"Anexos"

Dpto. Modulprüflabor Solar World Innovations GmbH

Pilar Beltrán Relancio

Contents

A. Devices List

A.1. By suppliers

METEOROLOGICAL STATION

AHLBORN

(These are the only ones who have all the equipment, but that they have a lot less offers from any device, hence the prices are expensive.)

ANEMOMETER

FVA 615-2

- Measuring range 0.5 ... 50 m / s -30 ... 70 ř C with heating

- Accuracy ś 0.5 m / s ś 3

- Resolution 0,1 m / s

- Adapter cable with ALMEMO connector, with cable supply for heating.

- Price 615 Euros

BAROMETER

FDA612SA

- Measuring range .. 1050 700th mbar

- Accuracy ś 0.5

- Operating range 10th .. 90

- Price 285 Euros

DATA LOGGER

MA85909

- 9 primary Data Logger

- 1 inputs 2 outputs

- 600mA AC Adapter

- Price 920 Euros

PSYCHORMETER

MT8xx6AGx

- Measuring range 0 .. 100

- Accuracy ś 2

- Price 655 Euros 500th ..

PLUVIOMETER

FRA916

- Measuring range 0.2 mm / pulse -30 ... 50 ř C with heating

- Price 590 Euros

STERNPYRANOMETER

FRA628S

- Measuring range 0 to 1500 W / mš

- Spectral Range 0.3 ... 3 microns

- Operative range - 40 ... 60 ř C

- Resolution 0.1 W / mš

- Price 1500 Euros

UV INSTRUMENTS

FLA613UVB

- Measurement range 0 .. 50?W / mš

- Spectral sensitivity 215th .. 315nm

- Price 820 Euros

FLA613UVA

- Measuring range 0...3mW/cm š

- Spectral sensitivity 310th .. 400nm

- Price 780 Euros

FLA613VLM

- Measuring range 0 .. 250W / mš

- Spectral sensitivity 360th .. 760nm

- Price 655 Euros

FLA613GS

- Measuring range 0 .. 1200 W / mš

- Spectral sensitivity 400th .. 1100nm

- Price 780 Euros

`WIND VANE`

FVA614

- Measuring range 0 .. 360

- Accuracy ś 5 ř

- Resolution 11.25 ř

- Power supply through ALMEMO

- Price 720 Euros

If all the devices of Ahlborn are:

Sum = 615 + 285 +920 + VAT +....= 6885. + Shipping

`PCE`

`ANEMOMETER`

P670M

- Measuring range 0 .. 40 m / s -200 ... 850 ř C

- Accuracy 0.5

- Resolution 0.01 m / s 0.1 ř C

- Without Software

- Price 288 euros

PCE008

- Measuring range 0.2 ... 25 m / s 0th .. min 50 ř C 0 .. 9900 mş /

- Accuracy ś 2

- Resolution 0.01 m / s 0.1 ř C

- RS232

- With software

- The data can also be sent in MS Excel.

- Additional Power Supply 9V / 230V.

- Price 260 Euros

PCE009

- Measuring range 0.2 ... 20 m / s 0 .. 50 ř C min 0 .. 36 000 mş /

- Accuracy ś 1

- Resolution 0,1 m / s 0.1 ř C

- RS232

- With software

- The data can also be sent in MS Excel.

- Additional Power Supply 9V / 230V.

- Additional RS232 to USB

- Price 290 Euros

BAROMETER

PCE THB 38

- Measuring range 10th .. 1100 mbar

- Accuracy ś 1.5 mbar ś 0.8 ř C

- Operating range 95

- Price 145 Euros

PCE DB2

- Measuring range 0 .. 2000 mbar

- Accuracy ś 1 mbar

- Without software (additional 49 Euros)

- Price 325 Euros

DATA LOGGER

DaqPro

- 8-Channel Data Logger

- Input range 0 .. 50 mV or 0 ... 10 V ś 0.5

- Electricitz network

- 512 values

- Price 325 Euros

PSYCHROMETER

PCE 313A

- Measuring range 0 .. 100

- Accuracy ś 2.5

- Resolution 0.1

- Software

- RS 232

- Battery 9V PP3

- Price 195 Euros

PCE HT11

- Measuring range 5th .. 98

- Accuracy ś 3

- Resolution 0.1

- Without Software

- 2 Battery 1.5V

- Price 134 Euros

PCE 3000

- Measuring range 0 .. 100

- Accuracy ś 2

- Resolution 0.01

- RS232

- 9V Battery pack

- Price 295 Euros

PCE 320

- Measuring range 0 .. 100

- Accuracy ś 3

- Resolution 0.1

- Without Software

- 4AA Battery 1.5V

- Price 135 Euros

`PLUVIOMETER` (you have none)

`STERNPYRANOMETER` (you have none)

`UV INSTRUMENTS`

PCE UV34

- Measuring range 0 .. 1.999 W / cm š or 1.99 to 19.99 mW / cm š

- Spectral sensitivity 290th .. 390 nm

- Price 110 Euros

PCE SPM1

- Measuring range 0 .. 2000 W / cm š

- Spectral sensitivity 400th .. 1100 nm

- RS232 USB to USB 32Euros

- Price 175 Euros

`WIND VANE` (you have none)

You have no pluviometer, Sternpyranometer and weather vane. With PCE can I make two list. The most expensive devices:

Total = 288 + 325 + 1408 + VAT + ..= 295th Returns

The cheapest devices:

Total = 260 + 145 + 775 + vat + ..= 195th Returns

And when I am most expensive devices of PCE plus I think the best of the rest Devices are (according to list prices Meteorologischestation equipment):

The most expensive (PCE) + pluviometer (Ahlborn) + Spyre. (K&Z) + Wfahne (M&T) = = 2662 + VAT + Shipping

Much cheaper than all of Ahlborn.

Or if I PCE + Ahlborn combine devices:

The most expensive PCE + Pluv, Stpyr, Wfahne (Ahlborn) = 4218 + VAT + Shipping

Even cheaper.

KIPP&ZONEN

ANEMOMETER

MW41

- Measuring range 0 .. 30 m / s -20 ... 60 ř C

- Non accuracy <0.9 m / s

- Resolution 0.07 m / s

- Price (you have sent me the price list, but this price is not there)

BAROMETER (you have none)

DATA LOGGER

COMBILOG

- Analog inputs 8 with 16 bits resolution, single ended ...

- Input ranges ś 6.25 mV ... ś 10 V

- Digital inputs 6 (entries and exits)

- Supply V DC 10th .18

- Internal memory 256 kB for data.

- Comunication RS 232 and RS 485

- Price 3250 Euros

LOGBOX SD

- Analog inputs with eight single ended, 3 diff. +2 Single ended

- Input ranges 2.5 ... 20 V mV

- Digital inputs 4 to 15 V maximum input

- Supply V DC 4 .20

- Internal memory 128 kB for data.

- Comunication RS 232 and RS 485

- Price 1050 Euros

`PSYCHOMETER` (you have none)

`PLUVIOMETER` (you have none)

`STERNPYRANOMETER`

CMP3

- Range 0 to 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <2.5

- Sensitivity fifth 20?W .. / mš

- Output voltage 20 mV for 1000W/qm 5th ..

- (Can not be used with the ventilation system CV2 or CV3)

- Price 650 Euros

CM4

- Measuring range 0 to 2500 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 150 ř C

- Non-linearity <3

- Sensitivity fifth 10?W .. / mš

- Price 1455 Euros

CMP6

- Measuring range 0 to 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <1

- Sensitivity fifth 20?W .. / mš

- Output voltage 16mV for 1000W/qm 5th ..

- Price 1195 Euros

CMP11

- Measuring range 0 ... 1000 W / mš
- Spectral Range 310 ... 2800 nm
- Operative range - 40 ... 80 ř C
- Non-linearity <0.2
- Sensitivity 7th .. 14?W / mš
- Output voltage 14mV for 1000W/qm 7th ..
- Price 1790 Euros

CMP21

- Measuring range 0 ... 1000 W / mš
- Spectral Range 310 ... 2800 nm
- Operative range - 40 ... 80 ř C
- Nonlinearity <0.2
- Sensitivity 7th .. 14?W / mš
- Output voltage 14mV for 1000W/qm 7th ..
- Price 2290 Euros

CMP22

- Measuring range 0 ... 1000 W / mš
- Spectral Range 200 ... 3600 nm
- Operative range - 40 ... 80 ř C
- Non-linearity <0.2
- Sensitivity 7th .. 14?W / mš
- Price 4980 Euros

`UV INSTRUMENTS`

CUV4

- Measuring range 0 .. 1000 W / mš
- Spectral sensitivity 300th and 315th .. .. 400 nm 385 nm
- Broadband UV Radiometer (UVA-UVB)

- Price 2165 Euros

`WIND VANE`

MW42

- Measuring range 0 .. 360

- Non-precision <15 ř

- Resolution 22.5 ř

- Together MW41 -20 ... 60 ř C

- Price

You do not have a price of anemometer and weather vane. You do not have a barometer, hygrometer, pluviometer. The most expensive devices:

Total = 6065 Euro + VAT + Shipping

The cheapest devices:

Total = 3865 Euro + VAT + Shipping

`Mencke&Tegtmeyer`

`ANEMOMETER`

Vwind - Measuring range 0.5 ... 50 m / s

- Accuracy ś 3

- Output 0-10 V or 4-20 mA

- Price 346 Euros

`BAROMETER` (you have none)

`DATA LOGGER`

IDL 100

- 8 input channels anal (16 bit resolution) 6 dig.

- Output Channels 6 dig.

- RS232 RS 485

- Price 1940 Euros

IDL 101

- Input Channels 6 anal (10/12 bit resolution) 16 dig.

- Output Channels dig 16th

- 512 kB RAM

- RS232 RS 485

- Price 990 Euros

ICL 100

- Connecting to ISK 100 ISM and IDL.

- Price 15 Euros

ICL 101

- Data cable RS 232 for reading out the data logger 100, or IDL IDL101.

- Price 20 Euros

ICL 103

- Maggots cable for data logger or IDL 100 IDL 101 (DB 25 connector)

- Price 20 Euros

ICL 104

- Almond cable for data logger IDL 100 or 101 (DB 9 connector).

- Price 20 Euros

ICP 100

- Configuration software for W98 to XP for ISM, IDL, IDM including Profibus-DP and modern connectivity.

- Price 195 Euros

`PSYCHROMETER` (you have none)

`PLUVIOMETER` (you have none)

`STERNPYRANOMETER` (you sell, which are of Kipp&zonen)

`UV INSTRUMENTS` (There is no price, no data)

STRAHLUNGSSENSORUVA

STRAHLUNGSSENSORUVB

`WIND VANE`

RWind

- Measuring range 0 .. 360

- Accuracy ś 5 ř

- Resolution 11.25 ř

- Output 0-10 V or 4-20 mA

- Price 404 Euros

You have no barometers, hygrometers, pluviometer, Spyranometer, UVMessg.

Sum = 346 + 990 + 404 = 1740 + VAT + Shipping

M&T + Ahlborn = 6370 + VAT + Shipping

CONTACT

PCE

-jge@warensortiment.de

Ahlborn

-info from Catalogs

Mencke&Tegtmeyer

-mencke@ib-mut.de; maris@ib-mut.de

Kipp&Zonen

-Silvia.Groeger @ rg-messtechnik.de

A.2. By devices

METEOROLOGICAL STATION

`ANEMOMETER`

PCE (site delivery in 1-3 days)

P670M

- Measuring range 0 .. 40 m / s -200 ... 850 ř C

- Accuracy 0.5

- Resolution 0.01 m / s 0.1 ř C

- Without Software

- Price 288 euros

PCE008

- Measuring range 0.2 ... 25 m / s 0th .. min 50 ř C 0 .. 9900 mş /

- Accuracy ś 2

- Resolution 0.01 m / s 0.1 ř C

- RS232

- With software

- The data can also be sent in MS Excel.

- Additional Power Supply 9V / 230V.

- Price 260 Euros

PCE009

- Measuring range 0.2 ... 20 m / s 0 .. 50 ř C min 0 .. 36 000 mş /

- Accuracy ś 1

- Resolution 0,1 m / s 0.1 ř C

- RS232

- With software

- The data can also be sent in MS Excel.

- Additional Power Supply 9V / 230V.

- Additional RS232 to USB

- Price 290 Euros

Ahlborn

FVA 615-2

- Measuring range 0.5 ... 50 m / s -30 ... 70 ř C with heating

- Accuracy ś 0.5 m / s ś 3

- Resolution 0,1 m / s

- Adapter cable with ALMEMO connector, with cable supply for heating.

- Price 615 Euros

Mencke&Tegtmeyer

Vwind

- Measuring range 0.5 ... 50 m / s

- Accuracy ś 3

- Output 0-10 V or 4-20 mA

- Price 346 Euros

Kipp&Zonen

MW41

- Measuring range 0 .. 30 m / s -20 ... 60 ř C

- Non accuracy <0.9 m / s

- Resolution 0.07 m / s

- Price (you have sent me the price list, but this price is not there)

Campbell

05 103, 05 106 & 05103-45

- Maximum wind speed measurements 60m / s gust survival 100 m / s

- Accuracy ś 0.3 m / s and ś 3 ř

- Combined sensor simplifies installation, with weather vane anemometer.

- Direct connection to Campbell Scientific dataloggers

- Other models, including the LM 05 103 (with 4-20 mA line driver) and 05 106 Marine version.

- Price 797.75 euros 05103-5

- Price 05106-5 991.5 e uros

A100R

- Max speed> 75m / s

- Accuracy ś 0.1 m / s

- Anemometer

- Price 626 Euros A100R

A100LK

- Max speed 77.22 m / s

- Accuracy 1

- Anemometer temperature range -30 to 70 ř C

- Price A100LK 659Euros

Sonic Wind

- Range. speed 0 to 60 m / s 0 ř - 359 ř

- Accuracy ś 2

- Anemometer temperature range -35 to 70 ř C

- Sonic Wind Price 641.85 Euros

Selection criterion

- Measuring range

- Accuracy

- Software

- Price

`BAROMETER`

PCE (site delivery in 1-3 days)

PCE THB 38

- Measuring range 10th .. 1100 mbar

- Accuracy ś 1.5 mbar ś 0.8 ř C

- Operating range 95

- Price 145 Euros

PCE DB2

- Measuring range 0 .. 2000 mbar

- Accuracy ś 1 mbar

- Without software (additional 49 Euros)

- Price 325 Euros

Ahlborn

FDA612SA

- Measuring range .. 1050 700th mbar

- Accuracy ś 0.5

- Operating range 10th .. 90

- Price 285 Euros

Mencke&Tegtmeyer, (you have none)

Kipp&Zonen (you have none)

Campbell

CS100

- 1100 mbar range 600th ..

- Accuracy ś 0.5 mb @ 20 ř C

- Manufactured by Setra

- Built-in power switch to power Minimise use

- Outputs a linear signal of 0 to 2.5 Vdc

- Price 481 Euros

RPT410F

- 1100 mbar range 600th ..

- Accuracy ś 0.3 mb @ 20 ř C

- Manufactured by Setra

- Built-in power switch to power Minimise use

- Outputs a a variable frecuency 600 to 1100 Hz

- Integral switching circuit Allows power consumption only during measurement cycle

- Manufactured by pressure

- Price 738 Euros

Selection criterion

- Measuring range

- Temperature (Here I think it is normal at 0 ř C)

- Price

`DATA LOGGER`

PCE (site delivery in 1-3 days)

DaqPro

- 8-Channel Data Logger

- Input range 0 .. 50 mV or 0 ... 10 V ś 0.5

- Electricitz network

- 512 values

- Price 325 Euros

Ahlborn

MA85909

- 9 primary Data Logger

- 1 inputs 2 outputs

- 600mA AC Adapter

- Price 920 Euros

Mencke&Tegtmeyer

IDL 100

- 8 input channels anal (16 bit resolution) 6 dig.

- Output Channels 6 dig.

- RS232 RS 485

- Price 1940 Euros

IDL 101

- Input Channels 6 anal (10/12 bit resolution) 16 dig.

- Output Channels dig 16th

- 512 kB RAM

- RS232 RS 485

- Price 990 Euros

ICL 100

- Connecting to ISK 100 ISM and IDL.

- Price 15 Euros

ICL 101

- Data cable RS 232 for reading out the data logger 100, or IDL IDL101.

- Price 20 Euros

ICL 103

- Maggots cable for data logger or IDL 100 IDL 101 (DB 25 connector)

- Price 20 Euros

ICL 104

- Almond cable for data logger IDL 100 or 101 (DB 9 connector).

- Price 20 Euros

ICP 100

- Configuration software for W98 to XP for ISM, IDL, IDM including Profibus-DP and modern connectivity.

- Price 195 Euros

Kipp&Zonen

COMBILOG

- Analog inputs 8 with 16 bits resolution, single ended ...

- Input ranges ś 6.25 mV ... ś 10 V

- Digital inputs 6 (entries and exits)

- Supply V DC 10th .18

- Internal memory 256 kB for data.

- Comunication RS 232 and RS 485

- Price 3250 Euros

LOGBOX SD

- Analog inputs with eight single ended, 3 diff. +2 Single ended

- Input ranges 2.5 ... 20 V mV

- Digital inputs 4 to 15 V maximum input

- Supply V DC 4 .20

- Internal memory 128 kB for data.

- Comunication RS 232 and RS 485

- Price 1050 Euros

Campbell (I think that you need a software (price list))

CR200Serie

- Low-cost data logger, available with built-in wideband wireless modem.
- Digital / pulse inputs 2 / 2
- Differential analog inputs (five single-ended)
- Typical Program expiring rate 1 Hz
- Price 408 Euros CR200
- Price 629 Euros CR216

CR800Serie

- Entry-level model
- Digital / pulse inputs 4 / 2
- Differential Analog Inputs 3
- Typical Program expiring rate 100 Hz
- Price 975 Euros CR800
- Price 1214 Euros CR850

CR1000

- Measuring and recording equipment with 4MB of memory standard.
- Digital / pulse inputs 8 / 2
- Differential analog inputs 8
- Typical Program expiring rate 100 Hz
- Price 1350 Euros CR1000

CR3000

- Stand-alone data logger with power supply, keyboard and display.
- Digital / pulse inputs 8 / 4
- Differential analog inputs 14
- Typical Program expiring rate 100 Hz
- Price 2673 Euros CR3000

- The logger needs a rear wall and electricity options.

CR5000

- Stand-alone data logger with power supply, keyboard and display.
- Digital / pulse inputs 8 / 2
- Differential analog inputs 20
- Typical Program expiring rate 2000 Hz
- Price CR 5000 4360 Euros
- The logger takes a back wall and electricity options.

CR9000X

- Modular
- Digital / pulse inputs 16/122
- Differential analog inputs typically 42 or more
- Typical Program expiring rate 4500 Hz
- Price 8906 Euros CR9000X

CR7

- Modular
- Digital / pulse inputs 8 / 4
- Differential analog inputs typically 42 or more
- Typical Program expiring rate 80 Hz
- Price on request

Selection criterion

- Aus-/Eingänge
- RS232 Connection
- Price

`PSYCHROMETER`

Mencke&Tegtmeyer, (you have none)

Kipp&Zonen (you have none)

PCE (site delivery in 1-3 days)

PCE 313A

- Measuring range 0 .. 100
- Accuracy ś 2.5
- Resolution 0.1
- Software
- RS 232
- Battery 9V PP3
- Price 195 Euros

PCE HT11

- Measuring range 5th .. 98
- Accuracy ś 3
- Resolution 0.1
- Without Software
- 2 Battery 1.5V
- Price 134 Euros

PCE 3000

- Measuring range 0 .. 100
- Accuracy ś 2
- Resolution 0.01
- RS232
- 9V Battery pack
- Price 295 Euros

PCE 320

- Measuring range 0 .. 100
- Accuracy ś 3
- Resolution 0.1
- Without Software
- 4AA Battery 1.5V

- Price 135 Euros

Ahlborn

MT8xx6AGx

- Measuring range 0 .. 100

- Accuracy ś 2

- Price 655 Euros 500th ..

Campbell

CS215

- Measuring range 0 .. 100

- Accuracy ś 2

- Price 259 Euros

HMP45C

- Measuring range 0.8 ... 100

- Accuracy ś 2

- Price 374 Euros

MP100A

- Measuring range 0 ... 100

- Accuracy ś 1

- Price 365 Euros

Selection criterion

- Measuring range, accuracy, resolution

- Software Included

- Price

`Pluviometer`

Mencke&Tegtmeyer, (you have none)

Kipp&Zonen (you have none)

PCE (you have none) Campbell

52203 (unheated)

- Measuring range 0.1 mm / pulse -20 ... 50 ř C with heating

- Accuracy 2

- Price 465 Euros

52202 (HEATED)

- Measuring range 0.1 mm / pulse -20 ... 50 ř C with heating

- Accuracy 2

- Price 621 Euros

Ahlborn

FRA916

- Measuring range 0.2 mm / pulse -30 ... 50 ř C with heating

- Price 590 Euros

`STERNPYRANOMETER`

PCE (you have none)

Ahlborn

FRA628S

- Measuring range 0 to 1500 W / mš

- Spectral Range 0.3 ... 3 microns

- Operative range - 40 ... 60 ř C

- Resolution 0.1 W / mš

- Price 1500 Euros

Kipp&Zonen

CMP3

- Measuring range 0 to 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <2.5

- Sensitivity fifth 20?W .. / mš

- Output voltage 20 mV for 1000W/qm 5th ..

- (Can not be used with the ventilation system CV2 or CV3)

- Price 650 Euros

CM4

- Measuring range 0 to 2500 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 150 ř C

- Non-linearity <3

- Sensitivity fifth 10?W .. / mš

- Price 1455 Euros

CMP6

- Measuring range 0 to 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <1

- Sensitivity fifth 20?W .. / mš

- Output voltage 16mV for 1000W/qm 5th ..

- Price 1195 Euros

CMP11

- Measuring range 0 ... 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <0.2

- Sensitivity 7th .. 14?W / mš

- Output voltage 14mV for 1000W/qm 7th ..

- Price 1790 Euros

CMP21

- Measuring range 0 ... 1000 W / mš

- Spectral Range 310 ... 2800 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <0.2

- Sensitivity 7th .. 14?W / mš

- Output voltage 14mV for 1000W/qm 7th ..

- Price 2290 Euros

CMP22

- Measuring range 0 ... 1000 W / mš

- Spectral Range 200 ... 3600 nm

- Operative range - 40 ... 80 ř C

- Non-linearity <0.2

- Sensitivity 7th .. 14?W / mš

- Price 4980 Euros

Mencke&Tegtmeyer, (you sell, which are of Kipp&Zonen)

Campbell data-not enough

CS300

- SI-photodiode pyranometer

- Price 194 Euros

- Mounting plate 36 Euros

SP1110

- SI-photodiode pyranometer

- Price 210 Euros

- Mounting plate 76 Euros

Selection criterion

- Measuring range, spectral range, range, operational, non-linearity, sensitivity

- Price

`UV INSTRUMENTS`

Ahlborn

FLA613UVB

- Measurement range 0 .. 50?W / mš

- Spectral sensitivity 215th .. 315nm

- Price 820 Euros

FLA613UVA

- Measuring range 0...3mW/cm š

- Spectral sensitivity 310th .. 400nm

- Price 780 Euros

FLA613VLM

- Measuring range 0 .. 250W / mš

- Spectral sensitivity 360th .. 760nm

- Price 655 Euros

FLA613GS

- Measuring range 0 .. 1200 W / mš

- Spectral sensitivity 400th .. 1100nm

- Price 780 Euros

Mencke&Tegtmeyer(There is no price, no data)

STRAHLUNGSSENSORUVA

STRAHLUNGSSENSORUVB

Kipp&Zonen

CUV4

- Measuring range 0 .. 1000 W / mš

- Spectral sensitivity 300th and 315th .. .. 400 nm 385 nm

- Broadband UV Radiometer (UVA-UVB)

- Price 2165 Euros

PCE

PCE UV34

- Measuring range 0 .. 1.999 W / cm š or 1.99 to 19.99 mW / cm š

- Spectral sensitivity 290th .. 390 nm

- Price 110 Euros

PCE SPM1

- Measuring range 0 .. 2000 W / cm š

- Spectral sensitivity 400th .. 1100 nm

- RS232 USB to USB 32Euros

- Price 175 Euros

Campbell (none)

Selection criterion

- UVA 315 - 400nm

- UVB 280-315 nm

- Measuring range

- Price

`WINDVANE`

Kipp&Zonen

MW42

- Measuring range 0 .. 360

- Non-precision <15 ř

- Resolution 22.5 ř

- Together MW41 -20 ... 60 ř C

- Price

Mencke&Tegtmeyer

RWind

- Measuring range 0 .. 360

- Accuracy ś 5 ř

- Resolution 11.25 ř

- Output 0-10 V or 4-20 mA

- Price 404 Euros

Ahlborn

FVA614

- Measuring range 0 .. 360

- Accuracy ś 5 ř

- Resolution 11.25 ř

- Power supply through ALMEMO

- Price 720 Euros

PCE (you have none)

Campbell (in anemometer)

Selection criterion

- Resolution, Accuracy

- Measuring range

- Price

I think the best opportunities, the red one.

CONTACT

PCE

-jge@warensortiment.de

Ahlborn

-info from Catalogs

Mencke&Tegtmeyer

-mencke@ib-mut.de; maris@ib-mut.de

Kipp&Zonen

-Silvia.Groeger @ rg-messtechnik.de

Campbell

-Kristin.Droege @ campbellsci.de

A.3. IV curve device

Schuster

info@schuster-elektronik.de

KML 605-5

-Sperrspanung individual resistors 00-10 kV.

- Current ranges 20 / 40 / 100 / 200 / 400 mA.

- Peak power and peak voltage (digital display)

- Price

- Delivery

KML 595

- blocking voltage 0-12 kV.

- Current ranges 1 / 2 / 5 / 10 / 20 / 50 / 100 / 200 / 500 / 1000 / 2000 mA.

- Peak power and peak voltage (digital display)

- Price

- Delivery

KML 585

- blocking voltage 00-10 kV.

- Current ranges 20 / 40 / 100 / 200 / 400 mA.

- Peak power and peak voltage

- Price

- Delivery

KML 621

- blocking voltage 0-12 kV.

- Current ranges 10 / 20 / 50 / 100 / 200 / 500 mA.

- Peak power and peak voltage

- Analog X-Y recorders.

- Price

- Delivery

PVE

md@pv-e.de

PVPM 2540C

- 250 V, 40 A

- Power cord set 600 V.

- Price 4950 Euros

- (There is one of PVPM2540C in laboratory)

PVPM 6020C

- 600 V, 20 A

- Power cord set 600 V.

- Price 5650 Euros

PVPM 1000C

- 1000 V, 20 A

- Power cord set 1000 V.

- Price 6400 Euros

PVPM 1000C40

- 1000 V, 40 A

- Power cord set 1000 V.

- (Peli 1650 Case with foam insert grid +358 Euros)

- Price 6800 Euros

- (This is outdoor mobile tester)

Mencke&Tegtmeyer

mencke@ib-mut.de

maris@ib-mut.de

PV-KLA

- Range 25 / 50 / 100 / 400 V 4 / 8 / 16 / 32 A

- RS 232 interface

- Incl. Operating software for MS Windows

- (With multiplexer, we can simultaneously measure up to 8 modules)

- Price 5300 Euros

KLA MULTIPLEXER

- Channel switch for extensions del PVKLA to the possibility of multiple (to measure up to 32) PV-PC-controlled elements.

- Price

UPDATE PVK

- Control Software for Windows PVK to 2000.

- Price 510 Euros.

METEO ADAPTER

- Meteorology adapter to extend the PVKLA to 7 additional meteorological logical channels (standard signals 0 .. 10 V or 0 .. 20 mA)

- Price 2450 Euros

METEOADAPTERS

- Meteorology adapter to extend the PVKLA to 7 additional meteorological logical channels (input signals as required)

- Price

CELL ADAPTER

- To extend the measure to the possibility PVKLA PV cells.

ET-PVMTS

Wolfgang.Elischer @ ettgmbh.de

PV module test bed ET-PVMTS

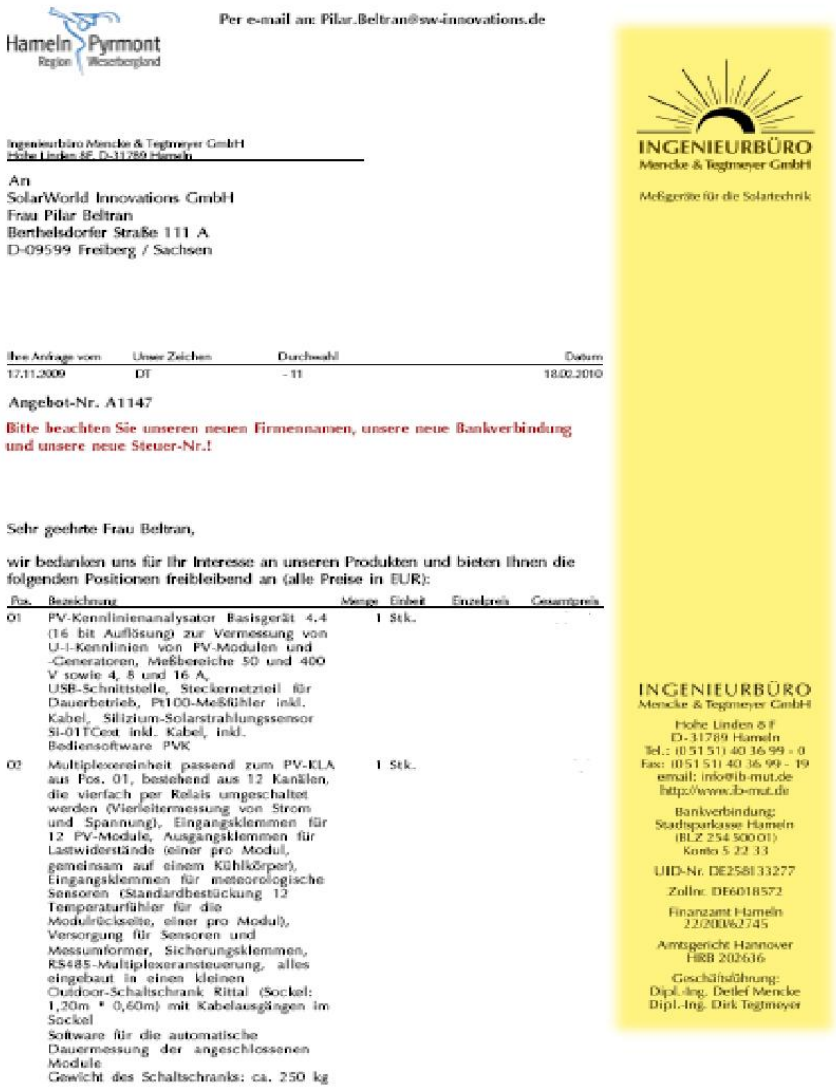
- Only valid to measure two Modullen. Too expensive for the table ...

- Price 9980 Euros

I think the best opportunities, the red one.

B. Agreed budget

After several offers and counteroffers with this and other companies, dated 02/18/2010 received from Mencke&Tegtmeyer which would be the purchase order. This offer allows us to buy as many devices we needed within budget constraints. All devices are bought except those in positions 14, 22, 23, 24. All prices have been erased for privacy reasons.



Hameln Pyrmont Region Weserbergland

Per e-mail an: Pilar.Beltran@sw-innovations.de

Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F, D-31789 Hameln

An
SolarWorld Innovations GmbH
Frau Pilar Beltran
Berthelsdorfer Straße 111 A
D-09599 Freiberg / Sachsen

Ihre Anfrage vom	Unser Zeichen	Durchwahl	Datum
17.11.2009	DT	- 11	18.02.2010

Angebot-Nr. A1147

Bitte beachten Sie unseren neuen Firmennamen, unsere neue Bankverbindung und unsere neue Steuer-Nr.!

Sehr geehrte Frau Beltran,

wir bedanken uns für Ihr Interesse an unseren Produkten und bieten Ihnen die folgenden Positionen freibleibend an (alle Preise in EUR):

Pos.	Bezeichnung	Menge	Einheit	Einzelpreis	Gesamtpreis
01	PV-Kennlinienanalysator Basisgerät 4.4 (16 bit Auflösung) zur Vermessung von U-I-Kennlinien von PV-Modulen und -Generatoren, Meßbereiche 50 und 400 V sowie 4, 8 und 16 A, USB-Schnittstelle, Steckernetzteil für Dauerbetrieb, Pt100-Meßfühler inkl. Kabel, Silizium-Solarstrahlungssensor SI-01TCext inkl. Kabel, inkl. Bediensoftware PVK	1	Stk.		
02	Multiplexereinheit passend zum PV-KLA aus Pos. 01, bestehend aus 12 Kanälen, die vierfach per Relais umgeschaltet werden (Vierleitermessung von Strom und Spannung), Eingangsklemmen für 12 PV-Module, Ausgangsklemmen für Lastwiderstände (einer pro Modul, gemeinsam auf einem Kühlkörper), Eingangsklemmen für meteorologische Sensoren (Standardbestückung 12 Temperaturfühler für die Modulrückseite, einer pro Modul), Versorgung für Sensoren und Messumformer, Sicherungsklemmen, RS485-Multiplexeransteuerung, alles eingebaut in einen kleinen Outdoor-Schaltschrank Rittal (Sockel: 1,20m * 0,60m) mit Kabelausgängen im Sockel Software für die automatische Dauermessung der angeschlossenen Module Gewicht des Schaltschranks: ca. 250 kg	1	Stk.		

INGENIEURBÜRO
Mencke & Tegtmeyer GmbH
Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer GmbH
Hohe Linden 8 F
D-31789 Hameln
Tel.: (0 51 51) 40 36 99 - 0
Fax: (0 51 51) 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de
Bankverbindung:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Konto 5 22 33
UID-Nr. DE258133277
Zollnr. DE6018572
Finanzamt Hameln
22/208/62745
Amtsgericht Hannover
HRB 202616
Geschäftsführung:
Dipl.-Ing. Detlef Mencke
Dipl.-Ing. Dirk Tegtmeyer

Ingenieurbüro Mencke & Tegtmeyer GmbH, Hohe Linden 8F, D-31789 Hameln 18.02.2010
Angebot-Nr. A1147 Seite 2

Pos.	Bezeichnung	Menge	Einheit	Einzelpreis	Gesamtpreis
03	Vwind, Windgeschwindigkeitsgeber Schalensternanemometer mit Heizung, Meßbereich 0,5 bis 50 m/s, Genauigkeit ± 3 % vom Meßwert oder 0,5 m/s, Ausgang 0 bis 10 V oder 4 bis 20 mA	1	Stk.		
04	Rwind, Windrichtungsgeber, Windfahne mit Heizung, Meßbereich 0° bis 360°, Auflösung 11,25°, Genauigkeit ± 5°, Ausgang 0 bis 10 V oder 4 bis 20 mA	1	Stk.		
05	TMeteo, Umgebungstemperatursensor, Pt100 zum Einbau in Wetter- und Strahlungsschutz	1	Stk.		
06	Wetter- und Strahlungsschutz für Umgebungstemperatursensor TMeteo	1	Stk.		
07	Feuchtefühler FA54 zur Messung der relativen Feuchte mit integriertem Meßumformer (Ausgang 0...10V / 4...20mA)	1	Stk.		
08	CMP 11 Pyranometer Kipp&Zonen, Secondary Standard nach WMO (Ausgangsspannung: 7...14 mV für 1000 W/m2)	3	Stk.		
09	Schattenring CM121 zur Abschattung eines Pyranometers vor direkter Sonnenstrahlung; genaue Messung der diffusen Himmelstrahlung	1	Stk.		
10	Halter für Windsensoren, aus Edelstahl zur Montage der Sensoren an Masten (ø 35 bis 50 mm) oder direkt an der Unterkonstruktion der Module	4	Stk.		
11	Halter für Pyranometer und Helligkeitsgeber aus Edelstahl zur Montage der Sensoren an Masten (ø 35 bis 50 mm) / 1 Halter - Compact + 1 Adapter für Strahlungs- oder Helligkeitsgeber	2	Stk.		
12	TModul10, Modultemperatursensor, selbstklebend, Polyamid Gehäuse für Meßumformer, 0..10 V bei -10..+120°C	10	Stk.		
13	Vwind-Si, Windgeschwindigkeitsgeber mit Reedrelais-Kontakt, 0,8..40 m/s, Digitalausgang typ. 2,3 Hz/(m/s), Meßfehler 0,5% FSR, inkl. 5 m Anschlußkabel	3	Stk.		
14	Aufpreis pro zusätzlichem Meteokanal, inkl. Messumformer UgT-MU (sofern notwendig) und Aufnahme in die Software	opt.	Stk.		
15	Traverse 2, 0,8 m Traverse aus Aluminium zur Montage von 2 meteorologischen Sensor	2	Stk.		
16	Mast zur Montage der Traversen und Halter für die Sensoren, Höhe 2m, mit Fußgestell 1m * 1m zur Beschwerung mit Betonplatten	1	Stk.		
17	Zusätzlicher PV-Kennlinienanalysator Basisgerät 4.4 (16 bit Auflösung) zur Vermessung von U-I-Kennlinien von PV-Modulen und -Generatoren, Meßbereiche 100 und 400 V sowie 4, 8 und 16 A, USB-Schnittstelle, Steckernetzteil für Dauerbetrieb, Pt100-Meßfühler inkl. Kabel, Silizium-Solarstrahlungssensor SI-01TCext inkl. Kabel, inkl. Bediensoftware PVK	1	Stk.		

Ingenieurbüro Mencke & Tegtmeyer GmbH, Hohe Linden 8F, D-31789 Hameln 18.02.2010
Angebot-Nr. A1147 Seite 3

Pos.	Bezeichnung	Menge	Einheit	Einzelpreis	Gesamtpreis
18	Installation und Auflegen der Sensorsignale für alle Meorologiekanäle und Temperatursensoren	28	Stk.		
19	Kabel AWG 26/4, 4x014qmm zum Anschluß der Sensoren (Verlegung in bauseits gestellten Kabelbühnen oder -kanälen)	500	m		
20	Verlegen der Kabel pro Sensorkanal	28	Stk.		
21	Si-13TC-K, Silizium-Solarstrahlungssensor, 0..10 V bei 0..1300 W/qm, temperaturkompensiert, inkl. Steckverbinder	4	Stk.		
22	Elektronische Last ESL-Solar 500 (Messbereiche: 100 V, 10 A), inkl. Software E-Pret Solar, Netzkabel, USB-Kabel, Bedienungsanleitung	opt.	Stk.		
23	Thies Clima Ultrasonic Anemometer 2D - compact, Ausgabe 0...10 V für Windgeschwindigkeit und -richtung, 16bit Auflösung, Gehäuse aus eloxiertem Aluminium, seewasserbeständig, 0...65 m/s, Genauigkeit ±2% rms, 0....360°, Genauigkeit ±2° bei >1m/s	opt.	Stk.		
24	Thies Clima Ultrasonic Anemometer 3D, Ausgabe 0...10V, Genauigkeit ±2% rms, ±2°, Gehäuse aus V4A-Edelstahl und seewasserfest eloxiertem Aluminium	opt.	Stk.		
25	Datenloggersystem auf Basis IDL100 und ISM111 als Ersatz für die in Pos. 14 angebotene Messdatenerfassung, Daten werden im IDL100 als einstellbare Mittelwerte gespeichert, gleichzeitig aber auch per Modulmessung vom LabView-Programm als Momentanwerte mitgeschrieben Bestehend aus: 1*IDL100, 4*ISM111, 1*Speichererweiterung 2MB SRAM	1	Stk.		
26	Kabelbahn Metall verzinkt, Höhe 60mm, Breite 100mm, Stärke 0,7mm, verzinkt nach DIN:PN-EN 10327:2005, mit Deckel nur auf der Hälfte der Länge und Deckelclips, Schnellverbindern aus V4A	210	m		
27	Montage der Kabelbahn an vorhandener Unterkonstruktion der PV-Module	210	m		

Summe (excl. MwSt.)
zzgl. 19% MwSt
Gesamtsumme

Mit freundlichen Grüßen

Dirk Tegtmeyer

Dieses Angebot ist gültig bis:31.03.2010

Lieferzeit: ca. 8 Wochen nach Erhalt der Anzahlung

Zahlungsbedingungen: 50% Anzahlung bei Auftrag, 50% nach Lieferung

Technische Änderungen im Sinne des Fortschritts vorbehalten. Garantie: 1 Jahr;
Gewährleistung: Instandsetzung defekter Geräte im Hause des Herstellers, Aufwendungen für Transport innerhalb der BRD werden von uns, Verpackung vom Käufer getragen.
Es gelten die Allgemeinen Lieferbedingungen für Erzeugnisse und Leistungen der Elektroindustrie (ZVEI, 6/2006).
Die Haftung von Mangelfolgeschäden wird ausgeschlossen. Gerichtsstand: Hameln

C. Technical data

C.1. Modules technical data sheets

Sunmodule®

SW 220/225/230/235 poly

German quality standards
Fully-automated production lines and seamless monitoring of the process and material ensure the quality that the company sets as its benchmark for its sites worldwide.

Outstanding products
In Photon test laboratory's endurance test SolarWorld's modules occupied first place in 2008 and in 2009 and were assessed by the ÖKO-TEST consumer magazine as "very good".

SolarWorld Plus-sorting
Plus-sorting guarantees the highest system efficiency. Only modules that achieve the designated nominal performance or greater in performance tests are despatched.

25 years linear performance guarantee*
SolarWorld guarantees a maximum degression in performance of 0.7% p.a. for more than 25 years – a clear additional benefit compared with the conventional two-stage industry guarantees. In addition, there is a product warranty that covers 5 years.

*in accordance with the applicable SolarWorld service certificate at purchase
www.solarworld-global.com/service-certificate

SOLARWORLD

We turn sunlight into power.

www.solarworld-global.com

Sunmodule®
SW 220/225/230/235 poly

PERFORMANCE UNDER STANDARD TEST CONDITIONS (STC)*

		SW 220	SW 225	SW 230	SW 235
Maximum power	P_{max}	220 Wp	225 Wp	230 Wp	235 Wp
Open circuit voltage	U_{oc}	36,6 V	36,8 V	36,9 V	37,0 V
Maximum power point voltage	U_{mpp}	29,2 V	29,5 V	29,8 V	30,0 V
Short circuit current	I_{sc}	8,08 A	8,17 A	8,25 A	8,35 A
Maximum power point current	I_{mpp}	7,54 A	7,63 A	7,72 A	7,85 A

*STC: 1000W/m², 25°C, AM 1.5

PERFORMANCE AT 800 W/m², NOCT, AM 1.5

		SW 220	SW 225	SW 230	SW 235
Maximum power	P_{max}	157,3 Wp	160,9 Wp	164,4 Wp	170,4 Wp
Open circuit voltage	U_{oc}	33,1 V	33,3 V	33,4 V	33,5 V
Maximum power point voltage	U_{mpp}	26,2 V	26,5 V	26,7 V	27,1 V
Short circuit current	I_{sc}	6,68 A	6,75 A	6,82 A	6,73 A
Maximum power point current	I_{mpp}	6,01 A	6,08 A	6,15 A	6,28 A

Minor reduction in efficiency under partial load conditions at 25°C: at 200W/m², 95% (+/-3%) of the STC efficiency (1000 W/m²) is achieved.

COMPONENT MATERIALS

Cells per module	60
Cell type	Poly crystalline
Cell dimensions	156 mm x 156 mm
Front	tempered glass (EN12150)

SYSTEM INTEGRATION PARAMETERS

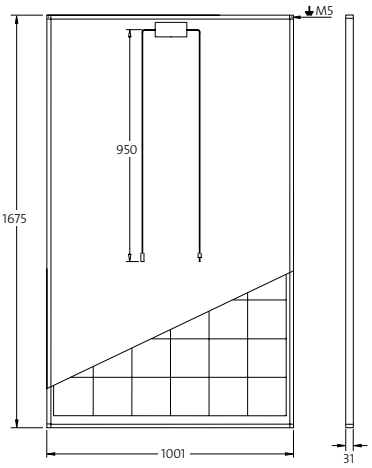
Maximum system voltage SC II	1000 V
Maximum reverse current	16 A
Increased snowload acc. to IEC 61215	5,4 kN/m²
Number of bypass diodes	3

THERMAL CHARACTERISTICS

NOCT	46 °C
TC I_{sc}	0,034 %/K
TC U_{oc}	-0,34 %/K
TC P_{mpp}	-0,48 %/K

ADDITIONAL DATA

Measuring tolerance	+/- 3 %
Junction box	IP65
Connector	MC4
SolarWorld Plus-Sorting[2]	$P_{Flash} \geq P_{max}$



· Qualified, IEC 61215
· Safety tested, IEC 61730
· Periodic Inspection

3)

1) Temporarily, also modules with a frame height of 34 mm can be supplied. Please inquire with your specialist dealer.
2) The output identified by SolarWorld (P_{Flash}) is always higher than the nominal output (P_{max}) of the module.
3) depending on the market
SolarWorld AG reserves the right to make specification changes without notice. This data sheet complies with the requirements of EN 50380.

23.07.2010 EN

Sunmodule®
SW 225/230/235/240 mono

German quality standards
Fully-automated production lines and seamless monitoring of the process and material ensure the quality that the company sets as its benchmark for its sites worldwide.

Outstanding products
In Photon test laboratory's endurance test SolarWorld's modules occupied first place in 2008 and in 2009.

SolarWorld Plus-sorting
Plus-sorting guarantees the highest system efficiency. Only modules that achieve the designated nominal performance or greater in performance tests are despatched.

25 years linear performance guarantee*
SolarWorld guarantees a maximum degression in performance of 0.7% p.a. for more than 25 years – a clear additional benefit compared with the conventional two-stage industry guarantees. In addition, there is a product warranty that covers 5 years.

*in accordance with the applicable SolarWorld service certificate at purchase
www.solarworld-global.com/service-certificate

SOLARWORLD

We turn sunlight into power.

www.solarworld-global.com

Sunmodule®
SW 225/230/235/240 mono

PERFORMANCE UNDER STANDARD TEST CONDITIONS (STC)*

		SW 225	SW 230	SW 235	SW 240
Maximum power	P_{max}	225 Wp	230 Wp	235 Wp	240 Wp
Open circuit voltage	U_{oc}	37,3 V	37,4 V	37,5 V	37,7 V
Maximum power point voltage	U_{mpp}	29,7 V	30,0 V	30,3 V	30,8 V
Short circuit current	I_{sc}	8,13 A	8,16 A	8,19 A	8,25 A
Maximum power point current	I_{mpp}	7,59 A	7,68 A	7,77 A	7,96 A

*STC: 1000W/m², 25°C, AM 1.5

PERFORMANCE AT 800 W/m², NOCT, AM 1.5

		SW 225	SW 230	SW 235	SW 240
Maximum power	P_{max}	162,9 Wp	166,9 Wp	170,9 Wp	179,1 Wp
Open circuit voltage	U_{oc}	33,7 V	33,9 V	34,0 V	34,4 V
Maximum power point voltage	U_{mpp}	26,8 V	27,2 V	27,5 V	28,1 V
Short circuit current	I_{sc}	6,56 A	6,58 A	6,60 A	6,65 A
Maximum power point current	I_{mpp}	6,07 A	6,14 A	6,22 A	6,37 A

Minor reduction in efficiency under partial load conditions at 25°C: at 200W/m², 95% (+/-3%) of the STC efficiency (1000 W/m²) is achieved.

COMPONENT MATERIALS

Cells per module	60
Cell type	Mono crystalline
Cell dimensions	156 mm x 156 mm
Front	tempered glass (EN12150)

SYSTEM INTEGRATION PARAMETERS

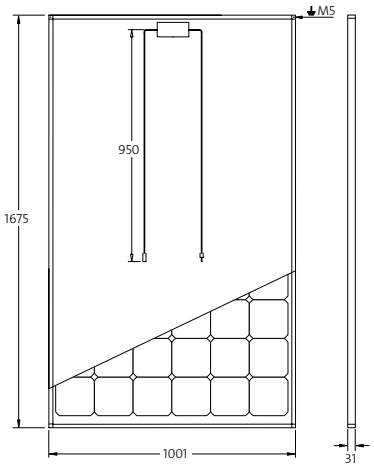
Maximum system voltage SC II	1000 V
Maximum reverse current	16 A
Increased snowload acc. to IEC 61215	5,4 kN/m²
Number of bypass diodes	3

THERMAL CHARACTERISTICS

NOCT	47 °C
TC I_{sc}	0,042 %/K
TC U_{oc}	-0,33 %/K
TC P_{mpp}	-0,45 %/K

ADDITIONAL DATA

Measuring tolerance	+/- 3 %
Junction box	IP65
Connector	MC4
SolarWorld Plus-Sorting[2]	$P_{Flash} \geq P_{max}$



· Qualified, IEC 61215
· Safety tested, IEC 61730
· Periodic Inspection

3)

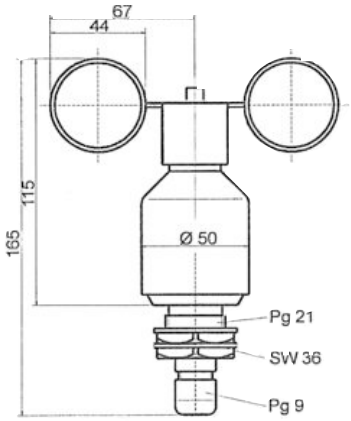
1) Temporarily, also modules with a frame height of 34 mm can be supplied. Please inquire with your specialist dealer.
2) The output identified by SolarWorld (P_{Flash}) is always higher than the nominal output (P_{max}) of the module.
3) depending on the market
SolarWorld AG reserves the right to make specification changes without notice. This data sheet complies with the requirements of EN 50380.

23.07.2010 EN

C.2. Devices technical data sheet

VWIND
Windspeed Sensor

INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik



Dimensional Drawing

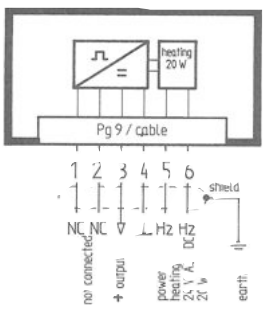
Technical Data

General

Measuring Range	0,5 50 m/s
Meas. Accuracy	± 3% of measuring value
Heating	24 V AC/DC max. 20 W
Ambient Temperature	30 °C + 70 °C
Mounting	onto masttube (boring thread Pg 21)
	or onto traverse, boring ∅ 29 mm

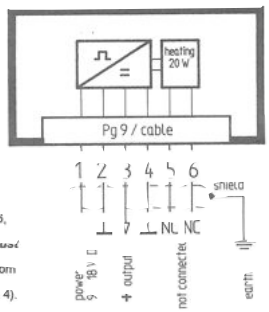
Resolution	< 0,1 m/s
Electr. Output	
Ord. No. **4.3519.00.040**	0 20 mA Load max. 500Ω (> 13 V DC Operating voltage)
4.3519.00.041	4 20 mA Load max 500Ω > 13 V DC Operating voltage)
4.3519.00.061	0 10 V Load resistance min. 1 kΩ
4.3519.00.067	0 2 V Load resistance min. 1 kΩ
4.3519.00.073	0 5 V Load resistance min. 1 kΩ
Operating voltage	9 18 V DC or 24 V AC/DC
	(13 8 V DC for 0 10 V Output)
	see Connecting diagrams
Current input	approx. 10 mA
Scanning	Light barrier slotted disc
Supply cable	LiYCY 6 x 0,25 mm² 12 m long
Weight	: 0,75 kg

Operation with Heating



In case of power supply for heating at pin 5 and 6, power supply voltage must be galvanic separated from signal voltage (pin 3 and 4).

Operation without Heating



INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Tel: +49 51 51 / 40 36 99 - 0
Fax: +49 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

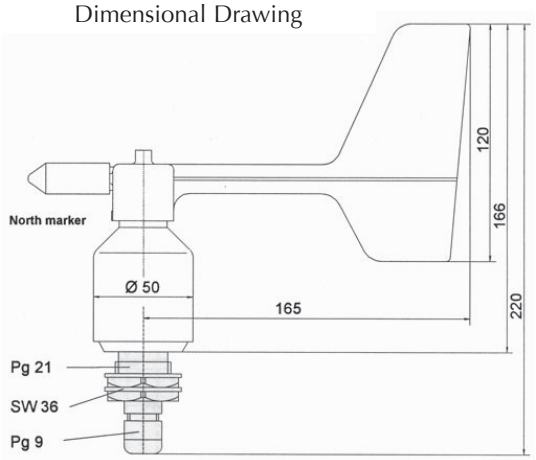
Banking Account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

USt-IdNr. DE160212135

RWIND
Winddirection Sensor

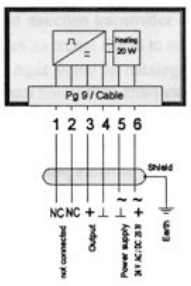
INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

Dimensional Drawing



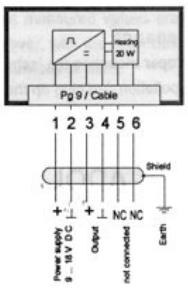
Resolution	: 11,25°	5 bit Gray code
Electr. Output		
Ord.-No. **4.3129.00.040**	: 0 - 20 mA	Load max. 500Ω (> 13 V DC Operating voltage)
4.3129.00.041	: 4 - 20 mA	Load max. 500Ω (> 13 V DC Operating voltage)
4.3129.00.061	: 0 - 10 V	Load resistance min. 1 kΩ
4.3129.00.067	: 0 - 2 V	Load resistance min. 1 kΩ
4.3129.00.073	: 0 - 5 V	Load resistance min. 1 kΩ
Operating voltage	: 9 - 18 V DC or 24 V AC/DC (13 - 18 V DC for 0 - 10 V Output) see connecting diagrams	
Current input	: approx. 10 mA	
Supply cable	: LiYCY 6 x 0,25 mm^2, 12 m long	
Weight	: 1,10 kg	

Operation with Heating



In case of power supply for heating at pin 5 and 6, power supply voltage ***must be*** galvanic separated from signal voltage (pin 3 and 4).

Operation without Heating



INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Tel: +49 51 51 / 40 36 99 - 0
Fax: +49 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking Account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

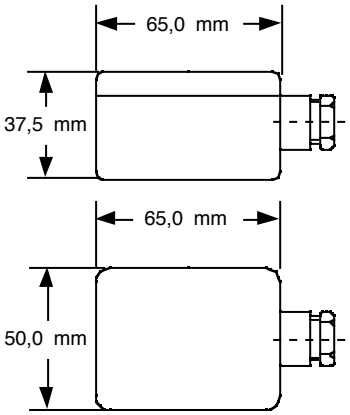
USt-IdNr. DE160212135

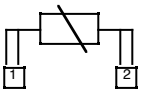
Tamb

(outdoor temperature sensor)

For measuring outside temperature, in cold stores and greenhouses, workshops and warehouses, stables etc.
In case of outdoor installation avoid direct rain and sun contact.

Housing:	polyamide, colour white, protection IP65
Ambient temp.:	-50 °C ... +90 °C
Accuracy:	Pt100,
Cable entry:	M16
Output signal:	4 mA ... 20 ma for -50 °C ... + 50 °C
Connection:	





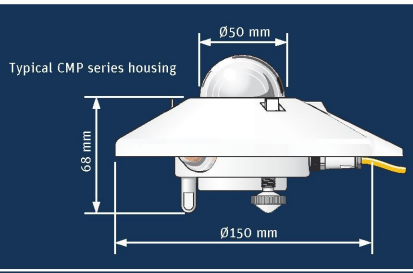
INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

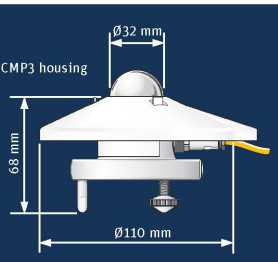
INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Germany
Tel: +49 (0) 5151 / 40 36 99 - 0
Fax: +49 (0) 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

USt-IdNr. DE160212135





Specifications	CMP 3	CMP 6	CMP 11	CMP 21	CMP 22
ISO CLASSIFICATION	Second Class	First Class	Secondary Standard	Secondary Standard	Secondary Standard
Response time (95 %)	18 s	18 s	5 s	5 s	5 s
Zero offsets (a) thermal radiation (200 W/m²) (b) temperature change (5 K/hr)	± 15 W/m² ± 5 W/m²	± 15 W/m² ± 4 W/m²	± 7 W/m² ± 2 W/m²	± 7 W/m² ± 2 W/m²	± 3 W/m² ± 1 W/m²
Non-stability (change/year)	± 1 %	± 1 %	± 0.5 %	± 0.5 %	± 0.5 %
Non-linearity (0 to 1000 W/m²)	± 2.5 %	± 1 %	± 0.2 %	± 0.2 %	± 0.2 %
Directional error (at 80 ° with 1000 W/m² beam)	± 20 W/m²	± 20 W/m²	± 10 W/m²	± 10 W/m²	± 5 W/m²
Temperature dependence of sensitivity	± 5 % (-10 °C to +40 °C)	± 4 % (-10 °C to +40 °C)	± 1 % (-10 °C to +40 °C)	± 1 % (-20 °C to +50 °C)	± 0.5 % (-20 °C to +50 °C)
Tilt error (at 1000 W/m²)	± 3 %	± 1 %	± 0.2 %	± 0.2 %	± 0.2 %
Other specifications					
	5 to 20 µV/W/m²	5 to 20 µV/W/m²	7 to 14 µV/W/m²	7 to 14 µV/W/m²	7 to 14 µV/W/m²
	20 to 200 Ω	20 to 200 Ω	10 to 100 Ω	10 to 100 Ω	10 to 100 Ω
	1 °	0.5 °	0.1 °	0.1 °	0.1 °
	-40 °C to +80 °C	-40 °C to +80 °C	-40 °C to +80 °C	-40 °C to +80 °C	-40 °C to +80 °C
	310 to 2800 nm	310 to 2800 nm	310 to 2800 nm	310 to 2800 nm	200 to 3600 nm
Typical signal output for atmospheric applications	0 to 15 mV	0 to 15 mV	0 to 15 mV	0 to 15 mV	0 to 15 mV
	2000 W/m²	2000 W/m²	4000 W/m²	4000 W/m²	4000 W/m²
	± 10 %	± 5 %	± 2 %	± 2 %	± 1 %
	Economical solution for routine measurements in weather stations, field testing	Good quality measurements for hydrology networks, greenhouse climate control	Meteorological networks, PV panel and thermal collector testing, materials testing	Meteorological networks, reference measurements in extreme climates, polar or arid	Scientific research requiring the highest level of measurement accuracy and reliability

CMP instruments have a standard cable length of 10 m. Optional cable lengths 25 m and 50 m

Standard 10k Thermistor or optional Pt-100 temperature sensor with CMP 21 and CMP 22

Individual directional response and temperature dependence test data with CMP 21 and CMP 22

Note: The performance specifications quoted are worst-case and/or maximum values

KIPP & ZONEN
SINCE 1830

Go to www.kippzonen.com for your local distributor

HEAD OFFICE

Kipp & Zonen B.V.
Delftechpark 36, 2628 XH Delft
P.O. Box 507, 2600 AM Delft
The Netherlands
T: +31 (0) 15 2755 210
F: +31 (0) 15 2620 351
info@kippzonen.com

4414470-0912

METEOROLOGICAL SENSORS

In conjunction with controlling of the performance of photovoltaic and solarthermal system the measurement of meteorological signal as global irradiance, wind velocity and direction as well as ambient temperature is essential. We offer all necessary sensors produced by well known manufacturers all at interesting prices.

General Informations

Global Irradiance

The best way for measuring the horizontal global irradiance is a thermo-electric pyranometer with a glass-dome. No silicon sensor meets the standards of the World Meteorological Organization (WMO). For precise measurement of the irradiance we therefore only use and offer pyranometers produced by Kipp & Zonen, Delft, the Netherlands.

Wind Velocity and Wind Direction

Our wind sensors have different output signals to meet the requirements of the connected data acquisition system. The case of the sensor consists of eloxadized aluminium and the cupstar and wind vane are made of plastic. Both sensors have an electronically regulated heating system to prevent the ball-bearings and the external rotation parts from freezing.

Ambient Temperature

We offer two different sensors for the measurement of ambient temperature. Both sensors use a Pt100 sensor according to the specification of EN 60751.

The sensor TMeteo consists of the essential Pt100 sensor and a weather and thermal radiation shield. Therefore this sensors can be installed nearly everywhere.

Our own sensor (Tamb) is realized with a Pt100 in a stainless steel tube mounted in a plastic case. Due to the very reduced thermal radiation shield the sensor must be protected from solar radiation to ensure correct measurement.

PV Module Temperature

The module temperature is measured by a special self adhesive Pt100 sensor on the rear side of the module. The sensor is delivered with a 2 m cable.

PYRANOMETER
Global Irradiance

Kipp & Zonen CM 3 / CM 6B / CM 11

- All pyranometers meet the specifications of the WMO and ISO 9060 respectively
- WMO class: Second class / First class / Secondary standard
- Measuring range: 0 to 1400 W/m²
- Sensitivity: 10 - 35 µV per W/m² / 9 - 15 µV per W/m² / 4 -6 µV per W/m²
- Non-linearity: 2,5 % / 1,5 % / 0,6 % (<1000 W/m²)
- Weight: 350 g / 850 g / 850 g

VWIND
Wind Velocity

Cupstar Anemometer

- Measuring range: 0,5 to 50 m/s
- Accuracy: ±3 % from rdg.
- Output signal: 10 (11) Imp. per rotation, 0(4) to 20 mA, 0 to 2(5)(10) V
- Heating: 24 V, max. 20 W
- Ambient temperature: -30 to +70 °C
- Weight: 400 g / 750 g

WRICHT
Wind Direction

Wind Vane

- Measuring range: 0 to 360 °
- Accuracy: 5 °
- Output signal: Graycode, 0(4) to 20 mA, 0 to 2(5)(10) V
- Heating: 24 V, max. 20 W
- Ambient temperature: -30 to +70 °C
- Weight: 1100 g

TMETEO
Ambient Temperature

Pt100 with Weather and Radiation Shield

- Measuring range: -30 to +70 °C /
- Accuracy: 1/3 DIN EN 60751 (corresponding to ±0,3 K)
- Output signal: Pt100, 4 to 20 mA, 0 to 1(10) V
- Voltage supply: 12 to 30 V_{DC}, 24 V_{DC} ±10 % for 0 to 10 V version
- Weather and radiation shield of UV resistant plastic with white lamella
- Dimensions Pt100 / shield: ø 20 mm x 138 mm / ø 120 mm x 140 mm
- Supply cable: 5 m long

TAMB
Ambient Temperature

Pt100 with Polycarbonat Case

- Measuring range: -70 to 200 °C
- Accuracy: 100 Ω ±0,1 Ω (corresponding ±0,3 °C)
- Dimensions Pt100 / case: ø 40 mm x 150 mm / 110 mm x 110 mm x 66 mm
- Weight: 250 g

TMODUL
PV Module Temperature

Self adhesive Pt100 for Surface Temperatures

- Measuring range: -50 to +150 °C /
- Accuracy: 100 Ω ±0,1 Ω (corresponding ±0,3 K)
- Dimensions Pt100: 40 mm x 13 mm
- Supply cable: 2 m lang, PTFE shielded

INTELLIGENT DATA ACQUISITION- AND LOGGING SYSTEM

Data Acquisition Using Fieldbus

The intelligent modules for data acquisition and logging are especially suitable where measured values and sensor signals occures decentral and the acquisition should be realized at one central place. There are various types of in- and output channels available. Therefore all measuring tasks can be realized with this system.

The high resolution (16 bit), multifunctional analog inputs, linearization and scaling direct within the modules, alarm report at cabling defects or surpassing of threshold values, RS 485 fieldbus with industrial used protocols, easy configuration and direct connection of numerical displays via fieldbus are some of the highlights of this system.

Description of Function

The intelligent data acquisition system consists of the dataloggers (IDL 100, ITL 104) and the registration modules (ISM, IDM and IMX). The system was especially designed for the acquistion of distributed signals, that should be logged at a central place.

The architecture of such a system as well as the configuration by using the Windows™-Software ICP 100 is quite simple. The multi-channel modules have multifunctional analog (16 bit resolution) and digital I/O ports for the measurement of current, voltage, resistance, bridges, frequency, counter etc.

The data acquisition is very exact because the pre-processing of the signals is realized within the registration modules. Simple linearizations or scalings can be managed within the decentral modules as well as more complicated mathematical linking and PID regulating. Due to this pre-processing the central datalogger is exonerated from data processing.

Additionally the registration modules detect different faults such as defect cabling, short-circuit and supervising of threshold values. Even in case of communication problems between the datalogger and the decentral modules the whole system can be shut down in a safe manner.

The data acquisition system can use different data transfer modes such as line, radio or GSM modems.

With the PCMCIA-Slot of the datalogger IDL 100 the memory capacity can be increased easily. By using specific firmware the IDL 100 is able to control numeric LED and LCD displays.

With the 32 bit-Software ICP 100 all parameters of each measuring channel is defined. Every datalogger and registration module gets its own configuration. Therefore the whole system can easily be adapted to the special requirements. During the start of the software the fieldbus is automatically scanned to find all connected modules.

INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Germany
Tel: +49 (0) 5151 / 40 36 99 - 0
Fax: +49 (0) 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

USt-IdNr. DE160212135

INTELLIGENT DATALOGGER

IDL 100
IDL 101

Technical Data

- Modular system with two-wire fieldbus (easy realization of wide distributed data acquisition systems
- Datalogger usable stand-alone or as central master
- RS 485 and RS 232 port
- Direct connection to a PC via RS 232 and to external numerical displays via RS 485
- Extension of memory and data transport by PCMCIA cards (2 MB SRAM, up to 10 MB Flash)
- Remote data transmission by using line, radio or GSM modem
- Comfortable, easy to use configuration software for Windows 95 and NT (also with remote data control), automatic busscan at software start
- Extension up to 60 channels
- Mathematical and logical linking of channels
- Very low power consumption (depending on the sampling rate)

Type	Input Channels	Output Channels	Ports	Specialities
IDL 100	8*analog[1, 3] 6*digital[2]	6*digital[2]	RS 232, RS 485	Complete datalogger PCMCIA slot
IDL 101	6*analog[1] 16*digital[2]	16*digital[2]	RS 232, RS 485	Complete datalogger 1*analog output
ISM 111	4*analog[1] 4*digital[2]	4*digital[2]	RS 485, Infrared	Additional mathematical functions
ISM 112	4*analog[1] 2*digital[2]	2*digital[2] 1*analog	RS 485, Infrared	Configurable analog output (14 bit resolution)

[1] Multifunctional [2] Usable as input or output [3] 10bit resolution

INTELLIGENT REGISTRATION MODULE

ISM 111, 112

Technical Data

- Modular system with two-wire fieldbus (easy realization of wide distributed acquisition systems
- Multifunctional analog inputs (current, voltage, resistance, temperature with Pt100 or thermocouple)
- Scaling and linearization direct within the module
- RS485 port, via converter IRK 100 also connectable to RS232 ports

PV-KLA

I-V Curve Analyser for Photovoltaics

The I-V curve analyser for photovoltaics PV-KLA can be used on many terms for I-V curve tracing of PV modules and generators in research and industry. Because of its flexibility it can be used for indoor tests as well as for outddor tests. Several devices are in action at german and international producer sites and research institutes. The PV-KLA is directly and comfortably controlled via the com port of a PC by using the software PVK.

Functional Description

For I-V curve tracing of PV generators the device uses a capacitive load. For a wide range of different PV generators the device uses only one capacitor without the need to exchange hardware. Optimization is done only by software. All 4 channels (voltage, current, irradiance and temperature) are sampled at the same time. The maximal sampling rate for one voltage-current-irradiance-temperature value is 66 kHz.

The portable, accu supplied device is suitable for indoor laboratory tests as well as for portable outdoor test. Large memory for every I-V curve, high accuracy and high sampling rates in combination with irradiance and temperature measurement in standard version enable the device to take curves with high quality. Meteorological sensors are sampled and displayed with 1 Hz. With optional specific adaptors the PV-KLA can be expanded for additional usage.

The meteorological adaptor multiplexes the temperature channel, so that 7 additional channels for the use of several meteorological sensors (Pt100 sensors, wind, pyranometer) are free.

The cell adaptor gives the opportunity to measure the I-V curve of single PV cells. It is the ideal partner for the steady-state solar simulator Mini-SuSi produced by our bureau.

The comfortable control software PVK is programmed object orientated (german and english versions are available) and shows in only one measuring window all relevant parameters for the I-V curve, incl. the curve as graphics.

It can be controlled by using the mouse or the keyboard. Long-time measurements with user defined time periods are possible as well as calculation of temperature coefficients for parameters like open-circuit voltage, short-circuit current or power within the maximum-power-point (MPP).

A multiplexer for up to 8 PV modules or generators is also available. It is controlled via the parallel port of the PC.

INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Germany
Tel: +49 (0) 5151 / 40 36 99 - 0
Fax: +49 (0) 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

USt-IdNr. DE160212135

PV-KLA
I-V Curve Analyser for Photovoltaics

Technical Data

- Basic accuracy: ±0,2 % fsr
  Meteorological sensors depend of used sensor type
- Voltage ranges: 25, 50, 100, 200 and 400 V
  (1 V option for cell adaptor)
- Current ranges: 4, 8, 16 and 32 A
  (1, 2, and 4 A option for cell adaptor)
- All current and voltage ranges can be combined to each other
- Irradiance range: 1300 W/m²
- Temperature range: 0 to 100 °C
- Maximal sampling rate for one current-voltage data pair: 66 ksamples/s
- Sampling of all channels by using 4 sample and hold devices
- Measuring time for I-V curve: 10 to 500 ms
  (about 6 I-V curves per minute possible)
- Maximal measuring quadruples for I-V curve: 8192
- User interface: controlled directly via PC und Software PVK
- Power supply: 6 V, 5*accu (baby type)
- PC port: RS232 (19,2 kBaud)
- Dimensions / weight (incl. acckus): 30 cm*20 cm*6 cm (DIN A4) / 1.800 g

SI-01TC
Silicon Irradiance Sensor

Technical Data

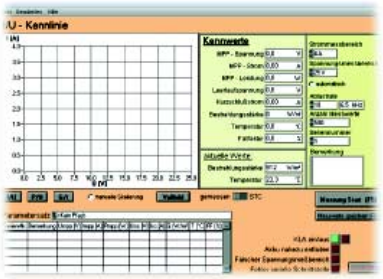
- Silicon irradiance sensor with active temperature compensation
- Calibration value for irradiance: 1 V for 1000 W/m²
- Monocristalline solar cell, embedded in ethylen-vinyl-acetat (EVA) between glass and Tedlar
- Eloxadized aluminium case

EXTEND OF SUPPLY

Standard version

Complete device PV-KLA with following accessories:

- Serial com port (RS232), incl. cable
- Charging unit for charging of accus
- Silicon irradiance sensor Si-01TC with active temperature compensation (incl. 2m sensor cable, uv- and heat resistant)



- Pt100 sensor for measurement of module temperature (incl. 2m cable)
- Control software PVK for Windows
- Manual

OPTIONS

- Meteorological adaptor for 7 additional meteorological sensors such as Pt100 sensors, pyranometer, silicon irradiance sensors, wind, and so on (temperature channel is multiplexed)
- Cell adaptor for I-V curve tracing of PV cells
- Module multiplexer for I-V curve tracing of up to 8 PV modules or generators without changing of cabling
- Personal software, output masks for data sheets on request
- Different current and voltage channels on request

Detailed infos, reference lists and demo versions of software PVK with complete manual on request or under www.ib-mut.de.

TModul10

(surface temperature sensor)

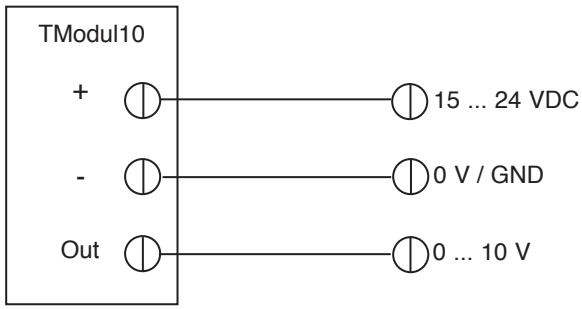
INGENIEURBÜRO
Mencke & Tegtmeyer
Meßgeräte für die Solartechnik

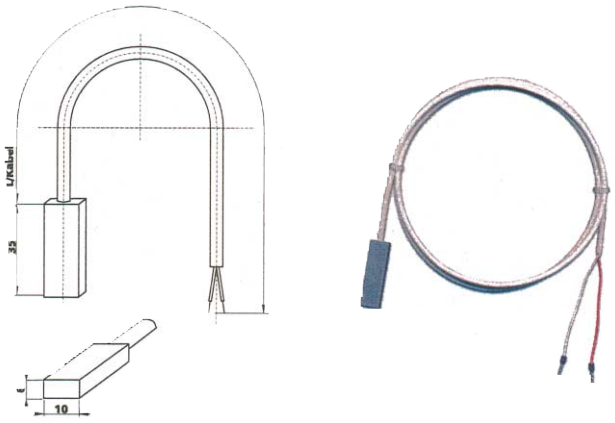
For measuring surface temperatures, e.g. PV module rear side

Cable connection: 1,0 m, fixed connection lead

Pockets:	aluminium, H x W x L: 6 mm x 10 mm x 35mm
Housing:	polyamide, colour white, protection IP65
Ambient temp.:	-50 °C ... +90 °C
Accuracy:	typ. ±1% of measuring range
Cable entry:	M16
Output signal:	0 to 10 V for -10 °C ... + 120 °C

Connection:





INGENIEURBÜRO
Mencke & Tegtmeyer

Hohe Linden 8F
D-31789 Hameln
Germany
Tel: +49 (0) 5151 / 40 36 99 - 0
Fax: +49 (0) 51 51 / 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 10 00 38 38

USt-IdNr. DE160212135

SILICON IRRADIANCE SENSOR

Measurement of Solar Irradiance

Since 1994 we have been developing and producing different types of silicon irradiance sensors. Until the year 2008 more than 9000 sensors were sold worldwide. Our silicon sensor is an affordable solution for measurement of solar irradiance. The powder-coated aluminium case in conjunction with the solar cell laminated between glass and Tedlar foil builds a very reliable and rugged sensor.

General Information

Mode of Operation

A silicon solar cell can be used as an irradiance sensor, because the short-circuit current is proportional to irradiance. Our sensors are build out of a monocrystalline solar cell connected to a shunt. Due to the low resistance of the shunt the cell operates next to short-circuit.

The temperature coefficient of the short-circuit current creates a small error. Therefore all of our silicon sensors with the extension „TC“ have an active temperature compensation, which reduces this error by factor 20. The compensation is realized by using a specific temperature sensor laminated to the rear side of the solar cell. The electronic circuit integrated for this compensation has a very low power consumption. The current consumption of the Si-01TC out of the internal Lithium battery is only about 15 µA.

Our silicon sensors are manufactured in different types with an internal or external power supply, with different output signals and with an optional sensor output for the cell temperature.

All sensors are calibrated in simulated sunlight against a reference cell of the same type. The reference cell is periodically calibrated against a reference cell calibrated by Fraunhofer ISE, Freiburg.

Mechanical Construction

The solar cell is embedded in Ethylen-Vinyl-Acetat (EVA) between glass and Tedlar. The laminated cell is integrated into a case of powder-coated aluminium. Therefore the sensor construction is comparable to that of a standard PV module. The electrical connection is realized by a 3 m cable or a waterproof (IP67) connector.

Optional Temperature Measurement

Additionally to the irradiance measurement our silicon sensors with the extension “-T” are able to measure the temperature of the solar cell. Therefore a temperature sensor is mounted to the rear side of the cell to detect the correct cell temperature.

INGENIEURBÜRO
Mencke & Tegtmeyer GmbH

Meßgeräte für die Solartechnik

INGENIEURBÜRO
Mencke & Tegtmeyer GmbH

Hohe Linden 8F
D-31789 Hameln
Tel: +49 (51 51) 40 36 99 - 0
Fax: +49 (51 51) 40 36 99 - 19
email: info@ib-mut.de
http://www.ib-mut.de

Banking Account:
Stadtsparkasse Hameln
(BLZ 254 500 01)
Kto.-Nr. 5 22 33

VAT-Nr. DE258133277

Customsno. DE6018572

Finanzamt Hameln
22/200/62745

Amtsgericht Hannover
HRB 202636

Management:
Dipl.-Ing. Detlef Mencke
Dipl.-Ing. Dirk Tegtmeyer

SILICON IRRADIANCE SENSOR

Technical Data

SI-SENSOR
General information

- Solar cell: Monocrystalline Silicon (50 mm x 33 mm)
- Current shunt: 0,1 Ω (TC = 20 ppm / K) with 1 V and 20 mA output
  0,12 Ω (TC = 20 ppm / K) with10 V output
- Operating temperature: –20 °C to 70 °C
- Electrical connection done via shielded cable, length 3 m
- Case, protection mode: Powder-coated aluminum, IP 65
- Dimension, weight: 145mm x 86mm x 39mm, appr. 340 g

ACCURACY
Irradiance

- Error with temperature compensation compared to pyranometer within the operating range of -20 °C to 70 °C and vertically beam of irradiance: ± 5 %
- Non-linearity of the electronic circuit:± 0,3% from reading for 50 to 1300 W/m²

Temperature

- Accuracy at 25 °C: ± 1,5 °C
- Non-linearity: ± 0,5 °C
- Error (over operating temperature range of -20…-70°C): ± 2,0 °C

Customs Numbers

for all silicon irradiance sensors 85 41 40 90

Sensor Types:

Typ	Irradiance			Cell Temperature
	Power Supply	Temperature Compensation	Output Signal	Output Signal
Si-01TC	Internal Li-Battery	Yes	0 to 1 V per 0 to 1000 W/m²	./.
Si-01TCext-K	5 to 28 V_{DC}	Yes	0 to 1 V per 0 to 1000 W/m²	./.
Si01TC-T-K	5 to 28 V_{DC}	Yes	0 to 1 V per 0 to 1000 W/m²	1,235 V + T[°C]*10mV/°C
Si-02-K	./.	No	appr. 60 mV per 1000 W/m²	./.
Si-02-Pt100-K, Si-02-T-K Si-02-Pt1000-K	./.	No	appr. 60 mV per 1000 W/m²	KTY 81-210 Pt100 / Pt1000
Si-10TC-K	12 to 28 V_{DC}	Yes	0 to 10 V per 0 to 1000 W/m²	./.
Si-420TC-K	12 to 28 V_{DC}	Yes	4 to 20 mA per 0 to 1200 W/m²	./.
Si-420TC-T-K	12 to 28 V_{DC}	Yes	4 to 20 mA per 0 to 1200 W/m²	(13,88+0,08/°C*T[°C])mA
Si-13TC-K	12 to 28 V_{DC}	Yes	0 to 10 V 0 to 1300 W/m²	./.
Si-13TC-T-K	12 to 28 V_{DC}	Yes	0 to 10 V 0 to 1300 W/m²	2,268V+86,9mV/°C*T

EXTEND OF SUPPLY

- Silicon sensor with shielded cable, 0,14 mm², UV- and temperature resistant

Options

- Ready-made cable of the requested length
- Version with waterproof connector (Si-01TC always with connector)

SILICON IRRADIANCE SENSOR

Electrical connection and Pin numbers

ELECTRICAL CONNECTION

Colors of cable (all versions „-K“)

Irradiance:	Orange
Power Supply (Plus):	Red
Power Supply / Signal (Minus):	Black
Temperature:	Brown; only versions with „-T“
Shield:	Black (bigger diameter)

ELECTRICAL CONNECTION OPTIONAL VERSION WITH CONNECTOR

The electrical connection of the Si sensor is realized with the inbuilt connector and the suitable plug.

Technical Data of the plug

- Cable dimensions (best / max): 0,14 mm^2 / 0,25 mm^2 (AWG26 / AWG24)
- Diameter for cable: 3,5 … 5 mm
- Protection mode: IP67 in conjunction with the suitable connector

The connection of the different Silicon irradiance sensors are listed in the following table. The pin numbers are printed at the inside of the plug. Please take care of the mounting of the plugs as shown at the end of the next page. Only if these mounting steps are done the plug reaches IP67 when connected.

Typ	Pin Numbers of the plug			
	Pin 1	Pin 2	Pin 3	Pin 4
Si-01TC	Plus Signal Irradiance	Minus Signal Irradiance	not available	not available
Si-01TCext	Plus Signal Irradiance	Minus Signal* Irradiance	Plus Signal Power Supply	not available
Si01TC-T	Plus Signal Temperature	Plus Signal Irradiance	Plus Signal Power Supply	Minus Signal* Irradiance
Si-02	Plus Signal Irradiance	Minus Signal Irradiance	not available	not available
Si-02-Pt100, Si-02-T Si-02-Pt1000	Plus Signal Irradiance	Minus Signal Irradiance	Pt100, KTY, Pt1000	Pt100, KTY, Pt1000
Si-10TC	Plus Signal Irradiance	Minus Signal* Irradiance	Plus Signal Power Supply	not available
Si-420TC	Plus Signal Irradiance	Minus Signal* Irradiance	Plus Signal Power Supply	not available
Si-420TC-T	Plus Signal Temperature	Plus Signal Irradiance	Plus Signal Power Supply	Minus Signal* Irradiance
Si-13TC	Plus Signal Irradiance	Minus Signal* Irradiance	Plus Signal Power Supply	not available
Si-13TC-T	Plus Signal Temperature	Plus Signal Irradiance	Plus Signal Power Supply	Minus Signal* Irradiance

* Minus signals of all sensors are identical to supply ground

SILICON IRRADIANCE SENSOR

Handling and Installation

SPECIALS

- The Si-01TC can be used for direct reading of irradiance via a digital voltmeter because of its internal Lithium battery and the calibration factor of 1 V per 1000 W/m².
- **Attention:** Because of the internal Lithium battery of the Si-01TC please use only voltmeters with a high input resistance (more than 1 MΩ).
- **Attention:** The maximum load at the output signals with the Si-420TC(-T) is 400 Ω.
- **Attention:** Horizontally mounting leads to increased reflexion on the glass surface and therefore to higher measurement errors.

MECHANICAL INSTALLATION

The Si sensor has two tounges with each three M6 drills. The installation at a suitable construction must use at least one M6 screw with washers at each mounting bracket.

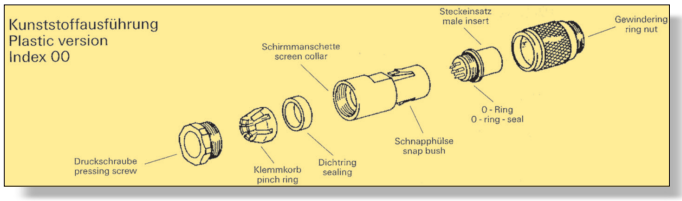
During installation the pressure compensation element near the electrical connection must not be damaged. If the cap of the element has loosened, it can be snapped on again.

HANDLING CASE

The Si sensor can be cleaned using a smooth cotton cloth, water and a mild cleaning fluid.

An opening of the sensor case by the user or installation staff is not necessary. If the case is opened, we can not guarantee the seal of the case anymore.

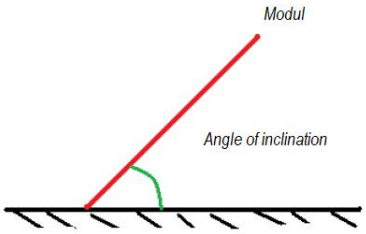
MOUNTING OF PLUG

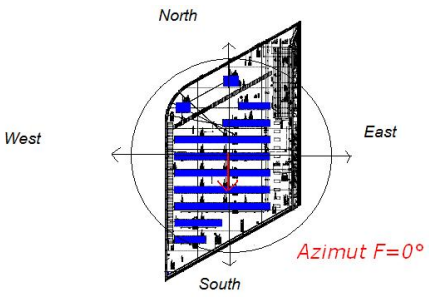


D. Shadows and sunhours per day calculation

D.1. Shadows calculation

This appendix explains the process followed for the calculation of shadows on the installation of the roof of the DS1400 due to the higher part of the building, the railing and the Meteorological Station.





1. For the measurement of the shadow, first of all must take a point from the module and some points of the object that causes the shadow.

$$F = F_m + 90 + tg^{-1}\left(\frac{b}{a}\right)$$

F is azimuth.

b, c and d is the distance between points

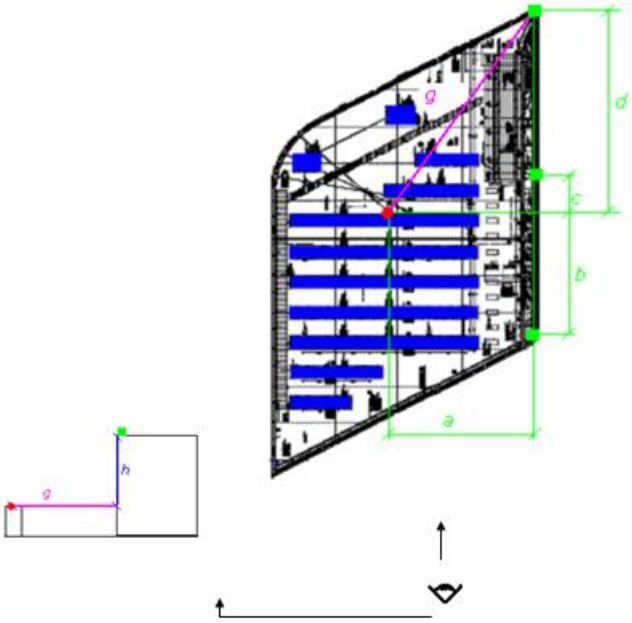
h height difference

$$\alpha = tg^{-1}\left(\frac{h}{g}\right)$$

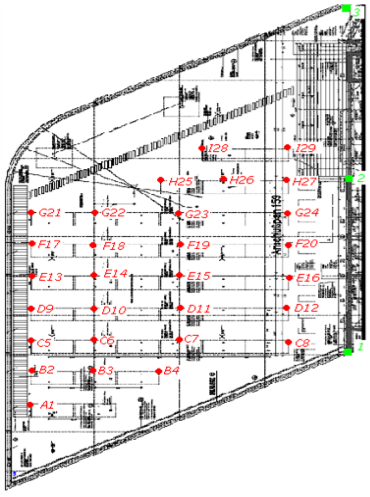
α solar altitude.

g distance between points.

We can see these parameters in Figure 1



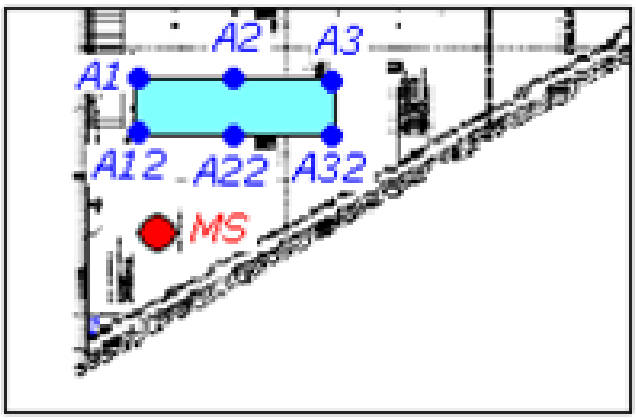
In the next Figure D.1, we can see the points to calculate the influence of the higher part of the building. Red points are the points in the installation and green points are the points in the building. The solar altitude is calculated for an angle of 15 °and 30 °. The angle of inclination 15 °limit us more than 30 °.



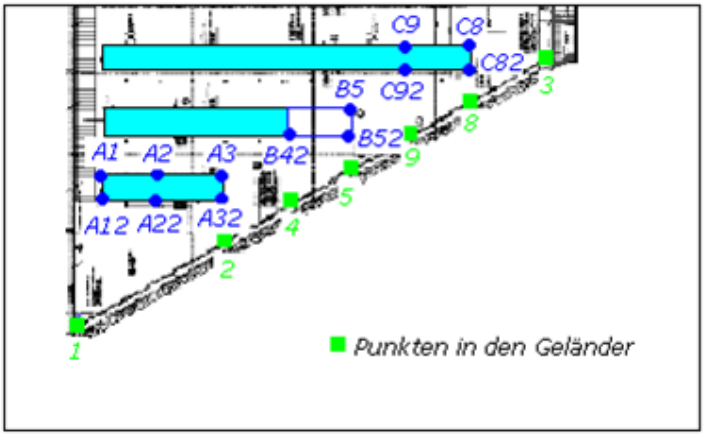
Here, Figure D.1 is a brief overview of the Excel file in which all calculations are performed. This overview also exists for the influence of the railing and the influence of the meteorological station. For these calculations other points were adopted.

PUNKT	a	b1	c2	d3	e30°	e15°	F1	F2	F3
A1	30,56	7,04	31,84	56,64	12,5625	12,9664781	102,972706	136,175133	151,651007
B2	30,56	2,56	27,36	52,16	12,5625	12,9664781	94,7884666	131,837702	149,634407
B3	24,48	2,56	27,36	52,16	12,5625	12,9664781	95,970016	138,17983	154,858154
B4	18,24	2,56	27,36	52,16	12,5625	12,9664781	97,9893268	146,309932	160,725613
PUNKT	g1	g2	g3	alpha1 30°	alpha2 30°	alpha3 30°	alpha1 15°	alpha2 15°	alpha3 15°
A1	31,3604082	44,1327452	64,3583965	21,8302842	15,8891402	11,0450259	22,4634551	16,3731198	11,3910591
B2	30,6670377	41,0180838	60,4531157	22,2760563	17,0281164	11,7393104	22,9193619	17,5425859	12,1058472
B3	24,6134922	36,7129405	57,6188858	27,0393482	18,890013	12,299574	27,7804261	19,4524818	12,682493
B4	18,418773	32,8826276	55,2572457	34,2958919	20,9088887	12,8082444	35,1448422	21,5206116	13,2059048

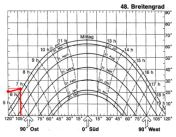
In Figure D.1, we can observe the points that we will use to calculate the influence of Meteorological Station. We suppose a MS height of 3 m.

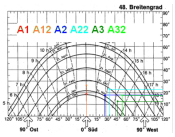


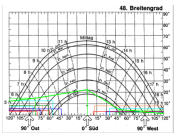
In the next Figure D.1, the points of the lines A, B and C, which can be affected by the railing, supposed height 1,10 m. Green points are railing points and blue points are points in moduls.

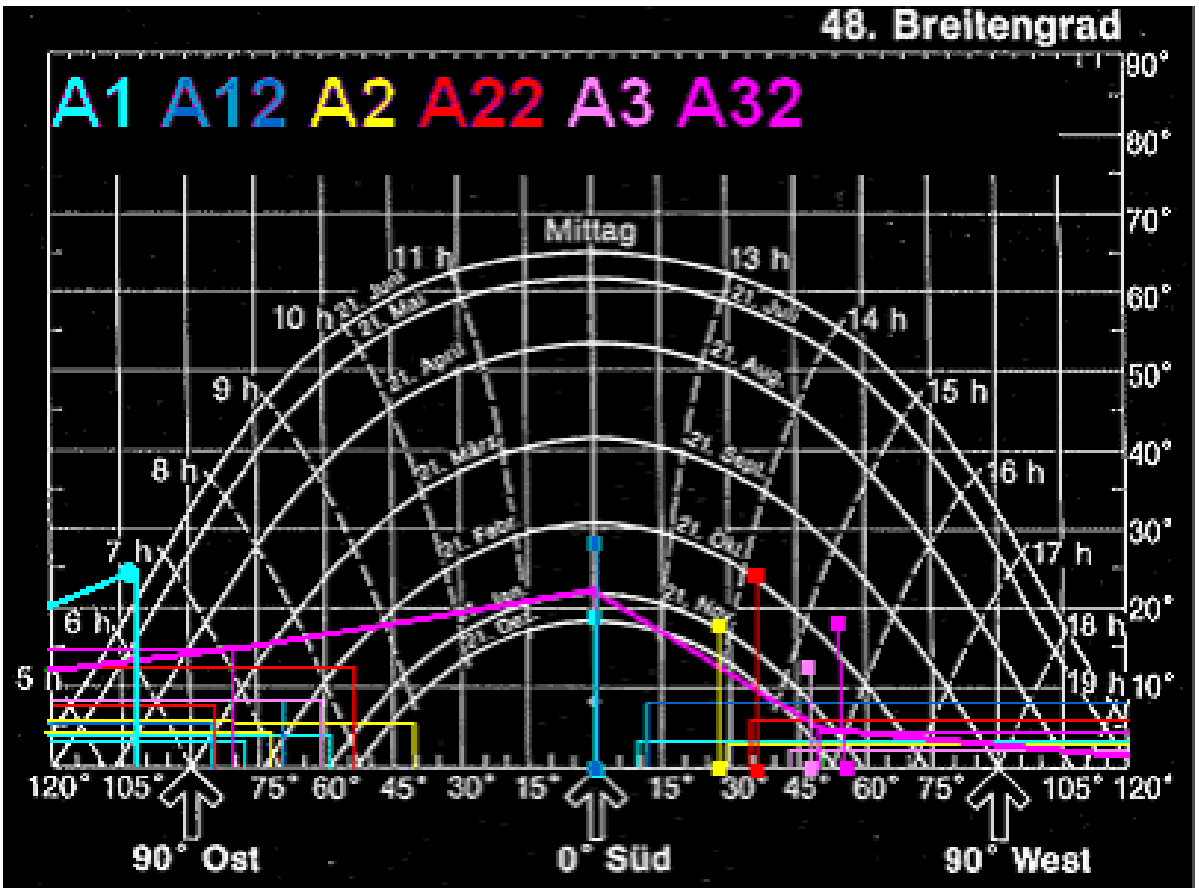


These are the representations D.1 of the Excel data transfered on a Sun angle chart, on the left we can find the building shadows, in the middle MS shadows, and on the right the railing shadows. They are represented only to line A. In black D.1 is the summary of all influences in line A.









We try to analyze point A1, Figure D.1

Building:

- 21 June: shadow until 6:30 AM.

- 21 May and July 21: Shadow up to 6:00 AM.

- Since that moment, the shadow decreases every day. For example: 21 April and 21 August: Shadow until 5:30 AM.

Meteorological Station:

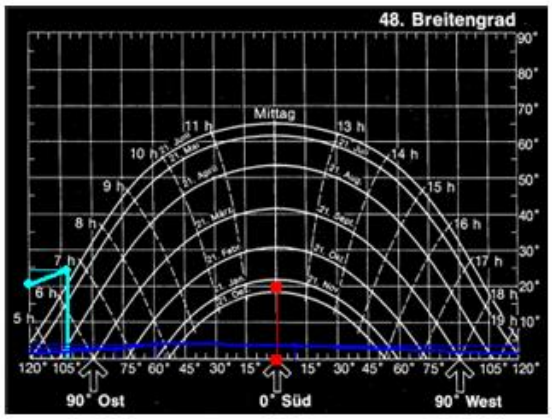
- Only temporary and specific effect.

- For example: between 18 December and 25 December, spot shadow at 12:00 AM.

Railing:

-Due to the height of the railing, the influence is lower.

-For example, earlier than 9 AM(start time of measurements), later than 16 PM(in winter, the sky is already dark).



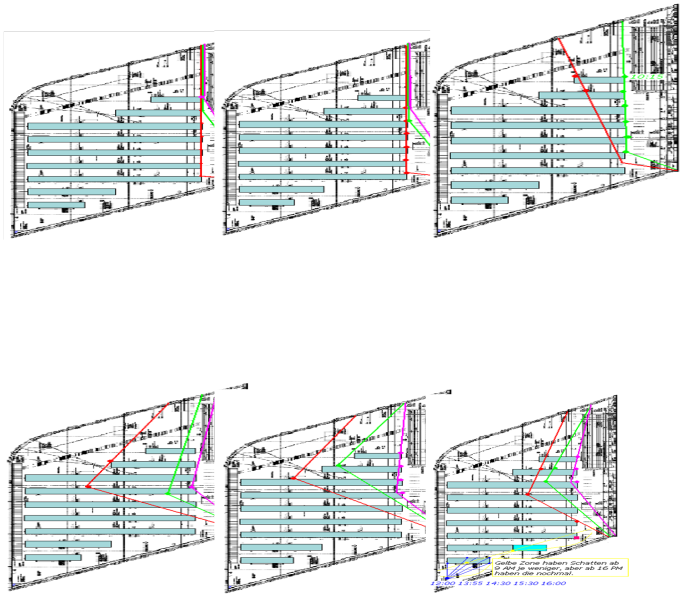
Punkt	A1				
Building	5-7	7-10	10-13	13-16	16-19
Mai Juni Juni Juli	Until 6.30	X	X	X	X
April Mai Juli August	Until 6	X	X	X	X
März April Aug. Sept.	Until 5.30	X	X	X	X
Feb. März Sept. Okt.	X	X	X	X	X
Jan. Feb. Okt. Nov.	X	X	X	X	X
Nov. Dez. Dez. Jan.	X	X	X	X	X

PUNKT	F1	alpha1 15°
A1	0	19,2239834

PUNKT	F1	F2	F3	alpha1 15°	alpha2 15°	alpha3 15°
A1	102,972706	136,175133	151,651007	22,4634551	16,3731198	11,3910591

	9,75242494	-59,8862668	-78,8351198	3,67022037	4,12192565	3,07792951

Here is a sequence D.1, in which we observe how the shadow moves on installation depending on the months of the year



D.2. Sunhours per day

It is shown here D.2, the sunrise and the sunset of the first and last days of every month.

SA Sunrise.

SU Sunset.

MW mean value of the month.

SSpT sun hours per day.

	Januar		Februar		März		April		Mai		Juni		Juli		August		September		Oktober		November		Dezember	
Tage	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU	SA	SU
1	07:58	16:39	07:45	16:56	06:54	17:45	06:46	19:37	05:43	20:26	04:59	21:10	04:56	21:24	05:30	20:55	06:18	19:55	07:05	18:48	06:56	16:44	07:45	16:05
28			06:56	17:44																				
30							05:45	20:24			04:55	21:24					07:03	18:50			07:44	16:05		
31	07:47	16:54			05:48	18:35			05:00	21:09			05:29	20:56	06:16	19:57			06:54	16:45			08:09	16:09
MW	08:01	16:31	07:21	17:19	06:21	18:10	06:14	20:00	05:19	20:48	04:53	21:20	05:10	21:13	05:53	20:27	06:40	19:22	07:28	18:16	07:18	16:21	08:00	16:02
SSpT	08:30		09:58		11:49		13:46		15:29		16:27		16:03		14:34		12:42		10:48		09:03		08:02	

This will help us to determine the best measure time, if we have only six hours to measure. This D.2 is an estimate of the 6 best times to take measurements, allowing us to obtain better values obtained during the six hours where the sun is at its highest position.

F.e: June: measurements 10:10 to 16:10

F.e: January: Measurements 9:07 to 15:07

EM First measurement of day.

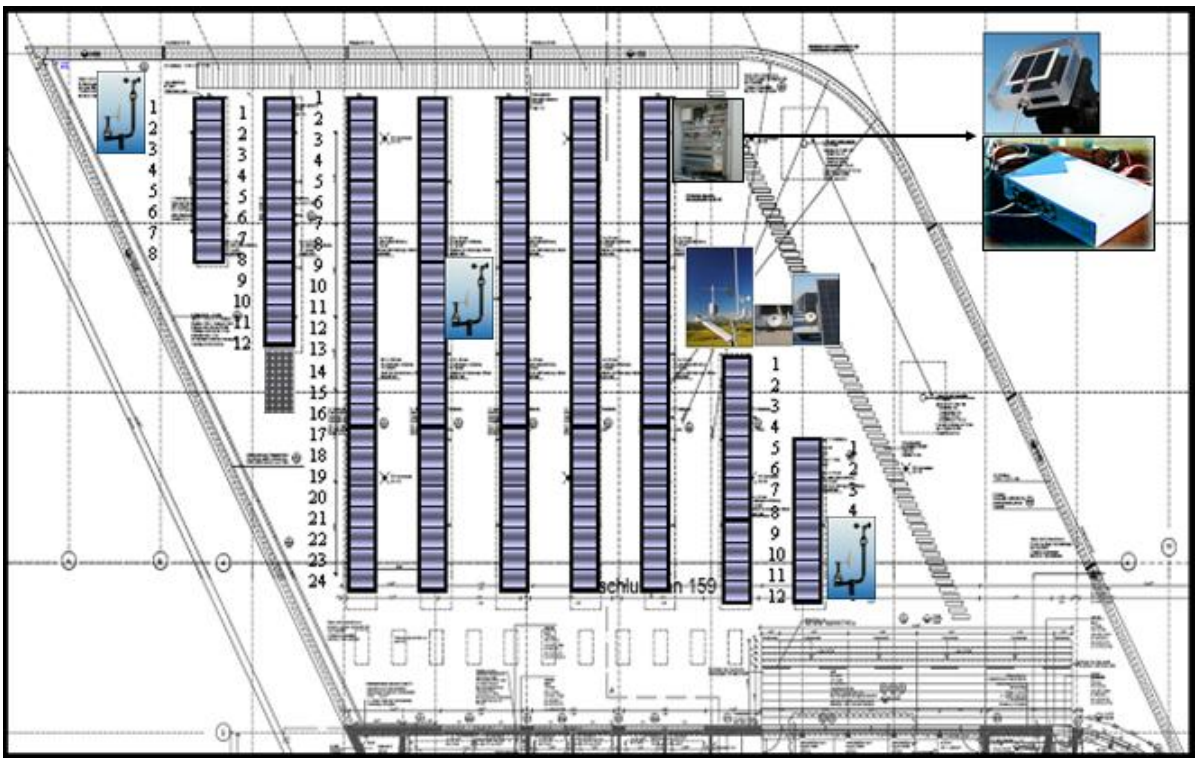
LM Last measurement in day.

MW mean value of the EM and LM for each month.

	Januar		Februar		März		April		Mai		Juni		Juli		August		September		Oktober		November		Dezember	
Tage	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM	EM	LM
1	09:13	15:24	09:44	14:57	09:48	14:51	10:39	15:44	10:27	15:42	10:12	15:57	09:57	16:23	09:47	16:38	09:39	16:34	09:29	16:24	08:27	15:13	08:46	15:04
28			08:55	15:45																				
30							09:38	16:31			10:08	16:11					10:24	15:29			09:15	14:34		
31	09:02	15:39			08:42	15:41			09:44	16:25			10:30	15:55	10:33	15:40			09:18	14:21			09:10	15:08
MW	09:07	15:07	09:19	15:19	09:15	15:15	10:08	16:08	10:05	16:05	10:10	16:10	10:13	16:13	10:10	16:10	10:01	16:01	09:23	15:23	08:51	14:51	08:58	14:58

Summary:

- In summer, the influence of the shadow is small, due to the higher altitude‘s sun that causes short shadows.
- Influence of the shadow increases with decreasing duration of sunshine and lower sun’s altitude.
- The influence of the railing is low because it is not closed, that means,they are actually bars.
- Inaccuracies in the sun angle chart, because it is taken a sun angle chart for a degree latitude of 48°, and Freiberg is 50,9167°. In addition, make the used points from only an approximation to the real environmental conditions.
- Then, from these new data, the final devices distribution is in Figure D.2.



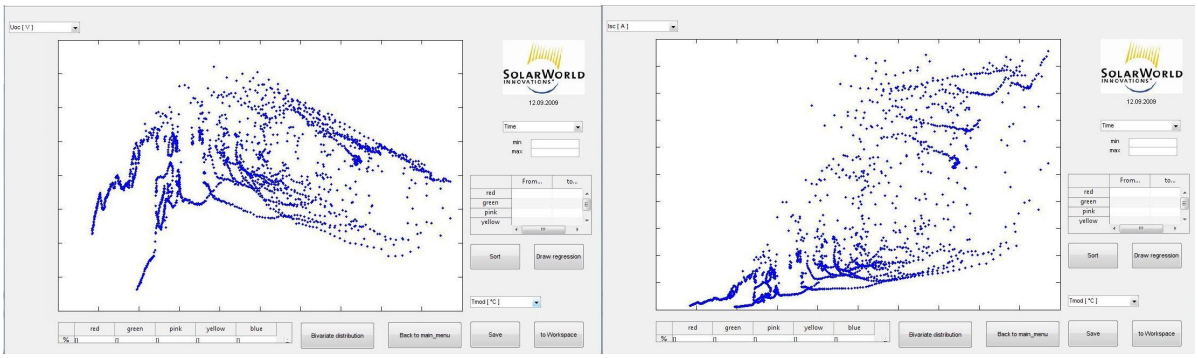
E. Example of analysis with the interfaces: Tmod

Given the amount of analysis that allows the program developed, and the amount of variables that would be interesting to focus, one variable will be chosen to evaluate, which has an special interest, module temperature. Since the influence of this variable in the operating of the module, and its dependence of external variables, makes it an interesting study for the development of improvements operation.

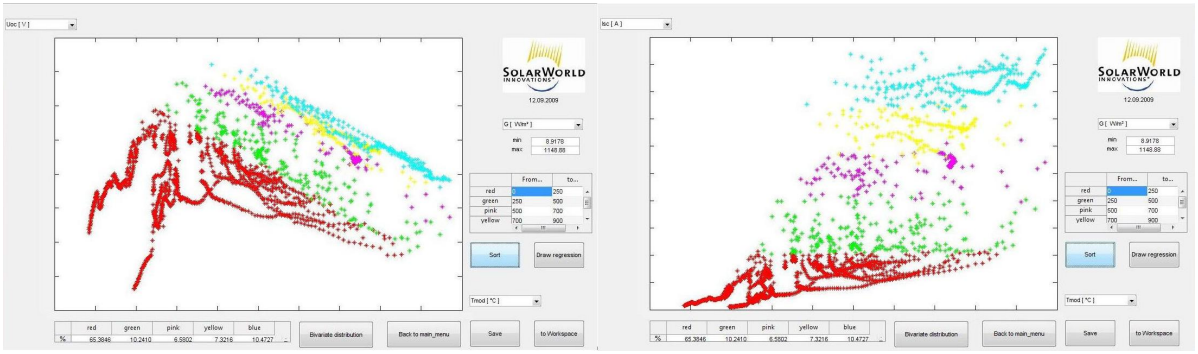
In this section several plot will be shown, and the conclusions will be presented in the next subject, conclusions.

E.1. Influence of temperature of the module in Isc and Voc

First observe how these parameters are influenced by the module temperature E.1



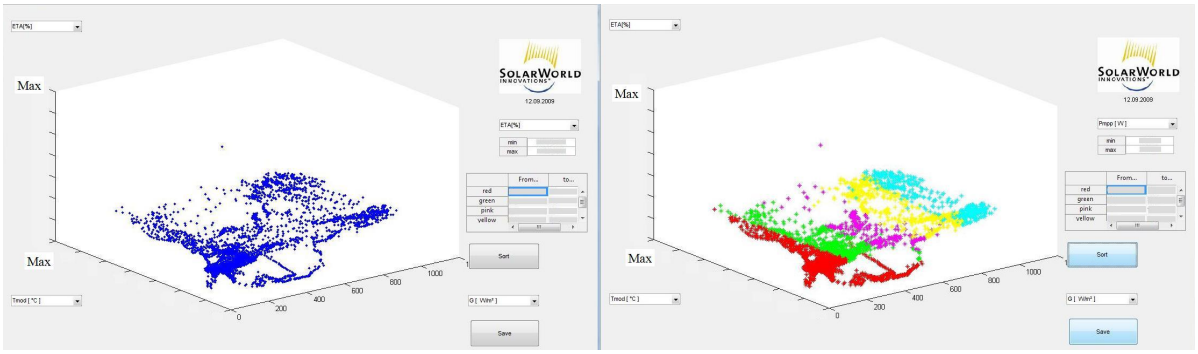
As we have clarified earlier,in the section **??**, with increasing module temperature, then Voc decreases. However, it pointed in the same paragraph that the Isc remained constant, and we can see that in our case increases. The reason is both Voc decreased as the constancy of the Isc are given to increase TMOD at conditions of constant radiation, so in the following graph E.1 we can observe that if we make a differentiation of five ranges in the radiation of our data these conclusions are easier to see.



It is observed very clearly as the Isc is almost constant for each range of radiation despite the increases in temperature, color lines horizontal. As we can see downlines for Voc.

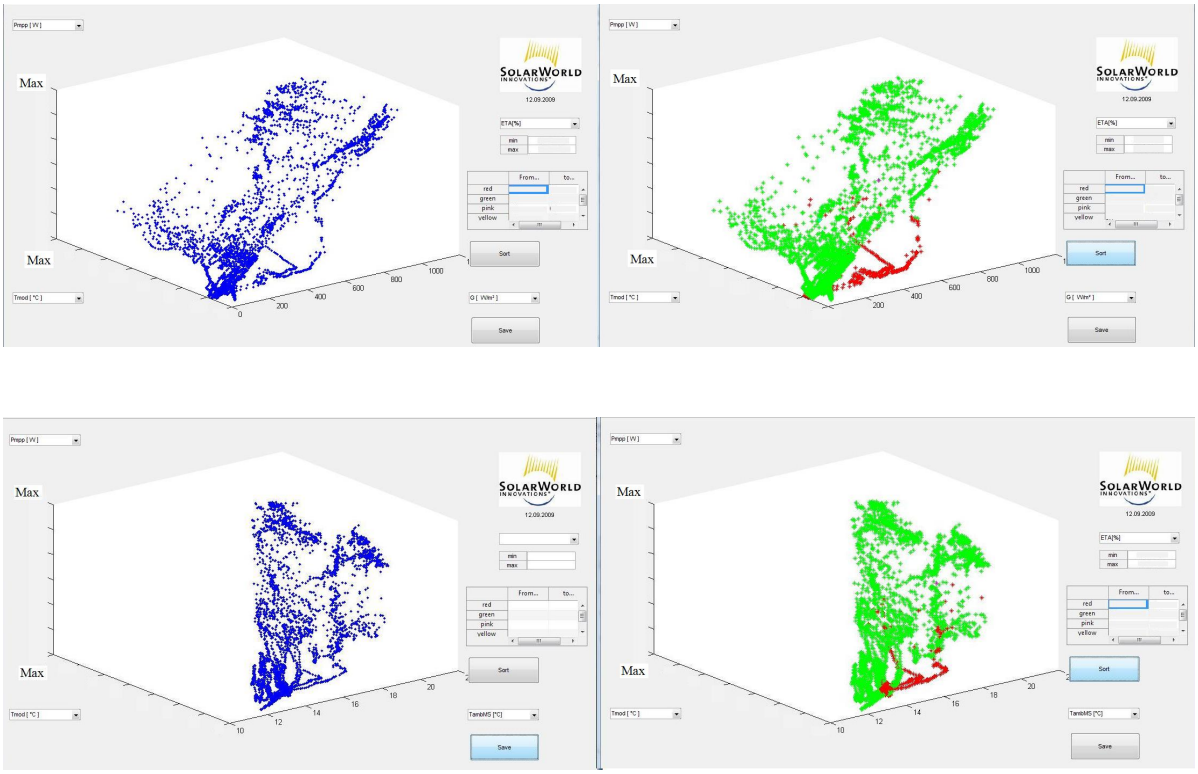
E.2. Influence of module temperature on output and power at the point of maximum power

First look at the graph of three-axis represents the performance we front module temperature and radiation, and then with this same a classification according Pmpp E.2, being the order of lowest to highest red, green, pink, yellow and cyan, in order to be followed all the representations.



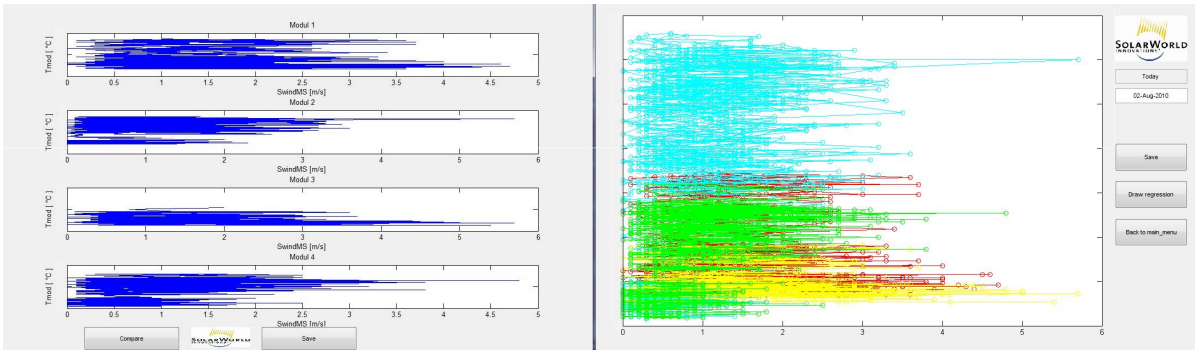
In the next figure E.2 can be seen the representation of the dependence of Pmpp, TMOD and G, also classified on the right side for output.

Finally we will see Pmpp dependence with module temperature and room temperature E.2, instead of radiation, on the right side find classification based on performance.



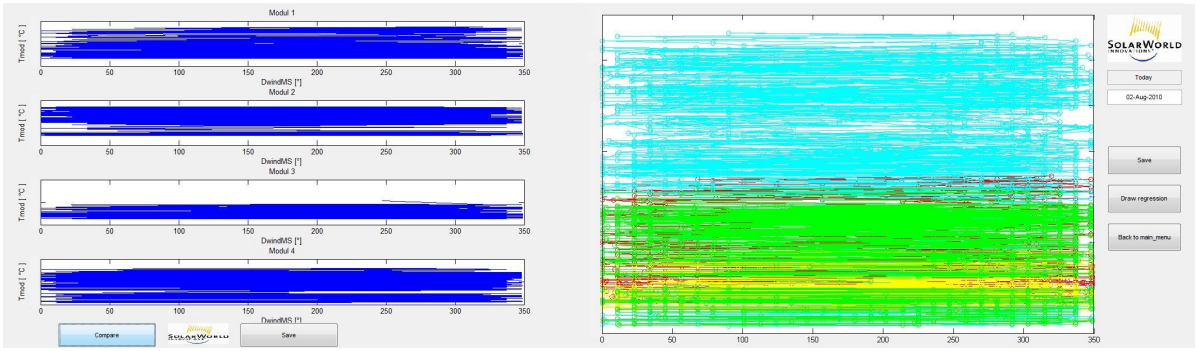
E.3. Module temperature dependence with the environmental conditions

In the next figure E.3 can be observed the module temperature dependence with the wind to four modules.

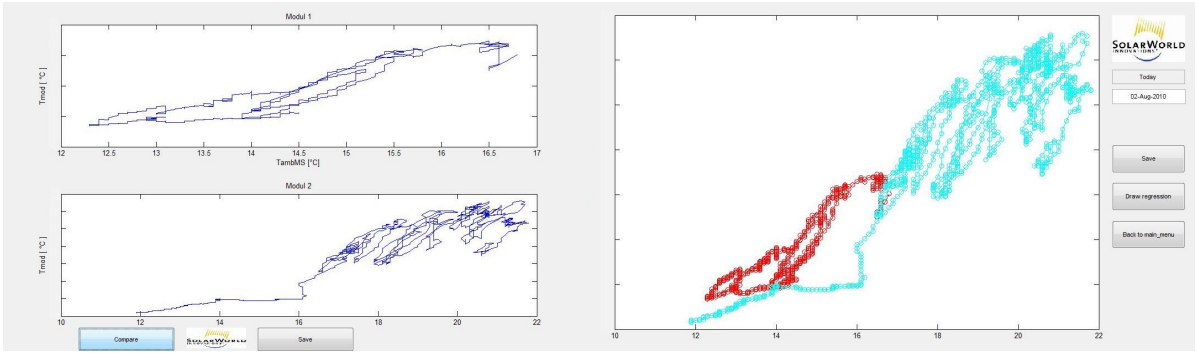


But we could not establish a clear dependence if we do not know what way the wind blows, so Figure E.3 try to clarify it.

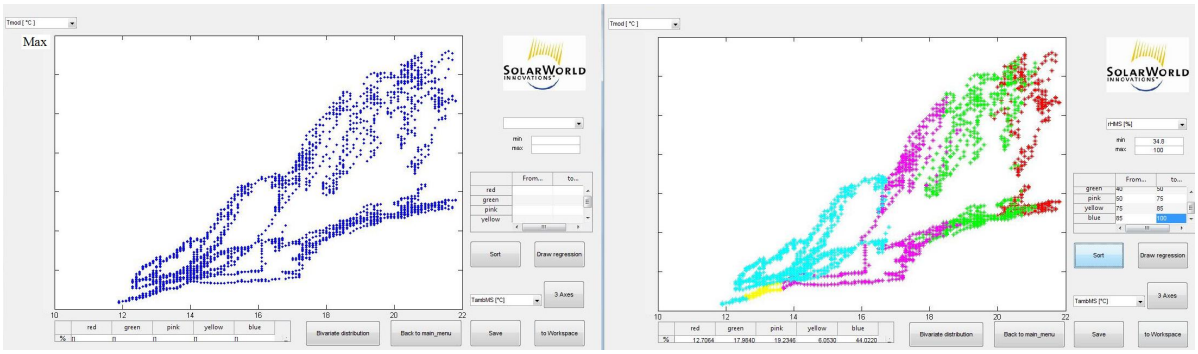
A dependency far more obvious and easily observed is that between module temper-



ature and room temperature, as seen for Figure E.3, that shows the dependence for the same module in two different days.

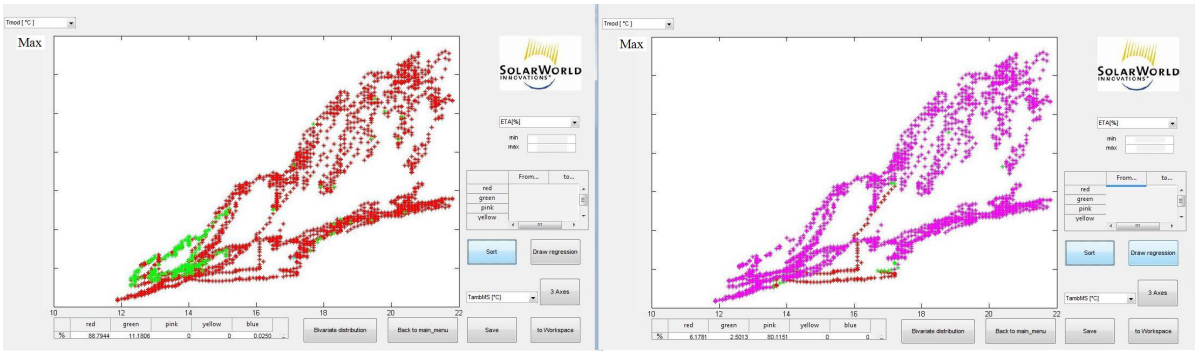


The three previous figures allow the comparison of different modules, to observe whether the behavior serves a type of pattern is what we'll evaluate all data together, may well see a trend. On the right side of the first Figure E.3 are the same representation as in the graph above E.3, ie Tmod versus Tamb, however in this case it is taken data from four modules and all have been assessed as if it were only one. On the right side the figure we have the same representation classified according to relative humidity air.

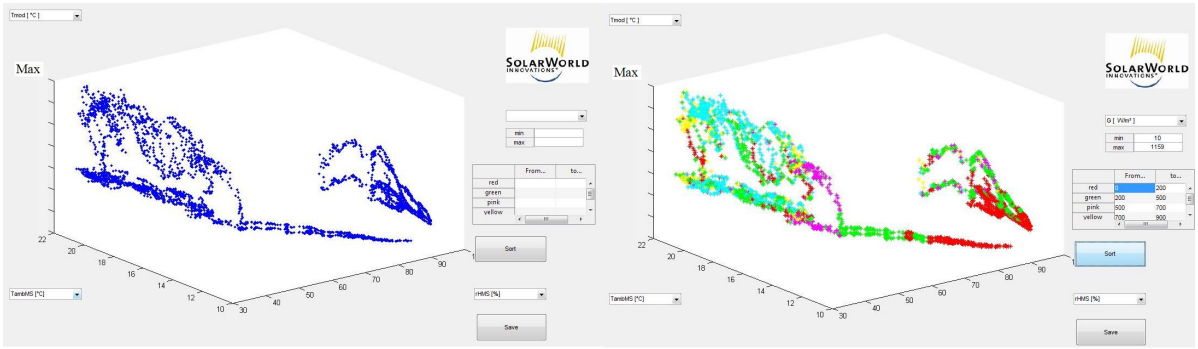


The Figure E.3 starts from the right side of E.3, but in this case has been made a

classification based on module output, the difference between right and left is that on the left the classification has been made into five ranks, ranging from minimum to maximum value, noting that the majority of points belong one of the ranks, has once again made another classification, right side, which has had three classification ranges limited by the maximum and minimum red range in the figure on the left. This allows us to classify and limit values of our variables, observing the behavior of the huge percentage.



Finally, the three axis representation E.3 of the module temperature compared to ambient temperature and humidity, all classified as radiation, right side.



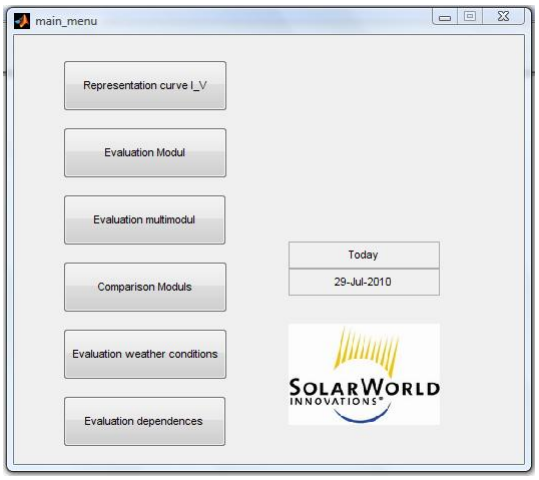
F. Programming

All the interfaces of the programming are here, first of all it is shown the interface, followed by the programming of this interface being easier to find the relation between the programming and the interface. There are some things that are common to all the interfaces:

- The first function of the programming of each interface has the name of the interface, it can be observed in the right up corner of the interface.
- The programming of the button has the name of the button followed by Callback.
- The parts that are in cyan are old attempts to solve which failed.
- The rest are command which can be found in the Help section of MATLAB.
- The programming of loops is relative easy to understand for people that are used to work it, furthermore there are comments, which trying to clarify the programming, they are preceded by %.

F.1. Main menu

This is the main interface, where you can choose an option to analyze.

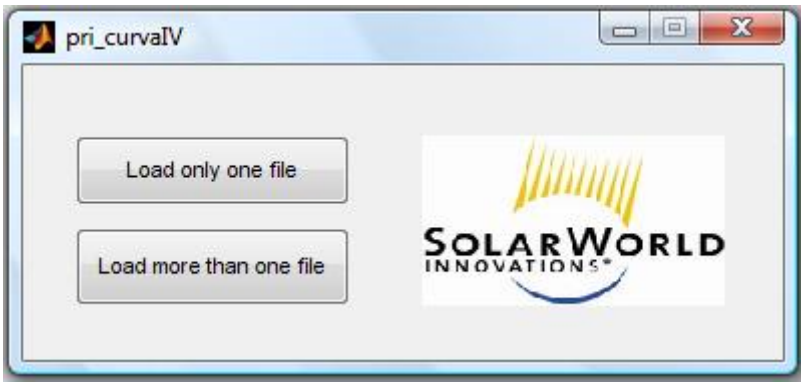


```
function varargout = main_menu(varargin)
% MAIN_MENU M-file for main_menu.fig
%      MAIN_MENU, by itself, creates a new MAIN_MENU or raises the
existing
%      singleton*.
%
%      H = MAIN_MENU returns the handle to a new MAIN_MENU or the handle
to
%      the existing singleton*.
%
%      MAIN_MENU('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in MAIN_MENU.M with the given input
arguments.
%
%      MAIN_MENU('Property','Value',...) creates a new MAIN_MENU or
raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before main_menu_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to main_menu_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main_menu

% Last Modified by GUIDE v2.5 01-Jun-2010 11:18:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_menu_OpeningFcn, ...
                   'gui_OutputFcn',  @main_menu_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT


% --- Executes just before main_menu is made visible.
```

```
function main_menu_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main_menu (see VARARGIN)

% Choose default command line output for main_menu
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

hoy=date;
set(handles.fecha, 'String', hoy)
% UIWAIT makes main_menu wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = main_menu_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in evaluacion.
function evaluacion_Callback(hObject, eventdata, handles)
% hObject    handle to evaluacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
int1
% --- Executes on button press in comparacion.
function comparacion_Callback(hObject, eventdata, handles)
% hObject    handle to comparacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
compare_12_try2


% --- Executes on button press in evtiempo.
function evtiempo_Callback(hObject, eventdata, handles)
% hObject    handle to evtiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
meteoevaluation
```

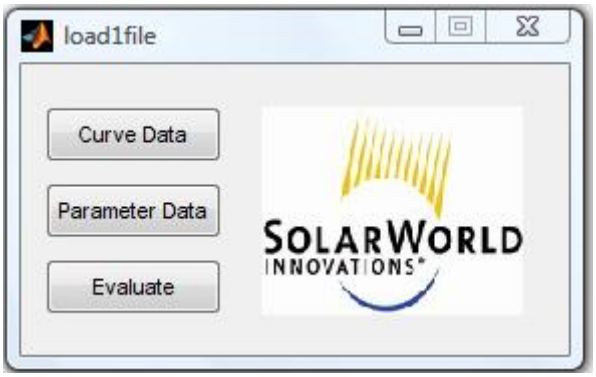
```
% --- Executes on button press in evdependencia.
function evdependencia_Callback(hObject, eventdata, handles)
% hObject    handle to evdependencia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Dependences_evaluation
% --- Executes on button press in curva.
function curva_Callback(hObject, eventdata, handles)
% hObject    handle to curva (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
pri_curvaIV


% --- Executes on button press in mmevaluation.
function mmevaluation_Callback(hObject, eventdata, handles)
% hObject    handle to mmevaluation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
int1_multiple
```

F.2. Representation curve IV



```
function varargout = pri_curvaIV(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @pri_curvaIV_OpeningFcn, ...
                   'gui_OutputFcn',  @pri_curvaIV_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function pri_curvaIV_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
%put logotype
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
%initialize Datos_IV
Datos_IV=[];


function varargout = pri_curvaIV_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;


function cargar_Callback(hObject, eventdata, handles)
guidata(hObject, handles);
% go to loadmorefiles
loadmorefiles

% global C_fecha Datos_IV  C_hora C_IV fid  car_xls Impp Vmpp Vocv Iscv
loadmul l
%
% [filename, pathname, filterindex] = uigetfile( {'*.xls','EXCEL-files
(*.xls)'; '*.txt','Text files (*.txt)'; '*.*', 'All Files (*.*)'}, 'Pick
a file', 'MultiSelect', 'on');
% l=length(filename);
% filename;
% %C_fecha=[];
% %C_hora=[];
% Datos_IV=[];
% %fprintf('pasa por el multiple');
% Datos_IV_old=[];
% Datos_IV_pro=[];
% Datos_IV_new=[];
% %Imppv=[];
% %Vmppv=[];
```

```
% Iscv=[];
% Vocv=[];
% for i=1:l
%     s = char(filename{1,i});
%     fid = fopen(s);
%     C_no1 = textscan(fid,'%*[^\n]',1);
% %     C_fecha(i) = textscan(fid,'%s',1);
% %     C_hora(i) = textscan(fid,'%s',1);
%     % C_no2 = textscan(fid,'%*[^\n]',14);
%     C_no2 = textscan(fid,'%*[^\n]',9);
%     C_MPP = textscan(fid,'%s %s %s %s %s %*[^\n]',1);
%
% %Imppv(i)=str2num(strrep(C_MPP{1,2}{1,1},',','.'));
%     %Vmppv(i)=str2num(strrep(C_MPP{1,1}{1,1},',','.'));
%     Iscv(i)=str2num(strrep(C_MPP{1,5}{1,1},',','.'));
%     Vocv(i)=str2num(strrep(C_MPP{1,4}{1,1},',','.'));
%     C_no3 = textscan(fid,'%*[^\n]',5);
%     C_IV = textscan(fid,'%s %s %*s ');
%     l=length(C_IV{1,1});
%     Datos_IV_new=zeros(l-1,2);
%     for i=1:l-1
%         for j=1:2
%             Datos_IV_new(i,j)=str2num(strrep(C_IV{1,j}{i,1},',','.'));
%         end
%     end
%     Datos_IV_pro=[Datos_IV_old;Datos_IV_new];
%     Datos_IV_old=Datos_IV_pro;
%     fclose(fid);
% end
% Datos_IV=Datos_IV_old;
%
% Datos_IV;
% loadmul=1;
%
% seg_curvaIV


%
% % --- Executes on button press in dibujar.
% function dibujar_Callback(hObject, eventdata, handles)
% % hObject    handle to dibujar (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
function cargauno_Callback(hObject, eventdata, handles)
load1file
```


load1file
Curve Data
Parameter Data
Evaluate
SOLARWORLD
INNOVATIONS

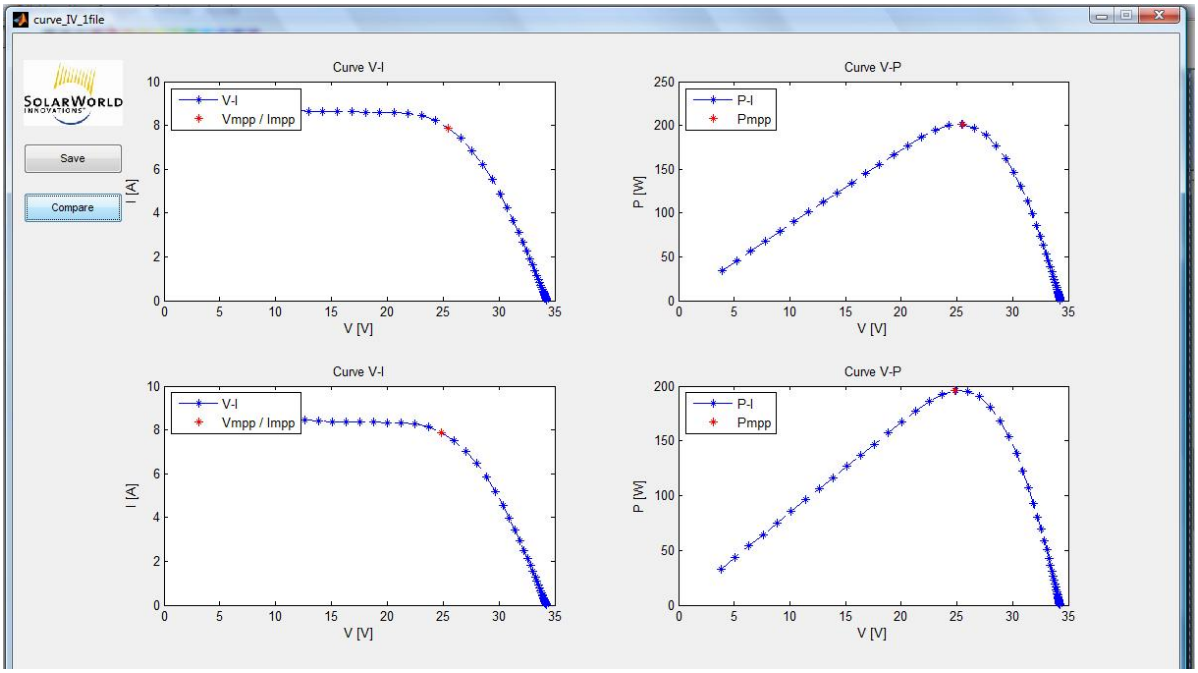
```
function varargout = load1file(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @load1file_OpeningFcn, ...
                   'gui_OutputFcn',  @load1file_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function load1file_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
%put logotype
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = load1file_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function loadcurvedata_Callback(hObject, eventdata, handles)
%Global variables
global G_c I_c V_c time date l3
%Load and open file
handles.fileName = uigetfile('*.txt');
fileName = handles.fileName;
fid = fopen(fileName);
%Read data and save in a cell
%C_IV{1,1}=V;C_IV{1,2}=I;C_IV{1,3}=G; (C_...=cell)
C_IV = textscan(fid,'%s %s %s ');
%l=amount of data
l=length(C_IV{1,1});
%Change comma to point in Irradiance vector, because Matlab read numbers
with point.
C_g=strrep(C_IV{1,3},',','.');
vector=[];
%Convert strings into numbers of the Irradiance vector
for i=1:l
    vector(i)=str2double(C_g(i));
end
%Find the Not a Number (NaN) and return a logical vector.
f=isnan(vector);
nanvector=[];
j=1;
%Find the position of NaNs and make a vector to save the positions.
for h=1:l
    if true(f(h))
```

```
        nanvector(j)=h;
        j=j+1;
    end
end
%how many NaN are there
l_nan=length(nanvector);
%To each group of data there are 3 NaN. So we know how many group of data
%there are, and so how many measurements of I-V data in a day.
l3=l_nan/3;
%Initialize cell for time, I, V and G data (..._c=cell)
time=cell(l3,1);
I_c=cell(l3,1);
V_c=cell(l3,1);
G_c=cell(l3,1);
%m = counter for counting with the 3 NaN of each group of data.
m=0;
%date is in these place in file.
date=C_IV{1,1}(nanvector(1),:);
%A cell for each Data (I,V,G,time), n sets the place of each group of
data
%in Cell.
for n=1:l3-1
    I_c{n,1}=C_IV{1,2}((nanvector(3+m)+1):(nanvector(4+m)-1));
    V_c{n,1}=C_IV{1,1}((nanvector(3+m)+1):(nanvector(4+m)-1));
    G_c{n,1}=C_IV{1,3}((nanvector(3+m)+1):(nanvector(4+m)-1));
    time{n,1}=C_IV{1,2}(nanvector(1+m),:);
    m=m+3;
end
I_c{l3,1}=C_IV{1,2}((nanvector(l_nan)+1):l);
V_c{l3,1}=C_IV{1,1}((nanvector(l_nan)+1):l);
G_c{l3,1}=C_IV{1,3}((nanvector(l_nan)+1):l);
time{l3,1}=C_IV{1,2}(nanvector(l_nan-2),:);
fclose(fid);

function loadparameterdata_Callback(hObject, eventdata, handles)
%Global variables
global Vmpp Impp Pmpp Voc Isc FF G Tmodul Vwind Rwind Tamb rF  Pyr1 Pyr2
Pyr3 Vwind1 Vwind2 Vwind3 SiC1 SiC2 SiC3 SiC4 Modul Date_c
%Load and open file
handles.fileName = uigetfile('*.txt');
fileName = handles.fileName;
fid = fopen(fileName);
%Skip 2 lines.
C_no=textscan(fid,'%*[^\n]',2);
%Read data and save in a cell, 27 columns, the data are electrical and
%meteorological
C_PVMeteodaten = textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s');
%Number of modul 1...12
Modul=str2double(C_PVMeteodaten{1,1}(1,:));
Date_c{1,1}=C_PVMeteodaten{1,2}(1,:);
%How many data, this must be equal to l3.
l_pvm=length(C_PVMeteodaten{1,1});
Vmpp=0;
Impp=0;
Pmpp=0;
%Vectors to each variable, time is equal to time in curve file
```

```
Vmpp=str2double(strrep(C_PVMeteodaten{1,4},',','.'));
Impp=str2double(strrep(C_PVMeteodaten{1,5},',','.'));
Pmpp=str2double(strrep(C_PVMeteodaten{1,6},',','.'));
Voc=str2double(strrep(C_PVMeteodaten{1,7},',','.'));
Isc=str2double(strrep(C_PVMeteodaten{1,8},',','.'));
FF=str2double(strrep(C_PVMeteodaten{1,9},',','.'));
G=str2double(strrep(C_PVMeteodaten{1,10},',','.'));
Tmodul=str2double(strrep(C_PVMeteodaten{1,11},',','.'));
Vwind=str2double(strrep(C_PVMeteodaten{1,14},',','.'));
Rwind=str2double(strrep(C_PVMeteodaten{1,15},',','.'));
Tamb=str2double(strrep(C_PVMeteodaten{1,16},',','.'));
rF=str2double(strrep(C_PVMeteodaten{1,17},',','.'));
Pyr1=str2double(strrep(C_PVMeteodaten{1,18},',','.'));
Pyr2=str2double(strrep(C_PVMeteodaten{1,19},',','.'));
Pyr3=str2double(strrep(C_PVMeteodaten{1,20},',','.'));
Vwind1=str2double(strrep(C_PVMeteodaten{1,21},',','.'));
Vwind2=str2double(strrep(C_PVMeteodaten{1,22},',','.'));
Vwind3=str2double(strrep(C_PVMeteodaten{1,23},',','.'));
SiC1=str2double(strrep(C_PVMeteodaten{1,24},',','.'));
SiC2=str2double(strrep(C_PVMeteodaten{1,25},',','.'));
SiC3=str2double(strrep(C_PVMeteodaten{1,26},',','.'));
SiC4=str2double(strrep(C_PVMeteodaten{1,27},',','.'));
fclose(fid);

% --- Executes on button press in evaluatebutton.
function evaluatebutton_Callback(hObject, eventdata, handles)
% hObject    handle to evaluatebutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
writehour
```


writehour
Write an hour hh:mm, for example: 09 43
Continue
SOLARWORLD
INNOVATIONS

```
function varargout = writehour(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @writehour_OpeningFcn, ...
                   'gui_OutputFcn',  @writehour_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function writehour_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = writehour_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function hour_Callback(hObject, eventdata, handles)

function hour_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function continuebutton_Callback(hObject, eventdata, handles)
global hourtodrawn
%take the string and go to next interface curve_IV_1file
hourtodrawn=get(handles.hour, 'String');
curve_IV_1file
```


curve_IV_1file
SOLARWORLD
INNOVATIONS
Save
Compare
Curve V-I
V-I
Vmpp / Impp
I [A]
V [V]
Curve V-P
P-I
Pmpp
P [W]
V [V]
Curve V-I
V-I
Vmpp / Impp
I [A]
V [V]
Curve V-P
P-I
Pmpp
P [W]
V [V]

```
function varargout = curve_IV_1file(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @curve_IV_1file_OpeningFcn, ...
                   'gui_OutputFcn',  @curve_IV_1file_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function curve_IV_1file_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global G_c I_c V_c time l3 hourtodrawn Vmpp Impp Pmpp position i1 v1 g1
p1 i2 v2 g2 p2 i3 v3 g3 p3 i4 v4 g4 p4 i5 v5 g5 p5 i6 v6 g6 p6
timetodrawn
% global vmpp1 impp1 pmpp1 vmpp2 impp2 pmpp2 vmpp3 impp3 pmpp3 vmpp4
impp4 pmpp4  vmpp5 impp5 pmpp5 vmpp6 impp6 pmpp6
global vmpp_v impp_v pmpp_v
axes(handles.logo)

handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
timetodrawn=[];
j=0;
%Find the time, and save the position of these hours.
for i=1:l3
    a=time{i,1};
    hhmmdrawn=a{1,1}(1:5);
    if hourtodrawn==hhmmdrawn
        j=j+1;
        timetodrawn(j)=i;

    end
end
%j number of group of data in this hh:mm, for example 2 when a
measurement
%each 30 seconds, and save the position
%if j=0, there are no data to this time
if j==0
    errordlg('You must other hour choose','File Error');
end

i=[];
v=[];
g=[];
p=[];
```

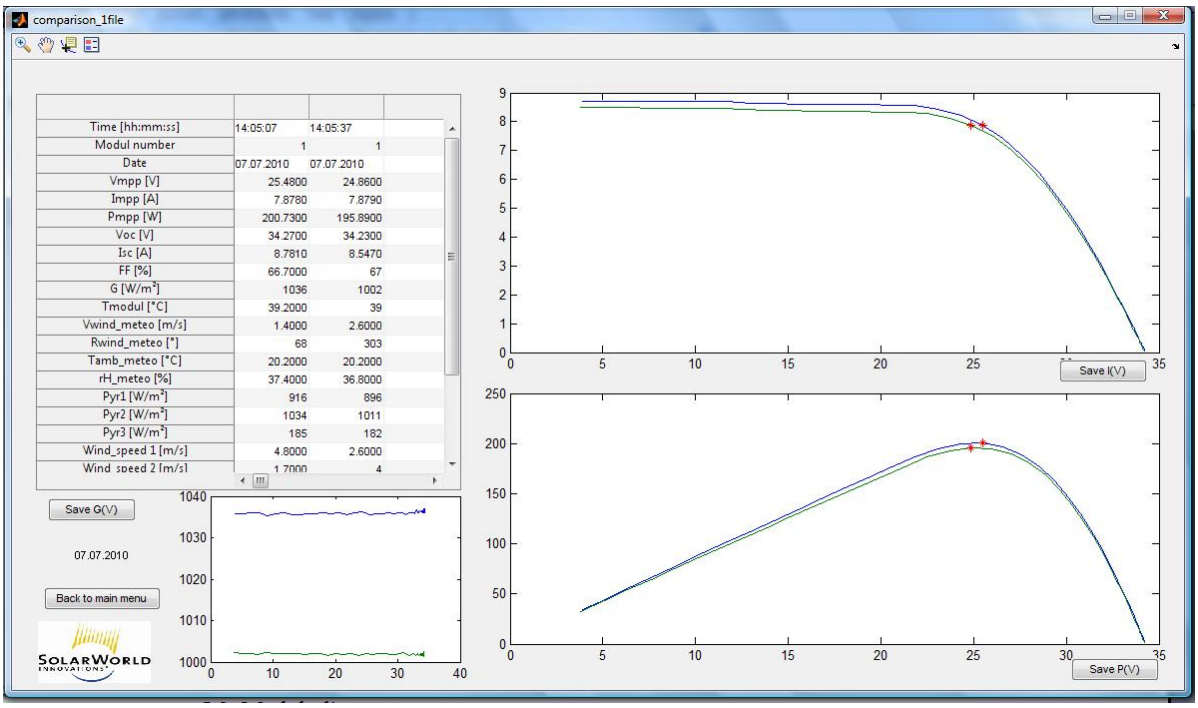
```
ioutzero=[];
voutzero=[];
goutzero=[];
poutzero=[];
q=0;
cont=0;
position=0;

i1=[];
v1=[];
g1=[];
p1=[];
i2=[];
v2=[];
g2=[];
p2=[];
i3=[];
v3=[];
g3=[];
p3=[];
i4=[];
v4=[];
g4=[];
p4=[];
i5=[];
v5=[];
g5=[];
p5=[];
i6=[];
v6=[];
g6=[];
p6=[];
vmpp=0;
impp=0;
pmpp=0;
% vmpp1=0;
% impp1=0;
% pmpp1=0;
% vmpp2=0;
% impp2=0;
% pmpp2=0;
% vmpp3=0;
% impp3=0;
% pmpp3=0;
% vmpp4=0;
% impp4=0;
% pmpp4=0;
% vmpp5=0;
% impp5=0;
% pmpp5=0;
% vmpp6=0;
% impp6=0;
% pmpp6=0;
vmpp_v=[];
impp_v=[];
pmpp_v=[];
```

```
%transfer data of a cell to a vector, and plot
for h=1:j
    position=timetodrawn(h);
    l_c=length(I_c{position,1});
    for k=1:l_c
        i(k)=str2double(strrep(I_c{position,1}(k,1),',','.'));
        v(k)=str2double(strrep(V_c{position,1}(k,1),',','.'));
        g(k)=str2double(strrep(G_c{position,1}(k,1),',','.'));

        if i(k)~=0 && v(k)~=0
            q=q+1;
            ioutzero(q)=i(k);
            voutzero(q)=v(k);
            goutzero(q)=g(k);
            poutzero(q)=ioutzero(q)*voutzero(q);
        end
    end
    vmpp=Vmpp(position);
    impp=Impp(position);
    pmpp=Pmpp(position);
    subplot(j,2,1+cont), plot(voutzero,ioutzero,'b*--',vmpp,impp,'r*')
    title('Curve V-I')
    xlabel('V [V]')
    ylabel('I [A]')
    m = legend('V-I','Vmpp / Impp',2);
    set(m,'Interpreter','none')
    subplot(j,2,2+cont), plot(voutzero,poutzero,'b*--',vmpp, pmpp,'r*')
    title('Curve V-P')
    xlabel('V [V]')
    ylabel('P [W]')
    o = legend('P-I','Pmpp',2);
    set(o,'Interpreter','none')


    if h==1
        i1=ioutzero;
        v1=voutzero;
        g1=goutzero;
        p1=poutzero;
        vmpp_v(1)=vmpp;
        impp_v(1)=impp;
        pmpp_v(1)=pmpp;
    elseif h==2
        i2=ioutzero;
        v2=voutzero;
        g2=goutzero;
        p2=poutzero;
        vmpp_v(2)=vmpp;
        impp_v(2)=impp;
        pmpp_v(2)=pmpp;
    elseif h==3
        i3=ioutzero;
        v3=voutzero;
        g3=goutzero;
        p3=poutzero;
        vmpp_v(3)=vmpp;
```

```
        impp_v(3)=impp;
        pmpp_v(3)=pmpp;
    elseif h==4
        i4=ioutzero;
        v4=voutzero;
        g4=goutzero;
        p4=poutzero;
        vmpp_v(4)=vmpp;
        impp_v(4)=impp;
        pmpp_v(4)=pmpp;
    elseif h==5
        i5=ioutzero;
        v5=voutzero;
        g5=goutzero;
        p5=poutzero;
        vmpp_v(5)=vmpp;
        impp_v(5)=impp;
        pmpp_v(5)=pmpp;
    else
        i6=ioutzero;
        v6=voutzero;
        g6=goutzero;
        p6=poutzero;
        vmpp_v(6)=vmpp;
        impp_v(6)=impp;
        pmpp_v(6)=pmpp;
    end
    cont=cont+2;
    i=[];
    v=[];
    g=[];
    p=[];
    ioutzero=[];
    voutzero=[];
    goutzero=[];
    poutzero=[];
    q=0;
    vmpp=0;
    impp=0;
    pmpp=0;

end

function varargout = curve_IV_1file_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function savebutton_Callback(hObject, eventdata, handles)
global G_c I_c V_c time l3 hourtodrawn Vmpp Impp Pmpp position

figure

timetodrawn=[];
j=0;
%Find the time, and save the position of these hours.
for i=1:l3
```

```
    a=time{i,1};
    hhmmdrawn=a{1,1}(1:5);
    if hourtodrawn==hhmmdrawn
        j=j+1;
        timetodrawn(j)=i;

    end
end
%j number of group of data in this hh:mm, for example 2 when a
measurement
%each 30 seconds, and save the position
%if j=0, there are no data to this time
if j==0
    errordlg('You must other hour choose','File Error');
end

i=[];
v=[];
g=[];
p=[];
ioutzero=[];
voutzero=[];
goutzero=[];
poutzero=[];
q=0;
cont=0;
position=0;

%transfer data of a cell to a vector, and plot
for h=1:j
    position=timetodrawn(h);
    l_c=length(I_c{position,1});
    for k=1:l_c
        i(k)=str2double(strrep(I_c{position,1}(k,1),',','.'));
        v(k)=str2double(strrep(V_c{position,1}(k,1),',','.'));
        g(k)=str2double(strrep(G_c{position,1}(k,1),',','.'));

        if i(k)~=0 && v(k)~=0
            q=q+1;
            ioutzero(q)=i(k);
            voutzero(q)=v(k);
            goutzero(q)=g(k);
            poutzero(q)=ioutzero(q)*voutzero(q);
        end
    end

    subplot(j,2,1+cont), plot(voutzero,ioutzero,'b*--
',Vmpp(position),Impp(position),'r*')
    title('Curve V-I')
    xlabel('V [V]')
    ylabel('I [A]')
    m = legend('V-I','Vmpp / Impp',2);
    set(m,'Interpreter','none')
    subplot(j,2,2+cont), plot(voutzero,poutzero,'b*--
',Vmpp(position),Pmpp(position),'r*')
    title('Curve V-P')
```

```
    xlabel('V [V]')
    ylabel('P [W]')
    o = legend('P-I','Pmpp',2);
    set(o,'Interpreter','none')

    cont=cont+2;
    i=[];
    v=[];
    g=[];
    p=[];
    ioutzero=[];
    voutzero=[];
    goutzero=[];
    poutzero=[];
    q=0;
end

function comparebutton_Callback(hObject, eventdata, handles)
comparison_1file
```


comparison_1file
Time [hh:mm:ss] 14:05:07 14:05:37
Modul number 1 1
Date 07.07.2010 07.07.2010
Vmpp [V] 25.4800 24.8600
Impp [A] 7.8780 7.8790
Pmpp [W] 200.7300 195.8900
Voc [V] 34.2700 34.2300
Isc [A] 8.7810 8.5470
FF [%] 66.7000 67
G [W/m²] 1036 1002
Tmodul [°C] 39.2000 39
Vwind_meteo [m/s] 1.4000 2.6000
Rwind_meteo [°] 68 303
Tamb_meteo [°C] 20.2000 20.2000
rH_meteo [%] 37.4000 36.8000
Pyr1 [W/m²] 916 896
Pyr2 [W/m²] 1034 1011
Pyr3 [W/m²] 185 182
Wind_speed 1 [m/s] 4.8000 2.6000
Wind_speed 2 [m/s] 1.7000 4
Save G(V)
07.07.2010
Back to main menu
SolarWorld
Save I(V)
Save P(V)

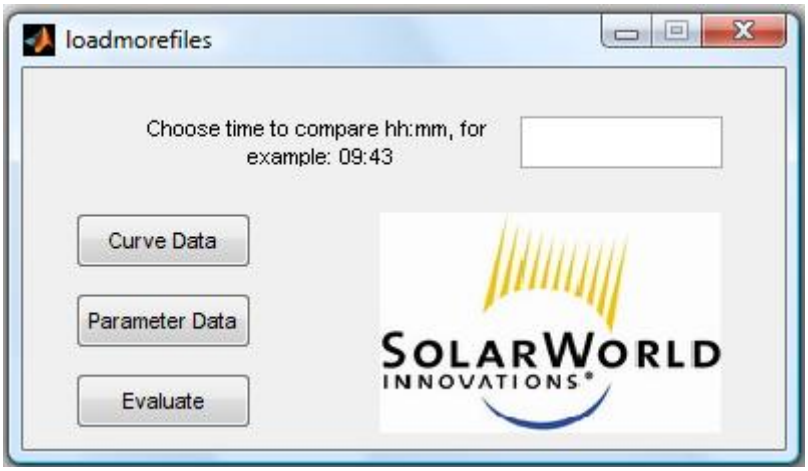
```
function varargout = comparison_1file(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @comparison_1file_OpeningFcn, ...
                   'gui_OutputFcn',  @comparison_1file_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function comparison_1file_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);

global  time Vmpp Impp Pmpp Voc Isc FF G Tmodul Vwind Rwind Tamb rF  Pyr1
Pyr2 Pyr3 Vwind1 Vwind2 Vwind3 SiC1 SiC2 SiC3 SiC4 position  date
global i1 v1 g1 p1 i2 v2 g2 p2 i3 v3 g3 p3 i4 v4 g4 p4 i5 v5 g5 p5 i6 v6
g6 p6 timetodrawn Modul Date_c
global vmpp_v impp_v pmpp_v l_v
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
set(handles.dayofevaluation, 'String', date)
ltime=length(timetodrawn);

datatable=cell(ltime,26);
position=0;
vmp=[];
imp=[];
pmp=[];
% put the data in the table, by means of the cell
for j=1:ltime
    position=timetodrawn(j);
    a=datestr(time{position,1},'HH:MM:SS');
    datatable{1,j}=a;
    datatable{2,j}=Modul;
    b=datenum(Date_c{1,1},'dd.mm.yyyy');
    datatable{3,j}=datestr(b,'dd.mm.yyyy');
    datatable{4,j}=Vmpp(position);
    datatable{5,j}=Impp(position);
    datatable{6,j}=Pmpp(position);
    datatable{7,j}=Voc(position);
    datatable{8,j}=Isc(position);
    datatable{9,j}=FF(position);
    datatable{10,j}=G(position);
    datatable{11,j}=Tmodul(position);
```

```
    datatable{12,j}=Vwind(position);
    datatable{13,j}=Rwind(position);
    datatable{14,j}=Tamb(position);
    datatable{15,j}=rF(position);
    datatable{16,j}=Pyr1(position);
    datatable{17,j}=Pyr2(position);
    datatable{18,j}=Pyr3(position);
    datatable{19,j}=Vwind1(position);
    datatable{20,j}=Vwind2(position);
    datatable{21,j}=Vwind3(position);
    datatable{22,j}=SiC1(position);
    datatable{23,j}=SiC2(position);
    datatable{24,j}=SiC3(position);
    datatable{25,j}=SiC4(position);
    Pmpp_parameter=Pmpp(position);
    G_parameter=G(position);
    nu=1-(Pmpp_parameter/(1.67601*G_parameter));
    datatable{26,j}=nu;
    position=0;
end
% plot(handles.V_I, v1,i1,vmp(1),imp(1), v2,i2,vmp(2),imp(2),
v3,i3,vmp(3),imp(3), v4,i4,vmp(4),imp(4), v5,i5,vmp(5),imp(5),
v6,i6,vmp(6),imp(6))
% plot(handles.V_P, v1,p1,vmp(1),pmp(1), v2,p2,vmp(2),pmp(2),
v3,p3,vmp(3),pmp(3), v4,p4,vmp(4),pmp(4), v5,p5,vmp(5),pmp(5),
v6,p6,vmp(6),pmp(6))
% plot(handles.G_V, v1,g1, v2,g2, v3,g3, v4,g4, v5,g5, v6,g6)

%draw plot, depends on number of data in this hour
l_v=length(vmpp_v);
if l_v==6
plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4),
impp_v(4),'r*', v5,i5,vmpp_v(5), impp_v(5),'r*', v6,i6,vmpp_v(6),
impp_v(6),'r*')
plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4),
pmpp_v(4),'r*', v5,p5,vmpp_v(5), pmpp_v(5),'r*', v6,p6,vmpp_v(6),
pmpp_v(6),'r*')
plot(handles.G_V, v1,g1, v2,g2, v3,g3, v4,g4, v5,g5, v6,g6)
elseif l_v==5
plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4),
impp_v(4),'r*', v5,i5,vmpp_v(5), impp_v(5),'r*')
plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4),
pmpp_v(4),'r*', v5,p5,vmpp_v(5), pmpp_v(5),'r*')
plot(handles.G_V, v1,g1, v2,g2, v3,g3, v4,g4, v5,g5)

elseif l_v==4
    plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4),
impp_v(4),'r*')
    plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4),
pmpp_v(4),'r*')
    plot(handles.G_V, v1,g1, v2,g2, v3,g3, v4,g4)
```

```
elseif l_v==3
    plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*')
    plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*')
    plot(handles.G_V, v1,g1, v2,g2, v3,g3)

elseif l_v==2
    plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*')
    plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*')
    plot(handles.G_V, v1,g1, v2,g2)

else
     plot(handles.V_I, v1,i1,vmpp_v(1), impp_v(1),'r*')
     plot(handles.V_P, v1,p1,vmpp_v(1), pmpp_v(1),'r*')
     plot(handles.G_V, v1,g1)
end
%set the data in the table
set(handles.alldatatable,'Data',datatable)

function varargout = comparison_1file_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function backbutton_Callback(hObject, eventdata, handles)
%close every interfaces, what no more useful are, and open the main_menu
close pri_curvaIV
close load1file
close writehour
close curve_IV_1file
main_menu

function saveivbutton_Callback(hObject, eventdata, handles)
global i1 v1  i2 v2 i3 v3 i4 v4 i5 v5 i6 v6
global vmpp_v impp_v  l_v
%save plot IV in a extra figure
figure
if l_v==6
plot(v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2), impp_v(2),'r*',
v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4), impp_v(4),'r*',
v5,i5,vmpp_v(5), impp_v(5),'r*', v6,i6,vmpp_v(6), impp_v(6),'r*')

elseif l_v==5
plot( v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2), impp_v(2),'r*',
v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4), impp_v(4),'r*',
v5,i5,vmpp_v(5), impp_v(5),'r*')

elseif l_v==4
    plot( v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*', v4,i4,vmpp_v(4),
impp_v(4),'r*')
```

```
elseif l_v==3
    plot( v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*', v3,i3,vmpp_v(3), impp_v(3),'r*')

elseif l_v==2
    plot( v1,i1,vmpp_v(1), impp_v(1),'r*', v2,i2,vmpp_v(2),
impp_v(2),'r*')


else
     plot( v1,i1,vmpp_v(1), impp_v(1),'r*')
end
xlabel('V [V]')
ylabel('I [A]')
title('Curve I-V')

function savepvbutton_Callback(hObject, eventdata, handles)
global i1 v1 g1 p1 i2 v2 g2 p2 i3 v3 g3 p3 i4 v4 g4 p4 i5 v5 g5 p5 i6 v6
g6 p6
global vmpp_v pmpp_v  l_v
%save plot PV in a extra figure
figure
if l_v==6
plot(v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2), pmpp_v(2),'r*',
v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4), pmpp_v(4),'r*',
v5,p5,vmpp_v(5), pmpp_v(5),'r*', v6,p6,vmpp_v(6), pmpp_v(6),'r*')
elseif l_v==5
plot(v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2), pmpp_v(2),'r*',
v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4), pmpp_v(4),'r*',
v5,p5,vmpp_v(5), pmpp_v(5),'r*')
elseif l_v==4
    plot( v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*', v4,p4,vmpp_v(4),
pmpp_v(4),'r*')

elseif l_v==3
    plot( v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*', v3,p3,vmpp_v(3), pmpp_v(3),'r*')

elseif l_v==2
    plot( v1,p1,vmpp_v(1), pmpp_v(1),'r*' ,v2,p2,vmpp_v(2),
pmpp_v(2),'r*')

else
     plot( v1,p1,vmpp_v(1), pmpp_v(1),'r*')

end
xlabel('V [V]')
ylabel('P [W]')
title('Curve P-V')

function savegvbutton_Callback(hObject, eventdata, handles)
global i1 v1 g1 p1 i2 v2 g2 p2 i3 v3 g3 p3 i4 v4 g4 p4 i5 v5 g5 p5 i6 v6
g6 p6
figure
%save plot GV in a extra figure
```

```
plot( v1,g1, v2,g2, v3,g3, v4,g4, v5,g5, v6,g6)
xlabel('V [V]')
ylabel('G [W/m²]')
title('Curve G-V')
```


loadmorefiles
Choose time to compare hh:mm, for example: 09:43
Curve Data
Parameter Data
Evaluate
SOLARWORLD
INNOVATIONS

```
function varargout = loadmorefiles(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @loadmorefiles_OpeningFcn, ...
                   'gui_OutputFcn',  @loadmorefiles_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function loadmorefiles_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = loadmorefiles_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function loadcurvedata_Callback(hObject, eventdata, handles)
global G_c I_c V_c time date l3
global I1 P1 V1 G1 I2 P2 V2 G2 I3 P3 V3 G3 I4 P4 V4 G4 I5 P5 V5 G5 I6 P6
V6 G6 I7 P7 V7 G7 I8 P8 V8 G8 I9 P9 V9 G9 I10 P10 V10 G10 I11 P11 V11 G11
I12 P12 V12 G12
global hourtodrawn pos
hourtodrawn=get(handles.hour, 'String');
[filename, pathname, filterindex] = uigetfile( {'*.xls','EXCEL-files
(*.xls)'; '*.txt','Text files (*.txt)'; '*.*', 'All Files (*.*)'}, 'Pick
a file', 'MultiSelect', 'on');
l_files=length(filename);
I1=[];
P1=[];
V1=[];
G1=[];
I2=[];
P2=[];
V2=[];
G2=[];
I3=[];
P3=[];
V3=[];
G3=[];
I4=[];
P4=[];
V4=[];
G4=[];
I5=[];
```

```
P5=[];
V5=[];
G5=[];
I6=[];
P6=[];
V6=[];
G6=[];
I7=[];
P7=[];
V7=[];
G7=[];
I8=[];
P8=[];
V8=[];
G8=[];
I9=[];
P9=[];
V9=[];
G9=[];
I10=[];
P10=[];
V10=[];
G10=[];
I11=[];
P11=[];
V11=[];
G11=[];
I12 =[];
P12=[];
V12=[];
G12=[];
pos=[];
for k=1:l_files
    s = char(filename{1,k});
    fid = fopen(s);
    C_IV = textscan(fid,'%s %s %s ');
    %l=amount of data
    l=length(C_IV{1,1});
    %Change comma to point in Irradiance vector, because Matlab read
numbers with point.
    C_g=strrep(C_IV{1,3},',','.');
    vector=[];
    %Convert strings into numbers of the Irradiance vector
    for iv=1:l
        vector(iv)=str2double(C_g(iv));
    end
    %Find the Not a Number (NaN) and return a logical vector.
    f=isnan(vector);
    nanvector=[];
    j=1;
    %Find the position of NaNs and make a vector to save the positions.
    for h=1:l
        if true(f(h))
            nanvector(j)=h;
            j=j+1;
        end
    end
```

```
    %how many NaN are there
    l_nan=length(nanvector);
    %To each group of data there are 3 NaN. So we know how many group of
data
    %there are, and so how many measurements of I-V data in a day.
    l3=l_nan/3;
    %Initialize cell for time, I, V and G data (..._c=cell)
    time=cell(l3,1);
    I_c=cell(l3,1);
    V_c=cell(l3,1);
    G_c=cell(l3,1);
    %m = counter for counting with the 3 NaN of each group of data.
    m=0;
    %date is in these place in file.
    date=C_IV{1,1}(nanvector(1),:);
    %A cell for each Data (I,V,G,time), n sets the place of each group of
data
    %in Cell.
    for n=1:l3-1
        I_c{n,1}=C_IV{1,2}((nanvector(3+m)+1):(nanvector(4+m)-1));
        V_c{n,1}=C_IV{1,1}((nanvector(3+m)+1):(nanvector(4+m)-1));
        G_c{n,1}=C_IV{1,3}((nanvector(3+m)+1):(nanvector(4+m)-1));
        time{n,1}=C_IV{1,2}(nanvector(1+m),:);
        m=m+3;
    end
    I_c{l3,1}=C_IV{1,2}((nanvector(l_nan)+1):l);
    V_c{l3,1}=C_IV{1,1}((nanvector(l_nan)+1):l);
    G_c{l3,1}=C_IV{1,3}((nanvector(l_nan)+1):l);
    time{l3,1}=C_IV{1,2}(nanvector(l_nan-2),:);


    timetodrawn=[];
    jj=0;
    %Find the time, and save the position of these hours.
    for ii=1:l3
        a=time{ii,1};
        hhmmdrawn=a{1,1}(1:5);
        if hourtodrawn==hhmmdrawn
            jj=jj+1;
            timetodrawn(jj)=ii;

        end
    end
    %j number of group of data in this hh:mm, for example 2 when a
measurement
    %each 30 seconds, and save the position
    %if j=0, there are no data to this time
    if jj==0
        errordlg('You must other hour choose','File Error');
    end

    i=[];
    v=[];
    g=[];
    p=[];
    ioutzero=[];
```

```
voutzero=[];
goutzero=[];
poutzero=[];
q=0;
position=0;
%transfer data of a cell to a vector, and plot
position=timetodrawn(1);
pos(k)=position;
l_c=length(I_c{position,1});
for kk=1:l_c
    i(kk)=str2double(strrep(I_c{position,1}(kk,1),',','.'));
    v(kk)=str2double(strrep(V_c{position,1}(kk,1),',','.'));
    g(kk)=str2double(strrep(G_c{position,1}(kk,1),',','.'));

    if i(kk)~=0 && v(kk)~=0
        q=q+1;
        ioutzero(q)=i(kk);
        voutzero(q)=v(kk);
        goutzero(q)=g(kk);
        poutzero(q)=ioutzero(q)*voutzero(q);
    end
end
l_outzero=length(ioutzero);
if k==1
    for pp=1:l_outzero

        I1(pp)=ioutzero(pp);
        V1(pp)=voutzero(pp);
        G1(pp)=goutzero(pp);
        P1(pp)=poutzero(pp);
    end
elseif k==2
    for pp=1:l_outzero

        I2(pp)=ioutzero(pp);
        V2(pp)=voutzero(pp);
        G2(pp)=goutzero(pp);
        P2(pp)=poutzero(pp);
    end
elseif k==3
    for pp=1:l_outzero

        I3(pp)=ioutzero(pp);
        V3(pp)=voutzero(pp);
        G3(pp)=goutzero(pp);
        P3(pp)=poutzero(pp);
    end
elseif k==4
    for pp=1:l_outzero

        I4(pp)=ioutzero(pp);
        V4(pp)=voutzero(pp);
        G4(pp)=goutzero(pp);
        P4(pp)=poutzero(pp);
    end
elseif k==5
```

```
    for pp=1:l_outzero

        I5(pp)=ioutzero(pp);
        V5(pp)=voutzero(pp);
        G5(pp)=goutzero(pp);
        P5(pp)=poutzero(pp);
    end
elseif k==6
    for pp=1:l_outzero

        I6(pp)=ioutzero(pp);
        V6(pp)=voutzero(pp);
        G6(pp)=goutzero(pp);
        P6(pp)=poutzero(pp);
    end
elseif k==7
    for pp=1:l_outzero

        I7(pp)=ioutzero(pp);
        V7(pp)=voutzero(pp);
        G7(pp)=goutzero(pp);
        P7(pp)=poutzero(pp);
    end
elseif k==8
    for pp=1:l_outzero

        I8(pp)=ioutzero(pp);
        V8(pp)=voutzero(pp);
        G8(pp)=goutzero(pp);
        P8(pp)=poutzero(pp);
    end
elseif k==9
    for pp=1:l_outzero

        I9(pp)=ioutzero(pp);
        V9(pp)=voutzero(pp);
        G9(pp)=goutzero(pp);
        P9(pp)=poutzero(pp);
    end
elseif k==10
    for pp=1:l_outzero

        I10(pp)=ioutzero(pp);
        V10(pp)=voutzero(pp);
        G10(pp)=goutzero(pp);
        P10(pp)=poutzero(pp);
    end
elseif k==11
    for pp=1:l_outzero

        I11(pp)=ioutzero(pp);
        V11(pp)=voutzero(pp);
        G11(pp)=goutzero(pp);
        P11(pp)=poutzero(pp);
    end
else
```

```
        for pp=1:l_outzero

            I12(pp)=ioutzero(pp);
            V12(pp)=voutzero(pp);
            G12(pp)=goutzero(pp);
            P12(pp)=poutzero(pp);
        end
    end
    fclose(fid);
end


% for i=1:l
%     s = char(filename{1,i});
%     fid = fopen(s);
%     C_IV = textscan(fid,'%s %s %s ');
%     l=length(C_IV{1,1});
%
%     C_g=strrep(C_IV{1,3},',','.');
%     vector=[];
%     for k=1:l
%         vector(k)=str2double(C_g(k));
%     end
%     f=isnan(vector);
%     nanvector=[];
%     j=1;
%     for h=1:l
%         if true(f(h))
%             nanvector(j)=h;
%             j=j+1;
%         end
%     end
%     l_nan=length(nanvector);
%     l3=l_nan/3;
%     time=cell(l3,1);
%     I_c=cell(l3,1);
%     V_c=cell(l3,1);
%     G_c=cell(l3,1);
%     m=0;
%
%     for n=1:l3-1
%         I_c_new{n,1}=C_IV{1,2}((nanvector(3+m)+1):(nanvector(4+m)-1));
%         V_c_new{n,1}=C_IV{1,1}((nanvector(3+m)+1):(nanvector(4+m)-1));
%         G_c_new{n,1}=C_IV{1,3}((nanvector(3+m)+1):(nanvector(4+m)-1));
%         time{n,1}=C_IV{1,2}(nanvector(1+m),:);
%         m=m+3;
%     end
%     I_c_new{l3,1}=C_IV{1,2}((nanvector(l_nan)+1):l);
%     V_c_new{l3,1}=C_IV{1,1}((nanvector(l_nan)+1):l);
%     G_c_new{l3,1}=C_IV{1,3}((nanvector(l_nan)+1):l);
%     time_new{l3,1}=C_IV{1,2}(nanvector(l_nan-2),:);
%
%     I_c_process=[I_c;I_c_new];
%     I_c=I_c_process;
%     V_c_process=[V_c;V_c_new];
%     V_c=V_c_process;
```

```
%     G_c_process=[G_c;G_c_new];
%     G_c=G_c_process;
%     time_process=[time;time_new];
%     time=time_process;
%     fclose(fid);
% end

function loadparameterdata_Callback(hObject, eventdata, handles)
%Global variables
global Vmpp Impp Pmpp Voc Isc FF G Tmodul Vwind Rwind Tamb rF  Pyr1 Pyr2
Pyr3 Vwind1 Vwind2 Vwind3 SiC1 SiC2 SiC3 SiC4
global Vmpp_all Impp_all Pmpp_all Voc_all Isc_all FF_all G_all Tmodul_all
Vwind_all Rwind_all Tamb_all rF_all  Pyr1_all Pyr2_all Pyr3_all
Vwind1_all Vwind2_all Vwind3_all SiC1_all SiC2_all SiC3_all SiC4_all
global hourtodrawn pos Modul Date_c C_PVMeteodaten
%Load and open file
[filename, pathname, filterindex] = uigetfile( {'*.xls','EXCEL-files
(*.xls)'; '*.txt','Text files (*.txt)'; '*.*', 'All Files (*.*)'}, 'Pick
a file', 'MultiSelect', 'on');
l=length(filename);
Modul= [];
    Impp=[];
    Vmpp=[];
    Pmpp=[];
    Voc=[];
    Isc=[];
    FF=[];
    G=[];
    Tmodul=[];
    Vwind=[];
    Rwind=[];
    Tamb=[];
    rF=[];
    Pyr1=[];
    Pyr2=[];
    Pyr3=[];
    Vwind1=[];
    Vwind2=[];
    Vwind3=[];
    SiC1=[];
    SiC2=[];
    SiC3=[];
    SiC4=[];
    Impp_all=[];
    Vmpp_all=[];
    Pmpp_all=[];
    Voc_all=[];
    Isc_all=[];
    FF_all=[];
    G_all=[];
    Tmodul_all=[];
    Vwind_all=[];
    Rwind_all=[];
    Tamb_all=[];
    rF_all=[];
    Pyr1_all=[];
    Pyr2_all=[];
```

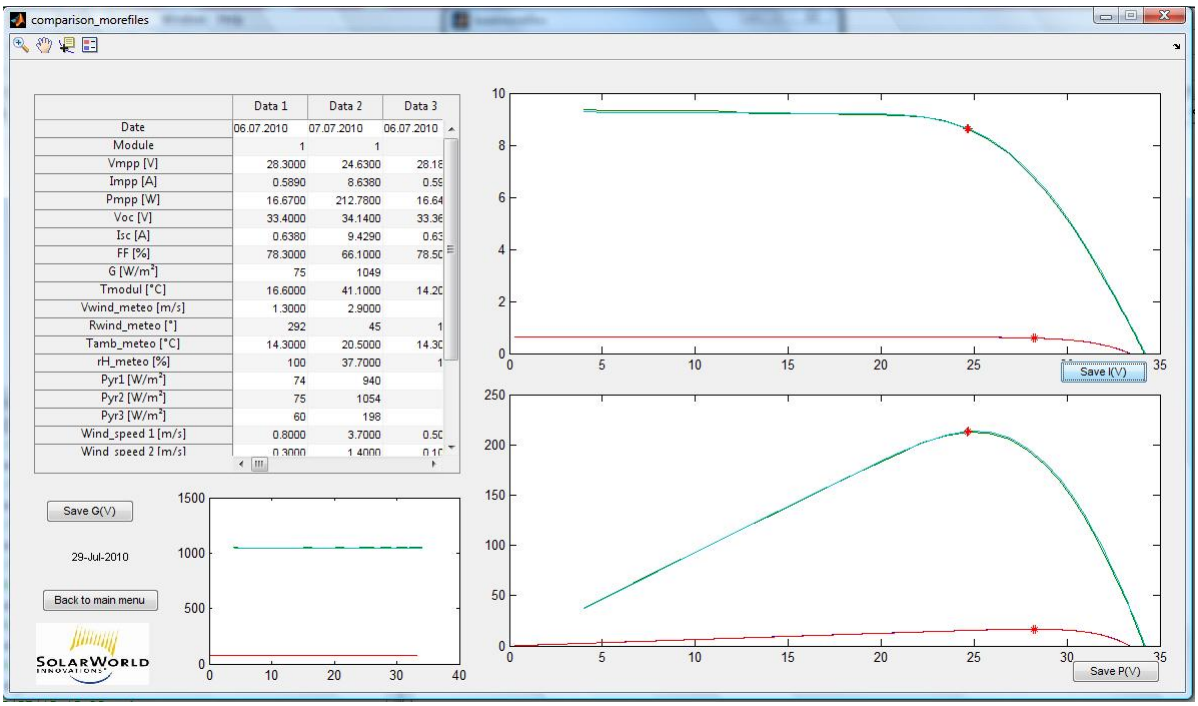
```
    Pyr3_all=[];
    Vwind1_all=[];
    Vwind2_all=[];
    Vwind3_all=[];
    SiC1_all=[];
    SiC2_all=[];
    SiC3_all=[];
    SiC4_all=[];
    Date_c=cell(l,1);
for i=1:l
    s = char(filename{1,i});
    fid = fopen(s);
    %Skip 2 lines.
    C_no=textscan(fid,'%*[^\n]',2);
    %Read data and save in a cell, 27 columns, the data are electrical
and
    %meteorological
    C_PVMeteodaten = textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s');
    %Number of modul 1...12

    Modul(i)=str2double(C_PVMeteodaten{1,1}(1,:));
    Date_c{i,1}=C_PVMeteodaten{1,2}(1,:);

    %How many data, this must be equal to l3.
    l_pvm=length(C_PVMeteodaten{1,1});
    %Vectors to each variable, time is equal to time in curve file
    Vmpp=str2double(strrep(C_PVMeteodaten{1,4},',','.'));
    Impp=str2double(strrep(C_PVMeteodaten{1,5},',','.'));
    Pmpp=str2double(strrep(C_PVMeteodaten{1,6},',','.'));
    Voc=str2double(strrep(C_PVMeteodaten{1,7},',','.'));
    Isc=str2double(strrep(C_PVMeteodaten{1,8},',','.'));
    FF=str2double(strrep(C_PVMeteodaten{1,9},',','.'));
    G=str2double(strrep(C_PVMeteodaten{1,10},',','.'));
    Tmodul=str2double(strrep(C_PVMeteodaten{1,11},',','.'));
    Vwind=str2double(strrep(C_PVMeteodaten{1,14},',','.'));
    Rwind=str2double(strrep(C_PVMeteodaten{1,15},',','.'));
    Tamb=str2double(strrep(C_PVMeteodaten{1,16},',','.'));
    rF=str2double(strrep(C_PVMeteodaten{1,17},',','.'));
    Pyr1=str2double(strrep(C_PVMeteodaten{1,18},',','.'));
    Pyr2=str2double(strrep(C_PVMeteodaten{1,19},',','.'));
    Pyr3=str2double(strrep(C_PVMeteodaten{1,20},',','.'));
    Vwind1=str2double(strrep(C_PVMeteodaten{1,21},',','.'));
    Vwind2=str2double(strrep(C_PVMeteodaten{1,22},',','.'));
    Vwind3=str2double(strrep(C_PVMeteodaten{1,23},',','.'));
    SiC1=str2double(strrep(C_PVMeteodaten{1,24},',','.'));
    SiC2=str2double(strrep(C_PVMeteodaten{1,25},',','.'));
    SiC3=str2double(strrep(C_PVMeteodaten{1,26},',','.'));
    SiC4=str2double(strrep(C_PVMeteodaten{1,27},',','.'));

    position=pos(i);
    Vmpp_all(i)=Vmpp(position);
    Impp_all(i)=Impp(position);
    Pmpp_all(i)=Pmpp(position);
    Voc_all(i)=Voc(position);
    Isc_all(i)=Isc(position);
```

```
    FF_all(i)=FF(position);
    G_all(i)=G(position);
    Tmodul_all(i)=Tmodul(position);
    Vwind_all(i)=Vwind(position);
    Rwind_all(i)=Rwind(position);
    Tamb_all(i)=Tamb(position);
    rF_all(i)=rF(position);
    Pyr1_all(i)=Pyr1(position);
    Pyr2_all(i)=Pyr2(position);
    Pyr3_all(i)=Pyr3(position);
    Vwind1_all(i)=Vwind1(position);
    Vwind2_all(i)=Vwind2(position);
    Vwind3_all(i)=Vwind3(position);
    SiC1_all(i)=SiC1(position);
    SiC2_all(i)=SiC2(position);
    SiC3_all(i)=SiC3(position);
    SiC4_all(i)=SiC4(position);
    fclose(fid);
end

function evaluatebutton_Callback(hObject, eventdata, handles)
comparison_morefiles

function filesnumber_Callback(hObject, eventdata, handles)

function filesnumber_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function hour_Callback(hObject, eventdata, handles)

function hour_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function varargout = load1file(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @load1file_OpeningFcn, ...
                   'gui_OutputFcn',  @load1file_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
function load1file_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
%put logotype
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = load1file_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function loadcurvedata_Callback(hObject, eventdata, handles)
%Global variables
global G_c I_c V_c time date l3
%Load and open file
handles.fileName = uigetfile('*.txt');
fileName = handles.fileName;
fid = fopen(fileName);
%Read data and save in a cell
%C_IV{1,1}=V;C_IV{1,2}=I;C_IV{1,3}=G; (C_...=cell)
C_IV = textscan(fid,'%s %s %s ');
%l=amount of data
l=length(C_IV{1,1});
%Change comma to point in Irradiance vector, because Matlab read numbers
with point.
C_g=strrep(C_IV{1,3},',','.');
vector=[];
%Convert strings into numbers of the Irradiance vector
for i=1:l
    vector(i)=str2double(C_g(i));
end
%Find the Not a Number (NaN) and return a logical vector.
f=isnan(vector);
nanvector=[];
j=1;
%Find the position of NaNs and make a vector to save the positions.
for h=1:l
    if true(f(h))
        nanvector(j)=h;
        j=j+1;
    end
end
%how many NaN are there
l_nan=length(nanvector);
%To each group of data there are 3 NaN. So we know how many group of data
%there are, and so how many measurements of I-V data in a day.
l3=l_nan/3;
%Initialize cell for time, I, V and G data (..._c=cell)
time=cell(l3,1);
I_c=cell(l3,1);
V_c=cell(l3,1);
G_c=cell(l3,1);
%m = counter for counting with the 3 NaN of each group of data.
m=0;
%date is in these place in file.
```

```
date=C_IV{1,1}(nanvector(1),:);
%A cell for each Data (I,V,G,time), n sets the place of each group of
data
%in Cell.
for n=1:l3-1
    I_c{n,1}=C_IV{1,2}((nanvector(3+m)+1):(nanvector(4+m)-1));
    V_c{n,1}=C_IV{1,1}((nanvector(3+m)+1):(nanvector(4+m)-1));
    G_c{n,1}=C_IV{1,3}((nanvector(3+m)+1):(nanvector(4+m)-1));
    time{n,1}=C_IV{1,2}(nanvector(1+m),:);
    m=m+3;
end
I_c{l3,1}=C_IV{1,2}((nanvector(l_nan)+1):l);
V_c{l3,1}=C_IV{1,1}((nanvector(l_nan)+1):l);
G_c{l3,1}=C_IV{1,3}((nanvector(l_nan)+1):l);
time{l3,1}=C_IV{1,2}(nanvector(l_nan-2),:);
fclose(fid);

function loadparameterdata_Callback(hObject, eventdata, handles)
%Global variables
global Vmpp Impp Pmpp Voc Isc FF G Tmodul Vwind Rwind Tamb rF  Pyr1 Pyr2
Pyr3 Vwind1 Vwind2 Vwind3 SiC1 SiC2 SiC3 SiC4 Modul Date_c
%Load and open file
handles.fileName = uigetfile('*.txt');
fileName = handles.fileName;
fid = fopen(fileName);
%Skip 2 lines.
C_no=textscan(fid,'%*[^\n]',2);
%Read data and save in a cell, 27 columns, the data are electrical and
%meteorological
C_PVMeteodaten = textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s');
%Number of modul 1...12
Modul=str2double(C_PVMeteodaten{1,1}(1,:));
Date_c{1,1}=C_PVMeteodaten{1,2}(1,:);
%How many data, this must be equal to l3.
l_pvm=length(C_PVMeteodaten{1,1});
Vmpp=0;
Impp=0;
Pmpp=0;
%Vectors to each variable, time is equal to time in curve file
Vmpp=str2double(strrep(C_PVMeteodaten{1,4},',','.'));
Impp=str2double(strrep(C_PVMeteodaten{1,5},',','.'));
Pmpp=str2double(strrep(C_PVMeteodaten{1,6},',','.'));
Voc=str2double(strrep(C_PVMeteodaten{1,7},',','.'));
Isc=str2double(strrep(C_PVMeteodaten{1,8},',','.'));
FF=str2double(strrep(C_PVMeteodaten{1,9},',','.'));
G=str2double(strrep(C_PVMeteodaten{1,10},',','.'));
Tmodul=str2double(strrep(C_PVMeteodaten{1,11},',','.'));
Vwind=str2double(strrep(C_PVMeteodaten{1,14},',','.'));
Rwind=str2double(strrep(C_PVMeteodaten{1,15},',','.'));
Tamb=str2double(strrep(C_PVMeteodaten{1,16},',','.'));
rF=str2double(strrep(C_PVMeteodaten{1,17},',','.'));
Pyr1=str2double(strrep(C_PVMeteodaten{1,18},',','.'));
Pyr2=str2double(strrep(C_PVMeteodaten{1,19},',','.'));
Pyr3=str2double(strrep(C_PVMeteodaten{1,20},',','.'));
Vwind1=str2double(strrep(C_PVMeteodaten{1,21},',','.'));
Vwind2=str2double(strrep(C_PVMeteodaten{1,22},',','.'));
```

```
Vwind3=str2double(strrep(C_PVMeteodaten{1,23},',','.'));
SiC1=str2double(strrep(C_PVMeteodaten{1,24},',','.'));
SiC2=str2double(strrep(C_PVMeteodaten{1,25},',','.'));
SiC3=str2double(strrep(C_PVMeteodaten{1,26},',','.'));
SiC4=str2double(strrep(C_PVMeteodaten{1,27},',','.'));
fclose(fid);

function evaluatebutton_Callback(hObject, eventdata, handles)
writehour
```


comparison_morefiles
Data 1
Data 2
Data 3
Date
06.07.2010
07.07.2010
06.07.2010
Module
Vmpp [V]
Impp [A]
Pmpp [W]
Voc [V]
Isc [A]
FF [%]
G [W/m²]
Tmodul [°C]
Vwind_meteo [m/s]
Rwind_meteo [°]
Tamb_meteo [°C]
rH_meteo [%]
Pyr1 [W/m²]
Pyr2 [W/m²]
Pyr3 [W/m²]
Wind_speed 1 [m/s]
Save G(V)
29-Jul-2010
Back to main menu
SOLARWORLD
Save I(V)
Save P(V)

```
function varargout = comparison_morefiles(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @comparison_morefiles_OpeningFcn,
...
                   'gui_OutputFcn',  @comparison_morefiles_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function comparison_morefiles_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
% declare global variables
global I1 P1 V1 G1 I2 P2 V2 G2 I3 P3 V3 G3 I4 P4 V4 G4 I5 P5 V5 G5 I6 P6
V6 G6 I7 P7 V7 G7 I8 P8 V8 G8 I9 P9 V9 G9 I10 P10 V10 G10 I11 P11 V11 G11
I12 P12 V12 G12
global Modul Date_c  C_PVMeteodaten lpos vmpp impp pmpp
global  Vmpp_all Impp_all Pmpp_all Voc_all Isc_all FF_all G_all
Tmodul_all Vwind_all Rwind_all Tamb_all rF_all  Pyr1_all Pyr2_all
Pyr3_all Vwind1_all Vwind2_all Vwind3_all SiC1_all SiC2_all SiC3_all
SiC4_all pos
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
set(handles.dayofevaluation, 'String', date)
lpos=length(pos);
I_matrix=[];
V_matrix=[];
G_matrix=[];
P_matrix=[];
datatable=cell(lpos,25);
% i1=[];
% v1=[];
% p1=[];
% g1=[];
% i2=[];
% v2=[];
% p2=[];
% g2=[];
% i3=[];
% v3=[];
% p3=[];
% g3=[];
% i4=[];
% v4=[];
```

```
% p4=[];
% g4=[];
% i5=[];
% v5=[];
% p5=[];
% g5=[];
% i6=[];
% v6=[];
% p6=[];
% g6=[];
% i7=[];
% v7=[];
% p7=[];
% g7=[];
% i8=[];
% v8=[];
% p8=[];
% g8=[];
% i9=[];
% v9=[];
% p9=[];
% g9=[];
% i10=[];
% v10=[];
% p10=[];
% g10=[];
% i11=[];
% v11=[];
% p11=[];
% g11=[];
% i12=[];
% v12=[];
% p12=[];
% g12=[];
vmpp=[];
impp=[];
pmpp=[];
%set data in a cell
for j=1:lpos
%     a=Date_c{j,1};
%     datatable{1,j}=a;
    a=datenum(Date_c{j,1}(1,:),'dd.mm.yyyy');
    datatable{1,j}=datestr(a,'dd.mm.yyyy');
    datatable{2,j}=Modul(j);
    datatable{3,j}=Vmpp_all(j);
    vmpp(j)=Vmpp_all(j);
    datatable{4,j}=Impp_all(j);
    impp(j)=Impp_all(j);
    datatable{5,j}=Pmpp_all(j);
     pmpp(j)=Pmpp_all(j);
    datatable{6,j}=Voc_all(j);
    datatable{7,j}=Isc_all(j);
    datatable{8,j}=FF_all(j);
    datatable{9,j}=G_all(j);
    datatable{10,j}=Tmodul_all(j);
    datatable{11,j}=Vwind_all(j);
    datatable{12,j}=Rwind_all(j);
```

```
    datatable{13,j}=Tamb_all(j);
    datatable{14,j}=rF_all(j);
    datatable{15,j}=Pyr1_all(j);
    datatable{16,j}=Pyr2_all(j);
    datatable{17,j}=Pyr3_all(j);
    datatable{18,j}=Vwind1_all(j);
    datatable{19,j}=Vwind2_all(j);
    datatable{20,j}=Vwind3_all(j);
    datatable{21,j}=SiC1_all(j);
    datatable{22,j}=SiC2_all(j);
    datatable{23,j}=SiC3_all(j);
    datatable{24,j}=SiC4_all(j);
     Pmpp_parameter=Pmpp_all(j);
    G_parameter=G_all(j);
    nu=1-(Pmpp_parameter/(1.67601*G_parameter));
    datatable{25,j}=nu;

end
%plot a maximal of 12 groups of data
if lpos==1
    plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*')
    plot(handles.V_I, V1,I1,vmpp(1),impp(1),'r*')

elseif lpos==11
plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*',V11,I11,vmpp(11),i
mpp(11),'r*')
plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8)
,pmpp(8),'r*',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*',V
11,P11,vmpp(11),pmpp(11),'r*')
elseif lpos==10
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*')
plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8)
,pmpp(8),'r*',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*')
elseif lpos==9

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8)
,pmpp(8),'r*',V9,P9,vmpp(9),pmpp(9),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*')
elseif lpos==8
```

```
plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8)
,pmpp(8),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*')
elseif lpos==7

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*')
elseif lpos==6

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*')
elseif lpos==5

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*')

elseif lpos==4

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*')
elseif lpos==3

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*')
elseif lpos==2

plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*')
    plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*')
else
```

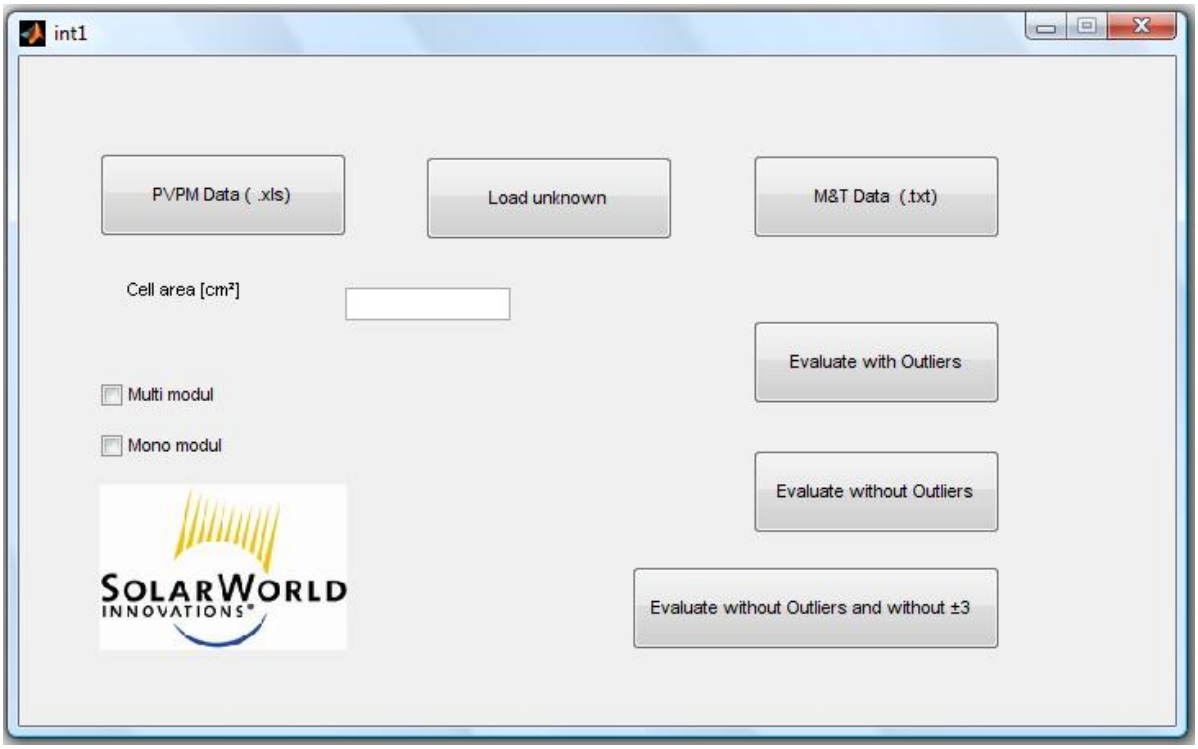
```
plot(handles.V_I,
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*',V11,I11,vmpp(11),i
mpp(11),'r*',V12,I12,vmpp(12),impp(12),'r*')
plot(handles.V_P,V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3
,P3,vmpp(3),pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5)
,'r*',V6,P6,vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8)
,pmpp(8),'r*',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*',V
11,P11,vmpp(11),pmpp(11),'r*',V12,P12,vmpp(12),pmpp(12),'r*')
end

plot(handles.G_V,
V1,G1,V2,G2,V3,G3,V4,G4,V5,G5,V6,G6,V7,G7,V8,G8,V9,G9,V10,G10,V11,G11,V12
,G12)
%set the cell of data in the table of the interface
set(handles.alldatatable,'Data',datatable)

function varargout = comparison_morefiles_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function backbutton_Callback(hObject, eventdata, handles)
close pri_curvaIV
close loadmorefiles
main_menu

function saveivbutton_Callback(hObject, eventdata, handles)
global I1 P1 V1 G1 I2 P2 V2 G2 I3 P3 V3 G3 I4 P4 V4 G4 I5 P5 V5 G5 I6 P6
V6 G6 I7 P7 V7 G7 I8 P8 V8 G8 I9 P9 V9 G9 I10 P10 V10 G10 I11 P11 V11 G11
I12 P12 V12 G12
global lpos vmpp impp pmpp
%open a new figure
figure
%draw in this figure this plot IV, the same to PV and GV
if lpos==1
        plot( V1,I1,vmpp(1),impp(1),'r*')
elseif lpos==11
plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*',V11,I11,vmpp(11),i
mpp(11),'r*')
elseif lpos==10
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*')
elseif lpos==9
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
```

```
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*')
elseif lpos==8
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*')
elseif lpos==7
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*')
elseif lpos==6
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*')
elseif lpos==5

plot(V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),
impp(3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*')

elseif lpos==4

plot(V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),
impp(3),'r*',V4,I4,vmpp(4),impp(4),'r*')
elseif lpos==3
    plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*')
elseif lpos==2
    plot( V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*')
else
plot(
V1,I1,vmpp(1),impp(1),'r*',V2,I2,vmpp(2),impp(2),'r*',V3,I3,vmpp(3),impp(
3),'r*',V4,I4,vmpp(4),impp(4),'r*',V5,I5,vmpp(5),impp(5),'r*',V6,I6,vmpp(
6),impp(6),'r*',V7,I7,vmpp(7),impp(7),'r*',V8,I8,vmpp(8),impp(8),'r*',V9,
I9,vmpp(9),impp(9),'r*',V10,I10,vmpp(10),impp(10),'r*',V11,I11,vmpp(11),i
mpp(11),'r*',V12,I12,vmpp(12),impp(12),'r*')
end
xlabel('V [V]')
ylabel('I [A]')
title('Curve I-V')

function savepvbutton_Callback(hObject, eventdata, handles)
global I1 P1 V1 G1 I2 P2 V2 G2 I3 P3 V3 G3 I4 P4 V4 G4 I5 P5 V5 G5 I6 P6
V6 G6 I7 P7 V7 G7 I8 P8 V8 G8 I9 P9 V9 G9 I10 P10 V10 G10 I11 P11 V11 G11
I12 P12 V12 G12
global lpos vmpp impp pmpp
figure
if lpos==1
    plot(V1,P1,vmpp(1),pmpp(1),'r*')
elseif lpos==11
plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8),pmpp(8),'r*
```

```
',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*',V11,P11,vmpp(
11),pmpp(11),'r*')
elseif lpos==10
plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8),pmpp(8),'r*
',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*')
elseif lpos==9

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8),pmpp(8),'r*
',V9,P9,vmpp(9),pmpp(9),'r*')
elseif lpos==8

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8),pmpp(8),'r*
')
elseif lpos==7

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*')
elseif lpos==6

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*')
elseif lpos==5

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*')

elseif lpos==4

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*')
elseif lpos==3

plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*')

elseif lpos==2
    plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*')


else
plot(V1,P1,vmpp(1),pmpp(1),'r*',V2,P2,vmpp(2),pmpp(2),'r*',V3,P3,vmpp(3),
pmpp(3),'r*',V4,P4,vmpp(4),pmpp(4),'r*',V5,P5,vmpp(5),pmpp(5),'r*',V6,P6,
vmpp(6),pmpp(6),'r*',V7,P7,vmpp(7),pmpp(7),'r*',V8,P8,vmpp(8),pmpp(8),'r*
',V9,P9,vmpp(9),pmpp(9),'r*',V10,P10,vmpp(10),pmpp(10),'r*',V11,P11,vmpp(
11),pmpp(11),'r*',V12,P12,vmpp(12),pmpp(12),'r*')
end
xlabel('V [V]')
ylabel('P [W]')
```

```
title('Curve P-V')

function savegvbutton_Callback(hObject, eventdata, handles)
global I1 P1 V1 G1 I2 P2 V2 G2 I3 P3 V3 G3 I4 P4 V4 G4 I5 P5 V5 G5 I6 P6
V6 G6 I7 P7 V7 G7 I8 P8 V8 G8 I9 P9 V9 G9 I10 P10 V10 G10 I11 P11 V11 G11
I12 P12 V12 G12
figure
plot(
V1,G1,V2,G2,V3,G3,V4,G4,V5,G5,V6,G6,V7,G7,V8,G8,V9,G9,V10,G10,V11,G11,V12
,G12)

xlabel('V [V]')
ylabel('G [W/m²]')
title('Curve G-V')
```

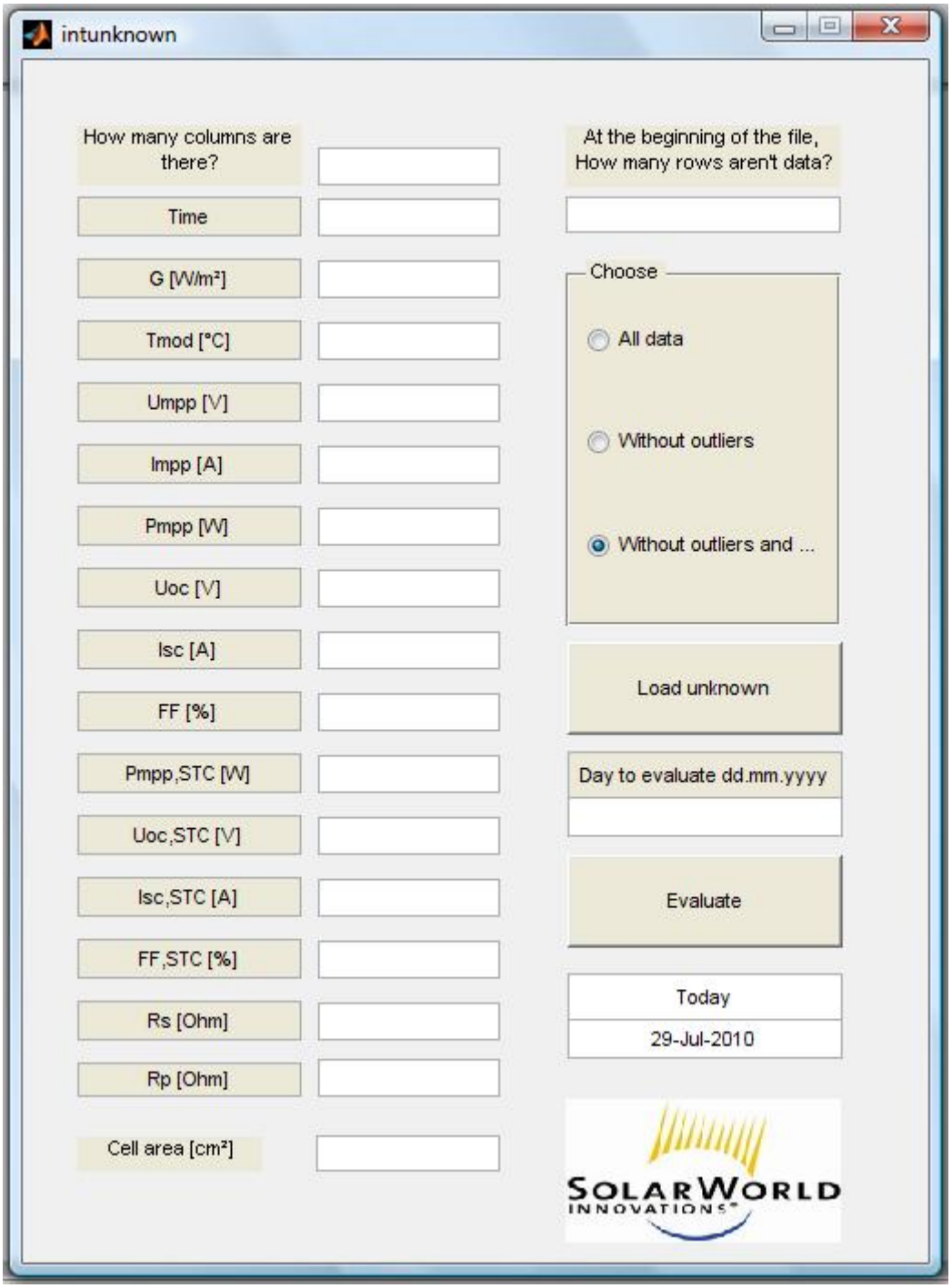
F.3. Evaluation Modul



```
function varargout = int1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int1_OpeningFcn, ...
                   'gui_OutputFcn',  @int1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = int1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function evaluate_Callback(hObject, eventdata, handles)
global MDM titMDM c_a C_datos fid car_xls
%go to evaluation without outliers of xls when car_xls=1 or to evaluation
%without outliers of a file of MEncke&Tegtmeyer
if car_xls==1
    int2wo
else
    int2wo_txt
end

function cargaxls_Callback(hObject, eventdata, handles)
%load the data of xls, and save it in a Matrix MDM and the data, that
%characters are in a Matrix of characters.
global MDM titMDM car_xls
handles.fileName = uigetfile('*.xls');
guidata(hObject, handles) ;
fileName = handles.fileName;
MDM=[];
titMDM=[];
[MDM,titMDM]=xlsread(fileName);
car_xls=1;

function cargatxt_Callback(hObject, eventdata, handles)
global C_datos MDM fid  car_xls
%load Data of M&T files and read them, than save them in a cell C_datos
handles.fileName = uigetfile('*.txt');
```

```
guidata(hObject, handles);
fileName = handles.fileName;
fid = fopen(fileName);
C_text=textscan(fid,'%*[^\n]',2);
C_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s ');
% length of C_datos, so how many data are there
l=length(C_datos{1,1});
C_datosp=zeros(l,12);
MDMtxt=[];
MDM=[];
%save all the data from 4 column to 11 column, the electrical parameters,
%in a Matrix, and change ',' to '.' and so identify as numbers, and
convert
%them into a number
for i=1:l
for j=4:11
MDMtxt(i,j-3)=str2num(strrep(C_datos{1,j}{i,1},',','.'));
end
end
fclose(fid);
MDM=MDMtxt;
car_xls=0;

function area_Callback(hObject, eventdata, handles)
%get the cell area
global c_a
b=get(handles.area,'String');
c=strrep(b, ',','.');
c_a=str2num(c);

function area_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function evaluate_co_Callback(hObject, eventdata, handles)
global MDM titMDM c_a C_datos fid car_xls
%when click in all data
if car_xls==1
    int2ad
else
    int2ad_txt
end

function evaluate_sosds_Callback(hObject, eventdata, handles)
global MDM titMDM c_a C_datos fid car_xls
%when click in without outliers and without 3*Standard Desviation
if car_xls==1
    int2wowsd
else
    int2wowsd_txt
end

function unknownload_Callback(hObject, eventdata, handles)
```

```
%when click in load unknown
close int1
intunknown

function multibox_Callback(hObject, eventdata, handles)
% Value to multicristaline modul
global TKisc TKvoc TKpmpp
TKisc=0.034;
TKvoc=-0.34;
TKpmpp=-0.48;

function monobox_Callback(hObject, eventdata, handles)
%Values to monocristaline modul
global TKisc TKvoc TKpmpp
TKisc=0.042;
TKvoc=-0.33;
TKpmpp=-0.45;
```


intunknown
How many columns are there?
Time
G [W/m²]
Tmod [°C]
Umpp [V]
Impp [A]
Pmpp [W]
Uoc [V]
Isc [A]
FF [%]
Pmpp,STC [W]
Uoc,STC [V]
Isc,STC [A]
FF,STC [%]
Rs [Ohm]
Rp [Ohm]
Cell area [cm²]
At the beginning of the file, How many rows aren't data?
Choose
All data
Without outliers
Without outliers and ...
Load unknown
Day to evaluate dd.mm.yyyy
Evaluate
Today
29-Jul-2010
SOLARWORLD
INNOVATIONS

```
function varargout = intunknown(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @intunknown_OpeningFcn, ...
                   'gui_OutputFcn',  @intunknown_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function intunknown_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global hoy
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
hoy=date;
set(handles.today,'String',hoy);

function varargout = intunknown_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function loadunk_Callback(hObject, eventdata, handles)
%take the file, and open
global c_a r_o c_time c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 c_9 c_10 c_11 c_12
c_13 c_14 n_c fid
handles.fileName = uigetfile('*.*');
fileName = handles.fileName;
fid = fopen(fileName,'r');

function areau_Callback(hObject, eventdata, handles)
global c_a
%take area, and convert it in a number.
b=get(handles.areau,'String');
c=strrep(b, ',','.');
c_a=str2num(c);

function areau_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

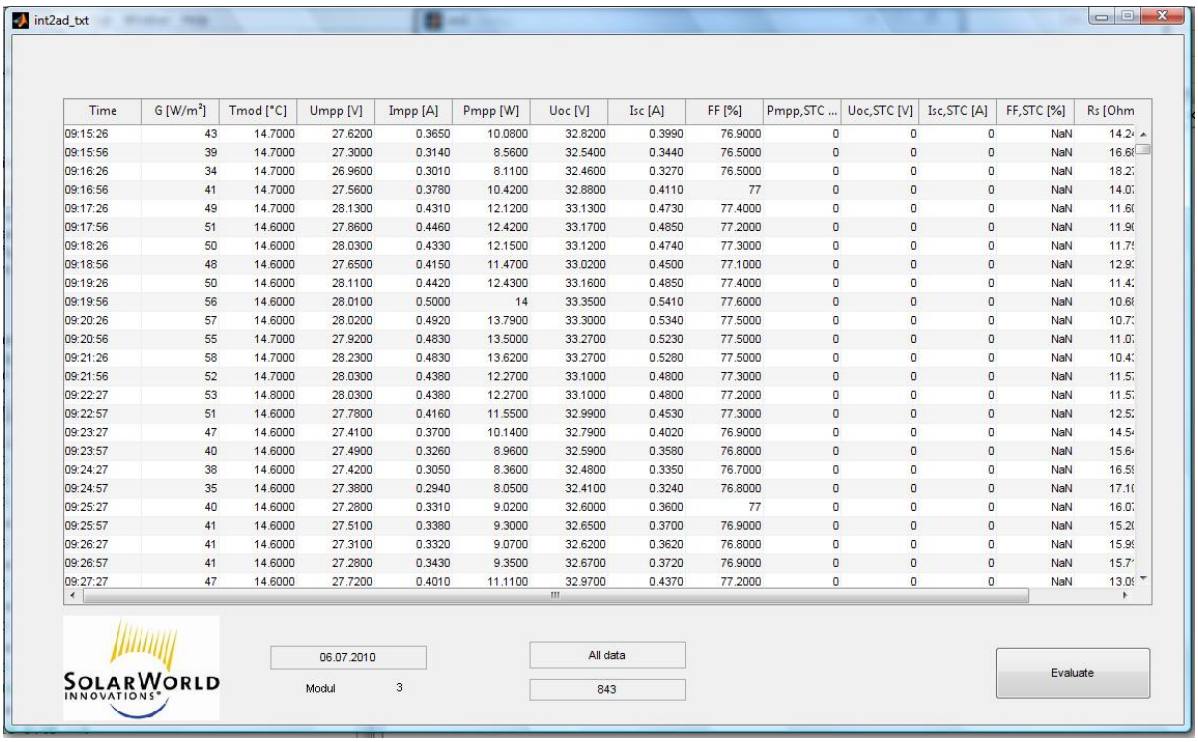
```
function coltime_Callback(hObject, eventdata, handles)
%take the number of time column
global c_time
c_time=str2num(get(handles.coltime, 'String'));

function coltime_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col1_Callback(hObject, eventdata, handles)
%take the number of Radiation
global c_1
c_1=str2num(get(handles.col1, 'String'));

function col1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col2_Callback(hObject, eventdata, handles)
%take the number of Temperature Modul
global c_2
c_2=str2num(get(handles.col2, 'String'));

function col2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col3_Callback(hObject, eventdata, handles)
% take the number of Umpp column
global c_3
c_3=str2num(get(handles.col3, 'String'));

function col3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col4_Callback(hObject, eventdata, handles)
% take the number of Impp column
global c_4
c_4=str2num(get(handles.col4, 'String'));

function col4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function col5_Callback(hObject, eventdata, handles)
% take the number of Pmpp column
global c_5
c_5=str2num(get(handles.col5, 'String'));

function col5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col6_Callback(hObject, eventdata, handles)
% take the number of Uoc column
global c_6
c_6=str2num(get(handles.col6, 'String'));

function col6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col7_Callback(hObject, eventdata, handles)
% take the number of Isc column
global c_7
c_7=str2num(get(handles.col7, 'String'));

function col7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col8_Callback(hObject, eventdata, handles)
% take the number of FF column
global c_8
c_8=str2num(get(handles.col8, 'String'));

function col8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col9_Callback(hObject, eventdata, handles)
% take the number of Pmpp,STC column
global c_9
c_9=str2num(get(handles.col9, 'String'));

function col9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end

function col10_Callback(hObject, eventdata, handles)
% take the number of Uoc,STC column
global c_10
c_10=str2num(get(handles.col10, 'String'));

function col10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col11_Callback(hObject, eventdata, handles)
% take the number of Isc,STC column
global c_11
c_11=str2num(get(handles.col11, 'String'));

function col11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col12_Callback(hObject, eventdata, handles)
% take the number of FF,STC column
global c_12
c_12=str2num(get(handles.col12, 'String'));

function col12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col13_Callback(hObject, eventdata, handles)
% take the number of Rs column
global c_13
c_13=str2num(get(handles.col13, 'String'));

function col13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function col14_Callback(hObject, eventdata, handles)
% take the number of Rp column
global c_14
c_14=str2num(get(handles.col14, 'String'));

function col14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end

function rowout_Callback(hObject, eventdata, handles)
% take the number of rows , that no data have.
global r_o
r_o=str2num(get(handles.rowout, 'String'));

function rowout_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_CreateFcn(hObject, eventdata, handles)

function num_col_Callback(hObject, eventdata, handles)
global n_c
n_c=str2num(get(handles.num_col, 'String'));

function num_col_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
% choose the option
global process
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'buttonad'
        % to evaluate all data
       process=1;
    case 'buttonwo'
        % Code for when radiobutton2 is selected.
        % to evaluate without outliers
        process=2;
    case 'buttonwowsd'
        % to evaluate without outliers and standard desviation
        process=3;
        % Code for when togglebutton1 is selected.
end

function dayev_Callback(hObject, eventdata, handles)
% take the day of evaluation
global d_ev
d_ev=get(handles.dayev, 'String');

function dayev_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit23_Callback(hObject, eventdata, handles)
```

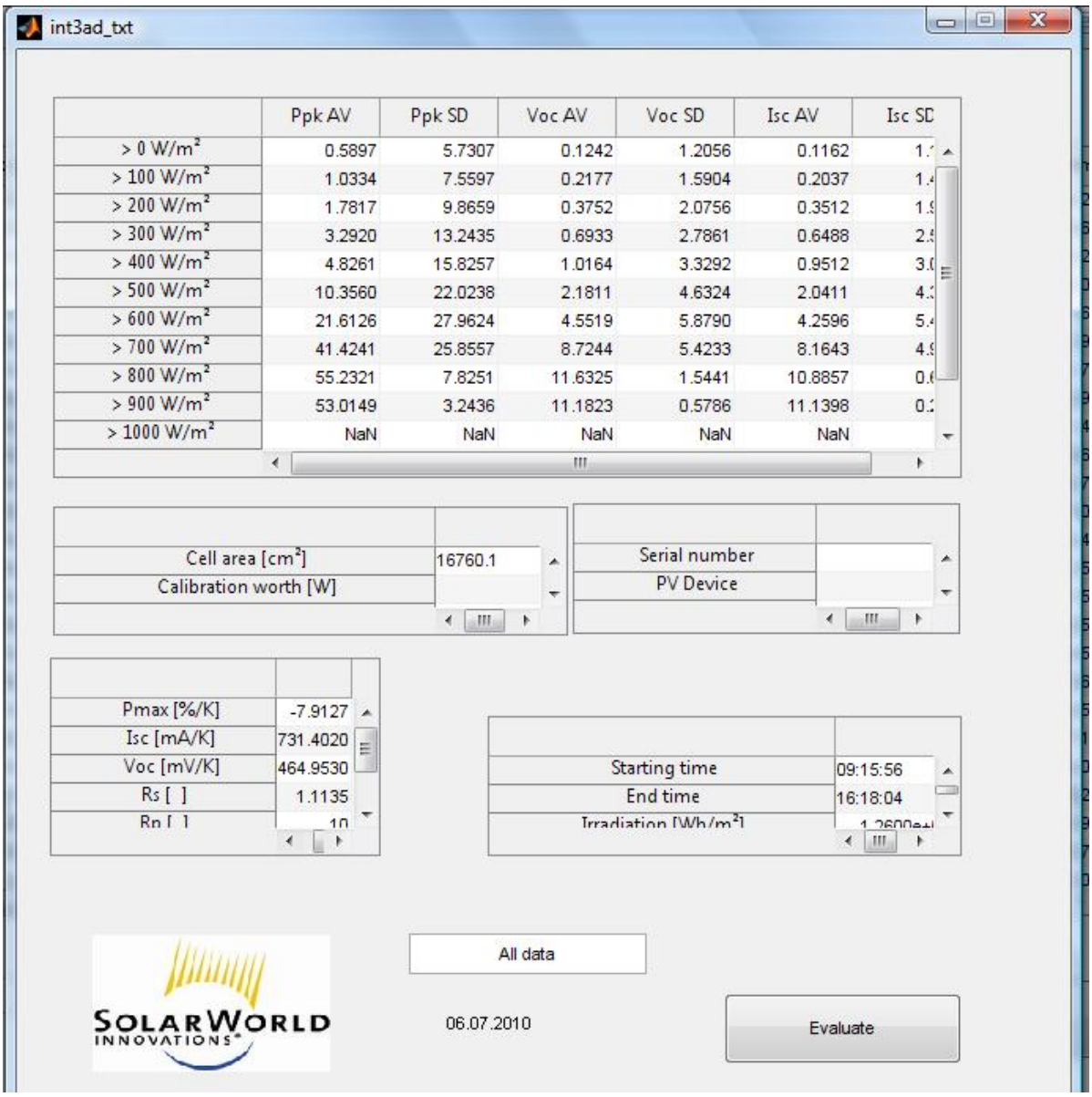
```
function edit23_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function today_Callback(hObject, eventdata, handles)

function today_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function evaluate_button_Callback(hObject, eventdata, handles)
global c_a r_o c_time c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 c_9 c_10 c_11 c_12
c_13 c_14 n_c process d_ev
% Go to differents interfaces
if process==1
    % all data
    int2ad_u
elseif process==2
    % without outliers
    int2wo_u
else
    % without outliers and standard desviation
    int2wowsd_u
end
```


int2ad_txt
Time
G [W/m²]
Tmod [°C]
Umpp [V]
Impp [A]
Pmpp [W]
Uoc [V]
Isc [A]
FF [%]
Pmpp,STC ...
Uoc,STC [V]
Isc,STC [A]
FF,STC [%]
Rs [Ohm
09:15:26 43 14.7000 27.6200 0.3650 10.0800 32.8200 0.3990 76.9000 0 0 0 NaN
09:15:56 39 14.7000 27.3000 0.3140 8.5600 32.5400 0.3440 76.5000 0 0 0 NaN
09:16:26 34 14.7000 26.9600 0.3010 8.1100 32.4600 0.3270 76.5000 0 0 0 NaN
09:16:56 41 14.7000 27.5600 0.3780 10.4200 32.8800 0.4110 77 0 0 0 NaN
09:17:26 49 14.7000 28.1300 0.4310 12.1200 33.1300 0.4730 77.4000 0 0 0 NaN
09:17:56 51 14.6000 27.8600 0.4460 12.4200 33.1700 0.4850 77.2000 0 0 0 NaN
09:18:26 50 14.6000 28.0300 0.4330 12.1500 33.1200 0.4740 77.3000 0 0 0 NaN
09:18:56 48 14.6000 27.6500 0.4150 11.4700 33.0200 0.4500 77.1000 0 0 0 NaN
09:19:26 50 14.6000 28.1100 0.4420 12.4300 33.1600 0.4850 77.4000 0 0 0 NaN
09:19:56 56 14.6000 28.0100 0.5000 14 33.3500 0.5410 77.6000 0 0 0 NaN
09:20:26 57 14.6000 28.0200 0.4920 13.7900 33.3000 0.5340 77.5000 0 0 0 NaN
09:20:56 55 14.7000 27.9200 0.4830 13.5000 33.2700 0.5230 77.5000 0 0 0 NaN
09:21:26 58 14.7000 28.2300 0.4830 13.6200 33.2700 0.5280 77.5000 0 0 0 NaN
09:21:56 52 14.7000 28.0300 0.4380 12.2700 33.1000 0.4800 77.3000 0 0 0 NaN
09:22:27 53 14.8000 28.0300 0.4380 12.2700 33.1000 0.4800 77.2000 0 0 0 NaN
09:22:57 51 14.6000 27.7800 0.4160 11.5500 32.9900 0.4530 77.3000 0 0 0 NaN
09:23:27 47 14.6000 27.4100 0.3700 10.1400 32.7900 0.4020 76.9000 0 0 0 NaN
09:23:57 40 14.6000 27.4900 0.3260 8.9600 32.5900 0.3580 76.8000 0 0 0 NaN
09:24:27 38 14.6000 27.4200 0.3050 8.3600 32.4800 0.3350 76.7000 0 0 0 NaN
09:24:57 35 14.6000 27.3800 0.2940 8.0500 32.4100 0.3240 76.8000 0 0 0 NaN
09:25:27 40 14.6000 27.2800 0.3310 9.0200 32.6000 0.3600 77 0 0 0 NaN
09:25:57 41 14.6000 27.5100 0.3380 9.3000 32.6500 0.3700 76.9000 0 0 0 NaN
09:26:27 41 14.6000 27.3100 0.3320 9.0700 32.6200 0.3620 76.8000 0 0 0 NaN
09:26:57 41 14.6000 27.2800 0.3430 9.3500 32.6700 0.3720 76.9000 0 0 0 NaN
09:27:27 47 14.6000 27.7200 0.4010 11.1100 32.9700 0.4370 77.2000 0 0 0 NaN
SOLARWORLD INNOVATIONS
06.07.2010
Modul 3
All data
843
Evaluate

```
function varargout = int2wo(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int2wo_OpeningFcn, ...
                   'gui_OutputFcn',  @int2wo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int2wo_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global MDM MDMi titMDM titchMDM fechaMDM horai horaf C_datos fid car_xls
Irr
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
%first of all, to put in order the matrix, there is a funktion to put in
%order xls matrix
    MDM = ordenxls(MDM);
    %Save the raw data
    MDMi=MDM;
    %take the length of the first column
    l=length(MDM(:,1));
    t=zeros(l,19);
    %take the characters, and make it text
    for i=3:l+2
        t(i-2,:)=titMDM{i,2};
    end
    %text to date
    tcdate=char(t(:,1:10));
    %text to hous
    tchour=char(t(:,11:end));
    horai=tchour(2,:);
    horaf=tchour(l,:);
    %and make the time in a vector of numbers
    horan=datenum(tchour,'HH:MM:SS');
    valor=zeros(l-1);
    % calculation of Irradiation
    for i=2:l
     valor(i-1)=(horan(i)-horan(i-1))*MDM(i-1,1);
    end
    Irr=sum(valor(:,1))*24;
    % take the date
    fechaMDM=tcdate(1,:);
    % set it in GUIDE
    set(handles.date, 'String', fechaMDM)
```

```
    % Call to function to remove the outliers, this will be the same to
    % without outliers and without 3*Standard desviation, but for all
data
    % there is no function because this prozess takes all the data .
[MDM, titchMDM] = outlierRemoval(MDM, tchour);
%the new length without outliers
l_out=length(MDM(:,1));
%when without outliers there isn´t data, than
if l_out==0
errordlg('You must another file choose, there are not enough data','Bad
Input','modal')
uicontrol(hObject)
return
end
%make a cell
num_elem=cell(l_out,15);
%set the first amount of data and the end amount of data
set(handles.inicial, 'String', l)
set(handles.final, 'String', l_out)
%set the data in a cell num_elem, to set the data in the table of guide
for i2=1:l_out
    for j=2:15
    num_elem{i2,j}=MDM(i2,j-1);
    end
 num_elem{i2,1}=titchMDM(i2,:);
end
set(handles.tablaseg,'Data',num_elem)

function varargout = int2wo_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function evaluate_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM
int3wo

function tablaseg_CellEditCallback(hObject, eventdata, handles)
% % global MDM titMDM titchMDM fechaMDM
% % l_out=length(MDM(:,1));
% % num_elem=cell(l_out,15);
% %
% % for i=1:20
% %     num_elem{i,1}=titchMDM(i,1);
% %     for j=2:15
% %         num_elem{i,j}=MDM(i,j-1);
% %     end
% % end
% % set(handles.tabla,'Data', num_elem)

function final_Callback(hObject, eventdata, handles)

function final_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

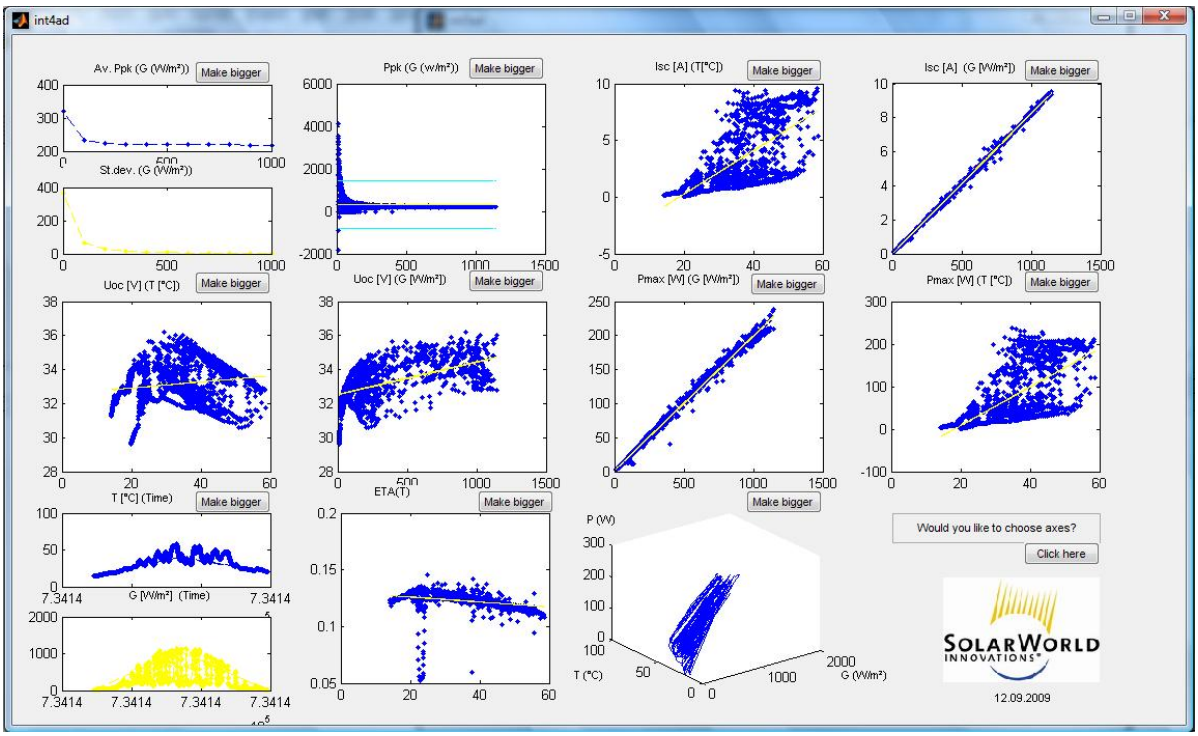
```
function inicial_Callback(hObject, eventdata, handles)

function inicial_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```


int3ad_txt
Ppk AV
Ppk SD
Voc AV
Voc SD
Isc AV
Isc SD
> 0 W/m²
0.5897
5.7307
0.1242
1.2056
0.1162
> 100 W/m²
1.0334
7.5597
0.2177
1.5904
0.2037
> 200 W/m²
1.7817
9.8659
0.3752
2.0756
0.3512
> 300 W/m²
3.2920
13.2435
0.6933
2.7861
0.6488
> 400 W/m²
4.8261
15.8257
1.0164
3.3292
0.9512
> 500 W/m²
10.3560
22.0238
2.1811
4.6324
2.0411
> 600 W/m²
21.6126
27.9624
4.5519
5.8790
4.2596
> 700 W/m²
41.4241
25.8557
8.7244
5.4233
8.1643
> 800 W/m²
55.2321
7.8251
11.6325
1.5441
10.8857
> 900 W/m²
53.0149
3.2436
11.1823
0.5786
11.1398
> 1000 W/m²
NaN
NaN
NaN
NaN
NaN
Cell area [cm²]
16760.1
Calibration worth [W]
Serial number
PV Device
Pmax [%/K]
-7.9127
Isc [mA/K]
731.4020
Voc [mV/K]
464.9530
Rs [ ]
1.1135
Starting time
09:15:56
End time
16:18:04
All data
06.07.2010
Evaluate

```
function varargout = int3wo(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int3wo_OpeningFcn, ...
                   'gui_OutputFcn',  @int3wo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int3wo_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global MDM titMDM titchMDM fechaMDM horai horaf c_a nu Irr
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
set(handles.date, 'String', fechaMDM)

%make a cell and put the data on it
%vbusc looks for all the data, that are bigger than the amount of (i-
1)*100
%take them and call to function sacaMedia, this function returns the
values
%of average, and standard desviation of Power, voltage and intensity
num_datos = cell(13,6);
% Xmed= average
% Xsd=standard desviation
for i=1:11
    vbusc = find(MDM(:,1)>((i-1)*100));
    [Pmed, Psd, Vmed, Vsd, Imed, Isd] = sacaMedia (MDM , vbusc);
    num_datos{i,1} = Pmed;
    num_datos{i,2} = Psd;
    num_datos{i,3} = Vmed;
    num_datos{i,4} = Vsd;
    num_datos{i,5} = Imed;
    num_datos{i,6} = Isd;
    %if is a number than save it
     if ~isnan(Pmed)
        Pmedn = Pmed;
        Psdn = Psd;
        Vmedn = Vmed;
        Vsdn = Vsd;
        Imedn = Imed;
        Isdn = Isd;
     end
```

```
end

num_datos{12,2}= Pmedn+(3*Psdn);
num_datos{12,4}= Vmedn+(3*Vsdn);
num_datos{12,6}= Imedn+(3*Isdn);

num_datos{13,2}= Pmedn-(3*Psdn);
num_datos{13,4}= Vmedn-(3*Vsdn);
num_datos{13,6}= Imedn-(3*Isdn);

set(handles.tablater1,'Data',num_datos)

% a cell for different data
num_datos2 = cell(3,1);
num_datos2= cell(3,1);
num_datos2{1,1}=num2str(c_a);
set(handles.tablater2,'Data',num_datos2)
num_datos3 = cell(7,1);
Gmax=max(MDM(:,1));
Gmaxdiv=Gmax/100;
Gmaxent=floor(Gmaxdiv);
Gtope=Gmaxent*100;

vbusctope=find(MDM(:,1)>Gtope);
l_tope=length(vbusctope);
Rs=zeros(l_tope,1);
Rp=zeros(l_tope,1);
FF=zeros(l_tope,1);
for i=1:l_tope

    Rs(i)=MDM(vbusctope(i),13);
    Rp(i)=MDM(vbusctope(i),14);
    FF(i)=MDM(vbusctope(i),12);
end
Rsm=mean(Rs);
Rpm=mean(Rp);
FFm=mean(FF);
%calculation of temperatur coefficients
cp=polyfit(MDM(:,2), MDM(:,5),1);
cv=polyfit(MDM(:,2), MDM(:,6),1);
ci=polyfit(MDM(:,2), MDM(:,7),1);
c_T_p = (cp(1)/cp(2))*100;
c_T_v = cv(1)*1000;
c_T_i = ci(1)*1000;
num_datos3{1,1}=c_T_p;
num_datos3{2,1}=c_T_i;
num_datos3{3,1}=c_T_v;
%average of resistances and fill factor
num_datos3{4,1}=Rsm;
num_datos3{5,1}=Rpm;
num_datos3{6,1}=FFm;
%calculation for nu=eta(%)=Pmpp/(area*W/m²)
c_a_m=c_a/10000;
nu = MDM(:,5)./(c_a_m.*MDM(:,1));
numed=mean(nu);
num_datos3{7,1}=numed;
```

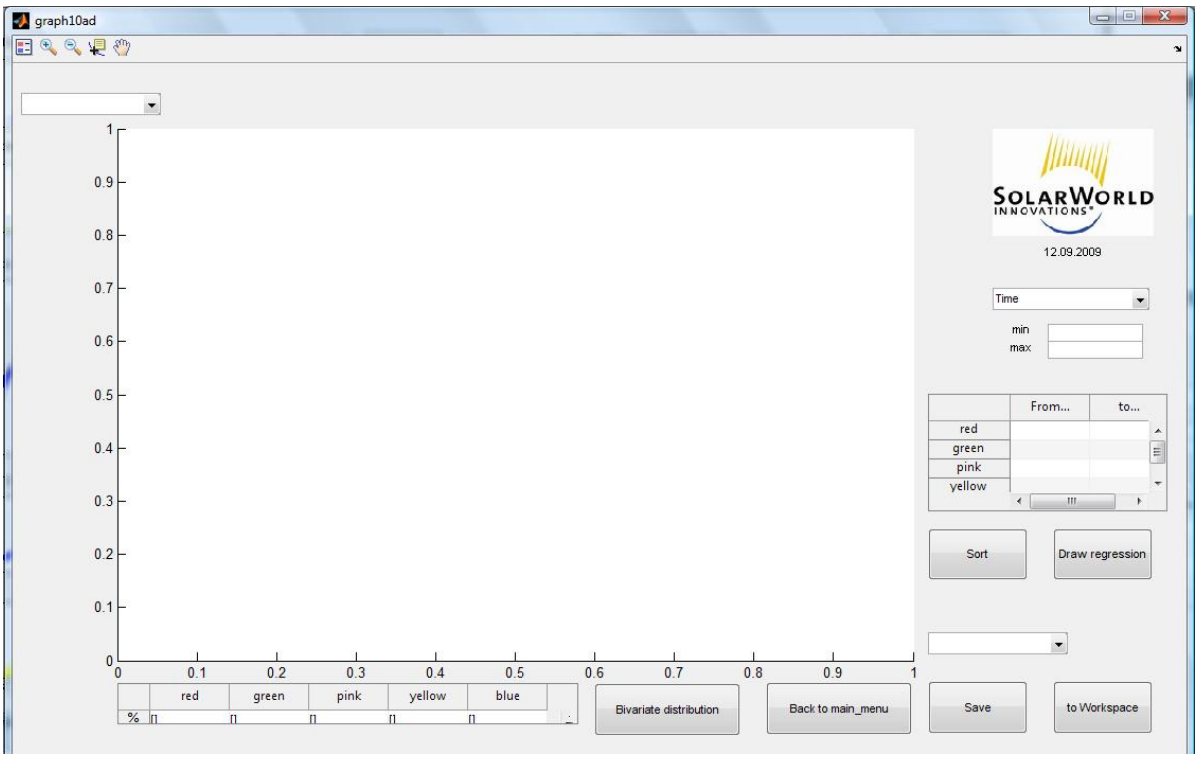
```
set(handles.tablater3,'Data',num_datos3)

% differents data in a table
num_datos4 = cell(2,1);
set(handles.tablater4,'Data',num_datos4)

num_datos5 = cell(5,1);
Gmed=mean(MDM(:,1));
Tmed=mean(MDM(:,2));
num_datos5{1,1}=horai;
num_datos5{2,1}=horaf;
num_datos5{3,1}= Irr;
num_datos5{4,1}=Gmed;
num_datos5{5,1}=Tmed;
set(handles.tablater5,'Data',num_datos5)

function varargout = int3wo_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function evaluate_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a
% cua_int
int4wo
```

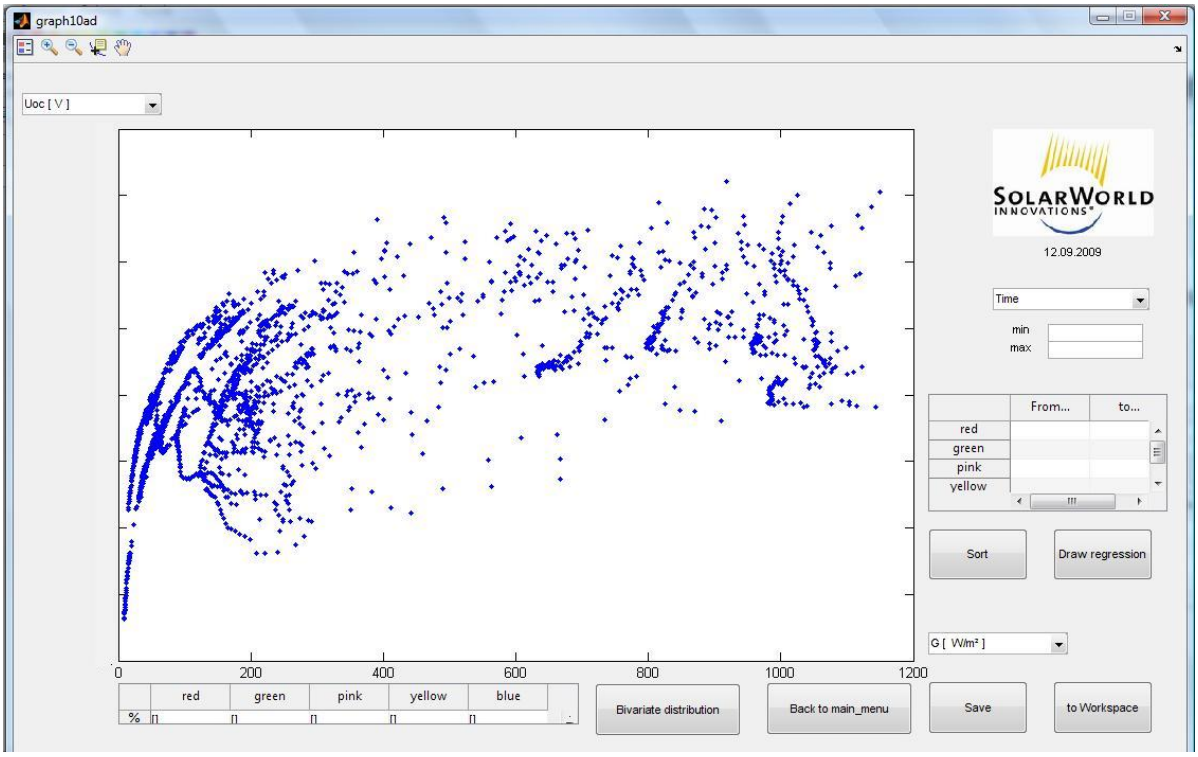

int4ad
Av. Ppk (G (W/m²))
Make bigger
St.dev. (G (W/m²))
Ppk (G (w/m²))
Make bigger
Isc [A] (T[°C])
Make bigger
Isc [A] (G [W/m²])
Make bigger
Uoc [V] (T [°C])
Make bigger
Uoc [V] (G [W/m²])
Make bigger
Pmax [W] (G [W/m²])
Make bigger
Pmax [W] (T [°C])
Make bigger
T [°C] (Time)
Make bigger
G [W/m²] (Time)
ETA(T)
Make bigger
P (W)
T (°C)
G (W/m²)
Would you like to choose axes?
Click here
SOLARWORLD
INNOVATIONS
12.09.2009

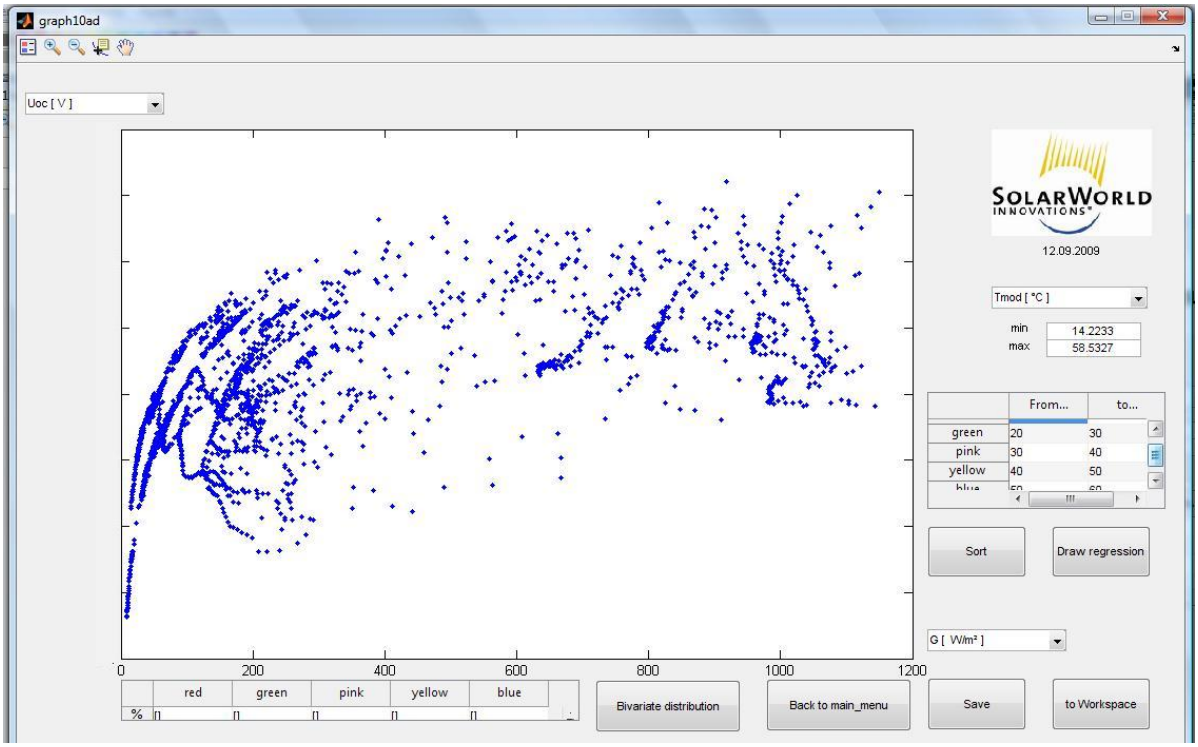
```
function varargout = int4wo(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int4wo_OpeningFcn, ...
                   'gui_OutputFcn',  @int4wo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int4wo_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

global MDM MDMi titMDM titchMDM fechaMDM c_a nu
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
set(handles.date, 'String', fechaMDM)
Pav = zeros(11,1);
Pde = zeros(11,1);
for i=1:11
    vbusc = find(MDM(:,1)>((i-1)*100));
    [Pmed, Psd, Vmed, Vsd, Imed, Isd] = sacaMedia (MDM , vbusc);
    Pav(i,1)=Pmed;
    Pde(i,1)=Psd;
end
     vG= [0 100 200 300 400 500 600 700 800 900 1000];
%graphic y=power average x=G(W/m²)
     plot(handles.eje1,  vG(:), Pav(:,1), 'b.', vG(:), Pav(:,1), '--b')
%graphic y=power standard desviation x=G(W/m²)
     plot(handles.eje11, vG(:), Pde(:,1), 'y.' ,vG(:), Pde(:,1), '--y')
%x=G(W/m²) y=power
media = mean(MDMi(:,9));
desv = std (MDMi(:,9));
dsa= media + (3*desv);
dsb= media - (3*desv);
plot(handles.eje2,  MDM(:,1), MDM(:,9), 'b.', MDM(:,1),media, '--
y',MDM(:,1), dsa, '--c', MDM(:,1),dsb, '--c')
% Isc T
c=polyfit(MDM(:,2), MDM(:,7),1);
l=length(MDM(:,1));
aplin3 = zeros(l,1);
for i=1:l
aplin3(i,1)= c(1)*MDM(i,2)+ c(2);
end
```

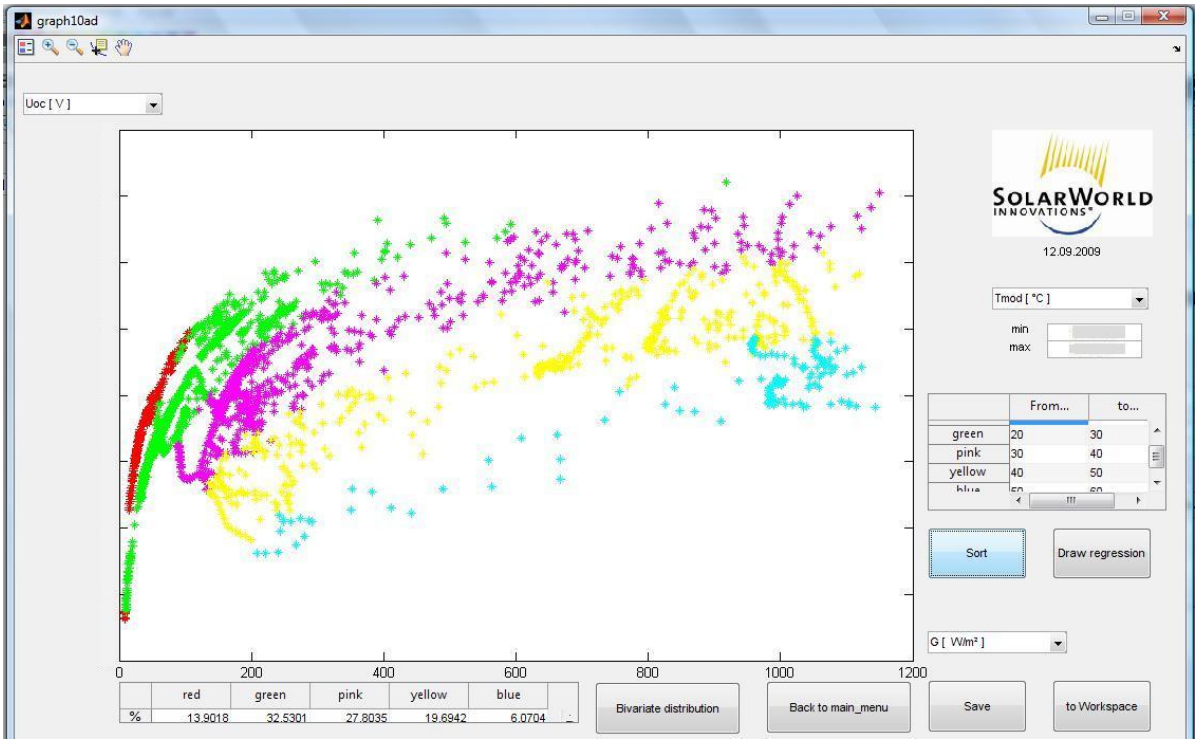
```
plot(handles.eje3,  MDM(:,2), MDM(:,7), 'b.',MDM(:,2),aplin3(:,1),'--y')
% Isc G
c=polyfit(MDM(:,1), MDM(:,7),1);
l=length(MDM(:,1));
aplin4 = zeros(l,1);
for i=1:l
aplin4(i,1)= c(1)*MDM(i,1)+ c(2);
end
plot(handles.eje4,  MDM(:,1), MDM(:,7), 'b.',MDM(:,1),aplin4(:,1),'--y')
% Voc T
c=polyfit(MDM(:,2), MDM(:,6),1);
l=length(MDM(:,1));
aplin5 = zeros(l,1);
for i=1:l
aplin5(i,1)= c(1)*MDM(i,2)+ c(2);
end
plot(handles.eje5,  MDM(:,2), MDM(:,6), 'b.',MDM(:,2),aplin5(:,1),'--y')
% Voc G
c=polyfit(MDM(:,1), MDM(:,6),1);
l=length(MDM(:,1));
aplin6 = zeros(l,1);
for i=1:l
aplin6(i,1)= c(1)*MDM(i,1)+ c(2);
end
plot(handles.eje6,  MDM(:,1), MDM(:,6), 'b.',MDM(:,1),aplin6(:,1),'--y')
% Pmpp G
c=polyfit(MDM(:,1), MDM(:,5),1);
l=length(MDM(:,1));
aplin7 = zeros(l,1);
for i=1:l
aplin7(i,1)= c(1)*MDM(i,1)+ c(2);
end
plot(handles.eje7,  MDM(:,1), MDM(:,5), 'b.',MDM(:,1),aplin7(:,1),'--y')
% Pmpp T
c=polyfit(MDM(:,2), MDM(:,5),1);
l=length(MDM(:,1));
aplin8 = zeros(l,1);
for i=1:l
aplin8(i,1)= c(1)*MDM(i,2)+ c(2);
end
plot(handles.eje8,  MDM(:,2), MDM(:,5), 'b.',MDM(:,2),aplin8(:,1),'--y')
% modul temperature time
    horas=datenum(titchMDM,'HH:MM:SS');
     plot(handles.eje9, horas, MDM(:,2), 'b.', horas, MDM(:,2), '--b')
% radiation time
     plot(handles.eje12, horas, MDM(:,1),'y.', horas, MDM(:,1), '--y')
%eta temperature
c=polyfit(MDM(:,2), nu(:),1);
l=length(MDM(:,1));
aplin14 = zeros(l,1);
for i=1:l
aplin14(i,1)= c(1)*MDM(i,2)+ c(2);
end
plot(handles.eje14,  MDM(:,2), nu(:), 'b.',MDM(:,2),aplin14(:,1),'--y')
%x= radiation y= modultemperature z=Pmpp
plot3(handles.eje10, MDM(:,1),MDM(:,2),MDM(:,5))
```

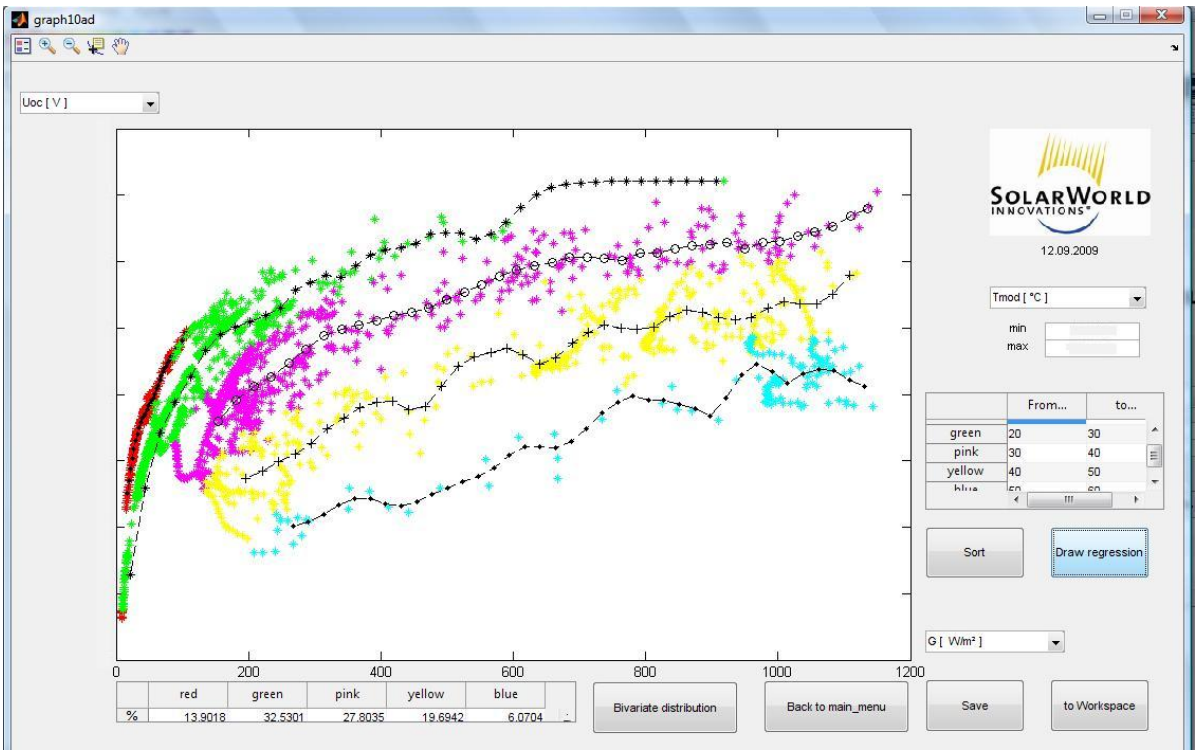
```
function varargout = int4wo_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function boton1_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph1wo

function boton2_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph2wo

function boton3_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph3wo

function boton4_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph4wo

function boton8_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph8wo

function boton7_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph7wo

function boton6_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph6wo

function boton5_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph5wo

function boton9_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph9wo

function boton10_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph11wo

function boton11_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graphtridwo

function choosebutton_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM c_a nu
graph10wo2
```


graph10ad
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
SOLARWORLD
INNOVATIONS
12.09.2009
Time
min
max
From...
to...
red
green
pink
yellow
Sort
Draw regression
red
green
pink
yellow
blue
%
Bivariate distribution
Back to main_menu
Save
to Workspace


graph10ad
Uoc [V]
SOLARWORLD
INNOVATIONS
12.09.2009
Time
min
max
From...
to...
red
green
pink
yellow
Sort
Draw regression
G [W/m²]
0 200 400 600 800 1000 1200
red
green
pink
yellow
blue
%
Bivariate distribution
Back to main_menu
Save
to Workspace


graph10ad
Uoc [V]
SOLARWORLD
INNOVATIONS
12.09.2009
Tmod [°C]
min
14.2233
max
58.5327
From...
to...
green
20
30
pink
30
40
yellow
40
50
Sort
Draw regression
G [W/m²]
0
200
400
600
800
1000
1200
red
green
pink
yellow
blue
%
Bivariate distribution
Back to main_menu
Save
to Workspace


graph10ad
Uoc [V]
SOLARWORLD
INNOVATIONS
12.09.2009
Tmod [°C]
min
max
From...
to...
green
20
30
pink
30
40
yellow
40
50
Sort
Draw regression
G [W/m²]
0
200
400
600
800
1000
1200
red
green
pink
yellow
blue
%
13.9018
32.5301
27.8035
19.6942
6.0704
Bivariate distribution
Back to main_menu
Save
to Workspace


graph10ad
Uoc [V]
0
200
400
600
800
1000
1200
red
green
pink
yellow
blue
%
13.9018
32.5301
27.8035
19.6942
6.0704
Bivariate distribution
Back to main_menu
SOLARWORLD
INNOVATIONS
12.09.2009
Tmod [°C]
min
max
From...
to...
green
20
30
pink
30
40
yellow
40
50
Sort
Draw regression
G [W/m²]
Save
to Workspace

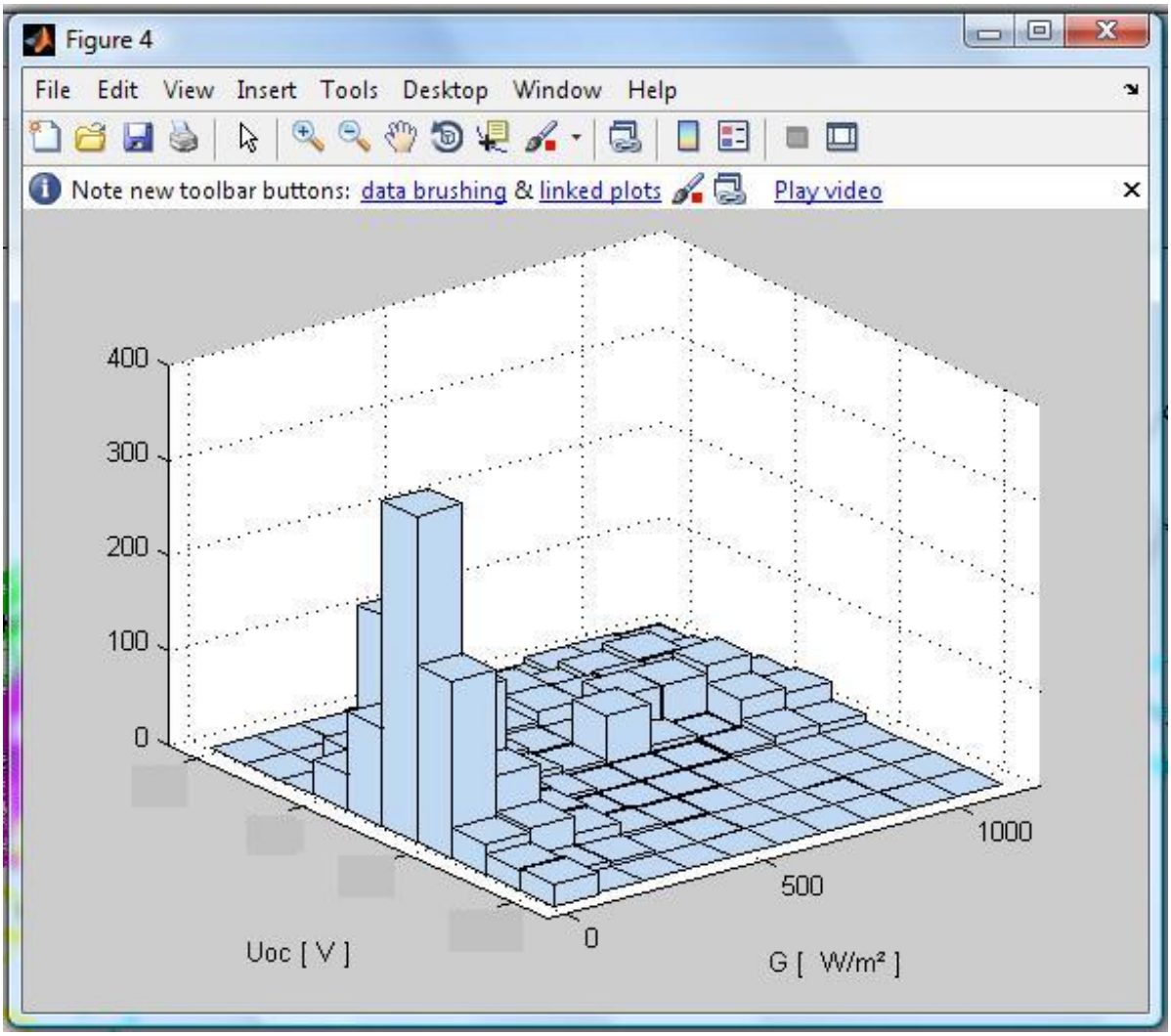


Figure 4
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
400
300
200
100
0
0
500
1000
Uoc [V]
G [W/m²]

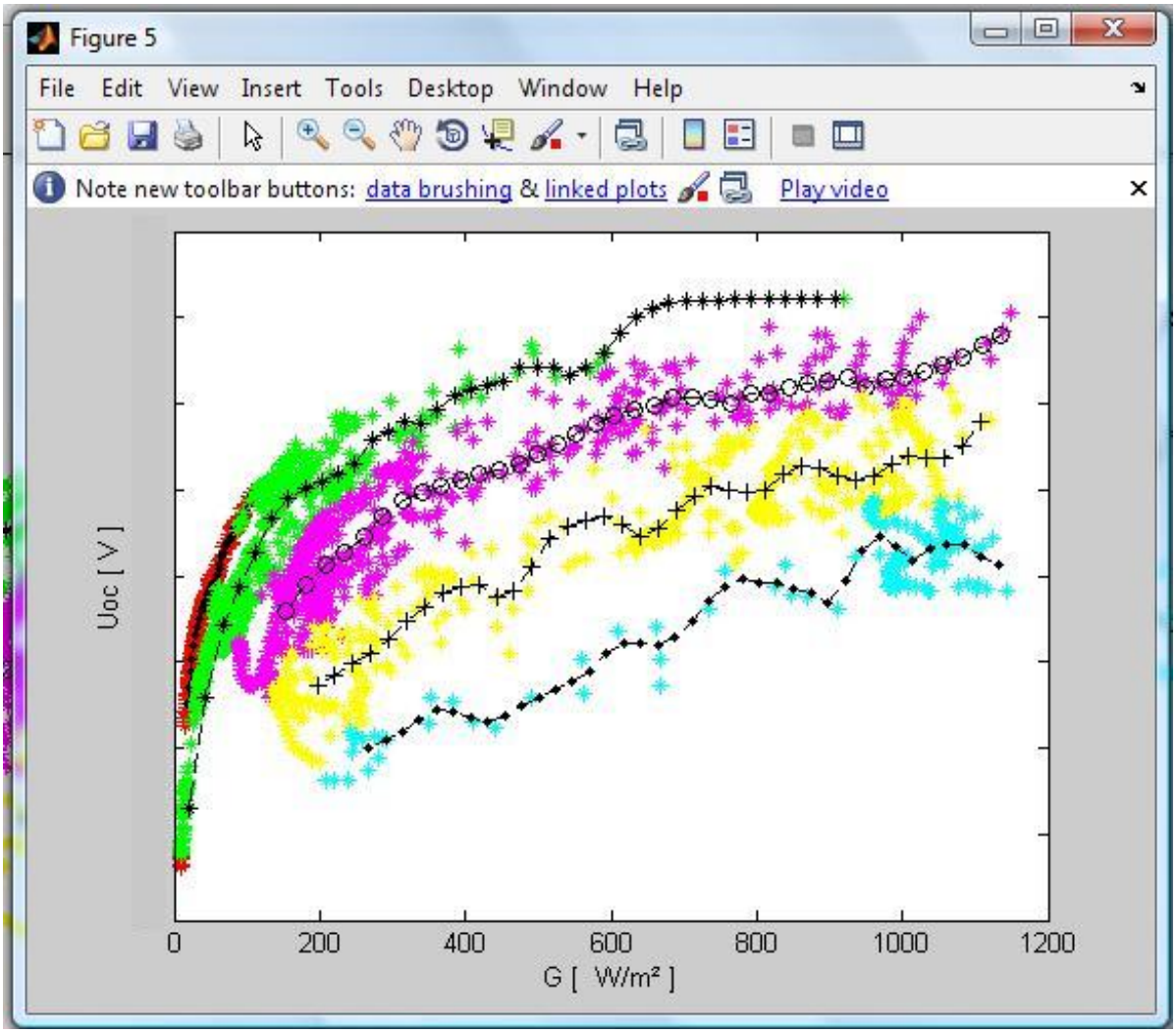
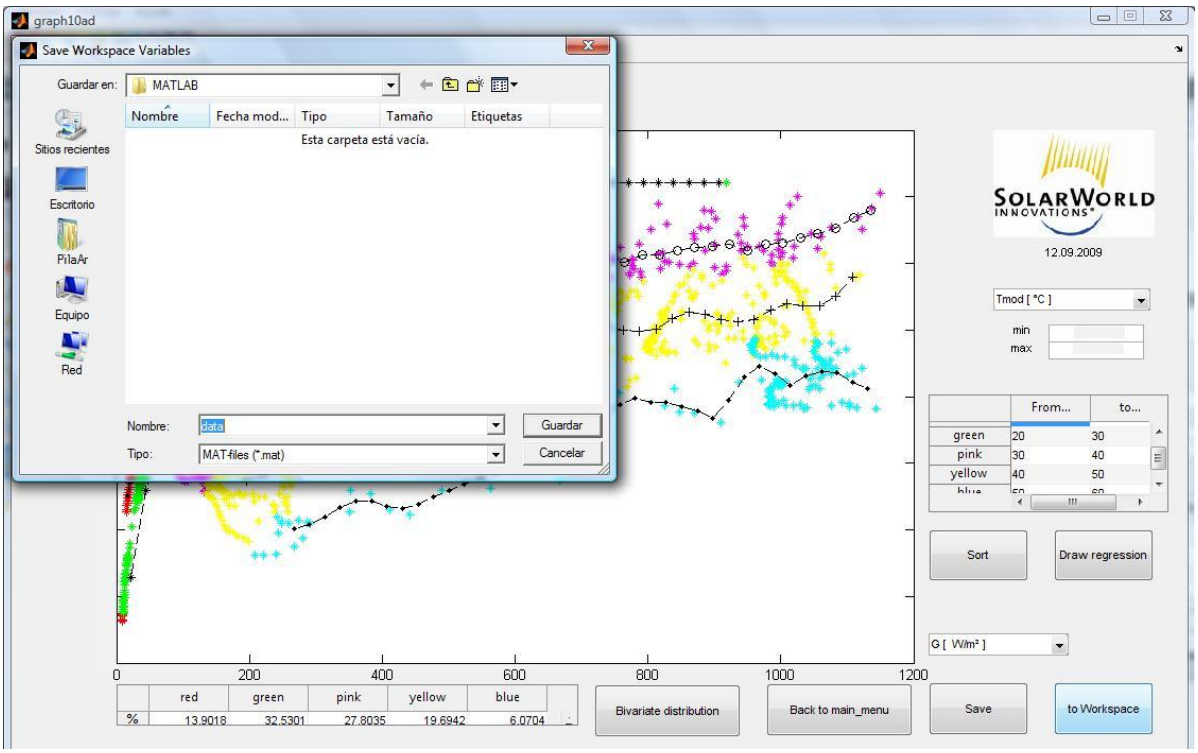


Figure 5
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
Uoc [V]
0
200
400
600
800
1000
1200
G [W/m²]


graph10ad
Save Workspace Variables
Guardar en:
MATLAB
Nombre
Fecha mod...
Tipo
Tamaño
Etiquetas
Esta carpeta está vacía.
Sitios recientes
Escritorio
PilaAr
Equipo
Red
Nombre:
data
Tipo:
MAT-files (*.mat)
Guardar
Cancelar
SOLARWORLD
INNOVATIONS
12.09.2009
Tmod [°C]
min
max
From...
to...
green
20
30
pink
30
40
yellow
40
50
Sort
Draw regression
G [W/m²]
0
200
400
600
800
1000
1200
red
green
pink
yellow
blue
%
13.9018
32.5301
27.8035
19.6942
6.0704
Bivariate distribution
Back to main_menu
Save
to Workspace

```
function varargout = graph10wo2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @graph10wo2_OpeningFcn, ...
                   'gui_OutputFcn',  @graph10wo2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function graph10wo2_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

global MDM titMDM titchMDM fechaMDM c_a nu sort rxo ryo gxo gyo pxo pyo
yxo yyo bxo byo regression
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
set(handles.date, 'String', fechaMDM)
%initialize variables to empty
rxo=[];
ryo=[];
gxo=[];
gyo=[];
pxo=[];
pyo=[];
yxo=[];
yyo=[];
bxo=[];
byo=[];
%create a cell
num_elem=cell(5,2);
set(handles.table,'Data',num_elem);
%initialize sort and regression options to zero
sort=0;
regression=0;

function varargout = graph10wo2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupy_Callback(hObject, eventdata, handles)
% choose in pop up menu y axes
% it depends on the varialbe , than y take the value of this variable
global MDM titMDM titchMDM fechaMDM y x c_a nu ymark ytag
y=[];
stry = get(hObject, 'String');
valy = get(hObject,'Value');
switch stry{valy};
    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        y = horas;
        ymark=0;
        ytag='Time';
    case 'G [  W/m² ]'
        y = MDM(:,1);
        ymark=1;
        ytag='G [  W/m² ]';
    case 'Tmod [ °C ]'
        y = MDM(:,2);
        ymark=2;
        ytag='Tmod [ °C ]';
    case 'Umpp [ V ]'
        y = MDM(:,3);
        ymark=3;
        ytag='Umpp [ V ]';
    case 'Impp [ A ]'
        y = MDM(:,4);
        ymark=4;
        ytag='Impp [ A ]';
    case 'Pmpp [ W ]'
        y = MDM(:,5);
        ymark=5;
        ytag='Pmpp [ W ]';
    case 'Uoc [ V ]'
        y = MDM(:,6);
        ymark=6;
        ytag='Uoc [ V ]';
    case 'Isc [ A ]'
```

```
        y = MDM(:,7);
        ymark=7;
        ytag='Isc [ A ]';
    case 'FF [ % ]'
        y = MDM(:,8);
        ymark=8;
        ytag='FF [ % ]';
    case 'Pmpp,STC [ W ]'
        y = MDM(:,9);
        ymark=9;
        ytag='Pmpp,STC [ W ]';
    case 'Uoc,STC [ V ]'
        y = MDM(:,10);
        ymark=10;
        ytag='Uoc,STC [ V ]';
    case 'Isc,STC [ A ]'
        y = MDM(:,11);
        ymark=11;
        ytag='Isc,STC [ A ]';
    case 'FF,STC [ % ]'
        y = MDM(:,12);
        ymark=12;
        ytag='FF,STC [ % ]';
    case 'Rs [ Ohm ]'
        y = MDM(:,13);
        ymark=13;
        ytag='Rs [ Ohm ]';
    case 'Rp [ Ohm ]'
        y = MDM(:,14);
        ymark=14;
        ytag='Rp [ Ohm ]';
    case 'ETA[%]'
        y = nu(:);
        ymark=15;
        ytag='ETA[%]';
end

function popupy_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupx_Callback(hObject, eventdata, handles)
%the same as the y popup menu
global MDM titMDM titchMDM fechaMDM y x c_a nu xmark xtag titchMDM sort
regression
x=[];
strx = get(hObject, 'String');
valx = get(hObject,'Value');
switch strx{valx};

    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        x = horas;
        xmark=0;
```

```
    xtag='Time';
case 'G [  W/m² ]'
    x = MDM(:,1);
    xmark=1;
    xtag='G [  W/m² ]';
case 'Tmod [ °C ]'
    x = MDM(:,2);
    xmark=2;
    xtag='Tmod [ °C ]';
case 'Umpp [ V ]'
    x = MDM(:,3);
    xmark=3;
    xtag='Umpp [ V ]';
case 'Impp [ A ]'
    x = MDM(:,4);
    xmark=4;
    xtag='Impp [ A ]';
case 'Pmpp [ W ]'
    x = MDM(:,5);
    xmark=5;
    xtag='Pmpp [ W ]';
case 'Uoc [ V ]'
    x = MDM(:,6);
    xmark=6;
    xtag='Uoc [ V ]';
case 'Isc [ A ]'
    x = MDM(:,7);
    xmark=7;
    xtag='Isc [ A ]';
case 'FF [ % ]'
    x = MDM(:,8);
    xmark=8;
    xtag='FF [ % ]';
case 'Pmpp,STC [ W ]'
    x = MDM(:,9);
    xmark=9;
    xtag='Pmpp,STC [ W ]';
case 'Uoc,STC [ V ]'
    x = MDM(:,10);
    xmark=10;
    xtag='Uoc,STC [ V ]';
case 'Isc,STC [ A ]'
    x = MDM(:,11);
    xmark=11;
    xtag='Isc,STC [ A ]';
case 'FF,STC [ % ]'
    x = MDM(:,12);
    xmark=12;
    xtag='FF,STC [ % ]';
case 'Rs [ Ohm ]'
    x = MDM(:,13);
    xmark=13;
    xtag='Rs [ Ohm ]';
case 'Rp [ Ohm ]'
    x = MDM(:,14);
    xmark=14;
    xtag='Rp [ Ohm ]';
```

```
    case 'ETA[%]'
        x = nu(:);
        xmark=15;
        xtag='ETA[%]';
end
%plot x y
lx=length(x);
ly=length(y);
plot(handles.eje10, x, y, 'b.')
sort=0;
regression=0;

function popupx_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupselection_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM y x c_a nu z zmark
%choose a variable to sort
%and show the minimum and maximum of this variable
z=[];
strz = get(hObject, 'String');
valz = get(hObject,'Value');
switch strz{valz};

    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        z = horas;
        zmark=0;
    case 'G [  W/m² ]'
        z = MDM(:,1);
        zmark=1;
    case 'Tmod [ °C ]'
        zmark=1;
        z = MDM(:,2);
    case 'Umpp [ V ]'
        z = MDM(:,3);
        zmark=1;
    case 'Impp [ A ]'
        z = MDM(:,4);
        zmark=1;
    case 'Pmpp [ W ]'
        z = MDM(:,5);
        zmark=1;
    case 'Uoc [ V ]'
        z = MDM(:,6);
        zmark=1;
    case 'Isc [ A ]'
        z = MDM(:,7);
        zmark=1;
    case 'FF [ % ]'
        z = MDM(:,8);
        zmark=1;
    case 'Pmpp,STC [ W ]'
```

```
        zmark=1;
        z = MDM(:,9);
    case 'Uoc,STC [ V ]'
        zmark=1;
        z = MDM(:,10);
    case 'Isc,STC [ A ]'
        zmark=1;
        z = MDM(:,11);
    case 'FF,STC [ % ]'
        zmark=1;
        z = MDM(:,12);
    case 'Rs [ Ohm ]'
        zmark=1;
        z = MDM(:,13);
    case 'Rp [ Ohm ]'
        zmark=1;
        z = MDM(:,14);
    case 'ETA[%]'
        zmark=1;
        z = nu(:);
end

minz=min(z);
maxz=max(z);
%message to remember, the way to put the data
if zmark==0
    warndlg('Always hh:mm:ss and if you want to put an empty field, you
must writen 00:00:00','Bad Input','modal')
    uicontrol(hObject)

end
if zmark==1
    warndlg('If you want to put an empty field, you must writen 0','Bad
Input','modal')
    uicontrol(hObject)

end
%when the data selected is time
if zmark==0
    timemin=datevec(minz);
    hhmin=num2str(timemin(end-2));
    mmmin=num2str(timemin(end-1));
    if timemin(end-1)<10
        mmmin=['0' mmmin];
    end
    ssmin=num2str(timemin(end));
    if timemin(end)<10
        ssmin=['0' ssmin];
    end
    minz=[hhmin ':' mmmin ':' ssmin];
    timemax=datevec(maxz);
    hhmax=num2str(timemax(end-2));
    mmmax=num2str(timemax(end-1));
    if timemax(end-1)<10
        mmmax=['0' mmmax];
    end
```

```
    ssmax=num2str(timemax(end));
    if timemax(end)<10
        ssmax=['0' ssmax];
    end
    maxz=[hhmax ':' mmmax ':' ssmax];
end
set(handles.minbox, 'String', minz)
set(handles.maxbox, 'String', maxz)

function popupselection_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function minbox_Callback(hObject, eventdata, handles)

function minbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function maxbox_Callback(hObject, eventdata, handles)

function maxbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sortbutton_Callback(hObject, eventdata, handles)
%to sort
global MDM titMDM titchMDM fechaMDM y x c_a nu z sort rxo ryo gxo gyo pxo
pyo yxo yyo bxo byo zmark
%take the values of the table, where the differents fields are.
a=get(handles.table, 'Data');
sort=1;
%take the values of limits
if zmark==0
    rmin=datenum(a{1,1},'HH:MM:SS');
    rmax=datenum(a{1,2},'HH:MM:SS');
    gmin=datenum(a{2,1},'HH:MM:SS');
    gmax=datenum(a{2,2},'HH:MM:SS');
    pmin=datenum(a{3,1},'HH:MM:SS');
    pmax=datenum(a{3,2},'HH:MM:SS');
    ymin=datenum(a{4,1},'HH:MM:SS');
    ymax=datenum(a{4,2},'HH:MM:SS');
    bmin=datenum(a{5,1},'HH:MM:SS');
    bmax=datenum(a{5,2},'HH:MM:SS');

else
    rmin=str2num(strrep(a{1,1},',','.'));
    rmax=str2num(strrep(a{1,2},',','.'));
    gmin=str2num(strrep(a{2,1},',','.'));
    gmax=str2num(strrep(a{2,2},',','.'));
```

```
    pmin=str2num(strrep(a{3,1},',','.'));
    pmax=str2num(strrep(a{3,2},',','.'));
    ymin=str2num(strrep(a{4,1},',','.'));
    ymax=str2num(strrep(a{4,2},',','.'));
    bmin=str2num(strrep(a{5,1},',','.'));
    bmax=str2num(strrep(a{5,2},',','.'));
end

l=length(MDM(:,1));
rxo=[];
ryo=[];
gxo=[];
gyo=[];
pxo=[];
pyo=[];
yxo=[];
yyo=[];
bxo=[];
byo=[];
contr=1;
contg=1;
contp=1;
conty=1;
contb=1;
%make a vector for all the fields.
for i=1:l
    if ((rmin<=z(i)) && (rmax>=z(i)))
        rxo(contr)=x(i);
        ryo(contr)=y(i);
        contr=contr+1;
    end
    if ((gmin<z(i)) && (gmax>=z(i)))
        gxo(contg)=x(i);
        gyo(contg)=y(i);
        contg=contg+1;
    end
    if ((pmin<z(i)) && (pmax>=z(i)))
        pxo(contp)=x(i);
        pyo(contp)=y(i);
        contp=contp+1;
    end
    if ((ymin<z(i)) && (ymax>=z(i)))
        yxo(conty)=x(i);
        yyo(conty)=y(i);
        conty=conty+1;
    end
    if ((bmin<z(i)) && (bmax>=z(i)))
        bxo(contb)=x(i);
        byo(contb)=y(i);
        contb=contb+1;
    end
end

 lr=length(rxo);
 lry=length(ryo);
```

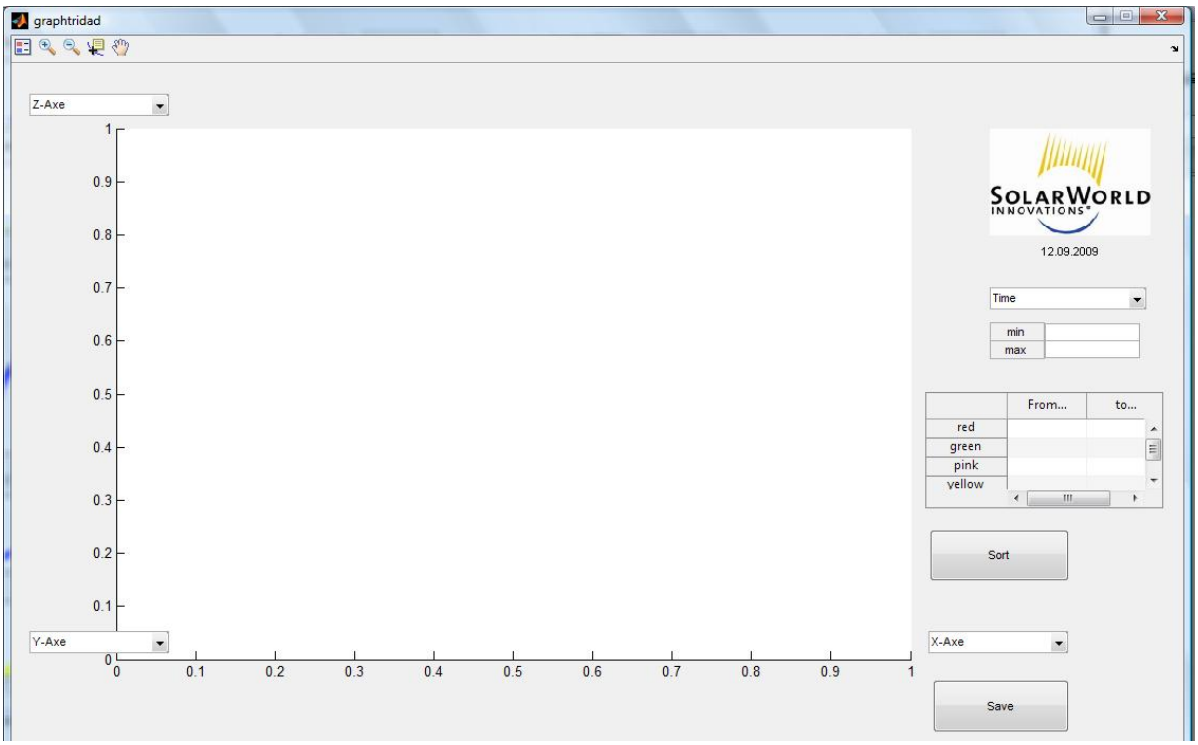
```
 lg=length(gxo);
  lgy=length(gyo);

 lp=length(pxo);
 lpy=length(pyo);

 ly=length(yxo);
  lyy=length(yyo);

 lb=length(bxo);
 lby=length(byo);
%plot the vectors of all fields
plot(handles.eje10,rxo(:),ryo(:),'r*',gxo(:),gyo(:),'g*',pxo(:),pyo(:),'m
*',yxo(:), yyo(:), 'y*', bxo(:),byo(:),'c*')
%put the procent of all the fields
procent=cell(1,5);
largered=lr/l;
largegreen=lg/l;
largepink=lp/l;
largeyellow=ly/l;
largeblue=lb/l;
procent{1,1}=(lr/l)*100;
procent{1,2}=(lg/l)*100;
procent{1,3}=(lp/l)*100;
procent{1,4}=(ly/l)*100;
procent{1,5}=(lb/l)*100;
set(handles.tableprocent,'Data',procent)
% --- Executes on button press in savebutton.
function savebutton_Callback(hObject, eventdata, handles)
%take the plot and save it in a figure of Matlab
global xmark xtag ymark ytag x y sort rxo ryo gxo gyo pxo pyo yxo yyo bxo
byo xsupr ysupr xsupg ysupg xsupp ysupp xsupy ysupy xsupb ysupb
regression
rgb = getframe(handles.eje10);
figure
if (sort==1) && (regression==1)
    plot(rxo(:),ryo(:),'r*',xsupr(:),ysupr(:),'k.--
',gxo(:),gyo(:),'g*',xsupg(:),ysupg(:),'k*--
',pxo(:),pyo(:),'m*',xsupp(:),ysupp(:),'ko--',yxo(:), yyo(:),
'y*',xsupy(:),ysupy(:),'k+--', bxo(:),byo(:),'c*',xsupb(:),ysupb(:),'k.--
')
elseif sort==1
    plot(rxo(:),ryo(:),'r*',gxo(:),gyo(:),'g*',pxo(:),pyo(:),'m*',yxo(:),
yyo(:), 'y*', bxo(:),byo(:),'c*')
else

    plot(x,y,'b*')
end

if xmark==0

    datetick('x','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)

elseif ymark==0
```

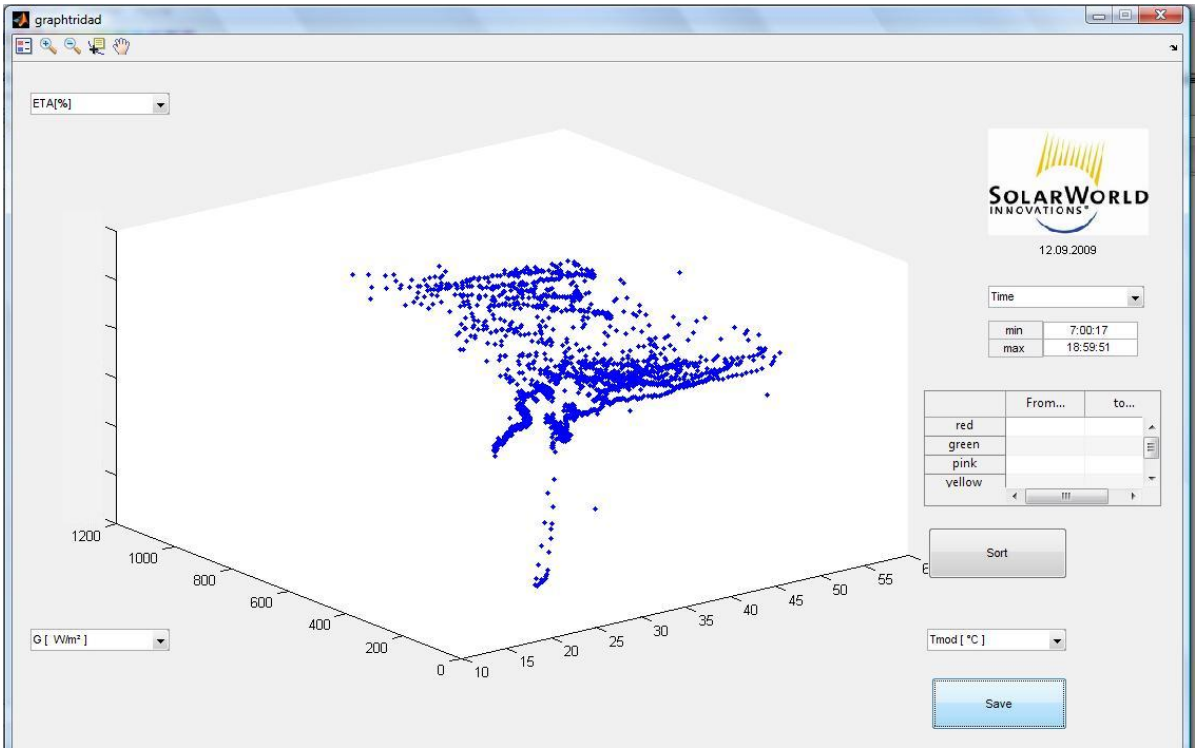
```
    datetick('y','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
else

    xlabel(xtag)
    ylabel(ytag)

end

function regredraw_Callback(hObject, eventdata, handles)
global xmark xtag ymark ytag x y sort rxo ryo gxo gyo pxo pyo yxo yyo bxo
byo regression xsupr ysupr xsupg ysupg xsupp ysupp xsupy ysupy xsupb
ysupb
regression=1;
% draw regression, in fact is a average point between two points
%take all the vectors and save it in a cell
classifycell = {rxo ; ryo; gxo; gyo; pxo; pyo; yxo; yyo; bxo; byo};
%initialize all vectors to empty
xx=[];
yy=[];
xsupr=[];
ysupr=[];
xsupg=[];
ysupg=[];
xsupp=[];
ysupp=[];
xsupy=[];
ysupy=[];
xsupb=[];
ysupb=[];
%look around the cell, than take the vectors
for n=1:2:10
    xx=classifycell{n};
    yy=classifycell{n+1};
   %if the vector is empty, than fixes the corresponding vector to empty
   a=length(xx);
    if a == 0
        if n==1
            xsupr=[];
            ysupr=[];
        elseif n==3
            xsupg=[];
            ysupg=[];
        elseif n==5
            xsupp=[];
            ysupp=[];
        elseif n==7
            xsupy=[];
            ysupy=[];
        else
            xsupb=[];
            ysupb=[];
        end
    else
```

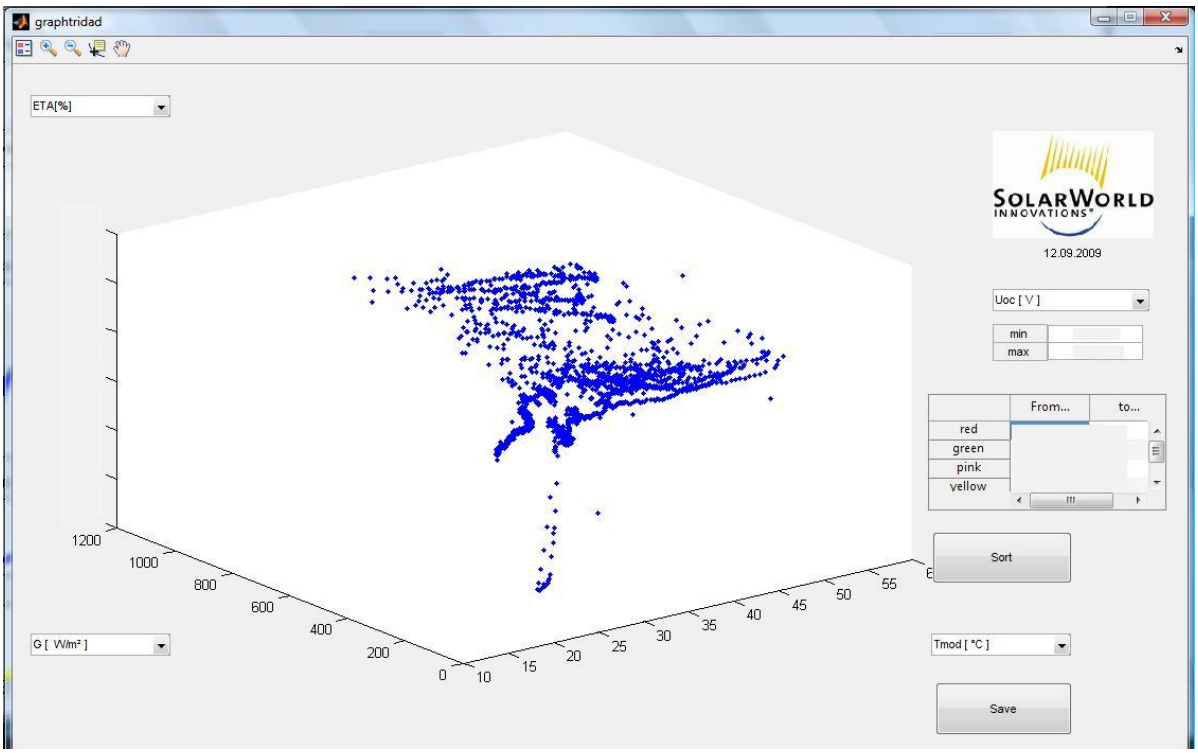
```
        %else take the maximum and minimum
        l=length(xx);
        maxxx=max(xx);
        minxx=min(xx);
        % take a parameter as 'd', to make 40 equidistant points
        d=(maxxx-minxx)/40;
        xsup=[];
        ysup=[];
        vector=[];
        l_v=0;
        y_2=[];
        %first poins
        xsup(1)=minxx+(d/2);
        %find all the points, that are between d and the minimum
        v=find(d > xx & xx > minxx );
        l_v1=length(v);
        y_1=[];
        %take it in a vector
        for h=1:l_v1
            y_1(h)=yy(v(h));
        end
        %and calculate the average
        ysup(1)=mean(y_1);
        %from 2 to 40 the same prozess, take a value plus d, and then
find
        %the positions of the x vector, that are between this value plus
and
        %minus d/2 .
        for i=2:40
            xsup(i)=xsup(i-1)+d;
            vector=find((xsup(i)+(d/2)) > xx & xx > (xsup(i)-(d/2)));
            %identify the position to y vector, and save it, and
calculate
            %the average
            l_v=length(vector);
            for j=1:l_v
                y_2(j)=yy(vector(j));
            end
            ysup(i)=mean(y_2);
            vector=[];
            y_2=[];

        end
        yliminf=ysup(1);
        cont=0;
        %when a value is not a number that breaks the vector
        %so find the first value, that is not a number, and go on up to
        %find a number value, so  have always number values
        for k=1:40
            if isnan(ysup(k))

                while isnan(ysup(k+cont)) && (k+cont)<40
                    cont=cont+1;
                end

                ysup(k)=(ysup(k+cont)+yliminf)/2;
```

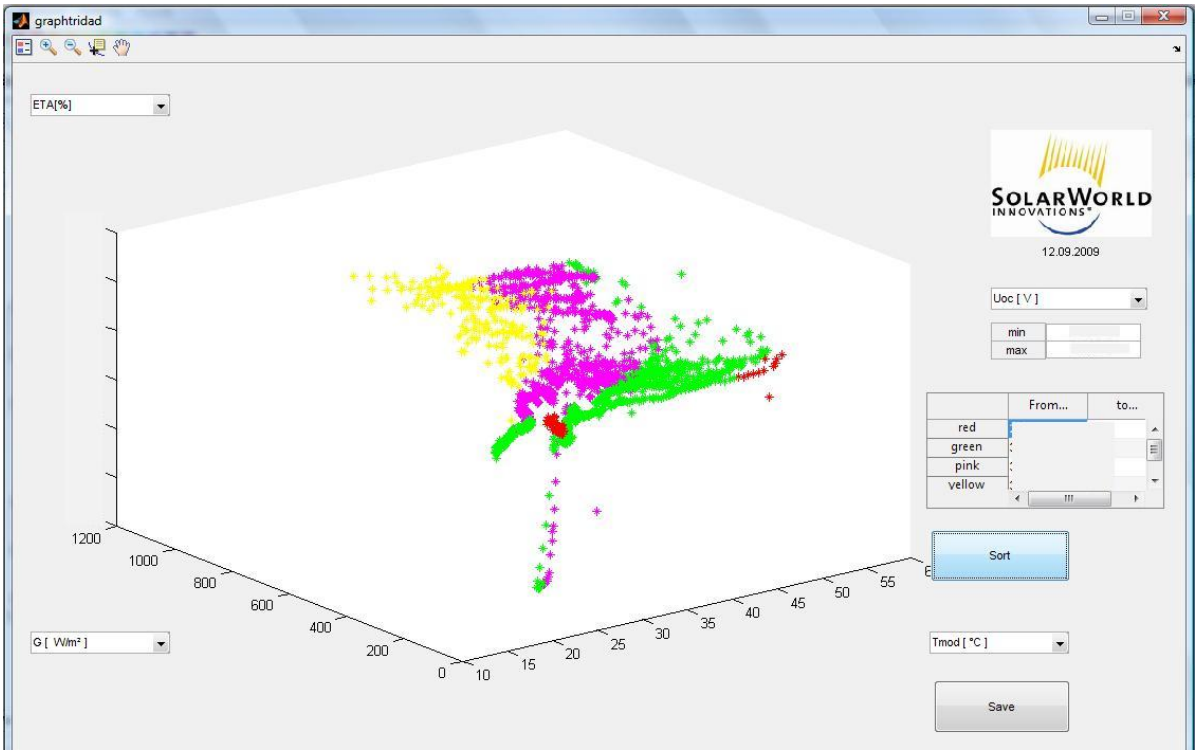
```
            end
            yliminf=ysup(k);
            cont=0;
        end
        yfin=[];
        yfin(1)=ysup(1);
        yfin(40)=ysup(40);
        %take all the values except the first and the last, and make an
        %average with this value, the previous and the following
        for m=2:39
            yfin(m)=(ysup(m-1)+ysup(m)+ysup(m+1))/3;
        end
        %when n=1 save it in red vector, and the same for all.
        if n==1
            xsupr=xsup;
            ysupr=yfin;
        elseif n==3
            xsupg=xsup;
            ysupg=yfin;
        elseif n==5
            xsupp=xsup;
            ysupp=yfin;
        elseif n==7
            xsupy=xsup;
            ysupy=yfin;
        else
            xsupb=xsup;
            ysupb=yfin;
        end
    end
    xx=[];
    yy=[];
end

plot(handles.eje10,rxo(:),ryo(:),'r*',xsupr(:),ysupr(:),'k.--
',gxo(:),gyo(:),'g*',xsupg(:),ysupg(:),'k*--
',pxo(:),pyo(:),'m*',xsupp(:),ysupp(:),'ko--',yxo(:), yyo(:),
'y*',xsupy(:),ysupy(:),'k+--', bxo(:),byo(:),'c*',xsupb(:),ysupb(:),'k.--
')
% --- Executes on button press in workspacebutton.
function workspacebutton_Callback(hObject, eventdata, handles)
%take all data, and save it in Workspace
global MDM titchMDM nu
horasch=titchMDM;
horas=datenum(titchMDM,'HH:MM:SS');
G=MDM(:,1);
Tmod=MDM(:,2);
Umpp=MDM(:,3);
Impp=MDM(:,4);
Pmpp=MDM(:,5);
Uoc=MDM(:,6);
Isc=MDM(:,7);
FF=MDM(:,8);
Pmppstc=MDM(:,9);
Uocstc=MDM(:,10);
Iscstc=MDM(:,11);
FFstc=MDM(:,12);
```

```
Rs=MDM(:,13);
Rp=MDM(:,14);
ETA=nu;
%save datos.mat horas horasch G Tmod Umpp Impp Pmpp Uoc Isc FF Pmppstc
Uocstc Iscstc FFstc Rs Rp
% save ..\data\datos.mat horas horasch G Tmod Umpp Impp Pmpp Uoc Isc FF
Pmppstc Uocstc Iscstc FFstc Rs Rp
uisave({'horas','horasch','G','Tmod','Umpp','Impp','Pmpp','Uoc','Isc','FF
','Pmppstc','Uocstc','Iscstc','FFstc','Rs','Rp','ETA'},'data')

function pushbutton5_Callback(hObject, eventdata, handles)
% back to main_menu and close these interfaces
close int1
close int2wo
close int3wo
main_menu

function pushbutton7_Callback(hObject, eventdata, handles)
%plot a bivariate graphic
global x y xtag ytag
A=[];
A=[x y];
l_x=length(x);
figure
hist3(A(1:l_x,1:2))
xlabel(xtag)
ylabel(ytag)
```


graphtridad
Z-Axe
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Y-Axe
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
SOLARWORLD
INNOVATIONS
12.09.2009
Time
min
max
From...
to...
red
green
pink
yellow
Sort
X-Axe
Save


graphtridad
ETA[%]
SOLARWORLD
INNOVATIONS
12.09.2009
Time
min
7:00:17
max
18:59:51
From...
to...
red
green
pink
yellow
Sort
1200
1000
800
600
400
200
0
G [W/m²]
10
15
20
25
30
35
40
45
50
55
Tmod [°C]
Save


graphtridad
ETA[%]
SOLARWORLD
INNOVATIONS
12.09.2009
Uoc [V]
min
max
From...
to...
red
green
pink
yellow
Sort
G [W/m²]
Tmod [°C]
Save
1200
1000
800
600
400
200
0
10
15
20
25
30
35
40
45
50
55


graphtridad
ETA[%]
SOLARWORLD
INNOVATIONS
12.09.2009
Uoc [V]
min
max
From...
to...
red
green
pink
yellow
Sort
G [W/m²]
Tmod [°C]
Save
1200
1000
800
600
400
200
0
10
15
20
25
30
35
40
45
50
55

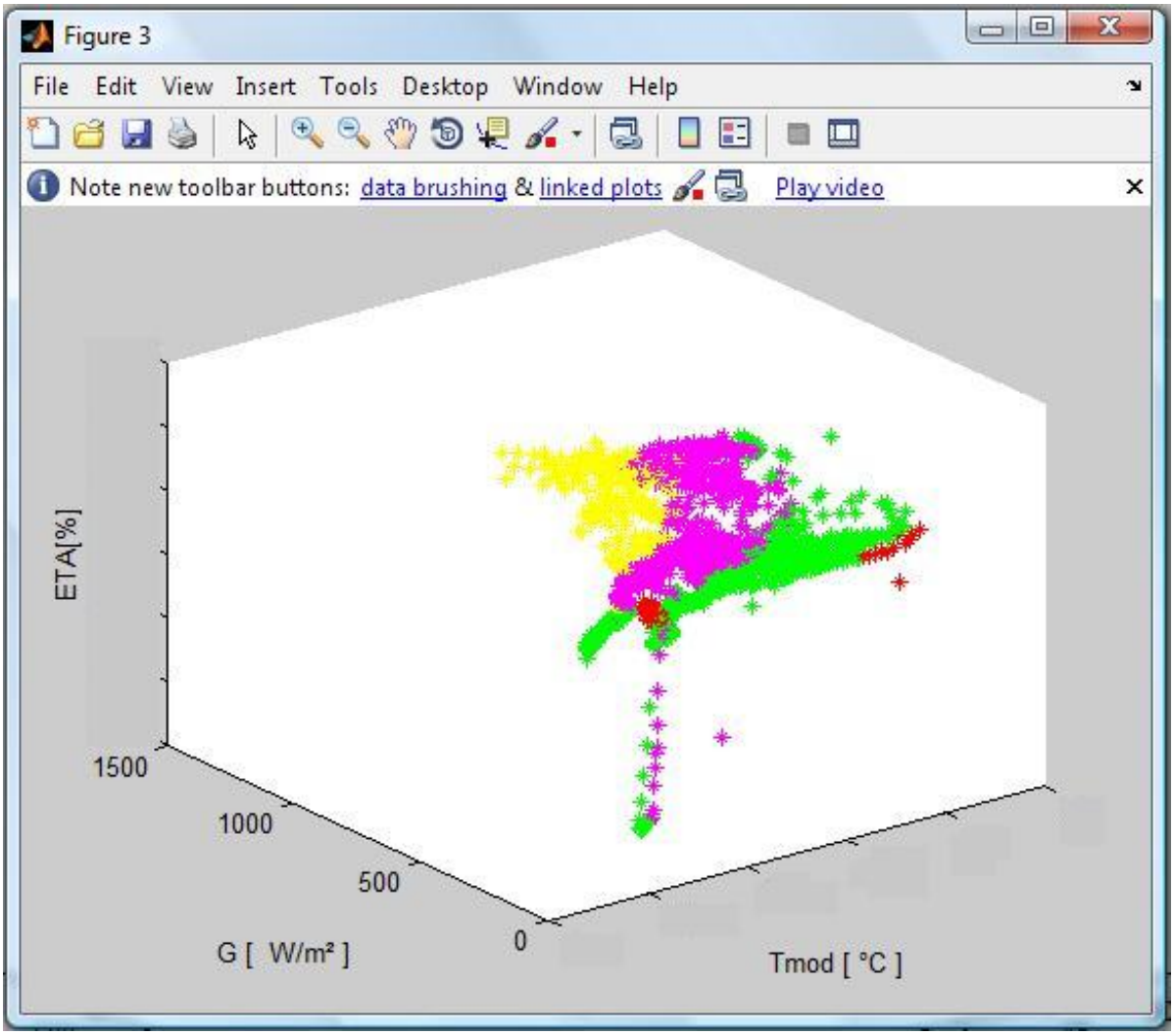


Figure 3
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
ETA[%]
1500
1000
500
0
G [W/m²]
Tmod [°C]

```
function varargout = graphtridwo(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @graphtridwo_OpeningFcn, ...
                   'gui_OutputFcn',  @graphtridwo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function graphtridwo_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

global MDM titMDM titchMDM fechaMDM c_a nu sort rxo ryo rzo gxo gyo gzo
pxo pyo pzo yxo yyo yzo bxo byo bzo
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
set(handles.date, 'String', fechaMDM)
rxo=[];
ryo=[];
rzo=[];
gxo=[];
gyo=[];
gzo=[];
pxo=[];
pyo=[];
pzo=[];
yxo=[];
yyo=[];
yzo=[];
bxo=[];
byo=[];
bzo=[];
% c=polyfit(MDM(:,2), MDM(:,5),1);
% l=length(MDM(:,1));
% aplin8 = zeros(l,1);
% for i=1:l
% aplin8(i,1)= c(1)*MDM(i,2)+ c(2);
% end
%
% plot(handles.eje10,  MDM(:,2), MDM(:,5), 'b.',MDM(:,2),aplin8(:,1),'--
y')
num_elem=cell(5,2);
set(handles.table,'Data',num_elem);
sort=0;
```

```
function varargout = graphtridwo_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupz_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM y x z c_a nu zmark ztag
strz = get(hObject, 'String');
valz = get(hObject,'Value');
switch strz{valz};
    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        z = horas;
        zmark=0;
        ztag='Time';
    case 'G [  W/m² ]'
        z = MDM(:,1);
        zmark=1;
        ztag='G [  W/m² ]';
    case 'Tmod [ °C ]'
        z = MDM(:,2);
        zmark=2;
        ztag='Tmod [ °C ]';
    case 'Umpp [ V ]'
        z = MDM(:,3);
        zmark=3;
        ztag='Umpp [ V ]';
    case 'Impp [ A ]'
        z = MDM(:,4);
        zmark=4;
        ztag='Impp [ A ]';
    case 'Pmpp [ W ]'
        z = MDM(:,5);
        zmark=5;
```

```
        ztag='Pmpp [ W ]';
    case 'Uoc [ V ]'
        z = MDM(:,6);
        zmark=6;
        ztag='Uoc [ V ]';
    case 'Isc [ A ]'
        z = MDM(:,7);
        zmark=7;
        ztag='Isc [ A ]';
    case 'FF [ % ]'
        z = MDM(:,8);
        zmark=8;
        ztag='FF [ % ]';
    case 'Pmpp,STC [ W ]'
        z = MDM(:,9);
        zmark=9;
        ztag='Pmpp,STC [ W ]';
    case 'Uoc,STC [ V ]'
        z = MDM(:,10);
        zmark=10;
        ztag='Uoc,STC [ V ]';
    case 'Isc,STC [ A ]'
        z = MDM(:,11);
        zmark=11;
        ztag='Isc,STC [ A ]';
    case 'FF,STC [ % ]'
        z = MDM(:,12);
        zmark=12;
        ztag='FF,STC [ % ]';
    case 'Rs [ Ohm ]'
        z = MDM(:,13);
        zmark=13;
        ztag='Rs [ Ohm ]';
    case 'Rp [ Ohm ]'
        z = MDM(:,14);
        zmark=14;
        ztag='Rp [ Ohm ]';
    case 'ETA[%]'
        z = nu(:);
        zmark=15;
        ztag='ETA[%]';
end

function popupz_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupx_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM y x z c_a nu xmark xtag
strx = get(hObject, 'String');
valx = get(hObject,'Value');
switch strx{valx};

    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
```

```
    x = horas;
    xmark=0;
    xtag='Time';
case 'G [  W/m² ]'
    x = MDM(:,1);
    xmark=1;
    xtag='G [  W/m² ]';
case 'Tmod [ °C ]'
    x = MDM(:,2);
    xmark=2;
    xtag='Tmod [ °C ]';
case 'Umpp [ V ]'
    x = MDM(:,3);
    xmark=3;
    xtag='Umpp [ V ]';
case 'Impp [ A ]'
    x = MDM(:,4);
    xmark=4;
    xtag='Impp [ A ]';
case 'Pmpp [ W ]'
    x = MDM(:,5);
    xmark=5;
    xtag='Pmpp [ W ]';
case 'Uoc [ V ]'
    x = MDM(:,6);
    xmark=6;
    xtag='Uoc [ V ]';
case 'Isc [ A ]'
    x = MDM(:,7);
    xmark=7;
    xtag='Isc [ A ]';
case 'FF [ % ]'
    x = MDM(:,8);
    xmark=8;
    xtag='FF [ % ]';
case 'Pmpp,STC [ W ]'
    x = MDM(:,9);
    xmark=9;
    xtag='Pmpp,STC [ W ]';
case 'Uoc,STC [ V ]'
    x = MDM(:,10);
    xmark=10;
    xtag='Uoc,STC [ V ]';
case 'Isc,STC [ A ]'
    x = MDM(:,11);
    xmark=11;
    xtag='Isc,STC [ A ]';
case 'FF,STC [ % ]'
    x = MDM(:,12);
    xmark=12;
    xtag='FF,STC [ % ]';
case 'Rs [ Ohm ]'
    x = MDM(:,13);
    xmark=13;
    xtag='Rs [ Ohm ]';
case 'Rp [ Ohm ]'
    x = MDM(:,14);
```

```
        xmark=14;
        xtag='Rp [ Ohm ]';
    case 'ETA[%]'
        x = nu(:);
        xmark=15;
        xtag='ETA[%]';
end
plot3(handles.eje10, x, y, z, 'b.')

function popupx_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupselection_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM  x c_a nu z s smark
strs = get(hObject, 'String');
vals = get(hObject,'Value');
switch strs{vals};

    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        s = horas;
        smark=0;
    case 'G [  W/m² ]'
        s = MDM(:,1);
        smark=1;
    case 'Tmod [ °C ]'
        smark=1;
        s = MDM(:,2);
    case 'Umpp [ V ]'
        s = MDM(:,3);
        smark=1;
    case 'Impp [ A ]'
        s = MDM(:,4);
        smark=1;
    case 'Pmpp [ W ]'
        s = MDM(:,5);
        smark=1;
    case 'Uoc [ V ]'
        s = MDM(:,6);
        smark=1;
    case 'Isc [ A ]'
        s = MDM(:,7);
        smark=1;
    case 'FF [ % ]'
        s = MDM(:,8);
        smark=1;
    case 'Pmpp,STC [ W ]'
        smark=1;
        s = MDM(:,9);
    case 'Uoc,STC [ V ]'
        smark=1;
        s = MDM(:,10);
    case 'Isc,STC [ A ]'
```

```
        smark=1;
        s = MDM(:,11);
    case 'FF,STC [ % ]'
        smark=1;
        s = MDM(:,12);
    case 'Rs [ Ohm ]'
        smark=1;
        s = MDM(:,13);
    case 'Rp [ Ohm ]'
        smark=1;
        s = MDM(:,14);
    case 'ETA[%]'
        smark=1;
        s = nu(:);
end

mins=min(s);
maxs=max(s);
if smark==0
    warndlg('Always hh:mm:ss and if you want to put an empty field, you
must writen 00:00:00','Bad Input','modal')
    uicontrol(hObject)

end
if smark==1
    warndlg('If you want to put an empty field, you must writen 0','Bad
Input','modal')
    uicontrol(hObject)

end
if smark==0
    timemin=datevec(mins);
    hhmin=num2str(timemin(end-2));
    mmmin=num2str(timemin(end-1));
    if timemin(end-1)<10
        mmmin=['0' mmmin];
    end
    ssmin=num2str(timemin(end));
    if timemin(end)<10
        ssmin=['0' ssmin];
    end
    mins=[hhmin ':' mmmin ':' ssmin];
    timemax=datevec(maxs);
    hhmax=num2str(timemax(end-2));
    mmmax=num2str(timemax(end-1));
    if timemax(end-1)<10
        mmmax=['0' mmmax];
    end
    ssmax=num2str(timemax(end));
    if timemax(end)<10
        ssmax=['0' ssmax];
    end
    maxs=[hhmax ':' mmmax ':' ssmax];
end
set(handles.minbox, 'String', mins)
set(handles.maxbox, 'String', maxs)
```

```
function popupselection_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function minbox_Callback(hObject, eventdata, handles)

function minbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function maxbox_Callback(hObject, eventdata, handles)

function maxbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sortbutton_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM y x c_a nu z s sort rxo ryo rzo gxo
gyo gzo pxo pyo pzo yxo yyo yzo bxo byo bzo zmark smark
a=get(handles.table, 'Data');
% rmin=0;
% rmax=0;
% gmin=0;
% gmax=0;
% pmin=0;
% pmax=0;
% ymin =0;
% ymax=0;
% bmin =0;
% bmax=0;
sort=1;
if smark==0
    rmin=datenum(a{1,1},'HH:MM:SS');
    rmax=datenum(a{1,2},'HH:MM:SS');
    gmin=datenum(a{2,1},'HH:MM:SS');
    gmax=datenum(a{2,2},'HH:MM:SS');
    pmin=datenum(a{3,1},'HH:MM:SS');
    pmax=datenum(a{3,2},'HH:MM:SS');
    ymin=datenum(a{4,1},'HH:MM:SS');
    ymax=datenum(a{4,2},'HH:MM:SS');
    bmin=datenum(a{5,1},'HH:MM:SS');
    bmax=datenum(a{5,2},'HH:MM:SS');

else
    rmin=str2num(strrep(a{1,1},',','.'));
    rmax=str2num(strrep(a{1,2},',','.'));
    gmin=str2num(strrep(a{2,1},',','.'));
    gmax=str2num(strrep(a{2,2},',','.'));
    pmin=str2num(strrep(a{3,1},',','.'));
    pmax=str2num(strrep(a{3,2},',','.'));
```

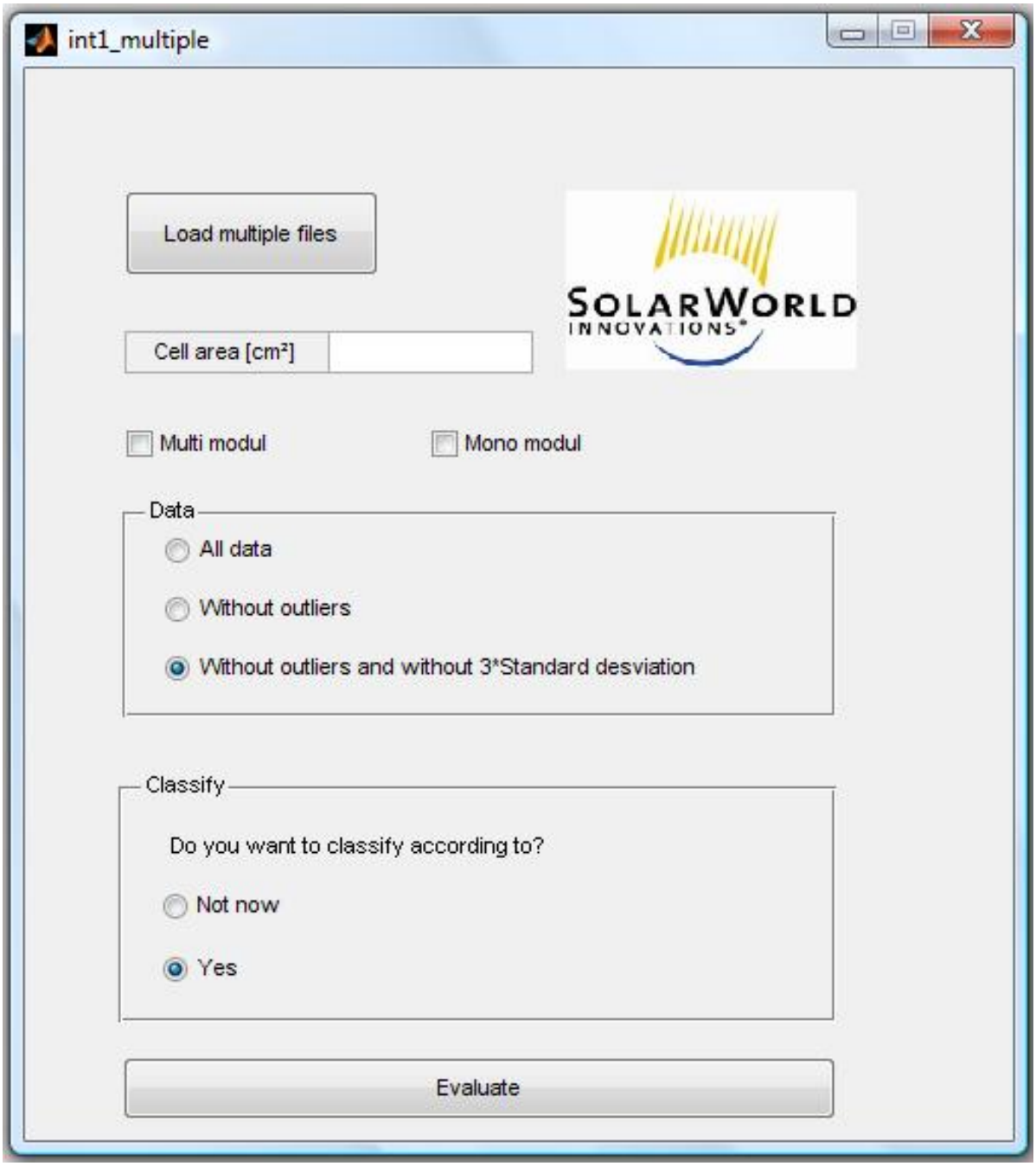
```
    ymin =str2num(strrep(a{4,1},',','.'));
    ymax=str2num(strrep(a{4,2},',','.'));
    bmin =str2num(strrep(a{5,1},',','.'));
    bmax=str2num(strrep(a{5,2},',','.'));
end

l=length(MDM(:,1));
rx=[];
ry=[];
rz=[];
gx=[];
gy=[];
gz=[];
px=[];
py=[];
pz=[];
yx=[];
yy=[];
yz=[];
bx=[];
by=[];
bz=[];
for i=1:l
    if ((rmin<s(i)) && (rmax>s(i)))
        rx(i)=x(i);
        ry(i)=y(i);
        rz(i)=z(i);
    end
    if ((gmin<s(i)) && (gmax>s(i)))
        gx(i)=x(i);
        gy(i)=y(i);
        gz(i)=z(i);
    end
    if ((pmin<s(i)) && (pmax>s(i)))
        px(i)=x(i);
        py(i)=y(i);
        pz(i)=z(i);
    end
    if ((ymin<s(i)) && (ymax>s(i)))
        yx(i)=x(i);
        yy(i)=y(i);
        yz(i)=z(i);
    end
    if ((bmin<s(i)) && (bmax>s(i)))
        bx(i)=x(i);
        by(i)=y(i);
        bz(i)=z(i);
    end
end

indr=find(rx(:)~=0);
lr=length(indr);
% rxo=zeros(lr,1);
% ryo=zeros(lr,1);
% rzo=zeros(lr,1);
for i=1:lr
rxo(i,1)=rx(indr(i));
```

```
ryo(i,1)=ry(indr(i));
rzo(i,1)=rz(indr(i));
end
indg=find(gx(:)~=0);
lg=length(indg);
% gxo=zeros(lg,1);
% gyo=zeros(lg,1);
% gzo=zeros(lg,1);
for i=1:lg
gxo(i,1)=gx(indg(i));
gyo(i,1)=gy(indg(i));
gzo(i,1)=gz(indg(i));
end
indp=find(px(:)~=0);
lp=length(indp);
% pxo=zeros(lp,1);
% pyo=zeros(lp,1);
% pzo=zeros(lp,1);
for i=1:lp
pxo(i,1)=px(indp(i));
pyo(i,1)=py(indp(i));
pzo(i,1)=pz(indp(i));
end
indy=find(yx(:)~=0);
ly=length(indy);
% yxo=zeros(ly,1);
% yyo=zeros(ly,1);
% yzo=zeros(ly,1);
for i=1:ly
yxo(i,1)=yx(indy(i));
yyo(i,1)=yy(indy(i));
yzo(i,1)=yz(indy(i));
end
indb=find(bx(:)~=0);
lb=length(indb);
% bxo=zeros(lb,1);
% byo=zeros(lb,1);
% bzo=zeros(lb,1);
for i=1:lb
bxo(i,1)=bx(indb(i));
byo(i,1)=by(indb(i));
bzo(i,1)=bz(indb(i));
end

plot3(handles.eje10,rxo(:),ryo(:),rzo(:),'r*',gxo(:),gyo(:),gzo(:),'g*',p
xo(:),pyo(:),pzo(:),'m*',yxo(:), yyo(:), yzo(:), 'y*', bxo(:),byo(:),
bzo(:),'c*')
%plot3(handles.eje10,rx(:),ry(:),rz(:),'r*',gx(:),gy(:),gz(:),'g*',px(:),
py(:),pz(:),'m*',yx(:), yy(:), yz(:), 'y*', bx(:),by(:), bz(:),'c*')

function savebutton_Callback(hObject, eventdata, handles)
global xmark xtag ymark ytag zmark ztag x y z sort  rxo ryo rzo gxo gyo
gzo pxo pyo pzo yxo yyo yzo bxo byo bzo rx ry gx gy px py yx yy bx by rz
gz pz bz yz
rgb = getframe(handles.eje10);
figure
if sort==1
```

```
plot3(rxo(:),ryo(:),rzo(:),'r*',gxo(:),gyo(:),gzo(:),'g*',pxo(:),pyo(:),p
zo(:),'m*',yxo(:), yyo(:),yzo(:), 'y*', bxo(:),byo(:),bzo(:),'c*')
else
    plot3(x,y,z,'b*')
end

if xmark==0

    datetick('x','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
    zlabel(ztag)

elseif ymark==0
%     figure
%     %     plot(x,y)
%     image(rgb.cdata)
    datetick('y','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
    zlabel(ztag)
%     hold off
elseif zmark==0
%     figure
%     image(rgb.cdata)
    %     plot(x,y)
    datetick('z','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
    zlabel(ztag)
%     hold off
else
    xlabel(xtag)
    ylabel(ytag)
    zlabel(ztag)
end

% figure
% image(rgb.cdata)

function popupy_Callback(hObject, eventdata, handles)
global MDM titMDM titchMDM fechaMDM y x z c_a nu ymark ytag
stry = get(hObject, 'String');
valy = get(hObject,'Value');
switch stry{valy};

    case 'Time'
        horas=datenum(titchMDM,'HH:MM:SS');
        y = horas;
        ymark=0;
        ytag='Time';
    case 'G [  W/m² ]'
        y = MDM(:,1);
        ymark=1;
        ytag='G [  W/m² ]';
```

```
    case 'Tmod [ °C ]'
        y = MDM(:,2);
        ymark=2;
        ytag='Tmod [ °C ]';
    case 'Umpp [ V ]'
        y = MDM(:,3);
        ymark=3;
        ytag='Umpp [ V ]';
    case 'Impp [ A ]'
        y = MDM(:,4);
        ymark=4;
        ytag='Impp [ A ]';
    case 'Pmpp [ W ]'
        y = MDM(:,5);
        ymark=5;
        ytag='Pmpp [ W ]';
    case 'Uoc [ V ]'
        y = MDM(:,6);
        ymark=6;
        ytag='Uoc [ V ]';
    case 'Isc [ A ]'
        y = MDM(:,7);
        ymark=7;
        ytag='Isc [ A ]';
    case 'FF [ % ]'
        y = MDM(:,8);
        ymark=8;
        ytag='FF [ % ]';
    case 'Pmpp,STC [ W ]'
        y = MDM(:,9);
        ymark=9;
        ytag='Pmpp,STC [ W ]';
    case 'Uoc,STC [ V ]'
        y = MDM(:,10);
        ymark=10;
        ytag='Uoc,STC [ V ]';
    case 'Isc,STC [ A ]'
        y = MDM(:,11);
        ymark=11;
        ytag='Isc,STC [ A ]';
    case 'FF,STC [ % ]'
        y = MDM(:,12);
        ymark=12;
        ytag='FF,STC [ % ]';
    case 'Rs [ Ohm ]'
        y = MDM(:,13);
        ymark=13;
        ytag='Rs [ Ohm ]';
    case 'Rp [ Ohm ]'
        y = MDM(:,14);
        ymark=14;
        ytag='Rp [ Ohm ]';
    case 'ETA[%]'
        y = nu(:);
        ymark=15;
        ytag='ETA[%]';
end
```

F.4. Evaluation Multimodul

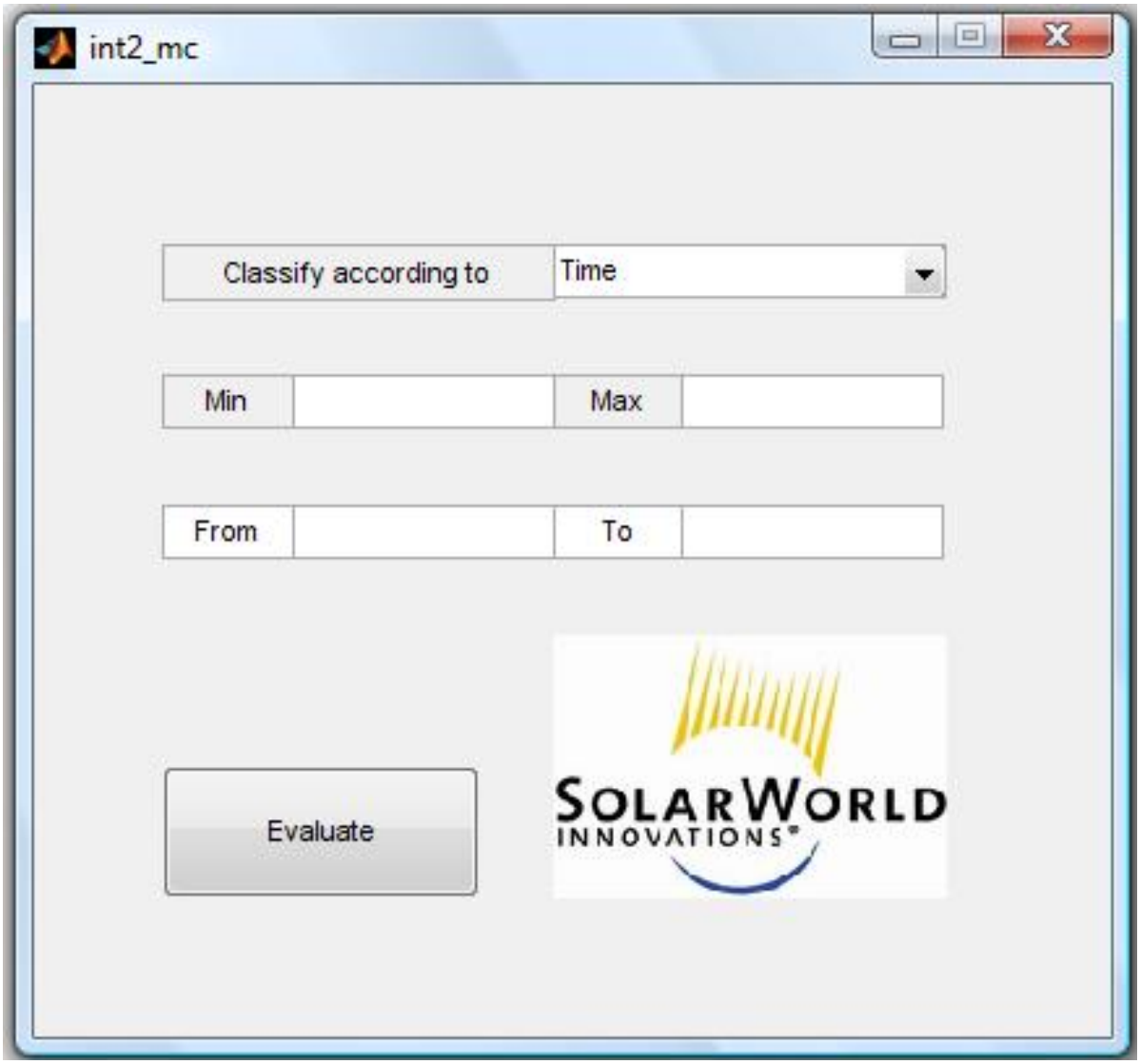


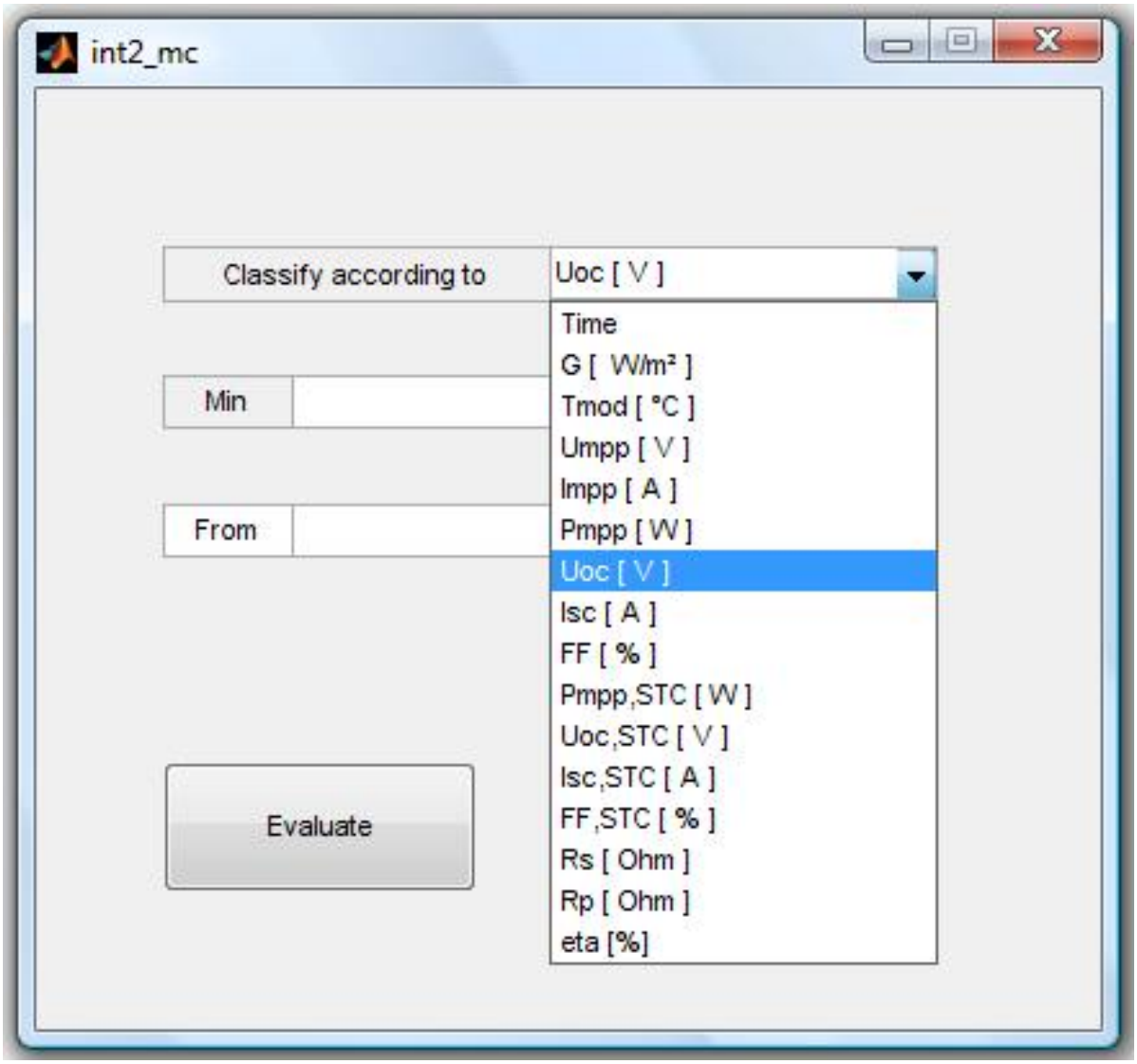
```
function varargout = int1_multiple(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int1_multiple_OpeningFcn, ...
                   'gui_OutputFcn',  @int1_multiple_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int1_multiple_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = int1_multiple_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function load_multiple_Callback(hObject, eventdata, handles)
% load many files of all kind of files
global MDM tchour MDM_old Irr filename l_i
global TKisc TKvoc TKpmpp titchMDM
[filename, pathname, filterindex] = uigetfile( {'*.xls','EXCEL-files
(*.xls)'; '*.txt','Text files (*.txt)'; '*.*', 'All Files (*.*)'}, 'Pick
a file', 'MultiSelect', 'on');
% how many files are there
l=length(filename);
%intialize variables
MDM=[];
MDM_old=[];
MDM_new=[];
MDM_pro=[];
tchour_old=[];
tchour_new=[];
tchour_newc=[];
tchour_realc=[];
tchour_pro=[];
titchMDM=[];
%for all files
for i=1:l
    %read the name of the file and save it in 's'
    s = char(filename{1,i});
    % read extension
    ext = s(end-2:end);
    MDM_new=[];
```

```
    %if excel file
    if ext=='XLS'
        %read the file
        [MDM_new,titMDM]=xlsread(s);
        % go to function ordenxls, that put it in order
        MDM_new = ordenxls(MDM_new);
        %how many data are there
        lxls=length(MDM_new(:,1));
        % take the column, that has date and time
        t=zeros(lxls,19);
        for i3=3:lxls+2
            t(i3-2,:)=titMDM{i3,2};
        end
        % to convert it to character
        tcdate=char(t(:,1:10));
        tchour_new=char(t(:,12:end));
        % make a vector  to time with numbers
        horan=datenum(tchour_new,'HH:MM:SS');
        valor=zeros(lxls-1);
        %calculate Irradiance
        for i2=2:lxls
            valor(i2-1)=(horan(i2)-horan(i2-1))*MDM_new(i2-1,1);
        end
        Irr(i)=sum(valor(:,1))*24;
      %if the asc file
    elseif ext=='asc'
        %open the file, save the number in fid
        fid = fopen(s);
        % don´t read first two rows
        C_text=textscan(fid,'%*[^\n]',2);
        % read all data
        C_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s ');
        % how many data there are
        ltxt=length(C_datos{1,1});
        C_datosp=zeros(ltxt,12);
        % convert ',' in a '.'
        for i4=1:ltxt
            for j=4:11
                MDM_new(i4,j-
3)=str2num(strrep(C_datos{1,j}{i4,1},',','.'));
            end
        end
       % call the function to put the matrix in orden.
        MDM_new = ordentxt(MDM_new,TKpmpp,TKisc,TKvoc);

        lltxt=length(C_datos{3});
        tchour_newc=[];
        tchour_new=[];
        % read time column and save it in tchour_newc
        for i5=1:lltxt
            tchour_newc(i5,:)=C_datos{1,3}{i5,1};
        end

        % convert it in character
        tchour_new=char(tchour_newc);
        % make a vector with the numbers corresponding to time
```

```
        horan=datenum(tchour_new,'HH:MM:SS');
        % calculate irradiance
        valor=zeros(lltxt-1);
        for i5=2:lltxt
            valor(i5-1)=(horan(i5)-horan(i5-1))*MDM_new(i5-1,1);
        end
        Irr(i)=sum(valor(:,1))*24;
        % close file
        fclose(fid);
    end
    % collect up old matrix with the new to make our matrix
    MDM_pro=[MDM_old;MDM_new];
    MDM_old=MDM_pro;
    % collect up old vector to time with the new.
    tchour_pro=[tchour_old;tchour_new];
    tchour_old=tchour_pro;
end
MDM=MDM_old;
tchour=tchour_old;
l_i=length(MDM);

function panel_process_SelectionChangeFcn(hObject, eventdata, handles)
global MDM tchour titchMDM l_f marcador
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'ad'
        marcador=1;
        % Code for when nn is selected.
        % to all data
    case 'wo'
        marcador=2;
        % Code for when yes is selected.
        % without outliers
    case 'wowsd'
       marcador=3;
        % without outliers and without 3*standar desviation
        % Code for when togglebutton1 is selected.
end

function panel_classify_SelectionChangeFcn(hObject, eventdata, handles)
global MDM tchour titchMDM nn
% go on to sort or not
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'nn'
        % not now, so we make a standard evaluation
        nn=1;

    case 'yes'
        % yes, now we can make other classification
        nn=0;

end
nn;

function area_Callback(hObject, eventdata, handles)
global c_a
b=get(handles.area,'String');
```

```
c=strrep(b, ',','.');
c_a=str2num(c);

function area_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function evaluation_multiple_Callback(hObject, eventdata, handles)
global nn marcador MDM titchMDM tchour l_f
switch marcador
    case 1
        % to all data, remain so.
        MDM=MDM;
        l_f=length(MDM);
        titchMDM=tchour;
        l_titchMDM=length(titchMDM);
    case 2
        % call to function without outliers
        [MDM, titchMDM] = outlierRemoval(MDM, tchour);
        l_f=length(MDM);
        l_titchMDM=length(titchMDM);
    case 3
        % call to function to get out outliers and standard desviation
        [MDM, titchMDM] = outlierRemoval_sosds(MDM, tchour);
        l_f=length(MDM);
        l_titchMDM=length(titchMDM);
end

if nn==1
    % standard evaluation
    int2_multiple

else
    % other classification
    int2_mc
end

function multibox_Callback(hObject, eventdata, handles)
global TKisc TKvoc TKpmpp
% Values of temperatur coefficients to multicristaline modul
TKisc=0.034;
TKvoc=-0.34;
TKpmpp=-0.48;

function monobox_Callback(hObject, eventdata, handles)
% Values of temperatur coefficients to monocristaline modul
global TKisc TKvoc TKpmpp
TKisc=0.042;
TKvoc=-0.33;
TKpmpp=-0.45;
```


int2_mc
Classify according to
Time
Min
Max
From
To
Evaluate
SOLARWORLD
INNOVATIONS


int2_mc
Classify according to
Uoc [V]
Time
G [W/m²]
Tmod [°C]
Umpp [V]
Impp [A]
Pmpp [W]
Uoc [V]
Isc [A]
FF [%]
Pmpp,STC [W]
Uoc,STC [V]
Isc,STC [A]
FF,STC [%]
Rs [Ohm]
Rp [Ohm]
eta [%]
Min
From
Evaluate

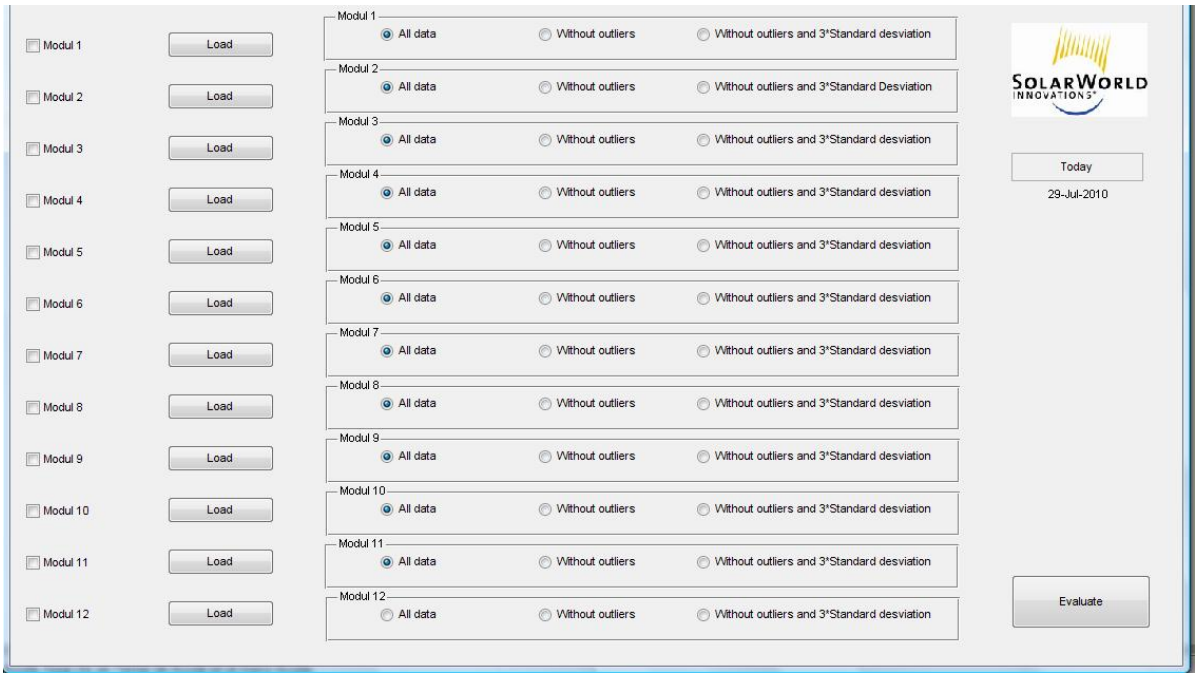
```
function varargout = int2_mc(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @int2_mc_OpeningFcn, ...
                   'gui_OutputFcn',  @int2_mc_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function int2_mc_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off

function varargout = int2_mc_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function popupmenu_mc_Callback(hObject, eventdata, handles)
% select a variable, and then show maximum and minimum
global y c_a MDM titchMDM zmark
stry = get(hObject, 'String');
valy = get(hObject,'Value');
switch stry{valy};

    case 'Time'

        horas=datenum(titchMDM,'HH:MM:SS');
        y = horas;
        zmark=0;
    case 'G [  W/m² ]'
        y = MDM(:,1);
         zmark=1;
    case 'Tmod [ °C ]'
         zmark=1;
        y = MDM(:,2);
    case 'Umpp [ V ]'
         zmark=1;
        y = MDM(:,3);
    case 'Impp [ A ]'
         zmark=1;
        y = MDM(:,4);
    case 'Pmpp [ W ]'
         zmark=1;
        y = MDM(:,5);
```

```
    case 'Uoc [ V ]'
         zmark=1;
        y = MDM(:,6);
    case 'Isc [ A ]'
         zmark=1;
        y = MDM(:,7);
    case 'FF [ % ]'
         zmark=1;
        y = MDM(:,8);
    case 'Pmpp,STC [ W ]'
         zmark=1;
        y = MDM(:,9);
    case 'Uoc,STC [ V ]'
         zmark=1;
        y = MDM(:,10);
    case 'Isc,STC [ A ]'
         zmark=1;
        y = MDM(:,11);
    case 'FF,STC [ % ]'
         zmark=1;
        y = MDM(:,12);
    case 'Rs [ Ohm ]'
         zmark=1;
        y = MDM(:,13);
    case 'Rp [ Ohm ]'
         zmark=1;
        y = MDM(:,14);
    case 'eta [%]'
         zmark=1;
        c_a_m=c_a/10000;
        nu = MDM(:,5)./(c_a_m.*MDM(:,1));
        y = nu(:);
end
% minimum and maximum of variable
miny = min(y);
maxy = max(y);
% if time is selected
if zmark==0
    timemin=datevec(miny);
    hhmin=num2str(timemin(end-2));
    mmmin=num2str(timemin(end-1));
    if timemin(end-1)<10
        mmmin=['0' mmmin];
    end
    ssmin=num2str(timemin(end));
    if timemin(end)<10
        ssmin=['0' ssmin];
    end
    miny=[hhmin ':' mmmin ':' ssmin];
    timemax=datevec(maxy);
    hhmax=num2str(timemax(end-2));
    mmmax=num2str(timemax(end-1));
    if timemax(end-1)<10
        mmmax=['0' mmmax];
    end
    ssmax=num2str(timemax(end));
    if timemax(end)<10
```

```
        ssmax=['0' ssmax];
    end
    maxy=[hhmax ':' mmmax ':' ssmax];
end
set(handles.min_mc, 'String', miny)
set(handles.max_mc, 'String', maxy)

function popupmenu_mc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function min_mc_Callback(hObject, eventdata, handles)

function min_mc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function max_mc_Callback(hObject, eventdata, handles)
function max_mc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)

function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function from_mc_Callback(hObject, eventdata, handles)

function from_mc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function to_mc_Callback(hObject, eventdata, handles)
```

```
function to_mc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function evaluate_mc_Callback(hObject, eventdata, handles)
% take superior value and inferior value
global y MDM titchMDM l_v_MDM zmark
a=get(handles.from_mc, 'String');
b=get(handles.to_mc, 'String');
%make it numbers
if zmark==0
    liminf = datenum(a,'HH:MM:SS');
    limsup=datenum(b,'HH:MM:SS');
else
    limina=strrep(a, ',','.');
    liminf=str2num(limina);
    limsupb=strrep(b, ',','.');
    limsup=str2num(limsupb);
end
%and take the matrix values, that are between the superior and inferior
v_MDM=[];
v_MDM=find((limsup>y) & (y>liminf));
l_v_MDM=length(v_MDM);
if l_v_MDM==0
    errordlg('You must another rank choose, there are not enough
data','Bad Input','modal')
    uicontrol(hObject)
    return
end
MDM_2=[];
titchMDM_2=[];
for i=1:l_v_MDM
    MDM_2(i,:)=MDM(v_MDM(i),:);
    titchMDM_2(i,:)=titchMDM(v_MDM(i),:);
end

MDM=MDM_2;
titchMDM=char(titchMDM_2);
%go to int3_multiple_classify, that is as int3wo
int3_multiple_classify
```

F.5. Comparison Multimodul



```
function varargout = compare_12_try2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compare_12_try2_OpeningFcn, ...
                   'gui_OutputFcn',  @compare_12_try2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function compare_12_try2_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

hoy=date;
set(handles.today, 'String', hoy)
global car1_m_1 car1_m_2 car1_m_3 car1_m_4 car1_m_5 car1_m_6 car1_m_7
car1_m_8 car1_m_9 car1_m_10 car1_m_11 car1_m_12
global s_m_1 s_m_2 s_m_3 s_m_4 s_m_5 s_m_6 s_m_7 s_m_8 s_m_9 s_m_10
s_m_11 s_m_12
global ext_m_1 ext_m_2 ext_m_3 ext_m_4 ext_m_5 ext_m_6 ext_m_7 ext_m_8
ext_m_9 ext_m_10 ext_m_11 ext_m_12
global process_m_1 process_m_2 process_m_3 process_m_4 process_m_5
process_m_6 process_m_7 process_m_8 process_m_9 process_m_10 process_m_11
process_m_12

car1_m_1=0;
car1_m_2=0;
car1_m_3=0;
car1_m_4=0;
car1_m_5=0;
car1_m_6=0;
car1_m_7=0;
car1_m_8=0;
car1_m_9=0;
car1_m_10=0;
car1_m_11=0;
car1_m_12=0;
s_m_1=0;
s_m_2=0;
s_m_3=0;
s_m_4=0;
s_m_5=0;
```

```
s_m_6=0;
s_m_7=0;
s_m_8=0;
s_m_9=0;
s_m_10=0;
s_m_11=0;
s_m_12=0;

ext_m_1=[];
ext_m_2=[];
ext_m_3=[];
ext_m_4=[];

ext_m_5=[];
ext_m_6=[];
ext_m_7=[];
ext_m_8=[];

ext_m_9=[];
ext_m_10=[];
ext_m_11=[];
ext_m_12=[];

process_m_1=1;
process_m_2=1;
process_m_3=1;
process_m_4=1;

process_m_5=1;
process_m_6=1;
process_m_7=1;
process_m_8=1;

process_m_9=1;
process_m_10=1;
process_m_11=1;
process_m_12=1;

function varargout = compare_12_try2_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function m_1_Callback(hObject, eventdata, handles)
global car1_m_1
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_1=1;
else
    car1_m_1=0;
    % Checkbox is not checked-take approriate action
end


function m_2_Callback(hObject, eventdata, handles)
global car1_m_2
% car1_m_2=0;
```

```
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_2=1;
else
    car1_m_2=0;
    % Checkbox is not checked-take approriate action
end

function m_3_Callback(hObject, eventdata, handles)
global car1_m_3
% car1_m_3=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_3=1;
else
    car1_m_3=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_3

function m_4_Callback(hObject, eventdata, handles)
global car1_m_4
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_4=1;
else
    car1_m_4=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_4

function m_5_Callback(hObject, eventdata, handles)
global car1_m_5
%car1_m_5=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_5=1;
else
    car1_m_5=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_5

function m_6_Callback(hObject, eventdata, handles)
global car1_m_6
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_6=1;
else
    car1_m_6=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_6

function m_7_Callback(hObject, eventdata, handles)
global car1_m_7
```

```
%car1_m_7=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_7=1;
else
    car1_m_7=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_7

function m_8_Callback(hObject, eventdata, handles)
global car1_m_8
%car1_m_8=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_8=1;
else
    car1_m_8=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_8

function m_9_Callback(hObject, eventdata, handles)
global car1_m_9
%car1_m_9=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_9=1;
else
    car1_m_9=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_9

function m_10_Callback(hObject, eventdata, handles)
global car1_m_10
%car1_m_10=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_10=1;
else
    car1_m_10=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_10

function m_12_Callback(hObject, eventdata, handles)
global car1_m_12

%car1_m_12=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
```

```
    car1_m_12=1;
else
    car1_m_12=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_12

function m_11_Callback(hObject, eventdata, handles)
global car1_m_11
%car1_m_11=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car1_m_11=1;
else
    car1_m_11=0;
    % Checkbox is not checked-take approriate action
end
%car1_m_11

function l_1_Callback(hObject, eventdata, handles)
global ext_m_1 s_m_1
% ext_m_1=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_1 = handles.fileName;
s_m_1 = char(fileName_m_1);
ext_m_1 = s_m_1(end-2:end);



% % % MDM_new=[];
% % % if ext=='XLS'
% % %     [MDM_new,titMDM]=xlsread(s);
% % %     MDM_new = ordenxls(MDM_new);
% % %     lxls=length(MDM_new(:,1));
% % %     t=zeros(lxls,19);
% % %     for i3=3:lxls+2
% % %         t(i3-2,:)=titMDM{i3,2};
% % %     end
% % %     tcdate=char(t(:,1:10));
% % %     tchour_new=char(t(:,11:end));
% % %     horan=datenum(tchour_new,'HH:MM:SS');
% % %     valor=zeros(lxls-1);
% % %     for i2=2:lxls
% % %         valor(i2-1)=(horan(i2)-horan(i2-1))*MDM_new(i2-1,1);
% % %     end
% % %     Irr_m_1=sum(valor(:,1))*24;
% % % elseif ext=='txt'
% % %     fid = fopen(s);
% % %     C_text=textscan(fid,'%*[^\n]',2);
% % %     C_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% % %     ltxt=length(C_datos{1,1});
% % %     C_datosp=zeros(ltxt,12);
% % %     for i4=1:ltxt
% % %         for j=4:15
```

```
% % %             MDM_new(i4,j-
3)=str2num(strrep(C_datos{1,j}{i4,1},',','.'));
% % %         end
% % %     end
% % %     MDM_new = ordentxt(MDM_new);
% % %     lltxt=length(C_datos{3});
% % %     for i5=1:lltxt
% % %         tchour_newc(i5,:)=C_datos{1,2}{i5,1};
% % %     end
% % %     %         C_datos{1,3}{1,
% % %     %tchour_new=char(tchour_newc);
% % %     tchour_newchar=char(tchour_newc);
% % %     for i8=1:lltxt
% % %         if C_datos{1,3}{i8,1}=='PM'
% % %             hour1=str2num(tchour_newchar(i8,1));
% % %             realhour=hour1+12;
% % %             realhourchar=num2str(realhour);
% % %             tchour_realc(i8,1)=' ';
% % %             tchour_realc(i8,2)=realhourchar(1);
% % %             tchour_realc(i8,3)=realhourchar(2);
% % %             for i6=1:6
% % %                 tchour_realc(i8,3+i6)=tchour_newchar(i8,1+i6);
% % %             end
% % %         else
% % %             tchour_realc(i8,1)=' ';
% % %             tchour_realc(i8,2)='0';
% % %             for i7=1:7
% % %                 tchour_realc(i8,2+i7)=tchour_newchar(i8,i7);
% % %             end
% % %         end
% % %     end
% % %     tchour_new=char(tchour_realc);
% % %     horan=datenum(tchour_new,'HH:MM:SS');
% % %     valor=zeros(lltxt-1);
% % %     for i5=2:lltxt
% % %         valor(i5-1)=(horan(i5)-horan(i5-1))*MDM_new(i5-1,1);
% % %     end
% % %     Irr_m_1=sum(valor(:,1))*24;
% % %     fclose(fid);
% % % end
% % % MDM_m_1=MDM_new;
% % % tchour_m_1=tchour_new;

% C_text=textscan(fid_m_1,'%*[^\n]',2);
% C_datos_m_1=textscan(fid_m_1,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_1{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_1(i,j-
3)=str2num(strrep(C_datos_m_1{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_1);
% MDM_m_1=MDMtxt_m_1;
```

```
function l_2_Callback(hObject, eventdata, handles)
% global C_datos_m_2 MDM_m_2 fid_m_2
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_2 = handles.fileName;
% fid_m_2 = fopen(fileName_m_2);
% C_text=textscan(fid_m_2,'%*[^\n]',2);
% C_datos_m_2=textscan(fid_m_2,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_2{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_2(i,j-
3)=str2num(strrep(C_datos_m_2{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_2);
% MDM_m_2=MDMtxt_m_2;
global ext_m_2 s_m_2
% ext_m_2=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_2 = handles.fileName;
s_m_2 = char(fileName_m_2);
ext_m_2 = s_m_2(end-2:end);

function l_3_Callback(hObject, eventdata, handles)
% global C_datos_m_3 MDM_m_3 fid_m_3
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_3 = handles.fileName;
% fid_m_3 = fopen(fileName_m_3);
% C_text=textscan(fid_m_3,'%*[^\n]',2);
% C_datos_m_3=textscan(fid_m_3,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_3{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_3(i,j-
3)=str2num(strrep(C_datos_m_3{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_3);
% MDM_m_3=MDMtxt_m_3;
global ext_m_3 s_m_3
% ext_m_3=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_3 = handles.fileName;
s_m_3 = char(fileName_m_3);
ext_m_3 = s_m_3(end-2:end);

function l_4_Callback(hObject, eventdata, handles)
% global C_datos_m_4 MDM_m_4 fid_m_4
% handles.fileName = uigetfile('*.txt');
```

```
% guidata(hObject, handles);
% fileName_m_4 = handles.fileName;
% fid_m_4 = fopen(fileName_m_4);
% C_text=textscan(fid_m_4,'%*[^\n]',2);
% C_datos_m_4=textscan(fid_m_4,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_4{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_4(i,j-
3)=str2num(strrep(C_datos_m_4{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_4);
% MDM_m_4=MDMtxt_m_4;

global ext_m_4 s_m_4
% ext_m_4=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_4 = handles.fileName;
s_m_4 = char(fileName_m_4);
ext_m_4 = s_m_4(end-2:end);

function l_5_Callback(hObject, eventdata, handles)
% global C_datos_m_5 MDM_m_5 fid_m_5
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_5 = handles.fileName;
% fid_m_5 = fopen(fileName_m_5);
% C_text=textscan(fid_m_5,'%*[^\n]',2);
% C_datos_m_5=textscan(fid_m_5,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_5{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_5(i,j-
3)=str2num(strrep(C_datos_m_5{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_5);
% MDM_m_5=MDMtxt_m_5;

global ext_m_5 s_m_5
% ext_m_5=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_5 = handles.fileName;
s_m_5 = char(fileName_m_5);
ext_m_5 = s_m_5(end-2:end);

function l_6_Callback(hObject, eventdata, handles)
% global C_datos_m_6 MDM_m_6 fid_m_6
% handles.fileName = uigetfile('*.txt');
```

```
% guidata(hObject, handles);
% fileName_m_6 = handles.fileName;
% fid_m_6 = fopen(fileName_m_6);
% C_text=textscan(fid_m_6,'%*[^\n]',2);
% C_datos_m_6=textscan(fid_m_6,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_6{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_6(i,j-
3)=str2num(strrep(C_datos_m_6{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_6);
% MDM_m_6=MDMtxt_m_6;

global ext_m_6 s_m_6
% ext_m_6=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_6 = handles.fileName;
s_m_6 = char(fileName_m_6);
ext_m_6 = s_m_6(end-2:end);

function l_7_Callback(hObject, eventdata, handles)
% global C_datos_m_7 MDM_m_7 fid_m_7
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_7 = handles.fileName;
% fid_m_7 = fopen(fileName_m_7);
% C_text=textscan(fid_m_7,'%*[^\n]',2);
% C_datos_m_7=textscan(fid_m_7,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_7{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_7(i,j-
3)=str2num(strrep(C_datos_m_7{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_7);
% MDM_m_7=MDMtxt_m_7;
global ext_m_7 s_m_7
% ext_m_7=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_7 = handles.fileName;
s_m_7 = char(fileName_m_7);
ext_m_7 = s_m_7(end-2:end);

function l_8_Callback(hObject, eventdata, handles)
% global C_datos_m_8 MDM_m_8 fid_m_8
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
```

```
% fileName_m_8 = handles.fileName;
% fid_m_8 = fopen(fileName_m_8);
% C_text=textscan(fid_m_8,'%*[^\n]',2);
% C_datos_m_8=textscan(fid_m_8,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_8{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_8(i,j-
3)=str2num(strrep(C_datos_m_8{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_8);
% MDM_m_8=MDMtxt_m_8;

global ext_m_8 s_m_8
% ext_m_8=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_8 = handles.fileName;
s_m_8 = char(fileName_m_8);
ext_m_8 = s_m_8(end-2:end);

function l_9_Callback(hObject, eventdata, handles)
% global C_datos_m_9 MDM_m_9 fid_m_9
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_9 = handles.fileName;
% fid_m_9 = fopen(fileName_m_9);
% C_text=textscan(fid_m_9,'%*[^\n]',2);
% C_datos_m_9=textscan(fid_m_9,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% l=length(C_datos_m_9{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_9(i,j-
3)=str2num(strrep(C_datos_m_9{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_9);
% MDM_m_9=MDMtxt_m_9;

global ext_m_9 s_m_9
% ext_m_9=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_9 = handles.fileName;
s_m_9 = char(fileName_m_9);
ext_m_9 = s_m_9(end-2:end);

function l_10_Callback(hObject, eventdata, handles)
% global C_datos_m_10 MDM_m_10 fid_m_10
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
```

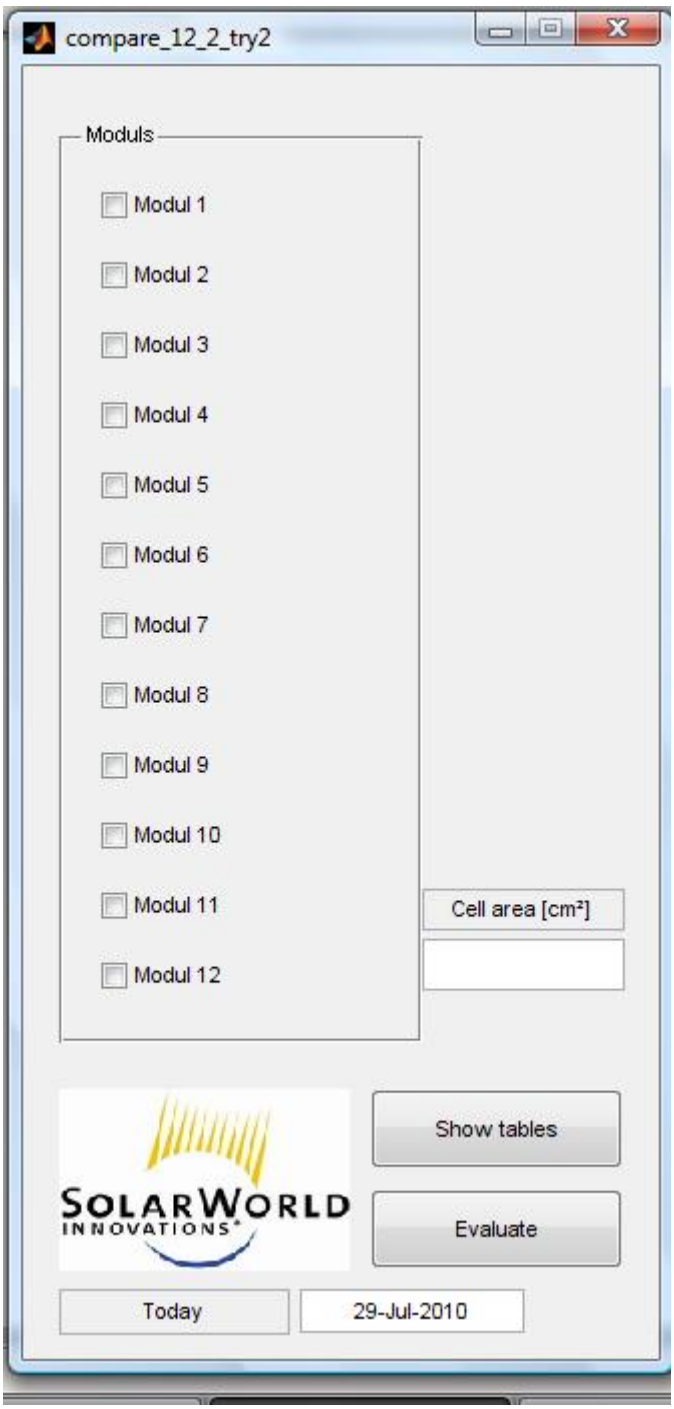
```
% fileName_m_10 = handles.fileName;
% fid_m_10 = fopen(fileName_m_10);
% C_text=textscan(fid_m_10,'%*[^\n]',2);
% C_datos_m_10=textscan(fid_m_10,'%s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s');
% l=length(C_datos_m_10{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_10(i,j-
3)=str2num(strrep(C_datos_m_10{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_10);
% MDM_m_10=MDMtxt_m_10;

global ext_m_10 s_m_10
% ext_m_10=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_10 = handles.fileName;
s_m_10 = char(fileName_m_10);
ext_m_10 = s_m_10(end-2:end);

function l_11_Callback(hObject, eventdata, handles)
% global C_datos_m_11 MDM_m_11 fid_m_11
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
% fileName_m_11 = handles.fileName;
% fid_m_11 = fopen(fileName_m_11);
% C_text=textscan(fid_m_11,'%*[^\n]',2);
% C_datos_m_11=textscan(fid_m_11,'%s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s');
% l=length(C_datos_m_11{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_11(i,j-
3)=str2num(strrep(C_datos_m_11{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_11);
% MDM_m_11=MDMtxt_m_11;

global ext_m_11 s_m_11
% ext_m_11=0;
handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_11 = handles.fileName;
s_m_11 = char(fileName_m_11);
ext_m_11 = s_m_11(end-2:end);

function l_12_Callback(hObject, eventdata, handles)
% global C_datos_m_12 MDM_m_12 fid_m_12
% handles.fileName = uigetfile('*.txt');
% guidata(hObject, handles);
```

```
% fileName_m_12 = handles.fileName;
% fid_m_12 = fopen(fileName_m_12);
% C_text=textscan(fid_m_12,'%*[^\n]',2);
% C_datos_m_12=textscan(fid_m_12,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s');
% l=length(C_datos_m_12{1,1});
% C_datosp=zeros(l,12);
% for i=1:l
%     for j=4:15
%         MDMtxt_m_12(i,j-
3)=str2num(strrep(C_datos_m_12{1,j}{i,1},',','.'));
%     end
% end
% fclose(fid_m_12);
% MDM_m_12=MDMtxt_m_12;

global ext_m_12 s_m_12

handles.fileName = uigetfile('*.txt','*.xls');
guidata(hObject, handles);
fileName_m_12 = handles.fileName;
s_m_12 = char(fileName_m_12);
ext_m_12 = s_m_12(end-2:end);

function p_m_1_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_1

switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_1_ad'
        % Code for when radiobutton1 is selected.
       process_m_1=1;
    case 'm_1_wo'
        % Code for when radiobutton2 is selected.
        process_m_1=2;
    case 'm_1_wowsd'
        process_m_1=3;
        % Code for when togglebutton1 is selected.
end

function p_m_2_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_2

switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_2_ad'
        % Code for when radiobutton1 is selected.
       process_m_2=1;
    case 'm_2_wo'
        % Code for when radiobutton2 is selected.
        process_m_2=2;
    case 'm_2_wowsd'
        process_m_2=3;
        % Code for when togglebutton1 is selected.
end

function p_m_3_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_3
```

```
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_3_ad'
        % Code for when radiobutton1 is selected.
       process_m_3=1;
    case 'm_3_wo'
        % Code for when radiobutton2 is selected.
        process_m_3=2;
    case 'm_3_wowsd'
        process_m_3=3;
        % Code for when togglebutton1 is selected.
end

function p_m_4_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_4

switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_4_ad'
        % Code for when radiobutton1 is selected.
       process_m_4=1;
    case 'm_4_wo'
        % Code for when radiobutton2 is selected.
        process_m_4=2;
    case 'm_4_wowsd'
        process_m_4=3;
        % Code for when togglebutton1 is selected.
end

function p_m_5_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_5
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_5_ad'
        % Code for when radiobutton1 is selected.
       process_m_5=1;
    case 'm_5_wo'
        % Code for when radiobutton2 is selected.
        process_m_5=2;
    case 'm_5_wowsd'
        process_m_5=3;
        % Code for when togglebutton1 is selected.
end

function p_m_6_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_6

switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_6_ad'
        % Code for when radiobutton1 is selected.
       process_m_6=1;
    case 'm_6_wo'
        % Code for when radiobutton2 is selected.
        process_m_6=2;
    case 'm_6_wowsd'
        process_m_6=3;
        % Code for when togglebutton1 is selected.
end
```

```
function p_m_7_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_7
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_7_ad'
        % Code for when radiobutton1 is selected.
       process_m_7=1;
    case 'm_7_wo'
        % Code for when radiobutton2 is selected.
        process_m_7=2;
    case 'm_7_wowsd'
        process_m_7=3;
        % Code for when togglebutton1 is selected.
end

function p_m_8_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_8
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_8_ad'
        % Code for when radiobutton1 is selected.
       process_m_8=1;
    case 'm_8_wo'
        % Code for when radiobutton2 is selected.
        process_m_8=2;
    case 'm_8_wowsd'
        process_m_8=3;
        % Code for when togglebutton1 is selected.
end

function p_m_9_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_9
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_9_ad'
        % Code for when radiobutton1 is selected.
       process_m_9=1;
    case 'm_9_wo'
        % Code for when radiobutton2 is selected.
        process_m_9=2;
    case 'm_9_wowsd'
        process_m_9=3;
        % Code for when togglebutton1 is selected.
end

function p_m_10_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_10
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_10_ad'
        % Code for when radiobutton1 is selected.
       process_m_10=1;
    case 'm_10_wo'
        % Code for when radiobutton2 is selected.
        process_m_10=2;
    case 'm_10_wowsd'
        process_m_10=3;
        % Code for when togglebutton1 is selected.
end
```

```
function p_m_11_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_11
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_11_ad'
        % Code for when radiobutton1 is selected.
       process_m_11=1;
    case 'm_11_wo'
        % Code for when radiobutton2 is selected.
        process_m_11=2;
    case 'm_11_wowsd'
        process_m_11=3;
        % Code for when togglebutton1 is selected.
end

function p_m_12_SelectionChangeFcn(hObject, eventdata, handles)
global process_m_12
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'm_12_ad'
        % Code for when radiobutton1 is selected.
       process_m_12=1;
    case 'm_12_wo'
        % Code for when radiobutton2 is selected.
        process_m_12=2;
    case 'm_12_wowsd'
        process_m_12=3;
        % Code for when togglebutton1 is selected.
end

function evaluation12_Callback(hObject, eventdata, handles)
% global process_m_1 process_m_2 process_m_3 process_m_4 process_m_5
process_m_6 process_m_7 process_m_8 process_m_9 process_m_10 process_m_11
process_m_12 C_datos_m_1 MDM_m_1 fid_m_1 C_datos_m_2 MDM_m_2 fid_m_2
C_datos_m_3 MDM_m_3 fid_m_3 C_datos_m_4 MDM_m_4 fid_m_4 C_datos_m_5
MDM_m_5 fid_m_5 C_datos_m_6 MDM_m_6 fid_m_6 C_datos_m_7 MDM_m_7 fid_m_7
C_datos_m_8 MDM_m_8 fid_m_8 C_datos_m_9 MDM_m_9 fid_m_9 C_datos_m_10
MDM_m_10 fid_m_10 C_datos_m_11 MDM_m_11 fid_m_11 C_datos_m_12 MDM_m_12
fid_m_12
%
% global car1_m_1 car1_m_2 car1_m_3 car1_m_4 car1_m_5 car1_m_6 car1_m_7
car1_m_8 car1_m_9 car1_m_10 car1_m_11 car1_m_12
% car1_m_1
% car1_m_2
% car1_m_3
% car1_m_4
% car1_m_5
% car1_m_6
% car1_m_7
% car1_m_8
% car1_m_9
% car1_m_10
% car1_m_11
% car1_m_12
compare_12_2_try2
```


compare_12_2_try2
Moduls
Modul 1
Modul 2
Modul 3
Modul 4
Modul 5
Modul 6
Modul 7
Modul 8
Modul 9
Modul 10
Modul 11
Modul 12
Cell area [cm²]
SOLARWORLD
INNOVATIONS
Show tables
Evaluate
Today
29-Jul-2010

```
function varargout = compare_12_2_try2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compare_12_2_try2_OpeningFcn, ...
                   'gui_OutputFcn',  @compare_12_2_try2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function compare_12_2_try2_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
global ext_m_1 s_m_1 ext_m_2 s_m_2 ext_m_3 s_m_3 ext_m_4 s_m_4 ext_m_5
s_m_5 ext_m_6 s_m_6 ext_m_7 s_m_7 ext_m_8 s_m_8 ext_m_9 s_m_9 ext_m_10
s_m_10 ext_m_11 s_m_11 ext_m_12 s_m_12
global car1_m_1 car1_m_2 car1_m_3 car1_m_4 car1_m_5 car1_m_6 car1_m_7
car1_m_8 car1_m_9 car1_m_10 car1_m_11 car1_m_12
global process_m_1 process_m_2 process_m_3 process_m_4 process_m_5
process_m_6 process_m_7 process_m_8 process_m_9 process_m_10 process_m_11
process_m_12
global MDM_m_1 MDM_m_2 MDM_m_3 MDM_m_4 MDM_m_5 MDM_m_6 MDM_m_7 MDM_m_8
MDM_m_9 MDM_m_10 MDM_m_11 MDM_m_12
global titchMDM_m_1 titchMDM_m_2 titchMDM_m_3 titchMDM_m_4 titchMDM_m_5
titchMDM_m_6 titchMDM_m_7 titchMDM_m_8 titchMDM_m_9 titchMDM_m_10
titchMDM_m_11 titchMDM_m_12
global car_m_1 car_m_2 car_m_3 car_m_4 car_m_5 car_m_6 car_m_7 car_m_8
car_m_9 car_m_10 car_m_11 car_m_12
global cont horai_m horaf_m Irr_m

cont=0;
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
cont=0;
hoy=date;
set(handles.today, 'String', hoy)
car_m_1=0;
car_m_2=0;
car_m_3=0;
car_m_4=0;
car_m_5=0;
car_m_6=0;
car_m_7=0;
car_m_8=0;
car_m_9=0;
car_m_10=0;
```

```
car_m_11=0;
car_m_12=0;
car1_m=[];
ext_m=[];
process_m=[];
s_m=[];
car1_m=[car1_m_1 car1_m_2 car1_m_3 car1_m_4 car1_m_5 car1_m_6 car1_m_7
car1_m_8 car1_m_9 car1_m_10 car1_m_11 car1_m_12];
process_m=[process_m_1 process_m_2 process_m_3 process_m_4 process_m_5
process_m_6 process_m_7 process_m_8 process_m_9 process_m_10 process_m_11
process_m_12];
ext_m = [ext_m_1; ext_m_2; ext_m_3; ext_m_4; ext_m_5; ext_m_6; ext_m_7;
ext_m_8; ext_m_9; ext_m_10; ext_m_11; ext_m_12 ];
s_m = {s_m_1; s_m_2; s_m_3; s_m_4; s_m_5; s_m_6; s_m_7; s_m_8; s_m_9;
s_m_10; s_m_11; s_m_12};
MDM_m_1 = [];
MDM_m_2 = [];

MDM_m_3 = [];
MDM_m_4 = [];

MDM_m_5 = [];
MDM_m_6 = [];

MDM_m_7 = [];
MDM_m_8 = [];


MDM_m_9 = [];
MDM_m_10= [];

MDM_m_11 = [];
MDM_m_12= [];

titchMDM_m_1= [];
titchMDM_m_2= [];

titchMDM_m_3= [];
titchMDM_m_4= [];


titchMDM_m_5= [];
titchMDM_m_6= [];

titchMDM_m_7= [];
titchMDM_m_8= [];

titchMDM_m_9= [];
titchMDM_m_10= [];

titchMDM_m_11= [];
titchMDM_m_12= [];
Irr_m=[];
horai_m=[];
```

```
horaf_m=[];
for i=1:12
    MDM_new=[];
    lxls=0;
    titMDM=[];
    t=[];
    tcdate=[];
    horan=[];
    ltxt=0;
    lltxt=0;
    tchour_new=[];
    tchour_newc=[];
    tchour_newchar=[];
    tchour_realc=[];
    C_datos=[];
    if car1_m(i)==1

        if ext_m(i,:)=='XLS'
            %fprintf('entra xls')
            [MDM_new,titMDM]=xlsread(s_m{i});
            MDM_new = ordenxls(MDM_new);  %%%%%%%%%%%%%%%%%%%
            lxls=length(MDM_new(:,1));
            t=zeros(lxls,19);
            for i3=3:lxls+2
                t(i3-2,:)=titMDM{i3,2};
            end
            tcdate=char(t(:,1:10));
            tchour_new=char(t(:,12:end));
            horai_m(i,:)=tchour_new(2,:);
            horaf_m(i,:)=tchour_new(lxls,:);


            horan=datenum(tchour_new,'HH:MM:SS');
            valor=zeros(lxls-1);
            for i2=2:lxls
                valor(i2-1)=(horan(i2)-horan(i2-1))*MDM_new(i2-1,1);
            end
            Irr_m(i)=sum(valor(:,1))*24;
            %         elseif ext_m(i,:)=='asc'
            %             %fprintf('entra txt')
            %             fid = fopen(s_m{i});
            %             C_text=textscan(fid,'%*[^\n]',2);
            %             C_datos=textscan(fid,'%s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s ');
            %             ltxt=length(C_datos{1,1});
            %             C_datosp=zeros(ltxt,12);
            %             for i4=1:ltxt
            %                 for j=4:15
            %                     MDM_new(i4,j-
3)=str2num(strrep(C_datos{1,j}{i4,1},',','.'));
            %                 end
            %             end
            %             MDM_new = ordentxt(MDM_new);
%%%%%%%%%%%%%%%%%%%
            %             lltxt=length(C_datos{3});
            %             for i5=1:lltxt
```

```
            %                 tchour_newc(i5,:)=C_datos{1,2}{i5,1};
            %             end
            %             %         C_datos{1,3}{1,
            %             %tchour_new=char(tchour_newc);
            %             tchour_newchar=char(tchour_newc);
            %             for i8=1:lltxt
            %                 if C_datos{1,3}{i8,1}=='PM'
            %                     hour1=str2num(tchour_newchar(i8,1));
            %                     realhour=hour1+12;
            %                     realhourchar=num2str(realhour);
            %                     tchour_realc(i8,1)=' ';
            %                     tchour_realc(i8,2)=realhourchar(1);
            %                     tchour_realc(i8,3)=realhourchar(2);
            %                     for i6=1:6
            %
tchour_realc(i8,3+i6)=tchour_newchar(i8,1+i6);
            %                     end
            %                 else
            %                     tchour_realc(i8,1)=' ';
            %                     tchour_realc(i8,2)='0';
            %                     for i7=1:7
            %
tchour_realc(i8,2+i7)=tchour_newchar(i8,i7);
            %                     end
            %                 end
            %             end
            %             tchour_new=char(tchour_realc);
            %             horai_m(i,:)=tchour_new(2,:);
            %             horaf_m(i,:)=tchour_new(lltxt,:);
            %
            %             horan=datenum(tchour_new,'HH:MM:SS');
            %             valor=zeros(lltxt-1);
            %             for i5=2:lltxt
            %                 valor(i5-1)=(horan(i5)-horan(i5-
1))*MDM_new(i5-1,1);
            %             end
            %             Irr_m(i)=sum(valor(:,1))*24;
            %             fclose(fid);
        elseif ext_m(i,:)=='asc'
            fid = fopen(s_m{i});
            C_text=textscan(fid,'%*[^\n]',2);
            C_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s %s %s ');
            ltxt=length(C_datos{1,1});
            C_datosp=zeros(ltxt,12);
            for i4=1:ltxt
                for j=4:11
                    MDM_new(i4,j-
3)=str2num(strrep(C_datos{1,j}{i4,1},',','.'));
                end
            end
            TKisc=0.034;
            TKvoc=-0.34;
            TKpmpp=-0.48;
            MDM_new = ordentxt(MDM_new,TKpmpp,TKisc,TKvoc);
            lltxt=length(C_datos{3});
            tchour_newc=[];
```

```
            tchour_new=[];
            for i5=1:lltxt
                tchour_newc(i5,:)=C_datos{1,3}{i5,1};
            end
            %         C_datos{1,3}{1,
            %tchour_new=char(tchour_newc);

            tchour_new=char(tchour_newc);
            horai_m(i,:)=tchour_new(2,:);
            horaf_m(i,:)=tchour_new(lltxt,:);
            horan=datenum(tchour_new,'HH:MM:SS');
            valor=zeros(lltxt-1);
            for i5=2:lltxt
                valor(i5-1)=(horan(i5)-horan(i5-1))*MDM_new(i5-1,1);
            end
            Irr_m(i)=sum(valor(:,1))*24;
            fclose(fid);

        end
        if process_m(i)==2
            [MDM_new, titchMDM_m] = outlierRemoval(MDM_new, tchour_new);
        elseif process_m(i)==3
            [MDM_new, titchMDM_m] = outlierRemoval_sosds(MDM_new,
tchour_new);
        elseif process_m(i)==1
            titchMDM_m=tchour_new;
        end
        if i==1
            MDM_m_1=MDM_new;
            titchMDM_m_1=titchMDM_m;
            l_m_1=length(MDM_m_1);
            if l_m_1==0
                errordlg('You must again choose, because the file of
Modul 1 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end

        elseif i==2
            MDM_m_2=MDM_new;
            titchMDM_m_2=titchMDM_m;
            l_m_2=length(MDM_m_2);
            if l_m_2==0
                errordlg('You must again choose, because the file of
Modul 2  have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==3
            MDM_m_3=MDM_new;
            titchMDM_m_3=titchMDM_m;
            l_m_3=length(MDM_m_3);
            if l_m_3==0
                errordlg('You must again choose, because the file of
Modul 3 have not enough data','Bad Input','modal')
                uicontrol(hObject)
```

```
                return
            end
        elseif i==4
            MDM_m_4=MDM_new;
            titchMDM_m_4=titchMDM_m;
            l_m_4=length(MDM_m_4);
            if l_m_4==0
                errordlg('You must again choose, because the file of
Modul 4 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==5
            MDM_m_5=MDM_new;
            titchMDM_m_5=titchMDM_m;
            l_m_5=length(MDM_m_5);
            if l_m_5==0
                errordlg('You must again choose, because the file of
Modul 5 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==6
            MDM_m_6=MDM_new;
            titchMDM_m_6=titchMDM_m;
            l_m_6=length(MDM_m_6);
            if l_m_6==0
                errordlg('You must again choose, because the file of
Modul 6 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==7
            MDM_m_7=MDM_new;
            titchMDM_m_7=titchMDM_m;
            l_m_7=length(MDM_m_7);
            if l_m_7==0
                errordlg('You must again choose, because the file of
Modul 7 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==8
            MDM_m_8=MDM_new;
            titchMDM_m_8=titchMDM_m;
            l_m_8=length(MDM_m_8);
            if l_m_8==0
                errordlg('You must again choose, because the file of
Modul 8 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==9
            MDM_m_9=MDM_new;
            titchMDM_m_9=titchMDM_m;
            l_m_9=length(MDM_m_9);
            if l_m_9==0
```

```
                errordlg('You must again choose, because the file of
Modul 9 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==10
            MDM_m_10=MDM_new;
            titchMDM_m_10=titchMDM_m;
            l_m_10=length(MDM_m_10);
            if l_m_10==0
                errordlg('You must again choose, because the file of
Modul 10 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        elseif i==11
            MDM_m_11=MDM_new;
            titchMDM_m_11=titchMDM_m;
            l_m_11=length(MDM_m_11);
            if l_m_11==0
                errordlg('You must again choose, because the file of
Modul 11 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        else
            MDM_m_12=MDM_new;
            titchMDM_m_12=titchMDM_m;
            l_m_12=length(MDM_m_12);
            if l_m_12==0
                errordlg('You must again choose, because the file of
Modul 12 have not enough data','Bad Input','modal')
                uicontrol(hObject)
                return
            end
        end
    end
end
% MDM_m_1
% MDM_m_2
%
% MDM_m_3
% MDM_m_4
%
% MDM_m_5
% MDM_m_6
%
% MDM_m_7
% MDM_m_8
%
%
% MDM_m_9
% MDM_m_10
%
% MDM_m_11
% MDM_m_12
%
```

```
% titchMDM_m_1
% titchMDM_m_2
%
% titchMDM_m_3
% titchMDM_m_4
%
%
% titchMDM_m_5
% titchMDM_m_6
%
% titchMDM_m_7
% titchMDM_m_8
%
% titchMDM_m_9
% titchMDM_m_10
%
% titchMDM_m_11
% titchMDM_m_12
```

```
% % if car1_m_1==1
% % MDM_new=[];
% % if ext_m_1=='XLS'
% %     [MDM_new,titMDM]=xlsread(s_m_1);
% %     MDM_new = ordenxls(MDM_new);
% %     lxls=length(MDM_new(:,1));
% %     t=zeros(lxls,19);
% %     for i3=3:lxls+2
% %         t(i3-2,:)=titMDM{i3,2};
% %     end
% %     tcdate=char(t(:,1:10));
% %     tchour_new=char(t(:,11:end));
% %     horan=datenum(tchour_new,'HH:MM:SS');
% %     valor=zeros(lxls-1);
% %     for i2=2:lxls
% %         valor(i2-1)=(horan(i2)-horan(i2-1))*MDM_new(i2-1,1);
% %     end
% %     Irr_m_1=sum(valor(:,1))*24;
% % elseif ext_m_1=='txt'
% %     fid = fopen(s_m_1);
% %     C_text=textscan(fid,'%*[^\n]',2);
% %     C_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s');
% %     ltxt=length(C_datos{1,1});
% %     C_datosp=zeros(ltxt,12);
% %     for i4=1:ltxt
% %         for j=4:15
% %             MDM_new(i4,j-
3)=str2num(strrep(C_datos{1,j}{i4,1},',','.'));
% %         end
% %     end
% %     MDM_new = ordentxt(MDM_new);
% %     lltxt=length(C_datos{3});
% %     for i5=1:lltxt
```

```
% %         tchour_newc(i5,:)=C_datos{1,2}{i5,1};
% %     end
% %     %         C_datos{1,3}{1,
% %     %tchour_new=char(tchour_newc);
% %     tchour_newchar=char(tchour_newc);
% %     for i8=1:lltxt
% %         if C_datos{1,3}{i8,1}=='PM'
% %             hour1=str2num(tchour_newchar(i8,1));
% %             realhour=hour1+12;
% %             realhourchar=num2str(realhour);
% %             tchour_realc(i8,1)=' ';
% %             tchour_realc(i8,2)=realhourchar(1);
% %             tchour_realc(i8,3)=realhourchar(2);
% %             for i6=1:6
% %                 tchour_realc(i8,3+i6)=tchour_newchar(i8,1+i6);
% %             end
% %         else
% %             tchour_realc(i8,1)=' ';
% %             tchour_realc(i8,2)='0';
% %             for i7=1:7
% %                 tchour_realc(i8,2+i7)=tchour_newchar(i8,i7);
% %             end
% %         end
% %     end
% %     tchour_new=char(tchour_realc);
% %     horan=datenum(tchour_new,'HH:MM:SS');
% %     valor=zeros(lltxt-1);
% %     for i5=2:lltxt
% %         valor(i5-1)=(horan(i5)-horan(i5-1))*MDM_new(i5-1,1);
% %     end
% %     Irr_m_1=sum(valor(:,1))*24;
% %     fclose(fid);
% % end
% % MDM_m_1=MDM_new
% % tchour_m_1=tchour_new
% % end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz modulo 1

% %     if process_m_1==2
% %         [MDM_m_1, titchMDM_m_1] = outlierRemoval(MDM_m_1,
tchour_m_1);
% %     elseif process_m_1==3
% %         [MDM_m_1, titchMDM_m_1] = outlierRemoval_sosds(MDM_m_1,
tchour_m_1);
% %     else
% %         titchMDM_m_1=tchour_m_1;
% %     end
% % end
%%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 2
% % % if car1_m_2==1
% % %     MDM_m_2 = ordentxt(MDM_m_2)
% % %     MDMi_m_2=MDM_m_2;
% % %     l=length(C_datos_m_2{3});
% % %     for i=1:l
% % %         tchour_m_2(i,:)=C_datos_m_2{1,2}{i,1};
% % %     end
```

```
% % %     horai_m_2=tchour_m_2(2,:);
% % %     horaf_m_2=tchour_m_2(l,:);
% % %     horan_m_2=datenum(tchour_m_2,'HH:MM:SS');
% % %     valor_m_2=zeros(l-1);
% % %     for i=2:l
% % %         valor_m_2(i-1)=(horan_m_2(i)-horan_m_2(i-1))*MDM_m_2(i-
1,1);
% % %     end
% % %     Irr_m_2=sum(valor_m_2(:,1))*24;
% % %     fechaMDM_m_2=C_datos_m_2{1,1}{1,1};
% % %     tchour_m_2
% % %
% % %     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz modulo 2
% % %
% % %     if process_m_2==2
% % %         [MDM_m_2, titchMDM_m_2] = outlierRemoval(MDM_m_2,
tchour_m_2);
% % %     elseif process_m_2==3
% % %         [MDM_m_2, titchMDM_m_2] = outlierRemoval_sosds(MDM_m_2,
tchour_m_2);
% % %     else
% % %         titchMDM_m_2=tchour_m_2;
% % %     end
% % % end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 3
% % %     if car1_m_3==1
% % %         MDM_m_3 = ordentxt(MDM_m_3);
% % %         MDMi_m_3=MDM_m_3;
% % %         l=length(C_datos_m_3{3});
% % %         for i=1:l
% % %             tchour_m_3(i,:)=C_datos_m_3{1,2}{i,1};
% % %         end
% % %         horai_m_3=tchour_m_3(2,:);
% % %         horaf_m_3=tchour_m_3(l,:);
% % %         horan_m_3=datenum(tchour_m_3,'HH:MM:SS');
% % %         valor_m_3=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_3(i-1)=(horan_m_3(i)-horan_m_3(i-1))*MDM_m_3(i-
1,1);
% % %         end
% % %         Irr_m_3=sum(valor_m_3(:,1))*24;
% % %         fechaMDM_m_3=C_datos_m_3{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 3
% % %
% % %         if process_m_3==2
% % %             [MDM_m_3, titchMDM_m_3] = outlierRemoval(MDM_m_3,
tchour_m_3);
% % %         elseif process_m_3==3
% % %             [MDM_m_3, titchMDM_m_3] = outlierRemoval_sosds(MDM_m_3,
tchour_m_3);
% % %         else
% % %             titchMDM_m_3=tchour_m_3;
% % %         end
% % %     end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 4
```

```
% % %     if car1_m_4==1
% % %         MDM_m_4 = ordentxt(MDM_m_4);
% % %         MDMi_m_4=MDM_m_4;
% % %         l=length(C_datos_m_4{3});
% % %         for i=1:l
% % %             tchour_m_4(i,:)=C_datos_m_4{1,2}{i,1};
% % %         end
% % %         horai_m_4=tchour_m_4(2,:);
% % %         horaf_m_4=tchour_m_4(l,:);
% % %         horan_m_4=datenum(tchour_m_4,'HH:MM:SS');
% % %         valor_m_4=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_4(i-1)=(horan_m_4(i)-horan_m_4(i-1))*MDM_m_4(i-
1,1);
% % %         end
% % %         Irr_m_4=sum(valor_m_4(:,1))*24;
% % %         fechaMDM_m_4=C_datos_m_4{1,1}{1,1};
% % %
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 4
% % %
% % %         if process_m_4==2
% % %             [MDM_m_4, titchMDM_m_4] = outlierRemoval(MDM_m_4,
tchour_m_4);
% % %         elseif process_m_4==3
% % %             [MDM_m_4, titchMDM_m_4] = outlierRemoval_sosds(MDM_m_4,
tchour_m_4);
% % %         else
% % %             titchMDM_m_4=tchour_m_4;
% % %         end
% % %     end
% % %     %%%%%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 5
% % %     if car1_m_5==1
% % %         MDM_m_5 = ordentxt(MDM_m_5);
% % %         MDMi_m_5=MDM_m_5;
% % %         l=length(C_datos_m_5{3});
% % %         for i=1:l
% % %             tchour_m_5(i,:)=C_datos_m_5{1,2}{i,1};
% % %         end
% % %         horai_m_5=tchour_m_5(2,:);
% % %         horaf_m_5=tchour_m_5(l,:);
% % %         horan_m_5=datenum(tchour_m_5,'HH:MM:SS');
% % %         valor_m_5=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_5(i-1)=(horan_m_5(i)-horan_m_5(i-1))*MDM_m_5(i-
1,1);
% % %         end
% % %         Irr_m_5=sum(valor_m_5(:,1))*24;
% % %         fechaMDM_m_5=C_datos_m_5{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 5
% % %
% % %         if process_m_5==2
% % %             [MDM_m_5, titchMDM_m_5] = outlierRemoval(MDM_m_5,
tchour_m_5);
```

```
% % %         elseif process_m_5==3
% % %             [MDM_m_5, titchMDM_m_5] = outlierRemoval_sosds(MDM_m_5,
tchour_m_5);
% % %         else
% % %             titchMDM_m_5=tchour_m_5;
% % %         end
% % %     end
% % % %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 6
% % % if car1_m_6==1
% % %     MDM_m_6 = ordentxt(MDM_m_6);
% % %     MDMi_m_6=MDM_m_6;
% % %     l=length(C_datos_m_6{3});
% % %     for i=1:l
% % %         tchour_m_6(i,:)=C_datos_m_6{1,2}{i,1};
% % %     end
% % %     horai_m_6=tchour_m_6(2,:);
% % %     horaf_m_6=tchour_m_6(l,:);
% % %     horan_m_6=datenum(tchour_m_6,'HH:MM:SS');
% % %     valor_m_6=zeros(l-1);
% % %     for i=2:l
% % %         valor_m_6(i-1)=(horan_m_6(i)-horan_m_6(i-1))*MDM_m_6(i-
1,1);
% % %     end
% % %     Irr_m_6=sum(valor_m_6(:,1))*24;
% % %     fechaMDM_m_6=C_datos_m_6{1,1}{1,1};
% % %
% % %
% % %     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz modulo 6
% % %
% % %     if process_m_6==2
% % %         [MDM_m_6, titchMDM_m_6] = outlierRemoval(MDM_m_6,
tchour_m_6);
% % %     elseif process_m_6==3
% % %         [MDM_m_6, titchMDM_m_6] = outlierRemoval_sosds(MDM_m_6,
tchour_m_6);
% % %     else
% % %         titchMDM_m_6=tchour_m_6;
% % %     end
% % % end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 7
% % %     if car1_m_7==1
% % %         MDM_m_7 = ordentxt(MDM_m_7);
% % %         MDMi_m_7=MDM_m_7;
% % %         l=length(C_datos_m_7{3});
% % %         for i=1:l
% % %             tchour_m_7(i,:)=C_datos_m_7{1,2}{i,1};
% % %         end
% % %         horai_m_7=tchour_m_7(2,:);
% % %         horaf_m_7=tchour_m_7(l,:);
% % %         horan_m_7=datenum(tchour_m_7,'HH:MM:SS');
% % %         valor_m_7=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_7(i-1)=(horan_m_7(i)-horan_m_7(i-1))*MDM_m_7(i-
1,1);
% % %         end
% % %         Irr_m_7=sum(valor_m_7(:,1))*24;
% % %         fechaMDM_m_7=C_datos_m_7{1,1}{1,1};
```

```
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 7
% % %
% % %         if process_m_7==2
% % %             [MDM_m_7, titchMDM_m_7] = outlierRemoval(MDM_m_7,
tchour_m_7);
% % %         elseif process_m_7==3
% % %             [MDM_m_7, titchMDM_m_7] = outlierRemoval_sosds(MDM_m_7,
tchour_m_7);
% % %         else
% % %             titchMDM_m_7=tchour_m_7;
% % %         end
% % %     end
% % %
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 8
% % %     if car1_m_8==1
% % %         MDM_m_8 = ordentxt(MDM_m_8);
% % %         MDMi_m_8=MDM_m_8;
% % %         l=length(C_datos_m_8{3});
% % %         for i=1:l
% % %             tchour_m_8(i,:)=C_datos_m_8{1,2}{i,1};
% % %         end
% % %         horai_m_8=tchour_m_8(2,:);
% % %         horaf_m_8=tchour_m_8(l,:);
% % %         horan_m_8=datenum(tchour_m_8,'HH:MM:SS');
% % %         valor_m_8=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_8(i-1)=(horan_m_8(i)-horan_m_8(i-1))*MDM_m_8(i-
1,1);
% % %         end
% % %         Irr_m_8=sum(valor_m_8(:,1))*24;
% % %         fechaMDM_m_8=C_datos_m_8{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 8
% % %
% % %         if process_m_8==2
% % %             [MDM_m_8, titchMDM_m_8] = outlierRemoval(MDM_m_8,
tchour_m_8);
% % %         elseif process_m_8==3
% % %             [MDM_m_8, titchMDM_m_8] = outlierRemoval_sosds(MDM_m_8,
tchour_m_8);
% % %         else
% % %             titchMDM_m_8=tchour_m_8;
% % %         end
% % %     end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 9
% % %     if car1_m_9==1
% % %         MDM_m_9 = ordentxt(MDM_m_9);
% % %         MDMi_m_9=MDM_m_9;
% % %         l=length(C_datos_m_9{3});
% % %         for i=1:l
% % %             tchour_m_9(i,:)=C_datos_m_9{1,2}{i,1};
% % %         end
% % %         horai_m_9=tchour_m_9(2,:);
% % %         horaf_m_9=tchour_m_9(l,:);
```

```
% % %         horan_m_9=datenum(tchour_m_9,'HH:MM:SS');
% % %         valor_m_9=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_9(i-1)=(horan_m_9(i)-horan_m_9(i-1))*MDM_m_9(i-
1,1);
% % %         end
% % %         Irr_m_9=sum(valor_m_9(:,1))*24;
% % %         fechaMDM_m_9=C_datos_m_9{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 9
% % %
% % %         if process_m_9==2
% % %             [MDM_m_9, titchMDM_m_9] = outlierRemoval(MDM_m_9,
tchour_m_9);
% % %         elseif process_m_9==3
% % %             [MDM_m_9, titchMDM_m_9] = outlierRemoval_sosds(MDM_m_9,
tchour_m_9);
% % %         else
% % %             titchMDM_m_9=tchour_m_9;
% % %         end
% % %     end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 1
% % %     if car1_m_10==1
% % %         MDM_m_10 = ordentxt(MDM_m_10);
% % %         MDMi_m_10=MDM_m_10;
% % %         l=length(C_datos_m_10{3});
% % %         for i=1:l
% % %             tchour_m_10(i,:)=C_datos_m_10{1,2}{i,1};
% % %         end
% % %         horai_m_10=tchour_m_10(2,:);
% % %         horaf_m_10=tchour_m_10(l,:);
% % %         horan_m_10=datenum(tchour_m_10,'HH:MM:SS');
% % %         valor_m_10=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_10(i-1)=(horan_m_10(i)-horan_m_10(i-
1))*MDM_m_10(i-1,1);
% % %         end
% % %         Irr_m_10=sum(valor_m_10(:,1))*24;
% % %         fechaMDM_m_10=C_datos_m_10{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 10
% % %
% % %         if process_m_10==2
% % %             [MDM_m_10, titchMDM_m_10] = outlierRemoval(MDM_m_10,
tchour_m_10);
% % %         elseif process_m_10==3
% % %             [MDM_m_10, titchMDM_m_10] =
outlierRemoval_sosds(MDM_m_10, tchour_m_10);
% % %         else
% % %             titchMDM_m_10=tchour_m_10;
% % %         end
% % %     end
% % %     %%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 11
% % %
% % %     if car1_m_11==1
```

```
% % %         MDM_m_11 = ordentxt(MDM_m_11);
% % %         MDMi_m_11=MDM_m_11;
% % %         l=length(C_datos_m_11{3});
% % %         for i=1:l
% % %             tchour_m_11(i,:)=C_datos_m_11{1,2}{i,1};
% % %         end
% % %         horai_m_11=tchour_m_11(2,:);
% % %         horaf_m_11=tchour_m_11(l,:);
% % %         horan_m_11=datenum(tchour_m_11,'HH:MM:SS');
% % %         valor_m_11=zeros(l-1);
% % %         for i=2:l
% % %             valor_m_11(i-1)=(horan_m_11(i)-horan_m_11(i-
1))*MDM_m_11(i-1,1);
% % %         end
% % %         Irr_m_11=sum(valor_m_11(:,1))*24;
% % %         fechaMDM_m_11=C_datos_m_11{1,1}{1,1};
% % %
% % %         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz
modulo 11
% % %
% % %         if process_m_11==2
% % %             [MDM_m_11, titchMDM_m_11] = outlierRemoval(MDM_m_11,
tchour_m_11);
% % %         elseif process_m_11==3
% % %             [MDM_m_11, titchMDM_m_11] =
outlierRemoval_sosds(MDM_m_11, tchour_m_11);
% % %         else
% % %             titchMDM_m_11=tchour_m_11;
% % %         end
% % %     end
% % % %%%%%%%%%%%%%%%%%%%%%%ordenar matriz del modulo 12
% % % if car1_m_12==1
% % %     MDM_m_12 = ordentxt(MDM_m_12);
% % %     MDMi_m_12=MDM_m_12;
% % %     l=length(C_datos_m_12{3});
% % %     for i=1:l
% % %         tchour_m_12(i,:)=C_datos_m_12{1,2}{i,1};
% % %     end
% % %     horai_m_12=tchour_m_12(2,:);
% % %     horaf_m_12=tchour_m_12(l,:);
% % %     horan_m_12=datenum(tchour_m_12,'HH:MM:SS');
% % %     valor_m_12=zeros(l-1);
% % %     for i=2:l
% % %         valor_m_12(i-1)=(horan_m_12(i)-horan_m_12(i-1))*MDM_m_12(i-
1,1);
% % %     end
% % %     Irr_m_12=sum(valor_m_12(:,1))*24;
% % %     fechaMDM_m_12=C_datos_m_12{1,1}{1,1};
% % %
% % %
% % %     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%procesar matriz modulo
12
% % %
% % %     if process_m_12==2
% % %         [MDM_m_12, titchMDM_m_12] = outlierRemoval(MDM_m_12,
tchour_m_12);
% % %     elseif process_m_12==3
```

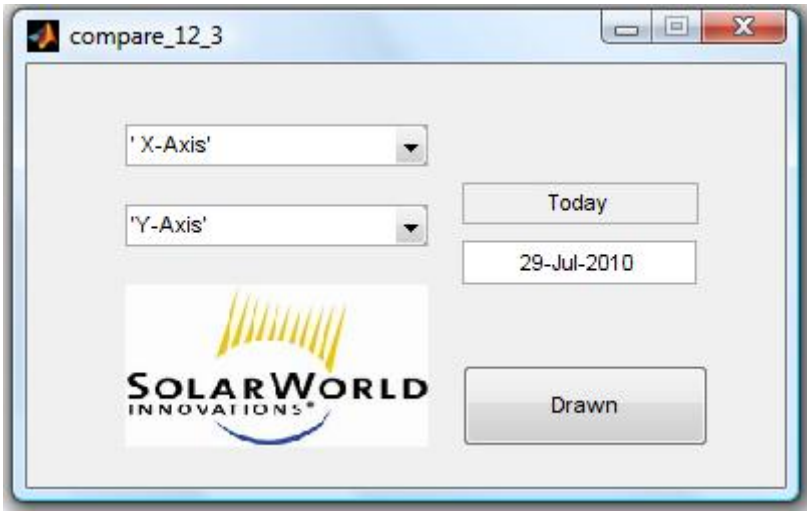
```
% % %         [MDM_m_12, titchMDM_m_12] = outlierRemoval_sosds(MDM_m_12,
tchour_m_12);
% % %     else
% % %         titchMDM_m_12=tchour_m_12;
% % %     end
% % % end

function varargout = compare_12_2_try2_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function today_Callback(hObject, eventdata, handles)

function today_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function checkbox1_Callback(hObject, eventdata, handles)
global car_m_1 cont
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_1=1;
    cont=cont+1;
else
    car_m_1=0;
    % Checkbox is not checked-take approriate action
end
% car_m_1

function checkbox2_Callback(hObject, eventdata, handles)
global car_m_2 cont
% car_m_2=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_2=1;
    cont=cont+1;
else
    car_m_2=0;

    % Checkbox is not checked-take approriate action
end

% car_m_2
function checkbox3_Callback(hObject, eventdata, handles)
global car_m_3 cont
% car_m_3=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_3=1;
    cont=cont+1;
else
    car_m_3=0;
    % Checkbox is not checked-take approriate action
```

```
end
% car_m_3

function checkbox4_Callback(hObject, eventdata, handles)
global car_m_4 cont
% car_m_4=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_4=1;
    cont=cont+1;
else
    car_m_4=0;
    % Checkbox is not checked-take approriate action
end
% car_m_4

function checkbox5_Callback(hObject, eventdata, handles)
global car_m_5 cont
% car_m_5=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_5=1;
    cont=cont+1;
else
    car_m_5=0;
    % Checkbox is not checked-take approriate action
end
% car_m_5
function checkbox6_Callback(hObject, eventdata, handles)
global car_m_6 cont
% car_m_6=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_6=1;
    cont=cont+1;
else
    car_m_6=0;
    % Checkbox is not checked-take approriate action
end
% car_m_6

function checkbox7_Callback(hObject, eventdata, handles)
global car_m_7 cont
% car_m_7=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_7=1;
    cont=cont+1;
else
    car_m_7=0;
    % Checkbox is not checked-take approriate action
end
```

```
% car_m_7

function checkbox8_Callback(hObject, eventdata, handles)
global car_m_12 cont

% car_m_12=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_12=1;
    cont=cont+1;
else
    car_m_12=0;
    % Checkbox is not checked-take approriate action
end
% car_m_12

function checkbox9_Callback(hObject, eventdata, handles)
global car_m_8 cont
% car_m_8=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_8=1;
    cont=cont+1;
else
    car_m_8=0;
    % Checkbox is not checked-take approriate action
end
% car_m_8

function checkbox10_Callback(hObject, eventdata, handles)
global car_m_9 cont
% car_m_9=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_9=1;
    cont=cont+1;
else
    car_m_9=0;
    % Checkbox is not checked-take approriate action
end
% car_m_9

function checkbox11_Callback(hObject, eventdata, handles)
global car_m_10 cont
% car_m_10=0;

if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_10=1;
    cont=cont+1;
else
    car_m_10=0;
    % Checkbox is not checked-take approriate action
end
% car_m_10
```

```
function checkbox12_Callback(hObject, eventdata, handles)
global car_m_11 cont
% car_m_11=0;
if (get(hObject,'Value') == get(hObject,'Max'))
    % Checkbox is checked-take approriate action
    car_m_11=1;
    cont=cont+1;
else
    car_m_11=0;
    % Checkbox is not checked-take approriate action
end
% car_m_11

function area_Callback(hObject, eventdata, handles)
global c_a
b=get(handles.area,'String');
c=strrep(b, ',','.');
c_a=str2num(c);

function area_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function buttonshowt_Callback(hObject, eventdata, handles)
global  car_m_1 car_m_2 car_m_3 car_m_4 car_m_5 car_m_6 car_m_7 car_m_8
car_m_9 car_m_10 car_m_11 car_m_12 cont
% cont=0;
% car_m_1
% car_m_2
% car_m_3
% car_m_4
% car_m_5
% car_m_6
% car_m_7
% car_m_8
% car_m_9
% car_m_10
% car_m_11
% car_m_12
if car_m_1==1
    Modul_1
%     cont=cont+1;
end
if car_m_2==1
    Modul_2
%     cont=cont+1;
end
if car_m_3==1
    Modul_3
%     cont=cont+1;
end
if car_m_4==1
    Modul_4
%     cont=cont+1;
```

```
end
if car_m_5==1
    Modul_5
%     cont=cont+1;
end
if car_m_6==1
    Modul_6
%     cont=cont+1;
end
if car_m_7==1
    Modul_7
%     cont=cont+1;
end
if car_m_8==1
    Modul_8
%     cont=cont+1;
end
if car_m_9==1
    Modul_9
%     cont=cont+1;
end
if car_m_10==1
    Modul_10
%     cont=cont+1;
end
if car_m_11==1
    Modul_11
%     cont=cont+1;
end
if car_m_12==1
    Modul_12
%     cont=cont+1;
end

function evalua_comp_Callback(hObject, eventdata, handles)
compare_12_3
```

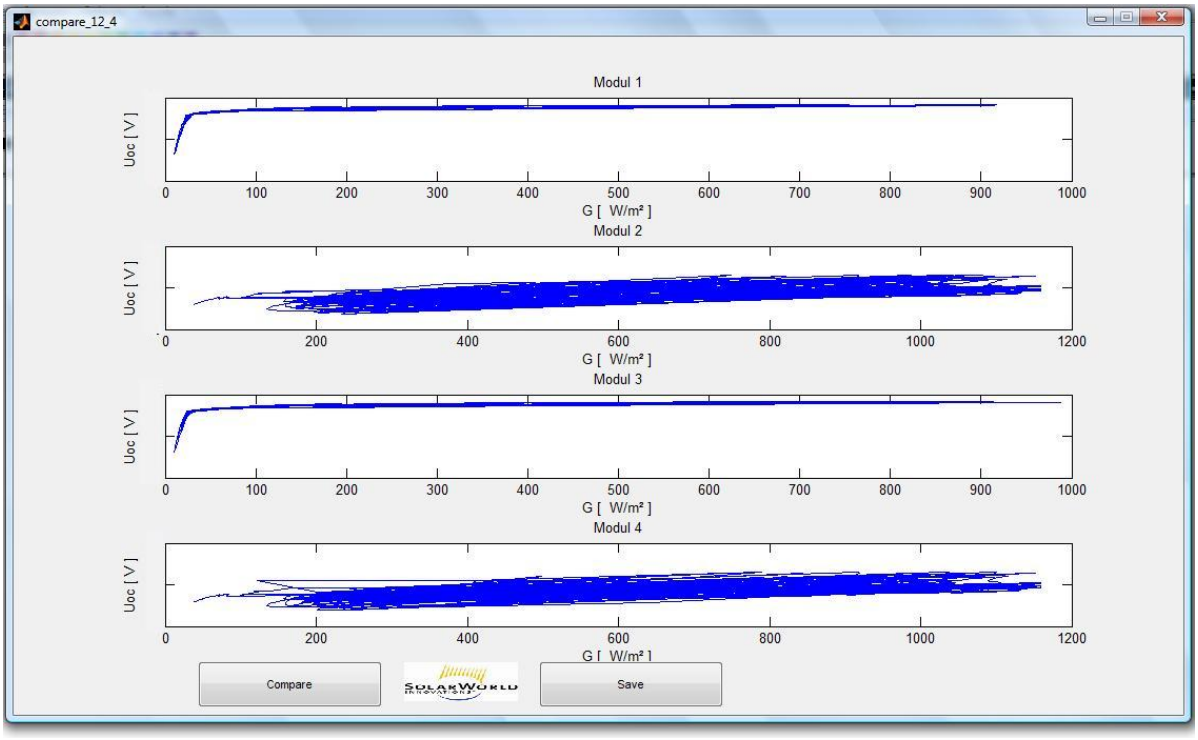

compare_12_3
' X-Axis'
'Y-Axis'
Today
29-Jul-2010
SOLARWORLD
INNOVATIONS
Drawn

```
function varargout = compare_12_3(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compare_12_3_OpeningFcn, ...
                   'gui_OutputFcn',  @compare_12_3_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function compare_12_3_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
hoy=date;
set(handles.today, 'String', hoy)

function varargout = compare_12_3_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function popupmenux_Callback(hObject, eventdata, handles)
global  y x xtag

strx = get(hObject, 'String');
valx = get(hObject,'Value');
switch strx{valx};

    case 'Time'
%         horas=datenum(titchMDM,'HH:MM:SS');
%         x = horas;
          x=0;
          xtag='Time';
    case 'G [  W/m² ]'
        %x = MDM(:,1);
        x=1;
        xtag='G [  W/m² ]';
    case 'Tmod [ °C ]'
        %x = MDM(:,2);
        x=2;
         xtag='Tmod [ °C ]';
    case 'Umpp [ V ]'
        x=3;
       xtag='Umpp [ V ]';
        %x = MDM(:,3);
    case 'Impp [ A ]'
```

```
        x=4;
        xtag='Impp [ A ]';
        %x = MDM(:,4);
    case 'Pmpp [ W ]'
        x=5;
        xtag='Pmpp [ W ]';
        %x = MDM(:,5);
    case 'Uoc [ V ]'
        x=6;
        xtag='Uoc [ V ]';
        %x = MDM(:,6);
    case 'Isc [ A ]'
        x=7;
        xtag='Isc [ A ]';
        %x = MDM(:,7);
    case 'FF [ % ]'
        x=8;
        xtag='FF [ % ]';
        %x = MDM(:,8);
    case 'Pmpp,STC [ W ]'
        x=9;
        xtag='Pmpp,STC [ W ]';
        %x = MDM(:,9);
    case 'Uoc,STC [ V ]'
        x=10;
        xtag='Uoc,STC [ V ]';
        %x = MDM(:,10);
    case 'Isc,STC [ A ]'
        x=11;
        xtag='Isc,STC [ A ]';
        %x = MDM(:,11);
    case 'FF,STC [ % ]'
       x=12;
       xtag='FF,STC [ % ]';
        %x = MDM(:,12);
    case 'Rs [ Ohm ]'
        x=13;
         xtag='Rs [ Ohm ]';
        %x = MDM(:,13);
    case 'Rp [ Ohm ]'
        x=14;
        xtag='Rp [ Ohm ]';
        %x = MDM(:,14);
    case 'eta [%]'
        x=15;
        xtag='eta [%]';
        %x = nu(:);
end

function popupmenux_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenuy_Callback(hObject, eventdata, handles)
global  y x ytag
```

```
stry = get(hObject, 'String');
valy = get(hObject,'Value');
switch stry{valy};

    case 'Time'
%         horas=datenum(titchMDM,'HH:MM:SS');
%         y = horas;
          y=0;
          ytag='Time';
    case 'G [  W/m² ]'
        %y = MDM(:,1);
        y=1;
        ytag='G [  W/m² ]';
    case 'Tmod [ °C ]'
        %y = MDM(:,2);
        y=2;
         ytag='Tmod [ °C ]';
    case 'Umpp [ V ]'
        y=3;
       ytag='Umpp [ V ]';
        %y = MDM(:,3);
    case 'Impp [ A ]'
        y=4;
        ytag='Impp [ A ]';
        %y = MDM(:,4);
    case 'Pmpp [ W ]'
        y=5;
        ytag='Pmpp [ W ]';
        %y = MDM(:,5);
    case 'Uoc [ V ]'
        y=6;
        ytag='Uoc [ V ]';
        %y = MDM(:,6);
    case 'Isc [ A ]'
        y=7;
        ytag='Isc [ A ]';
        %y = MDM(:,7);
    case 'FF [ % ]'
        y=8;
        ytag='FF [ % ]';
        %y = MDM(:,8);
    case 'Pmpp,STC [ W ]'
        y=9;
        ytag='Pmpp,STC [ W ]';
        %y = MDM(:,9);
    case 'Uoc,STC [ V ]'
        y=10;
        ytag='Uoc,STC [ V ]';
        %y = MDM(:,10);
    case 'Isc,STC [ A ]'
        y=11;
        ytag='Isc,STC [ A ]';
        %y = MDM(:,11);
    case 'FF,STC [ % ]'
       y=12;
       ytag='FF,STC [ % ]';
```

```
        %y = MDM(:,12);
    case 'Rs [ Ohm ]'
        y=13;
         ytag='Rs [ Ohm ]';
        %y = MDM(:,13);
    case 'Rp [ Ohm ]'
        y=14;
        ytag='Rp [ Ohm ]';
        %y = MDM(:,14);
    case 'eta [%]'
        y=15;
        ytag='eta [%]';
        %y = nu(:);
end

function popupmenuy_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function but_drawn_Callback(hObject, eventdata, handles)
global  y x
compare_12_4

function today_Callback(hObject, eventdata, handles)

function today_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```


compare_12_4
Modul 1
Uoc [V]
0 100 200 300 400 500 600 700 800 900 1000
G [W/m²]
Modul 2
Uoc [V]
0 200 400 600 800 1000 1200
G [W/m²]
Modul 3
Uoc [V]
0 100 200 300 400 500 600 700 800 900 1000
G [W/m²]
Modul 4
Uoc [V]
0 200 400 600 800 1000 1200
G [W/m²]
Compare
SOLARWORLD
Save

```
function varargout = compare_12_4(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compare_12_4_OpeningFcn, ...
                   'gui_OutputFcn',  @compare_12_4_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function compare_12_4_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

global x y
global  car_m_1 car_m_2 car_m_3 car_m_4 car_m_5 car_m_6 car_m_7 car_m_8
car_m_9 car_m_10 car_m_11 car_m_12 cont c_a
global MDM_m_1 MDMi_m_1 titMDM_m_1 titchMDM_m_1 fechaMDM_m_1 horai_m_1
horaf_m_1 C_datos_m_1 Irr_m_1
global MDM_m_2 MDMi_m_2 titMDM_m_2 titchMDM_m_2 fechaMDM_m_2 horai_m_2
horaf_m_2 C_datos_m_2 Irr_m_2
global MDM_m_3 MDMi_m_3 titMDM_m_3 titchMDM_m_3 fechaMDM_m_3 horai_m_3
horaf_m_3 C_datos_m_3 Irr_m_3
global MDM_m_4 MDMi_m_4 titMDM_m_4 titchMDM_m_4 fechaMDM_m_4 horai_m_4
horaf_m_4 C_datos_m_4 Irr_m_4
global MDM_m_5 MDMi_m_5 titMDM_m_5 titchMDM_m_5 fechaMDM_m_5 horai_m_5
horaf_m_5 C_datos_m_5 Irr_m_5
global MDM_m_6 MDMi_m_6 titMDM_m_6 titchMDM_m_6 fechaMDM_m_6 horai_m_6
horaf_m_6 C_datos_m_6 Irr_m_6
global MDM_m_7 MDMi_m_7 titMDM_m_7 titchMDM_m_7 fechaMDM_m_7 horai_m_7
horaf_m_7 C_datos_m_7 Irr_m_7
global MDM_m_8 MDMi_m_8 titMDM_m_8 titchMDM_m_8 fechaMDM_m_8 horai_m_8
horaf_m_8 C_datos_m_8 Irr_m_8
global MDM_m_9 MDMi_m_9 titMDM_m_9 titchMDM_m_9 fechaMDM_m_9 horai_m_9
horaf_m_9 C_datos_m_9 Irr_m_9
global MDM_m_10 MDMi_m_10 titMDM_m_10 titchMDM_m_10 fechaMDM_m_10
horai_m_10 horaf_m_10 C_datos_m_10 Irr_m_10
global MDM_m_11 MDMi_m_11 titMDM_m_11 titchMDM_m_11 fechaMDM_m_11
horai_m_11 horaf_m_11 C_datos_m_11 Irr_m_11
global MDM_m_12 MDMi_m_12 titMDM_m_12 titchMDM_m_12 fechaMDM_m_12
horai_m_12 horaf_m_12 C_datos_m_12 Irr_m_12
global nu_m_1 nu_m_2 nu_m_3 nu_m_4 nu_m_5 nu_m_6 nu_m_7 nu_m_8 nu_m_9
nu_m_10 nu_m_11 nu_m_12 xtag ytag
global x_m_1 y_m_1 x_m_2 y_m_2 x_m_3 y_m_3 x_m_4 y_m_4 x_m_5 y_m_5 x_m_6
y_m_6 x_m_7 y_m_7 x_m_8 y_m_8 x_m_9 y_m_9 x_m_10 y_m_10 x_m_11 y_m_11
x_m_12 y_m_12
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
```

```
axis off
x_m_1=[];
y_m_1=[];

x_m_2=[];
y_m_2=[];

x_m_3=[];
y_m_3=[];
x_m_4=[];
y_m_4=[];
x_m_5=[];
y_m_5=[];

x_m_6=[];
y_m_6=[];
x_m_7=[];
y_m_7=[];
x_m_8=[];
y_m_8=[];
x_m_9=[];
y_m_9=[];
x_m_10=[];
y_m_10=[];
x_m_11=[];
y_m_11=[];
x_m_12=[];
y_m_12=[];
i=0;
c_a_m=c_a/10000;
if car_m_1==1
    nu_m_1=MDM_m_1(:,5)./(c_a_m.*MDM_m_1(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_1,nu_m_1)
        datetick('x','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=horas_m_1;
        y_m_1=nu_m_1;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_1,horas_m_1)
        datetick('y','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=horas_m_1;
        x_m_1=nu_m_1;
    elseif x==0
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_1,MDM_m_1(:,y))
```

```
        datetick('x','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=horas_m_1;
        y_m_1=MDM_m_1(:,y);
    elseif y==0
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_1(:,x),horas_m_1)
        datetick('y','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=horas_m_1;
        x_m_1=MDM_m_1(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_1,MDM_m_1(:,y))
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=nu_m_1;
        y_m_1=MDM_m_1(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_1(:,x),nu_m_1)
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=nu_m_1;
        x_m_1=MDM_m_1(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_1(:,x),MDM_m_1(:,y))
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=MDM_m_1(:,y);
        x_m_1=MDM_m_1(:,x);
    end
end
if car_m_2==1
    nu_m_2=MDM_m_2(:,5)./(c_a_m.*MDM_m_2(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_2,nu_m_2)
        datetick('x','HH:MM:SS')
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        x_m_2=horas_m_2;
        y_m_2=nu_m_2;
    elseif (y==0) && (x==15)
        i=i+1;
```

```
        horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_2,horas_m_2)
        datetick('y','HH:MM:SS')
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        y_m_2=horas_m_2;
        x_m_2=nu_m_2;
    elseif x==0
        i=i+1;
        horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_2,MDM_m_2(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        x_m_2=horas_m_2;
        y_m_2=MDM_m_2(:,y);
    elseif y==0
        i=i+1;
        horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_2(:,x),horas_m_2)
        datetick('y','HH:MM:SS')
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        y_m_2=horas_m_2;
        x_m_2=MDM_m_2(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_2,MDM_m_2(:,y))
        %i=i+1;
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        x_m_2=nu_m_2;
        y_m_2=MDM_m_2(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_2(:,x),nu_m_2)
        %i=i+1;
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        y_m_2=nu_m_2;
        x_m_2=MDM_m_2(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_2(:,x),MDM_m_2(:,y))
        %i=i+1;
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        y_m_2=MDM_m_2(:,y);
        x_m_2=MDM_m_2(:,x);
    end
end
```

```
if car_m_3==1
    nu_m_3=MDM_m_3(:,5)./(c_a_m.*MDM_m_3(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_3,nu_m_3)
        datetick('x','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=horas_m_3;
        y_m_3=nu_m_3;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_3,horas_m_3)
        datetick('y','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=horas_m_3;
        x_m_3=nu_m_3;
    elseif x==0
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_3,MDM_m_3(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=horas_m_3;
        y_m_3=MDM_m_3(:,y);
    elseif y==0
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_3(:,x),horas_m_3)
        datetick('y','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=horas_m_3;
        x_m_3=MDM_m_3(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_3,MDM_m_3(:,y))
        %         i=i+1;
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=nu_m_3;
        y_m_3=MDM_m_3(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_3(:,x),nu_m_3)
        %         i=i+1;
        title('Modul 3')
        xlabel(xtag)
```

```
        ylabel(ytag)
        y_m_3=nu_m_3;
        x_m_3=MDM_m_3(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_3(:,x),MDM_m_3(:,y))
        %         i=i+1;
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=MDM_m_3(:,y);
        x_m_3=MDM_m_3(:,x);
    end
end
if car_m_4==1
    nu_m_4=MDM_m_4(:,5)./(c_a_m.*MDM_m_4(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_4,nu_m_4)
        datetick('x','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=horas_m_4;
        y_m_4=nu_m_4;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_4,horas_m_4)
        datetick('y','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=horas_m_4;
        x_m_4=nu_m_4;
    elseif x==0
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_4,MDM_m_4(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=horas_m_4;
        y_m_4=MDM_m_4(:,y);
    elseif y==0
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_4(:,x),horas_m_4)
        datetick('y','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=horas_m_4;
        x_m_4=MDM_m_4(:,x);
    elseif x==15
```

```
        i=i+1;
        subplot(cont,1,i),plot(nu_m_4,MDM_m_4(:,y))
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=nu_m_4;
        y_m_4=MDM_m_4(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_4(:,x),nu_m_4)
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=nu_m_4;
        x_m_4=MDM_m_4(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_4(:,x),MDM_m_4(:,y))
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=MDM_m_4(:,y);
        x_m_4=MDM_m_4(:,x);
    end
end
if car_m_5==1
    nu_m_5=MDM_m_5(:,5)./(c_a_m.*MDM_m_5(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_5,nu_m_5)
        datetick('x','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=horas_m_5;
        y_m_5=nu_m_5;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_5,horas_m_5)
        datetick('y','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=horas_m_5;
        x_m_5=nu_m_5;
    elseif x==0
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_5,MDM_m_5(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 5')
```

```
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=horas_m_5;
        y_m_5=MDM_m_5(:,y);
    elseif y==0
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_5(:,x),horas_m_5)
        datetick('y','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=horas_m_5;
        x_m_5=MDM_m_5(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_5,MDM_m_5(:,y))
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=nu_m_5;
        y_m_5=MDM_m_5(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_5(:,x),nu_m_5)
        %         i=i+1;
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=nu_m_5;
        x_m_5=MDM_m_5(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_5(:,x),MDM_m_5(:,y))
        %         i=i+1;
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=MDM_m_5(:,y);
        x_m_5=MDM_m_5(:,x);
    end
end
if car_m_6==1
    nu_m_6=MDM_m_6(:,5)./(c_a_m.*MDM_m_6(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_6,nu_m_6)
        datetick('x','HH:MM:SS')
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        x_m_6=horas_m_6;
        y_m_6=nu_m_6;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
```

```
        subplot(cont,1,i),plot(nu_m_6,horas_m_6)
        datetick('y','HH:MM:SS')
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        y_m_6=horas_m_6;
        x_m_6=nu_m_6;
    elseif x==0
        i=i+1;
        horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_6,MDM_m_6(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        x_m_6=horas_m_6;
        y_m_6=MDM_m_6(:,y);
    elseif y==0
        i=i+1;
        horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_6(:,x),horas_m_6)
        datetick('y','HH:MM:SS')
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        y_m_6=horas_m_6;
        x_m_6=MDM_m_6(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_6,MDM_m_6(:,y))
        %         i=i+1;
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        x_m_6=nu_m_6;
        y_m_6=MDM_m_6(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_6(:,x),nu_m_6)
        %         i=i+1;
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        y_m_6=nu_m_6;
        x_m_6=MDM_m_6(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_6(:,x),MDM_m_6(:,y))
        %         i=i+1;
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        y_m_6=MDM_m_6(:,y);
        x_m_6=MDM_m_6(:,x);
    end
end
if car_m_7==1
```

```
nu_m_7=MDM_m_7(:,5)./(c_a_m.*MDM_m_7(:,1));

if (x==0) && (y==15)
    i=i+1;
    horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_7,nu_m_7)
    datetick('x','HH:MM:SS')
    title('Modul 7')
    xlabel(xtag)
    ylabel(ytag)
    x_m_7=horas_m_7;
    y_m_7=nu_m_7;
elseif (y==0) && (x==15)
    i=i+1;
    horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
    subplot(cont,1,i),plot(nu_m_7,horas_m_7)
    datetick('y','HH:MM:SS')
    title('Modul 7')
    xlabel(xtag)
    ylabel(ytag)
    y_m_7=horas_m_7;
    x_m_7=nu_m_7;
elseif x==0
    i=i+1;
    horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_7,MDM_m_7(:,y))
    datetick('x','HH:MM:SS')
    title('Modul 7')
    xlabel(xtag)
    ylabel(ytag)
    x_m_7=horas_m_7;
    y_m_7=MDM_m_7(:,y);
elseif y==0
    i=i+1;
    horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
    subplot(cont,1,i),plot(MDM_m_7(:,x),horas_m_7)
    datetick('y','HH:MM:SS')
    title('Modul 7')
    xlabel(xtag)
    ylabel(ytag)
    y_m_7=horas_m_7;
    x_m_7=MDM_m_7(:,x);
elseif x==15
    i=i+1;
    subplot(cont,1,i),plot(nu_m_7,MDM_m_7(:,y))
    %         i=i+1;
    title('Modul 7')
    xlabel(xtag)
    ylabel(ytag)
    x_m_7=nu_m_7;
    y_m_7=MDM_m_7(:,y);
elseif y==15
    i=i+1;
    subplot(cont,1,i),plot(MDM_m_7(:,x),nu_m_7)
    %         i=i+1;
    title('Modul 7')
    xlabel(xtag)
```

```
        ylabel(ytag)
        y_m_7=nu_m_7;
        x_m_7=MDM_m_7(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_7(:,x),MDM_m_7(:,y))
        %         i=i+1;
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        y_m_7=MDM_m_7(:,y);
        x_m_7=MDM_m_7(:,x);
    end
end
if car_m_8==1
    nu_m_8=MDM_m_8(:,5)./(c_a_m.*MDM_m_8(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_8,nu_m_8)
        datetick('x','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=horas_m_8;
        y_m_8=nu_m_8;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_8,horas_m_8)
        datetick('y','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=horas_m_8;
        x_m_8=nu_m_8;
    elseif x==0
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_8,MDM_m_8(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=horas_m_8;
        y_m_8=MDM_m_8(:,y);
    elseif y==0
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_8(:,x),horas_m_8)
        datetick('y','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=horas_m_8;
        x_m_8=MDM_m_8(:,x);
```

```
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_8,MDM_m_8(:,y))
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=nu_m_8;
        y_m_8=MDM_m_8(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_8(:,x),nu_m_8)
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=nu_m_8;
        x_m_8=MDM_m_8(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_8(:,x),MDM_m_8(:,y))
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=MDM_m_8(:,y);
        x_m_8=MDM_m_8(:,x);
    end
end
if car_m_9==1
    nu_m_9=MDM_m_9(:,5)./(c_a_m.*MDM_m_9(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_9,nu_m_9)
        datetick('x','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=horas_m_9;
        y_m_9=nu_m_9;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_9,horas_m_9)
        datetick('y','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=horas_m_9;
        x_m_9=nu_m_9;
    elseif x==0
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_9,MDM_m_9(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 9')
```

```
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=horas_m_9;
        y_m_9=MDM_m_9(:,y);
    elseif y==0
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_9(:,x),horas_m_9)
        datetick('y','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=horas_m_9;
        x_m_9=MDM_m_9(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_9,MDM_m_9(:,y))
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=nu_m_9;
        y_m_9=MDM_m_9(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_9(:,x),nu_m_9)
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=nu_m_9;
        x_m_9=MDM_m_9(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_9(:,x),MDM_m_9(:,y))
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=MDM_m_9(:,y);
        x_m_9=MDM_m_9(:,x);
    end
end
if car_m_10==1
    nu_m_10=MDM_m_10(:,5)./(c_a_m.*MDM_m_10(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_10,nu_m_10)
        datetick('x','HH:MM:SS')
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        x_m_10=horas_m_10;
        y_m_10=nu_m_10;
    elseif (y==0) && (x==15)
        i=i+1;
```

```
        horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_10,horas_m_10)
        datetick('y','HH:MM:SS')
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        y_m_10=horas_m_10;
        x_m_10=nu_m_10;
    elseif x==0
        i=i+1;
        horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_10,MDM_m_10(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        x_m_10=horas_m_10;
        y_m_10=MDM_m_10(:,y);
    elseif y==0
        i=i+1;
        horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_10(:,x),horas_m_10)
        datetick('y','HH:MM:SS')
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        y_m_10=horas_m_10;
        x_m_10=MDM_m_10(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_10,MDM_m_10(:,y))
        %         i=i+1;
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        x_m_10=nu_m_10;
        y_m_10=MDM_m_10(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_10(:,x),nu_m_10)
        %         i=i+1;
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        y_m_10=nu_m_10;
        x_m_10=MDM_m_10(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_10(:,x),MDM_m_10(:,y))
        %         i=i+1;
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        y_m_10=MDM_m_10(:,y);
        x_m_10=MDM_m_10(:,x);
    end
end
```

```
if car_m_11==1
    nu_m_11=MDM_m_11(:,5)./(c_a_m.*MDM_m_11(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_11,nu_m_11)
        datetick('x','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=horas_m_11;
        y_m_11=nu_m_11;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_11,horas_m_11)
        datetick('y','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=horas_m_11;
        x_m_11=nu_m_11;
    elseif x==0
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_11,MDM_m_11(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=horas_m_11;
        y_m_11=MDM_m_11(:,y);
    elseif y==0
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_11(:,x),horas_m_11)
        datetick('y','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=horas_m_11;
        x_m_11=MDM_m_11(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_11,MDM_m_11(:,y))
        %         i=i+1;
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=nu_m_11;
        y_m_11=MDM_m_11(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_11(:,x),nu_m_11)
        %         i=i+1;
        title('Modul 11')
        xlabel(xtag)
```

```
        ylabel(ytag)
        y_m_11=nu_m_11;
        x_m_11=MDM_m_11(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_11(:,x),MDM_m_11(:,y))
        %         i=i+1;
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=MDM_m_11(:,y);
        x_m_11=MDM_m_11(:,x);
    end
end
if car_m_12==1
    nu_m_12=MDM_m_12(:,5)./(c_a_m.*MDM_m_12(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_12,nu_m_12)
        datetick('x','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=horas_m_12;
        y_m_12=nu_m_12;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_12,horas_m_12)
        datetick('y','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=horas_m_12;
        x_m_12=nu_m_12;
    elseif x==0
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_12,MDM_m_12(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=horas_m_12;
        y_m_12=MDM_m_12(:,y);
    elseif y==0
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_12(:,x),horas_m_12)
        datetick('y','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=horas_m_12;
        x_m_12=MDM_m_12(:,x);
    elseif x==15
```

```
        i=i+1;
        subplot(cont,1,i),plot(nu_m_12,MDM_m_12(:,y))
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=nu_m_12;
        y_m_12=MDM_m_12(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_12(:,x),nu_m_12)
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=nu_m_12;
        x_m_12=MDM_m_12(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_12(:,x),MDM_m_12(:,y))
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=MDM_m_12(:,y);
        x_m_12=MDM_m_12(:,x);
    end
end

function varargout = compare_12_4_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function but_compare_Callback(hObject, eventdata, handles)
compare_12_5

function savebutton_Callback(hObject, eventdata, handles)
global x y
global  car_m_1 car_m_2 car_m_3 car_m_4 car_m_5 car_m_6 car_m_7 car_m_8
car_m_9 car_m_10 car_m_11 car_m_12 cont c_a
global MDM_m_1 MDMi_m_1 titMDM_m_1 titchMDM_m_1 fechaMDM_m_1 horai_m_1
horaf_m_1 C_datos_m_1 Irr_m_1
global MDM_m_2 MDMi_m_2 titMDM_m_2 titchMDM_m_2 fechaMDM_m_2 horai_m_2
horaf_m_2 C_datos_m_2 Irr_m_2
global MDM_m_3 MDMi_m_3 titMDM_m_3 titchMDM_m_3 fechaMDM_m_3 horai_m_3
horaf_m_3 C_datos_m_3 Irr_m_3
global MDM_m_4 MDMi_m_4 titMDM_m_4 titchMDM_m_4 fechaMDM_m_4 horai_m_4
horaf_m_4 C_datos_m_4 Irr_m_4
global MDM_m_5 MDMi_m_5 titMDM_m_5 titchMDM_m_5 fechaMDM_m_5 horai_m_5
horaf_m_5 C_datos_m_5 Irr_m_5
global MDM_m_6 MDMi_m_6 titMDM_m_6 titchMDM_m_6 fechaMDM_m_6 horai_m_6
horaf_m_6 C_datos_m_6 Irr_m_6
global MDM_m_7 MDMi_m_7 titMDM_m_7 titchMDM_m_7 fechaMDM_m_7 horai_m_7
horaf_m_7 C_datos_m_7 Irr_m_7
global MDM_m_8 MDMi_m_8 titMDM_m_8 titchMDM_m_8 fechaMDM_m_8 horai_m_8
horaf_m_8 C_datos_m_8 Irr_m_8
global MDM_m_9 MDMi_m_9 titMDM_m_9 titchMDM_m_9 fechaMDM_m_9 horai_m_9
horaf_m_9 C_datos_m_9 Irr_m_9
```

```
global MDM_m_10 MDMi_m_10 titMDM_m_10 titchMDM_m_10 fechaMDM_m_10
horai_m_10 horaf_m_10 C_datos_m_10 Irr_m_10
global MDM_m_11 MDMi_m_11 titMDM_m_11 titchMDM_m_11 fechaMDM_m_11
horai_m_11 horaf_m_11 C_datos_m_11 Irr_m_11
global MDM_m_12 MDMi_m_12 titMDM_m_12 titchMDM_m_12 fechaMDM_m_12
horai_m_12 horaf_m_12 C_datos_m_12 Irr_m_12
global nu_m_1 nu_m_2 nu_m_3 nu_m_4 nu_m_5 nu_m_6 nu_m_7 nu_m_8 nu_m_9
nu_m_10 nu_m_11 nu_m_12 xtag ytag
global x_m_1 y_m_1 x_m_2 y_m_2 x_m_3 y_m_3 x_m_4 y_m_4 x_m_5 y_m_5 x_m_6
y_m_6 x_m_7 y_m_7 x_m_8 y_m_8 x_m_9 y_m_9 x_m_10 y_m_10 x_m_11 y_m_11
x_m_12 y_m_12
figure
x_m_1=[];
y_m_1=[];

x_m_2=[];
y_m_2=[];

x_m_3=[];
y_m_3=[];
x_m_4=[];
y_m_4=[];
x_m_5=[];
y_m_5=[];

x_m_6=[];
y_m_6=[];
x_m_7=[];
y_m_7=[];
x_m_8=[];
y_m_8=[];
x_m_9=[];
y_m_9=[];
x_m_10=[];
y_m_10=[];
x_m_11=[];
y_m_11=[];
x_m_12=[];
y_m_12=[];
i=0;
c_a_m=c_a/10000;
if car_m_1==1
    nu_m_1=MDM_m_1(:,5)./(c_a_m.*MDM_m_1(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_1,nu_m_1)
        datetick('x','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=horas_m_1;
        y_m_1=nu_m_1;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
```

```
        subplot(cont,1,i),plot(nu_m_1,horas_m_1)
        datetick('y','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=horas_m_1;
        x_m_1=nu_m_1;
    elseif x==0
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_1,MDM_m_1(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=horas_m_1;
        y_m_1=MDM_m_1(:,y);
    elseif y==0
        i=i+1;
        horas_m_1=datenum(titchMDM_m_1,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_1(:,x),horas_m_1)
        datetick('y','HH:MM:SS')
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=horas_m_1;
        x_m_1=MDM_m_1(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_1,MDM_m_1(:,y))
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        x_m_1=nu_m_1;
        y_m_1=MDM_m_1(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_1(:,x),nu_m_1)
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=nu_m_1;
        x_m_1=MDM_m_1(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_1(:,x),MDM_m_1(:,y))
        title('Modul 1')
        xlabel(xtag)
        ylabel(ytag)
        y_m_1=MDM_m_1(:,y);
        x_m_1=MDM_m_1(:,x);
    end
end
if car_m_2==1
    nu_m_2=MDM_m_2(:,5)./(c_a_m.*MDM_m_2(:,1));

    if (x==0) && (y==15)
```

```
    i=i+1;
    horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_2,nu_m_2)
    datetick('x','HH:MM:SS')
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    x_m_2=horas_m_2;
    y_m_2=nu_m_2;
elseif (y==0) && (x==15)
    i=i+1;
    horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
    subplot(cont,1,i),plot(nu_m_2,horas_m_2)
    datetick('y','HH:MM:SS')
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    y_m_2=horas_m_2;
    x_m_2=nu_m_2;
elseif x==0
    i=i+1;
    horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_2,MDM_m_2(:,y))
    datetick('x','HH:MM:SS')
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    x_m_2=horas_m_2;
    y_m_2=MDM_m_2(:,y);
elseif y==0
    i=i+1;
    horas_m_2=datenum(titchMDM_m_2,'HH:MM:SS');
    subplot(cont,1,i),plot(MDM_m_2(:,x),horas_m_2)
    datetick('y','HH:MM:SS')
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    y_m_2=horas_m_2;
    x_m_2=MDM_m_2(:,x);
elseif x==15
    i=i+1;
    subplot(cont,1,i),plot(nu_m_2,MDM_m_2(:,y))
    %i=i+1;
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    x_m_2=nu_m_2;
    y_m_2=MDM_m_2(:,y);
elseif y==15
    i=i+1;
    subplot(cont,1,i),plot(MDM_m_2(:,x),nu_m_2)
    %i=i+1;
    title('Modul 2')
    xlabel(xtag)
    ylabel(ytag)
    y_m_2=nu_m_2;
    x_m_2=MDM_m_2(:,x);
```

```
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_2(:,x),MDM_m_2(:,y))
        %i=i+1;
        title('Modul 2')
        xlabel(xtag)
        ylabel(ytag)
        y_m_2=MDM_m_2(:,y);
        x_m_2=MDM_m_2(:,x);
    end
end
if car_m_3==1
    nu_m_3=MDM_m_3(:,5)./(c_a_m.*MDM_m_3(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_3,nu_m_3)
        datetick('x','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=horas_m_3;
        y_m_3=nu_m_3;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_3,horas_m_3)
        datetick('y','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=horas_m_3;
        x_m_3=nu_m_3;
    elseif x==0
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_3,MDM_m_3(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=horas_m_3;
        y_m_3=MDM_m_3(:,y);
    elseif y==0
        i=i+1;
        horas_m_3=datenum(titchMDM_m_3,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_3(:,x),horas_m_3)
        datetick('y','HH:MM:SS')
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=horas_m_3;
        x_m_3=MDM_m_3(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_3,MDM_m_3(:,y))
        %         i=i+1;
```

```
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        x_m_3=nu_m_3;
        y_m_3=MDM_m_3(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_3(:,x),nu_m_3)
        %         i=i+1;
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=nu_m_3;
        x_m_3=MDM_m_3(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_3(:,x),MDM_m_3(:,y))
        %         i=i+1;
        title('Modul 3')
        xlabel(xtag)
        ylabel(ytag)
        y_m_3=MDM_m_3(:,y);
        x_m_3=MDM_m_3(:,x);
    end
end
if car_m_4==1
    nu_m_4=MDM_m_4(:,5)./(c_a_m.*MDM_m_4(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_4,nu_m_4)
        datetick('x','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=horas_m_4;
        y_m_4=nu_m_4;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_4,horas_m_4)
        datetick('y','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=horas_m_4;
        x_m_4=nu_m_4;
    elseif x==0
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_4,MDM_m_4(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=horas_m_4;
        y_m_4=MDM_m_4(:,y);
```

```
    elseif y==0
        i=i+1;
        horas_m_4=datenum(titchMDM_m_4,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_4(:,x),horas_m_4)
        datetick('y','HH:MM:SS')
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=horas_m_4;
        x_m_4=MDM_m_4(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_4,MDM_m_4(:,y))
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        x_m_4=nu_m_4;
        y_m_4=MDM_m_4(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_4(:,x),nu_m_4)
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=nu_m_4;
        x_m_4=MDM_m_4(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_4(:,x),MDM_m_4(:,y))
        %         i=i+1;
        title('Modul 4')
        xlabel(xtag)
        ylabel(ytag)
        y_m_4=MDM_m_4(:,y);
        x_m_4=MDM_m_4(:,x);
    end
end
if car_m_5==1
    nu_m_5=MDM_m_5(:,5)./(c_a_m.*MDM_m_5(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_5,nu_m_5)
        datetick('x','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=horas_m_5;
        y_m_5=nu_m_5;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_5,horas_m_5)
        datetick('y','HH:MM:SS')
```

```
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=horas_m_5;
        x_m_5=nu_m_5;
    elseif x==0
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_5,MDM_m_5(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=horas_m_5;
        y_m_5=MDM_m_5(:,y);
    elseif y==0
        i=i+1;
        horas_m_5=datenum(titchMDM_m_5,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_5(:,x),horas_m_5)
        datetick('y','HH:MM:SS')
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=horas_m_5;
        x_m_5=MDM_m_5(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_5,MDM_m_5(:,y))
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        x_m_5=nu_m_5;
        y_m_5=MDM_m_5(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_5(:,x),nu_m_5)
        %         i=i+1;
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=nu_m_5;
        x_m_5=MDM_m_5(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_5(:,x),MDM_m_5(:,y))
        %         i=i+1;
        title('Modul 5')
        xlabel(xtag)
        ylabel(ytag)
        y_m_5=MDM_m_5(:,y);
        x_m_5=MDM_m_5(:,x);
    end
end
if car_m_6==1
    nu_m_6=MDM_m_6(:,5)./(c_a_m.*MDM_m_6(:,1));
    if (x==0) && (y==15)
        i=i+1;
```

```
    horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_6,nu_m_6)
    datetick('x','HH:MM:SS')
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    x_m_6=horas_m_6;
    y_m_6=nu_m_6;
elseif (y==0) && (x==15)
    i=i+1;
    horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
    subplot(cont,1,i),plot(nu_m_6,horas_m_6)
    datetick('y','HH:MM:SS')
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    y_m_6=horas_m_6;
    x_m_6=nu_m_6;
elseif x==0
    i=i+1;
    horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_6,MDM_m_6(:,y))
    datetick('x','HH:MM:SS')
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    x_m_6=horas_m_6;
    y_m_6=MDM_m_6(:,y);
elseif y==0
    i=i+1;
    horas_m_6=datenum(titchMDM_m_6,'HH:MM:SS');
    subplot(cont,1,i),plot(MDM_m_6(:,x),horas_m_6)
    datetick('y','HH:MM:SS')
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    y_m_6=horas_m_6;
    x_m_6=MDM_m_6(:,x);
elseif x==15
    i=i+1;
    subplot(cont,1,i),plot(nu_m_6,MDM_m_6(:,y))
    %         i=i+1;
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    x_m_6=nu_m_6;
    y_m_6=MDM_m_6(:,y);
elseif y==15
    i=i+1;
    subplot(cont,1,i),plot(MDM_m_6(:,x),nu_m_6)
    %         i=i+1;
    title('Modul 6')
    xlabel(xtag)
    ylabel(ytag)
    y_m_6=nu_m_6;
    x_m_6=MDM_m_6(:,x);
else
```

```
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_6(:,x),MDM_m_6(:,y))
        %         i=i+1;
        title('Modul 6')
        xlabel(xtag)
        ylabel(ytag)
        y_m_6=MDM_m_6(:,y);
        x_m_6=MDM_m_6(:,x);
    end
end
if car_m_7==1
    nu_m_7=MDM_m_7(:,5)./(c_a_m.*MDM_m_7(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_7,nu_m_7)
        datetick('x','HH:MM:SS')
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        x_m_7=horas_m_7;
        y_m_7=nu_m_7;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_7,horas_m_7)
        datetick('y','HH:MM:SS')
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        y_m_7=horas_m_7;
        x_m_7=nu_m_7;
    elseif x==0
        i=i+1;
        horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_7,MDM_m_7(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        x_m_7=horas_m_7;
        y_m_7=MDM_m_7(:,y);
    elseif y==0
        i=i+1;
        horas_m_7=datenum(titchMDM_m_7,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_7(:,x),horas_m_7)
        datetick('y','HH:MM:SS')
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        y_m_7=horas_m_7;
        x_m_7=MDM_m_7(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_7,MDM_m_7(:,y))
        %         i=i+1;
```

```
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        x_m_7=nu_m_7;
        y_m_7=MDM_m_7(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_7(:,x),nu_m_7)
        %         i=i+1;
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        y_m_7=nu_m_7;
        x_m_7=MDM_m_7(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_7(:,x),MDM_m_7(:,y))
        %         i=i+1;
        title('Modul 7')
        xlabel(xtag)
        ylabel(ytag)
        y_m_7=MDM_m_7(:,y);
        x_m_7=MDM_m_7(:,x);
    end
end
if car_m_8==1
    nu_m_8=MDM_m_8(:,5)./(c_a_m.*MDM_m_8(:,1));

    if (x==0) && (y==15)
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_8,nu_m_8)
        datetick('x','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=horas_m_8;
        y_m_8=nu_m_8;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_8,horas_m_8)
        datetick('y','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=horas_m_8;
        x_m_8=nu_m_8;
    elseif x==0
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_8,MDM_m_8(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=horas_m_8;
```

```
        y_m_8=MDM_m_8(:,y);
    elseif y==0
        i=i+1;
        horas_m_8=datenum(titchMDM_m_8,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_8(:,x),horas_m_8)
        datetick('y','HH:MM:SS')
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=horas_m_8;
        x_m_8=MDM_m_8(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_8,MDM_m_8(:,y))
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        x_m_8=nu_m_8;
        y_m_8=MDM_m_8(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_8(:,x),nu_m_8)
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=nu_m_8;
        x_m_8=MDM_m_8(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_8(:,x),MDM_m_8(:,y))
        %         i=i+1;
        title('Modul 8')
        xlabel(xtag)
        ylabel(ytag)
        y_m_8=MDM_m_8(:,y);
        x_m_8=MDM_m_8(:,x);
    end
end
if car_m_9==1
    nu_m_9=MDM_m_9(:,5)./(c_a_m.*MDM_m_9(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_9,nu_m_9)
        datetick('x','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=horas_m_9;
        y_m_9=nu_m_9;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_9,horas_m_9)
        datetick('y','HH:MM:SS')
```

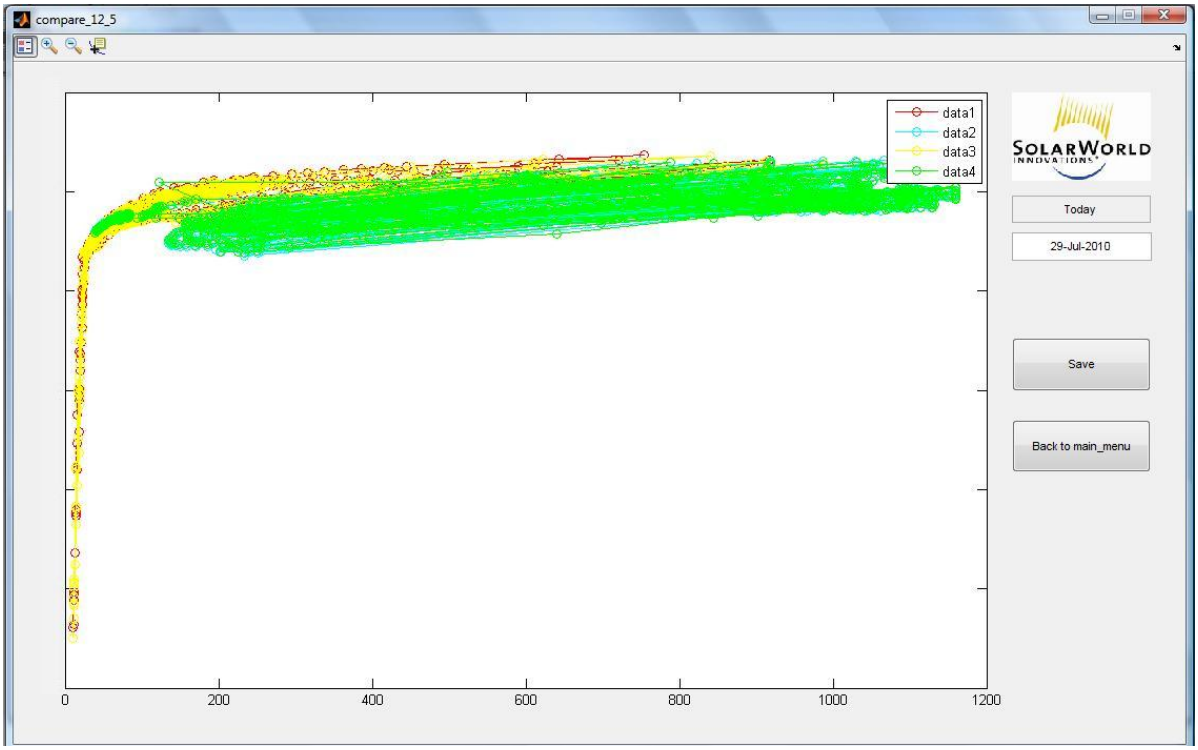
```
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=horas_m_9;
        x_m_9=nu_m_9;
    elseif x==0
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_9,MDM_m_9(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=horas_m_9;
        y_m_9=MDM_m_9(:,y);
    elseif y==0
        i=i+1;
        horas_m_9=datenum(titchMDM_m_9,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_9(:,x),horas_m_9)
        datetick('y','HH:MM:SS')
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=horas_m_9;
        x_m_9=MDM_m_9(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_9,MDM_m_9(:,y))
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        x_m_9=nu_m_9;
        y_m_9=MDM_m_9(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_9(:,x),nu_m_9)
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=nu_m_9;
        x_m_9=MDM_m_9(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_9(:,x),MDM_m_9(:,y))
        %         i=i+1;
        title('Modul 9')
        xlabel(xtag)
        ylabel(ytag)
        y_m_9=MDM_m_9(:,y);
        x_m_9=MDM_m_9(:,x);
    end
end
if car_m_10==1
    nu_m_10=MDM_m_10(:,5)./(c_a_m.*MDM_m_10(:,1));
    if (x==0) && (y==15)
```

```
    i=i+1;
    horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_10,nu_m_10)
    datetick('x','HH:MM:SS')
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    x_m_10=horas_m_10;
    y_m_10=nu_m_10;
elseif (y==0) && (x==15)
    i=i+1;
    horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
    subplot(cont,1,i),plot(nu_m_10,horas_m_10)
    datetick('y','HH:MM:SS')
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    y_m_10=horas_m_10;
    x_m_10=nu_m_10;
elseif x==0
    i=i+1;
    horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
    subplot(cont,1,i),plot(horas_m_10,MDM_m_10(:,y))
    datetick('x','HH:MM:SS')
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    x_m_10=horas_m_10;
    y_m_10=MDM_m_10(:,y);
elseif y==0
    i=i+1;
    horas_m_10=datenum(titchMDM_m_10,'HH:MM:SS');
    subplot(cont,1,i),plot(MDM_m_10(:,x),horas_m_10)
    datetick('y','HH:MM:SS')
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    y_m_10=horas_m_10;
    x_m_10=MDM_m_10(:,x);
elseif x==15
    i=i+1;
    subplot(cont,1,i),plot(nu_m_10,MDM_m_10(:,y))
    %         i=i+1;
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    x_m_10=nu_m_10;
    y_m_10=MDM_m_10(:,y);
elseif y==15
    i=i+1;
    subplot(cont,1,i),plot(MDM_m_10(:,x),nu_m_10)
    %         i=i+1;
    title('Modul 10')
    xlabel(xtag)
    ylabel(ytag)
    y_m_10=nu_m_10;
    x_m_10=MDM_m_10(:,x);
```

```
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_10(:,x),MDM_m_10(:,y))
        %         i=i+1;
        title('Modul 10')
        xlabel(xtag)
        ylabel(ytag)
        y_m_10=MDM_m_10(:,y);
        x_m_10=MDM_m_10(:,x);
    end
end
if car_m_11==1
    nu_m_11=MDM_m_11(:,5)./(c_a_m.*MDM_m_11(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_11,nu_m_11)
        datetick('x','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=horas_m_11;
        y_m_11=nu_m_11;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_11,horas_m_11)
        datetick('y','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=horas_m_11;
        x_m_11=nu_m_11;
    elseif x==0
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_11,MDM_m_11(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=horas_m_11;
        y_m_11=MDM_m_11(:,y);
    elseif y==0
        i=i+1;
        horas_m_11=datenum(titchMDM_m_11,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_11(:,x),horas_m_11)
        datetick('y','HH:MM:SS')
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=horas_m_11;
        x_m_11=MDM_m_11(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_11,MDM_m_11(:,y))
        %         i=i+1;
```

```
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        x_m_11=nu_m_11;
        y_m_11=MDM_m_11(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_11(:,x),nu_m_11)
        %         i=i+1;
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=nu_m_11;
        x_m_11=MDM_m_11(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_11(:,x),MDM_m_11(:,y))
        %         i=i+1;
        title('Modul 11')
        xlabel(xtag)
        ylabel(ytag)
        y_m_11=MDM_m_11(:,y);
        x_m_11=MDM_m_11(:,x);
    end
end
if car_m_12==1
    nu_m_12=MDM_m_12(:,5)./(c_a_m.*MDM_m_12(:,1));
    if (x==0) && (y==15)
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_12,nu_m_12)
        datetick('x','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=horas_m_12;
        y_m_12=nu_m_12;
    elseif (y==0) && (x==15)
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(nu_m_12,horas_m_12)
        datetick('y','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=horas_m_12;
        x_m_12=nu_m_12;
    elseif x==0
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(horas_m_12,MDM_m_12(:,y))
        datetick('x','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=horas_m_12;
        y_m_12=MDM_m_12(:,y);
```

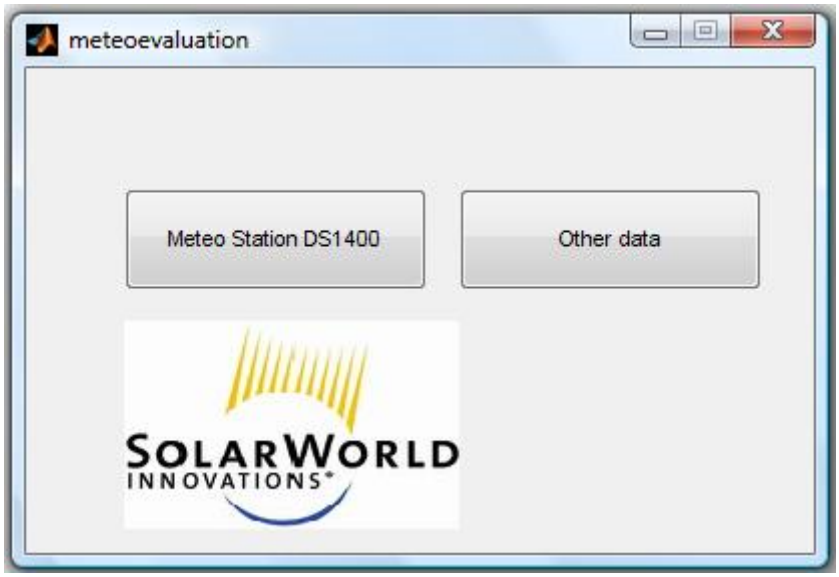
```
    elseif y==0
        i=i+1;
        horas_m_12=datenum(titchMDM_m_12,'HH:MM:SS');
        subplot(cont,1,i),plot(MDM_m_12(:,x),horas_m_12)
        datetick('y','HH:MM:SS')
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=horas_m_12;
        x_m_12=MDM_m_12(:,x);
    elseif x==15
        i=i+1;
        subplot(cont,1,i),plot(nu_m_12,MDM_m_12(:,y))
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        x_m_12=nu_m_12;
        y_m_12=MDM_m_12(:,y);
    elseif y==15
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_12(:,x),nu_m_12)
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=nu_m_12;
        x_m_12=MDM_m_12(:,x);
    else
        i=i+1;
        subplot(cont,1,i),plot(MDM_m_12(:,x),MDM_m_12(:,y))
        %         i=i+1;
        title('Modul 12')
        xlabel(xtag)
        ylabel(ytag)
        y_m_12=MDM_m_12(:,y);
        x_m_12=MDM_m_12(:,x);
    end
end
```


compare_12_5
data1
data2
data3
data4
0
200
400
600
800
1000
1200
SOLARWORLD
INNOVATIONS
Today
29-Jul-2010
Save
Back to main_menu

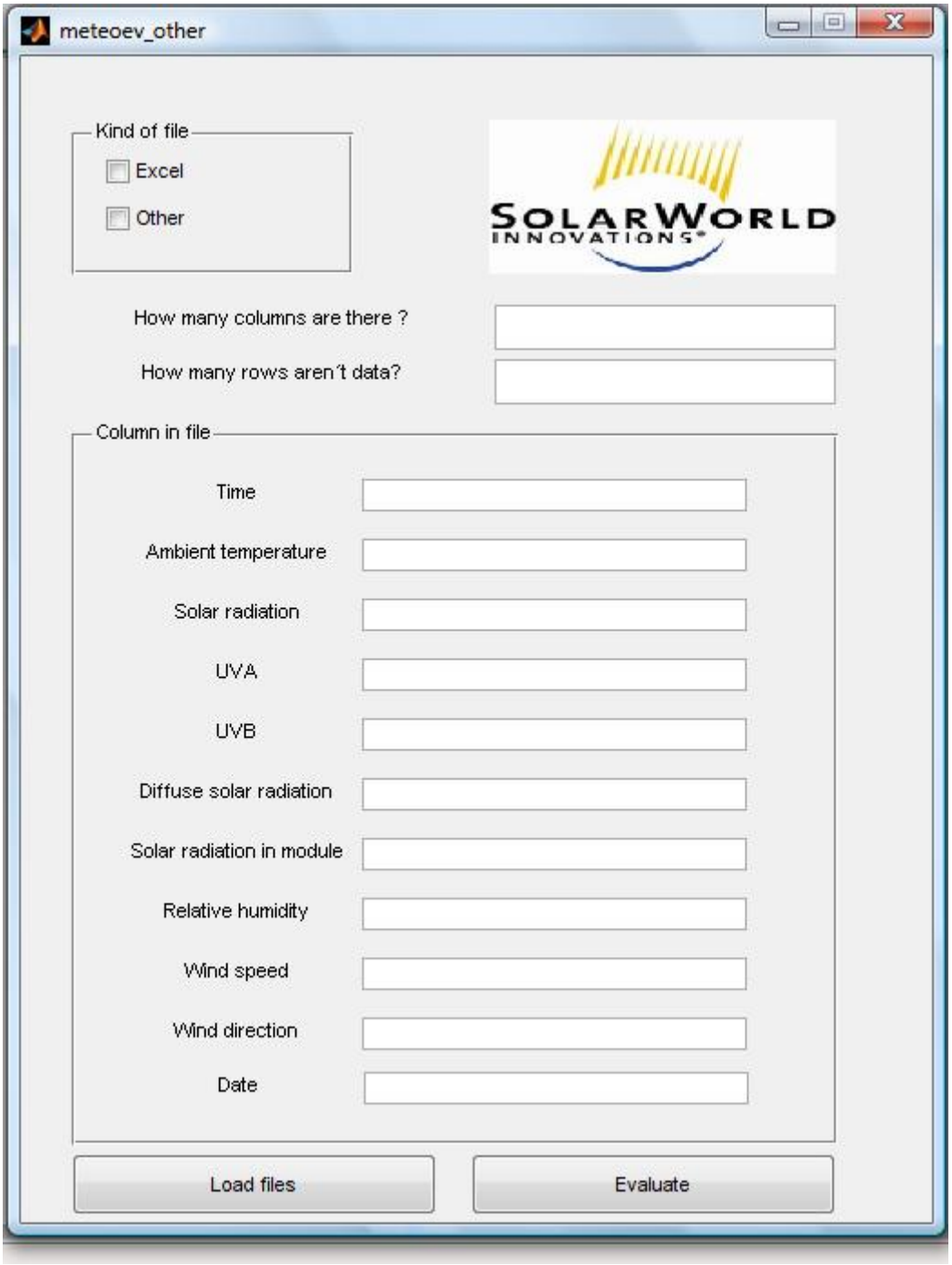
```
function varargout = compare_12_5(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compare_12_5_OpeningFcn, ...
                   'gui_OutputFcn',  @compare_12_5_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function compare_12_5_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global x_m_1 y_m_1 x_m_2 y_m_2 x_m_3 y_m_3 x_m_4 y_m_4 x_m_5 y_m_5 x_m_6
y_m_6 x_m_7 y_m_7 x_m_8 y_m_8 x_m_9 y_m_9 x_m_10 y_m_10 x_m_11 y_m_11
x_m_12 y_m_12
axes(handles.logo)
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen)
axis off
hoy=date;
set(handles.today,'String',hoy)


plot(handles.rep12, x_m_1, y_m_1, '-ro', x_m_2, y_m_2,'-co', x_m_3,
y_m_3,'-yo', x_m_4, y_m_4,'-go', x_m_5, y_m_5, '-mo', x_m_6, y_m_6,'-bo',
x_m_7, y_m_7,'-r*', x_m_8, y_m_8,'-c*', x_m_9, y_m_9,'-y*' ,x_m_10,
y_m_10,'-g*', x_m_11, y_m_11,'-m*', x_m_12, y_m_12,'-b*')

    % legend('Modul 1','Modul 2',2);
% % set(h,'Interpreter','none')
%'Modul 3','Modul 4','Modul 5','Modul 6','Modul 7','Modul 8','Modul
%9','Modul 10','Modul 11','Modul 12',1

function varargout = compare_12_5_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;



function today_Callback(hObject, eventdata, handles)

function today_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function savebutton_Callback(hObject, eventdata, handles)
global x y xtag ytag x_m_1 y_m_1 x_m_2 y_m_2 x_m_3 y_m_3 x_m_4 y_m_4
x_m_5 y_m_5 x_m_6 y_m_6 x_m_7 y_m_7 x_m_8 y_m_8 x_m_9 y_m_9 x_m_10 y_m_10
x_m_11 y_m_11 x_m_12 y_m_12
if x==0
    figure
    plot(x_m_1, y_m_1, '-ro', x_m_2, y_m_2,'-co', x_m_3, y_m_3,'-yo',
x_m_4, y_m_4,'-go', x_m_5, y_m_5, '-mo', x_m_6, y_m_6,'-bo', x_m_7,
y_m_7,'-r*', x_m_8, y_m_8,'-c*', x_m_9, y_m_9,'-y*' ,x_m_10, y_m_10,'-
g*', x_m_11, y_m_11,'-m*', x_m_12, y_m_12,'-b*')
    datetick('x','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
elseif y==0
    figure
    plot(x_m_1, y_m_1, '-ro', x_m_2, y_m_2,'-co', x_m_3, y_m_3,'-yo',
x_m_4, y_m_4,'-go', x_m_5, y_m_5, '-mo', x_m_6, y_m_6,'-bo', x_m_7,
y_m_7,'-r*', x_m_8, y_m_8,'-c*', x_m_9, y_m_9,'-y*' ,x_m_10, y_m_10,'-
g*', x_m_11, y_m_11,'-m*', x_m_12, y_m_12,'-b*')
    datetick('y','HH:MM:SS')
    xlabel(xtag)
    ylabel(ytag)
else
    figure(2)
    plot(x_m_1, y_m_1, '-ro', x_m_2, y_m_2,'-co', x_m_3, y_m_3,'-yo',
x_m_4, y_m_4,'-go', x_m_5, y_m_5, '-mo', x_m_6, y_m_6,'-bo', x_m_7,
y_m_7,'-r*', x_m_8, y_m_8,'-c*', x_m_9, y_m_9,'-y*' ,x_m_10, y_m_10,'-
g*', x_m_11, y_m_11,'-m*', x_m_12, y_m_12,'-b*')
    xlabel(xtag)
    ylabel(ytag)
end

function pushbutton2_Callback(hObject, eventdata, handles)
close compare_12_try2
close compare_12_2_try2
main_menu
```

F.6. Evaluation Meteo



```
function varargout = meteoevaluation(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @meteoevaluation_OpeningFcn, ...
                   'gui_OutputFcn',  @meteoevaluation_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function meteoevaluation_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;

function varargout = meteoevaluation_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function MSDS1400button_Callback(hObject, eventdata, handles)
global Meteomatrix  fid  Meteo_datos
handles.fileName = uigetfile('*.txt');
guidata(hObject, handles);
fileName = handles.fileName;
fid = fopen(fileName);
Meteo_text=textscan(fid,'%*[^\n]',2);
Meteo_datos=textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s
%s %s %s %s %s %s %s %s %s %s %s %s');
l=length(Meteo_datos{1,1});
Meteotxt=[];
Meteomatrix=[];
for ii=1:l
    Meteotxt(ii,1)=str2num(strrep(Meteo_datos{1,11}{ii,1},',','.'));
end
for i=1:l
for j=14:27
Meteotxt(i,j-12)=str2num(strrep(Meteo_datos{1,j}{i,1},',','.'));
end
end
fclose(fid);
Meteomatrix=Meteotxt;
meteoDS1400_1
function ODbutton_Callback(hObject, eventdata, handles)
meteoev_other
```


meteoev_other
Kind of file
Excel
Other
SOLARWORLD
INNOVATIONS
How many columns are there ?
How many rows aren´t data?
Column in file
Time
Ambient temperature
Solar radiation
UVA
UVB
Diffuse solar radiation
Solar radiation in module
Relative humidity
Wind speed
Wind direction
Date
Load files
Evaluate

```
function varargout = meteoev_other2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @meteoev_other2_OpeningFcn, ...
                   'gui_OutputFcn',  @meteoev_other2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function meteoev_other2_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global excelfile otherfile rowout datecol columnstotal coltime colTA
colMSR colDSR colSR colUVA colUVB colRH colWS colWD Meteomatrix Meteochar
fid lUVB lUVA lMSR lDSR lTA lSR lWS lWD lRH time TA UVA UVB SR MSR DSR RH
WS WD
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
time=[];
TA=[];
MSR=[];
DSR=[];
SR=[];
UVA=[];
UVB=[];
RH=[];
WS=[];
WD=[];
r_o=rowout;
if excelfile==1
    if coltime==0
        time=[];
    else
        time =Meteomatrix(:,coltime);
    end
    if colTA==0
        TA=[];
    else
        TA =Meteomatrix(:,colTA);
    end
    if colMSR==0
        MSR=[];
    else
        MSR =Meteomatrix(:,colMSR);
    end
    if colDSR==0
```

```
        DSR=[];
    else
        DSR =Meteomatrix(:,colDSR);
    end
    if colSR==0
        SR=[];
    else
        SR =Meteomatrix(:,colSR);
    end
    if colUVA==0
        UVA=[];
    else
        UVA =Meteomatrix(:,colUVA);
    end
    if colUVB==0
        UVB=[];
    else
        UVB =Meteomatrix(:,colUVB);
    end
    if colRH==0
        RH=[];
    else
        RH =Meteomatrix(:,colRH);
    end
    if colWS==0
        WS=[];
    else
        WS =Meteomatrix(:,colWS);
    end
    if colWD==0
        WD=[];
    else
        WD =Meteomatrix(:,colWD);
    end
elseif otherfile==1
    %read the whole file
    meteodata1= fscanf(fid, '%c');
    %vector with the positions of carry return
    eol_i=find(meteodata1==10);
    %l_i=length of the whole file
    l_i=length(meteodata1);
    %meteodata2 is a vector with all the data except the rows, which
don't
    %data
    meteodata2=meteodata1(eol_i(r_o):l_i);
    % change coma to point
    meteodata2=strrep(meteodata2,',','.');
    %length of new file
    l_f=length(meteodata2);
    %cc = vector, which saves the positions of tabs of meteodata2
    cc=find(meteodata2==9);
    %eol = vector, which saves the end of lines, carry returns, of
    %meteodata2
    eol=find(meteodata2==10);
    %arr = variable to count
    arr=0;
    %Initialize
```

```
ant=[];
% how many carry returns are there?
l_eol=length(eol);

%%%lo de quitar los 2 no estoy segura...
%%%l_eol???
Date_c=cell(l_eol,1);
Time_c=cell(l_eol,1);
anteol=l_eol-2;
date1=[];
date2=[];
time1=[];
time2=[];
Meteomatrix=[];
counteol=1;
countcc=1;
for i=1:anteol
    cont=1;
    for j=1:columnstotal
        if cont==1
            pal=meteodata2(eol(counteol):cc(countcc));
            %returns 1 when =NaN
            proof=isnan(str2double(pal));

            if proof==1
                % if str2double(pal)=NaN
                % if is date
                if cont==datecol
                    date1=datenum(pal,'dd.mm.yyyy');
                    date2=datestr(date1,'dd.mm.yyyy');
                    Date_c{i,1}=date2;
                    %if is hour
                elseif cont==coltime
                    time1=datenum(pal);
                    time2=datestr(time1,'HH:MM:SS');
                    Time_c{i,1}=time2;
                end
            else
                Meteomatrix(i,j)=str2double(pal);
            end
            proof=0;
            counteol=counteol+1;
            countcc=countcc+1;
            cont=cont+1;
        elseif cont==columnstotal

            pal=meteodata2(cc(countcc-1):eol(counteol));
            proof=isnan(str2double(pal));

            if proof==1
                % if str2double(pal)=NaN
                % if is date
                if cont==datecol
                    date1=datenum(pal,'dd.mm.yyyy');
                    date2=datestr(date1,'dd.mm.yyyy');
                    Date_c{i,1}=date2;
```

```
                %if is hour
            elseif cont==coltime
                time1=datenum(pal);
                time2=datestr(time1,'HH:MM:SS');
                Time_c{i,1}=time2;
            end
        else
            Meteomatrix(i,j)=str2double(pal);
        end
        proof=0;
    else

        pal=meteodata2(cc(countcc-1):cc(countcc));
        proof=isnan(str2double(pal));

        if proof==1
            % if str2double(pal)=NaN
            % if is date
            if cont==datecol
                date1=datenum(pal,'dd.mm.yyyy');
                date2=datestr(date1,'dd.mm.yyyy');
                Date_c{i,1}=date2;
                %if is hour
            elseif cont==coltime
                time1=datenum(pal);
                time2=datestr(time1,'HH:MM:SS');
                Time_c{i,1}=time2;
            end
        else
            Meteomatrix(i,j)=str2double(pal);
        end
        proof=0;
        countcc=countcc+1;
        cont=cont+1;
    end

  end
end
dayev=Date_c{1,1};
set(handles.dayofevalu,'String',dayev);
if coltime==0
    time=[];
else
    time=[];
    for ii=1:anteol
        time3=Time_c{ii,1};
        time(ii)=datenum(time3);
    end
end
if colTA==0
    TA=[];
else
    TA =Meteomatrix(:,colTA);
end
if colMSR==0
    MSR=[];
```

```
    else
        MSR =Meteomatrix(:,colMSR);
    end
    if colDSR==0
        DSR=[];
    else
        DSR =Meteomatrix(:,colDSR);
    end
    if colSR==0
        SR=[];
    else
        SR =Meteomatrix(:,colSR);
    end
    if colUVA==0
        UVA=[];
    else
        UVA =Meteomatrix(:,colUVA);
    end
    if colUVB==0
        UVB=[];
    else
        UVB =Meteomatrix(:,colUVB);
    end
    if colRH==0
        RH=[];
    else
        RH =Meteomatrix(:,colRH);
    end
    if colWS==0
        WS=[];
    else
        WS =Meteomatrix(:,colWS);
    end
    if colWD==0
        WD=[];
    else
        WD =Meteomatrix(:,colWD);
    end
end

% anteol=l_eol-2;
% MDM=zeros(anteol,14);
% for i=1:anteol
%     for j=1:14 %aquí tiene que ser hasta la longitud de c(:)
%         if c(j)~=0
%             caux=c(j)+arr;
%             ant=caux-1;
%         else
%             caux=0;
%         end
%         if caux==arr+1
%             pal=d_u(eol(i):cc(caux));
%             %fprintf('comienzo linea');
%             MDM(i,j)=str2num(pal);
%         elseif caux==n_c+arr
%             pal=d_u(cc(ant):eol(i+1));
%             %fprintf('fin de linea');
```

```
%             MDM(i,j)=str2num(pal);
%         elseif caux==0
%             %fprintf('fila que no tienen');
%             MDM(i,j)=9999;
%         else
%             pal=d_u(cc(ant):cc(caux));
%             %fprintf('entre columnas');
%             MDM(i,j)=str2num(pal);
%         end
%
%     end
%     arr=(arr)+(n_c)-1;
% end
ltime=length(time);
lTA=length(TA);
lUVA=length(UVA);
lUVB=length(UVB);
lSR=length(SR);
lMSR=length(MSR);
lDSR=length(DSR);
lWD=length(WD);
lWS=length(WS);
lRH=length(RH);
x=0;
y=0;
if ltime==0
    errordlg('There aren´t data for time','ERROR')
else
    if lTA==0
        subplot(2,3,1),plot(x,y)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Ambient temperature (°C)')
    else
        subplot(2,3,1),plot(time,TA)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Ambient temperature (°C)')
    end
    if lSR==0 && lUVA==0 && lUVB==0
        subplot(2,3,2),plot(x,y)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Global Solar Radiation, UVA, UVB  (W/m²)')
    elseif lUVA==0 && lUVB==0
        subplot(2,3,2),plot(time,SR)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Global Solar Radiation  (W/m²)')
    elseif lUVA==0
        subplot(2,3,2),plot(time,SR, time, UVB)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Global Solar Radiation  (W/m²)')
        h = legend('G','UVB',2);
        set(h,'Interpreter','none')
    elseif lUVB==0
```

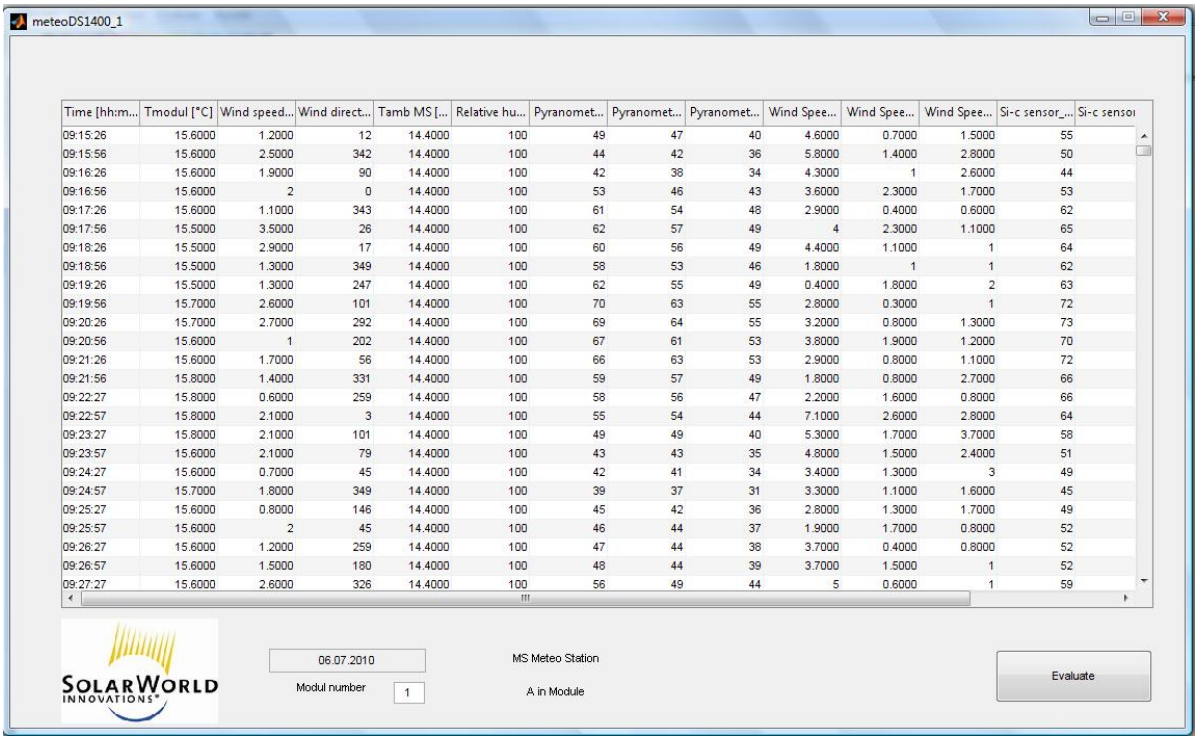
```
        subplot(2,3,2),plot(time,SR, time, UVA)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Global Solar Radiation  (W/m²)')
        h = legend('G','UVA',2);
        set(h,'Interpreter','none')
    elseif lSR==0
        subplot(2,3,2),plot( time, UVA, time, UVB)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('UVA, UVB(W/m²)')
        h = legend('UVA','UVB',2);
        set(h,'Interpreter','none')
    else
        subplot(2,3,2),plot(time,SR, time, UVA, time, UVB)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Global Solar Radiation  (W/m²)')
        h = legend('G','UVA','UVB',3);
        set(h,'Interpreter','none')
    end
    if lSR==0 && lMSR==0 && lDSR==0
        subplot(2,3,3),plot(x,y)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar Radiation  (W/m²)')

    elseif lMSR==0 && lDSR==0
        subplot(2,3,3),plot(time,SR)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar Radiation  (W/m²)')

    elseif lMSR==0
        subplot(2,3,3),plot(time,SR, time, DSR)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar Radiation  (W/m²)')
        j = legend('G','Diffuse solar radiation',2);
        set(j,'Interpreter','none')
    elseif lDSR==0
        subplot(2,3,3),plot(time,SR, time, MSR)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar Radiation  (W/m²)')
        j = legend('G','Diffuse solar radiation',2);
        set(j,'Interpreter','none')
    elseif lSR==0
        subplot(2,3,3),plot( time, MSR, time, DSR)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar radiation(W/m²)')
        j = legend('Solar radiation in module','Diffuse solar
radiation',2);
        set(j,'Interpreter','none')
    else
        subplot(2,3,3),plot(time,SR, time, MSR, time, DSR)
```

```
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Solar Radiation  (W/m²)')
        j = legend('G','Solar radiation module','Diffuse solar
radiation',3);
        set(j,'Interpreter','none')
    end
    if lRH==0
        subplot(2,3,4),plot(x,y)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Relative humidity (%)')
    else
        subplot(2,3,4),plot(time,RH)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Relative humidity (%)')
    end

    if lWS==0
        subplot(2,3,5),plot(x,y)
        xlabel('Wind  speed')

    else
        WD1=WD*pi/180;
        x1=[];
        y1=[];
        [x1,y1]=pol2cart(WD1,WS);
        subplot(2,3,5),compass(x1,y1)
        xlabel('Wind speed (m/s)')
    end
end

%%%%% Für die rose plot man braucht die winkeln in radians mira también
el
%%%%% compass

if lWD==0
     subplot(2,3,6),rose(x)
     xlabel ('Wind direction / Frequency')
else
    WD1=WD*pi/180;
    subplot(2,3,6),rose(WD1)
    xlabel ('Wind direction / Frequency')
end

function varargout = meteoev_other2_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

function firstbutton_Callback(hObject, eventdata, handles)
global time TA lTA x y
figure
x=0;
y=0;
if lTA==0
```

```
    plot(x,y)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Ambient temperature (°C)')
else
    plot(time,TA)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Ambient temperature (°C)')
end

function secondbutton_Callback(hObject, eventdata, handles)
global UVA UVB SR lUVA lUVB time x y lSR
figure
if lSR==0 && lUVA==0 && lUVB==0
    plot(x,y)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Global Solar Radiation, UVA, UVB  (W/m²)')
elseif lUVA==0 && lUVB==0
    plot(time,SR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Global Solar Radiation  (W/m²)')
elseif lUVA==0
    plot(time,SR, time, UVB)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Global Solar Radiation  (W/m²)')
    h = legend('G','UVB',2);
    set(h,'Interpreter','none')
elseif lUVB==0
    plot(time,SR, time, UVA)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Global Solar Radiation  (W/m²)')
    h = legend('G','UVA',2);
    set(h,'Interpreter','none')
elseif lSR==0
    plot( time, UVA, time, UVB)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('UVA, UVB(W/m²)')
    h = legend('UVA','UVB',2);
    set(h,'Interpreter','none')
else
    plot(time,SR, time, UVA, time, UVB)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Global Solar Radiation  (W/m²)')
    h = legend('G','UVA','UVB',3);
    set(h,'Interpreter','none')
end

function thirdbutton_Callback(hObject, eventdata, handles)
global MSR DSR SR lMSR lDSR time lSR x y
figure
```

```
if lSR==0 && lMSR==0 && lDSR==0
    plot(x,y)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar Radiation  (°C)')

elseif lMSR==0 && lDSR==0
    plot(time,SR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar Radiation  (W/m²)')

elseif lMSR==0
    plot(time,SR, time, DSR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar Radiation  (W/m²)')
    j = legend('G','Diffuse solar radiation',2);
    set(j,'Interpreter','none')
elseif lDSR==0
    plot(time,SR, time, MSR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar Radiation  (W/m²)')
    j = legend('G','Diffuse solar radiation',2);
    set(j,'Interpreter','none')
elseif lSR==0
    plot( time, MSR, time, DSR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar radiation(W/m²)')
    j = legend('Solar radiation in module','Diffuse solar radiation',2);
    set(j,'Interpreter','none')
else
    plot(time,SR, time, MSR, time, DSR)
    datetick('x','HH:MM:SS')
    xlabel('Time (HH:MM)')
    ylabel('Solar Radiation  (W/m²)')
    j = legend('G','Solar radiation module','Diffuse solar radiation',3);
    set(j,'Interpreter','none')
end

function fourthbutton_Callback(hObject, eventdata, handles)
global time RH lRH x y
figure
if lRH==0
        plot(x,y)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Relative humidity (%)')
    else
        plot(time,RH)
        datetick('x','HH:MM:SS')
        xlabel('Time (HH:MM)')
        ylabel('Relative humidity (%)')
    end
```

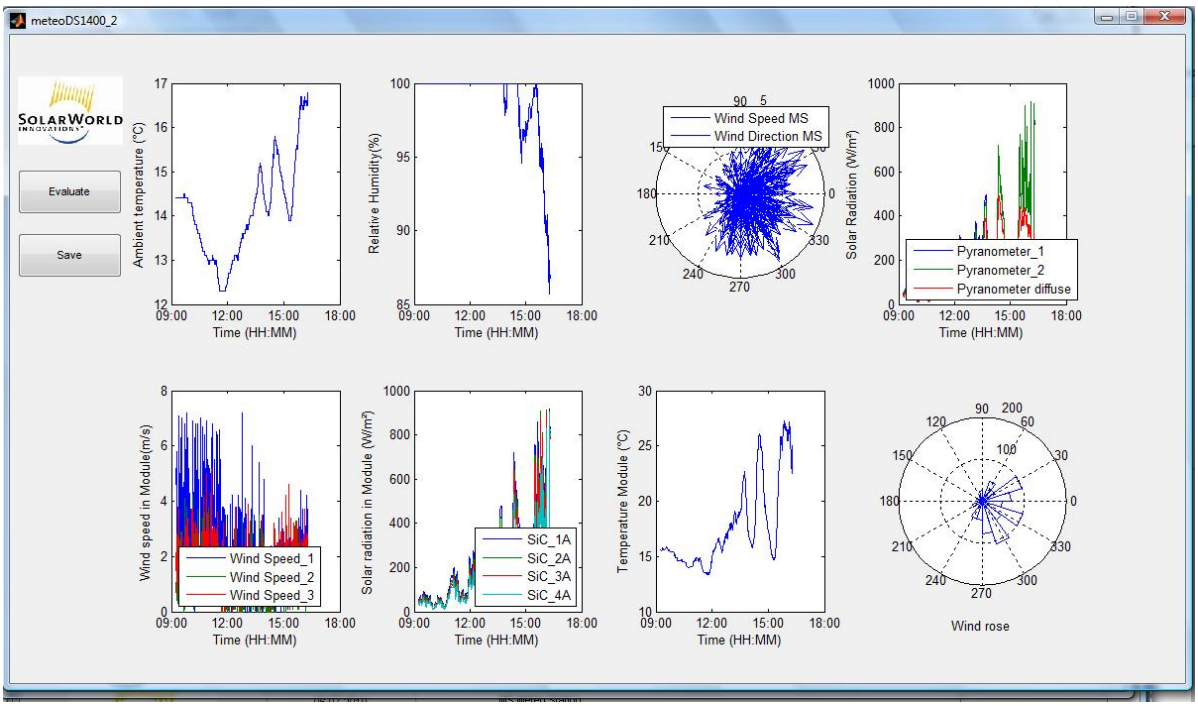
```
function fifthbutton_Callback(hObject, eventdata, handles)
global time WS WD lWS lWD x y
figure
if lWS==0
        plot(x,y)

        ylabel('Wind  speed')

    else
        WD1=WD*pi/180;
        x1=[];
        y1=[];
        [x1,y1]=pol2cart(WD1,WS);
        compass(x1,y1)
        xlabel('Wind direction (°)')
        ylabel('Wind speed (m/s)')
    end

function sexthbutton_Callback(hObject, eventdata, handles)
global WD lWD x
figure
if lWD==0
    rose(x)
     xlabel ('Wind direction / Frequency')
else
    WD1=WD*pi/180;
    rose(WD1)
    xlabel ('Wind direction / Frequency')
end

function saveworkspacebutton_Callback(hObject, eventdata, handles)
global time TA SR UVA UVB DSR MSR RH WS WD
%save datos.mat horas horasch G Tmod Umpp Impp Pmpp Uoc Isc FF Pmppstc
Uocstc Iscstc FFstc Rs Rp
% save ..\data\datos.mat horas horasch G Tmod Umpp Impp Pmpp Uoc Isc FF
Pmppstc Uocstc Iscstc FFstc Rs Rp
uisave({'time','TA','SR','UVA','UVB','DSR','MSR','RH','WS','WD'},'meteoda
ta')

function pushbutton8_Callback(hObject, eventdata, handles)
close meteoevaluation
close meteoev_other
main_menu
```


meteoDS1400_1
Time [hh:m...
Tmodul [°C]
Wind speed...
Wind direct...
Tamb MS [...
Relative hu...
Pyranomet...
Pyranomet...
Pyranomet...
Wind Spee...
Wind Spee...
Wind Spee...
Si-c sensor_...
Si-c sensor
09:15:26 15.6000 1.2000 12 14.4000 100 49 47 40 4.6000 0.7000 1.5000 55
09:15:56 15.6000 2.5000 342 14.4000 100 44 42 36 5.8000 1.4000 2.8000 50
09:16:26 15.6000 1.9000 90 14.4000 100 42 38 34 4.3000 1 2.6000 44
09:16:56 15.6000 2 0 14.4000 100 53 46 43 3.6000 2.3000 1.7000 53
09:17:26 15.6000 1.1000 343 14.4000 100 61 54 48 2.9000 0.4000 0.6000 62
09:17:56 15.5000 3.5000 26 14.4000 100 62 57 49 4 2.3000 1.1000 65
09:18:26 15.5000 2.9000 17 14.4000 100 60 56 49 4.4000 1.1000 1 64
09:18:56 15.5000 1.3000 349 14.4000 100 58 53 46 1.8000 1 1 62
09:19:26 15.5000 1.3000 247 14.4000 100 62 55 49 0.4000 1.8000 2 63
09:19:56 15.7000 2.6000 101 14.4000 100 70 63 55 2.8000 0.3000 1 72
09:20:26 15.7000 2.7000 292 14.4000 100 69 64 55 3.2000 0.8000 1.3000 73
09:20:56 15.6000 1 202 14.4000 100 67 61 53 3.8000 1.9000 1.2000 70
09:21:26 15.6000 1.7000 56 14.4000 100 66 63 53 2.9000 0.8000 1.1000 72
09:21:56 15.8000 1.4000 331 14.4000 100 59 57 49 1.8000 0.8000 2.7000 66
09:22:27 15.8000 0.6000 259 14.4000 100 58 56 47 2.2000 1.6000 0.8000 66
09:22:57 15.8000 2.1000 3 14.4000 100 55 54 44 7.1000 2.6000 2.8000 64
09:23:27 15.8000 2.1000 101 14.4000 100 49 49 40 5.3000 1.7000 3.7000 58
09:23:57 15.6000 2.1000 79 14.4000 100 43 43 35 4.8000 1.5000 2.4000 51
09:24:27 15.6000 0.7000 45 14.4000 100 42 41 34 3.4000 1.3000 3 49
09:24:57 15.7000 1.8000 349 14.4000 100 39 37 31 3.3000 1.1000 1.6000 45
09:25:27 15.6000 0.8000 146 14.4000 100 45 42 36 2.8000 1.3000 1.7000 49
09:25:57 15.6000 2 45 14.4000 100 46 44 37 1.9000 1.7000 0.8000 52
09:26:27 15.6000 1.2000 259 14.4000 100 47 44 38 3.7000 0.4000 0.8000 52
09:26:57 15.6000 1.5000 180 14.4000 100 48 44 39 3.7000 1.5000 1 52
09:27:27 15.6000 2.6000 326 14.4000 100 56 49 44 5 0.6000 1 59
SOLARWORLD INNOVATIONS
06.07.2010
Modul number
1
MS Meteo Station
A in Module
Evaluate

```
function varargout = meteoDS1400_1(varargin)
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @meteoDS1400_1_OpeningFcn, ...
                   'gui_OutputFcn',  @meteoDS1400_1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function meteoDS1400_1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global Meteohours Meteomatrix Meteotime SwindMS DwindMS TambMS rHMS
PyrMS1 PyrMS2 PyrdifMS Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A
SiC_4A TModul
global Meteomatrix  Meteo_datos Meteohours horai horaf fechaMeteo
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
fechaMeteo=Meteo_datos{1,2}{1,1};
number=Meteo_datos{1,1}{1,1};
set(handles.date, 'String', fechaMeteo)
set(handles.nummodul,'String',number)

l_out=length(Meteomatrix(:,1));
Meteohours=[];
Hour_c=cell(l_out,1);
    for i=1:l_out
        hour1=Meteo_datos{1,3}{i,1};
        Meteohours(i)=datenum(hour1);
        Hour_c{i,1}=datestr(Meteohours(i),'HH:MM:SS');
    end
    horai=Meteohours(1);
    horaf=Meteohours(l_out);

if l_out==0
errordlg('You must another file choose, there are not enough data','Bad
Input','modal')
uicontrol(hObject)
return
end

num_elem=cell(l_out,16);
for i2=1:l_out
    for j=2:16
        num_elem{i2,j}=Meteomatrix(i2,j-1);
    end
```

```
    num_elem{i2,1}=Hour_c{i2,1};
end
set(handles.meteotable,'Data',num_elem)

Meteotime=[];
TModul=[];
SwindMS=[];
DwindMS=[];
TambMS=[];
rHMS=[];
PyrMS1=[];
PyrMS2=[];
PyrdifMS=[];
Swind_1A=[];
Swind_2A=[];
Swind_3A=[];
SiC_1A=[];
SiC_2A=[];
SiC_3A=[];
SiC_4A=[];


Meteotime=Meteohours;
TModul=Meteomatrix(:,1);
SwindMS=Meteomatrix(:,2);
DwindMS=Meteomatrix(:,3);
TambMS=Meteomatrix(:,4);
rHMS=Meteomatrix(:,5);
PyrMS1=Meteomatrix(:,6);
PyrMS2=Meteomatrix(:,7);
PyrdifMS=Meteomatrix(:,8);
Swind_1A=Meteomatrix(:,9);
Swind_2A=Meteomatrix(:,10);
Swind_3A=Meteomatrix(:,11);
SiC_1A=Meteomatrix(:,12);
SiC_2A=Meteomatrix(:,13);
SiC_3A=Meteomatrix(:,14);
SiC_4A=Meteomatrix(:,15);

function varargout = meteoDS1400_1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function evaluate_Callback(hObject, eventdata, handles)
meteoDS1400_2

function meteotable_CellEditCallback(hObject, eventdata, handles)
% % global MDM titMDM titchMDM fechaMDM
% % l_out=length(MDM(:,1));
% % num_elem=cell(l_out,15);
% %
% % for i=1:20
% %     num_elem{i,1}=titchMDM(i,1);
% %     for j=2:15
% %         num_elem{i,j}=MDM(i,j-1);
% %     end
% % end
```

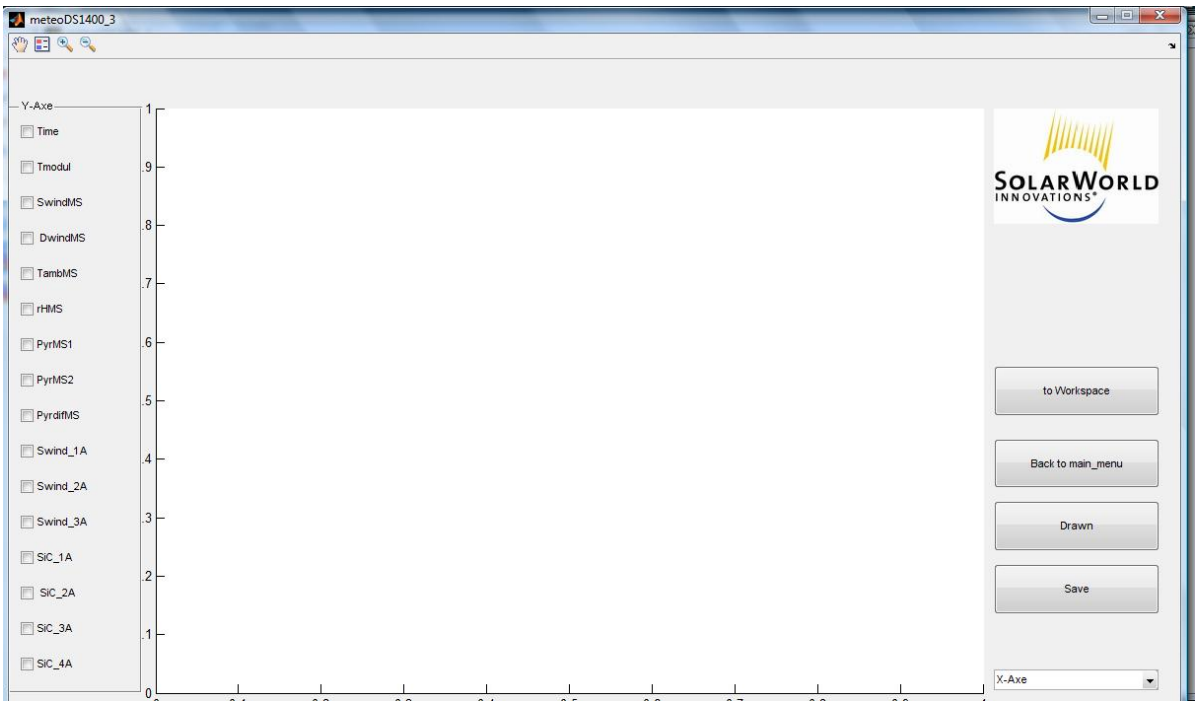
```
% % set(handles.tabla,'Data', num_elem)


% % --- Executes on button press in savebutton.
% function savebutton_Callback(hObject, eventdata, handles)
% % hObject    handle to savebutton (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
%


function nummodul_Callback(hObject, eventdata, handles)

function nummodul_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

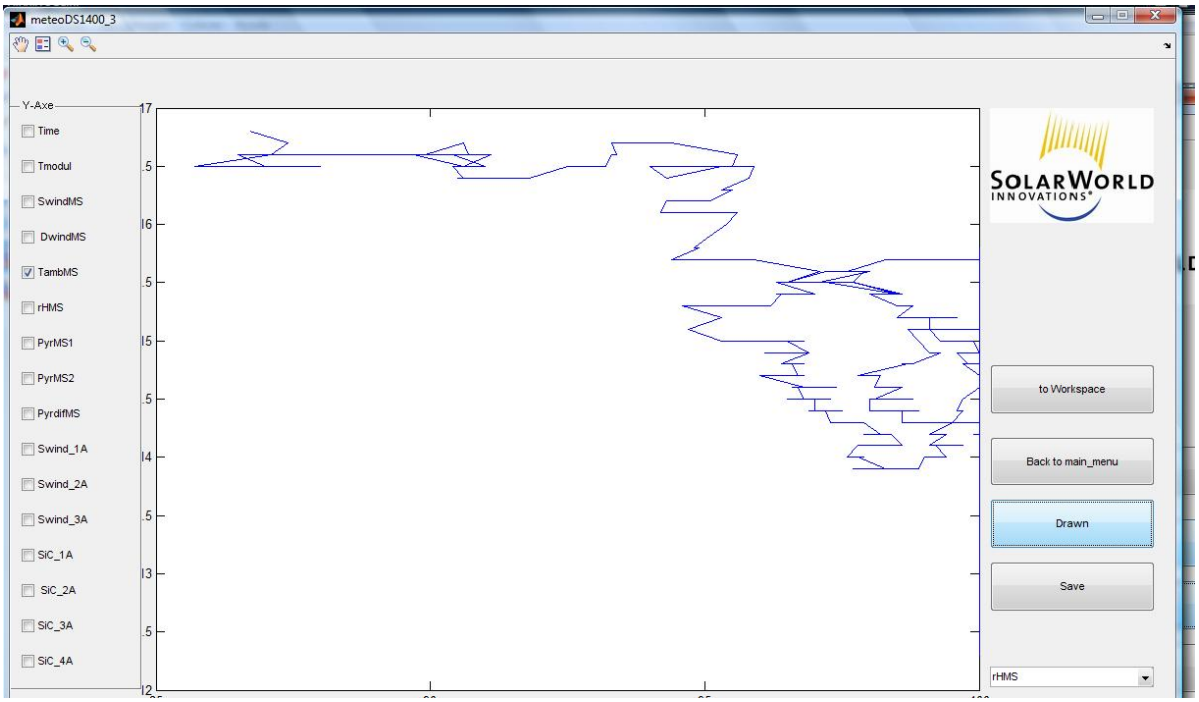

meteoDS1400_2
SOLARWORLD
Evaluate
Save
Ambient temperature (°C)
Time (HH:MM)
Relative Humidity(%)
Wind Speed MS
Wind Direction MS
Solar Radiation (W/m²)
Pyranometer_1
Pyranometer_2
Pyranometer diffuse
Wind speed in Module(m/s)
Wind Speed_1
Wind Speed_2
Wind Speed_3
Solar radiation in Module (W/m²)
SiC_1A
SiC_2A
SiC_3A
SiC_4A
Temperature Module (°C)
Wind rose

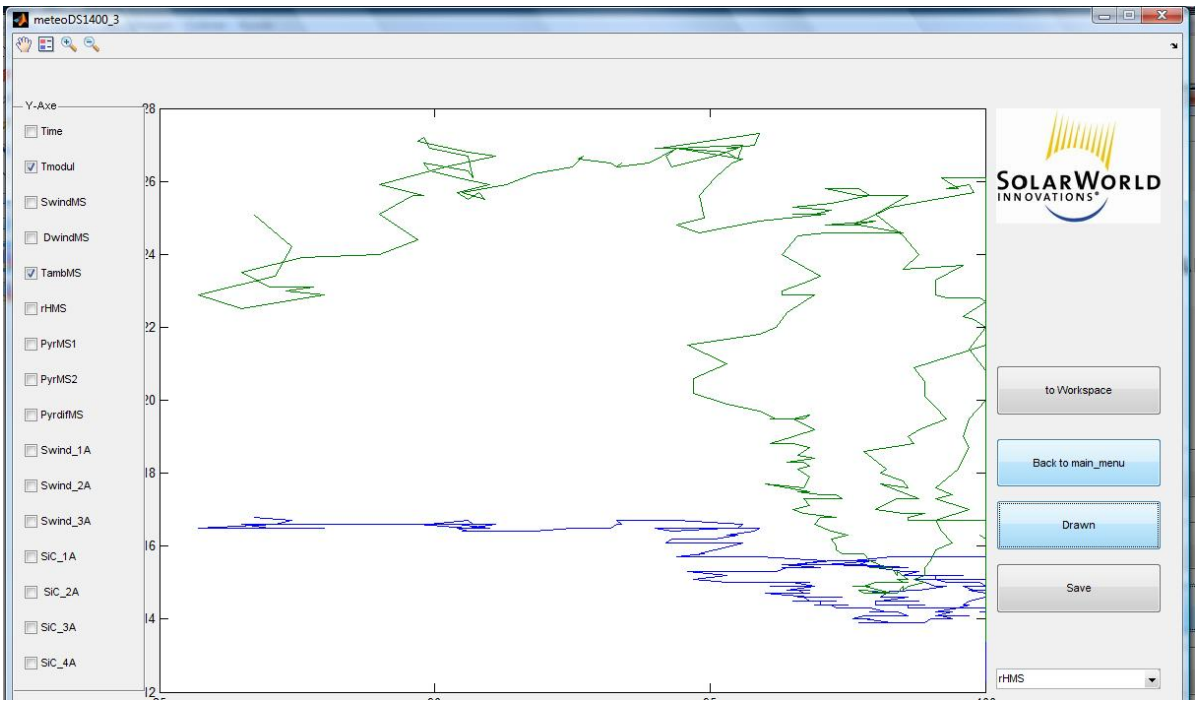
```
function varargout = meteoDS1400_2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @meteoDS1400_2_OpeningFcn, ...
                   'gui_OutputFcn',  @meteoDS1400_2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function meteoDS1400_2_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global Meteohours Meteomatrix Meteotime SwindMS DwindMS TambMS rHMS
PyrMS1 PyrMS2 PyrdifMS Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A
SiC_4A TModul
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
subplot(2,4,1),plot(Meteohours,TambMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Ambient temperature (°C)')

%%%%rH/t
subplot(2,4,2),plot(Meteohours,rHMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Relative Humidity(%)')

%%%%Wind speed and wind direction/t
x1=[];
y1=[];
WD=DwindMS*pi/180;
[x1,y1]=pol2cart(WD,SwindMS);
subplot(2,4,3),compass(x1,y1);
h = legend('Wind Speed MS','Wind Direction MS',2);
set(h,'Interpreter','none')

%%%%Pyr1 pyr2 pyrdifuse/t PyrMS1 PyrMS2 PyrdifMS
subplot(2,4,4),plot(Meteohours,PyrMS1,Meteohours,PyrMS2,
Meteohours,PyrdifMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Solar Radiation (W/m²)')
i = legend('Pyranometer_1','Pyranometer_2','Pyranometer diffuse',3);
set(i,'Interpreter','none')
```

```
%%%% Swind_1A Swind_2A Swind_3A/t
subplot(2,4,5),plot(Meteohours,Swind_1A,Meteohours,Swind_2A,
Meteohours,Swind_3A)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Wind speed in Module(m/s)')
j = legend('Wind Speed_1','Wind Speed_2','Wind Speed_3',3);
set(j,'Interpreter','none')

%%%% SiC_1A SiC_2A SiC_3A SiC_4A/t
subplot(2,4,6),plot(Meteohours,SiC_1A,Meteohours,SiC_2A,
Meteohours,SiC_3A, Meteohours,SiC_4A)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Solar radiation in Module (W/m²)')
k = legend('SiC_1A','SiC_2A','SiC_3A','SiC_4A',4);
set(k,'Interpreter','none')

%%%% Tmodul/t
subplot(2,4,7),plot(Meteohours,TModul)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Temperature Module (°C)')

%%%% Wind rose
WD=DwindMS*pi/180;
subplot(2,4,8),rose(WD)
xlabel('Wind rose')

function varargout = meteoDS1400_2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function evaluatebutton_Callback(hObject, eventdata, handles)
meteoDS1400_3

function savebutton_Callback(hObject, eventdata, handles)
global Meteohours Meteomatrix Meteotime SwindMS DwindMS TambMS rHMS
PyrMS1 PyrMS2 PyrdifMS Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A
SiC_4A TModul
figure
subplot(2,4,1),plot(Meteohours,TambMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Ambient temperature (°C)')

%%%%rH/t
subplot(2,4,2),plot(Meteohours,rHMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Relative Humidity(%)')

%%%%Wind speed and wind direction/t
%%%%Wind speed and wind direction/t
x1=[];
```

```
y1=[];
WD=DwindMS*pi/180;
[x1,y1]=pol2cart(WD,SwindMS);
subplot(2,4,3),compass(x1,y1);
h = legend('Wind Speed MS','Wind Direction MS',2);
set(h,'Interpreter','none')

%%%%Pyr1 pyr2 pyrdifuse/t PyrMS1 PyrMS2 PyrdifMS
subplot(2,4,4),plot(Meteohours,PyrMS1,Meteohours,PyrMS2,
Meteohours,PyrdifMS)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Solar Radiation (W/m²)')
i = legend('Pyranometer_1','Pyranometer_2','Pyranometer diffuse',3);
set(i,'Interpreter','none')

%%%% Swind_1A Swind_2A Swind_3A/t
subplot(2,4,5),plot(Meteohours,Swind_1A,Meteohours,Swind_2A,
Meteohours,Swind_3A)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Wind speed in Module(m/s)')
j = legend('Wind Speed_1','Wind Speed_2','Wind Speed_3',3);
set(j,'Interpreter','none')

%%%% SiC_1A SiC_2A SiC_3A SiC_4A/t
subplot(2,4,6),plot(Meteohours,SiC_1A,Meteohours,SiC_2A,
Meteohours,SiC_3A, Meteohours,SiC_4A)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Solar radiation in Module (W/m²)')
k = legend('SiC_1A','SiC_2A','SiC_3A','SiC_4A',4);
set(k,'Interpreter','none')

%%%% Tmodul/t
subplot(2,4,7),plot(Meteohours,TModul)
datetick('x','HH:MM')
xlabel('Time (HH:MM)')
ylabel('Temperature Module (°C)')


%%%% Wind rose
WD=DwindMS*pi/180;
subplot(2,4,8),rose(WD)
xlabel('Wind rose')
```


meteoDS1400_3
Y-Axe
Time
Tmodul
SwindMS
DwindMS
TambMS
rHMS
PyrMS1
PyrMS2
PyrdifMS
Swind_1A
Swind_2A
Swind_3A
SiC_1A
SiC_2A
SiC_3A
SiC_4A
SOLARWORLD
INNOVATIONS
to Workspace
Back to main_menu
Drawn
Save
X-Axe


meteoDS1400_3
Y-Axe
Time
Tmodul
SwindMS
DwindMS
TambMS
rHMS
PyrMS1
PyrMS2
PyrdifMS
Swind_1A
Swind_2A
Swind_3A
SiC_1A
SiC_2A
SiC_3A
SiC_4A
SOLARWORLD
INNOVATIONS
to Workspace
Back to main_menu
Drawn
Save
rHMS


meteoDS1400_3
Y-Axe
Time
Tmodul
SwindMS
DwindMS
TambMS
rHMS
PyrMS1
PyrMS2
PyrdiffMS
Swind_1A
Swind_2A
Swind_3A
SiC_1A
SiC_2A
SiC_3A
SiC_4A
SOLARWORLD
INNOVATIONS
to Workspace
Back to main_menu
Drawn
Save
rHMS

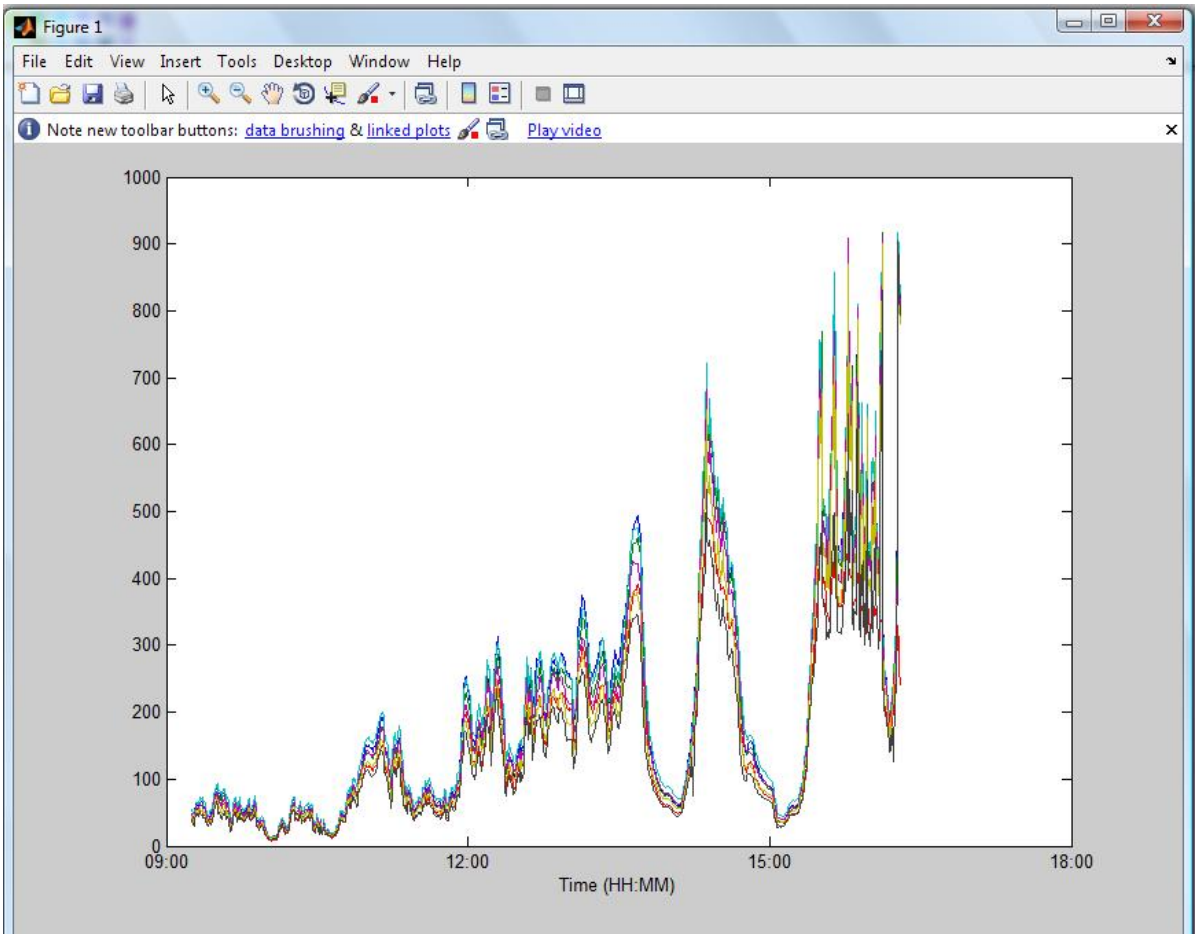


Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1000
900
800
700
600
500
400
300
200
100
0
09:00
12:00
15:00
18:00
Time (HH:MM)

```
function varargout = meteoDS1400_3(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @meteoDS1400_3_OpeningFcn, ...
                   'gui_OutputFcn',  @meteoDS1400_3_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function meteoDS1400_3_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global ytime ySwindMS yDwindMS yTambMS yrHMS yPyrMS1 yPyrMS2 yPyrdifMS
ySwind_1A ySwind_2A ySwind_3A ySiC_1A ySiC_2A ySiC_3A ySiC_4A yTModul
global l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 x lx
axes(handles.logo);
handles.imagen=imread('SWIN Navi.jpg');
imagesc(handles.imagen);
axis off;
ytime=[];
ySwindMS=[];
yDwindMS=[];
yTambMS=[];
yrHMS=[];
yPyrMS1=[];
yPyrMS2=[];
yPyrdifMS=[];
ySwind_1A=[];
ySwind_2A=[];
ySwind_3A=[];
ySiC_1A=[];
ySiC_2A=[];
ySiC_3A=[];
ySiC_4A=[];
yTModul=[];
x=[];
lx=0;
l1=0;
l2=0;
l3=0;
l4=0;
l5=0;
l6=0;
l7=0;
l8=0;
l9=0;
l10=0;
l11=0;
```

```
l12=0;
l13=0;
l14=0;
l15=0;
l16=0;

function varargout = meteoDS1400_3_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function popXaxe_Callback(hObject, eventdata, handles)
global Meteohours SwindMS DwindMS TambMS rHMS PyrMS1 PyrMS2 PyrdifMS
Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A SiC_4A TModul
global x lx
strx = get(hObject, 'String');
valx = get(hObject,'Value');
switch strx{valx};
    case 'Time'
        x=Meteohours;
        lx=1;
    case 'Tmodul'
        x=TModul;
    case 'SwindMS'
        x=SwindMS;
    case 'DwindMS'
        x=DwindMS;
    case 'TambMS'
        x=TambMS;
    case 'rHMS'
        x=rHMS;
    case 'PyrMS1'
        x=PyrMS1;
    case 'PyrMS2'
        x=PyrMS2;
    case 'PyrdifMS'
        x=PyrdifMS;
    case 'Swind_1A'
        x=Swind_1A;
    case 'Swind_2A'
        x=Swind_2A;
    case 'Swind_3A'
        x=Swind_3A;
    case 'SiC_1A'
        x=SiC_1A;
    case 'SiC_2A'
        x=SiC_2A;
    case 'SiC_3A'
        x=SiC_3A;
    case 'SiC_4A'
        x=SiC_4A;
end

function popXaxe_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Timebox_Callback(hObject, eventdata, handles)
global l1
if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l1=1;
else
% Checkbox is not checked-take approriate action
l1=0;
end

function SwindMSbox_Callback(hObject, eventdata, handles)
global l2

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l2=1;
else
% Checkbox is not checked-take approriate action
l2=0;
end

function DwindMSbox_Callback(hObject, eventdata, handles)
global l3

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l3=1;
else
% Checkbox is not checked-take approriate action
l3=0;
end

function TambMSbox_Callback(hObject, eventdata, handles)
global l4

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l4=1;
else
% Checkbox is not checked-take approriate action
l4=0;

end

function rHMSbox_Callback(hObject, eventdata, handles)
global l5

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l5=1;
else
% Checkbox is not checked-take approriate action
l5=0;
end
```

```
function PyrMS1box_Callback(hObject, eventdata, handles)
global l6

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l6=1;

else
% Checkbox is not checked-take approriate action
l6=0;

end

function PyrMS2box_Callback(hObject, eventdata, handles)
global  l7

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action

l7=1;
else
% Checkbox is not checked-take approriate action
l7=0;

end

function PyrdifMSbox_Callback(hObject, eventdata, handles)
global l8

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l8=1;
else
    l8=0;

end

function Swind_1Abox_Callback(hObject, eventdata, handles)
global  l9

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l9=1;

else
    l9=0;
end

function Swind_2Abox_Callback(hObject, eventdata, handles)
global l10

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l10=1;
```

```
else
% Checkbox is not checked-take approriate action
l10=0;
end

function Swind_3Abox_Callback(hObject, eventdata, handles)
global l11
if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l11=1;

else
    l11=0;
end

function SiC_1Abox_Callback(hObject, eventdata, handles)
global  l12

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l12=1;

else
% Checkbox is not checked-take approriate action
l12=0;
end

function SiC_2Abox_Callback(hObject, eventdata, handles)
global l13

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l13=1;
else
    l13=0;
% Checkbox is not checked-take approriate action
end

function SiC_3Abox_Callback(hObject, eventdata, handles)
global  l14

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l14=1;
else
    l14=0;
% Checkbox is not checked-take approriate action
end

function SiC_4Abox_Callback(hObject, eventdata, handles)
global l15

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
```

```
l15=1;
else
    l15=0;
% Checkbox is not checked-take approriate action
end

function Tmodulbox_Callback(hObject, eventdata, handles)
global l16

if (get(hObject,'Value') == get(hObject,'Max'))
% Checkbox is checked-take approriate action
l16=1;
else
% Checkbox is not checked-take approriate action
l16=0;
end

function savebutton_Callback(hObject, eventdata, handles)
global ytime ySwindMS yDwindMS yTambMS yrHMS yPyrMS1 yPyrMS2 yPyrdifMS
ySwind_1A ySwind_2A ySwind_3A ySiC_1A ySiC_2A ySiC_3A ySiC_4A yTModul
global lx xtime xSwindMS xDwindMS xTambMS xrHMS xPyrMS1 xPyrMS2 xPyrdifMS
xSwind_1A xSwind_2A xSwind_3A xSiC_1A xSiC_2A xSiC_3A xSiC_4A xTModul
rgb = getframe(handles.graphic);
figure
plot(xtime,ytime,xSwindMS,ySwindMS,xDwindMS, yDwindMS,xTambMS,
yTambMS,xrHMS, yrHMS,xPyrMS1, yPyrMS1,xPyrMS2, yPyrMS2,xPyrdifMS,
yPyrdifMS,xSwind_1A, ySwind_1A,xSwind_2A, ySwind_2A,xSwind_3A, ySwind_3A
,xSiC_1A, ySiC_1A,xSiC_2A, ySiC_2A ,xSiC_3A,ySiC_3A,xSiC_4A, ySiC_4A
,xTModul,yTModul )

if lx==1
    datetick('x','HH:MM')
    xlabel('Time (HH:MM)')
end

function drawnbutton_Callback(hObject, eventdata, handles)
global x ytime ySwindMS yDwindMS yTambMS yrHMS yPyrMS1 yPyrMS2 yPyrdifMS
ySwind_1A ySwind_2A ySwind_3A ySiC_1A ySiC_2A ySiC_3A ySiC_4A yTModul
global xtime xSwindMS xDwindMS xTambMS xrHMS xPyrMS1 xPyrMS2 xPyrdifMS
xSwind_1A xSwind_2A xSwind_3A xSiC_1A xSiC_2A xSiC_3A xSiC_4A xTModul
global Meteohours SwindMS DwindMS TambMS rHMS PyrMS1 PyrMS2 PyrdifMS
Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A SiC_4A TModul
global l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16

if l1==1
    ytime=Meteohours;
    xtime=x;
else
    ytime=[];
    xtime=[];
end
if l2==1
    ySwindMS =SwindMS;
    xSwindMS= x;
else
    ySwindMS=[];
```

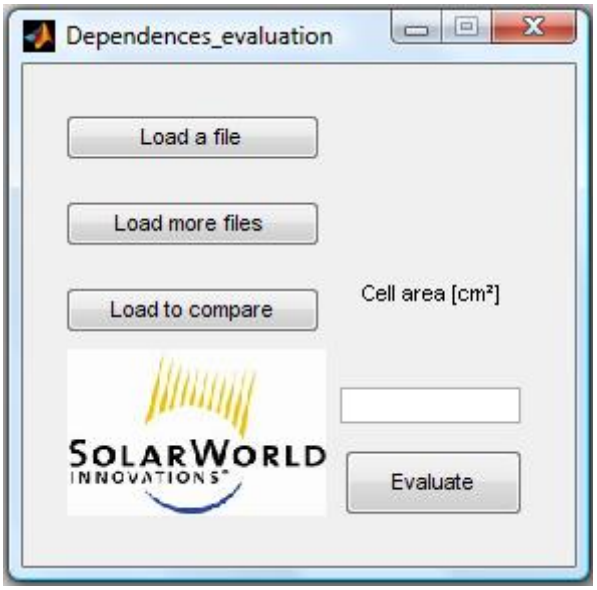
```
    xSwindMS=[];
end
if l3==1
    yDwindMS =DwindMS;
    xDwindMS =x;
else
    yDwindMS=[];
    xDwindMS=[];
end
if l4==1
    yTambMS =TambMS;
    xTambMS=x;
else
    yTambMS=[];
    xTambMS=[];
end
if l5==1
    yrHMS =rHMS;
    xrHMS=x;
else

    yrHMS=[];
    xrHMS=[];
end
if l6==1
    yPyrMS1 =PyrMS1;
    xPyrMS1 =x;
else
    yPyrMS1=[];
    xPyrMS1 =[];
end
if l7==1
    yPyrMS2 =PyrMS2;
    xPyrMS2=x;

else
    yPyrMS2=[];
    xPyrMS2=[];
end
if l8==1
    yPyrdifMS =PyrdifMS;
    xPyrdifMS=x;
else
    yPyrdifMS=[];
    xPyrdifMS=[];
end
if l9==1
    ySwind_1A =Swind_1A;
    xSwind_1A=x;
else
    ySwind_1A=[];
    xSwind_1A=[];
end
if l10==1
    ySwind_2A =Swind_2A;
    xSwind_2A= x;
else
```

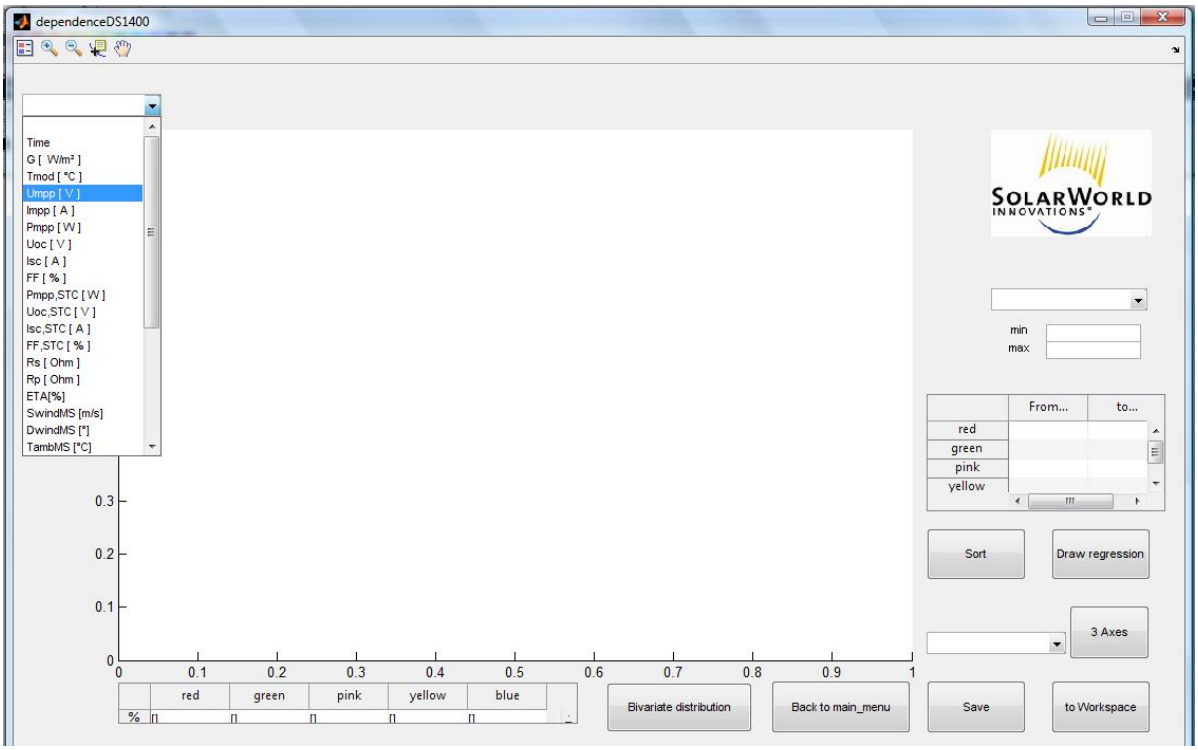
```
    ySwind_2A=[];
    xSwind_2A=[];
end
if l11==1
    ySwind_3A =Swind_3A;
    xSwind_3A=x;
else
    ySwind_3A=[];
    xSwind_3A=[];
end
if l12==1
    ySiC_1A =SiC_1A;
    xSiC_1A=x;
else
    ySiC_1A=[];
    xSiC_1A=[];
end
if l13==1
    ySiC_2A =SiC_2A;
    xSiC_2A=x;
else
    ySiC_2A=[];
    xSiC_2A=[];
end
if l14==1
    ySiC_3A =SiC_3A;
    xSiC_3A=x;
else
    ySiC_3A=[];
    xSiC_3A=[];
end
if l15==1
    ySiC_4A =SiC_4A;
    xSiC_4A=x;
else
    ySiC_4A=[];
    xSiC_4A=[];
end
if l16==1
    yTModul =TModul;
    xTModul=x;

else

    yTModul=[];
    xTModul=[];
end

plot(handles.graphic,xtime,ytime,xSwindMS,ySwindMS,xDwindMS,
yDwindMS,xTambMS, yTambMS,xrHMS, yrHMS,xPyrMS1, yPyrMS1,xPyrMS2,
yPyrMS2,xPyrdifMS, yPyrdifMS,xSwind_1A, ySwind_1A,xSwind_2A,
ySwind_2A,xSwind_3A, ySwind_3A ,xSiC_1A, ySiC_1A,xSiC_2A, ySiC_2A
,xSiC_3A,ySiC_3A,xSiC_4A, ySiC_4A ,xTModul,yTModul )


% --- Executes on button press in pushbutton3.
```

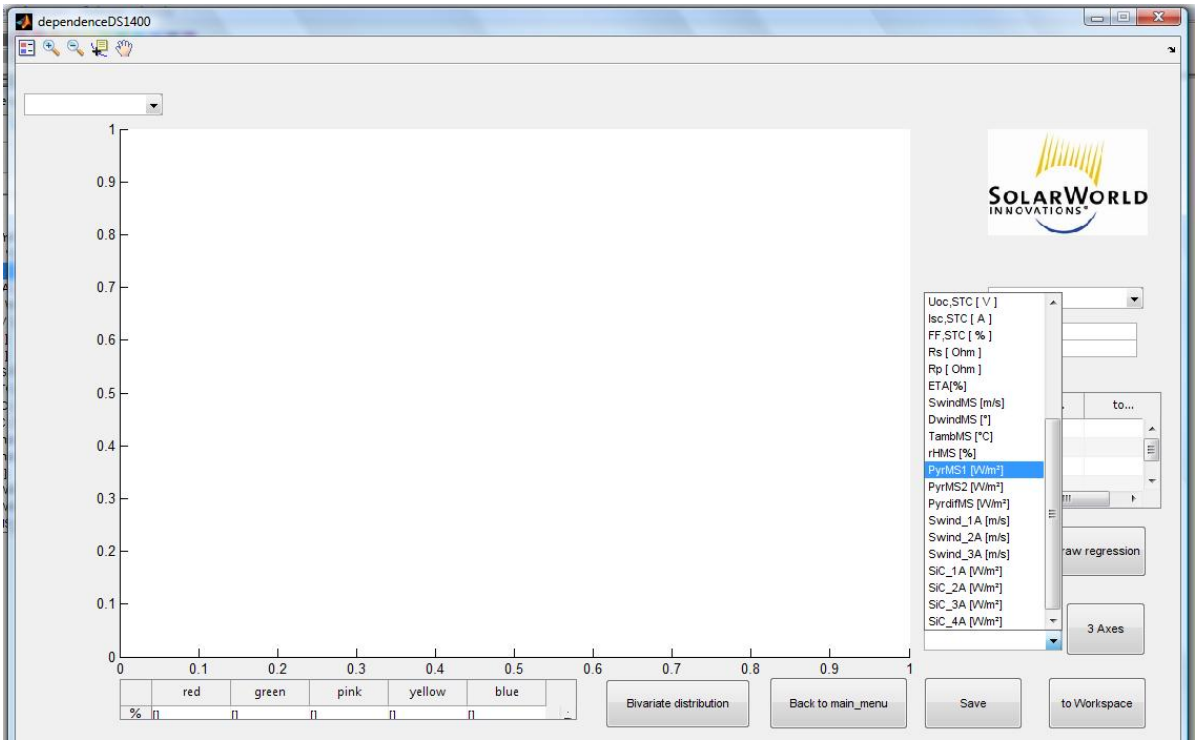
```
function pushbutton3_Callback(hObject, eventdata, handles)
close meteoevaluation
close meteoDS1400_1
main_menu

function pushbutton4_Callback(hObject, eventdata, handles)
global Meteohours Meteomatrix Meteotime SwindMS DwindMS TambMS rHMS
PyrMS1 PyrMS2 PyrdifMS Swind_1A Swind_2A Swind_3A SiC_1A SiC_2A SiC_3A
SiC_4A TModul

uisave({'Meteotime','TModul','SwindMS','DwindMS','TambMS','rHMS','PyrMS1'
,'PyrMS2','PyrdifMS','Swind_1A','Swind_2A','Swind_3A','SiC_1A','SiC_2A','
SiC_3A','SiC_4A'},'MSdataDS1400')
```

F.7. Dependences between modul parameters and weather conditions

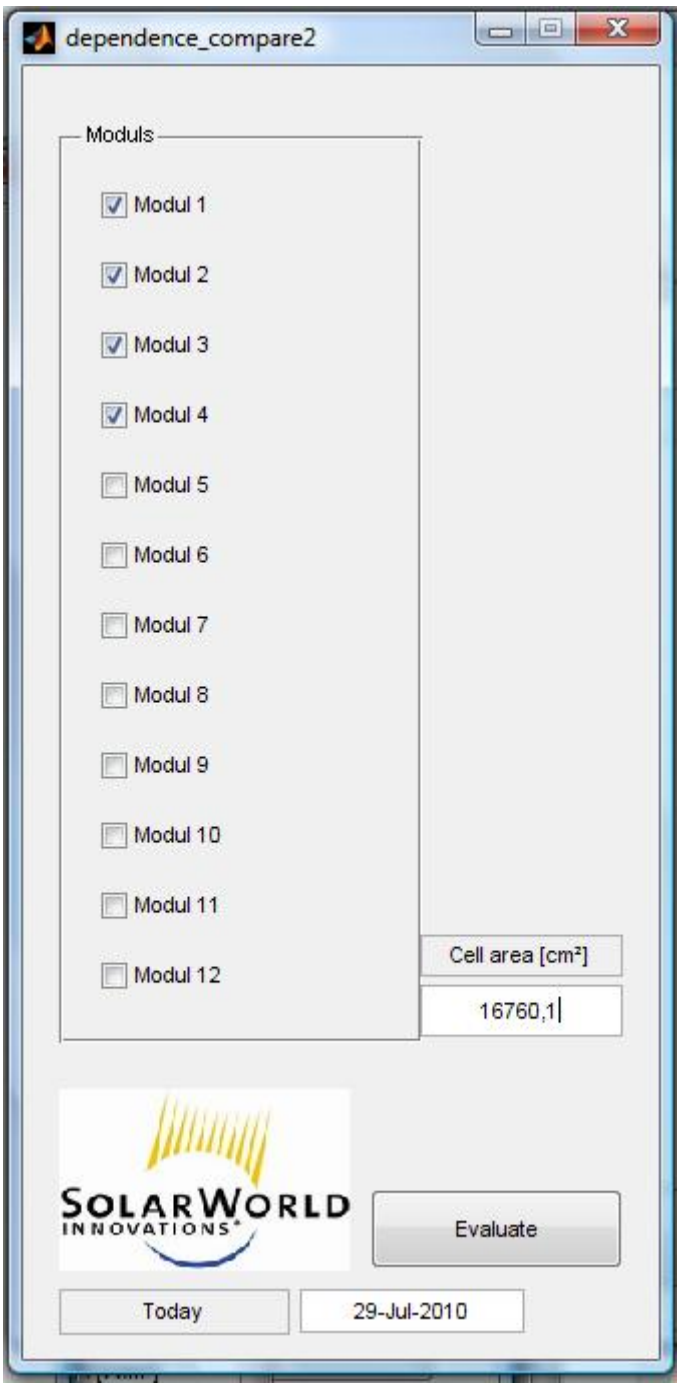
In this section, the programming is similar to the above, the difference lies in the variables used, in this case we use the electrical and thermical parameters of the modules together with meteorological parameters, and so dependences can be founded.

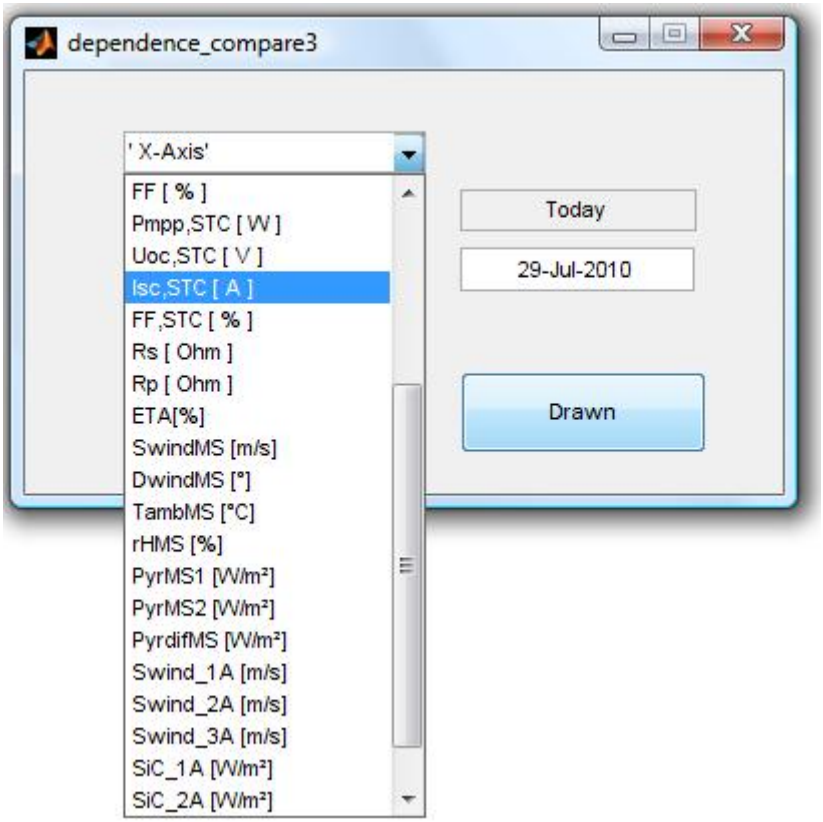


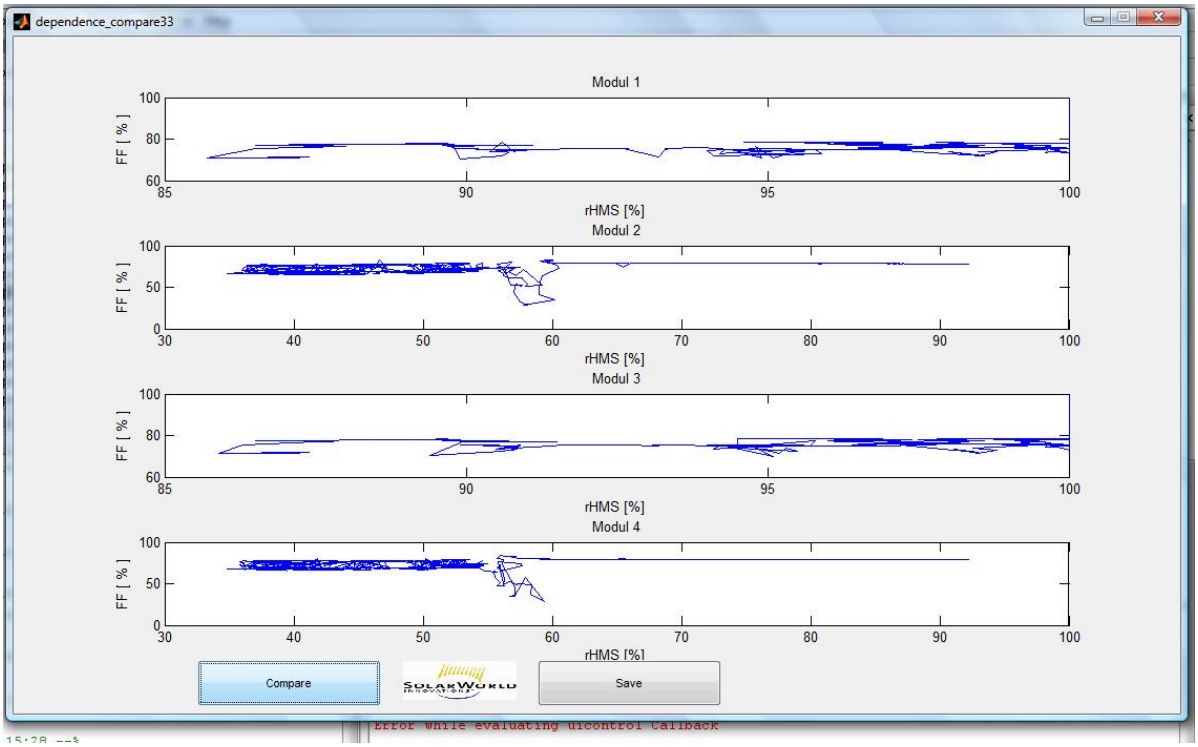


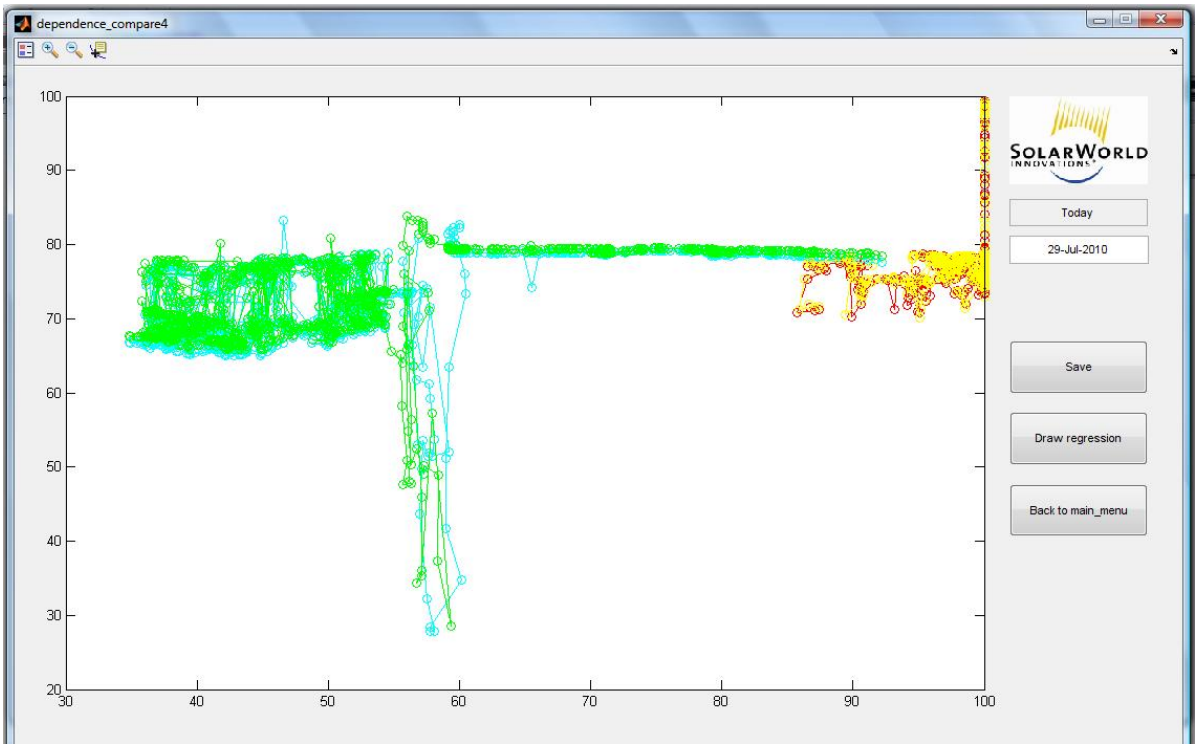

dependenceDS1400
SOLARWORLD
INNOVATIONS
Uoc,STC [V]
Isc,STC [A]
FF,STC [%]
Rs [Ohm]
Rp [Ohm]
ETA[%]
SwindMS [m/s]
DwindMS [°]
TambMS [°C]
rHMS [%]
PyrMS1 [W/m²]
PyrMS2 [W/m²]
PyrdifMS [W/m²]
Swind_1A [m/s]
Swind_2A [m/s]
Swind_3A [m/s]
SiC_1A [W/m²]
SiC_2A [W/m²]
SiC_3A [W/m²]
SiC_4A [W/m²]
to...
raw regression
3 Axes
red
green
pink
yellow
blue
%
Bivariate distribution
Back to main_menu
Save
to Workspace


dependence_compare
Modul 1
Load
Modul 2
Load
Modul 3
Load
Modul 4
Load
Modul 5
Load
Modul 6
Load
Modul 7
Load
Modul 8
Load
Modul 9
Load
Modul 10
Load
Modul 11
Load
Modul 12
Load
SOLARWORLD
INNOVATIONS
Today
29-Jul-2010
Evaluate

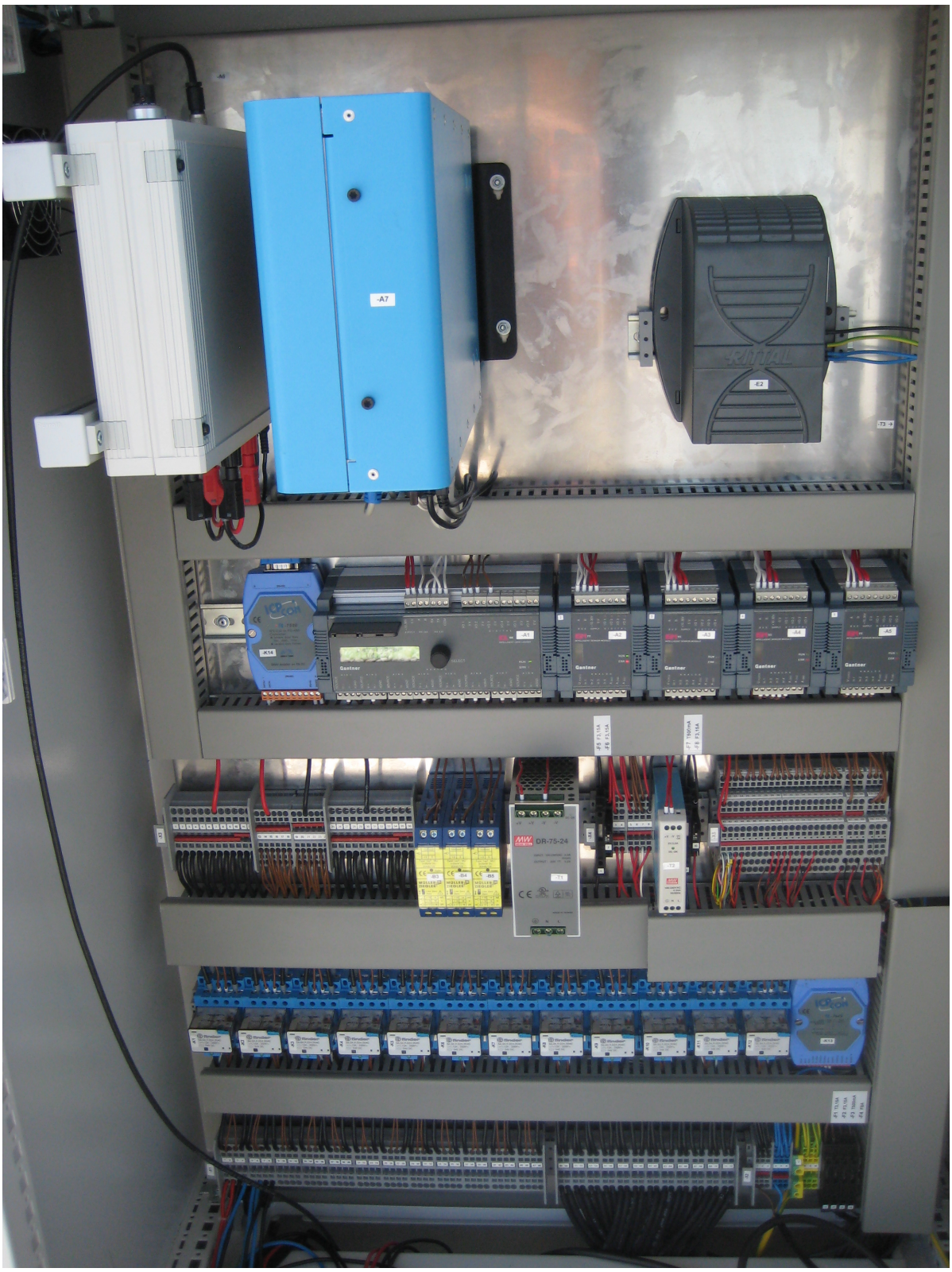

dependence_compare2
Moduls
Modul 1
Modul 2
Modul 3
Modul 4
Modul 5
Modul 6
Modul 7
Modul 8
Modul 9
Modul 10
Modul 11
Modul 12
Cell area [cm²]
16760,1
SOLARWORLD
INNOVATIONS
Evaluate
Today
29-Jul-2010


dependence_compare3
' X-Axis'
FF [%]
Pmpp,STC [W]
Uoc,STC [V]
Isc,STC [A]
FF,STC [%]
Rs [Ohm]
Rp [Ohm]
ETA[%]
SwindMS [m/s]
DwindMS [°]
TambMS [°C]
rHMS [%]
PyrMS1 [W/m²]
PyrMS2 [W/m²]
PyrdifMS [W/m²]
Swind_1A [m/s]
Swind_2A [m/s]
Swind_3A [m/s]
SiC_1A [W/m²]
SiC_2A [W/m²]
Today
29-Jul-2010
Drawn

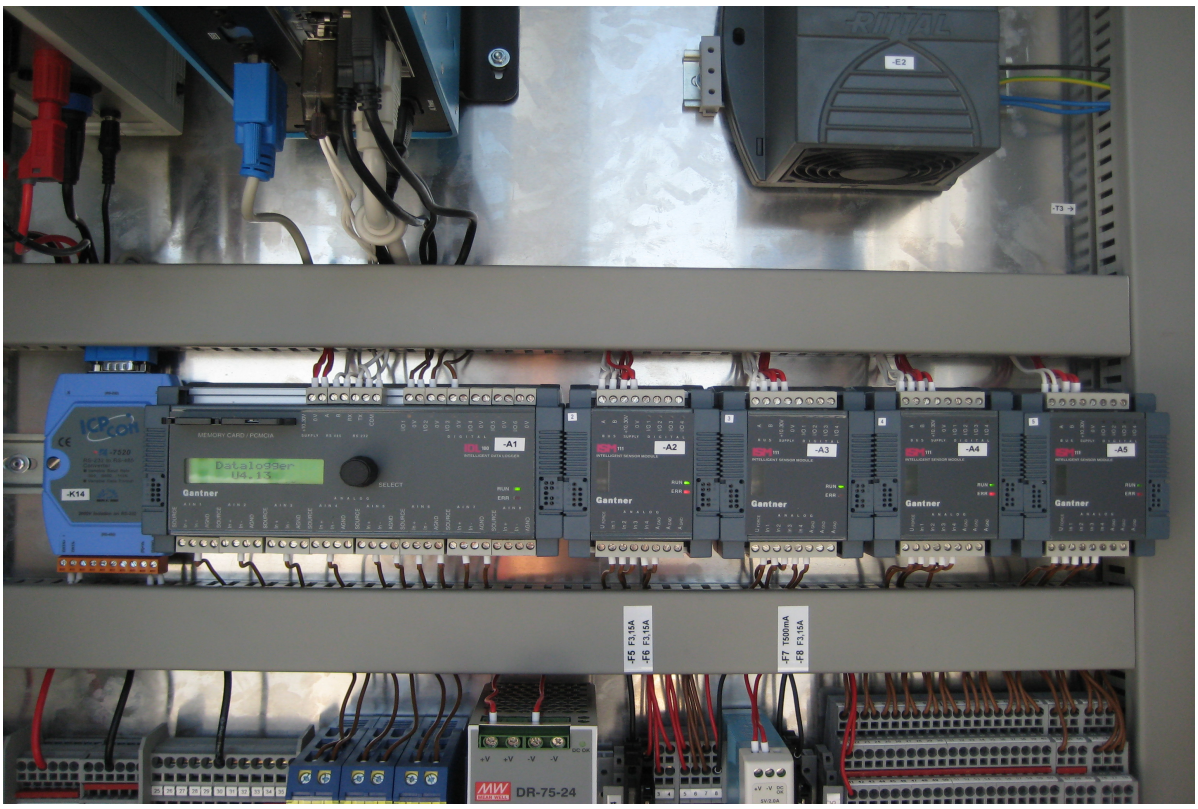

dependence_compare33
Modul 1
Modul 2
Modul 3
Modul 4
FF [%]
rHMS [%]
Compare
Save

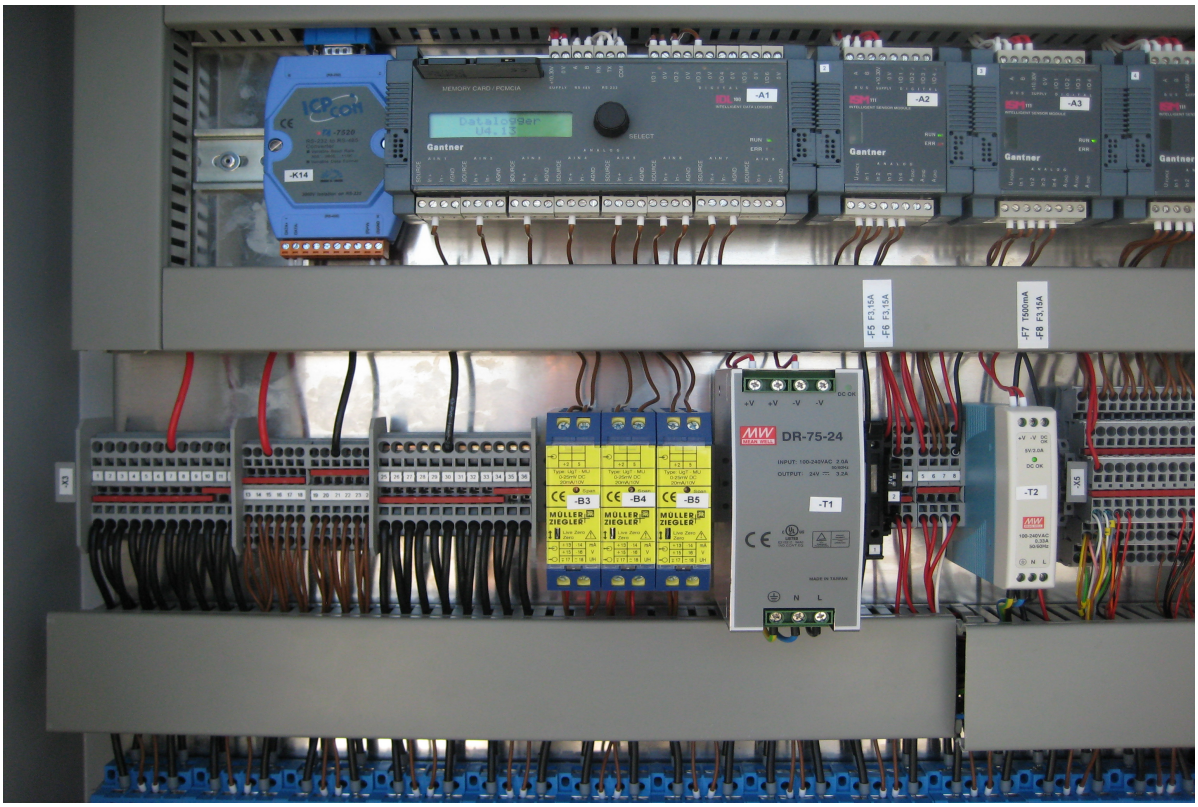

dependence_compare4
100
90
80
70
60
50
40
30
20
30
40
50
60
70
80
90
100
SOLARWORLD
INNOVATIONS
Today
29-Jul-2010
Save
Draw regression
Back to main_menu

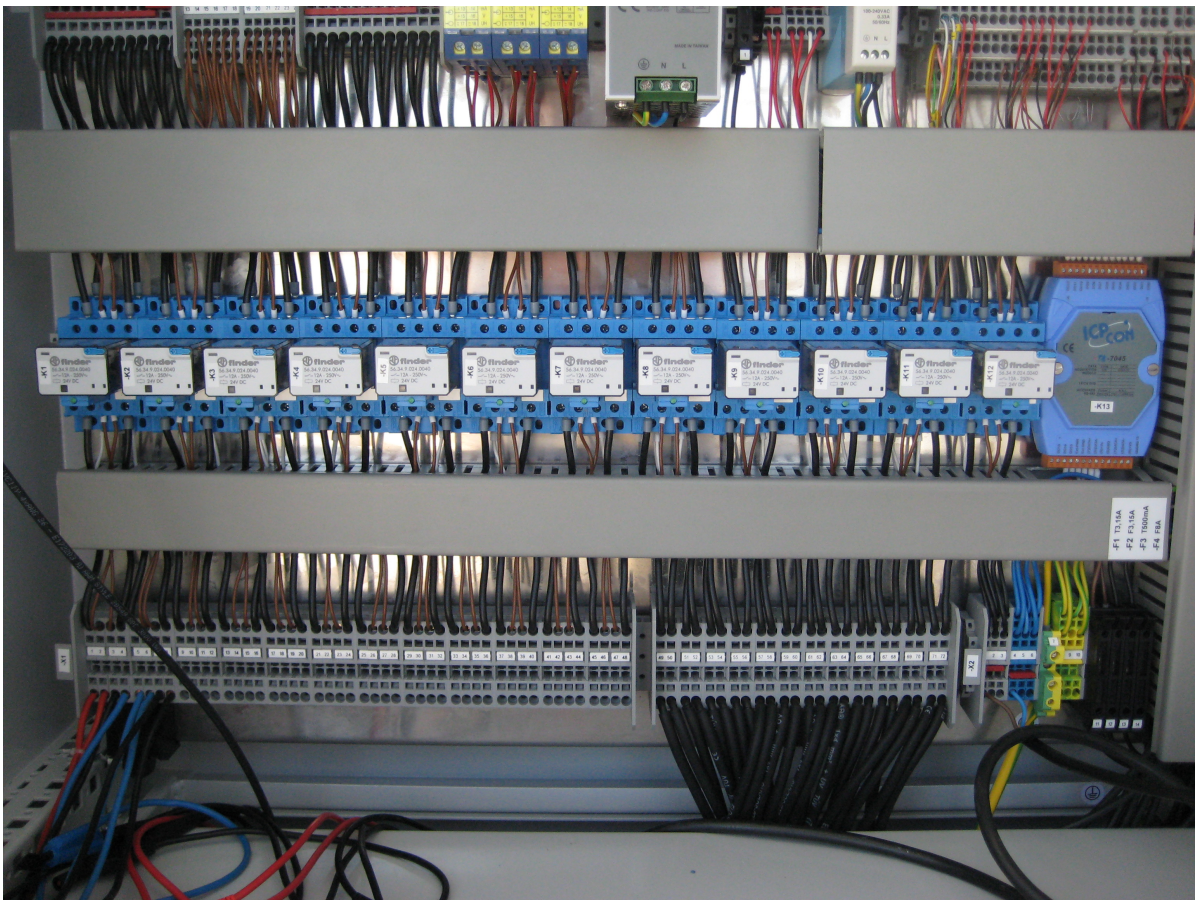
G. Electric Closet

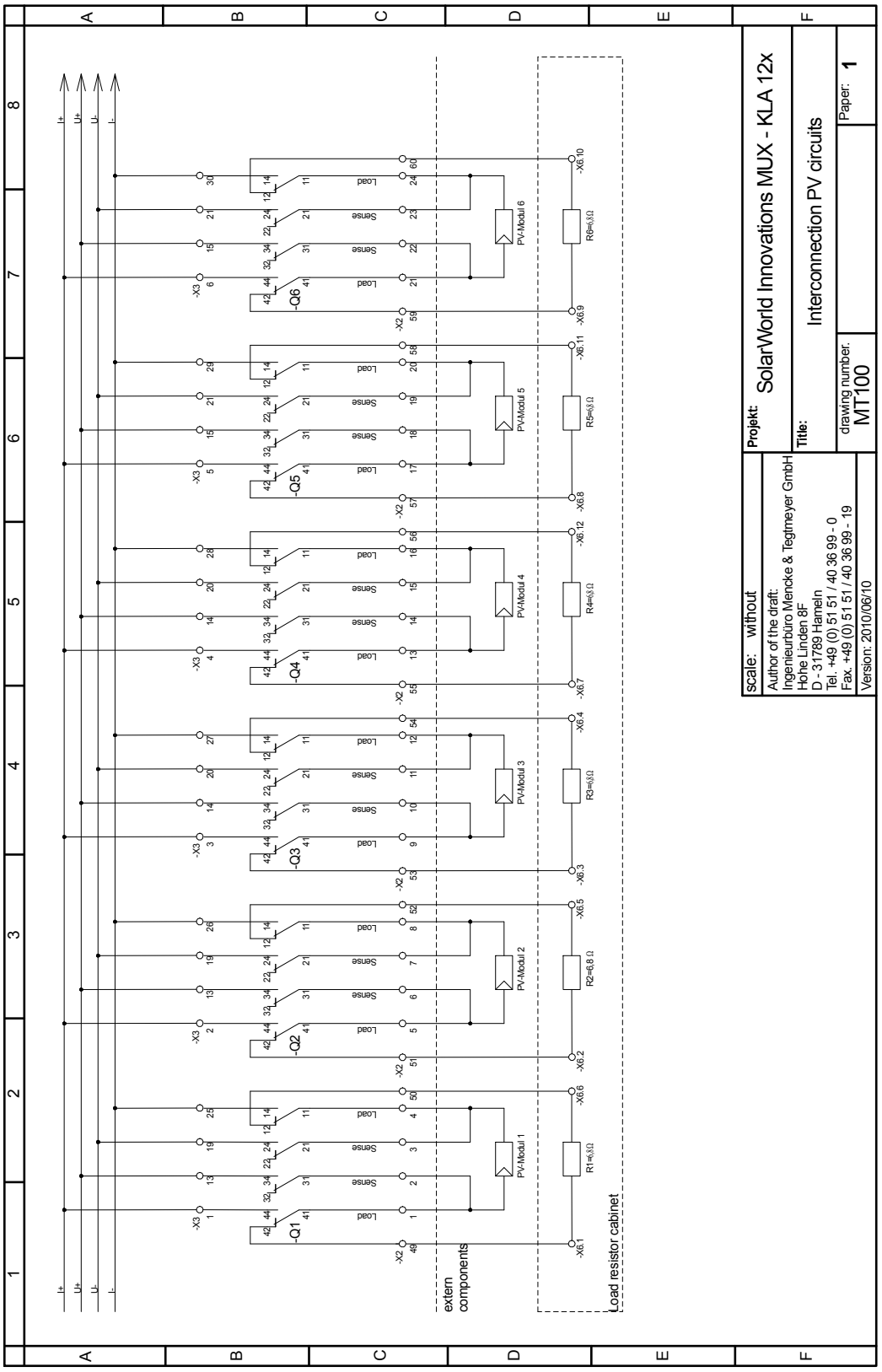

-A7
-E2
Gantner
DR-75-24

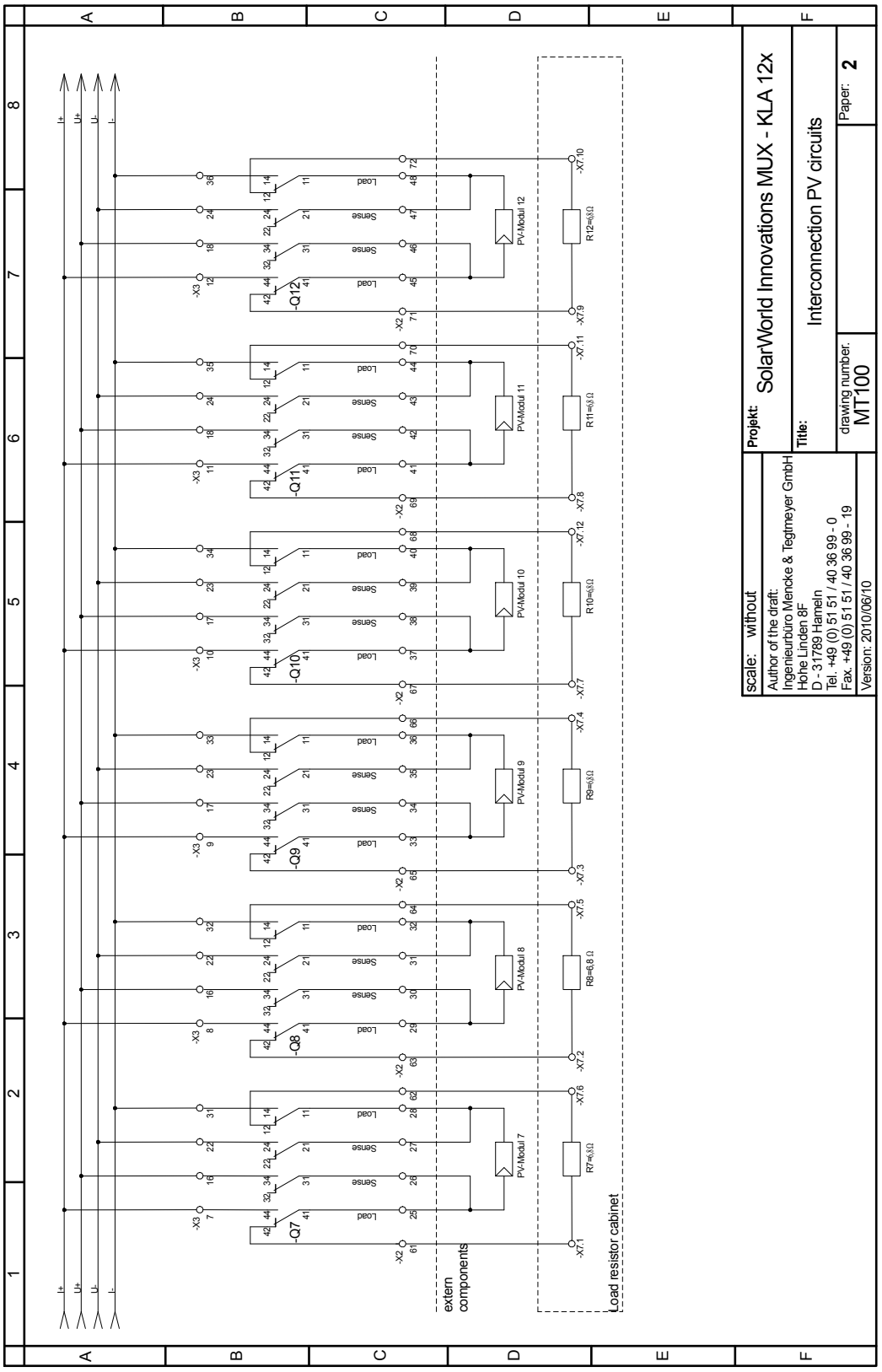

-B1
-A7
-E2

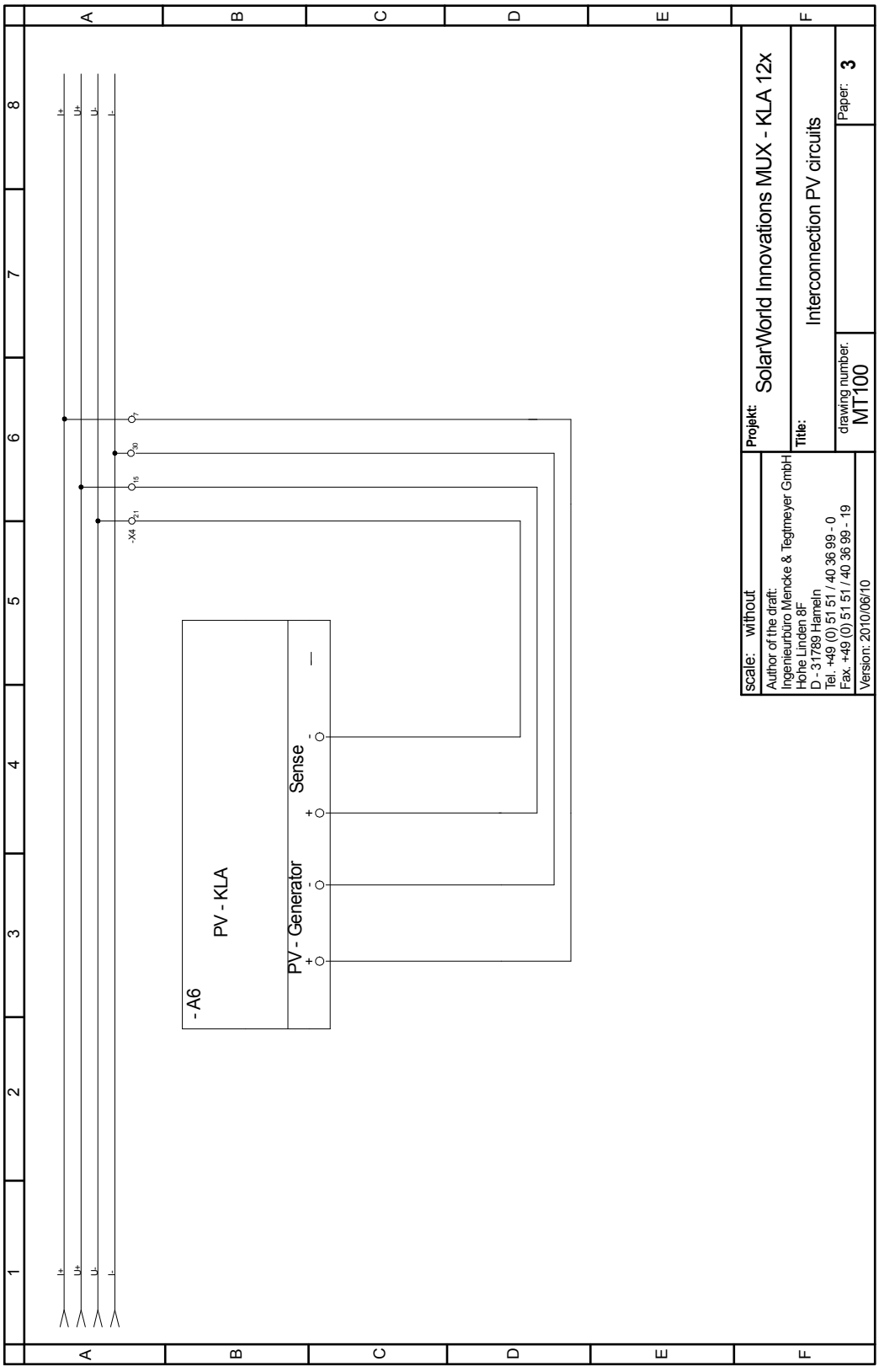

-E2
-A1
-A2
-A3
-A4
-A5
-K14
Gantner
-F5 F3,15A
-F6 F3,15A
-F7 T500mA
-F8 F3,15A
DR-75-24

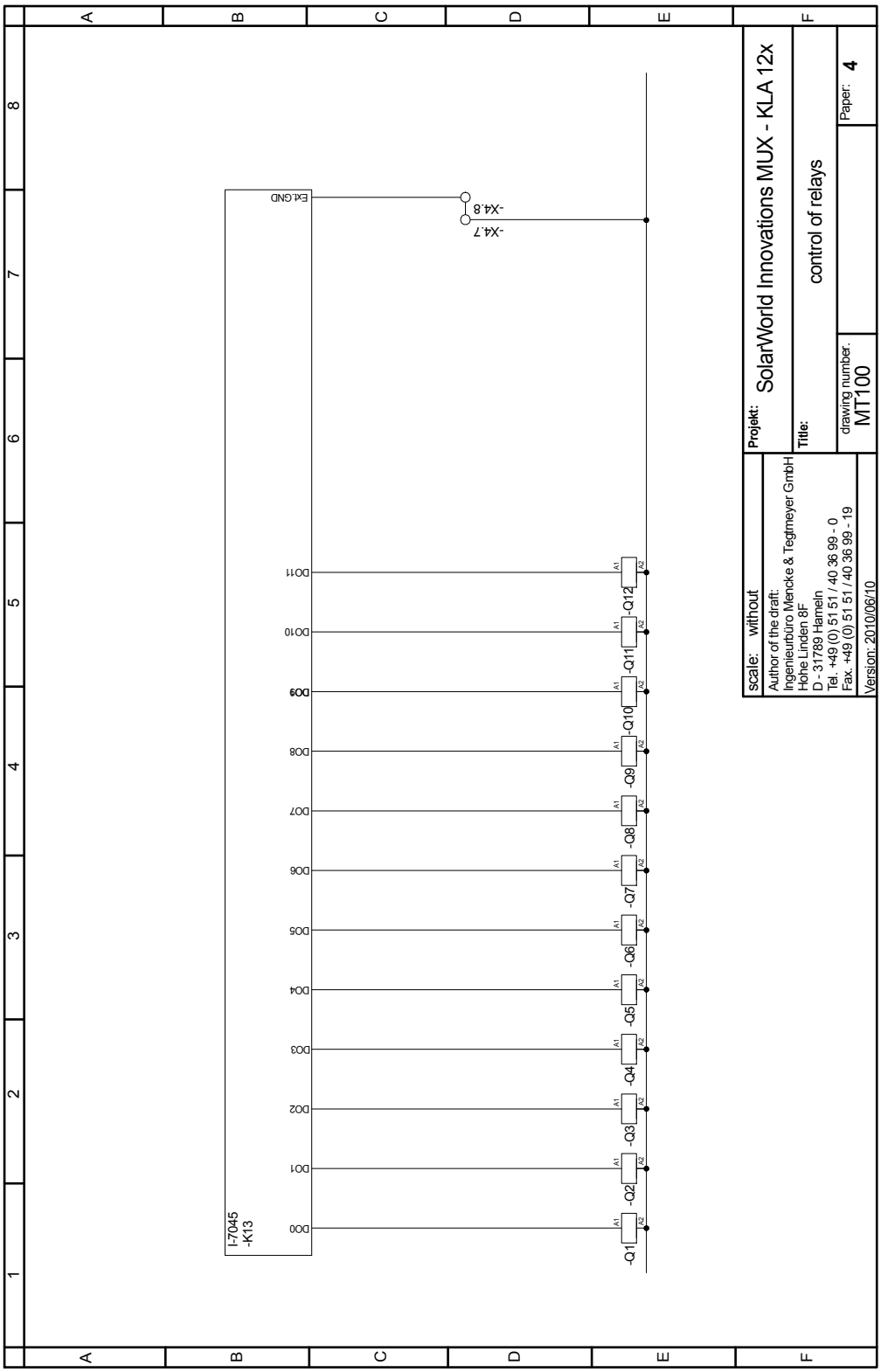

-A1
-A2
-A3
-K14
Gantner
-F5 F3,15A
-F6 F3,15A
-F7 1500mA
-F8 F3,15A
DR-75-24
-T1
-T2
-B3
-B4
-B5
MÜLLER ZIEGLER

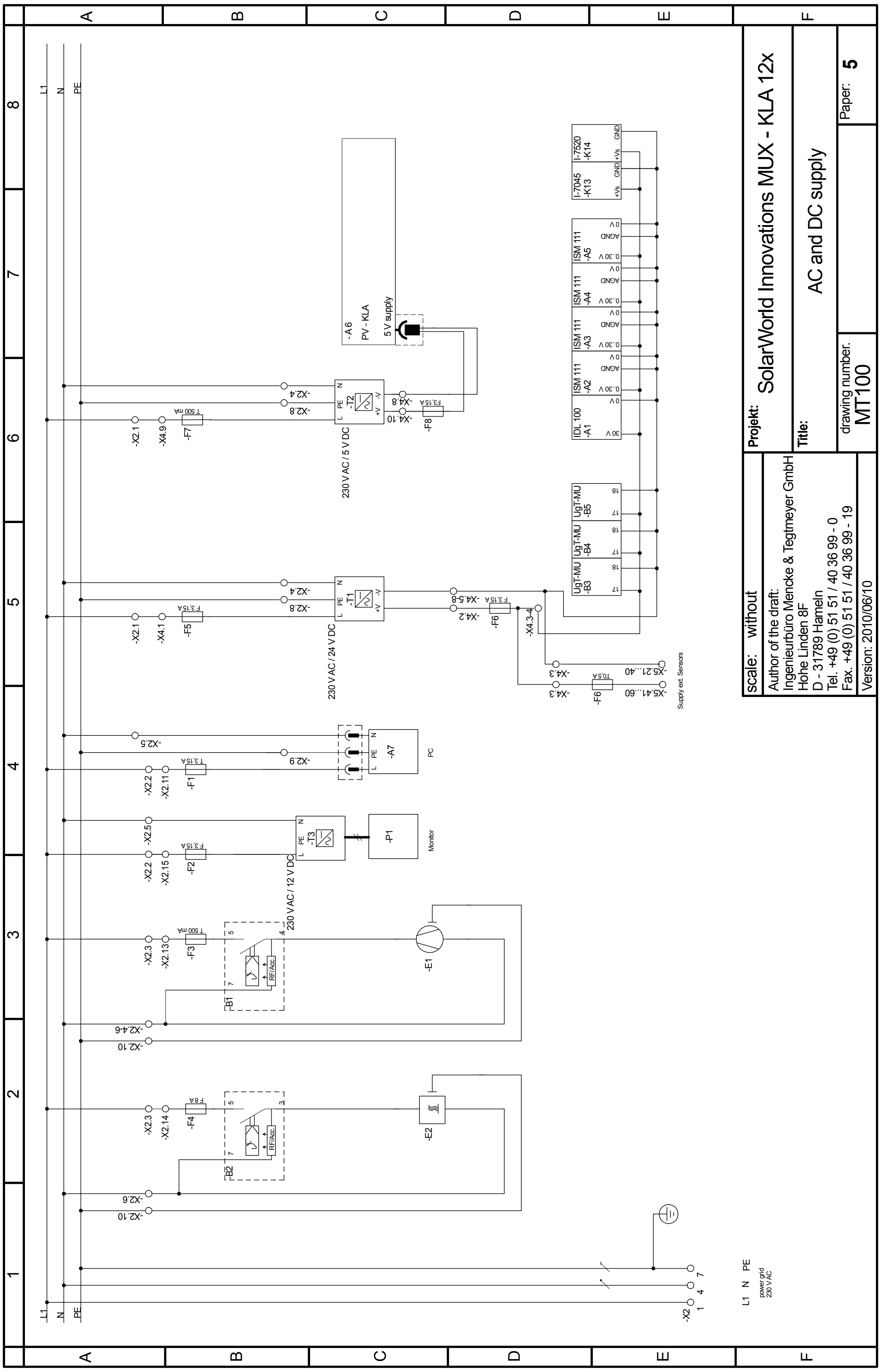

finder
ICP CON
-F1 T3.15A
-F2 T3.15A
-F3 T500mA
-F4 F8A

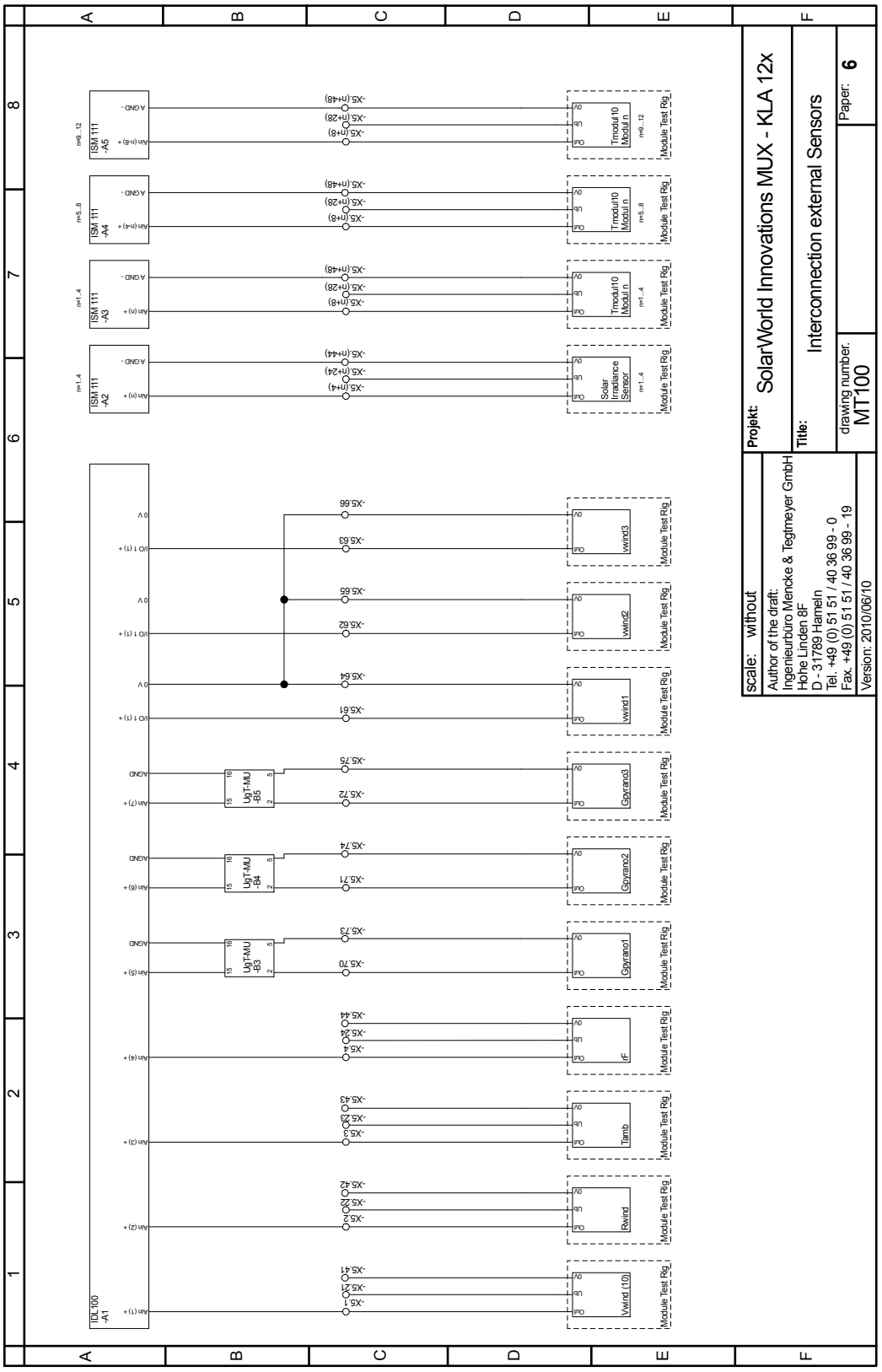

I+
U+
U-
I-
-X3
-X2
-Q1
-Q2
-Q3
-Q4
-Q5
-Q6
Load
Sense
PV-Modul 1
PV-Modul 2
PV-Modul 3
PV-Modul 4
PV-Modul 5
PV-Modul 6
-X6.1
-X6.2
-X6.3
-X6.4
-X6.5
-X6.6
-X6.7
-X6.8
-X6.9
-X6.10
-X6.11
-X6.12
R1=6,8 Ω
R2=6,8 Ω
R3=6,8 Ω
R4=6,8 Ω
R5=6,8 Ω
R6=6,8 Ω
extern components
Load resistor cabinet
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
Interconnection PV circuits
drawing number.
MT100
Paper: 1

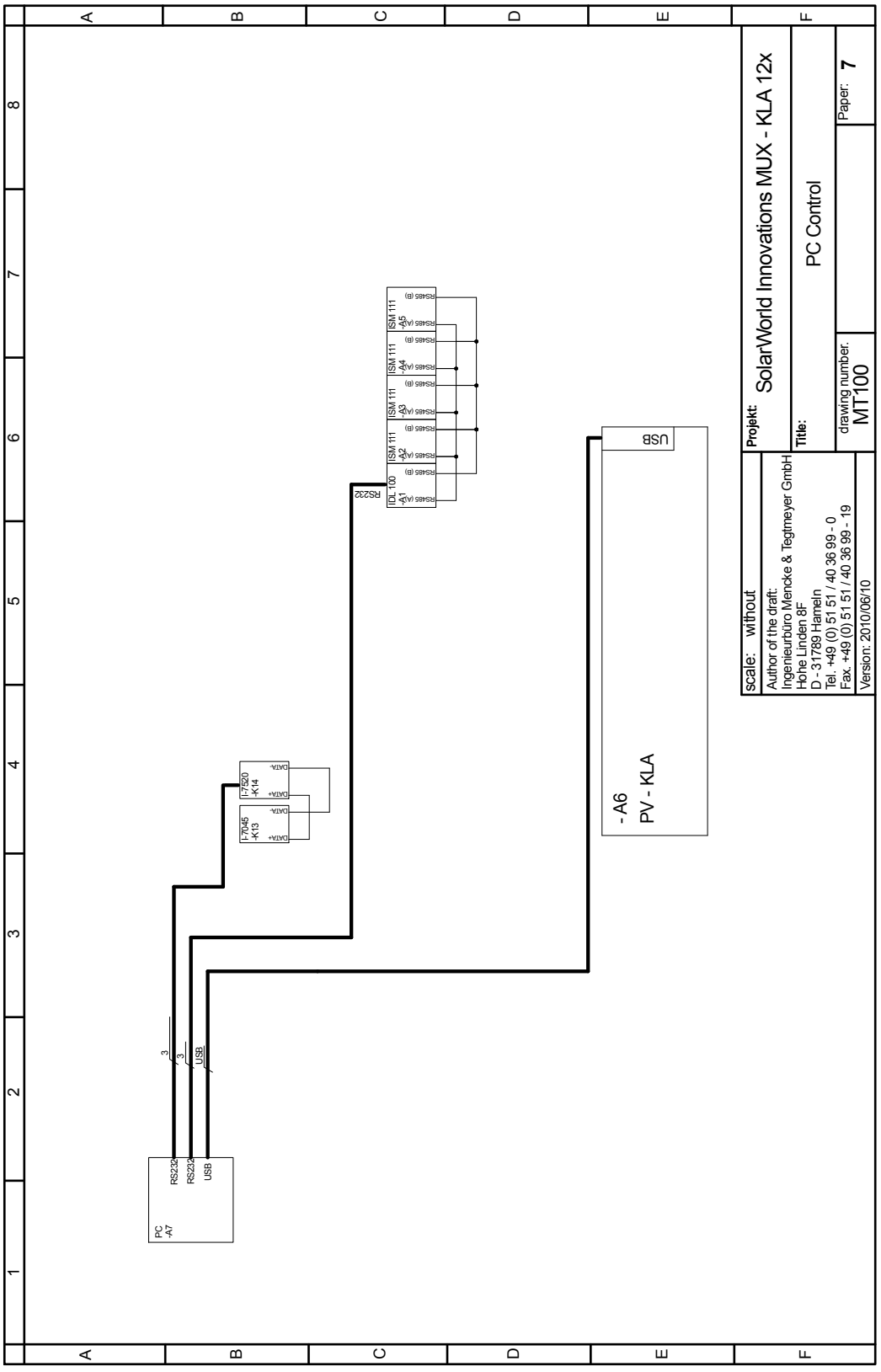

I+
U+
U-
I-
-X3
-X2
-Q7
-Q8
-Q9
-Q10
-Q11
-Q12
Load
Sense
PV-Modul 7
PV-Modul 8
PV-Modul 9
PV-Modul 10
PV-Modul 11
PV-Modul 12
-X7.1
-X7.2
-X7.3
-X7.4
-X7.5
-X7.6
-X7.7
-X7.8
-X7.9
-X7.10
-X7.11
-X7.12
R7=6,8Ω
R8=6,8 Ω
R9=6,8Ω
R10=6,8Ω
R11=6,8 Ω
R12=6,8Ω
extern components
Load resistor cabinet
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
Interconnection PV circuits
drawing number.
MT100
Paper: 2
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10


I+
U+
U-
I-
-X4
7
30
15
21
- A6
PV - KLA
PV - Generator
Sense
+
-
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Höhe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
Interconnection PV circuits
drawing number.
MT100
Paper: 3


Ex.GND
-X4.8
-X4.7
I-7045
-K13
DO0
DO1
DO2
DO3
DO4
DO5
DO6
DO7
DO8
DO9
DO10
DO11
-Q1
-Q2
-Q3
-Q4
-Q5
-Q6
-Q7
-Q8
-Q9
-Q10
-Q11
-Q12
A1
A2
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
control of relays
drawing number:
MT100
Paper: 4
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Höhe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10


scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
AC and DC supply
drawing number:
MT100
Paper: 5


Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
Interconnection external Sensors
drawing number.
MT100
Paper: 6
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10
IDL 100 -A1
ISM 111 -A2
ISM 111 -A3
ISM 111 -A4
ISM 111 -A5
UgT-MU -B3
UgT-MU -B4
UgT-MU -B5
Module Test Rig
Solar Irradiance Sensor
Tmodul10 Modul n
Vwind (10)
Rwind
Tamb
rF
Gpyrano1
Gpyrano2
Gpyrano3
vwind1
vwind2
vwind3


PC
-A7
RS232
RS232
USB
3
3
USB
I-7520
-K14
I-7045
-K13
DATA-
DATA+
DATA-
DATA+
RS232
IDL 100
-A1
ISM 111
-A2
ISM 111
-A3
ISM 111
-A4
ISM 111
-A5
RS485 (A)
RS485 (B)
USB
- A6
PV - KLA
Projekt:
SolarWorld Innovations MUX - KLA 12x
Title:
PC Control
drawing number.
MT100
Paper: 7
scale: without
Author of the draft:
Ingenieurbüro Mencke & Tegtmeyer GmbH
Hohe Linden 8F
D - 31789 Hameln
Tel. +49 (0) 51 51 / 40 36 99 - 0
Fax. +49 (0) 51 51 / 40 36 99 - 19
Version: 2010/06/10

H. Systems photos

H.1. Systems photos