Escuela Universitaria Politécnica - La Almunia
Centro adscrito
Universidad Zaragoza

ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)

Esquemas y planos

Dispositivo de testeo para la verificación del sistema de control de motores en máquinas de coser industriales

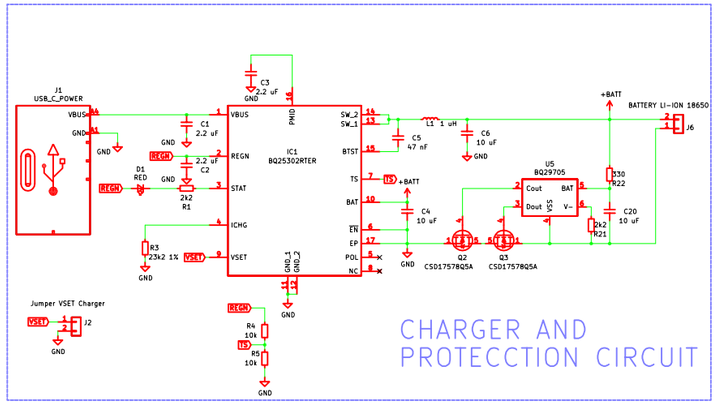
Monitoring Testing device for the Motor Control System in Industrial Sewing Machines

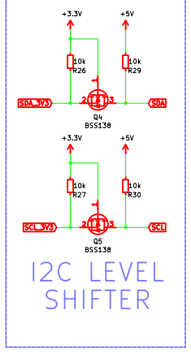
424.20.15

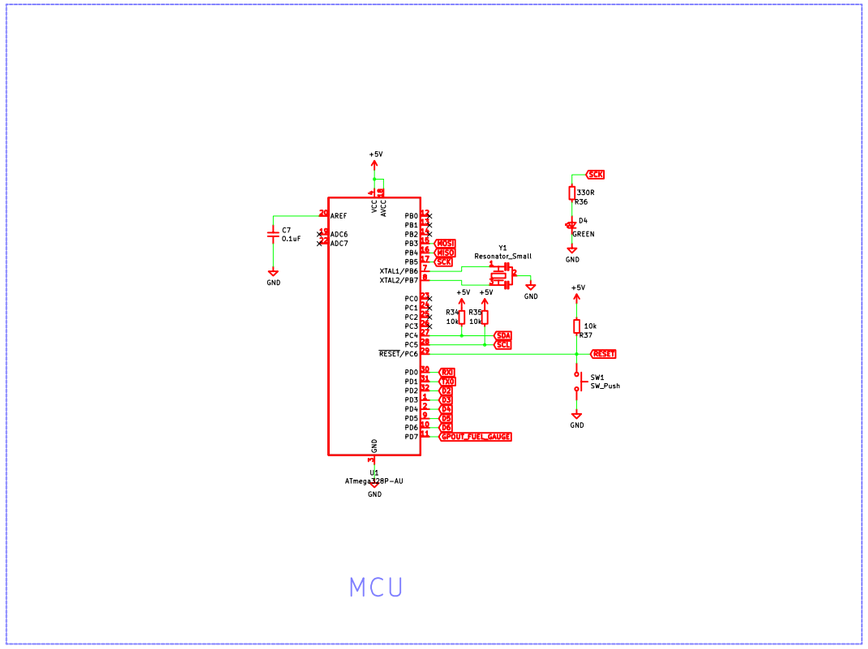
Autor: David Gallardo Ferrer

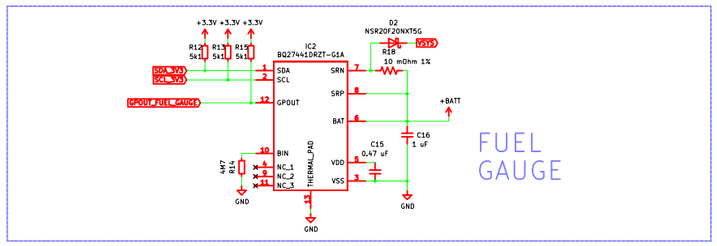
Director: Dr. Jesús Ponce de León Vázquez

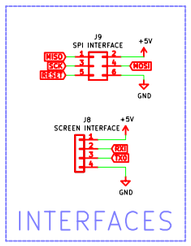
Fecha: 23 de junio del 2021

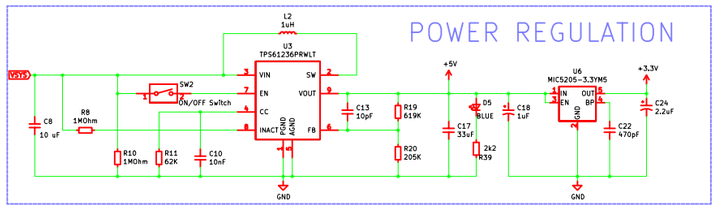


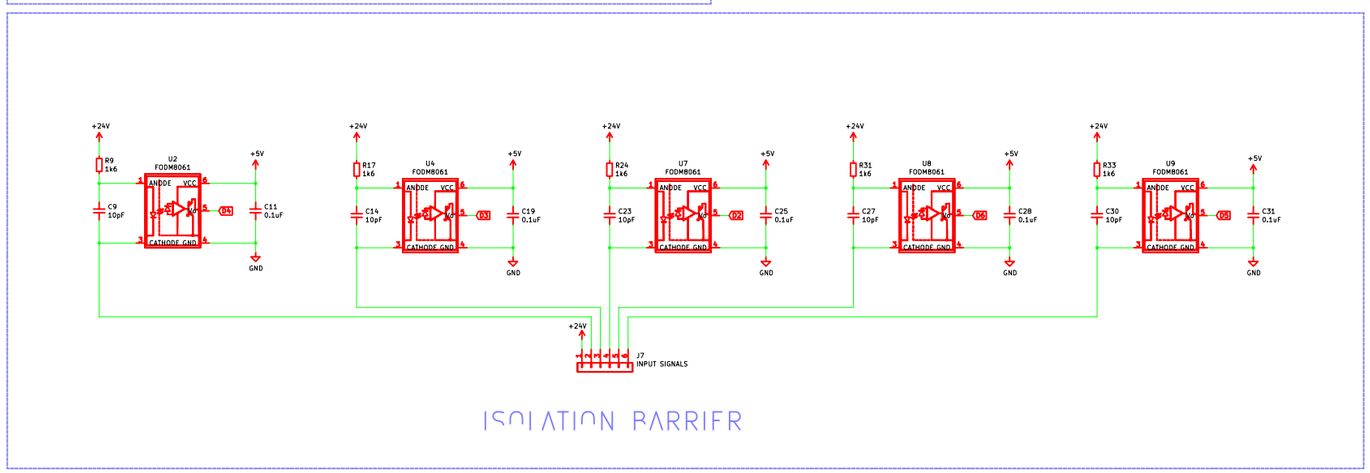




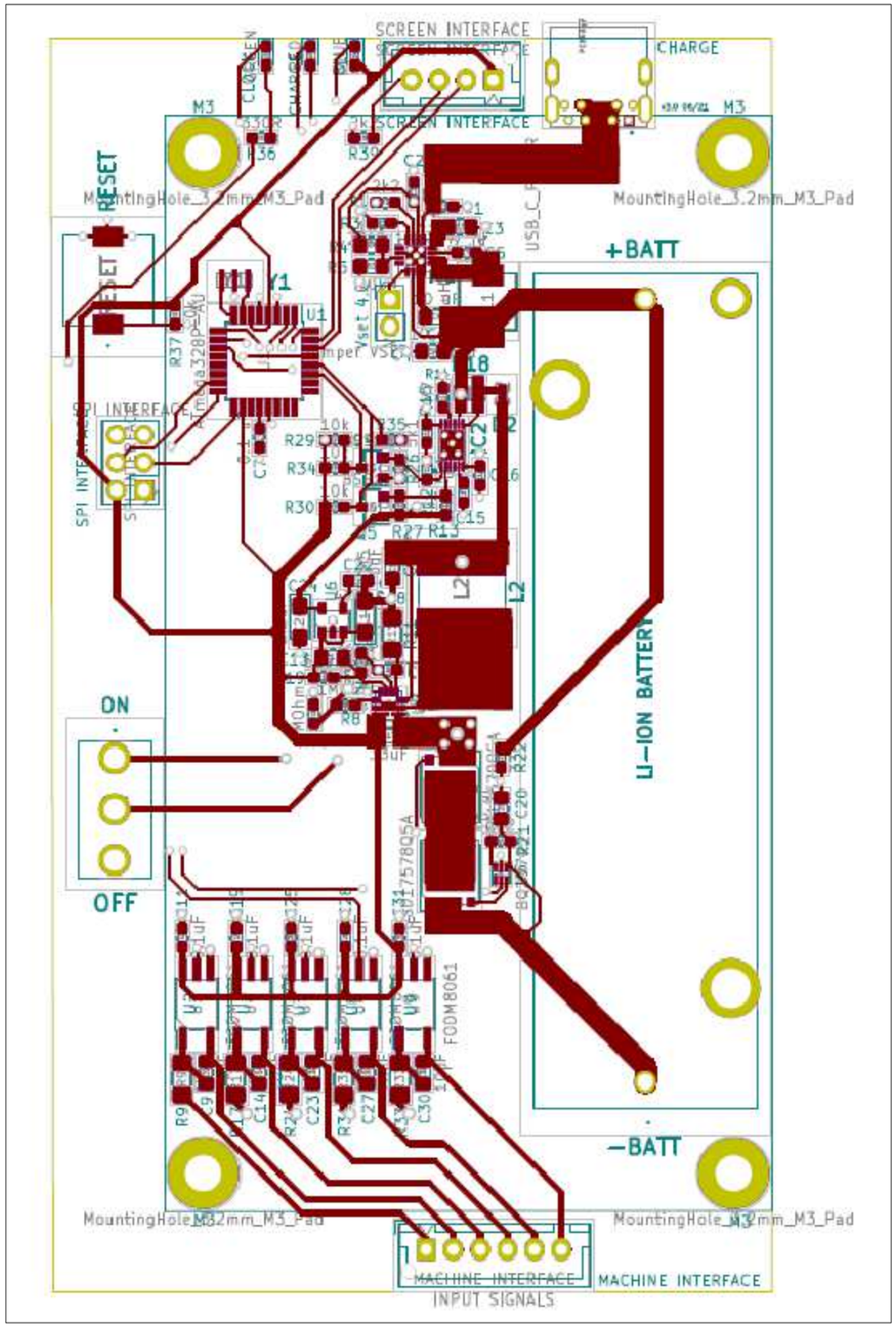


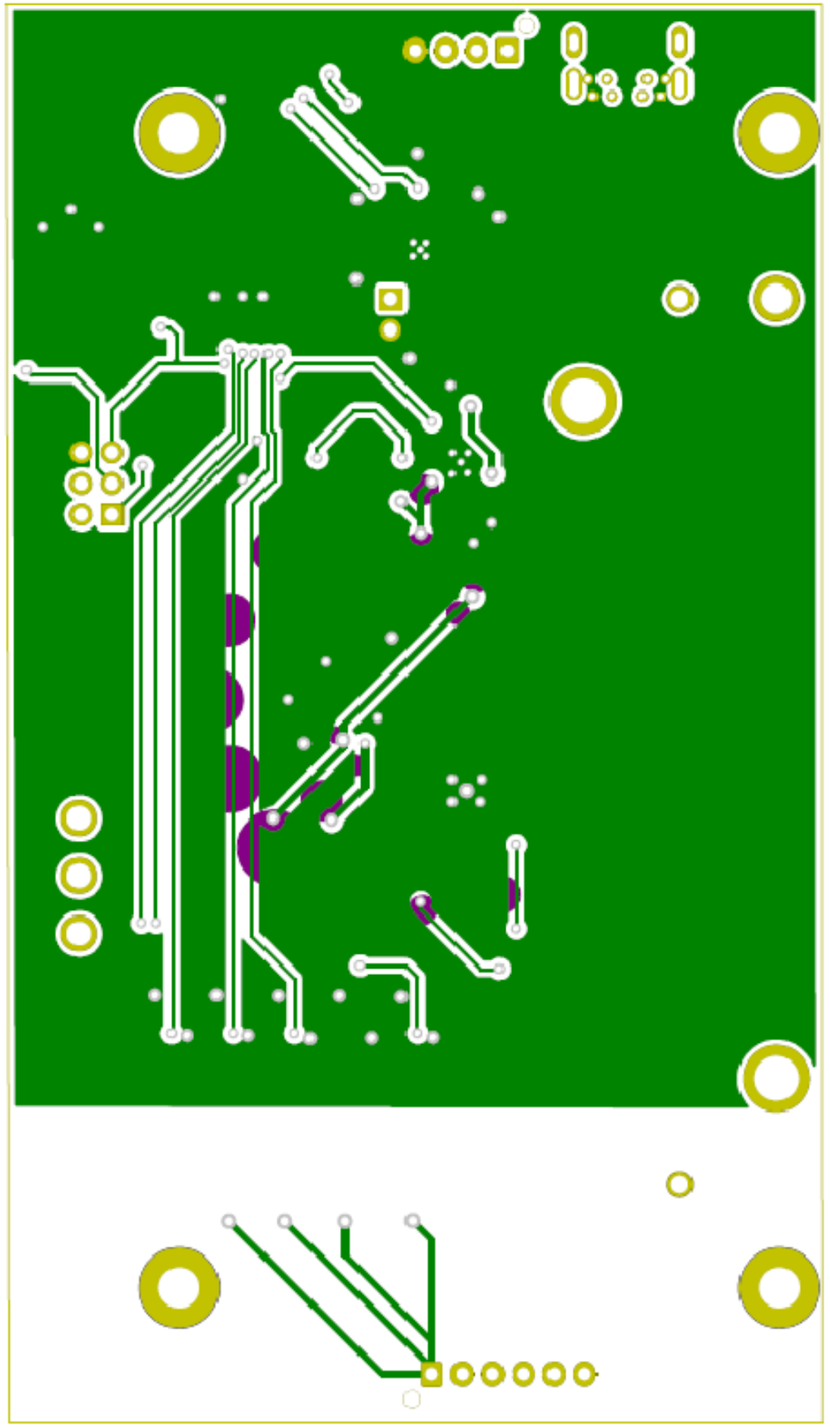




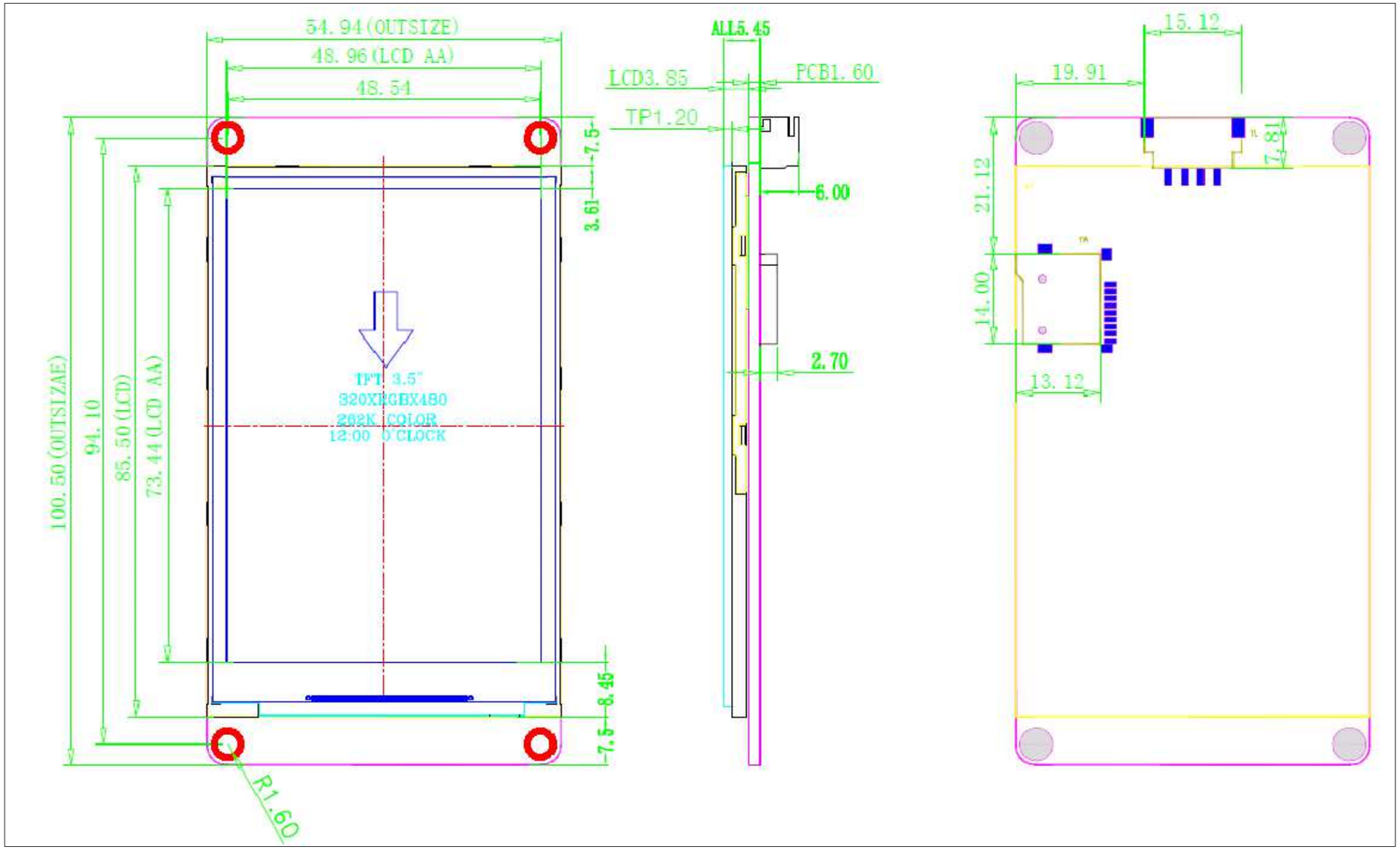


	Fecha	Nombre	Firma:	
Dibujado	05/2021	DAVID GALLARDO FERRER		Escuela Universitaria Politécnica - La Almunia, Centro adscrito, Universidad Zaragoza
Comprobado	07/2021	TRIBUNAL 3		
Rev. A Fecha: 05/2021	DISPOSITIVO DE TESTEO PARA LA VERIFICACIÓN DEL SISTEMA DE CONTROL DE MOTORES EN MÁQUINAS DE COSER INDUSTRIALES			Nº P: 424.20.15.000.01
ESCALA: -	ESQUEMA ELECTRÓNICO DEL DISPOSITIVO DE TESTEO			HOJA: 1/2





	Fecha	Nombre	Firma:	
Dibujado	05/2021	DAVID GALLARDO FERRER		Escuela Universitaria Politécnica - La Almunia, Centro adscrito Universidad Zaragoza
Comprobado	07/2021	TRIBUNAL 3		
Rev. A Fecha: 05/2021	DISPOSITIVO DE TESTEO PARA LA VERIFICACIÓN DEL SISTEMA DE CONTROL DE MOTORES EN MÁQUINAS DE COSER INDUSTRIALES			Nº P: 424.20.15.000.02
ESCALA: 2:1	CIRCUITO IMPRESO DEL DISPOSITIVO DE VERIFICACIÓN			HOJA: 2/2



	Fecha	Nombre	Firma:	
Dibujado	05/2021	NEXTION		Escuela Universitaria Politécnica - La Almunia Centro adscrito Universidad Zaragoza
Comprobado	07/2021	TRIBUNAL 3		
Rev. A Fecha: 05/2021	DISPOSITIVO DE TESTEO PARA LA VERIFICACIÓN DEL SISTEMA DE CONTROL DE MOTORES EN MÁQUINAS DE COSER INDUSTRIALES			Nº P: 424.20.15.000.03
ESCALA: -	ESQUEMA ELECTRÓNICO DEL DISPOSITIVO DE TESTEO			HOJA: 1/1

Escuela Universitaria Politécnica - La Almunia
Centro adscrito
Universidad Zaragoza

ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)

ANEXOS

Dispositivo de testeo para la verificación del sistema de control de motores en máquinas de coser industriales

Monitoring Testing device for the Motor Control System in Industrial Sewing Machines

424.20.15

Autor: David Gallardo Ferrer

Director: Dr. Jesús Ponce de León Vázquez

Fecha: 23 de junio de 2021

INDICE DE CONTENIDO

INDICE DE ILUSTRACIONES

INDICE DE TABLAS

1. ANEXO 1: CÓDIGO FUENTE

El código fuente del presente proyecto se ha realizado en el entorno de desarrollo de Arduino. Para programar el Atmega328p se ha hecho uso del programador AVRISP MKII.

Dado que se han hecho uso de librerías, tanto para el uso del HMI como del indicador de batería, el código se ha simplificado en gran medida. Por lo tanto, no ha sido necesario estructurar el programa en dos archivos. Sin embargo, se ha decidido utilizar distintas pestañas para localizar fácilmente distintas partes del código.

1.1. PROGRAMA PRINCIPAL

```
#include <BQ27441_Definitions.h>
#include <SparkFunBQ27441.h>

#include "Nextion.h"


#define LEVEL_HIGH (20)
#define LEVEL_LOW (0)

#define OFFSET (20 - LEVEL_HIGH/2)

#define BATTERY_CAPACITY 2200

bool flag_comms = 0;
unsigned long prevMillis = millis();
const int GPOUT_PIN = 7;

const byte SOCI_SET = 15; // Interrupt set threshold at 20%
const byte SOCI_CLR = 20; // Interrupt clear threshold at 25%
const byte SOCF_SET = 5; // Final threshold set at 5%
const byte SOCF_CLR = 10; // Final threshold clear at 10%

enum t_page {BLOCK,MENU,CHART,OPTIONS,LOW_BAT};
t_page current_page = BLOCK;

NexPage locked = NexPage(0, 0, "locked");
NexPage menu = NexPage(1, 0, "menu");
NexPage wave = NexPage(2, 0, "wave");
NexPage settings = NexPage(3, 0, "settings");
NexPage lowbat = NexPage(4, 0, "lowbat");

NexWaveform s0 = NexWaveform(2, 1, "s0");
NexWaveform s1 = NexWaveform(2, 8, "s1");
NexWaveform s2 = NexWaveform(2, 9, "s2");
NexWaveform s3 = NexWaveform(2, 10, "s3");
NexWaveform s4 = NexWaveform(2, 11, "s4");
```

Anexo 1: Código fuente

```
NexText soc_info = NexText(3, 10, "soc_info");
NexText volts_info = NexText(3, 11, "volts_info");
NexText current_info = NexText(3, 12, "current_info");
NexText full_cap_info = NexText(3, 13, "full_cap_info");
NexText cap_info = NexText(3, 14, "cap_info");
NexText power_info = NexText(3, 15, "power_info");
NexText soh_info = NexText(3, 16, "soh_info");

static uint8_t ch0_data = LEVEL_LOW;
static uint8_t ch1_data = LEVEL_LOW;
static uint8_t ch2_data = LEVEL_LOW;
static uint8_t ch3_data = LEVEL_LOW;
static uint8_t ch4_data = LEVEL_LOW;

NexTouch *nex_listen_list[] =
{
  &locked,
  &menu,
  &wave,
  &settings,
  &lowbat,
  NULL
};

void lockedPushCallback(void *ptr);
void menuPushCallback(void *ptr);
void wavePushCallback(void *ptr);
void settingsPushCallback(void *ptr);
void setupBQ27441(void);

void setup() {
  nexInit();
  locked.attachPush(lockedPushCallback);
  menu.attachPush(menuPushCallback);
  wave.attachPush(wavePushCallback);
  settings.attachPush(settingsPushCallback);
  dbSerialPrintln("setup done");

  setupBQ27441();


  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  pinMode(4,INPUT_PULLUP);
  pinMode(5,INPUT_PULLUP);
  pinMode(6,INPUT_PULLUP);

}

void loop() {
  nexLoop(nex_listen_list);
  if(!digitalRead(GPOUT_PIN) && lipo.socfFlag()){
    current_page = LOW_BAT;
  }
  switch(current_page){
      case BLOCK:

      break;
```


Autor: David Gallardo Ferrer

```
      case MENU:

      break;

      case CHART:
        if(digitalRead(2) == LOW){
          ch0_data = LEVEL_HIGH;
        }else{
          ch0_data = LEVEL_LOW;
        }

        if(digitalRead(3) == LOW){
          ch1_data = LEVEL_HIGH;
        }else{
          ch1_data = LEVEL_LOW;
        }

        if(digitalRead(4) == LOW){
          ch2_data = LEVEL_HIGH;
        }else{
          ch2_data = LEVEL_LOW;
        }

        if(digitalRead(5) == LOW){
          ch3_data = LEVEL_HIGH;
        }else{
          ch3_data = LEVEL_LOW;
        }

        if(digitalRead(6) == LOW){
          ch4_data = LEVEL_HIGH;
        }else{
          ch4_data = LEVEL_LOW;
        }

        s0.addValue(0, OFFSET + ch0_data);
        s1.addValue(0, OFFSET + ch1_data);
        s2.addValue(0, OFFSET + ch2_data);
        s3.addValue(0, OFFSET + ch3_data);
        s4.addValue(0, OFFSET + ch4_data);
      break;

      case OPTIONS:
        if(flag_comms){
          unsigned int soc = lipo.soc();  // Read state-of-charge (%)
          unsigned int volts = lipo.voltage(); // Read battery voltage
(mV)
          int current = lipo.current(AVG); // Read average current (mA)
          unsigned int fullCapacity = lipo.capacity(FULL); // Read full
capacity (mAh)
          unsigned int capacity = lipo.capacity(REMAIN); // Read
remaining capacity (mAh)
          int power = lipo.power(); // Read average power draw (mW)
          int health = lipo.soh(); // Read state-of-health (%)

          char buffer[20];
          sprintf(buffer,"%d %%",soc);
          soc_info.setText(buffer);
```

```
            sprintf(buffer,"%d mV",volts);
            volts_info.setText(buffer);

            sprintf(buffer,"%d mA",current);
            current_info.setText(buffer);

            sprintf(buffer,"%d mAh",fullCapacity);
            full_cap_info.setText(buffer);

            sprintf(buffer,"%d mAh",capacity);
            cap_info.setText(buffer);

            sprintf(buffer,"%d mW",power);
            power_info.setText(buffer);

            sprintf(buffer,"%d %%",health);
            soh_info.setText(buffer);
            prevMillis = millis();
          }

        break;

        case LOW_BAT:
          lowbat.show();
          if(digitalRead(GPOUT_PIN)){
            current_page = BLOCK;
            locked.show();
          }
        break;
      }

}
```

1.1.1. Interrupciones

```
void lockedPushCallback(void *ptr){
  current_page = BLOCK;
}

void menuPushCallback(void *ptr){
  current_page = MENU;
}

void wavePushCallback(void *ptr){
  current_page = CHART;
}

void settingsPushCallback(void *ptr){
  current_page = OPTIONS;
}
```

1.1.2. Inicialización del indicador de batería

```
void setupBQ27441(void){
  if (!lipo.begin()) // begin() will return true if communication is
successful
    {
      flag_comms = 0;
    }else{
      flag_comms = 1;
      lipo.enterConfig(); // To configure the values below, you must be
in config mode
      lipo.setCapacity(BATTERY_CAPACITY); // Set the battery capacity
      lipo.setGPOUTPolarity(LOW); // Set GPOUT to active-high
      lipo.setGPOUTFunction(BAT_LOW); // Set GPOUT to BAT_LOW mode
      lipo.setSOC1Thresholds(SOCI_SET, SOCI_CLR); // Set SOCI set and
clear thresholds
      lipo.setSOCFThresholds(SOCF_SET, SOCF_CLR); // Set SOCF set and
clear thresholds
      lipo.exitConfig();
    }
  // Uset lipo.setCapacity(BATTERY_CAPACITY) to set the design capacity
  // of your battery.
}
```

2. ANEXO 2: DIAGRAMAS DE ACTIVIDAD UML

En este apartado se incluyen los diagramas de flujo que describen el código fuente.

2.1. OPERACIÓN GENERAL

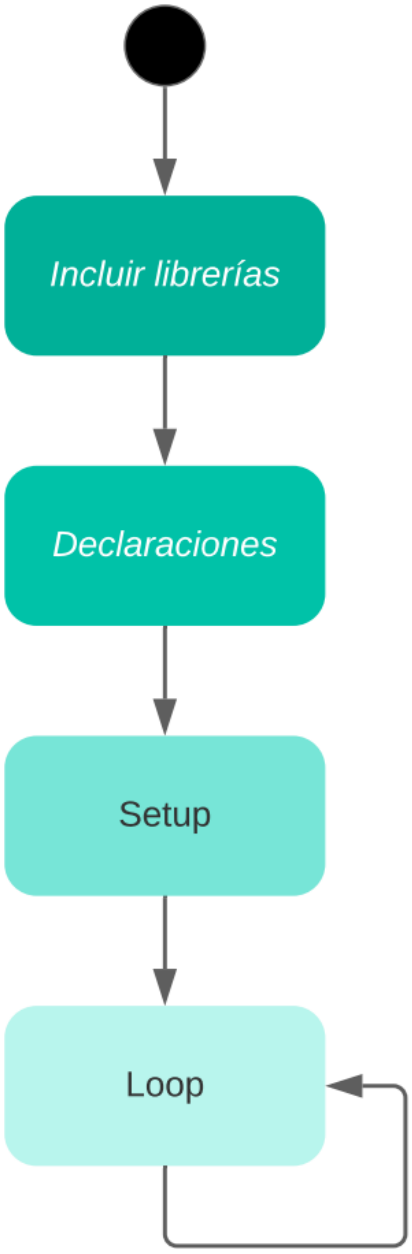


Ilustración 1 Diagrama de actividad: Operación general

2.2. SETUP

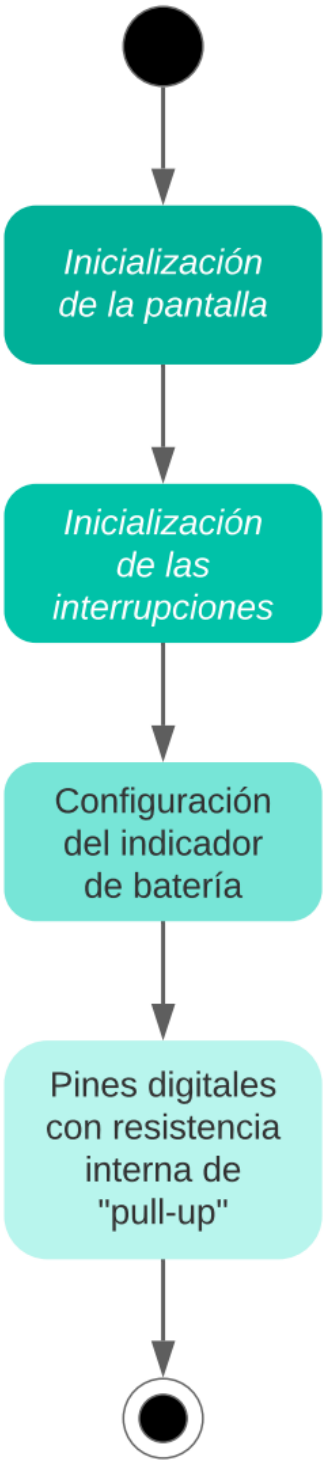


Ilustración 2 Diagrama de actividad: Setup

2.3. LOOP

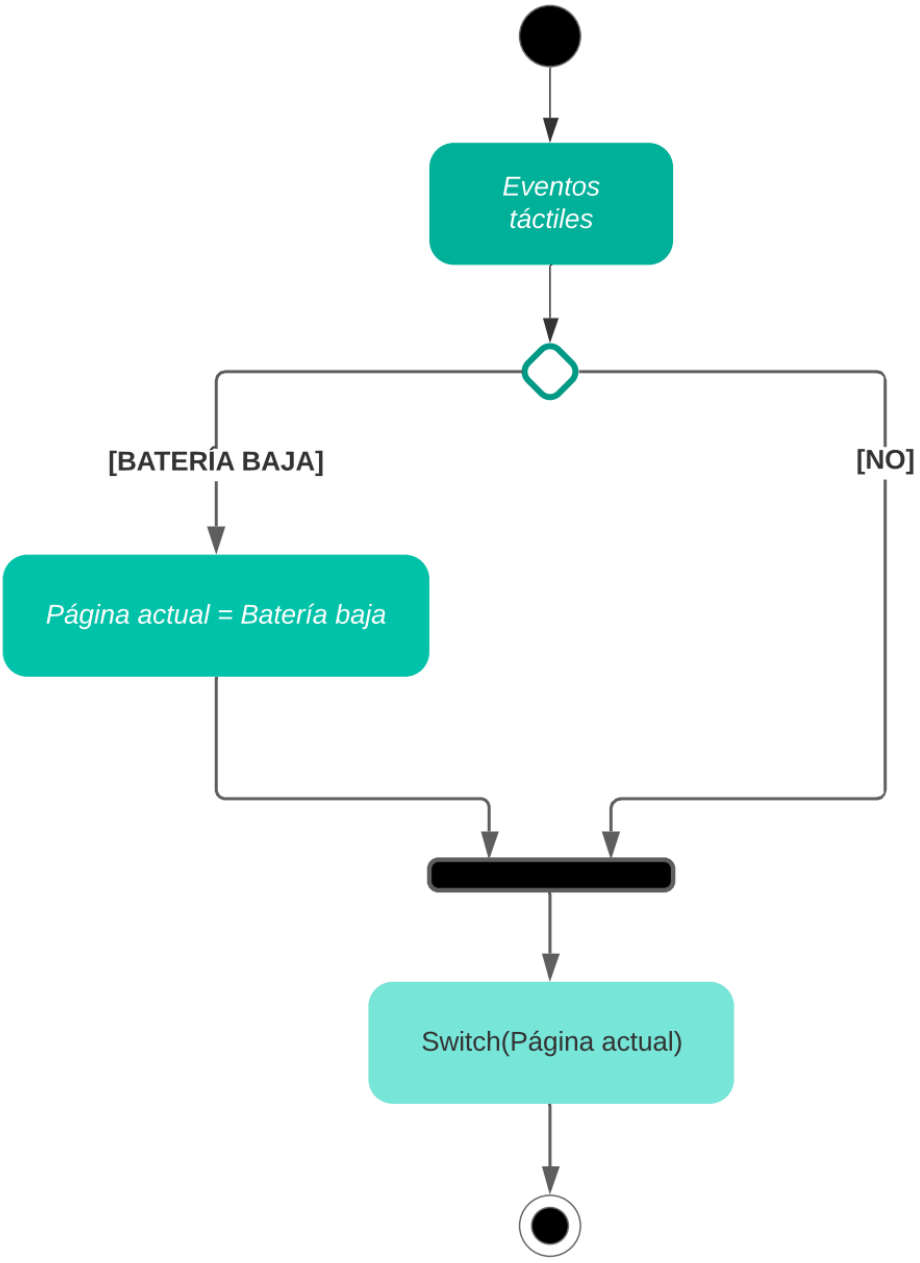


Ilustración 3 Diagrama de actividad: Loop

2.4. SWITCH-CASE

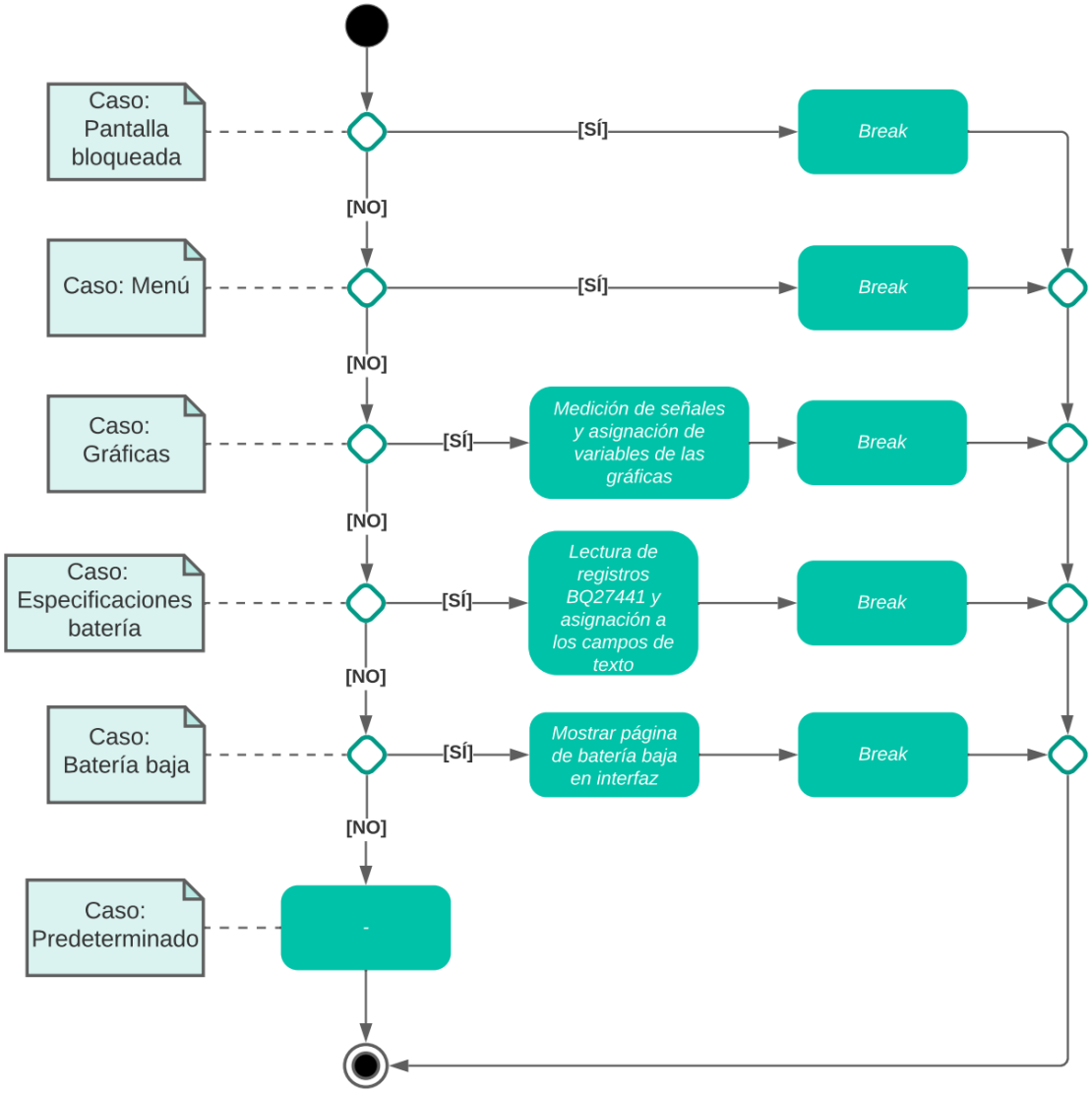


Ilustración 4 Diagrama de actividad: Switch-Case

2.5. SETUP BQ27441

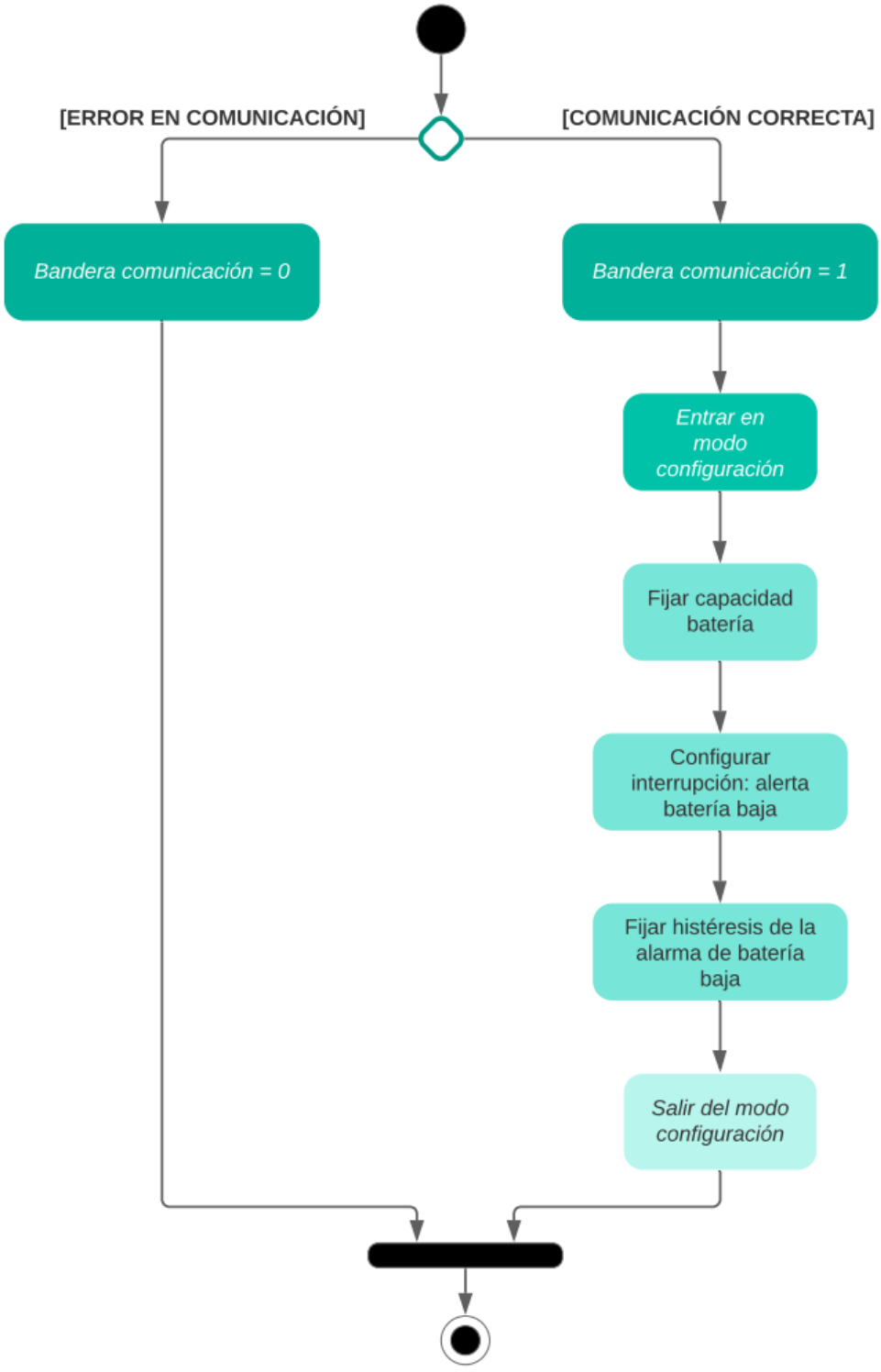


Ilustración 5 Diagrama de actividad: Setup BQ27441

2.6. EVENTOS DE CAMBIO DE PÁGINA

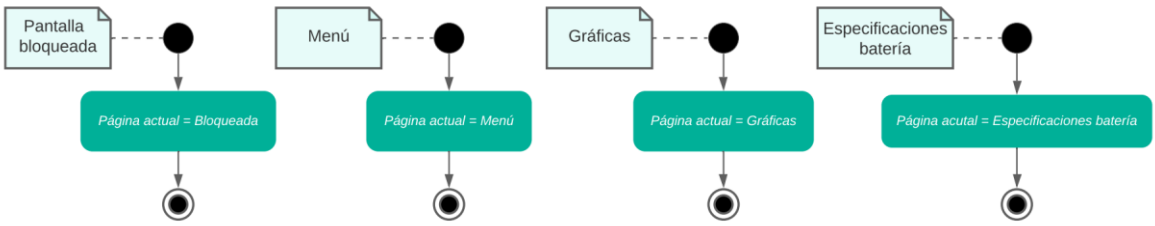


Ilustración 6 Diagrama de actividad: Eventos de cambio de página

3. ANEXO 3: LISTA DE MATERIALES DEL CIRCUITO IMPRESO (BOM)

En este apartado se incluye el listado de componentes utilizados para el montaje del circuito impreso diseñado.

Tabla 1 Lista de materiales (BOM)

ID	Empaquetado	Cantidad en la PCB	Valor / Referencia
LI-ION BATTERY	1043	1	BATTERY LI-ION 18650
R12, R13, R15	0603	3	5k1
R21, R39, R1	0603	3	2k2
IC1	QFN50P300X300X80-17N-D	1	BQ25302RTER
C1, C2, C3	0603	3	2.2 uF
C4, C6, C8, C20	0805	3	10 uF
C9, C13, C14, C23, C27, C30	0805	6	10pF
C7, C11, C19, C25, C28, C31	0603	5	0.1uF
C15	0603	1	0.47 uF
C16	0603	1	1 uF
C17	0805	1	33uF
C18	3216-18	1	1uF
C22	0603	1	470pF
C24	3216-18	1	2.2uF
LED CHARGE	0603	1	RED
LED CLOCK	0603	1	GREEN
LED ON	0603	1	BLUE
IC2	SON40P400X250X100-13N-D	1	BQ27441DRZT-G1A
J2	PinHeader_1x02_P2.54mm_Vertical	1	Jumper VSET Charger
MACHINE INTERFACE	JST_XH_B6B-XH-AM_1x06_P2.50mm_Vertical	1	INPUT SIGNALS
SCREEN INTERFACE	JST_XH_B4B-XH-AM_1x04_P2.50mm_Vertical	1	SCREEN INTERFACE
SPI INTERFACE	PinHeader_2x03_P2.54mm_Vertical	1	SPI INTERFACE
L1	2020	1	IHLP2020BZER1R0M11
L2	-	1	SRN8040-1R0Y

Q2, Q3	TDSON-8-1	2	CSD17578Q5A
Q4, Q5	SOT-23	2	BSS138
R9, R17, R24, R31, R33	1206	5	1k6
R11	1206	1	62K
R14	0603	1	4M7
R18	-	1	RESC1732X80N
R26, R27, R29, R30, R34, R35, R37, R4, R5	0603	9	10k
R22, R36	0603	1	330R
U2, U4, U7, U8, U9	SO-5_4.4x3.6mm_P1.27mm	5	FODM8061
U5	WSON-6_1.5x1.5mm_P0.5mm	1	BQ29705
U6	SOT-23-5	1	MIC5205-3.3YM5
Y1	-	1	CSTNE16M0V53L000R0
U3	-	1	TPS61236PRWLT
C5	0603	1	47 nF
C10	0603	1	10nF
R3	0603	1	23k2 1%
R8, R10	0603	2	1MOhm
R19	0603	1	619K
R20	0603	1	205K
U1	TQFP-32_7x7mm_P0.8mm	1	ATmega328P-AU
D2	DSN2	1	NSR20F20NXT5G
USB CHARGE	-	1	UJC-HP-G-SMT-TR
ON/OFF Switch	-	1	118251376
RESET BUTTON	-	1	TL9320AF400QG

Relación de documentos

(_) Memoria	178	páginas
(X) Anexos	13	páginas
(X) Planos	3	páginas

La Almunia, a 23 de 06 de 2021

Firmado: David Gallardo Ferrer