

Trabajo Fin de Grado

DISEÑO Y DESARROLLO DE UNA ESTACIÓN METEOROLÓGICA ALIMENTADA CON ENERGÍA SOLAR

Autor/es

Jorge Pérez Izquierdo

Director/es

Raúl Igual Catalán

Carlos Tomás Medrano Sánchez

Escuela Universitaria Politécnica de Teruel

2017

"Diseño y desarrollo de una estación meteorológica alimentada con energía solar"

RESUMEN

Este Trabajo Fin de Grado consiste en el diseño y desarrollo de un prototipo de estación meteorológica energéticamente autosuficiente para lugares remotos. Para alimentarlo se ha utilizado una placa solar y una batería para el almacenamiento de la energía excedente.

Se ha utilizado una placa Arduino para la realización de todas las mediciones de los sensores y, posteriormente, realizar un almacenamiento local y remoto de los datos automáticamente cada cierto tiempo.

"Design and development of a solar-powered weather station"

ABSTRACT

This End of Grade Work consists in the design and development of prototype of weather station energetically self-sufficient for remote places. A solar panel has been used to power it, including a battery for the storage of surplus energy.

An Arduino board has been used for the conduction of all sensor measurements. This board has also been used to store the collected data locally and remotely in an automatic way from time to time.



INDICE

1. INTRODUCCIÓN	5
1.1.Definiciones previas.....	5
1.2. Objetivo del proyecto.....	8
1.3. Estructura del proyecto.	8
2. TRABAJOS RELACIONADOS.....	9
3. INSTRUMENTACIÓN UTILIZADA.....	11
3.1. Arduino Mega.	11
3.2. GSM GPRS Arduino Shield.	12
3.3. Módulo solar, batería y placa solar.	15
3.4. Sensor de temperatura LM35.....	17
3.5. Sensor de temperatura NTC.	17
3.6. Sensor de humedad del suelo.....	18
3.7. Sensor de lluvia.....	19
3.8. Sensor de presión y temperatura (BMP180).	20
3.9. LCD y tarjeta SD.....	21
4. PLACA DE CIRCUITO IMPRESO.....	23
5. DISEÑO Y DESARROLLO DE LA ESTACIÓN.....	29
5.1.DISEÑO HARDWARE.....	29
5.2. DISEÑO SOFTWARE.	31
5.2.1.Librerías.....	31
5.2.2. Filtros software.	33
5.2.3. Diagramas de flujo.	33
5.2.4. Servidor.....	37



5.2.5. Página web.....	38
5.3. PROTOTIPO FINAL.....	40
6. CONSUMO DEL PROTOTIPO.....	41
7. PRUEBAS EN ENTORNOS REALES.....	43
8. COSTES DEL PROTOTIPO.....	48
9. CONCLUSIÓN Y POSIBLES MEJORAS.....	49
10. BIBLIOGRAFÍA.....	51
ANEXO 1: DATASHEETS.....	53
ANEXO 2: CÓDIGO DE ENLACE AL SERVIDOR (CD)	54
ANEXO 3: CÓDIGO DE LA ESTACIÓN METEOROLÓGICA (CD)	54
ANEXO 4: CÓDIGO DE LA PÁGINA WEB (CD)	54



LISTADO DE LAS FIGURAS

Figura 1: Observatorio sinóptico de superficie	5
Figura 2: Observatorio meteorológico aeronáutico	6
Figura 3: Estación termopluviométrica	6
Figura 4: Estación pluviométrica.....	7
Figura 5: Estación automática	7
Figura 6: Estación evaporimétrica.....	7
Figura 7: Estación con WiFi.....	9
Figura 8: Arduino Mega	12
Figura 9: Módulo GSM/GPRS	13
Figura 10: Jumpers para la utilización de los pines 7 y 8.....	13
Figura 11: Solar Charger Shield	15
Figura 12: Batería de 2000mAh	16
Figura 13: Placa solar	16
Figura 14: Circuito con el LM7805	16
Figura 15: Sensor LM35.....	17
Figura 16: NTC.....	18
Figura 17: Sensor de humedad del suelo	19
Figura 18: Sensor de lluvia.....	20
Figura 19-20: Sensor BMP180	20
Figura 21:RepRapDiscount Smart Contoller.....	21
Figura 22: RepRapDiscount Smart Controller, parte inferior	22
Figura 23: Esquema eléctrico de la placa de circuito impreso	25
Figura 24: Esquema eléctrico de la placa de circuito impreso del regulador de tensión.	25
Figura 25: Pistas, componentes y planos de masa de la placa de circuito impreso.....	26
Figura 26: Pistas, componentes y planos de masa de la placa del regulador	26
Figura 27: Placa de circuito impreso de los sensores soldada.....	27
Figura 28: Componentes de la placa de circuito impreso de los sensores.....	27
Figura 29: Placa de circuito impreso del regulador de tensión soldada	28
Figura 30: Componentes de la placa de circuito impreso del regulador de tensión	28
Figura 31: Señal del LM35 con un ruido de 100mV	30



Figura 32: Señal del LM35 tras hacer la placa de circuito impreso con un ruido de 64mV	30
Figura 33: Diagrama de flujo del programa principal	34
Figura 34: Diagrama de flujo de la interrupción	35
Figura 35: Diagrama de flujo de la función para enviar comandos AT al módulo GSM/GPRS	36
Figura 36: Diagrama de flujo de la función para enviar los datos al servidor.....	37
Figura 37: Página web: datos meteorológicos por meses.....	39
Figura 38: Página web: datos meteorológicos por sensores	39
Figura 39: Prototipo final	40
Figura 40: Estación meteorológica en una plantación trufera	45
Figura 41: Estación meteorológica en el jardín de la Universidad.....	46

LISTADO DE LAS TABLAS

Tabla 1: Comandos AT.....	14
Tabla 2: Consumo de la instrumentación de la estación meteorológica.....	41
Tabla 3: Prueba en una ventana.....	43
Tabla 4: Prueba en una plantación trufera.....	44
Tabla 5: Prueba en el jardín de la Universidad por la tarde.....	45
Tabla 6: Prueba en el jardín de la Universidad por la mañana.....	46
Tabla 7: Costes de la instrumentación del prototipo.....	48



1. INTRODUCCIÓN.

1.1. Definiciones previas.

Este proyecto versa sobre el diseño y desarrollo de una estación meteorológica alimentada con energía solar.

Una estación meteorológica es una instalación que tiene como función la medida y registro de distintos parámetros de la climatología como por ejemplo, temperatura, presión, humedad del aire, etc[1].

Hay diferentes tipos de estaciones meteorológicas. Algunas de ellas son:

- Observatorio sinóptico de superficie: Consta de una oficina donde se almacenan y envían los datos además de un jardín meteorológico donde se miden todo tipo de parámetros[2].



Figura 1: Observatorio sinóptico de superficie[3]

- Observatorio meteorológico aeronáutico: Están situados en aeródromos y obtienen variables meteorológicas útiles para la navegación aérea[4].



Figura 2: Observatorio meteorológico aeronáutico[5]

- Estación termopluviométrica: Toman datos de temperatura y precipitación únicamente[6].



Figura 3: Estación termopluviométrica[7]

- Estación pluviométrica: Obtiene datos de las precipitaciones a lo largo de un año o estación y se elabora un estudio[8].



Figura 4: Estación pluviométrica [9].

- Estación meteorológica automática: Realiza mediciones de distintos parámetros en lugares remotos sin necesidad de la acción humana[10].



Figura 5: Estación automática[11]

- Estación evaporimétrica: Mide la evaporación efectiva[12].



Figura 6: Estación evaporimétrica[13].



1.2. Objetivo del proyecto.

El objetivo de este Trabajo Fin de Grado es **el diseño y desarrollo de una estación meteorológica automática que sea autosuficiente energéticamente y la realización de un sistema de archivo de los datos meteorológicos capturados.**

A continuación vamos a aportar algunas directrices sobre cómo se alcanzará este objetivo. La autosuficiencia energética se conseguirá utilizando una placa solar con un módulo que cargue la batería con la energía excedente. Para este diseño se dispondrá de una serie de sensores de temperatura, presión, humedad de la tierra y de lluvia que proporcionarán las mediciones capturadas a una plataforma Arduino para su procesado. El archivo de los datos se hará mediante una tarjeta SD para guardar los datos de modo local y con un módulo GSM/GPRS (Global System for Mobile communications/ General Packet Radio Service) para enviar los datos a un servidor con el fin de que el usuario pueda verlos de una manera remota. Ambos componentes (tarjeta SD y módulo GSM/GPRS) serán gestionados desde la plataforma Arduino.

Así mismo, también es importante resaltar la utilidad real del objetivo planteado. La realización de este prototipo sería útil para hacer estudios de distintos cultivos y ver con qué condiciones meteorológicas se obtiene mayor rendimiento. De este modo, se podría llegar a automatizar dicha instalación para reproducir las condiciones de cultivo óptimas.

1.3. Estructura del proyecto.

El proyecto se ha dividido en una serie de etapas para la obtención del prototipo final:

- Aprendizaje de la programación de Arduino.
- Puesta en funcionamiento de los distintos sensores, módulo de comunicación(GSM/GPRS), módulo solar, almacenamiento de datos en la tarjeta SD y en el servidor.
- Diseño y desarrollo físico de la estación meteorológica.
- Prueba real del prototipo y captura de datos de muestra.
- Conclusiones y líneas de trabajo futuras.



2. TRABAJOS RELACIONADOS.

La plataforma Arduino es una de las más utilizadas para hacer proyectos con sensores como el que se ha propuesto en este TFG, por lo que ya se ha utilizado para la creación de estaciones meteorológicas.

Las han hecho de muchos tipos y los vamos a clasificar según el modo que tiene la estación meteorológica de recibir la alimentación eléctrica:

- Con alimentación de la red de una casa propia. Dentro de esta categoría encontramos:
 - Estación con visualización de los datos meteorológicos en el mismo lugar en el que se ubica[14].
 - Estación conectada a la red Ethernet o con WiFi y con una visualización de los parámetros en un servidor para poder consultarlos desde otro dispositivo sin tener que estar en el lugar de la estación[15].



Figura 7: Estación con WiFi[15].

- Estación con visualización en el mismo lugar pero con un almacenamiento de los datos en una tarjeta SD para tener un historial y poder hacer estudios de la climatología de un lugar[16].



- Con alimentación solar:
 - Estación que utiliza un módulo radiotransmisor que envía los datos recogidos en un determinado lugar a un receptor que se puede ubicar en un sitio más accesible. En esta estación se necesitan dos placas de Arduino, una para controlar los sensores de medida y el módulo emisor y la segunda para recibir la información y tratarla como desee el usuario[17].
- Con alimentación de baterías:
 - Este tipo también es inalámbrico. La estación envía los datos mediante Bluetooth a un smartphone el cual tiene que tener instalada una aplicación para recibirlos y visualizarlos posteriormente. Al utilizar este método de envío, la estación y el smartphone deben estar juntos o distanciados unos pocos metros ya que si no se perderá la comunicación entre ellos[18].

La estación realizada en este Trabajo Fin de Grado sería **novedosa** ya que no se ha encontrado ninguna con esta placa de desarrollo que tenga las mismas características. Sí que hay con alimentación solar, o que envíe los datos al servidor, pero no con **ambas características**. Además, las estaciones meteorológicas que envían los datos a un servidor lo hacen mediante red Ethernet o con WiFi pero no mediante la **red móvil**.



3. INSTRUMENTACIÓN UTILIZADA.

3.1. Arduino Mega.

Arduino es una plataforma de desarrollo de hardware y software compuesta por una placa que integra un microcontrolador y un entorno de desarrollo donde se programa la placa[19].

La elección de la placa Arduino Mega en lugar de otras placas de Arduino (como por ejemplo el UNO, NANO, Genuino, Zero, Micro, etc.) se debe a que ésta es una de las placas que mayor memoria tiene (256KB) y también ofrece una cantidad considerable de pines de entrada y salida (54 digitales y 16 analógicos). Algunos de los pines están dedicados a funciones específicas como 4 puertos serie, 6 interrupciones externas, PWM (Pulse-Width Modulation), SPI (Serial Peripheral Interface) e I²C (Inter-Integrated Circuit).

Inicialmente en este proyecto se utilizó la placa Arduino UNO que tiene una memoria de 32KB, 14 pines de entradas y salidas digitales y 6 analógicas. También tiene pines dedicados a puertos serie, interrupciones externas y otras funcionalidades. Sin embargo, esta placa no servía para la aplicación de este TFG ya que se quedaba corta tanto en cantidad de memoria como de número de pines digitales.

La motivación principal de utilizar Arduino frente a otras placas de desarrollo como Raspberry Pi es su menor consumo ya que no dispone de sistema operativo. Esto es especialmente importante ya que la estación meteorológica se tiene que alimentar mediante energía solar. Otra característica por la que se eligió esta placa es su facilidad de programación y de puesta en marcha de los distintos sensores y módulos.

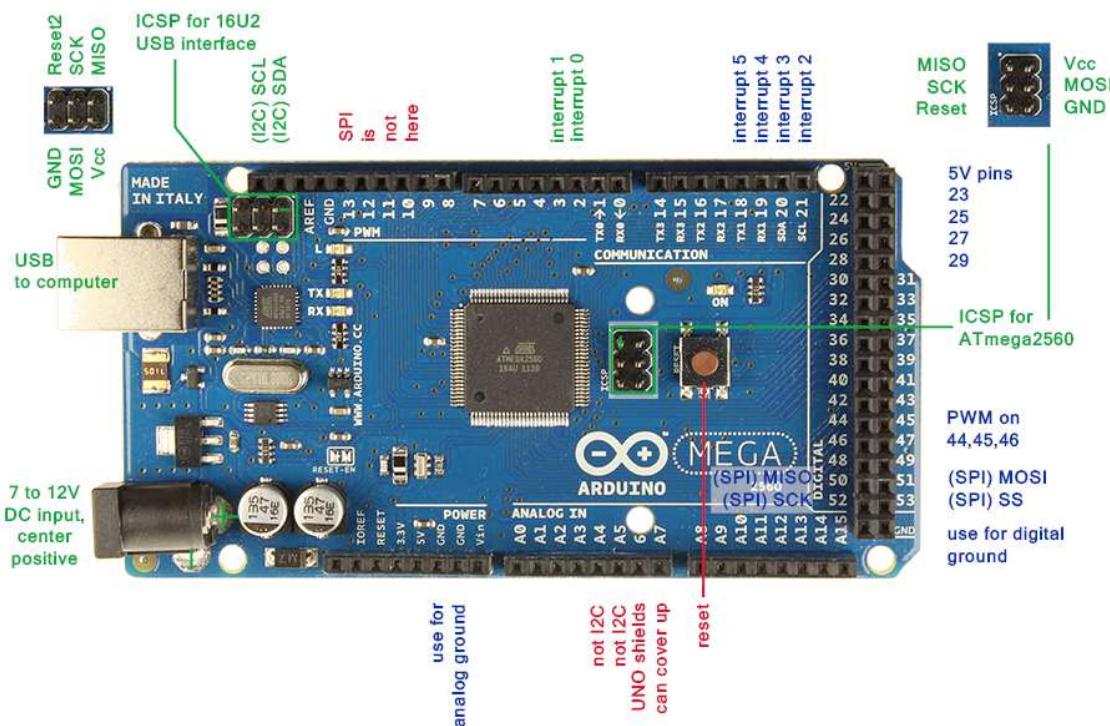


Figura 8: Arduino Mega [20].

3.2. GSM GPRS Arduino Shield.

El GSM/GPRS Arduino Shield es un módulo de comunicaciones que funciona mediante la red de telefonía móvil, por lo que necesita una tarjeta SIM. De este modo, la estación se podrá poner en un lugar alejado de los lugares poblados (teniendo en cuenta que tiene que haber cobertura móvil) sin necesidad de cableado como sí se necesitaría con un dispositivo de comunicación basado en Ethernet.

Este módulo tiene dos formas de comunicación, mediante GSM y mediante GPRS. Vamos a describir cada una de ellas:

- El GSM (Global System for Mobile communications) es más utilizado para la transmisión de voz aunque también se puede utilizar para datos y SMS pero a una velocidad muy reducida, 9 KB/s.
- El GPRS (General Packet Radio Service) se basa en la transmisión de paquetes y ofrece una mayor eficiencia en el envío de datos. Su velocidad asciende a 171KB/s[21].

La transmisión de datos entre el Arduino y el módulo GSM/GPRS se hace mediante comunicación serie. Además, hay que alimentarlo a 5 voltios, por lo que se



necesitan cuatro conexiones entre el módulo y el Arduino. Para utilizar los pines 7 y 8 para la comunicación del GSM con la placa Arduino hay que poner dos *jumpers* como se ven en la Figura 10.

Este módulo se programa mediante un conjunto de comandos *Hayes*, más conocidos como comandos AT (Atención), que es un lenguaje que se creó para configurar y parametrizar módem[22].

Por último, hay que decir que este módulo necesita ser encendido manualmente desde el botón o si se desea hacer por software habría que soldar un condensador en el pad "R13" y añadir un cable desde Arduino al pin 9 del módulo para mandarle un pulso de 1 segundo. Con esto se encendería sin necesidad de pulsar el botón.



Figura 9: Módulo GSM/GPRS[21].

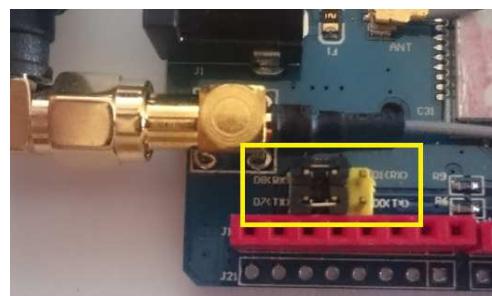


Figura 10: Jumpers para la utilización de los pines 7 y 8[21].



Los comandos AT más comunes se muestran en la siguiente tabla:

AT	Comprueba el estado del módulo
AT+CPIN = "XXXX"	Introducir el PIN de la SIM
AT+CREG	Comprueba la conexión a la red
ATDXXXXXXXXX;	Realiza una llamada
ATH	Finaliza la llamada
AT+CMGF = 1	Configurar el modo texto para enviar o recibir mensajes. Devuelve ">" como inductor
AT+CMGS = "XXXXXXXX"	Nº al que vamos a enviar el mensaje
AT+CLIP = 1	Activamos la identificación de la llamada
AT+CNMI = 2,2,0,0,0	Configuramos el módulo para que muestre los SMS por el puerto serie
AT+CGATT = 1	Conectamos a la red GPRS
AT+CSTT = "APN", "usuario", "contraseña"	Definimos APN. usuario y contraseña
AT+CIICR	Activamos el perfil de datos inalámbrico
AT+CIFSR	Obtenemos nuestra IP
AT+CIPSTART = "TCP", "direcciónIP", "puerto"	Indicamos el tipo de conexión, dirección IP y puerto al que realizamos la conexión
AT+CIPSEND	Preparamos el envío de datos. Devuelve ">" como inductor
AT+CIPCLOSE	Cerramos la conexión
AT+CIPSHUT	Cierra el contexto PDP del GPRS
AT+CGPSPWR = 1	Activar el GPS
AT+CGPSSTATUS?	Comprueba que el GPS ha encontrado la red
AT+CGPSINF = 0	Obtiene los datos del GPS: Modo, latitud, longitud...
AT+CGPSOUT = 32	Obtiene los datos del GPS usando la especificación NMEA: Horario UTC, estado, latitud, longitud, velocidad en nudos...
AT+CGPSPWR = 0	Cerrar el GPS

Tabla 1:Comandos AT [23].



Por ejemplo, para enviar las medidas de los sensores al servidor una vez el módulo está conectado a la red móvil se haría con los siguientes comandos:

```
"AT+CIPSTART=\"TCP\",\"155.210.68.182\",\"80\""
```

```
"AT+CIPSEND=GET
```

```
meteorem_jorge_eupt.php?LM35=16.49&NTC=17.05&humedad=10.54&lluvia=0.26  
&presion=995.61&temperatura=16.95"
```

De esta manera, estos datos se almacenarán en la tabla de la base de datos que se ha configurado. En el apartado 5.2 se explicará cómo se hace para enviar un dato al servidor de una manera más extensa.

3.3. Módulo solar, batería y placa solar.

El módulo solar es un sistema de alimentación mediante baterías LiPo. Permite mantenerlas cargadas ya que se puede conectar una placa solar u otro tipo de alimentación[24]. El sistema siempre será alimentado a través de la batería aunque se coloque una placa solar. Toda su energía se dedicará a cargar la batería.

Las características que tiene son:

- Voltaje operativo de la batería: 3,0 - 4,5V
- Voltaje operativo del USB: 4,75 - 5,25V
- Voltaje operativo solar: 4,8 - 6,0V
- Corriente máxima con batería: 3W (600mA a 5V)

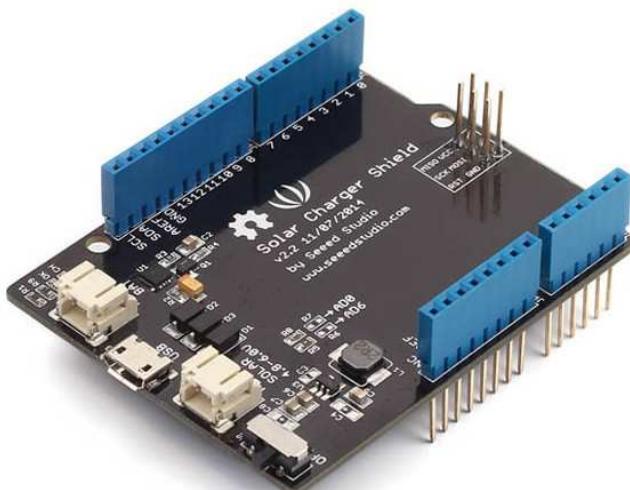


Figura 11: Solar Charger Shield [24].



La batería LiPo elegida ha sido una de 2000mAh para que tenga suficiente autonomía en caso de que durante un tiempo no se pueda cargar debido a la situación climatológica.



Figura 12: Batería de 2000mAh.

La placa solar podrá suministrar como máximo 1.5W y 12V con unas dimensiones de 145x145mm. Ya que puede suministrar hasta 12 V hay que realizar un circuito (Figura 14) para regular la tensión y que no sobrepase los 6V que admite el módulo solar. Para llegar a ese objetivo se ha puesto un LM7805 entre la salida de la placa solar y la entrada al módulo solar. Este circuito integrado es un regulador de tensión que admite una entrada entre 7 y 25V y proporciona una salida de 5V, asegurándonos así un margen para no dañar el módulo.

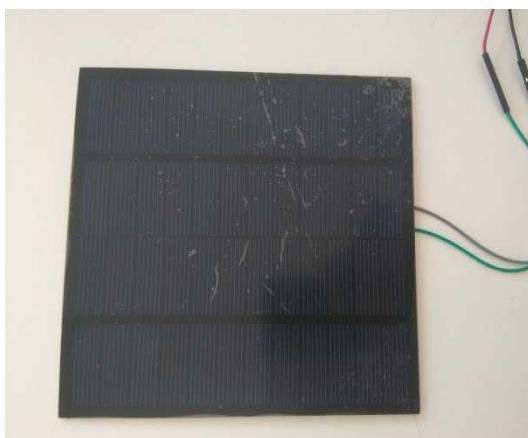


Figura 13: Placa solar.

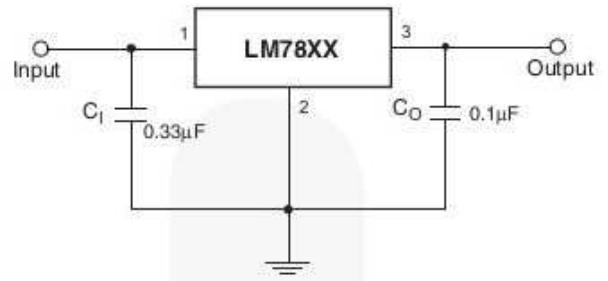


Figura 14: Circuito con el LM7805[25].



3.4. Sensor de temperatura LM35.

El sensor LM35 es un circuito integrado en el que el voltaje de salida varía linealmente de manera proporcional a los grados centígrados que haya en el ambiente. Cada grado varía 10mV en la salida. Este sensor tiene un rango de temperaturas desde -55°C hasta 150°C, que es mucho mayor al rango que se necesita. La precisión que ofrece es de $\pm 0.5^{\circ}\text{C}$. Además, tiene la ventaja de que su consumo es de 60 μA (Texas Instruments).



Figura 15: sensor LM35[26].

3.5. Sensor de temperatura NTC.

Una NTC (Negative Temperature Coefficient) es un tipo de termistor, que son sensores de temperatura de tipo resistivo. La NTC disminuye la resistencia al aumentar la temperatura. Este sensor necesita de una resistencia externa para así hacer un divisor resistivo y poder medir variaciones de voltaje en función de variaciones de resistencia. La variación de la resistencia, en función de la temperatura, de este sensor no es lineal sino exponencial. El rango de temperatura que admite el sensor está entre -70 y 500°C, mucho más de lo necesario.

La curva de resistencia frente a temperatura corresponde con esta fórmula:

$$R_T = R_0 e^{B * \left(\frac{1}{T} - \frac{1}{T_0} \right)}$$

R_T : Resistencia que presenta la NTC a la temperatura del ambiente.

R_0 : Resistencia a una temperatura de referencia.

B: Índice de sensibilidad propio del termistor.



De esa fórmula lo que hay que calcular es la temperatura ya que la resistencia de la NTC se puede obtener a partir del voltaje que cae en ella medido a través de Arduino. Aun así, esta expresión es aproximada por lo que hay error en la temperatura obtenida. Para obtener medidas más precisas se midió la resistencia a 25°C, ya que, aunque estaba en la hoja de características del componente tenía cierto error. Otra opción para su uso es linealizar el sensor con una resistencia en serie de valor adecuado. El sensor se puede calibrar con dos puntos (línea recta) o con más puntos y ajustar a un polinomio para medir con más precisión la temperatura. Se intentó esta opción pero no se pudo llevar a cabo ya que no se pudieron tomar medidas en todo el rango de temperaturas, así que se dejó con la fórmula tratada anteriormente.



Figura 16: NTC [27].

3.6. Sensor de humedad del suelo.

Este sensor mide la humedad del suelo por la variación de conductividad. Si se pone el sensor al aire libre la salida es de 5V que es el voltaje de alimentación ya que la conductividad del aire es mínima, y si se pone en un vaso con agua la salida sería de 0V ya que la conductividad aumenta mucho en el agua.

Una patilla de las cuatro que tiene es digital y se puede configurar con un potenciómetro que tiene la placa como alarma si se desea activar avisos sobre si la humedad está por debajo o por encima de cierto valor. Esto es muy útil para tener un riego automatizado y que el suelo siempre esté con un porcentaje de humedad necesario para ciertas plantas.

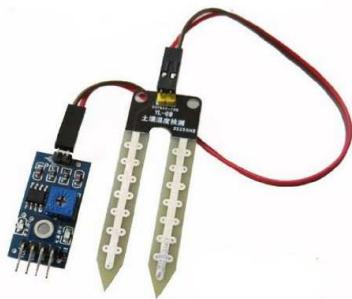


Figura 17: Sensor de humedad del suelo [28].

3.7. Sensor de lluvia.

Este sensor mide el área de la placa que está mojada por la variación de conductividad.

El funcionamiento del sensor es sencillo. Se basa en dos contactos unidos a unas pistas entrelazadas entre sí separadas una cierta distancia, sin ningún contacto. Al depositarse agua sobre la superficie, se pone en contacto eléctrico a ambos conductores, lo que puede ser detectado por el sensor [29].

Debido a ello, el sensor tiene mayor impedancia entre las pistas al estar seco y la salida por lo tanto es de 5V. Al estar totalmente mojado la impedancia entre las pistas será muy baja, entonces la salida será prácticamente 0V.

Al igual que el sensor de humedad del suelo, tiene una patilla digital que se puede configurar como alarma con un potenciómetro en la propia placa, si se desea que avise cuando cierta área del sensor se encuentre mojada o seca.



Figura 18: Sensor de lluvia [30].

3.8. Sensor de presión y temperatura (BMP180).

El sensor BMP180 mide la presión y la temperatura conjuntamente. Al tener esos dos parámetros también puede calcular la altura relativa a un punto mediante la diferencia de presión.

Este sensor transmite los datos mediante comunicación I²C que es un bus serie de datos que contiene dos líneas de transmisión (SDA: Serial Data y SCL: Serial Clock) más las de alimentación (3.3V) y de tierra. El I²C está diseñado como un bus maestro-esclavo. En este caso el Arduino (maestro) inicializa la recogida de datos de los dos sensores (esclavos) que se comunican de esta manera[31].

El BMP180 mide un rango de temperaturas desde -40 a 85°C y de presión desde 300 a 1100hPa. Los datos meteorológicos de las potenciales zonas a monitorizar se encuentran dentro de esos rangos, por lo que este sensor sirve.

Además, la precisión es muy buena en el caso de la presión y admisible en el caso de la temperatura, de hasta ±1hPa y de ±1°C.



Figura 19-20: Sensor BMP180[32].



3.9. LCD y tarjeta SD.

La LCD elegida para mostrar los valores en el lugar donde se encuentra la estación es la RepRapDiscount Smart Controller. Su pantalla tiene cuatro filas y 20 columnas.

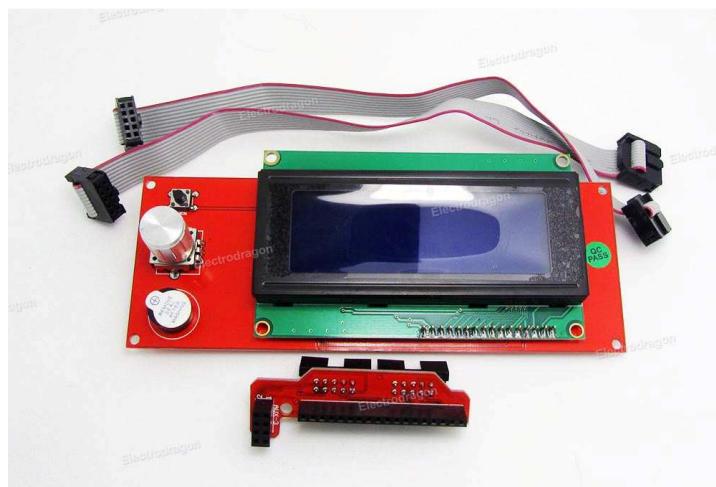


Figura 21:RepRapDiscount Smart Controller[33].

La conexión con el Arduino se hace con 10 cables: uno de alimentación a 5V, dos a masa, cuatro para los datos que se quieran mostrar, dos de control (reset y enable) y el último a un potenciómetro para controlar el contraste de la pantalla. Este potenciómetro lo lleva incorporado la LCD así que no es necesario ponerle uno externo. Todas estas conexiones se hacen desde un conector que la pantalla tiene en la parte posterior como se muestra en la Figura 22.

En la parte posterior (Figura 22) también tiene un módulo donde se puede insertar una tarjeta SD. Esto es una ventaja ya que dicha tarjeta ha sido utilizada para guardar los datos en modo local. La tarjeta SD se comunica mediante SPI por lo que solo necesita de cuatro conexiones de las cuales son dos de datos (MISO, MOSI), otra de reloj(CLK) y la última el CS. La alimentación de la SD es la misma que la de la LCD por lo que no es necesario alimentarla aparte.

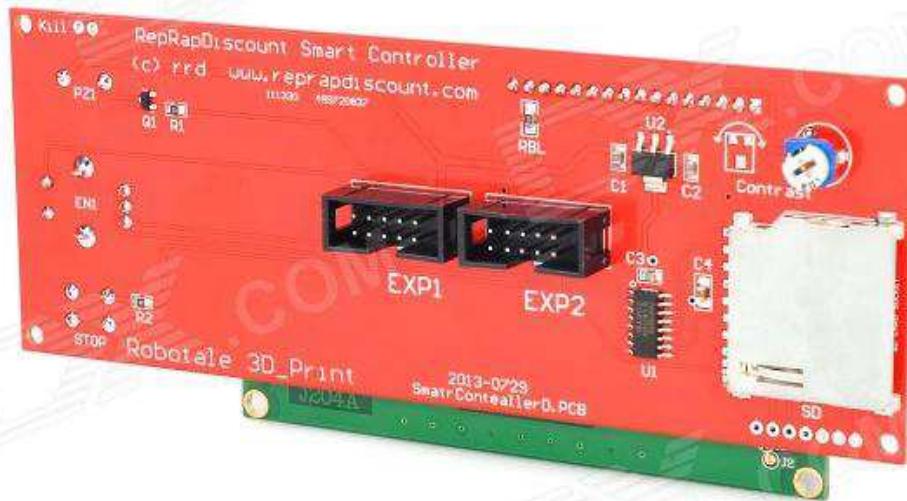


Figura 22: RepRapDiscount Smart Controller, parte inferior[34].



4. PLACA DE CIRCUITO IMPRESO.

Los circuitos de los diferentes sensores se han implementado en un placa de circuito impreso con el fin de mejorar el diseño del prototipo y de disminuir las interferencias electromagnéticas (EMI) con las técnicas que se explicarán a continuación. De esta manera se suprime gran parte del cableado.

Las medidas tomadas frente a las interferencias electromagnéticas[35] han sido:

- Se han creado tres planos de masa distintos para que el LM35 y la NTC estén aislados ya que presentaban una mayor EMI en la protoboard donde se montaron inicialmente. Las masas se han hecho en plano ya que así la corriente toma el camino de menor impedancia siendo así la solución óptima. Este punto es clave para solucionar los problemas de EMI.
- Las pistas deben ser cortas y anchas y evitar de esta manera pistas largas en paralelo. También hay que tener en cuenta que los cambios de dirección no se deben hacer mediante ángulos de 90° ya que los cambios fuertes de impedancias provocan reflexiones. De este modo, los cambios de dirección se deben hacer en la medida de lo posible con dos ángulos de 45°.
- Desacoplo de la alimentación: Es necesario hacer un desacoplo a nivel de placa y a nivel de dispositivo. Hay que evitar que los picos de corriente recorran un camino largo. Para ello se han puesto condensadores de desacoplo entre alimentación y masa en todos los sensores menos en la NTC ya que llevan circuitos integrados y éstos en las commutaciones pueden requerir altos niveles de intensidad en muy poco tiempo. Este condensador se ha puesto cercano al sensor ya que así se reducen las áreas de los bucles.
- Después de soldar los componentes, se han cortado las patillas lo más cortas posible ya que si no pueden actuar de antena.

Antes de hacer la placa de circuito impreso hay que realizar el esquema del circuito (Figura 23) en el programa que vamos a utilizar (Eagle). Para ello se necesitaban las librerías de los componentes utilizados. Sin embargo, hubo un problema con los sensores de humedad y lluvia ya que no se encontraron librerías asociadas a esos



componentes. La solución fue utilizar la librería del sensor BMP180, que ya se había utilizado, ya que tiene igual número de patillas y a la misma distancia unas de otras.

En el circuito del termistor NTC aparece un diodo Zener, el cual se ha puesto para reducir el ruido. Se explica en el apartado 5.1 de la memoria.

Además, se ha realizado una segunda placa de circuito impreso para la regulación de tensión de entrada al módulo solar ya que este dispositivo no admite los 12V que ofrece la placa solar. Esta placa también sigue las medidas contra las interferencias electromagnéticas nombradas anteriormente, aunque en este caso solo se ha necesitado un plano de masa. En la Figura 24 se muestra el esquema del circuito.

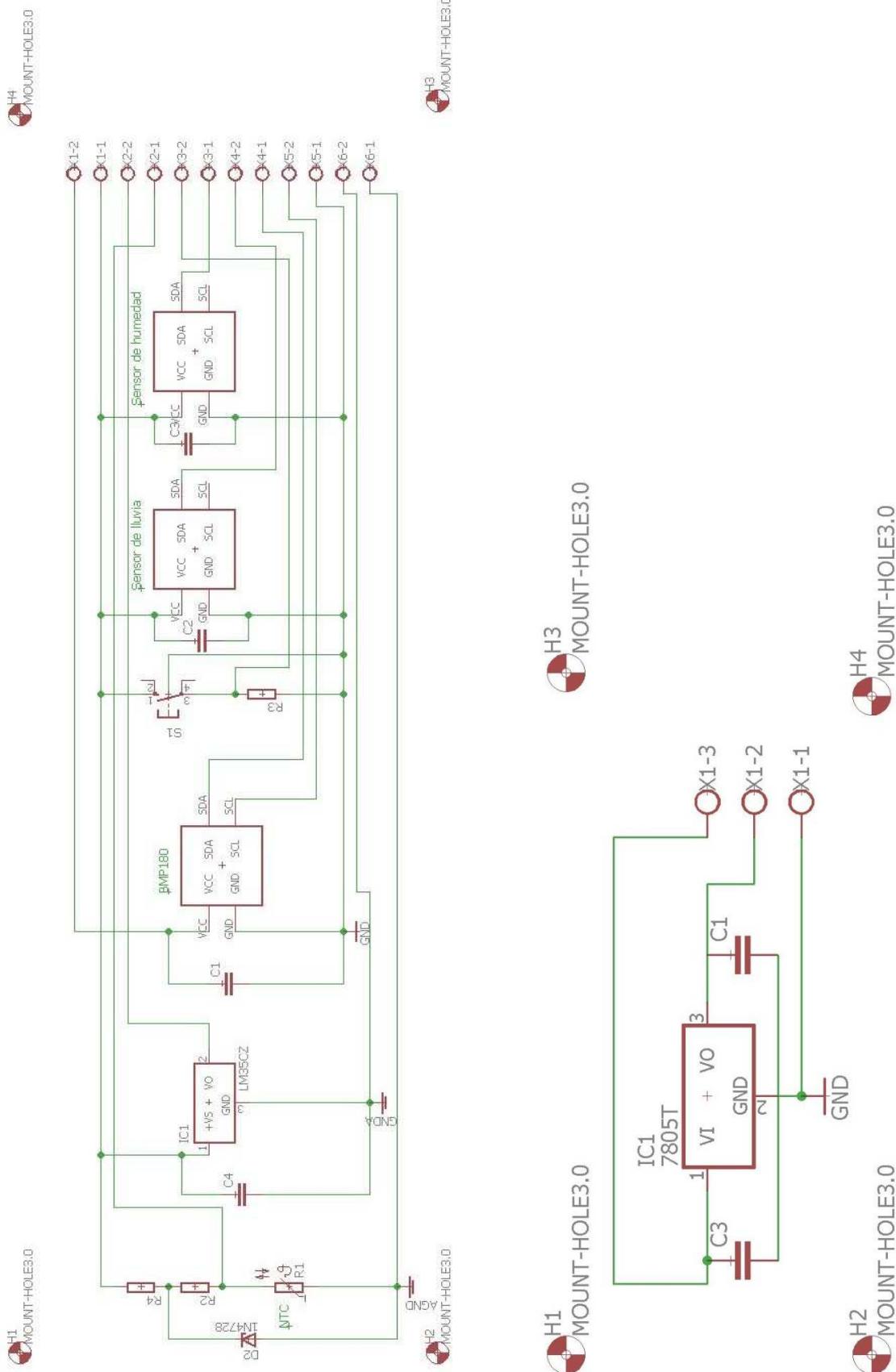


Figura 23: Esquema eléctrico de la placa de circuito impreso.

Figura 24: Esquema eléctrico de la placa de circuito impreso del regulador de tensión.



Una vez realizado el esquema, se elaboró otro archivo donde se ordenaron los componentes en la placa y se dibujaron las pistas. Por último, se realizaron los planos de masa. En las siguientes figuras se muestran las pistas, los componentes y los planos de masa que son cada polígono encerrado por la línea de puntos.

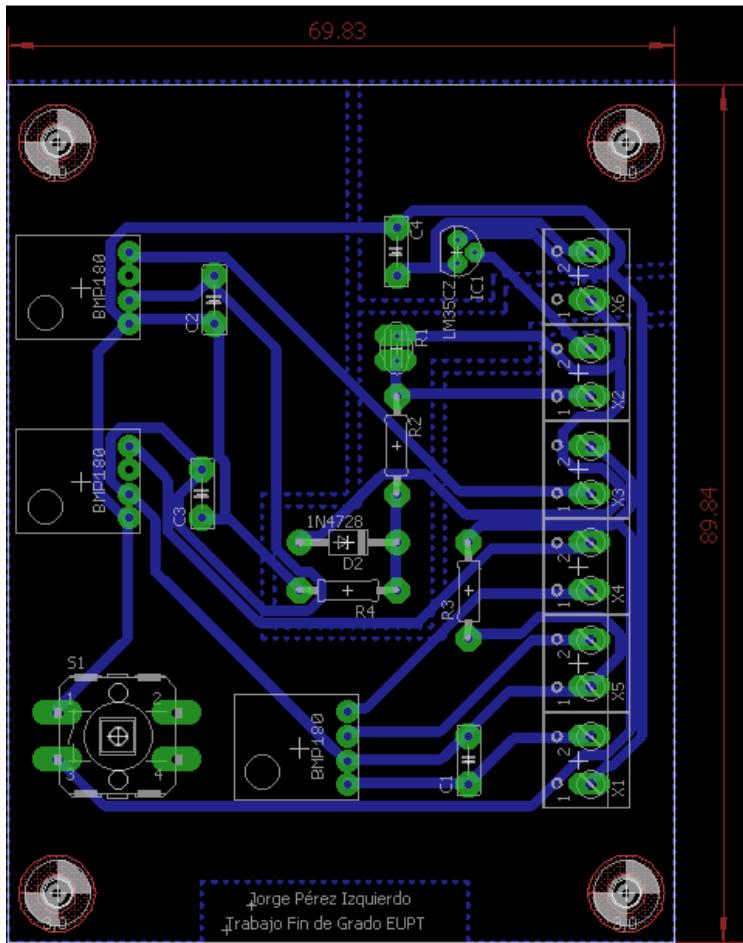


Figura 25: Pistas, componentes y planos de masa de la placa de circuito impreso.

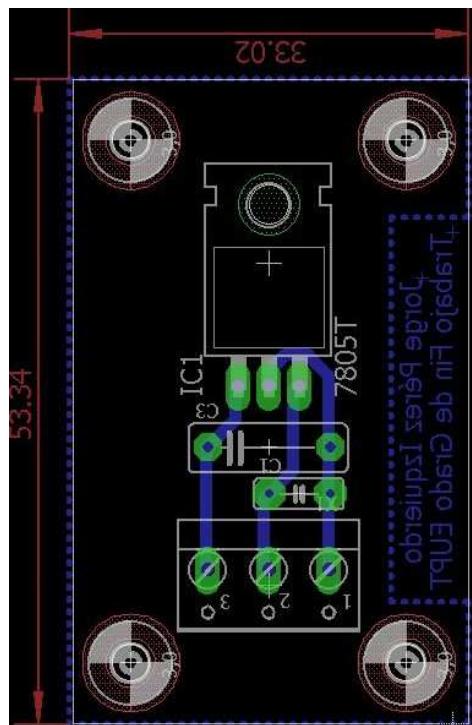


Figura 26: Pistas, componentes y planos de masa de la placa del regulador.

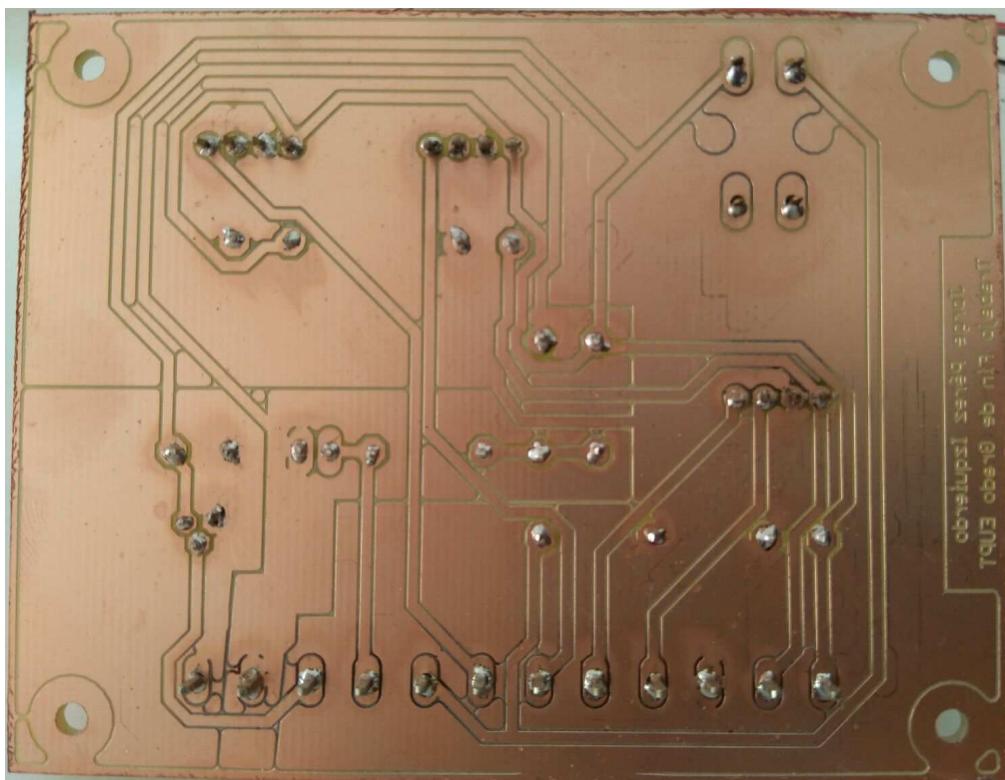


Figura 27: Placa de circuito impreso de los sensores soldada.



Figura 28: Componentes de la placa de circuito impreso de los sensores.

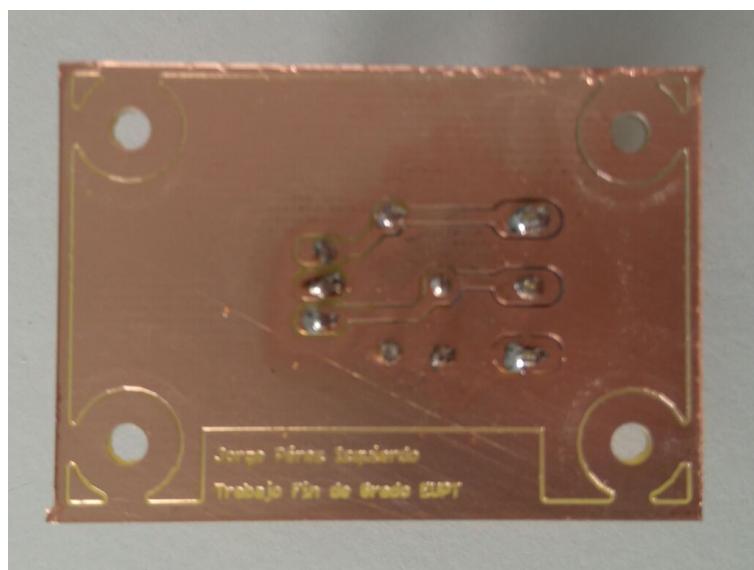


Figura 29: Placa de circuito impreso del regulador de tensión soldada.



Figura 30: Componentes de la placa de circuito impreso del regulador de tensión.



5. DISEÑO Y DESARROLLO DE LA ESTACIÓN.

En este apartado del trabajo se explicará detalladamente cómo se ha desarrollado la estación meteorológica, tanto el diseño hardware como el diseño software. Como se ha dicho anteriormente en el apartado 1.2, todo el diseño se hará conforme a las características del Arduino Mega 2560 ya que para este proyecto se pensó que era la mejor placa de desarrollo.

Para entender mejor el diseño que se ha realizado, lo primero que hay que tener en cuenta es el funcionamiento de esta estación meteorológica.

Como ya se ha dicho en el apartado 1.2, se trata de una estación meteorológica automática. El Arduino Mega 2560 es el encargado de recoger datos de temperatura (con los sensores LM35, NTC y BMP180), de humedad de la tierra, lluvia y presión. Estos datos se muestran en la LCD en distintos menús a los que se accede mediante un pulsador. Hay cuatro menús, en los que se muestra:

- Temperatura del LM35.
- Temperatura de la NTC.
- Humedad de la tierra y lluvia.
- Presión y temperatura con el sensor BMP180.

Todos estos modos tienen en común que en la primera línea de la LCD se escribe la fecha y la hora provenientes del shield GSM/GPRS que también lleva un reloj. En esta visualización, cada vez que activamos el pulsador, se recoge el dato del sensor o sensores necesarios para el modo y se actualiza la fecha y la hora.

Además, de manera automática, los datos son recogidos y almacenados cada media hora en la tarjeta SD y enviados al servidor remoto.

5.1.DISEÑO HARDWARE.

El diseño hardware se implementó en primera instancia en una protoboard. Así podríamos ver el comportamiento de los distintos sensores y las posibles mejoras de su funcionamiento.

Al comenzar el proyecto se fue probando cada sensor por separado y se vio que funcionaban perfectamente. En cambio, al ir poniendo varios sensores a la vez se detectaron algunos errores en las medidas ya que había mucho ruido proporcionado tanto por los sensores como por el propio Arduino. Esto se solucionó poniendo un filtro



paso-bajo software que se explicará posteriormente. Este problema era crítico en dos de los sensores de temperatura (NTC, LM35). Por ejemplo, en el LM35 un ruido de pico a pico de 100mV (Figura 31) podría variar la temperatura real $\pm 5^{\circ}\text{C}$ que es un error inadmisible en una estación meteorológica. También en la NTC, ya que una variación del voltaje que cae en dicho sensor hace variar la resistencia calculada en función a ello y por consiguiente ese error se va acumulando en los distintos cálculos posteriores. En este sensor se optó por poner un diodo Zener de 3,9V para alimentarlo reduciendo en 80mV el ruido en la entrada al sensor. El ruido en los demás sensores analógicos no es tan crítico ya que una variación de 100mV en el sensor de humedad y de lluvia solo cambiaría en $\pm 1\%$ su medida. Por este motivo todos ellos tienen una masa común y cada sensor tratado anteriormente (LM35 y NTC) tiene una masa distinta con el fin de disminuir ese ruido en ellos.

La placa de circuito impreso se creó con el fin de disminuir el ruido así como para disponer de un diseño más profesional.

En las siguientes imágenes se hace una comparativa del ruido en el sensor LM35 antes y después de hacer la placa de circuito impreso. Como se puede apreciar, el ruido disminuye considerablemente pero aun así sigue siendo crítico y se necesita un filtro software.

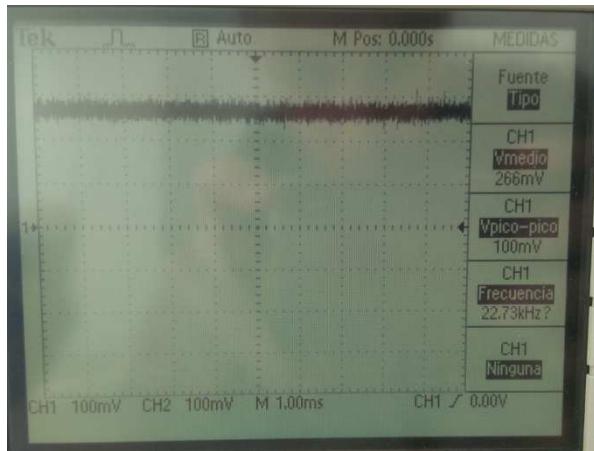


Figura 31: Señal del LM35 con un ruido de 100mV.

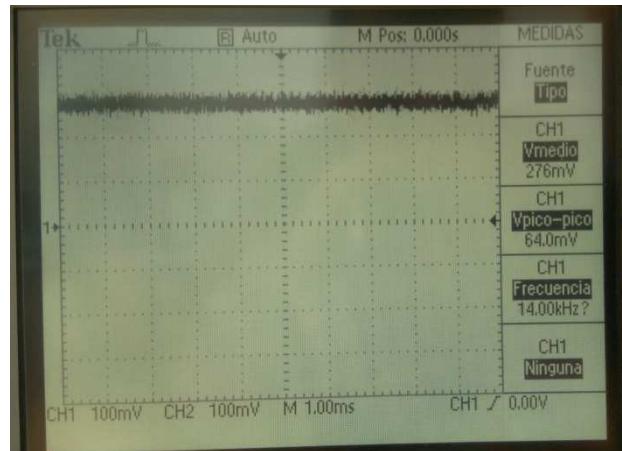


Figura 32: Señal del LM35 tras hacer la placa de circuito impreso con un ruido de 64mV.



5.2. DISEÑO SOFTWARE.

El software de la placa Arduino se ha hecho con el entorno asociado Arduino IDE, ya que nos proporciona muchas facilidades para crear el código y transferirlo a la placa de desarrollo.

5.2.1. Librerías.

En este proyecto se han utilizado diferentes librerías que nos han ofrecido herramientas para hacer el código más sencillo. En concreto:

- LiquidCrystal: Esta librería se ha incorporado para la utilización de la LCD. Con esta librería la configuración y escritura en la pantalla resulta muy sencilla ya que incluye funciones para ello. Entre ellas están:
 - LiquidCrystallcd(): En esta función hay que poner los pines digitales utilizados para la comunicación con la LCD (RS (reset), enable y las 4 líneas de datos (D4, D5, D6, D7)).
 - lcd.begin(): En esta función se indican las columnas y filas que tiene la LCD.
 - lcd.print(): Esta función sirve para escribir en la pantalla tanto variables como texto directamente.
 - lcd.setCursor(): Esta función sirve para elegir en qué columna y fila se desea comenzar a escribir.
 - lcd.clear(): Esta función borra todos los caracteres que están escritos en la pantalla.
- SFE_BMP180: Esta librería es para la utilización del sensor BMP180. Nos encapsula toda la parte de comunicación y de calibración, que es más compleja dado que este sensor tiene una comunicación mediante I²C. Las funciones utilizadas son:
 - pressure.begin(): Esta función devuelve un 1 si se ha inicializado la comunicación y un 0 si no lo ha hecho, por lo que es útil para detectar errores.
 - startTemperature(): Con esta función empieza la lectura de la temperatura. Devuelve un número, que posteriormente se utilizará como retardo, o un 0 si ha habido algún error en la comunicación.



- getTemperature(): Esta función guarda la temperatura en una variable externa que hay que pasarle. Previamente hay que iniciar la lectura. La función devuelve un 1 si ha tenido éxito la recuperación de la temperatura o un 0 si no lo ha tenido.
- startPressure() y getPressure() tienen la misma función que startTemperature() y getTemperature() pero varían las variables que necesitan. La función startPressure() permite cambiar la resolución de la presión poniendo un 0 entre los paréntesis, que sería la peor resolución posible, hasta un 3 que sería la mayor. La función getPressure() guarda el resultado de la presión en una variable y para calcularla necesita saber la temperatura anteriormente calculada. Al igual que las funciones de temperatura devuelven un 1 si ha tenido éxito la toma de datos o un 0 si no la ha tenido.
- Wire: Esta librería no se utiliza directamente pero la librería del sensor BMP180 hace uso de ella, así que es necesario incluirla para el funcionamiento del programa.
- SD: Esta librería se utiliza para escribir en un archivo de la tarjeta SD. Las funciones necesarias para ello son:
 - SD.begin(): Con esta función se comienza la comunicación, devolviendo un 1 si se ha realizado con éxito y un 0 si, al contrario, no lo ha hecho.
 - SD.open(): Esta función abre el archivo donde se quieren leer o escribir los datos. En nuestro caso solo se realizará la escritura, por lo que hay que indicarle, a parte del nombre del archivo donde se quiere escribir, que solo se va a hacer dicha acción. Devuelve un 1 si se ha abierto el archivo o un 0 si no se ha podido abrir.
 - Archivo.close(): Esta función cierra el archivo guardando los nuevos datos escritos en él.
 - Archivo.print(): Esta función se utiliza para escribir en el archivo.
- SofwareSerial: Esta librería permite configurar dos pines digitales para que se comporten como un puerto serie, que es necesario para la



comunicación con el módulo GSM/GPRS. En nuestro caso las funciones que se utilizan son:

- SIM900.available(): Esta función devuelve un 1 si está disponible el puerto configurado o un 0 si no lo está.
- SIM900.read(): Esta función lee los datos que recibe por el puerto que se ha configurado.
- SIM900.print(): Se envía un dato por el puerto configurado.

5.2.2. Filtros software.

Al realizar las mediciones de los distintos sensores de la estación meteorológica se comprobó que había mucha variación entre medidas de un mismo sensor en momentos poco distantes en el tiempo. Tras ver este suceso, se observó con el osciloscopio la señal que tenía la salida de los sensores y se vio que había mucho ruido.

Para que las medidas sean lo más próximas a la realidad posible, se decidió hacer un filtro paso bajo por software que consiste en realizar cuantiosas medidas en un corto periodo de tiempo y hacer la media entre todas ellas. De este modo se disminuye el error considerablemente. Por ejemplo, en el sensor LM35 que es el más sensible, tener un ruido de 100mV de pico a pico podría variar de una medida a otra 10°C y esto no se puede permitir.

Realizar este tipo de filtros ralentiza la ejecución del código ya que tiene que realizar bucles con muchas iteraciones. En este caso, el tiempo de ejecución no es crítico, por lo que ésta era la solución más adecuada para este proyecto.

Además, en la interrupción que activaba el pulsador se tuvo que poner un retardo de 400ms ya que al pulsarlo existen rebotes. En este caso la interrupción se activa con un flanco de subida, por lo que al existir rebotes entraba varias veces en ella y por lo tanto pasaba a un menú que no era el consecutivo.

5.2.3. Diagramas de flujo.

Diagrama de flujo del *setup* y del programa principal:

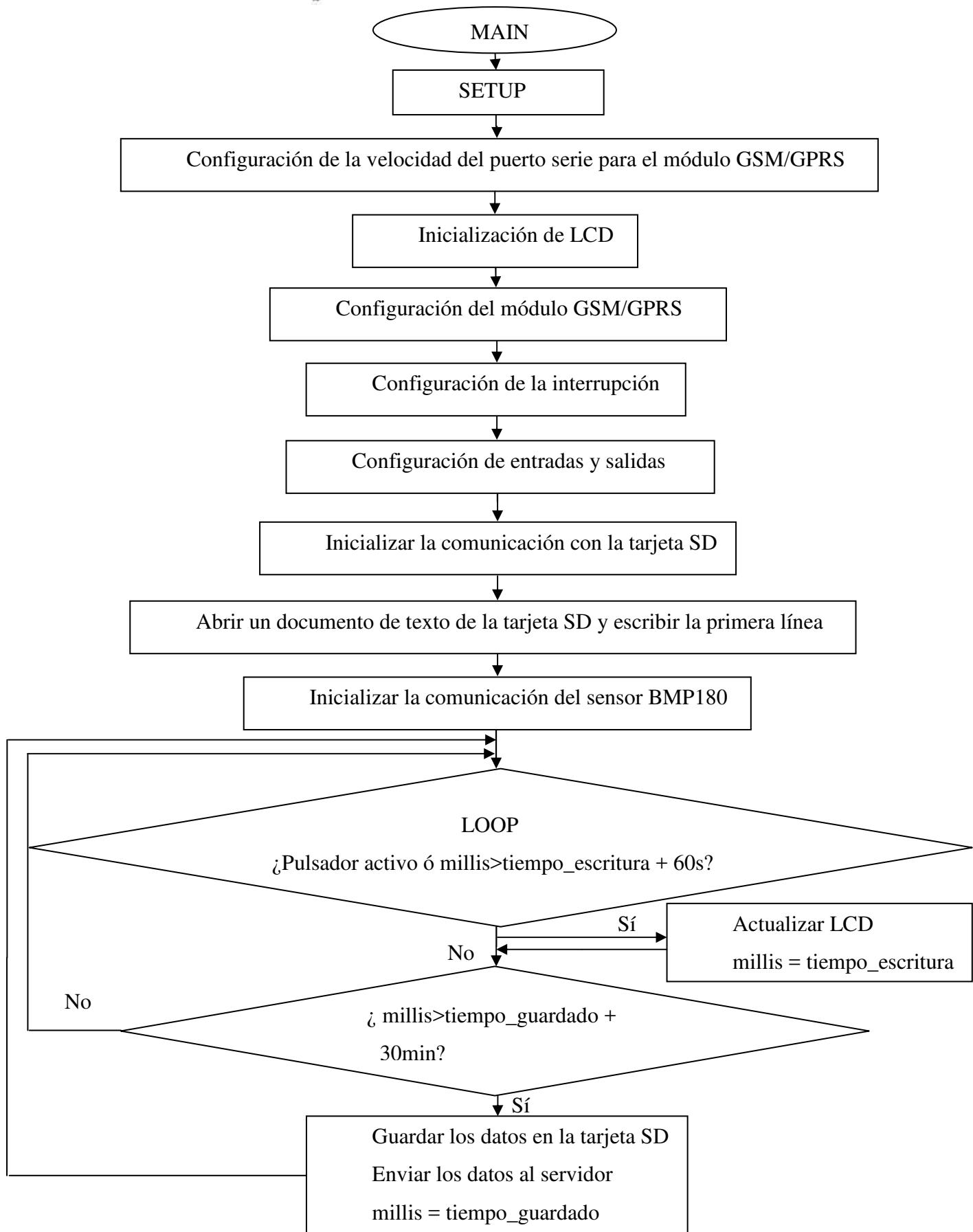


Figura 33: Diagrama de flujo del programa principal.



En la Figura 34 se muestra el diagrama de flujo de la interrupción:

La variable a lleva el control de los modos de visualización. A cada valor de a , que se incrementa al activar el pulsador, le corresponde un modo.

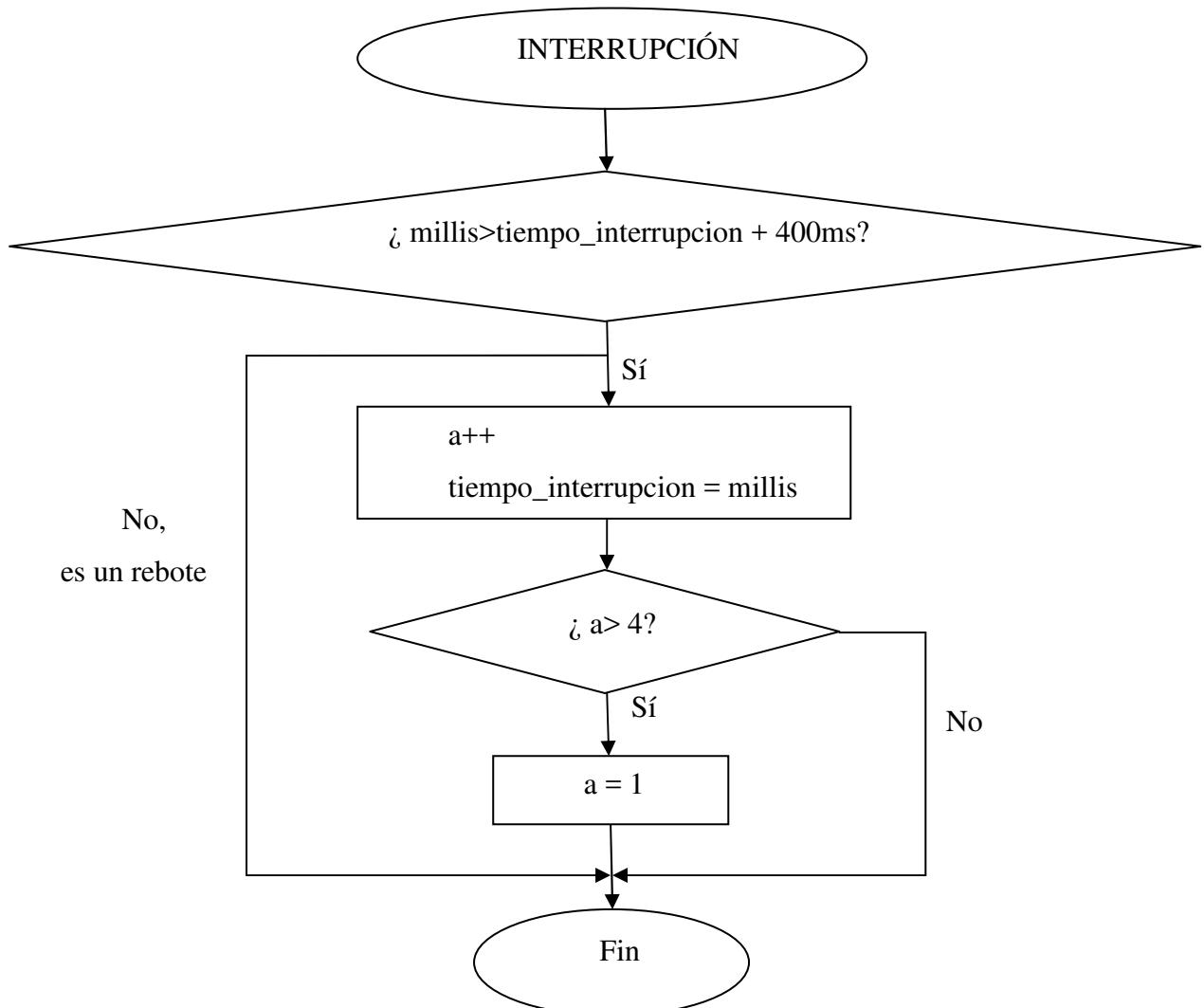


Figura 34: Diagrama de flujo de la interrupción.

En la Figura 35 se muestra el diagrama de flujo de la función que se utiliza para enviar comandos AT:

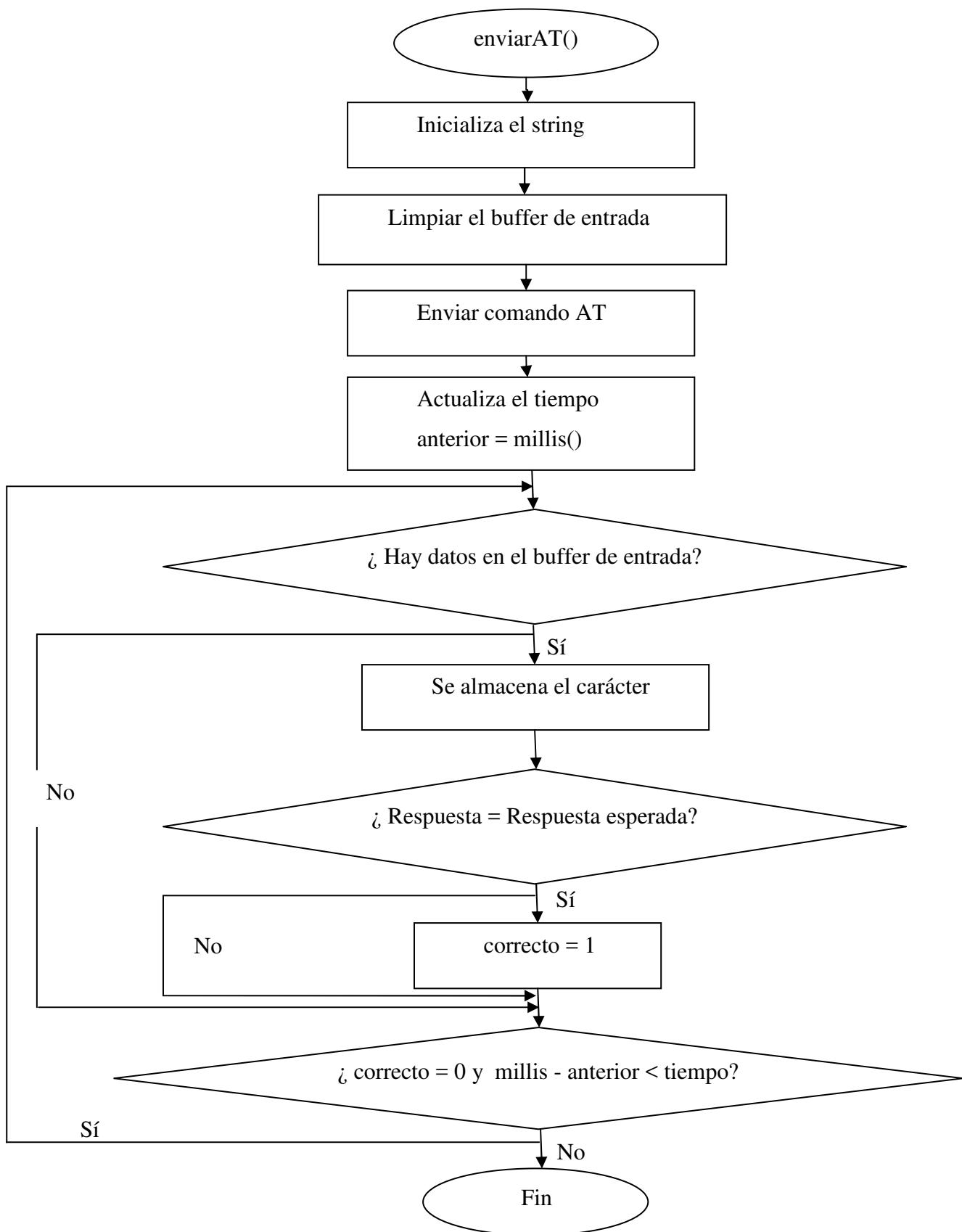


Figura 35: Diagrama de flujo de la función para enviar comandos AT al módulo GSM/GPRS.



En la Figura 36 se muestra el diagrama de flujo de la función de enviar datos al servidor:

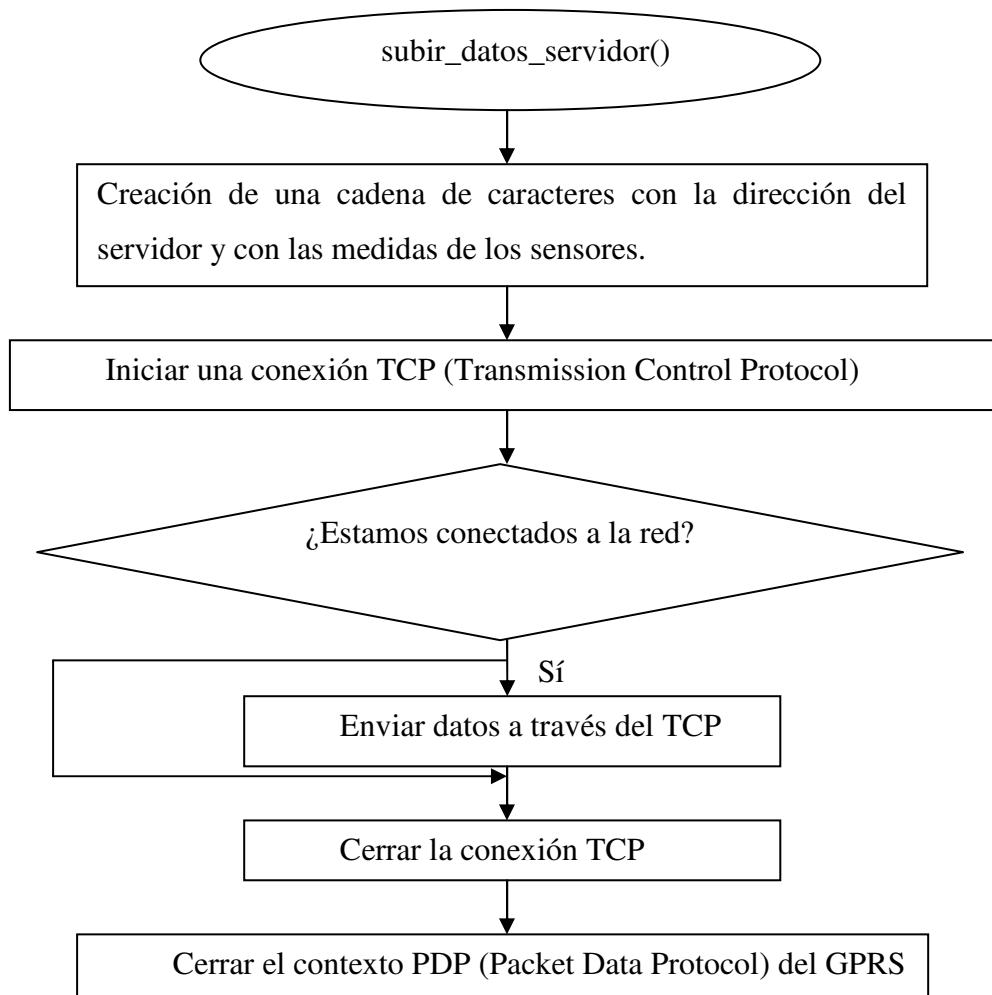


Figura 36: Diagrama de flujo de la función para enviar los datos al servidor.

5.2.4. Servidor.

Un servidor es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle la respuesta en concordancia. También ofrece la posibilidad de compartir datos, información y recursos hardware y software[36].

Es por esto último por lo que se necesita de un servidor en este proyecto ya que se requiere almacenar las medidas de los sensores recogidas con Arduino, para ser mostradas posteriormente en una página web.

Como primera instancia se creó un servidor local en una computadora propia con el programa Wampserver. Tras instalar el programa hubo que crear una cuenta para que no pudiese entrar cualquiera desde el exterior y manipular las mediciones.



Posteriormente, se creó una base de datos y una tabla donde almacenar los datos de los sensores.

Los datos que recoge el Arduino no se envían directamente al servidor sino que se hace a través de un archivo PHP (Hypertext Preprocessor) que hace de intermediario entre ellos. Este archivo tiene que acceder al nombre de usuario, contraseña, nombre de la base de datos y de la tabla para poder establecer conexión con el servidor. Después, hay que poner una función para que guarde los datos que enviamos con el Arduino en la tabla del servidor.

Todo lo comentado se tuvo que cambiar ya que no fue posible enviar los datos desde Arduino. Esto se debe a que la seguridad de la red de la Universidad no permite recibir datos enviados con una computadora o con otro tipo de máquinas desde el exterior a servidores locales conectados a la red como el que se creó. El problema se solucionó creando una cuenta en el servidor que tiene la propia Universidad el cual sí que tiene permisos para recibir datos desde el exterior. Además, también hubo que poner el archivo PHP comentado anteriormente en la carpeta correspondiente del servidor.

5.2.5. Página web.

Se creó una página web para una mejor visualización remota de los datos de los parámetros meteorológicos. De este modo, se pueden ver los datos recogidos, almacenados en el servidor, y también las medidas máximas, mínimas y medias de cada mes en cada sensor.

Para mostrar los datos del servidor en la página web hubo que poner código PHP dentro de HTML (HyperText Markup Language) que es el lenguaje en el que está programada la página. También se hizo uso del framework Bootstrap. El lenguaje PHP establece conexión con el servidor y, a través de unas funciones y el uso de SQL (Structured Query Language), es posible mostrar los datos que se desee de cualquier sensor. Todo ello es posible en caso de que dicha conexión con el servidor haya sido exitosa.

Dado que el objetivo de este proyecto no era la creación de una página web, ésta realiza lo que se desea de forma eficaz aunque no eficiente ya que no se ha podido emplear mucho tiempo perfeccionándola.



En las siguientes fotografías (Figura 37 y 38) se muestra la página web con algún dato meteorológico. En la Figura 37 se dividen los datos por meses y se muestran todos los parámetros meteorológicos que se miden además de la fecha y hora en que se realizaron las mediciones. En la Figura 38, se muestran los datos máximos, mínimos y medios por sensor en cada mes.



Información meteorológica por meses

■ Enero						
■ Febrero						
■ Marzo						
■ Abril						
Día	Temperatura	Temperatura	Temperatura	Humedad	Presión	Lluvia
2017-04-20 05:14:15	12	15	15	50	1111	10
2017-04-20 17:15:36	16	17	21	10	1010	0
■ Mayo						

Figura 37: Página web: datos meteorológicos por meses.

Información por sensor

■ Temperatura	■ Temperatura	■ Temperatura	■ Humedad	■ Presión	■ Lluvia
Temperatura					
Mes	Media		Máxima		Mínima
1	2		10		-6
2	7.5		16		-1
3	10		18		2
4	14		16		12
5	22		31		12
6	22		30		14
7	19.5		25		14
8	18.5		28		9

Figura 38: Página web: datos meteorológicos por sensores.



5.3. PROTOTIPO FINAL.

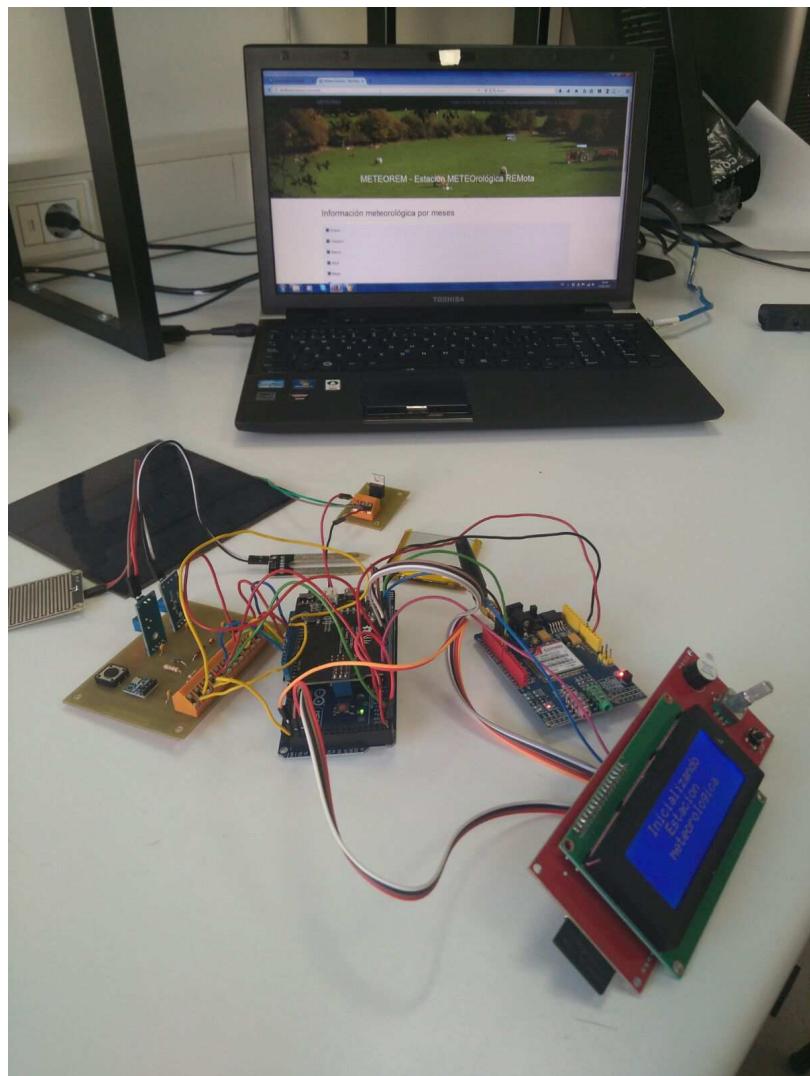


Figura 39: Prototipo final.



6. CONSUMO DEL PROTOTIPO.

En este apartado se van a mostrar los consumos de los distintos elementos de la estación meteorológica:

INSTRUMENTACIÓN	CONSUMO
Arduino Mega 2560	75mA en reposo
Módulo GSM/GPRS	2 mA en reposo 240 mA en plena transmisión de datos
LCD y módulo de la tarjeta SD	200 mA 300mA en la escritura de datos en la SD
NTC	2,8 mA
Sensor LM35	60 µA
Sensor de lluvia	4,15 mA
Sensor de humedad	4,15 mA
Sensor BMP180	1 mA durante la transmisión de datos
Botón	64 µA mientras está siendo pulsado
CONSUMO TOTAL	Entre 289 y 527mA

Tabla 2: Consumo de la instrumentación de la estación meteorológica.

El consumo máximo nunca sería de más de 527mA ya que los datos enviados por el módulo GSM/GPRS y a la tarjeta SD no se hacen en el mismo instante de tiempo sino que se hace uno detrás de otro.

El consumo medio es de poco más de 289mA ya que la mayor parte del tiempo el módulo GSM/GPRS y el de la tarjeta SD se encuentran en reposo y solo envían datos cada media hora durante un periodo de unos pocos segundos, por lo que ese gasto adicional de corriente es inapreciable. Sin embargo, hay que tener en cuenta el consumo máximo para ver si el módulo solar puede proporcionar dicha intensidad.

La duración de la batería de 2000mAh se calcula de la siguiente manera:

$$Duracion(h) = \frac{mAh(\text{batería})}{mA(\text{prototipo})} = \frac{2000mAh}{289mA} = 6h\ 55min$$

Esta duración podría resultar corta ya que es posible que no durase continuamente encendida dicha estación ya que en situaciones de noches largas como son las de invierno se agotaría la batería. Esto se podría mejorar utilizando un módulo para la tarjeta SD externo a la LCD para que ésta se pudiese apagar ya que es quien consume la



mayor parte de la corriente y no es necesario tenerla encendida todo el día debido a que esta estación se ha diseñado para estar en lugares remotos donde no se vaya a ir con frecuencia. Este módulo externo consumiría hasta 100mA durante la transmisión de datos pero al hacerse una transmisión cada media hora, al igual que el envío al servidor por parte del módulo GSM/GPRS, su consumo es mínimo. Tomando estas medidas la batería tendría un alcance de:

$$Duracion(h) = \frac{mA(batería)}{mA(prototipo)} = \frac{2000mAh}{89mA} = 22h\ 28min$$

Esta duración sería admisible ya que durante el día se podría cargar la batería y durante la noche no se descargaría.



7. PRUEBAS EN ENTORNOS REALES.

Una vez diseñado y elaborado el prototipo hay que realizar pruebas en entornos reales para ver cómo se comporta y si funciona todo como se previó de antemano. También hay que medir el tiempo de duración de la batería para así ver si en situaciones meteorológicas adversas y donde no haya suficiente radiación solar dicha estación tendría suficiente autonomía.

Las pruebas que se han realizado son:

- Dejar la estación meteorológica durante 7 horas en una ventana.
- Dejar la estación meteorológica 3 horas en el jardín de la Universidad por la mañana y 2,5 horas por la tarde.
- Realizar una prueba corta (1 hora de duración) en un entorno de uso agrícola (plantación trufera piloto del Centro Público Integrado de Formación Profesional “San Blas” -Teruel-).

La prueba de dejar la estación meteorológica en la ventana durante 7 horas se llevó a cabo el 12 de junio desde las 11 hasta las 18. Con ella se quería ver su funcionamiento durante un periodo largo de mediciones. Se pudo observar que no hubo ningún fallo. Los resultados de la prueba fueron los siguientes:

TEMPERATURA (ºC)		PRESIÓN (hPa)		HUMEDAD (%)	LLUVIA (%)
LM35	NTC	BMP180	BMP180		
24,96	25,74	25,02	996,61	30,89	0,00
25,75	26,05	25,35	996,66	31,77	0,00
26,43	26,91	26,30	996,44	31,09	0,00
26,43	27,63	27,08	996,29	30,69	0,00
27,73	28,76	26,98	996,1	31,57	0,00
29,87	31,01	28,80	996,02	31,18	0,00
30,77	31,68	31,67	995,5	26,39	0,00
31,35	32,37	32,41	995,27	22,78	0,00
32,26	33,05	33,10	995,09	18,08	0,00
32,29	33,14	33,78	995,04	15,64	0,00
32,29	33,05	33,59	994,92	12,12	0,00



TEMPERATURA (ºC)			PRESIÓN (hPa)	HUMEDAD (%)	LLUVIA (%)
32,29	33,17	33,50	994,62	9,29	0,00
32,29	26,10	33,49	994,39	7,04	0,00

Tabla 3: Prueba en una ventana.

La prueba en el Centro Público Integrado de Formación Profesional “San Blas” se realizó el día 8 de junio de las 11 a las 12 de la mañana en una plantación trufera. Esta prueba se vio oportuna ya que la mayor utilidad de este Trabajo Fin de Grado sería el estudio de plantaciones para poder tener una mayor eficiencia agrícola en el futuro. Los resultados fueron los siguientes:

TEMPERATURA (ºC)			PRESIÓN (hPa)	HUMEDAD (%)	LLUVIA (%)
LM35	NTC	BMP180	BMP180		
24,79	24,79	24,38	996,88	6,26	0,39
24,08	25,49	24,53	997,37	8,11	0,00
24,16	25,61	24,59	997,32	7,43	0,00
25,40	24,38	25,36	997,24	6,26	0,68
25,81	25,35	25,87	997,35	4,89	0,59

Tabla 4: Prueba en una plantación trufera.



Figura 40: Estación meteorológica en una plantación trufera.

A continuación se muestran los resultados de la prueba realizada en el jardín de la Universidad desde las 17:12 hasta las 19:42 el día 8 de junio a la sombra. Los resultados fueron los siguientes:

TEMPERATURA (ºC)			PRESIÓN (hPa)	HUMEDAD (%)	LLUVIA (%)
LM35	NTC	BMP180	BMP180		
27,88	29,25	28,26	991,29	71,85	0,00
24,49	26,03	24,64	991,05	64,22	0,00
24,47	25,98	24,49	990,81	63,15	0,00
23,96	25,28	23,74	990,96	62,85	0,00
23,5	24,96	23,46	990,59	62,85	0,00
22,05	23,8	22,35	991,17	62,76	0,00

Tabla 5: Prueba en el jardín de la Universidad por la tarde.



Figura 41: Estación meteorológica en el jardín de la Universidad.

Por último, se incluyen los datos recogidos en el jardín de la Universidad desde las 10:06 hasta las 13:06 del día 9 de junio a la sombra. Los resultados fueron los siguientes:

TEMPERATURA (ºC)			PRESIÓN (hPa)	HUMEDAD (%)	LLUVIA (%)
LM35	NTC	BMP180	BMP180		
20,58	22,67	21,69	992,78	74,00	0,00
19,09	21,23	19,99	992,78	67,94	0,00
19,58	21,55	20,02	992,75	68,43	0,00
22,16	24,02	23,19	992,48	68,62	0,00
24,96	27,05	27,19	992,54	68,13	0,00
25,93	28,14	28,56	992,31	68,43	0,00
24,47	26,33	26,28	992,53	64,61	0,00

Tabla 6: Prueba en el jardín de la Universidad por la mañana.



Una vez realizadas estas pruebas se ha podido observar que en la temperatura no hay demasiada precisión ya que de un sensor a otro puede variar hasta 2°C. En cambio, la presión, que fue contrastada con la de la estación meteorológica del Campus de Teruel, sí que es bastante precisa ya que no varía en más de 1hPa. Respecto a los sensores de humedad y de lluvia no se ha podido medir el error ya que no se dispone de los medios necesarios.



8. COSTES DEL PROTOTIPO.

En la tabla 2 se recogen los costes unitarios de todos los componentes que se han utilizado en esta estación meteorológica. Para su elaboración sería posible utilizar otros sensores, módulos, baterías, etc., que podrían resultar más baratos y esto podría disminuir el coste total.

INSTRUMENTACIÓN	COSTE
Arduino Mega 2560	14,69 €
Módulo GSM/GPRS	18,36 €
Módulo solar	17,99€
Placa solar de 12V y 1,5W	9,00 €
Batería Lipo 3,7V y 2000mAh	21,90 €
LM35	2,44 €
NTC 10kΩ	0,70€
Sensor de humedad de la tierra	1,89 €
Sensor de lluvia	1,7 €
BMP180	3,14 €
LCD + módulo para tarjeta SD	22,33 €
Tarjeta SD	3,75 €
Coste total del prototipo	117,89 €

Tabla 7: Costes de la instrumentación del prototipo.

A estos costes habría que añadir el gasto mensual de la tarjeta SIM que necesita en módulo GSM/GPRS para enviar los datos al servidor.

Como se ha podido observar este prototipo necesita bastante instrumentación para su correcto funcionamiento. Por ello su coste es bastante elevado aunque acorde con la cantidad de funcionalidades que realiza.



9. CONCLUSIÓN Y POSIBLES MEJoras.

La realización de este Trabajo Fin de Grado me ha parecido muy enriquecedora, a la vez que necesaria para mi formación ya que se han utilizado muchos de los conocimientos aprendidos en este Grado. También porque se han adquirido otros conocimientos nuevos, como crear un servidor, programar alguna función en PHP y cargar datos desde un servidor a una página web, entre otros.

Se ha visto también la importancia que tiene el ruido en proyectos que trabajan con sensores, ya que anteriormente no se imaginaba que era tan crítico. De este modo, hubo que admitir que no se podía eliminar del todo y hacer las medidas teniendo en cuenta este problema.

También hay que decir que dado que los sensores que se han elegido en este proyecto eran de un coste muy bajo, se ha comprobado que no tienen gran precisión. Entre los sensores de temperatura puede haber una variación de un sensor a otro, en el peor de los casos, de hasta 2°C. Además, este error siempre es entre el sensor LM35 y el termistor NTC ya que éste no está diseñado para rangos tan amplios de temperatura.

Por otra parte, la realización del Trabajo Fin de Grado supone un reto ya que se tiene que realizar la actividad propuesta con la menor ayuda posible y sacarlo adelante de la mejor manera. Siempre hay partes de la elaboración en las que falla la instrumentación o parámetros de los que no se tiene suficiente conocimiento y hay que buscar soluciones aun no sabiendo mucho sobre ese tema.

Al final, tras todos los problemas acaecidos durante su elaboración, la estación meteorológica es automática y energéticamente autosuficiente. Así mismo, captura datos de todos los sensores de los que se dispone y después hace un almacenamiento remoto, en la base de datos, y uno local, en la tarjeta SD. También realiza una visualización local de los datos meteorológicos en la pantalla LCD.

Respecto a las líneas de trabajo futuro, dado que esta estación no mide todos los parámetros meteorológicos, una posible mejora sería la implementación de más sensores de medida como un anemómetro, pluviómetro, higrómetro, etc. De este modo se conseguiría una estación meteorológica más completa. También sería deseable el diseño de un encapsulado sólido para toda la instrumentación para protegerla de los posibles daños que pueda tener del exterior.



Una vez se realizase lo anterior, y tras hacer un estudio de un cierto cultivo, se podría automatizar un invernadero a partir de los datos medidos con este proyecto aumentando así su producción.

Otra mejora sería cambiar la página web y en vez de mostrar los datos numéricos, hacer una gráfica por mes con cada uno de los sensores. Así se podrían ver las variaciones de los parámetros que hay entre el día y la noche y también entre los distintos días. De este modo, la página web quedaría más vistosa. Esta mejora no se ha hecho por la falta de tiempo y también de conocimientos ya que esto no entra dentro de los contenidos del Grado.

Además, crear una aplicación Android también sería útil ya que ahora los móviles son utilizados por todo el mundo y para el usuario quizás resultase más cómodo visualizarlos en su teléfono sin necesidad de entrar en la página web, aunque esta sea adaptativa como es el caso.



10. BIBLIOGRAFÍA.

- [1] «Estación meteorológica», *Wikipedia, la enciclopedia libre*. 22-mar-2017.
- [2] «Observatorio sinóptico de superficie», *Wikipedia, la enciclopedia libre*. 27-sep-2011.
- [3] Jorge, «METEOPRAT " PORTAL DE METEOROLOGIA ": Aparatos de Medida de Una Estacion Meteorologica Homologada», *METEOPRAT " PORTAL DE METEOROLOGIA "*, 24-nov-2010.
- [4] «Observatorio meteorológico aeronáutico», *Wikipedia, la enciclopedia libre*. 27-sep-2011.
- [5] «LA OBSERVACIÓN EN AEMET | Aemetblog». [En línea]. Disponible en: <https://aemetblog.es/2016/11/28/la-observacion-en-aemet/>. [Accedido: 04-may-2017].
- [6] «Estación termopluviométrica», *Wikipedia, la enciclopedia libre*. 27-sep-2011.
- [7] «STP Estacion Thermo-Pluviométrica», *Maranata-Madrid SL - NIF B-85746204*. [En línea]. Disponible en: <http://www.alphaomega-electronics.com/es/estaciones-nesa/1930-stp-estacion-thermo-pluviometrica.html>. [Accedido: 17-may-2017].
- [8] «Estación pluviométrica», *Wikipedia, la enciclopedia libre*. 03-feb-2014.
- [9] «iMetos RAIN - Estación para redes de monitoreo pluviométricas a Tiempo Real « Modelos « Estaciones meteorológicas iMetos « Productos | Seedmech: Estaciones Agrometeorológicas». [En línea]. Disponible en: http://www.seedmech.com/catalog.php%3Fcode%3D57%26page%3DModelos%26product%3D4%26product_name%3DiMetos_RAIN%26language%3D3). [Accedido: 17-may-2017].
- [10] «Estación meteorológica automática», *Wikipedia, la enciclopedia libre*. 31-dic-2015.
- [11] W. D. Góngora, «Instalan en Guáimaro nueva estación meteorológica automática», *CMHN Radio Guáimaro*. [En línea]. Disponible en: http://www.radioguaimaro.icrt.cu/index.php?option=com_content&view=article&id=17013:instalan-en-guaimaro-nueva-estacion-meteorologica-automatica&catid=55&Itemid=138&lang=es. [Accedido: 17-may-2017].
- [12] «Estación evaporimétrica», *Wikipedia, la enciclopedia libre*. 09-ene-2016.
- [13] «Medidores de Luz (Luxómetros), Pluviómetros, Estación de Evaporación, Medidor de Evapotranspiración - Estación de Evaporacion». [En línea]. Disponible en: <http://www.oremor.com/evaporation.html>. [Accedido: 17-may-2017].
- [14] «Como hacer una estación meteorológica con Arduino.» [En línea]. Disponible en: http://www.leantec.es/blog/38_Como-hacer-una-estaci%C3%B3n-meteorol%C3%B3gica-con-Ard.html. [Accedido: 14-abr-2017].
- [15] giltesa, «Estación Meteorológica con Arduino y WiFi», *el blog de giltesa*, 28-ene-2014.
- [16] «Estación1 archivos», *el blog de giltesa*.
- [17] Redacción, «Estación Meteorológica con Arduino alimentada con energía solar», *Arduino, Genuino, Raspberry Pi. Noticias y proyectos.*, 09-sep-2014. [En línea]. Disponible en: <http://descubrearduino.com/estacion-meteorologica-con-arduino/>. [Accedido: 14-abr-2017].
- [18] Andrey Ramirez, *Estación Meteorológica Con Arduino. «Inalambrica Bluetooth con aplicacion para dispositivos Android*. 2014.



- [19] «Arduino», *Wikipedia, la enciclopedia libre*. 11-abr-2017.
- [20] A. G. González, «Arduino Mega: Características, Capacidades y donde conseguirlo en Panamá», *Panama Hitek*, 23-ene-2013.
- [21] designthemes, «MÓDULO GSM/GPRS: llamar y enviar SMS | Tutoriales Arduino».
- [22] «Conjunto de comandos Hayes», *Wikipedia, la enciclopedia libre*. 14-mar-2017.
- [23] designthemes, «ANEXO COMANDOS AT PARA GSM/GPRS Y GPS | Tutoriales Arduino».
- [24] «Arduino Solar Shield - Cetronic». [En línea]. Disponible en: <http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idTienda=93&codProducto=151346002>. [Accedido: 17-abr-2017].
- [25] «Regulador de voltaje 7805 - para que sirve y como funciona», *Electrónica analógica y digital*, 09-mar-2016.
- [26] Luis, «Medir temperatura con Arduino y sensor LM35», *Luis Llamas*.
- [27] «VISHAY - 2322 640 63103. - THERMISTOR, NTC 10K 5% | CPC UK». [En línea]. Disponible en: <http://cpc.farnell.com/vishay/2322-640-63103/thermistor-ntc-10k-5/dp/SN35070>. [Accedido: 17-may-2017].
- [28] — Andres Cruz, «Sensor de humedad de suelo - Higrometro», *Electronilab*.
- [29] Luis, «Detector de lluvia con Arduino y sensor FC-37 o YL-83», *Luis Llamas*.
- [30] «MODULO SENSOR LLUVIA HUMEDAD AGUA CONDENSACION ARDUINO PI PROYECTOS ELECTRONICA». [En línea]. Disponible en: <http://www.ebay.es/itm/MODULO-SENSOR-LLUVIA-HUMEDAD-AGUA-CONDENSACION-ARDUINO-PI-PROYECTOS-ELECTRONICA-/152088537745>. [Accedido: 17-may-2017].
- [31] Alfonso Blesa, «Apuntes de Sistemas Empotrados». [Accedido: 3-may-2017].
- [32] «Sensor de Presión Barométrica - BMP180», *Electronilab*. [Accedido: 3-may-2017].
- [33] «Ramps 2004 LCD Smart Controller (RepRap)», *ElectroDragon*.
- [34] «Robatale Reprap Smart Controller Reprap Ramps 1.4 2004 LCD Controller for 3D Printer», *DX.com*. [En línea]. Disponible en: <http://www.dx.com/p/robatale-reprap-smart-controller-reprap-ramps-1-4-2004-lcd-controller-for-3d-printer-298336>. [Accedido: 01-jun-2017].
- [35] «Apuntes de Diseño Electrónico». 20-abr-2017.
- [36] «Servidor», *Wikipedia, la enciclopedia libre*. 05-may-2017.



ANEXO 1: DATASHEETS

- Datasheet del Arduino Mega 2560:

<http://www.mantech.co.za/datasheets/products/A000047.pdf>

- Datasheet del módulo solar:

http://wiki.seeed.cc/Solar_Charger_Shield_V2.2/

- Datasheet módulo GSM/GPRS:

http://linksprite.com/wiki/index.php5?title=SIM900_GPRS/GSM_Shield

- Datasheet LM35:

<http://www.ti.com/lit/ds/symlink/lm35.pdf>

- Datasheet NTC:

<http://eecs.oregonstate.edu/education/docs/datasheets/10kThermistor.pdf>

- Datasheet del sensor BMP180:

<https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

- Datasheet del regulador 7805:

<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>

- Datasheet RepRapDiscount Controller:

http://reprap.org/wiki/RepRapDiscount_Smart_Controller

- Tutorial del sensor de humedad:

<https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

- Tutorial del sensor de lluvia:

<https://www.luisllamas.es/arduino-lluvia/>



ANEXO 2: CÓDIGO DE ENLACE AL SERVIDOR (CD)

ANEXO 3: CÓDIGO DE LA ESTACIÓN METEOROLÓGICA (CD)

ANEXO 4: CÓDIGO DE LA PÁGINA WEB (CD)