



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería
Industrial

DISEÑO Y DESARROLLO DE UNA ESTACIÓN AGROCLIMÁTICA DE BAJO COSTE

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA



Universidad
Politécnica
de Cartagena

Autor: Jose Llovera Pujalte
Director: Dña. Dolores Parras Burgos
Codirector: D. Juan José Pérez Solano

INDICE

INDICE DE FIGURAS.....	5
INDICE DE TABLAS.....	7
RESUMEN	8
ABSTRACT.....	9
1. INTRODUCCIÓN.....	10
1.1 Breve recorrido histórico sobre meteorología.....	10
1.2 Definición de estación meteorológica	11
1.3 Estaciones meteorológicas.....	12
1.4 Sistema de Información Agro-meteorológica de la Región de Murcia.....	13
1.4.1 Ejemplos de Estaciones Automáticas de la Región de Murcia	14
2. MEMORIA DESCRIPTIVA	19
2.1 Medición de variables.....	19
3. REQUISITOS DEL PROYECTO	26
3.1 Etapa de alimentación	26
3.2 Consumo del dispositivo.....	27
3.3 Cálculo del panel solar y batería	29
3.4 Cálculos para dimensionado de la placa solar	31
3.5 Cálculos para el RTCC interno	34
3.4.1 Configuración del RTC.....	36
4. DISPOSITIVOS QUE INTEGRA NUESTRO NODO.....	39
4.1 Sensor de temperatura y humedad dht22.....	43
4.2 Pluviómetro de cubeta basculante	47
4.3 Anemómetro	49
4.4 Veleta.....	50
4.5 Piranómetro SP-110.....	55
4.6 COMPONENTES PARA ESTACIÓN 2.0	57
4.6.1 Sensor de Tª y Humedad HC2S3 de Campbell Scientific.....	57
4.6.2 Pluviómetro 52202-L de Young Company.....	64
4.6.3 Anemoveleta 05103 de Young Company.....	69
5. Diseño de la placa	71
5.1 Diseño placa de conexionado de sensores	71
5.1.1 Circuitstudio	71
5.1.2 Esquemático	72

5.1.3	Layout y Modelo 3D	74
6.	SOFTWARE	76
6.1	Protocolos de comunicación	76
6.1.1	GPIO	76
6.1.2	UART.....	77
6.1.3	SPI	78
6.1.4	I2C.....	80
6.2	MQTT	81
6.3	Node Red + Adafruit IO.....	82
6.4	Configuración FiPy	85
6.4.1	Boot.py	86
6.4.2	Main.py	87
6.4.3	Adafruit IO	87
6.4.4	Umqtt.py	90
6.4.5	Dashboard	95
7.	INSTALACIÓN Y MANTENIMIENTO DEL NODO	99
7.1	Instalación del nodo	99
7.2	Mantenimiento del nodo.....	102
8.	PRESUPUESTO	105
9.	CONCLUSIONES.....	106
10.	BIBLIOGRAFÍA.....	107

INDICE DE FIGURAS

Figura 1: Disposición general de una estación automática	13
Figura 2: Estación MU52, Cañada de San Pedro - Cabezo Plata (Murcia)	14
Figura 3: Estación de Jumilla en Las Encebras	16
Figura 4: Algunos sensores de la estación de Las Encebras	17
Figura 5: Glosario de términos para variables medidas	18
Figura 6: Pantallas de Stevenson	21
Figura 7: Sensores de humedad capacitivo y resistivo	22
Figura 8: Pluviómetro de cubeta basculante	23
Figura 9: Barómetro de mercurio y aneroides	23
Figura 10: Anemómetro rotatorio y de presión	24
Figura 11: Radiación solar	25
Figura 12: Espectro de la radiación solar	25
Figura 13: Radiación incidente en la Península	26
Figura 14: Cargador de batería BQ24040	27
Figura 15: Horas solares en Murcia, 2018-2019	30
Figura 16: Batería	31
Figura 17: Panel Solar	31
Figura 18: Web de la base de datos del GIS	32
Figura 19: Ficha técnica del panel solar	33
Figura 20: Gráfico de días nublados en la Región de Murcia	34
Figura 21: Módulo del RTC	36
Figura 22: Placa de desarrollo FiPy	39
Figura 23: Diagrama de pines de la placa de desarrollo FiPy	41
Figura 24: Diagrama de pines de la placa de expansión 3.0	42
Figura 25: Sensor DHT22	43
Figura 26: Distribución de pines del DHT22	43
Figura 27: Conexión del DHT22	45
Figura 28: Pluviómetro de cubeta basculante	47
Figura 29: Mecanismo interno de la cubeta	47
Figura 30: Conexión del pluviómetro	48
Figura 31: Anemómetro	49
Figura 32: Conexión del anemómetro	49
Figura 33: Veleta	50
Figura 34: Circuito equivalente veleta	50
Figura 35: Estructura interna de la veleta y divisor para su medición	51
Figura 36: Conexión veleta	52
Figura 37: Piranómetro	55
Figura 38: Terminales para conexión del SP-110	56
Figura 39: Conexión del piranómetro	56
Figura 40: Sensor HC2S3	57
Figura 41: Colocación del HC2S3	60
Figura 42: Comunicación del HC2S3 con el Datalogger	62
Figura 43: Cableado del HC2S3	62
Figura 44: Programación del Datalogger	63
Figura 45: Pluviómetro 52202-L de Young Company	64
Figura 46: Esquema del Pluviómetro	65
Figura 47: Cableado del Pluviómetro con el Datalogger	66
Figura 48: Programación Datalogger	68
Figura 49: Anemoveleta 05103 de Young Company	69

<i>Figura 50: Archivos del proyecto en Circuitstudio</i>	72
<i>Figura 51: Esquemático placa de conexionado</i>	73
<i>Figura 52: Vista Top + Bottom con cobre incluido</i>	74
<i>Figura 53: Vista Top del ruteado</i>	74
<i>Figura 54: Modelo 3D</i>	75
<i>Figura 55: Distribución de los GPIO</i>	76
<i>Figura 56: Comunicación UART</i>	77
<i>Figura 57: Topología SPI</i>	78
<i>Figura 58: Comunicación SPI</i>	79
<i>Figura 59: Comunicación SPI</i>	79
<i>Figura 60: Protocolo I2C</i>	80
<i>Figura 61: Arquitectura MQTT</i>	81
<i>Figura 62: Comunicación MQTT</i>	83
<i>Figura 63: Entorno de trabajo de Node-Red</i>	84
<i>Figura 64: Dashboard sobre temperatura de una sonda</i>	84
<i>Figura 65: Puente en la FiPy para actualización</i>	85
<i>Figura 66: Librerías de Pycom en GitHub</i>	86
<i>Figura 67: Feeds en Adafruit IO</i>	94
<i>Figura 68: Almacenamiento de datos en Adafruit IO</i>	95
<i>Figura 69: Esquema de nodos con Node-Red</i>	95
<i>Figura 70: Dashboard con datos en tiempo real e históricos de 24h</i>	97
<i>Figura 71: Disposición de los elementos que integran una estación</i>	99
<i>Figura 72: Kit meteorológico</i>	101
<i>Figura 73: Carcasa para el nodo</i>	101
<i>Figura 74: Ejemplos de cartillas de mantenimiento</i>	104

INDICE DE TABLAS

<i>Tabla 1: Datos de la estación MU52.....</i>	<i>15</i>
<i>Tabla 2: Componentes de la estación MU52.....</i>	<i>15</i>
<i>Tabla 3: Variables medidas de la estación MU52.....</i>	<i>15</i>
<i>Tabla 4: Datos de la estación JU71.....</i>	<i>16</i>
<i>Tabla 5: Componentes de la estación JU71.....</i>	<i>16</i>
<i>Tabla 6: Variables medidas de la estación JU71.....</i>	<i>17</i>
<i>Tabla 7: Algunas de las estaciones con comunicación GPRS.....</i>	<i>18</i>
<i>Tabla 8: Tabla de precisión mínima para medición de variables.....</i>	<i>19</i>
<i>Tabla 9: Requisitos mínimos según UNE 176101:2010.....</i>	<i>20</i>
<i>Tabla 10: Consumo de la placa de desarrollo por modos disponibles.....</i>	<i>28</i>
<i>Tabla 11: Consumo de la estación.....</i>	<i>28</i>
<i>Tabla 12: Datos obtenidos del servidor PVGIS-SARAH.....</i>	<i>32</i>
<i>Tabla 13: Media de irradiación anual.....</i>	<i>32</i>
<i>Tabla 14: MCP7940N-I/SN Datasheet.....</i>	<i>35</i>
<i>Tabla 15: Características de la FiPy.....</i>	<i>40</i>
<i>Tabla 16: Consumo de los servicios de la FiPy.....</i>	<i>41</i>
<i>Tabla 17: Características eléctricas del sensor DHT22.....</i>	<i>44</i>
<i>Tabla 18: Características eléctricas del DHT22.....</i>	<i>44</i>
<i>Tabla 19: Máximo error de Temperatura del DHT22.....</i>	<i>44</i>
<i>Tabla 20: Error de Humedad Relativa a 25°C del DHT22.....</i>	<i>45</i>
<i>Tabla 21: Tensión de salida de la veleta en función de la dirección.....</i>	<i>52</i>
<i>Tabla 22: Especificaciones del SP-110.....</i>	<i>55</i>
<i>Tabla 23: Precisión de Tª del HC2S3.....</i>	<i>58</i>
<i>Tabla 24: Precisión de %HR del HC2S3.....</i>	<i>59</i>
<i>Tabla 25: Color del cable, función, y conexión al datalogger para medida en modo single-ended.....</i>	<i>60</i>
<i>Tabla 26: Color del cable, función, y conexión al datalogger para medida en modo diferencial.....</i>	<i>61</i>
<i>Tabla 27: Cableado a un Canal de Entrada de Pulsos.....</i>	<i>66</i>
<i>Tabla 28: Cableado del Puerto de Entrada de Control.....</i>	<i>67</i>
<i>Tabla 29: Unidades de conteo de lluvia.....</i>	<i>67</i>

RESUMEN

En este proyecto se detallan los pasos para el diseño e implementación de una estación agroclimática de bajo coste, con localización remota, la cual permitirá la recopilación de datos meteorológicos tales como temperatura, humedad, dirección y velocidad del viento, precipitación y radiación solar a través de un despliegue de sensores que conforman la estación. Tras la adquisición de los datos, estos se transmiten mediante protocolo MQTT a un servidor donde se van almacenando para su posterior gestión y manipulación.

El fin de la adquisición de datos es conseguir un muestreo de las variables meteorológicas mediante la implementación de una interfaz para el usuario donde se pueda contemplar la evolución de estas variables a lo largo de los días, semanas o meses. Esto se ha trabajado con la implementación de un Dashboard, una interfaz gráfica en la que se muestran los datos recogidos por los sensores en tiempo real, así como los históricos que muestran la evolución de los datos a lo largo del tiempo en forma de gráficas.

El equipo de sensores ha sido escogido con la premisa del bajo coste, pero sin arriesgar demasiado la fiabilidad de los datos a recoger. La estructura de la estación tiene que instalarse de forma que aguante las inclemencias meteorológicas, y aislar de forma adecuada todos los componentes electrónicos que la integran.

Por último, para la autonomía del nodo se han realizado los cálculos pertinentes para la instalación de un panel solar que abastezca a la estación, y la integración de una batería para que disponga de energía en caso de emergencia.

ABSTRACT

In this project, the author has detailed the set of steps to follow in order to design and implement a low cost agroclimatic station with remote location, which will be able to collect data related to temperature, humidity, wind and solar radiation through different sensors that form the station. Once data is acquired, these are transmitted using a MQTT protocol, to a server where they are stocked up for a future manipulation.

The final goal of acquiring data, is to collect a big sample of the different meteorological variables through the implementation of an interface for the user, where it is possible to observe the evolution of these variables in time. This has been done setting up a Dashboard; a graphic interface in which every data collected by the sensors is shown in real time, accompanied by historical statistics, permitting the user to follow the evolution of data with different graphs.

The sensor system has been chosen with the premise of achieving a low cost station with the capacity of providing reliable data. The structure of the station needs to be installed in a way that allows the materials to resist bad weather, and isolate its electronic components.

At last, the author has done the calculations necessary to achieve autonomy for the station in terms of energy by installing a solar panel. At the same time it has been integrated in the system an emergency battery.

1. INTRODUCCIÓN

1.1 Breve recorrido histórico sobre meteorología

Me gustaría hacer un pequeño repaso histórico sobre la evolución de esta ciencia y la influencia que algunos personajes históricos han tenido sobre ella.

El interés por el tiempo atmosférico se remonta a los comienzos de la humanidad, y los intentos por comprenderlo y predecirlo a los primeros tiempos de los que tenemos testimonios escritos. Desde el principio el conocimiento de los fenómenos atmosféricos estuvo mediatizado por creencias sobrenaturales y asociado al movimiento de los astros, y hasta hace bien poco se han mantenido esas influencias en el estudio debido a que no se habían desarrollado las técnicas científicas necesarias. Los primeros intentos de explicar estos fenómenos mediante una aproximación científica se dieron en la época griega con obras como las de Tales de Mileto (624-545 a.C.), Anaximandro (611-547 a.C.), Hipócrates de Cos (460-375 a.C.), y demás sabios helénicos.

Aunque el estudio más amplio y difundido de aquella época corresponde a Aristóteles (384-322 a.C.), el cual introdujo el término “meteorología”, que etimológicamente está formado por las palabras griegas Meteoros, “alto en el cielo” y Lógica, “conocimiento o tratado”. Además, él introdujo el primer estudio relacionado principalmente con la atmósfera, los cuatro libros conocidos como “los Meteorológicos”.

Estos escritos supusieron el mejor compendio reconocido del saber meteorológico hasta el Renacimiento. Jose de Acosta (1517-1600) gran precursor de la climatología moderna y uno de los pocos españoles que han dejado su nombre en la historia de las ciencias atmosféricas, afirmó en una de sus obras: “no me determino a contradecir a Aristóteles, sino es en cosa muy cierta”.

Desarrollo de la instrumentación

Gracias a la navegación y a los primeros instrumentos utilizados para medir variables meteorológicas como el termómetro, el barómetro y el anemómetro (introducidos por Galileo, Torricelli y Hooke respectivamente a lo largo del S.XVII) y a las observaciones de marineros y hombres de ciencia, fueron asentando cada vez más las explicaciones sobre los fenómenos atmosféricos.

De esta forma se fueron dejando de lado las predicciones basadas en relaciones astrológicas, y a principios del S.XIX se empezaron a introducir los mapas de temperatura y presión, pero limitados a la superficie terrestre.

Meteorología moderna

Con la aparición del telégrafo en 1840 se empezaron a recopilar observaciones con rapidez y a transmitirse de forma mucho más cómoda entre meteorólogos de distintos puntos geográficos. Esto introdujo la necesidad de una coordinación internacional para contrastar y recopilar los estudios del clima, lo que dio origen al primer organismo de coordinación mundial en Viena en 1873: la Organización Meteorológica Internacional (OMI), que posteriormente se ha convertido en la Organización Meteorológica Mundial (OMM) en 1950.

A comienzos del S.XX el científico noruego Vilhelm Bjerknes, apodado “el padre de la Meteorología Moderna, expuso lo que son las bases de la predicción actual, exponiendo los principales problemas: como la dificultad de los problemas matemáticos, así como la escasez de datos y su limitación a la superficie terrestre.

Con el desarrollo de la computación los modelos matemáticos empezaron a tener la suficiente potencia para poder manejar la cantidad de datos requeridos en los cálculos, dando lugar a la “predicción numérica” utilizada en nuestros días. El avance en la colaboración internacional y la tecnología implicada en la meteorología hizo progresar la aviación, así como el interés de los gobiernos en apoyar el desarrollo de esta ciencia.

El lanzamiento del primer satélite artificial en 1957 dio paso a la etapa moderna de observación atmosférica, ya que permitió enviar imágenes desde el espacio. Con el paso de los años la observación atmosférica desde el espacio, unida a la computación de los modelos físico-matemáticos se han convertido en las herramientas principales que permiten el desarrollo de la meteorología presente en nuestros días.

1.2 Definición de estación meteorológica

Una estación meteorológica es una instalación formada por un conjunto de elementos de sensorización, gracias a la cual se miden y registran (de forma continuada en el tiempo) los valores de diferentes variables meteorológicas, cuya información es recogida por los sensores. Los datos recogidos en la estación se utilizan para modelar predicciones meteorológicas, en la elaboración de mapas climáticos, y en diversos estudios de interés climático. En definitiva, se utiliza para recoger datos del clima y el tiempo de cualquier zona de interés de estudio. En nuestro caso, nuestro núcleo de estudio sería el destinado a la agricultura.

-Según la OMM (Organización Meteorológica Mundial), una estación agrometeorológica:

Estación que proporciona datos meteorológicos para fines agrícolas y/o biológicos y que efectúa otras observaciones meteorológicas en el marco de los programas de los centros de investigación agrometeorológica y de otras entidades interesadas.

Ref: <https://unterm.un.org/unterm/display/record/wmo/na?OriginalId=a3136a3c-7de8-437b-bc13-0120cb375aae>

1.3 Estaciones meteorológicas

La Organización Meteorológica Mundial (OMM) clasifica las estaciones meteorológicas de la siguiente forma:

- Estaciones sinópticas de superficie
- Estaciones de observación en altitud
- Estaciones meteorológicas en aeronaves
- Estaciones meteorológicas aeronáuticas
- Estaciones sobre buques de investigación y para fines especiales
- Estaciones climatológicas
- Estaciones meteorológicas agrícolas
- Estaciones especiales

Pero de forma general, se pueden distinguir 2 tipos de estaciones: las automáticas (conocidas como EMAs) y las convencionales. Las estaciones convencionales (las manuales) tienen un registro de datos tomados diariamente por parte de personal profesional preparado para la toma, lectura y registro de datos, es lo que conocemos como el observador meteorológico; mientras, en las estaciones automáticas, se hace un registro y almacenamiento de variables de forma continua en el tiempo, sin necesidad de personal implicado en la tarea, además el avance de la tecnología en cuanto al uso de la energía solar obtenida por medio de placas, hace posible la instalación de estas estaciones en lugares inhóspitos donde no llega el cableado eléctrico o no es tan fácil realizar una revisión periódica de la misma.

En el presente trabajo, seguiré la línea de las estaciones automáticas para el diseño del nodo agroclimático.

En la siguiente imagen se puede apreciar la estructura general de una estación meteorológica automática (<https://pluselectric.wordpress.com/>):

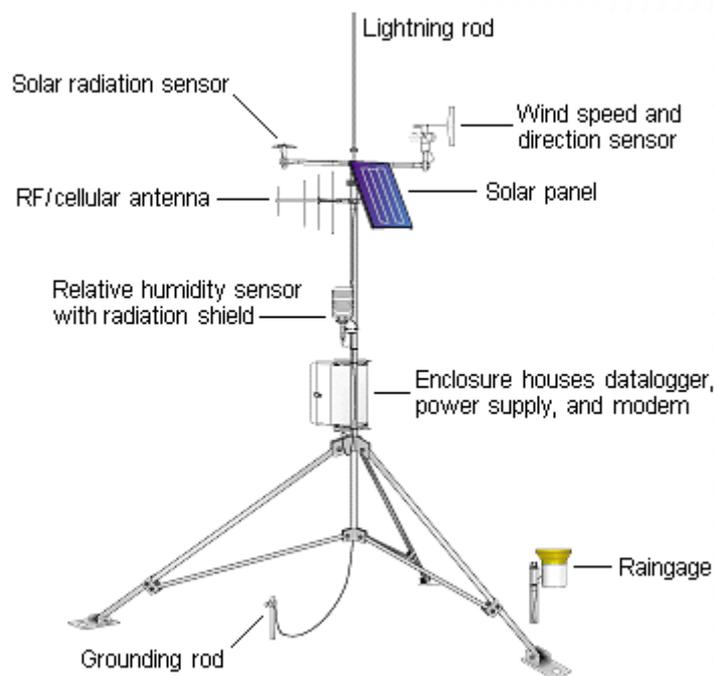


Figura 1: Disposición general de una estación automática

1.4 Sistema de Información Agro-meteorológica de la Región de Murcia

La red de estaciones agro-meteorológicas de la Región de Murcia está coordinada por el SIAM, el Sistema de Información Agro-meteorológica de la Región de Murcia, perteneciente al IMIDA, el Instituto Murciano de Investigación y Desarrollo Agrario y Alimentario. En octubre de 2020 esta red alcanzó ya las 50 estaciones automáticas.

El SIAM actúa como órgano de transferencia entre la comunidad científica y el sector productivo, haciendo uso de las tecnologías de información y los sistemas de comunicación más actuales. Es el encargado de fomentar éstas tecnologías en el sector agro-alimentario de la región entre el personal técnico del sector y agricultores cualificados.

Las acciones se engloban esencialmente en tres grupos: sistemas para la obtención de información, sistemas para la transferencia de información, e innovación en sistemas de transferencia tecnológica.

Cada estación registra o estima distintos datos agrometeorológicos, como la temperatura, humedad relativa, velocidad y dirección del viento, radiación global incidente, precipitación, punto de rocío, déficit de presión de vapor o evapotranspiración de referencia.

Todos estos datos proporcionados por las estaciones, una vez recibidos, evaluados y validados por el IMIDA, se almacenan en un servidor para que los interesados puedan acceder de forma gratuita desde la página web del SIAM.

Su página web: <https://www.imida.es/web/imida>

1.4.1 Ejemplos de Estaciones Automáticas de la Región de Murcia

Algunas estaciones con sistemas de comunicación GPRS pertenecientes a la red automática de la Región de Murcia:

<http://siam.imida.es/apex/f?p=101:110:887368143174247#>

<http://siam.imida.es/apex/f?p=101:1:887368143174247>

- Estación MU-52, en Cabezo Plata, Murcia.



Figura 2: Estación MU52, Cañada de San Pedro - Cabezo Plata (Murcia)

- Datos de su ubicación:

Nombre - Paraje (Municipio)	Cañada de San Pedro - Cabezo Plata (Murcia)
Finca	-
Coordenadas Geográficas	Lat:37° 58' 39,15" Lon:0° 59' ,69"
Coordenadas UTM	X:677062 Y:4205239
Altitud	125 m.
Fecha de Alta	18-FEB-1996
Último dato recibido	06-MAR-2021

Tabla 1: Datos de la estación MU52

- Componentes que la integran:

Marca	Modelo	Nº Serie	Descripción
YOUNG	05103-5	80813	ANEMOVELETA
KIPP & ZONEN	CMP6	070552	RADIÓMETRO
THIES-CLIMA	5.4031.30.007	29118	PLUVIÓMETRO
CAMPBELL	CR1000	11299	DATALOGGER
VAISALA	HMP155A	R4640886	TERMOHIGROMETRO

Tabla 2: Componentes de la estación MU52

- Variables que recoge la estación:

Tminabs (°C)	Tmaxabs (°C)	Tmin (°C)	Tmax (°C)	Vvmed (m/seg)	Vvmax (m/seg)
-1,11 27/12/10 08:00	44,96 27/08/20 12:00	1,57 18/01/17	34,67 01/08/20	7,73 08/01/10	26,17 24/01/09 08:00

Radmed (w/m2)	Radmax (w/m2)	Hsol (h)	Precmax (mm)	Prec (mm)	Eto (mm)
375,29 02/06/19	1.529 07/05/16 11:00	13 27/06/09	46,2 13/09/19 05:00	112,90 13/09/19	10,25 16/07/18

Tabla 3: Variables medidas de la estación MU52

- Estación JU-71, en Las Encebras, Jumilla.



Figura 3: Estación de Jumilla en Las Encebras

- Datos de su ubicación:

Nombre - Paraje (Municipio)	Las Encebras - Las Encebras (Jumilla)
Finca	Capellania
Coordenadas Geográficas	Lat:38° 23' 40,01" Lon:1° 14' 21,58"
Coordenadas UTM	X:653633 Y:4250840
Altitud	401 m.
Fecha de Alta	01-ENE-1996
Último dato recibido	06-MAR-2021

Tabla 4: Datos de la estación JU71

- Componentes que la integran:

Marca	Modelo	Nº Serie	Descripción
YOUNG	05103-5	87899	ANEMOVELETA
KIPP & ZONEN	CMP6	080274	RADIÓMETRO
THIES-CLIMA	5.4031.30.007	29116	PLUVIÓMETRO
CAMPBELL	CR1000	E2477	DATALOGGER
VAISALA	HMP45AC	D2930015	TERMO HIGRÓMETRO

Tabla 5: Componentes de la estación JU71



Figura 4: Algunos sensores de la estación de Las Encebras

- Variables que recoge la estación:

Tminabs (°C)	Tmaxabs (°C)	Tmin (°C)	Tmax (°C)	Vvmed (m/seg)	Vvmax (m/seg)
-9,60 28/01/05 05:00	45,50 24/07/09 15:00	-1,66 29/01/06	32,84 24/07/09	8,81 20/01/20	24,80 05/02/15 05:00

Radmed (w/m2)	Radmax (w/m2)	Hsol (h)	Precmax (mm)	Prec (mm)	Eto (mm)
380,45 30/06/17	1.554 17/05/11 12:00	13 29/06/17	44,5 28/09/12 15:00	69,10 28/09/12	9,35 27/07/19

Tabla 6: Variables medidas de la estación JU71

- Glosario de términos:

TMINABS	Temperatura mínima absoluta de una hora (°C)
TMAXABS	Temperatura máxima absoluta de una hora (°C)
TMIN	Día con la temperatura media mas baja (°C)
TMAX	Día con la temperatura media mas elevada (°C)
VVMED	Velocidad del viento media mas elevada de un dia (m/seg)
VVMAX	Velocidad del viento maxima de una hora (m/seg)
RADMED	Radiacion media mas elevada de un dia (w/m ²)
RADMAX	Radiacion maxima de una hora (w/m ²)
HSOL	Ultimo dia con maximo numero de horas de sol (nº horas)
PRECMAX	Precipitacion maxima de una hora (mm)
PREC	Precipitacion maxima de un día (mm)
ETO_PM_FAO	Evapotranspiracion maxima de un dia (mm)

Figura 5: Glosario de términos para variables medidas

Los datos recogidos en tiempo real por las estaciones del SIAM se pueden consultar desde la página <http://siam.imida.es/apex/f?p=101:110:1255956958716522>

Algunas de las estaciones con sistema de comunicación GPRS, para obtención de datos instantáneos, son:

Altiplano	
JU81	El Aljuzarejo, paraje Román, Jumilla
JU71	Las encebras, Jumilla
Vega Alta del Segura	
CI42	La Carrichosa, en el Municipio de Cieza
CI52	Rotas, en el municipio de Calasparra
Vega Media del Segura	
MO22	Campotejar, en Molina de Segura
Huerta de Murcia	
MU52	Cabezo de la Plata, en el Municipio de Murcia
Vega Media del Guadalentín	
AL41	Alhama Valle, en Alhama
Campo de Cartagena-Mar Menor	
TP52	El Mirador
TP73	Los Infiernos

Tabla 7: Algunas de las estaciones con comunicación GPRS

2. MEMORIA DESCRIPTIVA

2.1 Medición de variables

Cuando tenemos datos es necesario asegurarnos que sean de buena calidad. Cuando la calidad de un dato no puede ser asegurada, a veces, es mejor descartarlo. Debemos asegurarnos que venga de una fuente confiable. Para esto, la Organización Meteorológica Mundial (OMM) plantea las condiciones que deben cumplir los instrumentos y el lugar donde se toman las medidas, horarios fijos de toma de medidas, grado de exactitud de cada variable, etc.

Según la Organización Meteorológica Mundial la precisión mínima con la que se tienen que medir las variables es la siguiente:

Variable	Precisión requerida para valores diarios
Temperatura (incluyendo máx/mín, bulbo seco y mojado, y suelo)	$< \pm 0,5 \text{ }^{\circ}\text{C}$
Precipitación	$\pm 1\text{mm}$
Radiación solar	10% ($\pm 1\text{h}$)
Evaporación	$\pm 1\text{mm}$
Humedad relativa	$\pm 5\%$
Fotoperiodo	10% ($\pm 1\text{h}$)
Velocidad del viento	$\pm 0,5 \text{ m/s}$
Presión del aire	$\pm 0.1 \text{ hPa}$

Tabla 8: Tabla de precisión mínima para medición de variables

Ref: https://library.wmo.int/doc_num.php?explnum_id=3996 pág. 62

Y según UNE 176101:2010, Punto 7.2, para “requisitos de exactitud operativa y rendimiento de cada instrumento meteorológicos exigidos por la OMM, se nos facilita la siguiente tabla:

Variable	Límites de funcionamiento	Exactitud recomendable	Exactitud mínima	Resolución mínima
Temperatura del aire	De -40 °C a +60 °C	± 0,1 K	± 0,2 K (a 20 °C)	0,1 K
Humedad relativa	De 5% a 100%	± 3%	± 4%	1%
Precipitación (total)	De 0 mm a 400 mm	± 0,1 mm para ≤ 5 mm ± 2% para > 5 mm		0,2 mm
Precipitación (intensidad)			± 2% para < 25 mm/hora ± 3% para > 25 < 50 mm/hora	0,1 mm
Dirección del viento	Entre 0° y 360°	± 5°	± 5°	1°
Velocidad del viento	Entre 0 m/s y 75 m/s	± 0,5 m/s para ≤ 5 m/s ± 10% para > 5 m/s	± 0,5 m/s para ≤ 5 m/s ± 10% para > 5 m/s < 60 m/s	0,1 m/s
Radiación global	De 0 a 24 h Entre -35 °C y +75 °C	± 2%	± 5%	1 W/m ²
Radiación fotosintéticamente activa	De 0 a 24 h Entre -35 °C y +75 °C	± 3%	± 5%	1 W/m ²
Radiación neta	De 0 a 24 h Entre -35 °C y +75 °C	± 2%	± 5% típico (± 10% en condiciones extremas)	1 W/m ²
Temperatura del suelo	De -50 °C a +60 °C		± 0,4 °C de -20 a 48 °C	0,1 K
Presión atmosférica	Entre 920 hPa y 1 080 hPa (a nivel del mar)	± 0,3 hPa	± 0,5 hPa a 20 °C	0,1 hPa
Humectación en hoja	De 1 kΩ a 3 000 kΩ			1 kΩ
Placa de flujo	Entre -40 °C y +55 °C De -100 W/m ² a +100 W/m ²	± 5%	± 5%	1 W/m ²
Humedad del suelo	De 0 m ³ /m ³ hasta saturación Entre 0 °C y 50 °C	± 0,01 m ³ /m ³ (± 1%)	± 0,03 m ³ /m ³ (± 3%)	0,002 m ³ /m ³ (0,2%)

Tabla 9: Requisitos mínimos según UNE 176101:2010

Las mediciones de variables físicas recomendadas en las estaciones agroclimáticas son:

- Temperatura del aire y del suelo
- Viento: velocidad y dirección
- Radiación
- Humedad del aire y del suelo
- Presión atmosférica
- Horas solares
- Evapotranspiración

Nuestra estación, al tratarse de una estación meteorológica automática (EMA), y teniendo en cuenta los componentes que la integran, se encargará de recoger datos de variables de temperatura, viento, humedad, presión atmosférica y radiación solar.

- **Temperatura**

La temperatura es una magnitud física fácilmente reconocible, debido a la sensación que experimentamos al tocar objetos fríos o calientes, pero más específicamente, consiste en la cantidad de energía térmica que alberga un cuerpo, en el caso que nos atañe, la cantidad de energía térmica que albergan las moléculas del aire atmosférico. En el caso de los gases, este valor depende de la energía cinética de las moléculas, cuanto mayor sea la temperatura, mayor es la velocidad de estas partículas. Su unidad de medida en el SI es el Kelvin.

Para medir la temperatura se usan los termómetros, los hay de distintos tipos:

- Termómetros de líquido en tubo de vidrio. Los más conocidos son los de mercurio.
- Termómetros de líquido en envoltura metálica, basado realmente en un manómetro calibrado para indicar temperaturas.
- Termopares.
- Termómetros bimetálicos.
- Termistores.
- Termómetros de resistencia de platino.

Desde el aspecto meteorológico, la medida de temperatura que se busca es la de una temperatura seca, esta es aquella cuya medida no se vea afectada por la exposición directa del termómetro a la radiación solar, al flujo de aire del entorno donde este colocada y a la humedad relativa.

Para evitar la exposición a estos fenómenos se coloca una pantalla protectora a modo de escudo, en cuyo interior se encuentra el sensor de temperatura, esta se conoce como "Pantalla de Stevenson". Puede ser del tipo garita, dentro de la que se coloca el nodo de la estación, o puede ser únicamente para contener el sensor de temperatura:

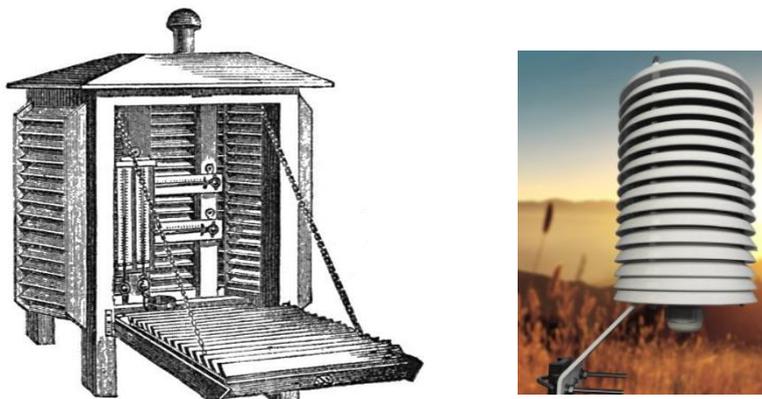


Figura 6: Pantallas de Stevenson

Estos blindajes disfrutaban de ventilación natural, pero para algunas mediciones específicas se utiliza ventilación artificial, creando un flujo de aire interno gracias a un ventilador.

- **Humedad relativa**

La medición de la HR es otra de las variables necesarias a considerar en los criterios mínimos de una EMA. La humedad se define como la cantidad de vapor de agua presente en el aire, procedente del fenómeno de la evaporación de distintas masas de agua, como los mares, ríos...etc. Esta depende de la temperatura, y cuanto más caliente se encuentra el aire más vapor de agua admite. La humedad relativa se mide en tanto por ciento de vapor de agua presente en el aire, %HR.

La humedad se mide a través de tres magnitudes, la humedad absoluta, la humedad específica y la humedad relativa, esta última es la utilizada para medición en las EMAs, y el instrumento que se usa para su medición es el higrómetro. Los sensores más utilizados en este ámbito son los capacitivos y los resistivos.



Figura 7: Sensores de humedad capacitivo y resistivo

- **Precipitación**

La precipitación en nuestro caso es la medida de cantidad de lluvia precipitada sobre una superficie, ya que la lluvia es el tipo de precipitación más frecuente, se mide en mm, y equivale a litros de agua por metro cuadrado de superficie, l/m^2 . El instrumento utilizado para medir la precipitación es el pluviómetro, y en nuestro caso el de cubeta basculante.

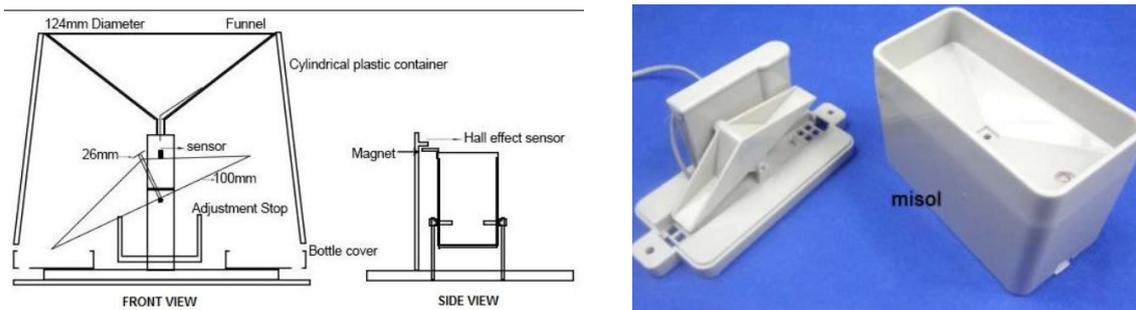


Figura 8: Pluviómetro de cubeta basculante

- **Presión atmosférica**

La presión atmosférica es la fuerza que ejerce la atmósfera sobre la superficie terrestre, a razón de su peso por unidad de superficie. A medida que aumenta la altura, disminuye la presión atmosférica

Su unidad en el SI es el Pascal (Pa), pero también se suele usar el Bar y los milímetros de mercurio (mmHg)

En meteorología se suele medir con barómetros de mercurio y con barómetros aneroides.

- El barómetro de mercurio es un tubo de vidrio que mide 850 mm lleno de mercurio, sobre un recipiente que también contiene mercurio, y se encuentra abierto por su parte inferior y cerrado por la superior. Y al nivel del mar el nivel del interior del tubo alcanza los 760 mm de mercurio, por ello se usa como medida de referencia.
- El barómetro aneroides no lleva mercurio y es el que se utiliza en navegación. Se trata de una caja metálica que se contrae cuando aumenta o disminuye la presión, y debido a ello la aguja que lo integra nos indica la presión a la que es sometido.

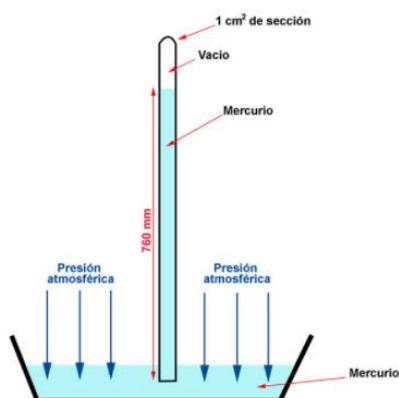


Figura 9: Barómetro de mercurio y aneroides

- **Dirección y velocidad del viento**

El viento es el movimiento irregular del aire atmosférico, y este es debido, en consecuencia, a gran parte de la energía proveniente de la radiación solar, la cual se transforma en energía cinética albergada en los gases de la atmósfera, cuyas moléculas son excitadas y quedan siempre en movimiento.

Se puede considerar al viento como un vector, con magnitud su intensidad y con una dirección. Esta intensidad se suele expresar en nudos, 1 nudo = 1 milla marina por hora o 0,51 m/s.

Pero la medida de estas magnitudes se suele complicar en superficie terrestre, ya que el movimiento del aire suele ser afectado por la presencia de cuerpos naturales o artificiales, focos de calor o la propia naturaleza de la superficie.

Por ello la altura normalizada para su medida con instrumentos es de unos 10 metros sobre el suelo, en superficie llana y descubierta. Para ello, se define un terreno como “descubierto” aquel en el que la distancia entre el resto de instrumentos y cualquier obstáculo es, como mínimo, 10 veces la altura del obstáculo.

La dirección del viento se define como la dirección de donde sopla.

Los instrumentos utilizados para la medida de la intensidad del viento son los anemómetros, pueden ser rotatorios o a presión. En nuestro caso hemos usado el rotatorio de cazoletas, ya que son más baratos y no dependen de la dirección del viento.

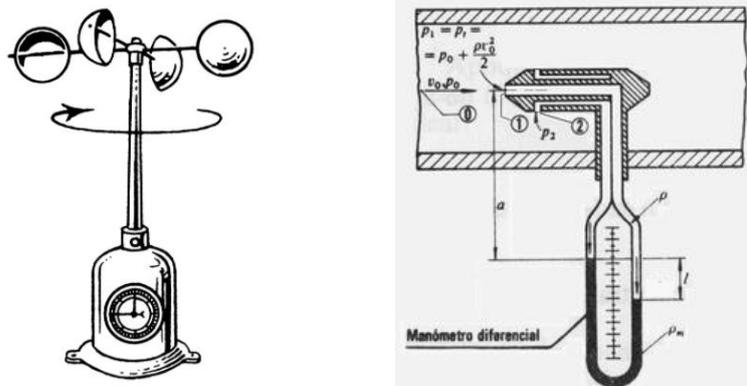


Figura 10: Anemómetro rotatorio y de presión

- **Radiación**

La radiación, también conocida como “irradiación solar” es la energía que incide en la atmósfera emitida en forma de energía electromagnética por el sol, sus unidades en el SI son los Julios por metro cuadrado, $\frac{J}{m^2}$.

Su valor depende de la época del año, de la latitud, del clima y también varía a lo largo del día.

Esta energía se dispersa por la presencia de los gases de la atmósfera, así que la que llega a la superficie se clasifica en radiación global, directa o difusa.

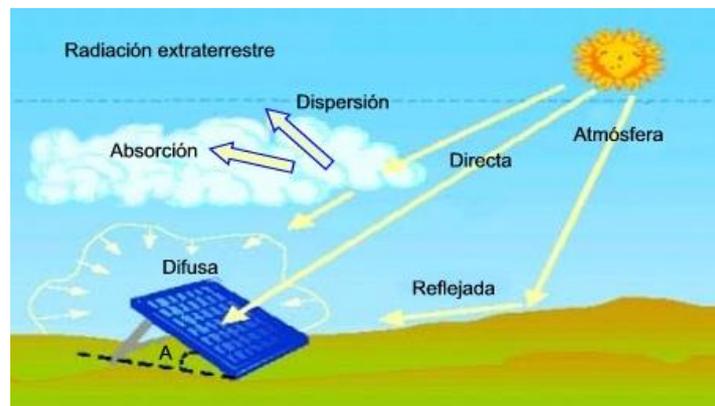


Figura 11: Radiación solar

El Sol proporciona energía presente a lo largo de todo el espectro electromagnético, aunque la radiación con longitud de onda más corta es absorbida por la atmósfera (debido al ozono – O₃), así que no toda la radiación nos llega a la superficie terrestre, y por supuesto, la mayor parte de ella no es visible al ojo humano.

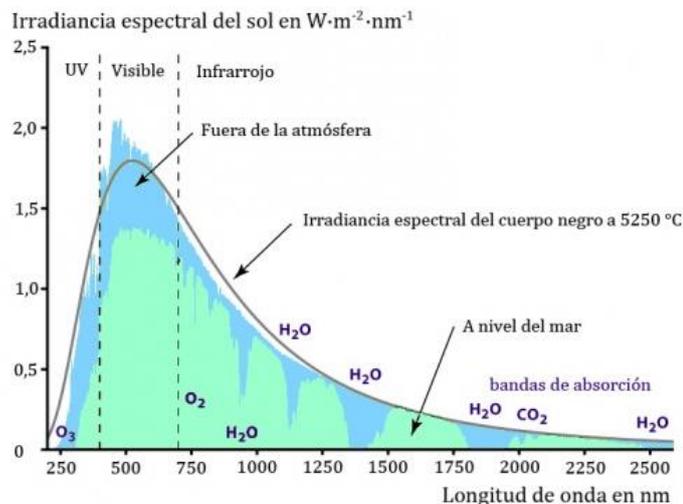


Figura 12: Espectro de la radiación solar

https://cyt-ar.com.ar/cyt-ar/index.php?title=Irradiaci%C3%B3n_solar&mobileaction=toggle_view_mobile

Por todo lo anterior este parámetro es el más complicado de medir de todos, y requiere de un instrumento cuya medida se basa en la diferencia de temperatura entre dos superficies, una blanca y otra negra, las cuales se encuentran dentro de una semiesfera de vidrio, el cual se conoce como piranómetro. Pero el piranómetro solo nos ofrece la medida de radiación global

En la siguiente imagen se puede observar cómo varía la incidencia en la Península Ibérica:

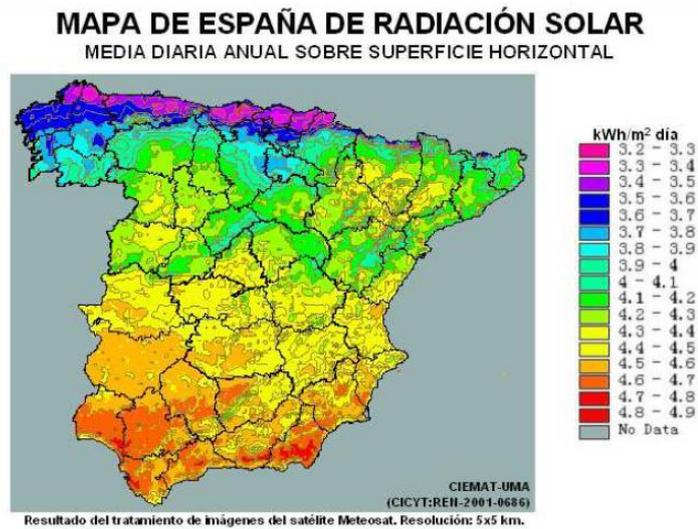


Figura 13: Radiación incidente en la Península

https://rinconeducativo.org/contenidoextra/radiacio/2radiaciones_ionizantes_y_no_ionizantes.html

3. REQUISITOS DEL PROYECTO

3.1 Etapa de alimentación

Al hacer uso de la placa de desarrollo FiPy podemos disfrutar de muchas funcionalidades ya incluidas, en este caso, para la etapa de alimentación ya tiene incluido un regulador de voltaje de 3.5-5.5V que regula la tensión de entrada desde el pin de Vin a 3.3V, suministrando un máximo de 1.2A, de este modo ya disfrutamos de la tensión de entrada requerida para el hardware escogido sin necesidad de usar más conversores de tensión.

Las conexiones de alimentación se encuentran en la Placa de Expansión 3.0, de este modo, para alimentar al conjunto de la estación hay 2 opciones:

1. Usar una batería tipo LiPo con 2 opciones diferentes de carga (100mA y 450mA). La placa de expansión cuenta con un cargador de baterías BQ24040



Figura 14: Cargador de batería BQ24040

2. Utilizar el conector micro-USB integrado en la placa de expansión, ya sea con una batería, con un panel solar o directamente de la red eléctrica.

Como la placa de expansión ya cuenta con un regulador de voltaje, el cargador solar, que es el que nos da energía a través de la batería recargable y el panel solar, irá conectado directamente a través del conector LiPo o utilizando el micro-USB.

En los siguientes apartados analizaremos el consumo de la estación para poder escoger el panel solar que mantendrá nuestra batería cargada, de esta forma mantendremos nuestro nodo abastecido durante todo el día para no perder la comunicación y la transmisión de datos.

Para ello analizaremos la autonomía necesaria de la estación para mantenerla activa todo el día durante los 7 días de la semana, aun cuando no pueda recibir energía directa de la radiación solar. De esta forma, se calcula la potencia exigida por el nodo, se hace el cálculo del panel solar que se va a utilizar y finalmente se hace una valoración del dimensionado del panel.

En los cálculos se tienen en cuenta los casos más desfavorables, tomando como referencia los días en los que la estación recibe menos luz solar debido hay que hay menos horas de sol, como lo son los meses de la época invernal.

Por último, se muestra el cálculo realizado para la elección del supercondensador, que mantendrá en alimentación constante el reloj interno de la estación para no perder los datos de tiempo en caso de que se pierda la alimentación por parte de la batería.

3.2 Consumo del dispositivo

-Consumo dependiendo de los servicios utilizados para la transmisión en el dispositivo:

SERVICIOS	CONSUMO	
Sigfox	192mA	
Punto de Acceso Wifi	126mA	
Cliente Wifi	137mA	
Bluetooth	121mA	
LoRa Receptor	LNA ON, BW=125KHz	9,7mA
	LNA ON, BW=250KHz	10,5mA
	LNA ON, BW=500KHz	12mA
	LNA OFF, BW=125KHz	10,8mA
	LNA OFF, BW=250KHz	11,6mA
	LNA OFF, BW=500KHz	13mA
LoRa Transmisor	13 dBm	28mA
	7dBm	18mA

Tabla 10: Consumo de la placa de desarrollo por modos disponibles

-Tabla resumen de consumo de la estación:

COMPONENTE	CONSUMO	
	V	mA
Sensor DHT22	3,3	2,1
Pluviómetro	3,3	3,3
Anemómetro	3,3	3,3
Veleta	3,3	0,33
Piranómetro SP-110	Autoalimentado	
Placa de desarrollo FiPy	3,3-5	62,7
TOTAL		71,73

Tabla 11: Consumo de la estación

Con los valores anteriores ya podemos analizar el consumo eléctrico diario de la estación, pero para ello se necesita saber el tiempo de funcionamiento de cada servicio.

La estación recibe datos de los sensores y los transmite al servidor mediante LoRa, puesto que el tiempo de muestreo de los sensores nos limita a la hora de recoger variables, pondremos un tiempo de muestreo de 3s (el tiempo de respuesta del sensor de temperatura y humedad es de unos 2s), para así asegurarnos de que se han recogido los datos satisfactoriamente, esto nos facilitaría unas 20 muestras por minuto, así que esos son los valores que consideraremos instantáneos.

Considerando que se reciben datos de todos los sensores a la vez y transmitimos a 13dBm con Lora tenemos un consumo de 99,73mA, suponiendo que alimentamos la placa a 5V, el consumo de cada transmisión:

$$P_{transmitiendo} = V * I = 5V * 99,73mA = 0,5 W$$

Para que no consuma tanto, almacenaremos los datos en una SD y los transmitiremos cada 3h, así tendremos unas 8 transmisiones al día, si cada transmisión dura unos 15s, estaremos transmitiendo a razón de 120seg/día, o lo que es lo mismo, a 0,033h/día.

Como sabemos, la FiPy en modo sleep consume solo 25uA, cuando no se esté transmitiendo estará en este modo de bajo consumo, eso son unos 86280seg/día, o lo que es lo mismo, 23,967h/día. De este modo nos queda:

$$P_{dormida} = V * I_{dormida} = 5V * 25uA = 0,125 mW$$

Este cálculo nos permite definir el consumo en Watios-hora/día de la estación cuando transmite los datos, así obtenemos:

$$P_{W-\frac{h}{día}}(transmitiendo) = 0,5 W * 0,033 \frac{h}{día} = 0,0165 W * h/día$$

$$P_{W-\frac{h}{día}}(dormida) = 0,125 mW * 23,967 \frac{h}{día} = 0,003 W * h/día$$

A este cálculo añadiremos un consumo adicional de 1/3 del consumo total, para disfrutar de potencia en caso de que la placa no esté aportándonos la energía exacta calculada, ya sea debido a la propia placa o a factores externos que produzcan una disminución en la generación de energía. Así, el consumo total mínimo de nuestro nodo asciende a:

$$P_{W-\frac{h}{día}}(Nodo) = P_{W-\frac{h}{día}}(transmitiendo) + P_{W-\frac{h}{día}}(dormida) = 0,0195 W * h/día$$

$$P_{W-\frac{h}{día}}(Total) = 0,0195 + 0,3333 * 0,0195 = 0,026 W * h/día$$

3.3 Cálculo del panel solar y batería

En referencia a las suposiciones hechas unos apartados más arriba para la selección del RTC, procederemos de manera similar para el cálculo del panel solar y la batería.

Nuestra estación tiene que estar operativa las 24 horas del día, 7 días a la semana. Pero lo que nos limitará para suministrar energía a través del panel solar será, como no, el sol.

De esta forma la temporada que más nos limita será la época invernal del año, en concreto el 21 de diciembre, que aproximadamente tiene unas 9,5 horas de sol (en el peor caso), lo que es:

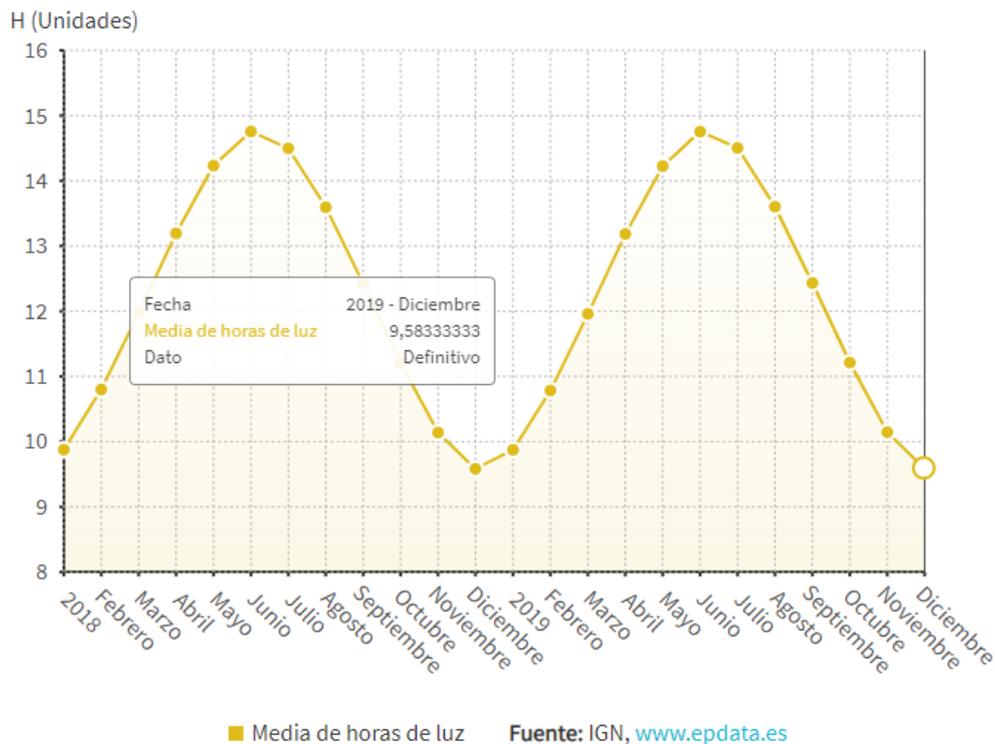


Figura 15: Horas solares en Murcia, 2018-2019

Ref: <https://www.epdata.es/cuantas-horas-luz-hay-provincia/207058ef-1b90-4e95-abe1-624319d83732/murcia/338>

A la vista de estos datos podemos calcular el tamaño del panel solar y la potencia de la batería. Contando con que la estación consume 71,73 mA cada hora, en todo el día consumirá:

$$71,73 \text{ mA} * 24 \text{ h} = 1721,52 \frac{\text{mA} * \text{h}}{\text{dia}}$$

Para las 9,5 horas de sol al día tendríamos un suministro de corriente de:

$$\frac{1721,52}{9,5} = 181,22 \text{ mAh}$$

Pero para asegurarnos sobradamente de que disponemos de suficiente energía deberíamos doblar ese valor, así que nuestro nodo consumirá unos 362,44 mAh.

De este modo, deberemos escoger un panel solar que nos suministre 362,44mAh a 5-7V. Conforme a esto he escogido un panel de Voltaic Systems que nos proporciona 6V a 530 mA, así nos aseguramos que el nodo reciba suficiente energía, y colocaremos una batería de polímero de litio que nos proporciona 1200 mAh.



Figura 16: Batería



Figura 17: Panel Solar

Ref: <https://www.adafruit.com/product/258> ; <https://www.adafruit.com/product/500>

3.4 Cálculos para dimensionado de la placa solar

Ahora toca exponer la irradiación incidente sobre la placa para averiguar la energía generada, para ello se puede hacer uso de una herramienta web, desarrollada por el fondo europeo, el PVGIS.

La base de datos GIS (sistemas de información geográfica) se ha desarrollado usando diversos modelos de radiación solar y técnicas de interpolación, entre otras la GIS GRASS. Las ecuaciones modelos de GIS están basadas conceptualmente en los resultados publicados en el Atlas Europeo de la Radiación Solar (ESRA). Esta estima el haz difuso y reflejado de los componentes del cielo, tanto cuando está despejado como nublado, y tanto la irradiancia como la radiación solar para superficies horizontales e inclinadas. Los valores totales de radiación [$kW * h/m^2$] son obtenidos mediante la integración de los valores de irradiación [W/m^2] calculados en un periodo de tiempo concreto y promediados entre el amanecer y el atardecer.

Ref: <https://ec.europa.eu/jrc/en/pvgis>

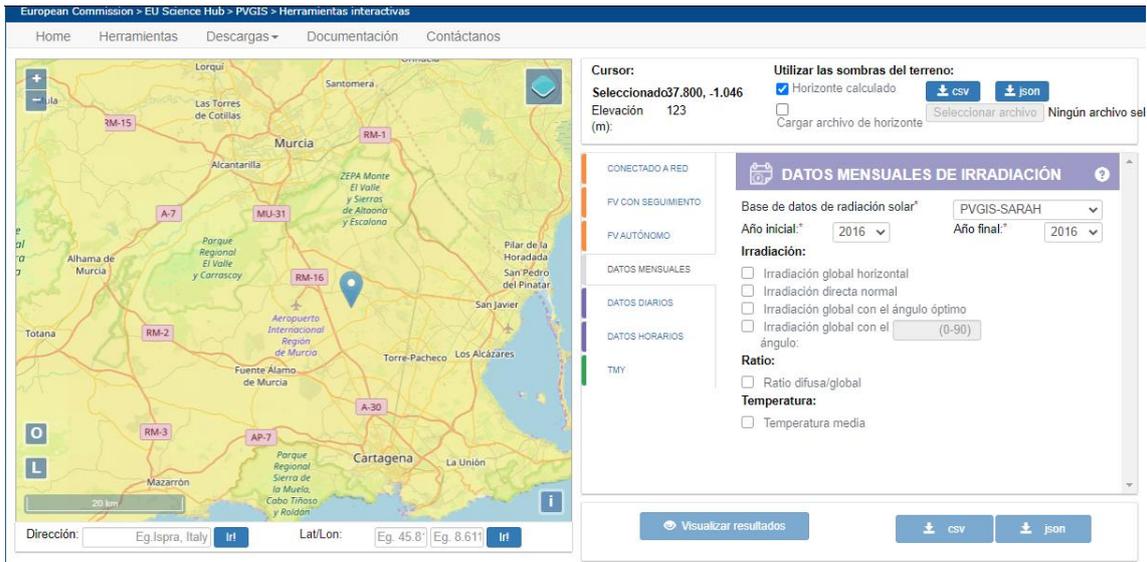


Figura 18: Web de la base de datos del GIS

Para averiguar el grado de inclinación óptimo del panel solar para aprovechar al máximo la radiación solar que incide sobre el mismo utilizaremos los datos facilitados por el PVGIS, desde el servidor de datos PVGIS-SARAH.

Para una irradiación mensual sobre los 27,8º obtenemos los siguientes datos, en $kW * h/m^2$:

Global at user angle	
Mes	2016
Enero	124.43
Febrero	129.9
Marzo	166.09
Abril	176.26
Mayo	198.83
Junio	223.79
Julio	227.39
Agosto	228.45
Septiembre	198.49
Octubre	153.18
Noviembre	116.64
Diciembre	104.13

Tabla 12: Datos obtenidos del servidor PVGIS-SARAH

Nos quedaría lo siguiente:

MESES	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	MEDIA
IRRADIACIÓN (KW*h/m ²)	124,43	129,9	166,09	176,26	198,83	223,79	170,63
	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	
	227,39	228,45	198,49	153,18	116,64	104,13	

Tabla 13: Media de irradiación anual

Si realizamos el dimensionado de acuerdo a la placa escogida, con una potencia nominal de 3.5W y una tensión máxima de 6V a 530mA, podemos calcular la energía generada por la placa, sabiendo que su dimensión es de 154 cm^2 , o lo que es lo mismo, $0,0154 \text{ m}^2$.

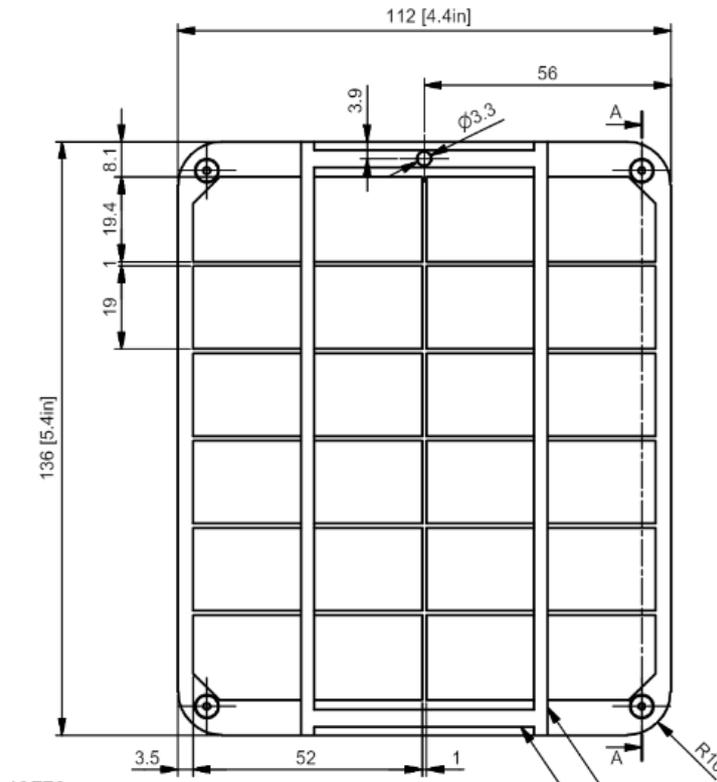


Figura 19: Ficha técnica del panel solar

Así podemos obtener la energía generada por la placa, multiplicando la superficie de la placa solar por la media de la irradiación obtenida a lo largo de los meses del año:

$$\text{Energía} = 0,0154 \text{ m}^2 * 7,109 \frac{\text{kW} * \text{dia}}{\text{m}^2} = 0,1095 \text{kW} * \text{dia} = 109,5 \text{ W} * \text{dia}$$

Ahora procedemos a calcular la hora solar pico. La hora solar pico (HSP), podríamos definirla como una unidad encargada de medir la irradiación solar y definirla como el tiempo (en horas) de una hipotética irradiancia solar constante de 1.000 W/m^2 :

$$HSP = \frac{7109 \frac{\text{W}}{\text{m}^2}}{1000 \text{ W/m}^2} = 7,11$$

La energía del panel para estas HSP sería de la forma:

$$E_p = I_p * HSP * \eta_p$$

Donde: $I_p = \text{Corriente nominal del panel}$

$\eta_p = \text{Rendimiento del panel (85\% - 95\%)}$

$$E_p = 0,33 * 7,11 * 0,9 = 2,11 \text{ W} * \text{h/dia}$$

Pasando el consumo necesario a A*h/día:

$$\text{Consumo} \left(\frac{A * h}{\text{día}} \right) = \frac{2,11 W * h/\text{día}}{6 V} = 0,352 A * h/\text{día}$$

Con los resultados obtenidos en este análisis se puede concluir que la energía que nos proporciona el panel es 13,5 veces superior a la energía que gasta nuestro sistema al día (unos 0,026 W * h/día). Esto sería un caso de sobredimensionamiento, el cual se podría evitar poniendo un panel menos productivo.

Ref: <https://www.grupoelektra.es/blog/wp-content/uploads/2014/10/como-somos-los-delektra-que-son-las-HSP.pdf>

3.5 Cálculos para el RTCC interno

El reloj se alimentará directamente de la Fipy cuando este en uso, y cuando no lo esté desconectaremos la patilla de GND, de esta forma se alimentará de su batería interna, a esto tenemos que añadir un supercondensador para prolongar el uso de la LiPo que usemos para el reloj.

La forma de calcular la capacidad del supercondensador que necesitamos dependiendo del tiempo que vaya a estar desconectado de la alimentación principal (no de la LiPo):

-Como la estación se alimentará de la placa solar, al estar la estación ubicada en Murcia, una zona mayormente soleada la gran parte del año, los meses más nubosos van de noviembre a marzo:

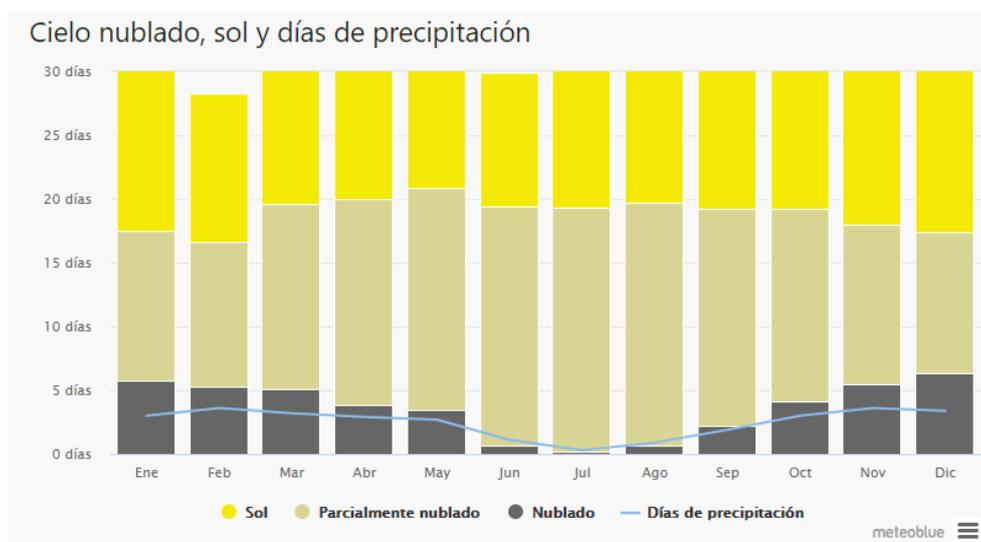


Figura 20: Gráfico de días nublados en la Región de Murcia

En caso de que la placa solar no pueda suministrar energía a la estación debido a los días nublados, necesitamos una alimentación auxiliar, en Murcia los días nublados no suelen durar mucho, podemos observar que la época más nublada se da en diciembre, así que pondremos un caso extremo de 7 días con cielos completamente nublados, sin un rayo de sol, de esta forma calcularemos el condensador necesario que permitirá que nuestra batería descargue energía para mantener el RTC funcionando ese intervalo de tiempo.

Como la FiPy ya consta de un RTC interno, para acceder a él y mantenerlo alimentado, usaremos otro RTC únicamente a modo de comunicación externa con el RTC interno a través del bus I2C. Para ello he elegido el integrado MCP7940N-I/SN, el cual consume 925nA cuando está solamente alimentado con la pila:

DC CHARACTERISTICS			Electrical Characteristics:				
			Industrial (I):		V _{CC} = +1.8V to 5.5V	T _A = -40°C to +85°C	
			Extended (E):		V _{CC} = +1.8V to 5.5V	T _A = -40°C to +125°C	
Param. No.	Sym.	Characteristic	Min.	Typ. ⁽²⁾	Max.	Units	Conditions
D1	V _{IH}	High-level input voltage	0.7 V _{CC}	—	—	V	—
D2	V _{IL}	Low-level input voltage	—	—	0.3 V _{CC} 0.2 V _{CC}	V V	V _{CC} ≥ 2.5V V _{CC} < 2.5V
D3	V _{HYS}	Hysteresis of Schmitt Trigger inputs (SDA, SCL pins)	0.05 V _{CC}	—	—	V	(Note 1)
D4	V _{OL}	Low-level output voltage (MFP, SDA pins)	—	—	0.40	V	I _{OL} = 3.0 mA @ V _{CC} = 4.5V I _{OL} = 2.1 mA @ V _{CC} = 2.5V
D5	I _{LI}	Input leakage current	—	—	±1	μA	V _{IN} = V _{SS} or V _{CC}
D6	I _{LO}	Output leakage current	—	—	±1	μA	V _{OUT} = V _{SS} or V _{CC}
D7	C _{IN} , C _{OUT}	Pin capacitance (SDA, SCL, MFP pins)	—	—	10	pF	V _{CC} = 5.0V (Note 1) T _A = 25°C, f = 1 MHz
D8	C _{OSC}	Oscillator pin capacitance (X1, X2 pins)	—	3	—	pF	(Note 1)
D9	I _{CCREAD} I _{CCWRITE}	SRAM/RTCC register operating current	—	—	300	μA	V _{CC} = 5.5V, SCL = 400 kHz
			—	—	400	μA	V _{CC} = 5.5V, SCL = 400 kHz
D10	I _{CCDAT}	V _{CC} data-retention current (oscillator off)	—	—	1	μA	SCL, SDA, V _{CC} = 5.5V (I-Temp)
			—	—	5	μA	SCL, SDA, V _{CC} = 5.5V (E-Temp)
D11	I _{CCCT}	Timekeeping current	—	1.2	—	μA	V _{CC} = 3.3V (Note 1)
D12	V _{TRIP}	Power-fail switchover voltage	1.3	1.5	1.7	V	—
D13	V _{BAT}	Backup supply voltage range	1.3	—	5.5	V	(Note 1)
D14	I _{BATT}	Timekeeping backup current	—	—	850	nA	V _{BAT} = 1.3V, V _{CC} = V _{SS} (Note 1)
			—	925	1200	nA	V _{BAT} = 3.0V, V _{CC} = V _{SS} (Note 1)
			—	—	9000	nA	V _{BAT} = 5.5V, V _{CC} = V _{SS} (Note 1)

Note 1: This parameter is not tested but ensured by characterization.

Note 2: Typical measurements taken at room temperature.

Tabla 14: MCP7940N-I/SN Datasheet

-Teniendo en cuenta que la caída de tensión en un condensador a corriente constante y dependiendo del tiempo es $\Delta V = \frac{I \cdot \Delta T}{C}$, que la tensión mínima a la que podemos alimentar el RTC para que funcione son 1.3V, queremos tener descargando la pila 1.2V durante 1 semana, de este modo nos queda:

$$1.2V = \frac{925 \times 10^{-9} A * 604800 \text{seg}}{C} \rightarrow C = 466.2 \text{ mF}$$

El condensador calculado es de 466.2 mF pero escogeremos uno estandarizado de unos 470mF.

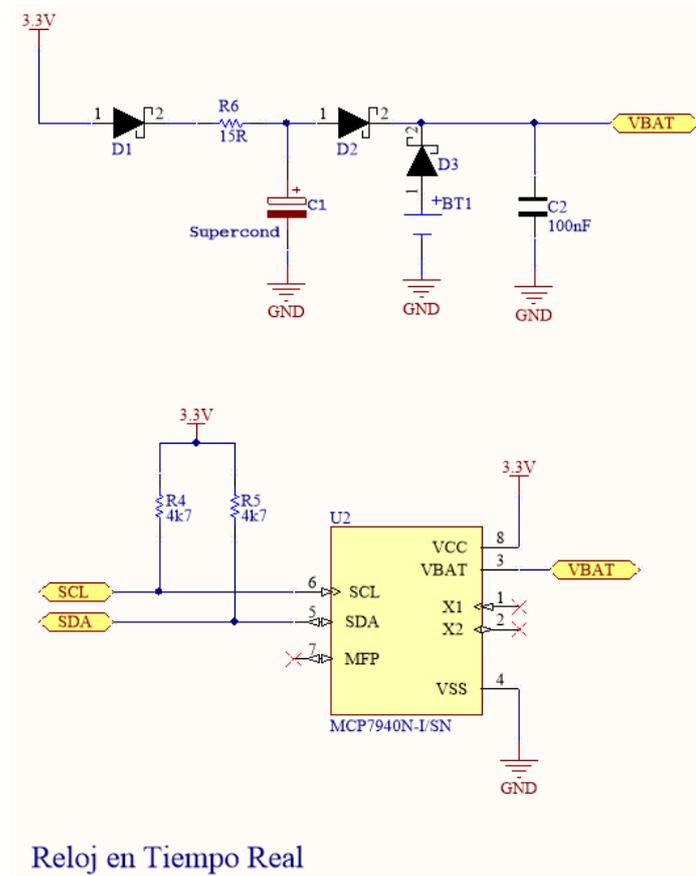


Figura 21: Módulo del RTC

3.4.1 Configuración del RTC

```
#include <MCP7940.h> // Include the MCP7940 RTC library

// Declare all program constants

const uint32_t SERIAL_SPEED{115200}; // Set the baud rate for Serial
I/O
const uint8_t LED_PIN{4}; // Arduino built-
in LED pin number
const uint8_t SPRINTF_BUFFER_SIZE{32}; // Buffer size for sprintf()

// Declare global variables and instantiate classes
```

```

MCP7940_Class MCP7940; // Create an instance of
the MCP7940
char inputBuffer[SPRINTF_BUFFER_SIZE]; // Buffer for sprintf()/
sscanf()

void setup() {
  Serial.begin(SERIAL_SPEED); // Start serial port at Baud rate
#ifdef __AVR_ATmega32U4__
  delay(3000);
#endif
  Serial.print(F("\nStarting SimpleBatteryBackup program\n"));
  Serial.print(F("- Compiled with c++ version "));
  Serial.print(F(__VERSION__));
  Serial.print(F("\n- On "));
  Serial.print(F(__DATE__));
  Serial.print(F(" at "));
  Serial.print(F(__TIME__));
  Serial.print(F("\n"));
  while (!MCP7940.begin()) { // Initialize RTC communications
    Serial.println(F("Unable to find MCP7940N. Checking again in 3s.));
    // Show error and wait
    delay(3000);
  } // of loop until device is located
  Serial.println(F("MCP7940N initialized.));
  if (MCP7940.getPowerFail()) { // Check for a power failure
    Serial.println(F("Power failure mode detected!\n"));
    Serial.print(F("Power failed at "));
    DateTime now = MCP7940.getPowerDown(); // Read w
hen the power failed
    sprintf(inputBuffer, "....-%02d-
%02d %02d:%02d:..", // Use sprintf() to pretty print
now.month(), now.day(), now.hour(), now.minute()); // date/t
ime with leading zeros
    Serial.println(inputBuffer);
    Serial.print(F("Power restored at "));
    now = MCP7940.getPowerUp(); // Read w
hen the power restored
    sprintf(inputBuffer, "....-%02d-
%02d %02d:%02d:..", // Use sprintf() to pretty print
now.month(), now.day(), now.hour(), now.minute()); // date/t
ime with leading zeros
    Serial.println(inputBuffer);
    MCP7940.clearPowerFail(); // Reset the power fail switch
  } else {
    while (!MCP7940.deviceStatus()) { // Turn oscillator on if necessary
      Serial.println(F("Oscillator is off, turning it on.));
      bool deviceStatus = MCP7940.deviceStart(); // Start oscillator and
return state

```

```

        if (!deviceStatus) { // If it didn't start
            Serial.println(F("Oscillator did not start, trying again.)); //
Show error and
            delay(1000); //
wait for a second
        } // of if-then oscillator didn't start
    } // of while the oscillator is off
    MCP7940.adjust(); // Set to library compile Date/Time
    Serial.println(F("Enabling battery backup mode"));
    MCP7940.setBattery(true); // enable battery backup mode
    if (!MCP7940.getBattery()) { // Check if successful
        Serial.println(F("Couldn't set Battery Backup, is this a MCP7940N?"
));
    } // if-then battery mode couldn't be set
    } // of if-then-
else we have detected a priorpower failure
    pinMode(LED_PIN, OUTPUT); // Declare built-in LED as output
} // of method setup()
/*****
*****
** This is the main program for the Arduino IDE, it is an infinite loop and
keeps on repeating. **
*****
*****/
void loop() {
    static uint8_t secs; // store the seconds value
    DateTime now = MCP7940.now(); // get the current time
    if (secs != now.second()) { // Output if seconds have changed
        sprintf(inputBuffer, "%04d-%02d-%02d %02d:%02d:%02d",
            now.year(), // Use sprintf() to pretty print
            now.month(), now.day(), now.hour(), now.minute(),
            now.second()); // date/time with leading
zeros
        Serial.print(inputBuffer); // Display the current
date/time
        secs = now.second(); // Set the counter variable
digitalWrite(LED_PIN, !digitalRead(LED_PIN)); // Toggle the LED
Serial.println();
    } // of if the seconds have changed
    delay(100); // Do nothing for 100 milliseconds
} // of method loop()

```

4. DISPOSITIVOS QUE INTEGRA NUESTRO NODO

INTRODUCCION

En este apartado vamos a abordar el diseño electrónico de la estación, para ello vamos a analizar las características de los componentes que se van a utilizar en nuestra estación, de esta forma quedará todo bien definido.

Después de esto, analizaremos los requerimientos energéticos de los componentes que integran nuestra estación, para poder definir bien la etapa de alimentación que requiere el nodo. De este modo podremos saber la energía que necesita nuestro sistema.

Más adelante mostraremos una opción con componentes más caros, pero de mayor calidad, con los que nuestra estación obtendría un muestreo de datos mayor y mucho más fiables, pero a un mayor costo.

-Características eléctricas de nuestra placa de desarrollo (FiPy):



Figura 22: Placa de desarrollo FiPy

La placa de desarrollo de Pycom es un kit de desarrollo de IoT que permite acceder a las redes LPWAN más recientes en una placa pequeña, basado en PIC-ESP32 y microPython. Dispone de un coprocesador ULP adicional que puede supervisar GPIO, los canales ADC y controlar la mayoría de los periféricos internos durante el modo de hibernación mientras consume solo 25uA.

El ESP32 es la evolución del popular ESP8266. Contiene un procesador Xtensa de 16 bits y dos núcleos que puede operar hasta una velocidad de 240 MHz, 520 KB de RAM interna, soporte para Wifi b/g/n y Bluetooth 2.0 y 4.0, 15 canales A/D, 2 UARTs, 25 salidas PWM, 2 convertidores D/A e interfaces para I2C, SPI y sensores touch pad. Puede ser programado con distintos lenguajes de programación como C/C++ y Python.

Antes de usar la placa se debe actualizar el firmware de la misma, para ello se puentea el pin GND con el pin G23 de la placa, después podremos conectarla al pc y actualizar el firmware.

La FiPy cuenta con un regulador de voltaje ya instalado en placa, admitiendo una tensión de 3.5V-5.5V desde el pin de VIN y regulándolo a 3.3V. Nos advierte de la importancia de que estos 3.3V solo pueden ser usados como salida, y no intentar alimentar a 3.3V a través de este pin ya que podríamos dañar el regulador.

17.0 Electrical Characteristics

17.1 Absolute maximum ratings

Table 25 – Absolute maximum ratings

Parameter	Symbol	Min	Typ.	Max	Unit
Supply Input Voltage	V_{IN}	3.5	–	5.5	V
Supply Output Current	I_{OUT}	–	–	1.2	A
Supply Output Voltage	V_{SV3}	–	3.3	–	V
Storage Temperature	T_{STR}	–	–	–	°C
Operating Temperature	T_{OPR}	–40	–	85	°C
Moisture Sensitivity Level	MSL	–	1	–	–

17.2 Input/Output characteristics

Table 26 – Input/Output characteristics

Parameter	Symbol	Min	Typ.	Max	Unit
Input low voltage	V_{IL}	–0.3	–	$0.25 \times V_{SV3}$	V
Input high voltage	V_{IH}	$0.75 \times V_{SV3}$	–	$V_{SV3} + 0.3$	V
Max Input sink current	I_{SINK}	–	6	12	mA
Input leakage current	I_{IL}	–	–	50	nA
Input pin capacitance	C_{pin}	–	–	2	pF
Output low voltage	V_{OL}	$0.1 \times V_{SV3}$	–	–	V
Output high voltage	V_{OH}	$0.8 \times V_{SV3}$	–	–	V
Max Output source current	I_{SOURCE}	–	6	12	mA

Tabla 15: Características de la FiPy

-Consumo de corriente por modos/funciones de energía medidos a 5V:

Mode	Min	Avg.	Max	Units
Idle (no radios)	-	62.7	-	mA
Sigfox†	-	192	-	mA
LTE Transmit	-	TBD	-	mA
WiFi AP	-	126	-	mA
WiFi client	-	137	-	mA
Bluetooth	-	121	-	mA
Deep sleep*	-	TBD	-	mA

Tabla 16: Consumo de los servicios de la FiPy

-Diagrama de conexiones de la FiPy:

<https://docs.pycom.io/datasheets/development/fipy/>

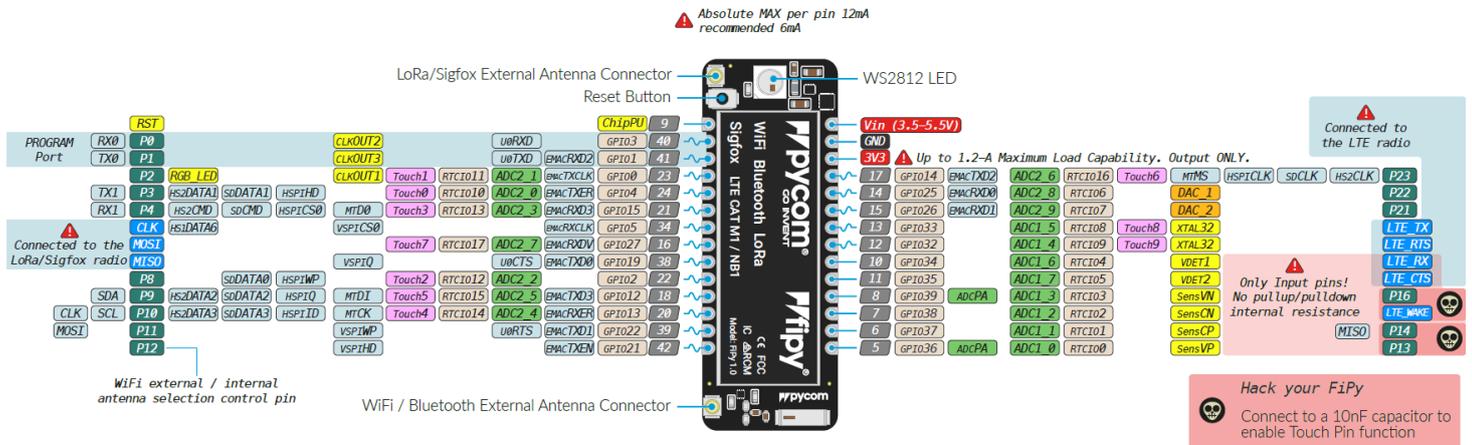
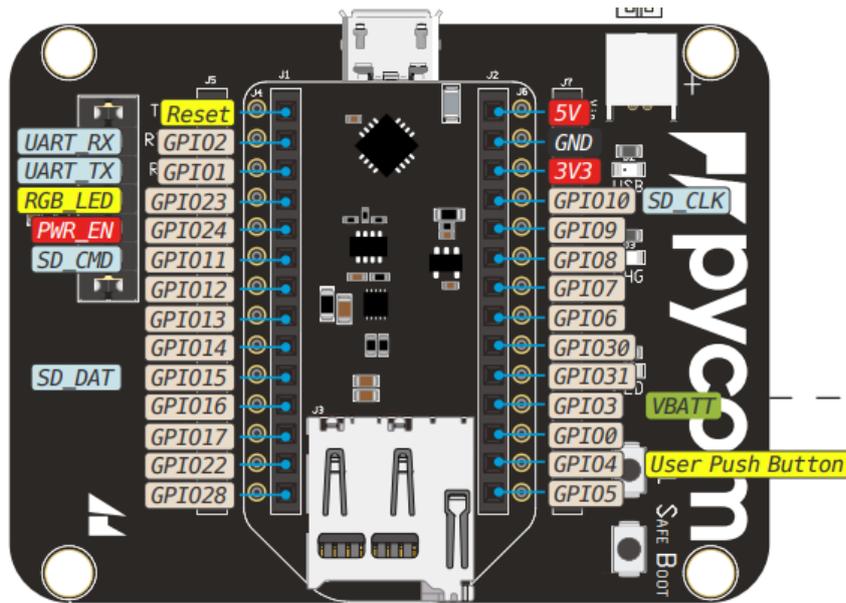


Figura 23: Diagrama de pines de la placa de desarrollo FiPy

La antena LTE tiene el conector justo debajo de la FiPy.

-Diagrama de la placa de expansión:

<https://docs.pycom.io/datasheets/expansionboards/expansion3/>



The expansion board contains 7 jumpers labeled:

- TX: TX UART (GPI01) communication of the development board to the Expansionboard
- RTS: Can be used for UART flow control. Connected to GPI07.
- RX: RX UART (GPI02) communication of the development board to the Expansionboard
- CTS: Can be used for UART flow control. Connected to GPI06.
- BAT: Connect the resistor divider on the Expansionboard to the Voltage monitoring ADC Pin (GPI03)
- LED: Connect the LED on the Expansionboard to GPI016
- CHG: Removing the jumper will decrease battery charging current from 450mA to 100mA



Figura 24: Diagrama de pines de la placa de expansión 3.0

4.1 Sensor de temperatura y humedad dht22

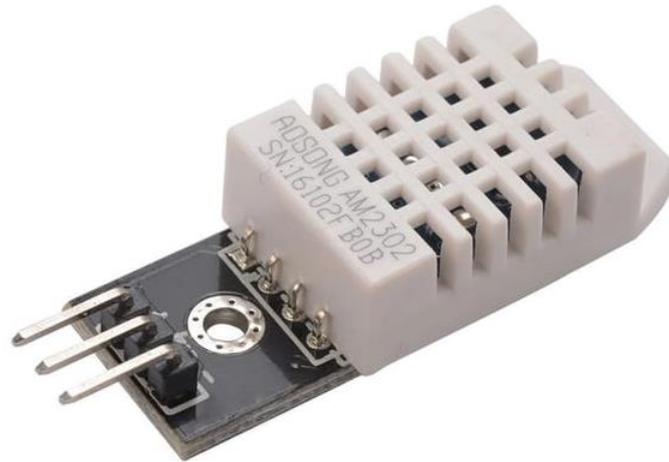


Figura 25: Sensor DHT22

Con el propósito de medir la temperatura y la humedad del aire hemos escogido el sensor DHT22 (digital), ya que dentro de las opciones barajadas es el único que mide temperatura y humedad dentro de un rango aceptable por el precio que ofrece.

En la siguiente imagen se puede apreciar mejor la disposición de sus pines, es la misma que la que nos ofrece su modelo anterior, el DHT11, por lo que no habría problema en caso de querer usar un modelo u otro a fin de intentar reducir el coste al máximo.

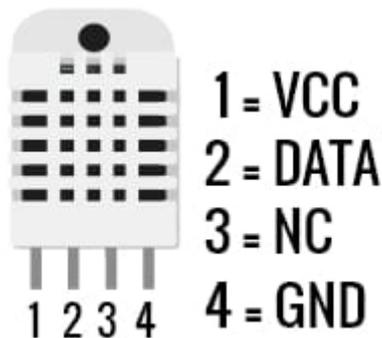


Figura 26: Distribución de pines del DHT22

-Podemos ver que consta de 4 pines:

- El pin 1 es el de VCC, por el cual recibe la energía con la que va a ser alimentado el sensor.
- El pin 2 es el de DATA, a través del cual se transmiten datos en ambos sentidos
- El pin 3 no se conecta porque no se utiliza
- El pin 4 es GND, el pin que va conectado a tierra o 0V.

El sensor admite una tensión de alimentación de 3,3 a 5,5 V en continua, con 5V de tensión recomendada, un consumo de corriente máximo muy reducido de 2,1mA (en medición), y con un tiempo de respuesta máximo de 2 segundos entre mediciones.

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	5.5	V
Current supply	Measuring	1.3	1.5	2.1	mA
	Average	0.5	0.8	1.1	mA
Collecting period	Second	1.7		2	Second

*Collecting period should be : >1.7 second.

Tabla 17: Características eléctricas del sensor DHT22

Atendiendo a las especificaciones de temperatura y humedad podemos observar:

- Rango de medición de temperatura de -40 a 80°C con una precisión de 0.5°C, y 0 a 100%HR de humedad relativa con una precisión del 5%HR.
- Resolución de 0,1°C de temperatura y 0,1% HR.

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Tabla 18: Características eléctricas del DHT22

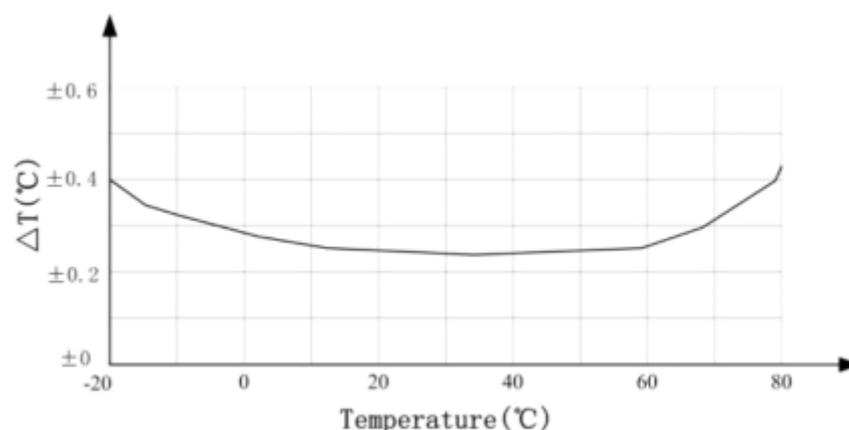


Tabla 19: Máximo error de Temperatura del DHT22

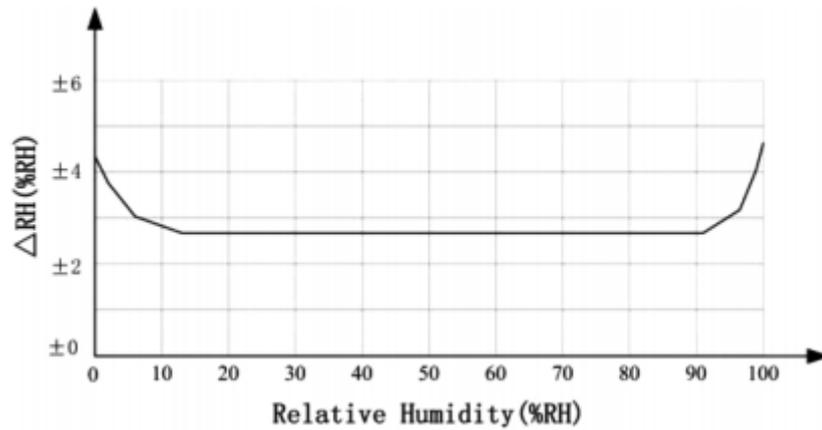


Tabla 20: Error de Humedad Relativa a 25°C del DHT22

El conexionado puede hacerse directamente a la placa con unos jumpers y headers hembra/macho.

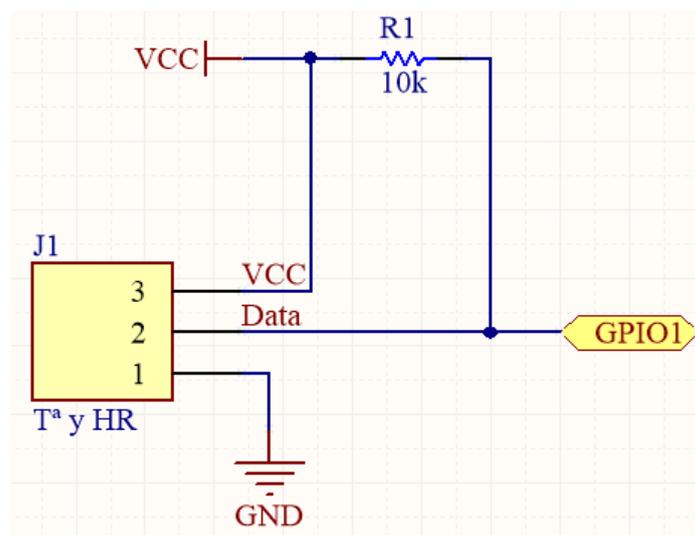


Figura 27: Conexionado del DHT22

La comunicación del sensor DHT22 con el ESP32 de la placa principal se realiza a través del bus I2C.

Para la programación del dispositivo haremos uso del siguiente código:

```

1  #include "DHT.h" // Include DHT Header file for DHT sensor.
2  #define DHTPIN 0 // Analog Pin sensor is connected to pin D2.
3  #define DHTTYPE DHT11
4  DHT dht(DHTPIN, DHTTYPE);
5  void setup()
6  {
7      Serial.begin(9600); // BaudRate para le monitor serial
8      dht.begin();
9      delay(500);
10     // inicializamos el LED de la placa como OUTPUT
11     pinMode(LED_BUILTIN, OUTPUT);
12 }
13
14 float hum=0.0; //Stores humidity value
15 float temp=0.0; //Stores temperature value
16
17 void loop()
18 {
19     //Lee los datos y los almacena en las variables hum y temp
20     hum = dht.readHumidity();
21     temp = dht.readTemperature();
22     //Print temp and humidity values to serial monitor
23     Serial.print("Humedad: ");
24     Serial.print(hum);
25     Serial.print("\n");
26     Serial.print("Temperatura: ");
27     Serial.print(temp);
28     Serial.println(" °C");
29     if (isnan(hum) || isnan(temp))
30     {
31         Serial.println("Fallo al leer el sensor DHT :(");
32         return;
33     }
34     digitalWrite(LED_BUILTIN, HIGH);
35     delay(1000);
36     digitalWrite(LED_BUILTIN, LOW);
37     delay(1000);
38 }

```

4.2 Pluviómetro de cubeta basculante



Figura 28: Pluviómetro de cubeta basculante

Para medir la cantidad de lluvia presente en la zona utilizaremos el pluviómetro de cubeta basculante, este modelo de pluviómetro cuenta con un balancín formado por 2 recipientes, el cual es nuestro sensor electromecánico. Cuando el balancín recibe 0.2794 mm de precipitación se inclinará hacia un lado, de este modo se registrará un pulso de balancín. Este modo de funcionamiento sería parecido al de un interruptor que se abre y cierra con cada balanceo, de modo que un pulso es equivalente a 0.2794 mm de agua precipitada.

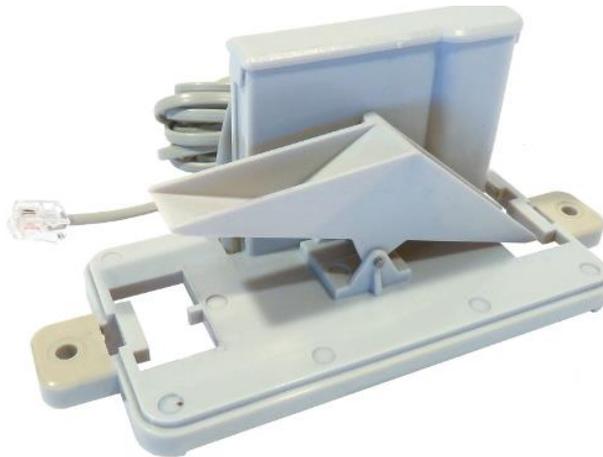


Figura 29: Mecanismo interno de la cubeta

Cada pulso registrado puede ser medido usando pines de interrupción o un contador digital. Los conductores centrales del conector RJ-11 se conectan al interruptor del medidor.

El circuito de acondicionamiento del pluviómetro consta de una configuración de pull-up a la que se le añade un condensador en paralelo, para filtrar la señal del pluviómetro

con el fin de evitar posibles rebotes eléctricos. El pluviómetro se conecta a la salida con una entrada digital (GPIO) del microprocesador.

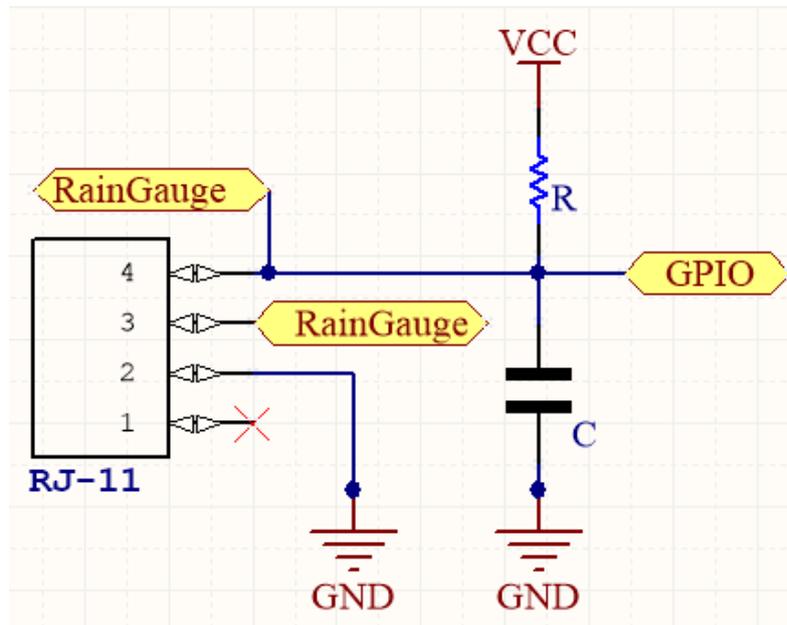


Figura 30: Conexión del pluviómetro

El pluviómetro alimentado a 3,3V con una resistencia de 1kOhm en el divisor de tensión consume 3,3mA de corriente.

El software quedaría de la siguiente forma:

```

1
2 void setup()
3 {
4     Serial.begin(9600);
5     //Configurar el pin para las interrupciones para el pluviometro
6     pinMode(PIN_INTERRUPT, INPUT_PULLUP);
7     attachInterrupt(digitalPinToInterrupt(PIN_INTERRUPT), Impulses, FALLING);
8     Change_clock_source();
9 }
10
11 //El pluviometro se balancea cada 0.2794 mm de agua
12
13 unsigned long rain_time, rain_interval, rain_last=0;
14
15 void Impulses(void){
16
17     REG_EIC_INTFLAG = REG_EIC_INTFLAG; //Se limpia el pin de petición
18     //de interrupciones internas (se pone a 0),
19     //es un acceso a bajo nivel de los registros del micro.
20     rain_time = millis(); //Solo contamos pulso cuando llega
21     //con un periodo de 10ms respecto del pulso anterior,
22     //asi evitamos rebotes y falsos pulsos.
23     rain_interval = raint_time - rain_last;
24     if(rain_interval > 10) //Aqui comprobamos que el pulso cumple los 10ms para contarlo.
25     {
26         rain = (rain+1) % 10000;
27         rain_last = rain_time;
28     }
29 }

```

4.3 Anemómetro



Figura 31: Anemómetro

Este componente está formado por unas cazoletas que provocan un movimiento giratorio sobre su eje cuando el viento incide sobre ellas, lo que hace rotar un imán encerrado en el interior. El imán cierra un interruptor de lengüeta en cada rotación, lo que se traduce en una tensión de salida. De esta forma, podemos tomar una medición a través de los conductores del conector RJ-11 (pines 2 y 3), usando un contador digital o unos pines de interrupción del microcontrolador.

Para convertir esta señal en velocidad de viento medible se usa la conversión de 2,4 km/h = 1 cierre de interruptor/segundo.

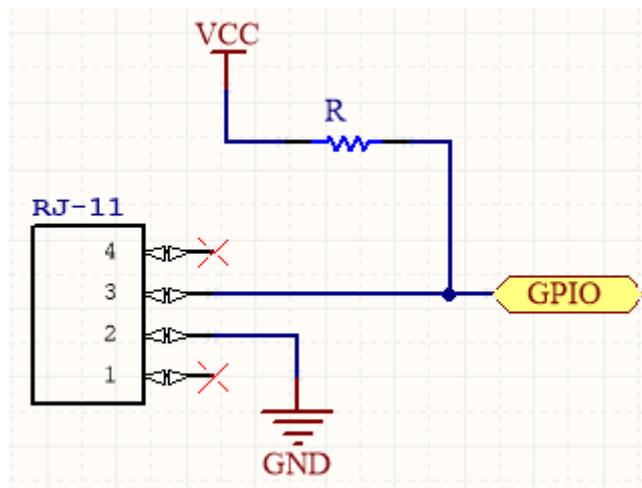


Figura 32: Conexión del anemómetro

El anemómetro alimentado a 3,3V con una resistencia de 1kOhm en el divisor de tensión consume 3,3mA de corriente.

4.4 Veleta



Figura 33: Veleta

La veleta es el sensor más complejo de los cuatro. Contiene 8 interruptores, cada uno conectado a una resistencia diferente. El imán de la veleta puede cerrar 2 interruptores a la vez, permitiendo usar 16 posiciones para definir la dirección. También se usa una resistencia externa para crear un divisor de voltaje, de esta forma obtenemos un voltaje de salida el cual podemos usar para medir con un conversor analógico/digital.

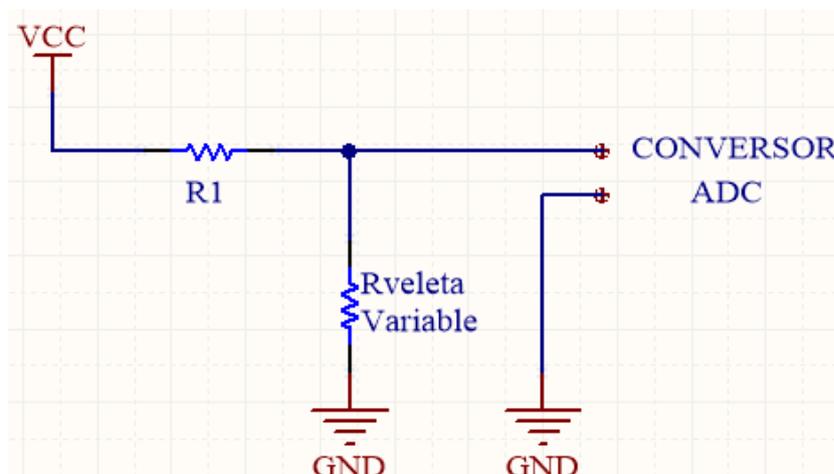


Figura 34: Circuito equivalente veleta

Mirando el circuito equivalente de la veleta se puede apreciar que con un sencillo divisor de tensión se puede obtener la tensión de salida de la veleta, siendo esta tensión dependiente del valor de resistencia que tenga la veleta en cada posición. De esta forma obtenemos las siguientes ecuaciones:

$$I = \frac{V_{in}}{R_1 + R_{veleta}}$$
$$V_{out} = I \cdot R_{veleta} = \frac{R_{veleta}}{R_1 + R_{veleta}} \cdot V_{in}$$

-El máximo consumo de la veleta serán 0,33mA, haciendo el cálculo con la Ley de Ohm, contando con la resistencia de pull-up al circuito (10kOhm) y el valor máximo de la resistencia variable de la veleta (unos 120kOhm), alimentando la veleta a 3,3V.

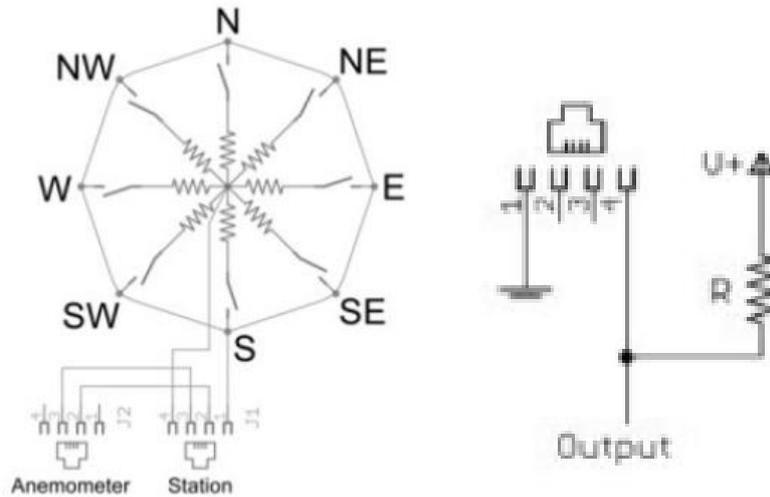


Figura 35: Estructura interna de la veleta y divisor para su medición

-Tensión de salida de la veleta según la dirección del viento:

DIRECCIÓN	GRADOS	RESISTENCIA	Voltaje (V=5V, R=10k)
N	0º	33k	3,84 V
NNE	22,5º	6,57k	1,98 V
NE	45º	8,2k	2,25 V
N	67,5º	891	0,41 V
E	90º	1k	0,45 V
ESE	112,5º	688	0,32 V
SE	135º	2,2k	0,90 V
SSE	157,5º	1,41k	0,62 V
S	180º	3,9k	1,40 V
SSO	202,5º	3,14k	1,19 V
SO	225º	16k	3,08 V
OSO	247,5º	14,12k	2,93 V
O	270º	120k	4,62 V
ONO	292,5º	42,12k	4,04 V
NO	315º	64,9k	4,33 V
NNO	337,5º	21,88kk	3,43 V

Tabla 21: Tensión de salida de la veleta en función de la dirección

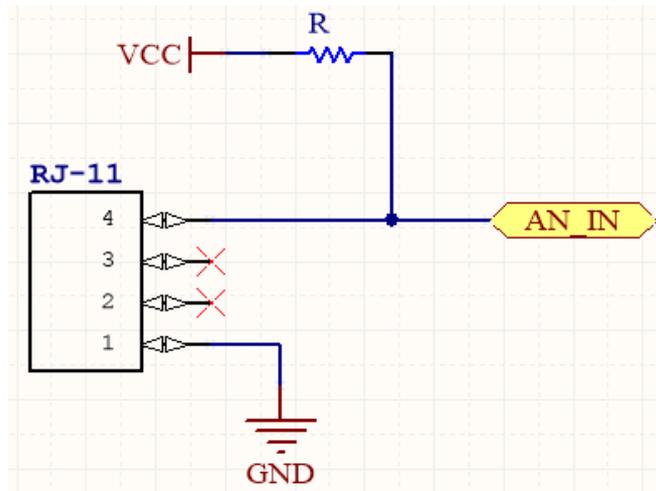


Figura 36: Conexión de la veleta

Para la programación de la veleta y el anemómetro:

```
1 #define Offset 0;
2
3 int ValorVeleta;
4 int Direccion;
5 int CalDireccion; // Valor convertido incluyendo el offset
6 int LastValue;
7
8 void setup() {
9     LastValue = 1;
10    Serial.begin(9600);
11    Serial.println("Valor Anemometro: \n");
12    Serial.println("Direccion: \n");
13    Serial.println("Orientacion: \n");
14 }
15
16 void loop() {
17
18     //map() nos permite transformar un valor del rango 0-1023 (valor analógico de entrada)
19     //al correspondiente en el rango 0-360 (su valor equivalente en grados)
20     ValorVeleta = analogRead(A4);
21     Direccion = map(ValorVeleta, 0, 1023, 0, 360);
22     CalDireccion = Direccion + Offset;
23
24     //Para la calibración en caso de que de una vuelta completa
25     if(CalDireccion > 360)
26         CalDireccion = CalDireccion - 360;
27
28     if(CalDireccion < 0)
29         CalDireccion = CalDireccion + 360;
30 }
```

```

31 // Se muestran los datos en caso de que haya variado mas de 2 grados
32 if(abs(CalDireccion - LastValue) > 5)
33 {
34     Serial.print("|\\t");
35     Serial.print(ValorVeleta);
36     Serial.print("\\t\\n");
37     Serial.print("\\t");
38     Serial.print(CalDireccion);
39     Serial.print("\\t\\n");
40     Serial.print("\\t");
41     Orientacion(CalDireccion);
42     Serial.print("\\t\\n");
43     LastValue = CalDireccion;
44 }
45 }
46
47 // Obtenemos la dirección a donde apunta la veleta
48 void Orientacion(int Direccion) {
49
50     if(direccion < 22)
51         Serial.println("N");
52     else if (direccion < 67)
53         Serial.println("NE");
54     else if (direccion < 112)
55         Serial.println("E");
56     else if (direccion < 157)
57         Serial.println("SE");
58     else if (direccion < 212)
59         Serial.println("S");
60     else if (direccion < 247)
61         Serial.println("SO");
62     else if (direccion < 292)
63         Serial.println("O");
64     else if (direccion < 337)
65         Serial.println("NO");
66     else
67         Serial.println("N");
68 }
69

```

```

70 //ANEMOMETRO
71
72 void loop()
73 {
74   float ValorAnem = analogRead(Ax);
75   Serial.print("Valor analógico =");
76   Serial.println(ValorAnem);
77
78   float voltage = (ValorAnem / 1023) * 5;
79   Serial.print("Voltage = ");
80   Serial.print(voltage);
81   Serial.println(" V");
82
83   float VelViento = mapfloat(voltage, 0.4, 2, 0, 32.4);
84   Serial.print("Velocidad Viento =");
85   Serial.print(VelViento);
86   Serial.println("m/s");
87
88   delay(300);
89 }
90
91 //Esta sería la expresión matemática de la expresión map()
92 float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
93 {
94   return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
95 }

```

4.5 Piranómetro SP-110

El piranómetro de célula de silicio nos ofrece una medida fiable de la irradiancia solar por un precio más asequible que los de termopila, aunque no goce de las ventajas de este último, sí que tiene mayor velocidad de respuesta



Figura 37: Piranómetro

SP-110 de LabFerrer:

	SP-110	SP-510	SP-610
Necesidades energéticas	Autoalimentado	780 0, 15,4 mA consumo de corriente y 185 mW alimentación a 12 V DC (Calentador)	
Salida (Sensibilidad)	0,2 mV / W m ²	0,057 mV / W m ²	0,15 mV 1W m ²
Factor calibración (recíproco de sensibilidad)	5,0 W m ² /mV	17,5 W m ² / mV	6,7 W m ² / mV
Incertidumbre de la calibración	± 5%		
No estabilidad (cambio/año)	< 2% / año		
Tiempo de respuesta	< 1 ms	5 s	
Campo de visión	180°		150°
Intervalo espectral	360 a 1120 nm	385 - 2105 nm	295 - 2685 nm
Respuesta direccional (coseno)	±5% hasta un ángulo de incidencia de 75°	< 30 W m ² hasta un ángulo de incidencia de 80°	< 20% hasta un ángulo de incidencia de 60°
Temperatura de respuesta	0,04 ± 0,04 % por o C	< 5%de-15 a 45°C	
Condiciones ambientales de funcionamiento	-40 a 70°C: 0 a 100% humedad relativa; se puede sumergir en agua hasta 30 m de profundidad	-50 a 50°C: 0 a 100% humedad relativa	
Dimensiones	24 mm diámetro, 28 mm alto	28,7 mm altura, 23,5 mm diámetro	
Peso (con 5m de cable)	90 g		100 g
Cable	5m de cable apantallado de par trenzado. Cubierta de caucho Santoprene (alta resistencia al agua, alta estabilidad UV, flexibilidad en condiciones frías), conector pigtail		
Garantía	4 años contra defectos en materiales y mano de obra		

Tabla 22: Especificaciones del SP-110

Habría que conectar el sensor a un dispositivo de medida (medidor-contador, datalogger, controlador) capaz de medir y visualizar o registrar una señal de milivoltios (mV), se requiere una entrada de medida aproximadamente del rango de 0-250mV para cubrir el rango entero de la longitud de onda de la radiación solar, el

rango de medida de entrada del dispositivo debe coincidir estrechamente con el rango de salida del piranómetro.

El conexionado del sensor a la estación no requiere de ningún acondicionamiento adicional debido a su funcionamiento interno.

SP-110: The sensor is self-powered and applying voltage will damage the sensor.

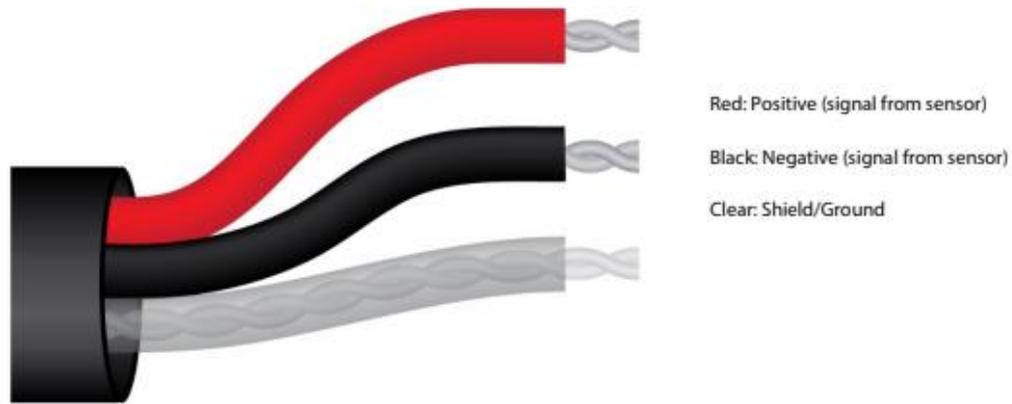


Figura 38: Terminales para conexionado del SP-110

-El conexionado del piranómetro sería directamente a un cabezal de 3 terminales, conectando el cable de señal del sensor a una entrada analógica del microprocesador:

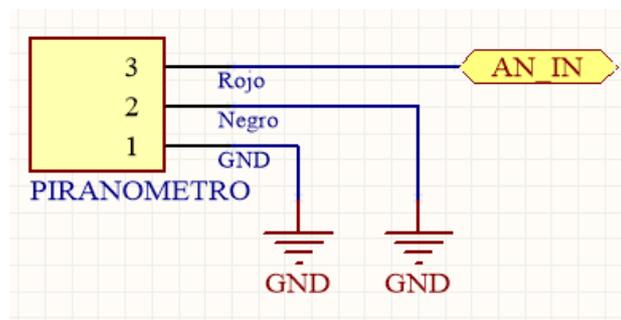


Figura 39: Conexionado del piranómetro

La programación podría ser de la siguiente forma:

```
void lectura_analogica_Radiacion() {  
    int SensorRad = analog.read(Ax); //Lee la entrada del pin analogico "x", siendo x el numero del pin.  
    float Radiacion = ((SensorRad * 1000.0) / 32768) / 6.67; //Convierte la lectura analogica |  
    // (va de 0 a 1023) en un valor de tensión (de 0-5 V)  
    Serial.println("Radiación: \n");  
    Serial.println(Radiacion, 1); //El 1 es para que nos muestre el primer decimal solo  
    delay(2000);  
}
```

4.6 COMPONENTES PARA ESTACIÓN 2.0

4.6.1 Sensor de Tª y Humedad HC2S3 de Campbell Scientific



Figura 40: Sensor HC2S3

El HC2S3 es una sonda digital con una salida de voltaje lineal para la temperatura y la humedad y un puerto serie para la UART. Las señales de voltaje pueden ser medidas con 2 entradas en modo single-ended o en diferencial en el datalogger.

El conversor digital-analógico se usa para generar señales de salida analógicas con 16 bits de resolución.

El cable de Campbell Scientific para el HC2S3 incluye un regulador de voltaje interno que aplica 3.3V a la sonda con una fuente de 5 a 24V. Se recomienda 12V de alimentación para usarla con los datalogger de Campbell Scientific. En aplicaciones donde reducir el consumo es importante, la energía puede apagarse y encenderse para las medidas, siempre que haya un retardo de preparación de 3 segundos. Cambiar la alimentación evita el flujo constante de corriente a través de la tierra del datalogger, lo que puede afectar a la precisión de las medidas de bajo nivel en single-ended.

-Especificaciones (<https://s.campbellsci.com/documents/es/manuals/hc2s3.pdf>):

- Límites de operación: -40 a 100 °C
- Temperatura de Almacenamiento: -50 a 100 °C
- Longitud de la sonda: 85 mm, 183 mm incluyendo el conector
- Diámetro de la sonda: 15 mm
- Peso de la sonda: 10 g
- Filtro: Polietileno, o Teflón opcionalmente.
- Consumo de energía: < 4,3 mA a 5V
< 2 mA a 12V
- Suministro de Voltaje (usando el cable CSI): 5 a 24 Vdc (12 Vdc recomendado)
- Tiempo de inicio:
1.5 s de típico (especificación de Rotronic)

- Recomendaciones de Campbell Scientific:
- 2 s a 60 °C
 - 3 s a 0 °C
 - 4 s a -40 °C
- Corriente máxima de arranque: < 50 mA durante 2us
 - Longitud máxima del cable:
 - 300 m (1000 ft) a 12 V de energía,
 - 3 m (10 ft) a 5 V de energía
 - Salidas analógicas:
 - Offset a 0 V: ± 3 mV (máximo)
 - Desviación de la señal Digital: < ± 1 mV (0.1 °C, 0.1% RH)

-Medición de Temperatura:

- Sensor: PT100 RTD, IEC 751 1/3 Clase B, con señal de acondicionamiento calibrada
- Rango de medida: -50 a 100 °C (por defecto de -40 a 60 °C)
- Rango de la señal de salida: 0 a 1 V
- Precisión a 23 °C: ± 0.1 °C con opciones de configuración estándar
- Estabilidad a largo plazo: < 0.1 °C/año
- Constante de tiempo del sensor [63% de cambio de escalón (1 m/s flujo de aire en el sensor)]:
 - Filtro PE Estándar: ≤ 22 s
 - Filtro de Teflón Opcional: ≤ 30 s
- Precisión sobre el Rango de Medida:

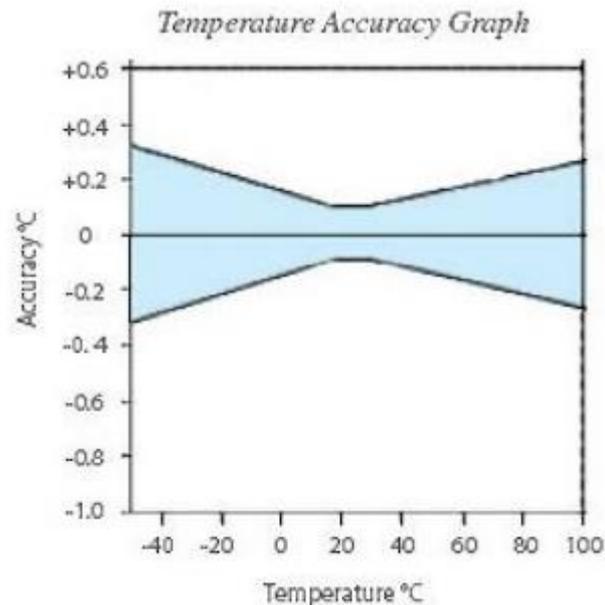


Tabla 23: Precisión de Tª del HC253

-Medición de la Humedad Relativa:

- Sensor: ROTRONIC Hygromer® IN1

- Rango de medida: 0 a 100% sin condensación
- Rango de la señal de salida: 0 a 1.0 Vdc
- Precisión a 23 °C: $\pm 0.8\%$ RH con opciones de configuración estándar
- Estabilidad a largo plazo: $<1\%$ HR por año
- Constante de tiempo del sensor [63% of a 35 to 80% RH (1 m/s flujo de aire en el sensor)]:
 - Filtro PE Estándar: ≤ 22 s
 - Filtro de Teflón Opcional: ≤ 30 s
- Precisión sobre el Rango de Temperatura:

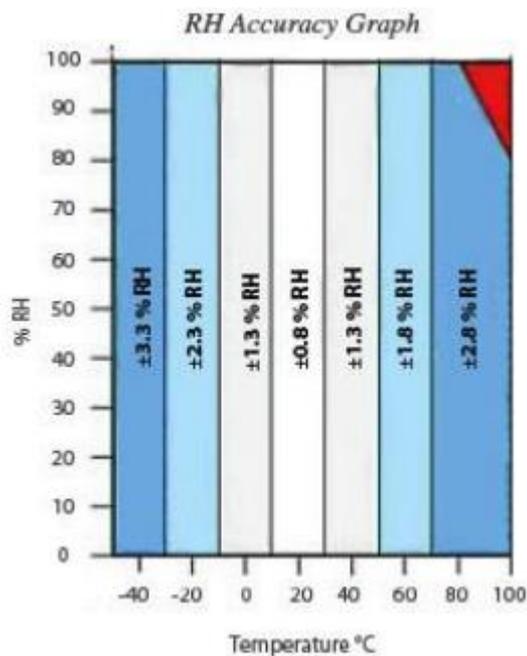


Tabla 24: Precisión de %HR del HC2S3

-Conexión del sensor HC2S3, recomendación de montaje y configuración de la comunicación:

-Recomendación de montaje (con shield):

El sensor debe ser colocado sobre un área abierta y nivelada de al menos 9 m (EPA) de diámetro. La superficie colindante al sensor debería estar cubierta por hierba muy corta o por suelo natural donde no haya hierba. Los sensores deberían estar localizados a una distancia de al menos 4 veces la altura de cualquier construcción cercana y al menos 30 m (EPA) de grandes zonas pavimentadas. El sensor debería protegerse de la radiación térmica y estar adecuadamente ventilado. El filtro se tiene que proteger de la exposición al agua colocándolo en la parte superior del sensor. La naturaleza hidrofóbica del filtro repele la lluvia cuando es leve, pero un flujo de lluvia más fuerte puede introducirse en la estructura porosa del filtro y tardar un tiempo en secarse.

Alturas de medida estándar:

1.5 m (AASC)
 1.25 to 2.0 m (WMO)
 2.0 m (EPA)

Figura 4.25: Altura para el HC2S3

Colocación del sensor en el mástil:



Figura 41: Colocación del HC2S3

-Cableado con el datalogger:

Las conexiones a los datalogger de Campbell Scientific para las medidas de humedad y temperatura usando 2 entradas analógicas single-ended o 2 diferenciales se explican en las tablas de más abajo. El uso de la medida con entrada analógica single-ended se debe usar cuando la longitud del cable sea menor a 6,1 m (20 ft), o si deja de suministrarse energía durante las medidas. De lo contrario, se usa la medición analógica en forma diferencial. Usar el modo single-ended para cables más largos de 6,1 m puede introducir errores de offset en la medida debido a la longitud del cable. Para evitarlo hay que conectar el cable amarillo y gris a la “tierra” del datalogger.

Wire Colour	Wire Function	Datalogger Connection Terminal
Brown	Temperature signal	U configured for single-ended analogue input ¹ , SE (single-ended, analogue-voltage input)
White	Relative humidity signal	U configured for single-ended analogue input, SE
Yellow	Signal reference	↓ (analogue ground)
Grey	Power ground	↓
Clear	Shield	↓
Green	Power	12V or SW12V

¹U channels are automatically configured by the measurement instruction.

Tabla 25: Color del cable, función, y conexión al datalogger para medida en modo single-ended

Wire Colour	Wire Function	Datalogger Connection Terminal
Brown	Temperature signal	U configured for differential analogue input H ¹ , Diff H
Jumper to Yellow	Temperature signal reference	U configured for differential analogue input L, Diff L
White	Relative humidity signal	U configured for differential analogue input H, Diff H
Yellow	Signal reference	U configured for differential analogue input L, Diff L
Grey	Power ground	G
Clear	Shield	↓
Green	Power	12V or SW12V
¹ U channels are automatically configured by the measurement instruction.		

Tabla 26: Color del cable, función, y conexión al datalogger para medida en modo diferencial

-Comunicación:

El HC2S3 puede conectarse a un datalogger de Campbell Scientific a través de una interfaz MD485 RS-485 usando el cable Rotronic E2-05XX-MOD RS485. La configuración para el puerto del RS485 en el MD485 se tiene que configurar para emparejar la configuración del HC2S3, la cual es 19200 baud, No Paridad, 8 Data Bits, 1 Stop bit, y No Flow Control.

Para configurar el dispositivo se usa la "Device Configuration Utility" de Campbell Scientific (software disponible para su descarga gratis). La configuración de los ajustes para el MD485 se hace de la siguiente forma:

MD485 Tab: CS I/O AND RS-485
 CS I/O Tab: SDC Address 7
 RS485 Tab: RS485 baud 19200

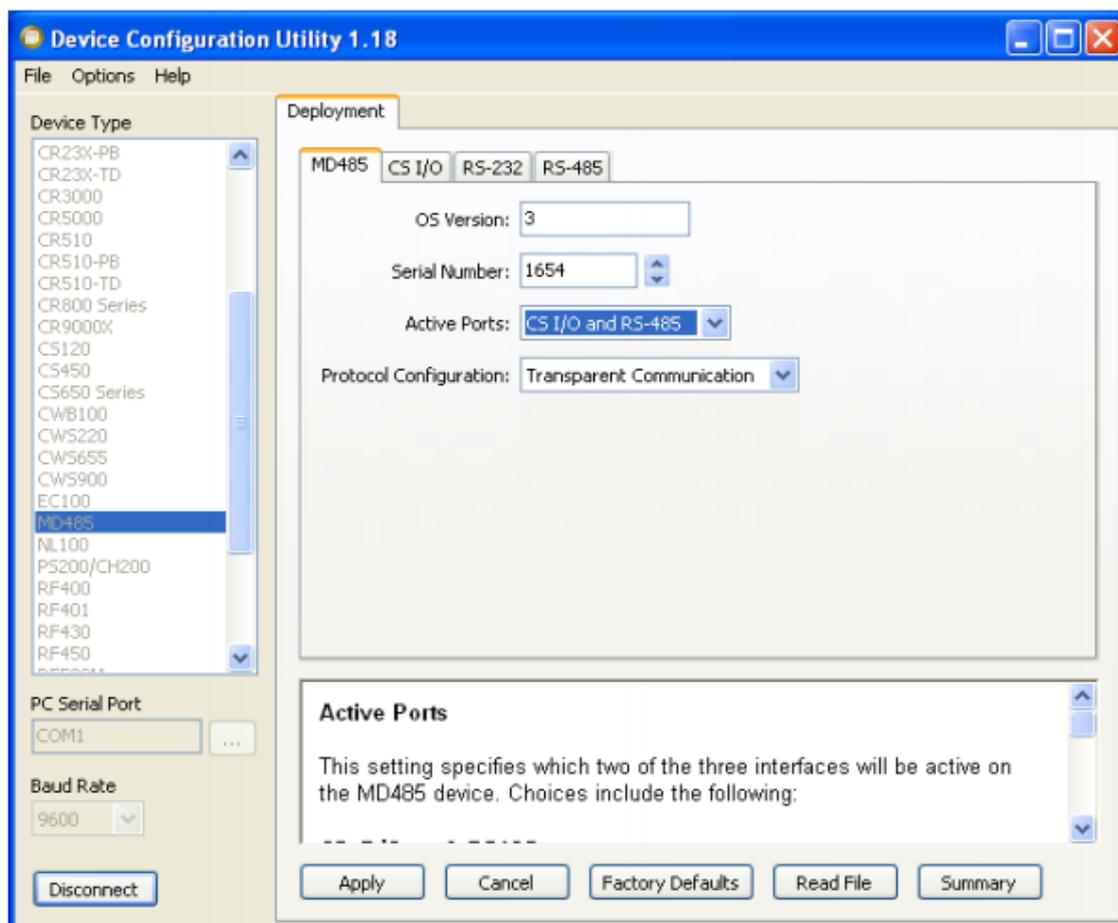


Figura 42: Comunicación del HC2S3 con el Datalogger

-Cableado del sensor:

Sensor Wiring:

E2-05XX-MOD Cable	MD485	CR1000
Blue	A	
Red	B	
Green		12V
Grey/Yellow		G
Clear		Ground Symbol

NOTE

If the Rotronic cable includes brown and white wires (voltage signals for temperature and humidity), Campbell Scientific recommends “capping” them with pn #27749 or equivalent insulated caps to prevent the possibility of shorting.

Connect the CS I/O port of MD485 to CS I/O port on CR1000 with an SC12 cable.

Figura 43: Cableado del HC2S3

-Programación para el CR1000:

El siguiente ejemplo de programa para el CR1000 configura el puerto CS I/O como COMSDC7 usando la instrucción SerialOpen, envía el comando RDD (Read Values) “|{F00RDD}CR” a la sonda, y analiza los valores de la temperatura y la humedad relativa del string de datos devuelto por la sonda.

Ejemplo de programación para el CR1000:

```
'CR1000 Program
'Declare variables

Public SerialIndest As String * 100
Dim String_1 As String
Const CRLF=CHR(13)+CHR(10)
Dim HC2S3_Split(17) As String * 40
Alias HC2S3_Split(2) = RH_Str           'RH string.
Alias HC2S3_Split(6) = TempC_Str       'Temp string.
Alias HC2S3_Split(17) = HC2S3_SN_Str   'HC2S3 serial number string.
Public TempC, RH, NBytesReturned

DataTable (Table1,1,-1)
  DataInterval (0,15,Min,10)
  Average (1,TempC,FP2,False)
  Sample (1,RH,FP2)
EndTable

BeginProg
  SerialOpen (ComSDC7,19200,0,0,100)    'Configure CS I/O port
  String_1 = "|{F00RDD}" + CRLF        'RS485 command to send data

  Scan (5,Sec,0,0)
    SerialFlush (34)
    SerialOut (ComSDC7,String_1,0,2,100) 'Send command to send data
    Delay (0,500,mSec)
    'Get data from probe
    SerialInRecord (ComSDC7,SerialIndest,&H6464,0,&H3B48,NBytesReturned,01)
    'Parse RH and temp from string
    SplitStr (HC2S3_Split(),SerialIndest,";",17,7)
    RH=RH_Str
    TempC=TempC_Str

    CallTable Table1
  NextScan
EndProg
```

Figura 44: Programación del Datalogger

4.6.2 Pluviómetro 52202-L de Young Company

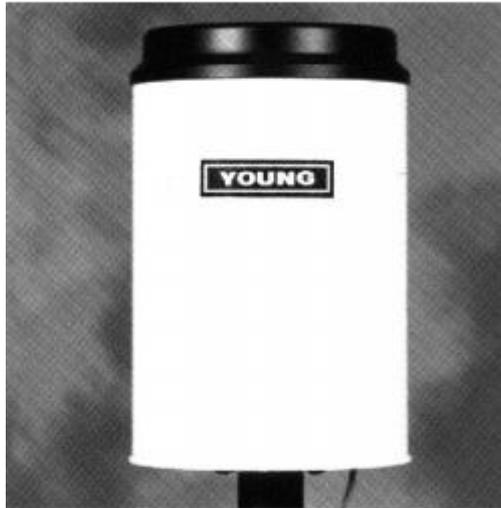


Figura 45: Pluviómetro 52202-L de Young Company

Los calibradores internos del 52202-L están contruidos con dispositivos de nivelación de diana para un ajuste preciso y sencillo.

El diseño del pluviómetro usa el típico método de cubeta basculante para medir la precipitación de forma sencilla y efectiva. La geometría de la cubeta y el material son seleccionados específicamente para expulsar la máxima cantidad de agua, reduciendo así la contaminación y los errores. La precipitación medida se descarga a través de un tubo colector para verificar la cantidad total de lluvia recogida. El uso de la versión calefactada se recomienda en climas donde haya nieve y hielo.

-Especificaciones

(https://s.campbellsci.com/documents/eu/manuals/52202_52203%20-%20382.pdf):

- Área de captación del calibrador: 200 cm²
- Resolución: 0.1 mm por extremidad
- Precisión: 2% hasta 25 mm/hora
3% hasta 50 mm/hora
- Salida: Interruptor de lengüeta magnético
- Índice de contacto: 24V AC/DC, 500 mA max.
- Temperatura de operación: -20°C a +50°C (para versión 52202 calefactada)
- Calentador (52202): Pot. Nominal 18W @ 24V AC
- Potencia requerida (52202): 18-24V AC, 0.75A min.

-Instalación y emplazamiento:

Emplazamiento

Hay que colocar el sensor con cuidado, evitando fuentes de error como árboles o edificaciones cercanas. Una regla de oro para el emplazamiento es que la distancia entre el sensor y cualquier obstáculo debe estar por lo menos al doble de distancia que la altura del obstáculo tenga sobre el nivel del suelo. Para emplazamientos meteorológicos estándar en Reino Unido, la Oficina Meteorológica especifica la altura a la que el borde de un pluviómetro debe estar por encima de la superficie del césped, y los modelos 52202/52203 deben exponerse de forma similar si las medidas son requeridas para comparar con las medidas de los sitios meteorológicos o sinópticos.

Mientras que los modelos 52202/52203 han sido diseñados para montarlos en poste, las investigaciones han enseñado que los pluviómetros obstruyen el flujo de aire y así aceleran el flujo y las turbulencias se incrementan por la parte superior del embudo. Esto puede provocar que se recolecte menos lluvia que de otra forma hubiese caído al suelo.

Instalación

Los pluviómetros 52202/52203 están diseñados específicamente para ser montados en un poste o un tubo de 34mm de diámetro externo. Si se requiere, el ensamblaje del poste se puede obviar y se puede montar el sensor directamente sobre una superficie plana usando unos tornillos que lo aseguren de forma apropiada. La brida de la base donde se monta viene con 3 agujeros en un círculo de paso de 160 mm de diámetro para poder atornillar. El pluviómetro es un instrumento que pesa poco, por lo que se tiene que atornillar con seguridad.

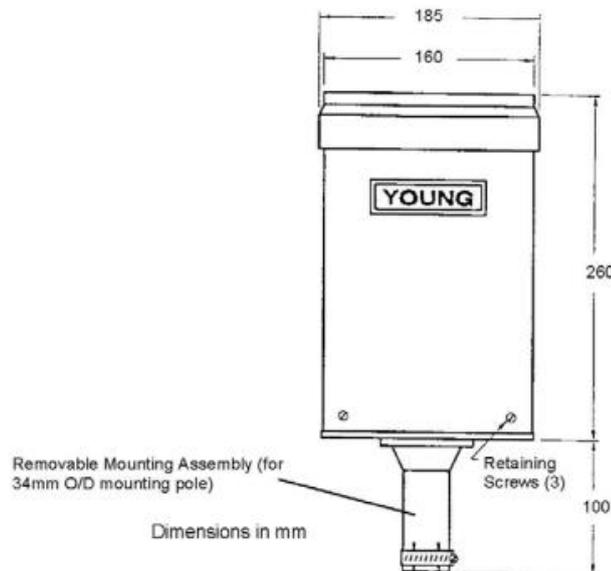


Figura 46: Esquema del Pluviómetro

-Cableado:

El pluviómetro viene con un cable de 4 hilos de 6m, pero se puede alargar si se requiere con un cable apantallado. Para algunas aplicaciones el pluviómetro puede conectarse directamente a una entrada del datalogger de conteo de pulsos como se muestra en la siguiente figura.

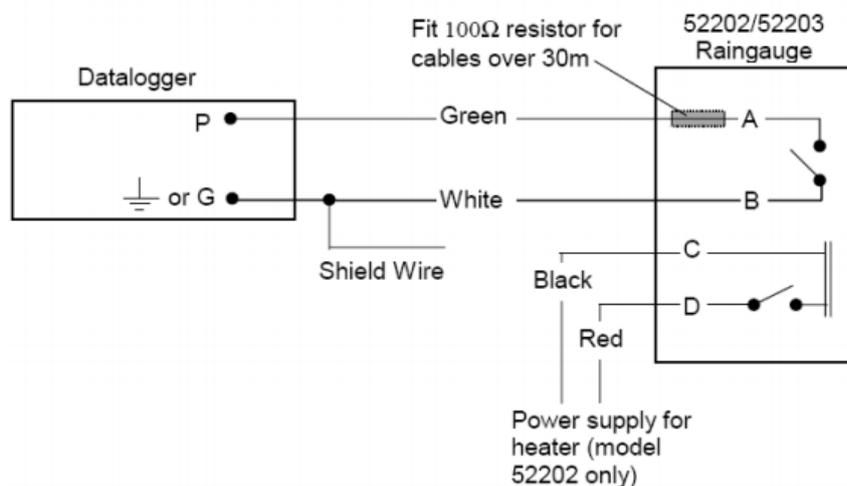


Figura 47: Cableado del Pluviómetro con el Datalogger

Para un cable muy largo, puede existir una capacitancia significativa entre los conductores, los cuales se descargan a través del interruptor de lengüeta cuando éste cierra. Conforme se va acortando la vida del interruptor, se puede inducir un voltaje transitorio en cualquier otro hilo que vaya cercano al cable del pluviómetro cada vez que el indicador se inclina. Si se intenta usar un cable mas largo (mayor de 30m) se debe ajustar una resistencia en serie de 100Ω en el cable tan cerca del sensor como sea posible (se puede observar en la Figura anterior), para proteger el interruptor de lengüeta de formaciones de arcos y prevenir transitorios. Lo mejor es colocar esta resistencia dentro del pluviómetro para protegerlo de cualquier contaminación de la lluvia o la nieve.

Cableado a un Canal de Pulsos

El modelo 52202 suele ir conectado al canal de pulsos del datalogger.

Colour	Description	CR800 CR1000 CR3000 CR5000	CR500 CR510 CR10X	21X CR7 CR23X	CR200(X) Series
Green	Signal	Pulse Channel	Pulse Channel	Pulse Channel	P_SW
White	Signal Return	⊥	G	⊥	⊥
Clear	Shield	⊥	G	⊥	⊥

Tabla 27: Cableado a un Canal de Entrada de Pulsos

Cableado a un Puerto de Control

Cuando un Puerto de Control se usa, el interruptor que vuelve del pluviómetro tiene que ir conectado a +5V en el datalogger.

Colour	Description	CR800 CR1000 CR3000	CR500 CR510	CR10X	CR23X
Green	Signal	Control Port	C2/P3	Control Port	Control Port
White	Signal Return	5 V	5 V	5 V	5 V
Clear	Shield	⊥	⊥	G	⊥

Tabla 28: Cableado del Puerto de Entrada de Control

-Programación del Datalogger:

El programa del datalogger para medir con el sensor se puede descargar directamente usando el "Campbell Scientific's Short Cut Program Builder".

La precipitación se mide usando un Contador de Pulsos con un código de configuración para cerrar el interruptor. El multiplicador usado en la instrucción de conteo de pulsos determina las unidades en las que se reporta la caída de lluvia.

Rain Gauge	0.01 in.	1 in.	0.1 mm	1 mm
52202	0.394	0.00394	1.0	0.1

Tabla 29: Unidades de conteo de lluvia

Ejemplos con el Canal de Pulsos

Los siguientes ejemplos de programas usan un canal de pulsos para leer la salida del indicador del pluviómetro. El ejemplo del CR1000 también funciona con el CR800, CR850, CR3000, y CR5000.

-Ejemplo de Programa con el CR1000:

```

'CR1000
'RM_Young Tipping Blk > P1
'                               Wht > ground
'Cabling for heater goes to 24VAC power supply

Public Rain_mm
Units Rain_mm=mm
DataTable(Rain,True,-1)
  DataInterval(0,60,Min,0)
  Totalize(1,Rain_mm,FP2,0)
EndTable

BeginProg
  Scan(1,Sec,1,0)
    PulseCount(Rain_mm,1,1,2,0,0.1,0)
    CallTable(Rain)
  NextScan
EndProg

```

Figura 48: Programación Datalogger

-Calibración:

Para controlar o recalibrar el pluviómetro, se sugiere lo siguiente:

1. Con el pluviómetro apropiadamente nivelado, precipitar lentamente una cantidad medida de agua dentro del embudo colector a razón de 3,3 ml/min. Esto es equivalente a una precipitación de 10mm/h.
2. Con cuidado contar cuantas veces toca el nivelador; esto debería contar 5 pulsos cada 10ml de agua. Por ejemplo, si usas 100ml de agua la cubeta debería inclinarse 50 veces. Se puede contar el número de inclinaciones manualmente o con el datalogger conectado al pluviómetro.
3. Si el conteo nos genera un error de más del 2%, ajustar los tornillos de calibración para corregir el error; sacar los tornillos si el conteo es bajo, meter los tornillos si el conteo es alto. Siempre ajustando los tornillos por igual.

4.6.3 Anemoveleta 05103 de Young Company



Figura 49: Anemoveleta 05103 de Young Company

El monitor de viento mide la velocidad y dirección de viento horizontal. Originalmente se desarrolló para recoger datos en boyas de uso oceánico, es un aparato robusto y resistente a la corrosión, pero preciso y liviano. La carcasa principal, la punta en cono, la hélice y otras partes internas están construidas en plástico resistente a U.V. con molde de inyección.

La hélice y los ejes verticales usan rodamientos de precisión de acero inoxidable. Los rodamientos llevan un contacto suave de teflón sellado y están rellenos de una grasa con un buen rango de Tª de trabajo para ayudar a excluir la humedad y la contaminación. La rotación de la hélice produce una señal AC sinusoidal de una frecuencia proporcional a la velocidad del viento. Esta señal AC es inducida en una bobina estacionaria por un imán de 6 polos montado en el eje de la hélice.

Una revolución de la hélice son 3 ciclos completos de la señal senoidal. La posición de la veleta es transmitida por un potenciómetro de plástico conductor de precisión de 10K Ohm, el cual requiere de una excitación regulada de voltaje. Con la aplicación de un voltaje constante al potenciómetro, la señal de salida es un voltaje analógico directamente proporcional al ángulo de la dirección del viento.

El anemómetro se suele montar en un tubo de unos 34mm de diámetro. Viene con un anillo de orientación para no perder la dirección de referencia del viento cuando se desmonte para su mantenimiento. Las conexiones eléctricas se hacen en una caja de conexiones colocada en la base.

Su temperatura de trabajo es de -50 a 50°C

-Especificaciones para la velocidad del viento:

- Rango: 0 a 100 m/s

- Sensor: 18 cm de diámetro de hélice de 4-aspas helicoidales en molde de polipropileno
- Inclinación: 29.4 cm de paso de aire por revolución
- Constante de distancia: 2.7 m para un 63% de recuperación
- Umbral de sensibilidad: 1.0 m/s
- Transductor: Bobina estacionaria montada en el centro, con una resistencia de 2K Ohm DC nominal
- Salida del transductor: -Señal AC senoidal inducida por imán giratorio colocado en el eje de la hélice.
- 80 mV pico-pico a 100 rpm.
- 8.0 V pico-pico at 10.000 rpm.
- Frecuencia de salida: 3 ciclos por revolución de la hélice (0.098 m/s por Hz)

-Especificaciones para la dirección del viento:

- Rango: 360° mecánicos
355° eléctricos (5° de apertura)
- Sensor: Veleta nivelada, 38 cm de radio de giro.
- Relación de amortiguación: 0.3
- Distancia de retardo: 1.3 m para recuperación del 50%
- Umbral de sensibilidad: 1.1 m/s (2.5 mph) a 10° de desplazamiento
- Longitud de onda natural amortiguada: 7.4 m
- Longitud de onda natural no amortiguada: 7.2 m
- Transductor: Potenciómetro de plástico conductor de precisión,
10KOhm de resistencia ($\pm 20\%$),
0.25% linealidad,
50mill. revoluciones de esperanza de vida,
- Requisito de excitación del transductor: Voltaje DC regulado, 15 VDC max
- Salida del transductor: Voltaje DC analógico proporcional al ángulo azimuth con un voltaje de excitación regulado aplicado a través del potenciómetro.

5. Diseño de la placa

5.1 Diseño placa de conexionado de sensores

Llegados a este punto ya tenemos diseñado el esquemático de nuestra placa de conexión, la cual irá conectada a la placa de expansión 3.0 para permitir la comunicación con la FiPy.

5.1.1 Circuitstudio

El diseño del esquemático se ha llevado a cabo a través del programa de diseño de PCB CircuitStudio, el cual es una versión más limitada que su versión profesional Altium, la cual cuenta con todas las funcionalidades que le faltan a Circuitstudio.



CIRCUITSTUDIO

Para el diseño del esquemático se ha tenido que incluir una biblioteca de componentes específica ya que el programa no contaba con biblioteca de componentes propia. Esto abarca desde el propio diseño del símbolo del componente, así como su footprint y el modelo 3D asociado.

Los símbolos son los componentes a nivel esquemático, los que conforman el propio esquema eléctrico de la placa del proyecto, por otra parte, los footprint son la apariencia física de los componentes en la placa, donde podemos diseñar el tipo de componente en 2D (forma, tamaño, capa física), así como sus pads (el contacto físico entre componente y placa). También podemos incluir el modelo 3D del componente asociado para después tener una visión real aproximada de cómo va a quedar la PCB.

Los símbolos asociados al esquemático van incluidos en sus bibliotecas de componentes, las cuales se incluyen en formato “.SchLib”, por el otro lado los footprints se incluyen en formato “.PcbLib”

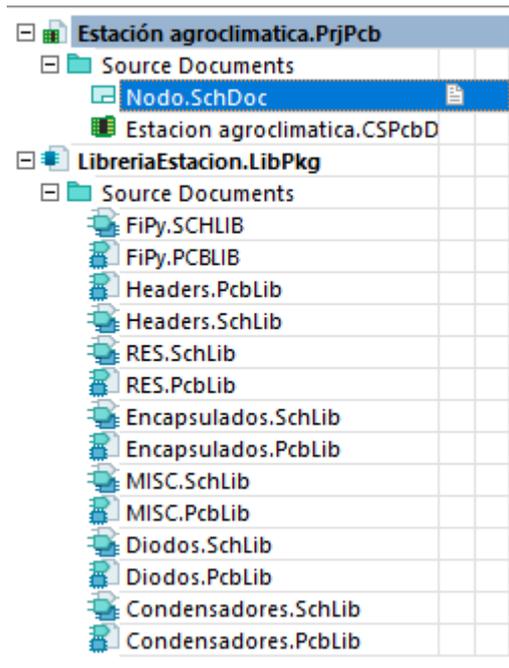
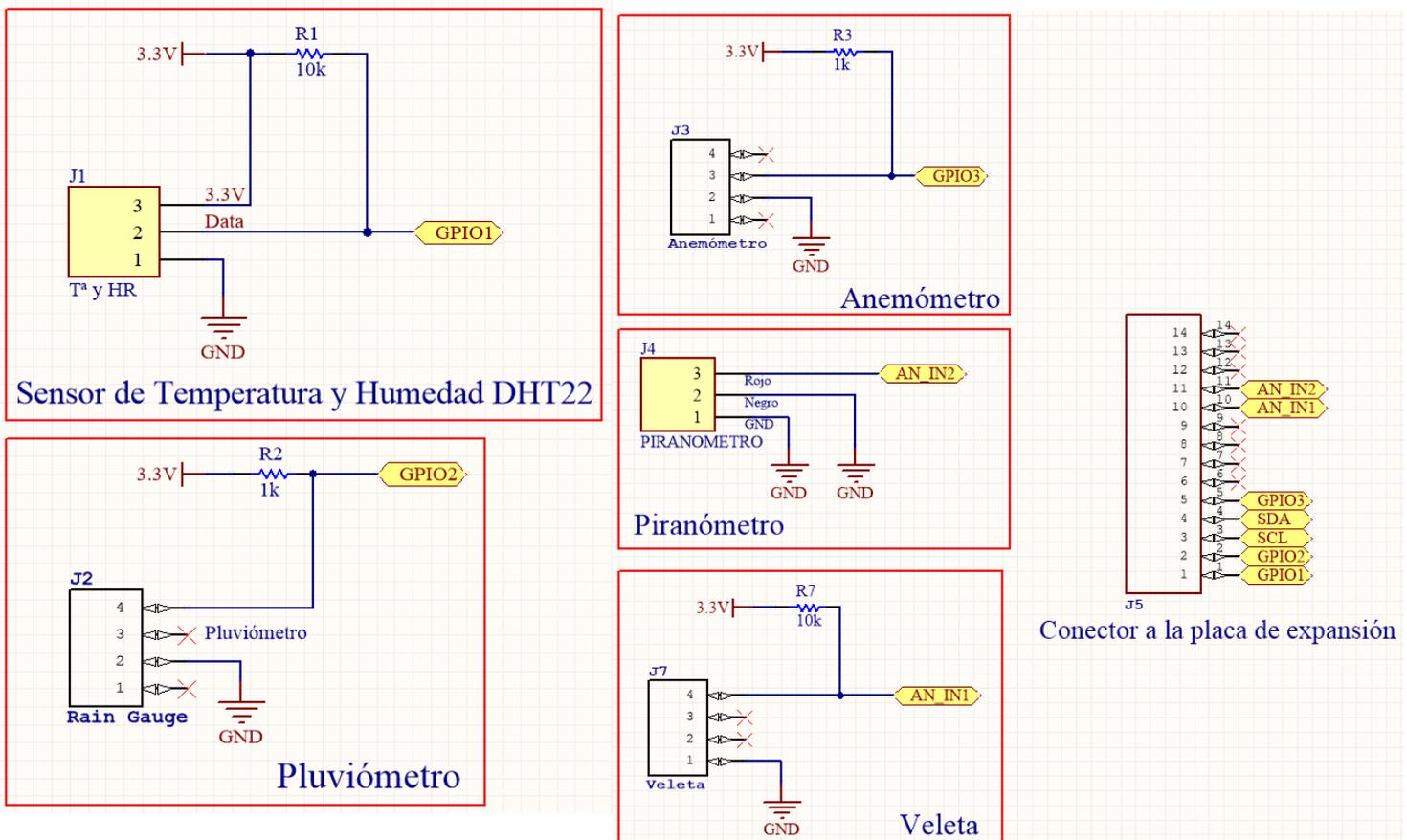


Figura 50: Archivos del proyecto en Circuitstudio

5.1.2 Esquemático

Gran parte del diseño de la placa de conexión de sensores no ha sido necesario, como la fase de alimentación, ya que la placa de desarrollo, la placa de expansión y el cargador solar cuentan con el acondicionamiento necesario para conectar dispositivos sin problemas.



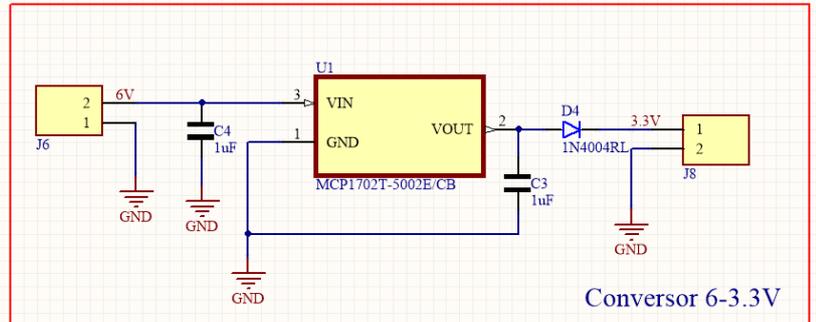
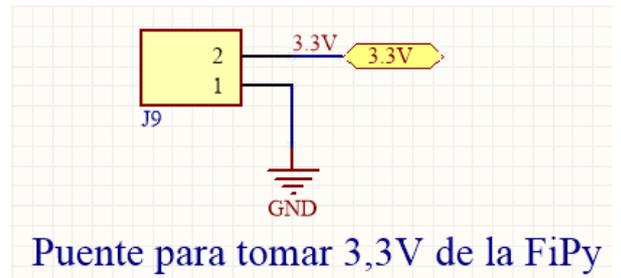
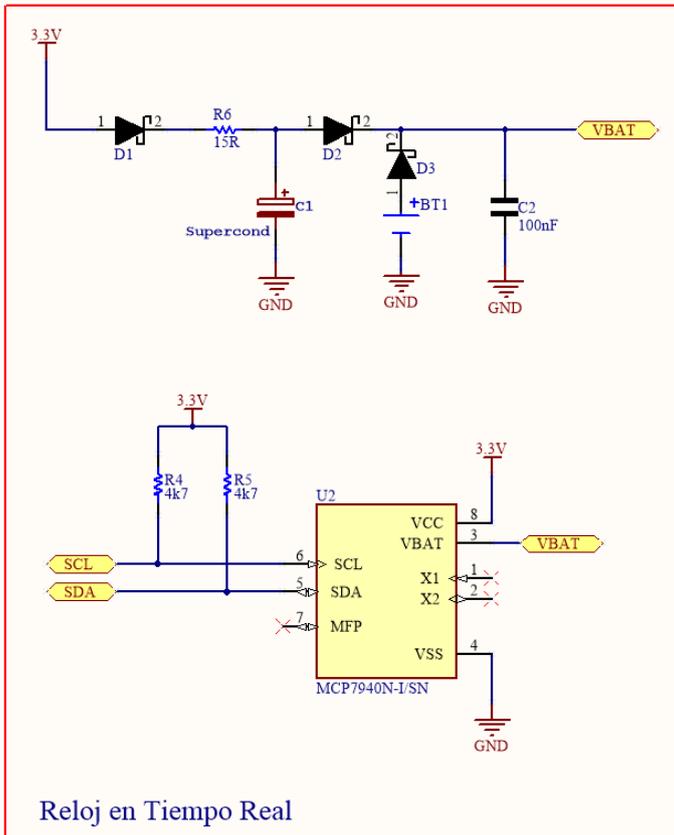


Figura 51: Esquemático placa de conexionado

- El regulador LDO de 6-3,3V se ha colocado como seguridad, en caso de que la estación se encuentre alimentada directamente desde el panel solar nos ofrecerá 6V a pleno rendimiento con sol directo, de esta forma nos aseguramos de que no se dañe ningún componente de la placa.
- El acondicionamiento de los sensores y el módulo RTC ya quedó explicado en el apartado de componentes.
- Se le ha incluido a la placa un conector tipo Header de 14 pines para que quede integrada en la placa de expansión a modo de Shield.
- Para el sensor de temperatura y humedad y para el piranómetro se han incluido conectores tipo PCB de tuerca de 3 contactos.
- Para el resto de sensores se han escogido conectores RJ11

5.1.3 Layout y Modelo 3D

- Disposición de los componentes en la placa y ruteado

Se ha seguido la premisa de los 45º para el flujo óptimo de la corriente por las pistas, al ser una placa con pocos componentes y sin zonas críticas, como procesamientos de señal de comunicaciones o regulación de energía, no se ha tenido en cuenta ninguna disposición específica para la distribución de los componentes, pero sí que se ha usado la ramificación de pistas desde el pin de 3,3V para la alimentación de los sensores.

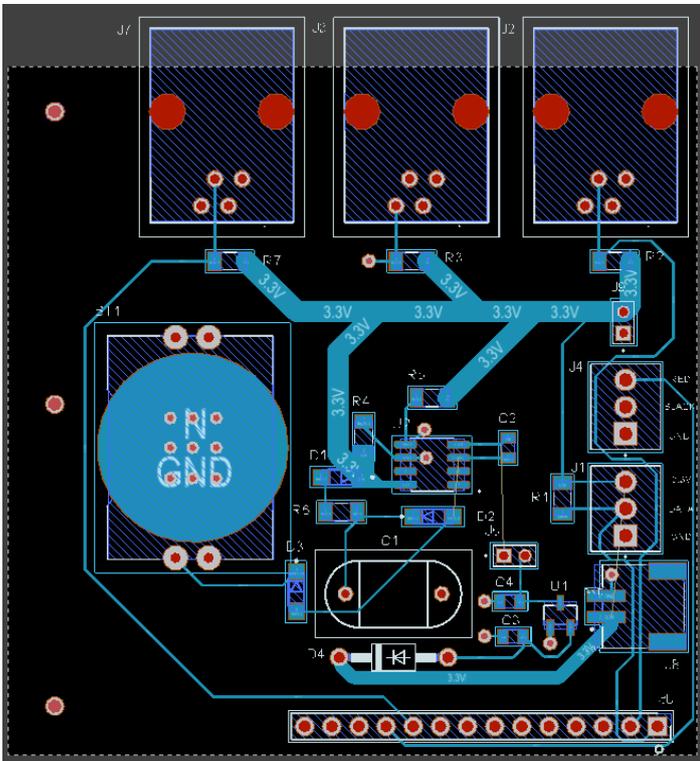


Figura 53: Vista Top del ruteado

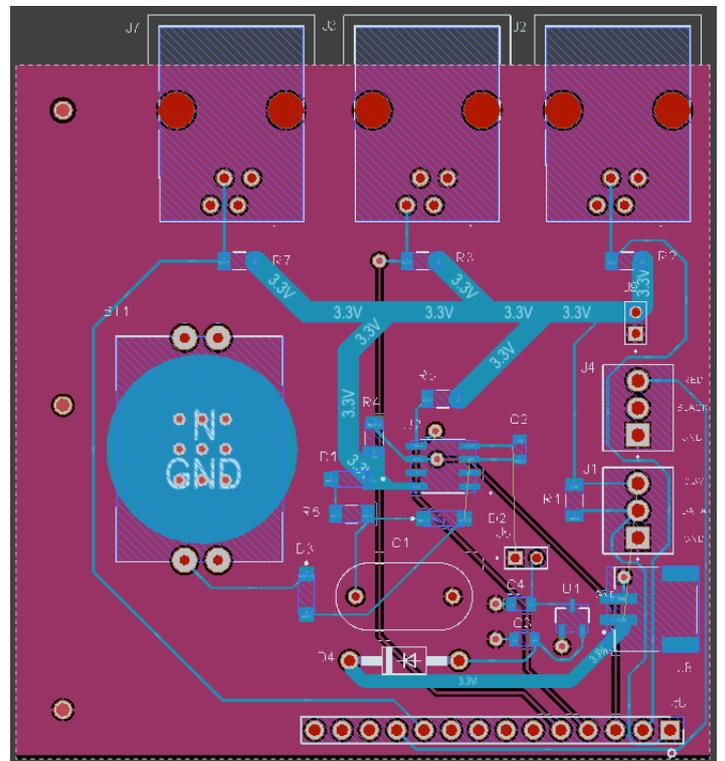


Figura 52: Vista Top + Bottom con cobre incluido

- Modelo 3D

Los footprint y modelos 3D incluidos en la placa han sido obtenidos de Snapeda.

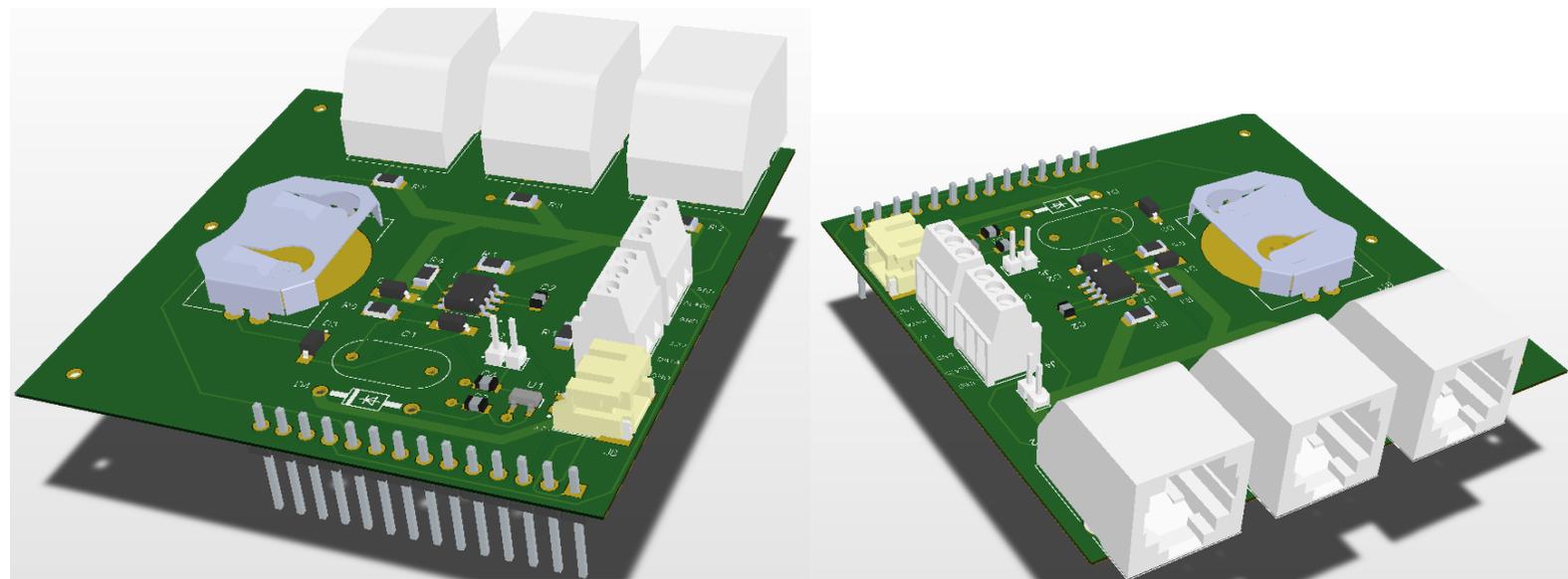
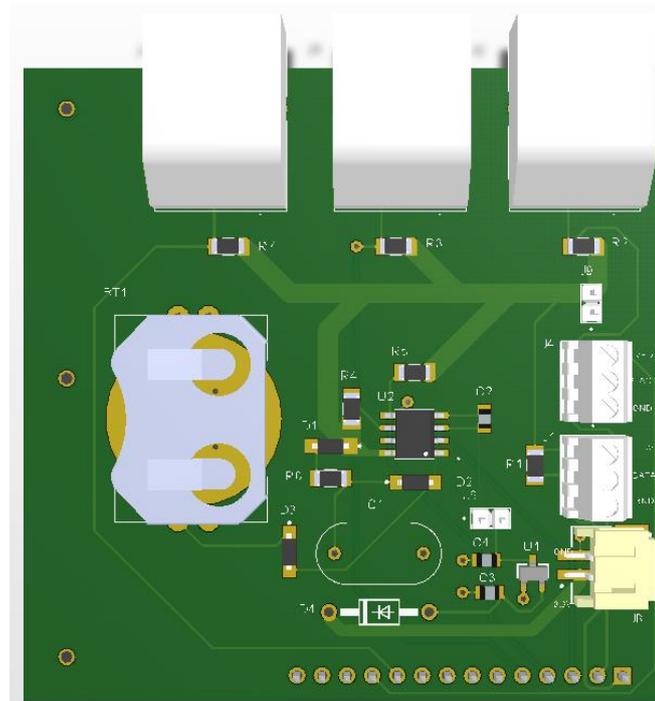


Figura 54: Modelo 3D

6. SOFTWARE

6.1 Protocolos de comunicación

6.1.1 GPIO

GPIO es la abreviatura de General Purpose Input Output, este tipo de entradas no tienen ningún propósito especial definido, es un sistema de entrada/salida de propósito general, este consta de una serie de pines que pueden usarse como entradas/salidas.

-Distribución de los pines de un conector GPIO:

BOARD	GPIO		GPIO	BOARD
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I²C)		DC Power 5v	04
05	GPIO03 (SCL1 , I²C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)		(I²C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Figura 55: Distribución de los GPIO

El número de pines de un conector GPIO suele variar dependiendo de la funcionalidad de la placa donde se aloje.

Se suelen presentar varios pines para el voltaje y masa disponibles en la placa, en la figura anterior se puede observar 2 conexiones disponibles para 3.3V y 5V, y hasta 8 pines para la conexión a masa.

Además de estas conexiones de alimentación se suelen encontrar los pines para la comunicación a nivel de placa de los dispositivos, lo que se conoce como comunicación serie:

1. UART
2. SPI
3. I2C

También podemos encontrar otro tipo de conexiones, como las ADC, DAC o PWM.

- ADC hace referencia al conversor analógico-digital que suele estar integrado en el procesador que controla las tareas en la placa. Cuantifica la señal analógica y transmite la señal codificada en digital.
- DAC por el contrario es la conexión que convierte los datos digitales a formato analógico.
- PWM o Pulse Width Modulation, emula una señal analógica, este tipo de conexiones nos permiten generar señales analógicas desde pines digitales. Mediante esta técnica se hace variar el ancho de pulso de una señal analógica, manteniendo la frecuencia constante, de forma que el promedio de esta tensión será igual al valor analógico que se desea.

6.1.2 UART

El Universal Asynchronous Receiver/ Transmitter es la clave para la transmisión de los sistemas de comunicación serie.

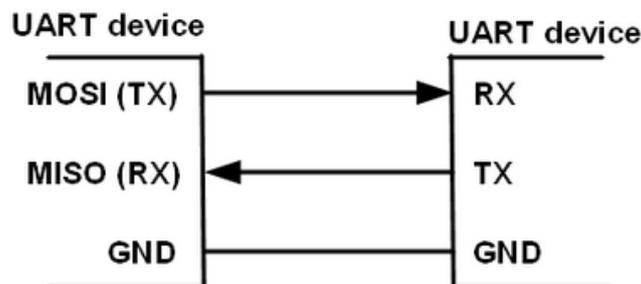


Figura 56: Comunicación UART

Este tipo de hardware se usa para convertir datos serie a paralelo cuando son datos recibidos (entrada), y convertir datos paralelos a datos serie para transmitirlos a otro dispositivo (salida). De este modo la UART se encargará de leer datos cuando llegan, generar y gestionar interrupciones, enviar datos y gestionar los tiempos de bit.

El funcionamiento de la UART siempre es el mismo en todos los dispositivos, esto es lo que facilita la comunicación entre ellos. Podemos ubicarlo con las siglas TX y RX, identificando las líneas de transmisión y recepción respectivamente.

Los datos enviados consisten en bits, normalmente entre 5 y 9, comenzando por el menos significativo.

El orden en la comunicación sería el siguiente:

- Inicio de comunicación – bit de start – siempre es un 0, el cual se mantiene durante un tiempo llamado “tiempo de bit”.
- Envío de datos. Una vez transcurrido el tiempo de bit se comienzan a enviar los datos.
- Fin de la comunicación o liberación de la línea. Una vez enviados los datos hay que indicar que ha concluido la comunicación, esto se hace enviando un “bit de stop”, como nos podemos imaginar se envía un 1 durante el “tiempo de bit”.

Una vez concluye la comunicación, la línea quedaría a la espera de recibir otro mensaje repitiendo el orden anterior.

TIEMPO DE BIT

El tiempo de bit es el tiempo que se mantiene un bit en la línea de transmisión. Está relacionado con el baudio o baudrate, el cual nos indica los bits que se pueden enviar en un segundo. Este parámetro es importante, ya que 2 dispositivos conectados a través de un puerto serie tienen que usar el mismo baudrate para poder comunicarse.

Su cálculo es simple:
$$T_{bit} = \frac{1}{baudrate} \mu s$$

6.1.3 SPI

Este protocolo de comunicación tiene una arquitectura maestro-esclavo. El dispositivo maestro es el que inicia la comunicación, enviando y recibiendo datos con uno o con varios dispositivos a los que llamamos esclavos. En esta topología los dispositivos que son esclavos no pueden iniciar ninguna comunicación, pero tampoco intercambiar datos entre ellos.



Figura 57: Topología SPI

La comunicación entre los dispositivos se realiza en líneas independientes, una línea para la transmisión maestro-esclavo y otra para esclavo-maestro. Este tipo de transmisión se conoce como Full Duplex, esto significa que el maestro puede enviar y recibir datos simultáneamente.

El bus SPI es un bus síncrono, esto se debe a que otra de las líneas de la comunicación es usada por el maestro para proporcionar una señal de reloj, lo que mantiene a todos los dispositivos esclavos sincronizados.

De esta forma podemos distinguir las líneas de la comunicación SPI como:

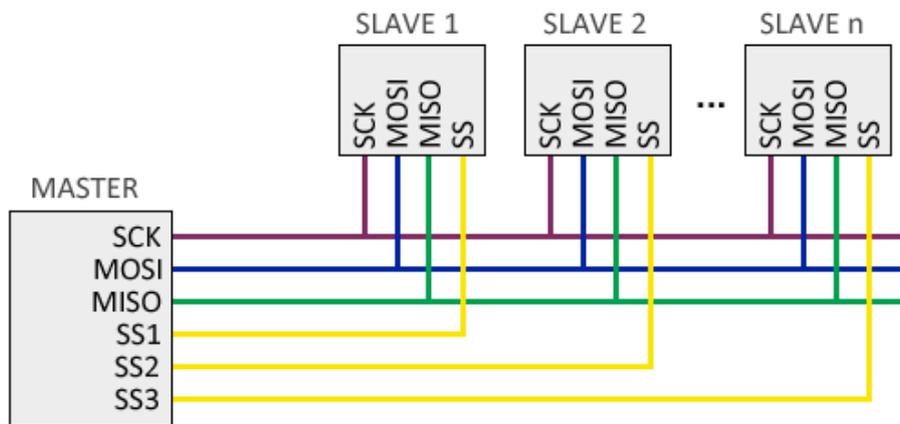


Figura 58: Comunicación SPI

- SCK – Signal Clock. Señal de reloj usada por el maestro
- MOSI – Master Out Slave In. Comunicación Maestro - Esclavo
- MISO – Master In Slave Out. Comunicación Esclavo – Maestro
- SS – Slave Select. Permite seleccionar el dispositivo con el que se va a comunicar el maestro

Aquí se podría observar el problema de añadir una línea por cada dispositivo conectado, lo que en la mayoría de aplicaciones es algo bastante poco deseable, por ello se puede cambiar por una distribución en cascada, aunque ello supone reducir la velocidad de transmisión por el bus debido a que cada comunicación tiene que pasar por varios dispositivos hasta su destino.

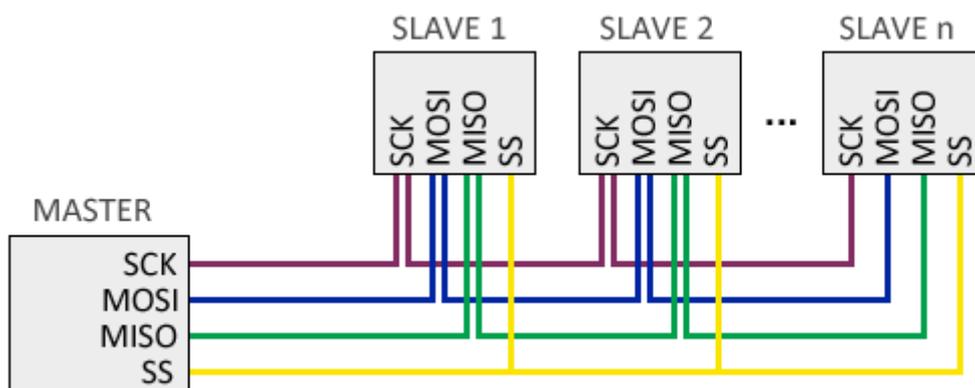


Figura 59: Comunicación SPI

6.1.4 I2C

Este protocolo también tiene una topología maestro-esclavo, pero solo usa dos líneas de transmisión. Por ello el maestro y el esclavo se comunican por el mismo cable, pero el maestro sigue controlando la transmisión con una señal de reloj. La clave de este protocolo no es la selección de esclavos como pasaba con el SPI, sino el direccionamiento.

El direccionamiento consiste en que cada dispositivo se identifica con una dirección, gracias a la que se puede elegir el dispositivo de destino de forma individual. La dirección puede definirse por hardware o por software.

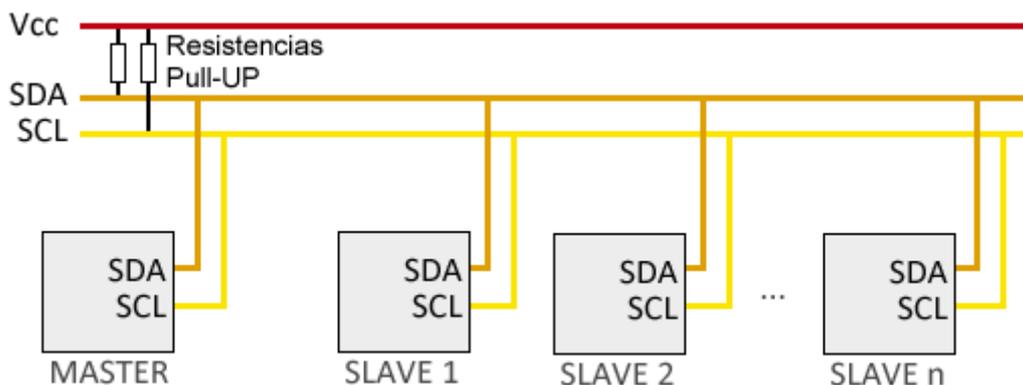


Figura 60: Protocolo I2C

Las dos líneas de transmisión son SDA para el envío de datos, y SCL para la señal de reloj.

La comunicación de este protocolo consiste en que el Maestro es el que inicia la comunicación con los dispositivos Esclavos, de esta forma puede enviar o recibir datos de ellos. Y los Esclavos no pueden iniciar ninguna comunicación con el Maestro, para ello, el Maestro tiene que preguntarles antes, tampoco puede haber comunicación entre esclavos.

Este tipo de bus es un bus síncrono, esto quiere decir que el maestro sincroniza los dispositivos esclavos mediante una señal de reloj.

Otra característica de este protocolo, como podemos observar en la figura anterior, son las resistencias Pull-up colocadas entre las líneas y VCC.

Las resistencias de pull-up o bien son colocadas físicamente o configuradas a través de software. El valor de estas resistencias también afecta a la comunicación, ya que si las resistencias son muy altas la variación en la señal de tensión será muy lenta, lo que limitará la velocidad de comunicación y la distancia a la que se pueden comunicar los dispositivos.

6.2 MQTT



MQTT son las siglas de Message Queue Telemetry Transport, fue creado por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom en 1999 como una buena solución para conectar dispositivos de monitoreo usados en la industria de petróleo y gas a servidores remotos. De esta forma, se planteó una topología de publicación-suscripción basado en el protocolo TCP/IP como base para la comunicación.

- TCP: El Protocolo de Control de Transmisión (TCP) es un protocolo de red importante que permite que dos anfitriones (*hosts*) se conecten e intercambien flujos de datos. TCP garantiza la entrega de datos y paquetes en el mismo orden en que se enviaron.

MQTT consiste en un protocolo de publicación/suscripción, permitiendo que los dispositivos de sensorización se comuniquen consumiendo muy poco ancho de banda, una comunicación bidireccional, reduciendo el consumo y garantizando una gran fiabilidad y relativa seguridad. Todo ello lo ha convertido en el protocolo más utilizado en el campo de Internet de las Cosas (IoT – Internet Of Things).

Este protocolo está diseñado con una topología en estrella, el broker hace de nodo central, y a él se conectan los clientes. De este modo, el broker se encarga de gestionar la red para la transmisión de los mensajes. Los clientes pueden publicar un mensaje en un determinado “topic”, mientras otros clientes pueden suscribirse a ese “topic” y el broker les hará llegar el mensaje publicado.

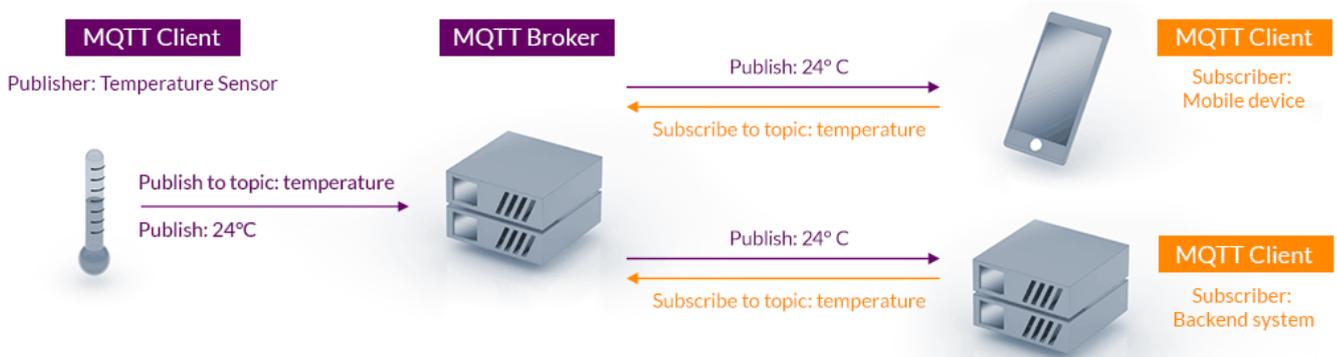


Figura 61: Arquitectura MQTT

En cuanto a seguridad, el protocolo MQTT dispone de diversas medidas de seguridad que se configuran directamente en el broker, incluyendo encriptación en el transporte SSL/TLS y acceso con usuario y contraseña. Aunque este tipo de encriptación a muchos dispositivos les puede suponer una carga adicional.

6.3 Node Red + Adafruit IO

Tras analizar y exponer el conjunto de componentes que integran el nodo de nuestra estación procedemos a detallar las comunicaciones a través de las tecnologías que nos facilita la placa de desarrollo: LoRa, Sigfox, LTE-CAT o NB-IoT.

Una de las facilidades de las que disponemos en la placa de desarrollo FiPy es un servidor FTP, el cual nos permite almacenar y extraer datos del dispositivo, con la ayuda de una tarjeta SD, sin necesidad de instalar ningún servidor de terceros en el mismo. La compañía PyCom recomienda el uso de las librerías Adafruit IO para las aplicaciones con sus dispositivos.



<https://io.adafruit.com/>

Adafruit IO es un servicio gratuito, que nos llega de la mano de la compañía de hardware open-source Adafruit, la cual nos ofrece un servicio en la nube que nos permite monitorizar datos mediante el uso de Feeds y dashboards, todo comunicado mediante protocolo MQTT, además, permite crearnos una cuenta gratis (pero con algunas limitaciones).

De este modo disfrutamos de una potente herramienta de IoT, que nos ofrece apoyo desde su API para almacenamiento de datos, conectividad con dispositivos IoT y visualización de variables en tiempo real.

Desde esta plataforma podremos suscribirnos a un topic mediante un broker MQTT y presentar los datos obtenidos de los sensores a través de Dashboards en Node-red.

El siguiente esquema refleja el sistema de publicación/suscripción en el que se basa el protocolo MQTT. Como ya hemos explicado anteriormente, un dispositivo se encarga

de publicar en un topic, lo cual consiste en mandar un mensaje al broker (servidor), y otros dispositivos podrán recibir el mensaje del publicador siempre y cuando estén suscritos a ese mismo topic.

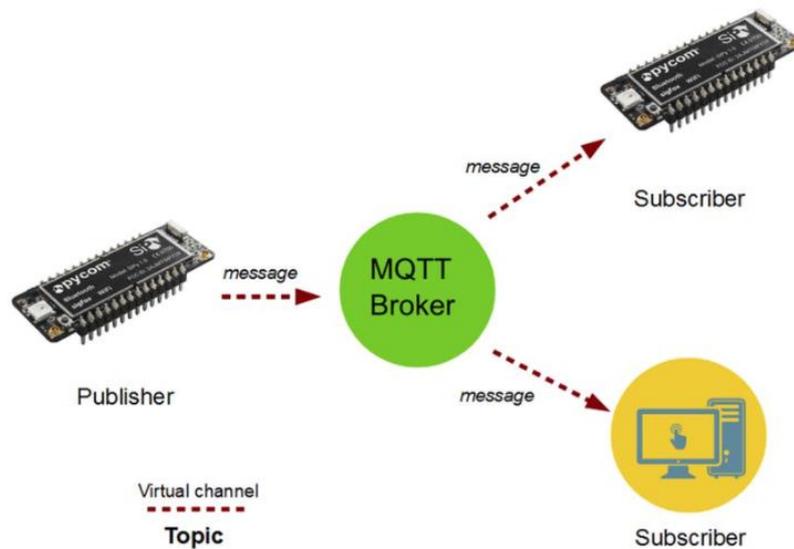
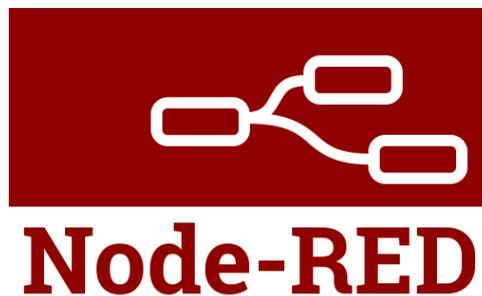


Figura 62: Comunicación MQTT

Para la gestión de datos en tiempo real haremos uso de una potente herramienta de programación en entorno visual, conocida como Node-Red.



<https://nodered.org/>

Es una herramienta de código libre (open source) construida a través de Node.js y Javascript, y nos permite conectar dispositivos, API's y servicios que conforman el IoT (Internet Of Things). La programación es muy sencilla, se realiza desde el lado del servidor, y consiste en la creación de flujos a partir de la conexión de nodos que se pueden arrastrar a través del entorno gráfico de programación para llevar a cabo las tareas deseadas.

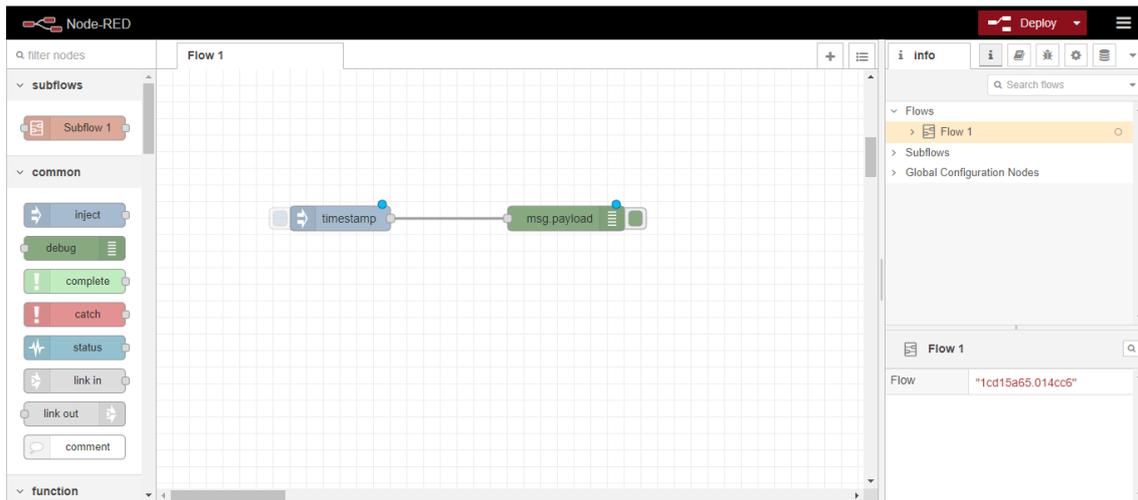


Figura 63: Entorno de trabajo de Node-Red

Además de la librería instalada por defecto, la comunidad ofrece un gran soporte y podemos encontrar una gran cantidad de nodos para distintos tipos de aplicaciones. Esta herramienta tiene un gran potencial, pero una característica a destacar es la creación de Dashboards, uno de los nodos más populares, el cual vamos a aprovechar para nuestro proyecto. Estos consisten en la creación de interfaces dinámicas, con las que podemos interactuar y obtener información de variables en tiempo real, como por ejemplo, la temperatura leída por una sonda:



Figura 64: Dashboard sobre temperatura de una sonda

6.4 Configuración FiPy

Para acceder a la FiPy y poder configurarla se ha elegido el compilador Visual Studio Code, una vez instalado hay que añadir la extensión Pymakr, el entorno de programación que nos ofrece Pycom para sus dispositivos.

Antes de manipular la placa de desarrollo, hay que actualizar su firmware, haciendo un puente entre GND y el pin G23, de esta forma podremos cargarle la última versión disponible del firmware y ya estará lista para trabajar.

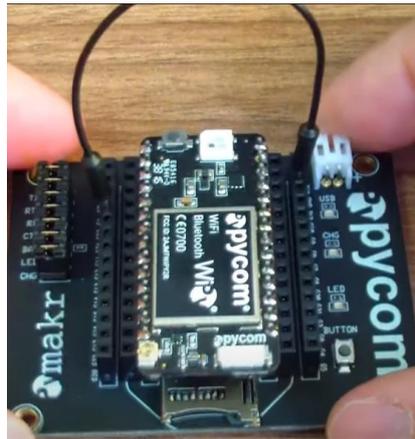


Figura 65: Puente en la FiPy para actualización

Una vez actualizado el firmware, para poder conectarnos al dispositivo tenemos que acceder desde el punto de acceso Wi-Fi que tiene por defecto la placa. Aparecerá en redes disponibles como “Fipy-wlan-XXX” y tendremos que introducir la contraseña “www.pycom.io”.

Para comunicarnos con el servidor FTP de la FiPy a través de Telnet podemos utilizar una aplicación externa, como por ejemplo el cliente Putty, el cual nos permite mantener un acceso Telnet al servidor de nuestra placa. Todo esto es posible ya que MycroPython no necesita ser compilado para ser ejecutado, si no que el código se escribe y se ejecuta inmediatamente una línea después de otra, es lo que se conoce como REPL, una abreviatura de Read-Evaluate-Print-Loop.

Para poder acceder al servidor FTP (File Transfer Protocol) de la FiPy tendremos que comunicarnos a través de Telnet mediante una aplicación externa, como por ejemplo el cliente Putty, el cual es un cliente que nos permite mantener un acceso Telnet al servidor de nuestra placa. Otra forma mucho más completa sería la que nos recomienda Pycom, a través del IDE de Atom, e instalando el paquete Pymakr.

Una vez tengamos el dispositivo conectado podemos cambiar el método de acceso, desde “Settings” en “Device adress” cambiamos la dirección del dispositivo por su puerto de comunicación serial, normalmente “COM3” o “COM4”. De esta forma no

tendremos que acceder vía Wi-Fi, podemos desconectarnos del punto de acceso inicial.

Las librerías de Pycom están disponibles en GitHub: <https://github.com/pycom/pycom-libraries>

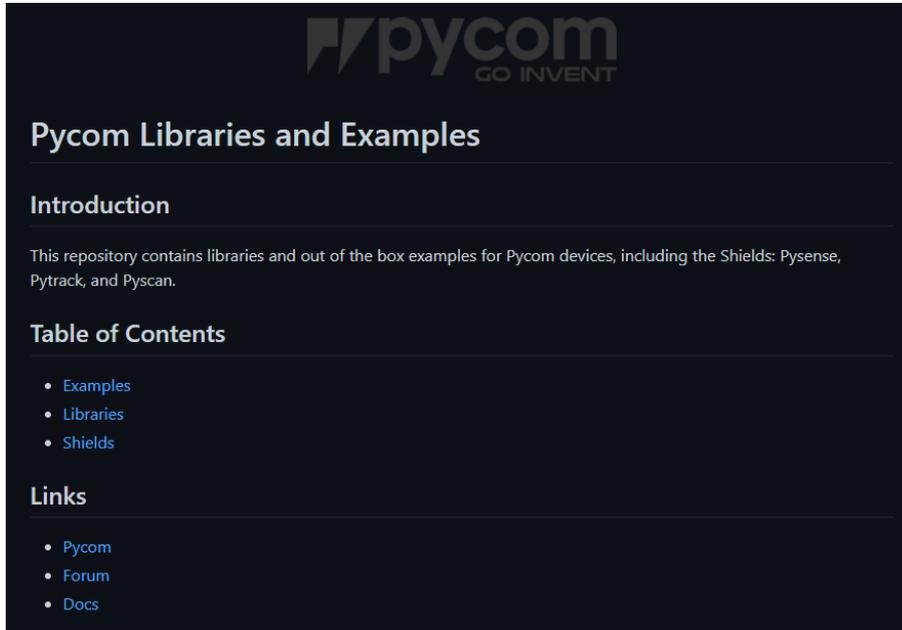
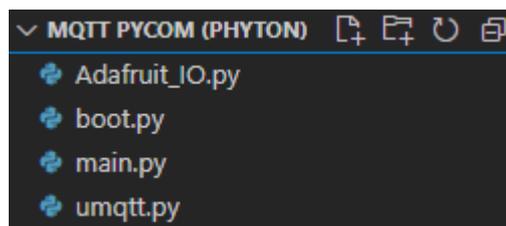


Figura 66: Librerías de Pycom en GitHub

Desde ahí tendremos acceso a la librería MQTT para poder configurar la comunicación desde nuestro dispositivo. Y podremos ejecutar los códigos desde la extensión Pymakr en Visual Studio Code o desde el IDE de Atom.



6.4.1 Boot.py

Desde el módulo de arranque se configura la UART con la tasa de baudios necesaria para el dispositivo y se llama a la función principal “main.py”.

```
from machine import UART
import machine
import os

uart = UART(0, baudrate=115200)
os.dupterm(uart)
```

```
machine.main('main.py')
```

6.4.2 Main.py

En este módulo se crea el acceso a la conexión a la red wifi para poder configurar el dispositivo, una vez lo consigue se configura el cliente mqtt.

```
from network import WLAN
from mqtt import MQTTClient
import machine
import time

def settimetype(duration):
    pass

wlan = WLAN(mode=WLAN.STA)
wlan.antenna(WLAN.EXT_ANT)
wlan.connect("yourwifinetwork", auth=(WLAN.WPA2, "wifipassword"), timeout
=5000)

while not wlan.isconnected():
    machine.idle()

print("Connected to Wifi\n")
client = MQTTClient("demo", "broker.hivemq.com", port=1883)
client.settimeout = settimetype
client.connect()

while True:
    print("Sending ON")
    client.publish("/lights", "ON")
    time.sleep(1)
    print("Sending OFF")
    client.publish("/lights", "OFF")
    time.sleep(1)
```

6.4.3 Adafruit IO

Desde este módulo se solicita la configuración mqtt del módulo "umqtt", así que si no la hemos incluido nos saltará un error como el siguiente

```
>>> Traceback (most recent call last):
  File "<stdin>", line 7, in <module>
ImportError: no module named 'umqtt'
>
MicroPython v1.8.6-849-d0dc708 on 2018-02-27; WiPy with ESP32
Type "help()" for more information.
```

También se introduce la configuración de la cuenta de Adafruit que estemos utilizando, para poder recibir los datos que se van procesando en la nube.

```
from network import WLAN      # For operation of WiFi network
import time                   # Allows use of time.sleep() for delays
import pycom                  # Base library for Pycom devices
from umqtt import MQTTClient # For use of MQTT protocol to talk to Adafruit IO
import ubinascii              # Needed to run any MicroPython code
import machine                 # Interfaces with hardware components
import micropython            # Needed to run any MicroPython code

# BEGIN SETTINGS
# These need to be change to suit your environment
RANDOMMS_INTERVAL = 5000 # milliseconds
last_random_sent_ticks = 0 # milliseconds

# Wireless network
WIFI_SSID = "IoT"
WIFI_PASS = "password"

# Adafruit IO (AIO) configuration
AIO_SERVER = "io.adafruit.com"
AIO_PORT = 1883
AIO_USER = "Pepellovera"
AIO_KEY = "aio_U0nD36b8Ru5Fq3dwaY38UMCfj1mq"

AIO_CLIENT_ID = ubinascii.hexlify(machine.unique_id()) # Can be anything
AIO_CONTROL_FEED = "Pepellovera/feeds/temp"

# END SETTINGS

# RGBLED
# Disable the on-board heartbeat (blue flash every 4 seconds)
# We'll use the LED to respond to messages from Adafruit IO
pycom.heartbeat(False)
time.sleep(0.1) # Workaround for a bug.
                # Above line is not actioned if another
                # process occurs immediately afterwards
pycom.rgbled(0xff0000) # Status red = not working

# WIFI
# We need to have a connection to WiFi for Internet access
# Code source: https://docs.pycom.io/chapter/tutorials/all/wlan.html

wlan = WLAN(mode=WLAN.STA)
wlan.connect(WIFI_SSID, auth=(WLAN.WPA2, WIFI_PASS), timeout=5000)

while not wlan.isconnected(): # Code waits here until WiFi connects
```

```

machine.idle()

print("Connected to Wifi")
pycom.rgbled(0xffd7000) # Status orange: partially working

# FUNCTIONS

# Function to respond to messages from Adafruit IO
def sub_cb(topic, msg):          # sub_cb means "callback subroutine"
    print((topic, msg))         # Outputs the message that was received.
    Debugging use.
    if msg == b"ON":            # If message says "ON" ...
        pycom.rgbled(0xffffffff) # ... then LED on
    elif msg == b"OFF":         # If message says "OFF" ...
        pycom.rgbled(0x000000)  # ... then LED off
    else:                        # If any other message is received ...
        print("Unknown message") # ... do nothing but output that it happened.

def random_integer(upper_bound):
    return machine.rng() % upper_bound

def send_random():
    global last_random_sent_ticks
    global RANDOMS_INTERVAL

    if ((time.ticks_ms() - last_random_sent_ticks) < RANDOMS_INTERVAL):
        return; # Too soon since last one sent.

    some_number = random_integer(100)
    print("Publishing: {0} to {1} ... ".format(some_number, AIO_RANDOMS_FEED), end='')
    try:
        client.publish(topic=AIO_RANDOMS_FEED, msg=str(some_number))
        print("DONE")
    except Exception as e:
        print("FAILED")
    finally:
        last_random_sent_ticks = time.ticks_ms()

# Use the MQTT protocol to connect to Adafruit IO
client = MQTTClient(AIO_CLIENT_ID, AIO_SERVER, AIO_PORT, AIO_USER, AIO_KEY)

# Subscribed messages will be delivered to this callback
client.set_callback(sub_cb)
client.connect()
client.subscribe(AIO_CONTROL_FEED)

```

```

print("Connected to %s, subscribed to %s topic" % (AIO_SERVER, AIO_CONTRO
L_FEED))

pycom.rgbled(0x00ff00) # Status green: online to Adafruit IO

try:
    # Code between try: and finally: may cause an e
    rror
    # so ensure the client disconnects the server i
    f
    # that happens.
    while 1:
        # Repeat this loop forever
        client.check_msg()# Action a message if one is received. Non-
        blocking.
        send_random() # Send a random number to Adafruit IO if it's t
        ime.
finally:
    # If an exception is thrown ...
    client.disconnect() # ... disconnect the client and clean up.
    client = None
    wlan.disconnect()
    wlan = None
    pycom.rgbled(0x000022)# Status blue: stopped
    print("Disconnected from Adafruit IO.")

```

6.4.4 Umqtt.py

Desde aquí se configura el protocolo de comunicación propio de MQTT

```

import usocket as socket
import ustruct as struct
from ubinascii import hexlify

class MQTTException(Exception):
    pass

class MQTTClient:

    def __init__(self, client_id, server, port=0, user=None, password=None,
    keepalive=0, ssl=False, ssl_params={}):
        if port == 0:
            port = 8883 if ssl else 1883
        self.client_id = client_id
        self.sock = None
        self.addr = socket.getaddrinfo(server, port)[0][-1]
        self.ssl = ssl
        self.ssl_params = ssl_params
        self.pid = 0
        self.cb = None
        self.user = user

```

```

self.pswd = password
self.keepalive = keepalive
self.lw_topic = None
self.lw_msg = None
self.lw_qos = 0
self.lw_retain = False

def _send_str(self, s):
    self.sock.write(struct.pack("!H", len(s)))
    self.sock.write(s)

def _recv_len(self):
    n = 0
    sh = 0
    while 1:
        b = self.sock.read(1)[0]
        n |= (b & 0x7f) << sh
        if not b & 0x80:
            return n
        sh += 7

def set_callback(self, f):
    self.cb = f

def set_last_will(self, topic, msg, retain=False, qos=0):
    assert 0 <= qos <= 2
    assert topic
    self.lw_topic = topic
    self.lw_msg = msg
    self.lw_qos = qos
    self.lw_retain = retain

def connect(self, clean_session=True):
    self.sock = socket.socket()
    self.sock.connect(self.addr)
    if self.ssl:
        import ssl
        self.sock = ssl.wrap_socket(self.sock, **self.ssl_params)
    msg = bytearray(b"\x10\0\0\x04MQTT\x04\x02\0\0")
    msg[1] = 10 + 2 + len(self.client_id)
    msg[9] = clean_session << 1
    if self.user is not None:
        msg[1] += 2 + len(self.user) + 2 + len(self.pswd)
        msg[9] |= 0xC0
    if self.keepalive:
        assert self.keepalive < 65536
        msg[10] |= self.keepalive >> 8
        msg[11] |= self.keepalive & 0x00FF
    if self.lw_topic:

```

```

        msg[1] += 2 + len(self.lw_topic) + 2 + len(self.lw_msg)
        msg[9] |= 0x4 | (self.lw_qos & 0x1) << 3 | (self.lw_qos & 0x2
) << 3
        msg[9] |= self.lw_retain << 5
self.sock.write(msg)
#print(hex(len(msg)), hexlify(msg, ":"))
self._send_str(self.client_id)
if self.lw_topic:
    self._send_str(self.lw_topic)
    self._send_str(self.lw_msg)
if self.user is not None:
    self._send_str(self.user)
    self._send_str(self.pswd)
resp = self.sock.read(4)
assert resp[0] == 0x20 and resp[1] == 0x02
if resp[3] != 0:
    raise MQTTException(resp[3])
return resp[2] & 1

def disconnect(self):
self.sock.write(b"\xe0\0")
self.sock.close()

def ping(self):
self.sock.write(b"\xc0\0")

def publish(self, topic, msg, retain=False, qos=0):
pkt = bytearray(b"\x30\0\0\0")
pkt[0] |= qos << 1 | retain
sz = 2 + len(topic) + len(msg)
if qos > 0:
    sz += 2
assert sz < 2097152
i = 1
while sz > 0x7f:
    pkt[i] = (sz & 0x7f) | 0x80
    sz >>= 7
    i += 1
pkt[i] = sz
#print(hex(len(pkt)), hexlify(pkt, ":"))
self.sock.write(pkt, i + 1)
self._send_str(topic)
if qos > 0:
    self.pid += 1
    pid = self.pid
    struct.pack_into("!H", pkt, 0, pid)
    self.sock.write(pkt, 2)
self.sock.write(msg)
if qos == 1:

```

```

        while 1:
            op = self.wait_msg()
            if op == 0x40:
                sz = self.sock.read(1)
                assert sz == b"\x02"
                rcv_pid = self.sock.read(2)
                rcv_pid = rcv_pid[0] << 8 | rcv_pid[1]
                if pid == rcv_pid:
                    return
            elif qos == 2:
                assert 0

def subscribe(self, topic, qos=0):
    assert self.cb is not None, "Subscribe callback is not set"
    pkt = bytearray(b"\x82\0\0\0")
    self.pid += 1
    struct.pack_into("!BH", pkt, 1, 2 + 2 + len(topic) + 1, self.pid)
    #print(hex(len(pkt)), hexlify(pkt, ":"))
    self.sock.write(pkt)
    self._send_str(topic)
    self.sock.write(qos.to_bytes(1, "little"))
    while 1:
        op = self.wait_msg()
        if op == 0x90:
            resp = self.sock.read(4)
            #print(resp)
            assert resp[1] == pkt[2] and resp[2] == pkt[3]
            if resp[3] == 0x80:
                raise MQTTException(resp[3])
            return

# Wait for a single incoming MQTT message and process it.
# Subscribed messages are delivered to a callback previously
# set by .set_callback() method. Other (internal) MQTT
# messages processed internally.
def wait_msg(self):
    res = self.sock.read(1)
    self.sock.setblocking(True)
    if res is None:
        return None
    if res == b"":
        raise OSError(-1)
    if res == b"\xd0": # PINGRESP
        sz = self.sock.read(1)[0]
        assert sz == 0
        return None
    op = res[0]
    if op & 0xf0 != 0x30:
        return op

```

```

sz = self._recv_len()
topic_len = self.sock.read(2)
topic_len = (topic_len[0] << 8) | topic_len[1]
topic = self.sock.read(topic_len)
sz -= topic_len + 2
if op & 6:
    pid = self.sock.read(2)
    pid = pid[0] << 8 | pid[1]
    sz -= 2
msg = self.sock.read(sz)
self.cb(topic, msg)
if op & 6 == 2:
    pkt = bytearray(b"\x40\x02\0\0")
    struct.pack_into("!H", pkt, 2, pid)
    self.sock.write(pkt)
elif op & 6 == 4:
    assert 0
# Checks whether a pending message from server is available.
# If not, returns immediately with None. Otherwise, does
# the same processing as wait_msg.
def check_msg(self):
    self.sock.setblocking(False)
    return self.wait_msg()

```

Una vez hayamos guardado los scripts, habría que subirlos al dispositivo y ya tendríamos la configuración MQTT para poder comunicarnos con el broker alojado en Adafruit IO.

Toda la comunicación también puede ser configurada desde Arduino. Adafruit nos ofrece una solución a través del IDE de Arduino con la biblioteca Adafruit_MQTT de una forma bastante sencilla.

Gracias a los feeds de Adafruit podemos guardar los datos en la nube provenientes de los sensores, sería algo parecido a lo siguiente:

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> DireccionViento	direccionviento	192	6 days ago
<input type="checkbox"/> Humedad	humedad	62	8 days ago
<input type="checkbox"/> Lluvia	lluvia	0	8 days ago
<input type="checkbox"/> Radiacion	radiacion-uv	1.6	6 days ago
<input type="checkbox"/> Temperatura	temp	20	6 days ago
<input type="checkbox"/> VelocidadViento	velocidadviento	15	8 days ago

Figura 67: Feeds en Adafruit IO

Donde se van almacenando los datos de la siguiente forma:

Created at	Value	Location
2021/06/12 7:37:28PM	20	
2021/06/12 7:37:27PM	23	
2021/06/12 7:37:26PM	26	
2021/06/12 7:37:26PM	27	
2021/06/12 7:37:25PM	30	
2021/06/12 7:37:24PM	32	
2021/06/12 7:37:24PM	33	
2021/06/12 7:37:23PM	28	

Figura 68: Almacenamiento de datos en Adafruit IO

6.4.5 Dashboard

A través de Node-Red crearemos un dashboard parecido al que Adafruit nos permite configurar desde su interfaz, de esta forma podremos mostrar los datos alojados en Adafruit IO, así como históricos con información de los datos de los sensores.

Así, se ha diseñado el siguiente esquema de nodos:

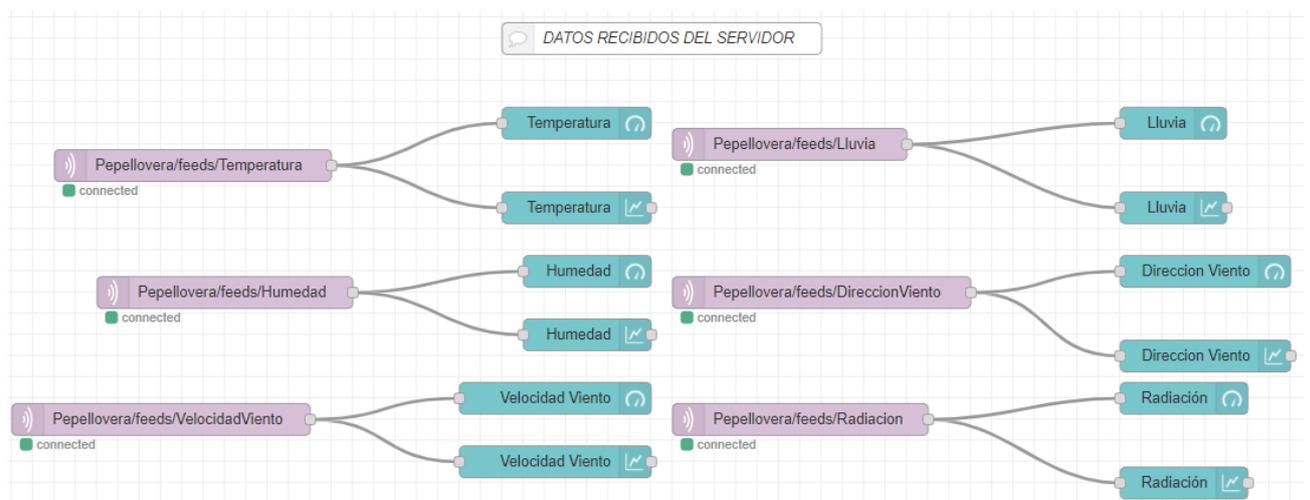


Figura 69: Esquema de nodos con Node-Red

Formado por los nodos:

- Mqtt in: es el nodo que se conecta al broker MQTT alojado en io.adafruit.com y se suscribe al topic deseado para recibir los mensajes que vayan llegando al mismo.
Por ejemplo, “Pepellovera/feeds/Temperatura” es el topic en el que leemos los datos de temperatura que va enviando nuestro sensor de temperatura al broker.
- Gauge: este nodo nos aporta un widget de tipo indicador, viene incluido en el paquete de widgets “dashboard”, y así muestra los datos recibidos en la interfaz usuario. Permite colocar varios tipos de indicadores para facilitar la visualización de la información.
- Chart: este nodo también es un widget de “dashboard”, nos permite colocar un gráfico que representa la evolución de la variable introducida. En este caso lo hemos usado para mostrar el histórico de datos de los sensores.

Así obtenemos el siguiente dashboard:

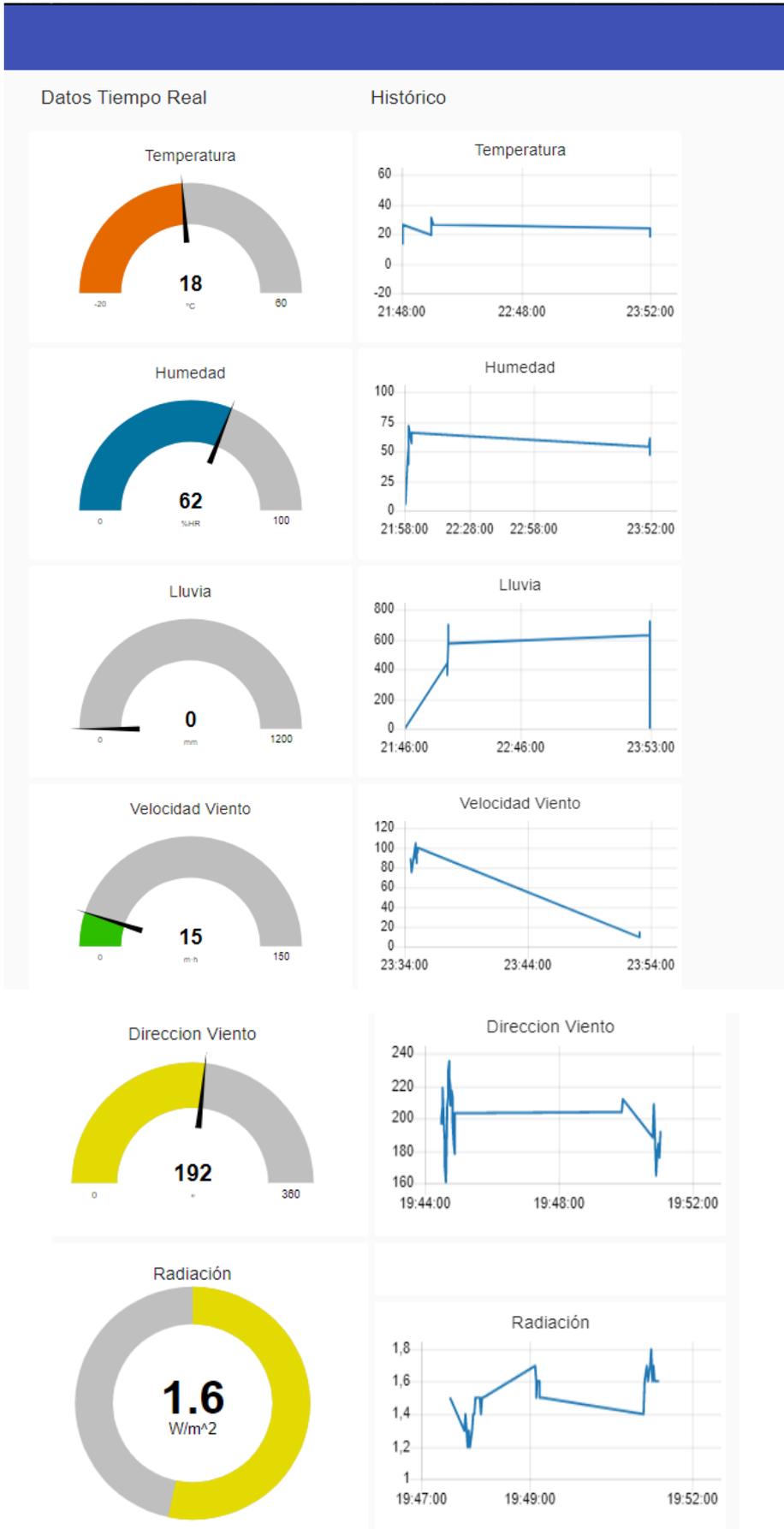


Figura 70: Dashboard con datos en tiempo real e históricos de 24h

7. INSTALACIÓN Y MANTENIMIENTO DEL NODO

7.1 Instalación del nodo

Para la selección de sitio de los componentes de una estación agroclimática la OMM nos aconseja la siguiente disposición de sensores:

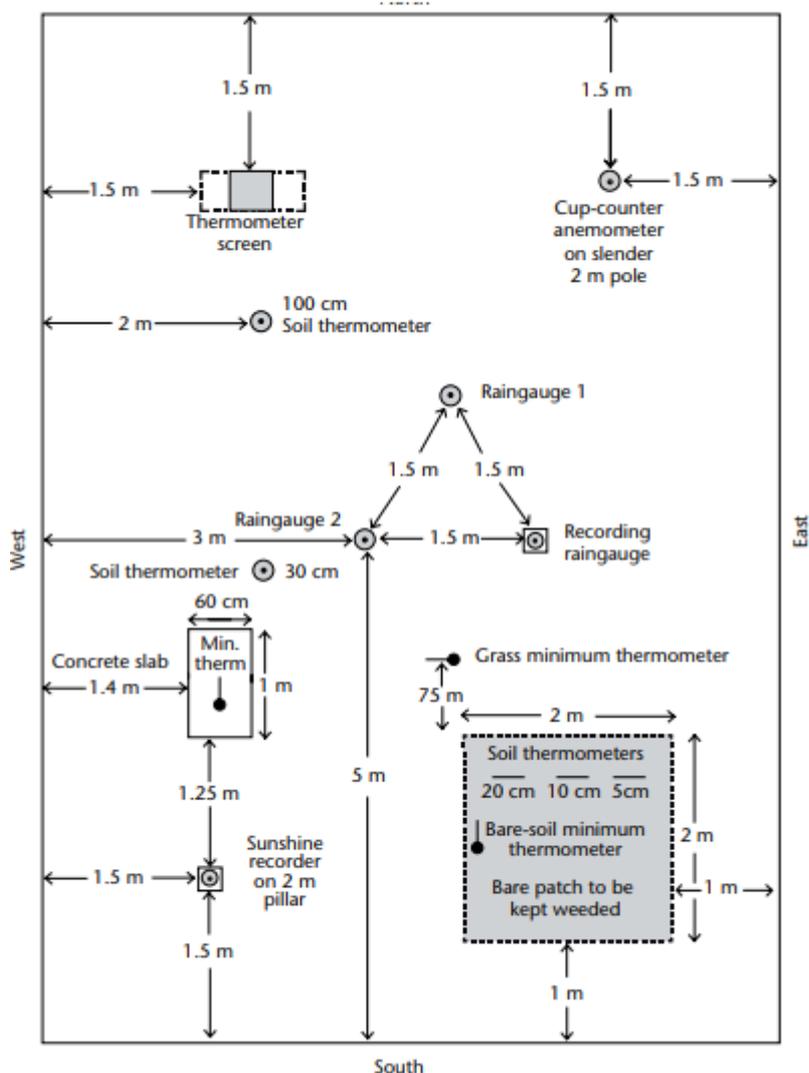


Figura 71: Disposición de los elementos que integran una estación

Ref. (pág. 2-5): https://library.wmo.int/doc_num.php?explnum_id=3996

Esta disposición permite eliminar posibles interferencias en la medición de los sensores por la proximidad entre ellos, ya sea por la sombra o por bloqueo de aire debido a la colocación de postes o también cercas en el perímetro, ya que si queremos limitar el

acceso a la estación para evitar manipulaciones no deseadas lo óptimo sería vallar el perímetro.

Además, el soporte de la estación debería quedar fijado a la superficie, de este modo salvamos a la estructura de cualquier accidente por agentes externos, como rachas fuertes de viento.

Según UNE 176101:2010, Punto 5, "LOCALIZACIÓN DE EMPLAZAMIENTOS":

- Las estaciones agrometeorológicas automáticas deben situarse en lugares con libre circulación de aire y visibilidad en todas direcciones, salvo en las que se quiera obtener la medida de determinadas variables en condiciones específicas diferentes, lo cual debe justificarse y documentarse en los ficheros informáticos utilizados para el intercambio y la difusión de los datos de cada estación (véase la norma UNE 500550).
- La estación debe estar situada a una distancia no inferior a cinco veces la altura de obstáculos, edificaciones, árboles y cortavientos naturales o artificiales, para evitar que su influencia pueda modificar las observaciones meteorológicas, salvo en las que se quiera obtener la medida de determinadas variables en condiciones específicas diferentes, lo cual debe justificarse y documentarse en los ficheros informáticos utilizados para el intercambio y la difusión de los datos de cada estación (véase la norma UNE 500550).
- Se recomienda que el recinto para la instalación de la estación meteorológica tenga unas dimensiones no inferiores a 10 m x 10 m. Si por razones de seguridad se cree necesario, dicho recinto estará vallado a una altura de 2 m, pudiendo utilizar un alambre de simple torsión o cualquier otro material que no impida la libre circulación de aire ni proyecte sombras sobre los sensores de radiación.

Respecto a la instalación de sensores en UNE 176101:2010, Punto 6.3 "Instalación de sensores":

- Sensores de velocidad y dirección de viento

Los sensores para la medición de la velocidad y la dirección del viento se recomiendan que estén instalados a una altura de 2 m.

Y respecto a la distancia de cualquier obstáculo se remite a lo comentado en el punto 5 (punto comentado con anterioridad).

- Radiación neta

Los sensores para la medición de radiación neta han de estar orientados al sur y en todo su campo de radiación solar no ha de haber ningún elemento, de la propia estación o ajeno, que produzca sombra ni reflejos artificiales sobre su medida. Se ha de procurar que la valla de protección no interfiera en esa medida.

Estos sensores deben colocarse a una altura de 1 m por encima del cultivo de referencia sobre el que se pretende medir.

Nuestro conjunto de sensores veleta-anemómetro-pluviómetro ha sido escogido en formato kit, ofrecido por Argent Data Systems, de esta forma la integración de los componentes es bastante sencilla, ya que viene con una guía bastante descriptiva para el montaje.

Guía en: https://learn.sparkfun.com/tutorials/weather-meter-hookup-guide?_ga=2.178757734.390438277.1624990639-1238838473.1616428567

Incluye un poste donde quedan fijados los mástiles sobre los que se montan los sensores, bridas y tornillería.

El conjunto quedaría de la siguiente forma:



Figura 72: Kit meteorológico

El nodo con nuestra placa de expansión tendría que aislarse del entorno, sobre todo por el agua y el polvo. Pycom nos ofrece carcasas con grado IP67 de protección contra polvo y agua como la siguiente:



Figura 73: Carcasa para el nodo

7.2 Mantenimiento del nodo

La situación de los instrumentos de medición de una EMA tiene que someterse a un mínimo de mantenimiento con la finalidad de que la calidad de los datos recogidos no se vea alterada. De este modo, la OMM nos recomienda un plan de mantenimiento preventivo para este tipo de estaciones.

Este tipo de planes requieren de un mantenimiento sistemático y se debe establecer un plan de visitas acorde a las necesidades de cada sensor instalado, así como del entorno en el que se encuentra instalado. Este plan incluye la rutina de mantenimiento, la inspección general del estado de la estación, la limpieza de los instrumentos según las recomendaciones del fabricante y registrar los metadatos.

Estructura principal

La primera inspección y la más sencilla es visualizar el estado del soporte de la estación, tenemos que asegurarnos de que no se haya alterado la sujeción que la mantiene unida al suelo, que los mástiles se encuentre en buen estado y que el eje principal tampoco haya sufrido ningún desperfecto. La estructura tendrá que encontrarse debidamente nivelada.

También tendremos que asegurarnos de que la tornillería que mantiene fijos los sensores no se ha soltado, en caso contrario se ajustará adecuadamente. Las carcasas del nodo y el sensor de temperatura tendrán que quedar bien revisadas de grietas y hermeticidad (en el caso de la carcasa que contiene el micro), y en caso contrario tendrán que ser reemplazadas con la mayor brevedad posible. Los cables también se tienen que someter a inspección, ya que si por accidente hubiese algún corte o perforación la veracidad de los datos tendría que darse por nula.

Alimentación

Como los sensores están alimentados mediante una conexión con el nodo principal, tendremos que asegurarnos que el cableado no ha sufrido ningún daño, y también comprobamos que nos sigue ofreciendo la alimentación necesaria con un multímetro.

El estado batería deberá ser visualizado y asegurar que no muestra señales de degradación. El cargador solar debe mantenerse limpio y seco, y comprobar que los conectores siguen en buen estado.

Por último, el panel solar, nos tenemos que asegurar que las celdas del panel se encuentran intacta, y también tendremos que limpiarlas con un paño suave, con cuidado de no rayar la superficie. Al estar conectado con el cargador solar por un cable con conexión de “jack” tendremos que verificar que no se ha visto alterado y que se encuentra bien aislado del exterior. Y para finalizar, claro está, asegurarnos que se

encuentra fijo, ninguna sujeción esta suelta y mantiene su inclinación sobre los 30 grados.

Sensores

Se podría decir que esta es la parte más importante de la rutina de mantenimiento, ya que los sensores son la parte del equipo encargada de recoger los datos, pero realmente la integración general de la estación es bastante importante, así que no hay que restarles importancia a los demás módulos de la estación, puesto que cada dispositivo es crítico para la correcta realización de la tarea final.

De este modo, el procedimiento es parecido que con el resto de componentes. De entrada, tendremos que comprobar el estado de cada uno de los sensores, atendiendo a su estado físico frente a posibles roturas de las carcasas externas, los brazos que los sostienen o la tornillería que los fija a estos mismos brazos, y ajustando las fijaciones que lo requieran.

Respecto a la limpieza y calibración de los sensores:

- El sensor que más cuidado requiere en este aspecto es el pluviómetro, sobre todo en otoño, que es la época en la que más hojas pueden introducirse dentro de la cubeta, tendremos que hacer al menos una revisión mensual. Destapar la cubierta y limpiar bien el mecanismo interno.
Para calibrarlo bastaría con derramar una cantidad de agua controlada, para asegurarnos de que el dispositivo la registra correctamente. Esta comprobación debería realizarse una vez al año.
- El sensor de temperatura y humedad es el que menos atención requiere, de todos modos, es necesaria una revisión. Tendremos que retirar el escudo protector y asegurarnos que las conexiones se encuentran en buen estado, también tendremos que limpiar la cubierta con un paño. En el caso de este sensor, se debería calibrar una vez al año, dada su poca complejidad.
- Los sensores de velocidad y dirección del viento no requieren mucha limpieza, aun así, deberemos retirar la suciedad acumulada. También tendremos que inspeccionar que los rodamientos siguen proporcionando el mínimo rozamiento para que no quede obstruida la rotación de ambos sensores.
- El sensor de radiación se encuentra a la altura del suelo, por ello tendremos que limpiarlo bien de cualquier deposición animal y de polvo, llevando cuidado con el cristal del sensor. La calibración del sensor de Lab Ferrer corre a cargo del servicio técnico, por lo que cada año solicitaremos su calibración.

Para llevar un registro del mantenimiento de la estación se suelen usar unos formularios donde se recoge la información del resultado del análisis del plan de mantenimiento, son las conocidas planillas o cartillas de mantenimiento:

PLANILLA DE VISITA DE RUTINA

A. INFORMACIÓN GENERAL	
1. Nombre de la Estación:	
2. ID de la estación:	
3. Fecha de visita:	
4. Datos personales del visitante:	
Nombre:	
Fono:	Celular:
Correo electrónico:	
Fotografías generales de la estación:	

D. SENSORES		
SENSOR	CONDICIÓN	COMENTARIO
	<input type="radio"/> Buen estado <input type="radio"/> Mal estado <input type="radio"/> Lectura incoherente	
	<input type="radio"/> Buen estado <input type="radio"/> Mal estado <input type="radio"/> Lectura Incoherente	
	<input type="radio"/> Buen estado <input type="radio"/> Mal estado <input type="radio"/> Lectura Incoherente	
	<input type="radio"/> Buen estado <input type="radio"/> Mal estado <input type="radio"/> Lectura Incoherente	
	<input type="radio"/> Buen estado <input type="radio"/> Mal estado <input type="radio"/> Lectura Incoherente	
		Fotografía del componente dañado

Figura 74: Ejemplos de cartillas de mantenimiento

Ref:

<https://biblioteca.inia.cl/bitstream/handle/123456789/7387/NR37804.pdf?sequence=1&isAllowed=y>

8. PRESUPUESTO

En este apartado se incluye el coste desglosado de todo el material involucrado en el diseño de la estación, los componentes usados en la pcb y los elementos que integran la estación, como el nodo, los sensores, el panel solar, la batería y el cargador de baterías.

No se han tenido en cuenta los costes referidos a mano de obra, software utilizado, ni otras herramientas requeridas para el desarrollo del proyecto.

Nombre	Cantidad	Precio
BT1	1	0,3 €
C1	1	2,58 €
C2	1	0,06 €
C3, C4	2	0,12 €
D1, D2, D3	3	1,11 €
D4	1	0,308 €
J1	1	0,24 €
J2	1	0,88 €
J3	1	0,88 €
J4	1	0,24 €
J5	1	0,68 €
J6, J9	2	0,34 €
J7	1	0,88 €
J8	1	0,17 €
R1, R7	2	0,2 €
R2, R3	2	0,2 €
R4, R5	2	0,2 €
R6	1	0,08 €
U1	1	0,51 €
U2	1	1 €
PCB	1	5 €
Panel Solar	1	38 €
Batería	1	8,42 €
Cargador Solar	1	14,81 €
Equipo estación	1	67,67 €
Piranómetro	1	74,64 €
FiPy	1	59,4 €
Expansion Board 3.0	1	17,6 €
Carcasa	1	8 €
TOTAL		304,52 €

9. CONCLUSIONES

Las estaciones meteorológicas automáticas han ido cobrando bastante importancia para el muestreo de variables climatológicas en tiempo real, ya sea para un uso doméstico, para aplicaciones agrícolas a mayor escala o para el estudio estadístico mediante análisis de históricos.

Las llamadas EMAs domésticas cada vez son más frecuentes, ya que dispositivos como las tarjetas de desarrollo de Arduino, Pycom o LilyGo son analizadas por la comunidad y mejoradas continuamente para ofrecer más prestaciones de una forma mas sencilla para todos los usuarios.

Frente al caso de estaciones para uso agrícola es bastante importante el monitoreo ambiental, ya que supone una mejora en la productividad de los cultivos. Pero para ello es importante dotar a la estación de un sistema de energía autónomo, de este modo se ha hecho un análisis de consumo que cumpla las necesidades de la estación para no comprometer la calidad de los datos recogidos.

También se han analizado distintas alternativas para los componentes de adquisición de variables meteorológicas, ya que la premisa inicial consistía en una estación económica, que fuera incluso accesible para el nivel usuario, y que el conjunto de sensores tuviera una integración sencilla con el nodo principal.

En base a lo anterior, en este proyecto se ha llevado a cabo el diseño de una pcb que complemente las funcionalidades de la placa de desarrollo FiPy, teniendo en cuenta los tipos de datos recogidos por los sensores, en su mayoría señales analógicas que requieren de un muestreo para convertirlas en las magnitudes deseadas. Para ello se han incluido conectores en la placa directamente ruteados a los pines correspondientes.

Para la transmisión de los datos se ha empleado el protocolo MQTT basado en la publicación/suscripción mediante topics, con un broker que se encarga de gestionar el flujo de mensajes. Y para la visualización de los datos en tiempo real se han utilizado Dashboard mediante la herramienta Node-Red, la cual nos ofrece un entorno visual bastante atractivo.

La dificultad del proyecto se ha debido en parte al dispositivo usado para el nodo de la estación, ya que no tiene tanto soporte como puede tenerlo cualquier dispositivo de Arduino, por lo que la mayoría de la información disponible se encuentra en la documentación del fabricante.

De cara al futuro se han hecho pruebas de monitoreo con el sensor de humedad y temperatura, pero no se ha alcanzado la implementación de todo el equipo en campo. Además de ello, se pueden implementar instrumentos de medición de mayor calidad para aumentar la frecuencia de muestreo y obtener datos más precisos, pero dejando a un lado la premisa del bajo coste.

10. BIBLIOGRAFÍA

1. OMM; 1991 (segunda ed.), Compendio de apuntes para la formación del personal meteorológico de la Clase IV - Volumen 2, Meteorología: partes I y II https://library.wmo.int/index.php?lvl=notice_display&id=6802#.YLFV86gzbIV
2. WMO; 2012 (2010 edition), Guide to Agricultural Meteorological Practices (GAMP). https://library.wmo.int/index.php?lvl=notice_display&id=12113#.YNSyFWgzblU
3. Maldonado I, Isaac, Ruiz M, Rubén y Fuentes B, Marcel (2010) Manual de mantenimiento de estaciones meteorológicas automáticas [en línea]. Chillan: Boletín INIA - Instituto de Investigaciones Agropecuarias. no. 199. Disponible en: <https://biblioteca.inia.cl/handle/123456789/7387>
4. <https://docs.pycom.io/datasheets/development/fipy/>
5. WMO; 2003 Guidelines on climate observation network and systems. Terry Allsopp ; Paul Llansó. https://library.wmo.int/index.php?lvl=notice_display&id=11634#.YPF02OgzaUI
6. Campbell Scientific, 52202-L Electrically Heated Rain and Snow Gage; <https://www.campbellsci.es/pn52202>
7. Campbell Scientific, HC2S3 Air temperature and relative humidity sensor; <https://www.campbellsci.es/hc2s3>
8. Campbell Scientific, 05103 Anemoveleta; <https://www.campbellsci.es/05103>
9. Weather Meter Hookup Guide, Sparkfun; https://learn.sparkfun.com/tutorials/weather-meter-hookup-guide?_ga=2.178757734.390438277.1624990639-1238838473.1616428567
10. Photovoltaic Geographical Information System (PVGIS); <https://ec.europa.eu/jrc/en/pvgis>
11. United Nations Terminology Data Base; <https://unterm.un.org/unterm/portal/welcome>
12. Journal of Agrometeorology, ASSOC Agrometeorologists; <https://jcr.clarivate.com/jcr-jp/journal-profile?journal=J%20AGROMETEOROL&year=2020&fromPage=%2Fjcr%2Fsearch-results>
13. Advances in Meteorology, Hindawi LTD; <https://jcr.clarivate.com/jcr-jp/journal-profile?journal=ADV%20METEOROL&year=2020&fromPage=%2Fjcr%2Fsearch-results>