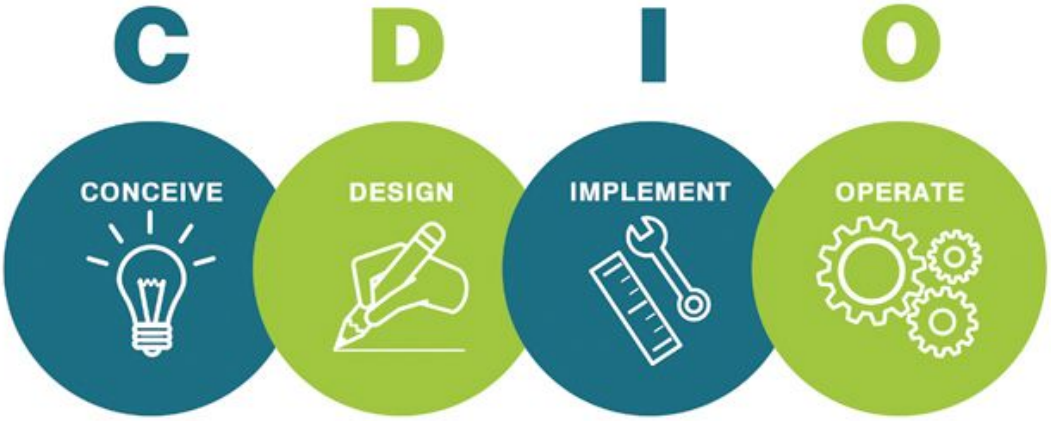
SONDA AGRÍCOLA
SPRINT 2



Abidán Brito Clavijo
Pablo Enguix Llopis
Luis Belloch Martínez
Elvira Montagud Hernandis

CDIO

ÍNDICE DE CONTENIDOS

CDIO

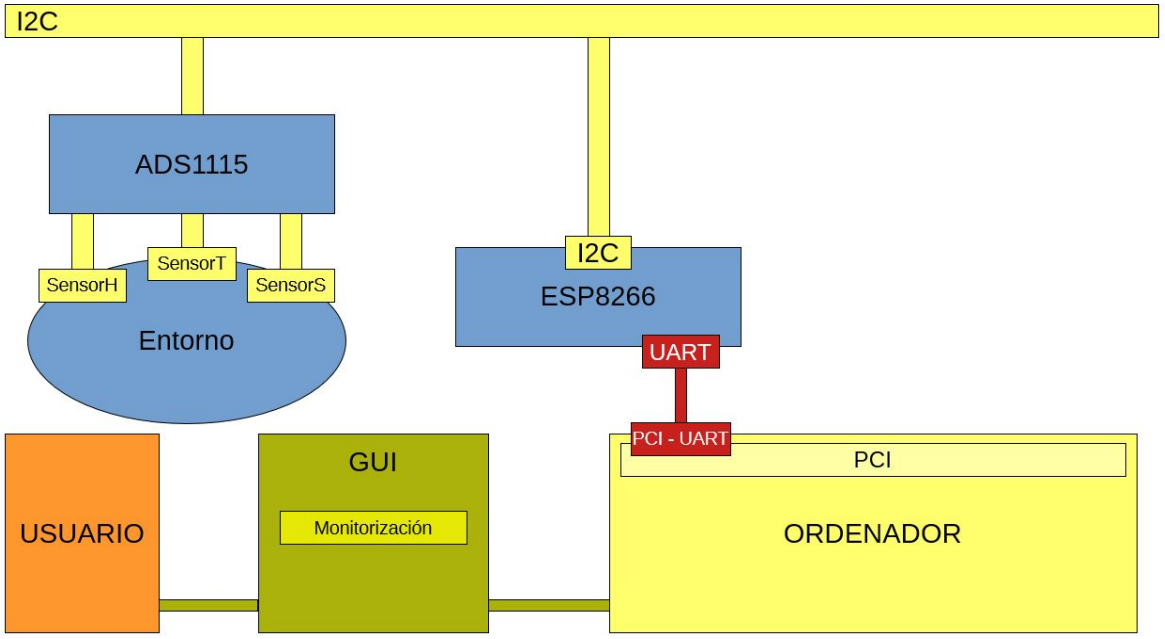
REPOSITORIO GIT Y LISTADO DOCUMENTAL

Para la base de código y documentos pertinentes de este proyecto, se ha utilizado un repositorio Git como sistema de control de versiones, alojado en la plataforma GitHub. Para acceder al repositorio, haga click sobre el hipervínculo aquí expuesto:
https://github.com/abidanBrito/CDIO_Agriculture_Sensors

Todos los documentos de esta versión se encuentran disponibles tanto en la aplicación de gestión de proyectos ágiles Worki, como en el repositorio de GitHub arriba citado. El repositorio está compuesto por los siguientes documentos y directorios:

1. **Sprint2_Documentación.pdf.** (Este documento.)
2. **Actas_Daily_Scrum.pdf.** (Actas de las reuniones.)

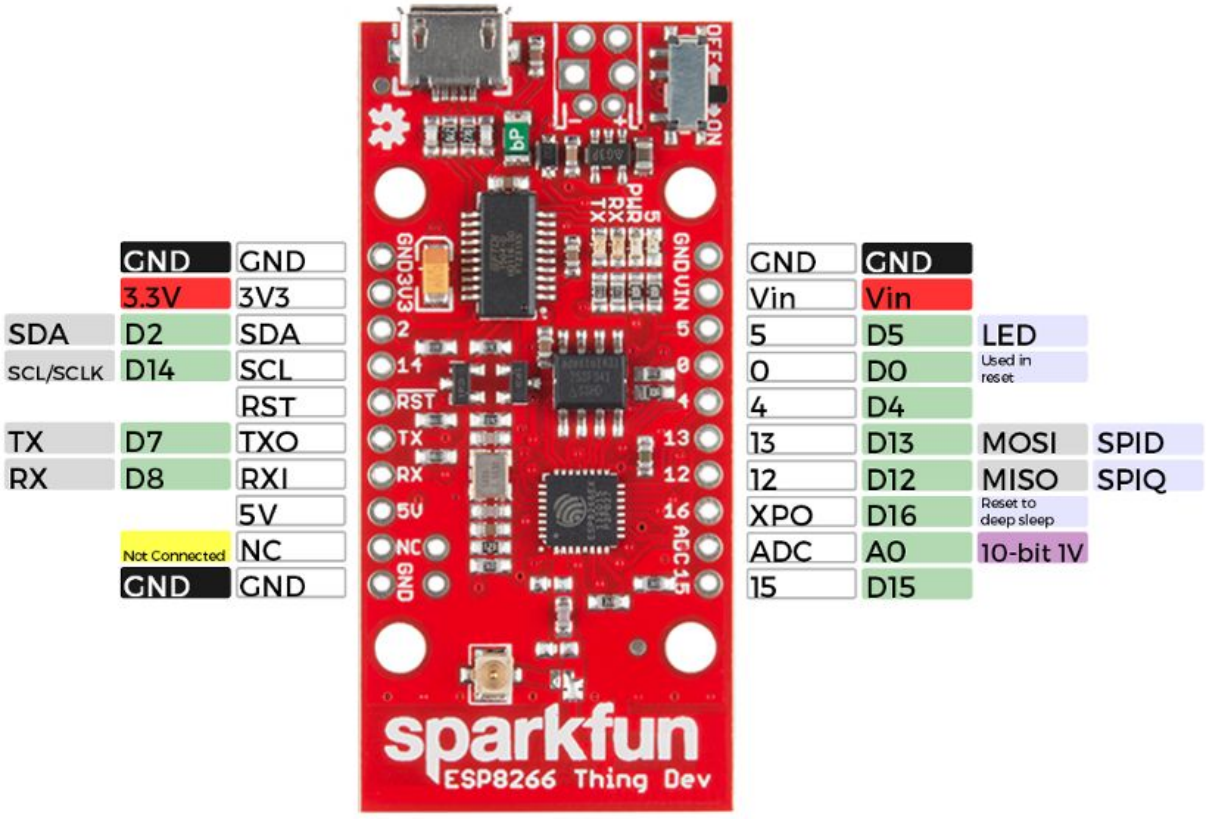
DISEÑO DEL SISTEMA



CDIO

COMPONENTES, BUSES Y PINES RELEVANTES

El sistema aquí expuesto parte del microcontrolador **ESP8266 Thing Dev** como base, acoplado al **convertidor Analógico-Digital** (A-D / ADC) **ADS1115**. Es necesario el uso de un ADC externo para poder leer / registrar varias señales analógicas a la vez. Por ende, los sensores se conectan al ADC. El ADS1115 dispone de 4 ADC de **16 bits**, 15 para la medición y 1 para el signo. A modo de inciso, cabe mencionar que la resolución del ADC se ajusta al rango de entrada.



Lo primero que se debe hacer es establecer una línea de tierra. Los pines de **GND** han de coincidir, y se ha de tener especial precaución con el sentido en que se acoplan los pines. Si se hiciese en una orientación errónea, se podría cortocircuitar la placa.

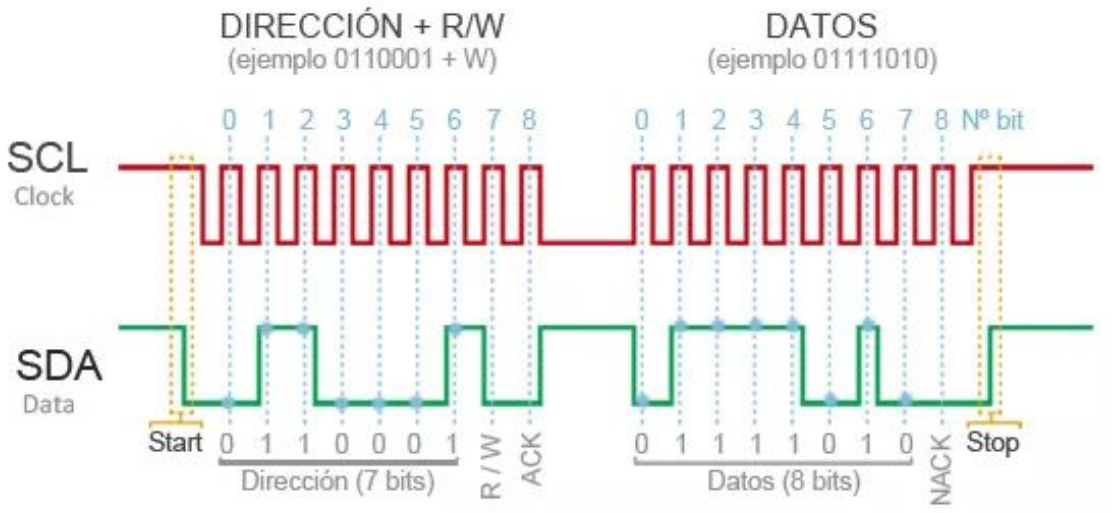
El circuito del **Transmisor-Receptor Asíncrono Universal (UART)** controla los puertos y dispositivos serie. Se conecta al ordenador, mediante un protocolo de comunicación en serie, permitiendo el análisis en tiempo real con el monitor en serie del Arduino IDE. La velocidad de transferencia ha de ser la misma que la configurada en el Arduino, para que la transmisión de datos coincida. Nótese que la placa ESP8266 solo tiene un UART. Es decir, no pueden haber varios periféricos conectados al mismo tiempo por la misma UART. La transmisión se lleva a cabo por el pin **TRx**. No obstante, al conectar la placa al PC por el puerto **microUSB**, se conecta directamente al UART.

CDIO

Nuestro sistema utiliza el **protocolo I2C.** Se trata de un protocolo de comunicación en serie, bidireccional (half-duplex) y síncrono. El **bus I2C** se conecta a todos los sensores. Es el mismo para todos. La conexión al microcontrolador se lleva a cabo a través de dos líneas o buses:

- **SDA** (data): primero se manda el bit de inicio (start) y luego la secuencia; el tren de datos en sí. Finalmente, un bit de acuerdo y otro de parada.
- **SCL / CLK** (clock): bus para el reloj. Marca el ritmo o compás, por así decirlo.

Ésto quiere decir que el maestro y el esclavo envían datos por el mismo bus, el cual es controlado por el maestro, que crea la señal de reloj. Nótese que I2C no utiliza selección de esclavo, sino direccionamiento.



Las líneas SCL del microcontrolador y SCL del convertidor Analógico-Digital se conectan entre sí, sincronizándolos. Análogamente, los SDA y GND se conectan por pares. En nuestro caso la facultad nos ha proporcionado los componentes, y el ADC consta de un circuito integrado que redirecciona las conexiones alineando los pines, facilitando el acoplamiento.

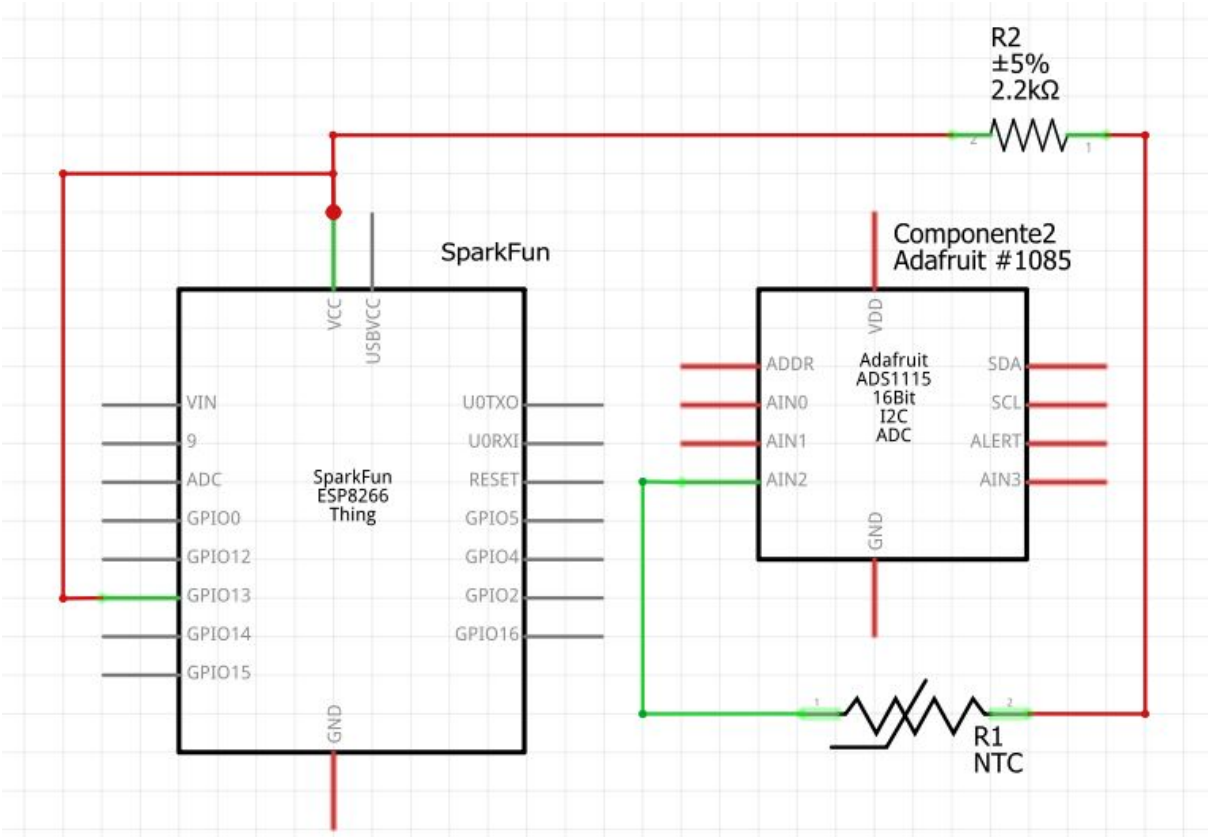
CDIO

ESQUEMÁTICOS ELECTRÓNICOS

Para la elaboración de los esquemáticos electrónicos se ha utilizado el programa informático *Fritzing*; una herramienta especializada para la creación de esquemáticos, circuitos electrónicos y diseño de PCBs.

Seguidamente se presenta el esquema del sensor de temperatura NTC, por ser el añadido con respecto al sprint anterior, y el esquema del montaje de la función *Deep Sleep*.

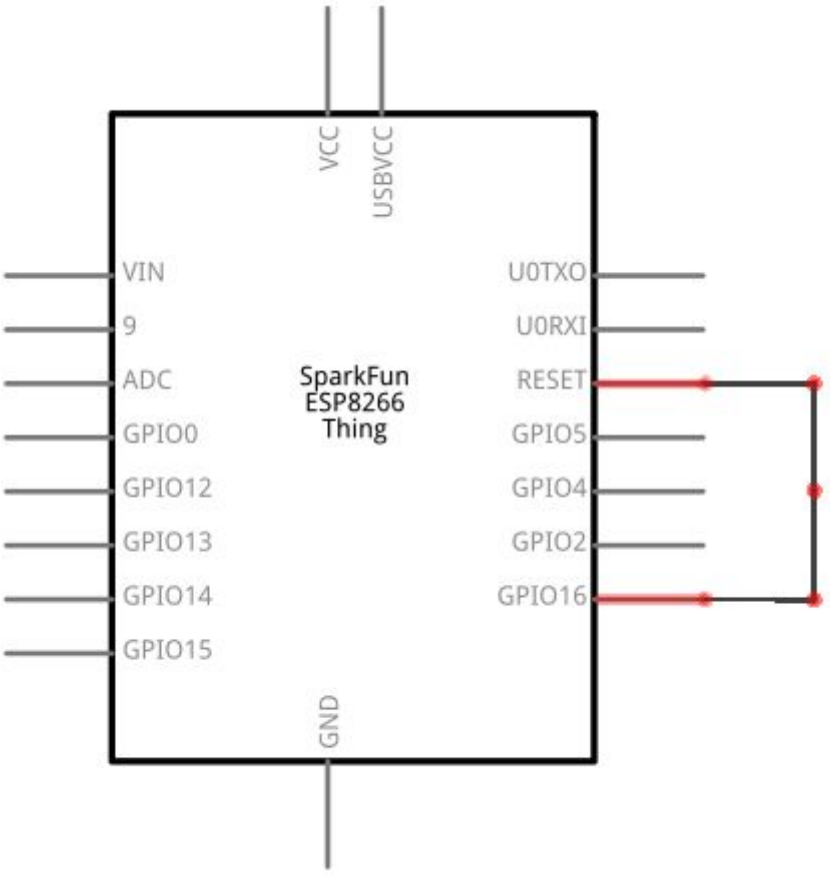
SENSOR DE TEMPERATURA



Véase el archivo original adjunto en el directorio de GIT Hub "*Sprint2/schematics/SensorTemperatura.fzz*" en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki.

CDIO

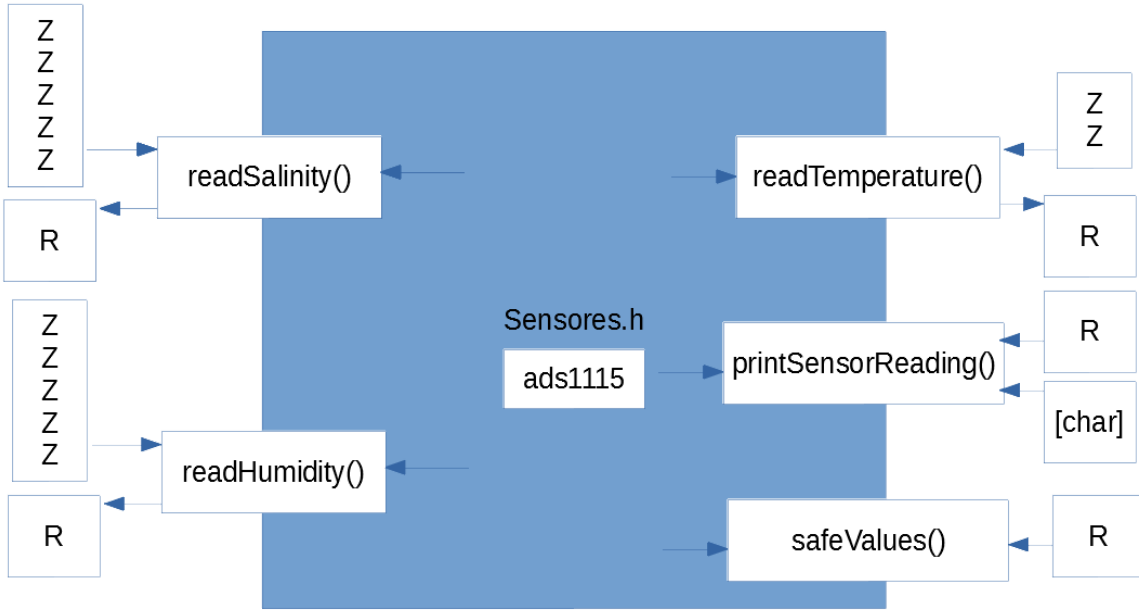
DEEP SLEEP



El archivo original se encuentra en el repositorio GIT “/Sprint2/schematics/DeepSleep.fzz”.

DISEÑO DEL PROGRAMA

Se adjunta a continuación el esquema de la clase Sensores.h, la cual agrupa las funciones de todos los sensores con el fin de simplificar el código.



El archivo original se encuentra en "*/Sprint2/schematics/Clase_Sensores.odg*".

En este diseño se presentan las funciones de los tres sensores implementados hasta este sprint: salinidad, humedad y temperatura. Además se incluye una función para escribir los resultados recibidos en el monitor Serial de Arduino IDE y una función llamada "SafeValues" la cual se asegura de corregir medidas que superen los límites del porcentaje de 0 a 100.

Esta clase es accedida desde *mainProgram.ino* el cual contiene dos funciones:

- **void Setup()**: se ejecuta una vez y define parámetros iniciales como la ganancia del ADS.
- **void Loop()**: hace las llamadas a las funciones de Sensores.h y guarda algunos datos. Se ejecuta cada minuto con la función Deep Sleep (configurable).

Para poder utilizar el ADC externo, es necesario:

- Añadir la **librería** correspondiente.
- Definir el **objeto** ADC.
- Inicializar el ADC (con la función **miADC.begin ()**).
- Definir la **ganancia** con la que se va a trabajar (al inicio del programa).

CDIO

El ADC sólo funciona en el **rango (0, 1) V**. No obstante, podemos utilizar la función **map()** de Arduino para transformar / escalar los valores de un rango de entrada a uno de salida. Es importante asignar un valor de **ADDR**, para determinar la dirección (valor

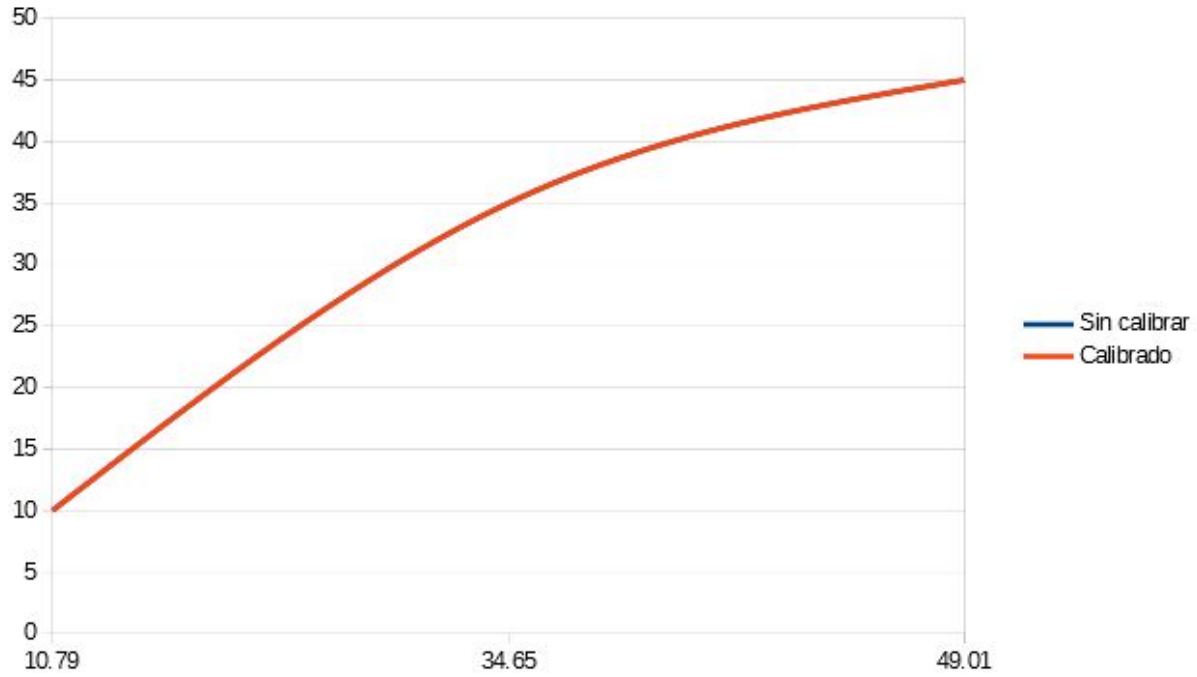
hexadecimal) en función del pin. Ésto es necesario para discernir qué convertidor analógico utilizar.

Sensores.h y *mainProgram.ino* constituyen el programa completo que se puede encontrar en el repositorio GIT: "*CDIO-SensorTech/Sprint2/code/mainProgram/*"

CALIBRACIONES Y TESTEOS

SENSOR DE TEMPERATURA

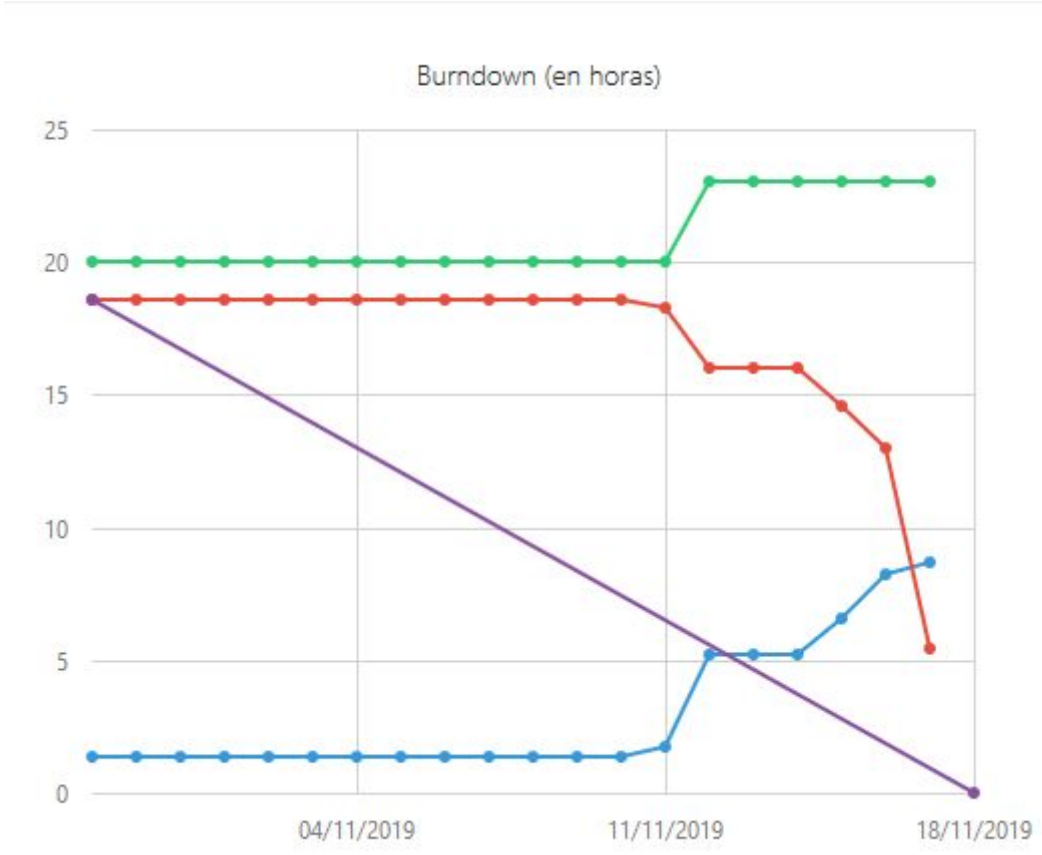
Para calibrar el sensor de temperatura hay que tener en cuenta que el valor de las temperaturas solo es óptimo en temperaturas que oscilan entre los 0ºC y los 45ºC, por ello en la calibración de este sensor se han usado como base un vaso con agua a temperatura ambiente y otro vaso de agua con un poco de hielo, para así realizar con la mayor precisión el calibrado. En la fase de calibrado es imprescindible el uso de un termómetro para así asignar a los valores de la NTC a los valores medidos con este. Los datos recibidos se ajustan con la variación media con respecto a la recta de regresión.



Podemos ver que la desviación es de -0.24ºC de media, por ello el cambio es casi imperceptible. El sensor es suficientemente preciso para este proyecto.

CDIO

DIAGRAMA DE BURNDOWN



Siendo la línea roja el trabajo restante, vemos que en este sprint hemos sobreestimado la cantidad de trabajo.

Véase el archivo original en el directorio "*Sprint2/img/Burndown.png*" en el repositorio o en la sección de documentos de la aplicación de gestión de proyectos ágiles Worki. Para un seguimiento más detallado, acceder a la sección de *Seguimiento*, subsección *Dashboard* en dicha aplicación.

DAILY SCRUM

Véase el documento adjunto en el directorio "*Sprint2/Actas_Daily_Scrum.pdf*".