



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



SPRINT 2

Proyecto CDIO

Mario Climent
Laura Gavilán
Hugo Martín
Alberto Alonso
Alexander Fuset



Contenido

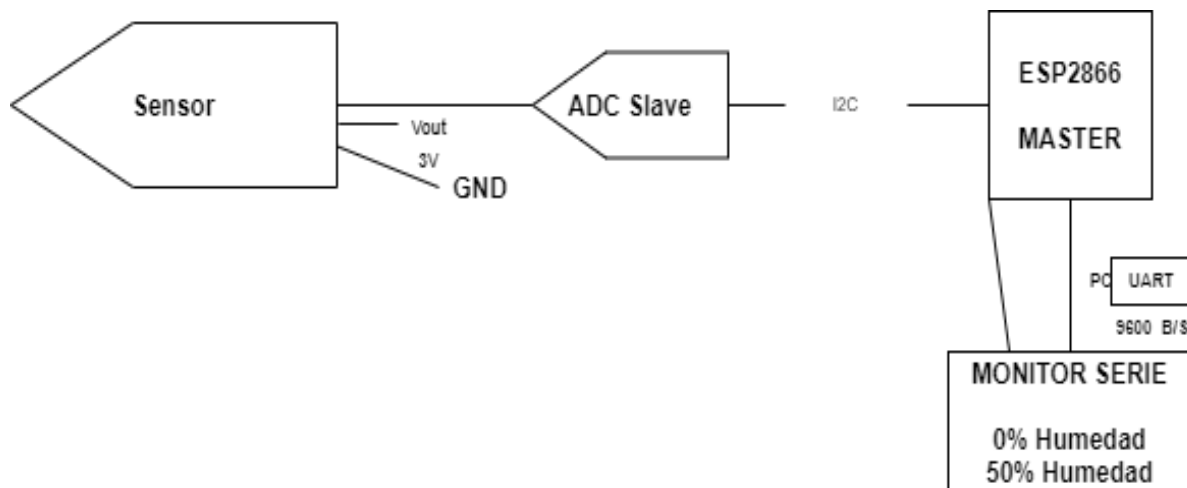
1-. Sensor de humedad.	2
Diseño del Circuito:	2
Calibración del sensor de humedad.	2
Testeo del sensor.	2
2-. Sensor de salinidad.....	3
Diseño del circuito:.....	4
Testeo del sensor.	5
Esquemático del circuito:	5
3.-Sensor de Temperatura	6
Diseño del circuito:	6
Calibración del sensor de temperatura:	6
Testeo del sensor de temperatura:	6
Esquemático del sensor:	7
4.- Deep Sleep.	8
Diseño del circuito:	8
Esquemático del Deep Sleep:.....	8
5.- circuito Completo.	9
6.- Actas	10
7.- Diagrama de Burndown.	12
8.-Programa.	13

Sensor Completo:

1-. Sensor de humedad.

Como agricultor nuestro cliente quiere conocer el nivel de humedad en su plantación para saber cuándo se deben regar y la cantidad de agua correspondiente, como también conocer el nivel de humedad en días de lluvia para preservar la salud de sus plantas.

Diseño del Circuito:



El Circuito está formado por un sensor de humedad conectado a un microcontrolador master “ESP8266 The Thing Dev-Board” conectado a otro microcontrolador slave “Adafruit ADS1115”.

El puerto de salida “A” del sensor de humedad está conectado al puerto “A0” del microcontrolador ADS1115 y los puertos “positivo” y “negativo” están conectados a la fuente de alimentación “5V” y la toma de tierra “GND” respectivamente.

Calibración del sensor de humedad.

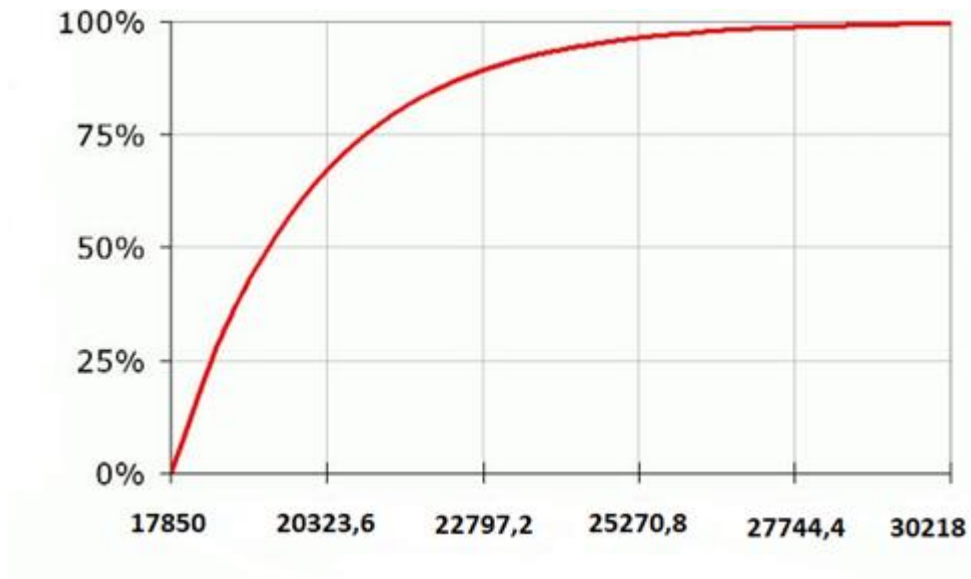
La calibración del sensor de humedad se ha llevado a cabo gracias a un programa de Arduino que con el sensor de humedad mide un valor en seco al que se asigna un 0% y otro en mojado que sería la referencia para el 100%.

Testeo del sensor.

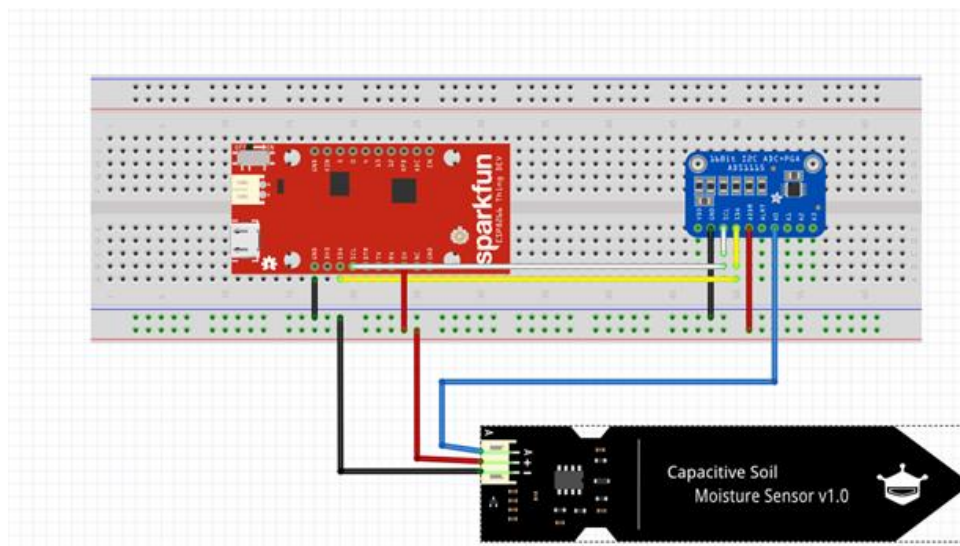
Tomando en cuenta los valores obtenidos con la calibración se calcula el porcentaje correspondiente que mide el sensor de humedad.

Este valor de humedad se obtiene multiplicando la división del valor en seco entre la diferencia entre ambos valores por cien a lo que se le resta el valor de entrada multiplicado por cien y dividido entre la diferencia de los valores en seco y mojado.

El valor medido por el sensor se muestra en el monitor serie gracias a nuestro circuito.



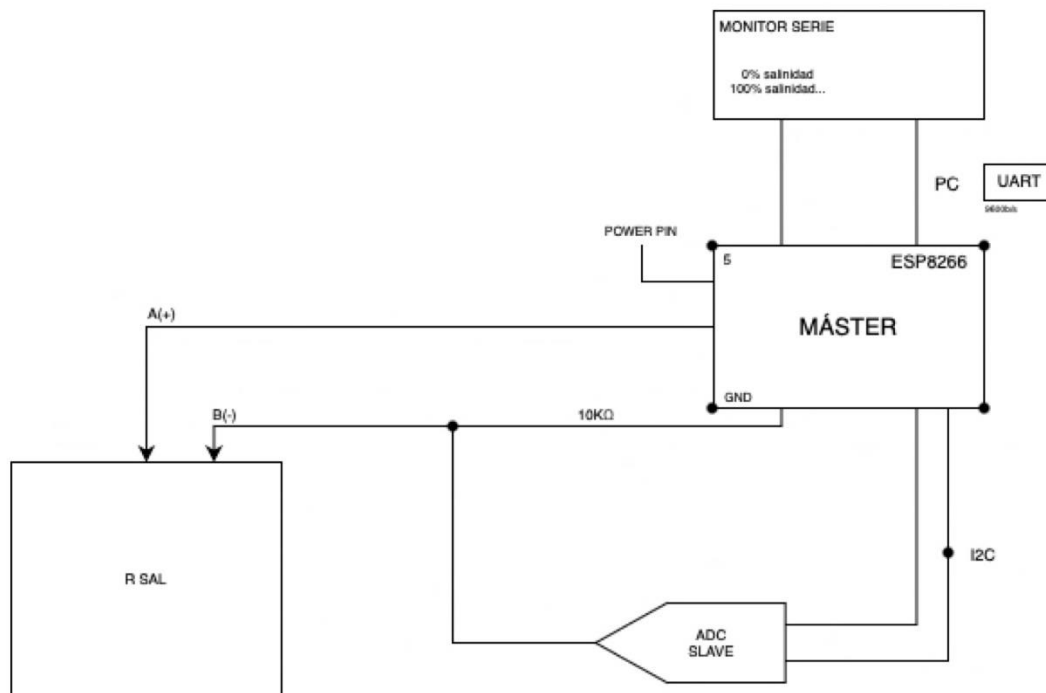
Esquemático del circuito:



2-. Sensor de salinidad.

Como agricultor nuestro cliente quiere conocer el nivel de salinidad en su plantación para que las plantas se mantengan en buen estado.

Diseño del circuito:



El Circuito está formado por un microcontrolador “master” “ESP8266 The Thing Dev-Board” conectado a otro microcontrolador “slave” “Adafruit ADS1115”.

A nuestro “master” se conecta a través del puerto “GND” una resistencia de 10k Ohm, que es conectada a la entrada “A0” del “slave”, así mismo, de este cable sale uno de los polos del sensor de salinidad, con el otro saliendo del microcontrolador ESP8266, ambos de estos polos se sumergen en agua con sal, la cual actúa como resistencia.

Calibración del sensor de salinidad.

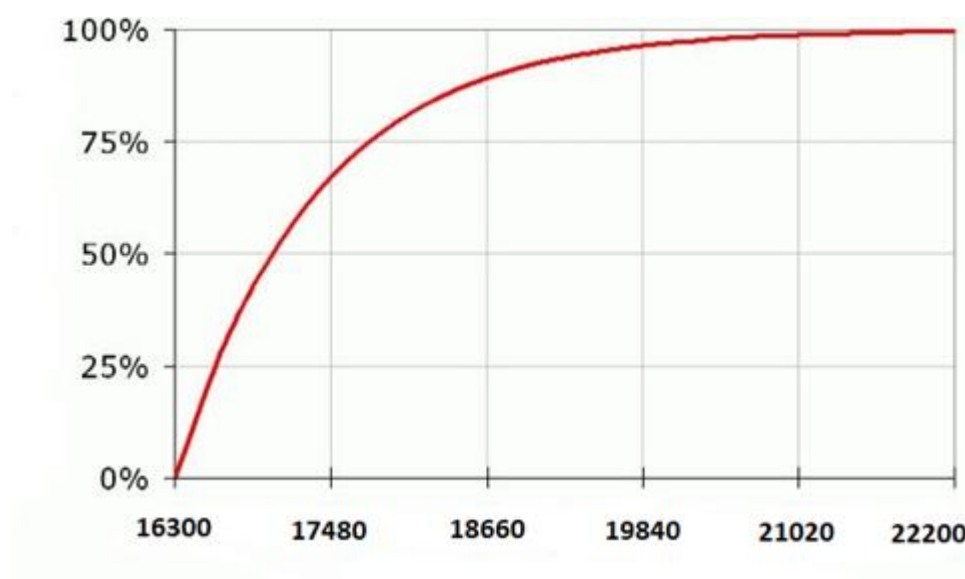
La calibración del sensor de salinidad se ha llevado a cabo gracias a un programa de Arduino que con el sensor de salinidad sumergido en agua sin sal se mide un valor al que se asigna un 0% y otro valor en agua con sal con el que el circuito final mediría 100%.

Testeo del sensor.

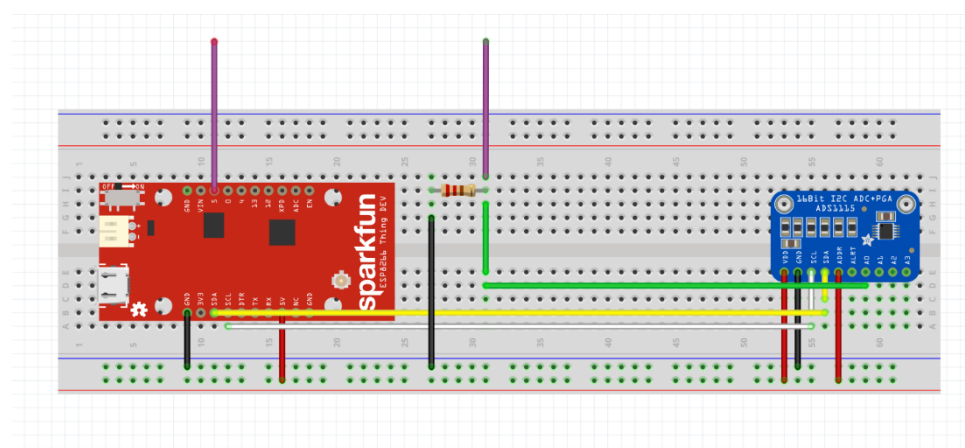
Con los valores obtenidos con la calibración se calcula el porcentaje correspondiente que mide el sensor de salinidad.

Este valor se obtiene multiplicando la división del valor sin sal entre la diferencia entre ambos valores por cien a lo que se le resta el valor de entrada multiplicado por cien y dividido entre la diferencia de los valores sin sal y con sal.

El valor medido por el sensor se muestra en el monitor serie para conocer el porcentaje.



Esquemático del circuito:



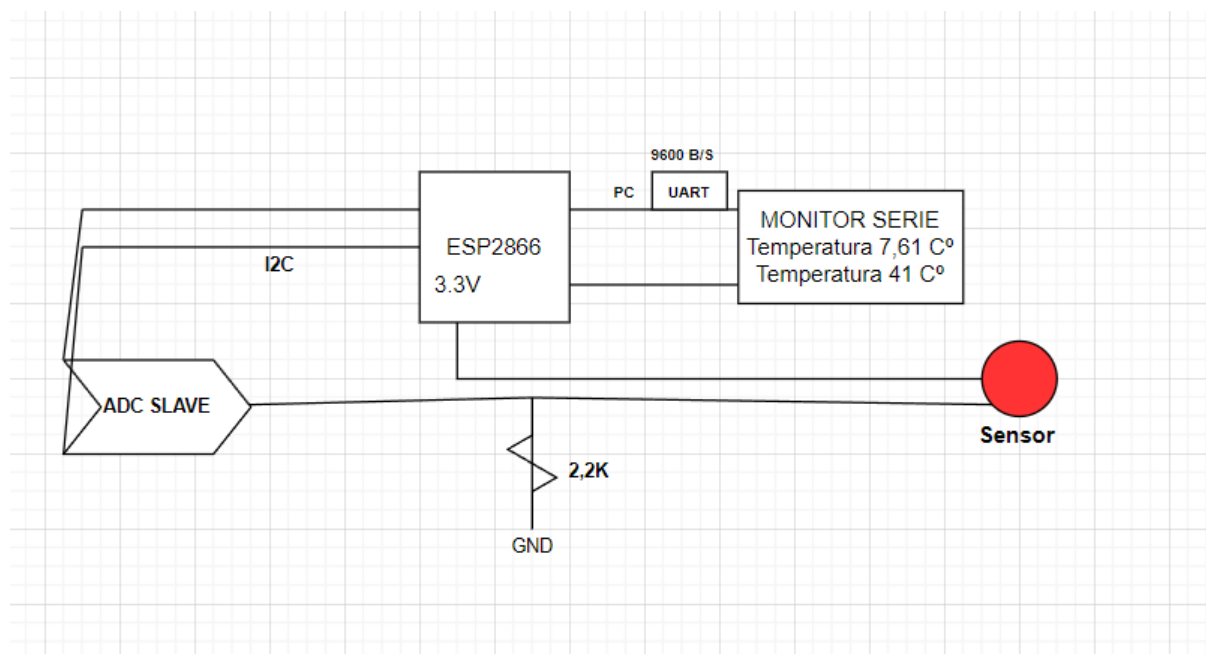
3.-Sensor de Temperatura

Como agricultor nuestro cliente quiere una medida correcta de la temperatura para un mejor control del crecimiento y salud de sus plantas.

Diseño del circuito:

El Circuito está formado por un microcontrolador “master” “ESP8266 The Thing Dev-Board” conectado a otro microcontrolador “slave” “Adafruit ADS1115”, a los cuales se conectan una resistencia de 2,2k Ohm y un sensor de temperatura.

El sensor de temperatura se conecta a nuestro “master” a través del puerto 3,3V por uno de sus cables y por otro es conectado a la resistencia de 2,2k Ohm, que está conectada a GND, y a una de las entradas de nuestro “slave”, que mediante el protocolo I2C es conectado al “master” que con el UART muestra los resultados por pantalla.



Calibración del sensor de temperatura:

La calibración del sensor de salinidad se llevó a cabo gracias a una serie de pruebas con un termistor NTC, que fue sumergido en agua a diferentes temperaturas (t^a ambiente, agua caliente y agua fría), en todas estas pruebas se anotó el voltaje medido por un voltímetro conectado a un circuito compuesto por una fuente de alimentación de 3,3 V, una resistencia de 2,2k Ohm y la NTC.

Con estos voltajes se sacan los umbrales de frío y calor, ya que la resistencia del NTC es más alta con el frío y más baja si entra en calor, por tanto, a voltajes bajos se les asignará una temperatura baja y a los voltajes altos una temperatura alta.

Testeo del sensor de temperatura:

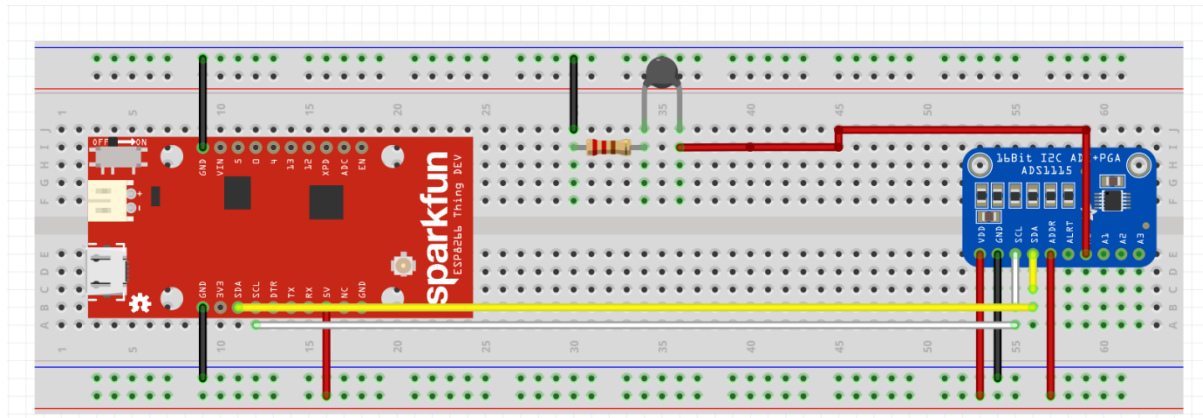
Tomando en cuenta los umbrales obtenidos se calcula la temperatura correspondiente, esto se consigue gracias a las ecuaciones:

$$Voltios = \frac{4,096 * adc2}{2^{15} - 1}$$

$$Temperatura(^{\circ}C) = \frac{Voltios - b}{m}$$

Donde: $b = 0.70$, $m = 0.035$ y $adc2$ es el valor leído por el monitor serie conectado al circuito.

Esquemático del sensor:



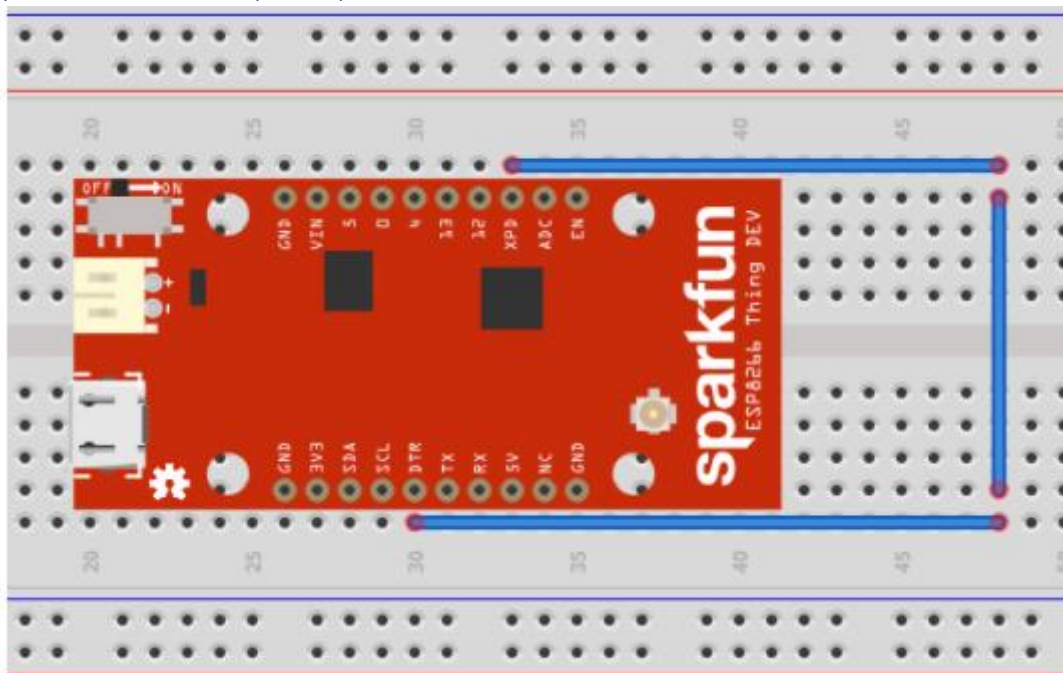
4.- Deep Sleep.

Para una mayor autonomía y utilidad nuestro sensor aplica un protocolo mediante el que después de tomar una serie de medidas entra en un periodo “Deep sleep” durante un tiempo determinado, y alternará estos dos estados de medición y suspensión.

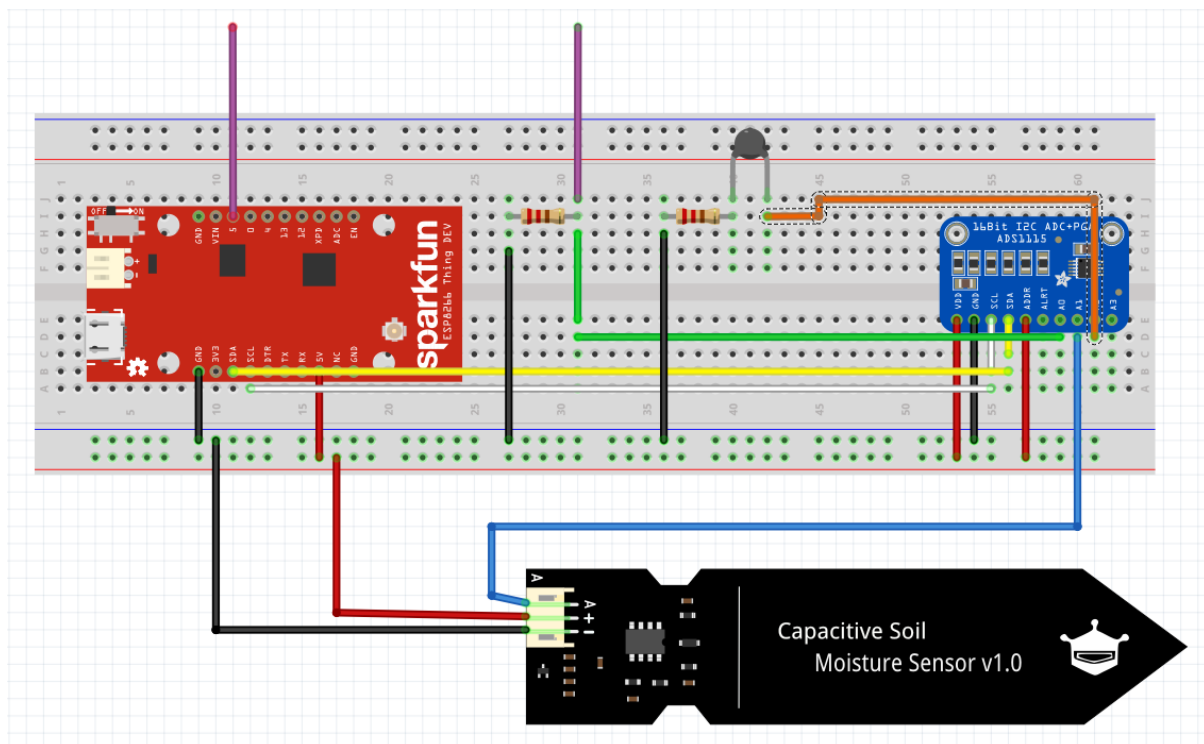
Diseño del circuito:

El Circuito está formado por un microcontrolador “master” “ESP8266 The Thing Dev-Board” conectado a otro microcontrolador “slave” “Adafruit ADS1115”. En este circuito compuesto por ambos microcontroladores se conecta un cable desde el pin reset(DTR) hasta el pin 16(XPD) de nuestro “master”.

Esquemático del Deep Sleep:



5.- circuito Completo.





6.- Actas

Acta 1

25.10.2021 → 15:00 - 18:30

Participantes:

- Laura Gavilán
- Alberto Alonso
- Mario Climent
- Hugo Martín
- Alex Fuset

Tareas individuales:

- Recogida de datos
- Montaje del circuito
- Montaje del circuito
- Montaje del circuito
- Montaje del circuito

Acta 2

26.10.2021 → 12:00 - 14:30 + 9:30 - 10:00

Participantes:

- Laura Gavilán
- Alberto Alonso
- Mario Climent
- Hugo Martín
- Alex Fuset

Tareas individuales:

- acta + toma de datos
- montaje del circuito
- montaje del circuito
-
- programación

Tareas grupales: añadir tareas al trello, organización del Trello y buscar la ecuación.

Acta 1 02.11.2021

Participantes:

- Laura Gavilán
- Alberto Alonso
- Mario Climent
- Hugo Martín
- Alex Fuset

Tareas individuales:

- actas
- Programación
- Programación
- Programación
- Programación

Acta 2 05.11.2021

Participantes:

- Laura Gavilán
- Alex Fuset

Tareas individuales:

- subir el programa, organizar Trello
- subir el programa, documentación.

Acta 3 06.11.2021

Participantes:

- Laura Gavilán
- Alberto Alonso
- Alex Fuset
- Mario Climent
- Hugo Martín

Tareas individuales:

- actas + prezi + diseños
- diseños analógicos
- documentación
- Programación
- Programación } + montaje

Acta 4 07.11.2021

Participantes:

- Alex Fuset
- Hugo Martín

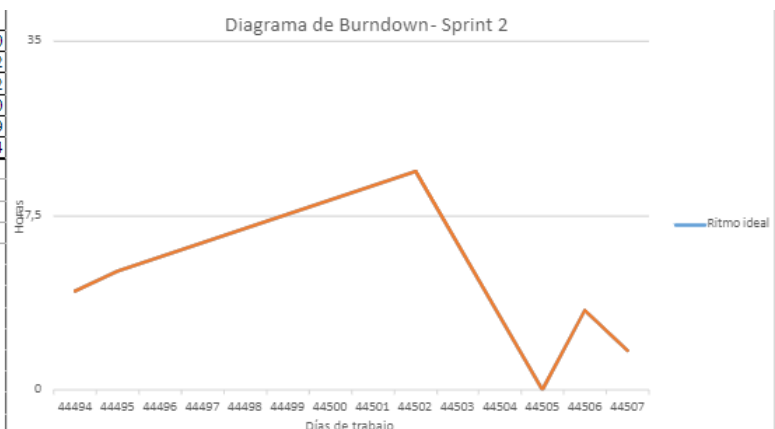
Tareas individuales:

- } diagrama de burndown
- acabar documentación y prezi

7.- Diagrama de Burndown.

Sprint 1 Días	Miembros del equipo (Puntos de Historia)						Tareas que hacen los Miembros del equipo				
	Hugo	Laura	Alberto	Alex	Mario	Total	Hugo	Laura	Alberto	Alex	Mario
25/10	1	1	1	1	1	5	01_T1 Estudiar el sensor	01_T1 Estudiar el sensor	01_T1 Estudiar el sensor	01_T1 Estudiar el sensor	01_T1 Estudiar el sensor
26/10	0,4	0,4	0,4	0,4	0,4	2	01_T2 Estudiar del Convertidor ADS1115	01_T2 Estudiar del Convertidor ADS1115	01_T2 Estudiar del Convertidor ADS1115	01_T2 Estudiar del Convertidor ADS1115	01_T2 Estudiar del Convertidor ADS1115
2/11	5,6	1,6	1,6	5,6	5,6	22	01_T3 Testear el programa	01_T3 Testear el programa	01_T3 Testear el programa	01_T3 Testear el programa	01_T3 Testear el programa
5/11						0	02_T1 Codificar el programa	02_T1 Codificar el programa	02_T1 Codificar el programa	02_T1 Codificar el programa	02_T1 Codificar el programa
6/11	2	1,33	1,33	1,33	2	7,99	02_T2 Testear el programa	02_T2 Testear el programa	02_T2 Testear el programa	02_T2 Testear el programa	02_T2 Testear el programa
7/11						4	03_T1 Revisar los circuitos de cada miembro para elegir el que mejor funciona	01_T6 Documentar	01_T6 Documentar	01_T6 Documentar	03_T2 Codificar el programa completo
miembro	10	4,33	4,33	18,33	19	55,99					
	Hugo	Laura	Alberto	Alex	Mario	Total					

Días	Ritmo ideal	Ritmo del equipo
1	25/10	10
2	26/10	12
3	2/11	22
4	5/11	0
5	6/11	8
6	7/11	4



Historia de Usuario	Tareas requeridas	Estimación de tiempo (Puntos de Historia)	Estimación tiempo total
01_Medida de Temperatura	01_T1 Estudiar el sensor	2	26
	01_T2 Estudiar del Convertidor ADS1115	3	
	01_T3 Diseñar el circuito	5	
	01_T4 Codificar el programa	10	
	01_T5 Testear el programa	2	
	01_T6 Documentar	4	
02_Deep Sleep	02_T1 Codificar el programa	2	8
	02_T2 Testear el programa	2	
	02_T3 Documentar	4	
03_Integrar las funcionalidades	03_T1 Revisar los circuitos de cada miembro para elegir el que mejor funciona	4	22
	03_T2 Codificar el programa completo	12	
	03_T3 Testear el programa completo	6	
			56



8.-Programa.

```
#include <Wire.h>    // Librería para usar el BUS I2C
```

```
#include <Adafruit_ADS1X15.h> // Librería del ADS1115
```

```
//---> Incluir el constructor del ads1115
```

```
Adafruit_ADS1115 ads1115;
```

```
//---> Incluir las variables de calibración del sensor de salinidad
```

```
const int valorSin = 16300;
```

```
const int valorSal = 22200;
```

```
int power_pin = 5;
```

```
//---> Incluir las variables de calibración del sensor de humedad
```

```
const int AirValue = 30218;
```

```
const int WaterValue = 17850;
```

```
// ---> Incluimos las variables del calculo de la temperatura.
```

```
float b = 0.79;
```



```
float m = 0.035;
```

```
//Contador para deepsleep
```

```
const int sleepTimeS = 30;
```

```
int cont=0;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    Serial.println("Inicializando...");
```

```
    ads1115.begin(); //Inicializa el ads1115
```

```
    // Configurar power_pin como pin de salida (Para sensor de salinidad)
```

```
    pinMode(power_pin, OUTPUT);
```

```
    Serial.println("Ajustando la ganancia...");
```

```
    ads1115.setGain(GAIN_ONE);
```



```
// Configurar power_pin como pin de salida (Para sensor de salinidad)
```

```
pinMode(power_pin, OUTPUT);
```

```
}
```

```
int funciónSalinidad (){
```

```
//---> Variables del programa
```

```
int16_t salinidad;
```

```
int16_t adc1;
```

```
//---> Poner power_pin a nivel alto
```

```
digitalWrite(power_pin, HIGH);
```

```
//---> Esperar 100ms
```




```
delay(100);
```

```
//---> Muestrear la tensión del sensor de salinidad
```

```
salinidad = analogRead( power_pin );
```

```
//---> Poner power_pin a nivel bajo
```

```
digitalWrite(power_pin,LOW);
```

```
adc1 = ads1115.readADC_SingleEnded(1);
```

```
salinidad = 100*valorSin/(valorSin-valorSal)-adc1*100/(valorSin-valorSal);
```

```
if (salinidad>=100){
```

```
    salinidad=100;
```

```
}
```

```
if (salinidad<=0){
```

```
    salinidad=0;
```

```
}
```



```
return salinidad;
```

```
}
```

```
int16_t funciónHumedad (){
```

```
int16_t adc0;
```

```
int16_t humedad;
```

```
adc0 = ads1115.readADC_SingleEnded(0);
```

```
humedad = 100*AirValue/(AirValue-WaterValue)-adc0*100/(AirValue-WaterValue);
```

```
if (humedad>=100){
```

```
    humedad=100;
```

```
}
```

```
if (humedad<=0){
```

```
    humedad=0;
```



}

return humedad;

}

```
float funciontemperatura(){  
    int16_t adc2;  
    float voltios, temperatura;  
    adc2=ads1115.readADC_SingleEnded(2);  
    voltios = 4.096*adc2/(pow(2,15)-1);  
    temperatura = (voltios - b)/m;  
    //Serial.println("ADC2: ");  
    //Serial.println(adc2);  
    //Serial.println("Grados(°C): ");  
    //Serial.println(temperatura);  
    return temperatura;  
}
```

```
void sleep(){  
  
    if (cont<20){  
        cont++;  
        delay(1000);  
    }  
    else{  
        Serial.println("ESP8266 in sleep mode");  
    }  
    ESP.deepSleep(sleepTimes * 1000000);  
}
```



```
void loop() {  
  
    int salinidad, humedad;  
    float temperatura;  
  
    int16_t adc0;  
  
    adc0 = ads1115.readADC_SingleEnded(0);  
  
    int adc1;  
  
    adc1 = ads1115.readADC_SingleEnded(1);  
  
    int adc2;  
  
    adc2 = ads1115.readADC_SingleEnded(2);  
  
    //int sleep;  
  
    humedad=funciónHumedad ();  
  
    salinidad=funciónSalinidad ();  
  
    temperatura=funciontemperatura();  
  
    ;  
  
    Serial.println("ESP8266 in normal mode");
```



```
Serial.print("-----");
```

```
Serial.print('\n');
```

```
Serial.print("AIN1: ");
```

```
Serial.println(adc1);
```

```
Serial.print("Salinidad (%): ");
```

```
Serial.print(salinidad);
```

```
Serial.println("%");
```

```
Serial.print("AIN0: ");
```

```
Serial.println(adc0);
```

```
Serial.print("Humedad (%): ");
```

```
Serial.print(humedad);
```

```
Serial.println("%");
```

```
Serial.println("ADC2: ");
```

```
Serial.println(adc2);
```



```
Serial.println("Grados(°C): ");
```

```
Serial.println(temperatura);
```

```
Serial.print("-----");
```

```
sleep();
```

```
delay(5000);
```

```
}
```