

Universidad Galileo

Gerson Lopez

Hardware/Microcontroladores/sistemas embebidos

Sistema de riego automatizado

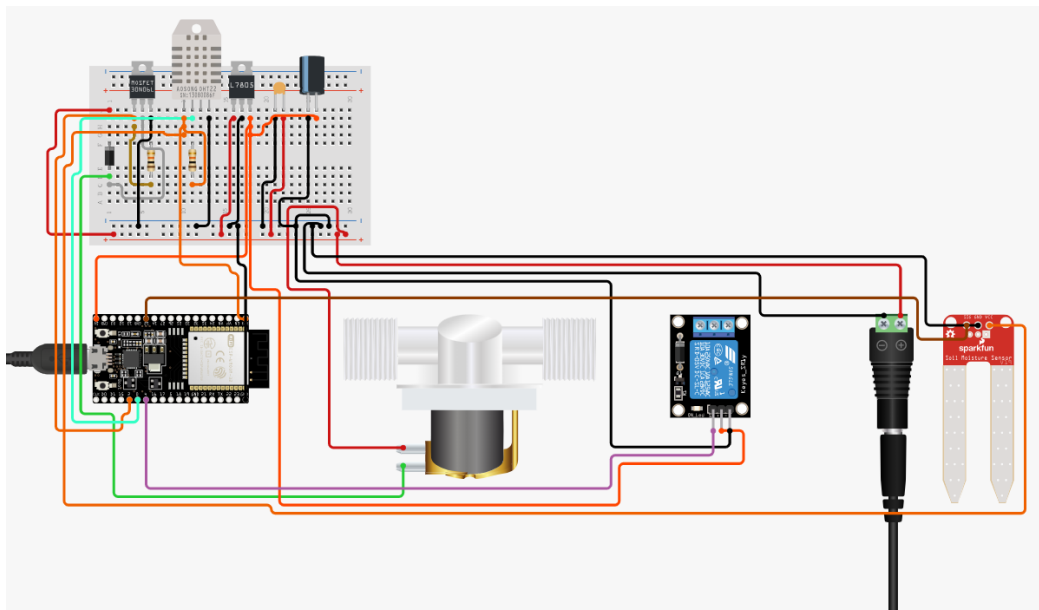
Guatemala 30 de jun. de 24

Sistema de riego automatizado

El riego automático con IoT permite gestionar de una forma inteligente y eficiente del agua mediante la integración de sensores, actuadores y red de comunicación en el ecosistema de riego, estos sistema recopilan datos en tiempo real sobre variables como la humedad del suelo, el estado del actuador(riego on/riego off) con esta información el sistema de reigo puede tomar decisiones automáticas para aplicar la cantidad exacta de agua necesaria para evitar el exceso o la escases del rieg, la idea principal por hacer este sistema es en base a las ventajas que tiene, el riego automático con IoT tienen los siguientes ventajas:

- Uso sostenible del agua
- Monitorización remota
- Limitar el recurso humano para el control
- Posibilidad de poder transpolar a otras áreas ya sea de la casa o agrícola

En este proyecto se usará un ESP32 para tomar el control del sistema de riego, este en base al código escrito para realizar tareas específicas en relación con las mediciones que este tomando con los sensores y se encargara de humedecer la tierra de las plantas, el diagrama es el siguiente:



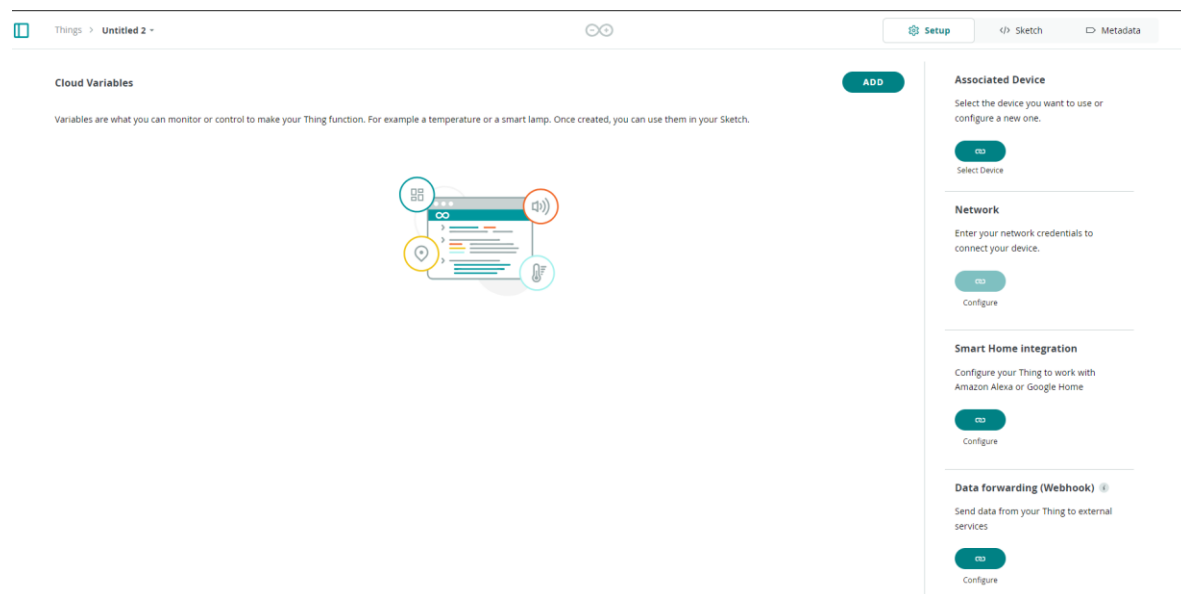
Circuito y materiales usados en el proyecto

Este circuito es un prototipado que simplifica la implementación de un riego automático ya que en una aplicación real es necesario utilizar cable extra para poder controlar la electroválvula y el sensor de humedad, además de hacer la base para colocar el microcontrolador en un área segura para que funcione.

Los materiales utilizados en este prototipo son los siguientes

- MCU ESP32
- Sensor DHT11
- Sensor de humedad con electrodos
- Bomba de agua
- Relay
- Jumpers

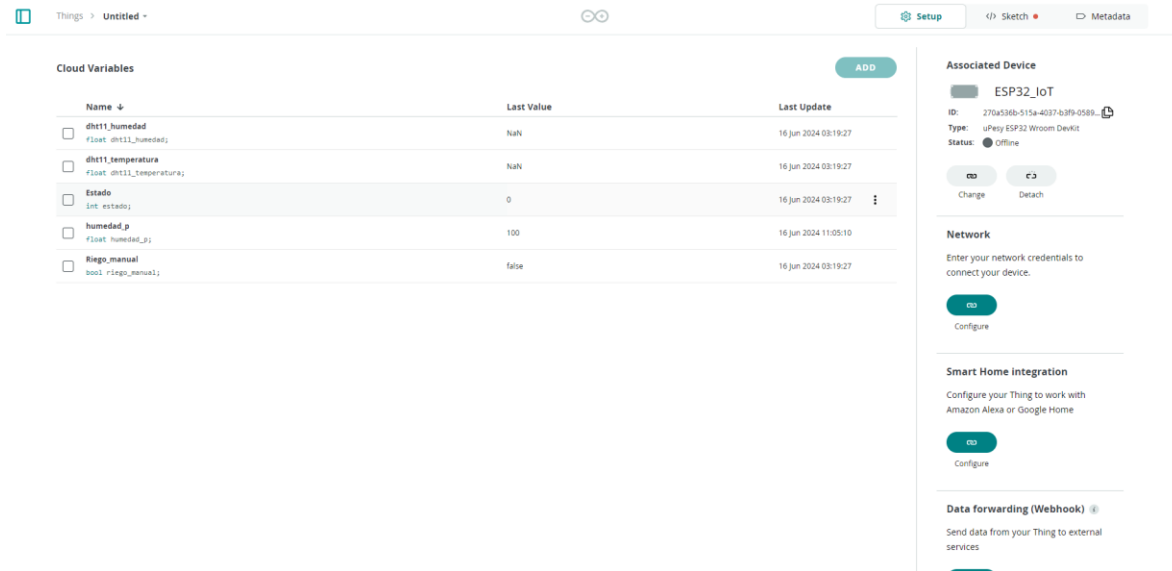
Para la implementación se utilizó Arduino cloud ya que es una plataforma en línea que facilita la creación, gestión y monitoreo del proyecto, esto permite que los usuarios puedan conectar varios dispositivos compatibles con Arduino, a simple vista y en programación es igual al Arduino IDE, SIN EMBARGO Añade la característica útil de poder conectarlo al internet y poder realizar monitorear y controlar distintas variables por medio de su dashboard también creado a medida de la necesidad del proyecto



Inicio de Arduino cloud

En la imagen anterior podemos ver como se inicializa un nuevo proyecto en Arduino cloud, la verdad si bastane intuitivo de usar, aun que lo único malo que puedo destacar es la limitación en cuantos “things” se puede agregar y también de las variables que se pueden monitorear.


Esta plataforma nos permite programar familia de arduinos y familia de ESP32 entre otras placas de desarrollo, una vez definida la placa de desarrollo a utilizar definimos las variables a utilizar.



Declaración de variables para el proyecto

En este apartado de cloud variable, definimos las variables necesarias para controlar el ecosistema IoT que querramos implementar para este proyecto se definió 5 variable.

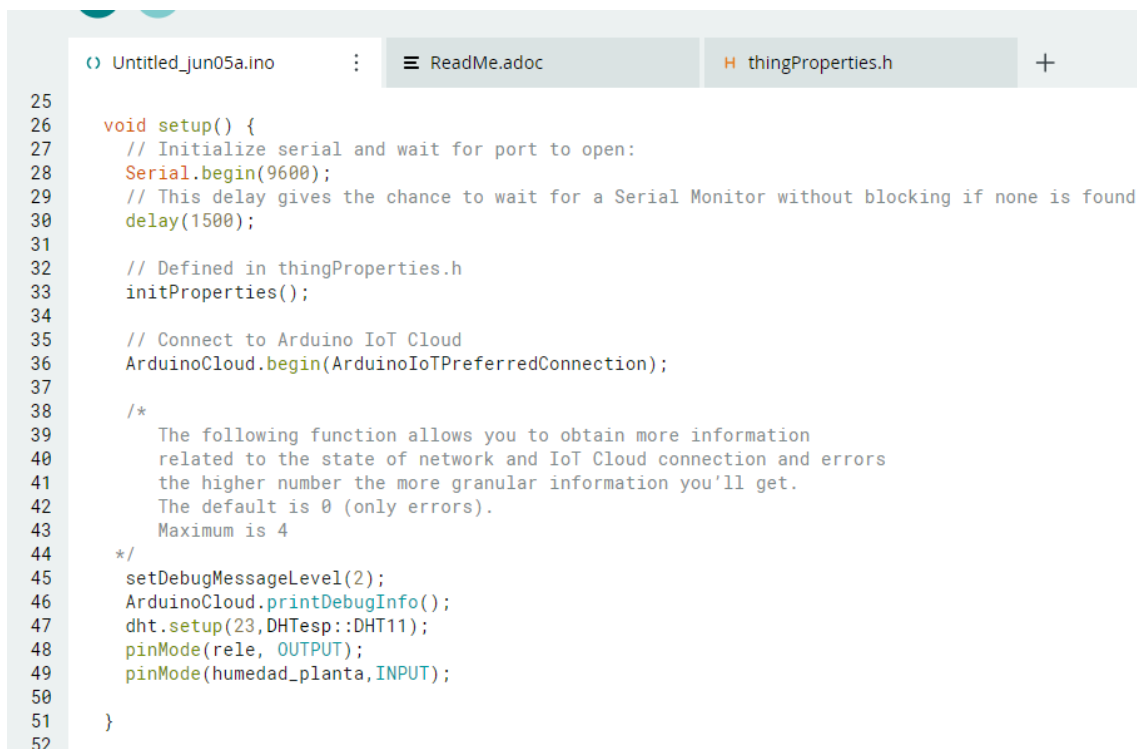
- Dht11_humedad
 - Es una variable de tipo flotante que nos permitirá monitorizar la humedad del ambiente donde esta operando el ESP32
- Dht11_temperatura
 - Es una variable de tipo flotante que nos permitirá monitorizar la temperatura del ambiente donde esta operando el ESP32
- Estado
 - Es una variable entera que nos permite monitorizar en que estado se encuentra el riego, este tiene 3 estados, riego manual, riego automático apagado, riego automático encendido
- Humedad_p
 - Esta variable es de tipo flotante ya que es utilizada para monitorear la humedad de la tierra de la planta
- Riego_manual
 - Esta variable es de tipo booleana ya que nos indica si estamos en el modo de riego manual o riego automático



```
1 #include <DHTesp.h>
2 /*
3  * Sketch generated by the Arduino IoT Cloud Thing "Untitled"
4  * https://create.arduino.cc/cloud/things/19bebfed-1b04-46da-bde9-a0dfa9d7c6fe
5  *
6  * Arduino IoT Cloud Variables description
7  *
8  * The following variables are automatically generated and updated when changes are made to the Thing
9  *
10  * float dht11_humedad;
11  * float dht11_temperatura;
12  * float humedad_p;
13  * int estado;
14  * bool riego_manual;
15  *
16  * Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
17  * which are called when their values are changed from the Dashboard.
18  * These functions are generated with the Thing and added at the end of this sketch.
19  */
```

Sketch creado a partir de las variables

Luego de crear las variables el siguiente paso es crear la lógica de programación para que el ecosistema de IoT se comporte a como uno quiere en relación a las variables medidas, Arduino cloud ya nos da toda la base de código para hacer la conexión entre el ESP32 y el internet, lo único que queda es programar la lógica para que en relación de la variables medidas, en este caso la humedad se inicie el riego.



```
25
26 void setup() {
27   // Initialize serial and wait for port to open:
28   Serial.begin(9600);
29   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
30   delay(1500);
31
32   // Defined in thingProperties.h
33   initProperties();
34
35   // Connect to Arduino IoT Cloud
36   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
37
38   /*
39   * The following function allows you to obtain more information
40   * related to the state of network and IoT Cloud connection and errors
41   * the higher number the more granular information you'll get.
42   * The default is 0 (only errors).
43   * Maximum is 4
44   */
45   setDebugMessageLevel(2);
46   ArduinoCloud.printDebugInfo();
47   dht.setup(23, DHTesp::DHT11);
48   pinMode(rele, OUTPUT);
49   pinMode(humedad_planta, INPUT);
50
51 }
52
```

Configuración de pines como entrada o salida



```
52
53 void loop() {
54   ArduinoCloud.update();
55   // Your code here
56   dht11_temperatura = dht.getTemperature();
57   dht11_humedad = dht.getHumidity();
58   humedad_p = map(analogRead(humedad_planta) ,0,4095, 100,0);
59
60   if(humedad_p <62 && estado == 0){
61     digitalWrite(rele,0); //activo rele para regar
62   }else if(estado == 0){
63     digitalWrite(rele,1);
64   }
65
66   Serial.println(digitalRead(rele));
67   Serial.println(humedad_p);
68   delay(5000);
69 }
70
```

Void loop

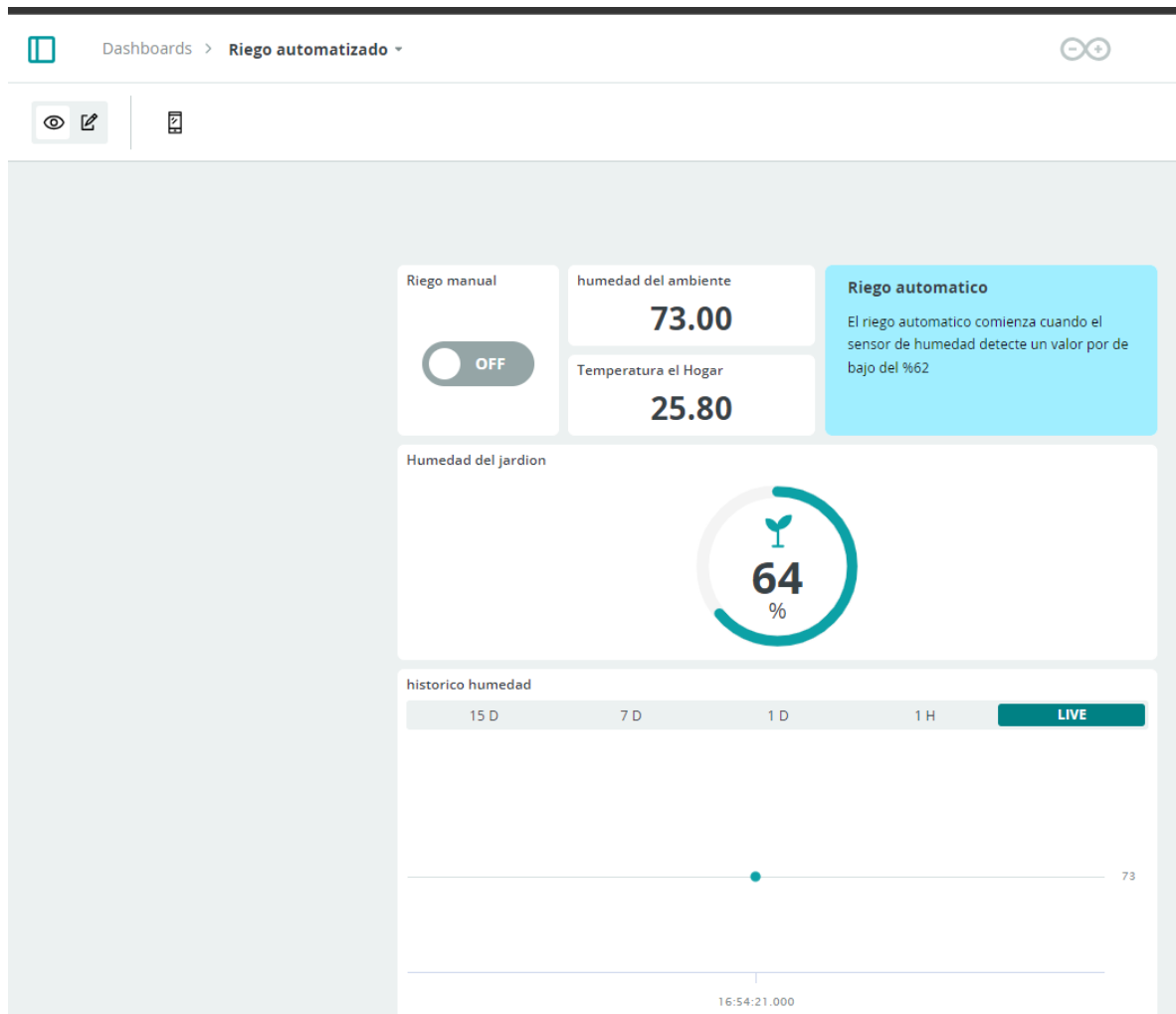
En el ciclo principal es donde por medio de la estructura condicional else-if controlamos el encendido y apagado de la bomba de agua, en la línea 60 podemos ver que la condición para encender la bomba es que la variable humedad_p sea menor a 62% y que el estado no sea 0, es decir no esté en modo manual



Dashboards			Search and filter Dashboards	+ DASHBOARD
<input type="checkbox"/> Name ↑	Last modified	Sharing with		
<input type="checkbox"/> Riego automatizado	12 Jun 2024, 4:35 p.m.	6		

Creación de dashboard

Luego de tener la lógica de programación, se crea el dashboard para poder monitorizar y controlar las variables



Dashboard

EL dashboard cuenta con 2 mediciones de humedad uno es para el ambiente en donde esta colocado el ESP32 y otro si es específicamente para la tierra de la planta, al igual que tiene un apartado para la temperatura al que esta operando el ESP32

Humedad del jardin



Indicador de la humedad de la planta

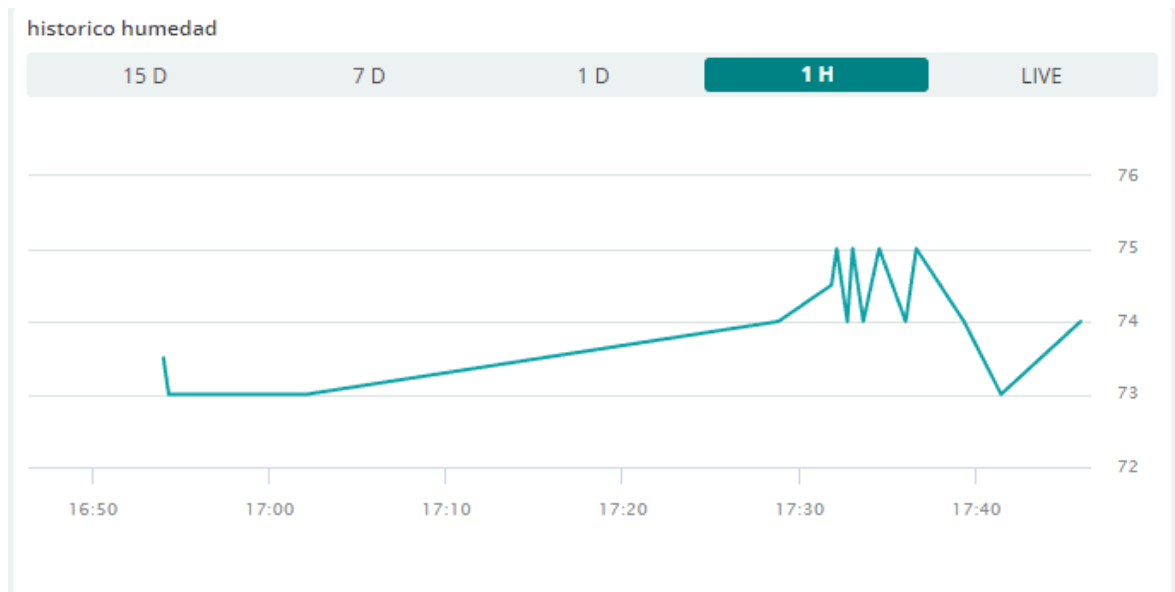


Indicador de la temperatura y la humedad del ambiente de trabajo del ESP32

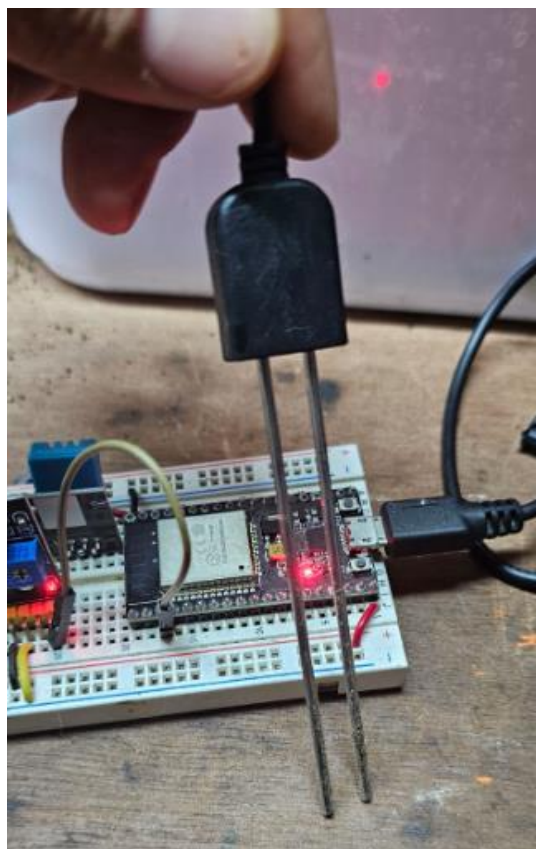
Riego manual



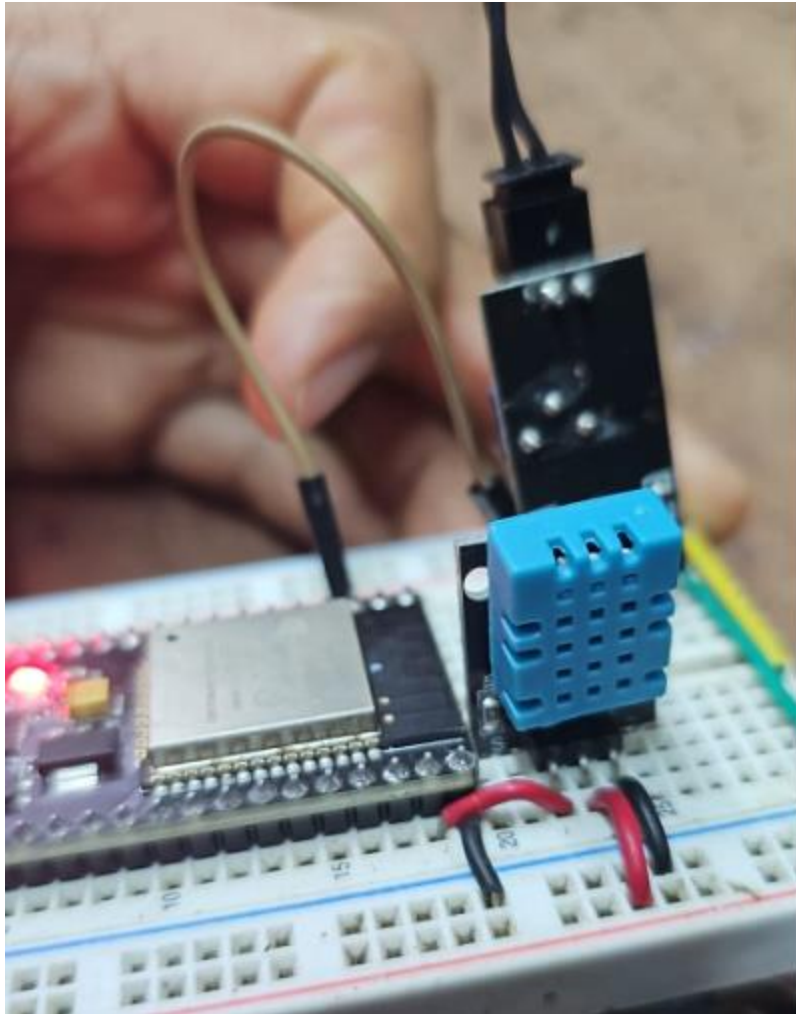
Indicador del estado del diego manual



Histórico de la temperatura



Sensor de humedad con electrodos alargados



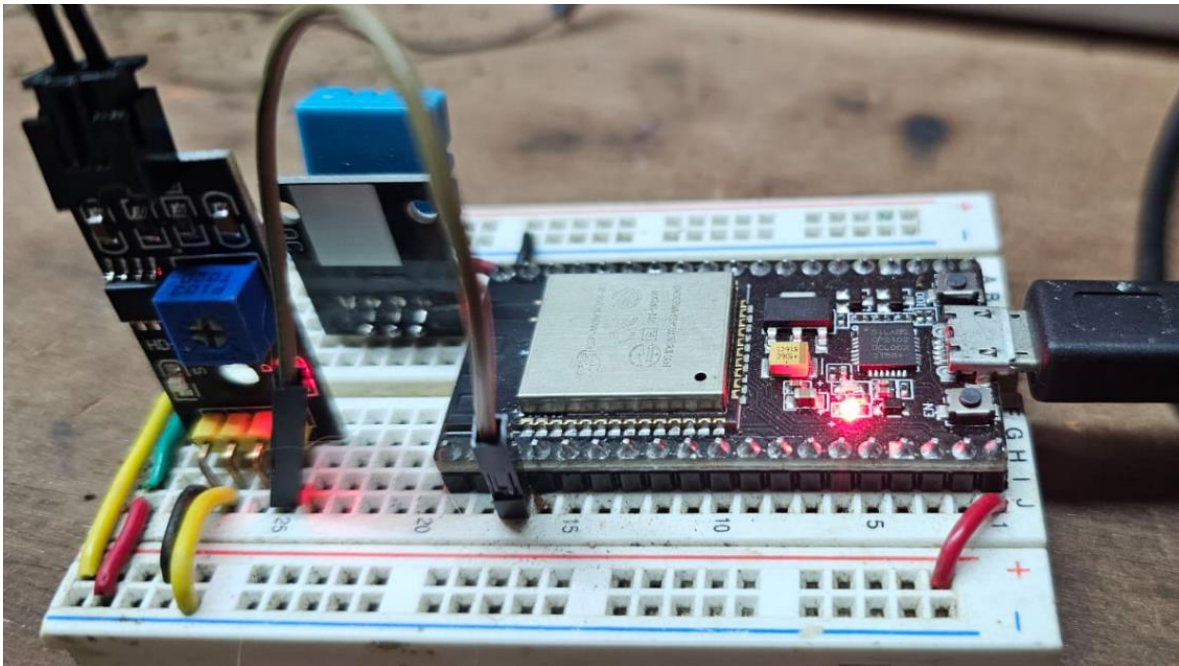
Sensor DTH11 para medir humedad y temperatura



Modulo rele



Bomba utilizada para el riego automático



Circuito utilizado en el riego automático



Implementación del riego automático

Conclusión:

Arduino cloud sin duda es una gran herramienta para la implementación de ecosistemas IoT ya que en las versiones de pago añade poder escalar este proyecto a grande escala para poder controlar un jardín grande o bien en la industria para automatizar el riego de una plantación agrícola, si bien una mejora notable seria poder tener el controlador aislada para evitar cablear tanto, y que este proyecto sea mas agradable a la vista y fácil de implementar.

Link video:

https://youtu.be/FlbxD_vwst4