

Sistema de riego Automático

Materiales necesarios:

- Protoboard o PCB
- Sensor de temperatura y humedad
- Cables
- Sensor YL-69
- MiniBomba
- Diodo 1N4007
- Transistor bipolar PN2222
- 3 Resistencias: 1x 220 ohms, 1x 1k, 1x fotoresistencia LDR
- Tanque de agua
- ESP32
- Tubo para conectar la mini bomba y llevar hasta las plantas

Concepto:

El sistema de riego automático regará siempre que detecte una serie de condiciones ambientales. Por ejemplo, cuando detecta poca luz o oscuridad la temperatura del aire es una concreta que configuraremos en el código de arduino y la humedad del suelo es baja. Ese sería el momento óptimo para regar la planta, Esto puede ser enviado a través de un Websocket e imprimirse en una interfaz de usuario.

ISPC - 2022 - Telecomunicaciones

Sensores y Actuadores

Docentes: Jorge Morales - Gonzalo Vera

Grupo 3 - Carla wayar - Gonzalo Passarell - Dario Arriola

Oscar Gazzola - Lionel Rios - Daniel Rodriguez - Jere Castro

IoT

Código de ejemplo para el proyecto:

```
#include <SimpleDHT.h>
#include <SPI.h>
#define humidity_sensor_pin A0
#define ldr_pin A5
//Bibliotecas para los módulos sensores usados necesarias
//Y definición de variables para los sensores de humedad y LDR en los pines A0 y A5

int pinDHT11 = 2;
SimpleDHT11 dht11;
int ldr_value = 0;
int water_pump_pin = 3;
int water_pump_speed = 255;
//Aquí puedes dar valores desde 0 a 255 para la velocidad a la que trabajará la minibomba
//Haz pruebas previas del caudal y configura la. Yo he //elegido 255 pero ustedes pueden
elegir la que estimen conveniente. A más velocidad, mayor //bombeo de agua
void setup() {
    Serial.begin(9600);
}
```

ISPC - 2022 - Telecomunicaciones

Sensores y Actuadores

Docentes: Jorge Morales - Gonzalo Vera

Grupo 3 - Carla wayar - Gonzalo Passarell - Dario Arriola

Oscar Gazzola - Lionel Rios - Daniel Rodriguez - Jere Castro

IoT

```
void loop() {  
  // Mide la temperatura y humedad relativa y muestra resultado  
  Serial.println("*****");  
  Serial.println("Muestra DHT11...");  
  
  byte temperature = 0;  
  byte humidity_in_air = 0;  
  byte data[40] = {0};  
  if (dht11.read(pinDHT11, &temperature, &humidity_in_air, data)) {  
    Serial.print("Lectura del sensor DHT11 fallida");  
    return;  
  }  
  
  Serial.print("Muestra RAW Bits: ");  
  
  
  for (int i = 0; i < 40; i++) { Serial.print((int)data[i]); if (i > 0 && ((i + 1) % 4) == 0) {  
    Serial.print(' ');  
  }  
}  
Serial.println("");  
  
Serial.print("Muestra OK: ");  
Serial.print("Temperatura: ");Serial.print((int)temperature); Serial.print(" *C, ");  
Serial.print("Humedad relativa en aire: ");Serial.print((int)humidity_in_air); Serial.println(" %");  
  
int ground_humidity_value = map(analogRead(humidity_sensor_pin), 0, 1023, 100, 0);  
Serial.print("Humedad en suelo: ");
```

ISPC - 2022 - Telecomunicaciones

Sensores y Actuadores

Docentes: Jorge Morales - Gonzalo Vera

Grupo 3 - Carla wayar - Gonzalo Passarell - Dario Arriola

Oscar Gazzola - Lionel Rios - Daniel Rodriguez - Jere Castro

IoT

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

```
int ldr_value = map(analogRead(ldr_pin), 1023, 0, 100, 0);  
Serial.print("Luz: ");  
Serial.print(ldr_value);  
Serial.println("%");  
Serial.println("*****");
```

```
//*****
```

```
// Condiciones de riego
```

```
// Si la humedad en el suelo es igual o inferior al 60%, si la luminosidad es inferior al 30%,
```

```
// Si la temperatura es inferior al 35%, entonces el sistema de riego riega.
```

```
// En caso de que no se cumpla alguno o ninguno de los 3 requisitos anteriores,
```

```
// el sistema de riego no riega
```

```
//*****
```

```
//Aquí puedes variar los parámetros que necesites de 60, 35 y 30, e incluso usar otros  
operandos <=>...
```

```
if( ground_humidity_value <= 60 && ldr_value<30 && temperature<35) {  
digitalWrite(water_pump_pin, HIGH);  
Serial.println("Irrigación");  
analogWrite(water_pump_pin, water_pump_speed);
```

```
}
```

```
else{
```

```
digitalWrite(water_pump_pin, LOW);  
Serial.println("Riego detenido");
```

```
}
```

```
delay (2000);
```

```
// Ejecuta el código cada 2000 milisegundos, es decir, 2 segundos. Puedes variar la  
frecuencia de muestreo
```

```
}
```


ISPC - 2022 - Telecomunicaciones

Sensores y Actuadores

Docentes: Jorge Morales - Gonzalo Vera

Grupo 3 - Carla wayar - Gonzalo Passarell - Dario Arriola

Oscar Gazzola - Lionel Rios - Daniel Rodriguez - Jere Castro

IoT