

Smart Macet



Profesores:

Alumnos:

Luca Sain

Ricardo Choque

Introducción

Este proyecto consistía en la implementación de algunos elementos electrónicos junto con unos microcontroladores como lo son le Arduino y el Raspberry, también resolver una necesidad.

En base a eso se nos ocurrió automatizar algo necesario pero simple, funcional y completamente comercial y un posible proyecto para lanzar al mercado inmediato. Como lo es una maceta automatizada que se cuida sola/e.

A este proyecto “SmartMacet” es una maceta tecnológica que se riega sola cuando lo necesite, tiene diferentes sensores para su automatización de regado que más adelante los nombraremos a detalle, también tienen un sistema que cada vez que se riega sola, mande un mail a alguna cuenta asociada perteneciente al usuario notificando que se está regando con regularidad, también tiene incorporada un “Led” para cuando el sistema esté en funcionamiento, incorporamos un botón para que si el usuario decide regarlo de forma manual, la pueda regar manualmente y sin necesidad que ir a buscar agua.

Planteamiento y ejecución del Proyecto

Nos planteamos que elementos utilizar, como debería funcionar, actuar, y como ejecutarse. Basándonos en eso, separamos los sensores, actuadores, resistencias y fuimos conectando todo a medida que probamos el correcto funcionamiento. Una vez conectado todo al arduino, cargamos el código, y conectamos por el puerto serial- usb al raspberry para tener una conexión, hacer un censo y que se envíen un mail al usuario

Elementos Utilizados

- 1 Arduino Mega 2560 v3
- 1 Raspberry PI 3
- 1 sensor de Humedad
- 1 Buzzer
- 1 Resistencia de 10k (Para el Pulsador)
- 1 Resistencia de 220 ohm (Para el Led)
- 2 Resistencias de 220 ohm (conectadas en paralelo simulando una resistencia de 110)
- 1 Protoboard (Para montar todo el Circuito electrónico)
- 1 Correo GMAIL (para en envió de correos)
- 1 server mqtt
- 1 server Node-Red

Código Arduino

```
Smart_Maceta.4.0 $  
  
// Sensor de Humedad  
  
// Conectamos el sensor de la siguiente forma:  
// GND    -> GND  
// VCC    -> 5V  
// DAT    -> A0  
// Por ejemplo conectamos a las entrada Analógica 0  
// http://arubia45.blogspot.com.es/  
  
// Descripción de valores del Sensor  
  
// 0  -300      Seco  
// 300-700      Húmedo  
// 700-950      En Agua  
const int relay_pin = 8;  
const int buzzer_pin = 10;  
const int tilt_pin = 11;  
const int humidity_pin = 8;  
const int buttonPin = 2;  
int trigger_humedad = NULL;  
  
int buttonState = 0;  
int Valor;  
|  
void setup() {  
    pinMode(relay_pin, OUTPUT);  
    pinMode(buzzer_pin, OUTPUT);  
    pinMode(tilt_pin, OUTPUT);  
    pinMode(buttonPin, INPUT);  
  
    Serial.begin(9600);  
    Serial.println("Smart Maceta");  
  
}  
  
void loop() {  
    buttonState = digitalRead(buttonPin);  
  
    Serial.print("Sensor de Humedad valor:");  
    Valor = analogRead(humidity_pin);  
    Serial.print(Valor);  
    if (Valor <= 300) {  
        trigger_humedad=0;  
        Serial.println(" Encharcado");  
        digitalWrite(relay_pin, LOW);  
    }  
}
```

```
        mqttduino("relay",String(digitalRead(relay_pin)));
    }
    if ((Valor > 300) and (Valor <= 700)){
        Serial.println(" Humedo, no regar");
        mqttduino("relay",String(digitalRead(relay_pin)));
    }
    if (Valor > 700){
        if(trigger_humedad==0){
            Serial.println(" Seco, necesitas regar");
            mqttduino("relay",String(digitalRead(relay_pin)));
            tone(buzzer_pin, 1000, 500);
            digitalWrite(relay_pin, HIGH);
            mqttduino("ya-regado","recien se regó la maceta");
            trigger_humedad=1;
        }

    }
    if ((buttonState == HIGH)and(Valor<300)) {
        digitalWrite(relay_pin, HIGH);
        Serial.println("BUTTON PRENDIDO");
    } else if((buttonState == LOW)and(Valor>300)){
        Serial.println("BUTTON APAGADO");
    }

    delay(1000);
}

void mqttduino (String channel, String data)
{
    Serial.print("@");
    Serial.print("mqtt");
    Serial.print(",");
    Serial.print(channel);
    Serial.print(",");
    Serial.println(data);
    delay(10);
}
```

Codigo Python

```
import paho.mqtt.client as mqttClient
import time
import serial
encoding = 'utf-8'
humidity = '@'
ser = serial.Serial('COM4', 9600)
#ser1 = serial.Serial('/dev/ttyACM0', 9600)
#https://techtutorialsx.com/2017/04/14/python-publishing-messages-to-mqtt-topic/
def on_connect(client, userdata, flags, rc):

    if rc == 0:

        print("Connected to broker")

        global Connected                #Use global variable
        Connected = True                #Signal connection

    else:

        print("Connection failed")

Connected = False    #global variable for the state of the connection

broker_address= "localhost"
port = 1883
user = ""
password = ""

client = mqttClient.Client("Python")           #create new instance
#client.username_pw_set(user, password=password)    #set username and password
client.on_connect= on_connect                  #attach function to callback
client.connect(broker_address, port=port)       #connect to broker

client.loop_start()          #start the loop

while Connected != True:     #Wait for connection
    time.sleep(0.1)

try:
    while True:

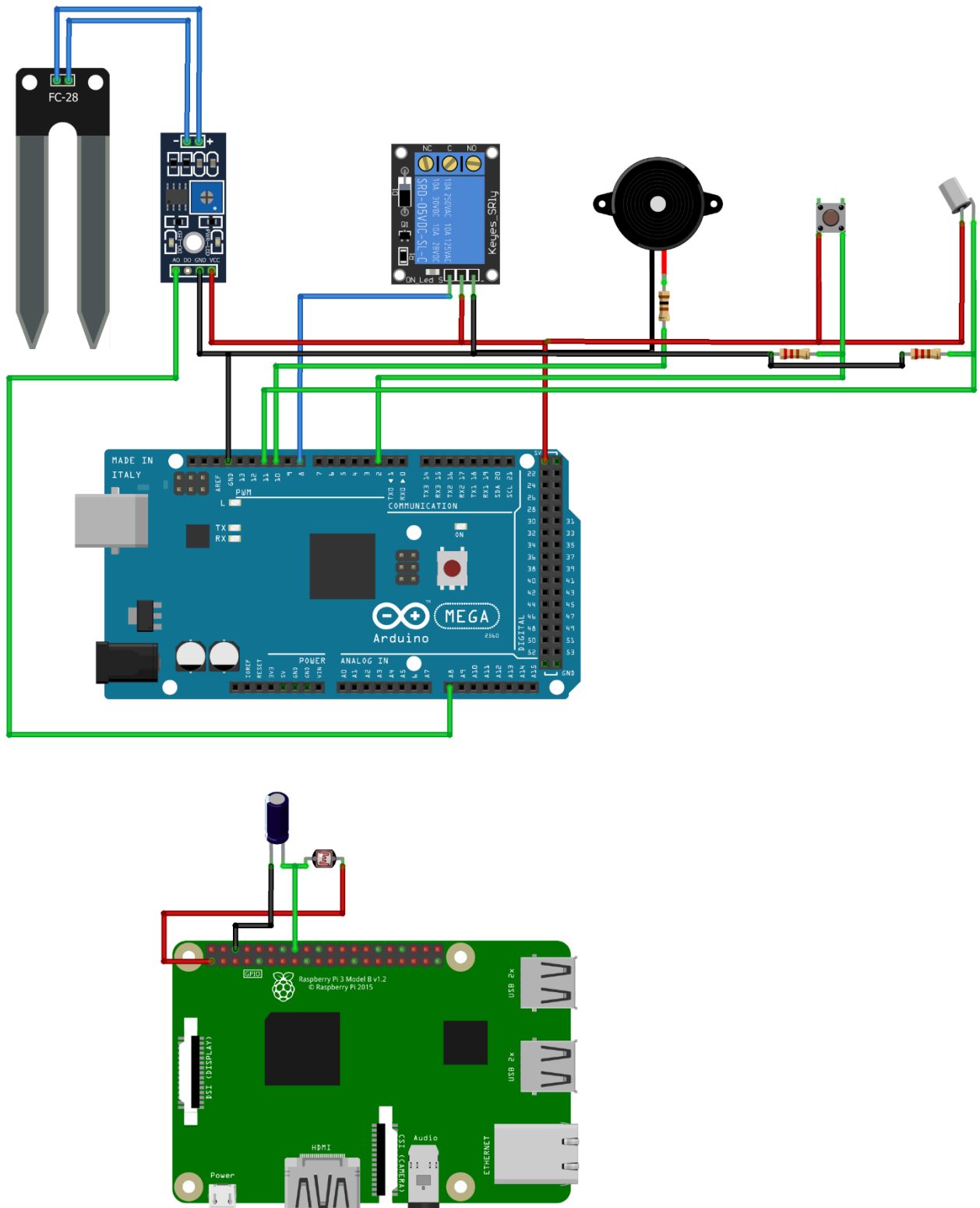
        #value = input('Enter the message:')
```

```
#client.publish("python/test",value)
line = str(ser.readline(), encoding)
if(line[0] == '@'):
    print ("Soy El Primer Parametro")
    print (line.split(",",-1)[0])
    print ("Soy El Segundo Parametro")
    print (line.split(",",-1)[1])
    print ("Soy El Tercer Parametro")
    print (line.split(",",-1)[2])
    client.publish(line.split(",",-1)[1],line.split(",",-1)[2])
    time.sleep(0.1)

except KeyboardInterrupt:

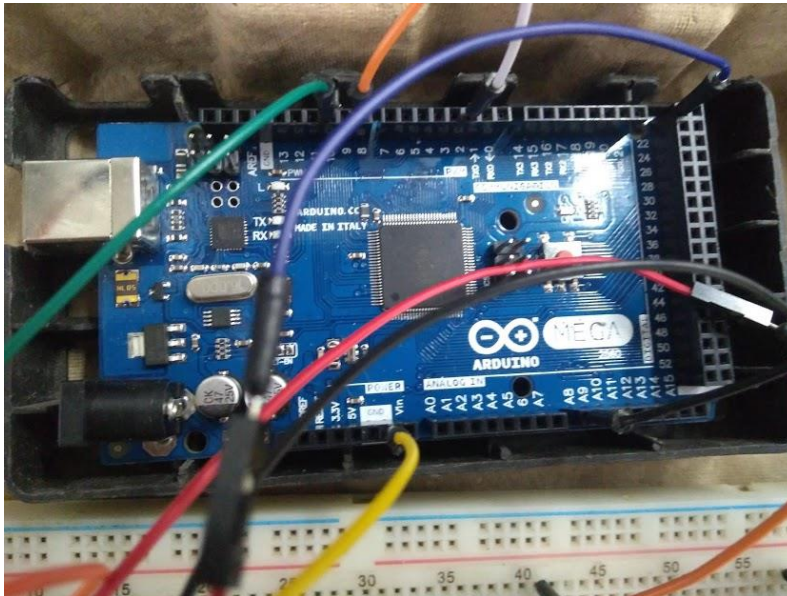
    client.disconnect()
    client.loop_stop()
```

Circuito



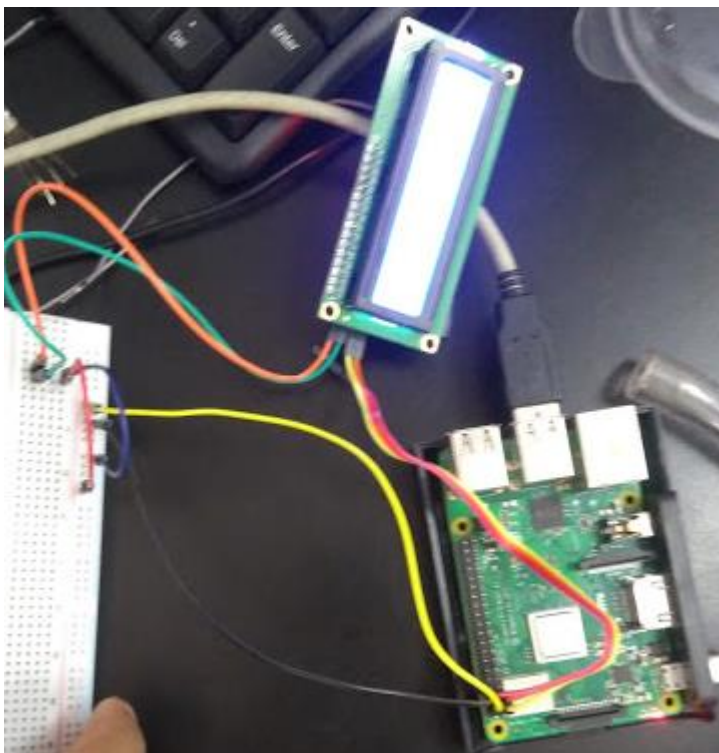
fritzing

Maqueta Arduino



Código Raspberry

Maqueta Raspberry PI



El LCD de la foto no lo llegamos a implementar al proyecto debido al corto tiempo

Integrantes con el Proyecto en funcionamiento

