

Programming Guide

made by:Alejandro Rizo Reyes

Introduction

Quick guide to learn how the programming part of the project was done.

Arduino Uno R3

Libraries

So as to use several sensor and modules we used this libraries

- dht.h (For the Humidity and Temperature sensor)
- LedControl.h (For the 8x8 led grid)
- LiquidCrystal_I2C.h (For the LCD adapater)
- Wire.h
- SoftwareSerial.h (For the connection with the ESP8226-01)

```
#include <dht.h>
#include <LedControl.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SoftwareSerial.h>
```

PIN connection

Here is the list of all the pin connections

```
SoftwareSerial mySerial(4,5);

LedControl lc = LedControl(10,8,9,1);
dht DHT;
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7);

#define DHT11_PIN 13
#define YL69_PIN A0
#define SEN2609_PIN A1
#define KY018_PIN A2
#define SCL_PIN A4
#define HCSR04_Tri_PIN 12
#define HCSR04_Echo_PIN 11
#define onButton_PIN 2
#define plantButton_PIN 3
```

Setup

So as to initiate the project we must set all the components, pin modes, lcd start and lc connection and Serial

```
void setup() {
    Serial.begin(9600);
    pinMode(onButton_PIN, INPUT);
    pinMode(HCSR04_Tri_PIN, OUTPUT);
    pinMode(HCSR04_Echo_PIN, INPUT);
    pinMode(onButton_PIN, INPUT);
    pinMode(plantButton_PIN, INPUT_PULLUP);

    lc.shutdown(0, false);
    lc.setIntensity(0, 4);
    lc.clearDisplay(0);

    lcd.begin (16,2);

    // Switch on the backlight
    lcd.setBacklightPin(3, POSITIVE);
    lcd.setBacklight(HIGH);
    lcd.home ();

    mySerial.begin(115200);
}
```

Loop

Read sensors

We use the read pin for each sensor and we store the data in a variable

```

//VALOR LUZ
valorLuz = analogRead(A1);

//SENSOR DE HUMEDAD Y TEMPERATURA SENCILLO: DHT11
chk = DHT.read11(DHT11_PIN);

//EXTRACCION DE DATOS TEMPERATURA Y HUMEDAD
tempPlant = DHT.temperature;
humPlant = DHT.humidity;

//EXTRACCION DE DATOS HUMEDAD TIERRA
lecturaSensorYL69 = analogRead(YL69_PIN);
humTPlant = map (lecturaSensorYL69, 1023, 0, 0, 100);

//SENSOR DE NIVEL DEL AGUA: SEN-2609
waterLvl = analogRead(SEN2609_PIN);

```

```

//EXTRACCION DE DATOS SENSOR DE LUZ: PON EL NOMBRE DEL SENSOR
valorLuz = analogRead(KY018_PIN);
valorLuzMapeado = map(valorLuz,0,1023,0,100);

//SENSOR DE PROXIMIDAD
digitalWrite(HCSR04_Tri_PIN, HIGH);
delay(1);
digitalWrite(HCSR04_Tri_PIN, LOW);

duracion = pulseIn(HCSR04_Echo_PIN, HIGH);
distancia = duracion/58.2;

```

Determine Water Level

We take the data from the water level sensor and we determine on which bracket we have our water tank, this value is different depending on the mineralisation of the water, this data was for the water we can find on the Valencian water system.

```

//COMPROBACION NIVEL AGUA
if(waterLvl <= 100){
    aguaStatus= "Vacio!";
    llenarAgua = 1;
} else if(waterLvl > 100 && waterLvl <= 230){
    aguaStatus= "Bajo!";
    llenarAgua = 0;
} else if(waterLvl > 230 && waterLvl <= 270){
    aguaStatus = "Medio";
    llenarAgua = 0;
} else{
    aguaStatus = "Alto";
    llenarAgua = 0;
}

```

Proximity Sensor

If the proximity sensor detects a presence in 15 cm or less it activates the LCD display and the 8x8 grid

```
if(distancia <= 15){  
    if(humTSalud == 0 && humSalud == 0 && tempSalud == 0){  
        lc.clearDisplay(0);  
  
        lcd.setBacklight(HIGH);  
        lcd.clear();  
        lcd.setCursor(0, 0);  
        lcd.print("Cuidando:" + nomPlant);  
        lcd.setCursor(0, 1);  
        lcd.print("T:" + String(tempPlant));  
        lcd.setCursor(5, 1);  
        lcd.print("H:" + String(humPlant));  
        lcd.setCursor(10, 1);  
        lcd.print("HT:" + String(humTPlant));  
        //CARITA FELIZ  
        //FILA 0  
        lc.setLed(0,0,1,true);  
        lc.setLed(0,0,2,true);  
        lc.setLed(0,0,3,true);  
        lc.setLed(0,0,4,true);  
        lc.setLed(0,0,5,true);  
        lc.setLed(0,0,6,true);  
  
        //FILA 1  
        lc.setLed(0,1,0,true);  
        lc.setLed(0,1,7,true);  
    }  
}
```

If not it clears the displays

```
} else {  
    lc.clearDisplay(0);  
    lcd.clear();  
    lcd.setBacklight(LOW);  
}
```

In this brackets we also print the happy or sad face depending on the status of the plant

Determine Status Of The Plant

Checking the personalyse values for each plant profile we determine if its necessities are being fulfilled.

```

//ESTADO PLANTA
if(tempMIN < DHT.temperature && DHT.temperature < tempMAX){
| tempSalud = 0;
} else {
| tempSalud= 1;
}

if(humMIN < DHT.temperature && DHT.temperature < humMAX){
| humSalud = 0;
} else {
| humSalud = 1;
}

if(humTMIN < DHT.temperature && DHT.temperature < humTMAX){
| humTSalud = 0;
} else {
| humTSalud = 1;
}

```

Button Set Mode

We determine the mode via the value that it is being held now by the variable and the button

```

if(buttonStateNew == 1 && buttonState == 1){
| if(pulsado){
| | pulsado = false;
| } else {
| | pulsado = true;
| }
}

```

Button Set Plant

We determine the plant profile by incrementing or decreasing the value of the selection variable.

```
lcd.setBacklight(HIGH);
if(buttonPlantStateNew == 0 && buttonPlantState == 1){
    if(plantSelect < numPlantas){
        plantSelect++;
    } else {
        plantSelect = 1;
    }
    delay(500);
}
```

```
switch(plantSelect){
    case 1:
        //VALORES DE TEMPERATURA ROSA
        tempMAX = 30;
        tempMIN = 15;
        //VALORES DE HUMEDAD AIRE ROSA
        humMAX = 80;
        humMIN = 50;
        //VALORES DE HUMEDAD TIERRA ROSA
        humTMAX = 90;
        humTMIN = 40;
        //NOMBRE PLANTA = ROSA
        nomPlant = "ROSA";
        break;

    case 2:
        //VALORES DE TEMPERATURA TULIPAN
        tempMAX = 28;
        tempMIN = 15;
        //VALORES DE HUMEDAD AIRE TULIPAN
        humMAX = 95;
        humMIN = 75;
        //VALORES DE HUMEDAD TIERRA TULIPAN
        humTMAX = 90;
        humTMIN = 50;
        //NOMBRE PLANTA = ROSA
        nomPlant = "TULIPAN";
        break;
```

ESP8226-01

Libraries

So as to correctly use the ESP8226 we use this libraries:

- SoftwareSerial.h
- ESP8266WiFi.h
- WiFiClientSecure.h
- UniversalTelegramBot.h

```
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
```

WiFi SetUp

We set the credentials to our network and in the setup we initiate the connection, later we send a “Conectado” message to our Telegram bot.

```
const char* ssid      = "WIFI NAME";
const char* password = "PASSWORD";
```

```
void setup() {
    Serial.begin(115200);
    Serial.print("Funciono");

    mySerial.begin(115200);

    WiFi.begin(ssid, password);
    while(WiFi.status() != WL_CONNECTED){
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi");

    net_ssl.setInsecure();

    bot.sendMessage(id,"Conectado");
}
```

Handle Telegram Messages

When a new message arrives to the Telegram Chat, the ESP8226-01 reads it and depending on the content of the messages it sends back the response.

```
void handleNewMessages(int numNewMessages){
    for(int i = 0; i<numNewMessages; i++){
        String id = String(bot.messages[i].chat_id);
        String text = bot.messages[i].text;

        String from_name = bot.messages[i].from_name;
        if(from_name == "") from_name = "Guest";

        if(text == "/status"){
            bot.sendMessage(id, "Su planta:");
        }

        if(text == "/help"){
            bot.sendMessage(id, "Todos los comandos que puedes usar son:\n /status:Te muestra el estado de tu planta");
        }
    }
}

void loop() {
    long now = millis();

    if(now >= checkTelegramDueTime){
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while(numNewMessages){
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        checkTelegramDueTime = now + checkTelegramDelay;
    }
}
```