



Irrigação IOT : Sistema de Irrigação

Carolina Rodrigues Fernandes, Wilian França

¹Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

Carolina_rfernandes@hotmail.com

1. Materiais e Métodos

1.1. Materiais

1. Arduino Uno (1 unidade): é uma placa de Arduino que tem como microcontrolador principal o ATmega328P. Ela possui 14 pinos digitais que podem ser entrada e/ou saída e desses, 6 pinos ser usados como saída PWM. E tem também uma conexão USB e uma ligação de fonte de energia, um conector de programação e um botão de reset.

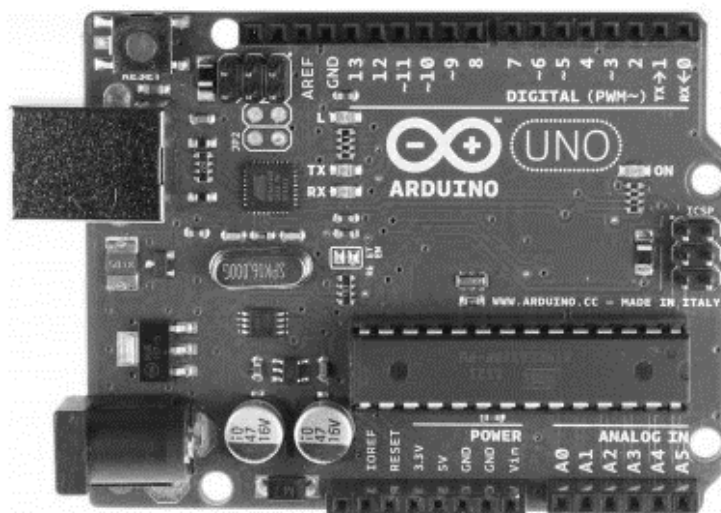


Figura 1 - Arduino Uno

2. Módulo ESP8266 NodeMcu ESP-12E com Wifi V3 (1 unidade): é uma placa que junta o ESP8266 (um chip de 32 bits com Wi-Fi integrado), regulador de tensão 3,3V. A programação pode ser feita pela IDE do Arduino. Possui antena embutida e um conector micro-USB, 11 pinos I/O e conversor analógico-digital.

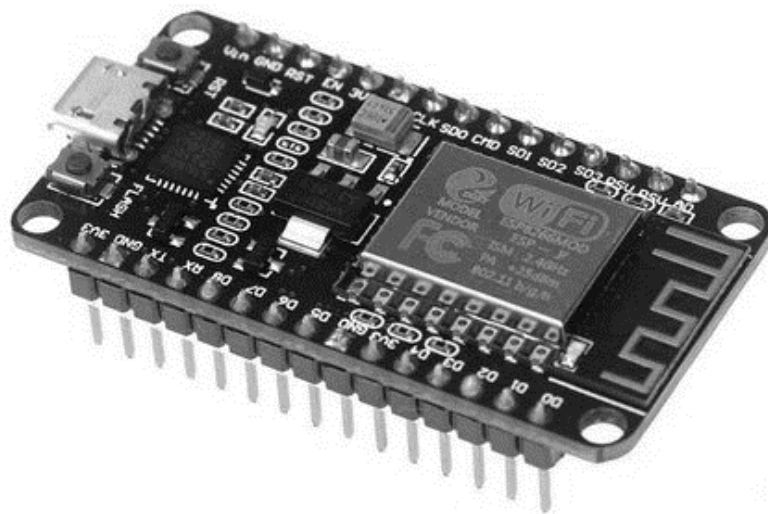


Figura 2 - Módulo WiFi ESP8266 ESP-12E

3. Jumper Premium 40p x 20cm – Macho/Fêmea (1 unidade): Cabo de 40 vias destacáveis, colorido de 20 cm Macho/Fêmea, para Arduino.



Figure 3 - Jumper Premium 40p x 20cm

4. Jumper Premium 40p x 20cm – Macho/Macho (1 unidade): Os Jumpers são responsáveis por desviar, ligar ou desligar o fluxo elétrico e assim permitindo configurações específicas. São cabos de 40 vias destacáveis de 20 cm.

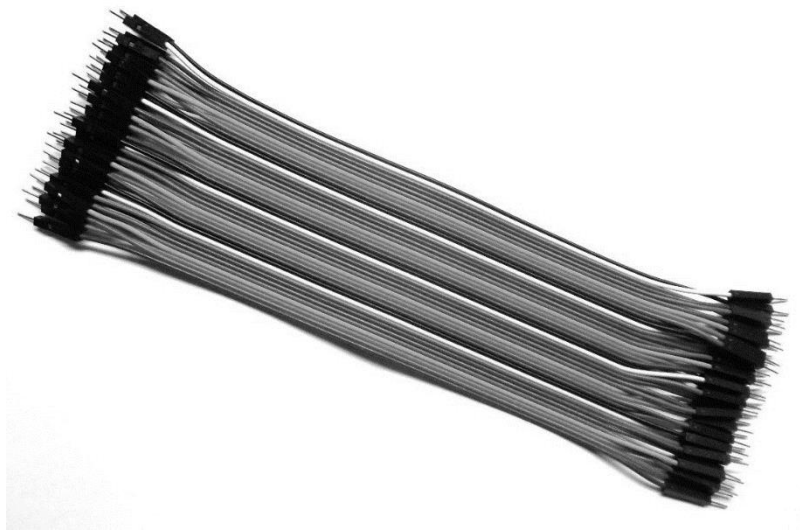


Figure 4 - Jumper Premium 40p x 20cm – Macho/Macho

5. Fonte Bivolt (1 unidade): Fonte de energia para circuitos eletrônicos. Sua Voltagem é de 12 e Amperagem de 1. É uma fonte chaveada, sendo assim traz estabilidade no controle de tensão. Possui o conector P4 e tomada no padrão brasileiro.



Figure 5 - Fonte Bivolt

6. Sensor de umidade de solo (1 unidade): Esse sensor é dividido em duas partes. A primeira, é a de Sonda, parte que efetivamente tem contato com o solo. Ela possui hastes com trilhas metalizadas para detector a umidade.

A segunda parte, é a parte que está o módulo com circuito integrado (CI) comparados LM393. Ele possui duas saídas, a digital (D0), quanto a analogia (A0), e o trimpot de ajuste para calibrar os níveis lógicos de saída.

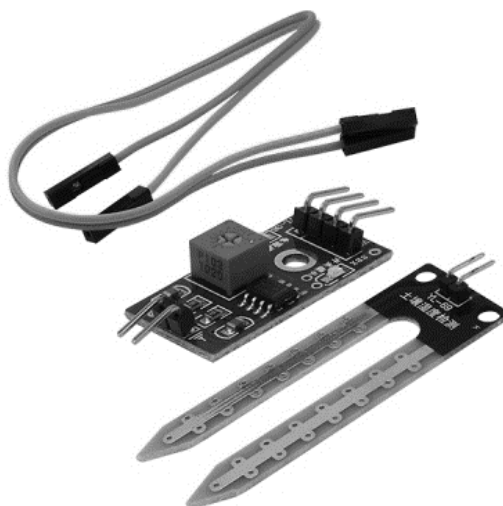


Figure 6 - Sensor de umidade

7. Válvula Solenoide de Entrada e Saída de Água (1 unidade): Esta válvula serve para controle de fluxo de água para o projeto. Quando é alimentada com uma tensão de 12V, permite a vazão da água e quando esse fluxo de energia para, a passagem de água é cortada. Sua entrada e saída é de $\frac{3}{4}$ polegada, opera com uma pressão de 0,2 a 8 kgf/cm².

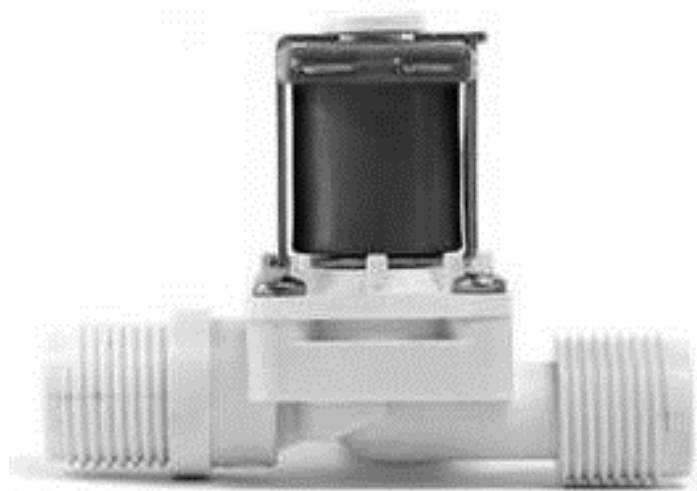


Figure 7 - Válvula Solenoide de Entrada e Saída de Água

8. Módulo Relé (1 unidade): O modulo relé pode ser usado para controle de luzes, equipamentos ligados a redes elétrica, entre outros. Sua função é acionar cargas de até 250VAC a 7A ou 125VAC a 14A. Ele opera com uma tensão de 5V.

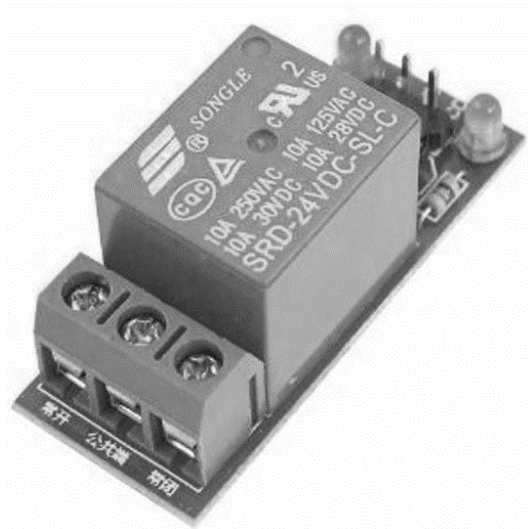


Figure 8 - Módulo Relé

9. Adaptador Borne de pressão para J4 Fêmea (1 unidade): O adaptador permite conectar uma placa de alimentação P4 e um cabo de terminais.



Figure 9 - Adaptador J4 Fêmea

10. Protoboard 830 pontos MB-102 (1 unidade): É uma placa com furos e conexões condutoras utilizada para a montagem de protótipos e projetos em estado inicial.

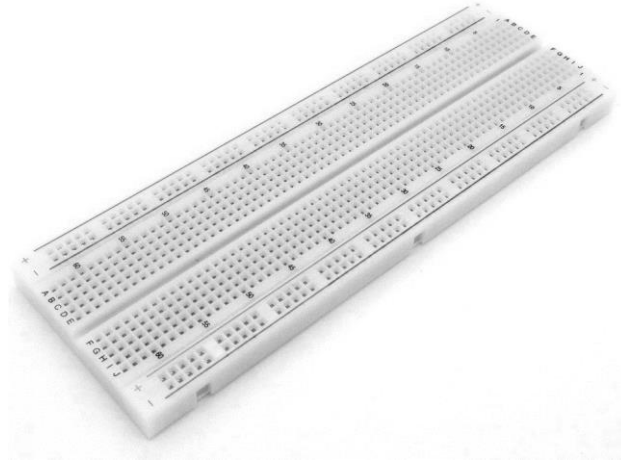


Figure 10 - 10. Protoboard 830 pontos MB-102

1.2. Método

Para realizar esse projeto, foi usado uma válvula solenoide de entrada e saída de água, ela funciona com uma tensão de 12V DC, ela é conectada a uma mangueira com fluxo contínuo, e quem faz a regulação é o Relé, que recebe as instruções do

Arduino UNO acionado a partir do sensor de umidade de solo, e nele também está conectado o NodeMCU utilizado apenas para conexão WI-FI.

Os dados são enviados e armazenados na plataforma ThingSpeak, onde pode ser visto os dados por meio de gráficos. O modo de comunicação é o MQTT.

2. Software desenvolvido – Código

```
#include <WiFi.h>
```

```
#include <WiFiClient.h>
```

```
#include <WiFiServer.h>
```

```
#include <WiFiUdp.h>
```

```
#define SSID_REDE    "House_Pokemon_"
```

```
#define SENHA_REDE   "Carolina17121998*"
```

```
#define INTERVALO_ENVIO_THINGSPEAK 20000
```

```
char EnderecoAPIThingSpeak[] = "api.thingspeak.com";
```

```
String ChaveEscritaThingSpeak = "J0AXPPG6M97P7145";
```

```
long lastConnectionTime;
```

```
WiFiClient client;
```

```
void EnvioDadosThingspeak(String StringDados);
```

```
void ConexaoWiFi(void);
```

```
float LeituraUmidade(void);
```

```
void EnvioDadosThingspeak(String StringDados)
```

```
{
```

```
    if (client.connect(EnderecoAPIThingSpeak, 80))
```

```
    {
```

```
        client.print("POST /update HTTP/1.1\n");
```

```
        client.print("Host: api.thingspeak.com\n");
```

```
        client.print("Connection: close\n");
```

```
        client.print("X-THINGSPEAKAPIKEY: "+ChaveEscritaThingSpeak+"\n");
```

```
        client.print("Content-Type: application/x-www-form-urlencoded\n");
```

```
        client.print("Content-Length: ");
```

```
        client.print(StringDados.length());
```

```
        client.print("\n\n");
```

```
        client.print(StringDados);
```

```
        lastConnectionTime = millis();
```

```
        Serial.println("- Informações enviadas ao ThingSpeak!");
```

```
    }
```

```
}
```

```
void ConexaoWiFi(void)
```

```
{
```

```
    client.stop();
```



```
Serial.println("Conectando-se à rede WiFi...");

Serial.println();

delay(1000);

WiFi.begin(SSID_REDE, SENHA_REDE);


while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}


Serial.println("");

Serial.println("WiFi conectado com sucesso!");

Serial.println("IP obtido: ");

Serial.println(WiFi.localIP());


delay(1000);
}
```

```
float UmidadeSolo(void)
{
    int ValorADC;

    float UmidadePercentual;
```

```

    ValorADC = analogRead(0);

    Serial.print("[Leitura ADC] ");

    Serial.println(ValorADC);


    UmidadePercentual = 100 * ((978-(float)ValorADC) / 978);

    Serial.print("[Umidade Percentual] ");

    Serial.print(UmidadePercentual);

    Serial.println("%");


    return UmidadePercentual;
}

void setup()
{
    Serial.begin(9600);

    lastConnectionTime = 0;

    FazConexaoWiFi();

    Serial.println("Sistema de Irrigacao");
}


void loop()
{
    float UmidadeLida;

    int UmidadeTruncada;

    char FieldUmidade[11];

```

```
if (client.connected())
{
    client.stop();
    Serial.println("- Desconectado do ThingSpeak");
    Serial.println();
}

UmidadePercentualLida = FazLeituraUmidade();
UmidadePercentualTruncada = (int)UmidadePercentualLida;

if(!client.connected() &&
    (millis() - lastConnectionTime > INTERVALO_ENVIO_THINGSPEAK))
{
    sprintf(FieldUmidade,"field1=%d",UmidadePercentualTruncada);
    EnviaInformacoesThingspeak(FieldUmidade);
}

delay(1000);
}
```

3. Interface

A interface utilizada é da plataforma ThingSpeak, foi utilizado o MQTT para transferência de mensagens e o protocolo TCP/IP para conexão de internet através do WI-FI.

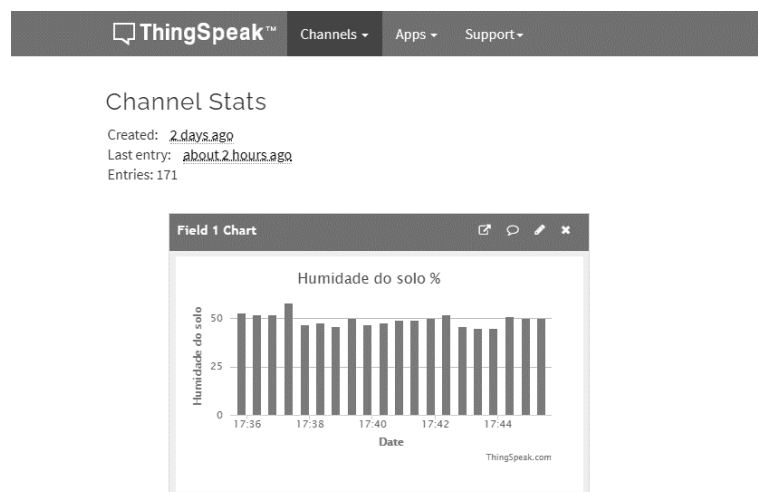


Figura 11 - Interface do ThingSpeak