

Projeto Code IoT Arduino: Cancela Automática Por WiFi

Lucas Chagas

Para a entrega do projeto do curso de Arduino, foi desenvolvido um circuito com um sensor ultrassônico HC-SR04 e um atuador, sendo o micro servo motor. Além disso, foi utilizado o módulo ESP-01S para comunicação com a rede LAN. O circuito pode ser visto na figura 1. Devido a falta de softwares em minha máquina, juntei os circuitos mostrados na figura 2 e na figura 3 para desenvolvimento na protoboard. Além do que, foi utilizado o Arduino Mega 2560. Contudo, esta junção é bem simples, basta utilizar as outras portas disponíveis conforme a lógica dos circuitos montados nas figuras. Outro atuador utilizado foi o led interno do Arduino, pois conforme o vídeo apresentado no link abaixo, quando o led está apagado, o sensor não ativa o servo motor. Porém, quando o led é ativado pelo link <http://10.0.0.180/LED=ON>, o sensor passa a ativar o servo motor.

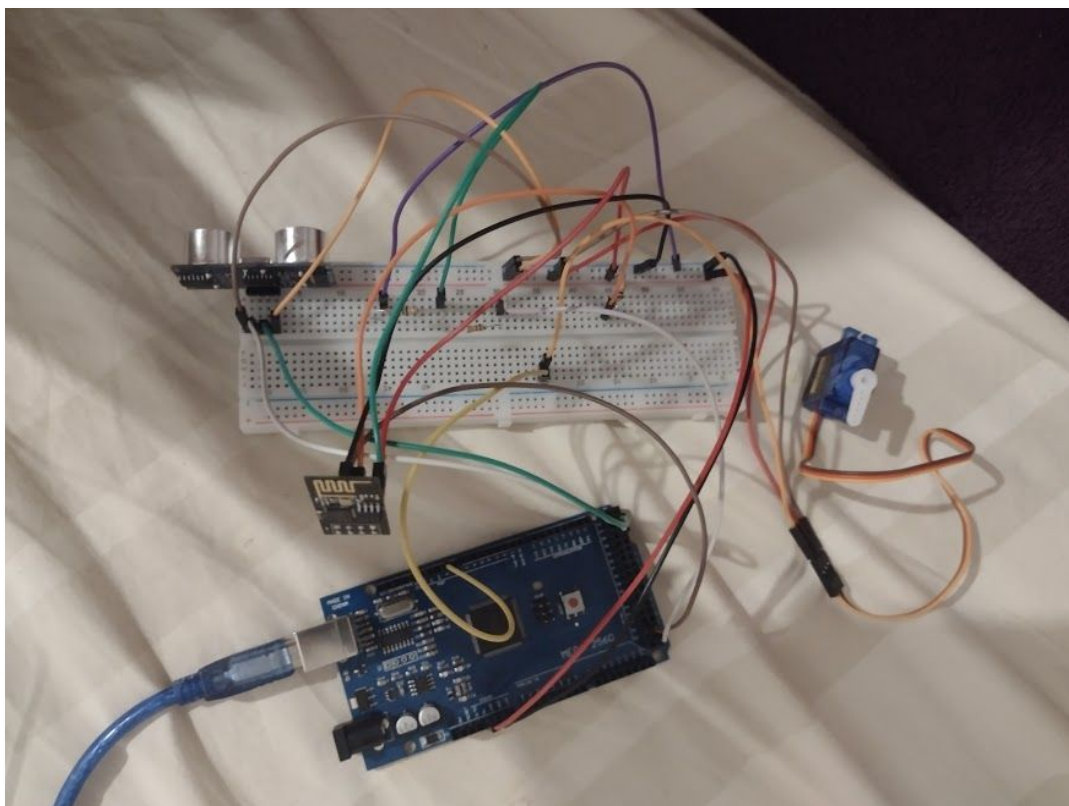
Uma possível aplicação para este projeto, seria a possibilidade de utilizar para liberação de alimentos para animais domésticos de pequeno porte, como pássaros (canários, periquitos, calopsitas) ou roedores. Alguns alimentos para estes animais, como frutas, ficam expostos a insetos. Com esta aplicação rodando, uma “portinha” poderia ser aberta apenas quando o animal se aproxima.

O código desenvolvido para a aplicação pode ser visto no final deste documento e segue abaixo o link do vídeo do seu funcionamento.

Link:

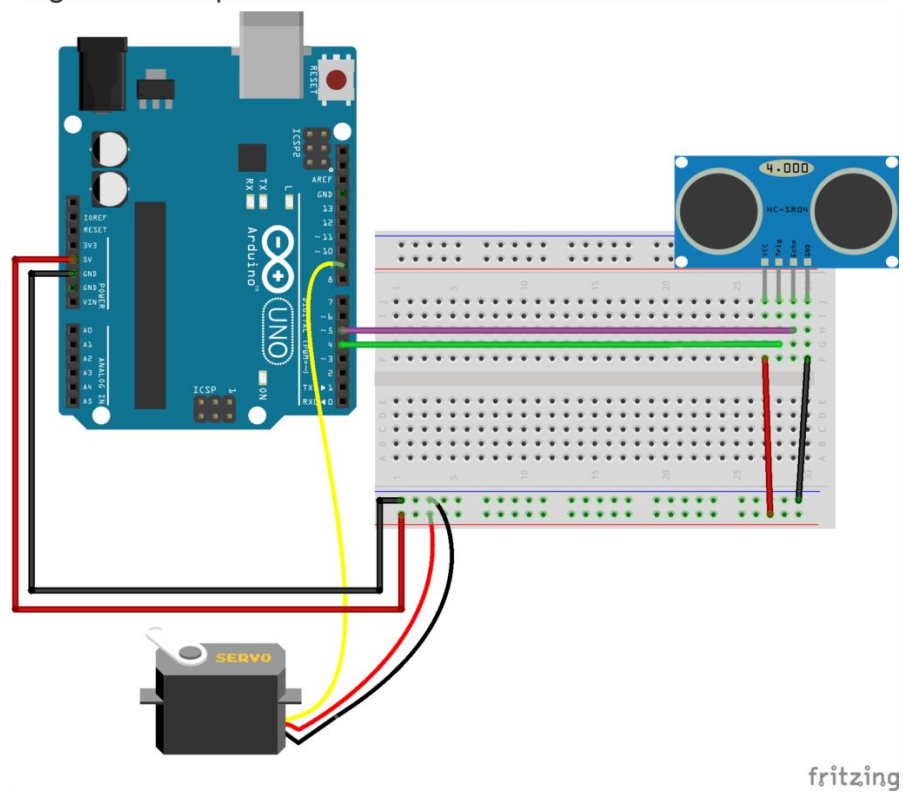
<https://drive.google.com/file/d/1Lo81ZL2LKw8btB0awPzHBRtgF1VM2FDq/view?usp=sharing>

Figura 1 - Circuito prático montado



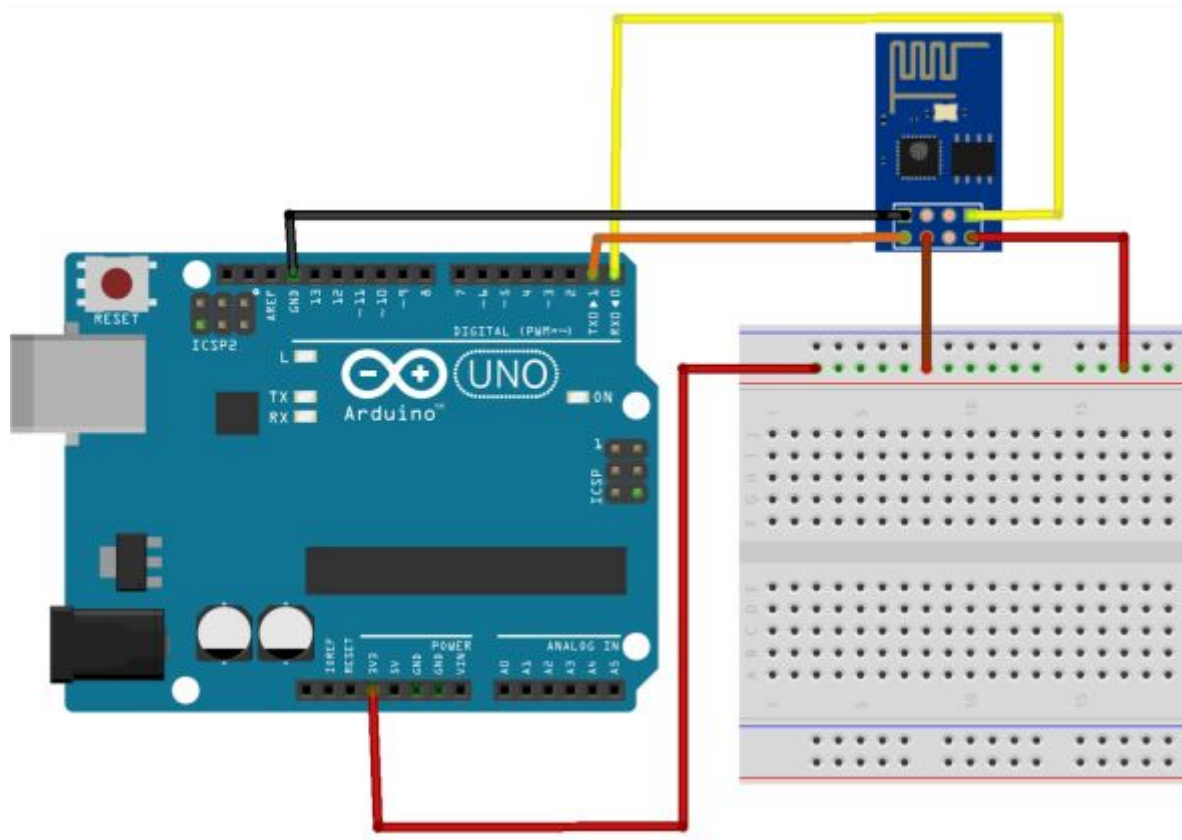
Fonte: Imagem Autoral

Figura 2 - Esquemático do circuito com sensor e servo motor



Fonte: site filipeflop

Figura 3 - Esquemático do circuito com ESP8266



Fonte: Documentação do curso Code IoT

Programa desenvolvido na IDE do Arduino

```
//Programa: Cancela automática
/*Baseado na documentação do curso Code IoT e
no site https://www.filipeflop.com/blog/como-juntar-dois-projetos-em-arduino/
*/

/*
 * Configura um servidor na ESP-01S; a ideia é, pelo navegador, permitir:
 * - controlar um atuador (liga ou desliga LED da Arduino);
 * - ler um sensor (abre uma página com valor de leitura de conversão AD).
 */

//Carrega as bibliotecas do sensor ultrassonico, wifi e do servo
#include <Ultrasonic.h>
#include <Servo.h>
#include <SoftwareSerial.h>
#include <stdlib.h>

// Para o sensor ultrassonico
//Define os pinos para o trigger e echo
#define pino_trigger 22
#define pino_echo 24

#define TIMEOUT 5000 // mS
#define LED 13 //verificação pelo led do arduino
//Inicializa o sensor nos pinos definidos acima
Ultrasonic ultrasonic(pino_trigger, pino_echo);

// Para o servo motor
Servo meuServo; // Declara o servo motor

//para comunicação com ESP 8266
SoftwareSerial mySerial(50, 52); // RX, TX
const int button = 8;

int test = 0; //POINT para saber se a cancela está automática

void setup()
{
  //serial para verificar a comunicação
  Serial.begin(9600);
```

```

pinMode(LED,OUTPUT);
mySerial.begin(9600);
SendCommand("AT+RST", "Ready");
delay(5000);

SendCommand("AT+CWMODE=1","OK");
SendCommand("AT+CIFSR", "OK");
SendCommand("AT+CIPMUX=1","OK");
SendCommand("AT+CIPSERVER=1,80","OK");
}

void loop()
{
  //iniciando comunicação com wifi
  String IncomingString="";
  boolean StringReady = false;

  //Le as informacoes do sensor ultrassonico em cm
  float cmMsec;
  long microsec = ultrasonic.timing();
  cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
  meuServo.attach(9); // Liga o servo motor

  while (mySerial.available()){
    IncomingString=mySerial.readString();
    StringReady= true;
  }

  if (StringReady){
    Serial.println("Received String: " + IncomingString);
    //verifica se o caminho é /LED=ON para ligar led
    if (IncomingString.indexOf("LED=ON") != -1 ) {
      digitalWrite(LED,HIGH);
      mySerial.println("AT+CIPSEND=0,18");
      delay(100);
      mySerial.println("<h1>LED Aceso</h1>");
      delay(1000);
      SendCommand("AT+CIPCLOSE=0","OK");
      test=1;
    }
    //verifica se o caminho é /LED=OFF para desligar led
    if (IncomingString.indexOf("LED=OFF") != -1) {
      digitalWrite(LED,LOW);
      mySerial.println("AT+CIPSEND=0,20");
    }
  }
}

```

```

    delay(100);
    mySerial.println("<h1>LED Apagado</h1>");
    delay(1000);
    SendCommand("AT+CIPCLOSE=0","OK");
    test=0;
}
} // fecha if ready

// Você pode alterar esse valor de acordo com o funcionamento desejado
if(cmMsec < 5 && test==1) // Verifica se o valor do sensor é menor que 5
{
    // Se o valor for
    meuServo.write(90); // Posiciona o motor em 90
    delay(1000); // Aguarda 1 segundo
    Serial.println("Abriu"); // Envia mensagem pela serial
}
else
{
    // Senão
    meuServo.write(0); // Posiciona o motor em 0
    delay(100); // Aguarda 0,1 segundo
    Serial.println("Fechou"); // Envia mensagem pela serial
}

meuServo.detach();// Desliga o servo motor

// Colocamos a distancia na serial para ajustarmos o sensor
Serial.print("Distancia em cm: ");
Serial.println(cmMsec);
}

//funções para comandos do esp com AT
boolean SendCommand(String cmd, String ack){
    mySerial.println(cmd); // Send "AT+" command to module
    if (!echoFind(ack)) // timed out waiting for ack string
        return true; // ack blank or ack found
}

//funcao do sensor de presenca
boolean echoFind(String keyword){
    byte current_char = 0;
    byte keyword_length = keyword.length();
    long deadline = millis() + TIMEOUT;
    while(millis() < deadline){

```

```
if (mySerial.available()){
  char ch = mySerial.read();
  Serial.write(ch);
  if (ch == keyword[current_char])
    if (++current_char == keyword_length){
      Serial.println();
      return true;
    }
  }
}
return false; // Timed out
}
```