

Objetos Inteligentes Conectados 1 sem. 2018

Turma 5K12

Projeto: "Semáforo com prevenção de acidentes"

Integrantes do grupo:

- Rafael Kacuta
- Felipe Macena Galhardo

Terceira Entrega:

Descrição da proposta: Dois semáforos (simulando um cruzamento) com um sensor de distância ligado em um semáforo, quando algo se aproximar do semáforo um LED branco acende no semáforo oposto. O usuário poderá acionar o semáforo após digitar "Start" no aplicativo de celular Blynk.

Codificação:

```
#include <Ultrasonic.h>
#include <Blynk.h>
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(2, 3); // RX, TX
#include <BlynkSimpleStream.h>
#define BLYNK_PRINT SwSerial
char auth[] = "a0a4668aecc14e9cbfee1cfebab7e003";
WidgetTerminal terminal(V1);
//Define os pinos para o trigger e echo
#define pino_trigger 2
#define pino_echo 3
//Inicializa o sensor nos pinos definidos acima
Ultrasonic ultrasonic(pino_trigger, pino_echo);

void setup()
{
  pinMode(13, OUTPUT); //red 1
  pinMode(12, OUTPUT); //yellow 1
  pinMode(11, OUTPUT); //green 1
  pinMode(10, OUTPUT); //red 2
  pinMode(8, OUTPUT); //yellow 2
  pinMode(7, OUTPUT); //green 2
  pinMode(6, OUTPUT); //branca
  SwSerial.begin(9600);
  Serial.begin(9600);
  Blynk.begin(Serial, auth);
  terminal.println(F("Blynk v" BLYNK_VERSION ": Device started"));
```

```

terminal.println(F("-----"));
terminal.println(F("Digite 'Start' para ligar o semáforo "));
terminal.println(F("Qualquer outra coisa e a mensagem será mostrada replicada"));
terminal.flush();
}

```

```

void loop()
{
  Blynk.run();
}

```

```

BLYNK_WRITE(V1){
  if (String("Start") == param.asStr()) {
    terminal.println("You said: 'Start'");
    terminal.println("I said: 'Iniciando...'");
    int op;
    op = 1;
    while (op=1){
      int cont=0;
      //Le as informacoes do sensor, em cm e pol
      float cmMsec, inMsec;
      long microsec;
      //inMsec = ultrasonic.convert(microsec, Ultrasonic::IN);
      digitalWrite(6,LOW);
      digitalWrite(10, LOW);
      digitalWrite(12,LOW);
      digitalWrite(13, HIGH);
      digitalWrite(7, HIGH);
      while (cont<3){
        //Exibe informacoes no serial monitor
        microsec = ultrasonic.timing();
        cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
        Serial.print("Distancia em cm: ");
        Serial.println(cmMsec);
        if (cmMsec<=10){
          digitalWrite(6,HIGH);
          delay(6000);
          //digitalWrite(7, LOW);
          //digitalWrite(8, HIGH);
          break;
        }else{
          cont+=1;
        }
        delay(1000);
      }
    }
  }
}

```

```

cont= 0;
delay(1000);
digitalWrite(6,LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
delay(1000);
while (cont <2){
microsec = ultrasonic.timing();
cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
Serial.print("Distancia em cm fase 2: ");
Serial.println(cmMsec);
if (cmMsec<=10){
digitalWrite(6,HIGH);
delay(1000);
digitalWrite(8, LOW);
digitalWrite(10, HIGH);
delay(2000);
break;
}else{
cont+=1;
}
delay(1000);
}
digitalWrite(6,LOW);
microsec = ultrasonic.timing();
cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
if (cmMsec<=10){
digitalWrite(6,HIGH);
Serial.print("Distancia em cm fase 3: ");
Serial.println(cmMsec);
}
digitalWrite(8, LOW);
digitalWrite(13, LOW);
digitalWrite(11, HIGH);
digitalWrite(10, HIGH);
delay(3000);
digitalWrite(6,LOW);
microsec = ultrasonic.timing();
cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
if (cmMsec<=10){
digitalWrite(6,HIGH);
Serial.print("Distancia em cm fase 4: ");
Serial.println(cmMsec);
}
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
delay(2000);

```

```
    }  
  }else{  
    // Send it back  
    terminal.print("Você disse:");  
    terminal.write(param.getBuffer(), param.getLength());  
    terminal.println();  
  }  
  
  // Ensure everything is sent  
  terminal.flush();  
}
```