

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, cada vez que a luz branca ativa, o terminal exibe a distância em CM, após o término do ciclo do semáforo será exibido no terminal a quantidade de carros detectadas.

Codificação:

```
#include
<Ultrasonic.h>

#include <Blynk.h>
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(0, 1); // 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...') ;
    terminal.flush();
    int op;
    op = 1;
    while (op=1){
      int cont=0;
      int i;
      int carro=0;
      //Le as informacoes do sensor, em cm e pol
      float cmMsec, inMsec;
      long microsec;
      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);
        if (cmMsec<=10){
          //Estende o sinal vermelho caso o sensor detecta algo
          digitalWrite(6,HIGH);
          carro+=1;
          terminal.print("Distancia em cm: ") ;
          terminal.println(cmMsec);

```

```

terminal.flush();
delay(2000);
for (i=0;i<=5;i++){
    digitalWrite(6,LOW);
    microsec = ultrasonic.timing();
    cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
    if (cmMsec<=10){
        digitalWrite(6,HIGH);
        carro+=1;
        terminal.print("Distancia em cm: ") ;
        terminal.println(cmMsec);
        terminal.flush();
        delay(1000);
    }
}
break;
}else{
cont+=1;
}
delay(1000);
}
cont= 0;
delay(1000);
digitalWrite(6,LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
delay(1000);
while (cont <2){
    //Sensor detecta algo e força mudança de sinal
    microsec = ultrasonic.timing();
    cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
    if (cmMsec<=10){
        digitalWrite(8, LOW);
        digitalWrite(10, HIGH);
        digitalWrite(6,HIGH);
        carro+=1;
        terminal.print("Distancia em cm: ") ;
        terminal.println(cmMsec);
        terminal.flush();
        delay(1000);
        digitalWrite(6,LOW);
        microsec = ultrasonic.timing();
        cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
        if (cmMsec<=10){
            digitalWrite(6,HIGH);
            carro+=1;
            terminal.print("Distancia em cm: ") ;
            terminal.println(cmMsec);
            terminal.flush();
            delay(1000);

```

```

    }
    break;
}else{
cont+=1;
}
delay(1000);
}
digitalWrite(6,LOW);
microsec = ultrasonic.timing();
cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
if (cmMsec<=10){
    digitalWrite(6,HIGH);
    carro+=1;
    terminal.print("Distancia em cm: ") ;
    terminal.println(cmMsec);
    terminal.flush();
    delay(1000);
}
digitalWrite(8, LOW);
digitalWrite(13, LOW);
digitalWrite(11, HIGH);
digitalWrite(10, HIGH);
delay(1000);
for (i=0;i<=2;i++){
    microsec = ultrasonic.timing();
    cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
    if (cmMsec<=10){
        digitalWrite(6,HIGH);
        carro+=1;
        terminal.print("Distancia em cm: ") ;
        terminal.println(cmMsec);
        terminal.flush();
        delay(1000);;
    }
}
digitalWrite(6,LOW);
microsec = ultrasonic.timing();
cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
if (cmMsec<=10){
    digitalWrite(6,HIGH);
    carro+=1;
    terminal.print("Distancia em cm: ") ;
    terminal.println(cmMsec);
    terminal.flush();
    delay(1000);
}
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
delay(1000);
microsec = ultrasonic.timing();

```

```

        cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
        if (cmMsec<=10){
            digitalWrite(6,HIGH);
            carro+=1;
            terminal.print("Distancia em cm: ") ;
            terminal.println(cmMsec);
            terminal.flush();
            delay(1000);
            digitalWrite(6,LOW);
        }
        terminal.print("Quantidade de carros: ");
        terminal.println(carro);
        terminal.flush();
    }
}

// Send it back
terminal.print("Você disse:");
terminal.write(param.getBuffer(), param.getLength());
terminal.println();
}

// Ensure everything is sent
terminal.flush();
}

```