**PONTICÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

RICARDO TAVERNA

VINICIOS VALERIO WEBER

YGOR STENGRAT ALVES

**RELATÓRIO TÉCNICO PROJETO SINALEIRO BLUETOOTH**

Documentação Técnica do Projeto de Sinaleiros Controlados por Arduino e Uma Aplicação Mobile

Curitiba/PR

2019

**RELATÓRIO TÉCNICO PROJETO SINALEIRO BLUETOOTH**

Documentação Técnica do Projeto de Sinaleiros Controlados por Arduino e Uma Aplicação Mobile

Relatório técnico do projeto dos sinaleiros controlados por Arduino em uma aplicação mobile entregue a Pontifícia Universidade Católica do Paraná

Curitiba, 28 de Março de 2019.

Sumário

[1- INTRODUÇÃO 4](#_Toc4700756)

[2- CÁLCULOS 4](#_Toc4700757)

[3- DIAGRAMA 5](#_Toc4700758)

[4- CÓDIGO 5](#_Toc4700759)

[5- Fotos 8](#_Toc4700760)

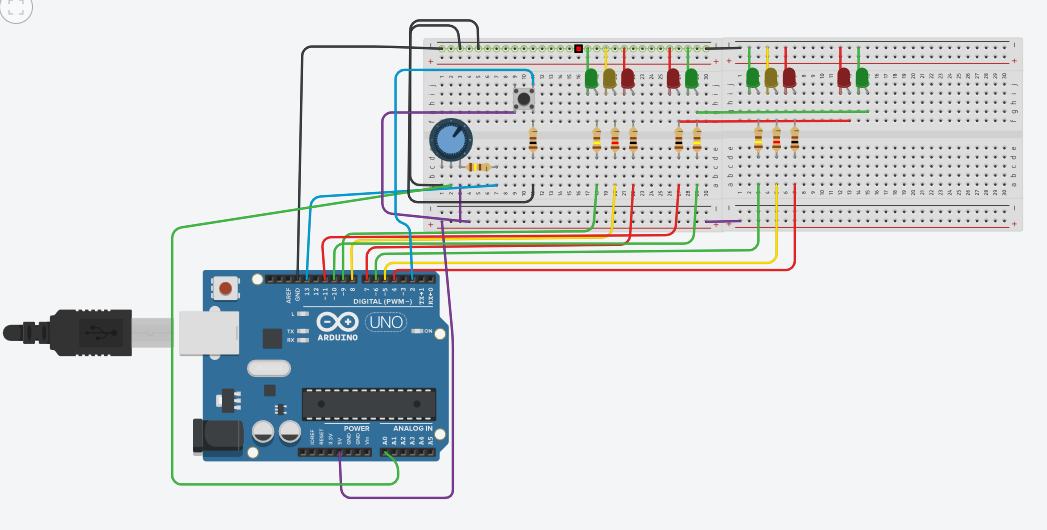
# INTRODUÇÃO

O problema apresentado refere-se à construção de uma representação de duas vias compostas por um par de sinaleiros para veículos e um par de sinaleiros para pedestres simulando o funcionamento de duas vias, sendo uma principal, em que o fluxo de veículos é emulado por um potenciômetro e todos os componentes ligados e controlados por um Arduino Uno.

O fluxo de veículos é passado no problema como sendo de 1024 veículos/hr na via principal, e na via secundária sendo de 1/3 desse valor. Para quando o fluxo de veículos for o mesmo em ambas as vias, o tempo de permanência dos semáforos deve ser o mesmo. O tempo para a via principal de ir aumento com fluxo até 2/3 do tempo total do ciclo e não sendo menor do que ¼ na via secundária.

O controle de trafego e os acionamentos dos botões são realizados por uma aplicação mobile construída no MIT Inventor para melhor utilização pelo usuário.

# IMAGENSS



# CÓDIGOS

|  |
| --- |
| unsigned int potenciometro;  float pwm;  unsigned int tempoPadrao = 2000;  unsigned int tempoAgregado = 10000;  float tempoS;  float tempoP;  unsigned int tempoPedestre = 15000;  unsigned int chamaSemafP;  void semaforoPedestre(){    if(chamaSemafP == 1){  digitalWrite(10, HIGH);  digitalWrite(11, LOW);  //semaforo principal em vermelho  digitalWrite(4, HIGH);  digitalWrite(5, LOW);  digitalWrite(6, LOW);  //semaforo secundario em verde  digitalWrite(7, HIGH);  digitalWrite(8, LOW);  digitalWrite(9, LOW);  delay(tempoPedestre);  for(int i = 0; i < 5; i++){  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoPadrao/2);  digitalWrite(10, LOW);  digitalWrite(11, LOW);  delay(tempoPadrao/2);  }  }  chamaSemafP = 0;  }  void setup(){  //pinMode(2, OUTPUT);//interrupção 0  attachInterrupt(0, chave, RISING);  //pinMode(3, OUTPUT);//interrupção 1  pinMode(4, OUTPUT);//semaforo primario vermelho  pinMode(5, OUTPUT);//semaforo primario amarelo  pinMode(6, OUTPUT);//semaforo primario verde  pinMode(7, OUTPUT);//semaforo secundario vermelho  pinMode(8, OUTPUT);//semaforo secundario amarelo  pinMode(9, OUTPUT);//semaforo secundario verde  pinMode(10, OUTPUT);//semaforo pedestre verde  pinMode(11, OUTPUT);//semaforo pedestre verde  pinMode(13, OUTPUT);//Regulador de fluxo  Serial.begin(9600);    }  void chave(){  chamaSemafP = 1;  }  void loop()  {  if(chamaSemafP == 0){  potenciometro = analogRead(A0);  if(potenciometro <= 341){  Serial.println(potenciometro);  pwm = (float)map(potenciometro, 0, 341, tempoAgregado/4 , tempoAgregado/2)/10000;//Controle de tempo com o regulador de fluxo  }else{  Serial.println(potenciometro);  pwm = (float)map(potenciometro, 342, 1023, tempoAgregado/2, (2\*tempoAgregado)/3)/10000;  }      tempoP = tempoAgregado\*pwm;  tempoS = tempoAgregado - tempoP;  Serial.println(pwm);  Serial.println(tempoP);  Serial.println(tempoS);    //semaforo principal em vermelho  digitalWrite(4, HIGH);  digitalWrite(5, LOW);  digitalWrite(6, LOW);  //semaforo secundario em verde  digitalWrite(7, LOW);  digitalWrite(8, LOW);  digitalWrite(9, HIGH);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoS);  //Amarelo ligado  digitalWrite(4, HIGH);  digitalWrite(5, LOW);  digitalWrite(6, LOW);  digitalWrite(7, LOW);  digitalWrite(8, HIGH);  digitalWrite(9, LOW);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoPadrao);  //Tempo de segurança em vermelho  digitalWrite(4, HIGH);  digitalWrite(5, LOW);  digitalWrite(6, LOW);  digitalWrite(7, HIGH);  digitalWrite(8, LOW);  digitalWrite(9, LOW);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoPadrao);  //semaforo Via Principal em verde usando o fluxo como delay  digitalWrite(4, LOW);  digitalWrite(5, LOW);  digitalWrite(6, HIGH);  //semaforo secundario em vermelho  digitalWrite(7, HIGH);  digitalWrite(8, LOW);  digitalWrite(9, LOW);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoP);  //Amarelo ligado  digitalWrite(4, LOW);  digitalWrite(5, HIGH);  digitalWrite(6, LOW);  digitalWrite(7, HIGH);  digitalWrite(8, LOW);  digitalWrite(9, LOW);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoPadrao);  //Tempo de segurança em vermelho  digitalWrite(4, HIGH);  digitalWrite(5, LOW);  digitalWrite(6, LOW);  digitalWrite(7, HIGH);  digitalWrite(8, LOW);  digitalWrite(9, LOW);  digitalWrite(10, LOW);  digitalWrite(11, HIGH);  delay(tempoPadrao);      semaforoPedestre();    }  } |

# Fotos