universidade de aveiro

Ano letivo 2023/2024

Controlo de semáforo com detetor de velocidade

Microcontroladores e Interação com Sensores e Atuadores

Grupo 2 (TP7-2): David Pelicano(113391); Gustavo Martins(114304); João Ferreira(113571); Salvador Caldeira(113390); Sebastião Filipe(113498); Tomás Frade(113507)

17/04/2024

Docente : Rui Martins

1. Objetivos:

Este trabalho foi desenvolvido no âmbito do módulo de Competências Transferíveis II, Microcontroladores e Interação com Sensores e Atuadores. Nosso objetivo era construir um projeto que incorporasse pelo menos um sensor e um atuador, utilizando o microcontrolador STM32F41XX para criar um programa que interagisse entre esses dispositivos. Este relatório tem como objetivo descrever o propósito deste trabalho, seu desenvolvimento e os resultados alcançados.

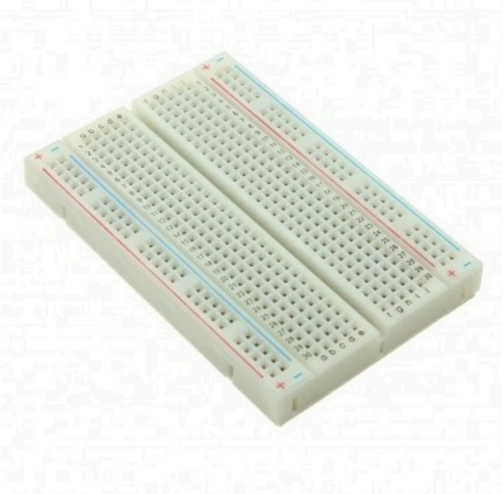
Para este projeto, o objetivo principal foi simular o controle de um semáforo, garantindo a segurança dos peões ao atravessar a rua e controlando o tráfego de veículos para evitar excesso de velocidade.

2. Introdução Teórica:

Neste projeto, propomos a simulação de um semáforo para veículos, que fica vermelho quando um veículo ultrapassa uma determinada velocidade. Este semáforo está interligado a outro destinado aos peões, ativado por um botão. Ao pressionar este botão, o semáforo dos veículos muda para vermelho, enquanto o dos peões exibe sinal verde. Após um intervalo de tempo predefinido, os sinais voltam à sua configuração normal. Este projeto inclui a implementação de um semáforo com botão para peões e a integração de um sensor que calcula a velocidade dos veículos em movimento.

3. Materiais

3.1 Placa de Ensaio:



A **placa de ensaio** foi fundamental para este trabalho, fornecendo uma plataforma organizada e flexível para prototipagem e testes. Ela permite a conexão e desconexão fácil dos componentes eletrónicos, facilitando ajustes e iterações durante o desenvolvimento. Além disso, simplifica as conexões elétricas entre o microcontrolador STM32F411, os LEDs, o botão para o peão e os sensores, tornando mais fácil a organização dos componentes e a depuração de problemas.

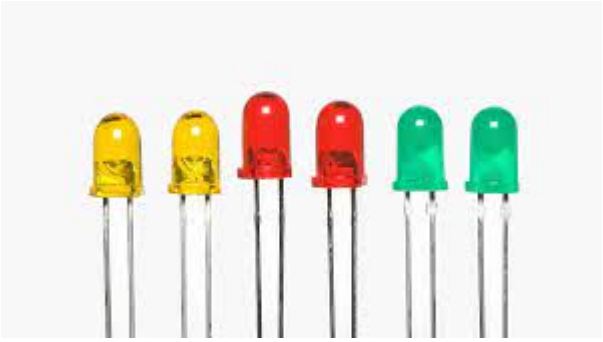
3.2 Sensores:

- **Sensor ultrassónico**: Este sensor emite pulsos de ultrassom e mede o tempo que leva para esses pulsos serem refletidos de volta por um objeto, permitindo calcular a distância até o objeto.

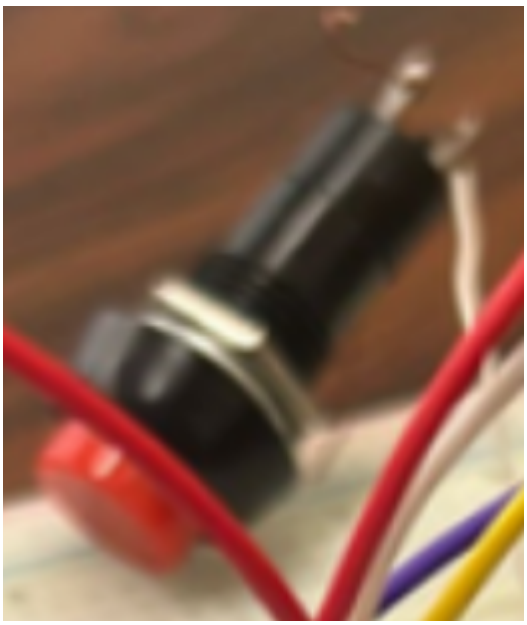
- **Sensor de infravermelhos**: Este sensor deteta a radiação infravermelha emitida por objetos e pode ser usado para deteção de movimento, temperatura, entre outros.

3.3 Atuadores:

- **5 LEDs**: Os LEDs emitem luz quando uma corrente elétrica passa por eles, sendo controlados para indicar informações visuais, como o estado dos semáforos. Foram usados dois LEDs vermelhos, dois verdes e um amarelo para sinalizar o estado dos semáforos.

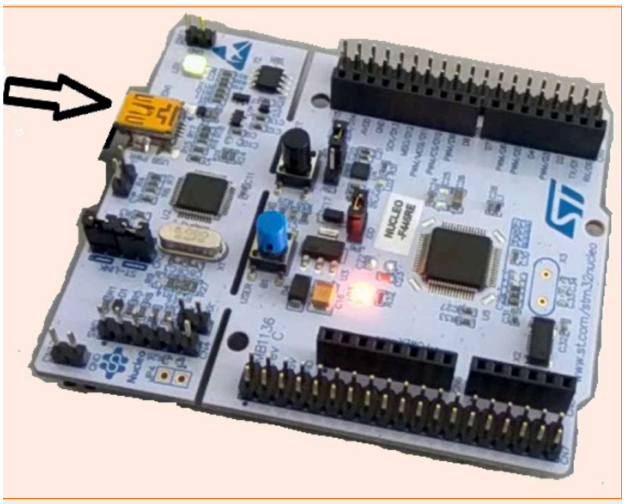


- **Botão**: Este atuador passivo fecha um circuito elétrico quando pressionado, sendo usado para simular o botão dos peões, permitindo que eles atravessem a passadeira quando pressionado.



3.4 Microcontrolador:

O microcontrolador **STM32F411** foi utilizado neste projeto. Ele é capaz de processar dados e enviar comandos para outros dispositivos, sendo fundamental para o controle dos semáforos e a interação com os sensores e atuadores.



3.5 Maquete:

Decidimos criar uma maquete simples para este projeto, servindo como base visual para uma melhor compreensão do projeto em si.

Figura 1: Maquete

4. Execução do projeto

Como se observa na figura 2, é um fluxograma, onde demonstra o funcionamento do controlo do semáforo, onde começa por verificar se o botão do peão foi pressionado. Se for pressionado, os semáforos para os carros mudam para amarelo e depois para vermelho, enquanto que o semáforo para os peões muda para verde. Se o botão do peão não for pressionado, o fluxograma passa a verificar o excesso de velocidade. Esta verificação é baseada na distância. Se a distância for inferior a 50cm, considera-se que há excesso de velocidade. Nesse caso, os semáforos para os carros mudam para amarelo e depois para vermelho. Se não houver excesso de velocidade, os semáforos para os carros permanecem verdes e o semáforo para os peões muda para vermelho.

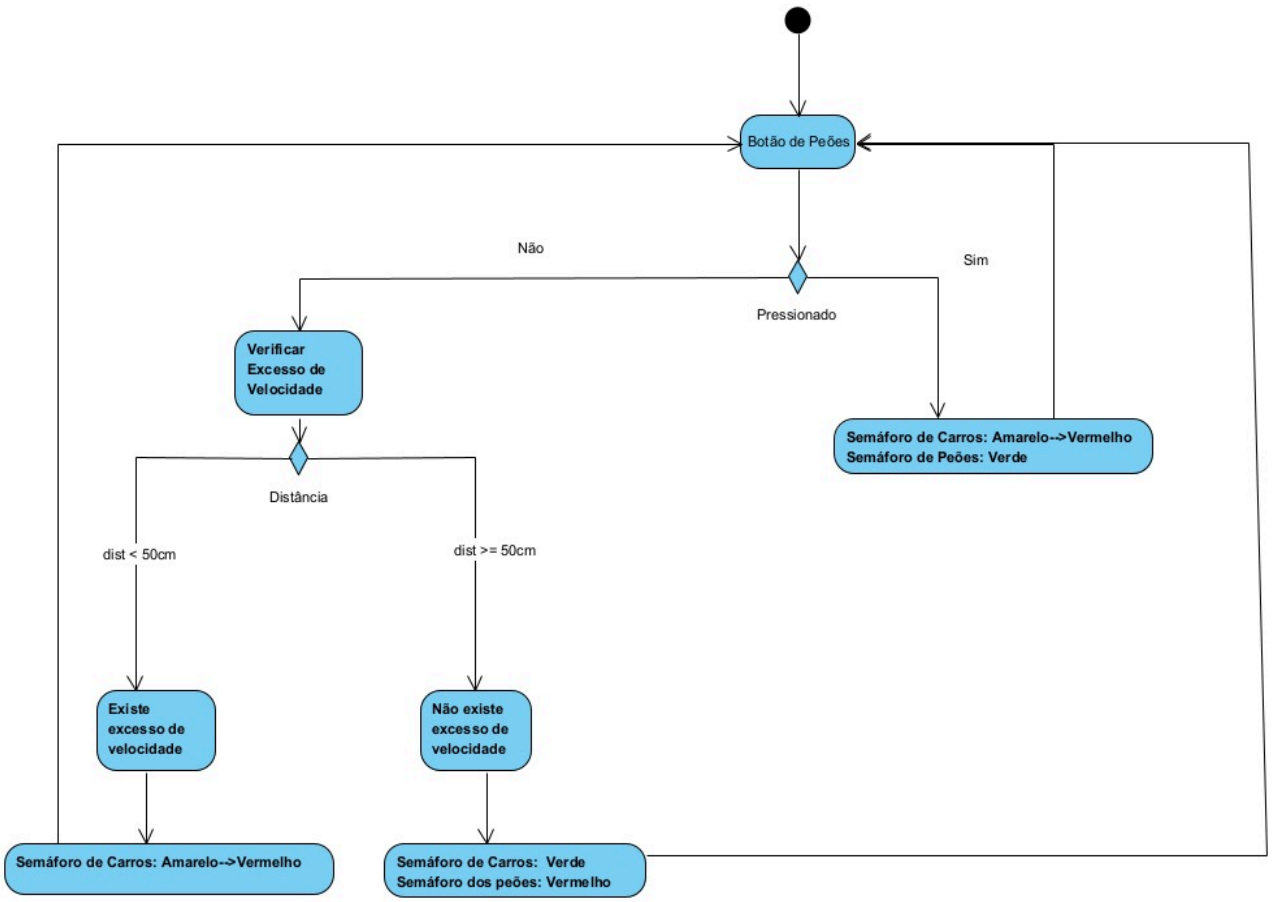


Figura 2: Fluxograma do código do controlo de semáforo

Este código implementa um sistema de semáforos inteligente utilizando o material indicado anteriormente. Este sistema controla tanto o tráfego de carros como o de peões, adaptando-se dinamicamente às condições do ambiente.

Quando o botão é pressionado, o ciclo do semáforo é iniciado. Durante esse ciclo, as luzes dos semáforos para carros e peões são alternadas de acordo com um padrão predefinido. Após um período de tempo, o semáforo para os peões é ativado, permitindo-lhes atravessar a rua com segurança.

Enquanto os carros estão em movimento, o sistema monitoriza a presença de objetos utilizando um sensor infravermelho e mede a distância até esses objetos com um sensor ultrassónico. Se um objeto for detetado e estiver a mover-se a alta velocidade, o semáforo dos carros é alterado para amarelo, alertando os condutores sobre a situação.

Essencialmente, o código integra diferentes componentes e sensores para criar um sistema de semáforos adaptativo, capaz de responder dinamicamente às condições do tráfego e garantir a segurança dos peões e condutores.

Código final do Projeto:

```
#include <Arduino.h>
#include <Ultrasonic.h>

#define BOTAO_PIN                       PA0
#define SEMAFORO_CARROS_VERDE_PIN       PB0
#define SEMAFORO_CARROS_AMARELO_PIN     PB1
#define SEMAFORO_CARROS_VERMELHO_PIN    PB2
#define SEMAFORO_PEDESTRES_VERDE_PIN    PB3
#define SEMAFORO_PEDESTRES_VERMELHO_PIN PB4
#define SENSOR_INFRARED_PIN             PA1
#define TRIGGER_PIN                     PB13
#define ECHO_PIN                        PB14
#define MAX_DISTANCE                    20
#define DISTANCIA_ENTRE_SENSORES        20
Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN);
const unsigned long READ_INTERVAL = 100;

unsigned long startTime = 0;
unsigned long tempoDecorrido = 0;
unsigned long velocidadeMedia = 0;
unsigned long Time;
unsigned long diff;
```

Figura 3: Configuração dos pinos e inicialização de variáveis

Como se observa na figura 4, este código é a configuração inicial de um sistema de semáforos para carros e peões, com um sensor infravermelho. Define-se a comunicação serial, configura-se os pinos dos LEDs dos semáforos e do sensor como entradas ou saídas, e estabelece-se o estado inicial dos semáforos: verde para carros e vermelho para peões

```
void setup() {
  Serial.begin(9600);
    pinMode(BOTAO_PIN, INPUT_PULLUP);
    pinMode(SEMAFORO_CARROS_VERDE_PIN, OUTPUT);
    pinMode(SEMAFORO_CARROS_AMARELO_PIN, OUTPUT);
    pinMode(SEMAFORO_CARROS_VERMELHO_PIN, OUTPUT);
    pinMode(SEMAFORO_PEDESTRES_VERDE_PIN, OUTPUT);
    pinMode(SEMAFORO_PEDESTRES_VERMELHO_PIN, OUTPUT);
    pinMode(SENSOR_INFRARED_PIN, INPUT);

    // Inicialmente, configuram-se os semáforos de carros e peões para um estado padrão
    digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
    digitalWrite(SEMAFORO_CARROS_VERDE_PIN, HIGH);
    digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);
    digitalWrite(SEMAFORO_PEDESTRES_VERMELHO_PIN, HIGH);
    digitalWrite(SEMAFORO_PEDESTRES_VERDE_PIN, LOW);
}
```

Figura 4: Configuração de um sistema de semáforos com um sensor infravermelho

Muito resumidamente, para não repetir-se o que foi dito anteriormente no relatório, como se observa na figura 5, esta parte do código,(void loop()), é a função principal de um sistema de semáforos com sensores infravermelhos e ultrassónicos. Quando um botão é pressionado, o semáforo para carros muda de verde para vermelho e o semáforo para peões muda de vermelho para verde, permitindo que os peões atravessem. Se um objeto é detectado pelo sensor infravermelho, o tempo é registado e quando o objeto passa pelo sensor ultrassónico, a velocidade média é calculada. Se a velocidade for superior a 5 m/s, o semáforo para carros muda para amarelo. Caso contrário, permanece verde.

```ino
void loop() {
    if (digitalRead(BOTAO_PIN) == LOW && estado==0) {
        // Se o botão foi pressionado, faça o ciclo do semáforo
        // Desliga o semáforo de carros
        digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
        digitalWrite(SEMAFORO_CARROS_VERDE_PIN, LOW);
        digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, HIGH);
        delay(3000);
        digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, HIGH);
        digitalWrite(SEMAFORO_CARROS_VERDE_PIN, LOW);
        digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);
        delay(2500);
        // Liga o semáforo de pedestres
        digitalWrite(SEMAFORO_PEDESTRES_VERMELHO_PIN, LOW);
        digitalWrite(SEMAFORO_PEDESTRES_VERDE_PIN, HIGH);
        // Delay de 4 segundos para os pedestres atravessarem
        delay(7000);
        for (int i = 0; i <= 5; i++) {
            digitalWrite(SEMAFORO_PEDESTRES_VERDE_PIN, HIGH); // Liga o LED
            delay(400); // Aguarda 500 milissegundos (0,5 segundo)
            digitalWrite(SEMAFORO_PEDESTRES_VERDE_PIN, LOW); // Desliga o LED
            delay(400); // Aguarda 500 milissegundos (0,5 segundo)
        }
        digitalWrite(SEMAFORO_PEDESTRES_VERDE_PIN, LOW);
        digitalWrite(SEMAFORO_PEDESTRES_VERMELHO_PIN, HIGH);
        delay(1500);

        digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
        digitalWrite(SEMAFORO_CARROS_VERDE_PIN, HIGH);
        digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);
        Serial.println("Fim botao");


    } else {
        // Verifica se o sensor infravermelho detectou um objet


        if (digitalRead(SENSOR_INFRARED_PIN) == LOW) {
            // Se sim, inicia o cronômetro
            estado=2;
            Time = millis();
            Serial.println("Sensor infravermelho detectou um objeto.");
        }

delay(10);
int distancia=ultrasonic.distanceRead(CM);
Serial.println(distancia);
Serial.println(estado);
        // Verifica se o sensor ultrassônico detectou o objeto a uma distância de até 20 cm
        if ((estado==2) && (distancia <= MAX_DISTANCE)) {
            // Se sim, calcula a velocidade média
            delay(10);
            unsigned long elapsedTime = millis() - Time; // Calcula o tempo decorrido
            long velocidadeMediaMs =1000* DISTANCIA_ENTRE_SENSORES / elapsedTime; // Calcula a velocidade média em m/s
            Serial.println(velocidadeMediaMs);

            estado=0;
             Serial.println("Fim");

            if (velocidadeMediaMs >= 5) {


                //Se a velocidade média for maior ou igual a 5 m/s, muda o semáforo dos carros para amarelo
                digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
                digitalWrite(SEMAFORO_CARROS_VERDE_PIN, LOW);
                digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, HIGH);
                delay(2000);
                digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, HIGH);
                digitalWrite(SEMAFORO_CARROS_VERDE_PIN, LOW);
                digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);
                delay(4000);
                digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
                digitalWrite(SEMAFORO_CARROS_VERDE_PIN, HIGH);
                digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);

            } else {
                // Caso contrário, mantém o semáforo dos carros verde
                digitalWrite(SEMAFORO_CARROS_VERMELHO_PIN, LOW);
                digitalWrite(SEMAFORO_CARROS_VERDE_PIN, HIGH);
                digitalWrite(SEMAFORO_CARROS_AMARELO_PIN, LOW);
            }
Share
```

Figura 5: Função principal do sistema

5. Resultado FInal

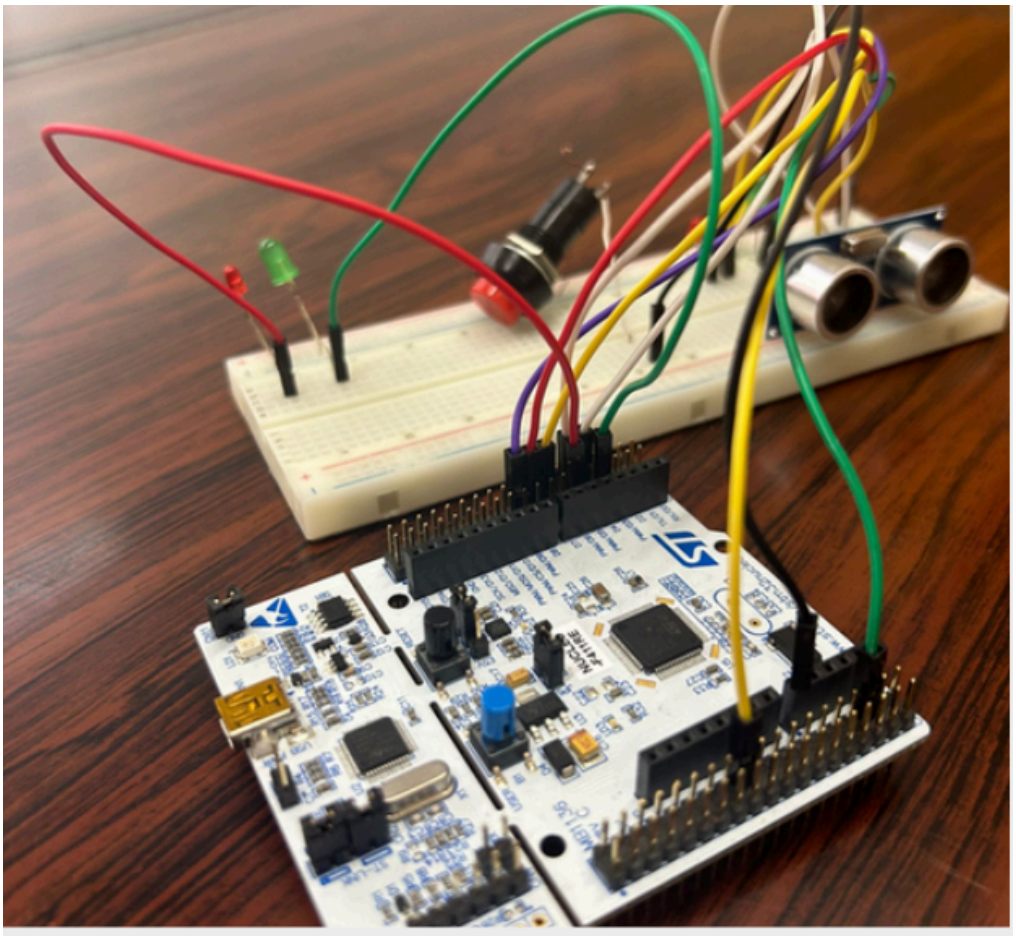


Figura 6: Projeto final

6. Conclusão

Em suma, este projeto atingiu com sucesso o seu objetivo principal de simular o controlo de um semáforo, assegurando a segurança dos peões e regulando o tráfego de veículos. A integração de sensores, atuadores e o microcontrolador STM32F411 possibilitou o desenvolvimento de um sistema eficaz e funcional. A maquete criada também proporcionou uma representação clara do projeto. Contudo, existem possibilidades de melhorias futuras, como a implementação de algoritmos mais avançados para deteção de velocidade e a inclusão de mais funcionalidades para aumentar a eficácia e segurança do sistema de semáforos.

7. Bibliografia

https://www.makerhero.com/blog/sensor-infravermelho-arduino/
https://www.makerhero.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/
Material disponibilizado no e-learning