Implementação de semáforo para carros e pedestres com LEDs e um botão utilizando multiprocessamento

Elias de Almeida Sombra Neto[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - Campus Maracanaú
Av. Parque Central, 1315 - Distrito Industrial I, Maracanaú-CE, Brasil

elias.almeida09@aluno.ifce.edu.br

Abstract. *This paper presents the implementation of a traffic light system for vehicles and pedestrians using the ESP32 microcontroller with multiprocessing support. The system uses LEDs for signaling and a button that allows pedestrians to request crossing. The button is constantly monitored on one core, while LED control is handled by another, ensuring fast responsiveness. The project was developed as part of a Microcontrollers course at IFCE, and the results showed the system worked effectively.*

Resumo. *Este trabalho apresenta a implementação de um sistema de semáforo para veículos e pedestres utilizando o microcontrolador ESP32 com suporte a multiprocessamento. O sistema inclui LEDs para sinalização e um botão que permite ao pedestre solicitar a travessia. A leitura constante do botão é realizada em um núcleo, enquanto o controle dos LEDs ocorre em outro, garantindo maior responsividade. A prática foi realizada no laboratório da disciplina de Microcontroladores do IFCE, e os resultados demonstraram o funcionamento eficaz do sistema proposto.*

1. Introdução

Este artigo descreve uma atividade prática realizada no Laboratório de Eletroeletrônica e Sistema Embarcados (LAESE) durante a disciplina de Microcontroladores no Instituto Federal do Ceará (IFCE).

O objetivo da atividade é simular o funcionamento de um semáforo de carrros e de pedestres. O botão serve para indicar que a pessoa gostaria de realizar a travessia. Portanto, seu acionamento deve priorizar o sinal verde para para pedestres e, consequentemente, ativar o sinal vermelho para carros.

Contudo, o botão deve ser lido constantemente pelo ESP32 sem grandes delays. Por isso, o código gravado no microcontrolador deve usar múltiplos núcleos de modo que um deles identifique o pressionamento do botão e o outro execute os semáforos normalmente.

2. Materiais utilizados

Os materiais para a construção do circuito incluem:

- 1 ESP32 30 pinos;
- 1 cabo micro USB;

- 1 protoboard;
- 2 LEDs vermelhos;
- 1 LED amarelo;
- 2 LEDs verdes;
- 1 botão;
- 3 resistores;
- 8 cabos jumper macho-fêmea;
- 2 cabo jumper macho-macho.

3. Montagem do circuito

Para realizar a montagem do circuito, é preciso seguir os seguintes passos:

- Conectar pino GND do ESP32 em um furo negativo da região de alimentação na protoboard através de um cabo jumper macho-fêmea.
- Posicionar o botão e os 5 LEDs na região central da protoboard. Os LEDs do semáforo de carros devem ficar em paralelo. Da mesma maneira, os LEDs do semáforo de pedestres devem ficar em paralelo.
- Conectar uma extremidade do jumper macho-fêmea em um GPIO do ESP32 e o outro na mesma coluna do pino positivo de cada LED.
- Conectar um resistor em paralelo com os LEDs do semáforo de carros e um resistor em paralelo com os LEDs do semáforo de pedestres. Uma extremidade do resistor deve ficar em um furo negativo da região de alimentação e a outra deve ficar na mesma coluna do pino negativo dos LEDs na região central da protoboard.
- Posicionar o Botão na região central da protoboard e ligá-lo na linha de alimentação negativa por meio de um jumper macho-macho.
- Conectar o resistor do botão em linhas distintas na região central da protoboard.
- Conectar uma extremidade do resistor do botão ao pino 3.3V do ESP32.
- A outra extremidade do resistor do botão deve ser conectada em um GPIO do ESP32. Conectar um pino do botão na mesma coluna dessa mesma extremidade por meio de um jumper macho-macho.

Além disso, é necessário reservar os seguintes pinos do ESP32:

- GPIO 13 para o LED vermelho de carros;
- GPIO 12 para o LED amarelo de carros;
- GPIO 14 para o LED verde de carros;
- GPIO 27 para o LED vermelho de pedestres;
- GPIO 26 para o LED verde de pedestres;
- GPIO 25 para o botão que aciona o circuito;

4. Implementação do código

O código responsável por simular os semáforos e acionar o sinal de pedestres por meio do botão foi desenvolvido com a linguagem C++ no editor de código Visual Studio Code juntamente com sua extensão PlatformIO.

```
1 #include <Arduino.h>
2
3 #define VERMELHO_CARRO 13
```

```
#define AMARELO_CARRO 12
#define VERDE_CARRO 14
#define VERMELHO_PEDESTRE 27
#define VERDE_PEDESTRE 26
#define BOTAO 25

bool solicitarTravessia = false;

void botaoTask(void *param);
void pedestreAtravessa();

void setup() {
  pinMode(VERMELHO_CARRO, OUTPUT);
  pinMode(AMARELO_CARRO, OUTPUT);
  pinMode(VERDE_CARRO, OUTPUT);
  pinMode(VERMELHO_PEDESTRE, OUTPUT);
  pinMode(VERDE_PEDESTRE, OUTPUT);
  pinMode(BOTAO, INPUT_PULLUP);
  xTaskCreate(botaoTask, "Botao Task", 1024, NULL, 1, NULL);
  Serial.begin(9600);
}

void botaoTask(void *param) {
  while (true) {
  vTaskDelay(5);
    if (digitalRead(BOTAO) == LOW) {
      solicitarTravessia = true;
    }
  }
}

void piscarLED(int cor) {
  for (int i = 0; i < 3; i++) {
    digitalWrite(cor, HIGH);
    delay(1000);
    digitalWrite(cor, LOW);
    delay(500);
  }
}

void pedestreAtravessa() {
  solicitarTravessia = false;
  digitalWrite(VERDE_CARRO, LOW);
  piscarLED(AMARELO_CARRO);
  digitalWrite(VERMELHO_CARRO, HIGH);
  digitalWrite(VERMELHO_PEDESTRE, LOW);
  digitalWrite(VERDE_PEDESTRE, HIGH);
  delay(6000);
  digitalWrite(VERDE_PEDESTRE, LOW);
```

```
  digitalWrite(VERMELHO_CARRO, LOW);
}

void loop() {
  solicitarTravessia = false;
  digitalWrite(VERDE_CARRO, HIGH);
  digitalWrite(VERMELHO_PEDESTRE, HIGH);
  for(int i = 1; i <= 400; i++){
    delay(10);
    if (solicitarTravessia == true) {
      break;
    }
  }
  if(solicitarTravessia == true){
    delay(2000);
    pedestreAtravessa();
  }
  else {
    digitalWrite(VERDE_CARRO, LOW);
    piscarLED(AMARELO_CARRO);
    digitalWrite(VERMELHO_CARRO, HIGH);
    digitalWrite(VERMELHO_PEDESTRE, LOW);
    digitalWrite(VERDE_PEDESTRE, HIGH);
    delay(3000);
    digitalWrite(VERMELHO_CARRO, LOW);
    digitalWrite(VERDE_PEDESTRE, LOW);
  }
}
```

5. Resultados

A montagem e implementação de código do botão para simular os semáforos acionar o sinal de pedestres foi um sucesso. A seguir, pode-se visualizar a forma do circuito no final da prática, assim como uma demonstração do seu funcionamento.

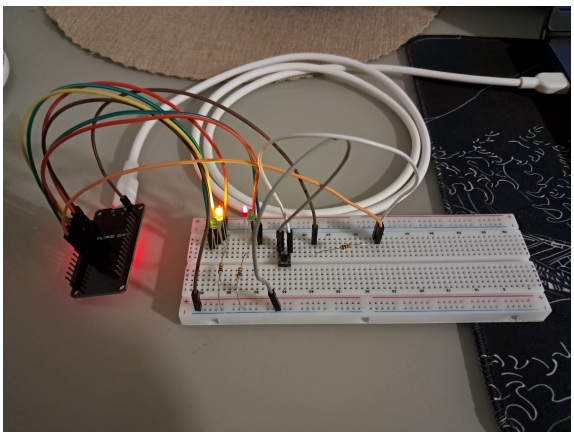

Figure 1. Circuito que simula o semáforo de carros e o de pedestres que pode ser acionado por um botão lido com multiprocessamento

O vídeo que demonstra o circuito de ambos semáforos em funcionamento está disponível em: Resultado da atividade prática 03

6. Conclusão

Essa atividade prática demonstrou como implementar um semáforo de carros e um semáforo de pedestres que pode ser acionado por meio de um botão. Ao pressionar, o microcontrolador imediatamente reconhece devido ao uso de múltiplos núcleos. A partir disso, o botão realiza a transição entre cores dos LEDs que simulam o semáforo, mantendo o sinal verde para pedestres aceso por um tempo maior. Para isso foi utilizado um microcontrolador ESP32, uma protoboard, alguns resistores e cabos jumper. Dessa forma, foi possível concluir a atividade e atingir o objetivo proposto.