

Semáforo com Led e display de 7 segmentos utilizando Esp32

José de Alencar de Sousa Júnior

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Avenida Treze de Maio, 2081 - Benfica - Fortaleza - Ceará - Brasil

²Departamento de Telemática (DETEL)

³Curso: Tecnólogo em Telemática

{SOUSA JÚNIOR, J. de A. de.}alencar.junior00@aluno.ifce.edu.br

Abstract. *The project proposes the simulation of an intersection with traffic lights with the possibility of pedestrians crossing by pressing a button, showing the time available for this crossing using a seven-segment display. The project includes timers for the signaling to change color. There is also an audible signal to help pedestrians cross. The microcontroller used was the ESP32, as it is low cost and easy to program.*

Resumo. *O projeto propõe a simulação de um cruzamento com semáforos com possibilidade de passagem para pedestre mediante acionamento de botão, exibindo o tempo disponível para essa passagem utilizando display de sete segmentos. O projeto apresenta temporizadores para a sinalização alterar a cor. Também há emissão de sinal sonoro para ajudar o pedestre na travessia. O microcontrolador utilizado foi o esp32, pois apresenta baixo custo e fácil programação.*

1. Introdução

A sinalização de trânsito tem um papel importantíssimo na segurança coletiva dos motoristas, passageiros e pedestres, pois permite que cada grupo tenha seu tempo certo de transitar sem que afete o próximo. Esse projeto tem como objetivo simular a sinalização de trânsito, utilizando programação em C, um microcontrolador esp32, um display de sete segmentos e um buzzer.

2. Fundamentação teórica

O Esp32 (Espressif Systems, 2024)[2] é um microcontrolador programável em várias linguagens, apresenta custo reduzido e eficiência. O display de 7 segmentos apresenta baixo custo e consegue mostrar informações alfanuméricas. O buzzer ativo também em baixo custo e é de fácil uso. Portanto, com a junção desses componentes, é possível simular uma passagem de trânsito com tempo para pedestres. O trabalho foi proposto pela disciplina de microcontroladores, afim de obter conhecimento sobre esquematização de circuitos (Silva, A., Correia E., Silva V., 2024)[1].

3. Materiais e métodos

Para realizar a demonstração do projeto, foram utilizados os materiais descritos a seguir.

3.1. Esp32

Esp32 é o microcontrolador com módulo Wi-Fi e bluetooth e que possui baixo consumo de energia. Esse equipamento é amplamente utilizado em projetos IoT (Datasheet Esp32, 2024)[3].

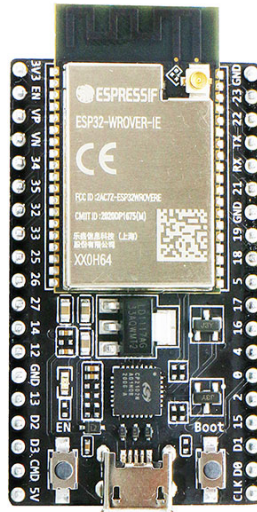


Figura 1. Esp32

3.2. Display de sete segmentos

É um display de baixo custo que mostra informações alfanuméricas. Foi utilizado o modelo do tipo anodo comum.

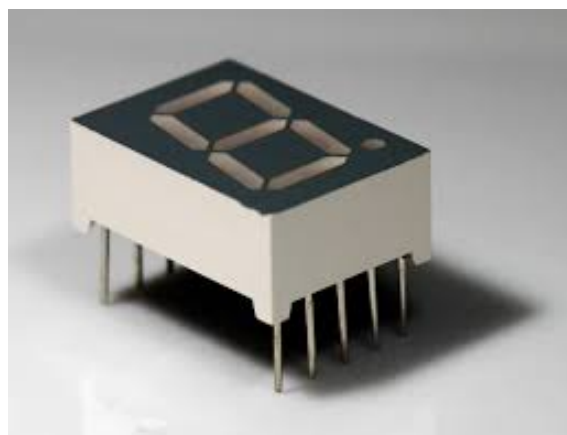


Figura 2. Display de 7 segmentos

3.3. Semáforos de prototipagem

É um módulo de prototipagem no formato de semáforo de trânsito e de baixo custo que permite a simulação de cruzamentos de passagem de veículos.

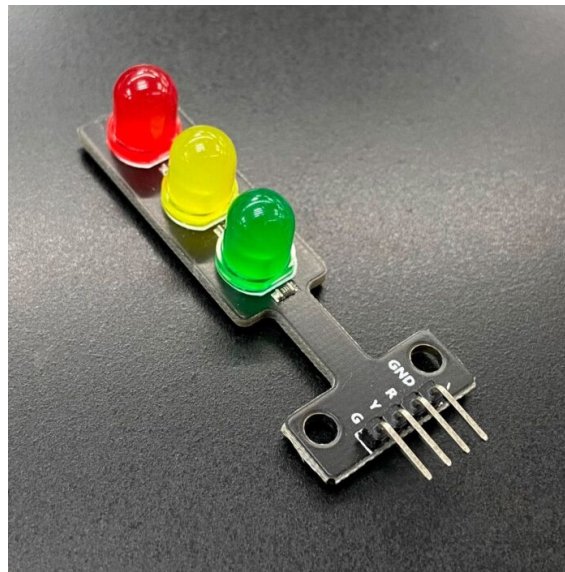


Figura 3. Semáforo de prototipagem

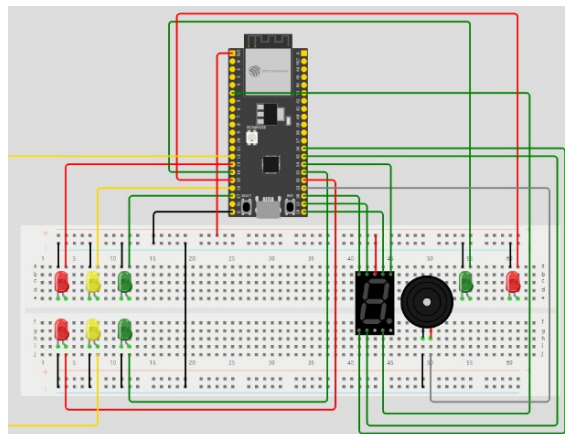


Figura 4. Esquema criado no site Wokwi

4. Método

O esquema do circuito pode ser representado conforme a figura 4.

O circuito real foi montado em protoboard com os componentes e condutores necessários, citados anteriormente. A exposição consiste em demonstrar a dinâmica de um cruzamento de vias com sinalização de dois semáforos para veículos, havendo um semáforo para passagem de pedestres (ativado mediante solicitação do mesmo), sendo sinalizando com semáforo respectivo (vermelho = pare; verde = siga), com exibição de tempo disponível para passagem com contagem regressiva e acompanhamento de sinal sonoro para auxiliar nessa passagem. O seguinte código foi utilizado na plataforma IDE Arduino para implementar o projeto:

```
/* Display 7 segmentos (Entre parênteses, as portas utilizadas no site wokwi, devido à ausência das utilizadas no hardware real.) */
const int segA = 18;
const int segB = 25;
// (34) const int segC = 4;
const int segD = 22;
// (35) const int segE = 23;
// (36) const int segF = 19;
const int segG = 21;
```

```

    // Semáforo 1 (veículos) const int sem1Verde = 17; const int sem1Amarelo = 16;
const int sem1Vermelho = 13;

    // Semáforo 2 (veículos) const int sem2Verde = 33; const int sem2Amarelo = 32;
// (12) const int sem2Vermelho = 26;

    // Semáforo pedestre const int pedestreVerde = 14; const int pedestreVermelho =
15;

    // Buzzer const int buzzer = 27; // (21)

const int botaoPedestre = 0; // Botão de interrupção volatile bool pedRequest =
false;

    // Declaração da função handleButtonPress antes do setup void handleButton-
Press();

    void setup() // Configuração dos segmentos do display pinMode(segA, OUT-
PUT); pinMode(segB, OUTPUT); pinMode(segC, OUTPUT); pinMode(segD, OUT-
PUT); pinMode(segE, OUTPUT); pinMode(segF, OUTPUT); pinMode(segG, OUT-
PUT);

    // Configuração dos semáforos, buzzer e botão de interrupção. pin-
Mode(sem1Vermelho, OUTPUT); pinMode(sem1Amarelo, OUTPUT); pin-
Mode(sem1Verde, OUTPUT); pinMode(sem2Vermelho, OUTPUT); pin-
Mode(sem2Amarelo, OUTPUT); pinMode(sem2Verde, OUTPUT); pin-
Mode(pedestreVermelho, OUTPUT); pinMode(pedestreVerde, OUTPUT); pin-
Mode(buzzer, OUTPUT); pinMode(botaoPedestre, INPUT_PULLUP);

    attachInterrupt(digitalPinToInterrupt(botaoPedestre), handleButtonPress, FAL-
LING);

    // Inicia semáforos com o Semáforo 1 em verde, Semáforo 2 em verme-
lho e Semáforo de pedestre em vermelho. digitalWrite(sem1Verde, HIGH); digital-
Write(sem2Vermelho, HIGH); digitalWrite(pedestreVermelho, HIGH);

    // Inicia display de 7 segmentos (apagado). digitalWrite(segA, HIGH); digi-
talWrite(segB, HIGH); digitalWrite(segC, HIGH); digitalWrite(segD, HIGH); digital-
Write(segE, HIGH); digitalWrite(segF, HIGH); digitalWrite(segG, HIGH);

    void loop() if (pedRequest) pedestrePassagem(); // Executa a função para a pas-
sagem de pedestre pedRequest = false; // Reseta o pedido do pedestre else alternancia-
Veiculos(); // Alternância normal dos semáforos de veículos

    // Função para conceder passagem ao pedestre void pedestrePassagem()
digitalWrite(sem1Vermelho, HIGH); digitalWrite(sem2Vermelho, HIGH); digital-
Write(pedestreVermelho, LOW); digitalWrite(pedestreVerde, HIGH);

    // Contagem regressiva no display e buzina for (int countdown = 9; countdown
!= 0; countdown--) exibirNumero(countdown); if (countdown == 0) tone(buzzer, 1000,
1000); // Emite 1 bip longo delay(1000); else if (countdown != 4) countdown -= 1)
tone(buzzer, 1000, 100); // Emite o primeiro bip breve delay(200); // Espera 200 ms (in-
tervalo entre os bipes) tone(buzzer, 1000, 100); // Emite o segundo bip breve delay(600);
// Espera o restante do tempo até completar 1 segundo else tone(buzzer, 1000, 100); //

```

Emite um bip normal delay(1000); // Espera 1 segundo

```
// Termina o tempo de pedestre digitalWrite(pedestreVerde, LOW); digitalWrite(pedestreVermelho, HIGH); digitalWrite(sem1Vermelho, LOW); digitalWrite(segA, HIGH); digitalWrite(segB, HIGH); digitalWrite(segC, HIGH); digitalWrite(segD, HIGH); digitalWrite(segE, HIGH); digitalWrite(segF, HIGH); digitalWrite(segG, HIGH);
```

```
// Função para exibir número no display de 7 segmentos void exibirNumero(int num) // Zera todos os segmentos antes de exibir o novo número digitalWrite(segA, HIGH); digitalWrite(segB, HIGH); digitalWrite(segC, HIGH); digitalWrite(segD, HIGH); digitalWrite(segE, HIGH); digitalWrite(segF, HIGH); digitalWrite(segG, HIGH);
```

```
switch (num) case 0: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segC, LOW); digitalWrite(segD, LOW); digitalWrite(segE, LOW); digitalWrite(segF, LOW); break; case 1: digitalWrite(segB, LOW); digitalWrite(segC, LOW); break; case 2: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segD, LOW); digitalWrite(segE, LOW); digitalWrite(segG, LOW); break; case 3: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segC, LOW); digitalWrite(segD, LOW); digitalWrite(segG, LOW); break; case 4: digitalWrite(segB, LOW); digitalWrite(segC, LOW); digitalWrite(segF, LOW); digitalWrite(segG, LOW); break; case 5: digitalWrite(segA, LOW); digitalWrite(segC, LOW); digitalWrite(segD, LOW); digitalWrite(segF, LOW); digitalWrite(segG, LOW); break; case 6: digitalWrite(segA, LOW); digitalWrite(segC, LOW); digitalWrite(segD, LOW); digitalWrite(segE, LOW); digitalWrite(segF, LOW); digitalWrite(segG, LOW); break; case 7: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segC, LOW); break; case 8: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segC, LOW); digitalWrite(segD, LOW); digitalWrite(segE, LOW); digitalWrite(segF, LOW); digitalWrite(segG, LOW); break; case 9: digitalWrite(segA, LOW); digitalWrite(segB, LOW); digitalWrite(segC, LOW); digitalWrite(segF, LOW); digitalWrite(segG, LOW); break; delay(1000); // Tempo entre a exibição dos números
```

```
// Função para alternância dos semáforos de veículos void alternanciaVeiculos() digitalWrite(sem1Verde, HIGH); delay(5000); digitalWrite(sem1Verde, LOW);
```

```
digitalWrite(sem1Amarelo, HIGH); delay(2000); digitalWrite(sem1Amarelo, LOW);
```

```
digitalWrite(sem1Vermelho, HIGH);
```

```
delay(1000); digitalWrite(sem2Vermelho, LOW); digitalWrite(sem2Verde, HIGH); delay(5000); digitalWrite(sem2Verde, LOW);
```

```
digitalWrite(sem2Amarelo, HIGH); delay(2000); digitalWrite(sem2Amarelo, LOW);
```

```
digitalWrite(sem2Vermelho, HIGH); delay(1000); digitalWrite(sem1Vermelho, LOW);
```

```
void IRAM_ATTR handleButtonPress() { pedRequest = true; }
```



Figura 5. Circuito montado em protoboard

5. Resultados e discussões

Após a montagem do circuito, o sistema funcionou como projetado, conforme ilustra a foto na Figura 5. O vídeo em SOUSA JÚNIOR, 2024[4] apresenta o funcionamento do projeto proposto.

Referências

- [1] SILVA, Alison Antônio André da; CORREIA, Enrico Vilela; SILVA, Vitor de Paula. TECNOLOGIA WIFI APLICADA NA COMUNICAÇÃO DE SEMÁFOROS, 2023. Disponível em: <https://ric.cps.sp.gov.br/bitstream/123456789/16421/1/TRAFECON.pdf>.
- [2] ESPRESSIF SYSTEMS. Documentação Oficial do ESP32. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>. Acesso em: 12 nov. 2024.
- [3] ESPRESSIF SYSTEMS. Datasheet do ESP32. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 17 nov. 2024.
- [4] SOUSA JÚNIOR, José de Alencar de. SemaforosComPassagemPedesres. YouTube, 2024. Disponível em: <https://youtube.com/shorts/IPDDG6ZffVoç>. Acesso em: 17 nov. 2024.