
Experiência Nº 07
Atuação de cargas simples 2 – Contador/Relógio Display 7 Segmentos

1. OBJETIVOS

- Aplicar os conceitos de acionamentos de cargas simples usando linguagem de programação C para microcontroladores (ARM).
- Realização de funções de Delay para visualização dinâmica de Displays 7 Segmentos.

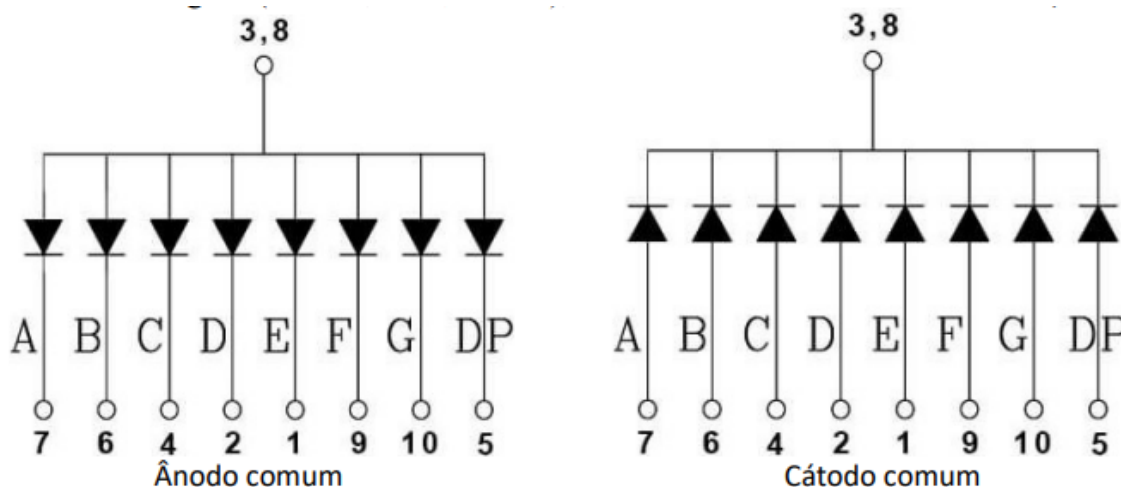
2. INTRODUÇÃO

2.1. Display 7 Segmentos

O display 7 segmentos é um dispositivo baseado em um arranjo de LEDs, que permite representar os dígitos de 0 até 9 (ou alguns caracteres). Duas configurações estão disponíveis dependendo da disposição interna dos LEDs, ânodo comum (Figura 6.1(a)) e catodo comum (Figura 6.1(b)). Para ligar cada segmento, basta com alimentar cada LED associado a esse segmento **sempre usando resistências para limitar a corrente** e garantir o correto funcionamento do dispositivo (evitar queimar/estragar o display ou a fonte/microcontrolador).

Temos 2 casos de ligação dependendo do tipo de display:

- Ânodo comum: No caso as resistências são ligadas entre cada segmento e uma saída digital (0 VDC, ou 3,3 VDC), e o comum vai conectado a VCC (3.3 V por exemplo).
- Cátodo comum: No caso as resistências são ligadas entre cada segmento e uma saída digital (0 VDC, ou 3,3 VDC), e o comum vai conectado a GND (0 VDC).



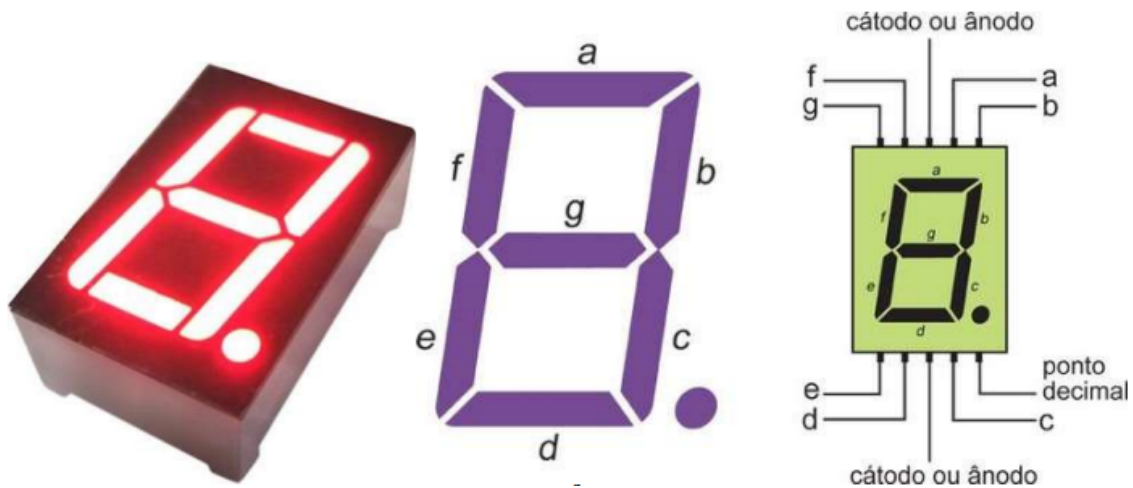


Figura 7.1 Display 7 segmentos. Ânodo comum e Catodo comum.

2.2. Visualização Dinâmica

Se é utilizado o esquema de conexão convencional, no caso de visualizar 4 dígitos, por exemplo, seriam necessários 32 bits/linhas de controle, uma para cada segmento. A visualização dinâmica é uma técnica que permite conectorizar vários *displays* 7 segmentos nas mesmas linhas de controle (no caso 8 linhas), ou seja, conectar os *displays* em paralelo compartilhando os mesmos segmentos (de **a** até **f**, e o **ponto**). Como apresentado na Figura 7.1, os displays contam com 1 comum (pinos 3 e 8), o qual pode ser conectado de forma independente para cada *display* para ativar/desativá-lo.

Para o exemplo de 4 *displays* (cátodo comum), se todos os comuns estão ligados a 0 VDC ou terra, então todos os 4 *displays* vão apresentar o mesmo valor (dependendo de qual segmento esteja ativo), mas se só um display é ligado por vez e em tempos diferentes, é possível apresentar em cada display um valor diferente. Para dar o efeito de todos os displays estarem ligados ao “mesmo tempo” com valores diferentes, é necessário fazer a ativação/desativação de cada display com uma frequência superior aos 50 Hz (para evitar o efeito de flickering). Este processo é conhecido como visualização dinâmica.

3. MATERIAL UTILIZADO

Durante esta experiência, serão utilizados os seguintes equipamentos e componentes:

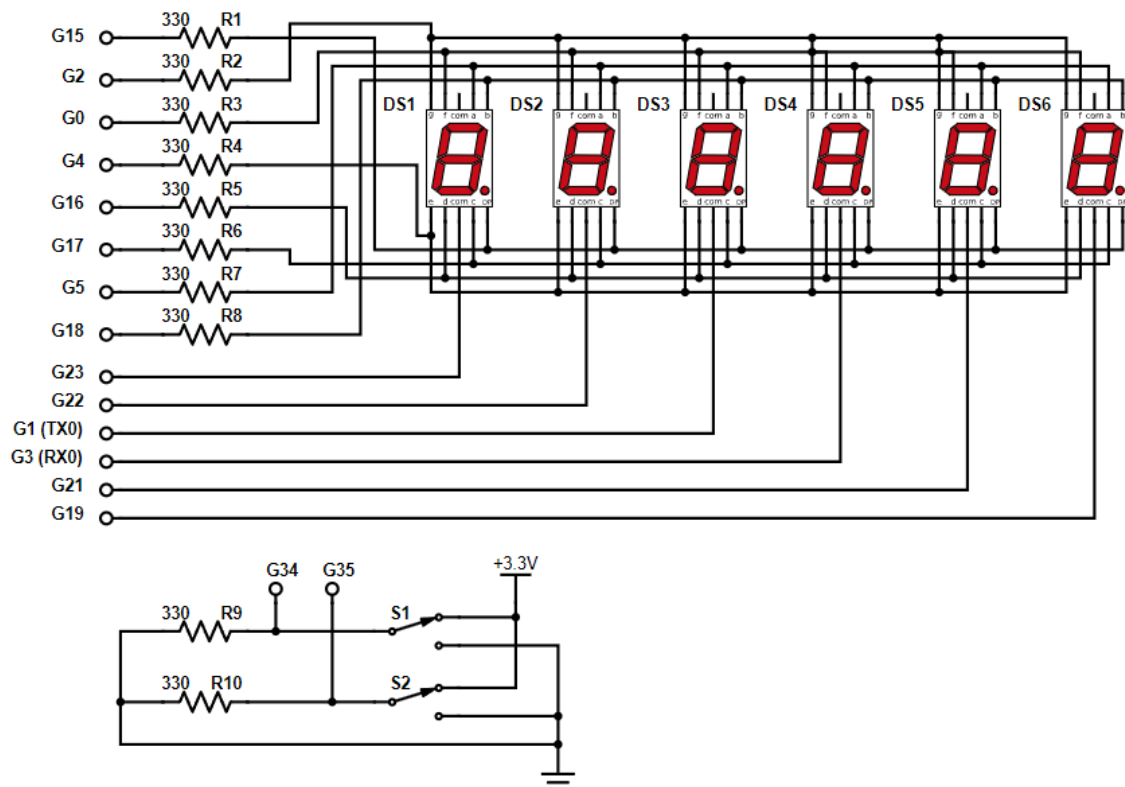
- Placa de desenvolvimento ESP32;
- 6 *Displays* 7 segmentos; ou 2 *Displays* 7 segmentos e 1 *Display* 7 segmentos de 4 dígitos;
- Um protoboard;
- Resistores fixos com os seguintes valores de resistência: 330 Ω ;
- Três chaves seletoras.

4. PROCEDIMIENTO

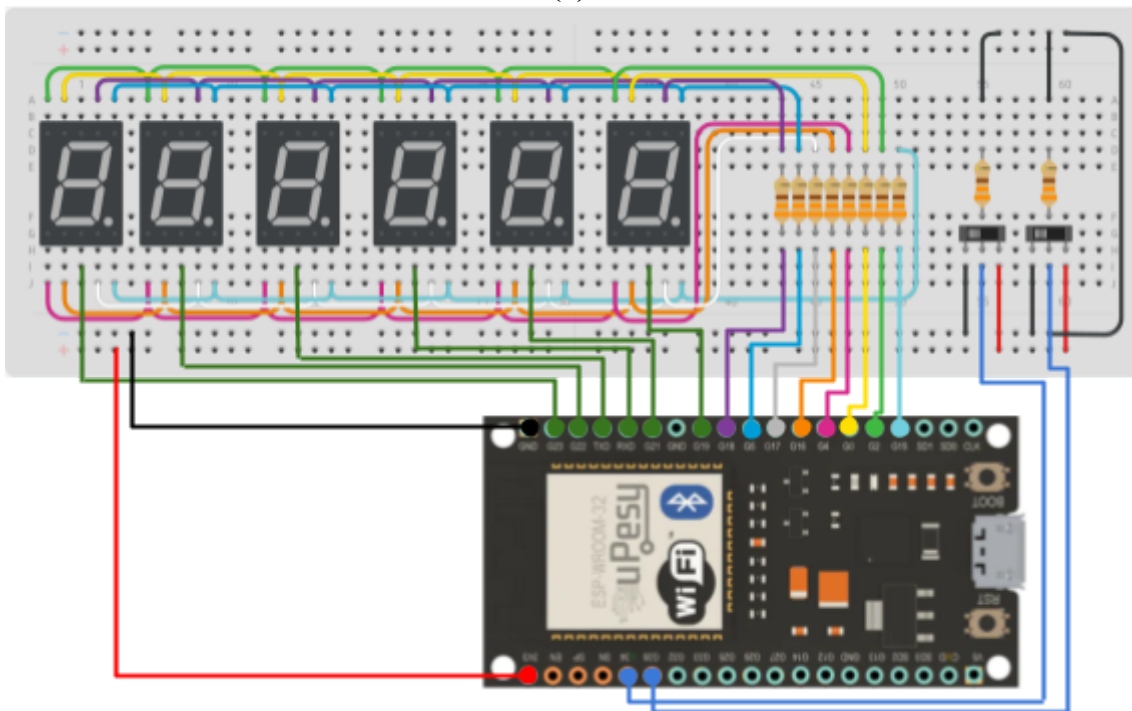
4.1. O circuito utilizado está montado na protoboard com os 6 *displays* de 7 segmentos e três chaves seletoras, de acordo com o diagrama esquemático (Figura 7.2(a)). [Para simplificar a montagem, é recomendado substituir 4 dos *Displays* de 7 segmentos por 1 *Display* de 4 dígitos (Figura 7.2(c)), que já apresenta internamente a interconexão dos segmentos de cada *display*].

4.2. Para ligar a placa, basta conectá-la via USB ao computador.

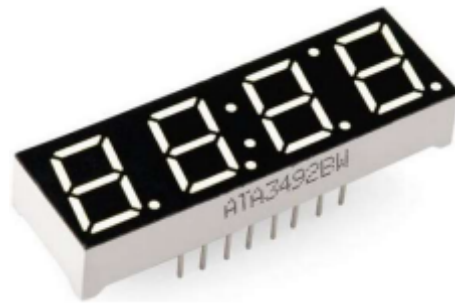
Experiência No 07 - Sistemas Embarcados I



(a)



(b)



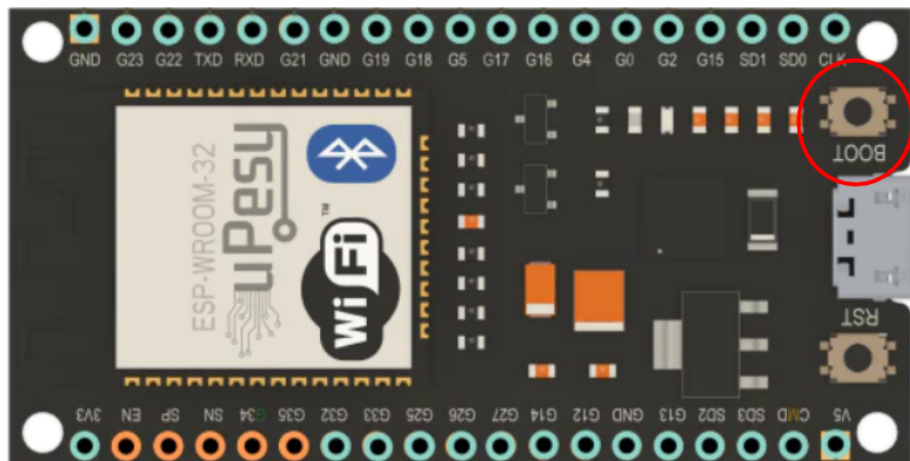
(c)

Figura 7.2 – Circuito de Visualização dinâmica. (a) Diagrama esquemático. (b) circuito de montagem. (c) Display 7 segmentos 4 dígitos.

4.3. Utilizando as três chaves seletoras:

- 4.3.1. Se a chave CS0 é acionada, a função de contador de corrida livre será ativada (de 0 até 9999).
- 4.3.2. No caso do acionamento de CS1, a função de relógio 24h será ativada (00:00h até 23:59h).
- 4.3.3. Se CS2 é acionada, a função de relógio 12h será ativada (12:00 a.m. até 11:59 a.m.). Para indicar o horário da tarde (12:00 p.m. até 11:59 p.m.), será necessário ligar o LED1 da PD. O LED0 da PD deve ser ligado para indicar o horário da manhã.

OBS.: Para programar a ESP32, em alguns casos, ao carregar o código é preciso manter pressionado o botão BOOT (Figura 7.3(a)) assim que a mensagem “Connecting...” aparecer no terminal da Arduino IDE (Figura 7.3(b)), até que seja iniciada a escrita na memória do ESP:



(a)

```
esptool.py v4.5.1
Serial port COM3
Connecting.....
```

(b)

Figura 7.3 – Programação da ESP32. (a) Localização do botão BOOT. (b) Mensagens no terminal da Arduino IDE.