

## Projeto No. 12 – Display de Led de 7 Segmentos

O objetivo deste projeto é demonstrar a utilização de um display de led de 7 segmentos controlado diretamente a partir das portas digitais do Arduino.

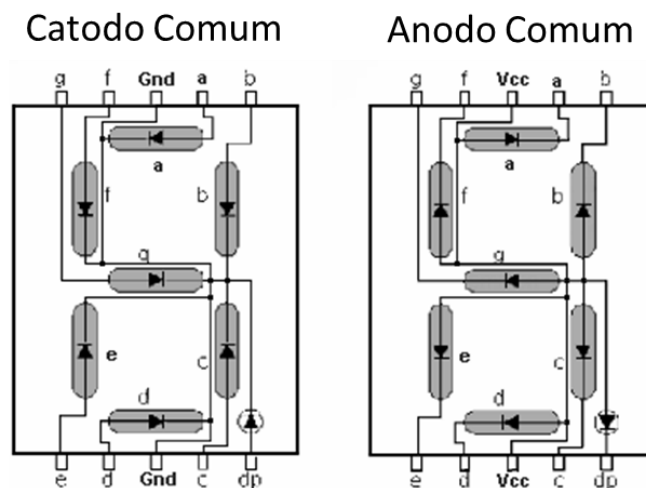
### Material necessário:

- 1 Arduino
- 1 Display de Led de 7 Segmentos Catodo ou Anodo Comum (1 dígito)\*
- 1 Resistor de 220 ohms (vermelho, vermelho, marrom) ou 330 ohms (laranja, laranja, marrom)\*
- 1 Protoboard\*
- Jumper cable

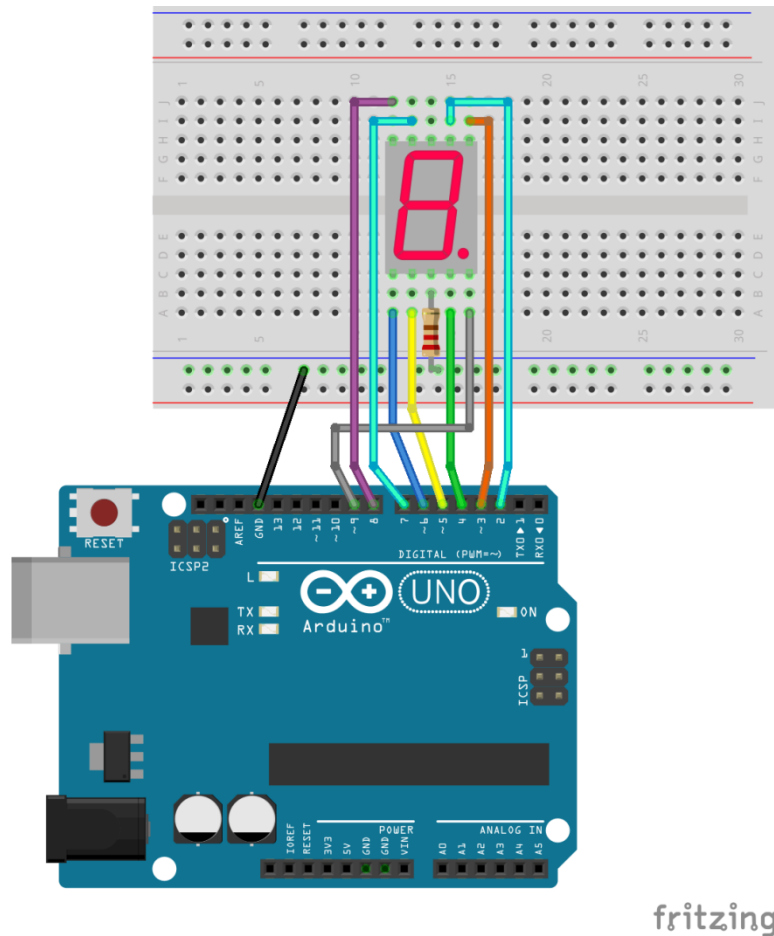
\* Podem ser substituídos pelo módulo P11-Display Simples da GBK Robotics.

### Passo 1: Displays de Led de 7 Segmentos

Os displays de led de sete segmentos são bastante comuns e muitos utilizados para exibir, principalmente, informações numéricas. Podem ser de dois tipos:



## Passo 2: Montagem do circuito



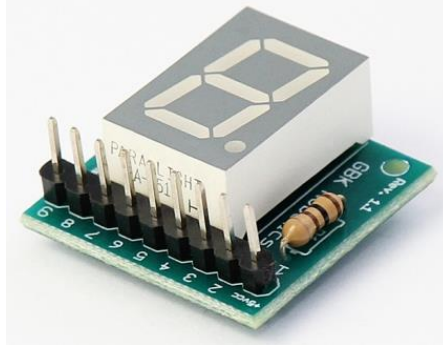
fritzing

Adotando como referência a figura acima, realizar a sequência de montagem:

- Pino 1 (segmento e) do display ligado ao 6 do Arduino;
- Pino 2 (segmento d) do display ligado ao 5 do Arduino;
- Pino 3 (Gnd, se catodo comum ou Vcc se anodo comum) do display ligado ao resistor de 220 ohms;
- Resistor de 220 ohms ligado ao Gnd (se catodo comum) ou Vcc (se anodo comum) do Arduino;
- Pino 4 (segmento c) do display ligado ao 4 do Arduino;
- Pino 5 (ponto decimal) do display ligado ao 9 do Arduino;
- Pino 6 (segmento b) do display ligado ao 3 do Arduino;
- Pino 7 (segmento a) do display ligado ao 2 do Arduino;
- Pino 9 (segmento f) do display ligado ao 7 do Arduino;
- Pino 10 (segmento g) do display ligado ao 8 do Arduino.

## Variação de Montagem

# Módulo P11-Display Simples da GBK Robotics



Este projeto pode ser montado substituindo o Display de Led de 7 Segmentos, o resistor de 220 ohms (ou 330 ohms) e a protoboard pelo módulo P11-Display Simples da GBK Robotics, neste caso:

- Conecte o pino 5V do Arduino ao pino +5Vcc do módulo P11;
- Conecte o pino 2 do Arduino ao pino 2 do módulo P11;
- Conecte o pino 3 do Arduino ao pino 3 do módulo P11;
- Conecte o pino 4 do Arduino ao pino 4 do módulo P11;
- Conecte o pino 5 do Arduino ao pino 5 do módulo P11;
- Conecte o pino 6 do Arduino ao pino 6 do módulo P11;
- Conecte o pino 7 do Arduino ao pino 7 do módulo P11;
- Conecte o pino 8 do Arduino ao pino 8 do módulo P11;
- Conecte o pino 9 do Arduino ao pino 9 do módulo P11.

**IMPORTANTE:** Para o módulo P11 nos sketches (programas) a seguir adotar a configuração para anodo comum.

### Passo 3: Programa 1 – Exibindo a Letra “A”

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
int SEG_A = 2;
int SEG_B = 3;
int SEG_C = 4;
int SEG_D = 5;
int SEG_E = 6;
int SEG_F = 7;
int SEG_G = 8;
int PONTO = 9;
int ATRASO = 1000;

void setup() {
  for (int pino = SEG_A; pino <= PONTO; pino++) {
    pinMode(pino, OUTPUT);
    // Para displays de catodo comum:
    digitalWrite(pino, LOW);
    // Para displays de anodo comum:
    // digitalWrite(pino, HIGH);
  }
}
```



```
}  
}  
  
void loop() {  
  // Para displays de catodo comum:  
  digitalWrite(SEG_A, HIGH);  
  digitalWrite(SEG_B, HIGH);  
  digitalWrite(SEG_C, HIGH);  
  digitalWrite(SEG_E, HIGH);  
  digitalWrite(SEG_F, HIGH);  
  digitalWrite(SEG_G, HIGH);  
  
  // Para displays de anodo comum:  
  // digitalWrite(SEG_A, LOW);  
  // digitalWrite(SEG_B, LOW);  
  // digitalWrite(SEG_C, LOW);  
  // digitalWrite(SEG_E, LOW);  
  // digitalWrite(SEG_F, LOW);  
  // digitalWrite(SEG_G, LOW);  
  
  delay(ATRASO);  
}
```

#### Passo 4: Programa 2 – Animação Simples

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
int SEG_A = 2;  
int SEG_B = 3;  
int SEG_C = 4;  
int SEG_D = 5;  
int SEG_E = 6;  
int SEG_F = 7;  
int SEG_G = 8;  
int PONTO = 9;  
  
int ATRASO = 150;  
  
void setup() {  
  for (int pino = SEG_A; pino <= PONTO; pino++) {  
    pinMode(pino, OUTPUT);  
    // Para displays de catodo comum:  
    digitalWrite(pino, LOW);  
    // Para displays de anodo comum:  
    // digitalWrite(pino, HIGH);  
  }  
}  
  
void loop() {  
  for (int pino = SEG_A; pino < SEG_G; pino++) {  
    // Para displays de catodo comum:  
    digitalWrite(pino, HIGH);  
  }  
}
```



```
// Para displays de anodo comum:
// digitalWrite(pino, LOW);
if (pino > SEG_A) {
    // Para displays de catodo comum:
    digitalWrite(pino - 1, LOW);
    // Para displays de anodo comum:
    // digitalWrite(pino - 1, HIGH);
}
else {
    // Para displays de catodo comum:
    digitalWrite(SEG_F, LOW);
    // Para displays de anodo comum:
    // digitalWrite(SEG_F, HIGH);
}
delay(ATRASO);
}
```

### Passo 5: Programa 3 – Contagem Regressiva

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
// Matriz com os dígitos de 0 a 9.
byte digitos[10][7] = {
    { 1,1,1,1,1,1,0 }, // = 0
    { 0,1,1,0,0,0,0 }, // = 1
    { 1,1,0,1,1,0,1 }, // = 2
    { 1,1,1,1,0,0,1 }, // = 3
    { 0,1,1,0,0,1,1 }, // = 4
    { 1,0,1,1,0,1,1 }, // = 5
    { 1,0,1,1,1,1,1 }, // = 6
    { 1,1,1,0,0,0,0 }, // = 7
    { 1,1,1,1,1,1,1 }, // = 8
    { 1,1,1,0,0,1,1 }  // = 9
};

void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pontoDecimal(false);
}

void pontoDecimal(boolean ponto) {
    digitalWrite(9, ponto);
}
```



```
void escrever(int digito) {
    int pino = 2;
    for (int segmento = 0; segmento < 7; segmento++) {
        // Para Catodo Comum:
        digitalWrite(pino, digitos[digito][segmento]);
        // Para Anodo Comum - apenas inverter o valor através do operador not (!):
        // digitalWrite(pino, !digitos[digito][segmento]);
        pino++;
    }
    pontoDecimal(false);
}

void limpar() {
    byte pino = 2;
    for (int segmento = 0; segmento < 7; segmento++) {
        // Para Catodo Comum:
        digitalWrite(pino, LOW);
        // Para Anodo Comum:
        // digitalWrite(pino, HIGH);
        pino++;
    }
}

void loop() {
    for (int cont = 9; cont >= 0; cont--) {
        escrever(cont);
        boolean ponto = true;
        for (int i = 0; i < 4; i++) {
            delay(250);
            pontoDecimal(ponto);
            ponto = !ponto;
        }
    }
    limpar();
    delay(1000);
}
```