

Projeto No. 13 – Display de Led de 7 Segmentos e 4 Dígitos

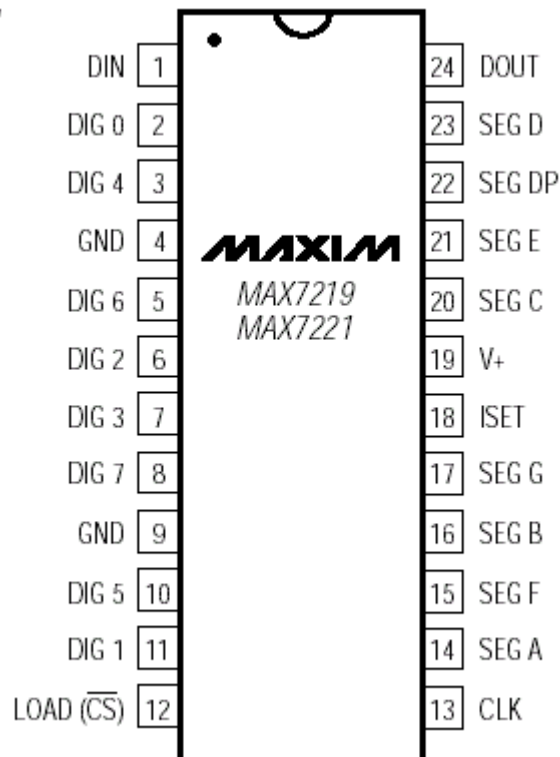
O objetivo deste projeto é demonstrar a utilização do display de led de 7 segmentos e 4 dígitos controlado através do CI MAX 7219 ou 7221.

Material necessário:

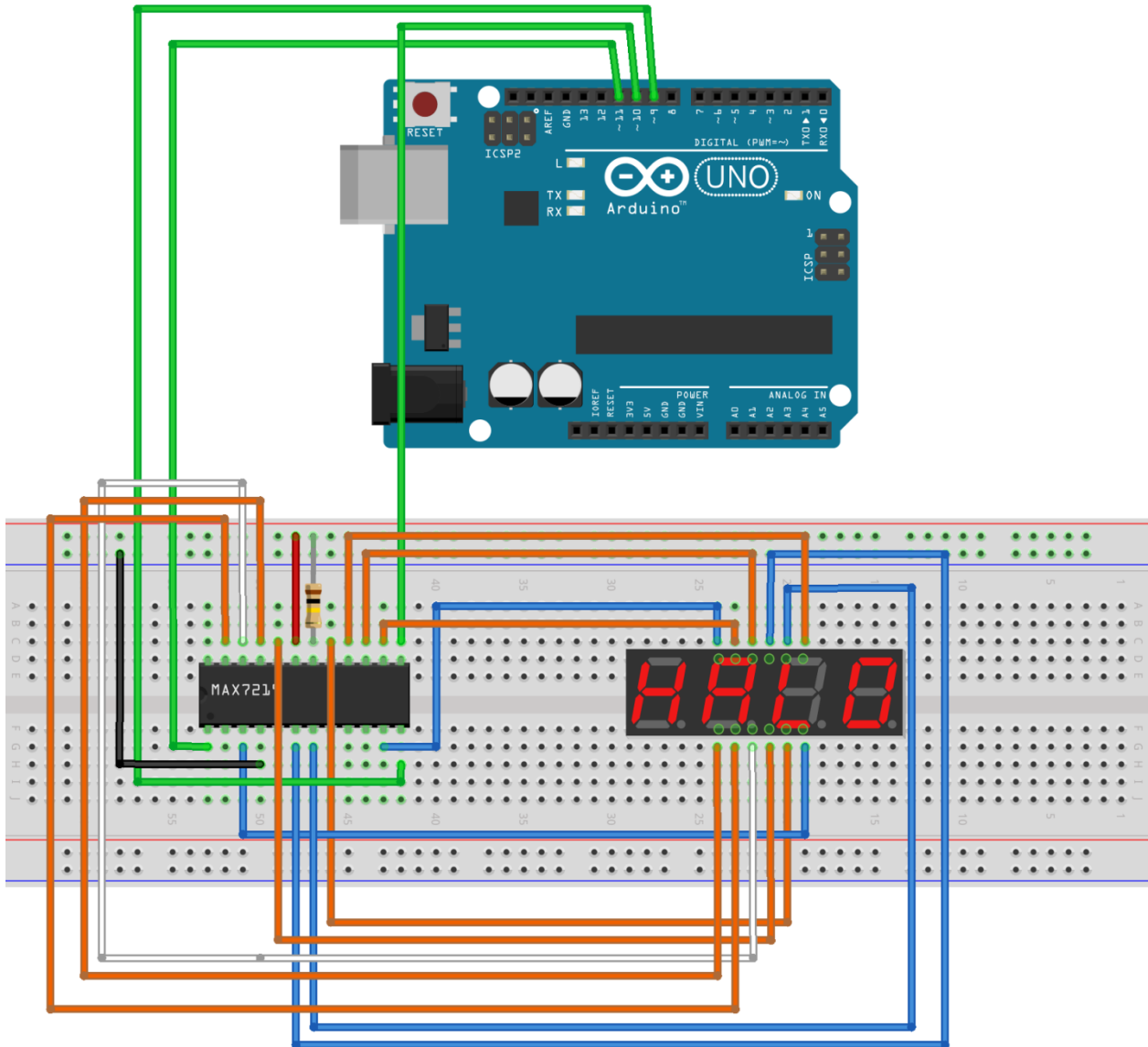
- 1 Arduino
- 1 Display de Led de 7 Segmentos (4 dígitos)
- 1 Circuito Integrado (CI) MAX 7219 ou 7221
- 1 Resistor de 100k ohms (marrom, preto, amarelo)
- 1 Protoboard
- Jumper cable

Passo 1: Uso de displays com múltiplos dígitos

Você já deve ter observado que quando precisamos utilizar displays de leds que apresentam mais do que um dígito, os portas disponíveis no Arduino não serão suficientes ou mesmo que sejam suficientes, não permitirão colocar novas funcionalidades ao seu projeto como, por exemplo, um sensor de temperatura ou um módulo de relógio em tempo real (RTC). Desta maneira, para otimizar o uso das portas do Arduino devemos utilizar um driver para displays de led sendo, o mais popular, o Maxim 7219 ou 7221, cuja pinagem pode ser observada a seguir.



Passo 2: Montagem do circuito



fritzing

Passo 3: Programa 1

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

```
// Este sketch exibe o número 1234 em um display de led de 7 segmentos com 4 dígitos

// Utilizar a biblioteca LedControl
#include "LedControl.h"

/*
 * Criar um LedControl (lc).
 * O pino 11 do Arduino deve ser conectado ao pino DATA IN do primeiro MAX7219/21
 * O pino 10 do Arduino deve ser conectado ao pino CLK do primeiro MAX7219/21
 */
```



```
* 0 pino 9 do Arduino deve ser conectado ao pino LOAD (/CS) do primeiro MAX7219/21
* 0 quarto parâmetro indica que há apenas um MAX7219/21 conectado ao Arduino
*/
LedControl lc = LedControl(11, 10, 9, 1);

void setup()
{ // Retira o MAX7219/21 do modo de economia de energia
  lc.shutdown(0, false);
  // Define a intensidade do brilho dos leds
  lc.setIntensity(0, 2);
  lc.clearDisplay(0);
}

void loop()
{ lc.setChar(0, 2, '0', false);
  lc.setChar(0, 2, '1', false);
  lc.setChar(0, 3, '2', false);
  lc.setChar(0, 4, '3', false);
}
```

Passo 4: Programa 2

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

```
/* Este sketch exibe os números inteiros entre -999 e 999 em um display de led de
* 7 segmentos com 4 dígitos
*/

// Utilizar a biblioteca LedControl
#include "LedControl.h"

/* Criar um LedControl (lc).
* 0 pino 11 do Arduino deve ser conectado ao pino DATA IN do primeiro MAX7219/21
* 0 pino 10 do Arduino deve ser conectado ao pino CLK do primeiro MAX7219/21
* 0 pino 9 do Arduino deve ser conectado ao pino LOAD (/CS) do primeiro MAX7219/21
* 0 quarto parâmetro indica que há apenas um MAX7219/21 conectado ao Arduino
*/
LedControl lc = LedControl(11, 10, 9, 1);
int i = -999;

void setup()
{ // Retira o MAX7219/21 do modo de economia de energia
  lc.shutdown(0, false);
  // Define a intensidade do brilho dos leds
  lc.setIntensity(0, 2);
  lc.clearDisplay(0);
}

void loop()
{ exibirInteiro(i++);
}
```



```
void exibirInteiro(int valor)
{
    int unidade;
    int dezena;
    int centena;
    boolean negativo = false;

    if(valor < -999 || valor > 999)
        return;

    if(valor < 0)
    {
        negativo = true;
        valor = valor * (-1);
    }

    unidade = valor % 10;
    valor = valor / 10;
    dezena = valor % 10;
    valor = valor / 10;
    centena = valor;

    if (negativo)
    { // Imprime o sinal de negativo no display que está mais a esquerda
        lc.setChar(0, 1, '-', false);
    }
    else
    { // Imprime um espaço no lugar do sinal de negativo
        lc.setChar(0, 1, ' ', false);
    }

    // Exibo o número dígito a dígito
    lc.setDigit(0, 2, (byte)centena, false);
    lc.setDigit(0, 3, (byte)dezena, false);
    lc.setDigit(0, 4, (byte)unidade, false);
    delay(100);
}
```