UTFPR
Engenharia de Computação
STCO7A - Sistemas Microcontrolados

Experimento - 7 segmentos

Aluno: Deivid da Silva Galvão RA: 2408740
Aluno: Eduardo Yuji Yoshida Yamada RA: 2320606
Professor orientador: Mauricio Eiji Nakai

Abril
2025

UTFPR
Engenharia de Computação
STCO7A - Sistemas Microcontrolados

Relatório

Relatório do Trabalho Prático Disciplinar apresentado como requisito parcial à obtenção de nota na disciplina de Sistemas Microcontrolados do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Aluno: Deivid da Silva Galvão RA: 2408740

Aluno: Eduardo Yuji Yoshida Yamada RA: 2320606

Professor orientador: Mauricio Eiji Nakai

Abril
2025

Conteúdo

1 Introdução

Neste experimento, exploramos a interface entre sensores analógicos e dispositivos de saída digitais utilizando um microcontrolador Arduino. O objetivo é desenvolver um sistema capaz de interpretar o valor analógico de um potenciômetro e convertê-lo em uma escala de 00 a 99, exibida em dois displays de sete segmentos. Para alcançar esse objetivo, limitamos o uso a apenas 9 portas digitais do Arduino, exigindo uma abordagem eficiente para a multiplexação dos sinais e a correta interpretação dos dados fornecidos pelo sensor.

2 Implementação

2.1 Código do Arduíno:

```
-----------------------------------------------------------------
int ledA = 22;
int ledB = 24;
int ledC = 26;
int ledD = 28;
int ledE = 30;
int ledF = 32;
int ledG = 34;
int jumper1 = 40;
int jumper2 = 42;
int contador, valor, dezena, unidade;
int analogPin = A1;

byte segmentos[10][7] = {
  { 1, 1, 1, 1, 1, 1, 0 },  //DIGITO 0
  { 0, 1, 1, 0, 0, 0, 0 },  //DIGITO 1
  { 1, 1, 0, 1, 1, 0, 1 },  //DIGITO 2
  { 1, 1, 1, 1, 0, 0, 1 },  //DIGITO 3
  { 0, 1, 1, 0, 0, 1, 1 },  //DIGITO 4
  { 1, 0, 1, 1, 0, 1, 1 },  //DIGITO 5
  { 1, 0, 1, 1, 1, 1, 1 },  //DIGITO 6
  { 1, 1, 1, 0, 0, 0, 0 },  //DIGITO 7
  { 1, 1, 1, 1, 1, 1, 1 },  //DIGITO 8
  { 1, 1, 1, 1, 0, 1, 1 }   //DIGITO 9
};

void setup() {
  Serial.begin(9600);
  pinMode(ledA, OUTPUT);
  pinMode(ledB, OUTPUT);
  pinMode(ledC, OUTPUT);
  pinMode(ledD, OUTPUT);
  pinMode(ledE, OUTPUT);
  pinMode(ledF, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(jumper1, OUTPUT);
  pinMode(jumper2, OUTPUT);
}

void loop() {
  valor = map(analogRead(analogPin), 0, 1023, 0, 99);
  Serial.println(valor);

  //contador++;
  //if (contador > 50) {
    //contador = 0;
    //valor++;
    dezena = valor / 10;
    unidade = valor % 10;
```

```
    //if (valor > 99) {
      //valor = 0;
    //}
  //}

  digitalWrite(ledA, segmentos[dezena][0]);
  digitalWrite(ledB, segmentos[dezena][1]);
  digitalWrite(ledC, segmentos[dezena][2]);
  digitalWrite(ledD, segmentos[dezena][3]);
  digitalWrite(ledE, segmentos[dezena][4]);
  digitalWrite(ledF, segmentos[dezena][5]);
  digitalWrite(ledG, segmentos[dezena][6]);
  digitalWrite(jumper1, HIGH);
  digitalWrite(jumper2, LOW);
  delay(5);
  digitalWrite(ledA, segmentos[unidade][0]);
  digitalWrite(ledB, segmentos[unidade][1]);
  digitalWrite(ledC, segmentos[unidade][2]);
  digitalWrite(ledD, segmentos[unidade][3]);
  digitalWrite(ledE, segmentos[unidade][4]);
  digitalWrite(ledF, segmentos[unidade][5]);
  digitalWrite(ledG, segmentos[unidade][6]);
  digitalWrite(jumper1, LOW);
  digitalWrite(jumper2, HIGH);
  delay(5);
}
```

2.2 Explicação do Código:

Primeiramente, são declaradas as variáveis dos LEDs correspondentes aos segmentos do display; por exemplo, ledA está associado à porta 22, ledB à porta 24 e assim por diante. Em seguida, definem-se os jumpers, que servem para controlar a seleção dos displays, e, por fim, o pino analógico destinado à leitura do potenciômetro. Após essas declarações, é criada uma matriz que facilita a conversão dos valores de binário para decimal, permitindo assim que os dígitos de 0 a 9 sejam exibidos corretamente.

A função setup é responsável por configurar todas as portas digitais utilizadas – tanto para os segmentos dos displays quanto para os jumpers – como saídas, garantindo o controle adequado de cada componente durante o funcionamento do sistema. Já na função loop, o Arduino realiza a leitura contínua do valor analógico obtido do potenciômetro e o converte para uma escala entre 0 e 99 utilizando a função map(). Com esse valor, procede-se à divisão entre a dezena e a unidade, assegurando que cada dígito seja mostrado no display correspondente.

Posteriormente, os segmentos dos displays são ativados com base na matriz segmentos. Primeiro, exibe-se a dezena, e em seguida, por meio da alternância dos jumpers (jumper1 e jumper2), é exibida a unidade. Essa troca ocorre com um pequeno atraso (delay de 5 milissegundos), permitindo a multiplexação eficiente dos displays e garantindo que a informação seja apresentada de maneira contínua e correta.

3 Resultados Obtidos

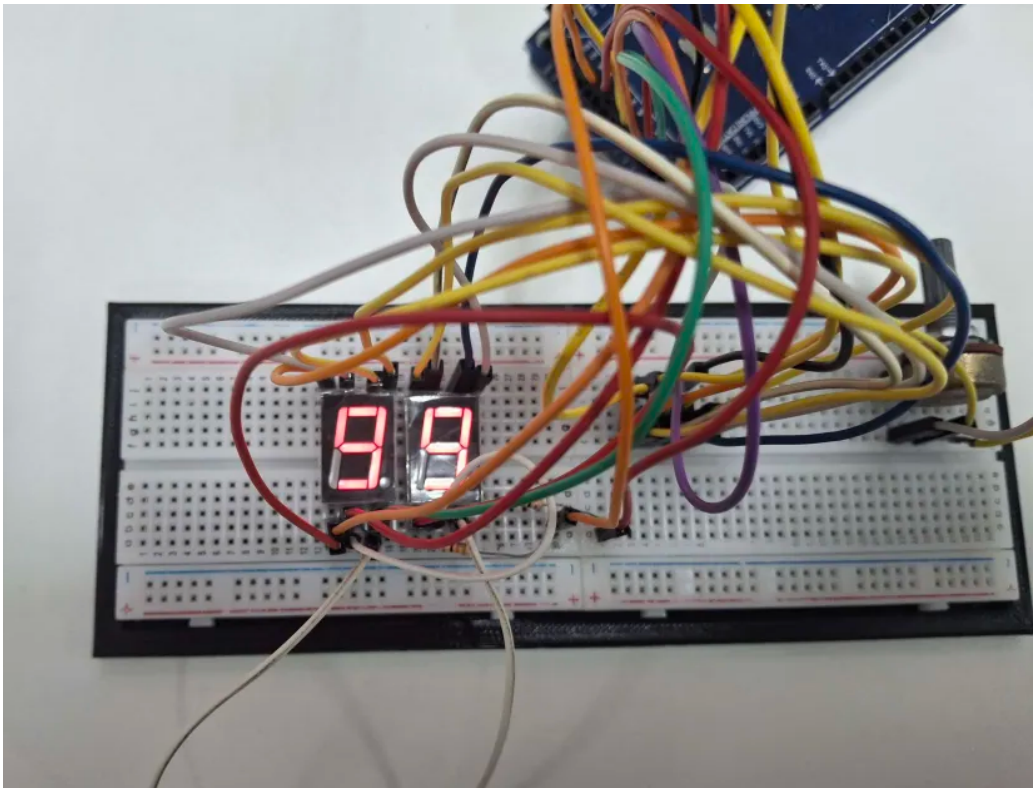

Figura 1: Foto do circuito montado no laboratório

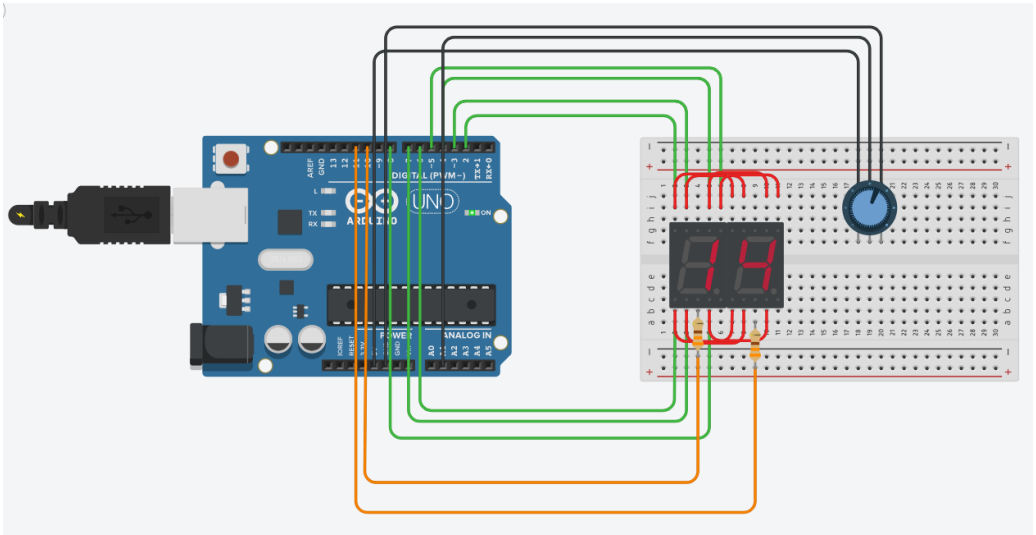

Figura 2: Foto do circuito montado no tinkercad

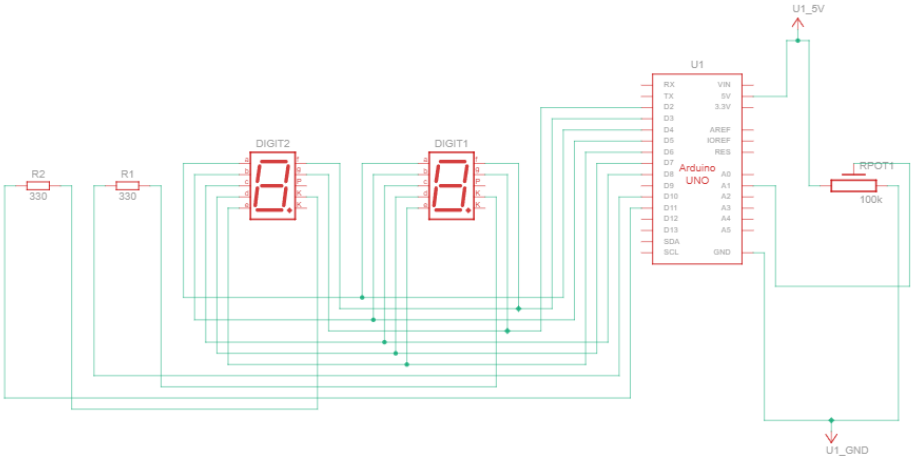

Figura 3: Foto do circuito montado no tinkercad(esquemático)

Quando movimentamos o potenciômetro, o *map* faz a leitura analógica da entrada e então a converte e joga para a variável *valor*. O valor fica limitado de 0 a 99, então é feita uma conversão, com base na variável 'valor', para dezena e unidade e assim conseguimos alterar cada display. O display da esquerda recebe o valor de dezena e o da direita os valores de unidade.

Ao movimentar o potenciômetro totalmente para a direita, temos o valor máximo que equivale, no display, a 99. E quando rodamos totalmente para a esquerda, o valor exibido é 00.

4 Conclusão

Por fim, pode-se concluir que o experimento demonstrou de forma prática como integrar a leitura de sinais analógicos e exibi-los em um display de sete segmentos. Ao utilizar a técnica de multiplexação, foi possível otimizar o uso das portas digitais. Essa abordagem reforça a importância de combinar hardware e software em projetos, podendo servir como base para o desenvolvimento de aplicações mais complexas futuramente.