



UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
Departamento de Engenharia Mecânica
Laboratório de Instrumentação Eletrônica

SEM0142 - Sensores e Sistemas de Medidas
Responsável: Prof. Dr. Leopoldo Pisanelli Rodrigues de Oliveira

Orientações Laboratório 02: Familiarização em Arduíno

São Carlos
2023

Sumário

1	Introdução	2
2	Materiais	2
3	Procedimentos	3
3.1	Semáforo - Prática #1 da Apostila Arduino	3
3.2	Leitura Analógica - Prática #3 da Apostila Arduino	4
3.3	Construção de Ohmímetro	6
4	Relatório	6

1 Introdução

Esta prática possui o o objetivo principal de permitir o aluno familiarizar-se com a plataforma, código e alguns comandos básicos do Arduino, implementando um código simples que controle um conjunto de LEDs (ver apostila, [??], na Seção 4 – Escrita Digital p. 18). Em seguida, trabalhar com leitura de dados analógicos (ver apostila, [??], na Seção 6 – Leitura Analógica p.27).

2 Materiais

- 1x Arduino UNO ou MEGA
- 1x Protoboard
- 3x LEDs
- 1x Potenciômetro
- Jumpers machoXmacho
- Resistores

3 Procedimentos

Conecte o arduino ao PC e abra a IDE. Verifique que tudo esteja funcionando, a IDE conectada na porta correta, o monitor serial abriu, etc...

3.1 Semáforo - Prática #1 da Apostila Arduino

No seu software preferido (ou bancada) monte o circuito da figura 1 e depois digite o código da figura 2:

Veja que este circuito é composto de três leds, representando as cores de um semáforo. O led verde fica aceso por 3 segundos, o amarelo depois acende por 1 segundo e o vermelho acende por último durante 2 segundos, então a sequência se reinicia. Quando um led está aceso os outros estão apagados, assim como no funcionamento do semáforo normal. Os pinos 13,12,11 estão conectados aos Leds verde, amarelo e vermelho respectivamente, na configuração cátodo comum, ou seja, acendem com nível lógico alto. Os resistores de 270R servem para limitar a corrente em cada led. No código mostrado na figura 2, teremos a diretiva 'define' para nomear os pinos usados e assim facilitar nossa leitura e atribuições no código. Usamos o comando 'pinMode' para definir os três pinos como saída de dados e o comando 'digitalWrite' para escolher qual pino vai a nível lógico alto ou baixo.

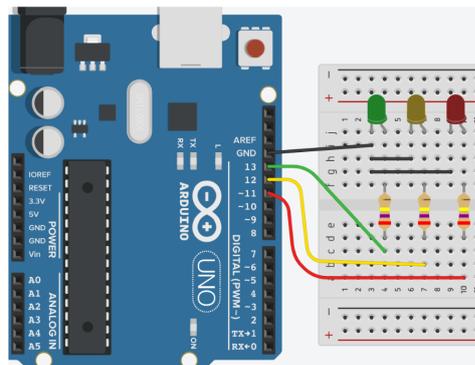


Figura 1 – Circuito a ser implementado para a Prática de Semáforo

Manipule as variáveis para ganhar familiaridade com a operação do sistema, testando diferentes posições de pinos e tempos de programação.

Se lembre de tirar uma foto da montagem para a escrita do relatório.

```
Texto [v] [Download] [Save] [Run] 1 (Arduino Uno R3)
1 //exercício 1 - semáforo
2 // tudo que estiver após // é comentário
3
4 #define VERDE 13 // o pino 13 acenderá o led verde
5 #define AMARELO 12 // o pino 12 acenderá o led amarelo
6 #define VERMELHO 11 // o pino 11 acenderá o led vermelho
7
8 void setup()
9 {
10     pinMode(VERDE, OUTPUT); //defino os pinos como saída
11     pinMode(AMARELO, OUTPUT);
12     pinMode(VERMELHO, OUTPUT);
13 }
14
15 void loop()
16 {
17     //primeiro ciclo o semáforo verde acende - 3segundos
18     digitalWrite(VERDE, HIGH);
19     digitalWrite(AMARELO, LOW);
20     digitalWrite(VERMELHO, LOW);
21     delay(3000);
22
23     //primeiro ciclo o semáforo amarelo acende - 1segundo
24     digitalWrite(VERDE, LOW);
25     digitalWrite(AMARELO, HIGH);
26     digitalWrite(VERMELHO, LOW);
27     delay(1000);
28
29     //primeiro ciclo o semáforo amarelo acende - 2segundos
30     digitalWrite(VERDE, LOW);
31     digitalWrite(AMARELO, LOW);
32     digitalWrite(VERMELHO, HIGH);
33     delay(2000);
34 }
```

Figura 2 – Código a ser implementado para a Prática de Semáforo

3.2 Leitura Analógica - Prática #3 da Apostila Arduino

O esquemático da figura 14 e o código da figura 15 são referentes a um circuito que lê a tensão de um potenciômetro e aciona três leds. O potenciômetro tem um cursor que rotaciona aproximadamente 180°. Neste caso, um led é aceso representando em qual faixa de ângulo o eixo foi rotacionado. Então de 0° a 59° acendemos um led, de 60° a 119° acendemos outro led e de 120° a 180° acendemos outro led. Perceba que é necessário realizar uma regra de três para achar os valores de ângulo a partir das tensões aferidas. Veja que o primeiro passo é ler o valor pelo pino analógico. A informação será um inteiro entre 0 e 1023. Depois precisamos converter para tensão, já que temos a resolução do Arduino, e depois convertemos para ângulo, já que o potenciômetro varia a tensão conforme giramos

seu eixo. Desta forma temos:

$$tensão = (resolução) * (leituraA/D) = 0.004882813 * (leituraA/D) \quad (1)$$

Depois para achar o valor do ângulo, se $180 = 5V$, então por regra de três:

$$\hat{angulo} = \frac{tensão * 180}{5} \quad (2)$$

Veja como isso é implementado no código (veja figura 4) e faça o experimento em bancada (por meio do circuito em figura 3):

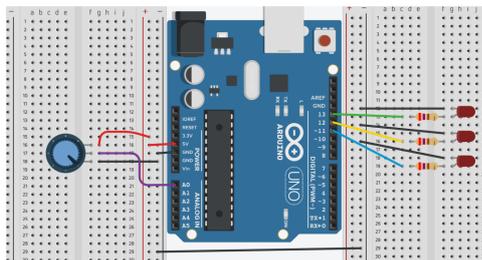


Figura 3 – Circuito a ser implementado para Prática de Leitura Analógica

```

1 //exercício 3 - leitura potenciômetro
2
3 #define LED1 11 //led ligado ao pino digital 11
4 #define LED2 12 //led ligado ao pino digital 12
5 #define LED3 13 //led ligado ao pino digital 13
6 #define POT A0 //potenciômetro ligado ao pino analogico A0
7
8 //variáveis
9 int leitura_pot = 0; //guarda valor da leitura do potenciômetro (vai de 0 a 1023)
10 float conversao_tensao = 0; //guarda o valor da conversão feita para tensão
11 float conversao_angulo = 0; //guarda o valor da conversão feita para angulo
12 void setup()
13 {
14     pinMode(LED1, OUTPUT); //pino 11 como saída
15     pinMode(LED2, OUTPUT); //pino 12 como saída
16     pinMode(LED3, OUTPUT); //pino 13 como saída
17     pinMode(POT, INPUT); //pino A0 como entrada
18 }
19
20 void loop()
21 {
22     leitura_pot = analogRead(POT); // afere o dado do potenciômetro e guarda
23     conversao_tensao = (float)(0.004882813*leitura_pot); // realiza conversão para tensão
24     conversao_angulo = (conversao_tensao*180)/(5); //realiza conversao para angulo
25
26     if(conversao_angulo <= 59) // se o angulo esta entre 0° e 59°
27     {
28         digitalWrite(LED1, HIGH); //acende apenas o led1
29         digitalWrite(LED2, LOW);
30         digitalWrite(LED3, LOW);
31     }
32
33     if((conversao_angulo > 59)&&(conversao_angulo <= 119)) // se o angulo esta entre 60° e 119°
34     {
35         digitalWrite(LED1, LOW);
36         digitalWrite(LED2, HIGH); //acende apenas o led2
37         digitalWrite(LED3, LOW);
38     }
39
40     if(conversao_angulo > 119) // se o angulo esta entre 120° e 180°
41     {
42         digitalWrite(LED1, LOW);
43         digitalWrite(LED2, LOW);
44         digitalWrite(LED3, HIGH); //acende apenas o led3
45     }
46 }

```

Figura 4 – Código a ser implementado para Prática de Leitura Analógica

Outro aspecto muito importante a salientar é que as conversões requerem números com ponto flutuante (float), então para não haver perda de dados, devemos realizar o 'casting' (operação matemática de conversão de tipos de variável), quando misturamos int com float. Dependendo do cast realizado, você pode consumir mais memória e aumentar a precisão dos seus dados (float), ou pode diminuir a memória consumida mas perder um pouco a precisão dos dados (int), então tome cuidado para não estourar a memória do microcontrolador, nem perder precisão nos cálculos. Este algoritmo não está otimizado, tente melhorá-lo como forma de treino e memorização dos comandos.

3.3 Construção de Ohmímetro

Em seguida manipule o sistema e código para realizar as seguintes tarefas:

1. Abra o monitor serial e observe o valor medido com o potenciômetro;
2. Visualize o valor do potenciômetro graficamente (usando monitor serial e ploter por exemplo);
3. Modifique o código para implementar um Ohmímetro para a coleta da resistência associada e mostre no monitor serial;

Para tanto, você deve implementar um divisor de tensão em que um dos resistores é conhecido e outro é o que se deseja determinar. Esboce o diagrama elétrico, aplique a lei de Ohm no circuito e isole a resistência desconhecida para chegar na fórmula necessária para esta implementação. Então ajuste o código do Arduino para executar esta operação. Use o monitor serial para mostrar o valor da resistência encontrada. Teste para 2 ou mais resistores para verificar sua implementação.

4 Relatório

Inclua o nome e NºUSP de todos os integrantes do grupo. O relatório deve apresentar:

- Materiais usados
- Fotos da montagem
- Foto do código
- Descrição dos circuitos usados (com fotos da implementação)
- Descrição do código (com código implementado)
- Descrição dos procedimentos realizados

Referências

- 1 JUNIOR, R. C. dos R.; OLIVEIRA, L. R. de. *RESUMO DE ARDUINO, CÓDIGOS, SENSORES E ATUADORES*. 2021. Apostila para disciplina SEM0142 - Sensores e Sistemas de Medidas.