

INSTITUTO FEDERAL

Santa Catarina

Câmpus Tubarão

Programação com Arduino

Curso Superior de Tecnologia em Sistemas Embarcados

Professor: Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

Março de 2023

Cronograma

Encontro	Data	Nº Aulas	Conteúdo
1	29-fev.	04	Recepção e Apresentação do Unidade / Apresentação do Plano de Ensino / Avaliação Diagnóstica / Introdução a sistemas embarcados
2	02-mar.	04	Conceitos e Características e Aplicações de Sistemas Embarcados / Histórico de Sistemas Embarcados / Práticas com Arduino
3	07-mar.	04	Microcontroladores, Microprocessadores / Periféricos / Introdução ao Arduino / Introdução ao C
4	14-mar.	04	Introdução à Linguagens de Programação / Comunicação Serial
5	21-mar.	04	Entrada de Dados via Serial
6	28-mar.	04	Estruturas de Decisão
7	04-abr.	04	Entradas Digital e Analógica / Conversor A/D / Estruturas de Repetição
8	11-abr.	04	Estruturas Condicionais
9	18-abr.	04	Estruturas de Repetição
10	25-abr.	04	Avaliação 01

Cronograma

Encontro	Data	Nº Aulas	Conteúdo
11	02-mai.	04	Microcontroladores
12	09-mai.	04	Entradas e Saídas Digitais
13	16-mai.	04	Conversor Analógico-Digital
14	18-mai.	04	Sensores
15	23-mai.	04	Comunicação Serial
16	06-jun.	04	PWM
17	13-jun.	04	Temporizadores
18	20-jun.	04	Interrupções
19	27-jun.	04	Avaliação 02
20	04-jul.	04	Conselho de Classe / Atividades de Encerramento da UC
		80	

Pauta

- Entradas digitais;
- Entradas Analógicas;
- Conversor A/D;
- Estruturas de Repetição;

Entradas

Entradas

Entradas

```
pinMode(A1, INPUT);  
pinMode(9, INPUT);
```

❑ Entrada Analógica:

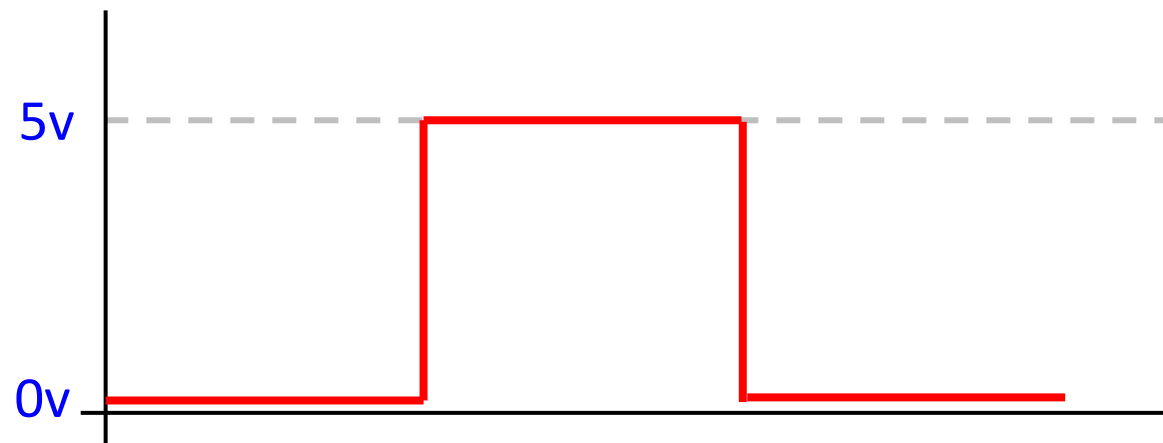
- ❑ Valores entre 0 – 1023;

❑ Entrada Digital

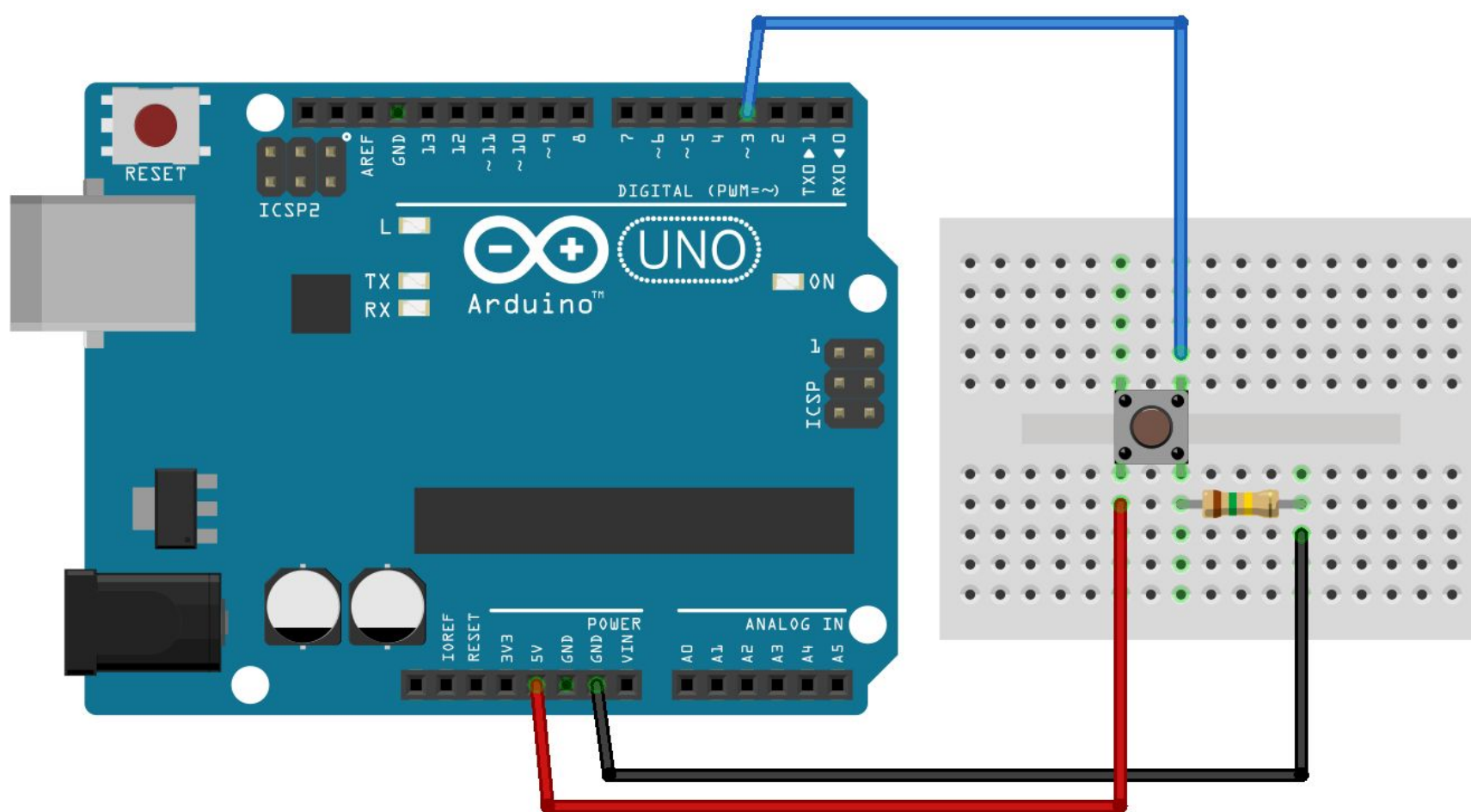
- ❑ Valor lógico 0 ou 1;

Entradas Digitais

- ❑ 14 Portas Digitais;
- ❑ 0 / 5v
 - ❑ LOW e HIGH;



Configurando Uma Entrada Digital



fritzing

Acionando um Led Utilizando Botão

```
#define pb 3
#define led 9
int pbValue;
int ledState;

void setup() {
  Serial.begin(9600);
  pinMode(pb, INPUT);
  pinMode(led, OUTPUT);
  ledState = LOW;
}
```

```
void loop(){
  pbValue = digitalRead(pb);
  Serial.println(pbValue);
  if(pbValue == 1){
    if(ledState == HIGH){
      ledState = LOW;
    } else {
      ledState = HIGH;
    }
    digitalWrite(led, ledState);
  }
  delay(300);
}
```

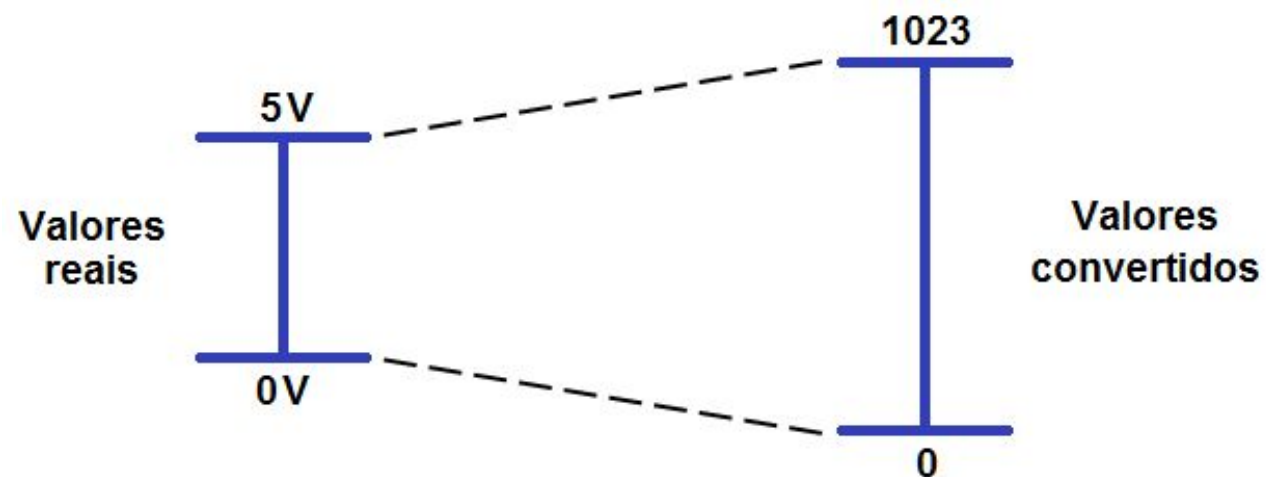
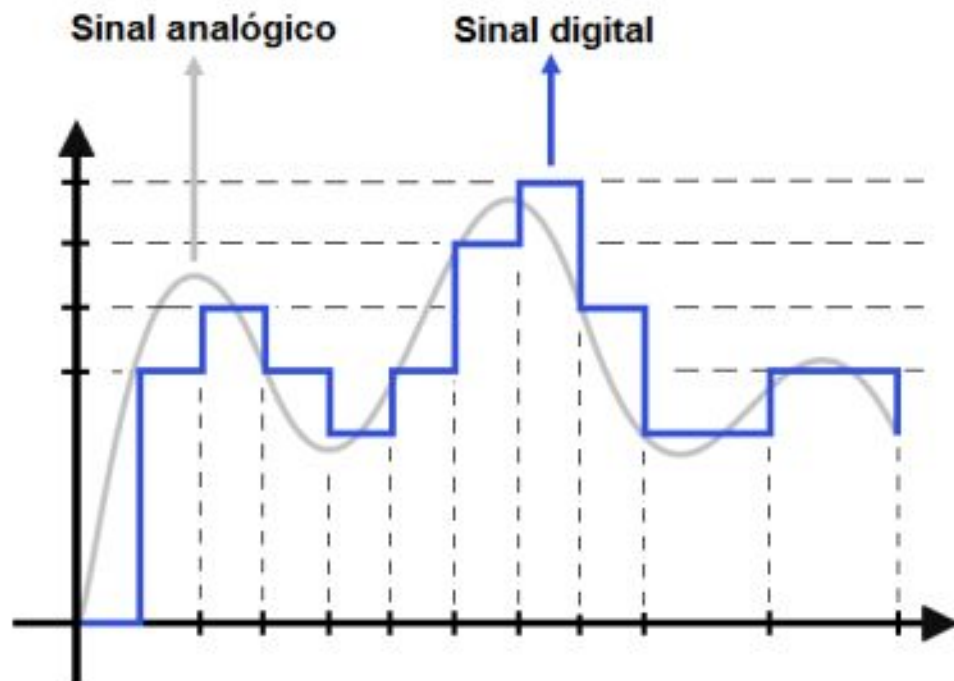
Atividade Com Botão e Leds

- ❑ Crie uma aplicação que faça piscar os três leds em sequência, alternando a cada 1 segundo.
 - Adicione um Botão ao seu projeto que quando pressionado, inverte o sentido que os leds estão piscando.

Portas Analógicas

6 Portas Analógicas

0 ~ 5V.



Conversor Analógico-Digital

Conversor Analógico-Digital

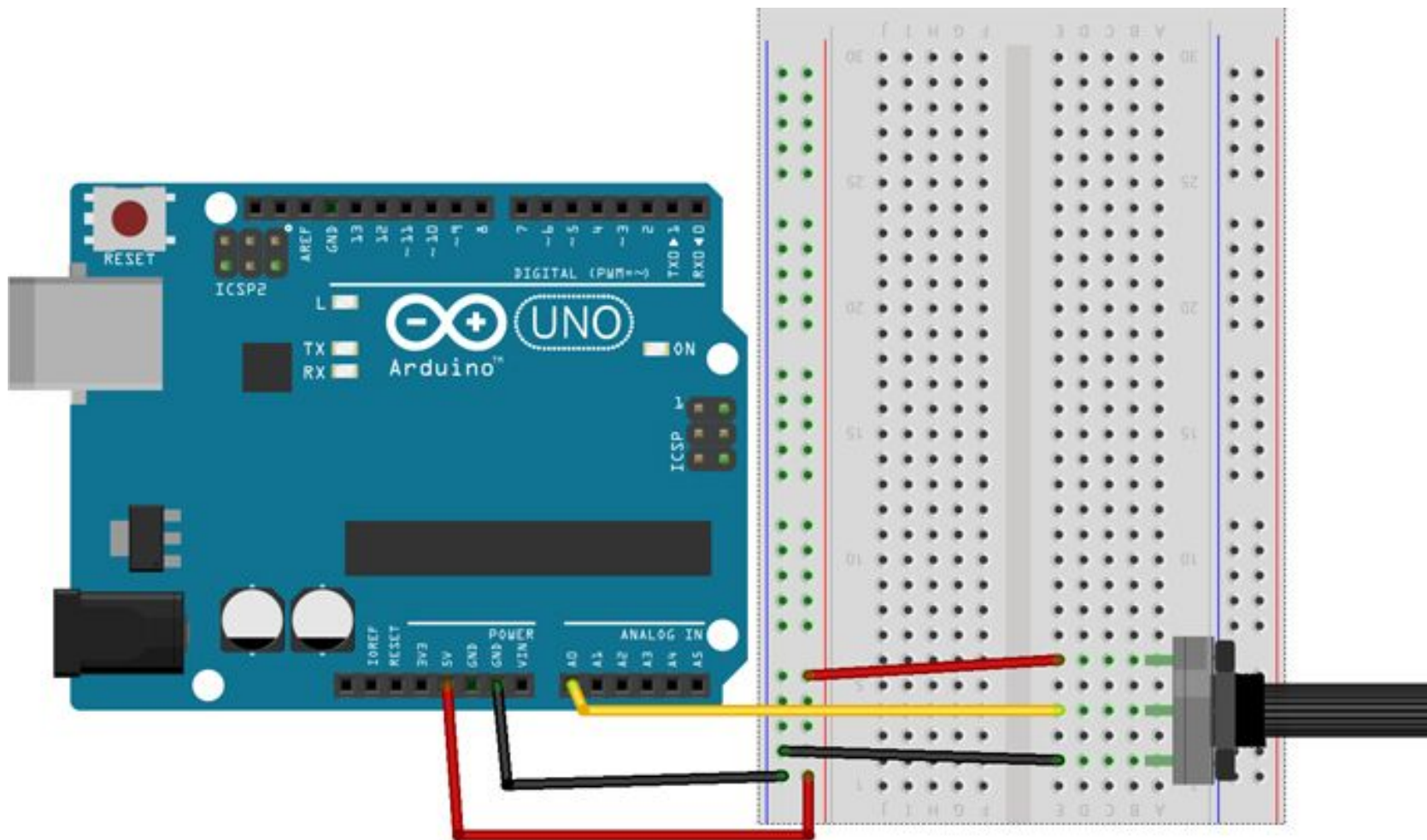
Conversor Analógico-Digital

- ❑ A eletrônica divide-se em dois tipos de sinais: digital e analógico.
- ❑ Infelizmente, nem tudo em nosso redor pode ser descrito de uma forma tão simples;
- ❑ Em alguns casos se faz necessário ler uma tensão numa faixa de 0 a 5V proporcionalmente.
- ❑ Esta é a abordagem analógica.
- ❑ **ATENÇÃO:** As entradas do Arduino só são compatíveis com tensões incluídas na seguinte faixa: 0-5V. Outras tensões poderão danificar a placa e até o computador ao qual ligar o circuito.

Conversor Analógico-Digital

- ❑ Para o processamento de sinais analógicos, é utilizado o chamado ADC: conversor analógico-digital;
- ❑ Este é um dos periféricos mais populares encontrados em microcontroladores;
- ❑ O seu trabalho é converter uma tensão aplicada na entrada do sistema num formato digital.
- ❑ Cada conversor possui uma quantidade de bits 8, 12, 16...
- ❑ Conforme a quantidade de bits, maior a precisão do conversor.

Exemplo de Uso Conversor Analógico-Digital



Exemplo de Uso Conversor Analógico-Digital

```
#define pot A0  
  
int potValue;  
  
void setup() {  
    Serial.begin(9600);  
    pinMode(pot, INPUT);  
}  
  
void loop(){  
    potValue = analogRead(pot);  
    Serial.println(potValue);  
    delay(1000);  
}
```


Prática com Conversor A/D




- ❑ Ajuste o programa com display e leds para apresentar na tela do display a leitura do potenciômetro;
- ❑ Conforme a leitura do potenciômetro faça o acionamento led conforme descrito abaixo:

Faixa de Valor	Leds Acionados
0 - 250	LED 1
251 - 500	LEDs 1 e 2
501 - 750	LEDs 1, 2 e 3
751 - 1023	LEDs 1, 2, 3 e 4

Estruturas de Repetição

Estruturas de Repetição

Laços for;

-  Laço de repetições até que a condição de saída especificada seja falsa;

Instrução do while;

-  Executa instruções enquanto a condição de saída seja falsa;

Instrução For

```
for([expressão inicial]; [condição]; [incremento]){  
    sequência de comandos  
}
```

- ❑ **Expressão inicial:** inicializa o laço de repetição;
- ❑ **Condição:** avaliada a cada laço, caso seja verdadeira o laço é executado, caso contrário a execução do laço termina;
- ❑ **Incremento:** atualiza a expressão e retorna para avaliação da condição;
- ❑ **Sequência de comandos:** executado a cada interação do laço de repetição;

Instrução For

```
int soma (){  
    int x, soma = 0;  
    for(x = 0; x < 30; x++){  
        soma += x;  
    }  
    return soma;  
}
```

Instrução Do While

```
do{  
    sequência de comandos  
} while([condição])
```

- ❑ **Sequência de comandos:** instrução executada pelo menos uma vez antes da verificação da condição.
- ❑ **Condição:** avaliada ao final de cada execução, caso seja verdadeira as instruções são executadas novamente, caso contrário a execução termina;

Instrução Do While

```
int main(){  
    int x = 0, soma = 0;  
    do{  
        soma += x;  
        x++;  
    } while (x < 5)  
    Serial.println(soma);  
}
```

Instrução While

```
while([condição]){  
    sequência de comandos  
}
```

- ❑ **Sequência de comandos:** instrução executada somente se a condição for satisfeita;
- ❑ **Condição:** avaliada no início de cada execução, caso seja verdadeira as instruções são executadas, caso contrário a execução termina;

Instrução While

```
int main(){  
    int x = 0, soma = 0;  
    while (x < 5){  
        soma += x;  
        x++;  
    }  
    Serial.println(soma);  
}
```

Break e Continue

```
void buscaLetra(char c){  
    char a[11] = "teste busca";  
    for(int i = 0; i < strlen(a); i++){  
        if(a[i] == c){  
            Serial.println(i);  
            break;  
        }  
    }  
}
```

```
while(i < 5){  
    i++;  
    if(i == 3){  
        continue;  
    }  
    n += 1;  
}  
  
Serial.println("%d", n);
```

Praticando com Estruturas de Repetição

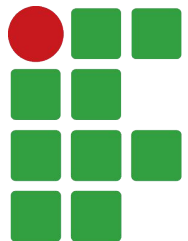
❑ Com base no conteúdo apresentado, crie as seguintes funções:

1) Crie um programa onde você tenha um Array de números:

- i. Peça ao usuário que informe um número que ele deseja buscar;
- ii. Percorra esse Array buscando o número informado;
- iii. Caso número seja encontrado imprimir a posição no array;
- iv. Caso o número não seja encontrado imprimir a mensagem “número não encontrado”.

2) Crie um programa que receba um número:

- i. A cada interação decrementa 1 do valor recebido e imprimir o valor restante;
- ii. Você deve parar de subtrair quando o valor for menor que 0 (zero);



INSTITUTO FEDERAL

Santa Catarina
Câmpus Tubarão

Obrigado!

Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

se.cst.tub@ifsc.edu.br