



OFICINA DE ROBÓTICA
INVENTAR E RECICLAR PARA EDUCAR
oficinaderobotica.ufsc.br

Oficina de Robótica

Programação Básica em Arduino – Aula 3

Execução:



LARM
Laboratório de Automação
e Robótica Móvel

Revisão

- ▶ Variáveis;
- ▶ Tipos de Dados(Int, Char, Float...);
- ▶ Operadores(Aritméticos, Relacionais, Lógicos);
- ▶ Monitor Serial;
- ▶ Resistores Pull Up e Pull Down;
- ▶ Comandos de Seleção...



LARM

Comandos de Seleção

▶ Seleção de múltipla escolha

- Na seleção de múltipla escolha é possível comparar vários valores.
- **No Arduino o comando de seleção de múltipla escolha é:**

```
switch (valor) {  
    case x: cmd1;  
        break;  
    case y: cmd2;  
        break;  
    default: cmd3;  
}
```

- **onde:**
 - *valor* – é um dado a ser avaliado. É representado por uma variável de memória.
 - *cmd_x* – comando a ser executado.
 - *case* – indica a opção a ser executada.
 - *default* – comando padrão que deverá ser executado se nenhuma outra escolha (*case*) tiver sido selecionada.



LARM

Comandos de Seleção

- ▶ Seleção de múltipla escolha
 - Acendendo e apagando leds pelo monitor serial

```
selecao_multipla_escolha
#define LED_VERDE 2
#define LED_AMARELO 3
#define LED_VERMELHO 4

char opcao;

void setup()
{
  pinMode(LED_VERDE, OUTPUT);
  pinMode(LED_AMARELO, OUTPUT);
  pinMode(LED_VERMELHO, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()) {
    opcao = Serial.read();

    switch (opcao) {
      case 'G': digitalWrite(LED_VERDE, HIGH);
                break;
      case 'g': digitalWrite(LED_VERDE, LOW);
                break;
      case 'Y': digitalWrite(LED_AMARELO, HIGH);
                break;
      case 'y': digitalWrite(LED_AMARELO, LOW);
                break;
      case 'R': digitalWrite(LED_VERMELHO, HIGH);
                break;
      case 'r': digitalWrite(LED_VERMELHO, LOW);
                break;
      default: Serial.println("Opcao Invalida.");
    }
  }
}
```



LARM

Entrada Digital de Dados

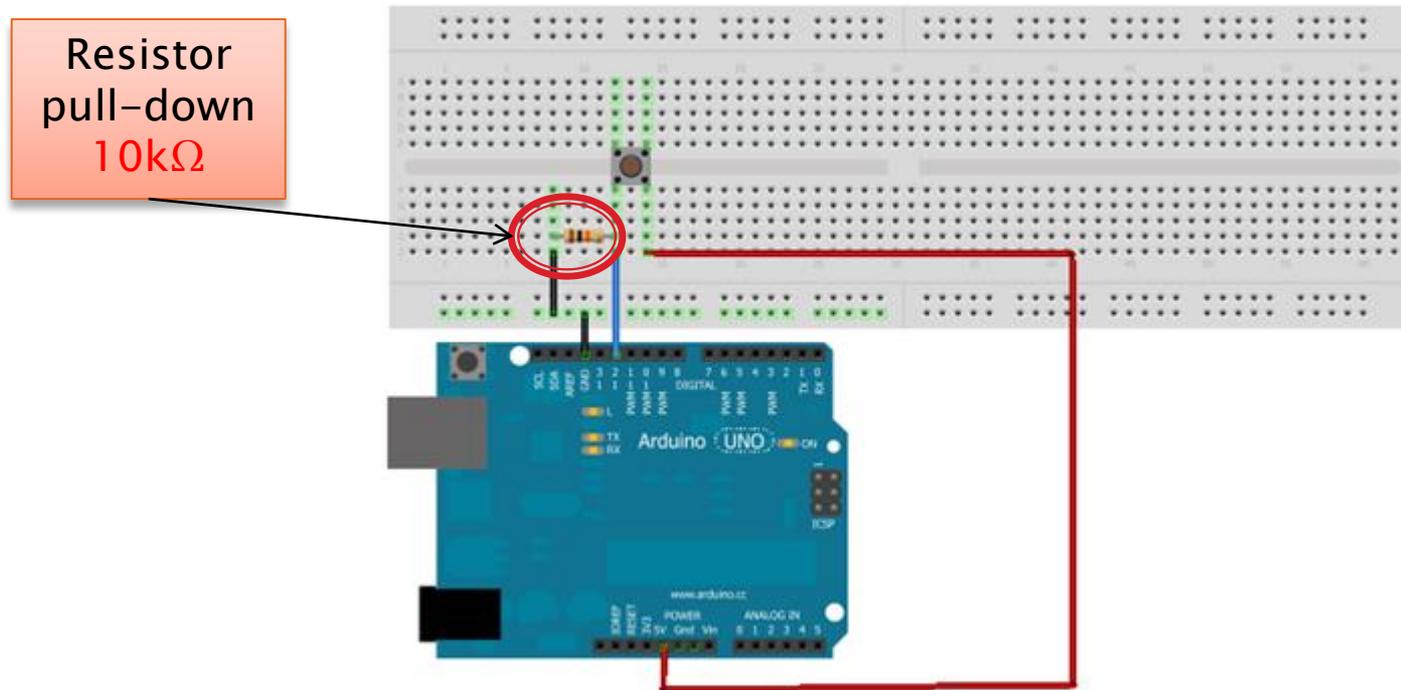
- ▶ Verificando o estado de um botão
 - Para verificar se um botão está **pressionado ou não** basta conectá-lo a uma **porta digital**.
 - Para que um circuito com botão funcione adequadamente é necessário o uso de resistores *pull-down* ou *pull-up*.
 - Os resistores *pull-down* e *pull-up* **garantem que os níveis lógicos estarão próximos às tensões esperadas.**



LARM

Entrada Digital de Dados

- ▶ Leitura de um botão com resistor *pull-down*
 - Ligação na protoboard



Entrada Digital de Dados

- ▶ Leitura de um botão com resistor *pull-down*
 - Programa

```
pull-down
#define BOTAO 12

void setup()
{
  pinMode(BOTAO, INPUT);
  Serial.begin(9600);
}

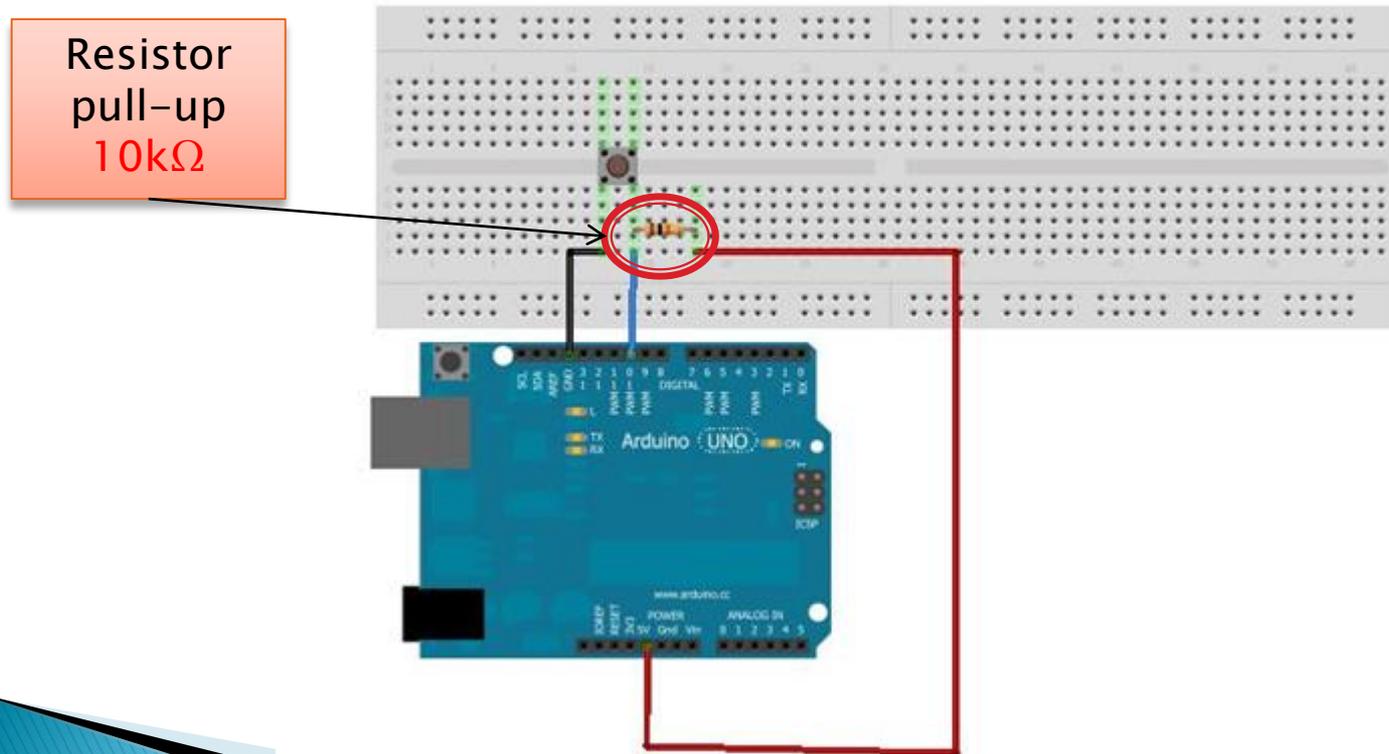
void loop()
{
  if (digitalRead(BOTAO) == HIGH)
    Serial.println("Botao pressionado!");
}
```



LARM

Entrada Digital de Dados

- ▶ Leitura de um botão com resistor *pull-up*
 - Ligação na protoboard



Entrada Digital de Dados

- ▶ Leitura de um botão com resistor *pull-up*
 - Programa

```
pullup
#define BOTAO 10

void setup()
{
  pinMode(BOTAO, INPUT);
  Serial.begin(9600);
}

void loop()
{
  if (digitalRead(BOTAO) == LOW)
    Serial.println("Botao pressionado!");
}
```



LARM

Entrada Digital de Dados

▶ Nota

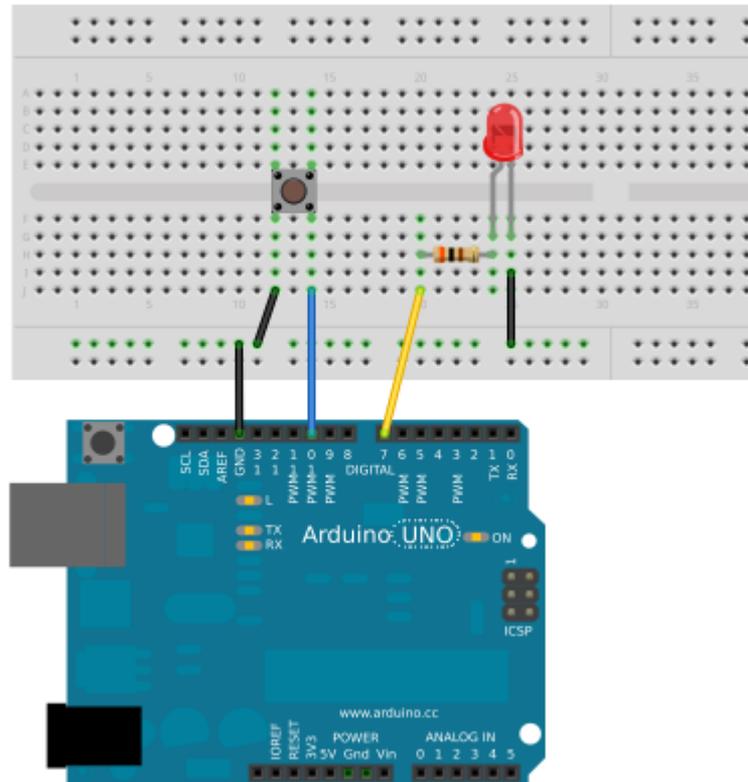
- O Arduino possui resistores *pull-up* nas portas digitais.
- Para **ativar** os resistores *pull-up* de uma porta digital **basta defini-la como entrada e colocá-la em nível alto (HIGH)**.
 - `pinMode(pin, INPUT)`
 - `digitalWrite(pin, HIGH)`
- Para **desativar** os resistores *pull-up* de uma porta digital **basta colocá-la em nível baixo**.
 - `digitalWrite(pin, LOW)`



LARM

Entrada Digital de Dados

- ▶ Ativando o *pull-up* de uma porta digital
 - Ligação na protoboard



Entrada Digital de Dados

- ▶ Ativando o *pull-up* de uma porta digital
 - Programa

```
pullup_arduino
#define BOTAO 10
#define LED 7

void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(BOTAO, INPUT);
  // ativa o resistor pull-up
  digitalWrite(BOTAO, HIGH);
}

void loop()
{
  if (digitalRead(BOTAO) == LOW)
    digitalWrite(LED, HIGH);
  else
    digitalWrite(LED, LOW);
}
```



Entrada Digital de Dados

▶ Nota

- O Arduino possui uma constante chamada *INPUT_PULLUP* que define que a porta será de entrada e o resistor *pull-up* da mesma será ativado.
- **Exemplo:**

```
void setup()  
{  
  pinMode(10, INPUT_PULLUP);  
}
```



Define a porta 10 como entrada de dados e ativa o resistor pull-up.



LARM

Exercícios

- ▶ Fazer um contador, com um botão pra incremento e um botão para decremento de uma variável. O valor da variável deve ser mostrado em tela.
- ▶ Criar um controle de LEDs (vermelho, amarelo e verde), onde o botão que for pressionado deve acender o LED correspondente a ele, e mostrar em tela qual LED está acesso.
- ▶ Criar um programa que faça leitura qualquer do teclado e acenda o led correspondente. Ex. 'A' acende led amarelo.



LARM