



Clube de Robótica IFTM Arduíno e Raspberry

# *Curso de Robótica Básica*

## 2º Semestre/2019

Professor Abel Alves

Professor João Marcos de Oliveira Machado

Professor Samuel Oliveira Serqueira



# Mãos na Massa

"Para aprender a fazer algo, deve-se começar tentar, pois é aos poucos que se aprende, no próprio processo de fazer."

Bernard Charlot





## Mão na Massa

Existem 5 coisas que precisamos saber fazer com o Arduino;

acender um led, piscar um led, criar uma função, enviar dados e receber dados.



## Liga LED

O mais simples e mais óbvio, acender um led. O pino digital 13 do Arduíno vem com um LED embutido e é possível fazê-lo acender usando programação.



# Liga LED

```
/****** Liga Led *****\
```

// O setup é executada uma vez quando é pressionado o botão reset ou a alimentação, é criado sem retorno (void) ou parâmetros ().

```
void setup() {  
    pinMode(13, OUTPUT); //Seta o  
    modo do pino 13 como saída.  
}
```



# Liga LED

// O loop é executada enquanto houver alimentação, é criado sem retorno (void) ou parâmetros ().

```
void loop() {  
    digitalWrite(13, HIGH); // Escreva de  
    forma digital no 13, Alto - Acende o  
    LED - se trocar HIGH por LOW ele  
    apagará  
}
```



# Liga LED

/\*\*\*\*\* Liga Led \*\*\*\*\*/

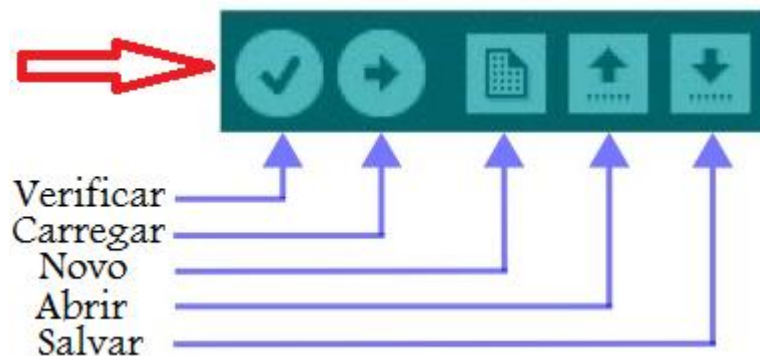
```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(13, HIGH);  
}
```



# Liga LED

Compile o programa clicando no botão Verificar do ambiente de desenvolvimento, para ver se não existe nenhum erro de código.

O botão é o seguinte:



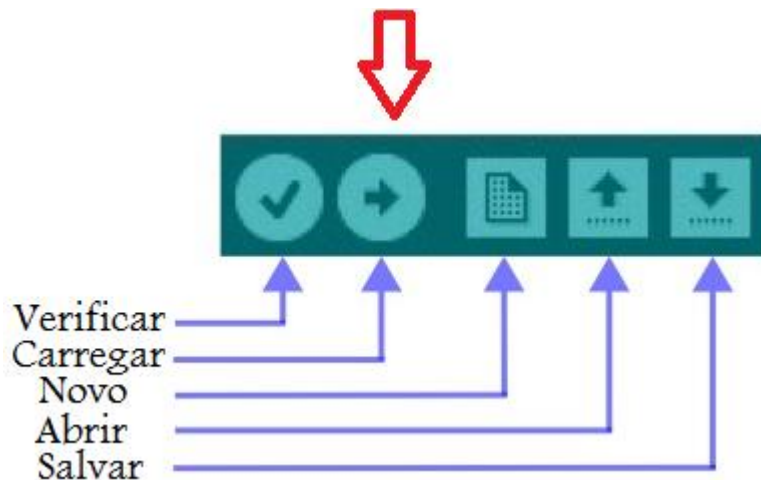




# Liga LED

Se na barra inferior não aparecer nenhuma mensagem de erro, o programa está pronto para ser enviado ao Arduíno.

Para tanto, basta clicar no botão Carregar que é o seguinte:





*Liga LED*

*Experimentar no lugar de*

*LOW/HIGH:*

*0, 1 e 2*

*false e true*



# *Pisca LED*

*O segundo mais simples de todos!  
Piscar um led. Basta alternar o pino  
digital entre ligado (HIGH) e desligado  
(LOW), aguardando um pequeno  
intervalo de tempo.*



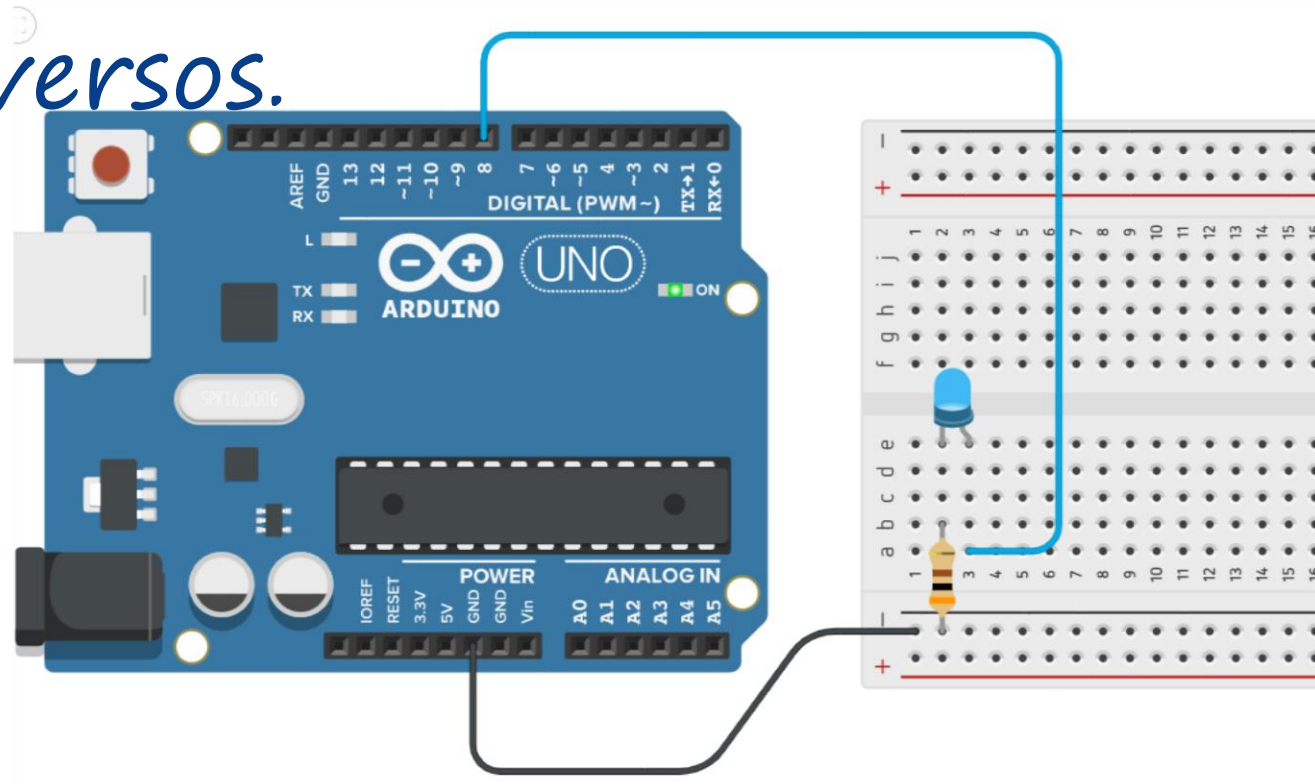
# Pisca LED

Componentes:

1 Led Vermelho

1 Resistor de  $300\Omega$

cabos diversos.





# Pisca LED

/\*\*\*\*\* Pisca Led \*\*\*\*\*/

Acende um LED por um segundo, então apaga por um segundo, repetidamente.

\\*\*\*\*\*/

//constantes (constantes não mudam de valor durante a execução do programa)

const int led = 8; //Cria a constante led, do tipo inteiro e com referencia ao pino 8

void setup() {

pinMode(led, OUTPUT); //Seta o modo do pino 8, com o nome de "led" como saída.

}



# Pisca LED

```
void loop() {  
  digitalWrite(led, HIGH); // Acende o LED (HIGH é o  
  nível de tensão - Alto)  
  delay(1000); // Aguarda 1 segundo (1.000  
  milissegundos)  
  digitalWrite(led, LOW); // Apaga o LED (LOW é o  
  outro nível de tensão - Baixo)  
  delay(1000); // Aguarda 1 segundo (1.000  
  milissegundos)  
}
```



# Pisca LED

```
const int led = 8;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```



# Pisca LED

Há várias maneiras de piscar um LED, porém vamos ver três.

A tradicional, que foi vista acima, a mais curta e a sem atraso (sem delay).





# Pisca LED

## A mais curta

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(13,!digitalRead(13));  
  delay(1000);  
}
```



# Pisca LED

Sem Atraso.

//Definição das constantes

const int pinoLed = 13;

const int intervalo = 500;

//Definição das variáveis

unsigned long passouTempo = 0;

unsigned long tempoAtual = 0;

int estadoLed = LOW;



# Pisca LED

```
void setup(){  
  pinMode (pinoLed, OUTPUT);  
  digitalWrite(pinoLed, estadoLed);  
}
```



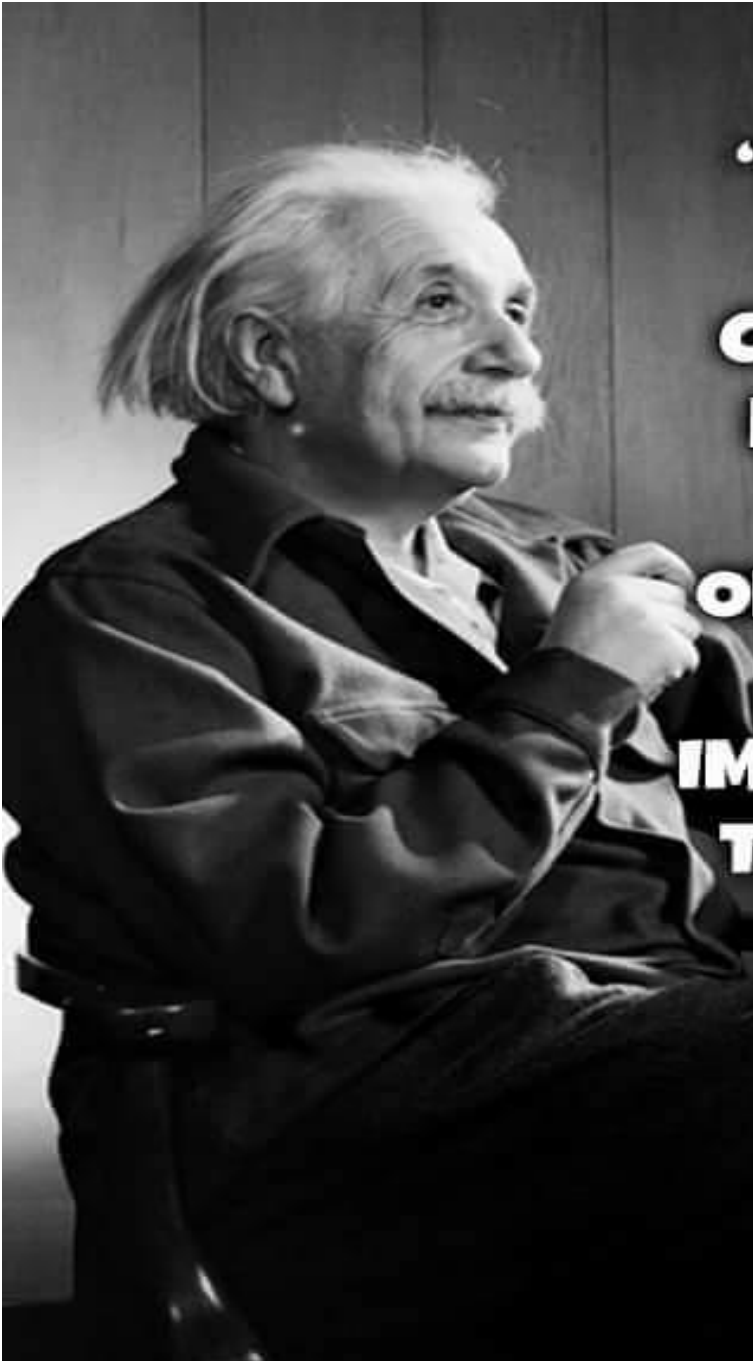
# Pisca LED

```
void loop() {  
    tempoAtual = millis();  
    if (tempoAtual - passouTempo >=  
intervalo){  
        passouTempo = tempoAtual;  
        if (estadoLed == LOW){  
            estadoLed = HIGH;  
        }  
    }  
}
```



# *Pisca LED*

```
else {  
    estadoLed = LOW;  
}  
}  
digitalWrite(pinoLed, estadoLed);  
}
```



**“EU TENTEI 99 VEZES E  
FALHEI, MAS NA  
CENTÉSIMA TENTATIVA  
EU CONSEGUI. NUNCA  
DESISTA DE SEUS  
OBJETIVOS! MESMO QUE  
ESSES PAREÇAM  
IMPOSSÍVEIS, A PRÓXIMA  
TENTATIVA PODE SER A  
VITORIOSA.”**

**CRAR**



# Função Pisca LED

Funções são essenciais para reaproveitamento do código, ao invés de ficar digitando várias vezes a mesma coisa apenas invoque a função alterando o estado do pino de HIGH para LOW e vice-versa.



# Função Pisca LED

/\*\*\*\*\*\

\*\*\*\*\* Função Pisca Led \*\*\*\*\*

\\*\*\*\*\*/

//constantes (constantes não mudam  
de valor durante a execução do  
programa)

const int led = 8;

const int atraso = 1000; // valor para  
o delay





# Função Pisca LED

//variáveis(variáveis mudam de valor durante a execução do programa

int valor = HIGH;//para acender ou apagar o led

```
void setup() {  
    pinMode(led, OUTPUT);  
}
```



# Função Pisca LED

```
void loop() {  
    funPisca(); // chama a função  
    delay(atraso); // Aguarda atraso de tempo  
}
```

```
void funPisca() { // Cria a função sem retorno
```



# Função Pisca LED

```
if(valor == HIGH)//Se valor for igual a  
HIGH
```

```
    valor = LOW;//Valor recebe LOW  
else//Senão
```

```
    valor = HIGH;//Valor recebe HIGH  
    digitalWrite(led, valor);//escreva  
digitalmente o valor no pino led  
}
```



# Função Pisca LED

```
const int led = 8;  
const int atraso = 1000;  
int valor = HIGH;  
  
void setup() {  
  pinMode(led, OUTPUT);  
}  
void loop() {  
  funPisca();  
  delay(atraso);  
}
```

```
void funPisca() {  
  if(valor == HIGH)  
    valor = LOW;  
  else  
    valor = HIGH;  
  digitalWrite(led, valor);  
}
```



# Enviar dados pela porta serial

Com o Arduíno é possível enviar dados dele para o computador. Neste código o Arduíno faz uma contagem a partir da variável “armDados” e imprime no Monitor Serial.



# Enviar dados pela porta serial

/\*\*\*\*\* Envia Dados\*\*\*\*\*/

int armDados = 0; // variável que será usada para armazenar a informação e enviar para o PC via porta serial



# Enviar dados pela porta serial

```
void setup() {  
    Serial.begin(9600); // inicia o  
    método Serial (.begin) para  
    comunicação com o PC ajusta a taxa de  
    transferência de dados para 9600 baud  
    ou bps (bits por segundo)  
}
```



# Enviar dados pela porta serial

```
void loop() {  
    Serial.println (armDados); //  
    enviar pela porta serial o valor da  
    variável armDados  
    delay(1000);  
    armDados++; // incrementa em 1  
}
```





# Enviar dados pela porta serial

```
int armDados = 0;
void setup() {
    Serial.begin(9600);
}
void loop() {
    Serial.println (armDados);
    delay(1000);
    armDados ++;
}
```



# Receber dados pela porta serial

Também é possível enviar dados do computador para o Arduino. Ao abrir o Monitor Serial, digitar a letra “a” (minúscula) e clicar em Enviar. O LED do pino digital acenderá e imprime “Aceso”, se enviar a letra “a” novamente vai apagar e imprimir “Apagado”.



# Receber dados pela porta serial

```
/******Recebe Dados*****\
```

```
//constantes
```

```
const int ledPin = 8;
```

```
//variáveis
```

```
char tecla; //variável do tipo char
```

```
int estado = HIGH; // variável que  
guarda o estado para o pino digital.
```



# Receber dados pela porta serial

```
void setup() {
```

```
    Serial.begin(9600); // inicia a  
    comunicação serial onde escreve os  
    dados
```

```
    pinMode(ledPin, OUTPUT);
```

```
    digitalWrite(ledPin, estado); // coloca  
    o valor de estado no pino ledPin
```

```
}
```



# Receber dados pela porta serial

```
void loop() {  
    if(Serial.available() > 0) { //o método "available"  
        verifica se há dados a serem transmitidos pela serial  
        retornando "true" ou "1" se sim ou "false" ou "0"  
        se não.  
    }
```



# Receber dados pela porta serial

`tecla = Serial.read();` // armazena em "tecla" o retorno do método `read()`, esse método lê um valor que é escrito na porta serial

`if(tecla == 'a') {` //se for igual ao caractere 'a'

`if(estado == HIGH) {` //se o LED estiver aceso  
`estado = LOW;` // apague



# Receber dados pela porta serial

```
Serial.println("Apagou"); //escreve mensagem  
no Monitor Serial  
{  
else {//senão  
    estado = HIGH;//estado recebe alto
```



# Receber dados pela porta serial

```
Serial.println("Acendeu");  
//escreve mensagem no Monitor Serial  
    }  
}  
}  
digitalWrite(ledPin, estado);  
}
```





# Receber dados pela porta serial

```
const int ledPin = 8;
char tecla;
int estado = HIGH;
void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, estado);
}
void loop() {
    if(Serial.available() > 0) {
        tecla = Serial.read();
        if(tecla == 'a') {
```

```
            if(estado == HIGH) {
                estado = LOW;
                Serial.println("Apagou");
            }
            else {
                estado = HIGH;
                Serial.println("Acendeu");
            }
        }
        digitalWrite(ledPin, estado);
    }
}
```



# Onde te encontro?

- ✓ [contatorobotica.criar@gmail.com](mailto:contatorobotica.criar@gmail.com)
- ✓ [arduinoemudi.blogspot.com](http://arduinoemudi.blogspot.com)