

# Guia de Referência Rápida Arduino

## Estrutura principal

```
void setup() {  
    // roda uma vez na inicialização  
}  
void loop() {  
    // roda repetidamente  
}
```

## Estruturas de controle

```
if (condicao) {  
    // condição true  
}else{  
    // condição false  
}  
  
for (int i = 0; i < 10; i++) {  
    // alguma coisa  
}  
  
while (condicao) {  
    // executa enquanto condição true  
}  
  
switch (variavel) {  
    case 1:  
        // caso variável igual a 1  
        break;  
    case 2:  
        // caso variável igual a 2  
        break;  
    default:  
        // caso não cair em um dos anteriores  
}  
  
do {  
    // executa ao menos uma vez e  
    // depois se condição true  
} while (condicao);
```

## Constantes

HIGH e LOW  
INPUT e OUTPUT  
true e false

## Comentários

```
// uma linha  
/* várias  
linhas */
```

## I/O Digital

pinMode(pino, INPUT ou OUTPUT); // define um pino como entrada ou saída  
digitalWrite(pino, HIGH ou LOW); // para pinos setados como OUTPUT  
digitalRead(pino); // para pinos setados como INPUT, retorna HIGH ou LOW

## I/O Analógico

analogReference(modo); /\* modo pode ser DEFAULT, INTERNAL(1.1v),  
INTERNAL1V1(mega), INTERNAL2V56(mega) e EXTERNAL(0 a 5v do AREF)\*/  
analogRead(pino); // retorna um valor entre 0 e 1023 (0 a 5v) do pino lido  
analogWrite(pino, 0 a 255); // gera uma onda que simula uma saída analógica

## I/O Avançado

tone(pino, frequencia); usa a saída PWM para gerar  
uma onda na frequência especificada  
tone(pino, frequencia, duração); // mesmo que o acima,  
porém com a duração em ms  
noTone(pino); // interrompe a execução dos comandos acima  
pulseIn(pino, valor); /\* lê a duração entre o pulso do  
valor em microssegundos valor pode ser HIGH ou LOW \*/  
pulseIn(pino, valor, timeout); //mesmo que o acima,  
porém com timeout de microssegundos

## Serial

Serial.begin(9600); //inicializa a serial  
Serial.print(dados); //envia dados pela serial  
Serial.println(dados); //mesmo que acima, mas  
com uma quebra de linha  
Serial.available(); //true se existem dados a serem lidos  
Serial.read(); //lê valores da serial  
Serial.parseFloat(); //lê o primeiro float válido da serial  
Serial.parseInt(); //lê o primeiro inteiro válido da serial

## Tempo

millis() // quantos milissegundos desde que ligou  
micros() // quantos microssegundos desde que ligou  
delay(milissegundos) // interrompe o programa por 'x' milissegundos  
delayMicroseconds(microssegundos) // interrompe o programa por 'x'  
microssegundos

## Randômicos

randomSeed(semente); //inicializa o gerador de números randômicos  
random(max); //gera um número entre 0 e max  
random(min, max); //gera um número entre min e max

## Tipos de dados

Tipo	Valores	Tamanho
boolean	true ou false	8 bit
byte	0 to 255	8 bit
char	-128 até 127	8 bit
unsigned char	0 a 255	8 bit
word	0 to 65535	16 bit
int / short	-32,768 to 32,767	16 bit
unsigned int	0 to 65,535	16 bit
long	-2,147,483,648 até 2,147,483,647	32 bit
unsigned long	0 to 4,294,967,295	32 bit
float / double	-3.4028235E38 até 3.4028235E38	32 bit
string	array de char	8 bit x tamanho

## Matemática

min(x, y); //retorna o menor  
max(x, y); //retorna o maior  
abs(x); //retorna o valor absoluto  
pow(base, expoente); //calcula potência  
sqrt(x); // calcula a raiz quadrada  
sin(radiano); //calcula o seno do ângulo  
cos(radiano); //calcula o cosseno do ângulo  
tan(radiano); //calcula a tangente do ângulo  
constrain(x, a, b); /\* retorna x se x entre a e b,  
retorna a se x < a ou b se x > b \*/  
map(valor, doMenor, doMaior, paraMenor, paraMaior);  
//remapeia valor 'doX' para 'paraX'

## IRQs externas

attachInterrupt(irq, função, modo);  
/\* irq pode ser 0 ou 1 (pino 2 ou 3 no Uno),  
função é o nome da função a ser chamada e  
modo pode ser CHANGE, RISING, FALLING, LOW ou HIGH(Mega) \*/  
detachInterrupt(irq); //desativa uma irq setada  
noInterrupts(); //pausa as IRQs  
interrupts(); //continua as IRQs

## Outros

#define ledPin 3 //cria variáveis usadas em tempo de  
compilação  
#include <avr/pgmspace.h> //inclui uma lib