

Arduino

Funções Nativas

`void setup()` Primeira função a ser executada ao ligar o Arduino, tudo o que está dentro dela é executado apenas 1 vez.
`void loop()` Função que se repete enquanto o Arduino estiver ligado.

Entradas e saídas Digitais

`pinMode()` Configura o pino especificado para funcionar como uma entrada ou saída.
Sintaxe: `pinMode(pino, modo)`
pino: Número do pino no qual se deseja configurar
modo: Modo do pino, pode ser:
- `INPUT`: Modo de entrada
- `OUTPUT`: Modo de saída
- `INPUT_PULLUP`: Modo de entrada com resistor de pull-up ativado.

`digitalRead()` Lê o valor de um pino digital especificado, que pode ser HIGH ou LOW.
Sintaxe: `digitalRead(pino)`
pino: Número do pino no qual se deseja ler uma informação

`digitalWrite()` Aciona um valor HIGH ou LOW em um pino digital.
Sintaxe: `digitalWrite(pino, valor)`
pino: Número do pino no qual se deseja manipular
valor: HIGH (Ligado) ou LOW (Desligado)

Entradas e Saídas Analógicas

`analogRead()` Lê o valor de um pino analógico especificado.
Sintaxe: `analogRead(pino)`
pino: Número do pino no qual se deseja ler o valor analógico

`analogWrite()` Aciona uma onda PWM em um pino.
Sintaxe: `analogWrite(pino, valor)`
pino: Número do pino para se acionar o PWM
valor: duty cycle: 0 (Sempre desligado) ou 255 (Sempre ligado)

Comunicação

Serial: Usada para comunicação entre uma placa Arduino e um computador ou outros dispositivos. Todas as placas Arduino possuem pelo menos uma porta serial.

Funções:

- `if(Serial)`
- `available()`
- `begin()`
- `end()`
- `flush()`
- `print()`: Imprime um valor na Serial
- `println()`: Imprime um valor na Serial e pula uma linha após
- `read()`
- `readBytes()`
- `readString()`
- `write()`

Funções Temporizadoras

`delay()` Pausa o programa por uma quantidade especificada de tempo (em milissegundos). Cada segundo equivale a 1000 milissegundos.
Sintaxe: `delay(ms)`
ms: Número de milissegundos para pausar o programa.

`delayMicroseconds()` Pausa o programa pela quantidade de tempo especificada como parâmetro (em microssegundos). Há mil microssegundos em um milissegundo, e um milhão de microssegundos em um segundo.
Sintaxe: `delayMicroseconds(us)`
us: Número em microssegundos para pausar o programa.

`micros()` Retorna o número de microssegundos passados desde que a placa Arduino começou a executar o programa atual.
Sintaxe: `time = micros()`
Esse número irá sofrer overflow (chegar ao maior número possível e então voltar pra zero), após aproximadamente 70 minutos. Em placas Arduino 16 MHz (ex. UNO e Nano), essa função possui uma resolução de quatro microssegundos (isto é, o número retornado é sempre um múltiplo de quatro). Em placas Arduino 8 MHz (ex. LilyPad), essa função possui uma resolução de oito microssegundos.

`millis()` Retorna o número de milissegundos passados desde que a placa Arduino começou a executar o programa atual.
Sintaxe: `time = millis()`
Esse número irá sofrer overflow (chegar ao maior número possível e então voltar pra zero), após aproximadamente 50 dias.

Funções Matemáticas

`abs()` Calcula o módulo (ou valor absoluto) de um número.
Sintaxe: `abs(x)`
x: Número do qual se deseja saber o módulo
Retorna: x se x é maior ou igual a 0.
Retorna: -x se x é menor que 0.

`map()` Remapeia um número de um intervalo para outro.
Sintaxe: `valor = map(valor, deMenor, deMaior, paraMenor, paraMaior)`
valor: o número a ser mapeado
deMenor: o menor limite do intervalo atual do valor
deMaior: o maior limite do intervalo atual do valor
paraMenor: o menor limite do intervalo alvo
paraMaior: o maior limite do intervalo alvo

`max()` Calcula o maior de dois números.
Sintaxe: `max(x, y)`
x: o primeiro número, qualquer tipo de dado
y: o segundo número, qualquer tipo de dado
Exemplo: `sensVal = max(sensVal, 20);`

`min()` Calcula o menor de dois números.
Sintaxe: `min(x, y)`
x: o primeiro número, qualquer tipo de dado
y: o segundo número, qualquer tipo de dado
Exemplo: `sensVal = min(sensVal, 100);`

`sqrt()` Calcula a raiz quadrada de um número.
Sintaxe: `sqrt(x)`
x: o número a se calcular a raiz quadrada, qualquer tipo de dados

Números Aleatórios

`random()` A função `random()` gera números pseudoaleatórios.
Sintaxe: `random(max)` ou `random(min, max)`
min: menor limite do valor aleatório, inclusivo e opcional (long)
max: maior limite do valor aleatório, exclusivo (long)