

Funções de entrada e saída (I/O):

`pinMode(pino, modo)` -> Define o pino como entrada ou saída.

- **INPUT:** Lê valores de tensão
- **OUTPUT:** Fornece valores de tensão

Leitura e escrita Digitais:

`digitalRead(pino)` -> Lê o nível de tensão do pino e retorna 1 se estiver em nível alto (5V) ou 0 em baixo (0V).

`digitalWrite(pino, nível)` -> Escreve um valor de tensão no pino..

- **HIGH:** Alto (5V)
- **LOW:** Baixo (0V)

Leitura e escrita Analógicas:

`analogRead(pino)` -> Lê o nível de tensão no pino e retorna de 0 (0V) até 1023 (5V).

`analogWrite(pino, valor)` -> Escreve um valor de tensão analógica (PWM) no pino.

- **0:** 0% de largura de pulso (0V)
- **até 255:** 100% de largura de pulso (5V)

Funções do LCD:

`lcd.begin(colunas, linhas)` -> Indica para o programa quantas colunas e linhas o LCD possui.

`lcd.print(variável ou string)` -> Imprime na tela um texto ou uma variável.

`lcd.clear()` -> Limpa a tela e volta o cursor para a posição (0,0).

`lcd.setCursor(coluna, linha)` -> Move o cursor para a coluna e linha desejadas.

Funções do termistor:

`thermistor(pino, resNominal, coefBeta, resSérie)` -> Define os parâmetros do NTC que está sendo utilizado.

`pino` = Pino em que o NTC está conectado
`resNominal` = Resistência nominal do NTC a 25°C
`coefBeta` = Coeficiente Beta do NTC
`resSérie` = Valor da resistência em série com o NTC

`thermistor.read()` -> Retorna o valor da temperatura, em graus Celsius

Variáveis:

int: número inteiro

float: ponto flutuante (número decimal)

bool: verdadeiro ou falso (true/false)

long: variável de 4 bytes

unsigned int: números naturais

Comunicação Serial:

Serial.begin(velocidade):

Inicia a comunicação do Arduino com o computador

Serial.print(variável):

Imprime no monitor serial alguma informação

Serial.println(variável ou texto):

Imprime no monitor serial alguma informação separada em uma linha

Outras funções:

tone(pino, frequência): Faz com que o buzzer vibre numa frequência definida.

sin(valor em radianos): Retorna o valor da função seno para determinado ângulo.

delay(tempo): Para o programa por um tempo determinado.

millis(tempo): Retorna o tempo desde que o Arduino foi ligado, em milissegundos.