

Arduíno - 1º Capítulo

Paula Laysa Amorim Silva

Estudo Dirigido – Anotações

Alguns comandos:

`moveforward();` = andar

`turnLeft();` = virar a esquerda

`turnRight();` = virar a direita

```
while (notDone()); {  
    moveforward(); = enquanto não terminar, ande.  
}
```

`if (is PathLeft())` =se: condição

`}else {` = senão: condição

Transmitindo dados

A função `setup()` é chamada quando um sketch inicia. A função `setup()` será executada apenas uma vez, após a placa ser alimentada ou acontecer um reset.

A função `loop()` faz precisamente o que o seu nome sugere, e repete-se consecutivamente enquanto a placa estiver ligada, permitindo o seu programa mudar e responder a essas mudanças.

O tipo de dado, é o tipo de informação que uma variável pode guardar.

A variável é utilizada para guardar dados de um programa.

Uma variável 'byte' armazena valores numéricos de 8-bit sem sinal, de 0 a 255.

Um tipo de dado usado para armazenar um caractere. Caracteres literais são escritos em aspas simples, dessa forma: 'A'.

Static é quando uma variável tem um valor permanente.

Atribuir um valor significa armazenar para usar depois. (=)

O ++ aumentará o valor de uma variável para mais um.

`Deley()` pausa o programa por uma quantidade especificada de tempo (em milissegundos).

Serial é usada para comunicação entre uma placa Arduino e um computador ou outros dispositivos

`Serial.Begin()` Configura a taxa de transferência em bits por segundo (baud rate) para transmissão serial.

`Serial.print()` imprime dados na mesma linha.

`Serial.println()` imprime dados em linhas distintas.

Recebendo dados

`Serial.available()` Retorna o número de bytes (caracteres) disponíveis para leitura da porta serial. Esses são dados que já chegaram e foram guardados no buffer de recebimento

`serial.read()` Lê dados recebidos na porta serial.

If para quando a condição for verdadeira e Else para quando for falsa. (ex.: se você não andar, você irá correr.), se não é uma coisa, é outra.

Exercício

1- O `Serial.Print` vai imprimir na mesma linha, já o `Serial.println()`, vai imprimir em linhas diferentes

2- Quando byte entrar na variável declarada como char, ela vai imprimir um caractere específico, seguindo o gráfico ASCII.

3-a) Porque são comandos utilizados em computadores velhos,

b) 65

c) 97

d) 48

e) 32

4- Para indicar que é uma variável permanente. Sem o static ele vai virar uma variável local e o valor não vai permanecer, vai ficar se destruindo toda as vezes que o loop reinicia.

5- Porque na tabela ASCII só vai até o 255.

6- Quando chega ao 255 ele continua em ordem crescente, como se tivesse voltado ao 0.

7-

```

1 void setup() {
2   Serial.begin(9600); // inicializa comunicação serial (9600 b
3 }
4
5 void loop() {
6   delay(100); // pausa o programa, por 100ms
7   static byte n = 0; // declara variável n, do tipo byte, e i
8   char c = n; // declara variável c, do tipo char, e a
9   Serial.print ( n, BIN ); // Converte o 'n' pra binário.
10  Serial.print ( " | " ); // separação
11  Serial.print ( n, OCT ); // Converte o 'n' pra octal.
12  Serial.print ( " | " ); // separação
13  Serial.print ( n ); // imprime o valor como um número (pq???
14  Serial.print ( " | " ); // separação
15  Serial.print ( n, HEX ); // Converte o 'n' pra hexadecimal.
16  Serial.print ( " | " ); // separação
17  Serial.println ( c ); // imprime o mesmo valor como um caracte
18  n++; // incrementa valor de n (faz n = n + 1)
19 }
20

```

Monitor serial

111111	//	03	3F	7
1000000		100		64 40 @
1000001		101		65 41 A
1000010		102		66 42 B
1000011		103		67 43 C
1000100		104		68 44 D

8- a) O código correspondente ao “{” é:

```
c = Serial.read();
```

```
n = c;
```

```
Serial.print ( n );
```

```
Serial.print ( " | " );
```

```
Serial.println ( c );
```

b) a=97; á=160; à=133.

9- Ele irá imprimir o código correspondente a cada caractere digitado, exemplo do a, que quando impresso, fica 97, seguindo a tabela ASCII.

10- Porque o valor precisará mudar conforme o usuário for digitando, ou seja, o valor não irá ser permanente.

11-

```

void setup() {
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}

```

```

void loop() {
  delay(10); // (somente para a simulação)
  byte n = 0; // declara variável n, e inicializa com valor 0
  char c; // declara variável c (não inicializa)
}

```

```

if ( Serial.available() > 0 ) { // se existem caracteres na fila de entrada
  c = Serial.read();          // leia 1 caractere da fila de entrada
  n = c;                      // atribua a n o valor de c

  Serial.print ( n );         // imprime o valor como um número
  Serial.print ( " | " );     // imprime a string " | "
  Serial.print ( c );         // imprime o valor como um caractere
  if ( n >= 47 && n <= 57 ) {
    Serial.print ( " | " );
    Serial.println ( "Categoria: numeros" );
  } else if ( n >= 97 && n <= 122 ) {
    Serial.print ( " | " );
    Serial.println ( "Categoria: letra minuscula" );
  } else if ( n >= 65 && n <= 90 ) {
    Serial.print ( " | " );
    Serial.println ( "Categoria: letra maiuscula" );
  } else {
    Serial.print ( " | " );
    Serial.println ( "Categoria: outros" );
  }
}
};

```

12-

```

void setup() {
  Serial.begin(9600); // inicializa comunicação serial (9600 bits/s)
}

void loop() {
  delay (100);        // pausa o programa, por 100ms
  static byte n = 0;  // declara variável n, do tipo byte, e inicializa com 0
  char c = n;         // declara variável c, do tipo char, e atribui valor de n
  Serial.print ( n ); // imprime o valor como um número (pq???)
  Serial.print ( " | " ); // imprime uma string ( " | " )
  Serial.println ( c ); // imprime o mesmo valor como um caractere (pq???)
  n++;                // incrementa valor de n (faz n = n + 1)
  if ( n >= 127 ) {
    n = 0;
  }
}

```