A LINGUAGEM C

Parte 1

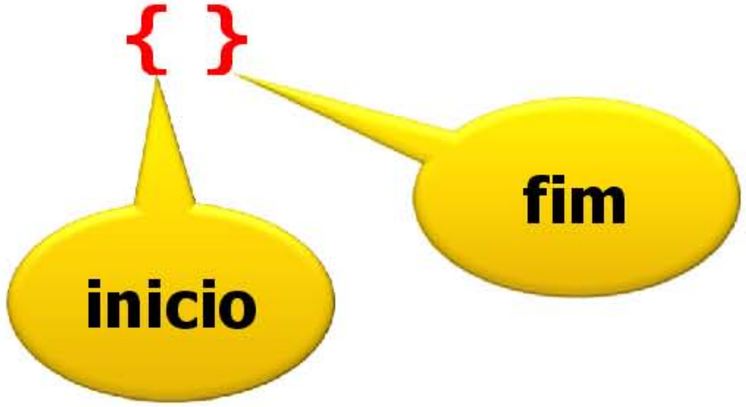
© Prof. Eng° esp Luiz Antonio Vargas Pinto
www.vargasp.com
Revisado em 02/2021

Origem

- Linguagem desenvolvida na Bell Laboratories.
- 1ª versão escrita e implementada por Denis Ritchie.
- 1ª utilização importante = reescrita do SO UNIX (original em Assembly)
- 1º Livro "The C Programming Language", por Denis Ritchie & Brian Kernighan - 1978.
- Diversas versões de C, incompatíveis, foram criadas.
  - Estas versões funcionavam somente com um determinado compilador, restritos a condições especiais.
- 1983, **ANSI** - **A**merican **N**ational **S**tandards **I**nstitute padronizou a linguagem C - ANSI C.

Características

- Case sensitive
- Separa parâmetros por vírgula
- Termina comando com ponto-e-vírgula.
- Blocos de código são delimitados por chaves



Exemplo

```
x = 0;
do
{
   PORTB = 0x23; //escrita direta no Port B
   digitalWrite (1,LOW);
     x++;
}while(x<3);
```

Fim de comando

início

fim

Separador de parâmetros

Comentários

- /* Comentário
  em mais de
  uma linha */
- // Comentário em uma linha

Bibliotecas

- Basicamente contém rotinas (funções) as quais complementam as funções internas
- Tem extensão **.h** de **header**
  - cabeçalho em inglês
  - **stdio.h** **st**and**d**art **i**nput/**o**utput
- **#include** <nome_da_biblioteca>
- Exemplo: **#include** <stdio.h>

Nome

Função loop()

- Todos os programas em C têm que ter uma função **main()**, a qual está embutida na IDE Arduino.
- A linha **loop()** indica que estamos definindo uma função de nome **loop**.
- O **loop()** em particular serve para manter a execução do programa restrito aos códigos dentro deste evitando acesso na memória além deste laço.

Tipos de Dados

Usuais

Data Type	Size (Bytes)	Range of Values
void	0	null
bool/boolean	1	True/False
char	1	-128 to +127
unsigned char	1	0 to 255
byte	1	0 to 255
int	2	-32,768 to 32,767
unsigned int	2	0 to 65,535
word	2	0 to 65,535
long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
float	4	-3.4028235E+38 to 3.4028235E+38
double	8	-1.7976932E+308 to 1.7976932E+308
string	-	character array

Declaração de Variáveis

- As variáveis devem ser declaradas antes de serem usadas.
- A forma geral da declaração de variáveis é:

  tipo nome1;

  Ou

  tipo nome1,nome2; //nome1 e nome2 são do mesmo tipo
- Exemplo:
  - int Dias;
  - float x,y;
- Podemos inicializar variáveis no momento de sua declaração:

  char numero = 12;
- Quando C cria uma variável, ele **não** atribui valor inicial ficando este *indefinido*.
  - ***Nunca* presuma que uma variável declarada vale zero ou qualquer outro valor**.

Tipos char e void

- São caracteres/símbolos como a, b, etc..
- Devem vir entre apóstrofes: 'a', 'b', etc..
- **char** também armazena valores numéricos inteiros de um byte (8 bits, isto é, até 255).
- **void** é o tipo vazio, ou um "tipo sem tipo".
  - Aplicação deste "tipo" será vista oportunamente.

Escopo de Variáveis

- Há três lugares nos quais podemos declarar variáveis:

1. Fora de todas as funções do programa → variável **global**
   - Todos podem utilizar – é o mesmo endereço de memória
2. No início de um bloco de código → variável **local**
   - só vale no bloco no qual são declaradas
3. Na lista de parâmetros de uma função
   - **Apesar de receberem valores externos, estas variáveis são conhecidas apenas pela função onde são declaradas.**

Exemplo:

```
float numero;
setup()
{
}
loop()
{
  int Dias;
  float Anos;
}
```

Variável Global

Foi declarada antes de tudo

Variável local

Foi declarada dentro de { }

Global

Protótipo da função

```
int contador;
int calculo(int j);
setup()
{
}
```

Local

```
loop()
{
  char condicao;
  int i;
  for (i=0; i<100; i=i+1)
  {
    float f2; // pertence ao for... é local
    calculo(i);
  }
  /* etc  ...  */

}
```

função

```
int calculo(int j)
{
   contador = j + 2;
}
```

j é local em calculo

Atribuição

- Ch='D';
- x=23.5;
- Anos=Dias/365.25;
- str[1] = 'a';

Ponto decimal

A função if

Forma:

```
if (condição)
{
   comandos;
}
```

ou:

```
if (condição)
{
   comandos_1;
}
else
{
   comandos_2;
}
```

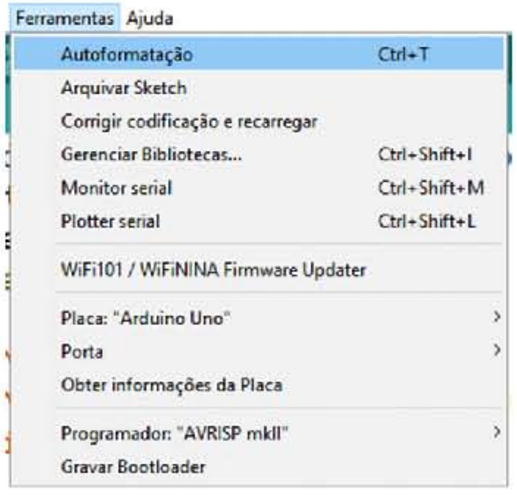
Opção do if

Usando a função if

```
loop()
{
    int num;  // ou use int num = 23;
    num = 23;
    if (num>10) digitalWrite(1,LOW);
    if (num==10)
    {
        digitalWrite(1,HIGH);
    }else
    {
        digitalWrite(2,HIGH);
    }
}
```

Espaço de identação

Na IDE Arduino



Condicionais

- **==** igual
- **!=** diferente de
- **>** maior que
- **<** menor que
- **>=** maior ou igual
- **<=** menor ou igual.

Operadores binários e aritméticos

Operador	Ação
&	And
|	Or
!	Not
^	Xor
~	Inverte
>>	Bits to right
<<	Bits to left

Operador	Ação
+	Soma inteira e fracionada
-	Subtração e troca de sinais
*	Multiplicação inteira ou fracionada
/	Divisão inteira e fracionada
%	Resto da divisão
++	Incremento
--	decremento

Obrigado !
A gente se vê na próxima..