

Provinha**1. O que é um microcontrolador?**

Circuito integrado que funciona como um mini computador, porém integrado em um único chip. Possui um núcleo que processa seus dados, memórias voláteis e não voláteis; e periféricos de entrada e saída (PIO's), como I/O's, geradores de sinais PWM, Timers, Conversores A/D, etc.

2. Qual a família de microcontroladores utilizada no curso?

A família de microcontroladores utilizada no curso é a família ARM Cortex M-4.

3. Das propriedades básicas do microcontrolador, explique duas de sua escolha:

(a) **Harvard / Von Neumann:** dois tipos de arquiteturas de computadores, com a diferença de que a arquitetura Harvard baseia-se na utilização de memórias diferentes para controle e dados, enquanto a arquitetura Von Neumann utilizada a mesma memória para armazenar ambas.

(b) **RISC/ CISC:** dois tipos de arquitetura de linguagem de processadores, que se diferenciam por o RISC (Reduced Instruction Set Computer) possuir uma linguagem de instruções mais reduzidas, porém mais rápidas de serem executadas, enquanto o CISC (Complex Instruction Set Computer) possui uma linguagem de instruções mais variadas e complexas, mas que demoram mais para serem executadas.

4. Explique as seguintes definições do C: volatile/const/static.

- Volatile: uma variável declarada como *volatile* pode ter seus dados alterados durante a execução do programa, porém sem o reconhecimento do compilador, impedindo-o de fazer otimizações no programa onde a variável encontra-se presente.
- Const: uma variável declarada como *const* pode apenas ter seus dados lidos, não podendo ser alterada. A variável *const* serve de proteção para que o valor armazenado na mesma não seja perdido no executar do programa.
- Static: uma variável declarada como *static* pode ser global ou local. Variáveis globais *static* funcionam como variáveis globais comuns, com a diferença de que só funcionam no programa em que foram declaradas, já variáveis locais *static* mantém o seu valor em duas chamadas distintas da mesma função.

5. Explique o uso do Watchdog Timer.

O Watchdog Timer é um dispositivo de clock independente que, com a ausência de reset até um tempo programado, entende que houve uma condição de erro no programa principal, reiniciando o sistema.

6. Dado um microcontrolador hipotético de 8bits, preencha o valor das variáveis conforme a evolução do código (sem considerar nenhuma otimização por nível do compilador). O subscrito no nome da variável indica a linha no código correspondente. Se o seu valor for indefinido, deixar o espaço em branco.

```

1 void main(){
2   int var;
3
4   var = 0xFA;
5
6   var |= 0x01;
7
8   var &= ~(0b10001000);
9
10  if( ((var >> 3) & 0x01) )
11      var = 0x02;
12  else
13      var = 0x07;
14
15  while(1){
16
17  }
18
19  var = var & 0x02;
20
21 }

```

Bit	7	6	5	4	3	2	1	0
Var ₄ =	1	1	1	1	1	0	1	0
Var ₆ =	1	1	1	1	1	0	1	1
Var ₈ =	0	1	1	1	0	0	1	1
Var ₁₁ =	-	-	-	-	-	-	-	-
Var ₁₃ =	0	0	0	0	0	1	1	1
Var ₁₉ =	-	-	-	-	-	-	-	-