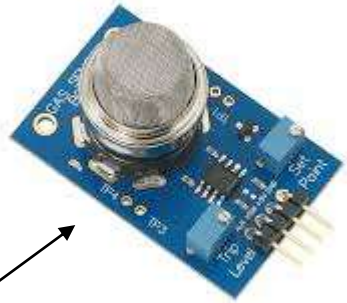
PROJETO com MCU

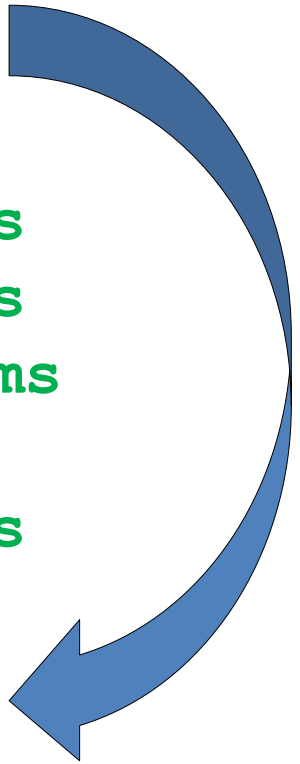


PROGRAMA em C

```
void main()
{
        InitSys();
        while(true)  {
          SensorCorrente();
          Termopar()
          Umidade();
          SensorPresenca();
          SendToUart();
          UpdateLcd();
    }
}
```

PROGRAMA em C

```
void main()
{
        InitSys();
        while(true) {
           SensorCorrente();      //2ms
           Termopar()             //10ms
           Umidade();             //15ms
           SensorPresenca();      //100ms
           SendToUart();          //1ms
           UpdateLcd();           //10ms
    }
}     //Tempo Gasto por loop: 138ms
```



SENAI

Como melhorar o desempenho do programa?

Prof° Fernando Simplicio

SENAI

Melhorias no Hardware | Software

Prof° Fernando Simplicio

Hardware (exemplos)

- **Escolher microcontroladores que atendam as necessidades do projeto.**
- **Conversores AD mais rápidos.**
- **Memórias Flash/EEPROM mais rápidas.**
- **Interfaces de comunicação mais rápidas. Ex: SPI ao invés de I2C e USB ao invés de RS232.**
- **Melhorar o layout PCB afim de diminuir interferências, *bounces*, etc.**

Software

- **Procurar deixar o código o mais objetivo e de fácil entendimento quanto possível.**
- **Usar com cautela macros e funções *inline*.**
- **Algoritmos recursivos podem chegar rapidamente ao resultado (poucas iterações), porém exigem o uso de máquina e recursos computacional melhores.**

SENAI

1° Alternativa:
Utilizar as Interrupções do MCU para atender e priorizar tarefas/processos.

Prof° Fernando Simplicio

SENAI

2° Alternativa:
Utilizar máquinas de estado para melhorar o tempo entre processos.

Prof° Fernando Simplicio

SENAI

3° Alternativa:
Utilizar máquinas de estado nos processos em conjunto com as interrupções de periféricos/CPU do MCU.

Prof° Fernando Simplicio

SENAI

4° Alternativa:
Usar um sistema operacional de Tempo Real RTO’s para individualizar, priorizar, sincronizar processos.

Prof° Fernando Simplicio

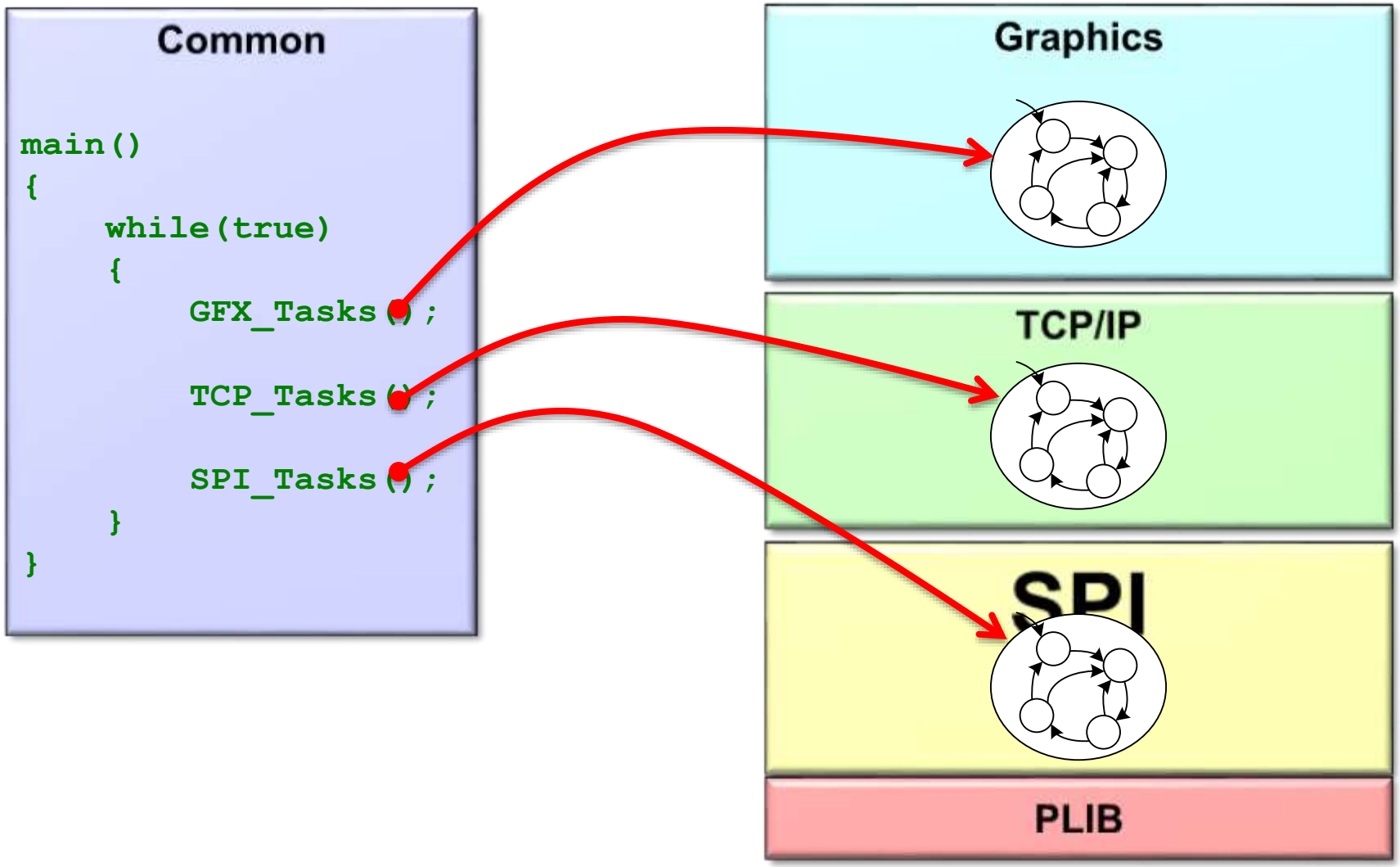
Máquina de Estado

- **Break long tasks into a state machine**

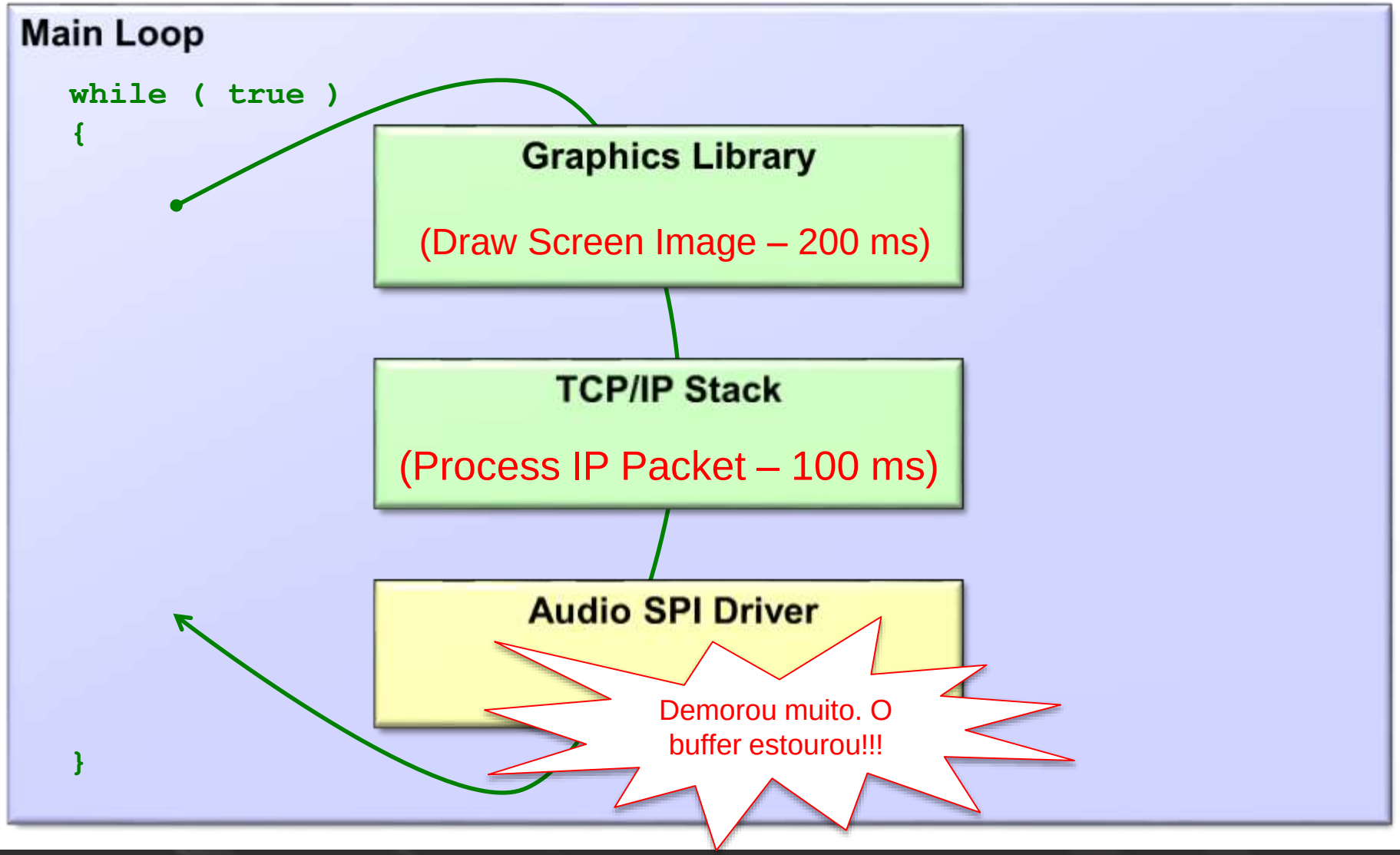
```
while (1)
{
  Task_1(); //60 us
  Task_2(); //2 us
}
//max loop time = 62 us
```

```
while (1)
{
  switch(Task_1_state)
  {
   case a:
     Task_1_state_a(); //20 us
     break;
   case b:
     Task_1_state_b(); //20 us
     break;
   case c:
     Task_1_state_c(); //20 us
     break;
  }
  Task_2();  //2 us
} //max loop time = 22 us
```

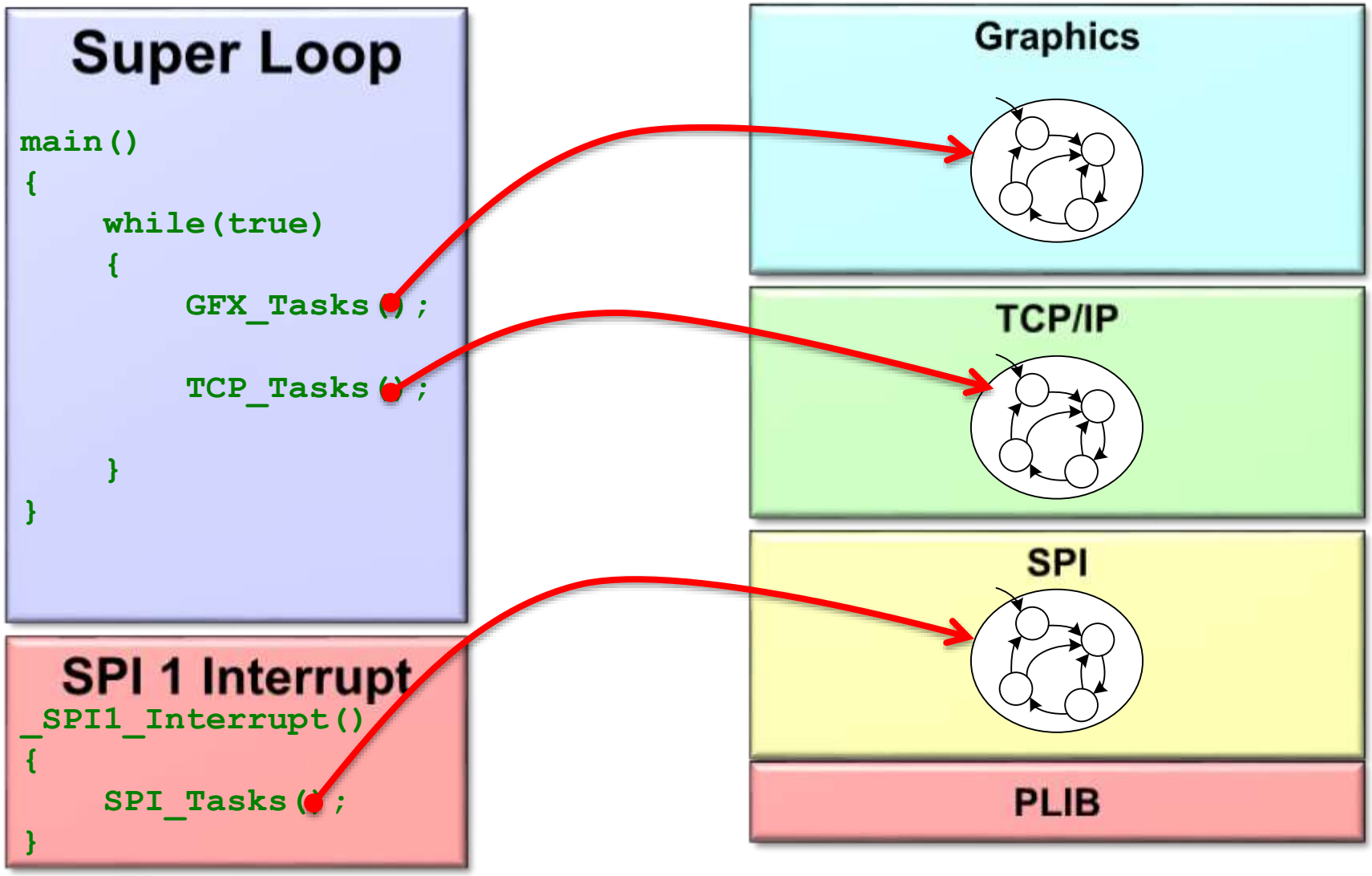
Polled Configurations



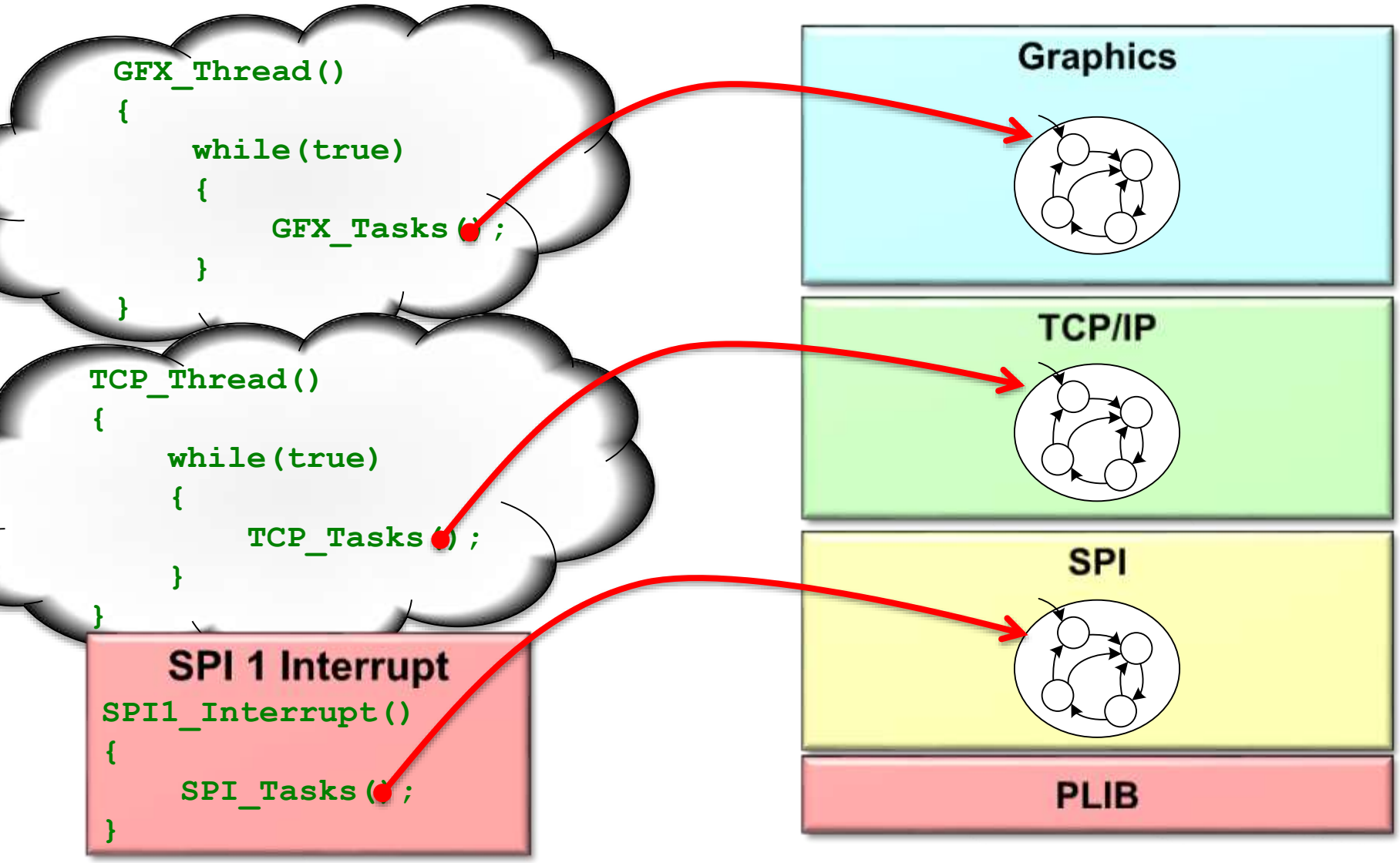
máquina de Estado



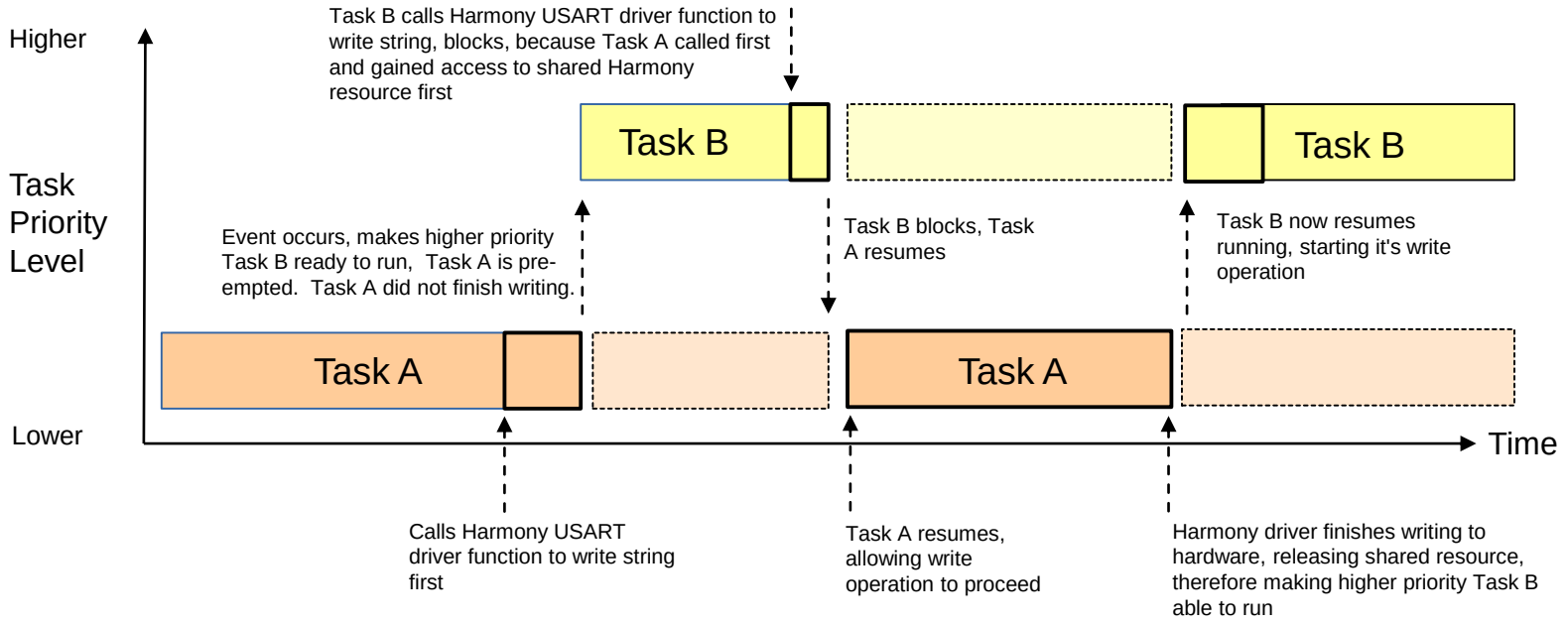
Interrupção dos Periféricos/CPU



Aplicando um RTO'S



Benefícios de um RTO's



Task A

```
signed char* pTaskA_str = "Hello World";

void Task_A_Func(void* pvParameter)
{
   while(true)
   {
      /* Perform a specific task job */
      DRV_USART_Write(UART_1_Handle,
         (void*)pTaskA_str,
      (size_t)(strlen(pTaskA_str)) );
   }
}
```

Task B

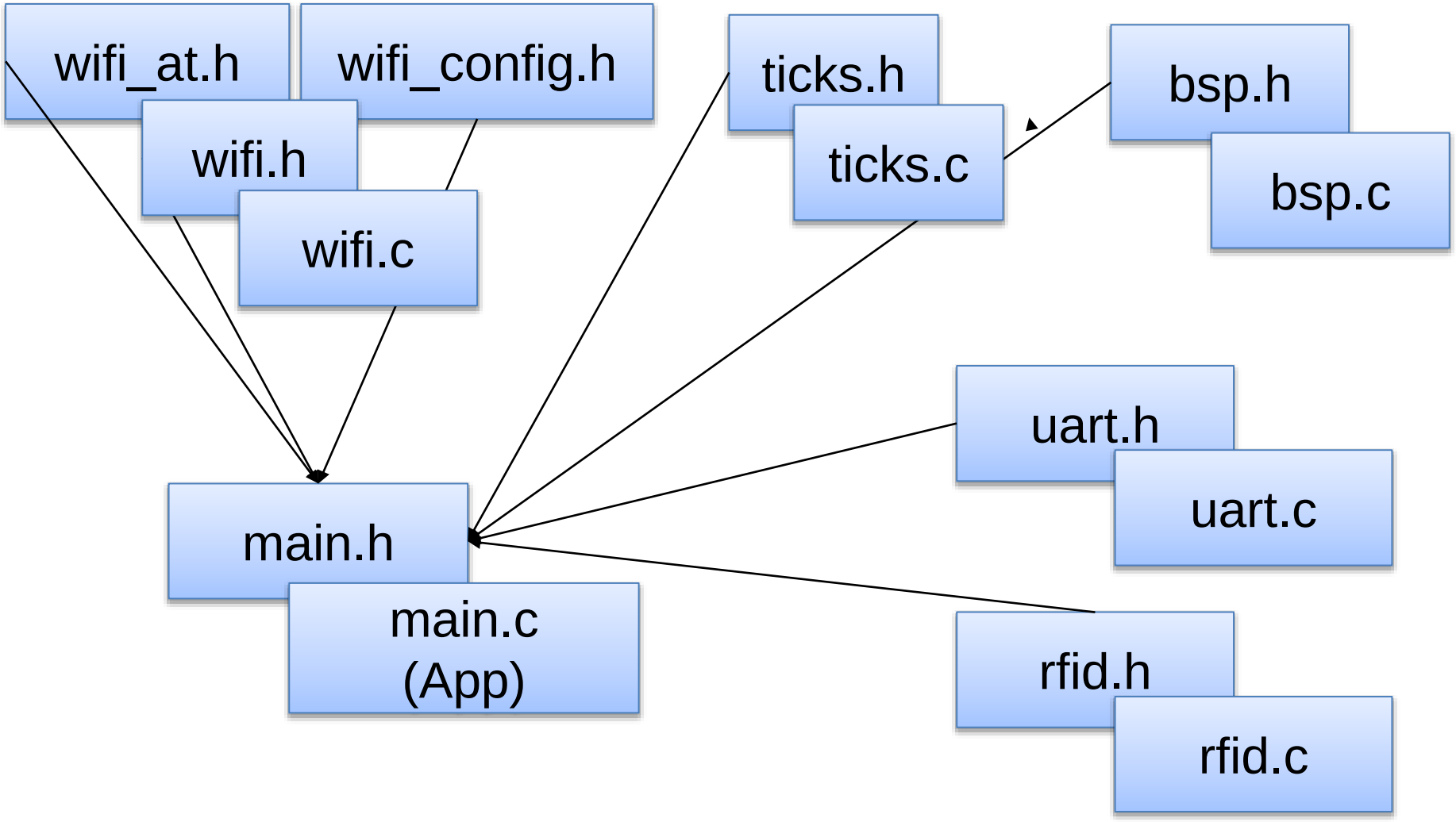
```
signed char* pTaskB_str = "Go Rattlers!";

void Task_B_Func(void* pvParameter)
{
   while(true)
   {
      /*perform Harmony or application specific
      task job*/
      DRV_USART_Write(UART_1_Handle,
         (void*)pTaskB_str,
      (size_t)(strlen(pTaskB_str)));
   }
}
```

SENAI

Análise de um projeto funcional com Microcontroladores

Prof° Fernando Simplicio

Exemplo de um Projeto para MCU



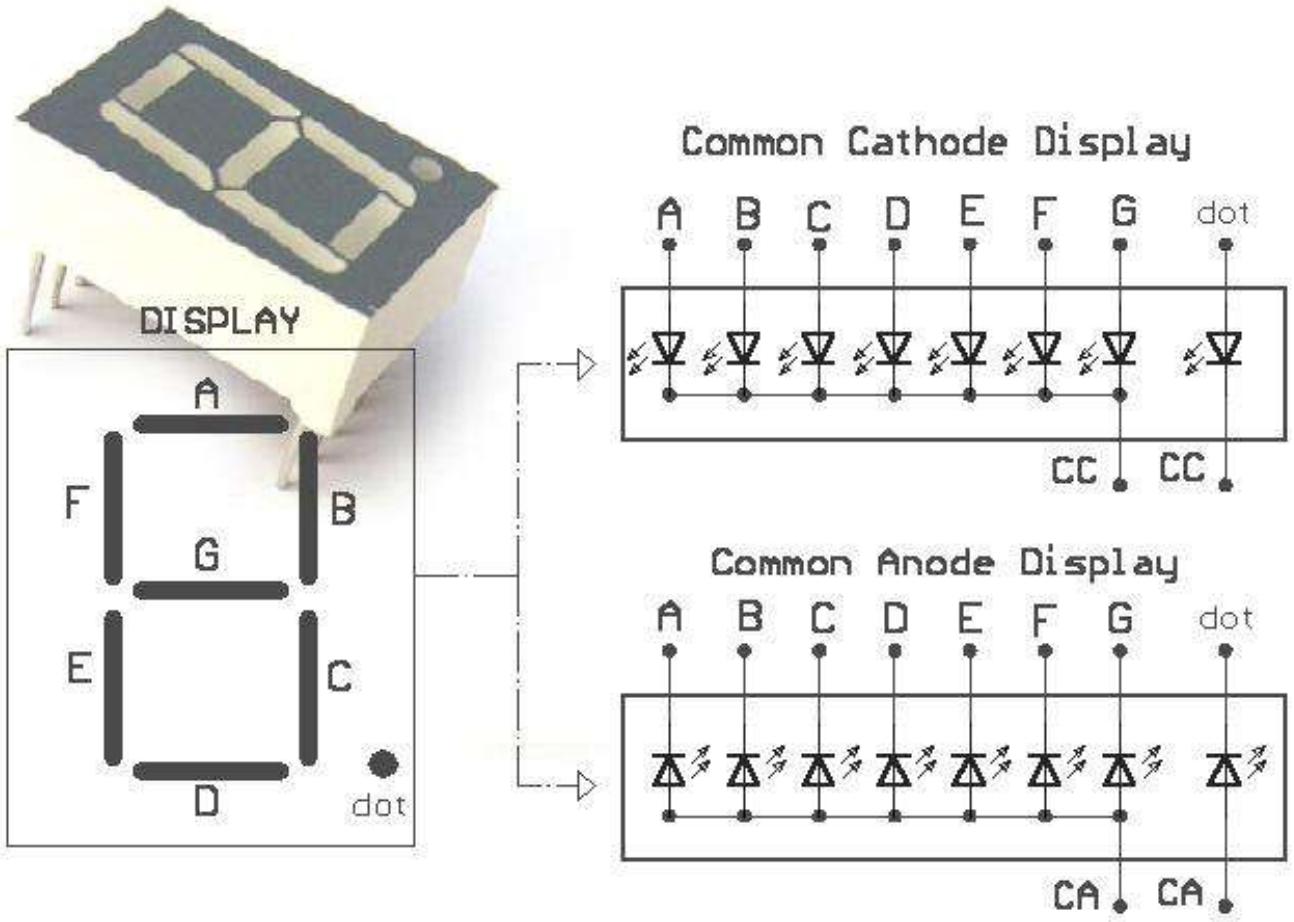
SENAI

Projeto: Voltimetro com Display de 7 Segmentos

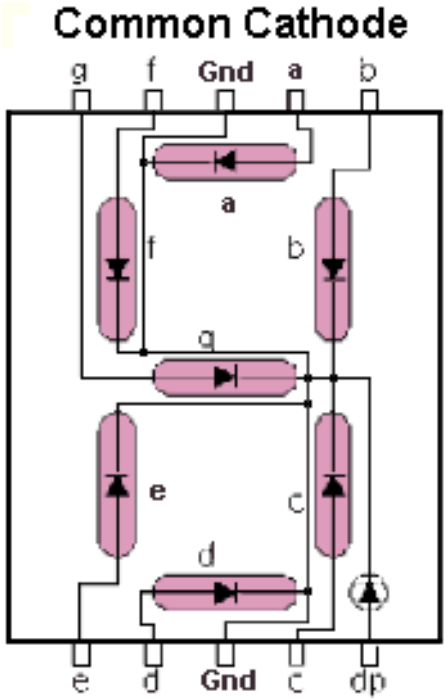
www.microgenios.com.br

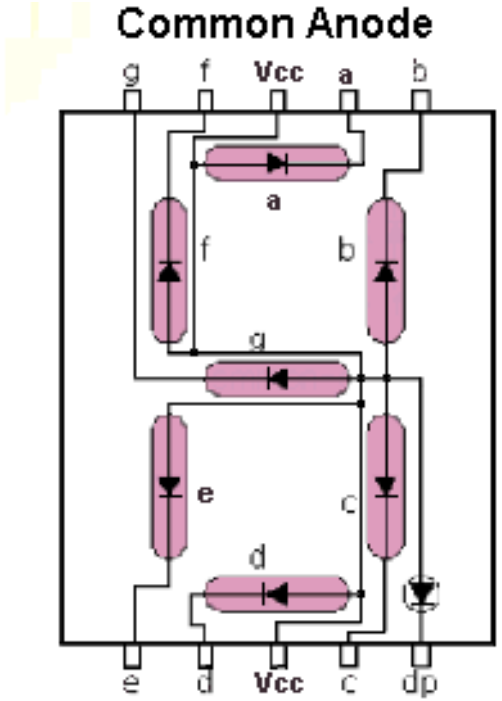
Teoria sobre
Display 7 Segmentos

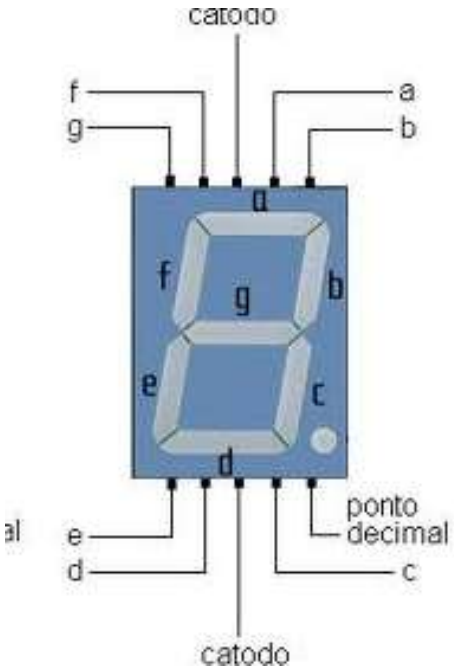
Teoria sobre
Display de 7 Segmentos



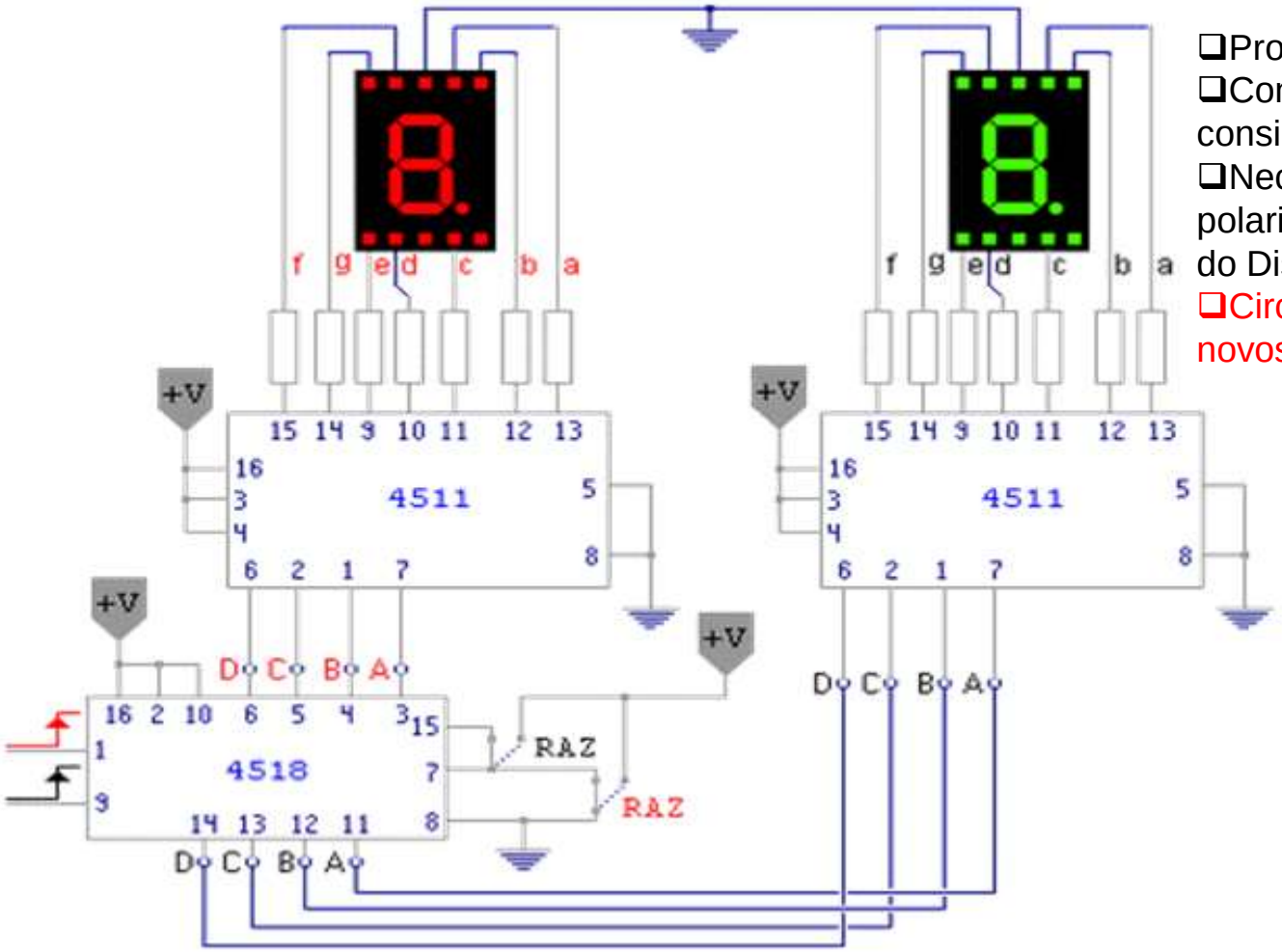
Teoria sobre
Display de 7 Segmentos





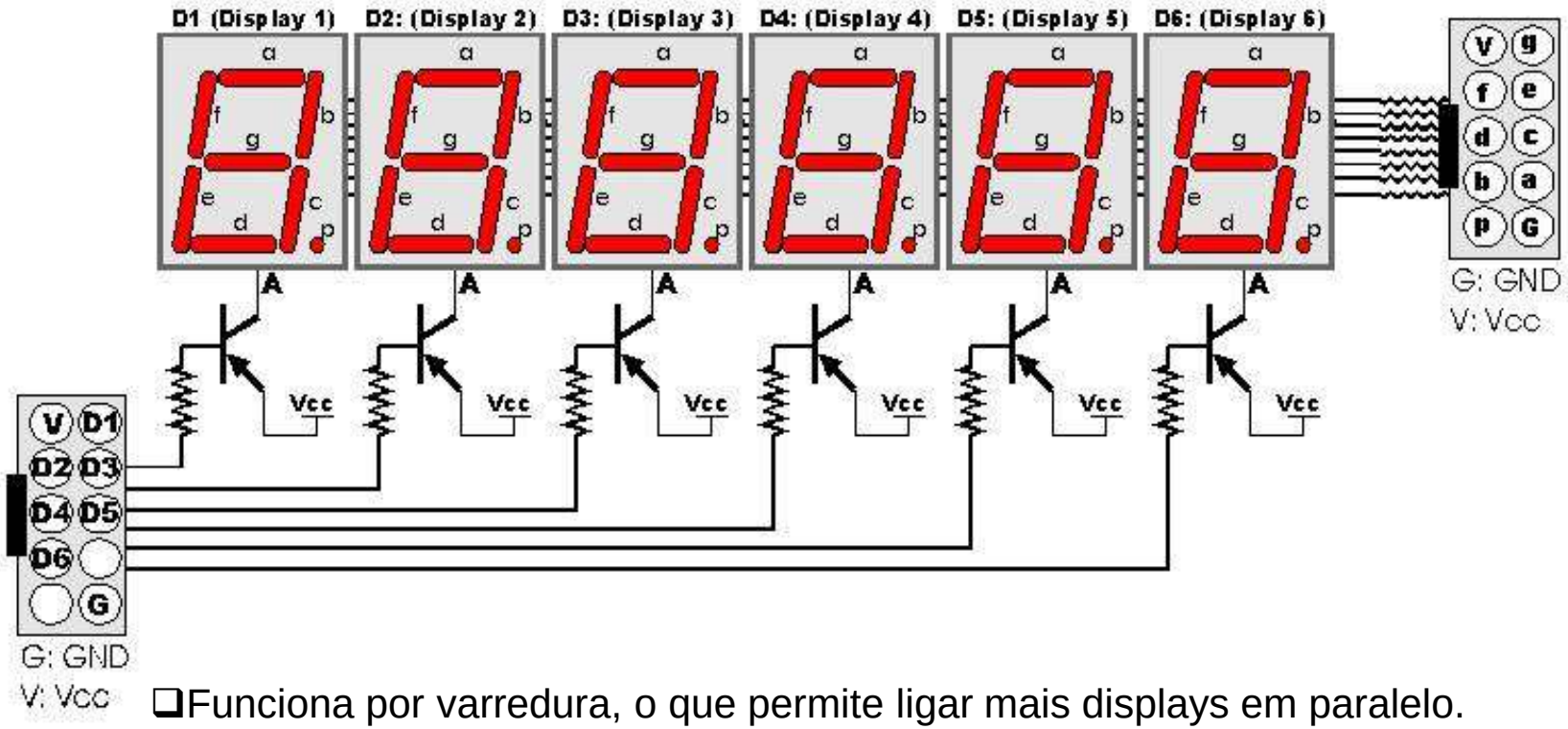


Projetos Antigos



- Projeto fixo e sem flexibilidade
- Consumo de corrente considerável.
- Necessário resistores de polarização dos leds para cada LED do Display.
- Circuito NÃO recomendável para novos projetos.

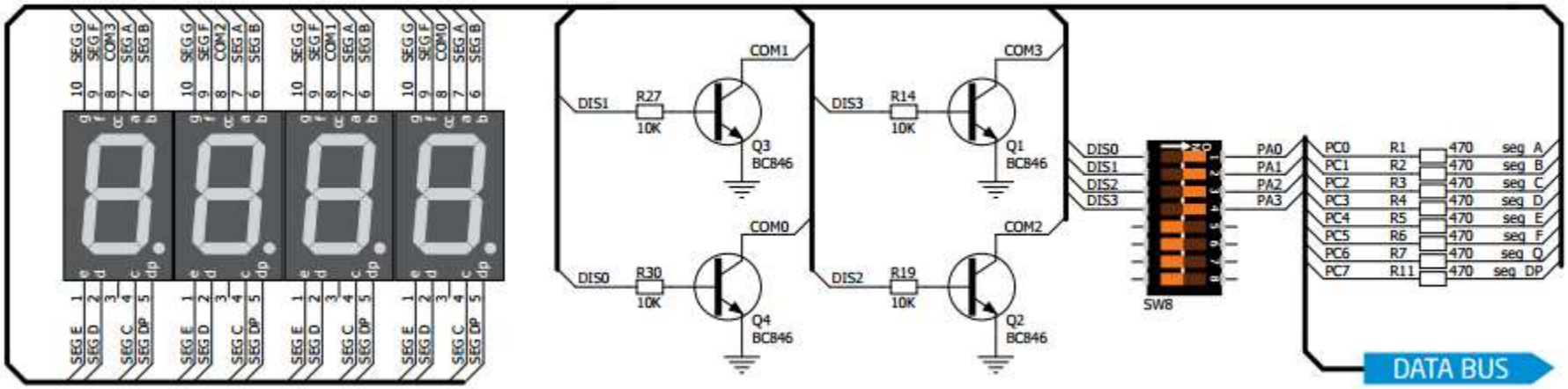
Novos Projetos
Sistema de Varredura



- Funciona por varredura, o que permite ligar mais displays em paralelo.
- Garante maior durabilidade dos LEDs, pois os NÃO ficam todo o tempo ligados.
- Economiza Pinos do Microcontrolador
- Permite maior flexibilidade na mudança do projeto
- Trabalha com picos de correntes, o que acarreta menor consumo de corrente.
- Menor número de componentes discretos para seu acionamento.
- Circuito recomendado para novos projetos.

Display 7 Segmentos no kit EasyAVR v7

4 digit
7-seg display



O que você acha do programa?

```
void main()
{
        InitMcu();
        InitADC();
        while(true) {
            LeituraADC();
            ConversaoADC();
            Display7();
            Button();
    }
}
```

SENAI

Como melhorar o programa?

Projeto: Voltimetro com Display de 7 Segmentos

Prof° Fernando Simplicio

O que você acha do programa?

```
void timer() interrupt_isr()
{
    Display7();
}

void main()
{
        InitMcu();
        InitADC();
        InitTimer();
        InitInterrupt();
        while(true) {
                LeituraADC();
                ConversaoADC();
                Button();
        }
}
```

Exemplo de varredura p/ display 7 seg

```
void Display7(){

        PORTC = seg_Unidade;            //Escreve no Barramento
        PORTA.B0 = 1;                   //Liga display7
        Delay_ms(2);                    //Espera o led acender por 2ms.
        PORTA.B0 = 0;                   //Desliga display7

        PORTC = seg_Dezena;             //Escreve no Barramento
        PORTA.B1 = 1;                   //Liga display7
        Delay_ms(2);                    //Espera o led acender por 2ms.
        PORTA.B1 = 0;                   //Desliga display7

        PORTC = seg_Centena;            //Escreve no Barramento
        PORTA.B2 = 1;                   //Liga display7
        Delay_ms(2);                    //Espera o led acender por 2ms.
        PORTA.B2 = 0;                   //Desliga display7

        PORTC = seg_Milhar;             //Escreve no Barramento
        PORTA.B3 = 1;                   //Liga display7
        Delay_ms(2);                    //Espera o led acender por 2ms.
        PORTA.B3 = 0;                   //Desliga display7

}
```

Exemplo Máquina de Estado

```
void Display7() {
      static enum { _disp1 = 1, _disp2, _disp3, _disp4} _maq_estado = _disp1;

      PORTA &= 0XF0;
       switch (_maq_estado)
       {
          case _disp1: {    //Unidade
                        PORTC = Dta[0];
                        PORTA.B0 = 1;
                        _maq_estado = _disp2;
                        break;
                       }
          case _disp2: {    //Dezena
                        PORTC = Dta[1];
                        PORTA.B1 = 1;
                        _maq_estado = _disp3;
                        break;
                       }
          (...)
       }
}
```

SENAI

Exemplo Prático: ATMEGA32 WinAVR no Atmel Studio 7.

Projeto: Voltimetro com Display de 7 Segmentos

Prof° Fernando Simplicio

SENAI

Exemplo Prático:
ATMEGA32 e mikroC PRO for AVR

Projeto: Voltimetro com Display de 7 Segmentos

Prof° Fernando Simplicio

SENAI

Exemplo Prático:
ATMEGA32 e WinAVR com Sistema Operacional de Tempo Real (RTO's).

Projeto: Voltimetro com Display de 7 Segmentos

Prof° Fernando Simplicio

Links

- RTO’s OSA: http://www.pic24.ru/
- WinAVR: https://sourceforge.net/projects/winavr/
- Atmel Studio 7: http://www.atmel.com/microsite/atmel-studio/
- mikroC PRO for AVR: https://www.mikroe.com/avr/
- ATMEGA32: http://www.microchip.com/wwwproducts/en/ATmega32

SENAI

Obrigado!

Prof° Fernando Simplicio