

Lab 4 – Bruno Guillen

1. Planejamento das fases do processo de desenvolvimento.

- Entendimento do 8N1
- Estudo do funcionamento do Tera Term
- Entendimento do Problema a ser resolvido
- Elaborar estratégia de desenvolvimento

2. Definição do problema a ser resolvido.

Elaborar um projeto no IAR cuja funcionalidade é receber dados pela porta serial (UART) do processador da Tiva, processar estes dados e enviá-los pela mesma porta serial.

A comunicação será feito com um emulador de terminal (TeraTerm ou similar) executando no PC.

O processamento de dados consiste em converter os caracteres maiúsculos (A-Z) nos correspondentes minúsculos (a-z) sem alterar os demais bytes recebidos. A taxa de comunicação deve ser de 115200 bps no formato 8N1.

3. Especificação da solução.

Requisitos funcionais:

RF1 - O programa deve receber mensagens.

RF2 - O programa deve converter letras maiúsculas para minusculas

RNF1- As mensagens serão recebidas por um simulador serial

Restrições:

- deve ser utilizada comunicação Serial
- a solução deve fazer uso de assembly.

4. Estudo da plataforma de HW (TIVA e seu processador).

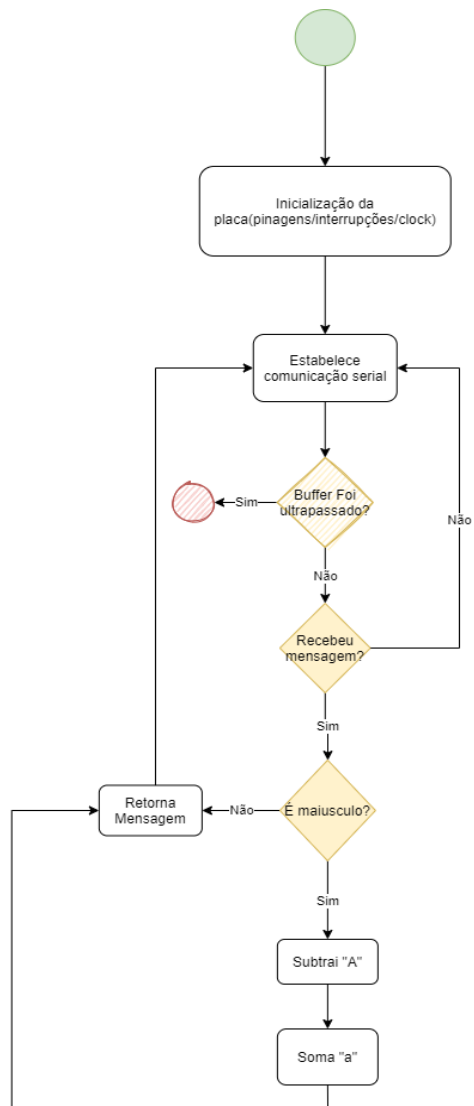


TivaWare™ Peripheral Driver Library

USER'S GUIDE

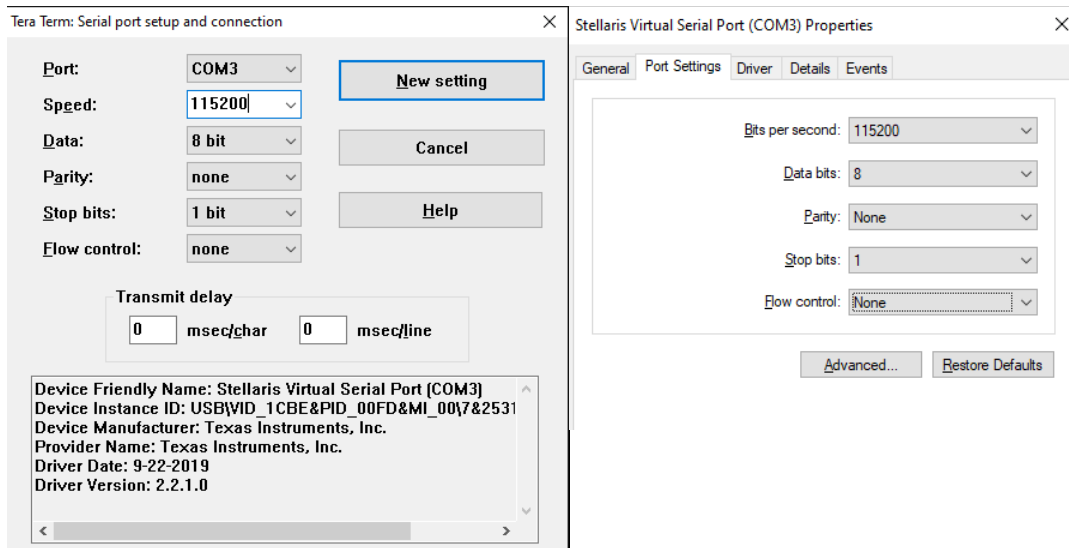
Além do conhecimento proveniente dos laboratorios anteriores foram consultados no material da TivaWare o Comportamento e funções presentes da UART da placa

6.Projeto (design) da solução:



7. Configuração do projeto na IDE(IAR) e da Stellaris.

Tera Term and Stellaris configuration:



A configuração da IDE seguiu os mesmos modelos do Laboratório 2

8. Edição do código da solução.

Além das próprias inicializações para UART/GPIO e importações de bibliotecas/tivaware.

Interrupção para se ler o valor do caractere digitado na plataforma Tera Term

```
void UART_Interruption_Handler(void){
    buffer[buffer_position%BUFFER_SIZE] = (uint8_t)UARTCharGetNonBlocking(UART0_BASE); //get the character from UART casted to uint to buff;
    UARTIntClear(UART0_BASE,UARTIntStatus(UART0_BASE,true)); //clear the interruptions on uart
    buffer_position++; //increment the position
}
```

Configuração para 8N1 (8bits 1 bit de parada sem paridade) a 120mhz(da placa e) Transferência pedida no exercício (115200):

```
// 8N1 120mhz 115200 bps
UARTConfigSetExpClk(UART0_BASE, (uint32_t)120000000, (uint32_t)115200, (UART_CONFIG_WLEN_8|UART_CONFIG_STOP_ONE|UART_CONFIG_PAR_NONE));
```

Função para conversão de Caracteres:

```
static uint8_t lowercase (uint8_t ch)
{
    if(isupper(ch)) //check if the values is upper
    {
        ch = (uint8_t)(ch - 'A'); // turn the value into making 0-26
        ch = (uint8_t)(ch + 'a'); // the sum make it corrected to a lower case
    }
    return ch;
}
```

Checando se valor é maiúsculo e então corrigindo o seu valor para o valor minúsculo utilizando o valor da primeira letra.

Para isso foi necessária a biblioteca **cctype.h**

Rotina para Conversão de caracteres

```
while(1){  
  
    if(buffercheck!=buffer_position){  
        charIO = &buffer[buffercheck%BUFFER_SIZE]; //get the message address in the buffer  
  
        *charIO = lowercase(*charIO); //function to turn into lower case  
  
        if(UARTCharPutNonBlocking(UART0_BASE,buffer[buffercheck%BUFFER_SIZE])){  
            buffercheck++; //update the buffer value  
        }  
    }  
}
```

9. Teste e depuração.

Digitando a Frase: Sistemas Embarcados ELF74 a saída do terminal foi:

