

EEL7030 - Microprocessadores



LCS

Laboratório de
Comunicações
e Sistemas
Embarcados

Prof. Raimes Moraes

EEL - UFSC

Exemplo 11 - Configurar ADC, DMA e USB

Exemplo que lê do conversor ADC e carrega em buffer da memória

DMA gera interrupção na metade do preenchimento do buffer e no final. Seta flags que são usados no while do programa principal para transmitir, pela USB, a primeira metade do buffer e a segunda metade do buffer, respectivamente

A execução deste exemplo requer a instalação de driver para o Windows. Este pode ser obtido em:

<http://www.st.com/en/development-tools/stsw-stm32102.html>

Para a visualização dos dados recebidos, utilizar programa para recepção serial:

Tera Term ou Bray Terminal (<https://sites.google.com/site/terminalbpp/>)

Exemplo 11 - Configurar ADC, DMA e USB

OBS: Para o caso de não haver sinal analógico na entrada do ADC, criou-se um vetor com dados (DMA_USB_teste) a ser transmitido para o PC pela USB.

Caso haja sinal analógico na entrada do ADC, modifique o endereço apontado pelos ponteiro entre os rótulos */* USER CODE BEGIN 2 */* e */* USER CODE END 2 */* para:

```
USBptr = (uint8_t*) DMA_USB_Buffer;  
USBptr2 = (uint8_t*)&DMA_USB_Buffer[8];
```

Exemplo 11 - Configurar ADC, DMA e USB

❑ Objetivo:

- Configurar ADC, DMA e USB no STM32CubeMX
- Gerar código e utilizar funções HAL
- Transmitir dados pela USB empregando DMA

❑ Crie projeto no STMCubeMX:

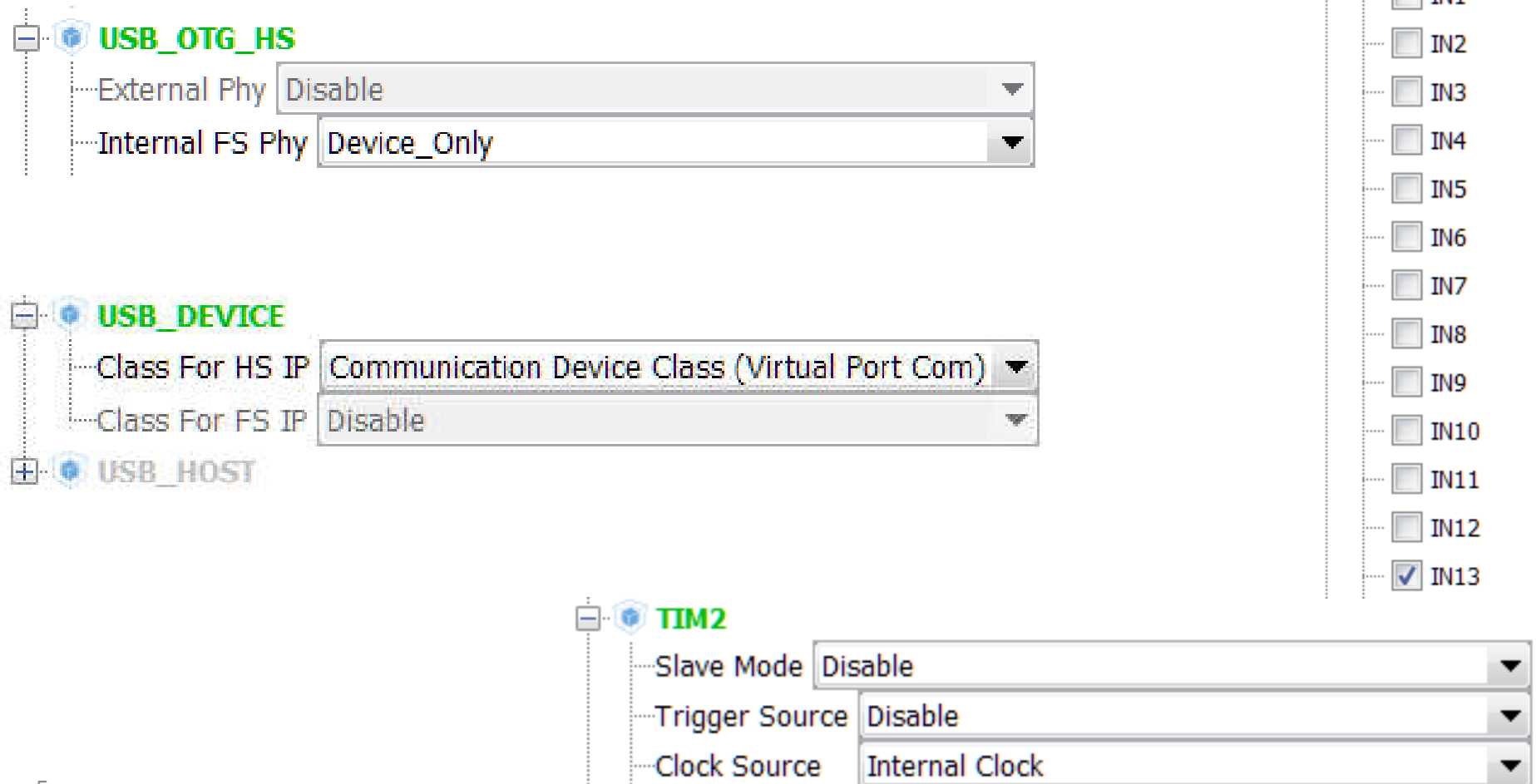
1. *New Project*

2. No MCU Selector: STM32F4 (series); STM32F429/439 (lines); LQFP144 (package); selecione: STM32F429ZITx - **OBS:** Selecionar *crystal* em RCC e configurar clocks; Selecionar SWD para Debug em SYS

OBS: o clock da interface USB deve ser de 48MHz

Exemplo 11 - Configurar ADC, DMA e USB

3. Configurar os modos de funcionamento da USB, ADC1 e TIM2



Exemplo 11 - Configurar ADC, DMA e USB

❑ Configure o TIM2 => Tab>Configuration>Control>TIM2

4. Especificar a taxa de amostragem e a fonte de trigger do ADC

TIM2 Configuration

✓ Parameter Settings ✓ User Constants ✓ NVIC Settings ✓ DMA Settings

Configure the below parameters :

Search :

☒ Counter Settings

Prescaler (PSC - 16 bits value)	2100
Counter Mode	Up
Counter Period (AutoReload Register - 32 ... 10000)	

Internal Clock Division (CKD) No Division

☒ Trigger Output (TRGO) Parameters

Master/Slave Mode	Disable (no sync between this TIM (Master) and its S...
Trigger Event Selection	Update Event

Restore Default Apply Ok Cancel

Exemplo 11 - Configurar ADC, DMA e USB

❑ Configure o ADC1 => Tab>Configuration>Analog>ADC1

5. Habilite solicitação contínua do DMA e especifique a fonte de trigger do ADC

ADC1 Configuration

✓ Parameter Settings ✓ User Constants ✓ NVIC Settings ✓ DMA Settings ✓ GPIO Settings

Configure the below parameters :

Search :

ADCs_Common_Settings	
Mode	Independent mode
ADC_Settings	
Clock Prescaler	PCLK2 divided by 4
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Enabled
End Of Conversion Selection	EOC flag at the end of single channel conversion
ADC_Regular_ConversionMode	
Number Of Conversion	1
External Trigger Conversion Source	Timer 2 Trigger Out event
External Trigger Conversion Edge	Trigger detection on the rising edge
Rank	1
ADC_Injected_ConversionMode	
Number Of Conversions	0
WatchDog	

Restore Default Apply Ok Cancel

Exemplo 11 - Configurar ADC, DMA e USB

❑ Configure o ADC1 => Tab>Configuration>Analog>ADC1

6. Configurar uso do DMA pelo ADC:

- Tab-> DMA Settings
- Clique em **Add**
- Em **Select**, selecione ADC1
- Configurar campos **Mode** e **Data Width** como ao lado

The screenshot shows the 'ADC1 Configuration' dialog box with the 'DMA Settings' tab selected. The 'DMA Request' table has one entry: 'ADC1' with 'DMA2 Stream 0', 'Peripheral To Memory' direction, and 'Low' priority. Below the table, the 'DMA Request Settings' section shows 'Mode' set to 'Circular' and 'Increment Address' checked. The 'Data Width' is set to 'Half Word'.

DMA Request	Stream	Direction	Priority
ADC1	DMA2 Stream 0	Peripheral To Memory	Low

Buttons: Add, Delete

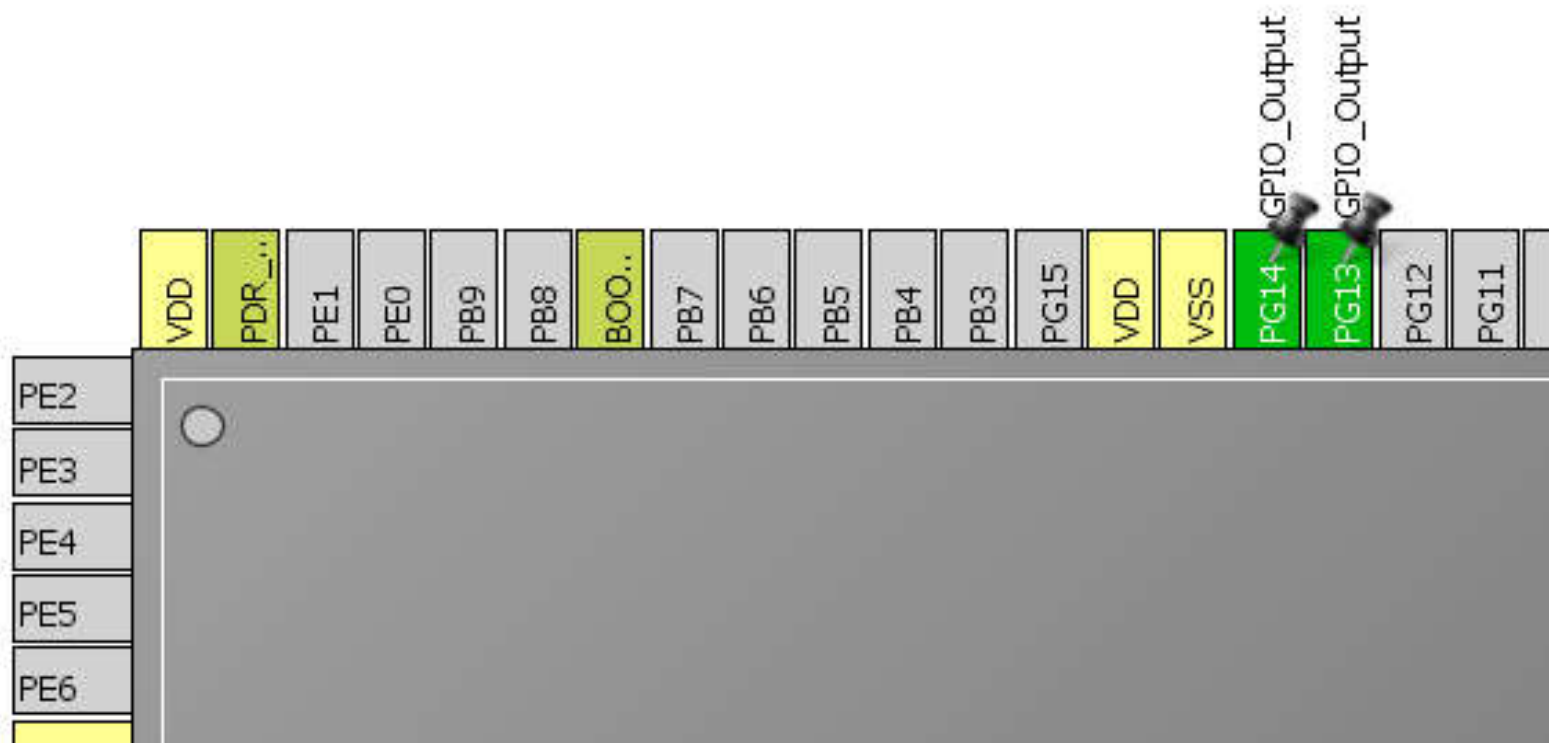
DMA Request Settings

Peripheral		Memory
Mode	Circular	Increment Address <input checked="" type="checkbox"/>
Use Fifo <input type="checkbox"/>	Threshold	Data Width: Half Word
Burst Size		Half Word

Buttons: Restore Default, Apply, Ok, Cancel

Exemplo 11 - Configurar ADC, DMA e USB

7. Configure os pinos PG13 e PG14 como pinos de saída



Exemplo 11 - Configurar ADC, DMA e USB

8. Forneça dados para geração do código:

- Menu -> Project -> Project Settings
- Digite *Project name*
- Digite *Project location*
- Escolha *toolchain* (MDK-ARM V5)
- Clique em Ok

9. Gere template para código:

- Menu -> Project -> Generate Code
- *Open Project*

The screenshot shows the 'Project Settings' dialog box with the 'Project' tab active. The 'Code Generator' tab is also visible. The 'Project Settings' section contains the following fields:

- Project Name:** First
- Project Location:** C:\Users\Raimes\STM\Exemplos\ (with a 'Browse' button)
- Toolchain Folder Location:** C:\Users\Raimes\STM\Exemplos\First\
- Toolchain / IDE:** MDK-ARM V5 (dropdown menu)

The 'Mcu and Firmware Package' section contains the following fields:

- Mcu Reference:** STM32F429ZITx
- Firmware Package Name and Version:** STM32Cube FW_F4 V1.9.0

At the bottom right, there are 'Ok' and 'Cancel' buttons.

Exemplo 11 - Configurar ADC, DMA e USB

10. Abra o projeto no Keil

11. Deve-se inserir constante em **main.c** (em **Application User**) entre os rótulos
/* USER CODE BEGIN Includes */ e /* USER CODE END Includes */

```
#define ADC_BUFFER_SIZE 16
```

12. Deve-se inserir variáveis em **main.c** (em **Application User**) entre os rótulos
/* USER CODE BEGIN PV */ e /* USER CODE END PV */

```
extern char VCPInitialized;      // indica interface USB pronta
```

```
uint8_t *USBptr;                // ponteiro para início do buffer de dados amostrados
```

```
uint8_t *USBptr2;               // ponteiro para meio do buffer de dados amostrados
```

```
uint16_t DMA_USB_Buffer[16];    // Buffer que recebe dados do ADC
```

```
uint16_t DMA_USB_teste[]={0x3030, 0x3131, 0x3232, 0x3333, 0x3434, 0x3535,  
    0x3636, 0x3737, 0x3838, 0x3939, 0x4141, 0x4242, 0x4343, 0x4444,  
    0x4545,0x4646};             // buffer que simula dados amostrados
```

```
uint32_t flag_buffer=0;         // flag para envio do buffer
```

Exemplo 11 - Configurar ADC, DMA e USB

13. Deve-se declarar função em **main.c** (em **Application User**) entre os rótulos `/*USER CODE BEGIN PFP */` e `/* USER CODE BEGIN PFP */`

```
extern uint8_t CDC_Transmit_HS(uint8_t*, uint16_t);
```

14. Deve-se inserir funções em **main.c** (em **Application User**) entre os rótulos `/* USER CODE BEGIN 2 */` e `/* USER CODE END 2 */`

```
USBptr = (uint8_t*) DMA_USB_teste;
```

```
USBptr2 = (uint8_t*)&DMA_USB_teste[8];
```

```
while(!VCPInitialized) {
```

```
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_14); // led red pisca se USB off
```

```
    HAL_Delay(500); }
```

```
HAL_TIM_Base_Start_IT(&htim2);
```

```
HAL_ADC_Start_DMA(&hadc1, (uint32_t *)DMA_USB_Buffer, (uint32_t)  
    ADC_BUFFER_SIZE );
```

Exemplo 11 - Configurar ADC, DMA e USB

13. Deve-se declarar função em **main.c** (em **Application User**) dentro do loop while(1)

```
if ( flag_buffer == 1) {          CDC_Transmit_HS(USBptr,(uint16_t) ADC_BUFFER_SIZE);
                                flag_buffer=0;
                                };

if ( flag_buffer == 2) {          CDC_Transmit_HS(USBptr2,(uint16_t) ADC_BUFFER_SIZE);
                                flag_buffer=0;
                                };
```

14. Deve-se inserir funções em **main.c** (em **Application User**) entre os rótulos `/* USER CODE BEGIN 4 */` e `/* USER CODE END 4 */`

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)      {
    flag_buffer=2;
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);    }

void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc)  {
    flag_buffer=1;
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);  }
```

Exemplo 11 - Configurar ADC, DMA e USB

15. Deve-se declarar variável em **usbd_cdc_if.c** (em *Application User*) entre os rótulos `/* USER CODE BEGIN PRIVATE_TYPES */` e `/* USER CODE END PRIVATE_TYPES */`

```
char VCPInitialized=0;
```

16. Deve-se alterar valor em **usbd_cdc_if.c** (em *Application User*) entre os rótulos `/* USER CODE BEGIN PRIVATE_DEFINES */` e `/* USER CODE END PRIVATE_DEFINES */`

```
#define APP_TX_DATA_SIZE 16                //!!!!!!!!!!!!!!!!!!!!!! MODIFICAR -- ESTAVA 4
```

Exemplo 11 - Configurar ADC, DMA e USB

17. Deve-se atribuir valor a variável em **usbd_cdc_if.c** (em *Application User*) nas **funções apresentadas** abaixo

```
static int8_t CDC_Init_HS(void)
{
    /* USER CODE BEGIN 8 */
    /* Set Application Buffers */
    USBDCDC_SetTxBuffer(&hUsbDeviceHS, UserTxBufferHS, 0);
    USBDCDC_SetRxBuffer(&hUsbDeviceHS, UserRxBufferHS);
```

VCPIinitialized=1; //!!!!!!!!!!!!!!!!!!!!!! INSERIR

```
    return (USBDCDC_OK);
    /* USER CODE END 8 */
}
```

```
static int8_t CDC_DeInit_HS(void)
{
    /* USER CODE BEGIN 9 */
```

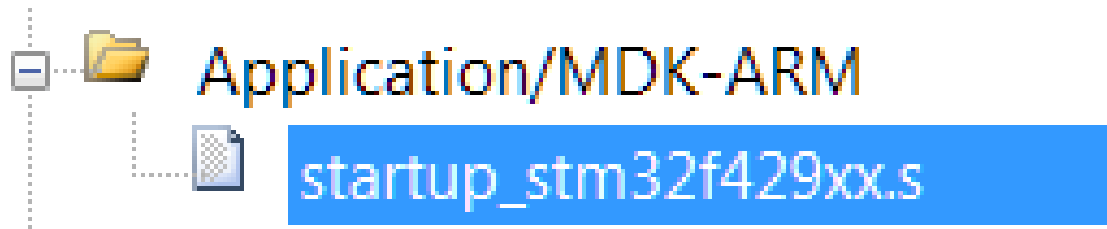
VCPIinitialized=0; //!!!!!!!!!!!!!!!!!!!!!! INSERIR

```
    return (USBDCDC_OK);
    /* USER CODE END 9 */
}
```

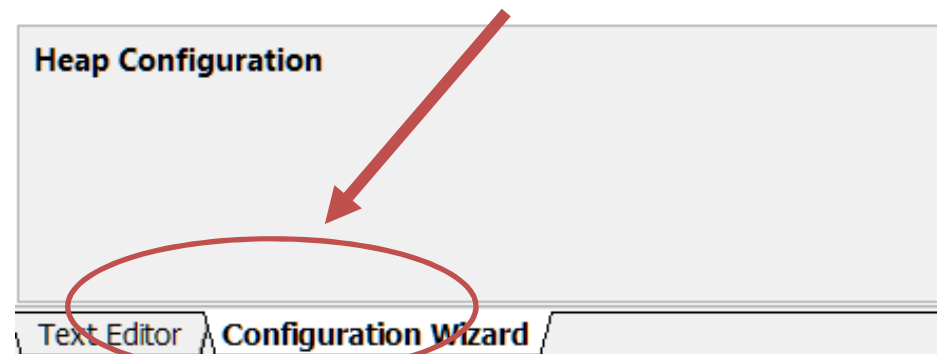
Exemplo 11 - Configurar ADC, DMA e USB

Expand All		Collapse All		Help	
Option			Value		
Stack Configuration					
Stack Size (in Bytes)			0x0000 0400		
Heap Configuration					
Heap Size (in Bytes)			0x0000 0400		

18. Abra o arquivo indicado abaixo:



19. Selecione o tab Confituration Wizard, Como mostrado ao lado.



20. Altere os valores do *Stack* e *Heap* para 0x400 (ao lado no topo).

Exemplo 11 - Configurar ADC, DMA e USB

Para ver a transmissão sendo executada, conecte a segunda USB do kit ao PC, abra o programa de recepção serial (Tera Term ou Bray Terminal) para visualizar a recepção dos dados. Reset o kit.

Caso haja problema na conexão, o led vermelho pisca. Caso a transmissão esteja sendo realizada, o led verde pisca.