

# **EEL7030 - Microprocessadores**



**LCS**

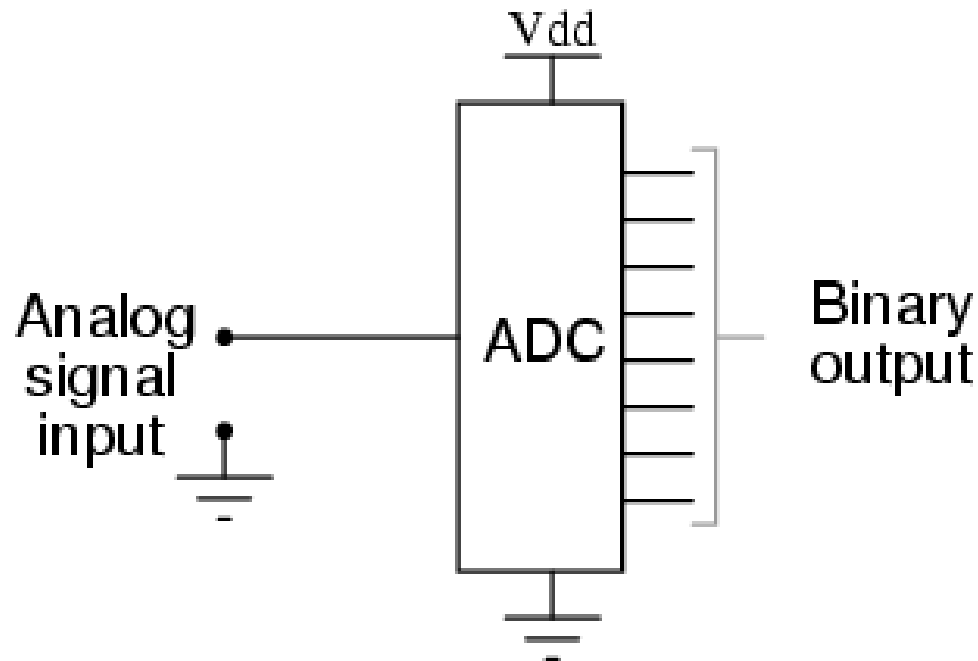
Laboratório de  
Comunicações  
e Sistemas  
Embarcados

**Prof. Raimes Moraes**

**EEL - UFSC**

## ***Analog to Digital Converter (DAC)***

O conversor analógico-digital converte sinal de tensão em valor binário. Como exemplo de emprego, tem-se a conversão de sinal elétrico proveniente de microfone em dado binário que pode ser armazenado em arquivo para transmissão via Internet.

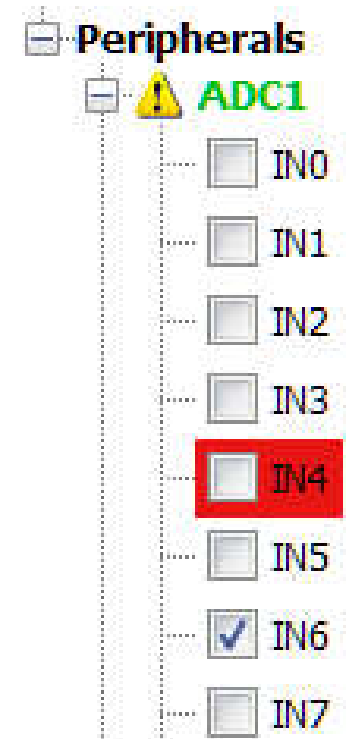


## Exemplo 9 – Configurar **ADC** para *polling*

### ❑ **Objetivo:**

- Configurar ADC no STM32CubeMX para amostrar sinal gerado pelo DAC
- Gerar código e utilizar funções HAL
- Configurar STMStudio para visualizar dados do ADC

1. Abra projeto anterior no STMCubeMX: - Selecione ADC IN6



## Exemplo 9 – Configurar **ADC** para *polling*

2. Conectar saída do DAC (PA4) na entrada 6 (IN 6) do ADC1 (PA6)

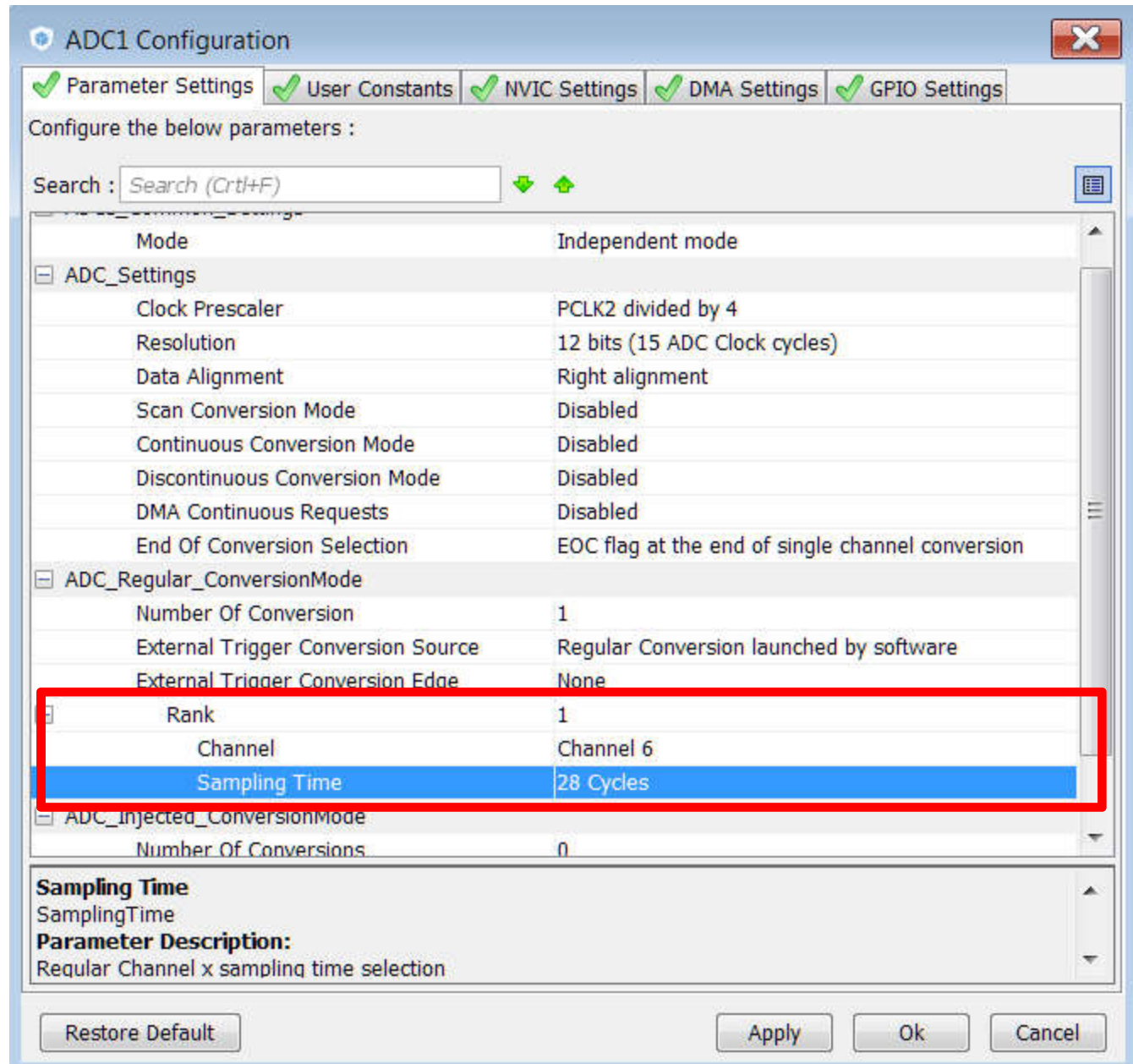


## Exemplo 9 – Configurar ADC para *polling*

### 3. Configurar ADC:

- TAB>Configuration>Anal og>ADC1>Parameter Settings
- Observar alterações do quadro ao lado
- Pressionar OK

OBS: Manter as demais configurações do exemplo anterior



## Exemplo 9 – Configurar **ADC** para *polling*

4. Forneça dados para geração do código:

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

5. Gere *template* para código:

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

The screenshot shows the 'Project Settings' dialog box with the 'Project' tab selected. 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 9 – Configurar ADC para *polling*

6. Tendo aberto o projeto no Keil, acrescente as variáveis abaixo em **main.c** (em **Application User**) entre os rótulos */\* USER CODE BEGIN PV \*/* e */\* USER CODE END PV \*/*

```
/* USER CODE BEGIN PV */
```

```
int value_adc;
```

```
int *address_value_adc; // utilizado para identificar endereço de value_adc - próximos slides
```

7. Inserir funções em **main.c** (em **Application User**) substituindo entre os rótulos */\* infinite loop \*/* e */\* USER CODE END 3 \*/* o código abaixo:

```
/* Infinite loop */
```

```
    address_value_adc=&value_adc;
```

```
/* USER CODE BEGIN WHILE */
```

```
while (1)
```

```
{
```

```
/* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 3 */
```

```
    HAL_ADC_Start(&hadc1);
```

```
    HAL_ADC_PollForConversion(&hadc1,10);
```

```
    value_adc=HAL_ADC_GetValue(&hadc1);
```

```
    HAL_Delay(1);
```

```
}
```

```
/* USER CODE END 3 */
```

## Exemplo 9 – Configurar **ADC** para *polling*

8. Compile o código, conecte o kit ao PC e inicie a depuração do programa.

**OBS:** Antes de conectar o kit no PC, lembre-se de conectar saída do DAC (PA4) na entrada 6 (IN 6) do ADC1 (PA6)





# STM-STUDIO-STM32

(<http://www.st.com/web/en/catalog/tools/PF251373>)

## visualização de formas de onda

- ☐ STM Studio comunica-se com processadores da ST (32bits) via ST-LINK
- ☐ Seu propósito é depurar código em tempo real, permitindo ler e mostrar o conteúdo de variáveis
- ☐ Possui ferramentas de visualização gráfica para acompanhar alteração de conteúdo das variáveis

## Exemplo 9 – Configurar STMStudio

- O acesso a variável **value\_adc** requer saber seu endereço na memória. Este pode ser encontrado no Keil no modo *debug*, colocando a variável **address\_value\_adc** em **watch1** como mostrado abaixo. Execute o programa no modo de depuração até esta variável ser atualizada durante a execução do programa.
  - OBS: no caso abaixo é 0x20000008 (pode ser diferente para outras otimizações)

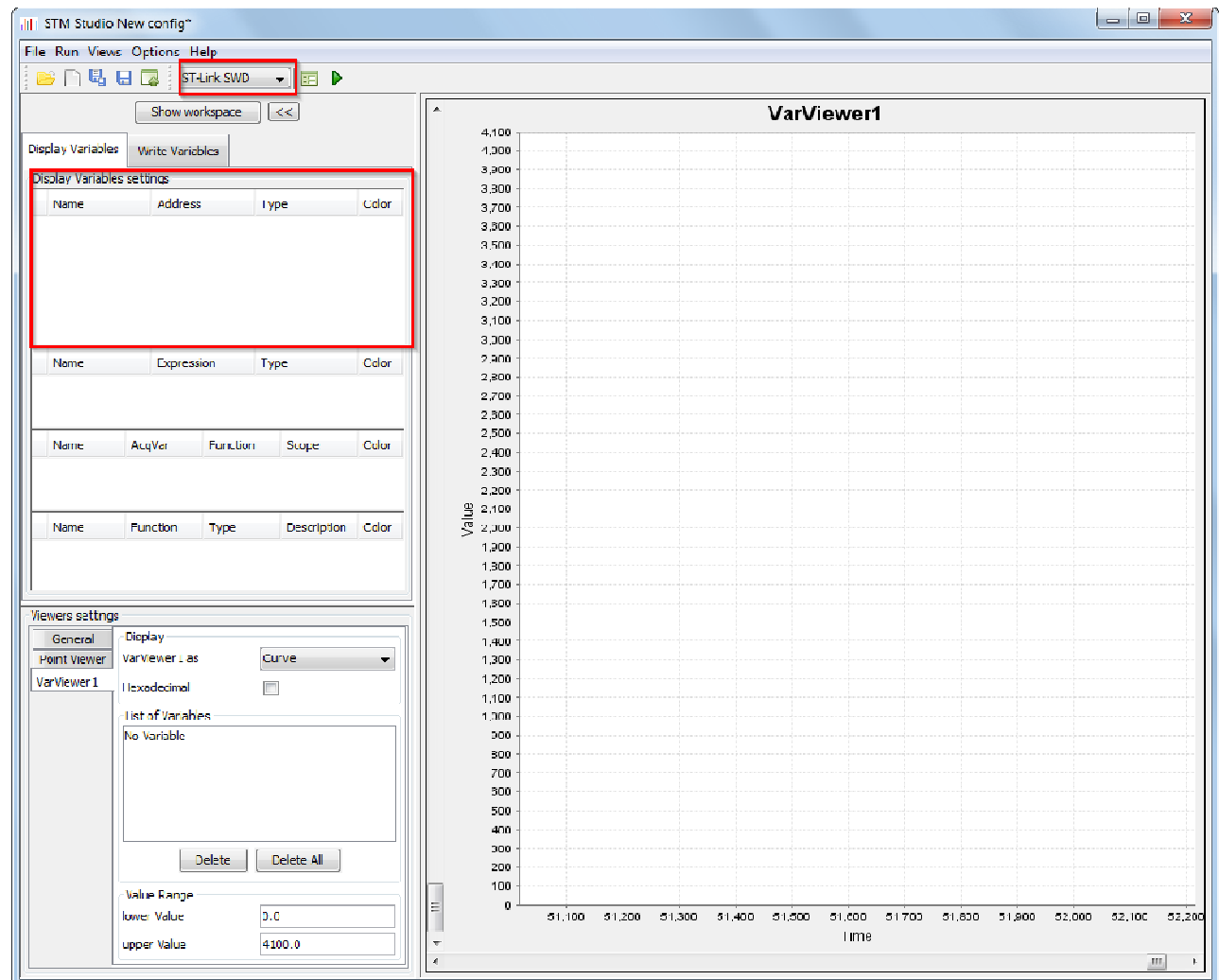
The screenshot shows the STMStudio interface. On the left, a context menu is open over the code line `address_value_adc=&value_adc;`. The menu includes options like 'Split Window horizontally', 'Insert '#include file'', 'Go to Headerfile "main.h"', 'Show Disassembly at 0x08001CDA', 'Set Program Counter', 'Insert/Remove Breakpoint' (F9), 'Enable/Disable Breakpoint' (Ctrl+F9), 'Go To Definition Of 'address\_value\_adc'', 'Go To Reference To 'address\_value\_adc'', and 'Add 'address\_value\_adc' to...'. The 'Add 'address\_value\_adc' to...' option is highlighted, and a sub-menu is visible showing 'Watch 1'.

On the right, the 'Watch 1' window displays a table with the following data:

Name	Value	Type
value_adc	0x0000016F	int
address_value_adc	0x20000008 &value_adc	int *
<Enter expression>		

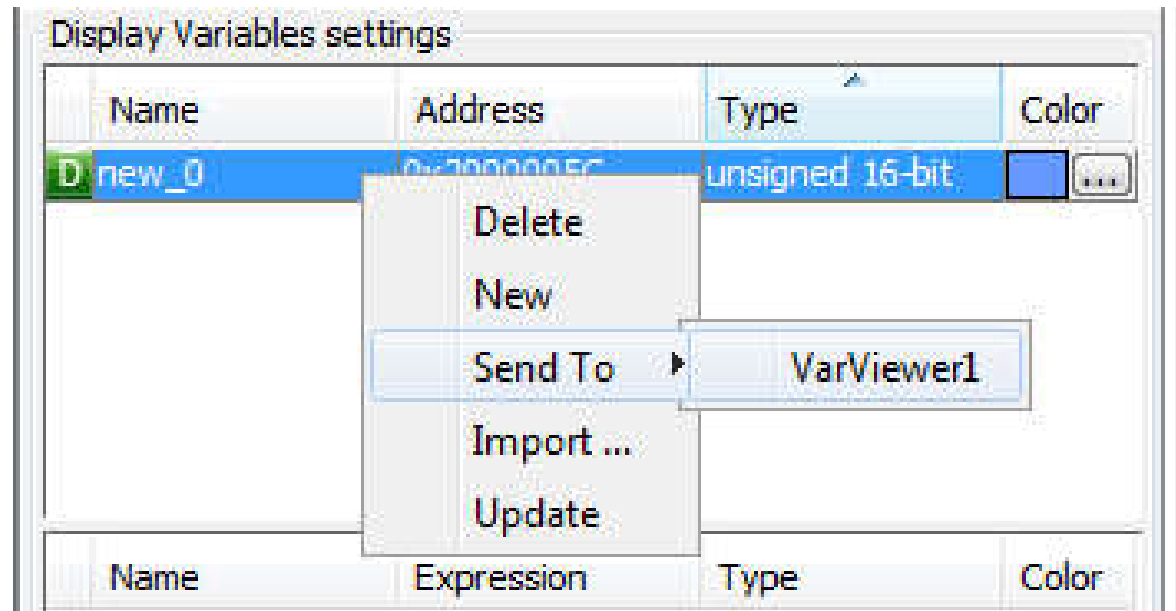
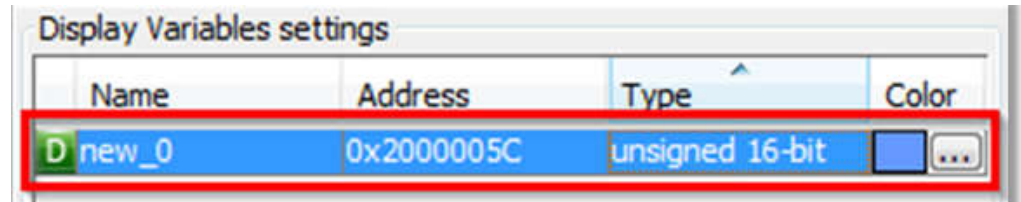
## Exemplo 9 – Configurar STMStudio

- ❑ Abra STMStudio
- ❑ Certifique-se da seleção de ST Link SWD (1º quadro vermelho)
- ❑ Clique sobre a área *Display variable settings* (2º quadro vermelho)
- ❑ Selecione NEW



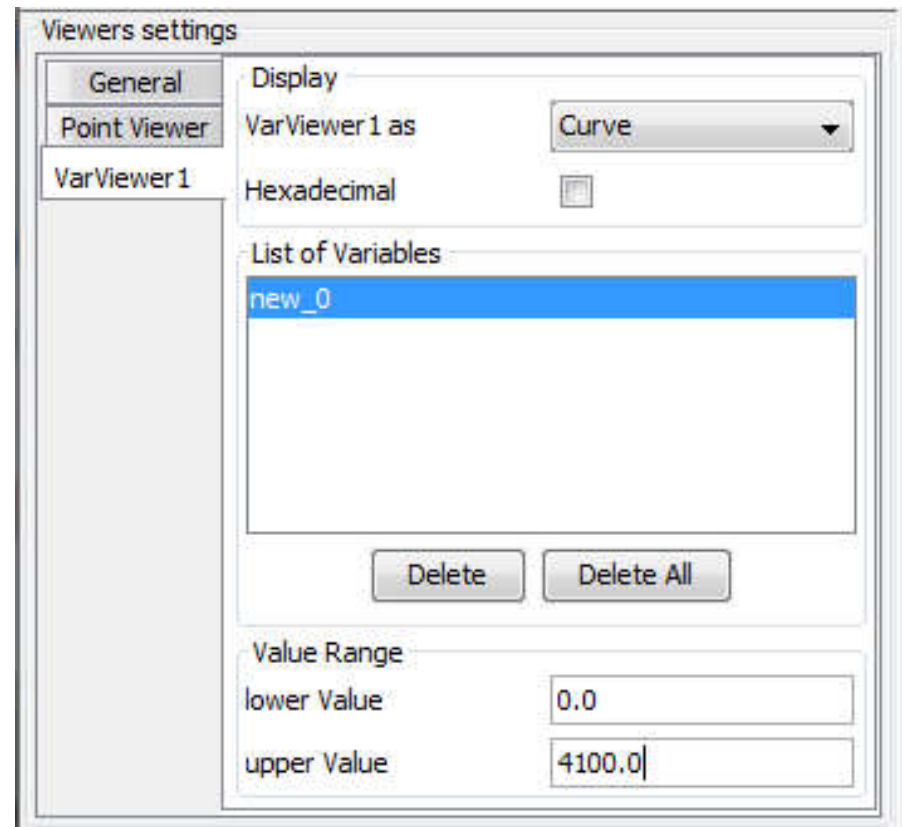
## Exemplo 9 – Configurar STMStudio

- ☐ Especifique campo **Address**  
com o endereço de value\_adc  
0x20000008
- ☐ Especifique **Type** = unsigned16bit
- ☐ Clique na linha indicada ao lado
- ☐ Selecione **Send To**  
**VarViewer1**



## Exemplo 9 – Configurar STMStudio

- ❑ É necessário definir escala vertical
- ❑ Selecione Viewers settings (canto inferior esquerdo)
- ❑ Informe máxima amplitude (12bits):4096



## Exemplo 9 – Configurar STMStudio

- ❑ Para ver forma de onda amostrada pelo ADC no grande quadro em vermelho, pressione play (seta verde)

- ❑ OBS:

- Keil não deve estar no modo debug
- Programa deve estar sendo executado no kit. Resete o kit.

