



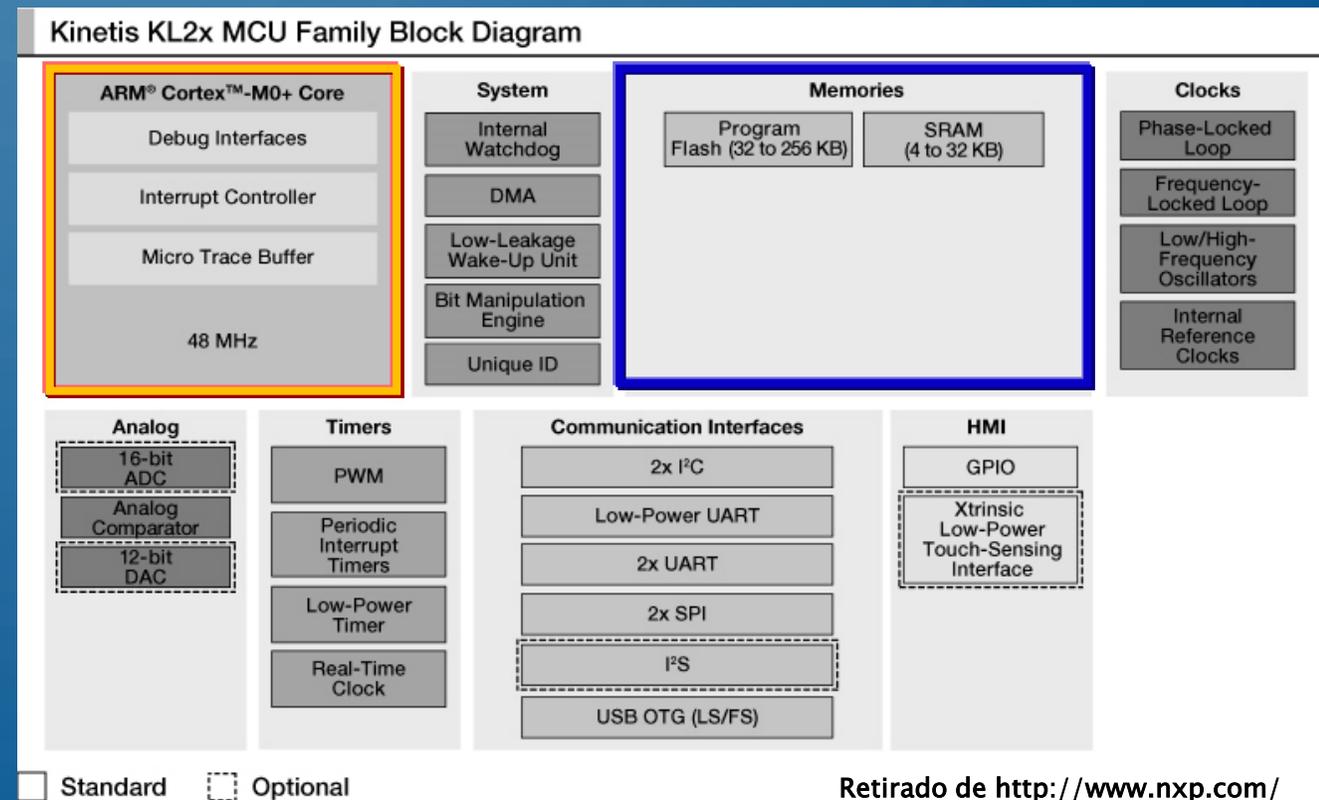
MANUAL RÁPIDO

# KIT NXP FREEDOM FRDM – KL25Z

PSI 3212 – LABORATÓRIO DE CIRCUITOS ELÉTRICOS

# ARQUITETURA ARM CORTEX-M

- Baixo consumo e alta integração
- Alto desempenho e *clock* (acima de 16,0 [MHz])
- 32 bits
- Alta quantidade de memória
- Amplo espectro de periféricos
- NXP Kinetis KL25Z
  - ARM Cortex-M0+
  - Single core 48,0 MHz de clock
  - 128,0 KB FLASH ROM
  - 16 KB SRAM

Retirado de <http://www.nxp.com/>

# KIT NXP FREEDOM FDRM-KL25Z

- Chip ARM CORTEX-M0+ NXP Kinetis MKL25Z128VLK4
- Sensores
  - Acelerômetro MEMS triaxial
  - Sensor *touch slider* capacitivo
- Atuadores
  - Um LED RGB (três LEDs – RGB)
- Interface USB OTG ligada direto ao microcontrolador KL25Z
  - Terminais GPIO (*General Purpose Input and Output*)
  - Pinagem compatível com padrão Arduino Revisão 3 (R3)
- Cabo de programação OpenSDA embutido (outro ARM!) – interface USB SDA



# CUIDADOS NA UTILIZAÇÃO!

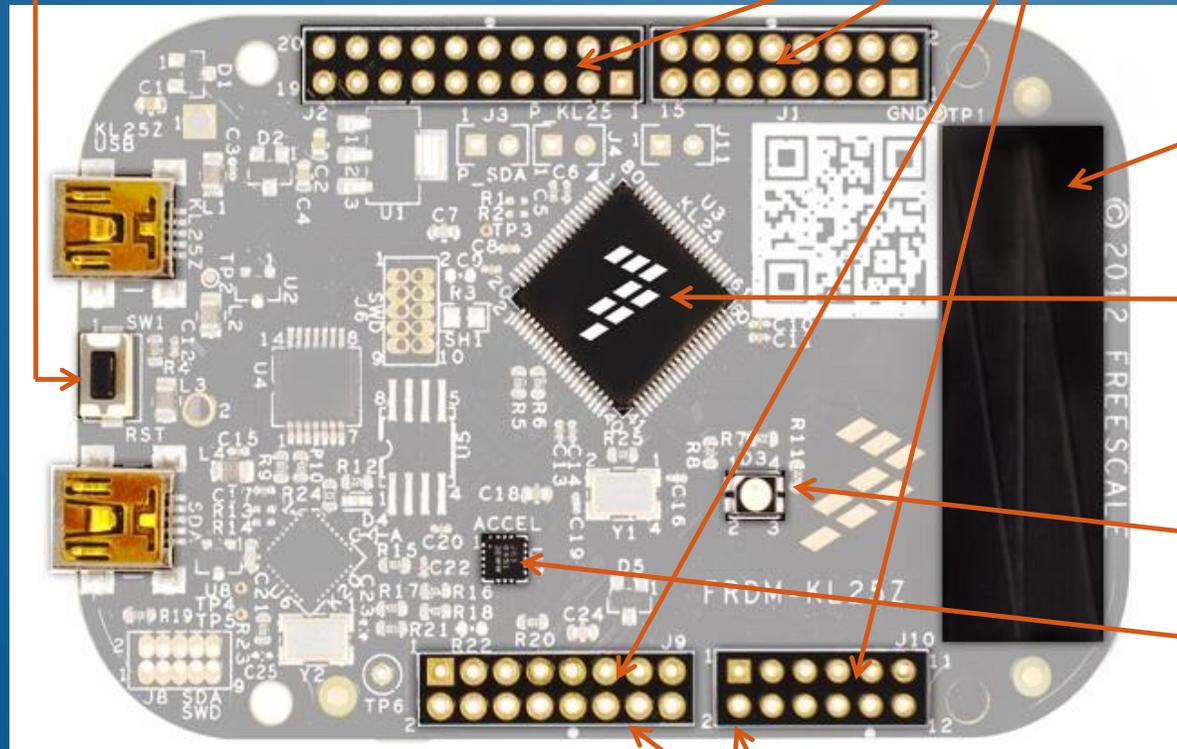


- Kit projetado para uso com outros dispositivos e interfaces COMPATÍVEIS
- Terminais e conectores de expansão NÃO podem ser ligados a qualquer componente, de qualquer forma, com qualquer tensão ou especificação
- Enquanto energizado, mantenha o kit afastado de objetos metálicos, condutores, fios, grafite, líquidos,...

# PLACA DO KIT E SEUS RECURSOS

Botão de RESET de programas

Acessos às Entradas e Saídas da placa



“Touch slider” Capacitivo

Microcontrolador

LED RGB

Acelerômetro 3D  
Micro-máquina MEMS:  
Incline a placa.

Conectores de  
expansão Arduino R3

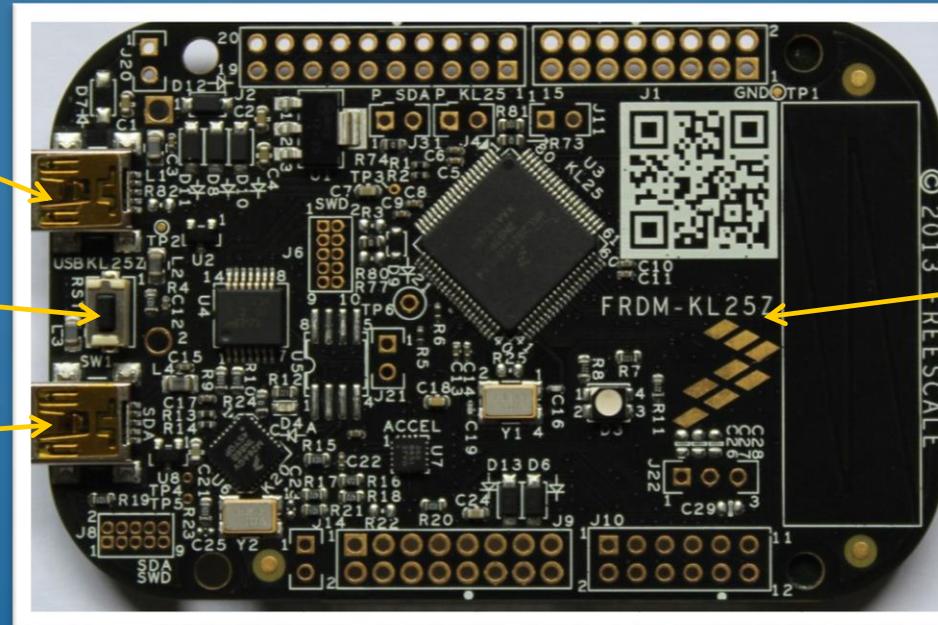
# PLACA DO KIT / CONEXÃO AO PC

USB – mini B  
USB KL25Z

Botão de reset

USB – mini B  
USB SDA

Usar esta para ligar ao PC



Placa FRDM-KL25Z

Observação: Há duas portas USB: uma denominada USB SDA e outra USB KL25Z.

CABO USB: Conectar ao computador a extremidade padrão A  
e à placa a extremidade padrão mini-B



Atenção: USB tipo Micro-B, usado como carregador de smartphones, não é compatível!

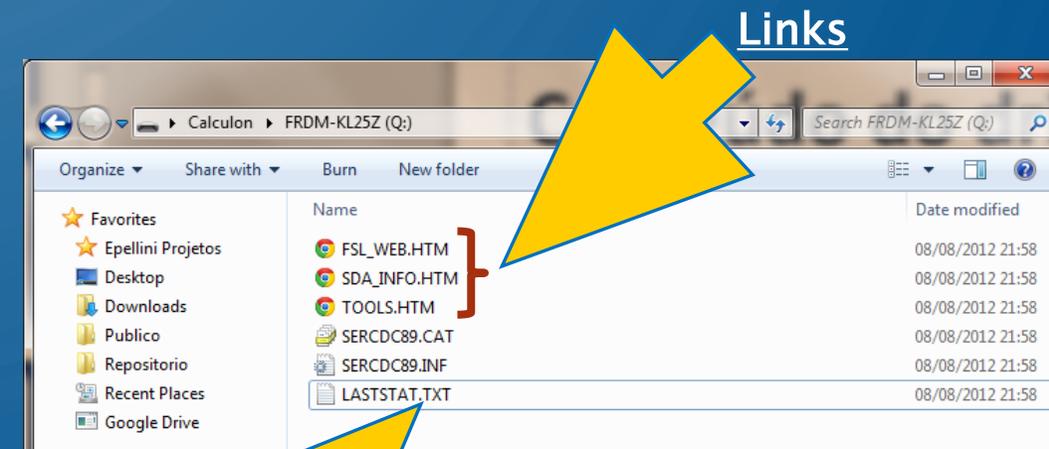
# NO SEU COMPUTADOR

- Surge um flash-drive junto aos demais dispositivos do seu computador
- Esse drive será utilizado para gravar novos programas no microcontrolador KL25Z



## Dentro do Drive

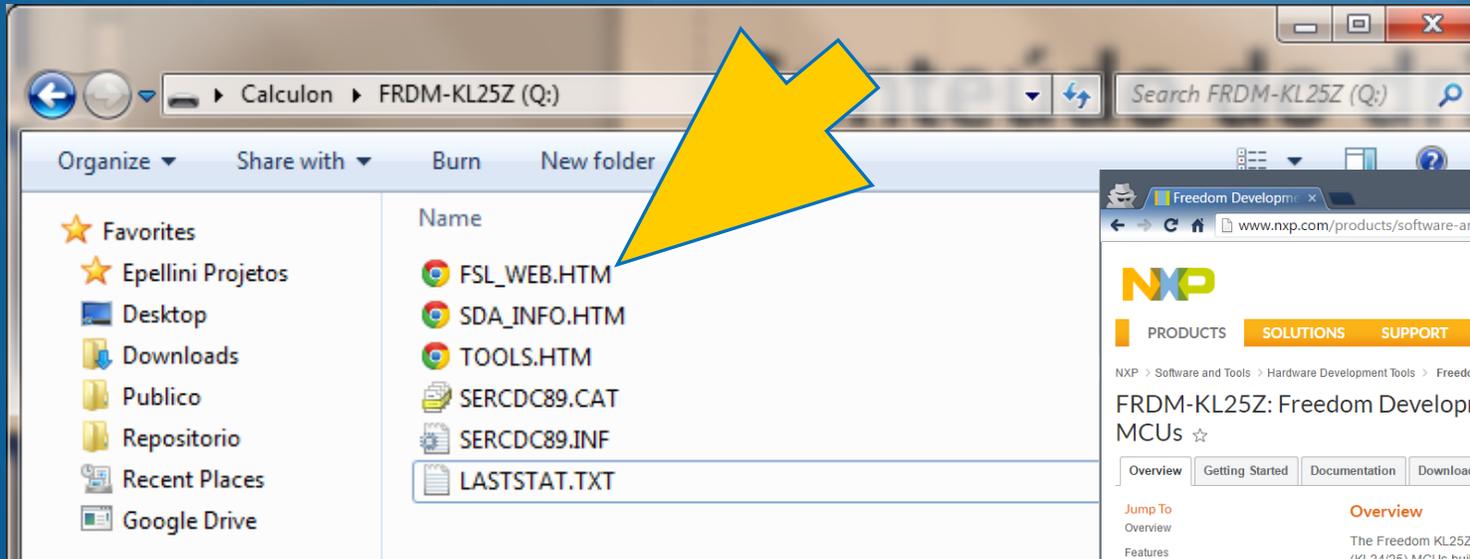
- Alguns arquivos são links para páginas de internet
- Outros arquivos são os drivers de uma porta serial virtual que você precisa instalar no seu Windows



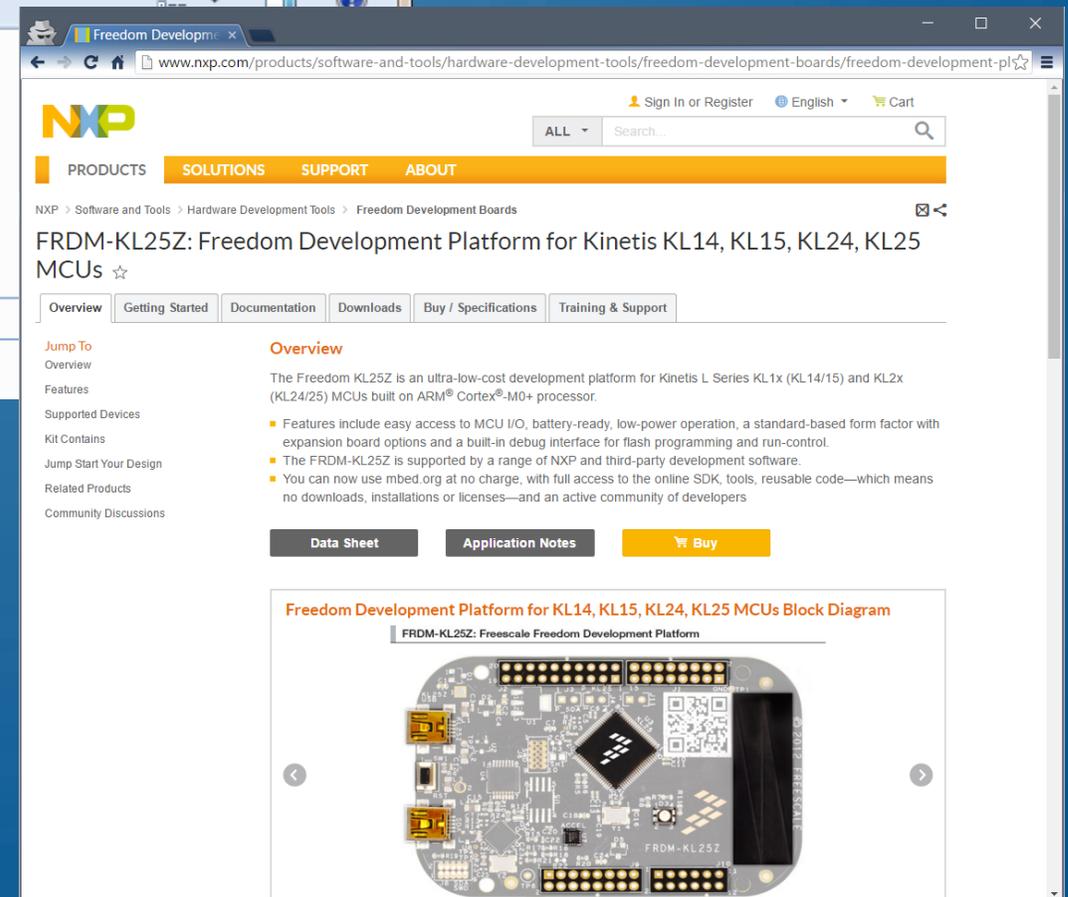
Arquivo de *Status*

# VISITANDO O SITE DA NXP

- No drive FRDM-KL25Z, clique duas vezes sobre o arquivo FSL\_WEB.HTM

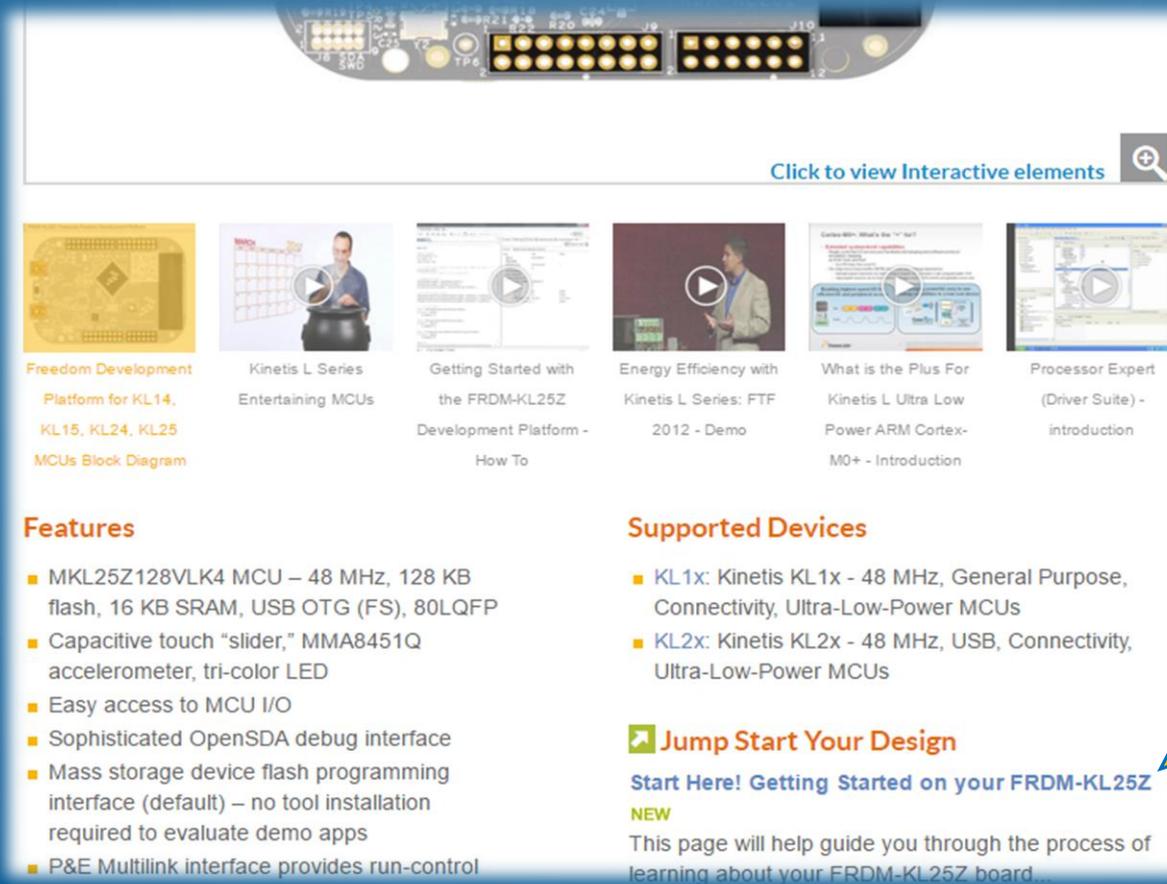


- Seu browser deve abrir a página do kit no site da NXP Freedom Development:



# QUICK START

- No site da NXP *Freedom Development* aberto, na seção “*Jump Start Your Design*”, clique sobre “*Start Here! Getting Started on your FRDM-KL25Z*”



Click to view Interactive elements

Freedom Development Platform for KL14, KL15, KL24, KL25 MCUs Block Diagram

Kinetis L Series Entertaining MCUs

Getting Started with the FRDM-KL25Z Development Platform - How To

Energy Efficiency with Kinetis L Series: FTF 2012 - Demo

What is the Plus For Kinetis L Ultra Low Power ARM Cortex-M0+ - Introduction

Processor Expert (Driver Suite) - introduction

### Features

- MKL25Z128VLK4 MCU – 48 MHz, 128 KB flash, 16 KB SRAM, USB OTG (FS), 80LQFP
- Capacitive touch “slider,” MMA8451Q accelerometer, tri-color LED
- Easy access to MCU I/O
- Sophisticated OpenSDA debug interface
- Mass storage device flash programming interface (default) – no tool installation required to evaluate demo apps
- P&E Multilink interface provides run-control

### Supported Devices

- KL1x: Kinetis KL1x - 48 MHz, General Purpose, Connectivity, Ultra-Low-Power MCUs
- KL2x: Kinetis KL2x - 48 MHz, USB, Connectivity, Ultra-Low-Power MCUs

### Jump Start Your Design

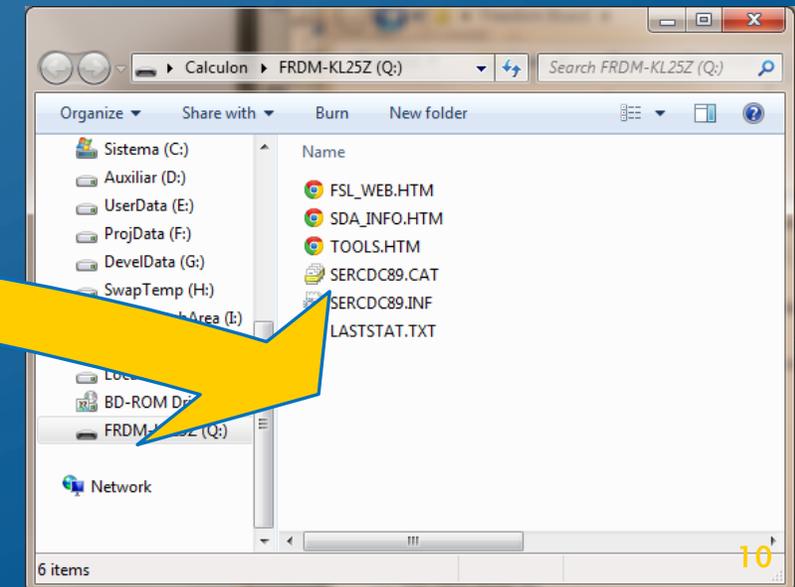
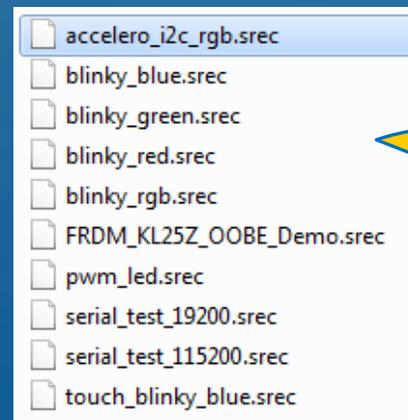
**Start Here! Getting Started on your FRDM-KL25Z**  
NEW

This page will help guide you through the process of learning about your FRDM-KL25Z board...

- Selecione a aba “Downloads” e baixe todos os arquivos, principalmente o *Quick Start Package* (FRDM-KL25Z\_QSP.ZIP)

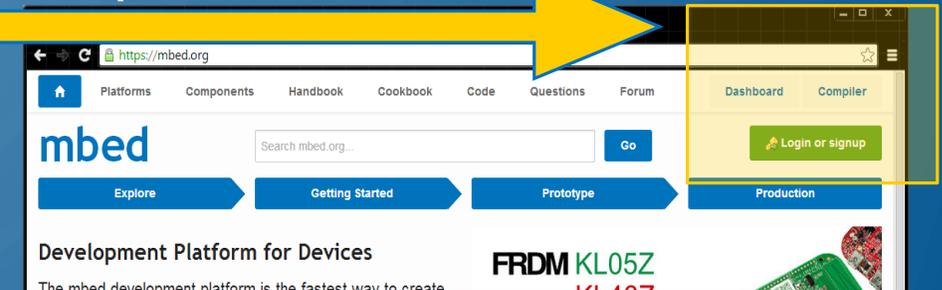
# TESTE DE PROGRAMAS EXEMPLO

- Gravação de novos programas via Flash-Drive
- Descomprima o arquivo “FRDM Quick Start Package FRDM-KL25Z\_QSP.ZIP” em um diretório no seu computador
- Procure a pasta “Precompiled Examples”
- Conecte o Cabo USB
- Arraste qualquer arquivo com extensão .SREC para dentro do drive FRDM-KL25Z
- Observe o funcionamento de cada programa



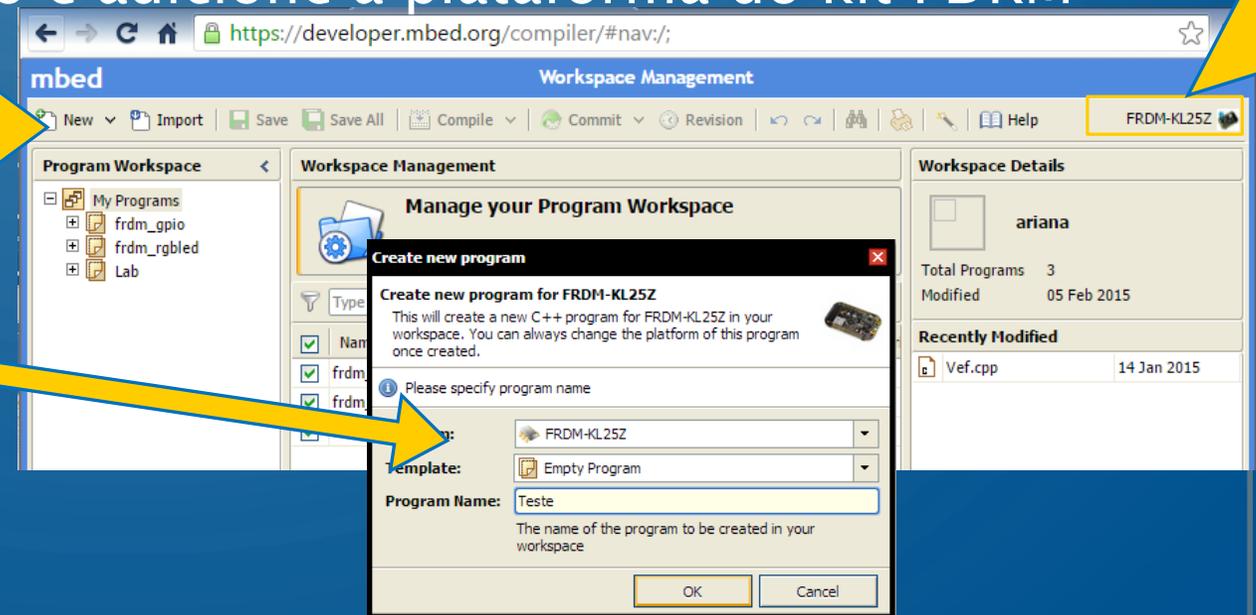
# MBED – PROGRAMAÇÃO NA NUVEM

- Acesse o site de desenvolvimento <http://developer.mbed.org>
- Clique em “*login or signup*” e crie uma conta pessoal
- Explore o Dashboard e o Compiler



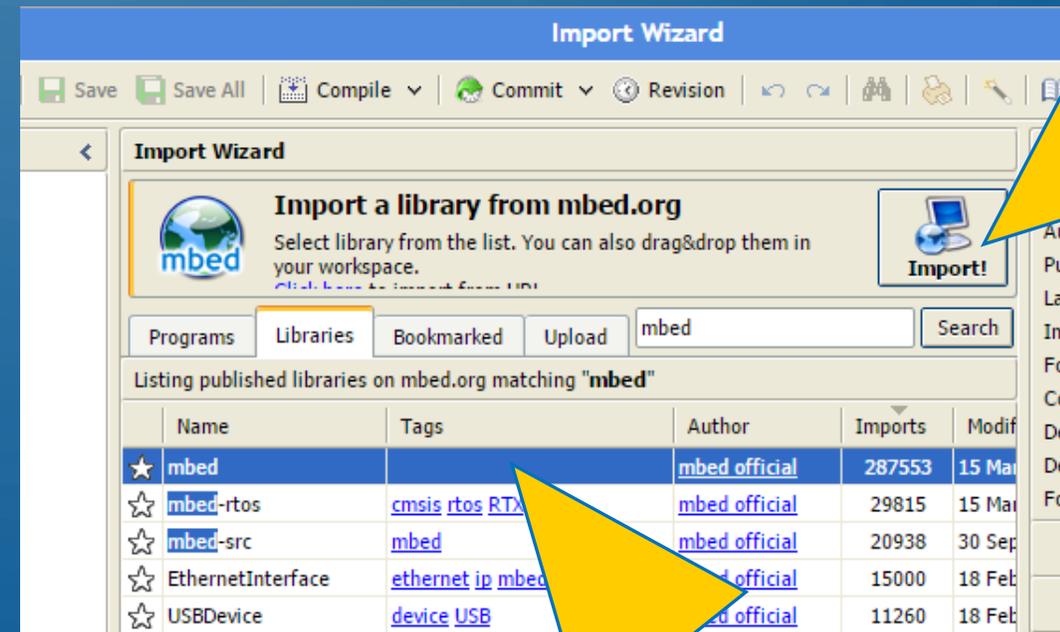
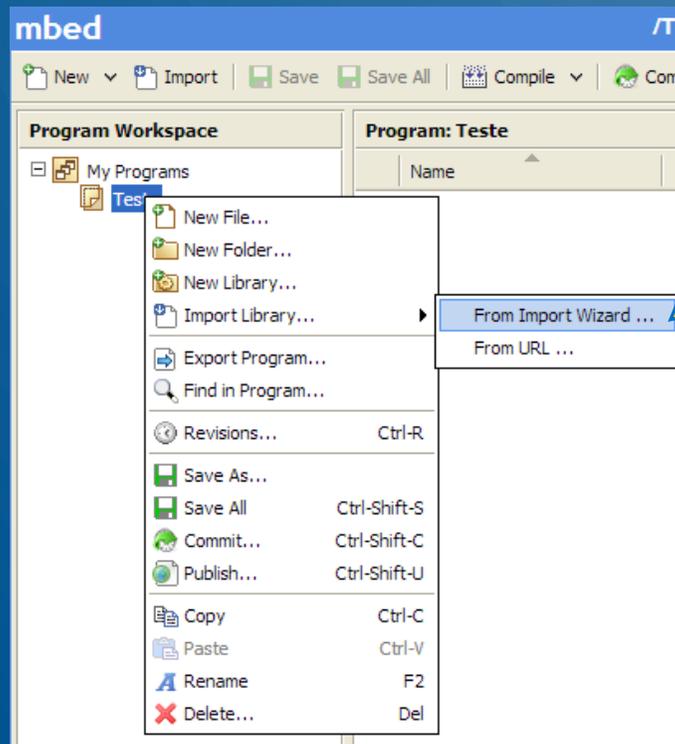
## MBED – COMPILER

- Suporte a várias plataformas, usuários e grupos
- Clique no canto superior direito e adicione a plataforma do kit FRDM-KL25Z no seu compilador
- Clique em “*New*”, escolha a plataforma “FRDM-KL25Z” e o template “Empty Program”
- Escolha um nome para seu primeiro programa (Teste) e clique em OK



# MBED – COMPILER – BIBLIOTECA MBED

- Na seção “Program Workspace” clique com o botão direito sobre o seu programa (Teste), escolha “Import Library...”, “From Import Wizard...”
- Na janela “Import a library from mbed.org”, digite “mbed” no campo e clique em “search”. Ao aparecer a lista, escolha a opção “mbed”, do autor “mbed official”, e clique no botão Import. Aceite as demais opções como padrão.

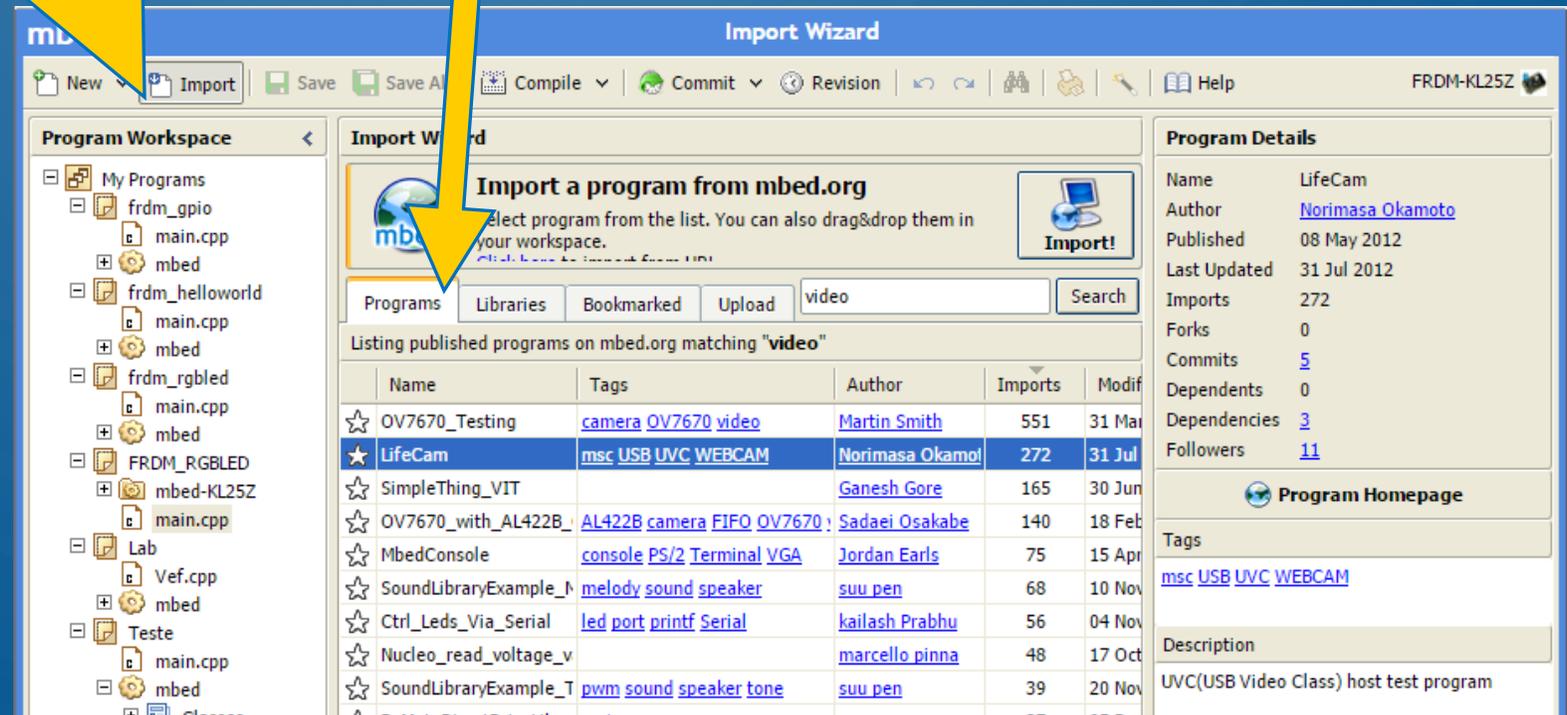


# MBED – COMPILER – “IMPORT”

Você pode escolher bibliotecas e até mesmo programas já desenvolvidos para os mais variados fins, como comunicação com câmeras de video, com o Matlab, com chips comerciais etc.

- Selecione “Import” e depois “Programas” ou “Libraries”, procure por palavra-chave e  comece a testar!

Mas deixe isso para quando estiver mais familiarizado com o kit, por agora, vamos seguindo os passos iniciais



The screenshot shows the mbed IDE's 'Import Wizard' window. The 'Programs' tab is selected, and a search for 'video' has been performed. The 'LifeCam' program is highlighted in the list. The 'Program Details' panel on the right shows information for the 'LifeCam' program.

Name	Tags	Author	Imports	Modif
OV7670_Testing	camera OV7670 video	Martin Smith	551	31 Mai
★ LifeCam	msc USB UVC WEBCAM	Norimasa Okamoto	272	31 Jul
SimpleThing_VIT		Ganesh Gore	165	30 Jun
OV7670_with_AL422B_	AL422B camera FIFO OV7670	Sadaei Osakabe	140	18 Feb
MbedConsole	console PS/2 Terminal VGA	Jordan Earls	75	15 Apr
SoundLibraryExample_	melody sound speaker	suu pen	68	10 Nov
Ctrl_Leds_Via_Serial	led port printf Serial	kailash Prabhu	56	04 Nov
Nucleo_read_voltage_v		marcello pinna	48	17 Oct
SoundLibraryExample_T	pwm sound speaker tone	suu pen	39	20 Nov

**Program Details**

Name	LifeCam
Author	<a href="#">Norimasa Okamoto</a>
Published	08 May 2012
Last Updated	31 Jul 2012
Imports	272
Forks	0
Commits	<a href="#">5</a>
Dependents	0
Dependencies	<a href="#">3</a>
Followers	<a href="#">11</a>

[Program Homepage](#)

Tags

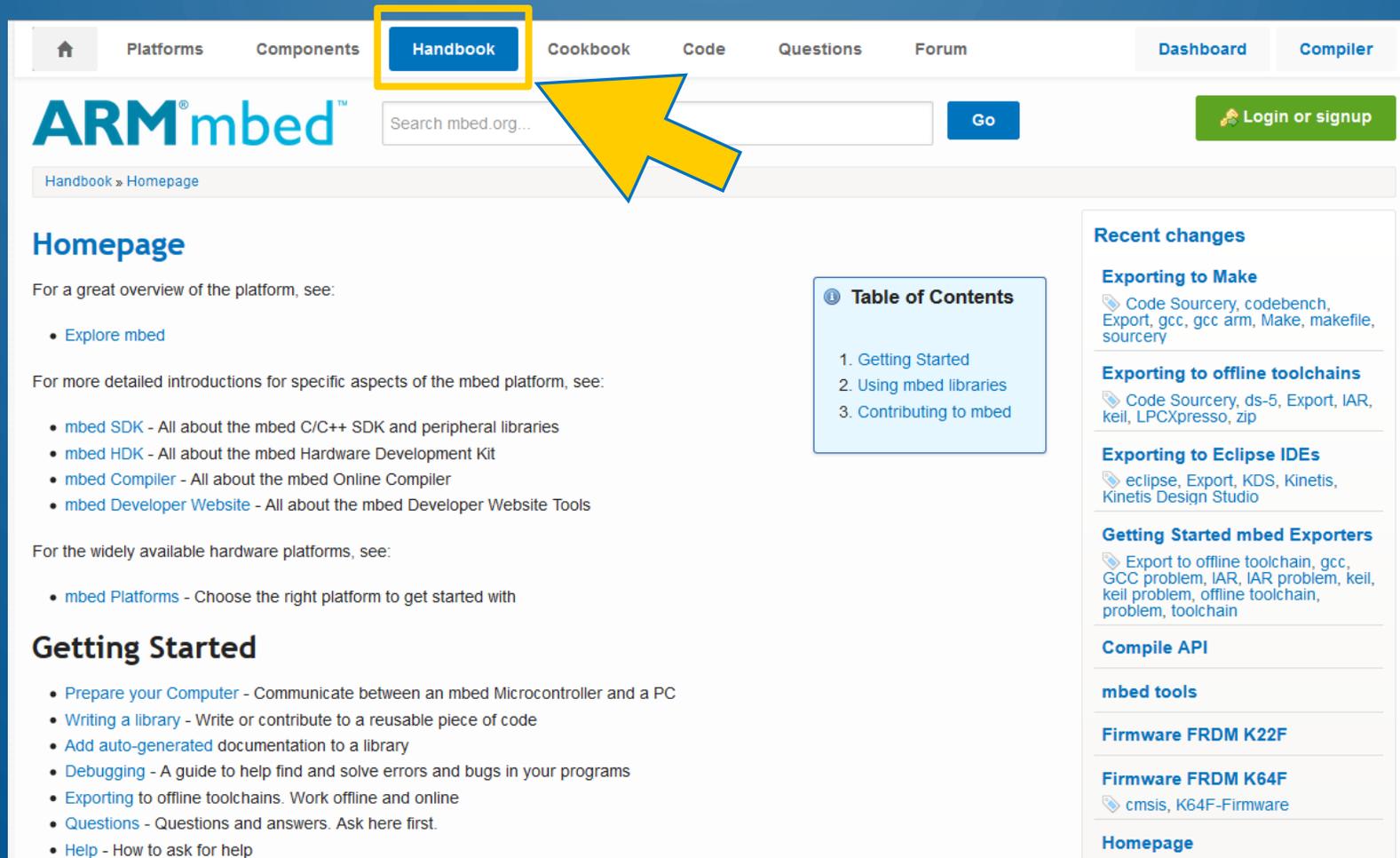
[msc USB UVC WEBCAM](#)

Description

UVC(USB Video Class) host test program

# MBED – HANDBOOK

- Selecione a aba “Handbook” para obter ajuda rápida e completa sobre como utilizar sua placa.



ARM mbed

Search mbed.org... Go Login or signup

Handbook » Homepage

## Homepage

For a great overview of the platform, see:

- [Explore mbed](#)

For more detailed introductions for specific aspects of the mbed platform, see:

- [mbed SDK](#) - All about the mbed C/C++ SDK and peripheral libraries
- [mbed HDK](#) - All about the mbed Hardware Development Kit
- [mbed Compiler](#) - All about the mbed Online Compiler
- [mbed Developer Website](#) - All about the mbed Developer Website Tools

For the widely available hardware platforms, see:

- [mbed Platforms](#) - Choose the right platform to get started with

## Getting Started

- [Prepare your Computer](#) - Communicate between an mbed Microcontroller and a PC
- [Writing a library](#) - Write or contribute to a reusable piece of code
- [Add auto-generated](#) documentation to a library
- [Debugging](#) - A guide to help find and solve errors and bugs in your programs
- [Exporting to offline toolchains](#). Work offline and online
- [Questions](#) - Questions and answers. Ask here first.
- [Help](#) - How to ask for help

### Table of Contents

1. Getting Started
2. Using mbed libraries
3. Contributing to mbed

### Recent changes

#### Exporting to Make

- [Code Sourcery, codebench, Export, gcc, gcc arm, Make, makefile, sourcery](#)

#### Exporting to offline toolchains

- [Code Sourcery, ds-5, Export, IAR, keil, LPCXpresso, zip](#)

#### Exporting to Eclipse IDEs

- [eclipse, Export, KDS, Kinetis, Kinetis Design Studio](#)

#### Getting Started mbed Exporters

- [Export to offline toolchain, gcc, GCC problem, IAR, IAR problem, keil, keil problem, offline toolchain, problem, toolchain](#)

#### Compile API

#### mbed tools

#### Firmware FRDM K22F

#### Firmware FRDM K64F

- [cmsis, K64F-Firmware](#)

#### Homepage

# MBED – COMPILER – PRIMEIRO CÓDIGO FONTE

- Clique (botão direito) no seu projeto (Teste) e escolha “New File...”
- Escolha como nome para o arquivo: main.cpp
- Como conteúdo do arquivo main.cpp, digite seu primeiro programa em C para a plataforma do kit:

```
#include "mbed.h"
DigitalOut myled(LED1);
int main() {
    while(1) {
        myled = 1;
        wait(0.2);
        myled = 0;
        wait(0.2);
    }
}
```

- Atenção à sintaxe.
- Cuidado com maiúsculas e minúsculas.
- Clique em “Compile”
- Se tudo estiver correto, será gerado um arquivo com extensão .bin (onde o seu browser baixa arquivos)
- Salve-o dentro do drive FRDM-KL25Z

# PENSE A RESPEITO E PESQUISE

- O que esse programa faz?
- Para que serve o `#include "mbed.h"`
- O que é `DigitalOut`?
- Porque o programa possui um laço do tipo `while(1)...?`
- O que faz a instrução `wait(0.2)`?
- Quem é `LED1`? Será que existe `LED2`? E `LED3` e `LED4`?
- Mas o que é `LED`?

# PORTA SERIAL VIRTUAL

- Podemos criar uma porta serial virtual entre um computador e uma placa utilizando a USB que os conecta. Para que serve?
  - Para comunicação com o Kit
  - Para “debugar” o programa
- Com a placa conectada a um computador com S.O. Microsoft Windows (XP/Vista/7/8/10), surgem dois itens no Gerenciador de Dispositivos
- Os novos dispositivos necessitam da instalação do driver:  
<http://www.pemicro.com/opensda/>  
PEDrivers\_install.exe (Executar – como administrador – a instalação padrão)

Após a instalação, verifique o número da porta serial (COMx)

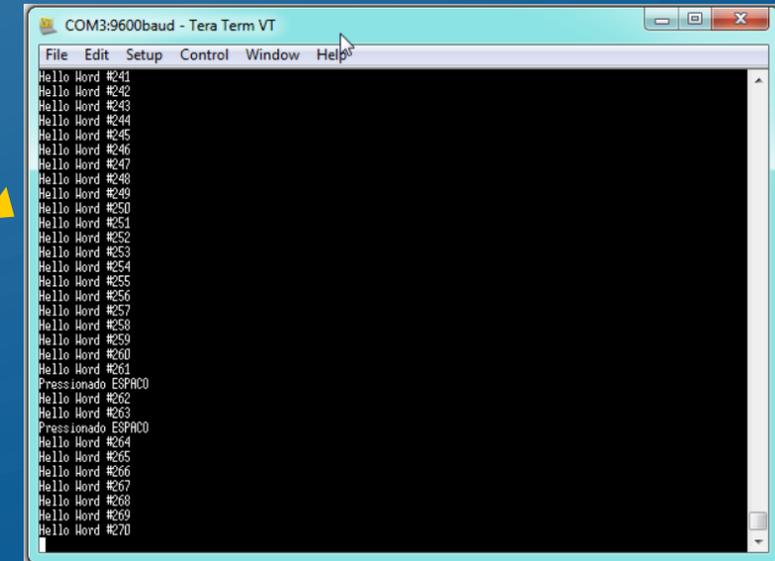
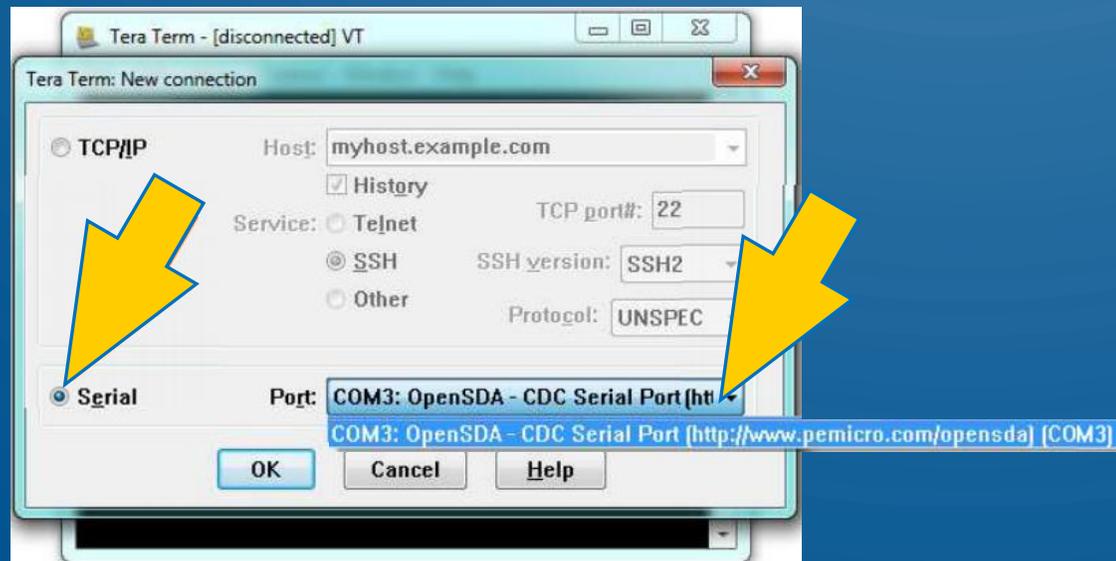


# EMULADOR DE TERMINAL

- Existem vários programas emuladores de terminais de comunicação serial.
- No Windows XP havia o “HyperTerminal”. Esse programa não acompanha mais os pacotes do Microsoft Windows a partir da versão Vista.
- Recomenda-se instalar o programa “Tera Term” para dar suporte às comunicações seriais entre o kit e o computador através da porta serial virtual do kit KL25Z.
- Acesse <http://ttssh2.sourceforge.jp>, faça download da versão 4.9 ou superior e instale em seu computador. **Cuidado:** há versões comerciais desse programa. A versão que estamos apontando o link é gratuita.
- Deve-se prosseguir com a instalação padrão, aceitando todas as configurações sugeridas.

# EXECUTANDO O TERA TERM

- Ao executar o TeraTerm, você será perguntado pela porta de conexão ao dispositivo externo.
  - Escolha a porta COMx da porta serial virtual citada no gerenciador de dispositivos do Windows (e clique “OK”).
- Se as comunicações com o kit estiverem operacionais:
  - Comandos enviados pelo kit (printf e etc.) devem surgir no terminal.
  - Comandos teclados pelo usuário no terminal são enviados de volta para o kit



# DICA DE COMUNICAÇÃO SERIAL

- A comunicação serial pode ser feita mais veloz, permitindo o envio de mais informações a uma velocidade superior que a normal.
  - Basta alterar o programa no MBED para que a serial seja configurada para uma velocidade de símbolos (*baud rate*) superior à tradicional.
  - O padrão de velocidade é 9600 bps. Pode-se mudar para outra.
- O programa TeraTerm **também deve ser reconfigurado**, senão não há comunicação
  - Acesse o menu “Setup”, opção “Serial Port”.
  - Altere o *baud rate* para a mesma taxa usada no MBED.
- As taxas padrão são: 9600, 19200, 38400, 57600 e 115200bps.

**Adicione essas linhas ao seu programa:**

```
Serial pc(USBTX, USBRX); //logo no começo do programa  
pc.baud(115200); //dentro do main()
```

# PINOS, TENSÕES E CORRENTES DISPONÍVEIS

- Um mesmo pino pode assumir diferentes funções, segundo configuração no programa feita pelo usuário:
  - Entrada, que recebe um sinal digital proveniente de um sensor externo (por exemplo um botão), ou;
  - Saída, que envia um sinal digital para acionamento ou comando de um outro dispositivo (por exemplo, um LED ou relé eletromecânico).
- Porta USB alimentada com 5 V – 9 V no pino “Vin”
- Saída regulada de 3.3 V para alimentar periféricos
- 5 V disponível da porta USB (se esta estiver conectada)

# LIMITES OPERACIONAIS DO MICROCONTROLADOR

## 1.4 Voltage and current operating ratings

Table 4. Voltage and current operating ratings

Symbol	Description	Min.	Max.	Unit
$V_{DD}$	Digital supply voltage	-0.3	3.8	V
$I_{DD}$	Digital supply current	—	120	mA
$V_{IO}$	IO pin input voltage	-0.3	$V_{DD} + 0.3$	V
$I_D$	Instantaneous maximum current single pin limit (applies to all port pins)	-25	25	mA
$V_{DDA}$	Analog supply voltage	$V_{DD} - 0.3$	$V_{DD} + 0.3$	V
$V_{USB\_DP}$	USB_DP input voltage	-0.3	3.63	V
$V_{USB\_DM}$	USB_DM input voltage	-0.3	3.63	V
$V_{REGIN}$	USB regulator input	-0.3	6.0	V

Cada terminal individual pode fornecer ou receber corrente com valor até 25,0 [mA], MAS...

...Todos os pinos de GPIO juntos só podem consumir ou fornecer no máximo 100,0 [mA]!!

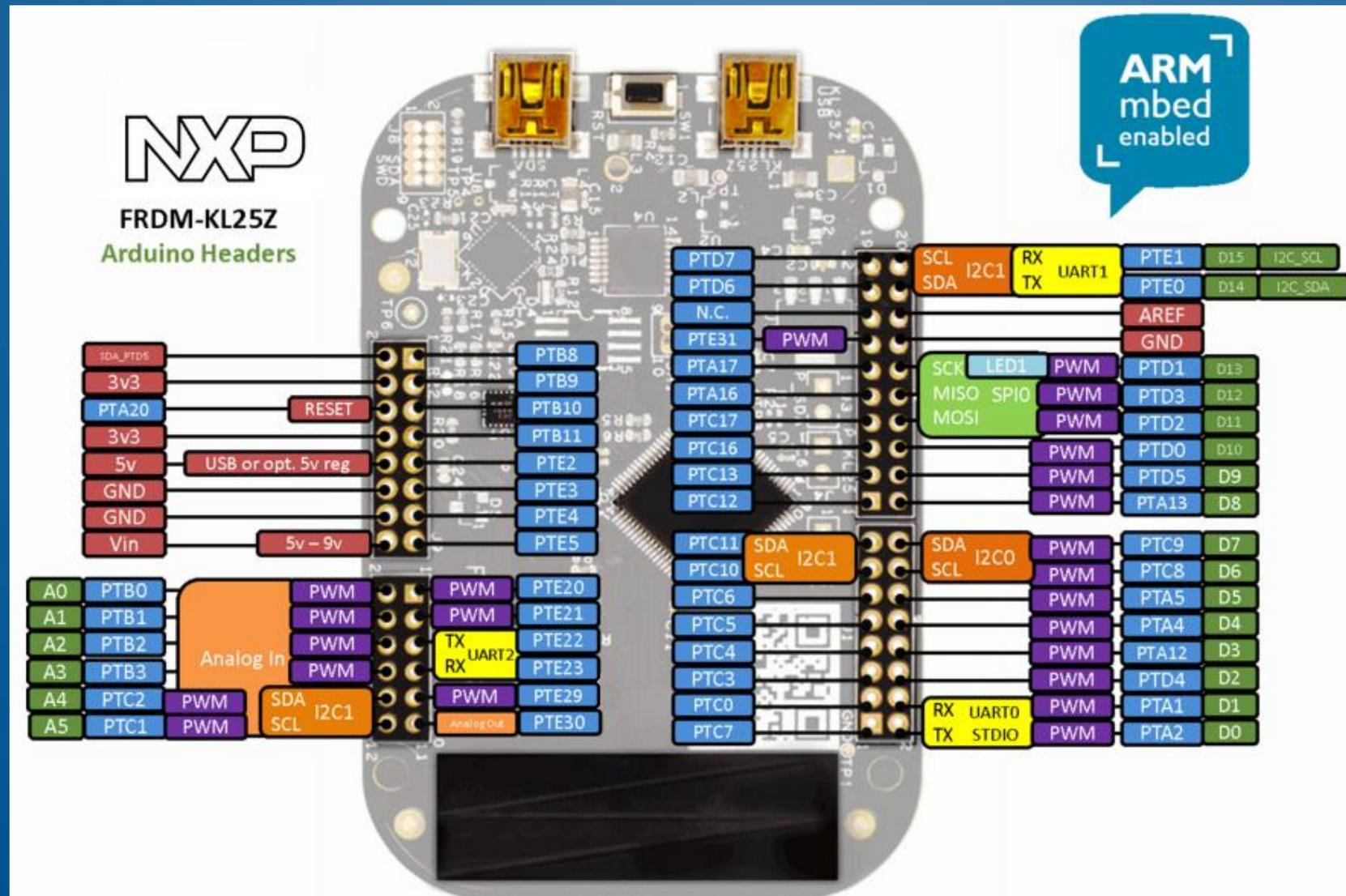
## 5.1.3 Voltage and current operating behaviors

Table 4. Voltage and current operating behaviors

Symbol	Description	Min.	Max.	Unit	
$V_{OH}$	Output high voltage — high drive strength	• $2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $I_{OH} = -10\text{ mA}$	$V_{DD} - 0.5$	—	V
		• $1.71\text{ V} \leq V_{DD} \leq 2.7\text{ V}$ , $I_{OH} = -3\text{ mA}$	$V_{DD} - 0.5$	—	V
	Output high voltage — low drive strength	• $2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $I_{OH} = -2\text{ mA}$	$V_{DD} - 0.5$	—	V
		• $1.71\text{ V} \leq V_{DD} \leq 2.7\text{ V}$ , $I_{OH} = -0.6\text{ mA}$	$V_{DD} - 0.5$	—	V
$I_{OHT}$	Output high current total for all ports	—	100	mA	
$V_{OL}$	Output low voltage — high drive strength	• $2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $I_{OL} = 10\text{ mA}$	—	0.5	V
		• $1.71\text{ V} \leq V_{DD} \leq 2.7\text{ V}$ , $I_{OL} = 3\text{ mA}$	—	0.5	V
	Output low voltage — low drive strength	• $2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ , $I_{OL} = 2\text{ mA}$	—	0.5	V
		• $1.71\text{ V} \leq V_{DD} \leq 2.7\text{ V}$ , $I_{OL} = 0.6\text{ mA}$	—	0.5	V
$I_{OLT}$	Output low current total for all ports	—	100	mA	
$I_{IN}$	Input leakage current (per pin)	—	1	$\mu\text{A}$	
$I_{OZ}$	Hi-Z (off-state) leakage current (per pin)	—	1	$\mu\text{A}$	
$R_{PU}$	Internal pullup resistors	30	50	$\text{k}\Omega$	
$R_{PD}$	Internal pulldown resistors	30	50	$\text{k}\Omega$	

# PINOS DE INTERFACE I/O

- No Homepage do MBED, vá em “Platforms” e selecione o kit KL25Z:



Esses são os nomes de cada pino e suas funções

# LISTA DE PORTAS E TERMINAIS DO KL25Z

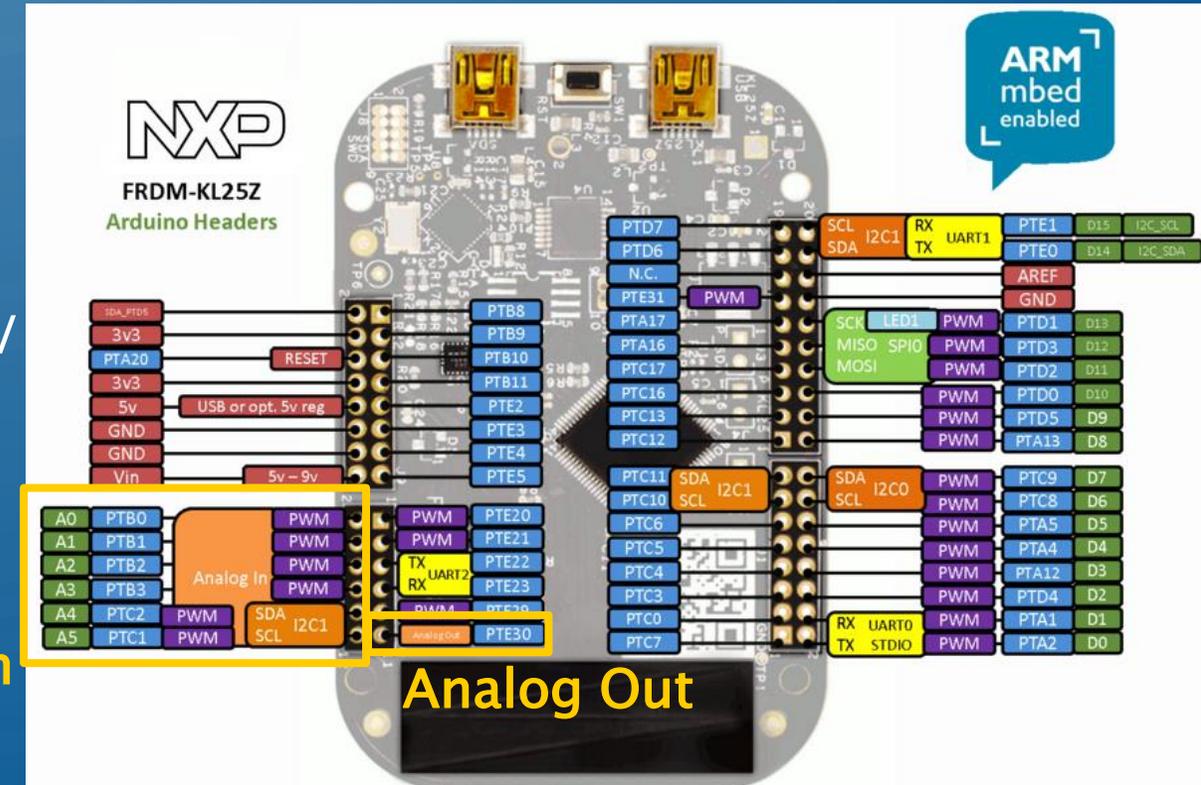
Microcontrolador KL25Z Encapsulamento de 80 pinos		Kit FREEDOM BOARD		
Terminal do chip	Nome e número do port	Disponível no conector e pino do kit	Nome padrão Arduino™ R3	Periférico do kit já conectado ao port
1	PTE0	J2 20	D14	—
2	PTE1	J2 18	D15	—
3	PTE2	J9 09	—	—
4	PTE3	J9 11	—	—
5	PTE4	J9 13	—	—
6	PTE5	J9 15	—	—
13	PTE20	J10 01	—	—
14	PTE21	J10 03	—	—
15	PTE22	J10 05	—	—
16	PTE23	J10 07	—	—
21	PTE29	J10 09	—	—
22	PTE30	J10 11	—	—
23	PTE31	J2 13	—	—
24	PTE24	—	—	Acelerometro
25	PTE25	—	—	Acelerometro
27	PTA1	J1 02	D0	—
28	PTA2	J1 04	D1	—
30	PTA4	J1 10	D4	—
31	PTA5	J1 12	D5	—
32	PTA12	J1 08	D3	—
33	PTA13	J2 02	D8	—
34	PTA14	—	—	Acelerometro
35	PTA15	—	—	Acelerometro
36	PTA16	J2 09	—	—
37	PTA17	J2 11	—	—
42	PTA20	J9 06	—	Botão Reset
43	PTB0	J10 02	A0	—
44	PTB1	J10 04	A1	—
45	PTB2	J10 06	A2	—
46	PTB3	J10 08	A3	—
47	PTB8	J9 01	—	—

Microcontrolador KL25Z Encapsulamento de 80 pinos		Kit FREEDOM BOARD		
Terminal do chip2	Nome e número do port3	Disponível no conector e pino do kit4	Nome padrão Arduino™ R3	Periférico do kit já conectado ao port5
48	PTB9	J9 03	—	—
49	PTB10	J9 05	—	—
50	PTB11	J9 07	—	—
51	PTB16	—	—	Touch Slider
52	PTB17	—	—	Touch Slider
53	PTB18	—	—	Led Vermelho
54	PTB19	—	—	LED Verde
55	PTC0	J1 03	—	—
56	PTC1	J10 12	A5	—
57	PTC2	J10 10	A4	—
58	PTC3	J1 05	—	—
61	PTC4	J1 07	—	—
62	PTC5	J1 09	—	—
63	PTC6	J1 11	—	—
64	PTC7	J1 01	—	—
65	PTC8	J1 14	D6	—
66	PTC9	J1 16	D7	—
67	PTC10	J1 13	—	—
68	PTC11	J1 15	—	—
69	PTC12	J2 01	—	—
70	PTC13	J2 03	—	—
71	PTC16	J2 05	—	—
72	PTC17	J2 07	—	—
73	PTD0	J2 06	D10	—
74	PTD1	J2 12	D13	Led Azul
75	PTD2	J2 08	D11	—
76	PTD3	J2 10	D12	—
77	PTD4	J1 06	D2	—
78	PTD5	J2 04	D9	—
79	PTD6	J2 17	—	—
80	PTD7	J2 19	—	—

# PERIFÉRICOS ANALÓGICOS I/O DO KIT KL25Z

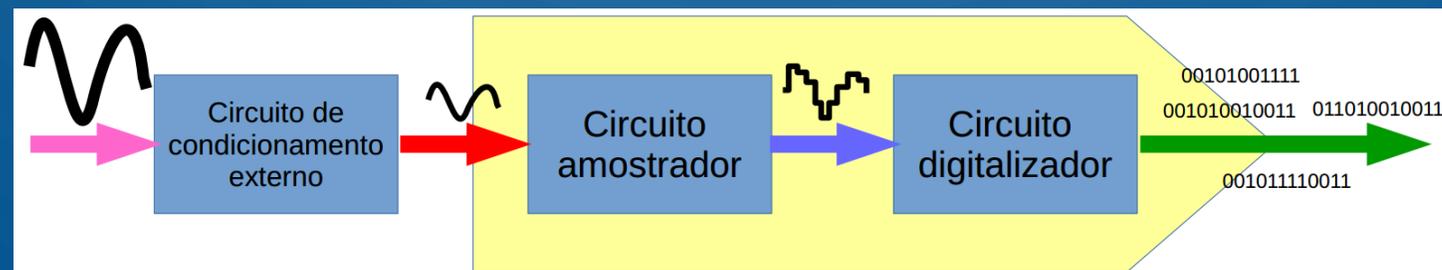
- 6 canais de ADC
  - Limites para conversão da tensão de entrada 0 a 3,3 V
  - Tensão máxima admissível de 3,6 V. Mínima de -0,3 V
  - Taxa de amostragem até 12 MHz
  - Resolução de 16 bits
- 1 canal de um DAC
  - Limites da tensão de saída de 0 a 3,3 V
  - Frequência do sinal – proporcional ao número de pontos em um período.
  - Corrente saída máxima de 1mA
  - Resolução 12 bits

Analog In



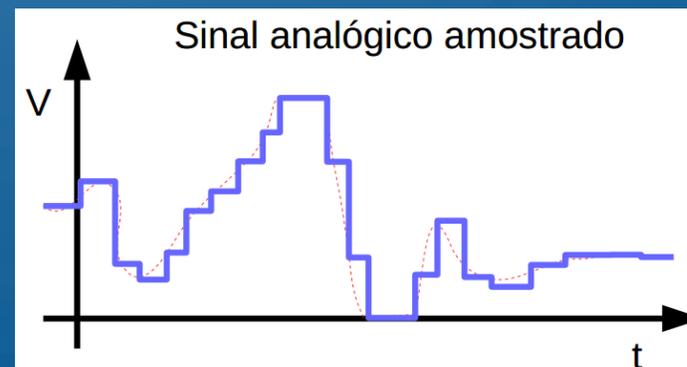
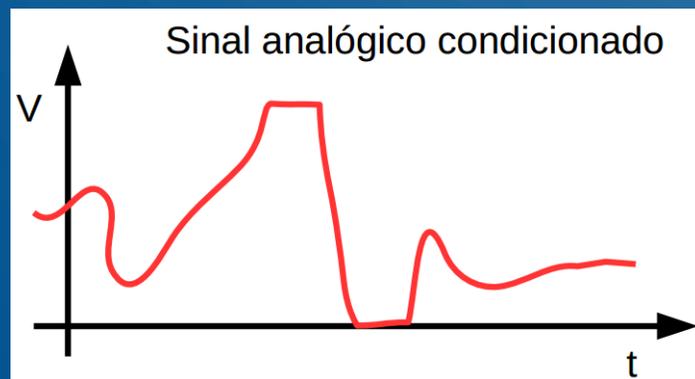
# CONVERSOR ANALÓGICO-DIGITAL (ADC)

- Como um computador/sistema eletrônico pode usar um sinal analógico?
  - Convertendo a grandeza física em uma tensão equivalente, ou fazendo sua transdução, usando um sensor por exemplo.
  - Transformando o valor instantâneo do sinal de tensão, em uma representação numérica simbólica, com uma regra de formação conhecida.
- Esse dispositivo é denominado de **ADC** – Analog to Digital Converter
  - Dispositivo eletrônico capaz de gerar uma representação digital de uma grandeza analógica, tipicamente uma tensão.
  - Empregados na interface entre dispositivos digitais (microcontroladores, etc) e dispositivos analógicos, como sensores de temperatura, pressão, audio, vídeo, etc.



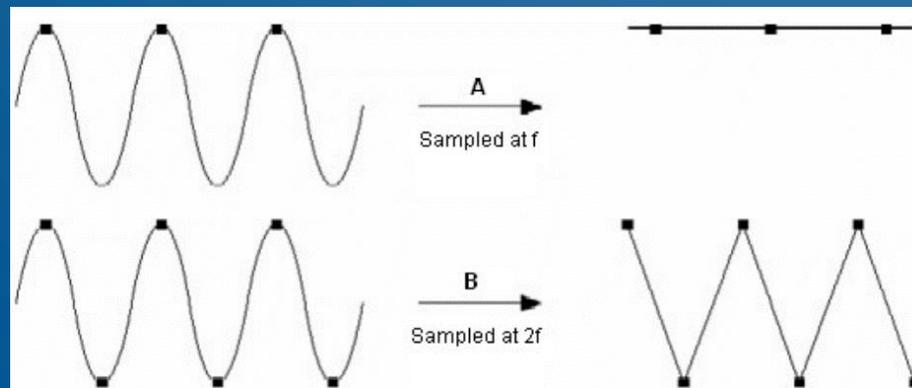
# ADC – AMOSTRAGEM

- Computadores executam operações de forma cíclica, conforme seu *clock* (relógio interno), enquanto que um sinal analógico pode variar continuamente ao longo do tempo. Dessa forma, como deve-se registrar um sinal contínuo?
- Solução: O computador registra o valor dos sinais analógicos periodicamente, conforme uma dada frequência de aquisição, ajustada adequadamente pelo usuário. Um sinal contínuo torna-se uma série, uma sequência de valores.



# ADC – AMOSTRAGEM

- Processo de capturar o valor de um sinal durante um intervalo de tempo.
- Normalmente, os ADCs são usados para capturar amostras em intervalos regulares de tempo.
- A taxa ou frequência de captura (amostras/s) deve ser escolhida corretamente para que seja possível continuar a representar (observar) o sinal original, mesmo após essa amostragem

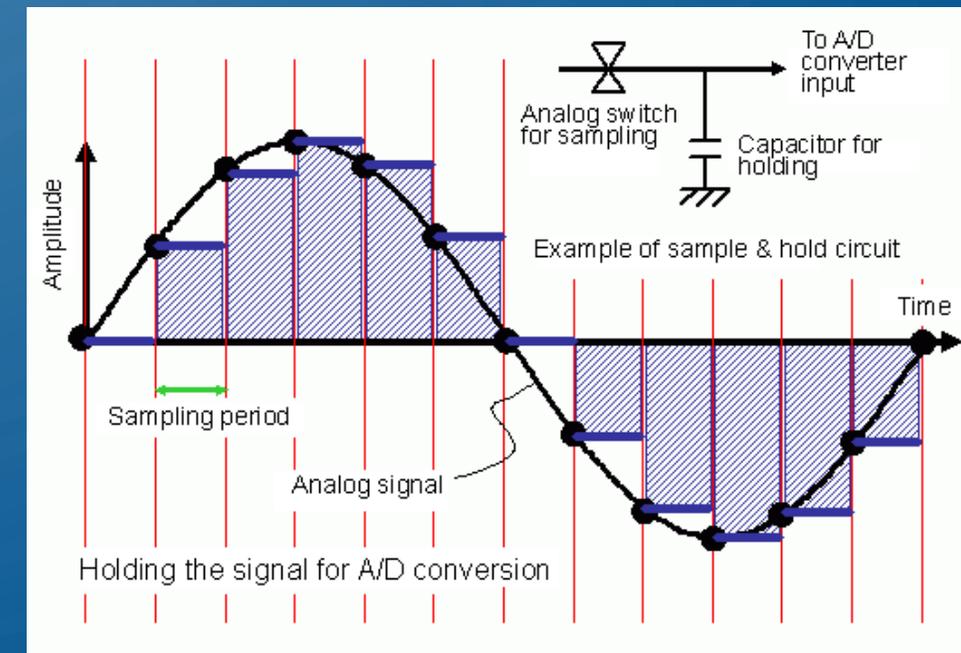


Exemplos de frequência de amostragem insuficiente para representar o sinal original analógico. Os pontos em preto são as amostras do sinal original

# ADC – AMOSTRAGEM / DIGITALIZAÇÃO

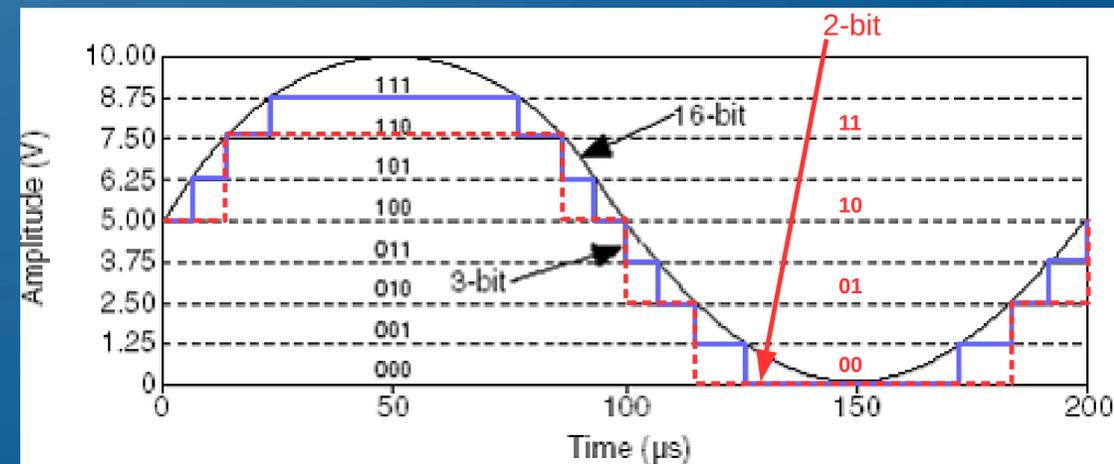
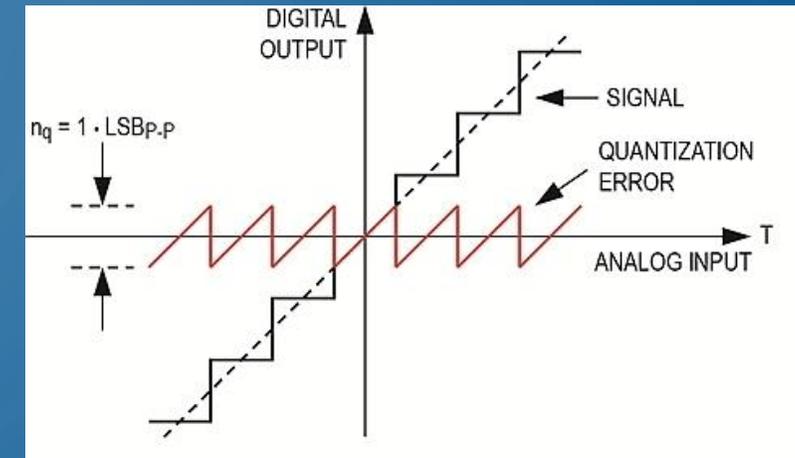
- A amostragem é realizada por um circuito chamado *sample-and-hold* (S/H)
- Ele mantém constante o valor do sinal analógico de entrada enquanto o circuito seguinte de digitalização realiza a quantização do sinal.
- A quantização é feita pelo conversor AD que classifica a tensão amostrada entre os níveis disponíveis. O nº de níveis depende do nº de bits existentes.
- O ADC do microcontrolador tem 16 bits de entrada, logo existem  $2^{16} = 65536$  níveis disponíveis para representar o sinal analógico.

Nível	Representação no seu programa	Tensão real
0	0	0 V
$0 < \text{nível} < 65535$	$\text{nível} * 1 / 65536$	$\text{nível} * 3,3 / 65536$
65535	1	3,3 V



# QUANTIZAÇÃO E EXEMPLO

- Normalmente, um ADC escolhe o valor mais próximo da amostra atual de uma lista uniformemente espaçada.
- Pode haver algum erro de aproximação (erro de quantização no processo)
- **Exemplo de Conversão**
  - Sinal senoidal de entrada:  
Frequência ( $f_s$ ) de 5 kHz, entre 0 e 10 V
  - Taxa de amostragem ( $f_a$ ) de 100 kHz  
(100.000 amostras/s)
  - ➔  $f_a/f_s = 100k/5k = 20$  amostras/período
  - ADC com:
    - 2 bits → Resolução de 2,5 V
    - 3 bits → Resolução  $(10-0)/(2^3) = 1,25$  V
    - 16 bits → Resolução de 0,000152 V



# EXEMPLO DE CÓDIGO – AMOSTRAGEM

```
#include "mbed.h"
```

```
AnalogIn ain(PTXX); //ain recebe o valor da porta PTXX
```

```
int main()
```

```
{
```

```
    printf("%f\r", ain.read());
```

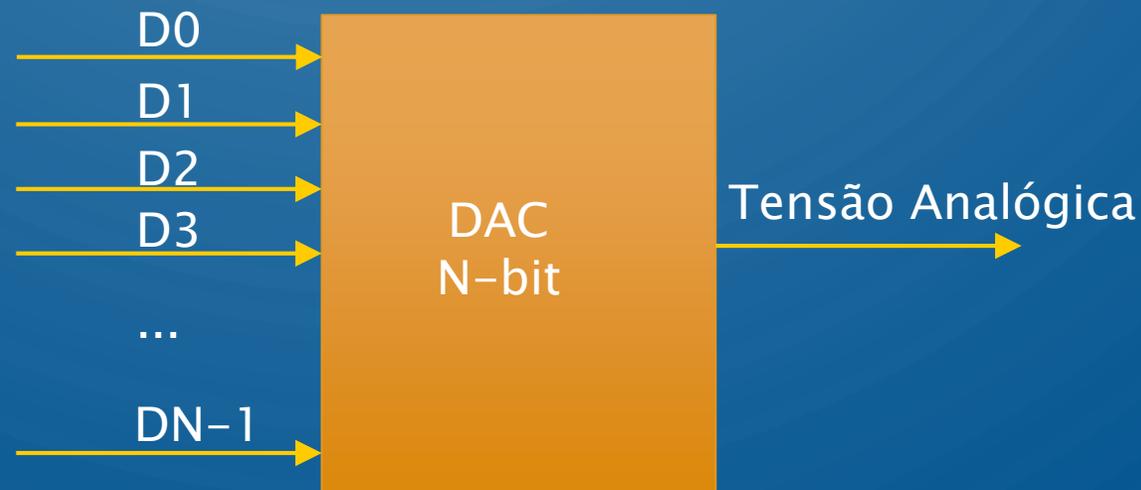
```
    // Coloque aqui o seu programa para tratamento dos dados.
```

```
}
```

# CONVERSOR DIGITAL ANALÓGICO (DAC)

- Circuito eletrônico capaz de converter uma grandeza digital em uma grandeza analógica.
- Esse processo é denominado de síntese digital–analógica.
- Características Básicas:
  - Velocidade de conversão
  - Resolução ou quantidade de níveis para síntese do sinal analógico (nº de bits)

O DAC do microcontrolador tem 12 bits



# EXEMPLO DE CÓDIGO – AMOSTRAGEM

```
#include "mbed.h"
```

```
AnalogOut saida(XXXXXX);
```

```
//saida é a variável relacionada à porta XXXXXX
```

```
int main()
```

```
{
```

```
    while(1){
```

```
        saida=saida+0.01;
```

```
        wait_us(1);
```

```
        if (saida==1)
```

```
            saída=0;
```

```
    }
```

```
}
```