

Introdução a Engenharia Elétrica - 323100

Aula S8

Módulo 1 – Conversores AD e DA

Escola Politécnica da Universidade de São Paulo

Departamentos da Engenharia Elétrica



PCS Computação e Sistemas Digitais

PEA Energia e Automação Elétricas

PSI Sistemas Eletrônicos

PTC Telecomunicações e Controle

V1.1

Setembro de 2018



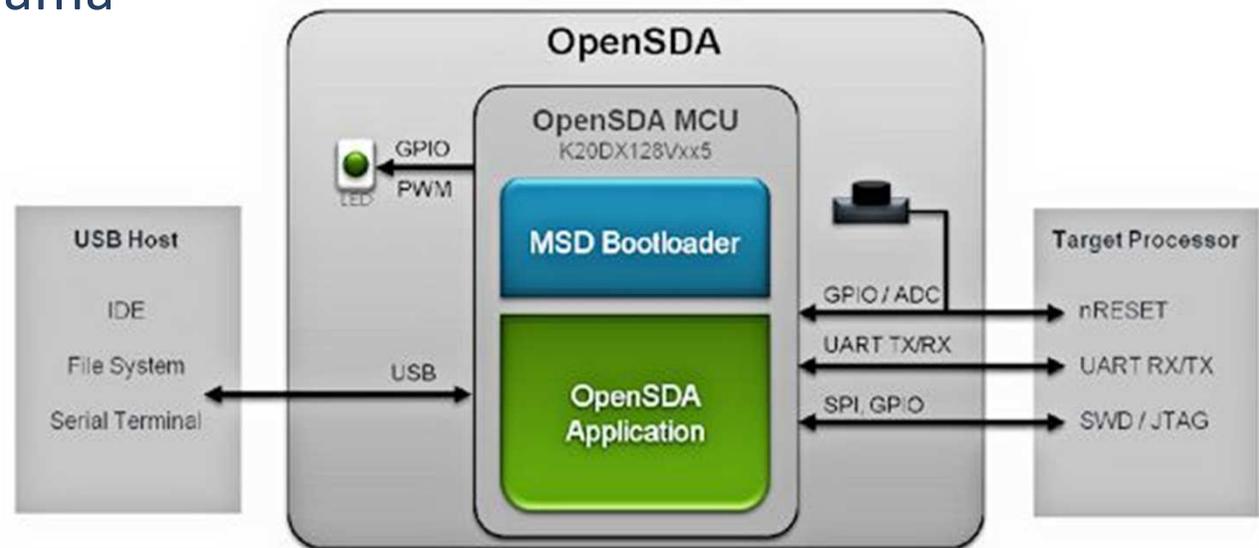
Sumário

1. Porta serial de comunicação com o Kit
2. Sinais analógicos e digitais
3. Conversores Analógico - Digitais
4. Conversor Digital - Analógico



Porta serial virtual via USB

- Atualmente: sistema de arquivos
 - Para armazenar os executáveis produzidos pelos programadores.
- Nova função: serial virtual através da USB
 - Para comunicação com o Kit
 - Para debugar o programa





Como se preparar para o uso...

- Instalar o driver da porta serial virtual

No windows, uso o arquivo **dpinst_x86.exe** ou **dpinst_amd64.exe** disponível no arquivo 'Drivers en.stsw-link009.zip'

Mais detalhes, vide apresentação específica no eDisciplinas.

- Instalar um programa emulador de terminal

No windows, uso o arquivo **teraterm-4.84.exe**

Mais detalhes, vide apresentação específica no eDisciplinas.

Obs.: Há outros programas emuladores de terminal, tais como o Hyperterminal, Putty, Minicom, etc.



Esses programas já devem estar instalados na sala de aula.



Como utilizar no seu programa

- Em geral, utilize chamadas para a rotina printf da forma tradicional do ANSI C.
- **printf(“Ola mundo!!!\r\n”);**
- O kit STM32 envia os caracteres através da porta serial virtual até o seu computador.
- Seu computador devem estar executando o programa emulador de terminal Teraterm (ou similar), com a serial USB virtual selecionada como porta de comunicação.
- As definições padrão da comunicação são: velocidade de 9600 bps, 8 bits de dados, paridade nenhuma, 1 bit de parada.



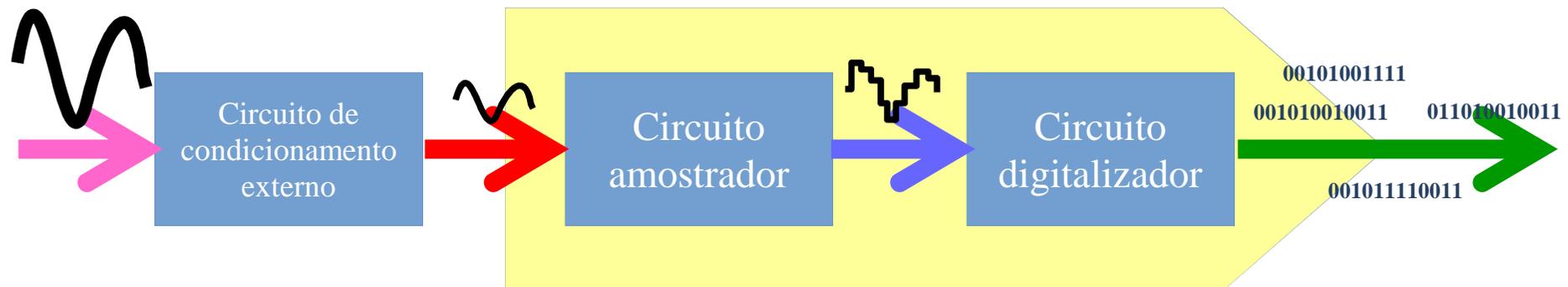
Conversão analógico - digital

- Sinais analógicos – sinais análogos às grandezas físicas.
 - São contínuos no tempo.
 - Podem ser medidos ou aferidos em uma ampla gama de valores e unidades.
- Como um computador/sistema eletrônico pode usar esses sinais?
 1. Convertendo a grandeza física em uma tensão equivalente, ou fazendo sua transdução, usando um sensor por exemplo.
 2. 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).



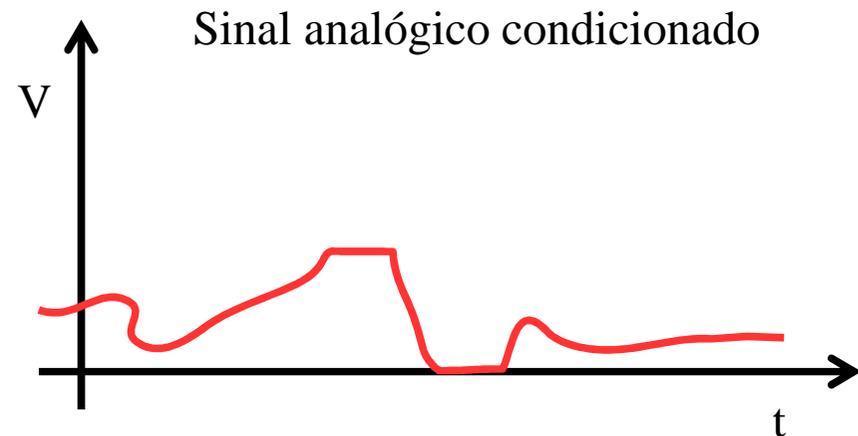
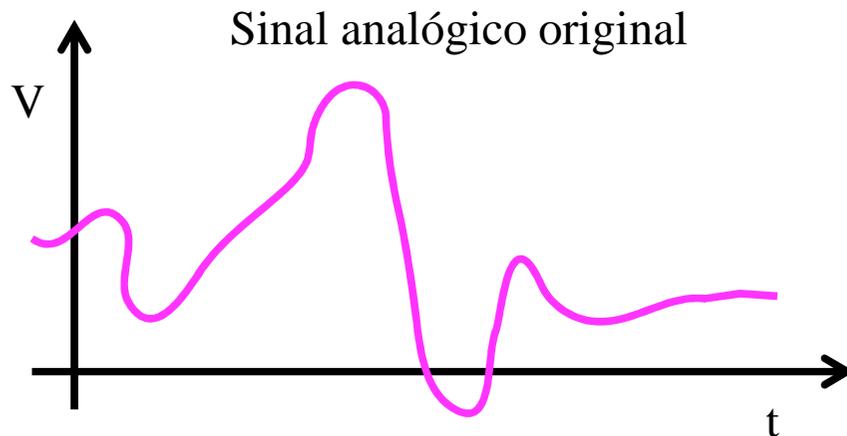
Conversor Analógico Digital (ADC)

- 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.





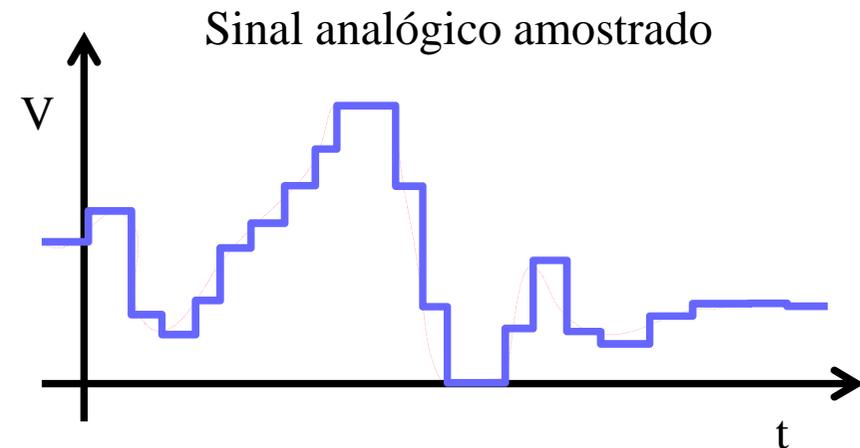
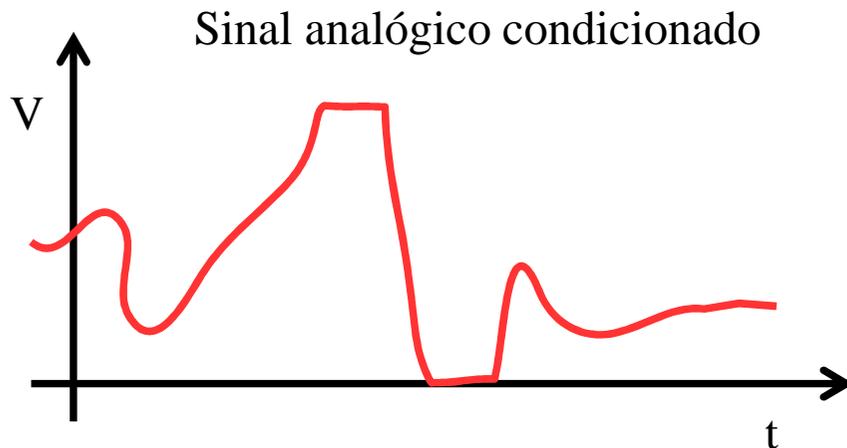
Conversão analógico/digital - condicionamento



- Problema: Os sinais analógicos podem assumir valores dos mais diversos em magnitude. Os computadores só trabalham com tensões próximas de suas tensões de alimentação. Como compatibilizar esses níveis de tensão?
- Solução: O computador utiliza um circuito de condicionamento para tornar as medidas condizentes com os limites do dispositivo. As magnitudes são reescaladas e limitadas, de forma a proteger os demais circuitos do ADC do microcontrolador



Conversão analógico/digital - amostragem

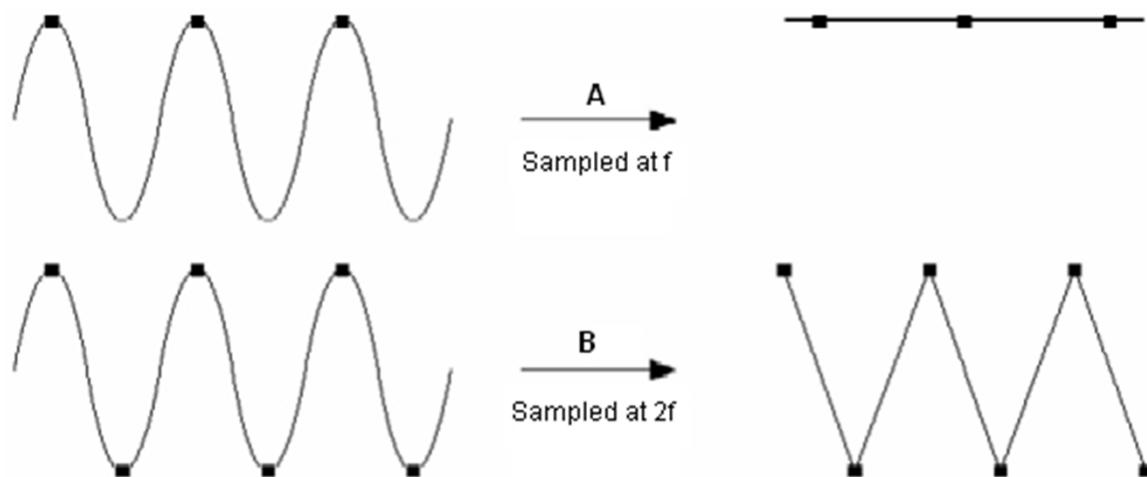


- Problema: Computadores executam operações de forma cíclica, conforme seu *clock* ou relógio interno, enquanto que um sinal analógico pode variar continuamente ao longo do tempo. Dessa forma, como deve-se registrar um sinal contínuo?
- Solução: O computador registra o valor dos sinais analógicos periodicamente,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



Amostragem no conversor analógico digital

- 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.

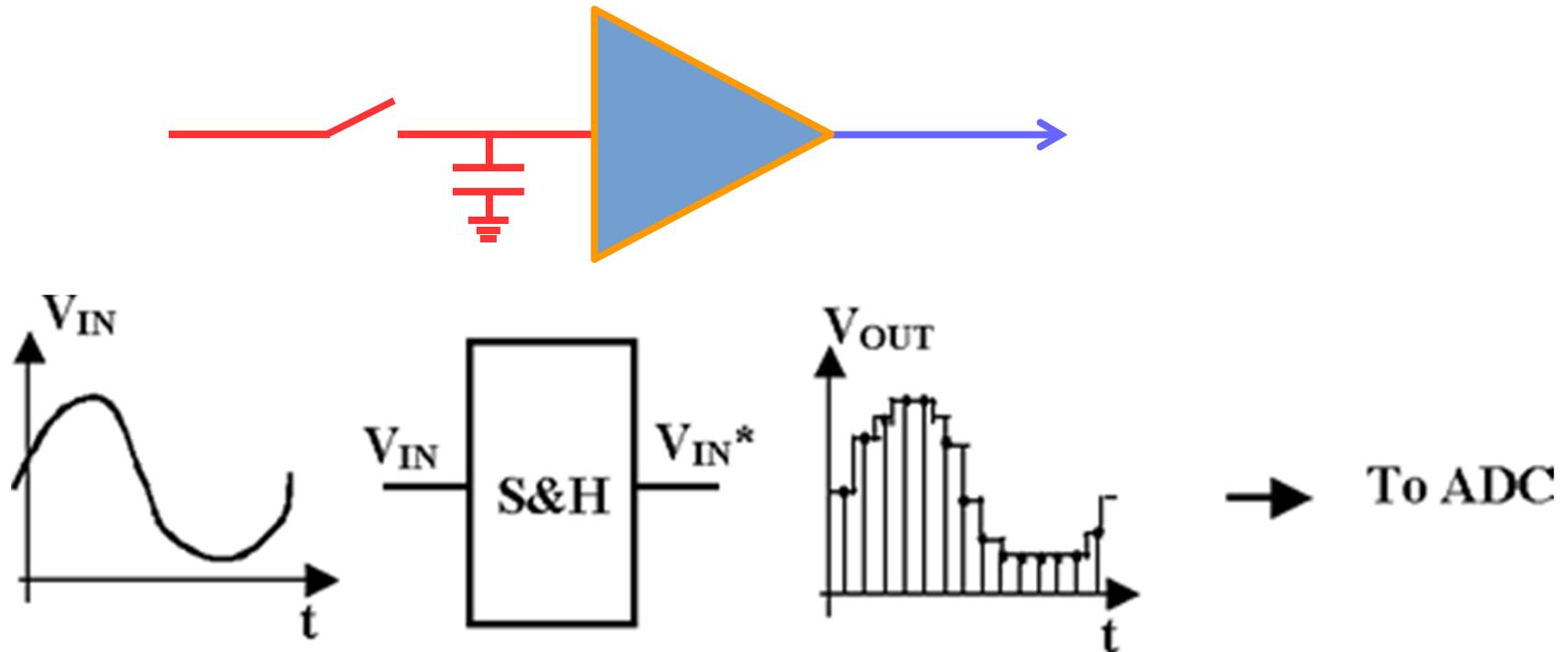


Exemplos de frequência de amostragem insuficiente para representar o sinal original analógico. Os pontos em preto são as amostras do sinal original.



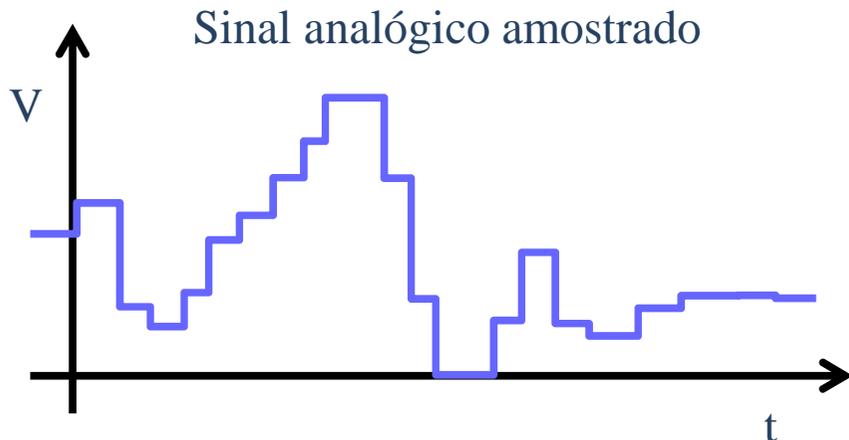
Funcionamento da amostragem

- 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.





Conversão analógico/digital - digitalização



Série de valores digitalizados

100, 130, 50, 30, 55, 89, 105,
145, 160, 255, 145, 56, 0, 0,
32, 78, 29, 25, 42, 48, 49, 51,
50, ...

- O computador pode classificar a tensão amostrada entre os níveis ou valores disponíveis em uma tabela.
- Os sinais que chamamos de digitais, utilizados até então no microcontrolador, foram digitalizados usando apenas dois níveis, 0 ou 1. Um conversor AD possui uma quantidade (resolução) muito maior de símbolos, valores ou patamares para representar esses sinal.
- Quanto melhor a resolução, melhor será o discernimento ou discretização dos valores assumidos pela tensão externa.



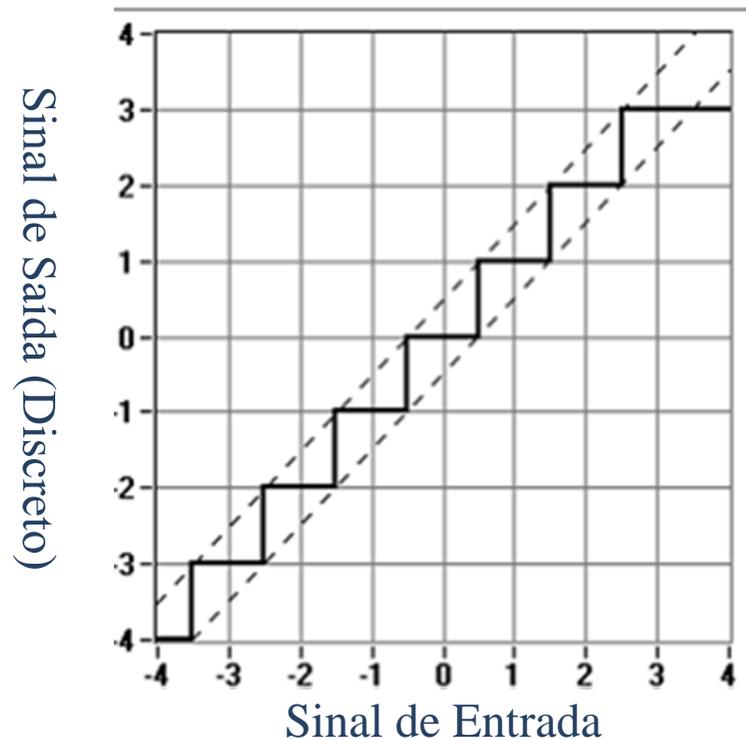
Funcionamento da digitalização

- Um ADC digitaliza um sinal amostrado utilizando um valor inteiro aproximado (quantizado), para representar cada amostra analógica, constituindo uma série de valores discretos (quantização).
- O valor escolhido é proveniente de uma lista pré-determinada.
 - O tamanho da lista é definida pelo número de bits do ADC (resolução).
 - Um conversor de N bits possui 2^N valores para representar o sinal amostrado.
- Exemplo: supondo um conversor de 8 bits
 - $2^8 = 256$ valores distintos, por exemplo, de 0 a 255
 - Para uma tensão máxima na entrada de 3,3V (referência) → utiliza-se o valor máximo da tabela, ou seja, 255.
 - Para uma tensão mínima na entrada de 0,0 V → utiliza-se o menor valor da tabela, 0.
 - Todos os valores intermediários estão separados por níveis equivalentes a $3,3/256 = 0,012$ [V/patamar]



Conversor Analógico Digital (ADC)

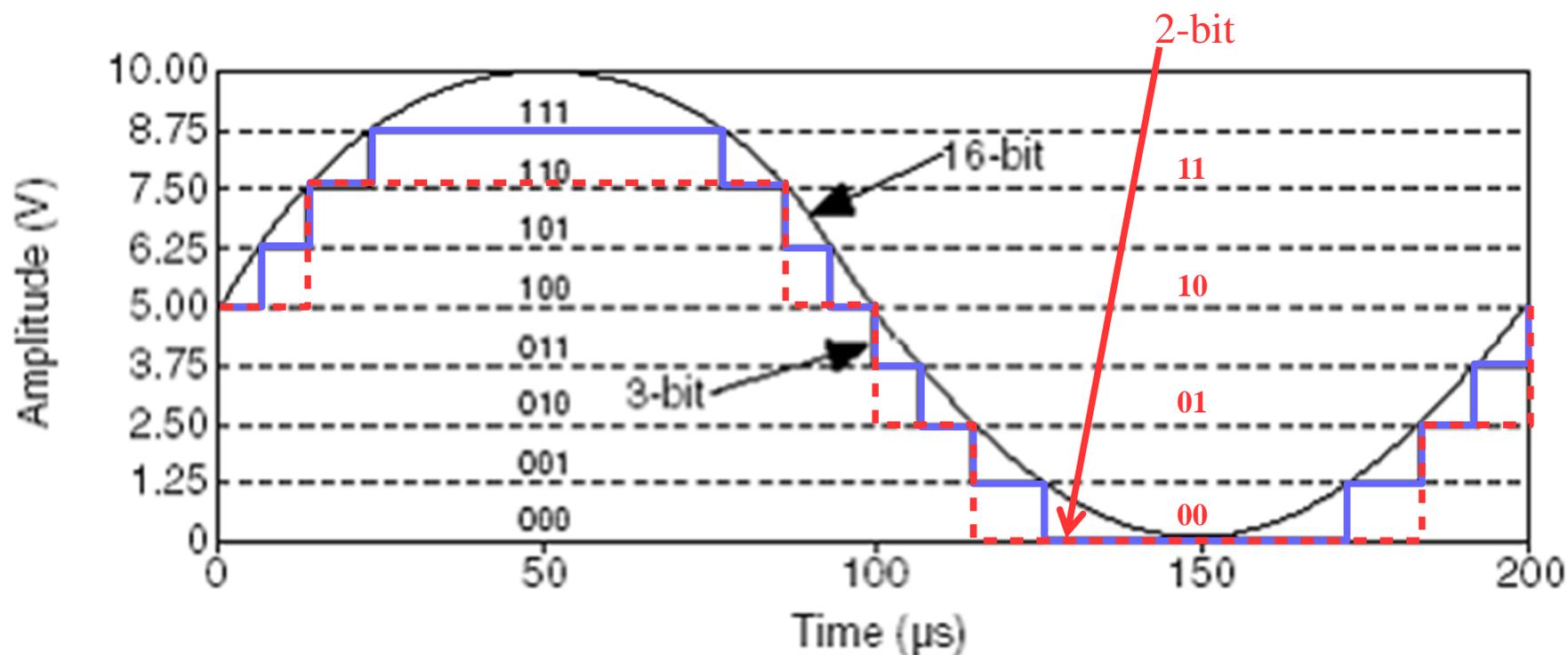
- Quantização
 - Normalmente, um ADC escolhe o valor mais próximo da amostra atual de uma lista uniformemente espaçada.
 - Pode haver algum erro de aproximação (erro de quantização no processo).





Conversor Digital Analógico (ADC)

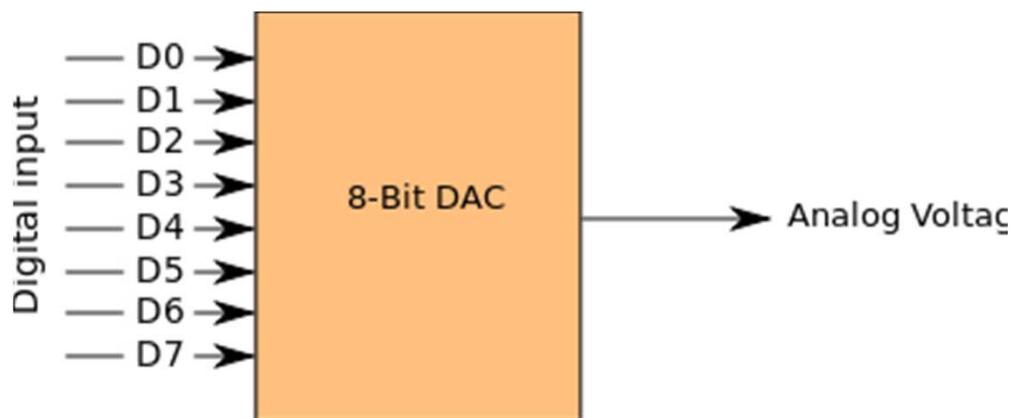
- Exemplo de Conversão
 - Sinal Senoidal de entrada de 5kHz, oscilando entre 0 e 10 [V].
 - Taxa de Amostragem de 100kHz (100.000 amostras/s).
 - ADC de 2 Bits → Resolução de 2,5 [V]
 - ADC de 3 Bits → Resolução $10/(2^3)=1,25$ [V]
 - ADC de 16 bits → Resolução de 0,000152 [V]





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 patamares para síntese do sinal analógico (número de bits).

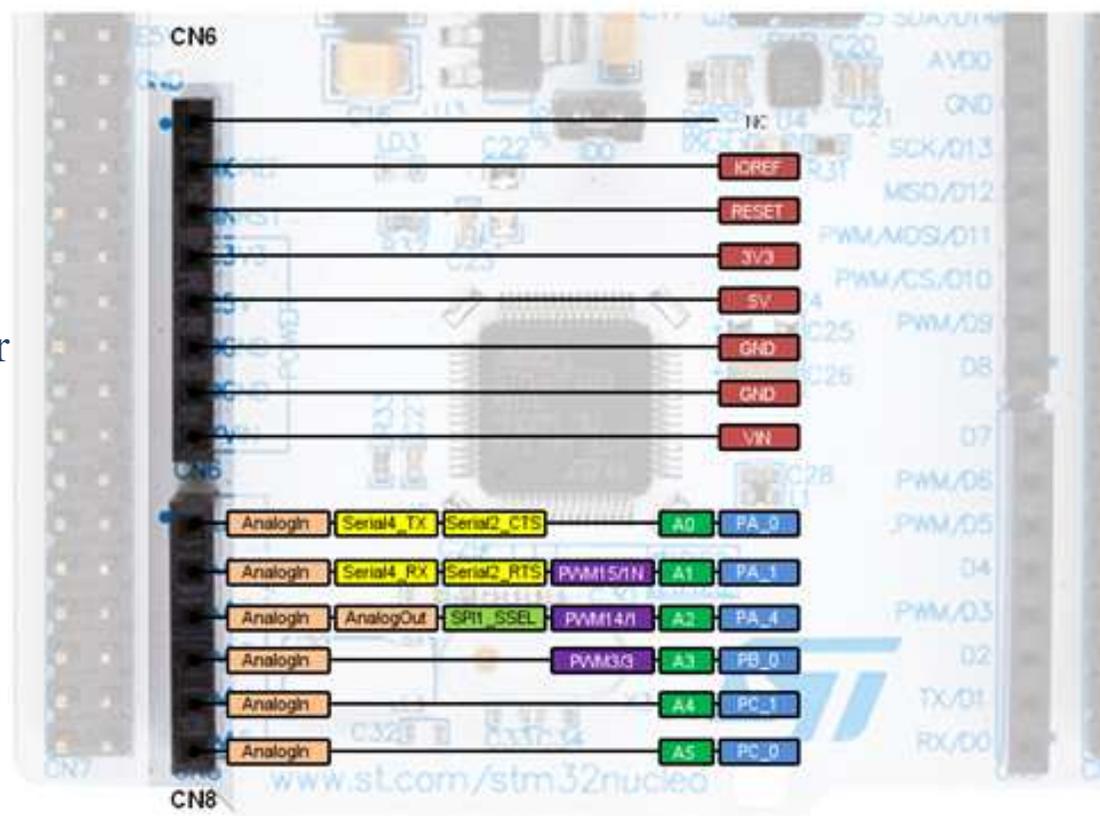




Periféricos analógicos de entrada do Kit STM32

- 16 canais de ADC (AnalogIn)
 - 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 14 [MHz].
 - Resolução de 12 bits.

As demais portas podem ser encontradas no apêndice

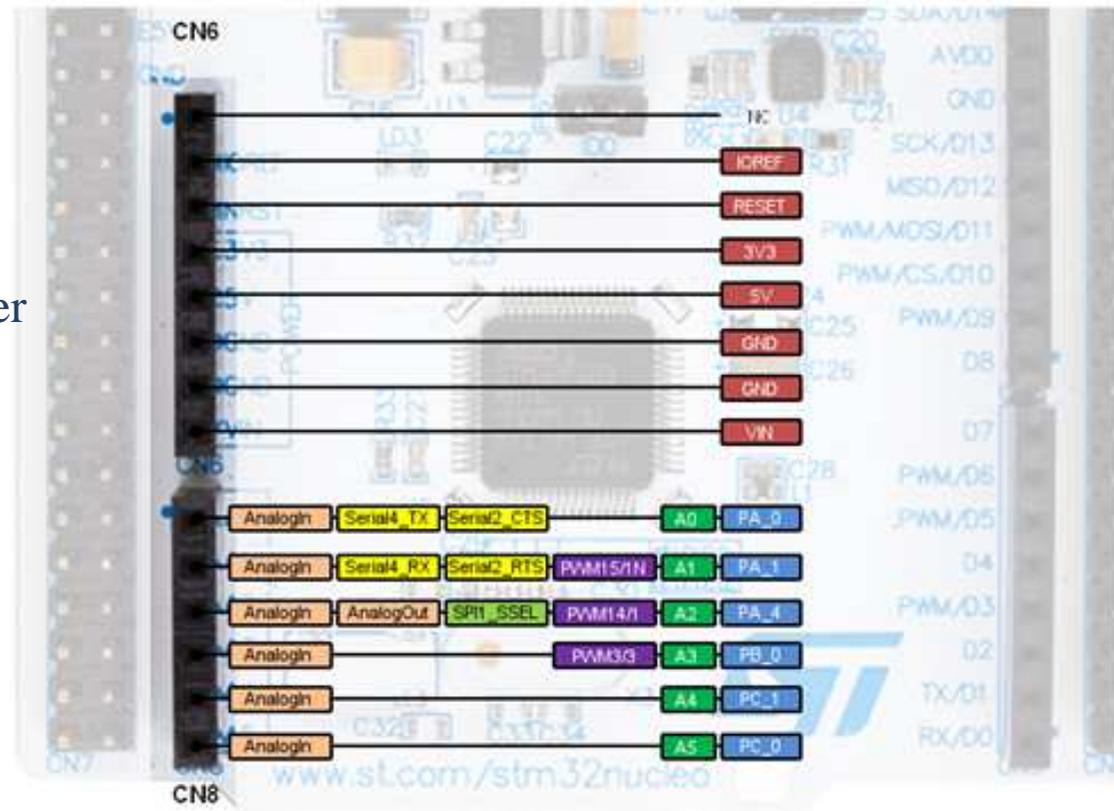




Periféricos analógicos de saída do Kit STM32

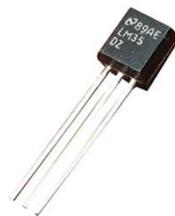
- 2 canais de um DAC (AnalogOut)
 - Limites da tensão de saída de 0,0 a 3,3 [V]
 - Taxa de amostragem 5 a 30kHz
 - Corrente saída máxima de 1mA
 - Resolução 12-bits

As demais portas podem ser encontradas no apêndice

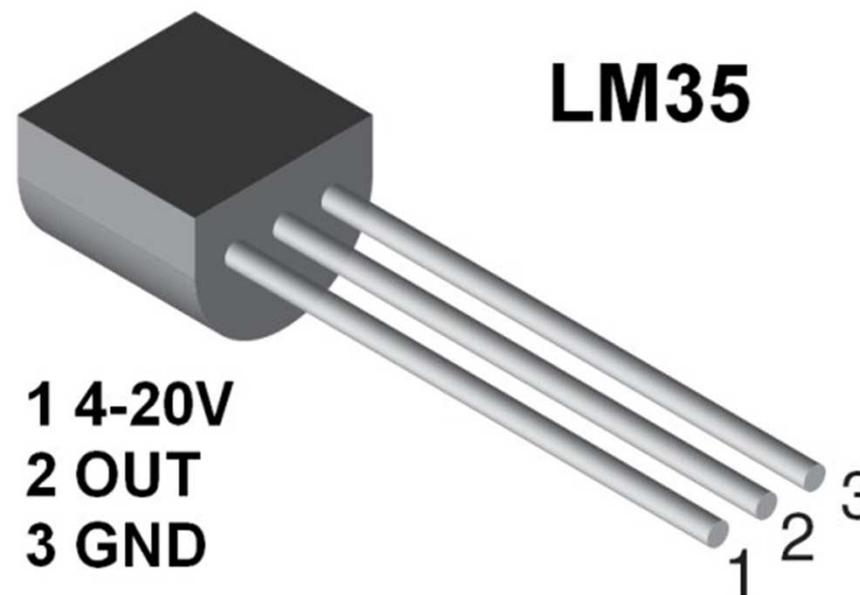




Sensor LM-35



- Transdutor Temperatura-Tensão
- Tensão de saída x Temperatura
 - Relação linear
 - +10 mV / °C
- Faixa de Operação
 - -55°C até 150°C
- Pinagem
 - Pino 1 – Tensão de alimentação entre 4 a 20,0 [V]
 - Pino 2 – Saída do sinal analógico de tensão
 - Pino 3 – Terminal de terra ou negativo.



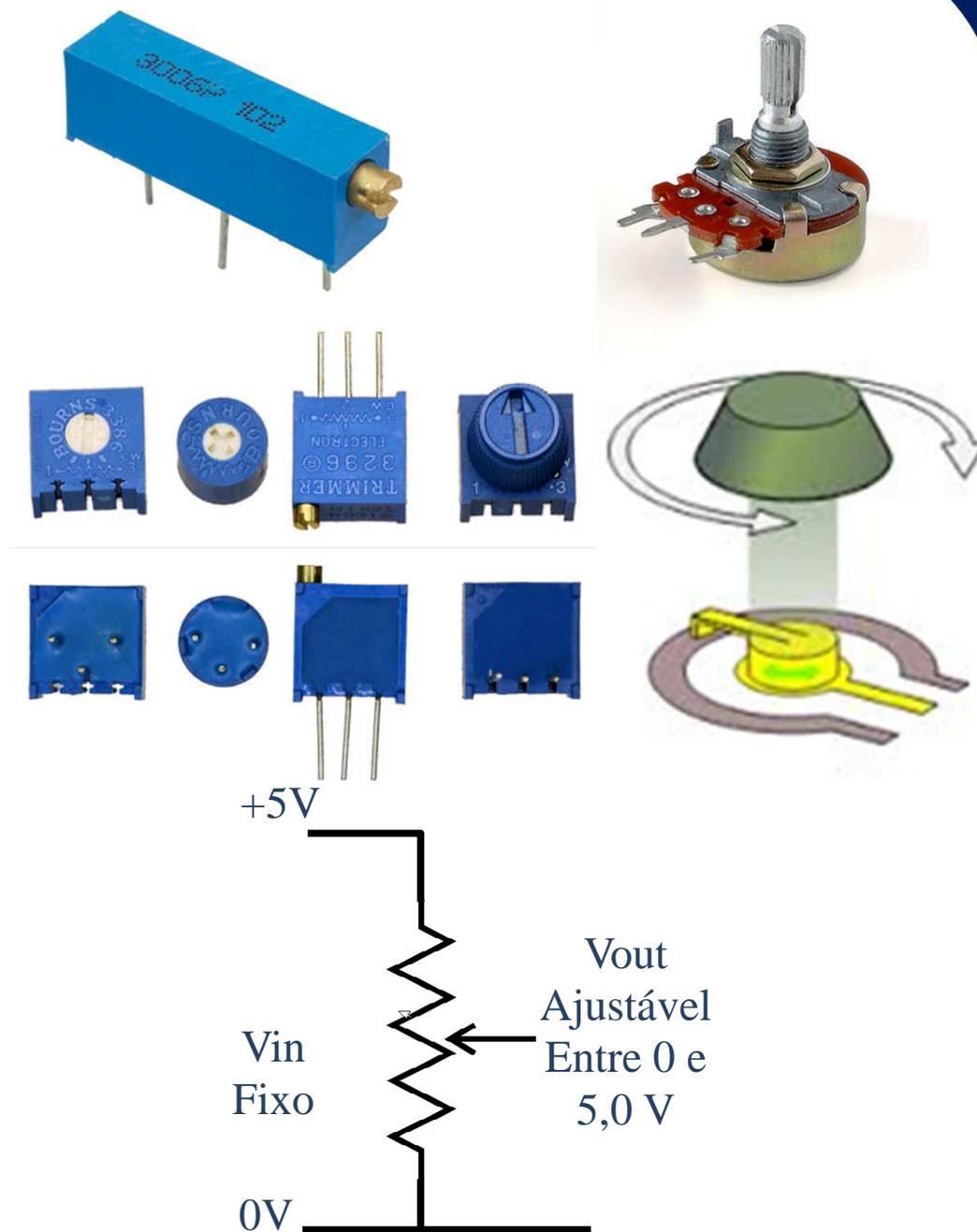
Atenção: O LM-35 não pode ser ligado invertido, com as tensões erradas. O dispositivo pode queimar!!!!





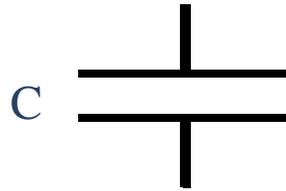
Potenciômetro

- Resistor variável.
- Possui um valor nominal de resistência e potência máximas.
- Possui três terminais.
- Permite implementar um divisor de tensão, para produzir uma tensão entre um valor máximo e mínimo.





Capacitores



- Duas placas metálicas paralelas, isoladas por um material dielétrico (isolante).
- Várias tecnologias de materiais isolantes (cerâmicos, eletrolíticos, etc).
- Armazenam energia na forma do acúmulo de cargas elétricas entre suas placas.
- Utilizados tipicamente como acumuladores, ou em circuitos de filtragem.
- Alguns tipos são polarizados. Atente à forma de ligação do componente ao circuito.
- A relação entre a tensão e a corrente em um capacitor obedece a expressão:

$$v(t) = \frac{1}{C} \cdot \int_{t_0}^t i(t) \cdot dt + v(t_0)$$



Poliéster



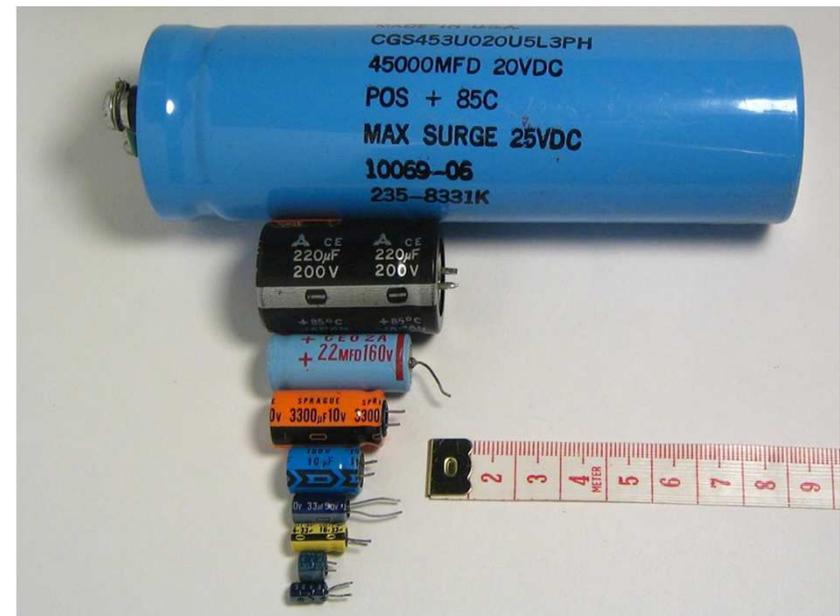
Cerâmico



Eletrolítico Polarizado

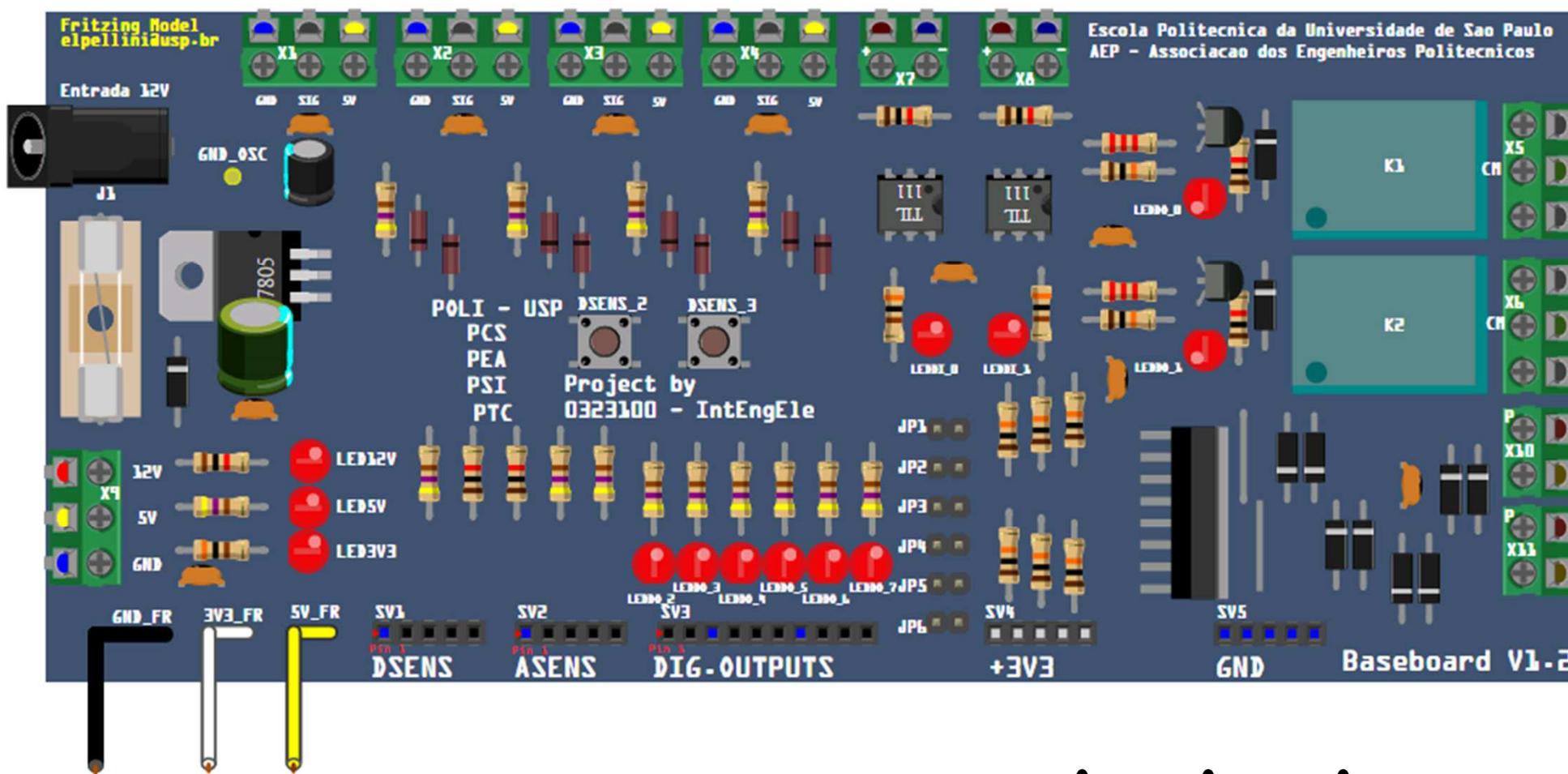


Tântalo - polarizado





Baseboard da disciplina



Atenção à documentação e manual de uso.



- A placa só está pronta para uso se as três luzes LED12V, LED5V e LED3V3 estiverem ligadas!!!



Exercício 1

Exercício SerialComm

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

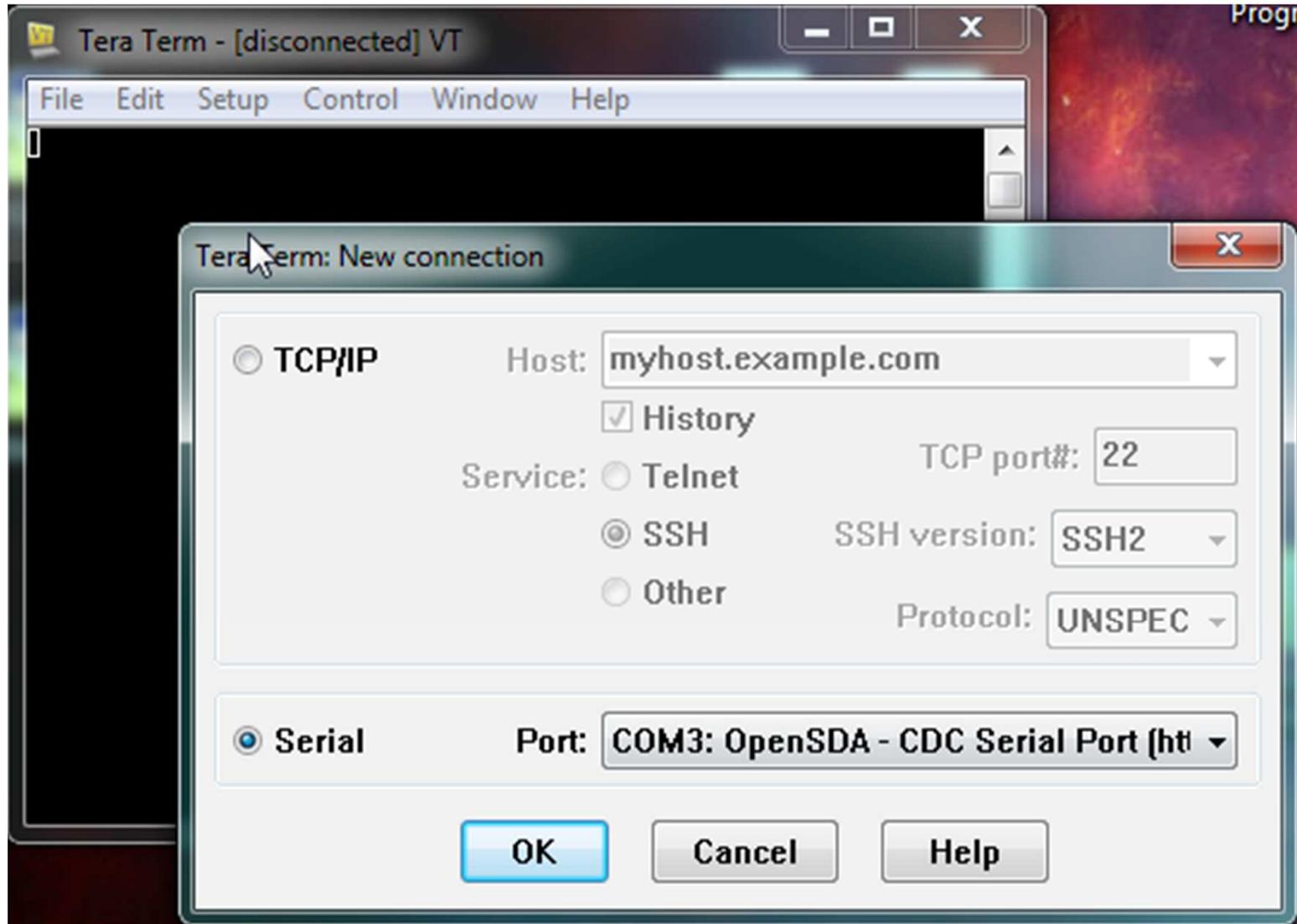
```
Serial PC(USBTX, USBRX);
```

```
int main(){  
    int Contador;  
    Contador = 0;  
    while(1)  
    {  
        PC.printf("Hello Word #%d\r\n", Contador);  
        Contador++;  
        wait(0.5); //500ms  
    }  
}
```



Exercício 1

Tela do teraterm para abertura da porta serial do kit





Exercício 1

Exercício SerialComm – Saída esperada

COM3:9600baud - Tera Term VT

File Edit Setup Control Window Help

```
Hello Word #158  
Hello Word #159  
Hello Word #160  
Hello Word #161  
Hello Word #162  
Hello Word #163  
Hello Word #164  
Hello Word #165  
Hello Word #166  
Hello Word #167  
Hello Word #168  
Hello Word #169  
Hello Word #170  
Hello Word #171  
Hello Word #172  
Hello Word #173  
Hello Word #174  
Hello Word #175  
Hello Word #176  
Hello Word #177  
Hello Word #178  
Hello Word #179  
Hello Word #180
```


Exercício ADC_Pot

```
#include "mbed.h"

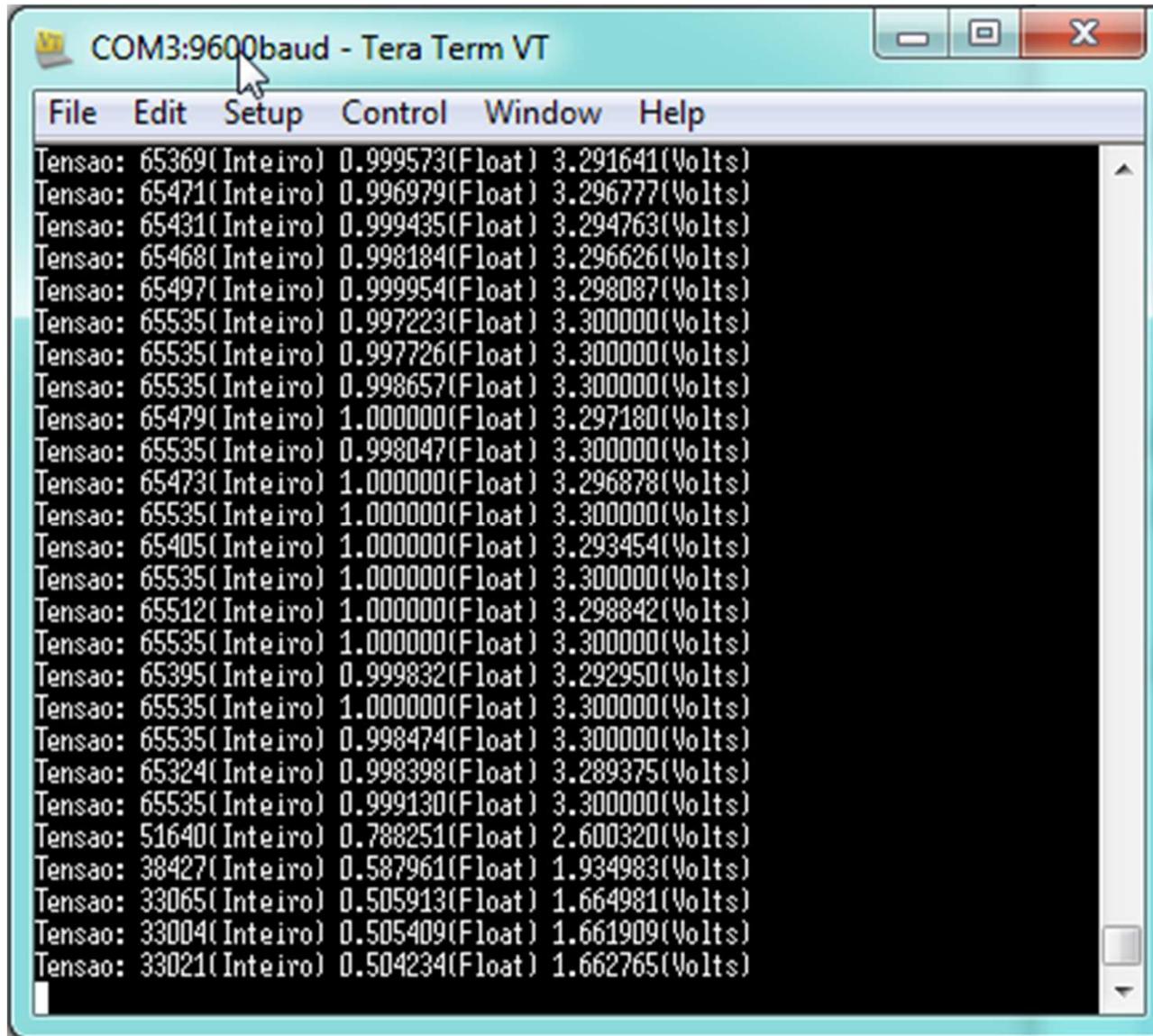
#define MV(x) ((3.3*x)/0xFFFF)

AnalogIn Entrada_AN(PA_0);
Serial PC(USBTX, USBRX);

int main() {
    unsigned short Medida_Inteira;
    float Valor_Normalizado;
    while(1) {
        Valor_Normalizado = Entrada_AN;
        Medida_Inteira = Entrada_AN.read_u16();
        PC.printf("Tensao: %d(Inteiro) %f(Float) %f(Volts)\r\n",
                Medida_Inteira, Valor_Normalizado,
                MV(Medida_Inteira));
        wait(0.5); // 500 ms
    }
}
```

Exercício 2 - Montagem ADC_Pot

Exercício ADC_Pot – Saída esperada



The screenshot shows a terminal window titled "COM3:9600baud - Tera Term VT". The window displays a list of ADC readings. Each line represents a reading with three columns: a decimal value in parentheses labeled "Inteiro", a floating-point value in parentheses labeled "Float", and a voltage value in parentheses labeled "Volts". The data shows a range of values, with the voltage values fluctuating between approximately 1.66V and 3.30V.

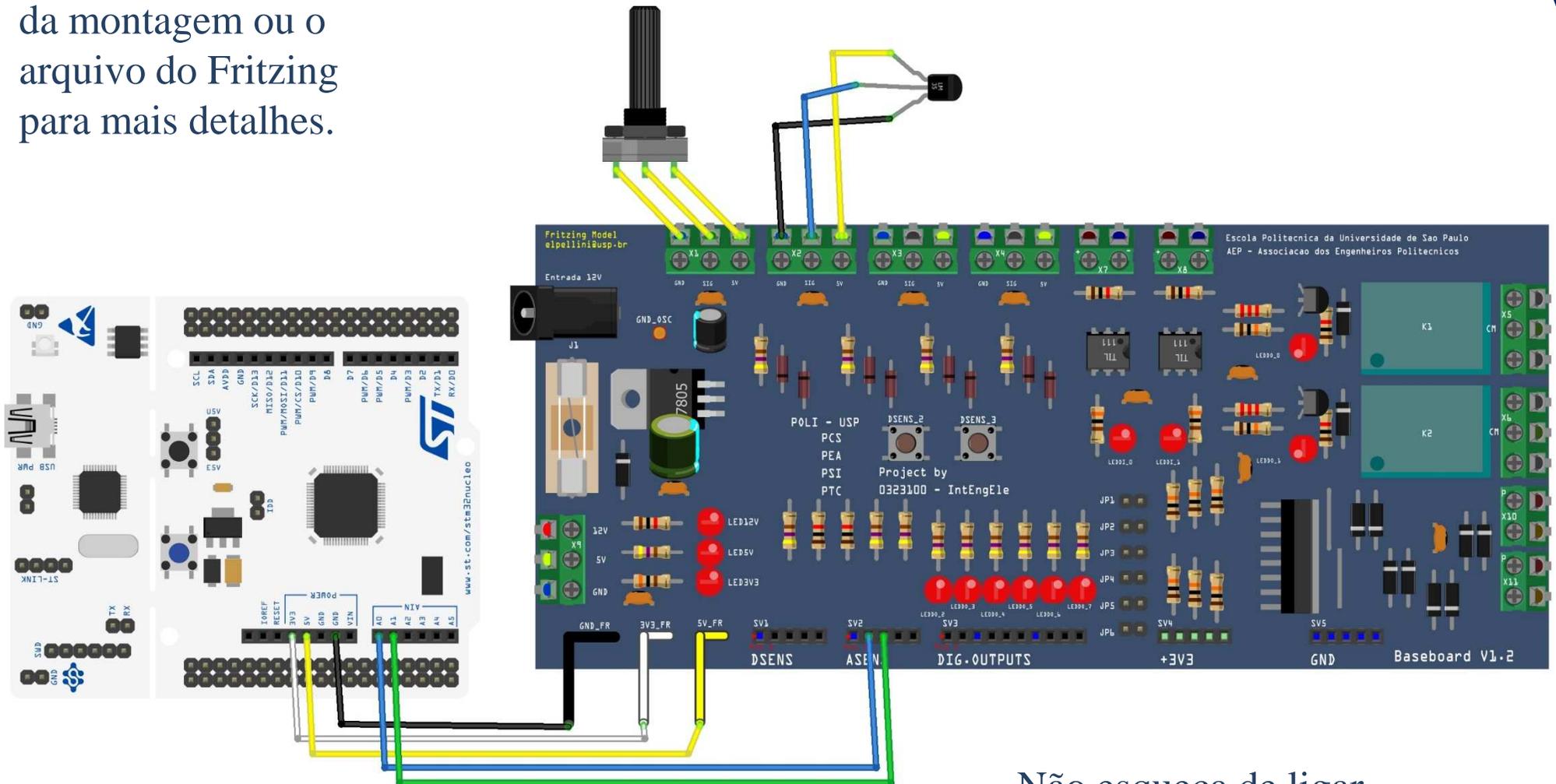
Tensao	Inteiro	Float	Volts
Tensao:	65369	0.999573	3.291641
Tensao:	65471	0.996979	3.296777
Tensao:	65431	0.999435	3.294763
Tensao:	65468	0.998184	3.296626
Tensao:	65497	0.999954	3.298087
Tensao:	65535	0.997223	3.300000
Tensao:	65535	0.997726	3.300000
Tensao:	65535	0.998657	3.300000
Tensao:	65479	1.000000	3.297180
Tensao:	65535	0.998047	3.300000
Tensao:	65473	1.000000	3.296878
Tensao:	65535	1.000000	3.300000
Tensao:	65405	1.000000	3.293454
Tensao:	65535	1.000000	3.300000
Tensao:	65512	1.000000	3.298842
Tensao:	65535	1.000000	3.300000
Tensao:	65395	0.999832	3.292950
Tensao:	65535	1.000000	3.300000
Tensao:	65535	0.998474	3.300000
Tensao:	65324	0.998398	3.289375
Tensao:	65535	0.999130	3.300000
Tensao:	51640	0.788251	2.600320
Tensao:	38427	0.587961	1.934983
Tensao:	33065	0.505913	1.664981
Tensao:	33004	0.505409	1.661909
Tensao:	33021	0.504234	1.662765

Varie o Pot!!



Exercício 3 - Montagem ADC_Pot_Temp

Veja o arquivo PDF da montagem ou o arquivo do Fritzing para mais detalhes.



Não esqueça de ligar a fonte de 12,0 V que alimenta a baseboard.

fritzing



Exercício 3 - Montagem ADC_Pot_Temp

Exercício ADC_Pot_Temp

```
#include "mbed.h"

AnalogIn Entrada_AN_Potenciometro(PA_0);
AnalogIn Entrada_AN_Sensor(PA_1);

Serial PC(USBTX, USBRX);

int main()
{
    float Medida_Potenciometro, Medida_Sensor, Temperatura;
    while(1) {
        Medida_Potenciometro=Entrada_AN_Potenciometro*3.3;
        Medida_Sensor=Entrada_AN_Sensor*3.3;
        Temperatura=Medida_Sensor/10E-3; //10mV/C
        PC.printf("Pot.: %3.4f [V] ", Medida_Potenciometro);
        PC.printf("LM35: %3.4f [V] ", Medida_Sensor);
        PC.printf("LM35: %3.1f [oC] \r\n", Temperatura);
        wait(0.5); // 500 ms
    }
}
```



Exercício 3 - Montagem ADC_Pot_Temp

Exercício ADC_Pot_Temp – Saída esperada

```
COM3:9600bau - Tera Term VT
File Edit Setup Control Window Help
Pot.: 1.4967 [V] LM35: 0.2600 [V] LM35: 26.0 [oC]
Pot.: 1.4416 [V] LM35: 0.2588 [V] LM35: 25.9 [oC]
Pot.: 1.4917 [V] LM35: 0.2538 [V] LM35: 25.4 [oC]
Pot.: 1.4883 [V] LM35: 0.2554 [V] LM35: 25.5 [oC]
Pot.: 1.4812 [V] LM35: 0.2485 [V] LM35: 24.9 [oC]
Pot.: 1.4486 [V] LM35: 0.2503 [V] LM35: 25.0 [oC]
Pot.: 1.4553 [V] LM35: 0.2533 [V] LM35: 25.3 [oC]
Pot.: 1.4520 [V] LM35: 0.2552 [V] LM35: 25.5 [oC]
Pot.: 1.4946 [V] LM35: 0.2583 [V] LM35: 25.8 [oC]
Pot.: 1.4503 [V] LM35: 0.2441 [V] LM35: 24.4 [oC]
Pot.: 1.4958 [V] LM35: 0.2561 [V] LM35: 25.6 [oC]
Pot.: 1.4500 [V] LM35: 0.2464 [V] LM35: 24.6 [oC]
Pot.: 1.4521 [V] LM35: 0.2475 [V] LM35: 24.7 [oC]
Pot.: 1.4786 [V] LM35: 0.2550 [V] LM35: 25.5 [oC]
Pot.: 1.5024 [V] LM35: 0.2521 [V] LM35: 25.2 [oC]
Pot.: 1.4967 [V] LM35: 0.2511 [V] LM35: 25.1 [oC]
Pot.: 1.4523 [V] LM35: 0.2517 [V] LM35: 25.2 [oC]
Pot.: 1.4963 [V] LM35: 0.2595 [V] LM35: 26.0 [oC]
Pot.: 1.4491 [V] LM35: 0.2577 [V] LM35: 25.8 [oC]
Pot.: 1.4403 [V] LM35: 0.2573 [V] LM35: 25.7 [oC]
Pot.: 1.4934 [V] LM35: 0.2525 [V] LM35: 25.3 [oC]
Pot.: 1.5021 [V] LM35: 0.2604 [V] LM35: 26.0 [oC]
Pot.: 1.4985 [V] LM35: 0.2513 [V] LM35: 25.1 [oC]
Pot.: 1.4954 [V] LM35: 0.2495 [V] LM35: 24.9 [oC]
Pot.: 1.4953 [V] LM35: 0.2534 [V] LM35: 25.3 [oC]
Pot.: 1.4908 [V] LM35: 0.2511 [V] LM35: 25.1 [oC]
```



Observe a variação da temperatura!!!

Além do sinal esperado, existe um ruído substancial!!!

As medidas possuem uma incerteza de mais de 5 °C!!!

Esse problema do ruído é ainda mais expressivo se o LM35 estiver distante da placa Baseboard, com o uso de um conjunto de cabos de comprimento maior.

Como resolver isso?!?!



Exercício ADC_Pot_Temp – Uso de médias

```
#include "mbed.h"
AnalogIn Entrada_AN_Potenciometro(PA_0);
AnalogIn Entrada_AN_Sensor(PA_1);
Serial PC(USBTX, USBRX);

int main()
{
    float Medida_Potenciometro, Medida_Sensor, Temperatura, Media=0.0;
    while(1) {
        Medida_Potenciometro=Entrada_AN_Potenciometro*3.3;
        Medida_Sensor=Entrada_AN_Sensor*3.3;
        Temperatura=Medida_Sensor/10E-3; //10mV/C
        Media = (4.0*Media + Temperatura)/5.0;
        PC.printf("Pot.: %3.4f [V] ", Medida_Potenciometro);
        PC.printf("LM35: %3.4f [V] ", Medida_Sensor);
        PC.printf("LM35: %3.1f [oC] ", Temperatura);
        PC.printf("Med.: %3.1f [oC] \r\n", Media);
        wait(0.5); // 500 ms
    }
}
```



Exercício 4 - Montagem DAC

Exercício DAC – Para verificar no osciloscópio

```
#include "mbed.h"
#define PI 3.14159

AnalogOut Saida_AN(PA_4);

int main()
{
    float Frequencia,dT,Angulo;
    dT=1E-3;
    Frequencia=60;
    Angulo=0;
    while(1) {
        Angulo=Angulo+2*PI*Frequencia*dT;
        Saida_AN=0.5*sin(Angulo)+0.5;
        wait(dT);
    }
}
```



Exercício 5 – DAC e potenciômetro no ADC

Exercício ADC-DAC – Para verificar no Osc.

```
#include "mbed.h"
#define PI 3.14159

AnalogOut Saida_AN(PA_4);
AnalogIn Entrada_AN_Potenciometro(PA_0);

int main()
{
    float Frequencia,dT,Angulo;
    dT=1E-3;
    Frequencia=60;
    Angulo=0;
    while(1) {
        Angulo=Angulo+2*PI*Frequencia*dT;
        Saida_AN=(Entrada_AN_Potenciometro/2)*sin(Angulo)+0.5;
        wait(dT);
    }
}
```



Para saber mais

- Mbed, <http://mbed.org>.
- STM32F072 microcontroller ,
<https://www.st.com/resource/en/datasheet/stm32f072rb.pdf>., ultimo acesso Set/2018.
- Silberschatz, A., Galvin, P. and Gagne, G., “Operating System Concepts”, Wiley, 8th Edition, 2008.
- Monk, S., “Hacking Electronics. An illustrated DIY guide for makers and hobbyists”, Mc Graw Hill Education, 2013.



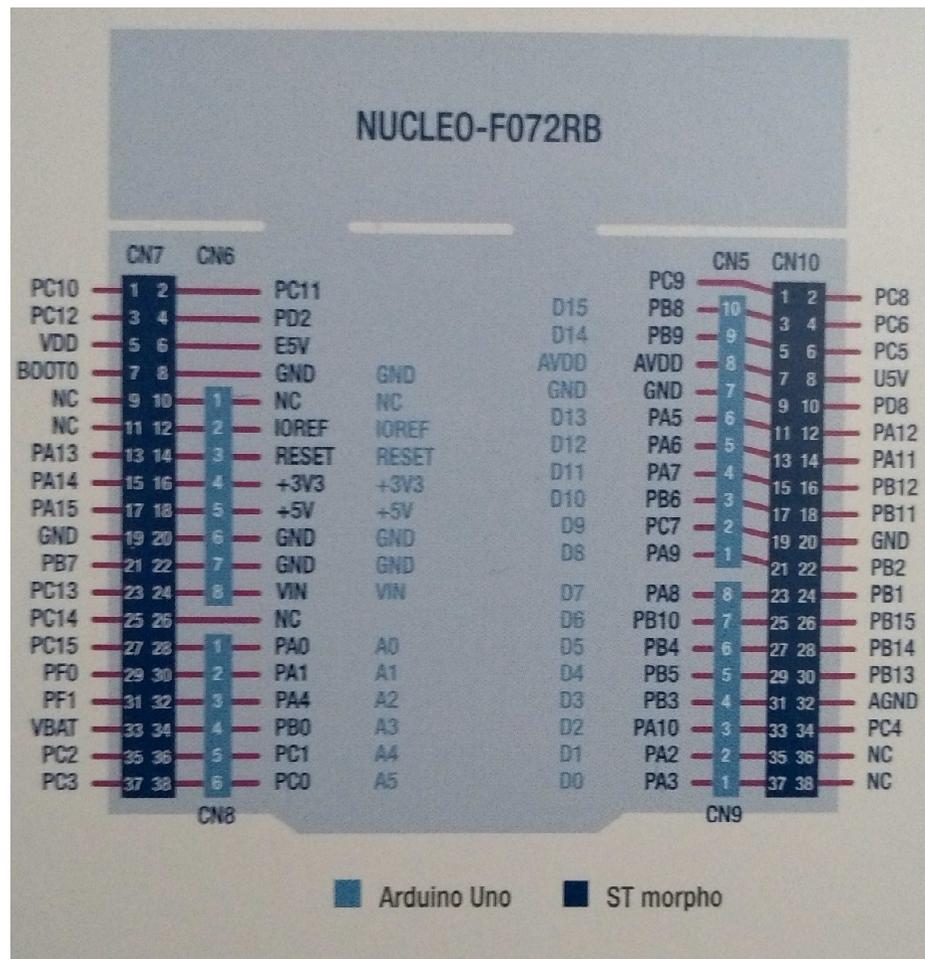
Apêndices

- Materiais para consulta de pinagens do kit.



Lista de ports e funções do kit

- No encarte da caixa do kit existe um guia de referência rápida dos pinos e ports.





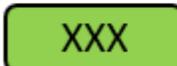
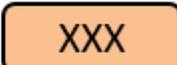
Detalhes dos pinos com suas funções especiais

Labels usable in code

-  MCU pin without conflict
-  MCU pin connected to other components
See [PeripheralPins.c](#) (link below) for more information

-  Arduino connector names (A0, D1, ...)
-  LEDs and Buttons (LED_1, USER_BUTTON, ...)

Labels not usable in code (for information only)

-  Serial pins (USART/UART)
-  SPI pins
-  I2C pins
-  PWMOut pins (TIMER n/c[N])
n = Timer number c = Channel
N = Inverted channel
-  AnalogIn (ADC) and AnalogOut pins (DAC)
-  CAN pins
-  Power and control pins (3V3, GND, RESET, ...)

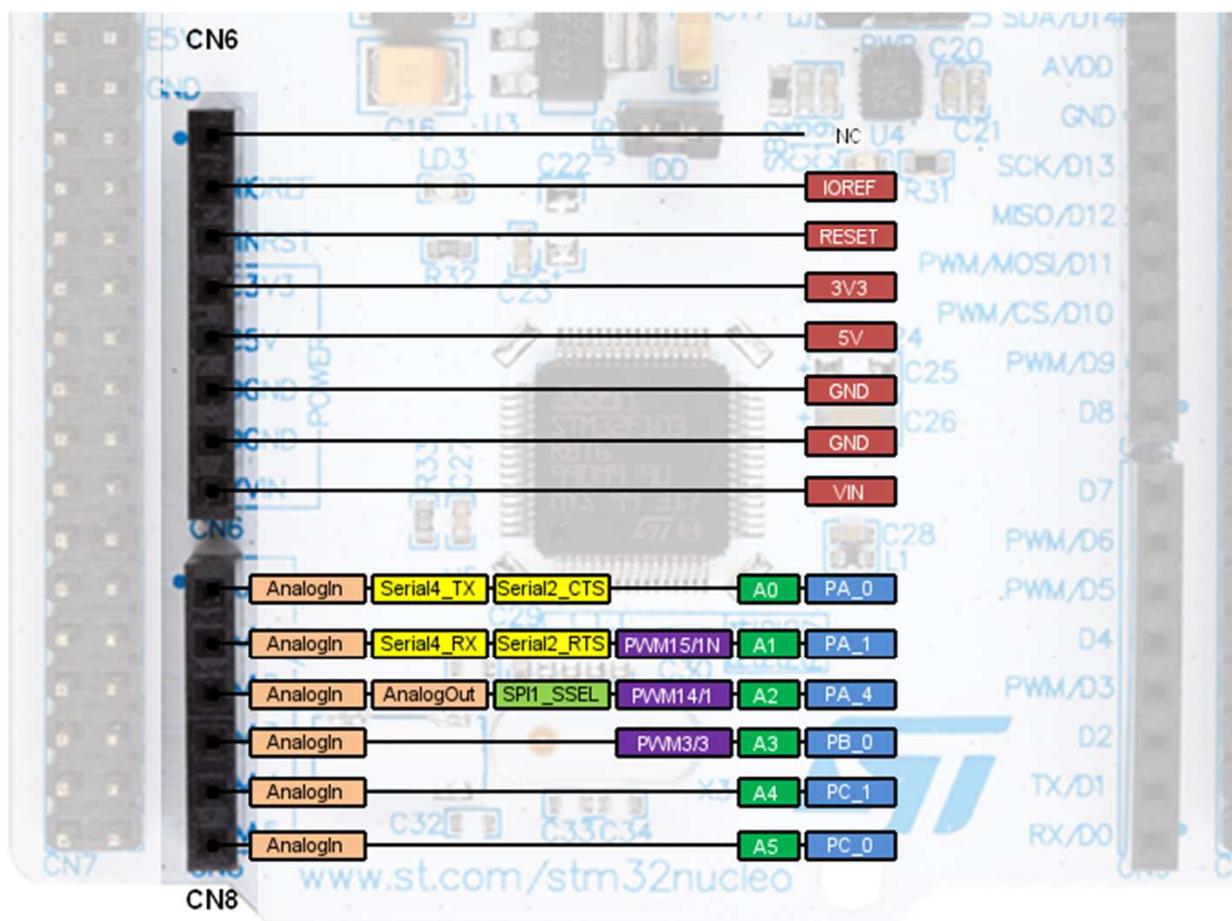


Detalhes dos pinos com suas funções especiais



life.augmented

NUCLEO-F072RB
ARDUINO HEADER
(top left side)



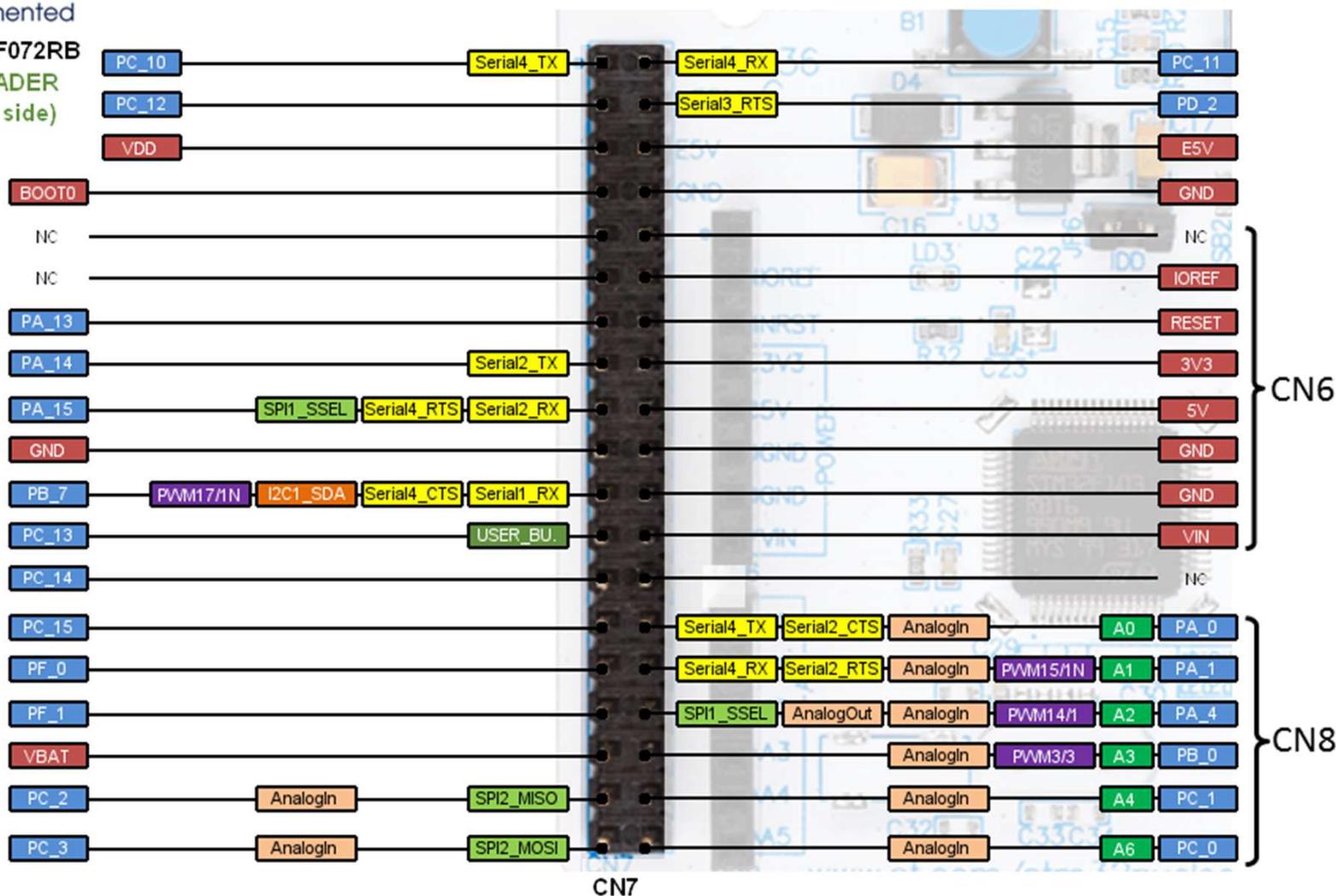


Detalhes dos pinos com suas funções especiais



life.augmented

NUCLEO-F072RB
CN7 HEADER
(top left side)

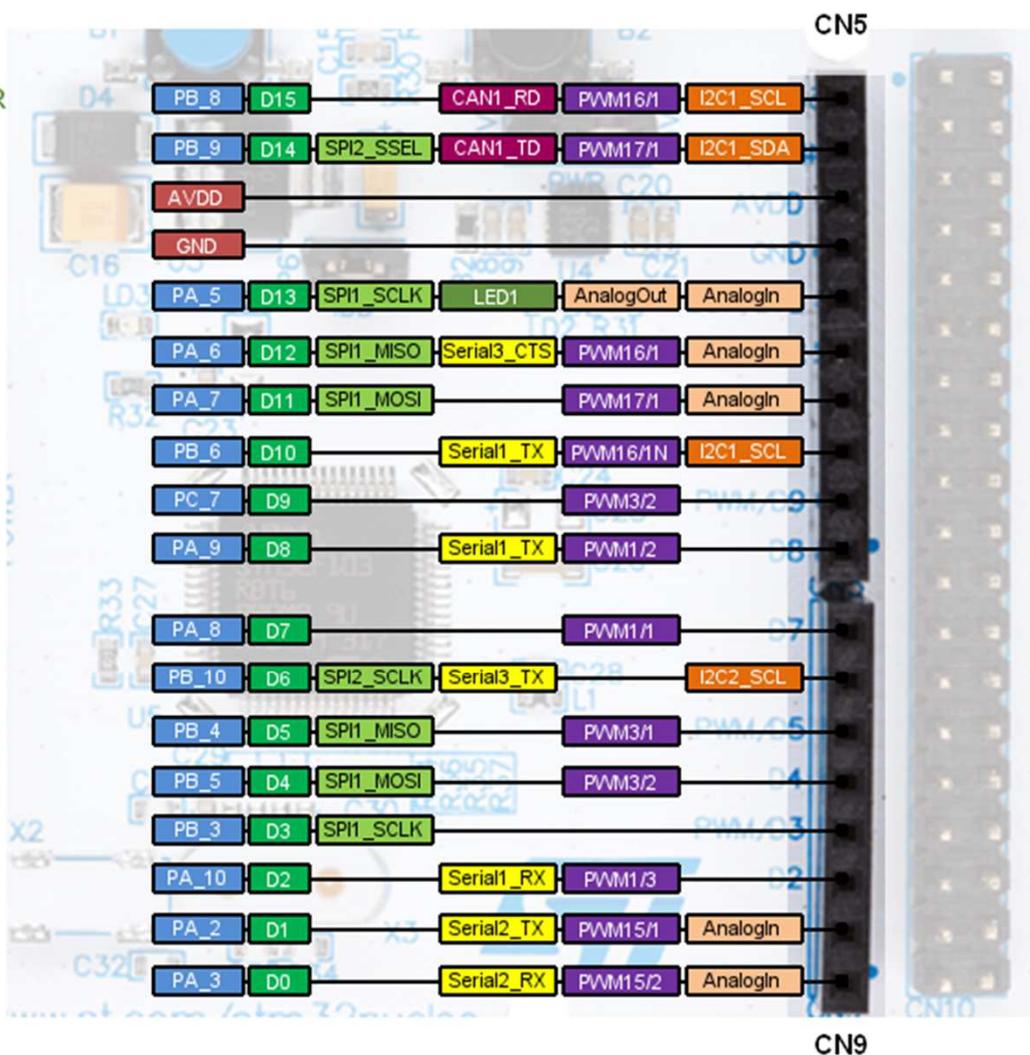




Detalhes dos pinos com suas funções especiais



life.augmented
NUCLEO-F072RB
ARDUINO HEADER
(top right side)





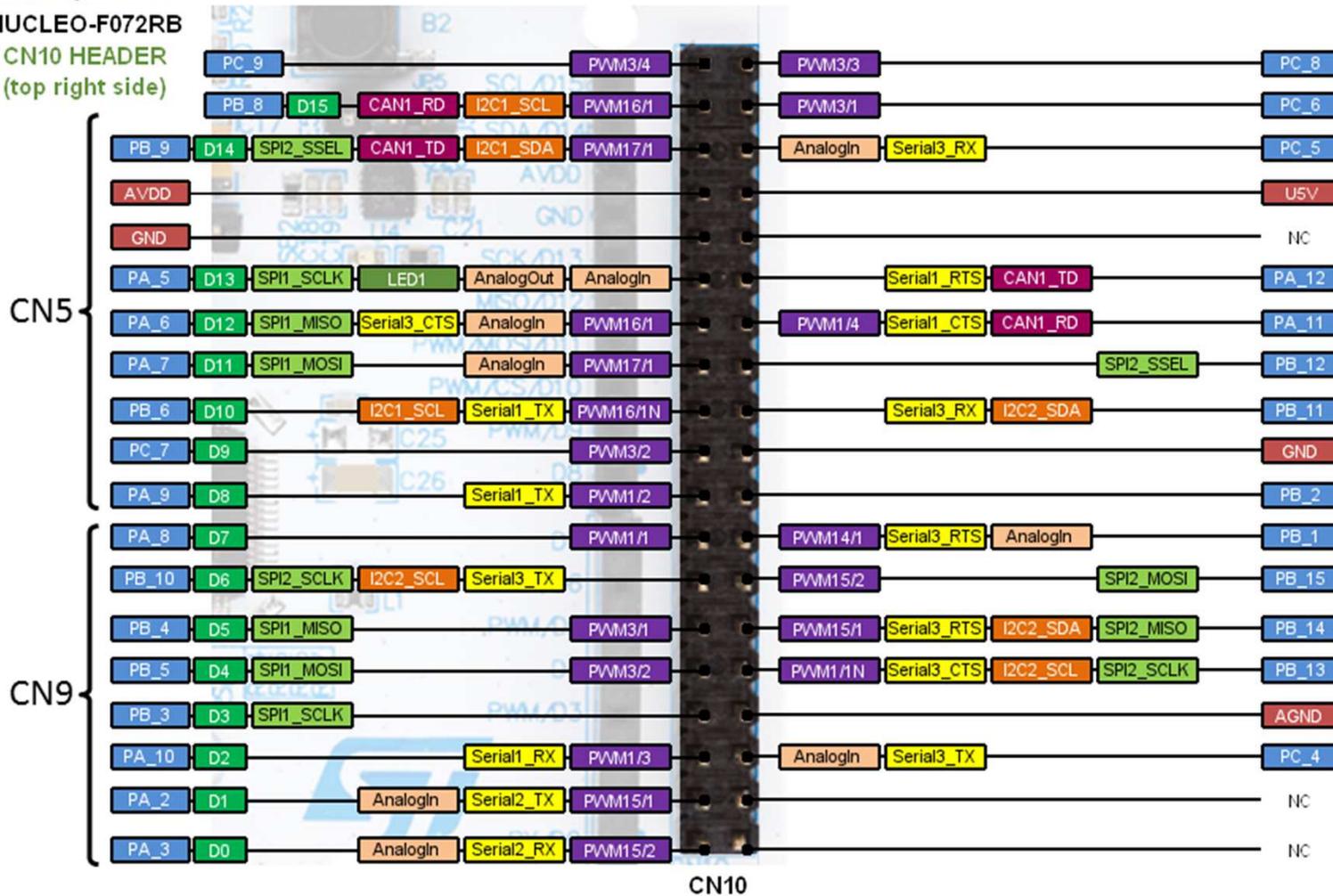
Detalhes dos pinos com suas funções especiais



life.augmented

NUCLEO-F072RB

CN10 HEADER
(top right side)





Lista de ports e terminais do KL25Z

Microcontrolador STM 32 Encapsulamento de 64 pinos		Kit STM32F072 Nucleo-64		
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	VBAT	CN7 33	-	-
2	PC_13	CN7 23	-	User Button
3	PC_14	CN7 25	-	-
4	PC_15	CN7 27	-	-
5	PF_0	CN7 29	-	-
6	PF_1	CN7 31	-	-
7	NRST	CN6 3/ CN7 14	RESET	-
8	PC_0	CN8 6	A5	-
9	PC_1	CN8 5	A4	-
10	PC_2	CN7 35	-	-
11	PC_3	CN7 37	-	-
12	VSSA	CN10 32	-	-
13	VDDA	CN5 8	AVDD	-
14	PA_0	CN8 1/ CN7 28	A0	-
15	PA_1	CN8 2/ CN7 30	A1	-
16	PA_2	CN9 2/ CN10 35	D1	-
17	PA_3	CN9 3/ CN10 37	D0	-
18	VSS	GND	GND	-
19	VDD	VDD	-	-
20	PA_4	CN8 3/ CN7 32	A2	-
21	PA_5	CN5 6/ CN10 11	D13	LED1
22	PA_6	CN5 5/ CN10 13	D12	-
23	PA_7	CN5 4/ CN10 15	D11	-
24	PC_4	CN10 34	-	-
25	PC_5	CN10 6	-	-
26	PB_0	CN8 4/ CN7 32	A3	-
27	PB_1	CN10 24	-	-
28	PB_2	CN10 22	-	-
29	PB_10	CN9 7/ CN10 25	D6	-
30	PB_11	CN10 18	-	-
31	VSS	GND	GND	-
32	VDD	VDD	-	-

Microcontrolador STM 32 Encapsulamento de 64 pinos		Kit STM32F072 Nucleo-64		
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
33	PB_12	CN10 16	-	-
34	PB_13	CN10 30	-	-
35	PB_14	CN10 28	-	-
36	PB_15	CN10 26	-	-
37	PC_6	GN10 4	-	-
38	PC_7	CN5 2/ CN10 19	D9	-
39	PC_8	CN10 2	-	-
40	PC_9	CN10 1	-	-
41	PA_8	CN9 8/ CN10 23	D7	-
42	PA_9	CN5 1/ CN10 21	D8	-
43	PA_10	CN9 3/ CN10 33	D2	-
44	PA_11	CN10 14	-	-
45	PA_12	CN10 12	-	-
46	PA_13	CN7 13	-	-
47	VSS	GND	-	-
48	VDDIO2	-	-	-
49	PA_14	CN7 15	-	-
50	PA_15	CN7 17	-	-
51	PC_10	CN7 1	-	-
52	PC_11	CN7 2	-	-
53	PC_12	CN7 3	-	-
54	PD_2	CN7 4	-	-
55	PB_3	CN9 4/ CN10 31	D3	-
56	PB_4	CN9 6/ CN10 27	D5	-
57	PB_5	CN9 5/ CN10 29	D4	-
58	PB_6	CN5 3/ CN10 17	D10	-
59	PB_7	CN7 21	-	-
60	BOOT0	CN7 7	-	-
61	PB_8	CN5 10/ CN10 3	D15	-
62	PB_9	CN5 9/ CN10 5	D14	-
63	VSS	GND	GND	-
64	VDD	VDD	-	-