

Introdução a Engenharia Elétrica - 323100

Aula S12

Comunicação serial, Display LCD

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

V1.5

Outubro de 2015



Comunicação Serial - revisão

- Comunicação com um PC host através de um "USB Virtual Serial Port" através do mesmo cabo USB que é usado para a programação. Isso permite que você:
 - Imprima mensagens para um terminal host PC (útil para depuração!)
 - Leia a entrada do teclado do PC host
 - Comunique-se com aplicativos e linguagens de programação em execução no PC host que podem se comunicar com uma porta serial, por exemplo, Perl, Python, Java, etc.



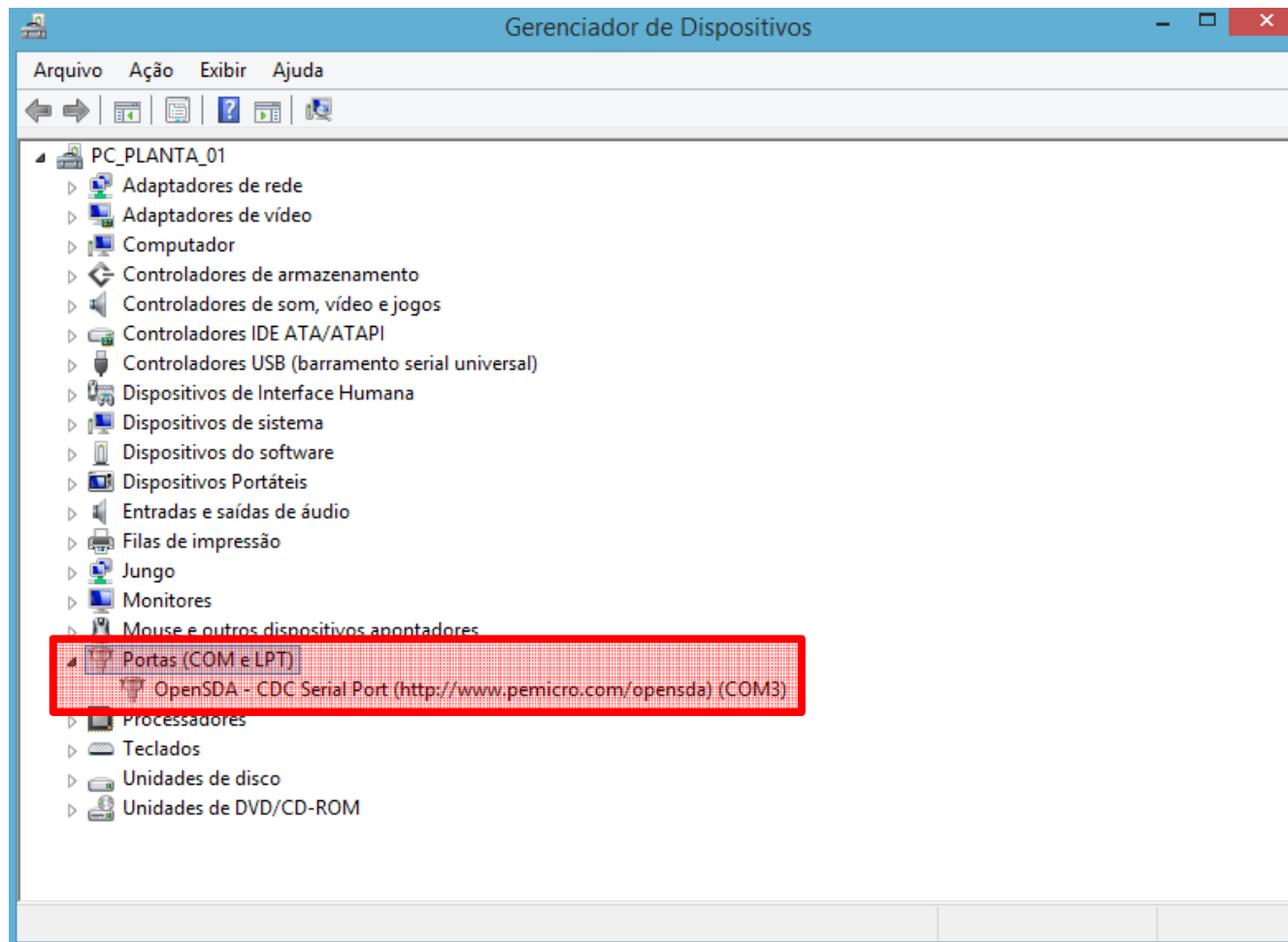
No Mbed:

- **Serial name(USBTX, USBRX)**: define a comunicação serial com TX e RX via USB DAS. Define o label name.
- **name.baud(9600)**: define a taxa de transmissão (baud rate) em 9600 kbps. Aceita outros valores (19200, 38400, etc).
- **name.printf**: imprime *string* STM32 → PC;
- **name.putc**: envia um caractere STM32 → PC;
- **name.getc**: recebe um caractere PC → STM32.



Comunicação Serial

- Ao conectar o cabo USB, aparece enumerada no computador uma porta OpenSDA – CDC.





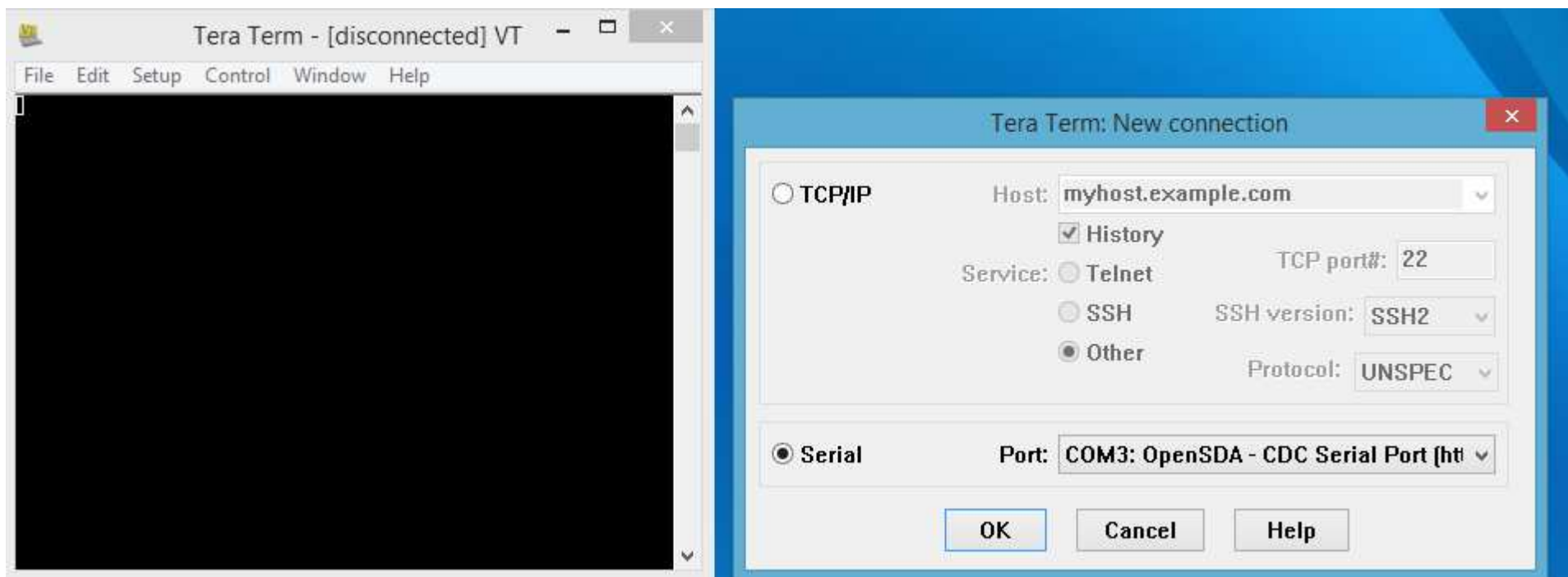
Comunicação Serial

- Caso não apareça no sistema windows, é necessário instalar o driver. Busque por **Download the mbed Windows serial port driver** no *Dashboard* do mbed.
- Para o curso, vamos utilizar um aplicativo emulador de terminal denominado Tera Term (<http://ttssh2.sourceforge.jp/index.html.en>).



Comunicação Serial

- Escolher a opção serial e a porta enumerada.





Exemplos Básicos no Mbed

- Hello World!

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX); // tx, rx
int main() {
    pc.printf("Hello World!\n");
}
```

- *Papagaio de pirata - Echo back characters you type*

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX);
int main() {

    pc.printf("Echoes back to the screen anything you type\n");
    while(1) {
        pc.putc(pc.getc());
    }
}
```



Outros Exemplos:

- Controla brilho do LED ('u' p/ UP e 'd' p/ DOWN).

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX); // tx, rx
PwmOut led(PA_9);
int main() {
    char c;
    float brightness = 1.0;
    pc.printf("Press 'u' para aumentar brilho e 'd' para diminuir \n\r");
    while(1) {
        c = pc.getc();
        if((c == 'u') && (brightness > 0.0)) {
            brightness -= 0.1;
            led = brightness;
        }
        if((c == 'd') && (brightness < 1.0)) {
            brightness += 0.1;
            led = brightness;
        }
        pc.printf("Brilho   = %2.0f %% \r", (1.0-led)*100);
    }
}
```




Comunicação Serial

- Escolhe o LED ('1', '2' o '3'. Qualquer outra tecla apaga tudo).

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX); // tx, rx
DigitalOut LED_1(PA_8); DigitalOut LED_2(PB_10); DigitalOut LED_3(PA_9);
int main() {
    char c;
    LED_1=0; LED_2=0; LED_3=0;
    pc.printf("Pressione o numero do led desejado (1 ou 2 ou 3)");
    while(1) {
        c = pc.getc();
        switch(c){
            case '1':
                LED_1=1; LED_2=0; LED_3=0;
                break;
            case '2':
                LED_1=0; LED_2=1; LED_3=0;
                break;
            case '3':
                LED_1=0; LED_2=0; LED_3=1;
                break;
            default:
                LED_1=0; LED_2=0; LED_3=0;
                break;
        }
    }
}
```



Display LCD

- Há diversos tipos e tamanhos de LCD alfanuméricos disponíveis comercialmente.
- Eles são sempre especificados em número de caracteres exibidos, no formato de colunas e linhas.
- Mais comuns: 08x02 (oito colunas por duas linhas), 16x01 (16 colunas por 1 linha), **16x02 (16 colunas por 2 linhas)**, 16x04 (16 colunas por 4 linhas), 20x01 (20 colunas por 1 linha), 20x02 (20 colunas por 2 linhas) e 20x04 (20 colunas por 4 linhas).

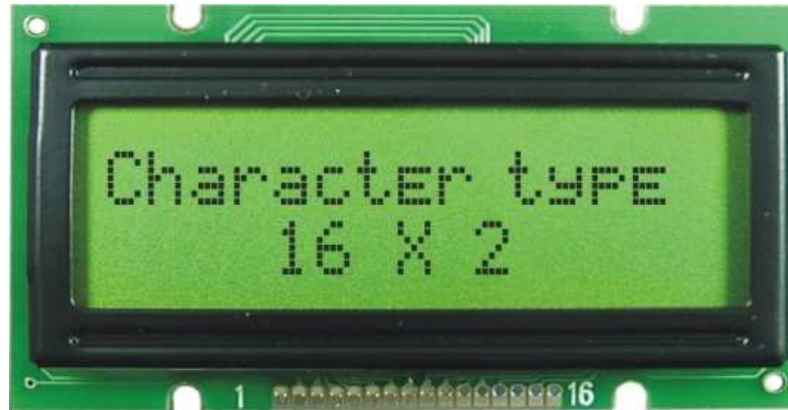


Funcionamento básico do cristal líquido

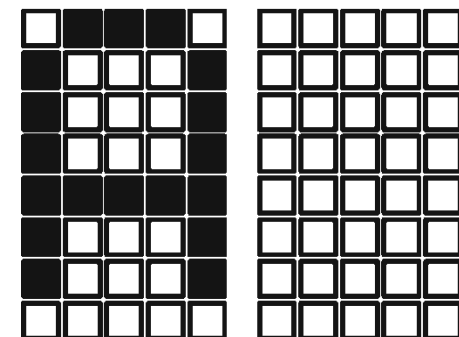
- São dispositivos que trabalham com luz polarizada com um líquido eletricamente controlável.
- Ao se aplicar ou retirar uma tensão em alguns locais (pixels), os cristais do líquido são polarizados, permitindo ou impedindo a passagem da luz.
- Não são elementos que emitem luz própria, apenas alteram o estado de reflexão da luz ambiente.
- Para funcionamento em locais escuros, precisam de uma fonte de luz na parte traseira, denominada *backlight*.



Exemplo de display LCD 16x02



- Cada “célula”(caractere) do LCD possui 8 pixels na vertical e 5 pixels na horizontal.
- Cada caractere é desenhado com até 35 pixels, pois a linha inferior é normalmente reservada para representar um cursor.





Interface com Display LCD

- Os LCDs mais comuns são os gerenciados por um chip controlador Hitachi HD44780.
- O módulo do display deve ser ligado à placa através de um conjunto de conexões elétricas, denominadas **barramento de dados**.
- O barramento de dados pode ser de quatro bits (modo *nibble*) ou oito bits (modo *byte*).
- No modo *nibble*, são usadas apenas quatro linhas de dados (D4 a D7), simplificando a montagem.



Pinagem

- Na maioria dos displays, a pinagem está impressa.

PINO	SÍMBOLO	FUNÇÃO
1	VSS	GND (0V)
2	VDD	DC +5V
3	V ₀ (VEE)	Ajuste de contraste
4	D/I (RS)	Dados/Instruções
5	R/W	Leitura / Escrita
6	E	Enable
7	DB0	Linha de dados 0
8	DB1	Linha de dados 1
9	DB2	Linha de dados 2
10	DB3	Linha de dados 3
11	DB4	Linha de dados 4
12	DB5	Linha de dados 5
13	DB6	Linha de dados 6
14	DB7	Linha de dados 7
15	A	LED +
16	K	LED –

} Alimentação

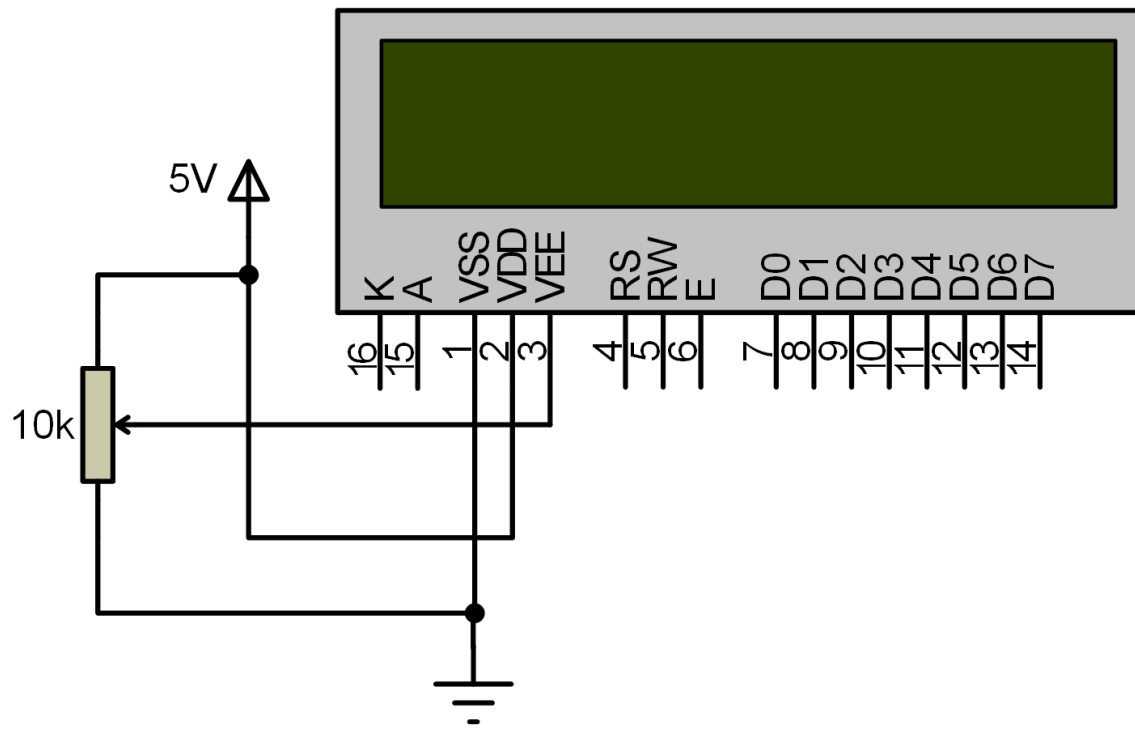
} Barramento de comunicação

} Energia para os LEDs de backlight



Interface com Display LCD

- O pino 1 (V_{ss}) é ligado ao terra e o pino 2 (V_{dd}) na tensão de 5V.
- No pino 3 (V_o) deve-se ter uma tensão entre 0 e 5V para ajuste do contraste. Isso pode ser feito com um potenciômetro, como abaixo:





Interface com Display LCD

- No modo *nibble*, o display é controlado por meio de comandos e dados, enviados pelos pinos RS, RW, D4, D5, D6 e D7)
- O pino RS designa o tipo de informação enviada:
 - Se RS=0: envio de comandos;
 - Se RS=1: envio de dados.
- O pino RW controla a natureza da operação.
 - Se RW=0: operação de escrita de dados no display;
 - Se RW=1: operação de leitura.
- Também possui um pino de habilitação (ENABLE).
- Em geral, deixa-se permanentemente o pino ENABLE ligado em 3,3V e o pino RW ligado em GND.



Biblioteca TextLCD

- Biblioteca disponível no MBED para utilização do display LDC no STM32, modo *nibble*.
- A mesma deve ser incluída no cabeçalho do programa principal: `#include "TextLCD.h"`
- O seguinte comando define a pinagem utilizada do kit para interface com o display LCD:

```
TextLCD lcd(RS, Enable, DB4, DB5, DB6, DB7);
```

- Como exemplo, o comando:

```
TextLCD lcd(PA_9, PC_7, PB_5, PB_4, PB_10, PA_8);
```

faz as seguintes atribuições:



Biblioteca TextLCD

/* PINAGEM DA LIGAÇÃO STM32 - LDC **

* - DB4	: d4 -> PB_5
* - DB5	: d5 -> PB_4
* - DB6	: d6 -> PB_10
* - DB7	: d7 -> PA_8
* - RS	: d8 -> PA_9
* - Enable	: d9 -> PC_7
* - RW	: GND (only W mode)

*** /



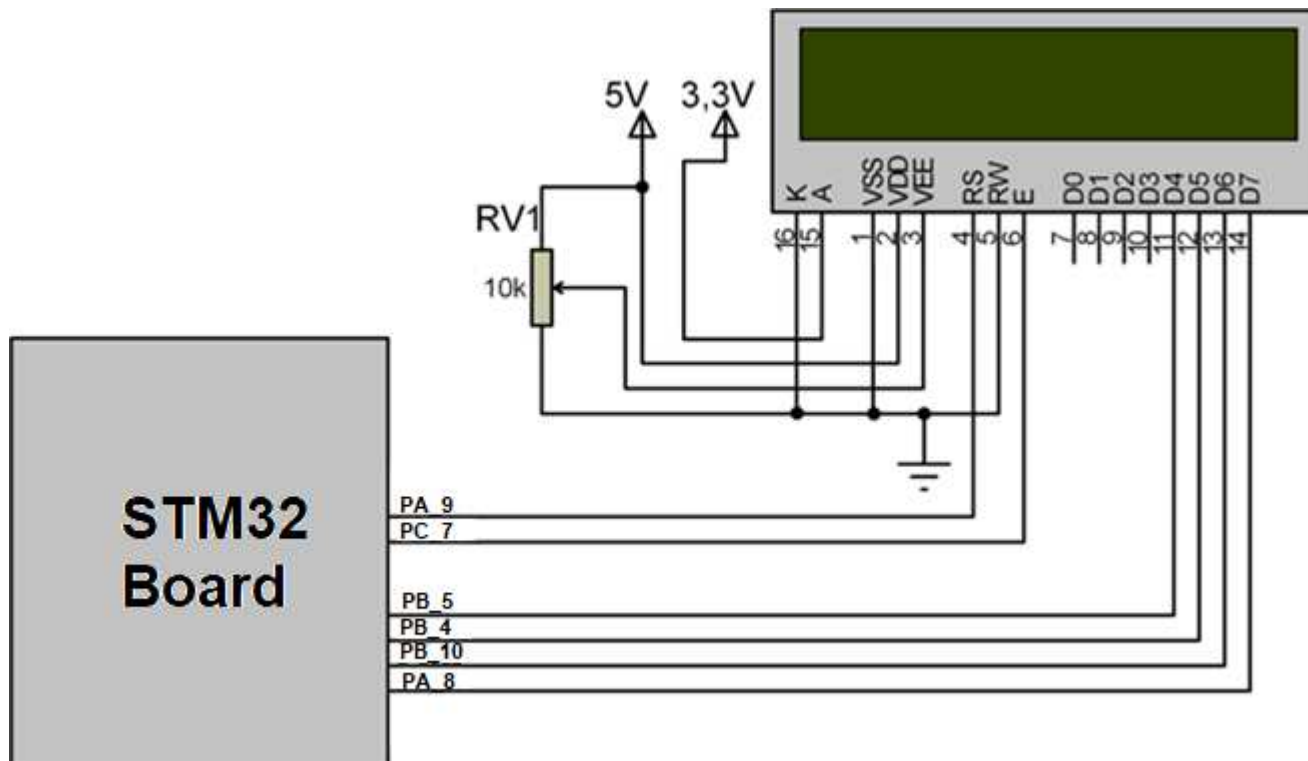
Biblioteca TextLCD

- O comando `lcd.cls()` limpa os caracteres do display;
- O comando `lcd.printf()` imprime *strings* constantes e variáveis no display.
- O comando `lcd.locate(x,y)` posiciona o display na coluna x (0 a 15), linha y (0 ou 1).



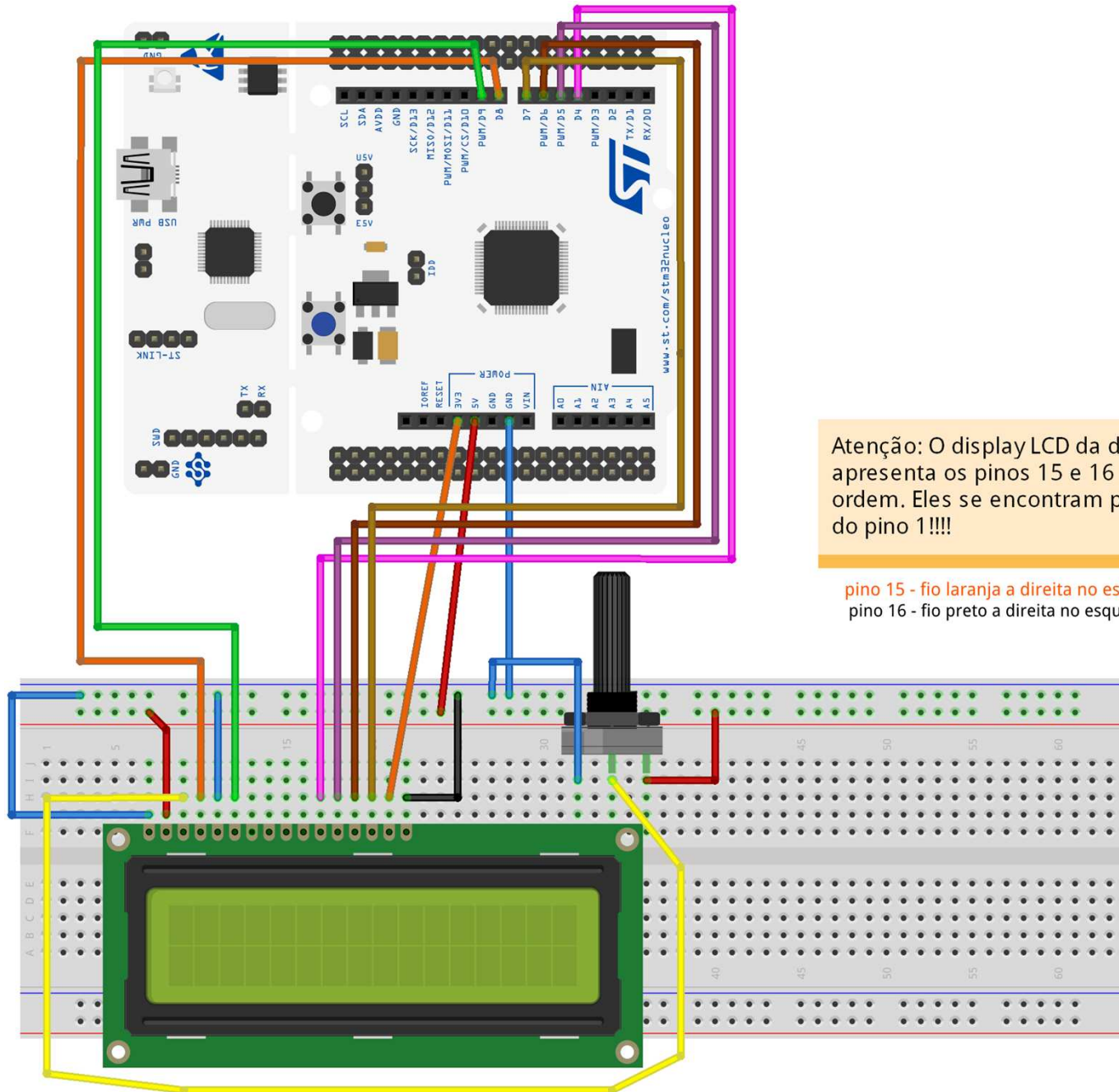
Exemplo de Aplicação

- Utilize o display LCD 16x02 para mostrar o valor do acumulador de um contador cíclico de 0 a 59, com incremento a cada 0,5 segundos.





Montagem com Display 7 Segmentos



Atenção: O display LCD da disciplina apresenta os pinos 15 e 16 fora de ordem. Eles se encontram próximos do pino 1!!!!

pino 15 - fio laranja a direita no esquema
pino 16 - fio preto a direita no esquema



Código

```
#include "mbed.h"
#include "TextLCD.h"
TextLCD lcd(PA_9, PC_7, PB_5, PB_4, PB_10, PA_8);
int main() {
    int count = 0;
    lcd.cls();
    lcd.printf("CONTA DE 0 A 59");
    wait(0.5);
    while (1) {
        lcd.locate(0,1);
        lcd.printf("CT=%2d", count);
        count = count + 1;
        wait(0.5);
        if (count == 60) count=0;
    }
}
```



Observação:

- A biblioteca TextLCD.h precisa ser importada no projeto.

