

# BARRAMENTO / PROTOCOLO I2C (I<sup>2</sup>C)

## Utilização do módulo I2C com o display LCD 1602 (16x2) Arduino UNO R3

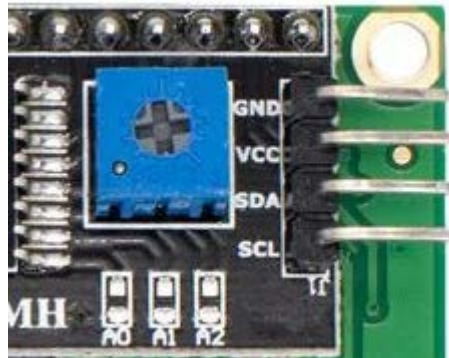
O módulo I2C, também denominado I<sup>2</sup>C ou IIC (Inter-Integrated Circuit) é um barramento compartilhado muito popular e poderoso usado para comunicação entre um mestre (ou vários mestres) e um ou vários dispositivos escravos.

Podemos então dizer que é uma comunicação mestre-escravo na qual podemos conectar e controlar vários escravos a partir de um único mestre.

É um barramento / protocolo de comunicação serial de dados, síncrono, no qual o mestre inicia a comunicação e os escravos são controlados por endereçamento.

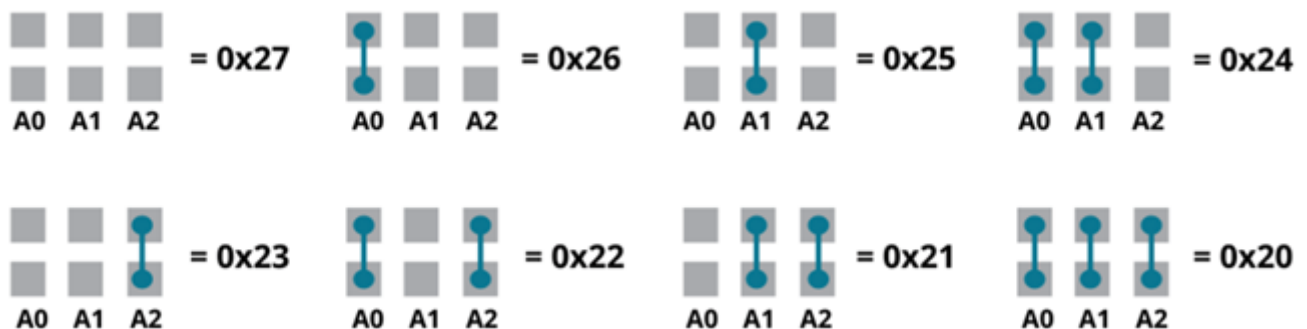
O protocolo I2C simplifica muito a comunicação entre dispositivos mestre-escravo por utilizar somente duas vias de comunicação. O módulo I2C possui 4 terminais, sendo dois para comunicação e dois destinados à alimentação do módulo.

A entrada de comunicação do módulo I2C é feita através de dois fios (terminais) identificados como: SDA (Serial Data) e SCL (Serial Clock). Os dois fios (terminais) restantes destinam-se à alimentação (VCC-GND) tipicamente de 3,3V ou 5V.



Neste artigo trataremos da utilização do módulo I2C com o display LCD 16x2, no entanto, pode ser totalmente aplicável nos displays LCD 20x4.

Observe que logo abaixo do trimpot existem 3 jumpers ou pads identificados como A0, A1 e A2 que servem para alterar o endereço de comunicação do dispositivo. Veja abaixo a forma como esses pads podem ser configurados e seus endereçamentos resultantes:



Conclui-se então que, se nenhum jumper for fechado o endereço do dispositivo é 0x27. Para certificar-se de que o endereçamento esteja configurado corretamente através dos jumpers, basta executar uma rotina bem simples:

### Sketch para verificar endereçamento do módulo I2C

```
#include <Wire.h>
void setup()
{
    Wire.begin();

    Serial.begin(9600);
    while (!Serial);
    Serial.println("\nI2C Scanner");
}

void loop()
{
    byte error, address;
    int nDevices;
    nDevices = 0;
    for(address = 1; address < 127; address++ )
    {
        Wire.beginTransmission(address);
        error = Wire.endTransmission();
        if (error == 0)
        {
            Serial.print("Endereço I2C encontrado: 0x");
            if (address<16)
                Serial.print("0 ");
            Serial.println(address,HEX);

            nDevices++;
        }
        else if (error==4)
        {
            Serial.print("ERRO ");
            if (address<16)
                Serial.print("0");
            Serial.println(address,HEX);
        }
    }
    if (nDevices == 0)
        Serial.println("Nenhum endereço i2C encontrado ");
    else

        Serial.println(" Feito !");

    delay(5000);
}
```

Executando o sketch e abrindo o Serial Monitor, deverá aparecer a informação do endereçamento, se tudo estiver correto.

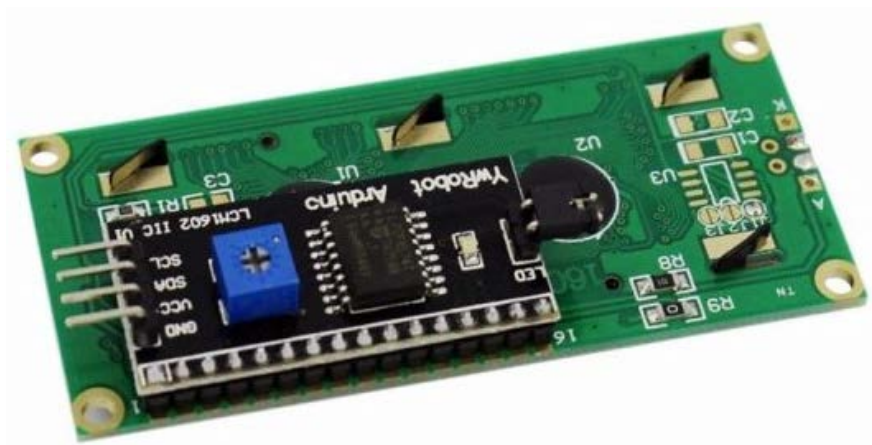
```
i2c-address | Arduino IDE 2.1.0
File Edit Sketch Tools Help

i2c-address.ino
30  ...else-if (error==4)
31  ...{
32  ...  Serial.print("ERRO-");
33  ...  if (address<16)
34  ...    Serial.print("0");
35  ...    Serial.println(address,HEX);
36  ...  }
37  ...}
38  ...if (nDevices==0)
39  ...  Serial.println("Nenhum endereço i2C encontrado.");
40  ...else
41  ...
42  ...  Serial.println("Feito!");
43  ...
44  ...  delay(5000);
45  ...}
46

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM11')

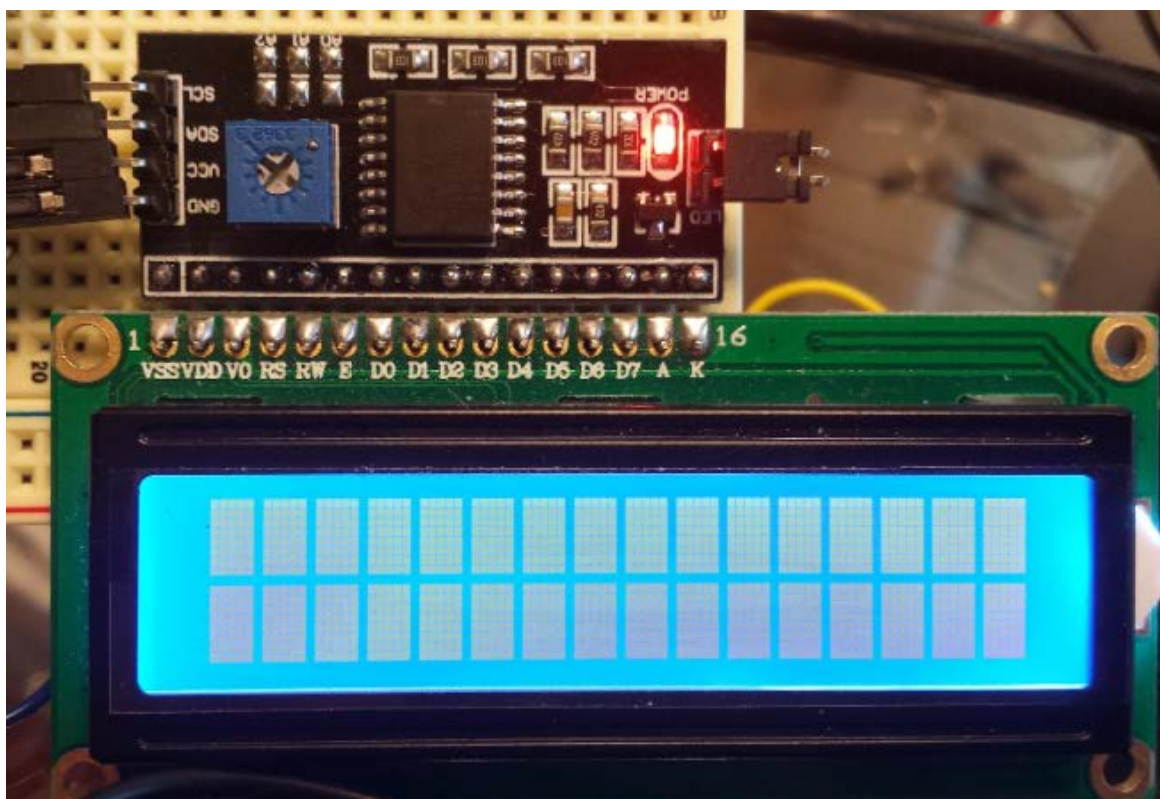
Endereço I2C encontrado: 0x27
Feito !
Endereço I2C encontrado: 0x27
Feito !
```

O endereço encontrado foi exatamente o 0x27 por estarem todos os jumpers abertos. Como temos 3 variáveis (A0, A1, A2) teremos 8 combinações ou opções de endereçamento uma vez que  $2^3 = 8$ . A partir daí podemos então interligar o I2C no display e rodar uma rotina de programação no Arduino para conferir os resultados.



*Módulo I2C soldado ao display LCD 1602*

A figura a seguir mostra outra opção para ligação do I2C ao display, utilizando o protoboard. É muito importante observar a posição do dispositivo por conta da conexão correta entre os terminais.



Controle da luminosidade  
(backlight)



Pinos de conexão  
I2C



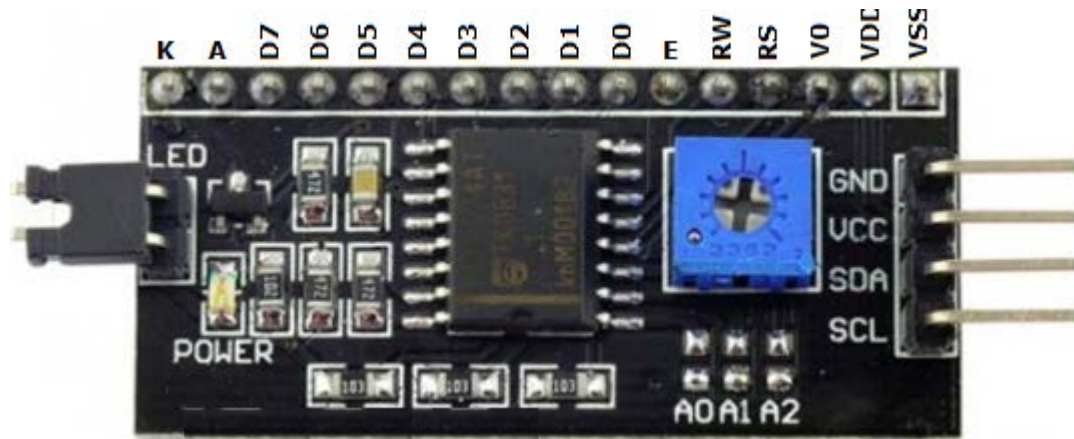
Configuração de  
endereçamento I2C



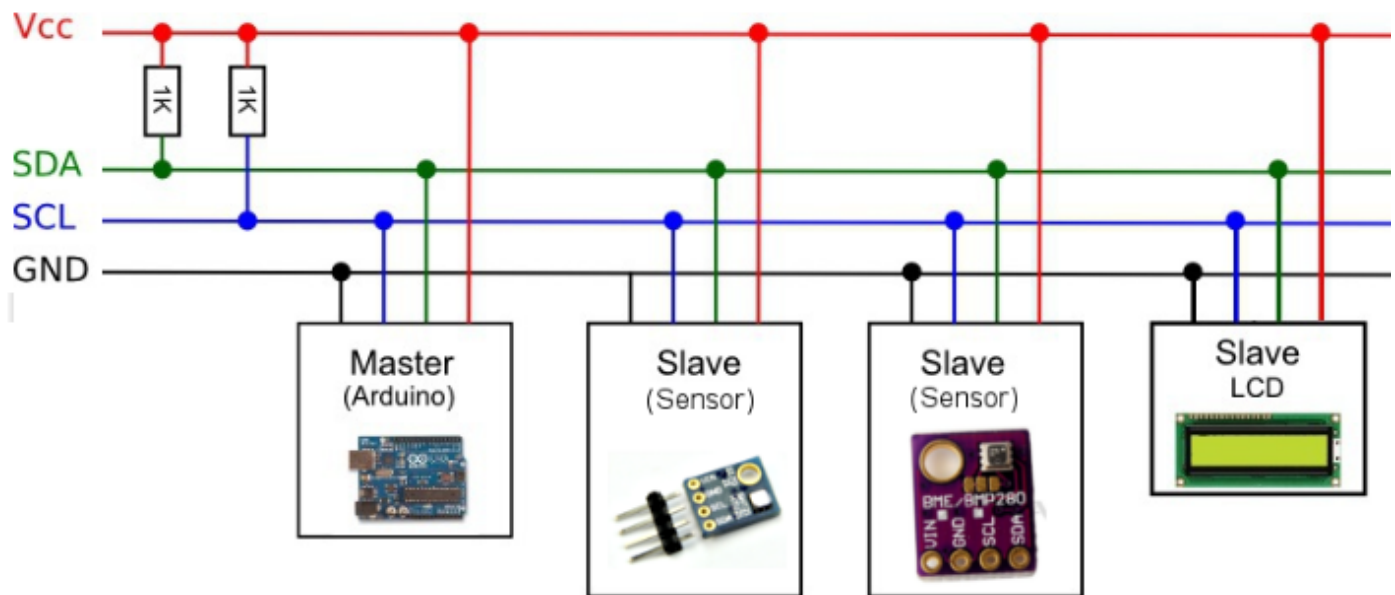
Jumper que permite ligar  
ou desligar a luz de fundo  
(backlight)

## Módulo I2C





Veja abaixo um diagrama bem simples de comunicação (protocolo I2C com barramento compartilhado) entre um dispositivo MASTER, no caso o Arduino, com dispositivos ESCRAVOS (Slave).



*Displays LCD 16 x 2 – backlight azul e verde*



**VCC do IC2 ao 5V do Arduino**  
**GND do IC2 ao GND do Arduino**  
**SDA do IC2 ao A4 do Arduino (Analog In)**  
**SCL do IC2 ao A5 do Arduino (Analog In)**

Veja a seguir o sketch para rodar um texto simples:

```
#include <Wire.h> // biblioteca utilizada para fazer a comunicação com o I2C
#include <LiquidCrystal_I2C.h> // biblioteca utilizada para fazer a comunicação com o
display 16 x 2

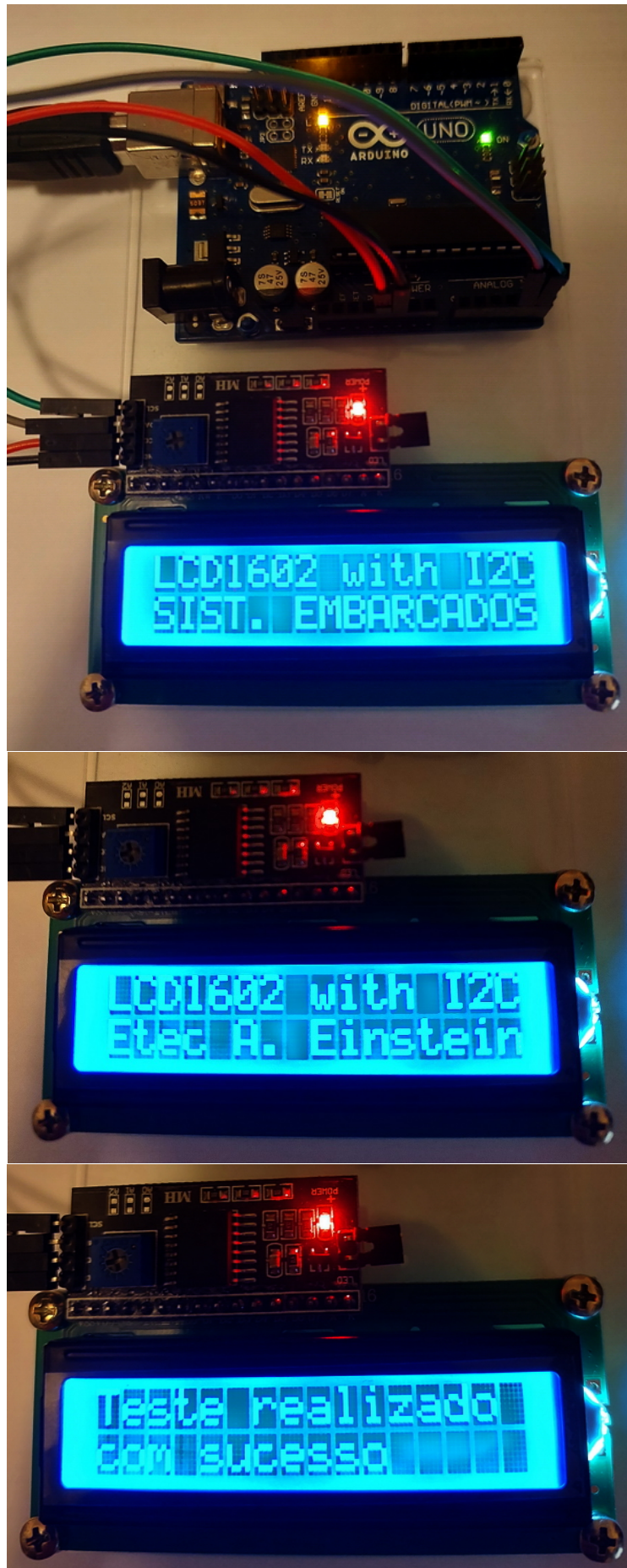
#define col 16 // identifica o tipo de display quanto as colunas (16 colunas)
#define lin 2 // identifica o tipo de display quanto as linhas (2 linhas)
#define ende 0x27 // identifica o endereçamento do display

LiquidCrystal_I2C lcd(ende,col,lin); // funções do lcd (ende= endereçamento; col =
colunas; lin = linhas)

void setup() // incia o display
{
    lcd.init(); // comunicação com o display que já está conectado - verificar o
endereçamento
    lcd.backlight(); // controla a luz de fundo do display (backlight on/off)
    lcd.clear(); // limpa a tela do display
}
void loop()
{
    lcd.setCursor(0,0); // Coloca o cursor do display na coluna 0 e linha 0
    lcd.print("LCD1602 with I2C"); // saída da mensagem que deve aparecer na primeira linha
do display
    delay(1000);
    lcd.setCursor(0,1); //Coloca o cursor do display na coluna 0 e linha 1
    lcd.print("ADDRESS:0x27"); // saída com a mensagem que deve aparecer na segunda linha do
display
    delay(5000);
    lcd.setCursor(0,1);
    lcd.print("SIST. EMBARCADOS");
    delay(5000);
    lcd.setCursor(0,1);
    lcd.print("Etec A. Einstein");
    delay(5000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Teste realizado");
    delay(500);
    lcd.setCursor(0,1);
    lcd.print("com sucesso");
    delay(5000);
    lcd.clear(); // Limpa o display até o loop ser reiniciado
    delay(5000); // tempo para reinício do loop
}
```



Veja abaixo figuras que mostram detalhes do funcionamento:





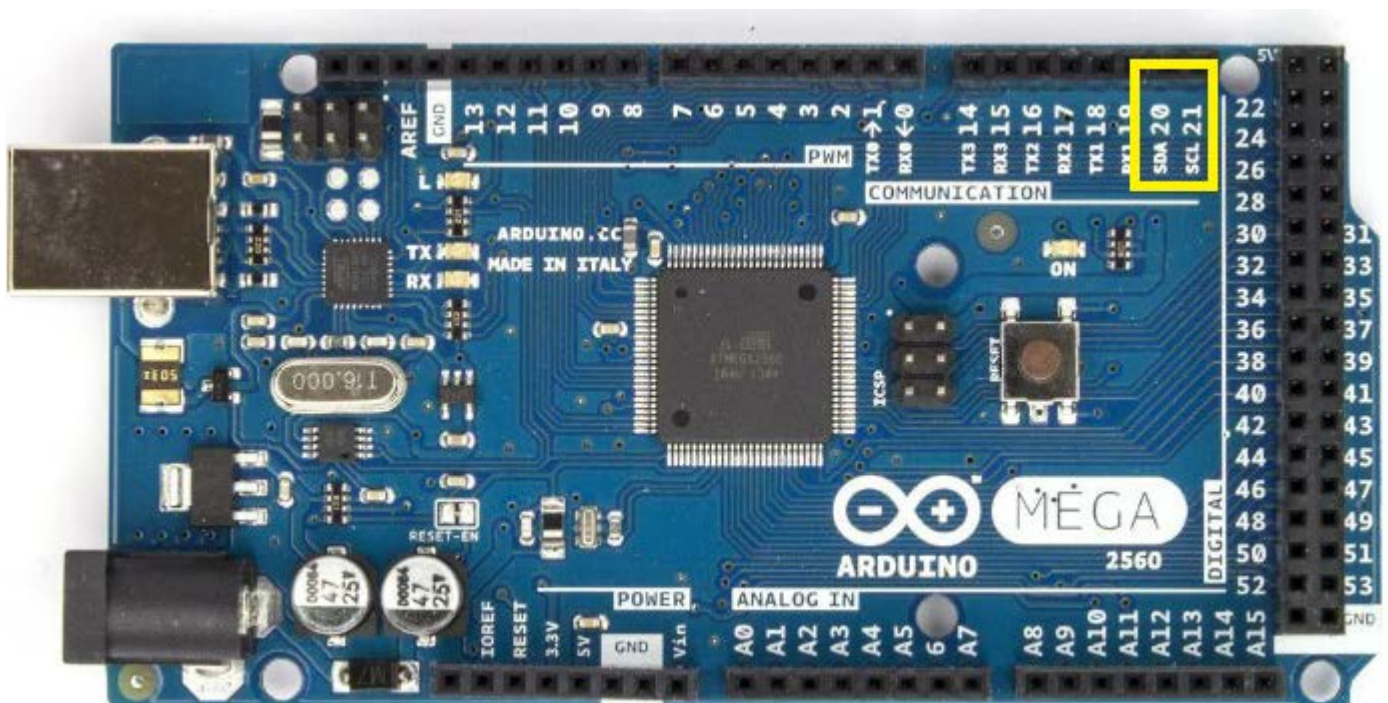
# CONCLUSÕES:

A comunicação I2C é um protocolo de comunicação serial que permite que dispositivos diferentes se comuniquem e troquem dados através de um barramento compartilhado.

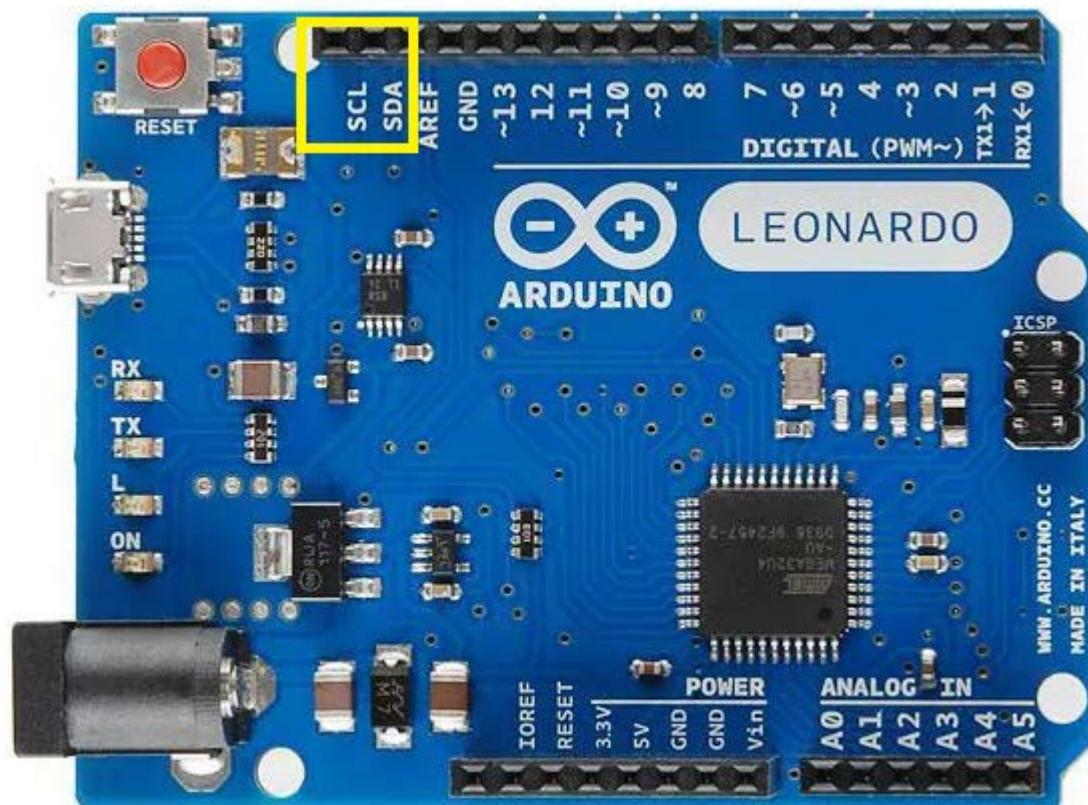
Nos Arduinos UNO e NANO essa comunicação é feita através dos pinos A4 (SDA) e A5 (SCL). Observe na figura abaixo a correlação entre A4 e A5 com SDA e SCL para o Arduino NANO.



Já no Arduino MEGA existem os terminais específicos para tal comunicação (terminais 20 e 21), conforme ilustra a figura abaixo:



A figura a seguir mostra os terminais exclusivos SDA e SCL para o Arduino LEONARDO:



A figura abaixo mostra mais um esquema de comunicação entre dispositivos, usando o protocolo/barramento I2C, onde pode-se observar que o Arduino pode operar tanto como Mestre (Master) ou como Escravo (Slave):

