Protocolos de Comunicação

Prof. Dr. Rafael Traldi Moura

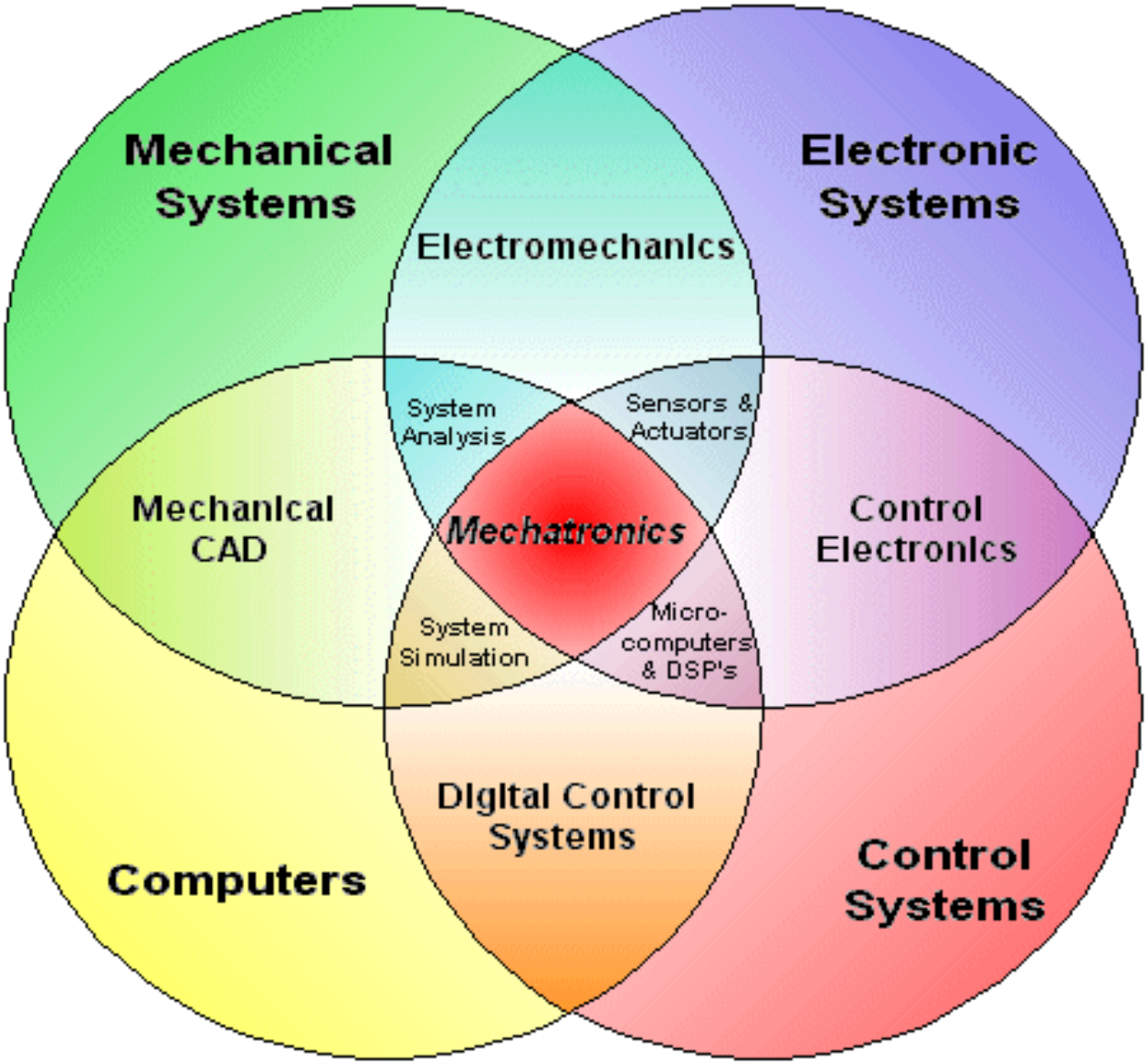
Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos

Mecatrônica

Objetivo

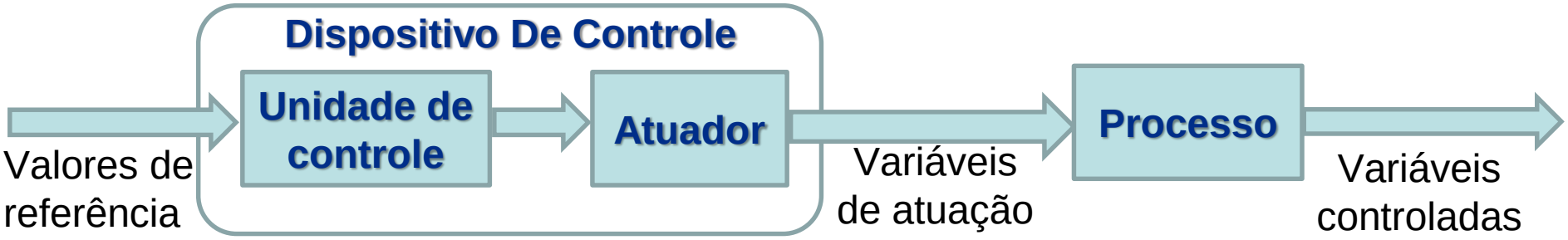
- Para se evitar os desastres do vídeo, é necessário o uso correto de sensores e atuadores, especialmente na **transferência dos dados**.
- Para tal, teremos uma aula **prática** de protocolos de comunicação e MEMs.

Mecatrônica

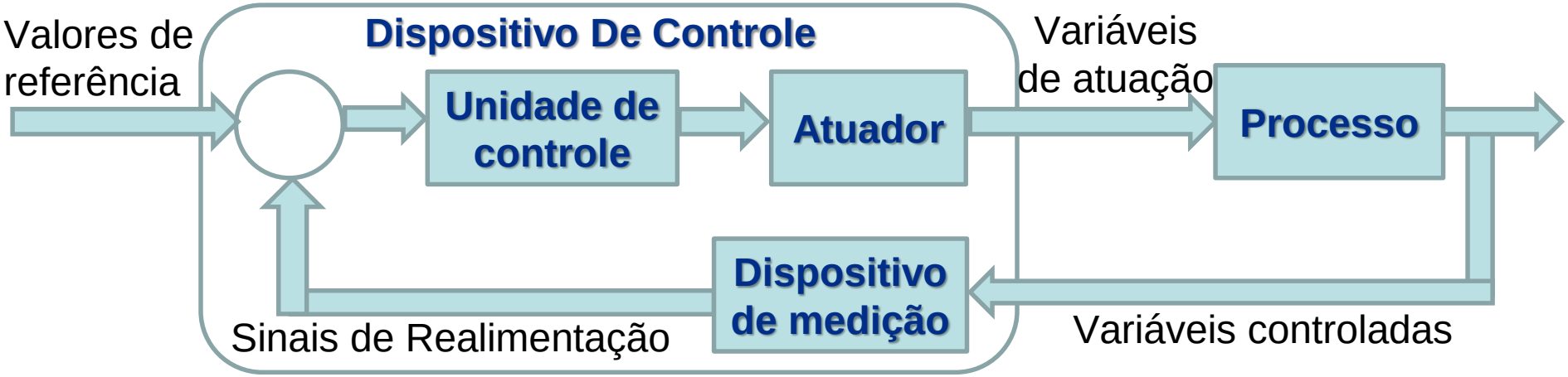


Controle

Sair de um sistema com controle de malha aberta:

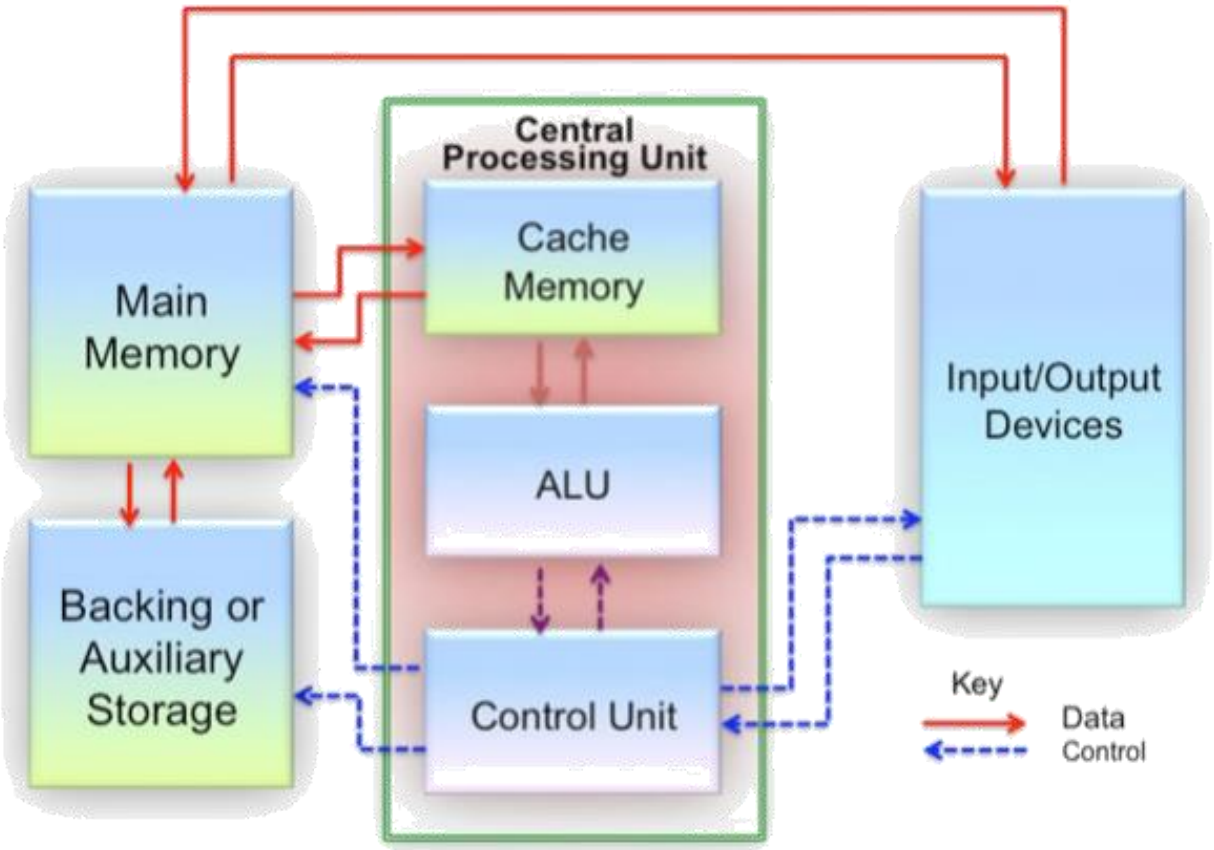


E implementar um sistema com controle de malha fechada:



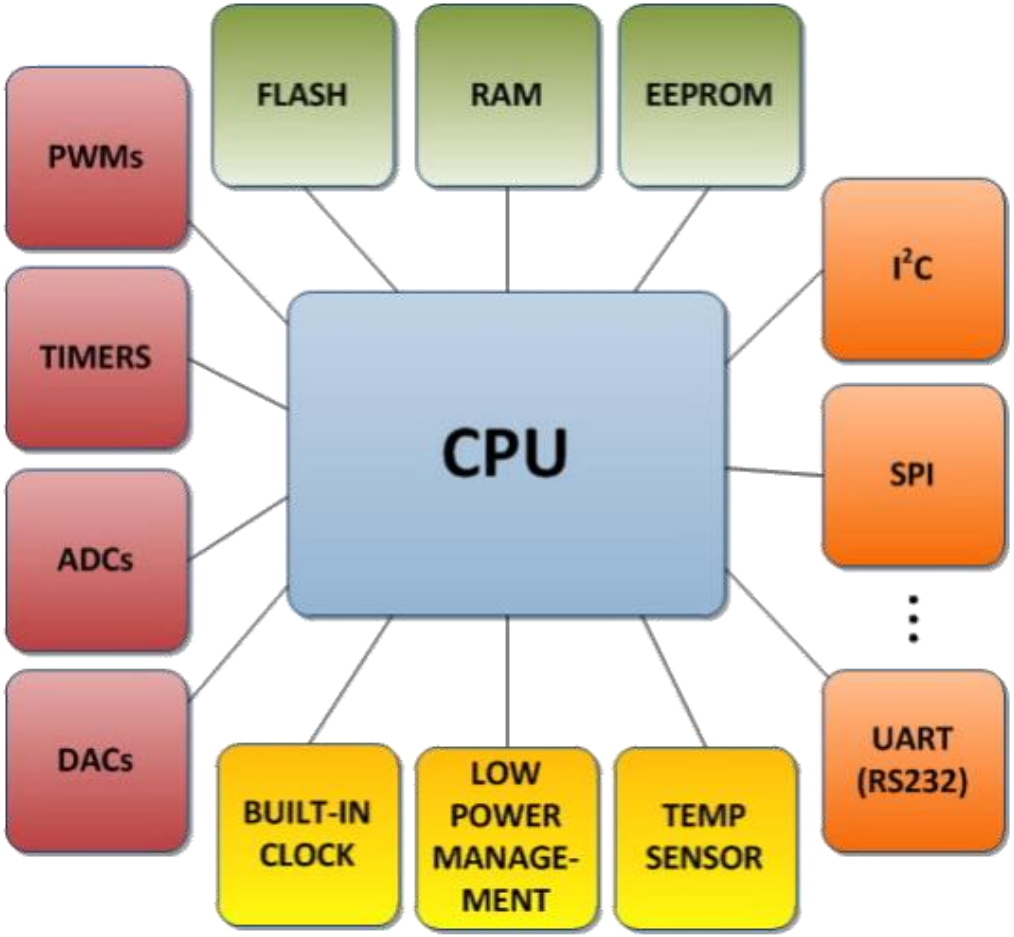
MPU

- Um microprocessador, MPU, é um CI (circuito integrado) composto por uma unidade de controle, uma ALU (Arithmetic Logic Unit) e memória (registradores e cache).

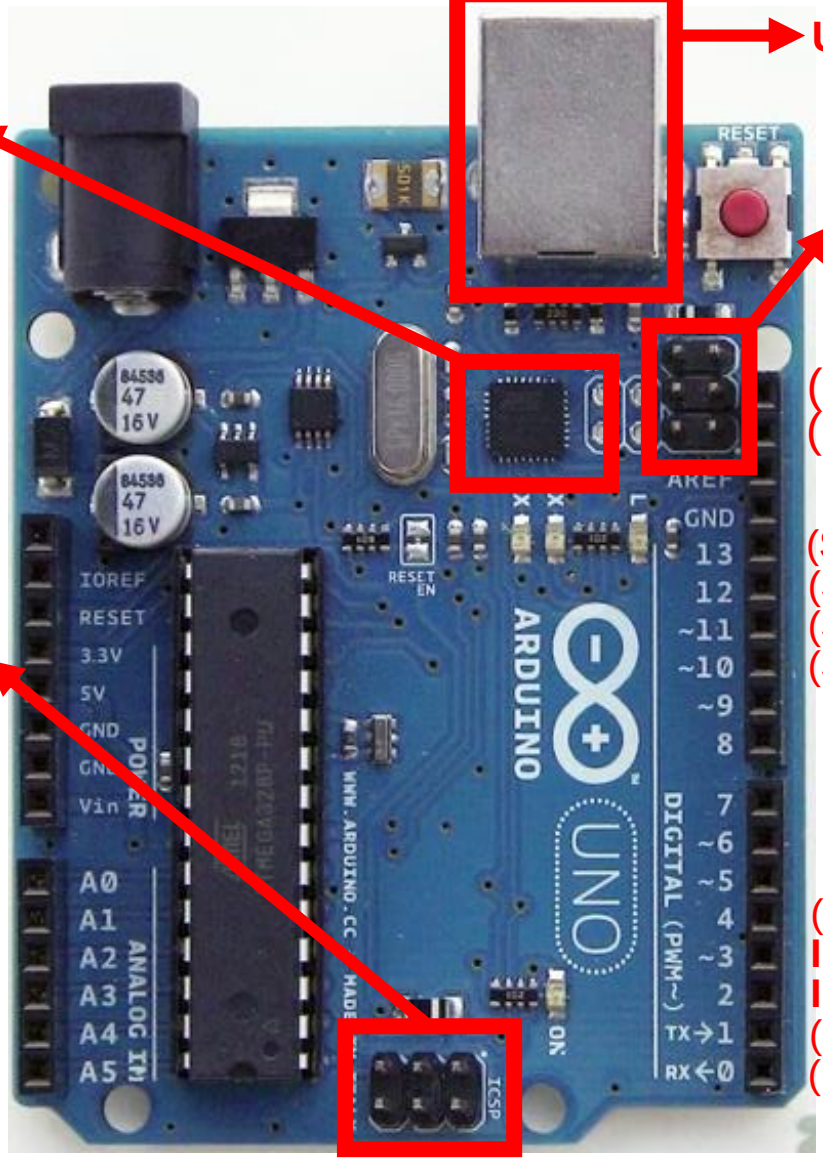


MCU

- Um microcrontolador, MCU, é um SoC (System-on-a-Chip), pois é um CI (circuito integrado) que contém uma CPU, memória, entradas e saídas, protocolos de comunicação, conversores analógico digital e digital analógico, timers e PWMs.



Arduino – Exemplos de implementação de protocolos



Interfaces

- Interfaces são maneiras pre-definidas de comunicação entre eletrônicos.
- As interfaces são utilizadas porque os microcontroladores possuem um número limitado de entradas e saídas, sendo que as interfaces usam poucas entradas e saídas. Além disso, a maioria dos dispositivos como sensores, controladores EEPRON requerem o uso de uma interface de comunicação.
- Os protocolos podem ser implementados de maneira serial ou paralela. Os dados em serial são enviados um de cada vez pelo mesmo fio. Ao contrario, os dados em paralelo são enviados por vários fios, de maneira simultânea. Entretanto, a comunicação entre o HD e a placa mãe no seu PC é serial. Porque?
- Exemplos: UART, I2C, RS232, SPI, 1WIRE, USB 2.0, CANBUS.

Interfaces

Algumas propriedades destas interfaces são:

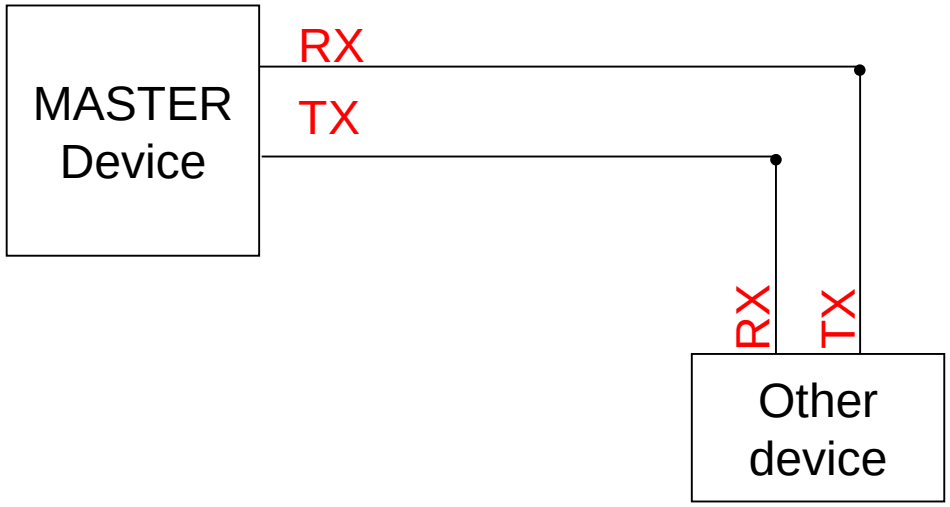
- Taxa de comunicação: é usualmente representada em bps (bits por segundo). Chamada de clock em comunicação síncronas e de baud rate em assíncronas;
- Método:
  - Síncrono -> depende de um sinal e clock;
  - Assíncrono -> uso de parâmetros como baud rate para diminuir erro e start e stop bits;
- Sentido de transmissão:
  - Full-duplex -> envia e recebe ao mesmo tempo;
  - Half-duplex -> ou envia ou recebe, mas não simultaneamente;
  - Simplex -> apenas envia ou apenas recebe;

Interfaces mais comuns

- I^2C : Inter-Integrated Circuit;
- SPI: Serial peripheral interface/interchange;
- RS-232: Recommended Standard 232 (conhecida por serial);
- UART: Universal asynchronous receiver/transmitter;
- 1Wire;
- USB: Universal Serial Bus.

Protocolo	(bps) Taxa Comunicação	Sentido	Método	N fios	Voltagem	Máximo de dispositivos
UART	1200 a 115200	Full-duplex	Assíncrono	2	0 a 5V	1
SPI	0 a 10M	Full-duplex	Síncrono	3+ 1 para cada Slave	0 a 5V	-
I^2C	100k a 400k	Half-duplex	Síncrono	2	0 a 5V	127 a 1024
RS 232	1200 a 115200	Full-duplex	Síncrono ou Assíncrono	2	-15 a 25V	1
1 Wire	0 a 16,3k ou 0 a142k	Half-duplex	Assíncrono	1	2,8 a 6,0V	256
USB	1,5M a 4,8G	Half-duplex ou Full-duplex	Assíncrono	2	0 a 20V	127

Interfaces: RS232

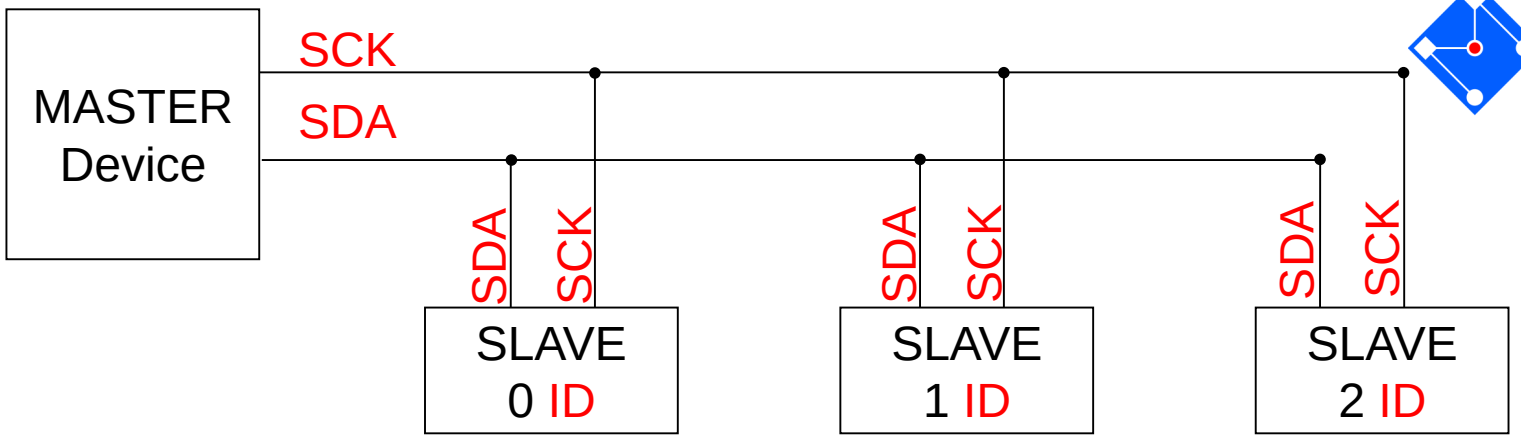


- RX – Receive line, TX – Transmit line

1. Assynchronous – no clock

Interfaces: I²C

- Colocar resistor de pullup de 4,7kΩ
- SCK – Serial Clock e SDA – Serial Data



- I²C : Inter-Integrated Circuit;
- O I²C pode colocar em comunicação diversos dispositivos mestre e diversos dispositivos escravos. Os mestres não podem se comunicar via I²C;
- A comunicação é síncrona e half-duplex;
- Precisa apenas de 2 fios: um de dados (SDA) e um de clock (SCL);
- ID de cada dispositivo é dados por um endereço de um byte;
- Incorpora bits de Acknowledge para verificar se o escravo está *ouvindo*.

Interfaces: I²C

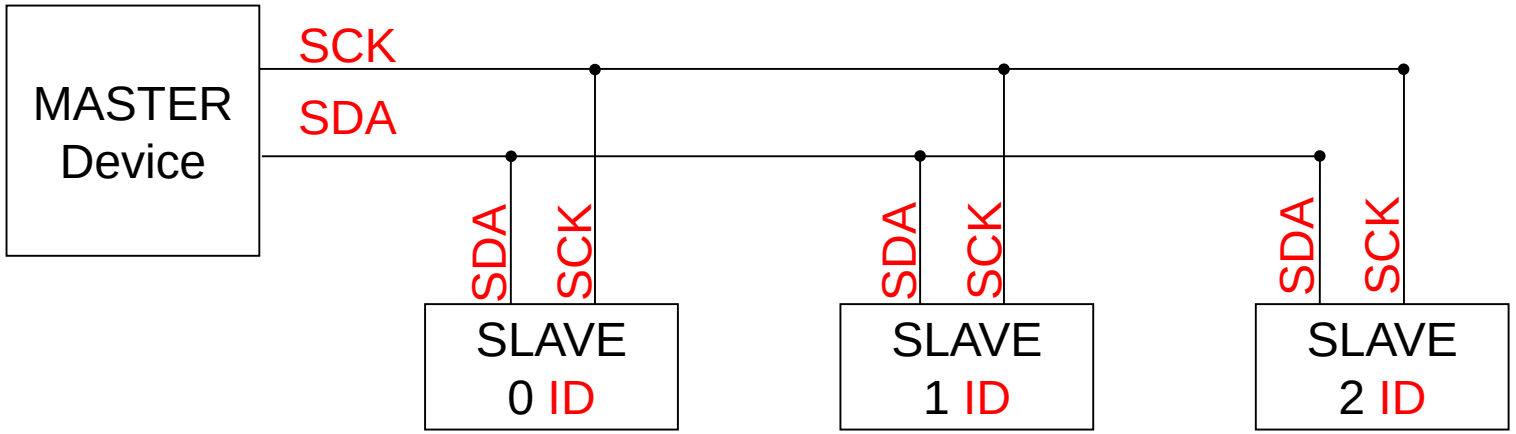
- Está implementado no Arduino com a biblioteca WIRE:
  - Começo: wire.begin();
  - Começo de transmissão: wire.beginTransmission(0x68) – procura escravo com ID 0x68;
  - Escrever: wire.write(0x6b) -> escreve 0x6B na linha;
  - Requisitar: wire.requestFrom(0x68, 6, true); -> pede 6 bytes do escravo com ID 0x68. Se o ultimo argumento é falso, o Arduino segura a linha I²C impedindo comunicações;
  - Ler: wire.read() -> le um byte requisitado pelo wire.requestFrom();
  - Fim: wire.endTransmission(true).

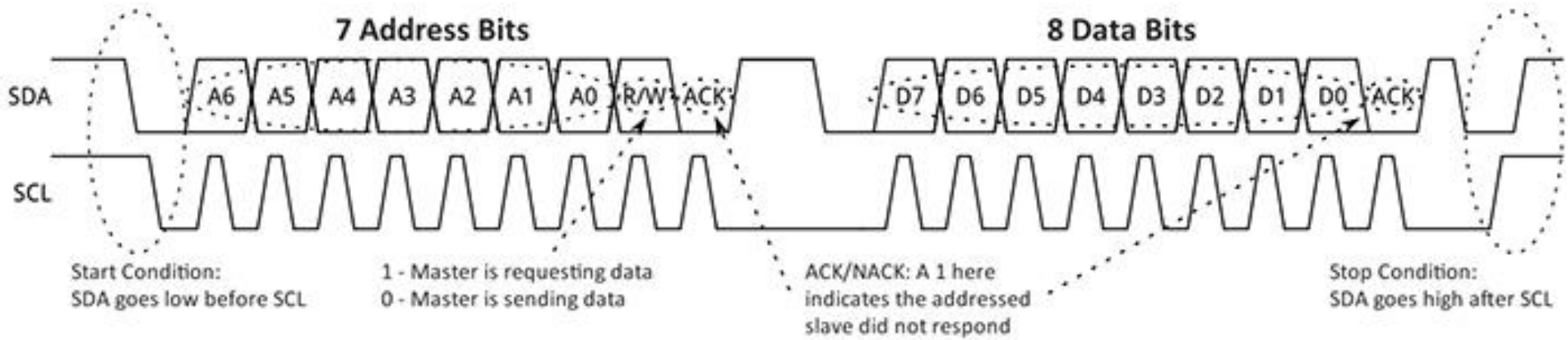
- Segue esta sequência:

1) O Dispositivo mestre escreve o comando de ID do dispositivo no BUS para identificar. O escravo responde que está pronto;
2) Dispositivo mestre envia comando;
3) Dispositivo mestre envia detalhe (endereço de memória);
4) Dispositivo escravo envia dado ou grava dado no end. de memória;

Interfaces: I²C

- Colocar resistor de pullup de 4,7kΩ
- SCK – Serial Clock e SDA – Serial Data





Interfaces: I²C

Exercício:
I²C Scanner

OLED usando biblioteca U8glib.h com o modelo SSD1306_128x64 e comunicação I²C.

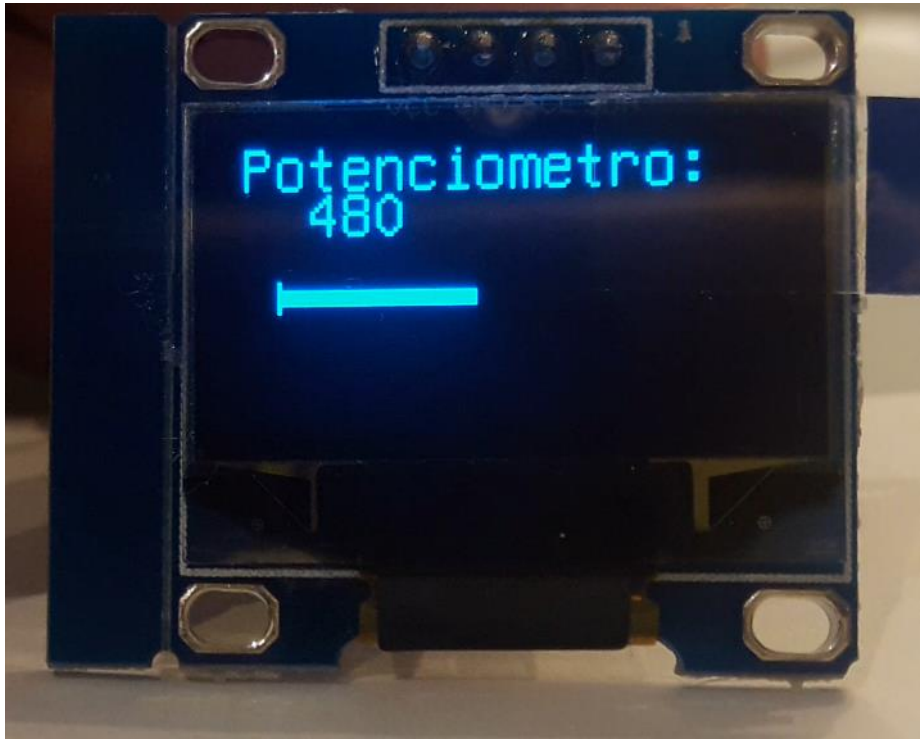
```
#include <Wire.h>
void setup(){
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);             // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");
}
void loop(){
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ ) {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
    if (error == 0){
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");
      nDevices++;
    }
    else if (error==4) {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");
  delay(5000);           // wait 5 seconds for next scan
}
```

Interfaces: I²C

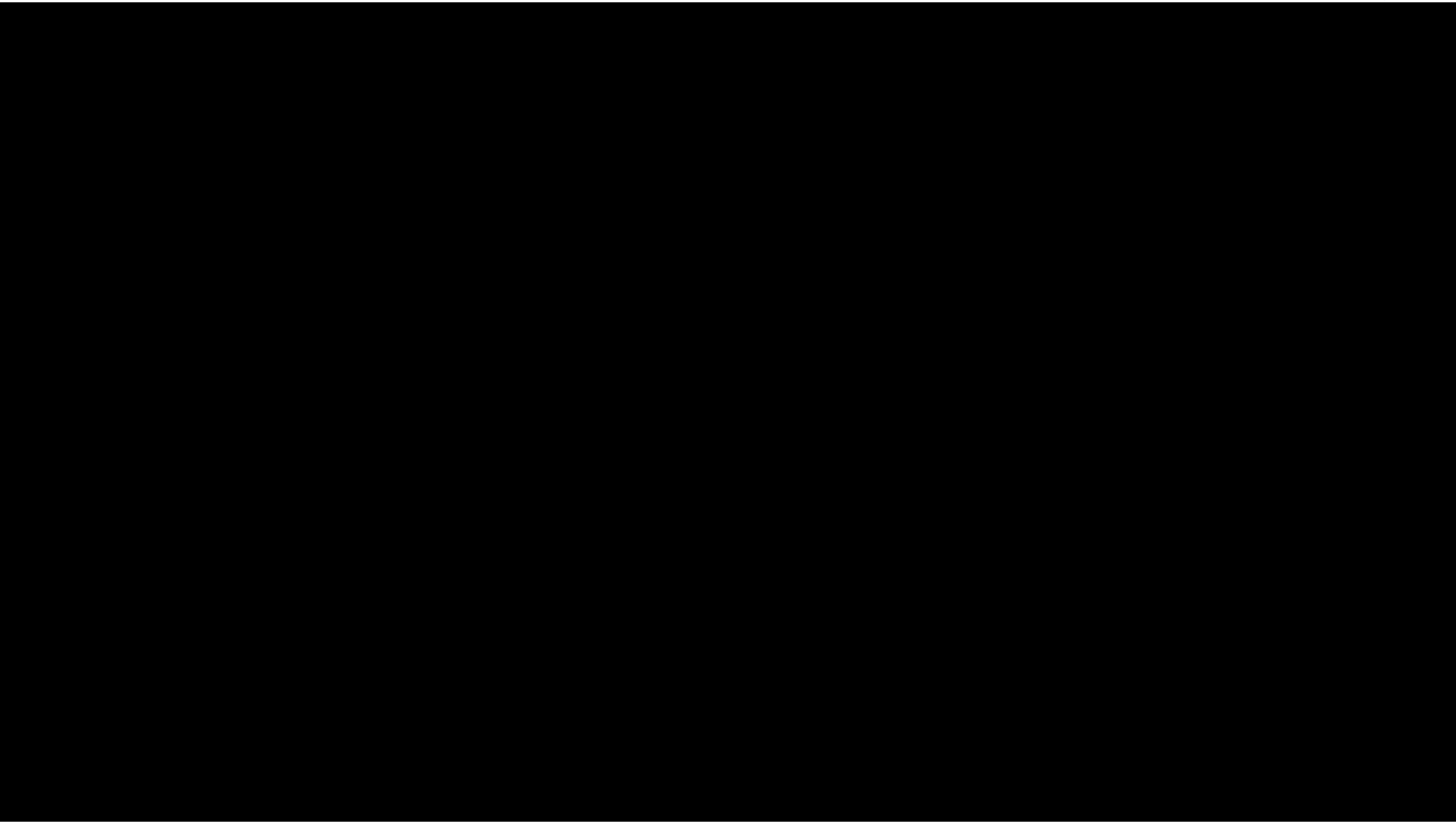
Exercício:
OLED, biblioteca U8glib.h, SSD1306_128x64 I²C

Faça um programa que leia a voltagem no potenciômetro e:
1) Escreva o valor no OLED, tanto no nível lógico (0 a 1023) quanto na voltagem (0 a 5V);
2) Tenha uma barra que varia de tamanho de acordo com o valor do potenciômetro.

Lembrar de verificar os comandos para configuração e os de dados, tentando compreender todos os passos no uso do I²C.



Sensor capacitivo: Acelerometro + I^2C



Interfaces: I²C

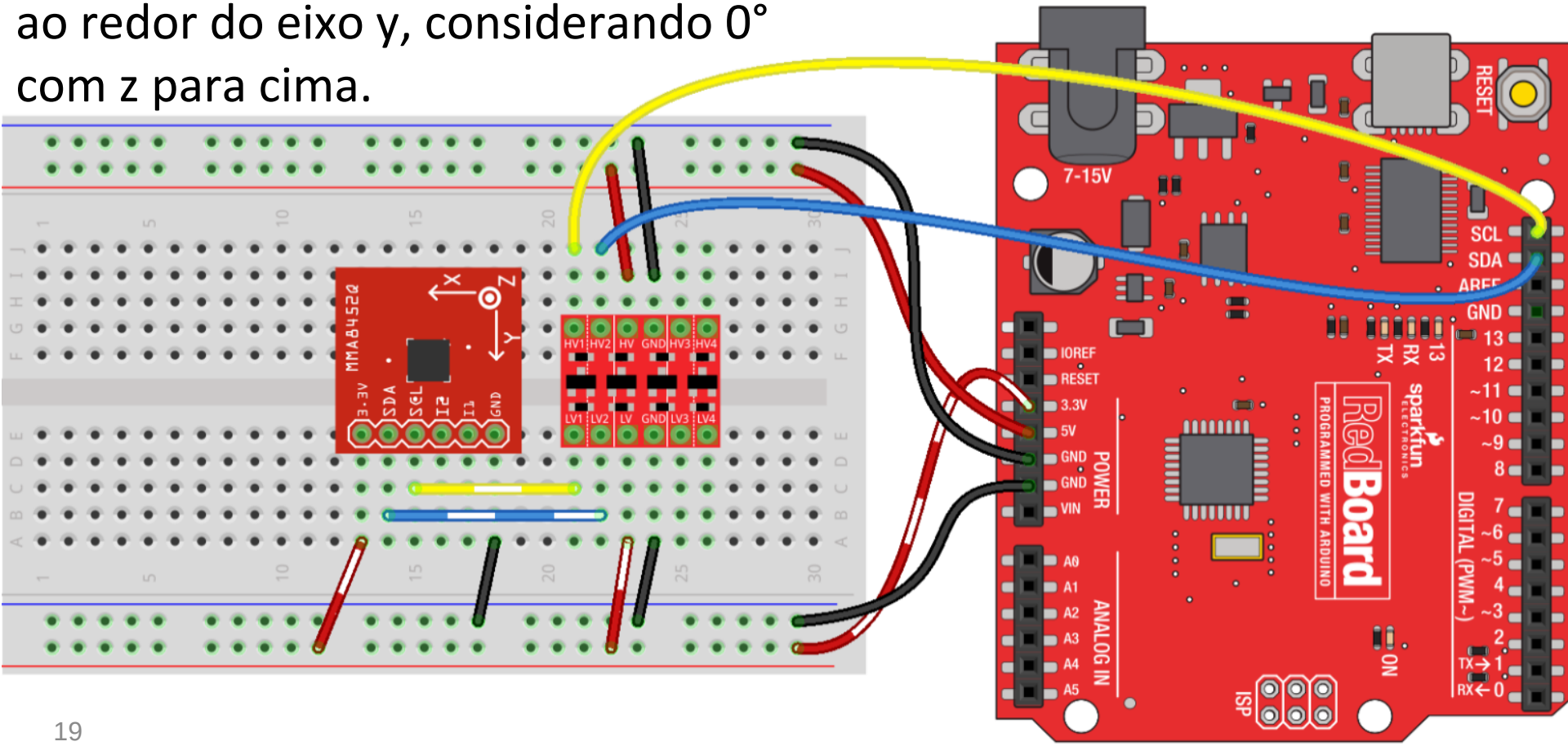
Exercício:

Acelerômetro MMA8452Q, com level shifter

Primeiro descubra qual o ID do acelerômetro utilizando o I²C Scanner.

Faça um programa que calcula a inclinação do acelerômetro girando-o ao redor do eixo y, considerando 0°

com z para cima.

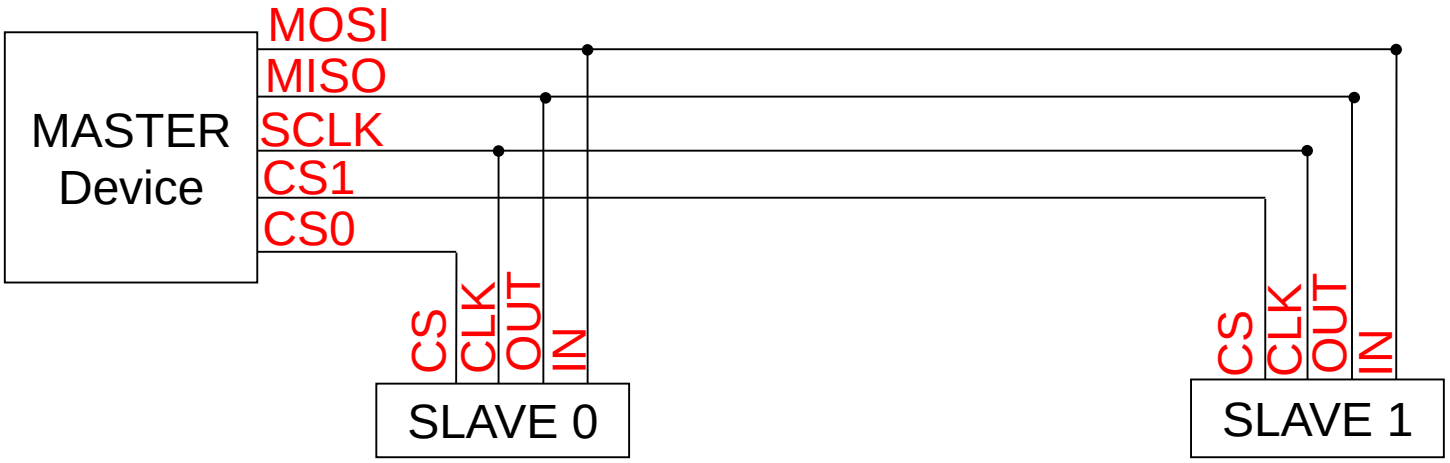


Interfaces: I²C

Exercício:
OLED, biblioteca U8glib.h, SSD1306_128x64 I²C
Acelerômetro MMA8452Q, com level shifter

Faça um programa que leia a inclinação do acelerômetro girando-o ao redor do eixo y, considerando 0° com z para cima e tanto escreva o valor no OLED, quanto gire uma barra em L que representa o acelerômetro, com a maior parte do L sendo a placa com circuito e a menor parte do L sendo os pinos em que são ligados os cabos.
Lembrar de verificar os comandos para configuração e os de dados, tentando compreender todos os passos no uso do I²C.

Interfaces: SPI



- SPI : Serial peripheral interface/interchange;
- O SPI pode colocar em comunicação um dispositivos mestre e diversos dispositivos escravos;
- A comunicação é síncrona e full-duplex;
- Precisa apenas de 3 fios mais um por escravo: um de clock (SCLK), dois de dados (MOSI e MISO) e um por escravo para seleção (SS ou CS);
- Geralmente é implementada com um shift register. Desta forma, quando um bit é recebido, os bits anteriores são movidos em uma posição e o bit mais antigo é enviado de volta pela SPI.

Interfaces: SPI

- Existem 4 modos de operação de SPI, variando tanto o começo da captura de dados e quanto momento da captura de dados de acordo com bordas de subida ou descida do clock.
- Sendo CPOL o valor da voltagem de clock em um momento e CPHA a borda de subida ou descida, a tabela explica os quatro modos de funcionamento do SPI.

	CPHA = 0 ("first edge" of clock signal)	CPHA = 1 ("second edge" of clock signal)
CPOL = 0 (idle 0)	**"SPI mode 0"** •Data capture começa na primeira borda de subida. •Data capture ocorre na borda de subida;	**"SPI mode 1"** •Data capture começa na primeira borda de subida. •Data capture ocorre na borda de descida.
CPOL = 1 (idle 1)	•**"SPI mode 2"** •Data capture começa na primeira borda de descida. •Data capture ocorre na borda de subida.	•**"SPI mode 3"** •Data capture começa na primeira borda de descida. •Data capture começa na primeira borda de descida

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

Interfaces: SPI

- Está implementado no Arduino com a biblioteca SPI:
  - Constructor: SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0)) -> define 14MHz, bit mais significativo primeiro e SPI Modo 0;
  - Escrever: digital.write(10,LOW) -> Seleciona o escravo que está no GPIO 10;
  - Transferir: SPI.transfer(0x00); -> envia um byte (neste caso 0) para o escravo;
  - Fim: SPI.endTransaction(); -> deve ser chamado somente depois de digitalWrite(10,HIGH).

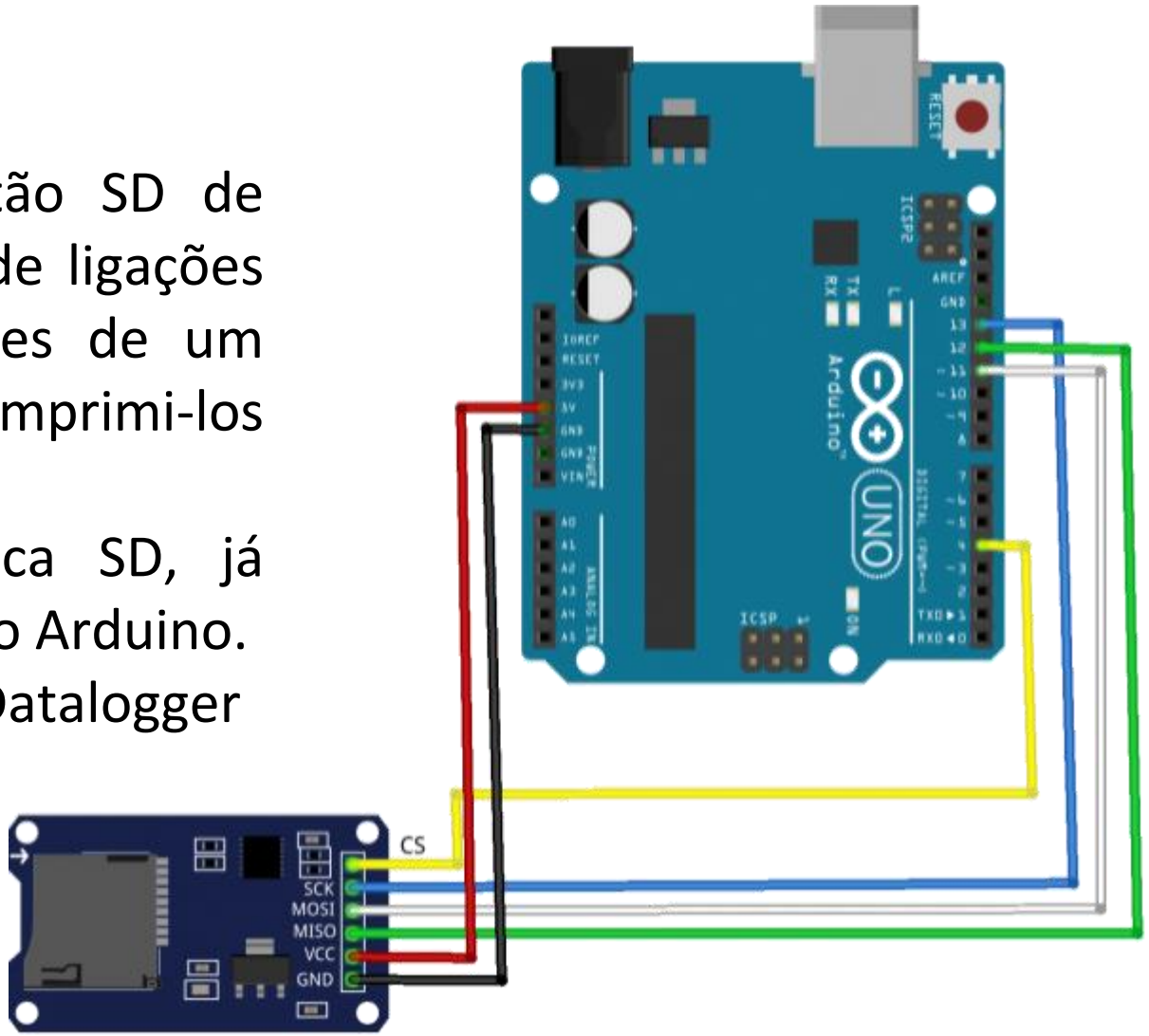
- Segue esta sequência:

1) Dispositivo mestre coloca CS **low** (depende do dispositivo) para selecionar o dispositivo desejado;
2) Dispositivo mestre manda comando através de MOSI e escravo retorna dados;
3) Dispositivo mestre manda vários 1 em MOSI e recebe valor desejado em MISO;

Interfaces: SPI

Exercício
Cartão SD

Use o gravador de cartão SD de acordo com o esquema de ligações abaixo e grave os valores de um potenciômetro, além de imprimi-los no Serial da IDE.
Obs.: Utilizar a biblioteca SD, já disponível na instalação do Arduino.
File -> Exemples -> SD -> Datalogger



Interfaces: SPI

Exercício:

Cartão SD

OLED, biblioteca U8glib.h, SSD1306_128x64 I^2C

Acelerômetro MMA8452Q, com level shifter

Faça um programa que leia a a inclinação do acelerômetro girando-o ao redor do eixo y, considerando 0° com z para cima e tanto escreva o valor no OLED, quanto gire uma barra em L que representa o acelerômetro, com a maior parte do L sendo a placa com circuito e a menor parte do L sendo os pinos em que são ligados os cabos E QUE ESCREVA O VALOR DA INCLINAÇÃO TANTO NO SD QUANTO NA SERIAL DA IDE ARDUINO.