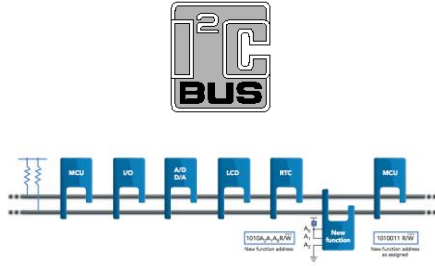


## Zbernica I<sup>2</sup>C



1

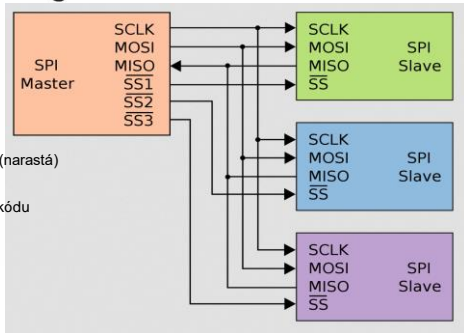
## RS-232 nie je zbernica What's Wrong with Serial Ports?

2

## Zbernica SPI

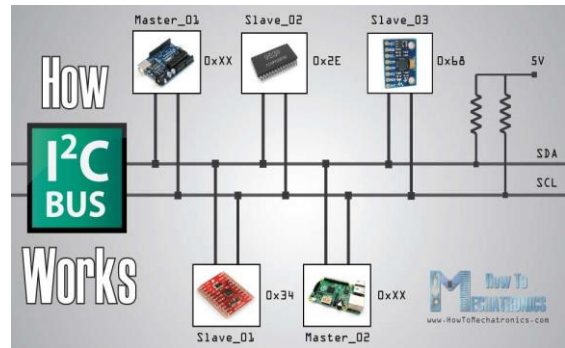
### What's Wrong with SPI?

- > Počet vodičov (narastá)
- > Single master
- > Malá ochrana kódu



3

## Zbernica I<sup>2</sup>C multimaster



4

## Porovnanie

| Peripheral          | Synchronous  |  | Asynchronous  |
|---------------------|--|--|---|
|                     | SPI  | I <sup>2</sup> C   | UART  |
| Max Bit Rate        | 10Mbit/s   | 1Mbit/s  | 500kbit/s   |
| Max Bus Size        | Limited by no. of pins   | 128 devices  | Point to point (RS232)<br>256 devices (RS485)                                   |
| Number of pins      | 3 + n x CS   | 2  | 2   |
| Pros                | Simple, low cost, high speed   | Small pin count, allows multiple masters   | Longer distance, improved noise immunity (requires transceivers)                |
| Cons                | Single master, short distance  | Slowest, short distance  | Requires accurate clock frequency   |
| Typical Application | Direct connection to ASICs and other peripherals on same PCB   | Bus connection with peripherals on same PCB  | Interface with terminals, personal computers and other data acquisition systems |
| Examples            | Serial EEPROMs (25CXXX series), MCP320X A/D converter, ENC28J60 Ethernet controller, MCP251X CAN controller... | Serial EEPROMs (24CXXX series), MCP98XX temperature sensors, MCP322x A/D converters... | RS232, RS422, RS485, LIN bus, MCP2550 IrDA interface...                         |

5

## Zbernica I<sup>2</sup>C

- > What is the I<sup>2</sup>C Bus and what is it used for?
- > Bus characteristics
- > I<sup>2</sup>C Bus Protocol
- > Data Format
- > Typical I<sup>2</sup>C devices
- > Example device
- > Sample pseudo code

6

## Zbernica I<sup>2</sup>C (TWI)

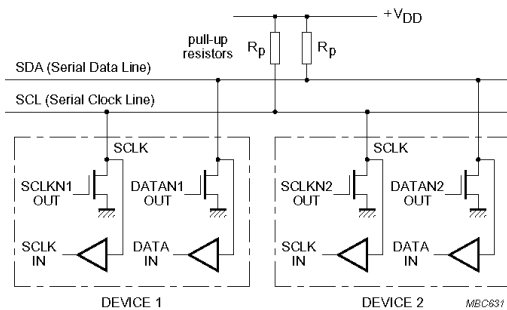


- Skratka **Inter - Integrated Circuit Bus**
- Vývoj: firma Philips, teraz NXP
- Malá sieť medzi integrovanými obvodmi
- **Pôvodne pre IO na jednej doske plošného spoja**
  - Synchronný sériový prenos dát
  - Master/Slave, Multi-master
  - Jeden vodič je pre data SDA
  - Jeden vodič je pre hodiny / clock
  - Jeden referenčný vodič GND
  - Napájanie 5V / 3,3V
- **Široká škála zariadení v súčasnosti**
  - Mikropočítače, EEPROM, RTC, LCD driver, ADC a DAC, **SENZORY**

7

7

## I<sup>2</sup>C Electrical Aspects



- I<sup>2</sup>C devices are wire ANDed together.
- If any single node writes a zero, the entire line is zero

9

9

## Zbernica I<sup>2</sup>C dôležité pojmy

- MASTER** – zariadenie, ktoré inicializuje (zahajuje) prenos na zbernici, generuje hodinové signály a ukončuje prenos. MASTER môže byť vysielateľ aj prijímač.
- SLAVE** – zariadenie, ktoré je oslovené (adresované) mastrom. Odpovedá, len ak je oslovené. SLAVE môže byť aj vysielateľ aj prijímač.
- Multi-master** – schopnosť súčasnej bezkolíznej koexistencie viacerých mastrov na zbernici
- Transmitter** – zariadenie vysielajúce dáta na zbernici.
- Receiver** – zariadenie prijímajúce dáta zo zbernice.
- Synchronizácia** – zabezpečenie zhody signálov medzi dvoma zariadeniami – zabezpečuje CLK, generuje Master.
- Arbitrácia** – vlastnosť, ktorá zabezpečí, že v danom čase len ovláda zbernici len jedno zariadenie (ACK/NACK, kontrola stavu H/L)

8

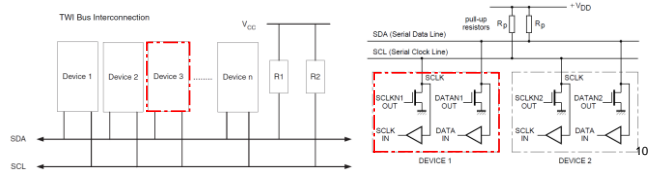
8

## I<sup>2</sup>C (TWI)

### Inter Integrated Circuit

- Počet obvodov pripojených na zbernici je obmedzený počtom
- adres a celkovou kapacitou zbernice <400 pF (jednotkou dĺžky je pF<sup>2</sup>).
- Na prenos informácie sú použité dve nesymetrické vedenia
  - SCL (hodiny)
  - SDA (data/adresa)

Tieto vodiče sú obojsmerné a pomocou PULLED UP rezistorov ťahané hore. Všetky zariadenia pripojené na zbernici musia mať „otvorený kolektor“ (alebo ...). Budiče zbernice majú implementované „drôtové AND“.



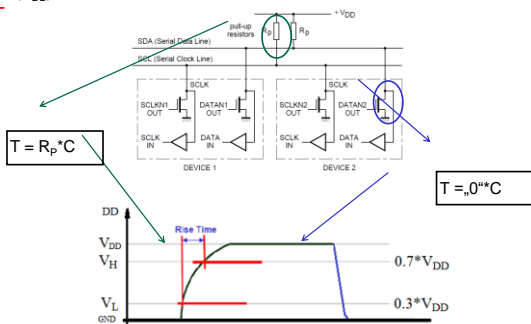
9

10

## I<sup>2</sup>C

Zdroj:  
Log. 1 } = f(V<sub>DD</sub>)  
Log. 0 }

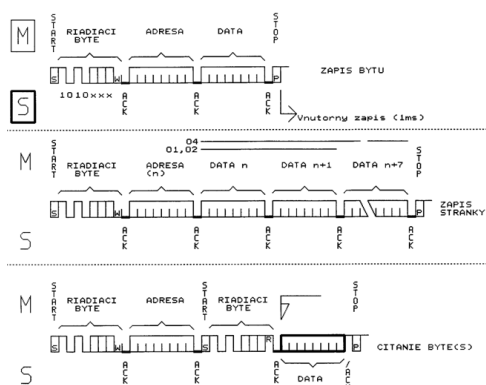
Prijímač filtruje krátke impulzy na zbernici.



11

11

## I<sup>2</sup>C – pripojenie EEPROM

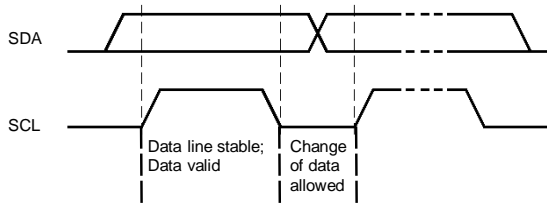


12

12

## Bit Transfer on the I<sup>2</sup>C Bus

- Počas nábežnej hrany SCL musia byť data stabilné, zmena nastáva počas SCL=0

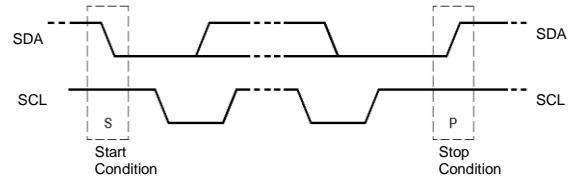


13

13

## Start and Stop Conditions

- H/H znamená, že zbernica je voľná
- Normálne musí byť D počas CLK H stabilná
- Prechod H/L na D počas CLK H je definované ako START
- Prechod L/H na D počas CLK H je definované ako STOP
- Obe podmienky generuje MASTER
- Medzi START a STOP sa zbernica považuje za obsadenú



14

14

## I<sup>2</sup>C Addressing

- Každý uzel má svoju jedinečnú 7- (10-) bitovú adresu
- Adresy začínajúce 0000 alebo 1111 majú špeciálny význam:
  - 0000000 – General Call Address
  - 0000001 – Null (CBUS) Address
  - 1111XXX – Address Extension
  - 1111111 – Address Extension – Next Bytes are the Actual Address

Adresy sú často pevné, alebo čiastočne programovateľné či nastaviteľné.

15

15

## First Byte in Data Transfer on the I<sup>2</sup>C Bus



R/W

- 0 – Slave written to by Master
- 1 – Slave read by Master

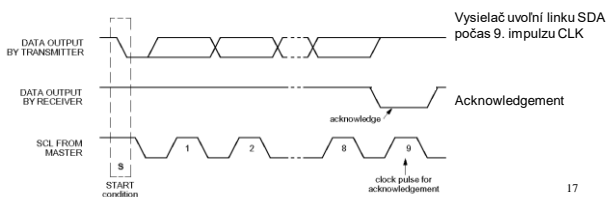
ACK – Generated by the slave whose address has been output.

16

16

## ACK – Acknowledgement

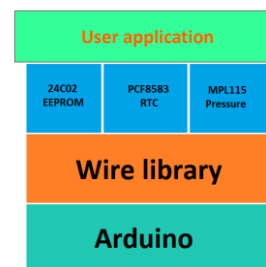
- Po prijatí každého bajtu prijímač (master/slave) stiahne SDA na 0 počas jedného hodinového impulzu na CLK
- Master receiver leaves data line high after receipt of the last byte requested NACK
- Slave receiver leaves data line high on the byte following the last byte it can accept NACK



17

17

## Arduino: Wire library



18

## Arduino: zápis

```
#include <Wire.h>

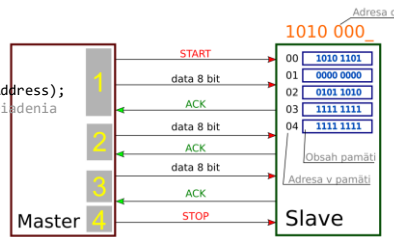
Wire.begin();

Wire.beginTransmission(DeviceAddress);
// 7-bitová adresa zariadenia

Wire.write(WriteToAddress);

Wire.write(Data);

Wire.endTransmission();
```



19

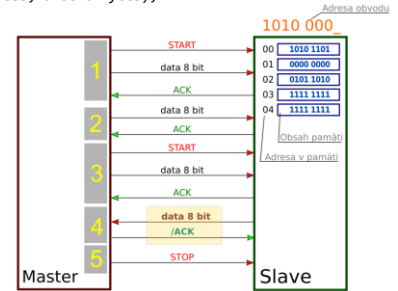
## Arduino: čítanie

```
#include <Wire.h>

Wire.begin();

Wire.requestFrom(DeviceAddress,NumberOfBytes);

While (Wire.available())
{
  char c = Wire.read();
  Serial.print(c);
}
```



20

## Arduino: čítanie zo špecifickej adresy

```
#include <Wire.h>

Wire.begin();

byte MyRead()
{
  /* STEP 1 */
  Wire.beginTransmission(DeviceAddress);
  Wire.write(MemoryAddress);
  Wire.endTransmission();

  /* STEP 2 */
  Wire.requestFrom(DeviceAddress, 1);

  if(Wire.available())
    return Wire.read();
  else
    return 0xFF;
}
```

21

## Arduino: Serial print

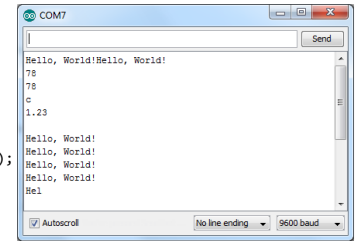
```
void setup()
{
  byte c = 78;

  Serial.begin(9600);

  Serial.print("Hello, World!");
  Serial.println("Hello, World!");

  Serial.println(78);
  Serial.println(c);
  Serial.println('c');
  Serial.println(1.23456);
  Serial.println();
}

void loop()
{
  Serial.println("Hello, World!");
}
```



22

## Arduino: Serial print

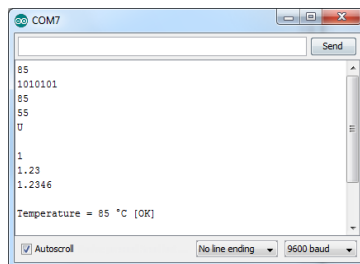
```
void setup()
{
  byte c = 85; // c = 'U';

  Serial.begin(9600);

  Serial.println(c);
  Serial.println(c, BIN);
  Serial.println(c, DEC);
  Serial.println(c, HEX);
  Serial.println((char)c);

  Serial.println(1.23456, 0);
  Serial.println(1.23456, 2);
  Serial.println(1.23456, 4);

  Serial.print("Temperature = ");
  Serial.print(c);
  Serial.print(" ");
  Serial.write(176);
  Serial.println("°C [OK]");
}
```



23