

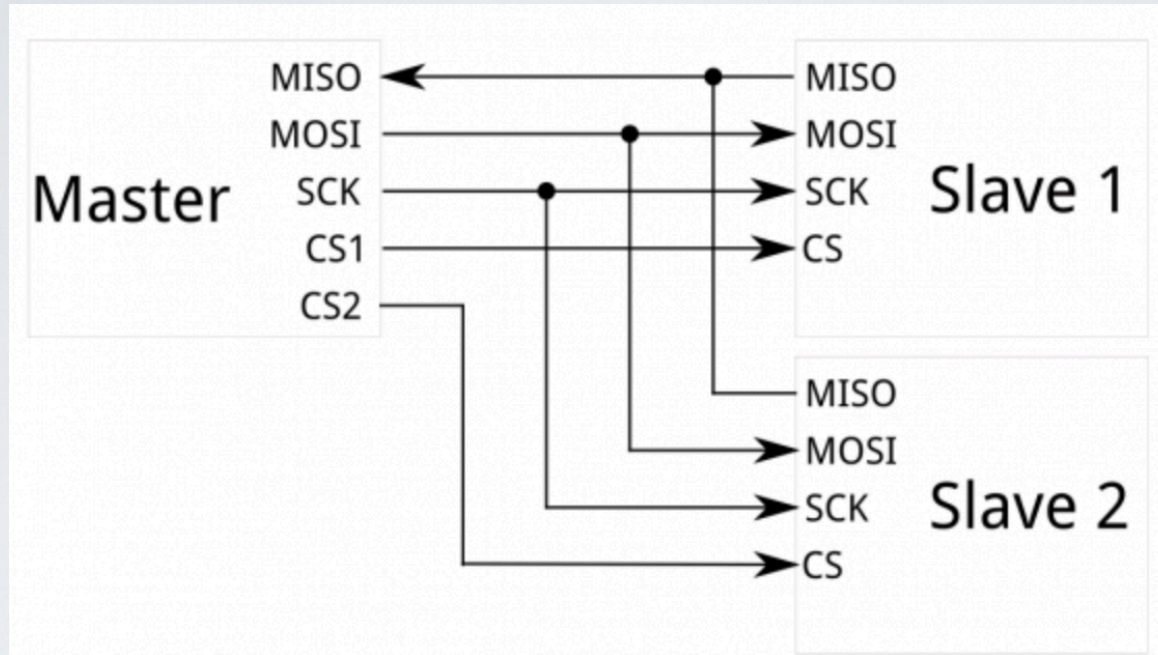


Arduino CURSUS

door Willy - w2@skynet.be , 22-sep-2017

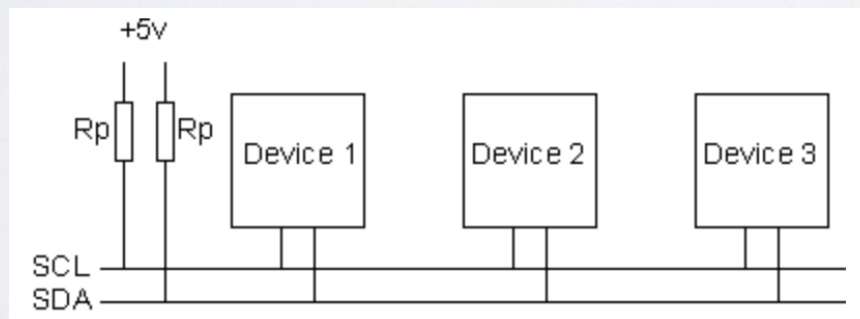
I2C of two wire protocol versus SPI

SPI



per slave een extra
chip select (CS) pin

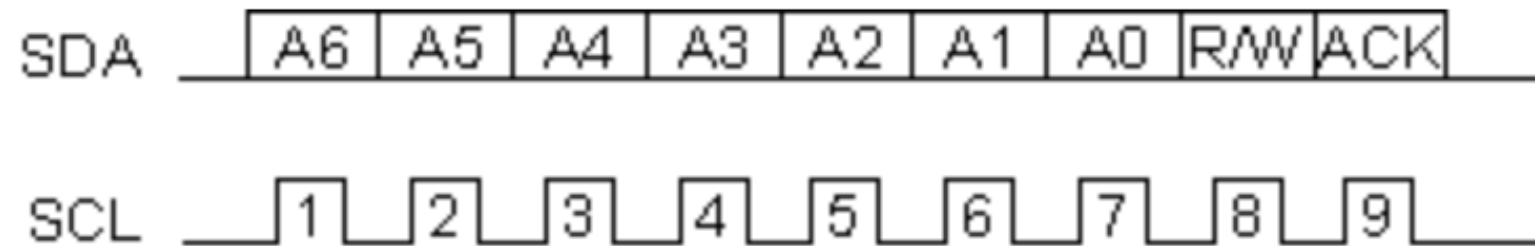
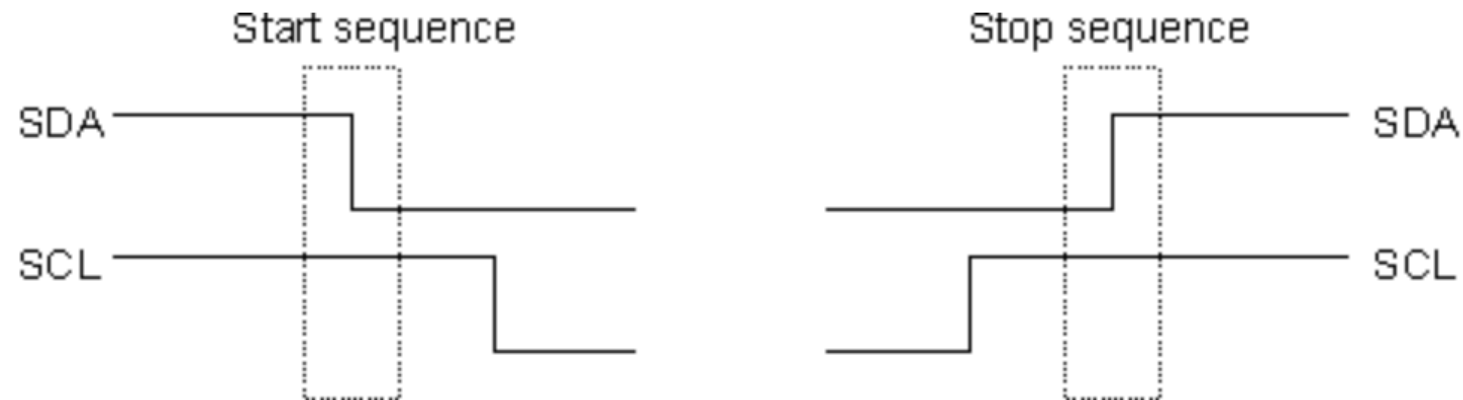
I2C



4 draden : Vcc , GND
SCL en SDA

oorsprong , historiek : Philips tv.

I2C - protocol



I2C - wat is belangrijk

- clock snelheden : 100 Khz. , pull up 5K ... 10K of
400 Khz. , pull up 2K ... 5K
- Arduino is de “master”
- ieder “slave” device heeft een uniek adres , 112 mogelijke adressen
- Arduino programma : “i2c_scanner.ino” om de bus af te tasten
- bij problemen : logic analyser , vb. SALEAE , www.saleae.com
- level shifter board , Arduino UNO = 5 Volt en sensor is 3.3 Volt
- sommige “slave’s” laten meerdere adressen toe.
- op de Arduino UNO : SDA is pin A4 , SCL is pin A5

I2C - Toepassingen

- **allerlei sensoren : temperatuur , druk**
- **display's , zowel tekst als grafisch.**
- **real time clock IC's.**
- **A/D en D/A convertors.**
- **EEPROM en RAM geheugens.**
- **enz....**

Een 1-ste voorbeeld

```
#include <Wire.h>

byte val=0 ;

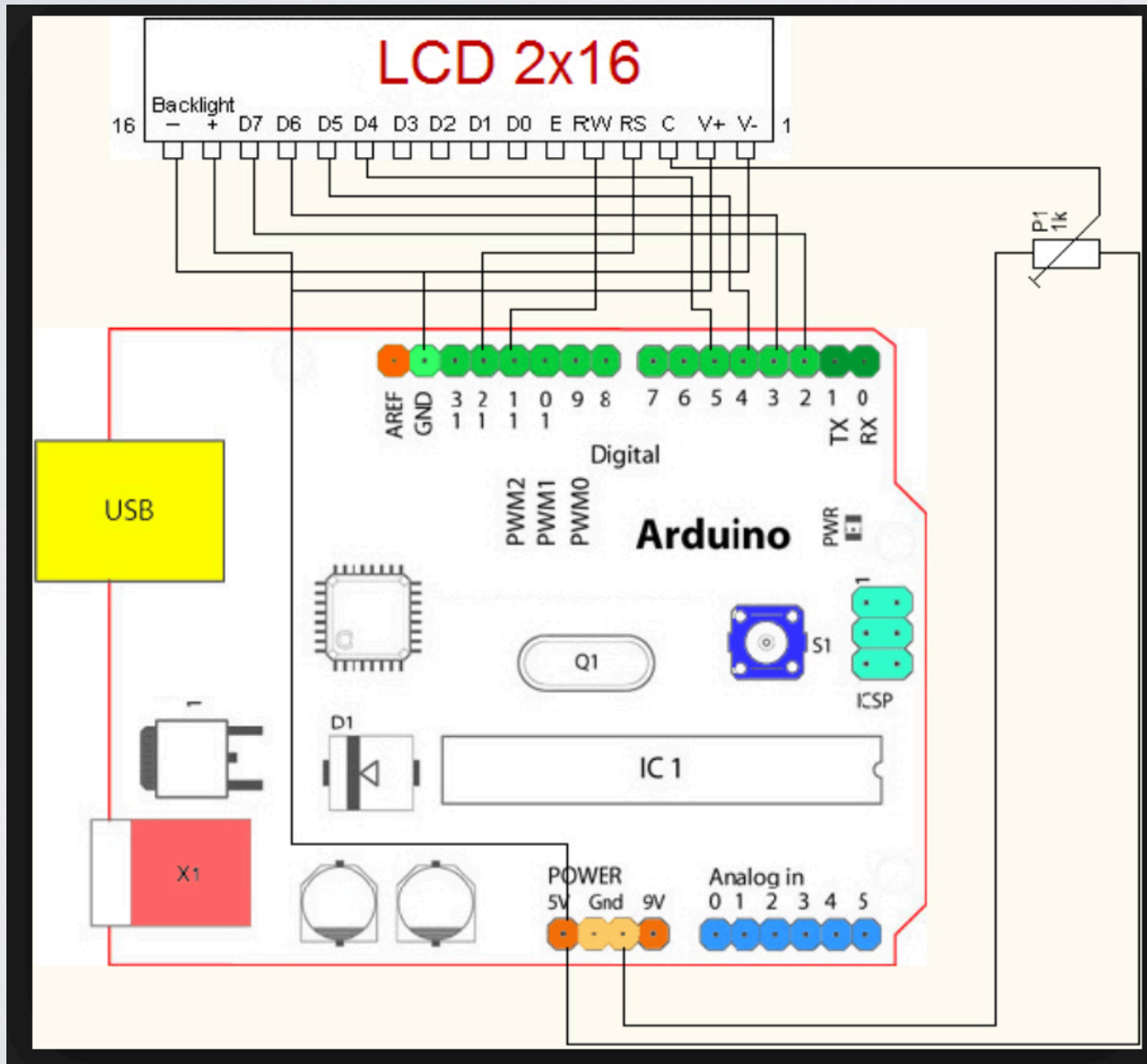
void setup() {
  Wire.begin(); // join i2c bus (address optional for master)
}

void loop() {
  Wire.beginTransmission(44) ; // transmit to device #44 (0x2c)
  Wire.write(byte(0x00)) ;      // sends instruction byte
  Wire.write(val) ;             // sends potentiometer value byte
  Wire.endTransmission() ;      // stop transmitting

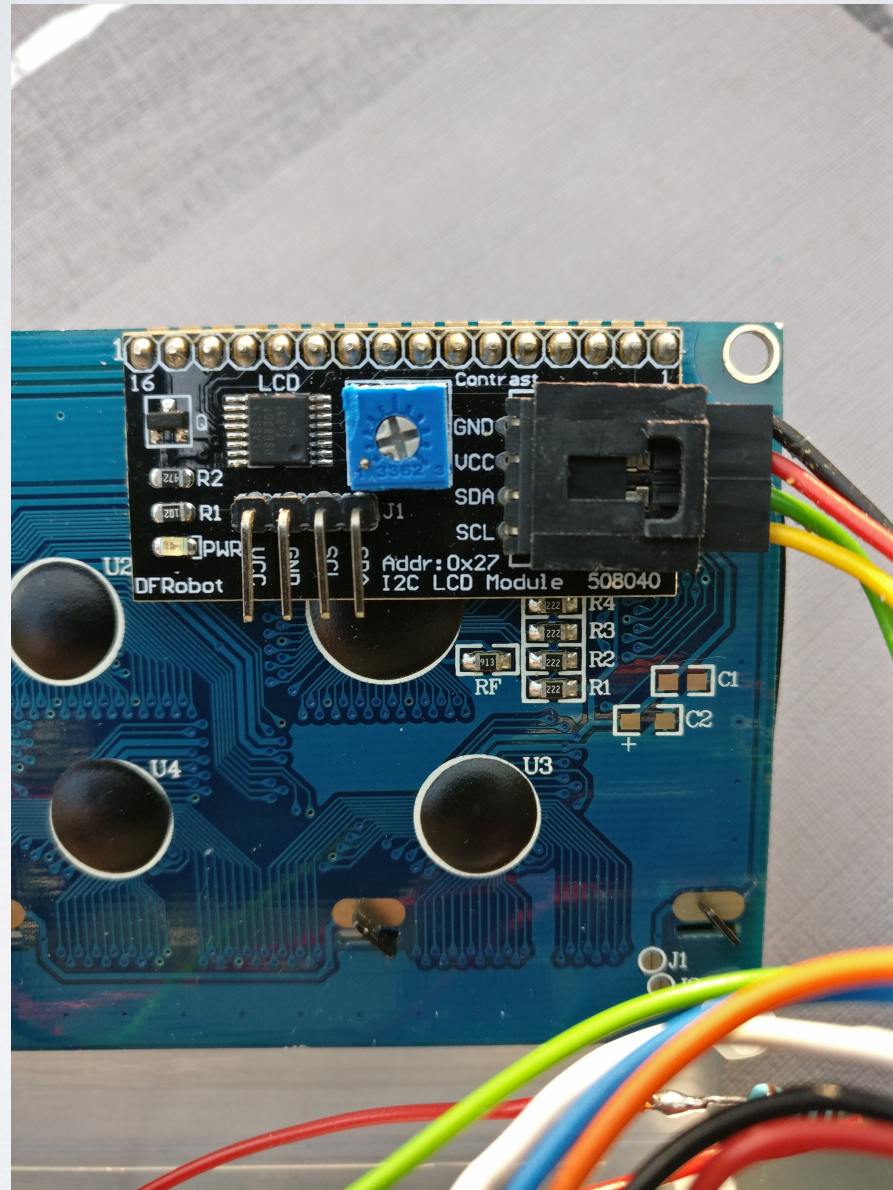
  val++ ;                      // increment value
  if (val == 64) { // if reached 64th position (max)
    val=0 ; // start over from lowest value
  }
  delay(500) ;
}
```

Voorbeeld : LCD

veel aansluitingen
nodig



LCD display met I2C



I2C - LCD - code (deel I)

```
#include <LiquidCrystal_I2C.h>

#define REV "Rev. 20-sep-2017"
#define lcd_adr 0x27 // the LCD I2C address
#define nr_lines 4
#define nr_char_line 20

LiquidCrystal_I2C lcd(lcd_adr,nr_char_line,nr_lines) ;

int counter ;

char lcd_mess[nr_char_line+1] ; // for the LCD message(s)

void setup() // 20-sep-2017
{
  lcd.init() ; // includes Wire.begin()
  lcd.backlight() ;
  welcome() ;
  counter=0 ;
}

void loop() // 20-sep-2017
{
  sprintf(lcd_mess,"counter : %d sec.",counter++) ;
  send_mess_to_lcd(3) ;
  delay(1000) ;
}
```

I2C - LCD - code (deel 2)

```
void welcome() // 20-sep-2017
```

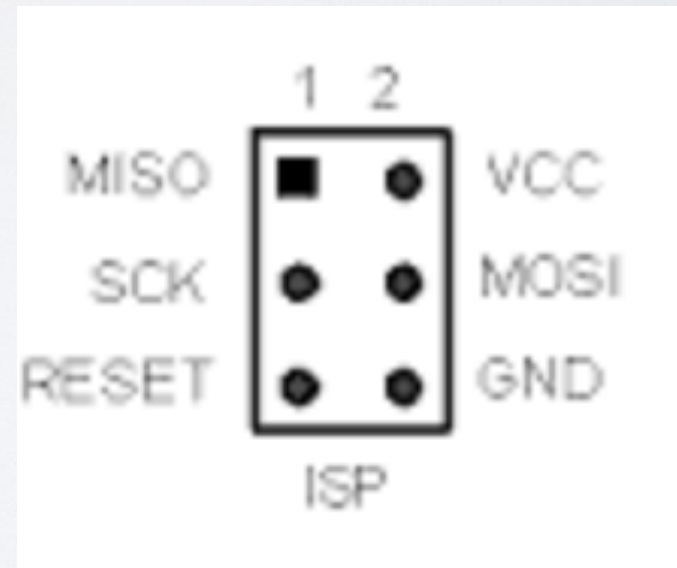
```
{  
    sprintf(lcd_mess,"LCD - I2C , demo") ;  
    send_mess_to_lcd(0) ;  
    delay(2000) ;  
}
```

```
void send_mess_to_lcd(int l) // 24-sep-2016
```

```
{  
    if ( (l < 0) || (l > nr_lines-1)) return ;  
    lcd.setCursor(0,l) ; // position at the beginning of the lcd line  
  
    if ((int)strlen(lcd_mess) > nr_char_line) lcd_mess[nr_char_line]='\0' ;  
  
    for (int n=strlen(lcd_mess) ; n < nr_char_line ; n++) strcat(lcd_mess," ") ;  
    for (int n=0 ; n < (int)strlen(lcd_mess) ; n++) lcd.print(lcd_mess[n]) ;  
}
```

Varia : Arduino - flashen

- kan op 2 manieren :
 - via de **USB poort** , dit vergt een boot-loader in de **AVR chip**
 - via de **ISP connector** , 6 pinnen , **AVR-MKII**



Besluit

- **Arduino (UNO) proberen te situeren.**
- **het is een mix van hardware (electronica) en software (C , C++).**
- **de voordracht was verre van compleet.**
- **leer het met vallen en opstaan.**
- **probeer voorbeelden van anderen te verstaan.**
- **lees de Arduino “Reference”.**
- **stel vragen : w2@skynet.be**

Wat houdt je nog tegen ?

