Arduino CURSUS

door Willy - w2@skynet.be , 09-juni-2017

OVERZICHT (1)

- historiek
- wat is een microcontroller
- het bordje : Arduino UNO
- wat basis electronica
- de programmeeromgeving (IDE).

OVERZICHT (2)

- de taal : C / C++

- oefeningen : LED laten knipperen , een drukknop inlezen enz.

- gevorderde code : serieel data versturen , I2C , SPI , A/D conversie , PWM , interrupts …

Historiek

- 2004 (2005) - Italie - oa. Massimo Banzi
- nood aan eenvoudige en betaalbare microcontroller electronica voor studenten en ze kozen voor een ATMEL AVR processor .
- zowel de hardware als de software zijn open-source.
- te programmeren via de USB poort v/e PC , Mac , Linux (zie oa. de boot-loader).

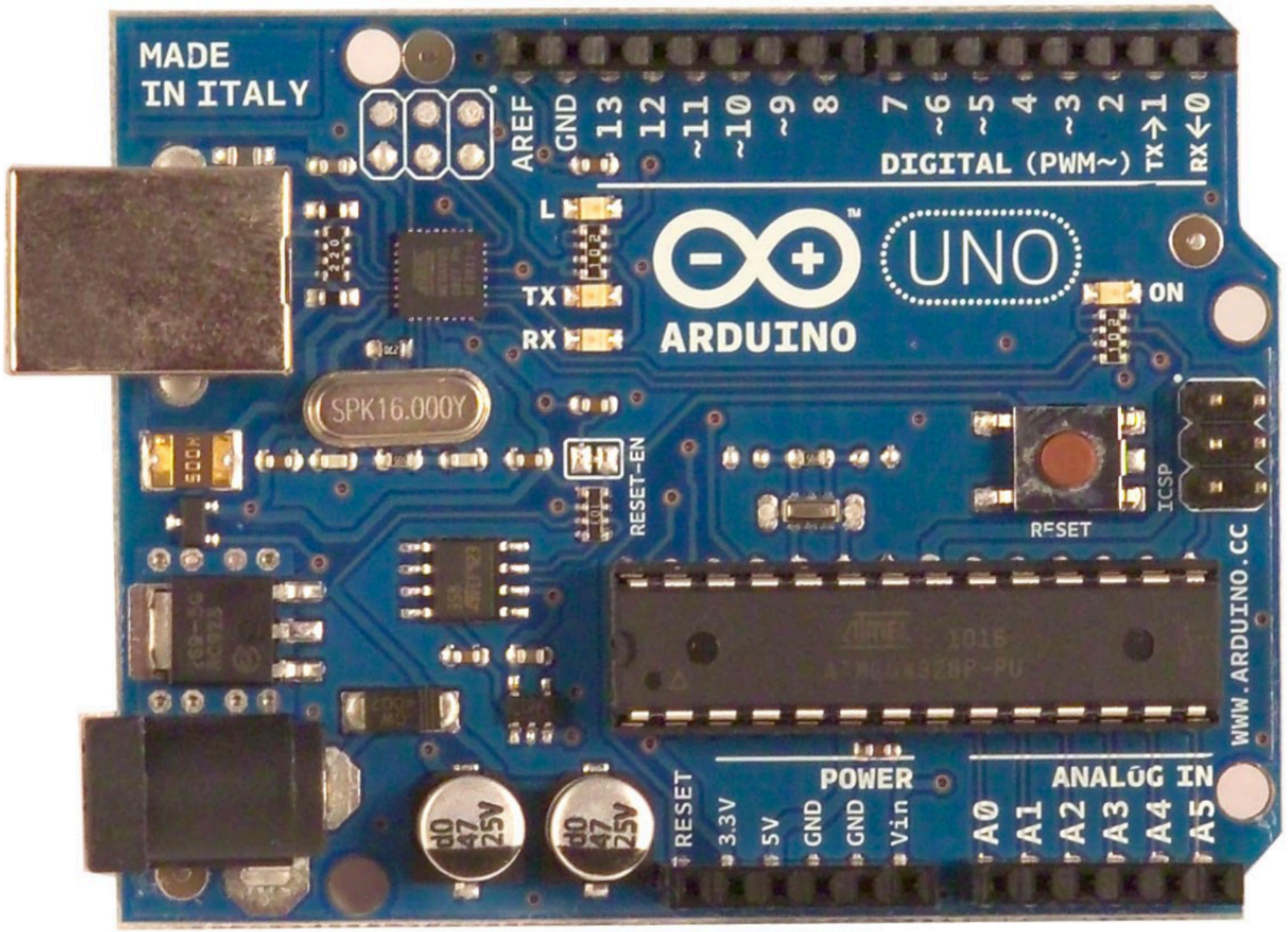
Microcontroller (1)

- klein computersysteem op 1 chip , vb. om sturingen te doen.
- veel Input/Output poorten
- ingebouwde Timers/Counters
- I/O voor A/D conversie , seriele poorten , I2C , SPI
- flash geheugen (voor het programma) en RAM

Microcontroller (2)

- in alle formaten qua kloksnelheid , opslagcapaciteit , speciale periferie zoals : ethernet , CAN bus , D/A enz…
- klein , vb. de Tiny85 van Atmel heeft slechts 8 pinnen
- low power , vb. de MSP430 reeks van Texas Instruments.
- heel krachtig : STM32 reeks met ARM processoren

Arduino UNO


MADE
IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
SPK16.000Y
RESET-EN
RESET
ICSP
WWW.ARDUINO.CC
POWER
ANALOG IN
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Specificaties - Arduino UNO bord

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Specificaties - ATmega328P chip

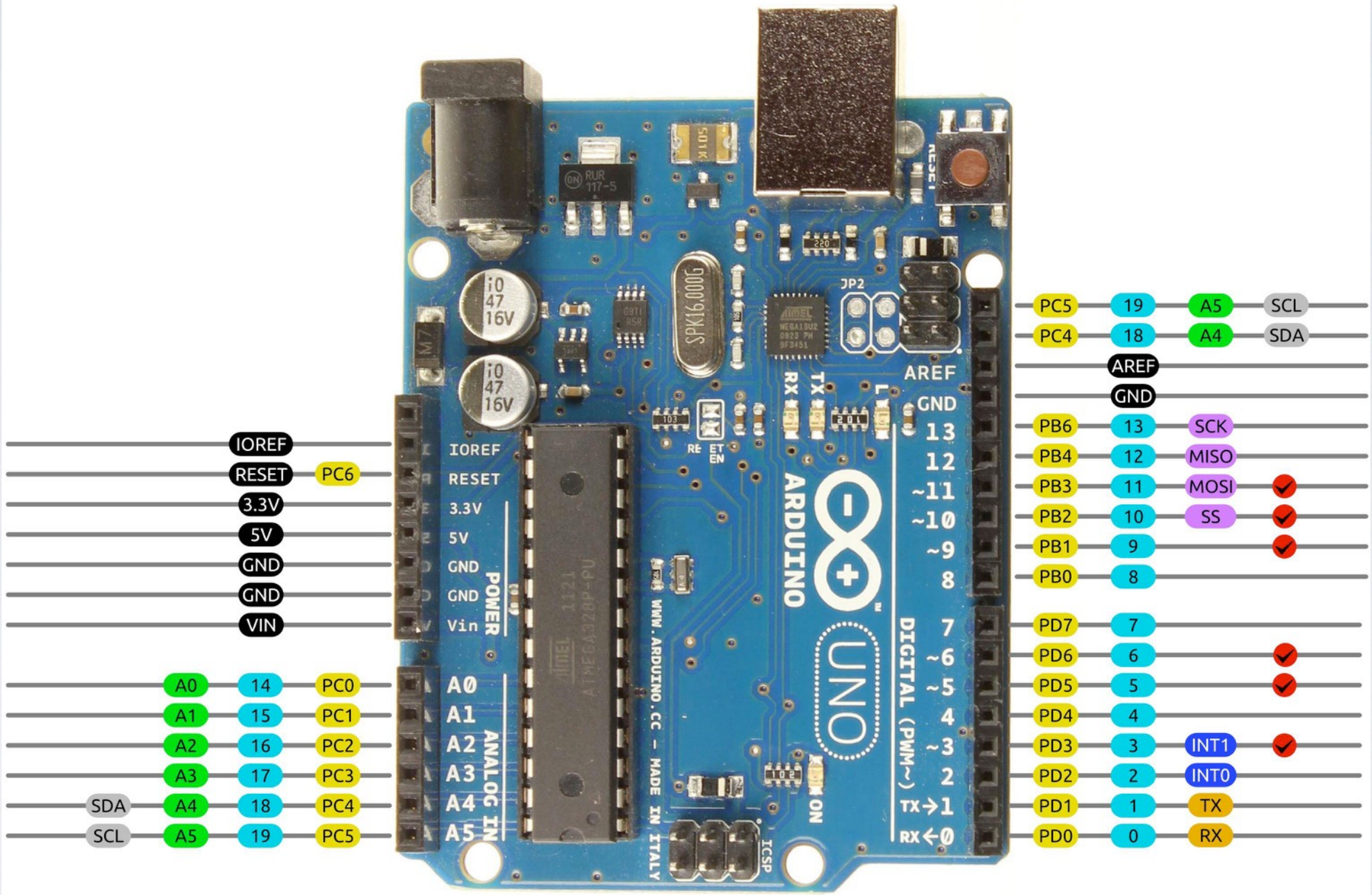
Configuration Summary

Features	ATmega328/P
Pin Count	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
General Purpose I/O Lines	23
SPI	2
TWI (I^2C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

LET wel !

- een aantal pinnen hebben een dubbele functie , vb. A4 en A5 (analoge ingangen) kunnen ook dienen als SDA en SCL voor de I2C bus.

Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

2014 by Bouni
Photo by Arduino.cc

Enkele andere ARDUINO borden

- Mega - ATMega2560 chip
- Mini - 328P chip
- Nano - 328P chip
- Zero
- Ethernet
- veel opsteekbordjes - de zogenaamde SHIELDS

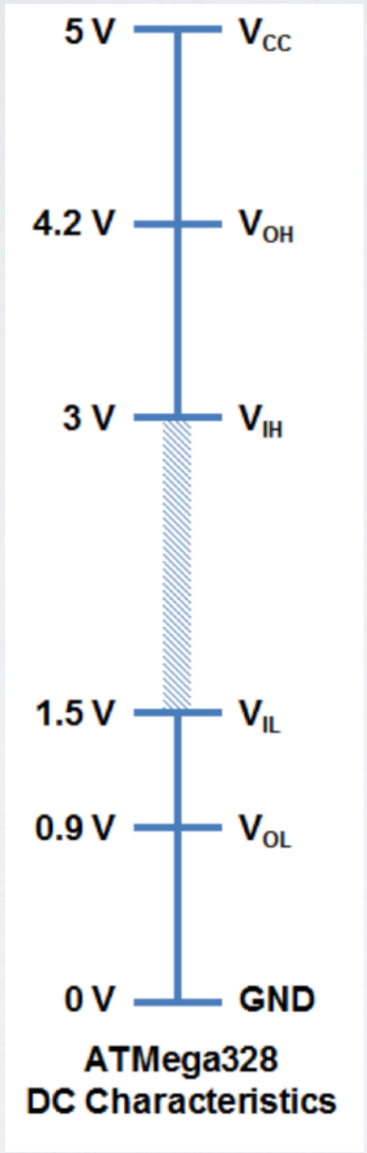
Wat basis electronica

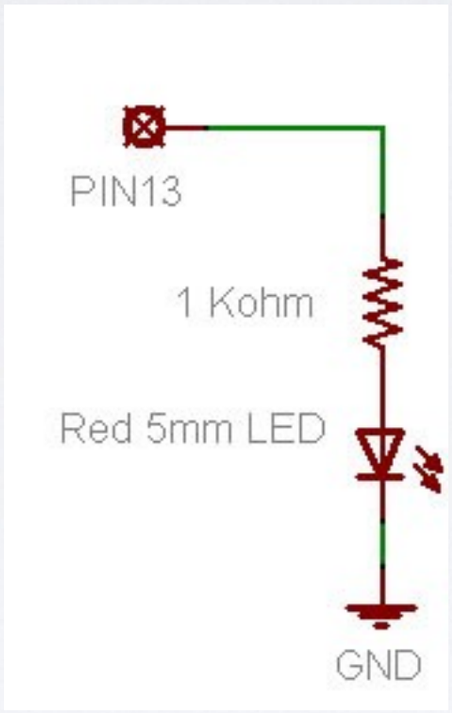
- de Arduino UNO werkt met 5 V logica , “0” = 0 tot 1,5 V , “1” = 3 tot 5 V

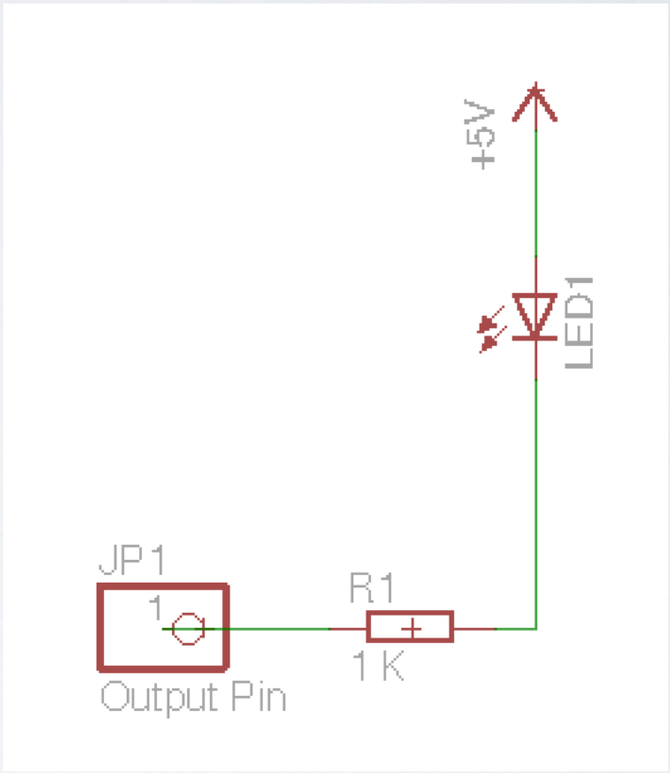
- de wet van OHM : U = I x R , met U in Volt , R in Kohm en I in mA.

- vb. rode LED via 1 K naar een logische “1” , dan loopt er ongeveer 3 mA. door de LED

Logische niveau's







enkele nuttige links

- https://www.arduino.cc , vb. om de software te downloaden voor PC , MAC of LINUX.

Vervolg

- **vrijdag 23 juni - de Arduino IDE , wat C/C++ en een eerste programma**
- vrijdag 30 juni - ????

Arduino CURSUS

door Willy - w2@skynet.be , 23-juni-2017

Arduino - IDE

Integrated Development Environment

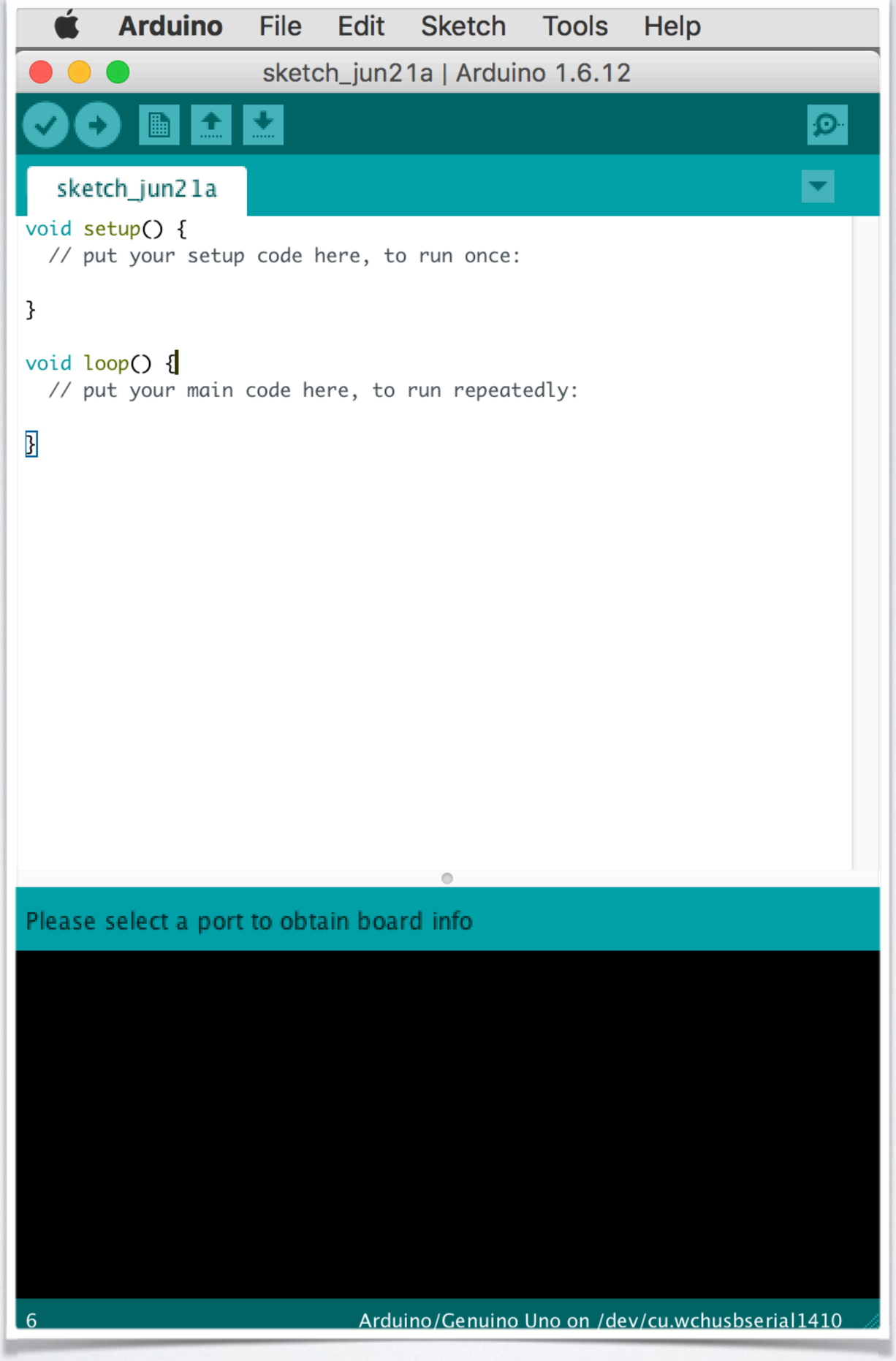
- opladen / saven van bestanden (*.ino , vroeger *.pde)

- instellingen kiezen (vb. soort Arduino bordje , COM poort)

- editten

- compileren

- uitvoeren


Arduino File Edit Sketch Tools Help
sketch_jun21a | Arduino 1.6.12
sketch_jun21a
void setup() {
// put your setup code here, to run once:
}
void loop() {
// put your main code here, to run repeatedly:
}
Please select a port to obtain board info
6
Arduino/Genuino Uno on /dev/cu.wchusbserial1410

classic-C versus Arduino-C

```c
#include <stdio.h>

void main()

{

 printf("hello world\n") ;

}
```

```c
void setup()

{



}

void loop()

{

}
```

setup() - loop()

- setup()
  - alle initialisaties , vb. input PIN , output PIN , baudrate v/d seriele poort
  - wordt slechts 1 maal uitgevoerd bij het starten v/h programma
- loop()
  - een oneindige lus

Een eerste voorbeeld

Emacs File Edit Options Tools C Buffers Help

test_1.c

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

-:--- test_1.c All L1 (C/l Abbrev)

Loading cc-langs...done

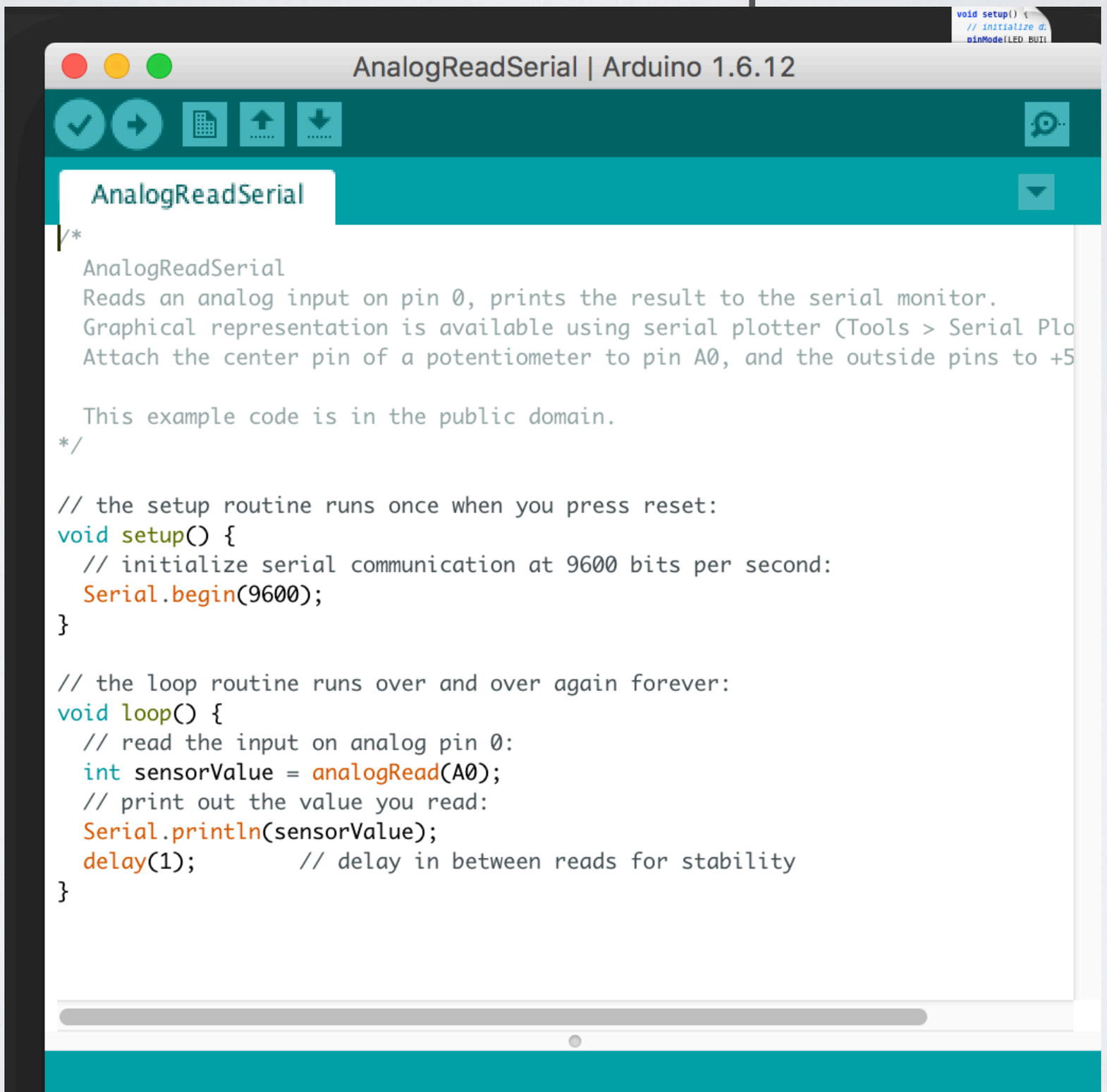
wat achtergrond info

- het compileren gebeurt met : avr-gcc (= cross-compileren).

- een programma noemt men een “sketch”

- het programma staat in een directory met dezelfde naam als het programma (zonder de extensie)

Seriele output



Gebruik de Help -> Reference

- daarin staat er heel veel uitleg.

- een paar voorbeelden

Arduino CURSUS

door Willy - w2@skynet.be , 30-juni-2017

Voorbeelden - Stoplicht

```
#define ROOD 7
#define ORANJE 6
#define GROEN 5

void alle_leds_uit()

{
 digitalWrite(ROOD,HIGH) ;
 digitalWrite(ORANJE,HIGH) ;
 digitalWrite(GROEN,HIGH) ;
}

void setup()

{
 pinMode(ROOD,OUTPUT) ;
 pinMode(ORANJE,OUTPUT) ;
 pinMode(GROEN,OUTPUT) ;

 pinMode(13,OUTPUT) ;
 digitalWrite(13,LOW)  ; // on-board LED OFF

 alle_leds_uit() ;
}
```

```
void led_on(int led)

{
 digitalWrite(led,LOW) ; // LED ON
 if (led == ORANJE) delay(2000) ;
  else delay(5000) ;
 digitalWrite(led,HIGH) ; // LED OFF
}

void loop()

{
 led_on(GROEN) ; // 5 sec.
 led_on(ORANJE) ; // 2 sec.
 led_on(ROOD) ; // 5 sec.
}
```

Met drukknop

```
#define ROOD 7
#define ORANJE 6
#define GROEN 5
#define SCHAKELAAR 2

void setup()

{
 pinMode(ROOD,OUTPUT) ;
 pinMode(ORANJE,OUTPUT) ;
 pinMode(GROEN,OUTPUT) ;

 pinMode(SCHAKELAAR,INPUT_PULLUP) ;

 pinMode(13,OUTPUT) ;
 digitalWrite(13,LOW)  ; // on-board LED OFF

 alle_leds_uit() ;
}
```

```
void led_on(int led)

{
 digitalWrite(led,LOW) ; // LED ON
 if (led == ORANJE) delay(2000) ;
  else delay(5000) ;
 digitalWrite(led,HIGH) ; // LED OFF
}

void alle_leds_uit()

{
 digitalWrite(ROOD,HIGH) ;
 digitalWrite(ORANJE,HIGH) ;
 digitalWrite(GROEN,HIGH) ;
}

void loop()

{
 if (digitalRead(SCHAKELAAR) == 0) {
  led_on(GROEN) ; // 5 sec.
  led_on(ORANJE) ; // 2 sec.
  led_on(ROOD) ; // 5 sec.
 }
}
```

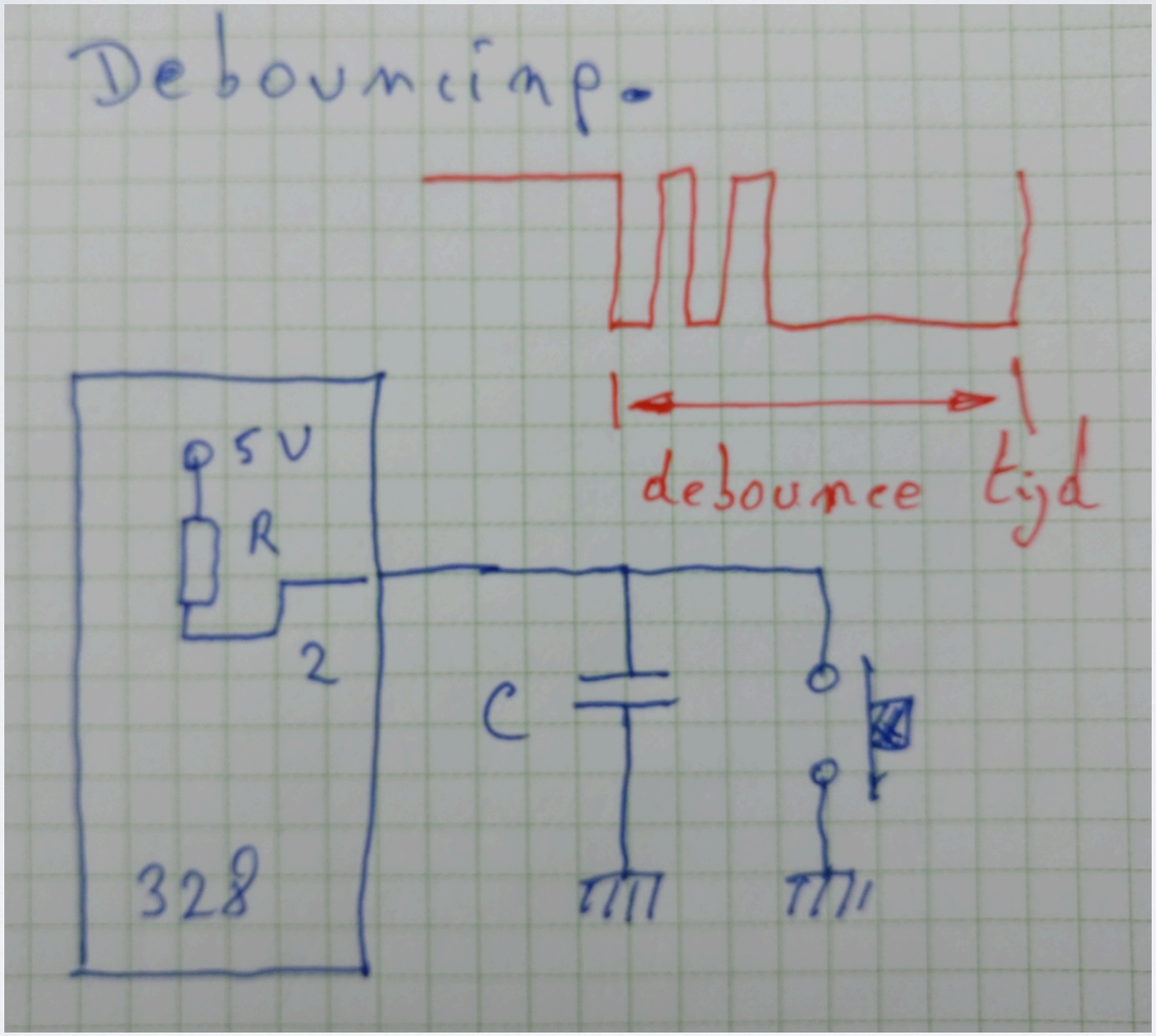
Interrupt (1)

```
#define ROOD 7
#define BUTTON 2

void setup()

{
 pinMode(ROOD,OUTPUT) ;
 digitalWrite(ROOD,HIGH) ;

 pinMode(BUTTON,INPUT_PULLUP) ;

 pinMode(13,OUTPUT) ;
 digitalWrite(13,LOW)  ; // on-board LED OFF

 attachInterrupt(digitalPinToInterrupt(BUTTON),switch_led,FALLING) ;
}
```

Interrupt (2)

```
void switch_led()

{
 byte status ;

 do { // debounce time = 200 mS.
  for (int n=0 ; n < 20 ; n++) delayMicroseconds(10000) ;
 } while (digitalRead(BUTTON) == 0) ;

 status=digitalRead(ROOD) ; // lees een output PIN
 digitalWrite(ROOD,!status) ;
}

void loop()

{

}
```

debouncing



Let wel : ISR (interrupt service routine)

gebruik geen delay(…) in de ISR

ISR routine zo kort mogelijk houden.

als ge een globale variabele wilt
veranderen dan deze declareren als “volatile”
vb. : volatile int getal ;

Vervolg ?

I2C , SPI , LCD display….

Arduino CURSUS

door Willy - w2@skynet.be , 01-sep-2017

de scope van variabelen

```
#define <.....>

// globale variabelen

int n , m ;
float f ;

void f1()
{
 int n ;
  ....
 n=5 ;
  ....
}

void f2()

{
 int m ;
  ....
 m=5 ;
  ....
}
```

```
void setup()

{
 Serial.begin(9600) ;

 n = 10 ; // de globale n

 m = 100 ; // de globale m
}

void loop()

{
 n=n+1 ; // de globale n
 m=m+1 ; // de globale m
 Serial.println(n) ;
 Serial.println(m) ;
 delay(1000) ;
}
```

soorten variabelen

- boolean : true/false , 0/1
- char : mychar='A' , str[10]
- byte : b=B00011100
- int : 16 bit getal , range : -32768 ... +32767
- unsigned int :
- long : 32 bits getal , a = 100000L
- unsigned long :
- float : a=3.14 (voor komma getallen)
- double : zelfde als float (op de UNO)

commentaar in een C programma

2 manieren :

```
// per regel
```

of :

```
/*

 een aantal regels

*/
```

A/D conversie

- meten v/e analoge spanning (0 ... 5 Volt) ,
  vb. op pin A0 ... A5

- het resultaat is een getal ts. 0 en 1023 (10 bit)

- omzetten naar Volt : getal*ADCST ,
  en ADCST = 5.0/1024.0 , resolutie = +- 5 mV.

- als ge niets zegt is de referentie de + 5V voeding

- of : analogReference(internal) , ref. = 1.1 Volt

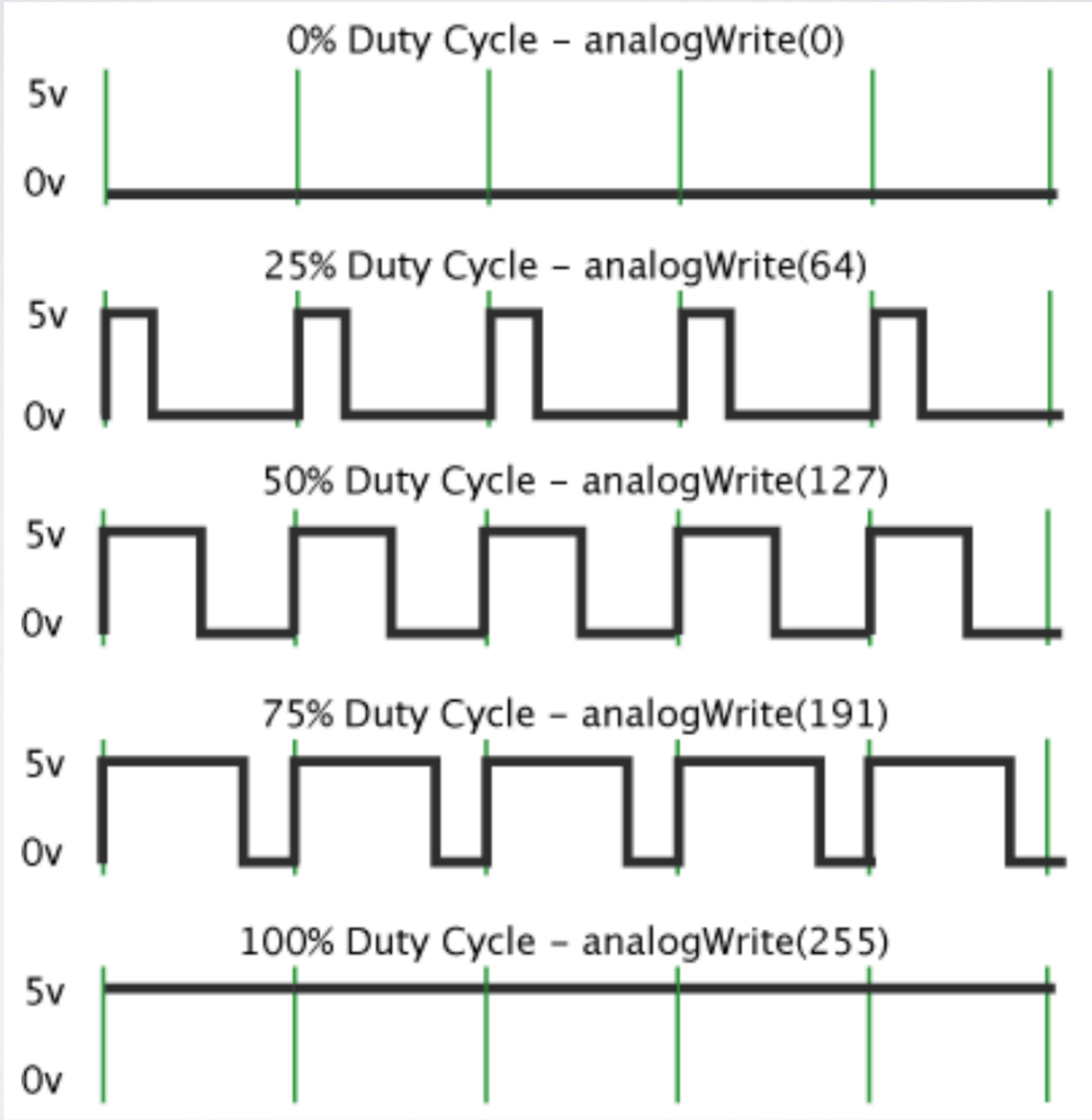
A/D : een voorbeeld

```
#define ADCST 5.0/1024.0 // 10 bit A/D

void setup()

{
 Serial.begin(9600) ; // baudrate 9600 baud
}

void loop()

{
 int analog ;
 float volt ;

 analog=analogRead(A0) ; // 0 .. 1023
 volt=analog*ADCST ; // 0.0 .. 5 volt
 Serial.println(volt) ;
 delay(1000) ;
}
```

A/D : verkorte versie

```
#define ADCST 5.0/1024.0 // 10 bit A/D

void setup()

{
 Serial.begin(9600) ; // baudrate 9600 baud
}

void loop()

{
 Serial.println(analogRead(A0)*ADCST) ;
 delay(1000) ;
}
```

PWM - pulse width modulation



PWM - waarvoor

- DC motor sturingen , vb. ventilatoren

- aansturen van LED verlichting

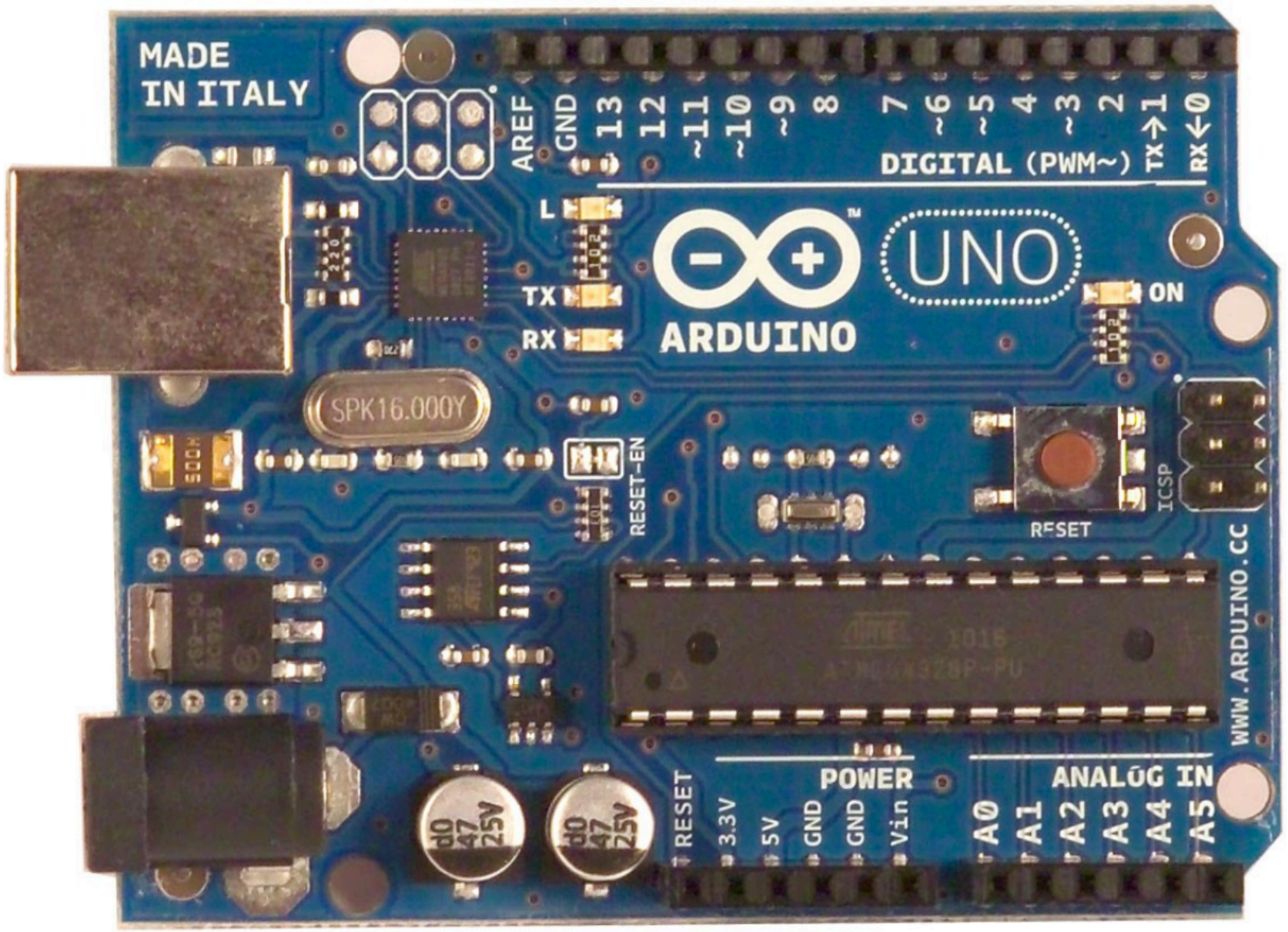
- voor het maken van analoge golfvormen

- eventueel via een vermogen FET om grotere stromen te kunnen aansturen , of een solid state relais om 230 V toestellen te sturen.

- op een Arduino kunnen alle pinnen met de aanduiding ~ (3 , 5 , 6 , 9 , 10 , 11) gebruikt worden voor PWM sturing.

- de basis herhaal frequentie is ongeveer 490 Hz.

Arduino UNO


MADE
IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
SPK16.000Y
RESET-EN
RESET
ICSP
WWW.ARDUINO.CC
POWER
ANALOG IN
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

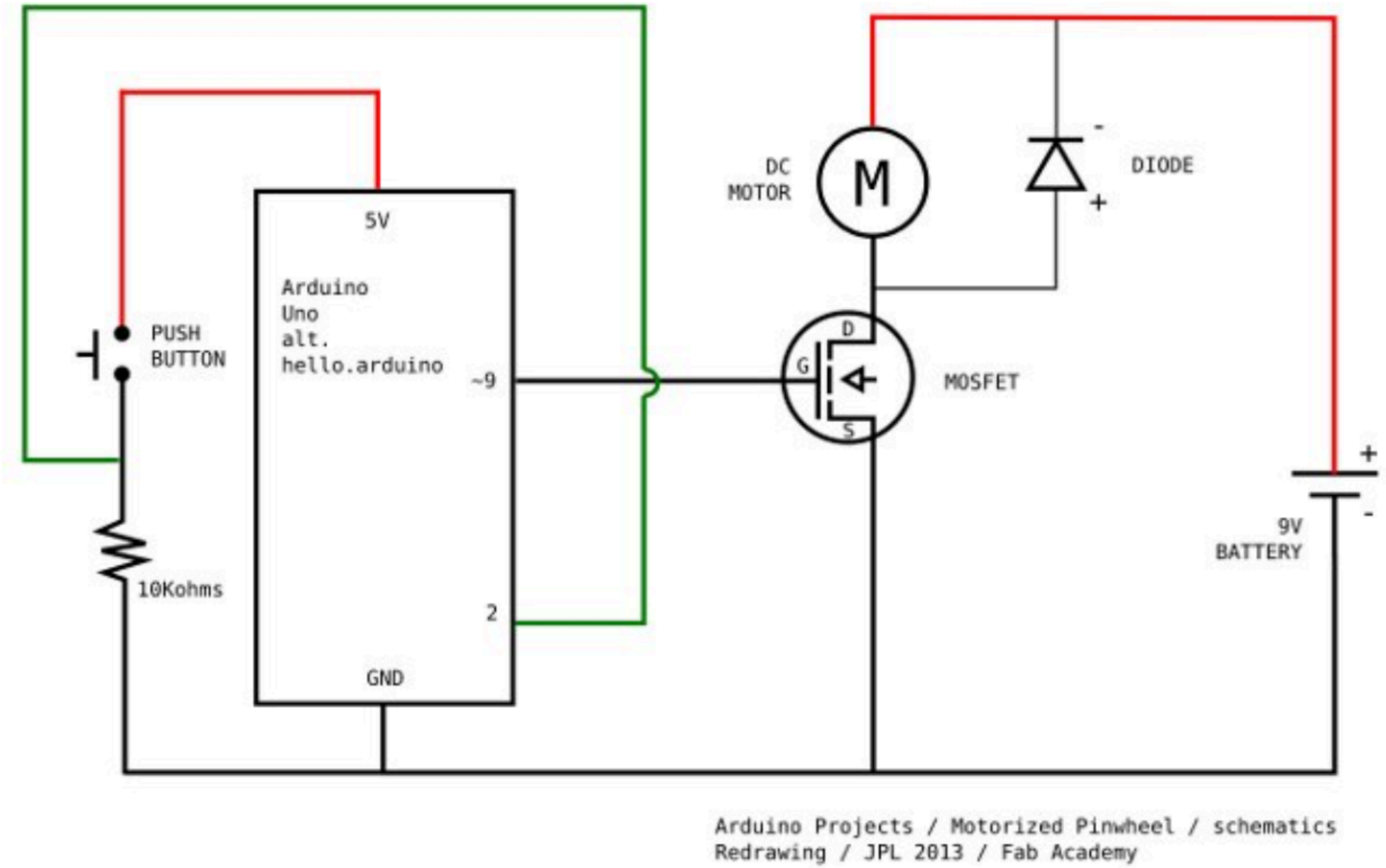
PWM voorbeeld

```
#define Relais 5

void setup()

{

}

void loop()

{
 int n ;

 for (n=0 ; n < 256 ; n++) {
  analogWrite(Relais,n) ;
  delay(200) ;
 }

 for (n=254 ; n > 0 ; n--) {
  analogWrite(Relais,n) ;
  delay(200) ;
 }
}
```

analogWrite(pin,value)

en value is een getal : 0 .. 255

FET sturing

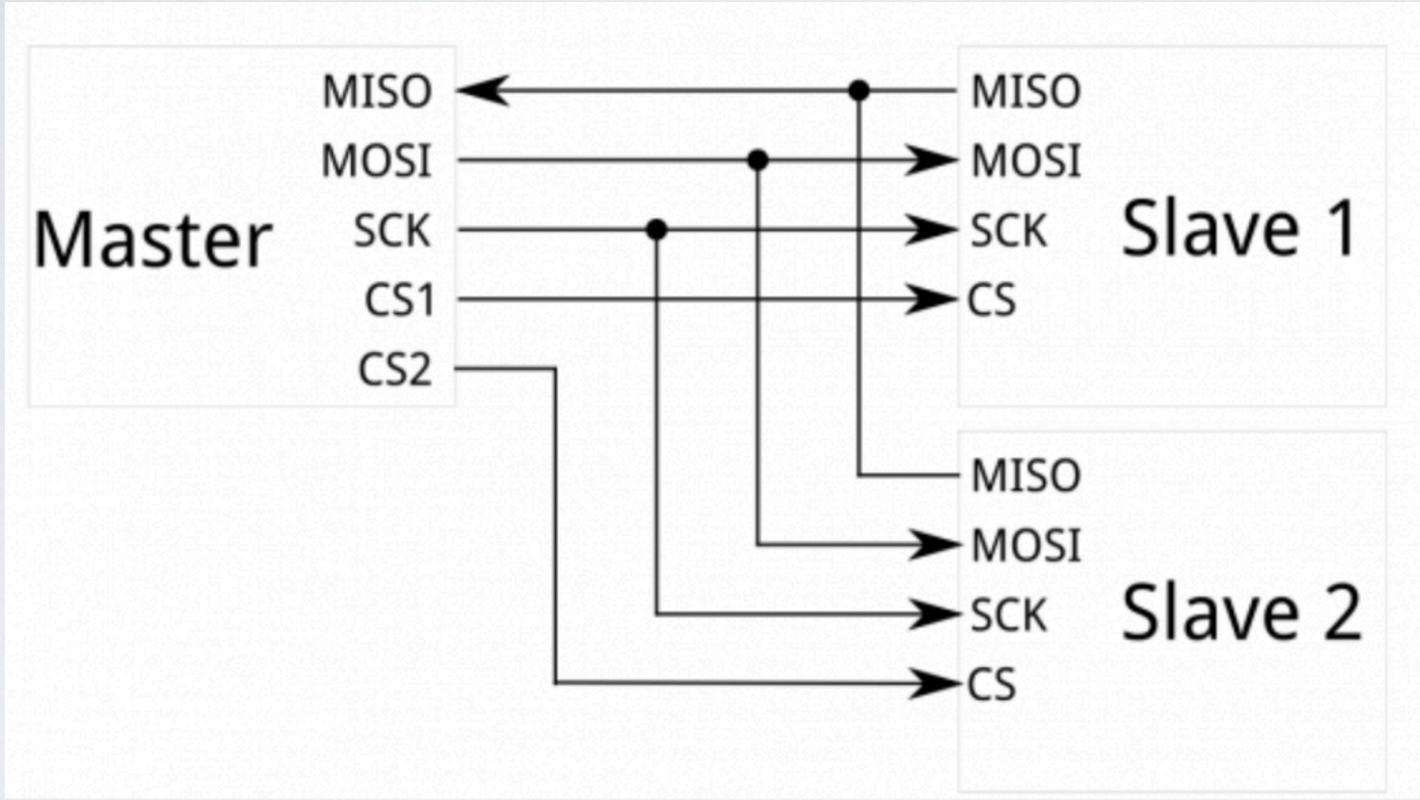

5V
Arduino
Uno
alt.
hello.arduino
~9
2
GND
PUSH
BUTTON
10Kohms
DC
MOTOR
M
DIODE
-
+
D
G
S
MOSFET
+
-
9V
BATTERY
Arduino Projects / Motorized Pinwheel / schematics
Redrawing / JPL 2013 / Fab Academy

Arduino CURSUS

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

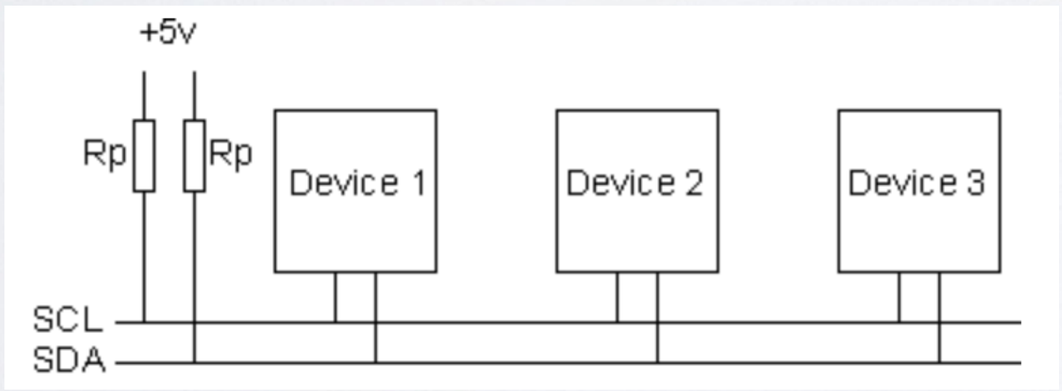
I2C of two wire protocol versus SPI

SPI



per slave een extra chip select (CS) pin

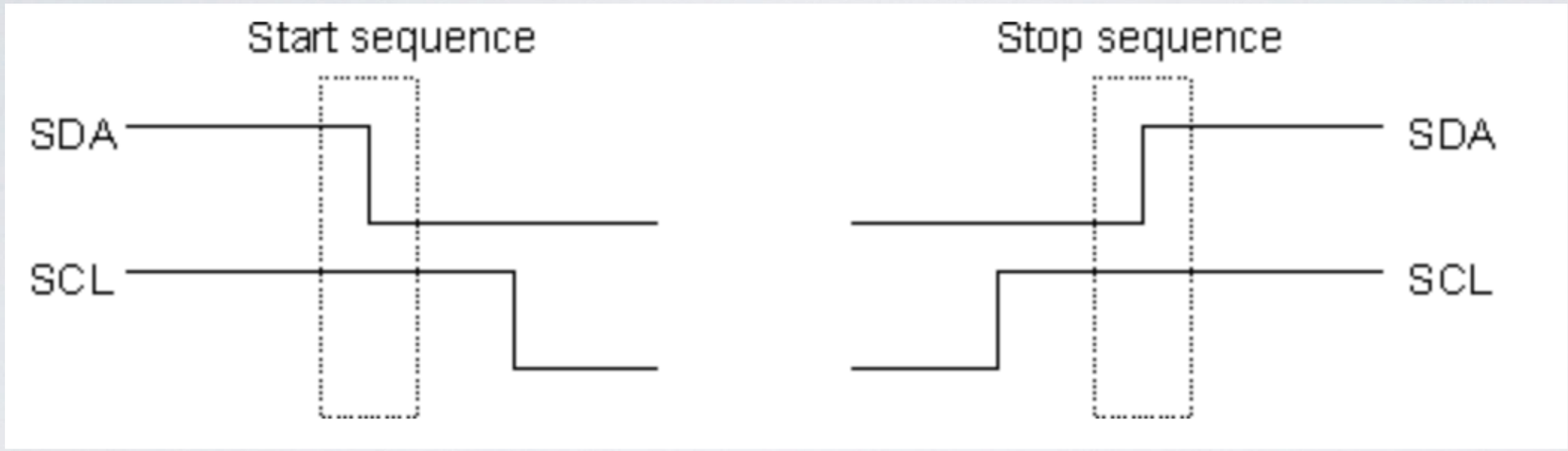
I2C

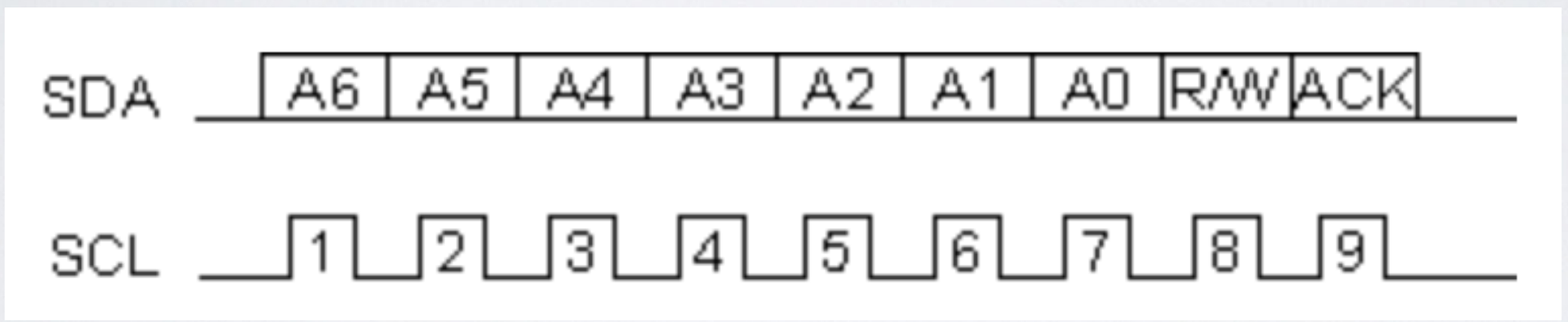


4 draden : Vcc , GND SCL en SDA

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.

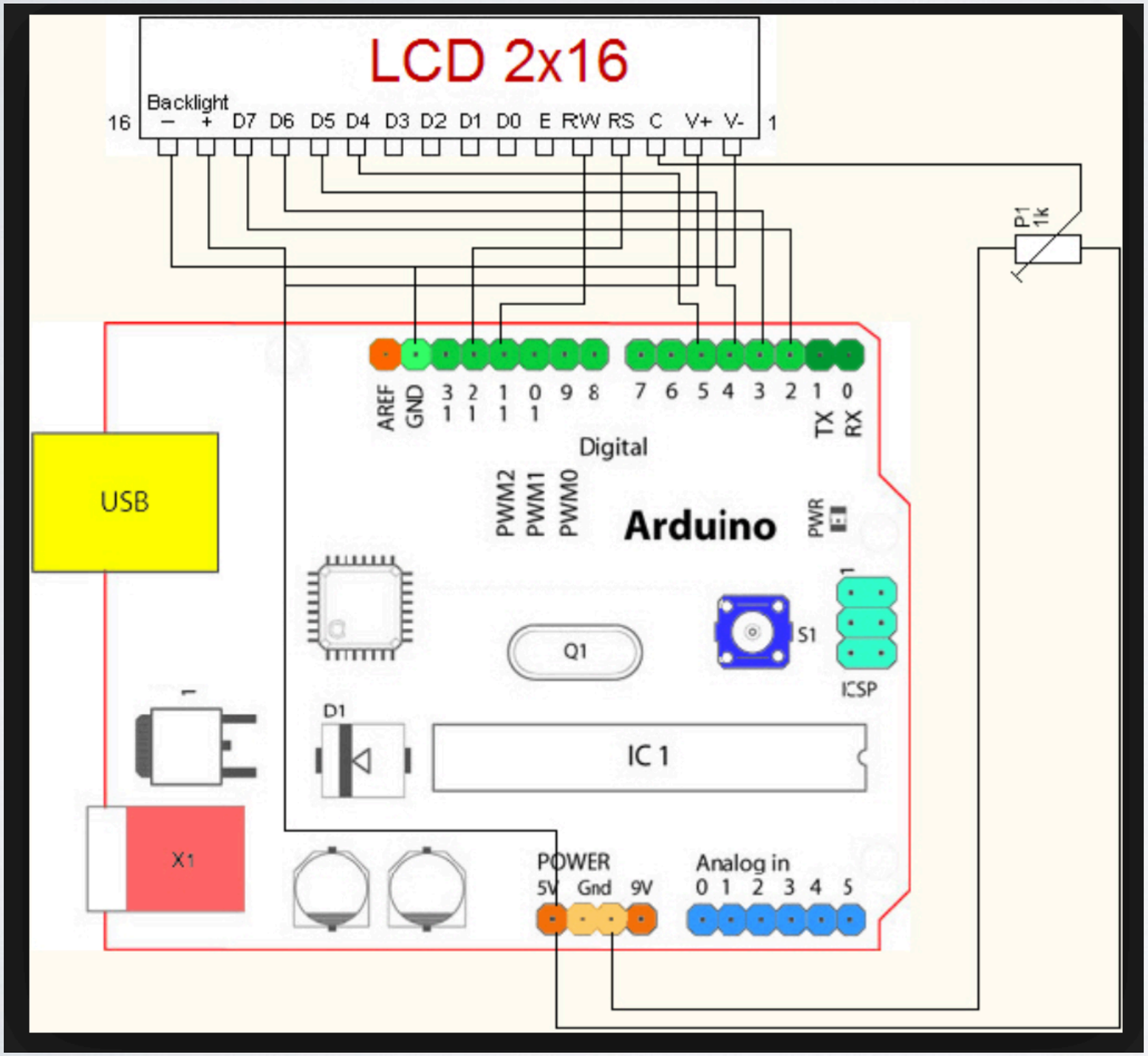
- EEPROM en RAM geheugens.

- enz....

Een I-ste voorbeeld

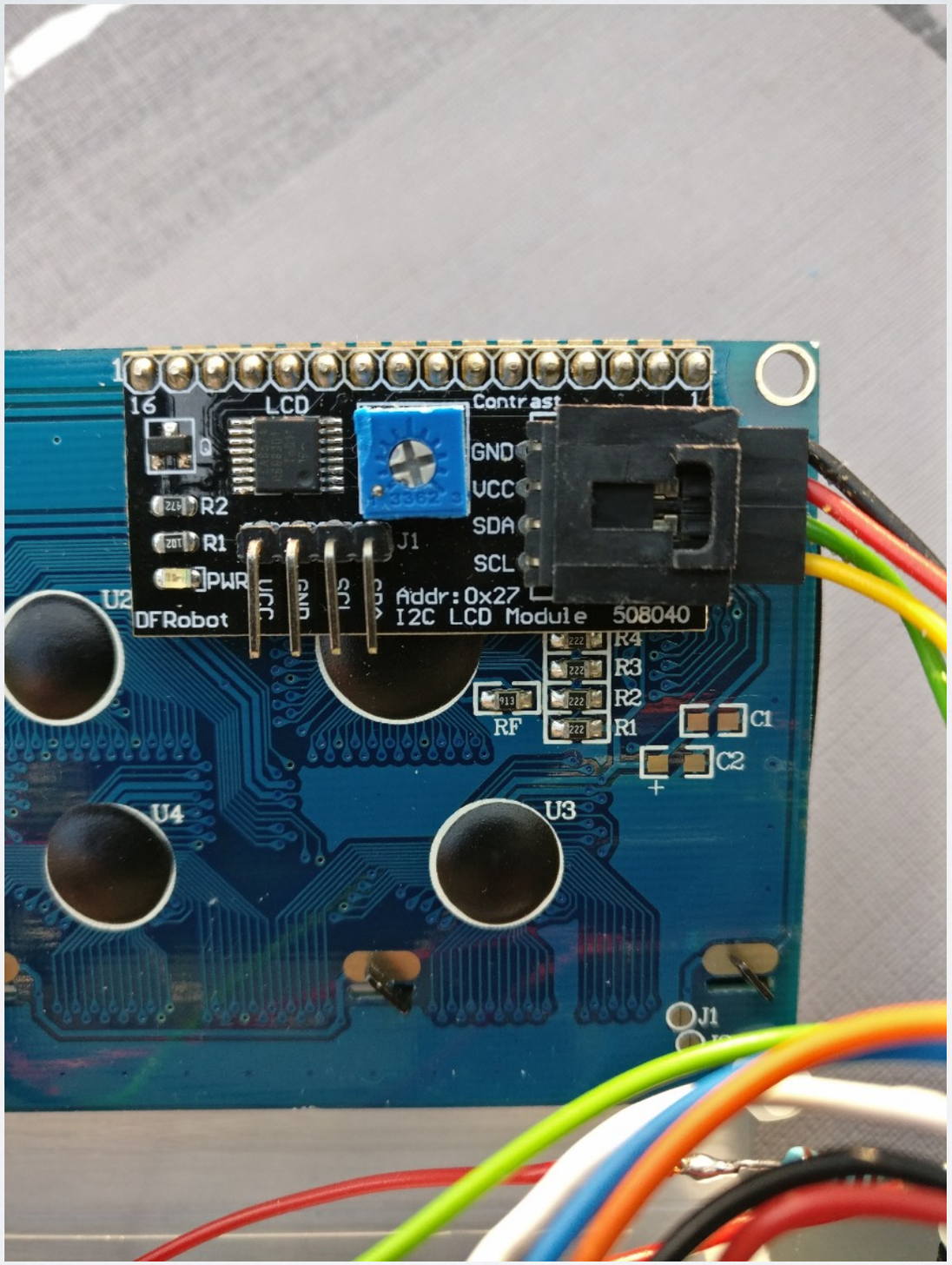
```
#include <Wire.h>

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)

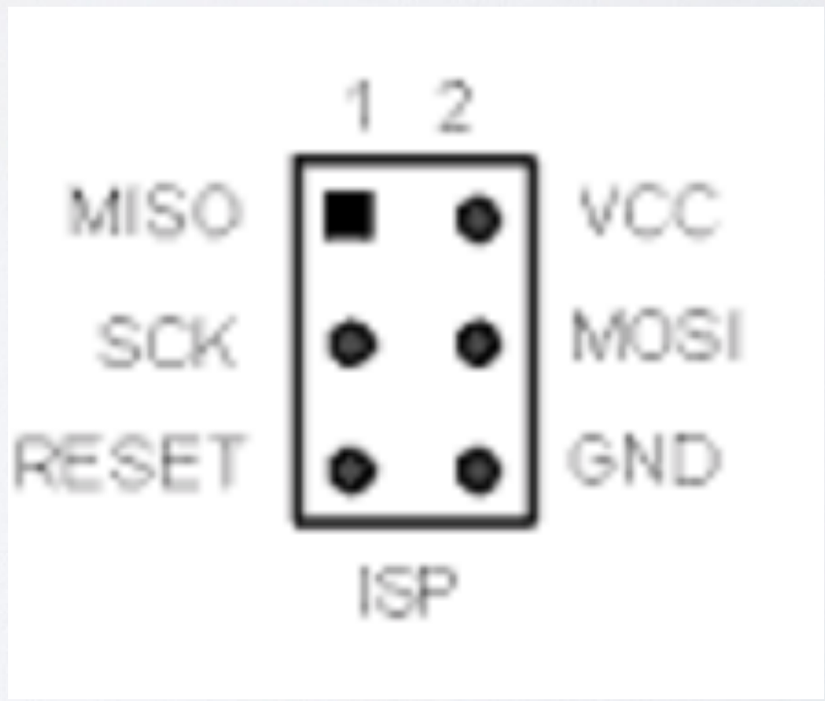
```
void welcome() // 20-sep-2017

{
 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 ?

`