



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.



Wireless Networks & Arduino

Bluetooth, Wifi, Xbee, Grove
LEDs, Servos, Steppenmotoren, Relais, Sensors (Temperatuur,
Luchtvochtigheid, Alcohol, druk, etc)

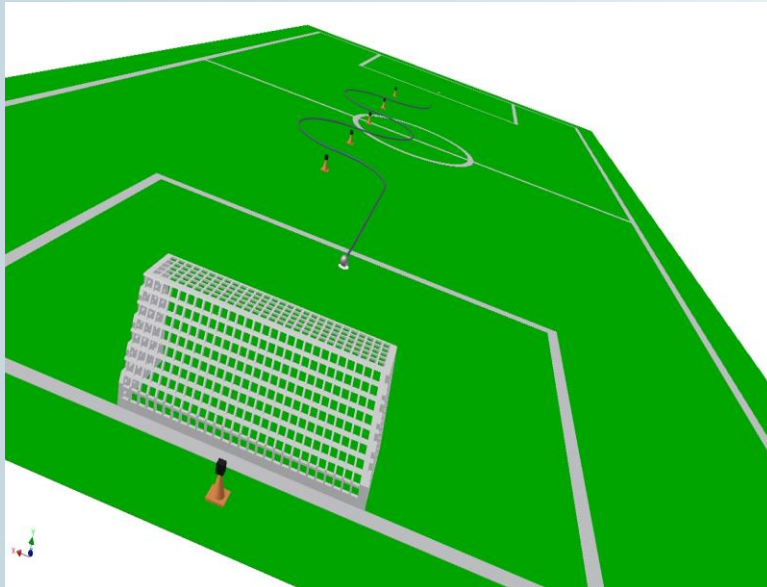


Els van Tol

- Els.vantol@gmail.com
- Docent Informatica College Hageveld







- Afstudeerscriptie:

Wireless Sensing Networks with the Arduino
In opdracht van Oracle.



← → ↻ www.footballnote.com/wordpress/

Sign In Register Contact

Features » Offerings Blog Brochure      







FOOTBALL NOTE
Footballnote.com


30 DAYS FREE TRIAL

Live your passion through innovation

Finally, it is here. FootballNote.com – solution, which helps football (some call it soccer) coaches in daily life.

[read more](#)

Share      



4 april 2013 – NIOC, <http://www.openarduino.nl/>; E.H. van Tol-Homan



Onderwerpen in deze presentatie





Arduino

De Arduino





Arduino Micro



Arduino Nano



Arduino Ethernet



Arduino Mini



Arduino Mega



Arduino Pro Mini



Arduino Pro Mini



Microcontroller

ATmega328

Operating Voltage

5V

Input Voltage (recommended)

7-12V

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) of which 0.5 KB used by bootloader

SRAM

2 KB (ATmega328)

EEPROM

1 KB (ATmega328)



Arduino Due

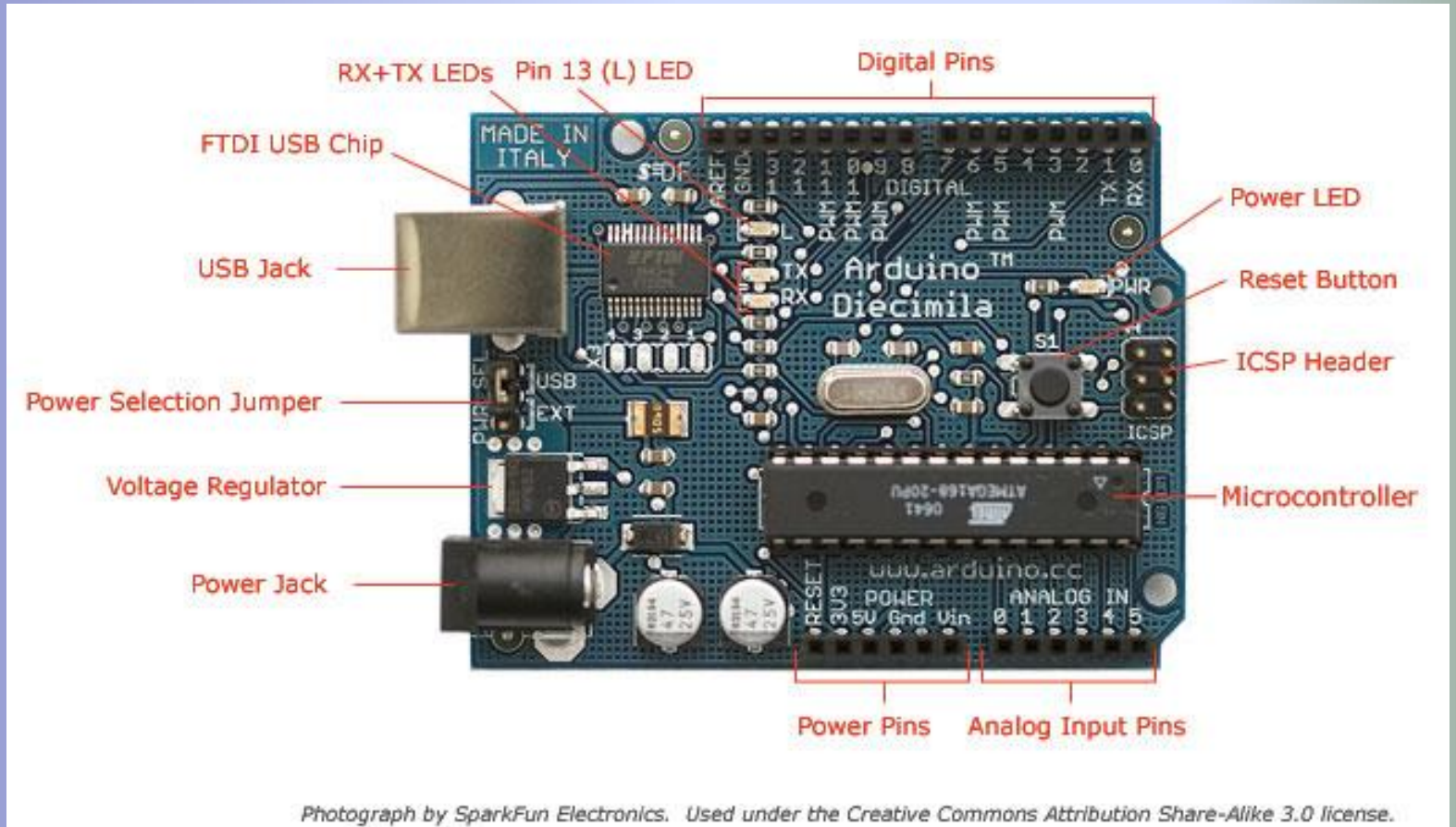


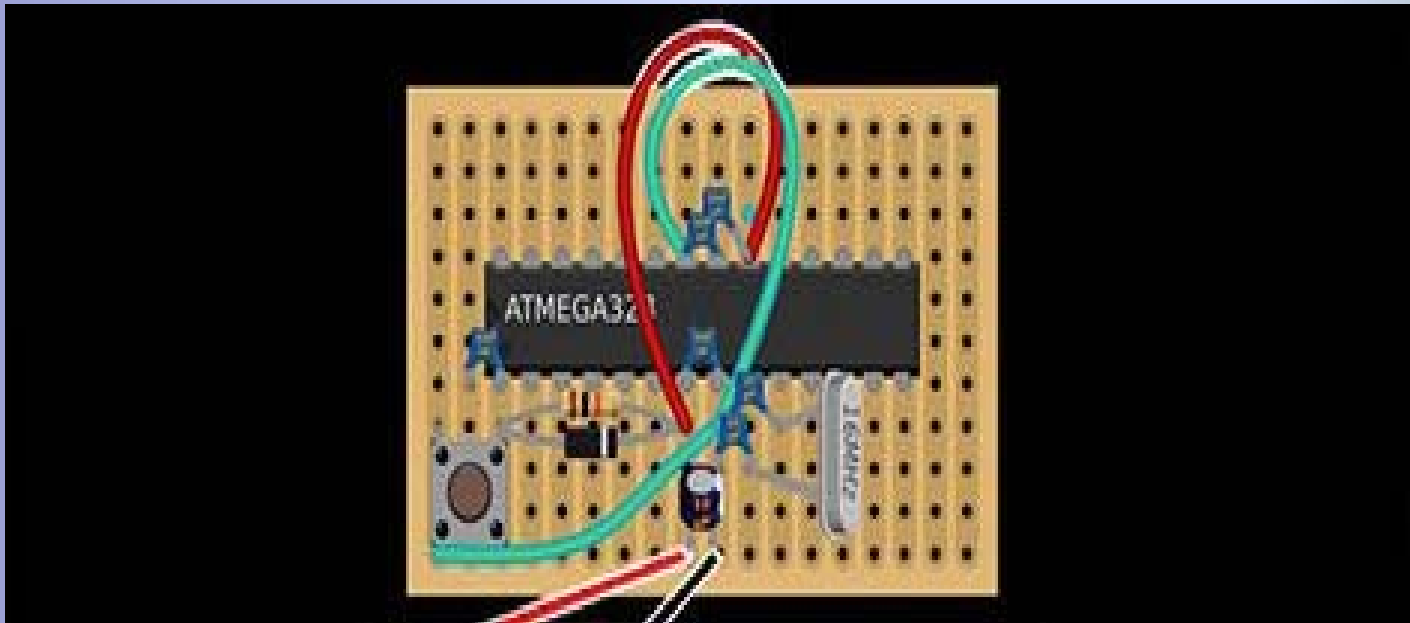
Arduino Uno

Clock Speed

16 MHz

Arduino





IDE

- Arduino IDE

C++

```
void setup()
```

```
{
```

```
void loop()
```

```
{
```

- Processing IDE

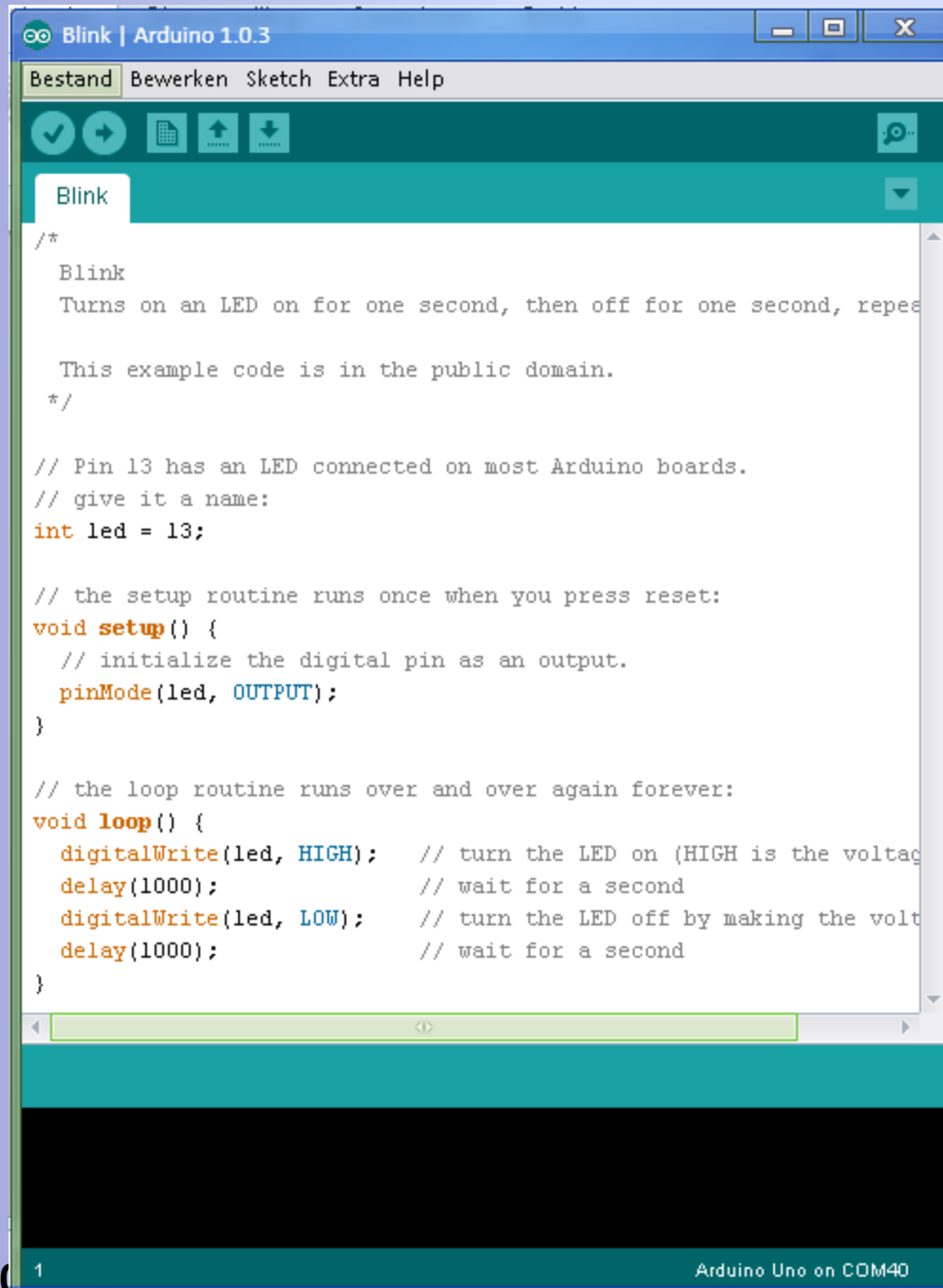
Java

```
void setup()
```

```
{
```

```
void draw()
```

```
{
```

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.3". The menu bar includes "Bestand", "Bewerken", "Sketch", "Extra", and "Help". The toolbar contains icons for saving, running, uploading, and downloading. The main text area displays the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeats.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

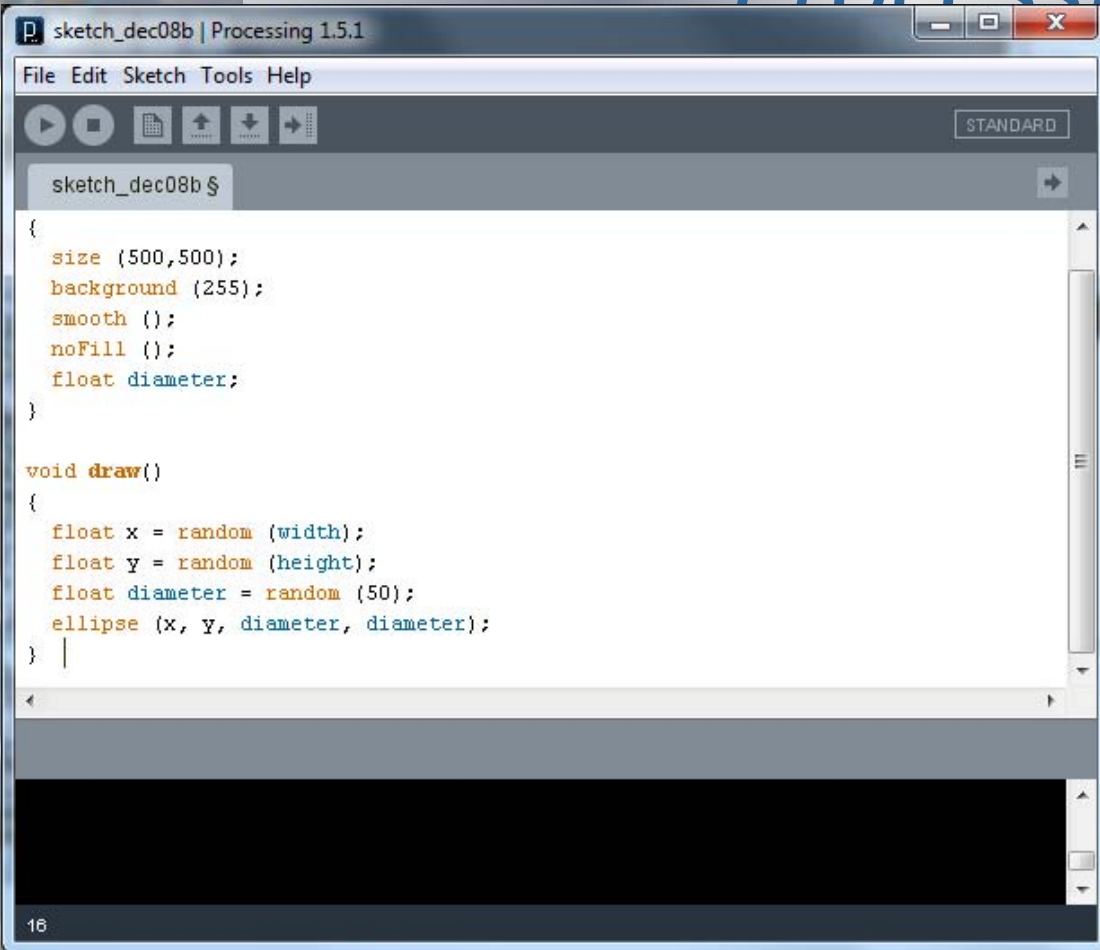
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for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

The status bar at the bottom shows "1" on the left and "Arduino Uno on COM40" on the right.

A close-up, angled view of an Arduino Uno board, showing the microcontroller, various components, and the white USB Type-B connector. The board is blue and populated with various electronic components.

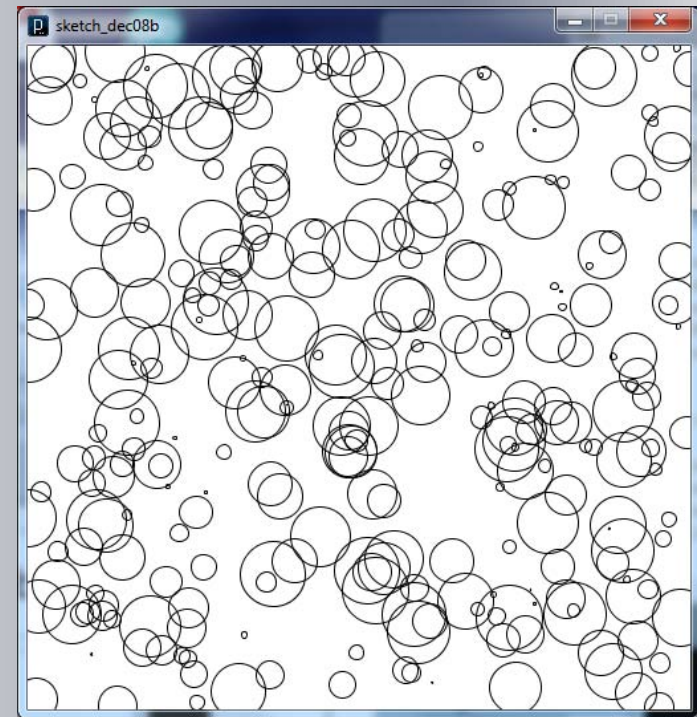
Processing

Processing IDE

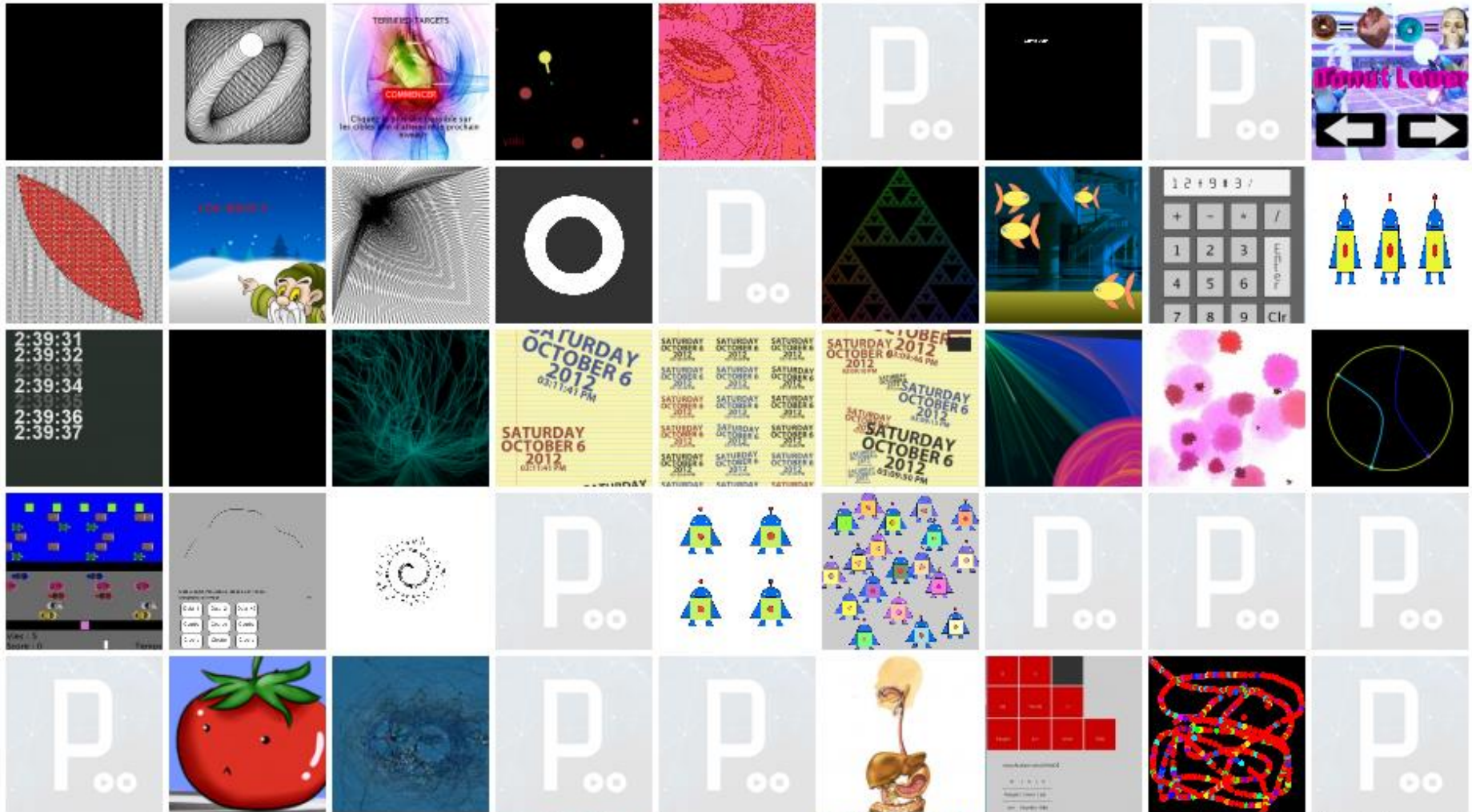


```
sketch_dec08b | Processing 1.5.1
File Edit Sketch Tools Help
STANDARD
sketch_dec08b $
{
  size (500,500);
  background (255);
  smooth ();
  noFill ();
  float diameter;
}

void draw()
{
  float x = random (width);
  float y = random (height);
  float diameter = random (50);
  ellipse (x, y, diameter, diameter);
}
```



OpenProcessing.org





Arduino INPUT

Input



A close-up, vertical view of an Arduino board, showing various components like resistors, capacitors, and a microcontroller chip. The board is blue and populated with numerous electronic components.

Analoog / Digitaal Input

analogRead()

Description

Reads the value from the specified analog pin. The Arduino board contains a 6 channel (8 channels on the Mini and Nano, 16 on the Mega), 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit. The input range and resolution can be changed using `analogReference()`.

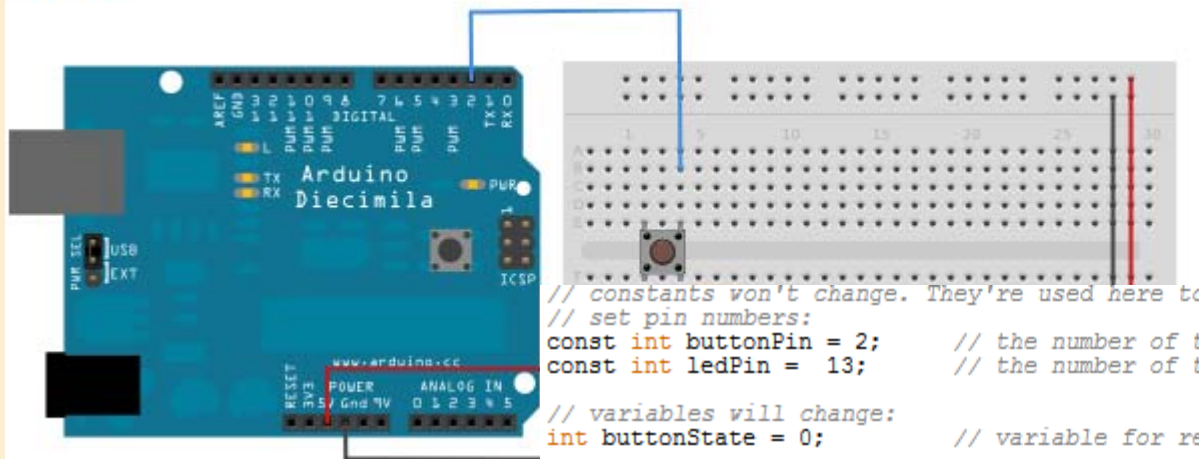
digitalRead()

Description

Reads the value from a specified digital pin, either `HIGH` or `LOW`.

DigitalRead

Circuit



```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;     // the number of the LED pin

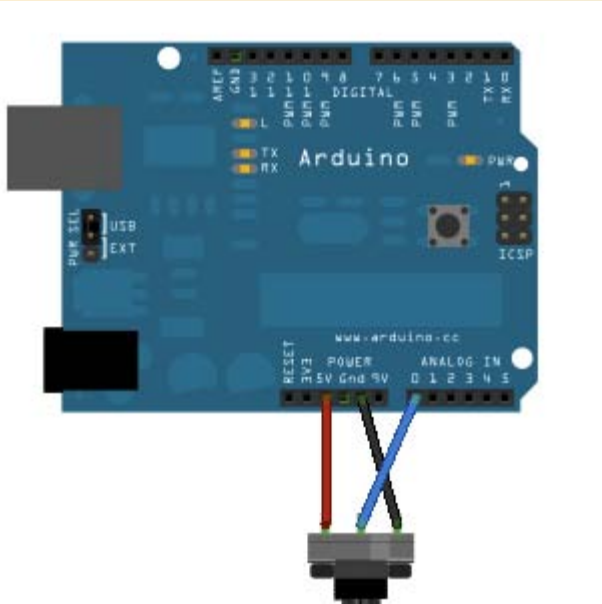
// variables will change:
int buttonState = 0;       // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

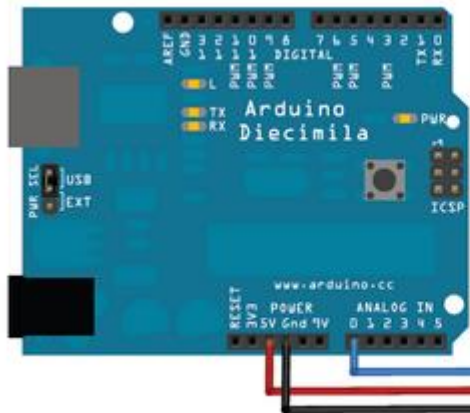
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

AnalogRead



```
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize serial communication at 9600 bits per second:  
  Serial.begin(9600);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  // print out the value you read:  
  Serial.println(sensorValue);  
  delay(1);      // delay in between reads for stability  
}
```

AnalogF



Copy Code

```
//TMP36 Pin Variables
int sensorPin = 0; //the analog pin the TMP36's Vout (sense) pin is connect
                    //the resolution is 10 mV / degree centigrade with
                    //500 mV offset to allow for negative temperatures

/*
 * setup() - this function runs once when you turn your Arduino on
 * We initialize the serial connection with the computer
 */
void setup()
{
  Serial.begin(9600); //Start the serial connection with the computer
                    //to view the result open the serial monitor
}

void loop()
    // run over and over again
{
  //getting the voltage reading from the temperature sensor
  int reading = analogRead(sensorPin);

  // converting that reading to voltage, for 3.3v arduino use 3.3
  float voltage = reading * 5.0;
  voltage /= 1024.0;

  // print out the voltage
  Serial.print(voltage); Serial.println(" volts");

  // now print out the temperature
  float temperatureC = (voltage - 0.5) * 100; //converting from 10 mv per
                                             //to degrees ((voltage - 500
  Serial.print(temperatureC); Serial.println(" degrees C");

  // now convert to Fahrenheit
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF); Serial.println(" degrees F");

  delay(1000); //waiting a second
}
```

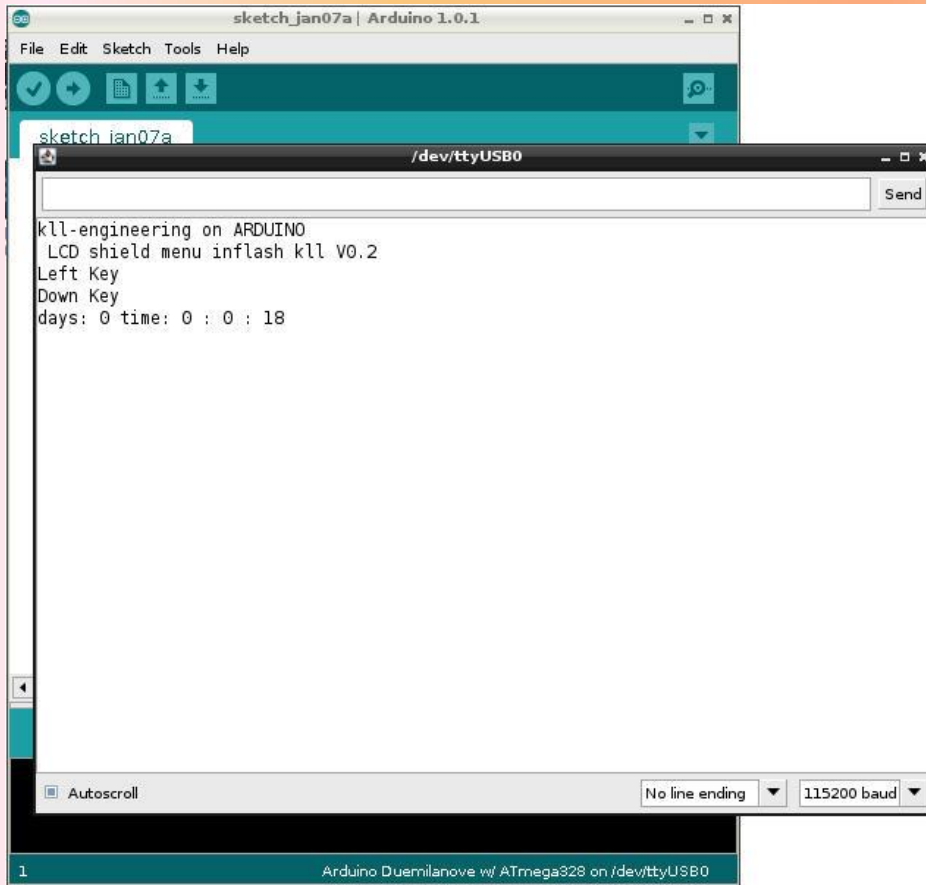


Arduino OUTPUT

Output



Terminal



The image shows a screenshot of an Arduino IDE terminal window. The window title is "sketch_jan07a | Arduino 1.0.1". The terminal output displays the following text:

```
kll-engineering on ARDUINO  
LCD shield menu inflash kll V0.2  
Left Key  
Down Key  
days: 0 time: 0 : 0 : 18
```

At the bottom of the terminal window, there is a status bar that reads "1 Arduino Duemilanove w/ ATmega328 on /dev/ttyUSB0". The terminal window also shows a "Send" button and a "No line ending" dropdown menu.

DigitalWrite, AnalogWrite

analogWrite()

Description

Writes an analog value (**PWM wave**) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to **analogWrite()**, the pin will generate a steady square wave of the specified duty cycle until the next call to **analogWrite()** (or a call to **digitalRead()** or **digitalWrite()** on the same pin). The frequency of the PWM signal is approximately 490 Hz.

On most Ar **digitalWrite()**

and 11. On 1

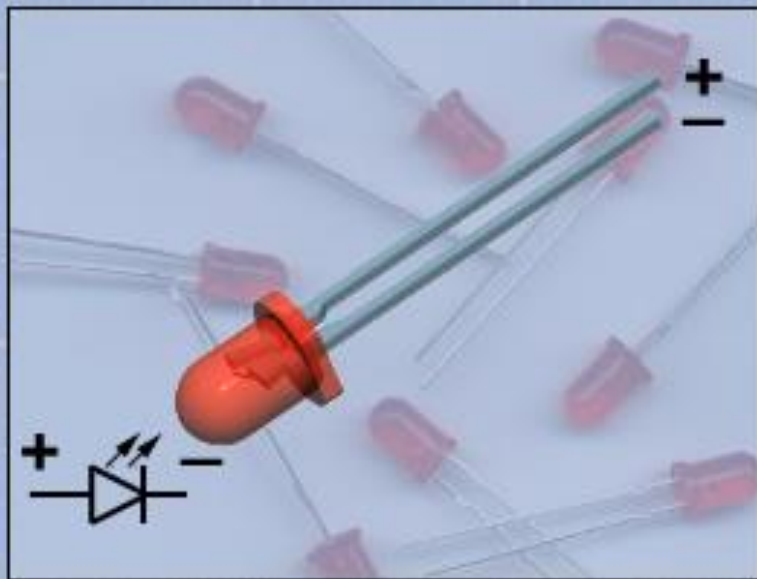
support an: **Description**

The Arduin Write a **HIGH** or a **LOW** value to a digital pin.

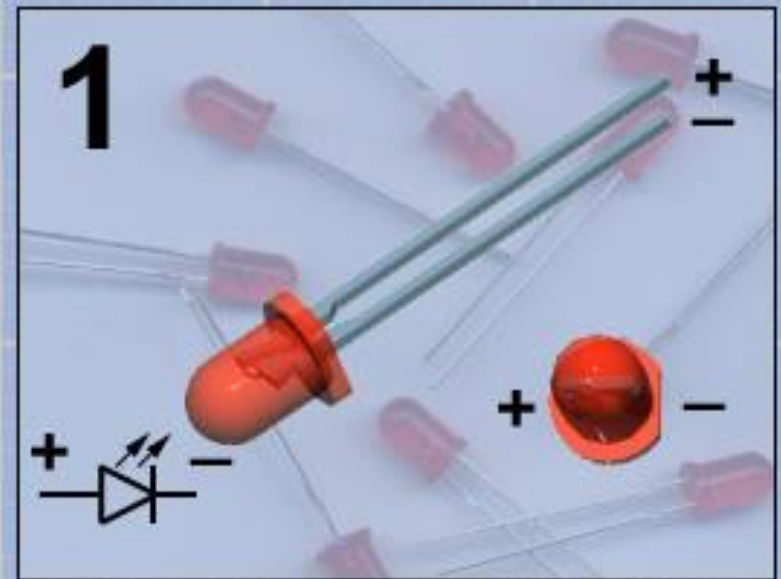
DACo and I If the pin has been configured as an OUTPUT with **pinMode()**, its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

You do not

The *analog* If the pin is configured as an INPUT, writing a HIGH value with **digitalWrite()** will enable an internal 20K pullup resistor (see the **tutorial on digital pins**). Writing LOW will disable the pullup. The pullup resistor is enough to light an LED dimly, so if LEDs appear to work, but very dimly, this is a likely cause. The remedy is to set the pin to an output with the **pinMode()** function.

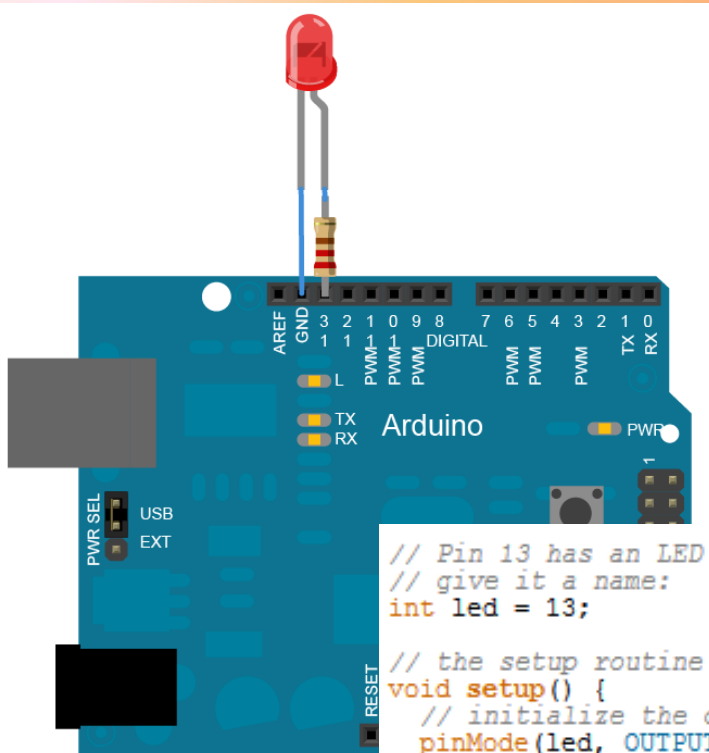


Een LED is een **diode** die licht geeft.
Je moet de LED goed aansluiten.
Gebruik altijd een **weerstand** anders brandt de LED door.



Een LED heeft 2 verschillende poten.
De lange poot is de +
De korte poot is de -
Je ziet het ook aan het rode lensje

DigitalWrite

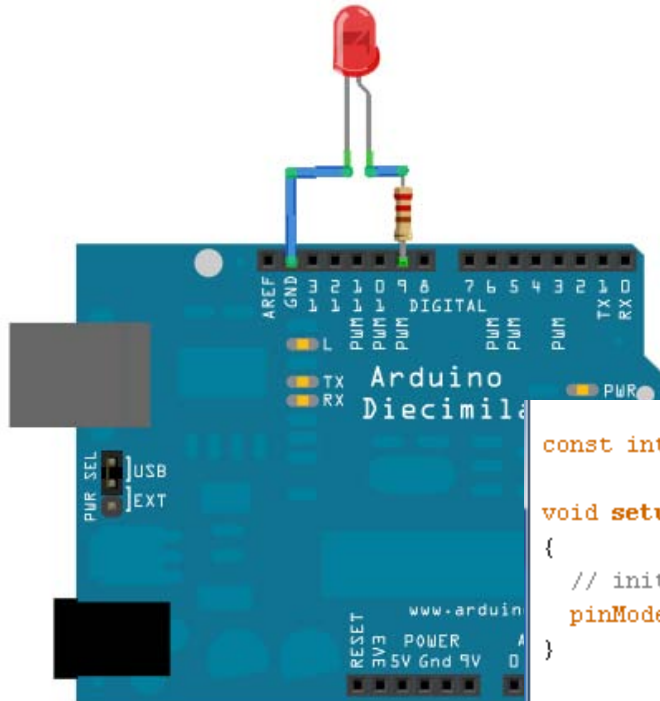


```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

AnalogWrite



```
const int ledPin = 9;      // the pin that the LED is attached to

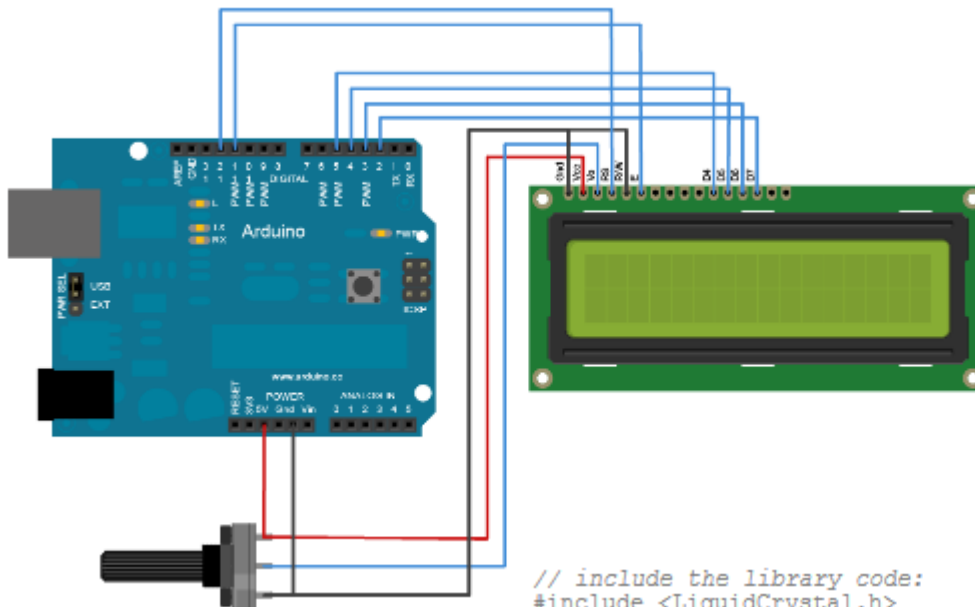
void setup()
{
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  byte brightness;

  // check if data has been sent from the computer:
  for (brightness=0;i<150;i++) {

    // set the brightness of the LED:
    analogWrite(ledPin, brightness);
  }
}
```

SCHERM



```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

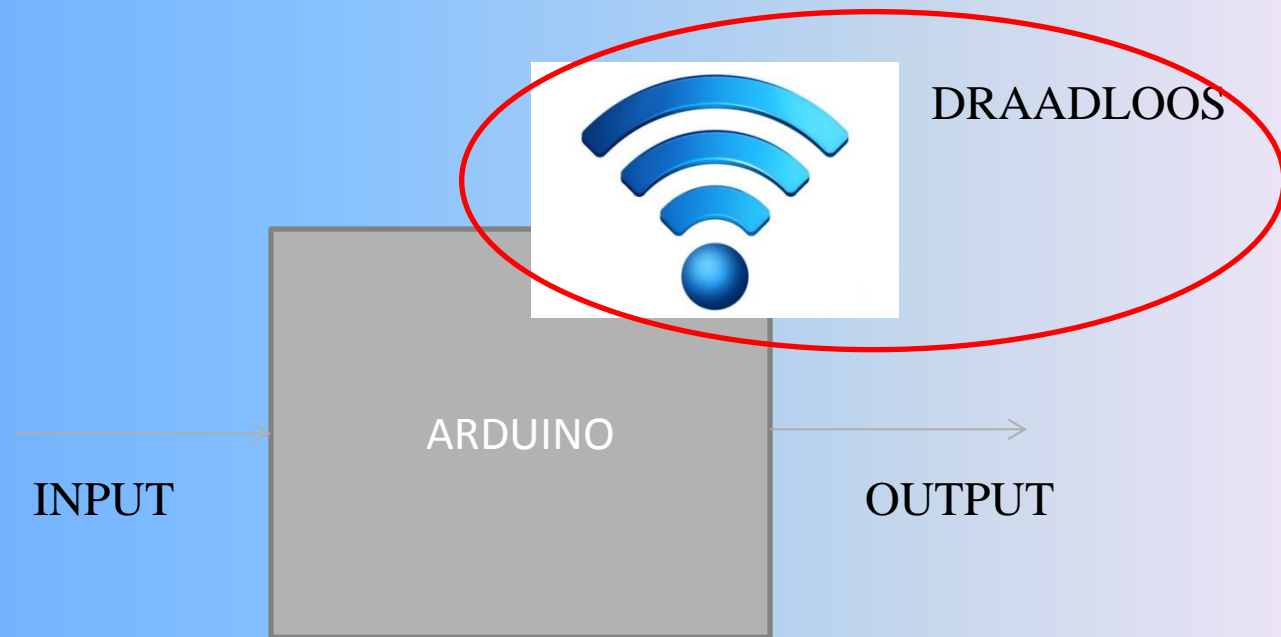
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

A vertical strip on the left side of the slide shows a close-up of an Arduino PCB. It features a microcontroller chip, various resistors, and a yellow LED that is illuminated. The board is blue with white silkscreen markings.

Arduino DRAADLOOS

Draadloos

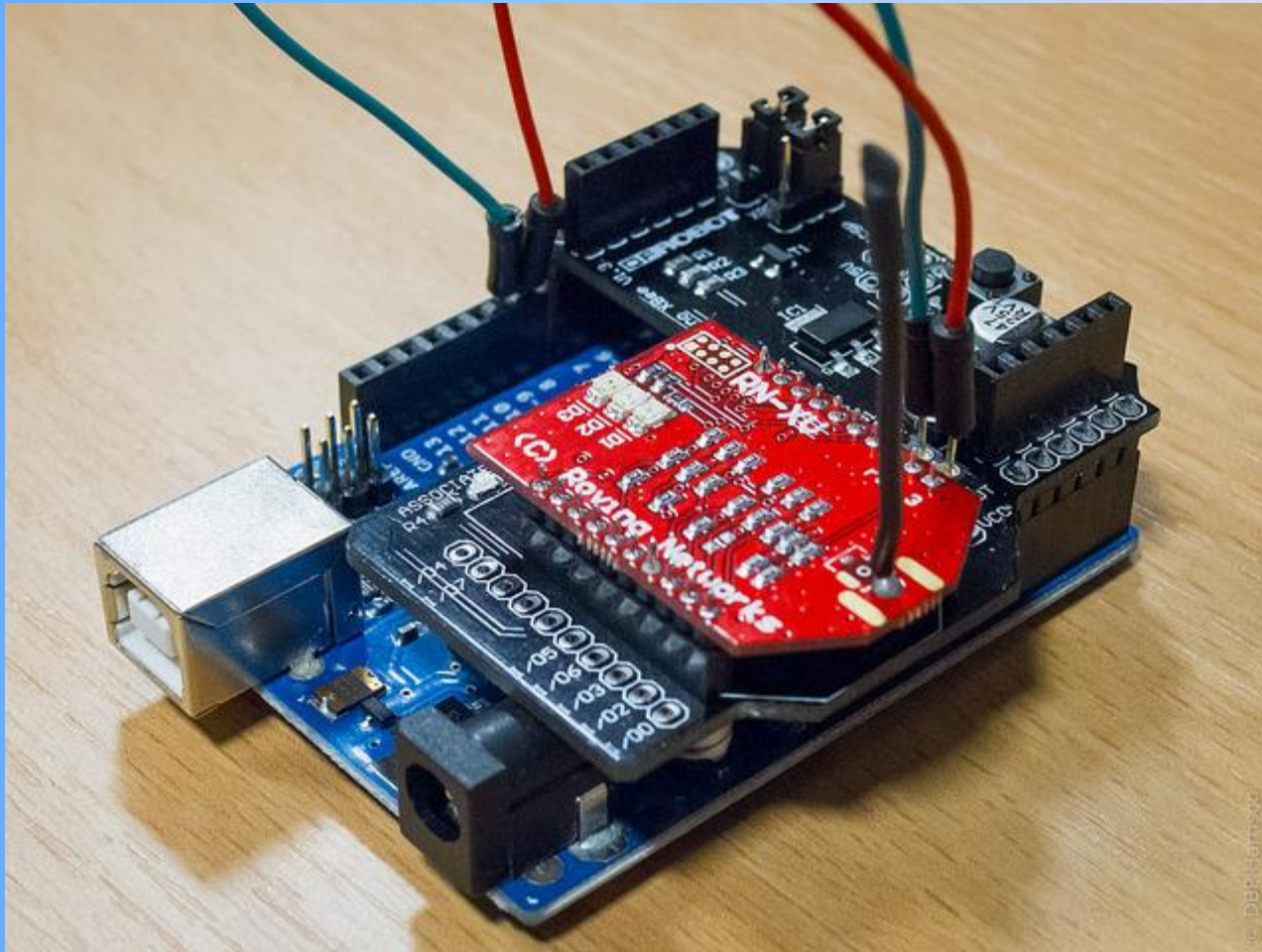


A vertical strip on the left side of the slide shows a close-up of an Arduino board, highlighting various components like a microcontroller, capacitors, and a USB port.

Draadloos

- Wifi
- Ethernet Shield
- Xbee
- Bluetooth
- GSM

Wifi

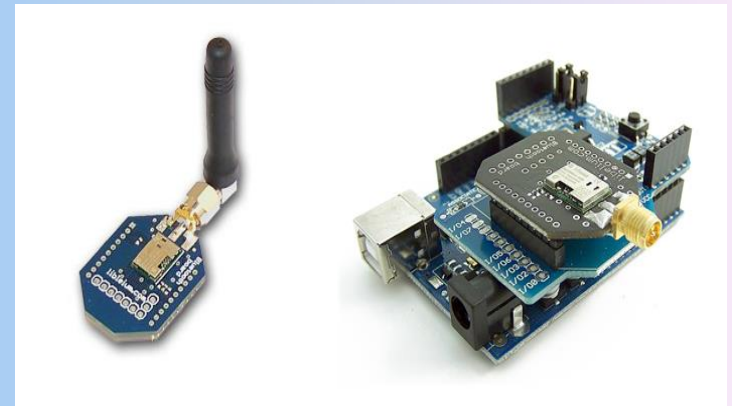


Shields

- Aangesloten op een Shield
- Wordt gebruikt voor:

Bluetooth, Xbee, Wifi

Let op bereik en toepassing !!



A vertical strip on the left side of the slide shows a close-up of an Arduino board, highlighting various components like a microcontroller, resistors, and a USB port.

Libraries

- Voor elke functionaliteit is een aparte library.
- Problemen:
 - afhankelijk van de versie van de Arduino IDE
 - niet altijd compatible met andere libraries
 - geschreven door iedereen
 - let op: de libraries zijn niet altijd gecontroleerd

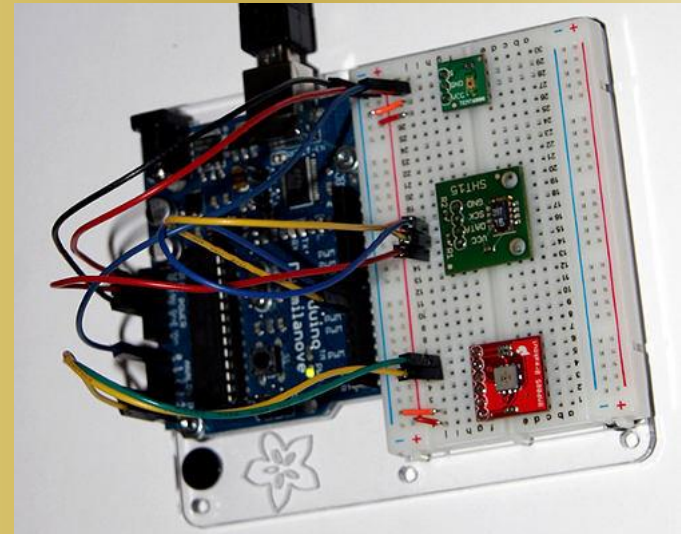
A vertical strip on the left side of the slide shows a close-up of an Arduino board. It features a blue PCB with various components like a microcontroller, resistors, and a USB port. A white infinity symbol is visible on the board.

Workshop

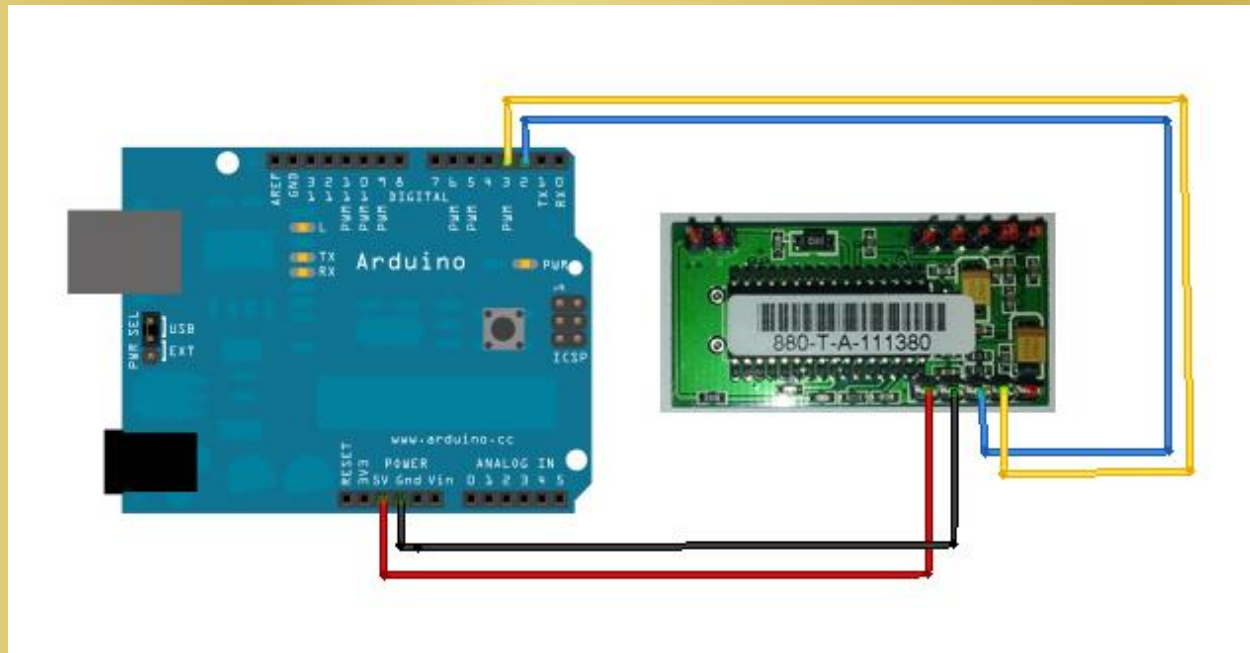
4 april 2013 – NIOC, <http://www.openarduino.nl>; E.H. van Tol-Homan

Onderwerp 1

- Eigen 'weerstation'
- Wifi
- Op het digibord wordt het zomerse weer bekeken:



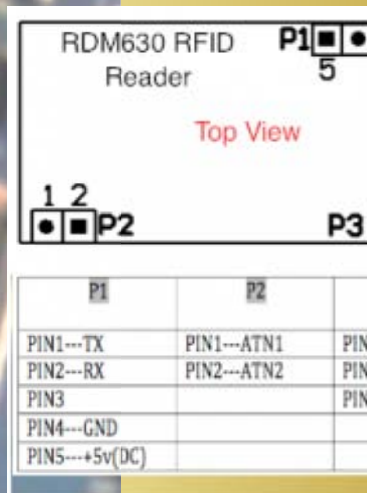
Onderwerp 2: OVChipkaart 13.56Mhz



Opdracht 3: Lilypad



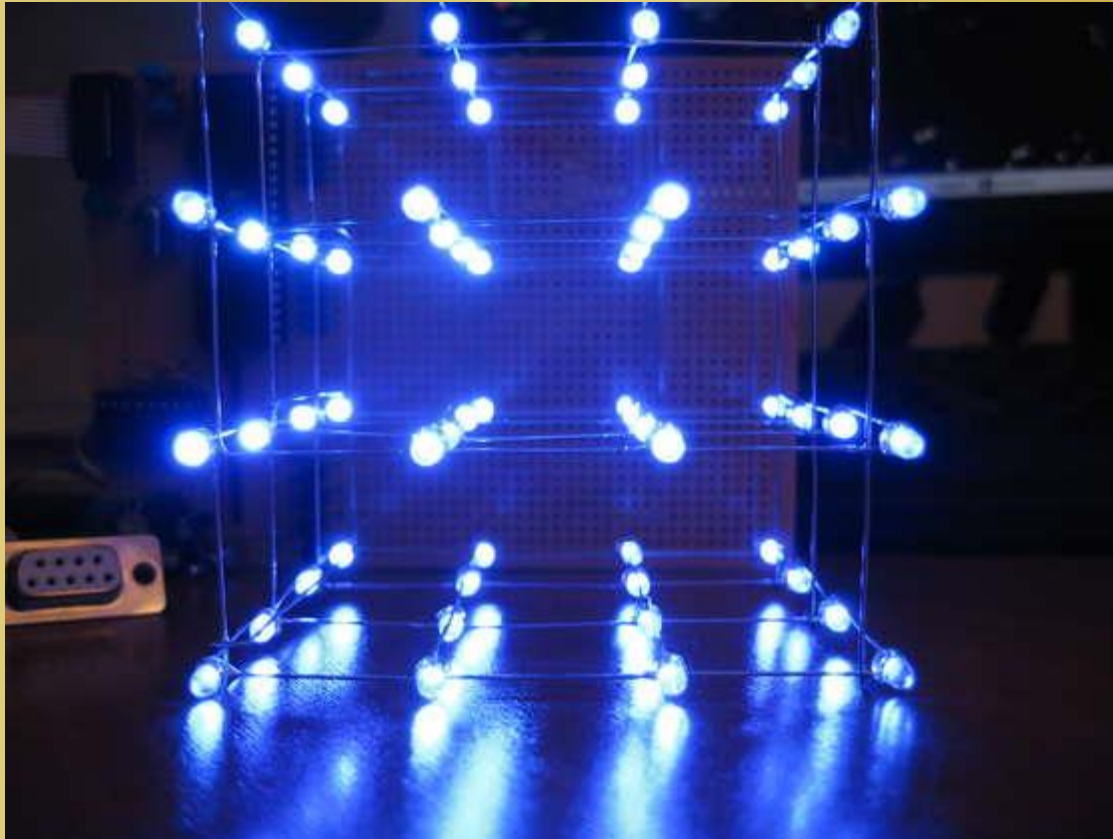
Opdracht 4, 6: RFID 125 kHz



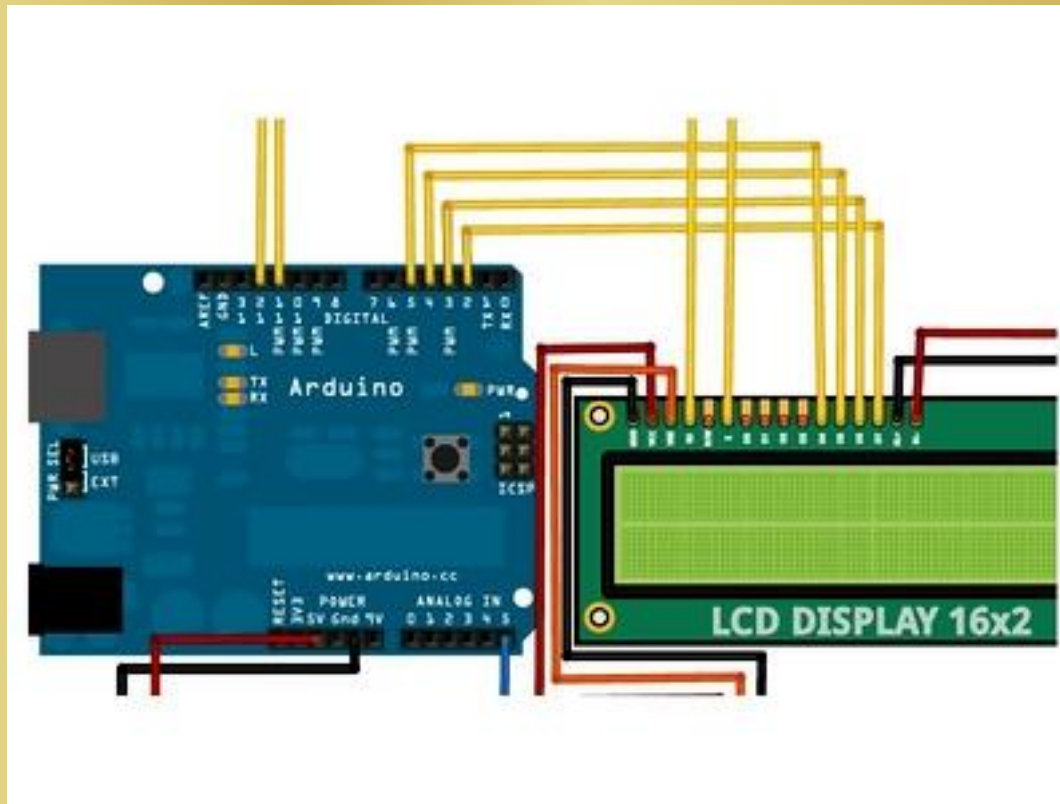
```
rfd | Arduino 1.0.1
Bestand Bewerken Sketch Extra Help
rfd Rfid_RDM630 $
http://creativecommons.org/licenses/by-nc-sa/3.0/
-----*/
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
char code[20];
int val = 0;
int bytesread = 0;
//-----
//create a Serial object RFID
SoftwareSerial RFID= SoftwareSerial(rxPin, txPin);

void setup()
{
  Serial.begin(9600);
  Serial.println("Serial Ready");
  RFID.begin(9600);
  Serial.println("RFID Ready");
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
}
}
```

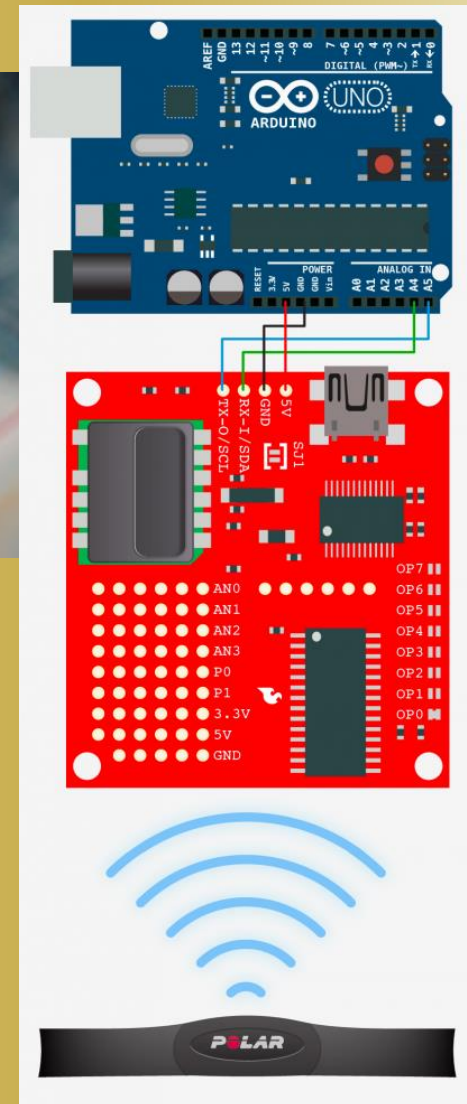
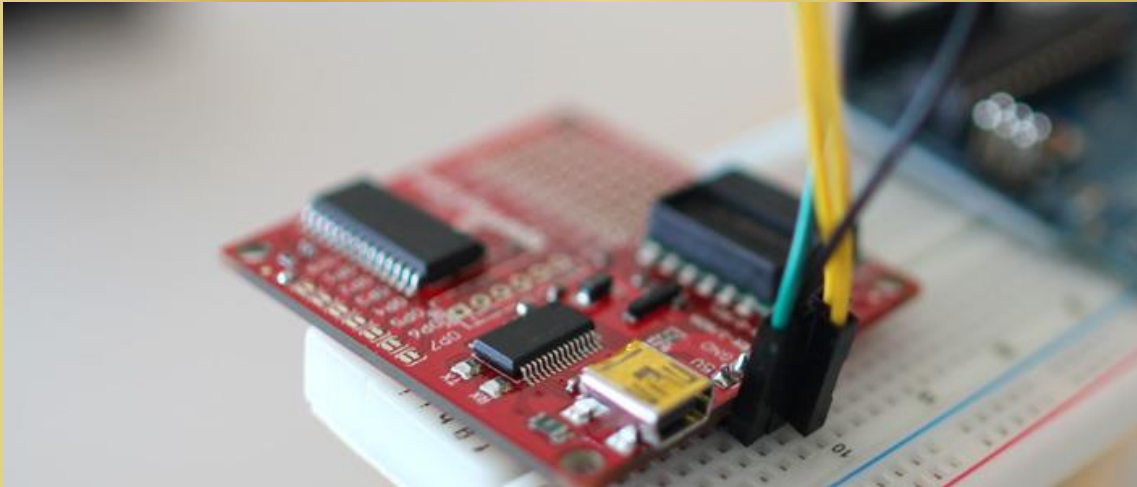
Opdracht 5: 3D cube



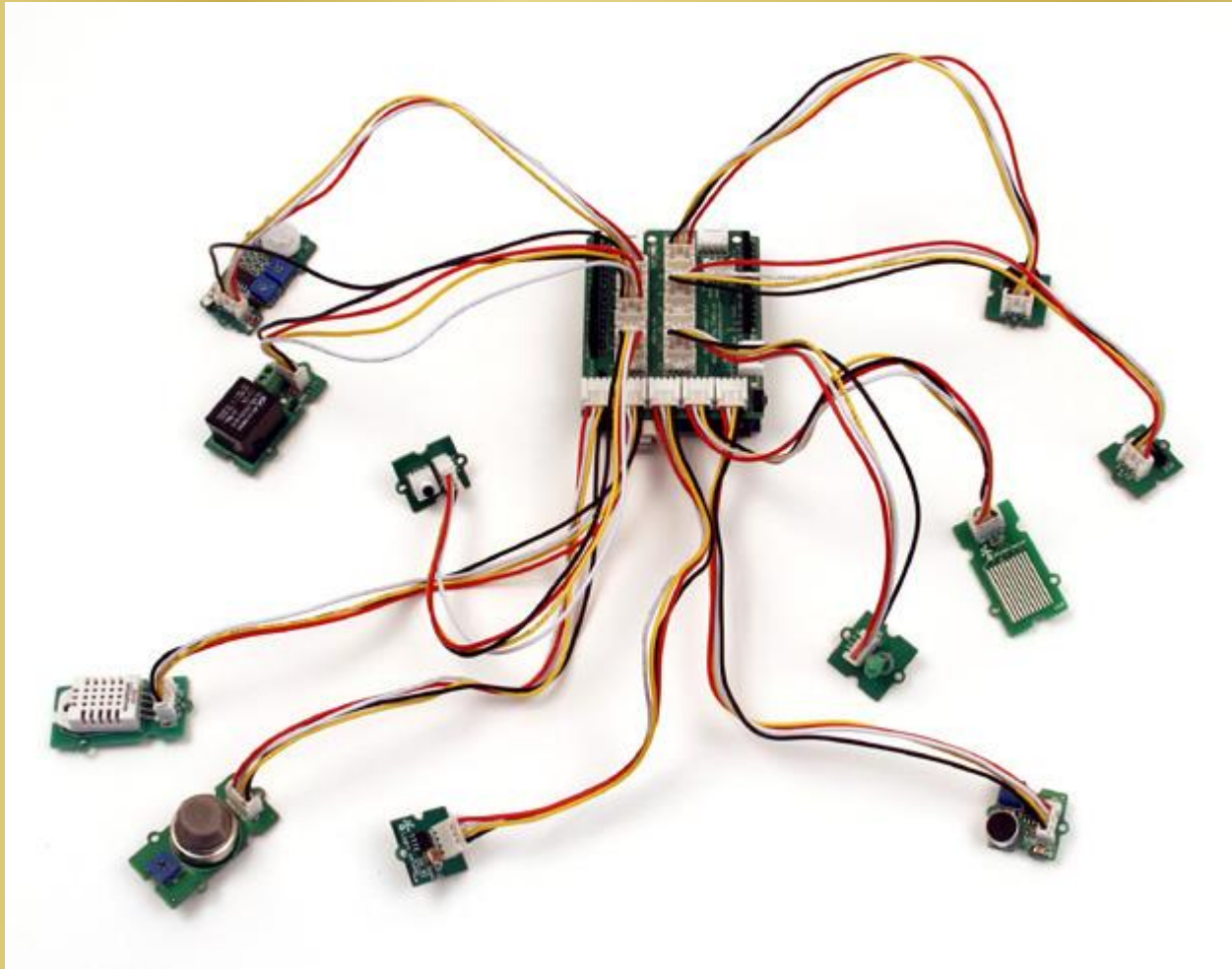
Opdracht 7: LCD



Opdracht 8: Hartslag



Opdracht 9: Grove



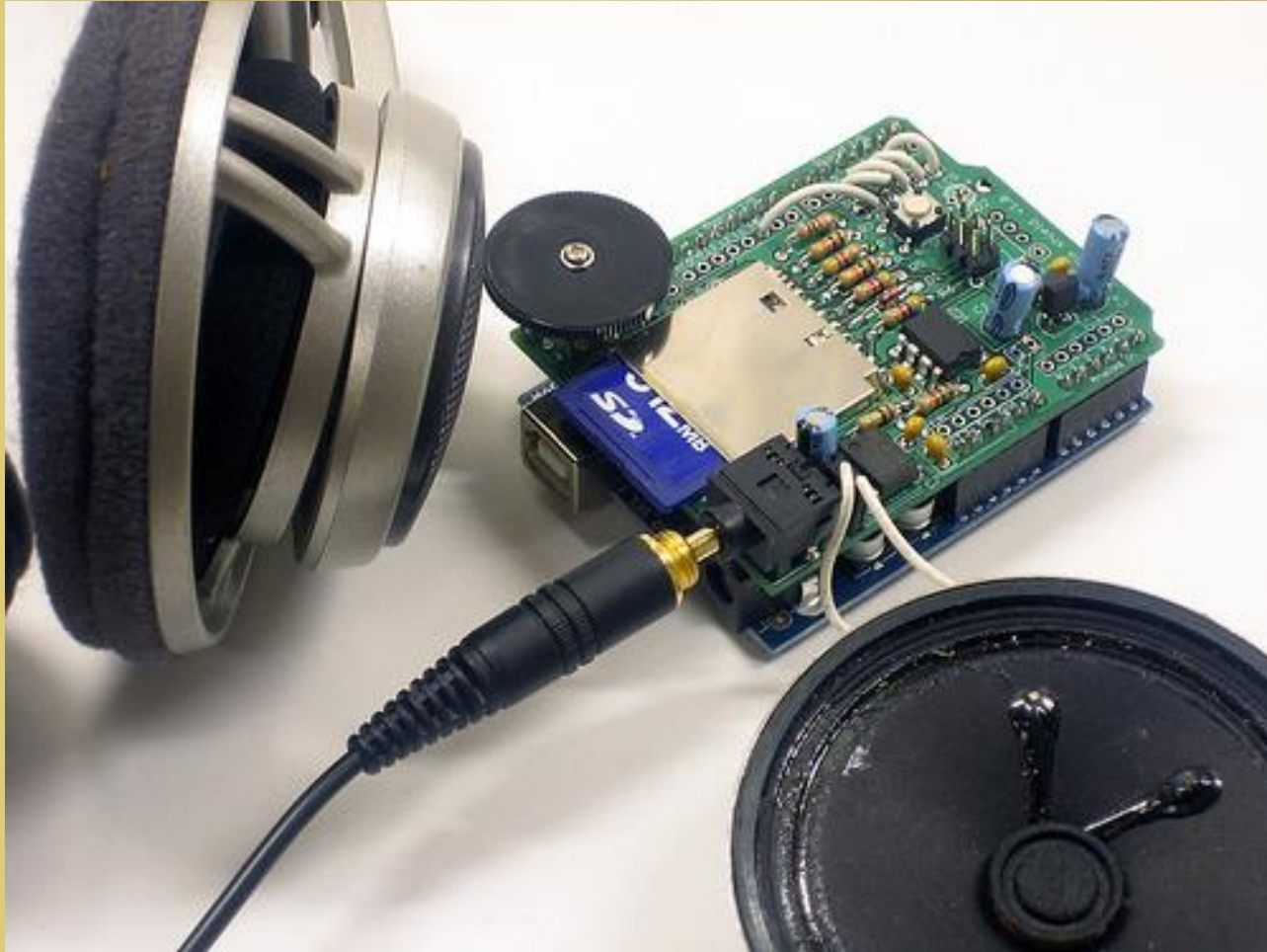
Opdracht 10,11: Robot



Opdracht 12 Deurbel, Xbee



Opdracht 13 Muziek



Opdracht 14: LOL



Opdracht 15-19



4 april 2013 – NIOC, <http://www.openarduino.nl>; E.H. van Tol-Homan

Opdracht 15-19



4 april 2013 – NIOC, <http://www.openarduino.nl>; E.H. van Tol-Homan

Onderwerp 19



```
sketch_121017a | Processing 2.0a5
File Edit Sketch Tools Help
sketch_121017a $
import processing.serial.*;
Serial port;

void setup() {
  size(256, 150);

  println("Available serial ports:");
  println(Serial.list());

  port = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  // draw a gradient from black to white
  for (int i = 0; i < 256; i++) {
    stroke(i);
    line(i, 0, i, 150);
  }

  // write the current X-position of the mouse to the serial port as
  port.write(mouseX);
}
```



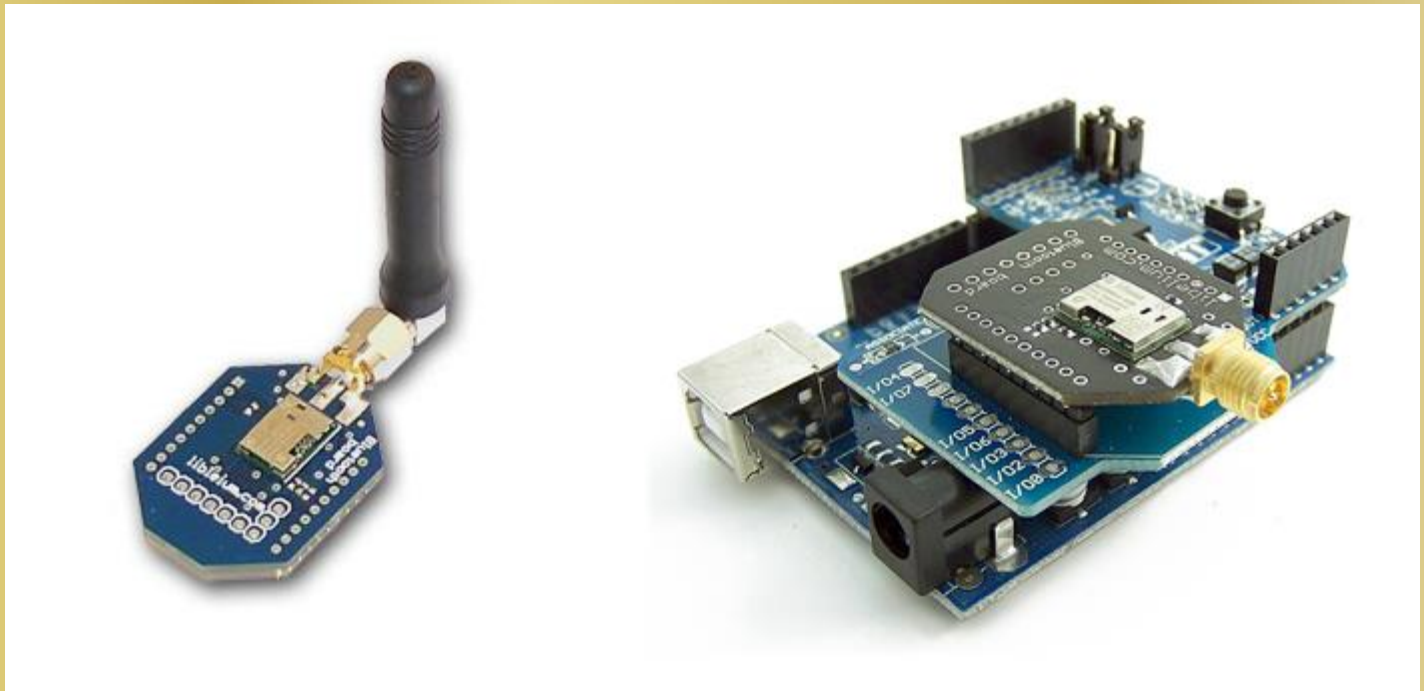
```
sketch_oct17a | Arduino 1.0.1
Bestand Bewerken Sketch Extra Help
sketch_oct17a $
void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  byte brightness;

  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    brightness = Serial.read();
    // set the brightness of the LED:
    analogWrite(ledPin, brightness);
  }
}

22 Arduino Mega 2560 or Mega ADK on COM16
```

Opdracht 20: Bluetooth



Opdracht 21: Android + Arduino ADK



A vertical strip on the left side of the slide shows a close-up of an Arduino circuit board. It features various components like a microcontroller, resistors, and a USB port, with a white infinity symbol printed on the board.

Arduino Workshop

Opdracht 1 tot met 21



Installatie

Installing drivers for the Arduino Duemilanove, Nano, or Diecimila with Windows7, Vista, or XP:

When you connect the board, Windows should initiate the driver installation process (if you haven't used the computer with an Arduino board before).

On Windows Vista, the driver should be automatically downloaded and installed. (Really, it works!)

On Windows XP, the Add New Hardware wizard will open:

- + When asked **Can Windows connect to Windows Update to search for software?** select **No, not this time**. Click next.
- + Select **Install from a list or specified location (Advanced)** and click next.
- + Make sure that **Search for the best driver in these locations** is checked; uncheck **Search removable media**; check **Include this location in the search** and browse to the **drivers/FTDI USB Drivers** directory of the Arduino distribution. (The latest version of the drivers can be found on the FTDI website.) Click next.
- + The wizard will search for the driver and then tell you that a "USB Serial Converter" was found. Click finish.
- + The new hardware wizard will appear again. Go through the same steps and select the same options and location to search. This time, a "USB Serial Port" will be found.

You can check that the drivers have been installed by opening the Windows Device Manager (in the Hardware tab of System control panel). Look for a "USB Serial Port" in the Ports section; that's the Arduino board.



Tenslotte

Syntax

- <http://www.planetb.ca/projects/syntaxHighlighter/>
- Layout:

```
01.  /*
02.   Blink
03.   Turns on an LED on for one second, then off for one second, repeatedly.
04.
05.   This example code is in the public domain.
06.  */
07.
08.  void setup() {
09.    // initialize the digital pin as an output.
10.    // Pin 13 has an LED connected on most Arduino boards:
11.    pinMode(13, OUTPUT);
12.  }
13.
14.  void loop() {
15.    digitalWrite(13, HIGH);  // set the LED on
16.    delay(1000);            // wait for a second
17.    digitalWrite(13, LOW);  // set the LED off
18.    delay(1000);            // wait for a second
19.  }
```

Maria Mole

- <http://dalpix.com/mariamole>

```
9
10 // Pin 13 has an LED connected on most Arduino boards.
11 // give it a name:
12 int led = 13;
13
14 // the setup routine runs once when you press reset:
15 void setup() {
16 // initialize the digital pin as an output.
17 pinMode(led, OUTPUT);
18 }
19
20 // the loop routine runs over and over again forever:
21 void loop() {
22 digitalWrite(led, HIGH); // turn the LED on (HIGH
23 delay(1000); // wait for a second
24 digitalWrite(led, LOW); // turn the LED off by ma
25 delay(1000); // wait for a second
26 }
27
```

Build Output Search 23:43

Project 'Blink' successfully built!
Binary file size: 1084 bytes (Uses 7% of the available memory at this board:14336 bytes)

Cursussen

- Voor scholen / docenten binnenkort de mogelijkheid om:
 - cursussen te volgen;
 - lesmateriaal te gebruiken;
 - arduino-sets te kopen;
- Zie: <http://www.openarduino.nl>

Website: OpenArduino.nl

Open Arduino | Arduino | x Nieuw tabblad x

www.openarduino.nl

open arduino
arduino in het onderwijs

home cursussen

Search Term Search

11011101100101
10110001001100
1000110010101000
110010001100101100
101111001011001011
10000110110001100001
10010100101101100101
01110111011001011101
01011000100110010110
01000110010101010001
01100100011001011001
01011110010110010111
10000110110001100001
10010100101101100101

Arduino in het onderwijs

Welkom op de website voor het gebruik van de Arduino in het Onderwijs.

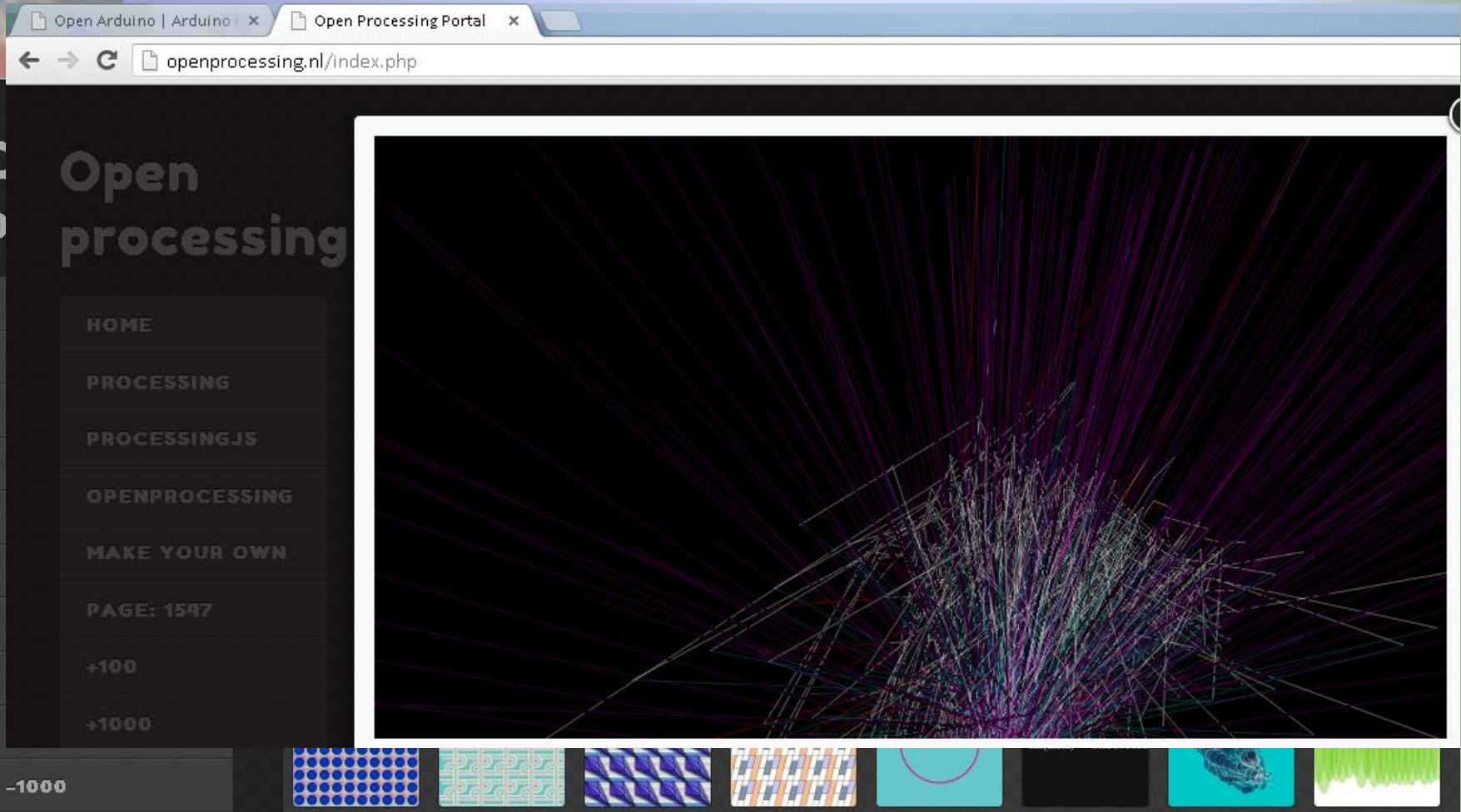
maart 26, 2013, 11:47 am
Arduino
0 Comments

RSS

categorieën
Categorie selecteren

meta
Registreren
Inloggen
Berichten RSS
Reacties RSS
WordPress.org

Website: *OpenProcessing.nl*





Vragen ???

Els.vantol@gmail.com