



## Stichting NIOC en de NIOC kennisbank

Stichting NIOC ([www.nioc.nl](http://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](http://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](mailto:kennisbank@nioc.nl).

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

# Wireless Sensing

## *Wireless Remote Sensing*

*Hoe kun je met goedkope elektronica een 'wireless remote sensing network' realiseren?*

*Door: Els van Tol-Homan, College Hageveld.*

*Trefwoorden: Arduino, wireless remote sensing.*

**Het masteronderzoek van Els van Tol-Homan vormde de basis van deze presentatie en draagt de titel: How to Generalize Wireless Remote Sensing Networking for Use-Case Independency. Sinds de opkomst van Arduino is het eenvoudig om eigen elektronica te ontwikkelen. U kunt bijvoorbeeld met een draadloze verbinding (Wifi of Zbee) een draadloos netwerk ontwerpen.**

Het wordt bijvoorbeeld toegepast in interactieve didactiek, het uitlezen van sensoren om fysische grootheden te meten of in de klas om Arduino te gebruiken in informaticaonderwijs. Tijdens deze presentatie en workshop kwam informatie aan bod over 'Wireless Remote Sensing'-technieken met behulp van Arduino en kregen de aanwezigen de mogelijkheid zelf een remote netwerk te bouwen.

### **Wireless Networks en Arduino.**

De workshop is een resultaat van de afstudeeropdracht "Wireless sensing networks with the Arduino" die is uitgevoerd in opdracht van Oracle. Deze bijdrage beschrijft de functies van "de Arduino" en een workshop voor VO-leerlingen in het vak Informatica om daarmee te leren werken en de mogelijkheden van 'sensing en wireless networks' praktisch te ervaren.

### **De Arduino**

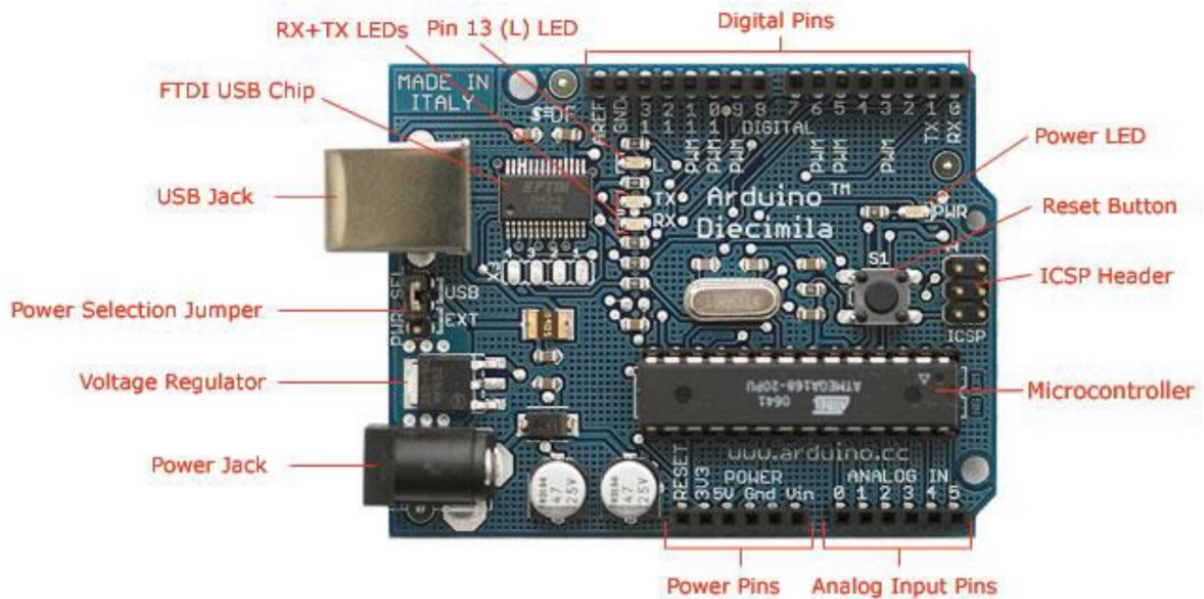
Arduino is een open source-computerplatform dat is opgebouwd rond de ATmega-microcontroller van Atmel en het softwareontwikkelplatform Processing. De Arduino kent afhankelijk van het type processor diverse uitvoeringen. De basisspecificaties van de ATmega328 zijn in figuur 1 beschreven.

| Kenmerk                     | Omschrijving   |
|-----------------------------|--|
| 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)                                     |
| Clock Speed                 | 16 Mhz   |

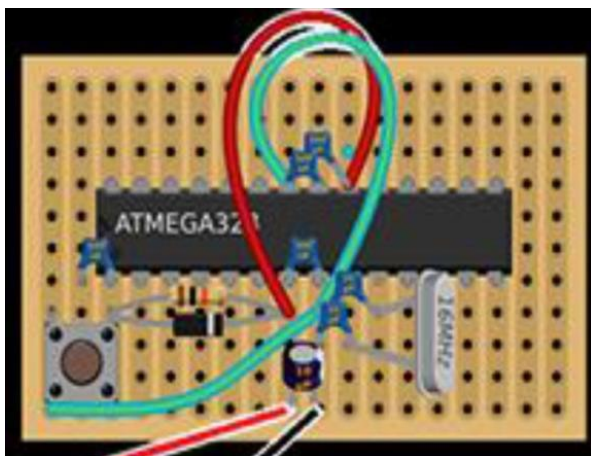
Figuur 1. Specificaties van de microcontroller ATmega328.

De microcontroller vormt de kern van het ontwikkelplatform en is op een board samengevoegd met elektronische componenten die de digitale en analoge functies voor invoer, uitvoer en wireless

communicatie realiseren (figuur 2). Figuur 3 toont een variant van een board waarop de microcontroller met een minimaal aantal componenten kan functioneren.



Figuur 2. Arduino-board (Sparkfun Electronics).



Figuur 3. Minimale configuratie voor functionele werking van de ATMEGA-microcontroller

De ontwikkeling van software voor het Arduino platform is mogelijk met diverse IDE's (Integrated Development Environment), ontwikkelsystemen voor de programmering van de microcontroller (figuur 4). Bij de Arduino worden diverse IDE's toegepast voor verschillende programmeertalen, zoals C++ of Java. De opensource IDE onder de naam 'Processing' voor de taal Java wordt veel toegepast in combinatie met de Arduino.

| Arduino IDE  | Processing IDE |
|--------------|----------------|
| C++          | Java           |
| Void setup() | Void setup()   |
| {}           | {}             |
| Void loop()  | Void draw()    |
| {}           | {}             |

Figuur 4. Programmaontwikkeling voor Arduino in C++ of Java.

```

/*
  Blink
  Turns on a LED for one second, the off for one second.
  This example code is in the public domain.
*/
// Pin 13 has a 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
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);    // turn the LED off
  delay(1000);                // wait for a second
}

```

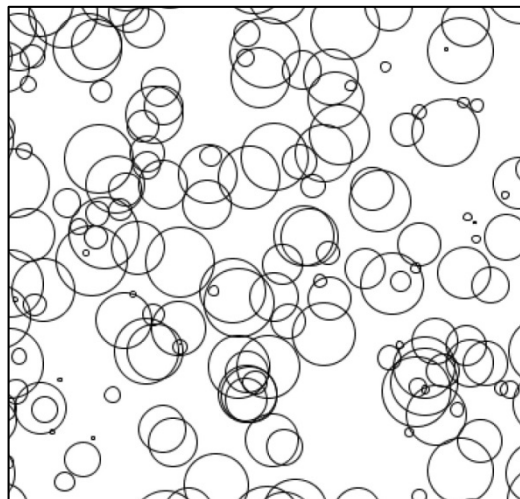
Figuur 5. Programmavoorbeeld 'knipperlicht'.

De Processing IDE biedt 'programmeerfuncties voor tekenen' en daarmee is onderwijs in inleidend programmeren goed mogelijk voor een brede doelgroep van leerlingen. De resultaten van de diverse opdrachten en veranderingen in de programmacode zijn snel concreet zichtbaar.

```

{
  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);
}

```



Figuur 6 programma 'cirkels' met schermresultaat.

Op de website van 'Openprocessing' zijn veel programmavoorbeelden te vinden die gebruik maken van de tekenfuncties van processing met vaak verrassend mooie voor de leerlingen aansprekende en uitdagende resultaten.

### De input

De invoer (input) van de Arduino is zowel digitaal als analoog mogelijk. De functie *analogRead()* heeft als omschrijving:

*Leest de waarde van de specifieke analoge pin. Het Arduino board heeft een 6 kanaals 10 bit converter (8 kanaals op de Mini en Nano en 16 kanaals op de Mega). Dat betekent dat het invoerspanningen tussen 0 en 5 Volt omzet in gehele waarden tussen 0 en 1023. Dat geeft een resolutie van 4,9 millivolt. Het input bereik en de resolutie kan worden aangepast door de functie *analogReference()*.*

De functie *digitalRead()* heeft als omschrijving:

*Leest de waarde van een specifieke digitale pin HIGH of LOW.*

```
// 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
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);
  }
}
```

Figuur 7 Voorbeeldprogramma digitale invoer en uitvoer.

### De output

De functie *analogWrite()* heeft als beschrijving:

Schrijft een analoge PWM-golf naar een pin. Kan worden gebruikt om een LED op variabele intensiteit te laten branden of een motor op verschillende snelheden te laten draaien. Na een aanroep *analogWrite()* zal de pin een blok golf genereren met de specifieke frequentie tot de volgende aanroep van *analogWrite()* (of een aanroep *digitalRead()* of *digitalWrite()* op dezelfde pin. De frequentie van het PWM-sigitaal is ongeveer 490 Hz.

De functie digitaleWrite() heeft als beschrijving:

Schrijft een HIGH of LOW waarde naar een digitale pin. Als de pin is geconfigureerd met een aanroep pinMode() als een OUTPUT, zal de spanning (voltage) worden gezet op de corresponderende waarde van 5 Volt voor HIGH en 0 Volt (aarde) voor LOW.

Als de pin is geconfigureerd als een INPUT, zal het schrijven van een HIGH waarde met digitalWrite() een interne 20K pullup weerstand activeren. Schrijven van LOW disabled de pullup weerstand.

```
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);
  }
}
```

Figuur 8. Programmavoorbeeld analoge uitvoer.

```
// include the library code
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interfacepins
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);
}
```

Figuur 9. Programmavoorbeeld bibliotheekgebruik en schermuitvoer.

## De wireless communicatie

Draadloze communicatie met Arduino is mogelijk via Wifi, Ethernet Shield, Xbee, Bluetooth en GSM.

Voor elke vorm is een specifieke component beschikbaar die op het Arduino board kan worden

aangesloten. Bij elke wireless toepassing wordt een eigen bibliotheek met programmafuncties geleverd voor de aansturing van de communicatie.

### **De Workshop**

In de workshop zijn diverse voorbeelden van opdrachten gepresenteerd en uitgevoerd.

1. Weerstation: Lees de gemeten temperatuur via wifi en presenteer die op het digibord in de klas.
2. OV-chipkaart: Analyseer en simuleer onderdelen van de werking van de OV-chipkaart.
3. Lilypad: LED-richtingaanwijzer op de rug van fietsers.
4. RFID: Analyseren van RFID-werking.
5. 3D-cube: aansturing van een 3-dimensionale LED-matrix.
6. RFID-lezer: Uitlezen van RFID-gegevens.
7. LCD: aansturing van een LCD-display.
8. Hartslag: Ontwikkel een hartslagmeter.
9. Grove: Sensoren en actuatoren.
10. Robot: Ontwikkeling van de aansturing van een robot
11. Robot: Zelfsturende (intelligentie) van een robot
12. Deurbel: ontwikkeling van wireless (Xbee) automatische identificerende deurbel.
13. Muziek: aansturing van luidsprekers met elektronische muziek.
14. LOL: plezier met aansturing van LED-arrays.
- 15 t/m 19: Zelfbouw, realisatie, testen en gebruiken van een Arduino testboard-kit.
20. Bluetooth: ontwikkeling van een Bluetooth toepassing.
21. Android en Arduino: Ontwikkeling van Arduino toepassing op Android (telefoon).

Op basis van de ervaringen met de workshops en de mogelijkheden van de diverse opdrachten worden cursussen, lesmaterialen en Arduino-sets aangeboden voor VO-scholen en VO-docenten.

Referenties.

[www.openarduino.nl](http://www.openarduino.nl);

[www.openprocessing.org](http://www.openprocessing.org)

Wilt u reageren op deze presentatie? Neem dan contact op met:

Els van Tol-Homan; docent informatica; College Hageveld

[evtol@hageveld.nl](mailto:evtol@hageveld.nl)