

Werkbladen voor Arduino workshop 2

In deze workshop zal gewerkt worden met de voorbeelden die meegeleverd worden met de Arduino IDE. Deze zijn te vinden onder het menu Bestand->Voorbeelden.

Oefening 1 - Seriële communicatie via USB

De Arduino heeft een seriële poort (UART), die beschikbaar is op I/O pinnen 0 en 1. Meestal worden deze pinnen niet gebruikt voor digitale I/O, omdat de seriële poort op de Arduino verbonden is met de USB naar serieel-omzetter. Je kan dit gebruiken om te communiceren met de Arduino via de PC.

Initialisatie:

Serial.begin(bits_per_sec);

Stelt de snelheid in van de seriële poort (in bits per seconde). 9600, 19200, 38400, 57600 en 115200 zijn veel gebruikte waarden. Het standaard formaat is 8N1.

Uitvoer:

Serial.print(waarde, formattering);

Serial.println(waarde, formattering);

Met print() kunnen we een waarde via de seriële poort printen
Met println() wordt tevens een nieuwe regel gegeven.

Met de formattering kunnen we bepalen hoe de waarde afgedrukt wordt:

```
Serial.print(78);           // print "78"  
Serial.print(1.23456);     // print "1.23"  
Serial.print('N');        // print "N"  
Serial.print("Hello, world."); // print "Hello, world."
```

Je kan ook een tweede parameter meegeven waarmee je het formaat bepaalt: voor gehele getallen is dit BIN, OCT, DEC of HEX. Voor kommagetallen bepaalt de tweede parameter hoeveel cijfers na de komma je wil zien.

```
Serial.print(78, BIN);      // print "1001110"  
Serial.print(78, OCT);     // print "116"  
Serial.print(78, DEC);     // print "78"  
Serial.print(78, HEX);     // print "4E"  
Serial.print(1.23456, 0);  // print "1"  
Serial.print(1.23456, 2);  // print "1.23"  
Serial.print(1.23456, 4);  // print "1.2346"
```

Invoer:

Serial.available()

Toont hoeveel karakters er ingelezen kunnen worden uit het seriële invoerbuffer.

Serial.read()

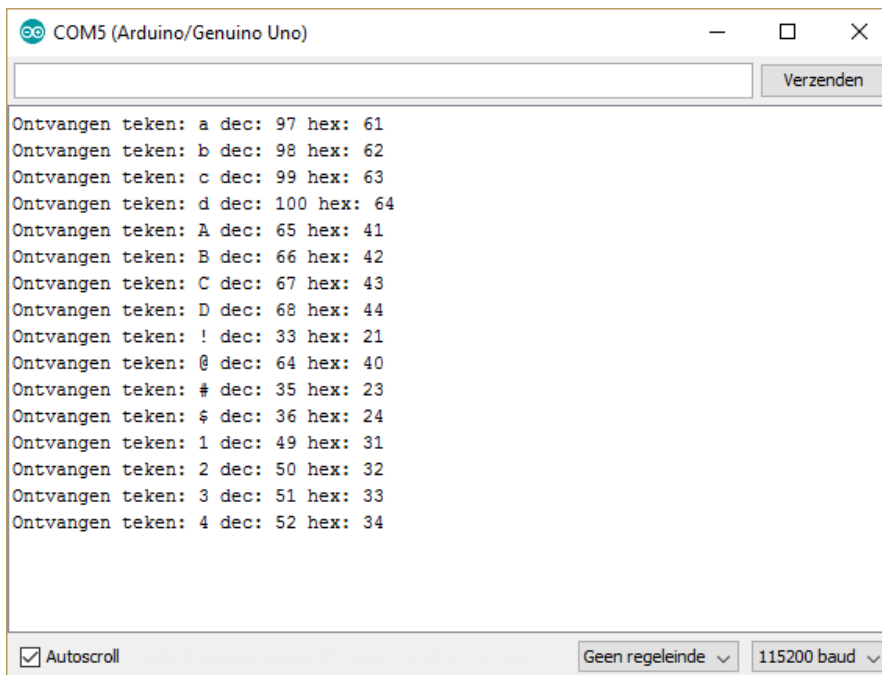
Leest een letter via de seriële poort.

```
#include <Wire.h>

char letterteken;           // binnenkomende seriële data

void setup() {
  Serial.begin(115200);     // open seriële poort op 115200 baud
}

void loop() {
  if (Serial.available() > 0) {
    letterteken = Serial.read(); // lees het ontvangen karakter
    Serial.print("Ontvangen teken: ");
    Serial.print(letterteken);   // Echo terug
    Serial.print(" dec: ");     // Spatie
    Serial.print(letterteken, DEC); // Echo terug
    Serial.print(" hex: ");     // Spatie
    Serial.println(letterteken, HEX); // Laat decimale waarde zien
  }
}
```



```
COM5 (Arduino/Genuino Uno)
```

Verzenden

Ontvangen teken: a dec: 97 hex: 61
Ontvangen teken: b dec: 98 hex: 62
Ontvangen teken: c dec: 99 hex: 63
Ontvangen teken: d dec: 100 hex: 64
Ontvangen teken: A dec: 65 hex: 41
Ontvangen teken: B dec: 66 hex: 42
Ontvangen teken: C dec: 67 hex: 43
Ontvangen teken: D dec: 68 hex: 44
Ontvangen teken: ! dec: 33 hex: 21
Ontvangen teken: @ dec: 64 hex: 40
Ontvangen teken: # dec: 35 hex: 23
Ontvangen teken: \$ dec: 36 hex: 24
Ontvangen teken: 1 dec: 49 hex: 31
Ontvangen teken: 2 dec: 50 hex: 32
Ontvangen teken: 3 dec: 51 hex: 33
Ontvangen teken: 4 dec: 52 hex: 34

Autoscroll Geen geregeleinde 115200 baud

Hello world

```
void setup()
{
  Serial.begin(9600); // Stel de seriële poort in
}

void loop()
{
  Serial.println("Hello world"); // Print "Hello world", en nieuwe regel
  delay(1000);
}
```

Invoer vanuit de PC

```
void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

int tijd = 500;
int input = 0;

void loop()
{
  if(Serial.available()>0)
  {
    input=Serial.parseInt();
    if(input!=0)
    {
      tijd = input;
    }
  }
  digitalWrite(13, HIGH);
  delay(tijd);
  digitalWrite(13, LOW);
  delay(tijd);
}
```

Opdracht1: maak een programma dat getallen inleest via de seriële poort, de som bijhoudt, en afdrukt.

Extra opdracht: maak een programma dat float getallen inleest via de seriële poort, de som bijhoudt, en afdrukt.

Oefening 2 - Analoge waarde inlezen (input)

analogRead(pin);

Meet de spanning op een analoge pin, en geeft de waarde als een getal tussen 0 en 1023 (10 bit ADC). De maximumwaarde wordt bereikt bij de referentiespanning, die normaal gezien 5V is maar ook extern kan aangereikt worden.

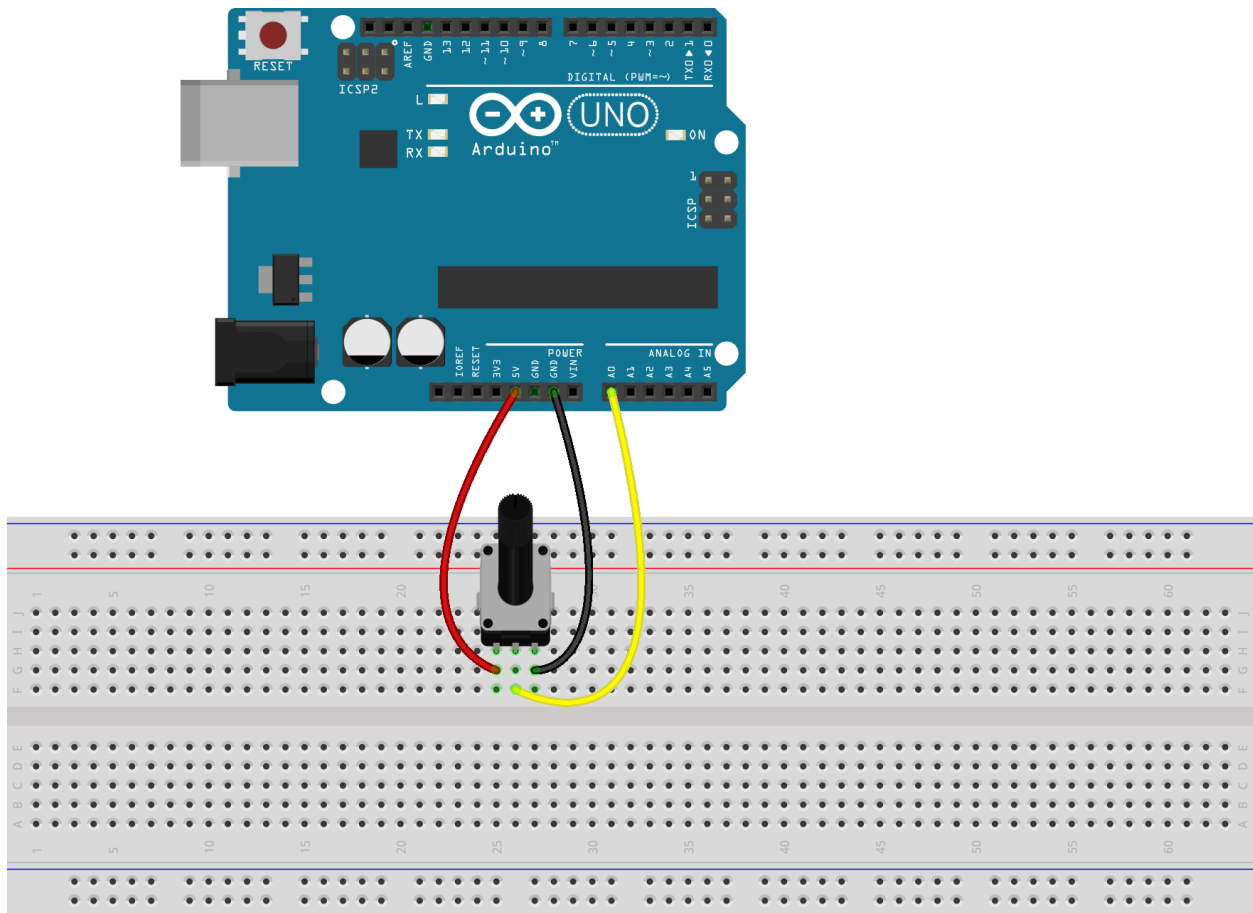
analogReference(type); // type: DEFAULT, INTERNAL of EXTERNAL

DEFAULT: de analoge spanning op de chip (5V op de UNO)

INTERNAL: interne 1.1V referentiespanning

EXTERNAL: een spanning die je zelf instelt via de pin AREF (0...5V max)

Opdracht 2a: Lees de stand van een potmeter in.



fritzing

Bouw de schakeling op zoals getoond, en laad het voorbeeld 03.Analog->AnalogInput
Pas de code aan zodat elke seconde de gemeten waarde op de seriële poort wordt geprint.
Bereken de spanning in Volt.

Opdracht 2b: Stel de interne referentiespanning in (pas ook de omzetting naar Volt aan).

Oefening 3 - Analoge waarde instellen (output)

AnalogWrite(pin, waarde);

De Arduino heeft geen analoge uitgangen, maar dat is voor veel toepassingen niet nodig. Door heel snel een digitale pin aan en uit te schakelen, kan je (gemiddeld) een gewenste spanning benaderen.

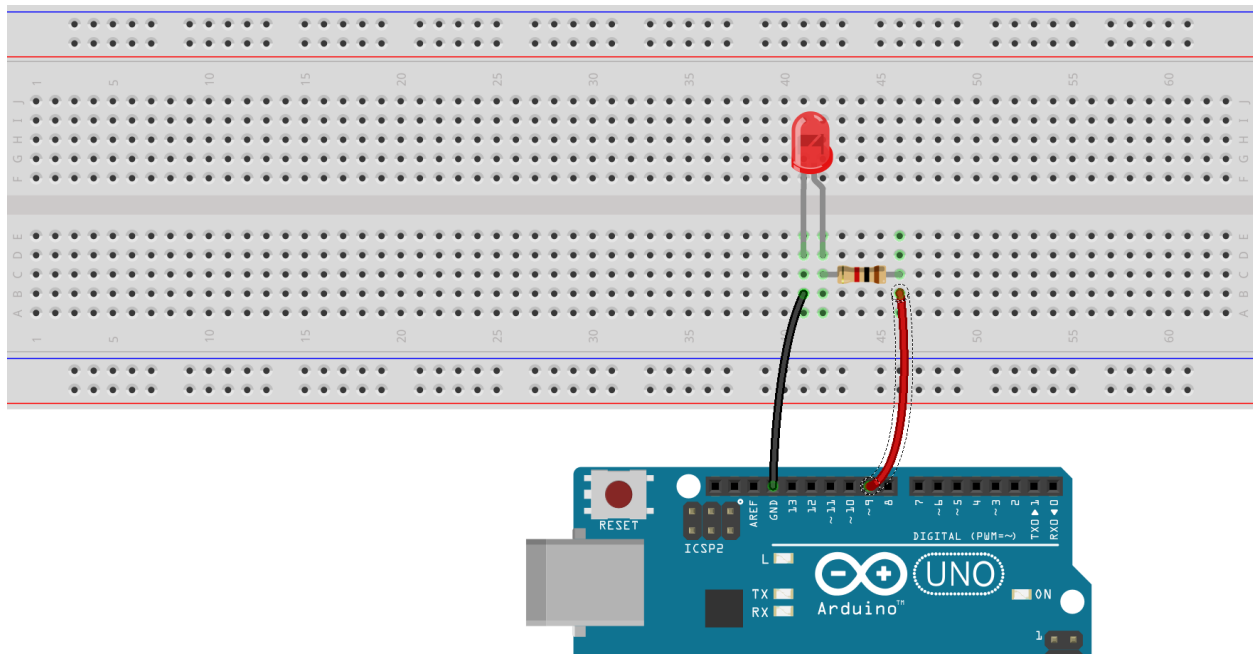
De waarde is een getal tussen 0 (0%, altijd uit) en 255 (100%, altijd aan).

Op de Arduino Uno kan dit alleen op de pinnen 3, 5, 6, 9, 10, en 11

Deze zijn aangeduid op het bordje met ~

Het is niet nodig de pin te configureren via pinMode !

Opdracht: Laat een LED langzaam aan- en uitgaan.



Bouw de schakeling op zoals getoond en laad het voorbeeld 03.Analog->Fading en upload het naar de Arduino.

Pas het programma als volgt aan:

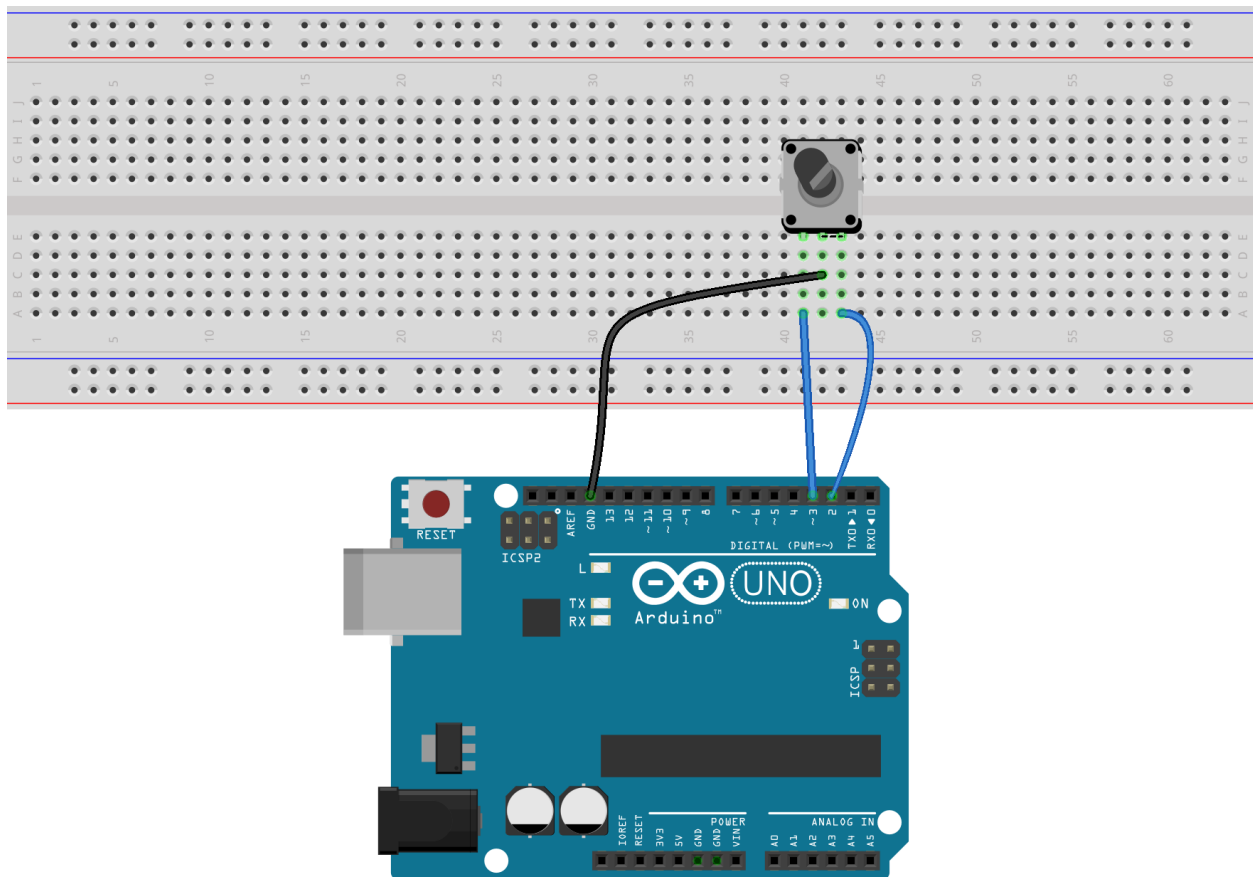
- 1) Laat het dimmen trager gaan.
- 2) Voeg een drukknop toe. Een druk op de knop laat de led langzaam aangaan. Een tweede druk op de knop laat de led langzaam uitdoven.

Oefening 4 - Rotary encoder

Er zijn oneindig veel sensoren, chips, displays enzovoort die je kan gebruiken in je Arduino projecten. Deze worden niet allemaal ondersteund door de standaard installatie, maar je kan wel op internet heel wat externe “Libraries” vinden. Als je een library installeert, worden meestal ook meteen enkele voorbeelden toegevoegd aan het “Examples” menu.

Encoder library: http://www.pjrc.com/teensy/td_libs_Encoder.html

- Deze functionaliteit gebruikt interrupt bij voorkeur interrupts.
Opgepast: op de Arduino Uno is dit alleen mogelijk op de pinnen 2 en 3.
- De gemeenschappelijke pin (de middelste van de encoder) wordt verbonden met GND (de library gebruikt de interne pull-up weerstanden).



fritzing

Opdracht: Laad Examples->Encoder->Basic in. Pas de code aan zodat pinnen 2 en 3 worden gebruikt. Lees de waarde uit op de seriële monitor

Extra:

Sluit een luidspreker aan en varieer de toonhoogte met de Rotary encoder.