

# ● Meten en sturen - Processing ●

Dit is een vervolg op Meten en sturen – Arduino (zie nr. 2024-2)

Henk Siewert

Processing is een programmeertaal, of als je dat liever zo noemt, een scripttaal, die gebaseerd is op Java. Behalve dat je er hele fraaie grafische dingen mee kunt doen, is de taal ook heel geschikt om meetgegevens mee weer te geven. He-las is dat laatste niet zo heel erg bekend. Maar daar gaan we verandering in aanbrengen.

## Ontstaan

Processing is in 2001 ontwikkeld door Ben Fry en Casey Reas aan het MIT (in Massachusetts). Het was de bedoeling om kunststudenten de mogelijkheid te bieden, op een niet al te ingewikkelde manier, programma's op de computer te maken om kunstwerken te genereren. Kenners van BASIC zullen misschien gelijk opveren en denken 'Ha, waar heb ik dat meer gehoord?'. Inderdaad, BASIC is ook ontworpen om studenten toegang te geven tot het maken van programma's op de toenmalige enorme mainframe computers. Eerst als een laag bovenop FORTRAN, als een soort preprocessor. En later als een zelfstandige programmeertaal.

Processing schijnt heel succesvol te zijn in de kunstwereld. Er worden zelfs tentoonstellingen georganiseerd in New York, van kunst die gemaakt is met behulp van Processing. Voorbeelden zijn te vinden op: <https://processing.org/examples> Maar wij gaan Processing op een hele andere manier gebruiken. De makers van Processing zijn namelijk zo verstandig geweest om er voor te zorgen dat de Processing-programma's ook verbinding kunnen maken met dingen buiten de computer waar het programma op draait. Dat kan onder andere via de seriële poorten van de computer. Kijk, en daar worden de gebruikers van microcontrollers vrolijk van. Want dat is nou precies waar zij ook gebruik van maken om met de computer te praten.

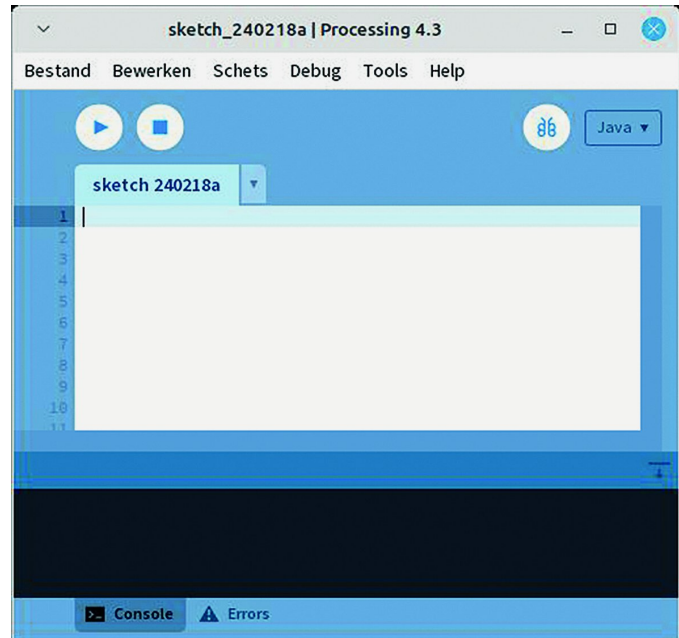
## Installeren

Zoals ik al eerder heb aangegeven werk ik voor dit project op een Linux Computer. Maar als je op Windows werkt, kun je Processing ook heel makkelijk downloaden en installeren. In Linux ga je naar <https://processing.org/> en haal je Processing op. Pak het uit en open in de directory waar je de bestanden in heb uitgepakt een terminalvenster. Ga naar de map ( directory ) en druk op F4.

Het terminal opent dan vanzelf. Typ dan: **sudo ./install.sh** Processing wordt dan geïnstalleerd en in het startmenu wordt dan een knop aangemaakt om het programma te kunnen starten. Meestal gebeurt dat onder het hoofdje 'Ontwikkeling'.

Als je Processing start, krijg je dit te zien: (zie afbeelding Processing IDE in de volgende kolom):

Wat zeg je? Lijkt wel erg veel op de Arduino IDE? Klopt. De Arduino IDE en de manier van coderen in de Arduino IDE zijn gebaseerd op de Processing IDE. Dat is toch leuk? Als je aan de ene gewend bent kun je ook overweg met de andere. Om eerlijk te zijn: er zit nog iets tussen. Voordat Arduino er was, was er Wiring. Wiring was een systeem en IDE voor het programmeren van de AVR-microcontrollers. Deze worden gebruikt op de Arduino-bordjes. Maar toen de Arduino populair werd, ging Wiring min of meer verloren. Er is een jarenlange strijd geweest tussen de maker van Wiring en de makers van Arduino. Ten slotte is enige tijd geleden de strijdbijl be-graven doordat de makers van de Arduino de maker van Wi-



Processing IDE

ring in dienst namen. Eind goed al goed... De Wiring-website bestaat nog steeds en je kunt Wiring nog steeds downloaden. Kijk maar eens op: <http://wiring.org.co/>

## Project 1

OK, terug naar ons potmeterproject. Het is de bedoeling een Processingprogramma te maken om de waarde die door de Arduino naar de seriële poort wordt gestuurd weer te geven in dat processingprogramma. Maar zoals gewoonlijk beginnen we eerst met elkaar gedag te zeggen.

```
/*
  Hallo World
  Auteur  : Henk Siewert
  Datum   : 18 februari 2024
  Sensor  : -
  Data input      : X\nl
  Taal           : Processing ( Java )
  Compiler       : Processing 4.3
  */

PFont font;           // Font object
String data = "Hallo World"; // Text to be printed

void setup()
{
  size(400,200);
  // Window size
  font = createFont("Arial Rounded MT Bold", 48 );
  // Create font
}

void draw()
```

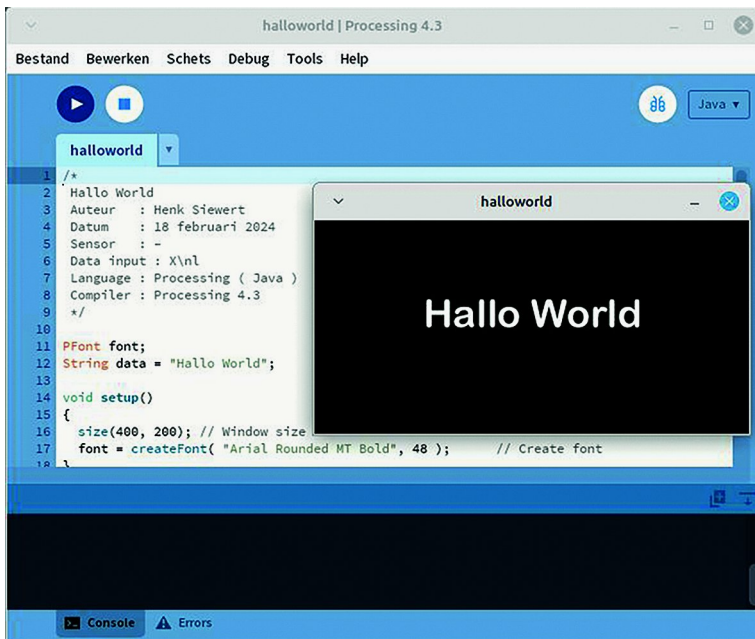
De code gaat verder op de volgende kolom >>

```
{
background( 0, 0, 0 ); // Window background color
fill( 255, 255, 255 ); // Text color
textFont( font, 36 ); // Text size
text( data, 100, 100 ); // Print text
}
```

Lijkt best wel veel op code voor de Arduino. Om te beginnen is een aantal duidelijke structuren te onderscheiden. Uiteraard `setup()`. Dat hebben we in de Arduino-code ook. Waar in de Arduino-code `loop()` staat gebruikt Processing `draw()`. De functie is identiek. Alles wat binnen `draw()` staat wordt herhaald uitgevoerd.

Er wordt begonnen met het aanmaken van een font object om met tekst te kunnen werken. In `setup()` wordt de grootte van het venster bepaald met `size(400, 200)`; waarna een font wordt aangemaakt waarmee de tekst in het venster kan worden weergegeven. Dat moet een bestaand systeemfont zijn. Het Windows Arial-font is ook onder Linux te installeren en te gebruiken. Xubuntu-gebruikers kunnen gebruik maken van het 'Lettertypen'-programma. Klik dan rechts op het te installeren TrueType bestand en klik op 'Open met Lettertypen'. Klik dan weer op de 'Installeren'-knop en klaar is Kees. Linux Mint-gebruikers zullen wat extra moeite moeten doen. Er staat op <https://www.fosslinux.com/104449/installing-and-using-custom-fonts-on-linux-mint.htm> een goede instructie om dat te doen. Lekker bakkie koffie er bij en dan lukt het vast wel.

In `draw` worden de kleuren bepaald en wordt de grootte van de tekst vastgelegd: `text( data, 100, 100 )`; bepaalt de plaats van de tekst in het venster en drukt de tekst af in het venster. Klik op het pijltje om het programma te starten. Als het goed gaat krijg je onderstaande te zien.



Processing IDE en Hallo World-venster

## Serieel

In het volgende programma ga je de waarde die door de Arduino op de seriële poort wordt gezet uitlezen en weergeven in een venster. Denk eraan om bij

```
String COMPOORT = "/dev/ttyACM0";
```

je eigen compoortnummer neer te zetten. Bij Windows gebruikers zal dat iets zijn als COM3. Ik heb zoveel mogelijk

commentaar gebruikt om duidelijk te maken wat er gebeurt. Sorry voor de mengeling van Engels en Nederlands. Maar het was een al bestaande code die ik verder heb aangevuld. Volgende keer alles in het Nederlands. Het commentaar dan...

```
/*
Serieel
Auteur   : Henk Siewert
Datum    : 18 februari 2024
Sensor   : -
Data input : X\nl
Taal     : Processing ( Java )
Compiler  : Processing 4.3
*/

// -----
// Global inits etc.
// -----

String COMPOORT = "/dev/ttyACM0"; // Hier uw eigen
                                   compoort invullen!!!
int SERIALSPEED = 9600;           // Serial port speed
PFont font;                       // Create space for
                                   font

import processing.serial.*;        // Seriele biblio-
                                   theek
Serial serialPort;                // Naam voor
                                   seriële poort

char BUFC = '\n';                 // Buffer end char
boolean gvalue = false;           // Flag value re-
                                   ceived
String data = "";                 // Data from serial
                                   string

void setup()
{
font = createFont( "Arial Rounded MT Bold", 32 );
// Font
serialPort = new Serial( this, COMPOORT, SERIALSPEED );
// Open serial port
serialPort.bufferUntil( BUFC );    // Buffer until \nl
size( 250, 130 );
}

// -----
// Main program
// -----
void draw()
{
if ( gvalue )                     // Is er een nieuwe waarde
                                   binnengekomen?
{
background ( 0, 70, 75 ); // Zet achtergrond op zwart

fill( 255, 255, 255 ); // Text color
textFont( font, 25 ); // Size of font
textAlign( CENTER );
textFont( font, 40 );

fill( 255, 255, 255 );
textFont( font, 15 );
textAlign( CENTER );

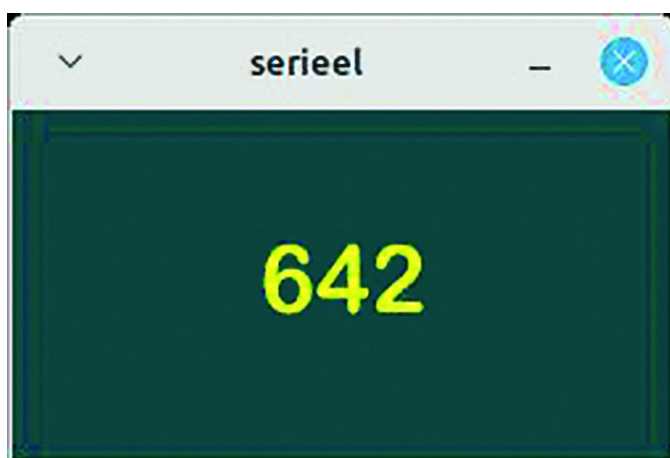
int ADCi = int( data ); // Convert string to integer
textAlign( CENTER );
fill( 255, 255, 0 );
textFont( font, 35 );
text( ADCi, 250 / 2, 75 );
}
```

De code gaat verder op de volgende pagina >>

```
data = "";           // Reset ADC sting
gvalue = false;     // Reset vlag
}
}

void serialEvent( Serial port )
{
  data = port.readStringUntil( BUFC ); // Read data in
                                     // serial buffer upto and including BUFC
  data = trim( data );                // Clean-up string
  gvalue = true;                      // Flag data received
}
```

Als alles het doet krijg je dit te zien:



*Processingprogramma weergave van de waarde op de seriële input*

Draai aan de potmeter en je ziet de waarde veranderen.  
Maximum 1023 en minimum 0. Probeer dat maar eens met  
een Raspberry Pi Pico.

Succes en veel plezier. Vergeet niet om op de website kijken:  
<http://winlin.atwebpages.com/>

Vragen en opmerkingen kun je kwijt op:  
[swtobserver@outlook.com](mailto:swtobserver@outlook.com).