Weekendschool – Programmeren – Les 3b – Programmeren van Arduino

# Wat gaan we doen?

In de vorige les heb je gezien dat je met programmeren ook kunt werken met knopjes en LEDjes. En je hebt gezien dat op het plankje met de eend ook een heel klein computertje zat: de Arduino Nano.

In deze les gaan we zo'n Arduino Nano programmeren. En wel zo, dat het plankje kan worden afgekoppeld en zonder de Raspberry Pi computer kan werken.

## Waarom geen Scratch?

In de vorige lessen hebben we met Scratch gewerkt als programmeertaal. Dat is een taal voor het onderwijs. Als je grote programma's wilt maken dan is het niet zo handig. Echte programmeurs gebruiken daarom meestal andere programmertalen. Heel populair is de taal C en C++. We gebruiken vandaag die talen, echter met wat beperkingen.

## Moeten we veel intikken?

In de vorige lessen hebben we het meeste met de muis gedaan. Het blijkt dat het voor programmeren toch meestal het snelste is om te tikken op het toetsenbord. Daarom gebruiken veel programmeurs programmeertalen waarbij het programma wordt ingetikt.

Echter, vandaag gebruik je veel voorbeelden en die kun je uit deze instructie met kopieren en plakken gewoon overzetten. Soms moet je – net als een een echte programmeur – wel dingen intikken.

## Waarom Arduino Nano?

Wat is het leuke aan de Arduino Nano? Hij is zo goedkoop dat jullie allemaal in het laatste deel van de les zelf zo'n plankje in elkaar mogen knutselen en dan het spel kunnen spelen dat je gaat programmeren. En je krijgt er naar keuze een batterij of een netvoeding bij zodat je het thuis kunt spelen zonder dat je verder iets nodig hebt! Maar alleen als je het leuk vindt om dat te doen en er thuis iets mee gaat doen. Want ookal is zo'n plankje niet zo duur, het is zonde als het bij het huisvuil belandt.

## En wat maken we dan?

We beginnen een spel te maken dat kijkt wie het vaakste op een knop kan drukken. We tellen het aantal keer van elke partij en laten dat zien.

Als er tijd is maken we de speeltijd instelbaar met een draaiknop.

En als er dan nog meer tijd over is, dan kun je hier het spel programmeren dat we vorige keer in Scratch gemaakt hebben: een rode LED gaat branden op willekeurige tijdstippen en degene die de eerste de knop indrukt krijgt een punt. Druk je op de knop terwijl de rode LED niet brandt, dan krijgt de ander een punt.

Met een draaiknop kunnen we de speeltijd instellen en ook de maximum tijd waarna de LED zal oplichten.

Dat lijkt heel veel, en dat is het ook. Maar je hoeft niet alles te maken om toch een leuk spel te hebben.

# Starten

## Geef je naam en emailadres

Als je het programma toegestuurd wilt krijgen, vergeet dan niet om je naam en emailadres op te geven. Ookal heb je dat vandaag al eerder gedaan, we gebruiken nu een andere Raspberry Pi en daarom moet het nog een keer.

## Openen van Arduino IDE

We beginnen met het openen van Les\_3b op het bureaublad. Dubbelklik deze map. Klik dan rechts op het bestand Les\_3b.ino en kies Arduino.

Je ziet dat de Arduino IDE geopend wordt. IDE staat voor Integrated Development Environment. Dat betekent *geintegreerde ontwikkelomgeving*.

En er opent een blanko blad. Hier gaan we het programma schrijven.

# Hebben we haast?

Deze instructie is aardig dik, maar er zitten dan ook extra opdrachten in voor als je heel snel klaar bent. Werk zorgvuldig en werk gewoon goed door. Maar veel belangrijker dan hoever je komt is dat je snapt wat je doet. Dus als er iets is wat je niet snapt, overleg dan met elkaar. En je kunt natuurlijk ook je begeleider vragen om het uit te leggen.

Verder: dit is de eerste keer dat we deze les doen, dus het is wat experimenteren hoe snel jullie zullen gaan.

## Wat je moet snappen

Er staat uitleg die je niet perse hoeft te weten, maar het geeft verdere uitleg.

Wat je wel echt moet weten staat op een groene achtergrond, net zoals deze regel.

# Beginnen met programmeren

Elke regel in het programma bevat een opdracht die wordt uitgevoerd. Er zijn 3 grote delen van het programma waar zulke regels in staan:

1. De declaraties. Hier staan definities die voor het hele programma gelden. Bijvoorbeeld op welke pennen de knoppen en LED's zijn aangesloten.
2. setup(): Het stukje programma dat eenmalig wordt uitgevoerd als de Arduino Nano start. Het stukje programma staat tussen accoulades { }.
3. loop(): Het stukje programma dat na setup() wordt uitgevoerd. Ook dit staat tussen accoulades.

En daarna kunnen nog stukjes programma komen die we kunnen gebruiken in het hoofdprogramma. Dat zien we straks.

## Declaraties

Zie het gele blok hieronder.

Alles na // tot het einde van de regel is kommentaar. Dat dient alleen om duidelijk te maken wat er gebeurt. Het is heel verstandig om als je aan het programmeren bent hier te noteren wat er wordt gedaan en waarom, want als je na een paar maanden naar het programma kijkt dan snap je er vaak niets meer van. En dat geldt ook voor iemand anders die er naar kijkt.

#define betekent dat overal waar het woord voorkomt dit wordt vervangen door wat er achter staat, tot het einde van de regel of het begin van het kommentaar.

Dus, bijvoorbeeld overal waar in het programma staat KnopLinks zal dan 3 worden ingevuld.

Dit is het eerste stel regels van het programma. Programmaregels staan in deze instructie altijd op een gele ondergrond. Je kunt dit blok tekst selecteren en dan kopieren met Ctrl-C. Ga dan naar het venster van de Arduino IDE en tik Ctrl-V. Je kunt ook de rechter knop van de muis gebruiken.

// Pennen van de Arduino. We geven ze een naam zodat we in het programma een naam kunnen

// gebruiken (dat is makkelijker) en omdat we dan, als we een wijziging moeten maken

// (bijvoorbeeld een knop op een andere pin aansluiten), slechts op 1 plek een wijziging

// hoeven te maken.

// Knoppen

#define KnopLinks 3 // De linker knop zit op pen 3

#define KnopRechts 12 // De rechter knop zit op pen 12

#define KnopKlein 4 // De kleine knop linksonder zit op pen 4

// LEDs

#define LEDGroenLinks 2 // De groene LED links is aangesloten op pen 2

#define LEDGroenRechts 11 // De rechter groene LED is aangesloten op pen 11

#define LEDRood 5 // De rode LED zit op pen 5

// Analoge ingangen

#define PotMeter A0 // De potentiometer (potmeter) is een regelbare

// weerstand. Die is aangesloten op pen A0

// De 4 cijfers worden bestuurd door een chip TM1637 die aan de onderkant zit

// (je kunt het transparante plaatje omdraaien en dan zie je de chip met het nummer.

// Je ziet dan ook dat er 4 pennen zijn: GND, VCC, CLK en DIO.

// VCC en GND zijn de draden voor de voeding en die gaan naar resp. +5 Volt en GND op

// de Arduino zodat de chip en het blok cijfers spanning krijgen.

// DIO staat voor Digitale Input / Output. Over deze draad gaat de data die vertelt welke

// cijfers moeten oplichten.

// CLK staat voor klok (clock in het Engels). Die vertelt wanneer de TM1637 chip op de DIO draad de data moet bekijken.

#define CLK 8 // Dit is de klok

#define DIO 9 // Dit is de digitale input / output

## setup()

Dit stuk van het programma wordt uitgevoerd als de Arduino Nano wordt gestart.

Kopieer het hele blok en plak het in het venster van de Arduino IDE. Doet dat achter het blok met declaraties wat er al staat.

Dit stuk programma vertelt hoe de pennen moeten werken. Is het voor invoer of uitvoer?

// ==========================================================================================

// Wat tussen de accoulades achter setup staat wordt uitgevoerd nadat de Arduino is gestart

void setup()

{ // Dit is openingsaccoulade die bij setup hoort.

pinMode(KnopLinks, INPUT\_PULLUP); // De pen waaraan de linker knop zit is voor invoer

// (input).

// En er moet een optrekweerstand (pullup) worden

// geconfigureerd om te zorgen dat

// er een 5 volt signaal (HIGH) op de pen staat als de

// knop NIET wordt ingedrukt.

// Als er op de knop wordt gedrukt komt er 0 volt (LOW)

// op.

pinMode(KnopRechts, INPUT\_PULLUP); // Idem voor de rechter knop

pinMode(KnopKlein, INPUT\_PULLUP); // Idem voor de kleine knop links onder

pinMode(LEDGroenLinks, OUTPUT); // De pen waaraan de linker groene LED zit is een

// uitvoer (output).

pinMode(LEDGroenRechts, OUTPUT); // Idem voor de rechter groene LED.

pinMode(LEDRood, OUTPUT); // Idem voor de rode LED.

Serial.begin(115200); // Deze opdracht is nodig om straks te kunnen zien op

// de Raspberry Pi wat de Arduino aan het doen is.

// 115200 wil zeggen dat de communicatie zal gaan met

// 115200 bit/seconde.

// Hoe je op de Raspberry Pi kunt zien wat de Arduino

// doet komt later.

} // Dit is de sluitaccoulade die bij setup hoort.

// Dit sluit de rij opdrachten af die bij

// het starten van de Arduino worden uitgevoerd.

## loop()

Tussen de accoulades na loop() komt het programma dat wordt uitgevoerd na de setup().

// ==========================================================================================

// Wat tussen de accoulades staat wordt uitgevoerd na setup()

void loop()

{

digitalWrite(LEDGroenLinks, HIGH); // Laat de linker LED branden

delay(1000); // Wacht 1000 milliseconde (ms) = 1 seconde

digitalWrite(LEDGroenLinks, LOW); // Doe de linker LED weer uit

delay(1000); // Wacht 1000 milliseconde (ms) = 1 seconde

}

En als alle opdrachtregels zijn uitgevoerd, dan start het gewoon weer van voren af aan bij de eerste regel in loop().

Plak nu dit stukje programma aan het eind.

Wat je hier ziet is zijn deze opdrachten:

digitalWrite De eerste keer wordt de pen HIGH (hoog) gemaakt, dus een hoge spanning (5 volt). Daardoor gaat de LED branden.   
De tweede keer maken we de pen LOW (laag), dus een lage spanning (0 volt). Dan gaat de LED weer uit.

delay Delay betekent vertraag. In dit geval: wacht het aantal milliseconde dat tussen haakjes staat.

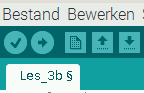
## Uittesten

We gaan nu het programma laden in de Arduino. Maar eerst moet het programma worden vertaald in een code die de Arduino begrijpt. Dat heet ***compileren***.

Controleer dat er rechtsonder op het scherm dit staat:  


Als er iets anders staat, vraag dan even je begeleider.

Het programma noemt Arduino een sketch. Sketch is het engelse woord voor *schets*.

Druk nu hier om te compileren en naar de Arduino Nano te *uploaden*.

## Werkt het?

Als het goed is gegaan, dan knippert het linker groene LEDje. Werkt het niet? Kijk dan even goed. Werkt het nog steeds niet? Vraag dan je begeleider.