**ARDUINO**

**ADVANCED**

Jeroen van Hattem

**Voordat we kunnen beginnen met de workshop moet je de Arduino IDE gedownload hebben. Dit is de omgeving die we gaan gebruiken om de Arduino te programmeren.**

# Intro

Bij deze workshop wordt er van je verwacht dat je al enige achtergrond kennis hebt met programmeren.

Als je nog geen kennis hebt van de Arduino IDE, raad ik je aan hoofdstuk 2 te lezen. Als je hier al bekend mee bent, kan je dit overslaan.

# 1. Wat gaan we maken?

Vandaag gaan we een bestuurbare auto maken met een bijbehorende controller. Dit alles gaan we zelf programmeren. De auto en de controller zullen samen communiceren via infrarood. Voor de auto maken we gebruik van de L9110 H-Bridge om de motoren aan te sturen, hier is op internet voorbeeld code voor te vinden, indien nodig. Ook gebruiken we een infrarood ontvanger, de XXX, in combinatie met het RC-5 protocol. Hier bestaan library’s voor, om dit nu helemaal vanaf scratch te maken hebben we niet genoeg tijd.

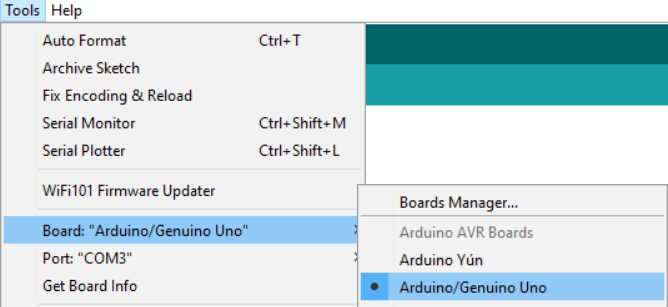
Voor de controller gebruiken we een eenvoudig infrarood ledje, in combinatie met het RC-5 protocol. Dit kan je eenvoudig programmeren als je de goede library gebruikt. In hoofdstuk 3 gaan we beginnen met hoe we onderdelen van de auto programmeren. In hoofdstuk XXX gaan we beginnen aan de controller.

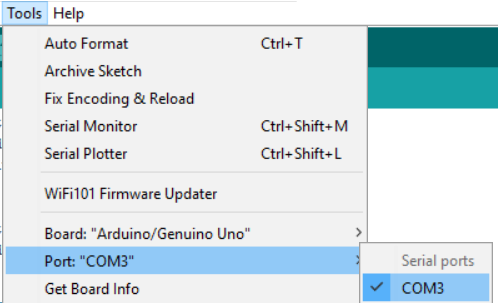
# 2. Hoe werkt de IDE?

Voordat we kunnen programmeren hebben we een IDE nodig, dit is een omgeving waar code kan worden geschreven en kan worden geüpload naar het Arduino bordje. Wij maken gebruik van de Arduino IDE.

De betekenis van de knoppen is als volgt:  
 controleren of jouw sketch (programma) correct is  
 uploaden van jouw sketch naar de Arduino  
 nieuwe sketch aanmaken  
 opslaan van jouw sketch  
 open de seriële monitor om de uitvoer van de Arduino te bekijken

Voordat we kunnen uploaden moeten we de instellingen goed zetten.

Ga naar “Tools” -> “Board” -> “Arduino / Genuino Uno”, dit doen we omdat er verschillende Arduino boards zijn. Deze optie selecteert de goede opties die nodig zijn bij het compilen en uploaden van de code.



Ga daarna naar “Tools” -> “Port”, kies hier de COM-port uit die er staat. De COM-port is het USB-adres waar de code naar moet worden geüpload.

Wanneer we alles hebben ingesteld kunnen we een stuk code schrijven, laten we beginnen met een eenvoudige code, genaamd “Hello World”. Dit is in veel programmeertalen de eenvoudigste manier om je code te testen, dit bevat ook de absolute basis van een project.

void setup() {

Serial.begin(9600);

}

void loop() {

Serial.println("Hello world");

delay(10 \* 1000);

}

Wanneer we de “Serial Monitor” openen, zullen we zien dat daar om de 10 seconden “Hello World” wordt geprint. Dit betekent dat de Arduino functioneert naar behoren.

# 3. Aansturen van de motoren

We gaan hier een stukje code schrijven waarmee je een ledje kan aansturen.

## 3.1 De opstelling

Voordat we de Arduino gaan programmeren, gaan we de hardware aansluiten.

We hebben voor dit stukje een ledje, een breadboard, een weerstand en wat kabeltjes nodig.

We doen de led in het insteekbord. De led heeft 2 pins, een korte pin en een lange gebogen pin. Op de korte pin sluiten we de “ground” aan, aangegeven met “GND” op de Arduino. Op de lange pin sluiten we, via een weerstand, een digitale pin aan, “D5” in dit geval. Deze gebruiken we als output signaal.

Wanneer je dit allemaal aansluit. Zorg ervoor dat de weerstand goed is aangesloten.

## 3.2 De code

const int buttonPin = 7;

const int ledPin = 5;

int buttonValue;

void setup() {

pinMode(buttonPin, INPUT\_PULLUP);

pinMode(ledPin, OUTPUT);

}

void loop() {

digitalWrite(ledPin, HIGH);

}

Als we deze code kopiëren naar de Arduino IDE kunnen we de code uittesten. Wanneer de code in het code vlak staat, drukken we op  om de code te valideren. Wanneer dit allemaal goed is drukken we op  om de code te uploaden naar de Arduino.

## 3.3 Code uitleg

const int buttonPin = 7;

We maken een constante variabele aan die het nummer van de pin bevat. Op pin 7 sluiten we het knopje aan. “const” houdt in dat de variabele niet meer veranderd kan worden. Als er geen “const” staat kan de variabele later in de code nog veranderd worden.

pinMode(buttonPin, INPUT\_PULLUP);

Hiermee wordt de gegeven pin, in dit geval “buttonPin”, geconfigureerd als een input of output. Mogelijke opties hier zijn “INPUT”, “OUTPUT” en “INPUT\_PULLUP”

digitalWrite(ledPin, HIGH);

Hier sturen we een pin aan. In dit geval sturen we een hoog signaal naar “ledPin”. Mogelijke opties hier zijn “HIGH”, “TRUE”, of “1” en “LOW”, “FALSE”, of “0”.

## 3.4 Testen

Wanneer we het programma runnen, gaat het lampje aan.

# 4. Data lezen van de infrarood ontvanger

We gaan hier een stukje code schrijven waarmee we een ledje kunnen laten knipperen.

## 4.1 De opstelling

Voordat we de Arduino gaan programmeren, gaan we de hardware aansluiten.

We hebben voor dit stukje een ledje, een breadboard, een weerstand en wat kabeltjes nodig.

We doen de led in het insteekbord. De led heeft 2 pins, een korte pin en een lange gebogen pin. Op de korte pin sluiten we de “ground” aan, aangegeven met “GND” op de Arduino. Op de lange pin sluiten we, via een weerstand, een digitale pin aan, “D5” in dit geval. Deze gebruiken we als output signaal.

Wanneer je dit allemaal aansluit. Zorg ervoor dat de weerstand goed is aangesloten.

## 4.2 De code

const int buttonPin = 7;

const int ledPin = 5;

int buttonValue;

void setup() {

pinMode(buttonPin, INPUT\_PULLUP);

pinMode(ledPin, OUTPUT);

}

void loop() {

digitalWrite(ledPin, HIGH);

delay(100);

digitalWrite(ledPin, LOW);

delay(100);

}

Als we deze code kopiëren naar de Arduino IDE kunnen we de code uittesten. Wanneer de code in het code vlak staat, drukken we op  om de code te valideren. Wanneer dit allemaal goed is drukken we op  om de code te uploaden naar de Arduino.

## 4.3 Code uitleg

delay(100);

De “delay()” functie zorgt ervoor dat het programma een bepaalde tijd wacht. In dit geval moet het programma 100ms wachten voordat het ledje van status kan worden veranderd.

## 4.4 Testen

Wanneer we het programma runnen zien we dat het ledje knippert om de 100ms.

# 5.Data lezen van de joystick

We gaan hier een stukje code schrijven waarmee we de data van de joystick kunnen uitlezen.

De joystick werkt analoog en stuurt een X en Y waarde door. We hebben hiervoor 2 pins nodig.

## 5.1 De opstelling

We bouwen vanaf hier verder vanaf de vorige opstelling. Hier voegen we een knopje aan toe. We gaan een opstelling maken waarbij het ledje gaat branden wanneer we het knopje indrukken.

We doen de led in het insteekbord. De led heeft 2 pins, een korte pin en een lange gebogen pin. Op de korte pin sluiten we de “ground” aan, aangegeven met “GND” op de Arduino. Op de lange pin sluiten we, via een weerstand, een digitale pin aan, “D5” in dit geval. Deze gebruiken we als output signaal.

Ook moeten we het knopje aansluiten, deze heeft 4 pins. 2x een ground pin en 2x een signaal pin. We sluiten de pins aan zoals aangegeven op het plaatje. Het oranje kabeltje gaat naar “D7”.

## 5.2 De code

const int buttonPin = 7;

const int ledPin = 5;

int buttonValue;

void setup() {

pinMode(buttonPin, INPUT\_PULLUP);

pinMode(ledPin, OUTPUT);

}

void loop() {

buttonValue = digitalRead(buttonPin);

if (buttonValue) {

digitalWrite(ledPin, HIGH);

}

else {

digitalWrite(ledPin, LOW);

}

}

Als we deze code kopiëren naar de Arduino IDE kunnen we de code uittesten. Wanneer de code in het code vlak staat, drukken we op  om de code te valideren. Wanneer dit allemaal goed is drukken we op  om de code te uploaden naar de Arduino.

## 5.3 Code uitleg

buttonValue = digitalRead(buttonPin);

Hier vragen we door middel van “digitalRead(buttonPin)” de waarde van “buttonPin” op. Deze waarde kennen we dan toe aan “buttonValue”.

if(buttonValue){} / else{}

Hier kijken we of “buttonValue” hoog is. Als dit het geval is zal de code die binnen de krulhaken staat uitgevoerd worden. Als dit niet het geval is wordt de code die binnen de krulhaken van de “else” uitgevoerd.

## 5.4 Testen

Wanneer we nu op de knop op het insteekbordje drukken zien we dat het ledje aan gaat.

# 6. Aansturen van de infrarood verzender

We gaan nu een reactiespelletje maken voor 2 spelers. Op het breadboard bevinden zich 2 knopjes. Wanneer het ledje aangaat moeten de spelers zo snel mogelijk op hun knopje drukken. De winnaar wordt dan met zijn tijd geprint op de “Serial Monitor” op de computer.

# 6.1 De opstelling

Voor het reactiespelletje moeten we een extra knopje toevoegen aan de vorige opstelling.

We doen de led in het insteekbord. De led heeft 2 pins, een korte pin en een lange gebogen pin. Op de korte pin sluiten we de “ground” aan, aangegeven met “GND” op de Arduino. Op de lange pin sluiten we, via een weerstand, een digitale pin aan, “D5” in dit geval. Deze gebruiken we als output signaal.

Ook moeten we de knopjes aansluiten, deze heeft 4 pins. 2x een ground pin en 2x een signaal pin. We sluiten de pins aan zoals aangegeven op het plaatje. Het oranje kabeltje gaat naar “D7”. Bij het tweede knopje doen we hetzelfde, alleen sluiten we deze aan op “D6”.

## 6.2 De code

const int firstButton = 7;

const int secondButton = 6;

const int led = 5;

int timer = 0;

void setup() {

Serial.begin(9600);

pinMode(firstButton, INPUT\_PULLUP);

pinMode(secondButton, INPUT\_PULLUP);

pinMode(led, OUTPUT);

digitalWrite(led, LOW);

Serial.println("Wacht tot de LED brandt en druk dan snel op de knop!");

}

void loop() {

delay(random(400, 2000));

digitalWrite(led, HIGH);

Serial.println("GO");

while((digitalRead(firstButton) == HIGH) && (digitalRead(secondButton) == HIGH)) {

delay(1);

timer++;

}

if(digitalRead(firstButton) == LOW) {

Serial.println("De winnaar is gebruiker 1: ");

Serial.println(timer);

Serial.println("ms");

digitalWrite(led, LOW);

timer = 0;

while(1);

}

else if(digitalRead(secondButton) == LOW) {

Serial.println("De winnaar is gebruiker 2: ");

Serial.println(timer);

Serial.println("ms");

digitalWrite(led, LOW);

timer = 0;

while(1);

}

}

Als we deze code kopiëren naar de Arduino IDE kunnen we de code uittesten. Wanneer de code in het code vlak staat, drukken we op  om de code te valideren. Wanneer dit allemaal goed is drukken we op  om de code te uploaden naar de Arduino. Als dit gedaan is drukken we op om de seriële monitor te openen om de uitvoer van de Arduino te bekijken. Hier moeten we de baud rate nog aanpassen naar “9600”

## 6.3 Code uitleg

Serial.begin(9600);

Begin een seriële connectie tussen de computer en de Arduino. Hierover kan informatie worden uitgewisseld. De parameter (het getal tussen de haakjes) is de “baud rate”, dit is het aantal bits die per seconde worden uitgewisseld. Deze moet op de “Serial Monitor” evenveel zijn als in de code.

Serial.println("Leuke tekst hier!");

Met deze code kan de Arduino tekst weergeven in de “Serial Monitor”. Dit kan ook worden gebruikt om variabelen weer te geven, zoals je reactietijd.

delay(random(400, 2000));

Hier gebruiken we 2 functies, de “delay()” functie en de “random()” functie. De “random()” functie genereert een willekeurig geval, in dit geval tussen 400 en 2000. Dit getal wordt dan teruggegeven en ingevuld in de “delay()” functie. De “delay()” functie geeft aan dat de Arduino een bepaalde tijd moet wachten. Als de “random()” functie 800 zou teruggeven, zou het programma 800ms moeten wachten voordat het verder kan gaan.

digitalRead(firstButton);

Deze functie leest de waarde in van de gegeven pin, in dit geval “firstButton”. Deze functie geeft een “HIGH” of “LOW” waarde terug. Vervolgens kan je deze waarde vergelijken door middel van “==”, dit is te zien in de voorbeeldcode. Let op! De buttons geven een omgekeerd signaal om. Dus “HIGH” is “LOW” en “LOW” is “HIGH”.

timer++;

Dit is hetzelfde als “timer = timer + 1;”

## 6.4 Testen

Wanneer we de code hebben geüpload zien we het ledje na een tijdje oplichten. Wanneer een knopje wordt ingedrukt, zien we in de “Serial Monitor” de winnaar staan.

# 7. Zelf aan de slag

In de doos vind je nog een ledje. Nu heb je een groen en een rood ledje. Zorg ervoor dat de rode led aanstaat wanneer de gebruiker moet wachten om te drukken. Wanneer het spel is begonnen en de gebruiker op het knopje moet klikken moet het rode ledje uitgaan en het groene ledje aan.

Denk hierbij aan het aansluiten van de weerstand.

# Nawoord

## Studievereniging

Deze workshop is gegeven in samenwerking met de studievereniging Indicium**.** Deze studievereniging is speciaal voor studenten van de opleiding HBO-ICT. Wanneer je lid bent van deze studievereniging heb je toegang

## Thuis verder aan de slag met een Arduino?

Als je het werken met een Arduino leuk hebt gevonden en je zou thuis verder willen gaan met het programmeren van een microcontroller, dan zou je kunnen overwegen om zelf een Arduino Uno aan te schaffen.

Officiële Arduino’s zijn te koop op veel plekken in Nederland, waaronder Conrad. Mocht je budget wat kleiner zijn, kan je overwegen om een Arduino “clone” te halen. Deze bordjes werken hetzelfde, maar komen niet van Arduino zelf.

Heb je toch geen zin om een Arduino te halen? Op [circuits.io](http://circuits.io) kan je online een Arduino simuleren. Als je problemen hebt met een code kan je hier ook kijken of het aan jouw Arduino ligt of aan je code.