

# Arduino Workshop

## Inleiding

Welkom bij de Arduino Workshop voor beginners! In deze workshop zul je kennismaken met de basisprincipes van Arduino en hands-on ervaring opdoen met het bouwen van schakelingen en het programmeren van een Arduino Uno R3. Laten we beginnen!

## Benodigde Materialen

- 1x Arduino Uno R3
- 1x Breadboard
- 1x LED
- 1x Drukknop
- 1x 10kΩ weerstand (Kleurcode: Bruin-Zwart-Oranje)
- 1x 1kΩ weerstand (Kleurcode: Bruin-Zwart-Rood)
- Jumper draden

## 1. Basisconcepten van Arduino

Arduino is een **microcontroller**. Dit is een kleine slimme chip die we kunnen vertellen wat te doen en die ons in staat stelt om allerlei leuke dingen te maken met elektronica en code!

### Wist je dat:

Arduino een fabrikant is van een ontwikkelboardje waarop een microcontroller aanwezig is. Er zijn nog veel meer andere microcontrollers en ontwikkelboardjes van andere fabrikanten die je OOK in je kit hebt zitten. Zelfs met WiFi en Bluetooth! (ESP32)

Een Arduino heeft diverse mogelijkheden om componenten op aan te sluiten. Dit worden **GPIO's (General Purpose Input Output)** genoemd. Hierop kunnen we componenten aansluiten en code matig van alles mee doen.

Soms hebben we ook enkel alleen een spanningsbron nodig. Het Arduino ontwikkelboardje heeft een 5 Volt(+), 3,3 Volt(+) en diverse GND (-) aansluitingen beschikbaar gesteld waarmee we iets constant mee kunnen voorzien van spanning. Denk hierbij aan sensoren chips en andere componenten.

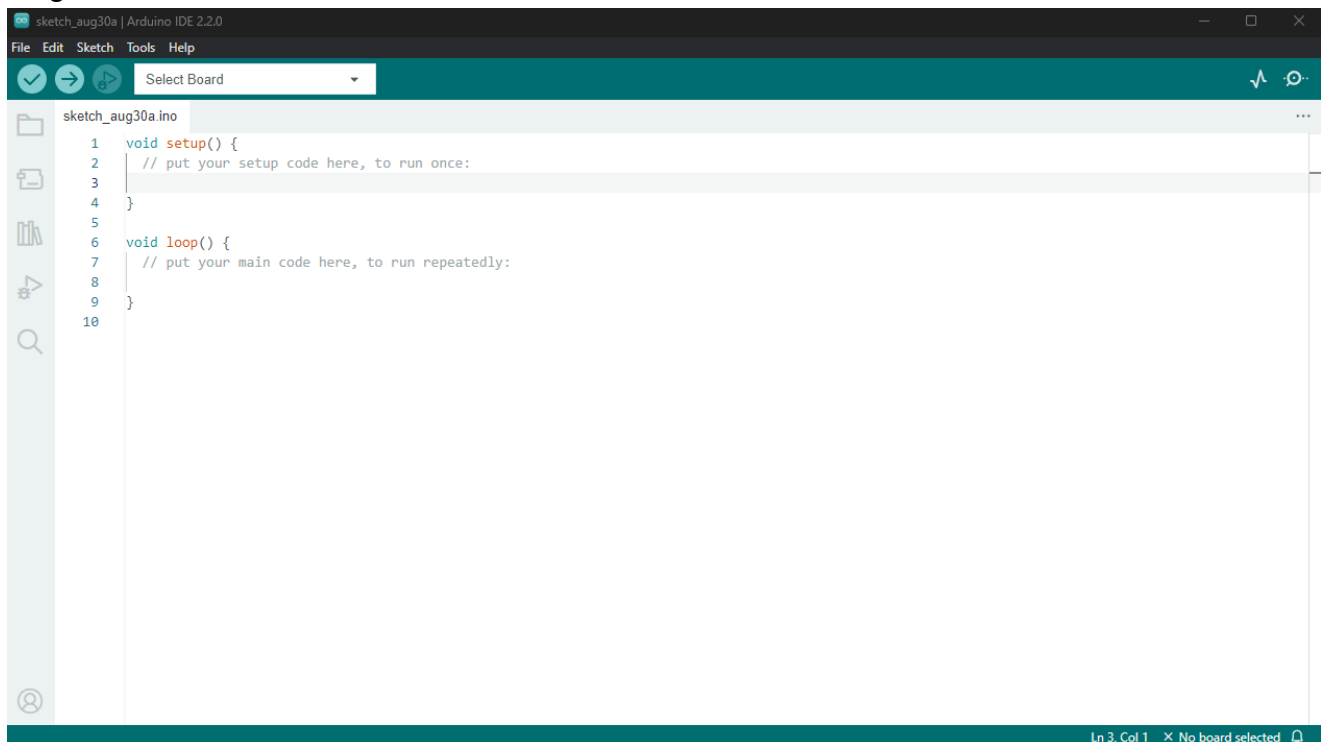
## 2. Installeren van de Arduino IDE

Om iets met GPIO's te kunnen doen moeten we natuurlijk programmeren en vervolgens dit kunnen sturen naar de microcontroller. Om het makkelijk te maken hebben we een omgeving nodig waarin we kunnen coderen en onze code kunnen uploaden naar de microcontroller. Deze omgeving heet voor Arduino ontwikkelbordjes de **Arduino IDE**. Dit is de software omgeving die op je computer moet worden geïnstalleerd.

### Psst... kom je er niet uit?

Hier heb je een eenvoudige tutorial: [Installatie documentatie](#)

Als je het goed hebt gedaan kan je de Arduino IDE openen en wordt je begroet met het volgende scherm:



Zoals je ziet staat er al wat code voor je klaar. Wat je ziet zijn twee **"functies"**. **Functies zijn een lijst van instructies die kunnen worden uitgevoerd**. de namen van onze functies zijn **"setup"** en **"loop"**. Als bijschrift staat in de functie beschreven dat de setup één keer wordt uitgevoerd en de loop blijft herhalen.

### LETOP!

De setup wordt als eerste uitgevoerd en daarna pas de loop.

### 3. Je eerste Arduino-code!

Nu je weet hoe Arduino werkt, kunnen we beginnen met het schrijven van je eerste programma. Laten we eerst eens een LED (Lampje) knipperen. Hiervoor kunnen we de volgende code gebruiken:

```
int LED_GPIO = 13;

void setup() {
  pinMode(LED_GPIO, OUTPUT);
}

void loop() {
  digitalWrite(LED_GPIO, HIGH);
  delay(1000);
  digitalWrite(LED_GPIO, LOW);
  delay(1000);
}
```

Deze Arduino-code laat een LED op pin 13 (deze is al op de Arduino aanwezig) aan en uit knipperen met een tussenpozen van 1 seconde.

#### 1. Instellingen (setup):

- De `LED_GPIO` wordt gedefinieerd en toegewezen met waarde 13, wat de pin is die de LED aanstuurt.
- De `setup`-functie initialiseert de pin met `pinMode(LED_GPIO, OUTPUT)` zodat de LED kan worden aangestuurd als uitgang.

#### 2. Herhalende handelingen (loop):

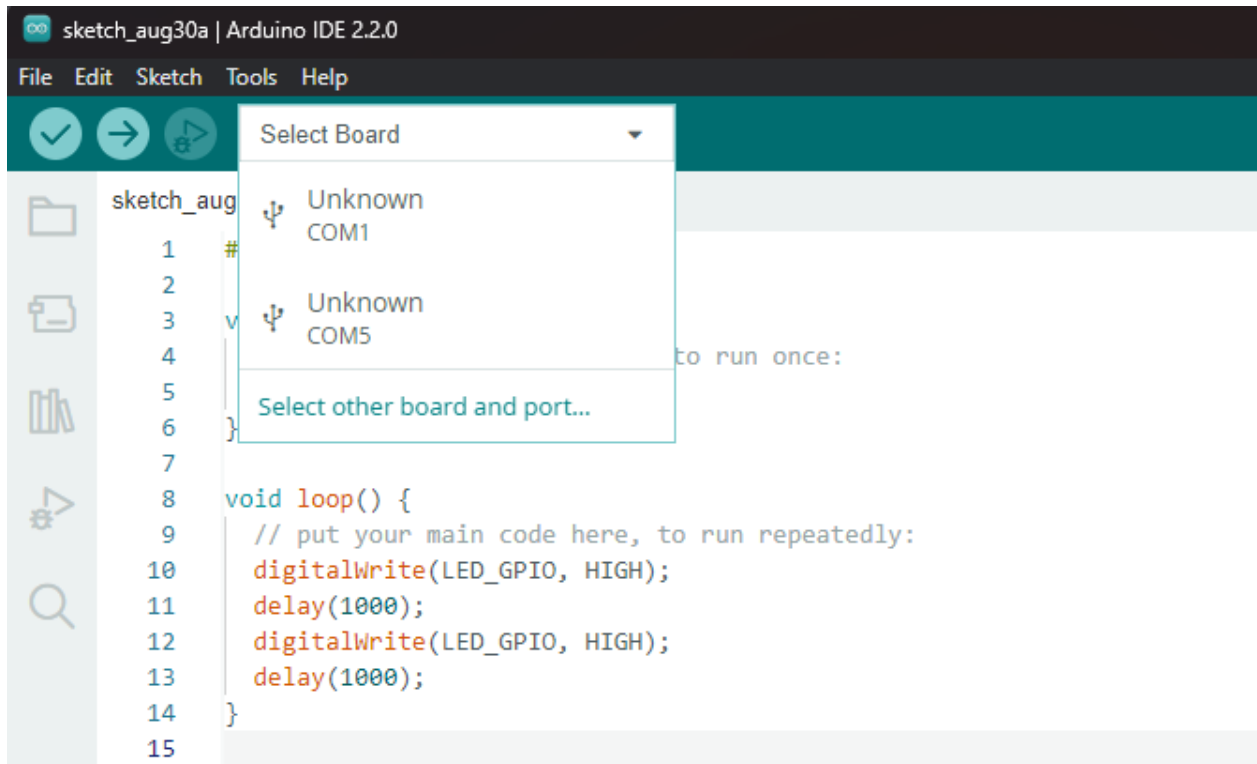
- De `loop`-functie herhaalt continu het volgende:
  - `digitalWrite(LED_GPIO, HIGH)` zet de LED aan (laat het licht geven).
  - `delay(1000)` wacht 1 seconde.
  - `digitalWrite(LED_GPIO, LOW)` zet de LED uit (dooft het licht).
  - `delay(1000)` wacht opnieuw 1 seconde voordat de cyclus zich herhaalt.

Hierdoor ontstaat het effect van een knipperende LED, die 1 seconde aan is, gevolgd door 1 seconde uit, en dit blijft herhalen zolang de Arduino actief is.

## 4. Het uploaden van je code naar de Microcontroller

Om je code uit te voeren moet hij worden **geupload** naar je microcontroller. Dit doe je als volgt:

1. Verbind je Arduino met een USB-kabel aan je computer.
2. In de Arduino IDE kan je klikken op "Select Board" en selecteer jouw Arduino.



### LETOP

Zoals je ziet in de afbeelding hierboven kunnen er meerdere zichtbaar zijn en wordt niet altijd aangegeven welke de Arduino is. Koppel dan je Arduino los en kijk welke is verdwenen, zo weet je dus welke je Arduino is.

3. Nadat je je Arduino hebt geselecteerd klik je op de pijl naar rechts (zichtbaar in de afbeelding hierboven) om je code te uploaden naar je Arduino. Er kan een scherm naar voren komen waarin gevraagd wordt naar je type Arduino. Zoek in deze lijst naar "Arduino Uno" en klik op "OK".
4. Als je alles goed hebt gedaan moet je nu een LED op je Arduino zien knipperen.

Gefeliciteerd! 🎉 Je hebt nu je eerste Arduino-code geschreven en op je microcontroller gezet.

## 5. Tijd Voor Hardware!

Nu we een lampje hebben laten knippen kunnen we hardware gaan koppelen aan de Arduino.

We gaan het volgende doen:

1. We gaan een knop en een LED toevoegen aan onze schakeling.
2. We gaan vervolgens een temperatuur-sensor toevoegen en de temperatuur sturen naar onze computer.

### 5.1 Een knop en een LED toevoegen.

Wanneer we knop en een LED gaan toevoegen hebben we het volgende nodig:

- 1x Arduino Uno R3
- 1x Breadboard
- 1x LED
- 1x Drukknop
- 1x 1kΩ weerstand (Kleurcode: Bruin-Zwart-Rood)



- Jumper draden

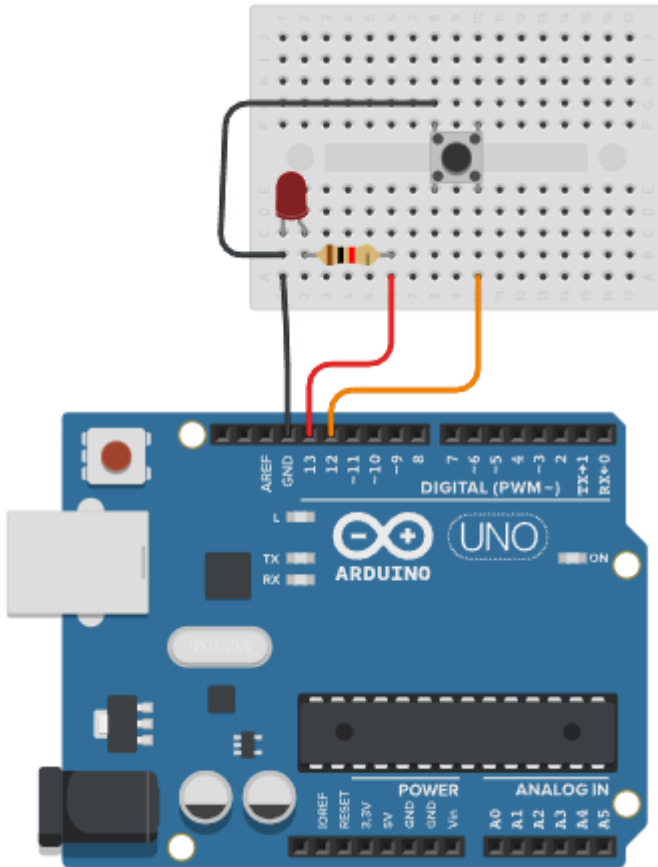
#### Kan je de juiste weerstand niet vinden?

Weerstanden hebben streepjes in diverse kleuren deze geven de waarde aan van de weerstand.

Online heb je hier diverse calculators voor: [Resistor Color Code Calculator](#)

Kijk welke weerstand het dichtste in de buurt komt van wat je nodig hebt.

Bouw vervolgens de volgende opstelling.



## LETOP

Zoals je ziet heeft een LED een lang pootje en een kort pootje. de lange kant is de plus en moet aan de de weerstand worden gekoppeld. De min is het korte pootje en moet aan de GND worden koppelt van de Arduino.

Met de volgende code kan je de knop gebruiken:

```
int BTN_GPIO = 12;

void setup()
{
    pinMode(BTN_GPIO, INPUT_PULLUP);
}

void loop()
{
    if(digitalRead(BTN_GPIO) == LOW){
        // Doe iets wanneer de knop is ingedrukt
    }
}
```

## Oefening

Probeer de code uit de vorige opdracht toe te voegen aan deze code.

## 5.2 Van Digitaal naar Analooog

Je zal waarschijnlijk wel denken "Een knopje is leuk, maar ik wil meer variatie dan alleen aan of uit.". Je hebt volkomen gelijk en dat is precies wat we nu gaan doen! Je hebt nu alleen maar gewerkt met digitale signalen. Echter is er ook zoiets als analoge signalen.

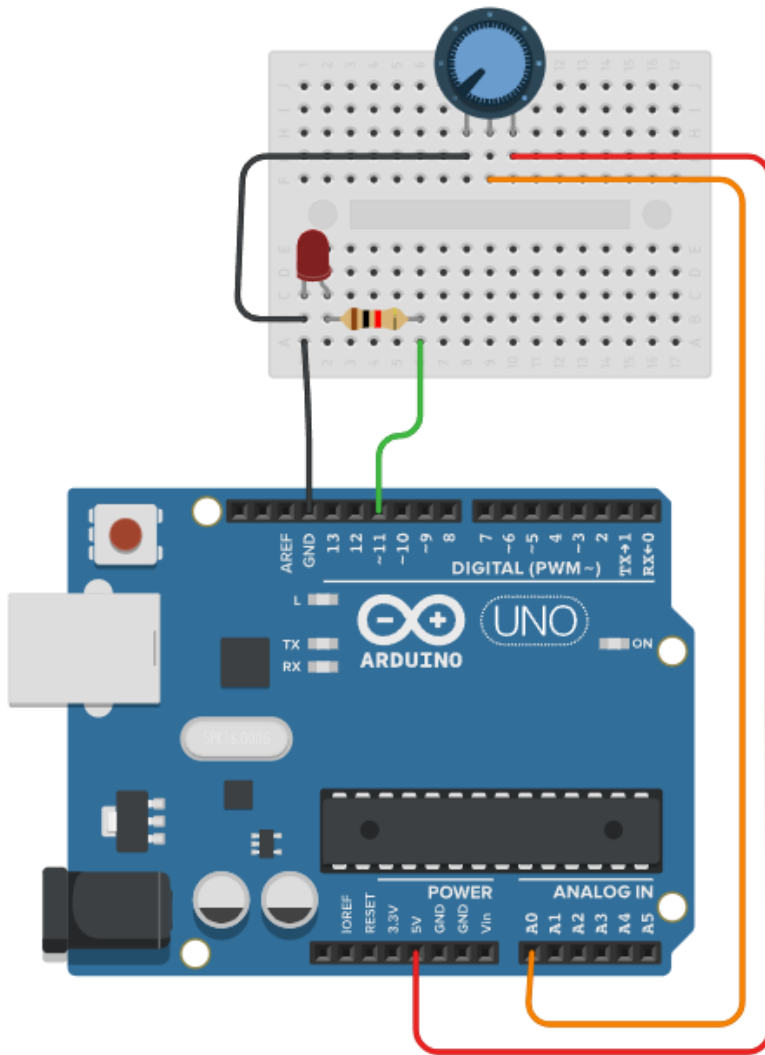
Een digitaal signaal kan je vergelijken met een één of een nul. Alle waardes hiertussen zoals 0.1, 0.15 of zelfs 0.0000000001 bestaan bij digitale signalen niet.

Wanneer we werken met analoge signalen bestaan deze wel! Wanneer we dit vertalen naar een spanning (Volts), kan je dit vergelijken met een waarde van 0V tot 5V. We moeten dus de spanning regelen! In je kit zit een component die dit voor je kan doen, namelijk een "Potmeter"



Deze kan je prikken in je Breadboard. Voordat we deze gaan koppelen aan de Arduino, is het belangrijk om te weten dat je Arduino alleen analoge signalen kan lezen met pinnen die dit soort signalen ondersteunen. Deze zijn gelabeld als pin nummers die beginnen met een "A" zoals A0-A5 (deze pinnen noem je **Analoge pinnen**).

Nu we dit weten, kunnen we de potmeter gaan verbinden\*:



**LETOP:** De LED zit nu aangesloten op pin 11

Wanneer je dit hebt gedaan kunnen we weer gaan coderen. Wanneer je terug kijkt in de code, zie je dat je hebt gewerkt met `digitalRead(...)`. Dit hebben we (zoals de naam al zegt) gebruikt voor het lezen van digitale signalen. nu hebben we een analoog signaal en gebruiken we dus `analogRead(...)`.

Kijk maar eens naar de code op de volgende pagina:



```

int potmeter_GPIO = A0;

void setup() {
    // Start de seriële monitor
    Serial.begin(9600);
    pinMode(potmeter_GPIO, INPUT);
}

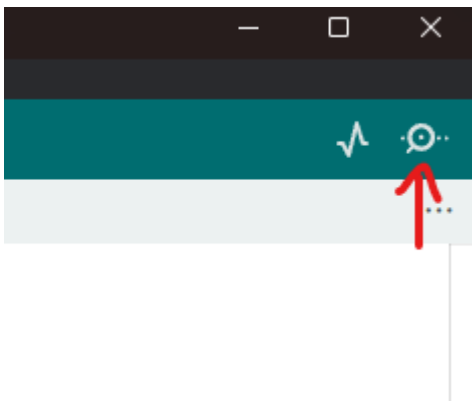
void loop() {
    // Lees de waarde en stop deze in een variabele
    int potmeter_value = analogRead(potmeter_GPIO);

    //Schrijf naar de seriële monitor
    Serial.println("Potmeter waarde = " + String(potmeter_value));
}

```

In de code hierboven definiëren we nog steeds onze pin als een `INPUT` omdat er een signaal in gaat. In de loop lezen we de analoge waarde en stoppen deze in een variabele zodat we hem later in de loop kunnen hergebruiken.

Daarbij staat ook nog iets anders, namelijk `Serial`. Dit gebruiken we om informatie naar onze computer te sturen in de vorm van **tekst (String)**. Je kan deze informatie op je computer lezen door de Seriële monitor te openen:



Wanneer je de nieuwe code upload en je alles goed hebt aangesloten, zie de waarde veranderen wanneer je aan de potmeter draait. HOERAAA!!! 🎉

## Oefening

Probeer nu de LED te dimmen met behulp van `analogRead(...)` en `analogWrite(...)`. Het schrijven van een analoge waarde is nieuw en vaardigt dus wat inzicht. Gebruik hier gerust Google voor.

### Leuke extra:

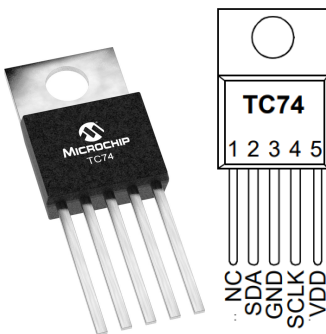
In de kit zit ook een LDR (**L**ight **D**ependent **R**esistor) wat samen met één weerstand exact fungeert als een potmeter. Echter kan je dan de spanning regelen aan de hand van de hoeveelheid licht in de omgeving.

Kijk hiervoor op de volgende website: [Arduino met LDR](#)

## 5.3 Sensoren en protocollen

Wat nou als we complexere informatie willen uitlezen van een sensor? Dan verloopt dit vaak niet via een analoog signaal, maar via een digitaal signaal met hierover heen een protocol. Eén zo'n protocol is **I<sup>2</sup>C**. Dit is een protocol dat werkt met behulp van twee lijnen (naast de plus en min) deze lijnen zijn **SDA** (voor de data) en **SCL** (voor de timing/klok). Het protocol werkt net als een straat met huisnummers (adressen). Er kunnen namelijk meerdere I<sup>2</sup>C sensoren en zelfs andere Arduino's worden aangesloten op deze lijnen. Het adres staat meestal vooraf in geprogrammeerd in de desbetreffende sensor. Hiermee kan je dus kan aangeven van welke sensor (huisnummer) je informatie wilt ontvangen.

De Arduino Uno heeft ook deze twee pinnen dit zijn namelijk: **A4 (SDA)** en **A5 (SCL)**. Laten we dit eens uit proberen! We gebruiken hiervoor de TC74 (zit in je kit). Dit is een digitale temperatuur-sensor dat zijn informatie stuurt via het I<sup>2</sup>C protocol. de sensor ziet er als volgt uit:



Bedraadt deze aan de Arduino Uno als volgt:

- **NC:** Not Connected (Deze hoeft je niet te verbinden)
- **SDA:** A4 (SDA) van de Arduino Uno
- **GND:** Dit is de min van de voeding en moet dus naar de GND op de Arduino Uno.
- **SCLK:** A5 (SCL) van de Arduino Uno
- **VDD:** Dit is de plus van de voeding en moet naar de 5V pin van de Arduino Uno.

Om deze sensor uit te lezen kunnen we een hulpmiddel gebruiken dat ook wel een **library** of **bibliotheek** wordt genoemd.

De bibliotheek die we nodig hebben kan je hier downloaden: [TC74 Library](#). Je installeert hem als volgt:

1. Hernoem het bestand naar `TC74.zip` (Pak hem niet uit!).
2. Open de Arduino IDE en navigeer naar "Sketch" > "Include Library" > "Add .ZIP Library".
3. Selecteer het zip bestand dat we zojuist hebben gedownload en hernoemt.  
Je hebt nu de bibliotheek geïnstalleerd zodat je hem in de code kan gebruiken. Het volgende voorbeeld kan je gebruiken om je sensor te testen.

```
#include "TC74.h"

TC74 sensor(0x4A); // 0x4A is hier het address

void setup() {
  sensor.begin(); // Start de sensor
  Serial.begin(9600); // Initialiseer de sensor
  while(sensor.isStandby()){ } // Wacht totdat de sensor is opgestart
}

void loop() {
  Serial.print("Graden Celcius: ");
  Serial.println(sensor.readTemperature('c'));
  Serial.print("Graden Farenheit: ");
  Serial.println(sensor.readTemperature('f'));
  Serial.print("Graden Kelvin: ");
  Serial.println(sensor.readTemperature('k'));
  delay(1500);
  Serial.println();
}
```

Deze code initialiseert de TC74-temperatuursensor, leest temperatuurgegevens uit in Celsius, Fahrenheit en Kelvin en stuurt deze gegevens naar de seriële monitor met tussenpozen van 1,5 seconden.

Zodra dit werkt, kan je deze code implementeren met de code van de andere oefeningen en kan je bijvoorbeeld een temperatuur sensor maken dat bij een instelbare waarde een LED laat branden.

## 6. Afsluiting

Gefeliciteerd met het verkennen van de basisprincipes van microcontrollers met Arduino! Deze reis door elektronica en programmeren is slechts het begin.

Onthoud dat Arduino niet de enige optie is. Andere microcontrollers zoals de Raspberry Pi Pico en ESP32 bieden ook geweldige mogelijkheden voor innovatie.

Met de kennis die je hebt opgedaan, ben je klaar om jouw creatieve projecten tot leven te brengen. Blijf experimenteren, leer van fouten en deel met de maker-community.

Bedankt voor je deelname aan deze workshop. Jouw avontuur in de wereld van microcontrollers is nu echt begonnen. Veel plezier en succes met het creëren van jouw eigen elektronische meesterwerken!

