

Bitlair/DJO arduino Workshop Opdrachten

Project 1: blink - Laat ledje knipperen op pin 13

volgorde:

open de arduino workspace

sluit arduino aan op computer met usb kabel

kijk bij tools – poort of de goede poort is aangesloten

kijk bij tools bord of de Arduino Uno is aangesloten

ga naar: file – example – 01 Basics – Blink

druk nu op vinkje linksboven (Verify), dit controleert de code

druk nu op het pijltje naast het vinkje (Upload), er gaan nu lampjes snel knipperen op de arduino, dit betekent dat er data verkeer is tussen de computer en de arduino.

Als de geel-oranje lampjes stoppen met knipperen blijft nu het groene lampje (L) knipperen.

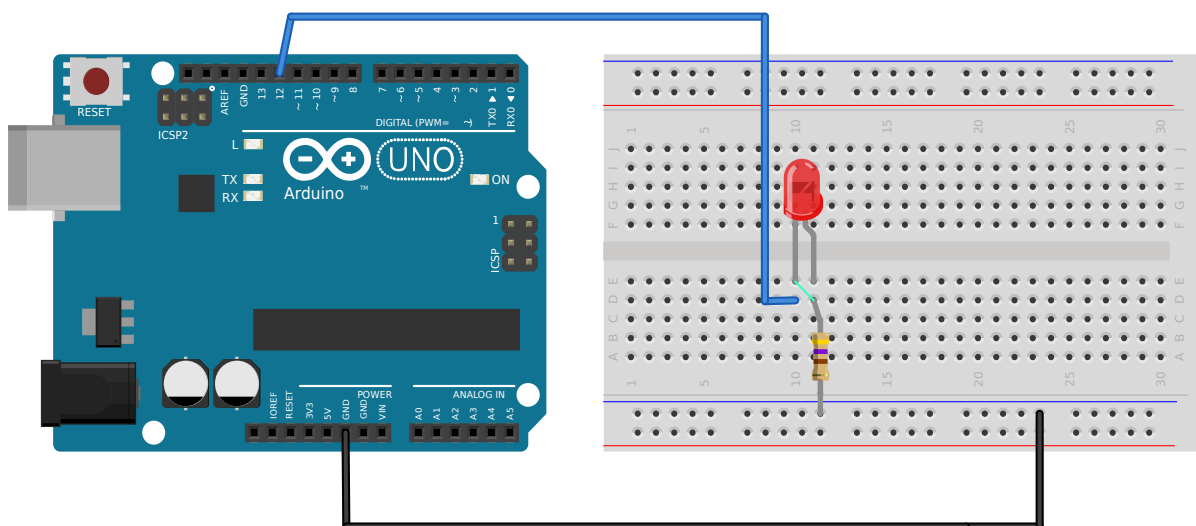
Dit lampje is rechtstreeks aangesloten op poort 13.

kijk nu in de code, in de code staat een `delay(1000);`

dit betekend dat de code 1000 milli seconde stopt, 1 seconde dus.

Probeer eens on het ledje langzamer of sneller te laten knipperen.

Sluit nu het breadboard aan volgens onderstaande schema:

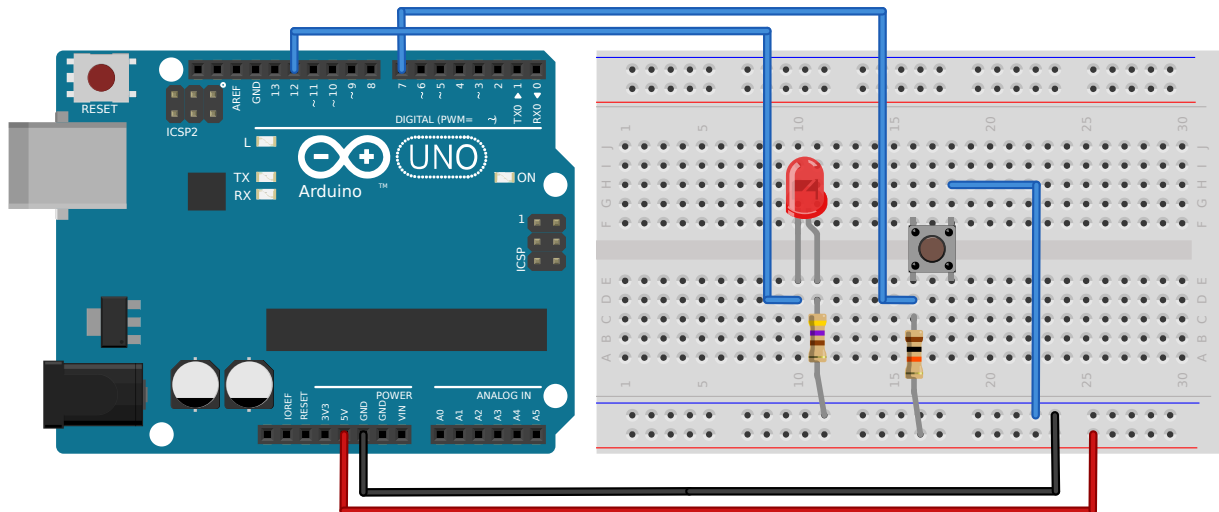


verander de code zodat poort 12 gebruikt wordt ipv poort 13

==> nu werkt de basis

Project 2: Knopje + led

Sluit het volgende schema aan:



De led blijft op poort 12

Op poort 7 (of andere digitale poort) wordt nu een button aangesloten met een z.g. pull-up weerstand. Als de button niet ingedrukt is staat er 5V (hoog) op poort 7, als de button ingedrukt is staat er 0V (laag) op poort 7.

Zet voor de setup routine het volgende:

```
const int ledPin = 12;
```

Dit definieert de variabele ledPin die naar poort 12 wijst.

Maak hieronder de variable buttonPin aan die naar poort 7 wijst.

In de setup() functie voeg je een regel toe waarin je de pinMode van buttonPin op INPUT zet.

En pas in setup() de bestaande regel aan zodat pin 12 een output pin wordt.

Verander pinMode(12, OUTPUT); naar pinMode(ledPin,OUTPUT);

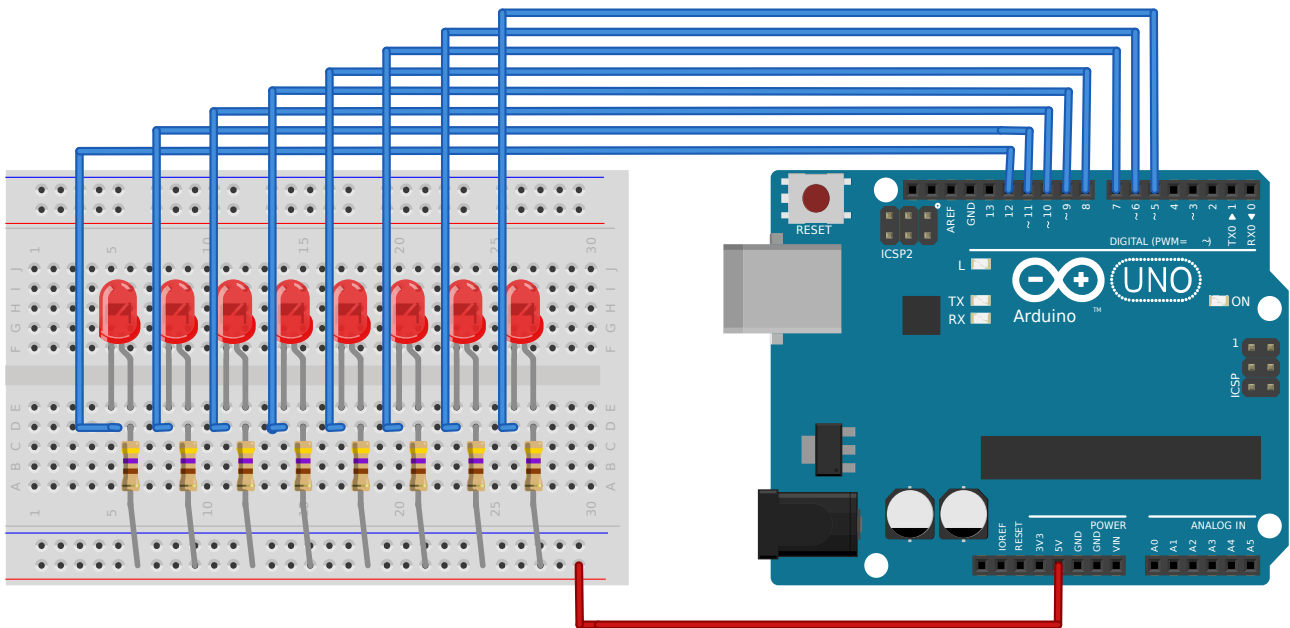
Verander de code in de LOOP functie met behulp van de volgende code:

```
int test = digitalRead(buttonPin);  
if (test == HIGH)  
{  
  digitalWrite(ledPin, HIGH);  
}  
else  
{  
  digitalWrite(ledPin, LOW);  
}
```

Als het goed is kan je nu met het knopje de led aan en uit zetten.

Project 3 - blink 8

Sluit het volgende schema aan:



Project 3a - Blink 8

Pas het programma van oefening 1 aan zodat 8 leds tegelijk knipperen.

Project 3b - looplicht 8

Pas het volgende programma in de arduino workspace, en pas het aan zodat alle leds achter elkaar aangaan.

```
void setup() // begin met setup parameters
{
  pinMode(9, OUTPUT); // initialiseer pin 9 (de linker led als uitgangs pin).
  pinMode(8, OUTPUT); // initialiseer pin 8 (de 2de led als uitgangs pin).
}
void loop() //Hier begint het programma te lopen
{
  digitalWrite(9, HIGH); // zet de led op pen 9 aan
  delay(1000);           // wacht 1 seconde
  digitalWrite(9, LOW);  // zet de led op pen 9 uit
  delay(1000);           // wacht 1 seconde
  digitalWrite(8, HIGH); // zet de led op pen 8 aan
  delay(1000);           // wacht 1 seconde
  digitalWrite(8, LOW);  // zet de led op pen 8 uit
  delay(1000);           // wacht 1 seconde
}
```

Als het werkt probeer dan dit programma:

```
int timer = 100; // The higher the number, the slower the timing.
```

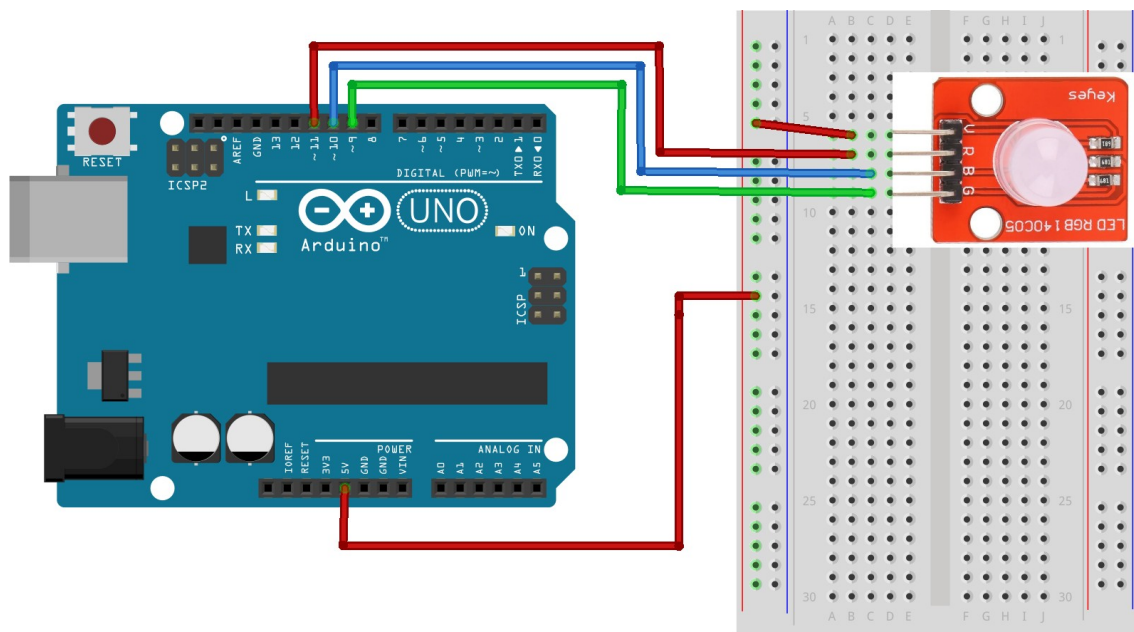
```
void setup
{
  // use a for loop to initialize each pin as an output:
  for (int thisPin = 5; thisPin < 13; thisPin++)
  {
    pinMode(thisPin, OUTPUT);
  }
}

void loop()
{
  // loop from the lowest pin to the highest:
  for (int thisPin = 5; thisPin < 13; thisPin++)
  {
    // turn the pin on:
    digitalWrite(thisPin, HIGH);
    delay(timer);
    // turn the pin off:
    digitalWrite(thisPin, LOW);
  }
}
```

Als dit ook werkt probeer dan eens om nog een loop te maken om de leds weer terug te laten lopen...

als dit je aan een televisie serie doet denken :-)

Project 4 - RGB led



4a – RGB led met digitale aansturing

Een RGB led bevat 3 verschillende leds met de kleuren rood (R), groen (G) en blauw (B). De drie leds zitten of aan de plus kant (genaamd: common anode) of aan de min kant (common kathode) aan elkaar. In ons geval is het de plus kant (common anode). Door de andere kant aan 3 outputs van de arduino te hangen kunnen de 3 kleuren onafhankelijk van elkaar aan- of uitgezet worden. Hierdoor kun je 8 kleuren maken:

Zwart (=alles uit)

Rood

Geel

Groen

Cyaan (licht blauw)

Blauw

Magenta (paars)

Wit

- Probeer dit uit.
- Kun je de verschillende kleuren achter elkaar laten verschijnen?
- Kun je de 7 kleuren (zonder zwart) ieder steeds 2 keer na elkaar laten knipperen?

4b – RGB led met analoge aansturing

Met PWM (pulse-width-modulation) kunnen in plaats van alleen hoog (5V) en laag (0V) ook waarden ertussen nagebootst worden (omdat ons oog snelle aan/uit schakeling van de led niet kan volgen en we in plaats daarvan alleen het gemiddelde zien, lijkt het alsof de led op bijv. halve of kwart intensiteit brandt).

Je kunt de leds met PWM aansturen door de pin in OUTPUT mode te zetten (net als hiervoor) en de pin aan te sturen met 'analogWrite(pin, waarde)'; hier in is waarde een getal van 0 t/m 255.

- Probeer dit uit door de 3 leds met verschillende waarden aan te sturen.
- Maak een for-loop die rood aanstuurt met alle waarden van 0 t/m 255.
- Doe hetzelfde voor groen en blauw en plaats de 3 loops achter elkaar.
- Probeer te bedenken hoe je alle kleuren van de regenboog na elkaar kunt laten zien.
Tip: Hiervoor heb je nog 3 for-loops nodig.

Project 5 – Waarden meten

Analoge waarden meten

De arduino kan ook spanningsniveaus tussen 0 en 5V meten op de pinnen A0 t/m A5.

De functie die hiervoor gebruikt wordt is `analogRead(pin)`. Deze functie geeft een waarde van 0 t/m 1023 terug. Bijv: Om de gemeten waarde op pin A0 te meten en deze in de variabele 'waarde' te zetten gebruik je '`int waarde = analogRead(A0);`'.

Seriële uitvoer

Tijdens het uploaden communiceert je computer met de arduino. Dit seriële communicatie kanaal kan ook gebruikt worden om je programma met je computer te laten communiceren. De arduino omgeving heeft ook een seriële monitor. De arduino kan hier tekst heen sturen en je kunt ook tekst intypen die naar de arduino gestuurd wordt en die je in een programma kunt inlezen.

Om de communicatie te starten moet je in de `setup()` functie de volgende regel toevoegen:

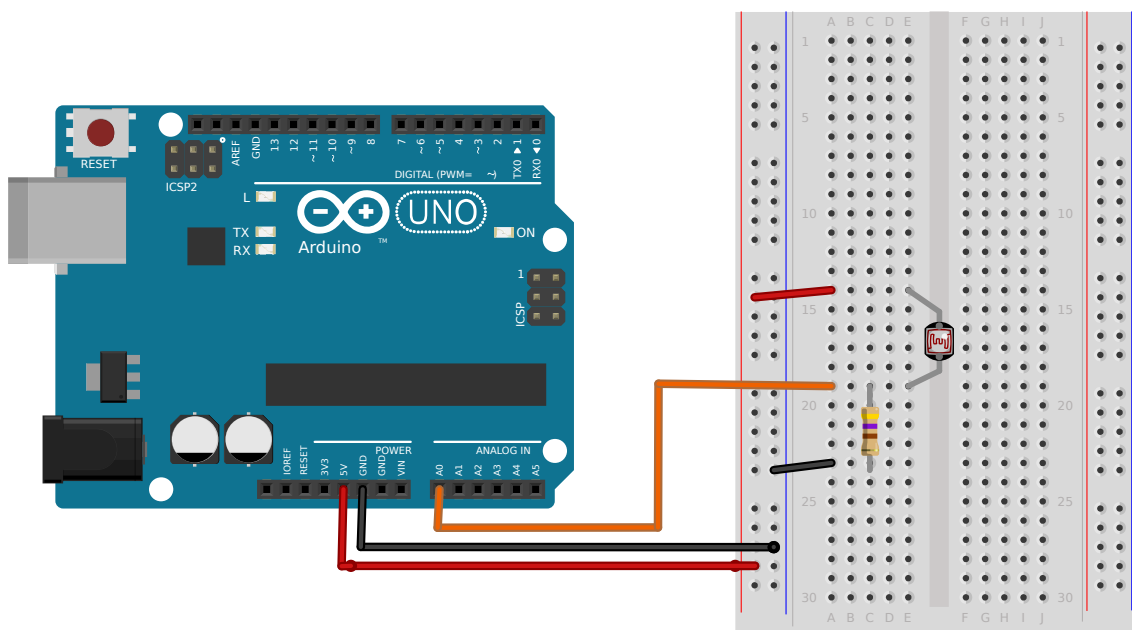
```
Serial.begin(9600);
```

De waarde 9600 geeft de snelheid aan waarmee de tekst naar de computer verstuurd wordt. De seriële monitor gebruikt standaard 9600. Je kunt de snelheid verhogen naar 115200 door dit aan te passen in zowel de seriële monitor als in de bovenstaande regel in de arduino. Voor deze opdracht maakt dat echter niet uit.

Hierna kun je de waarde van bijv. variabele waarde naar de seriële monitor sturen:

```
Serial.println(waarde);
```

5a – LDR (lichtgevoelige weerstand) en seriële uitvoer



Zet in een nieuwe sketch de aangeven regel met `Serial.open` in de `setup()` functie.

Zet in de `loop()` functie een regel die de waarde van pin A0 meet en die naar de seriële monitor

verstuurt.

Voeg daarachter een regel toe die 100 ms wacht.

Upload de code naar de arduino.

Open de seriële monitor.

Als het goed is zie je daarin waarden verschijnen. Dit zijn de gemeten waarden. Je kunt de waarden veranderen door je hand om de LDR te houden of er een fel licht op te schijnen.

Je kunt de waarden ook grafisch bekijken i.p.v. met de seriële monitor. Vraag hiervoor om een kopie van het demonstratie programma.

5b – LDR en RGB led

Maak een combinatie van de schakelingen van 5.1 (LDR) en 4.2 (RGB led met analoge aansturing).

De gemeten waarde varieert tussen 0 en 1023, de PWM (analoge) aansturing gebruikt waarden van 0 tot 255. Als je de RGB led wilt aansturen met PWM moet je dus de gemeten waarde door 4 delen:

```
int waarde = analogRead(A0) / 4;
```

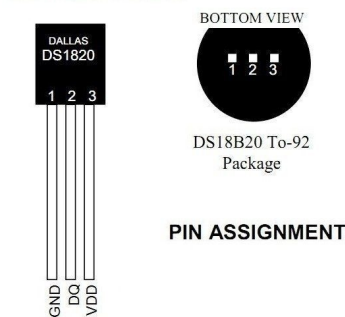
Maak een programma dat de RGB led aanstuurt met de gemeten waarde.

- a) Laat de led grijswaarden tonen; dus alle pinnen dezelfde waarde.
- b) Stuur de led aan met rood de gemeten waarde, groen 255 – waarde en blauw 0.

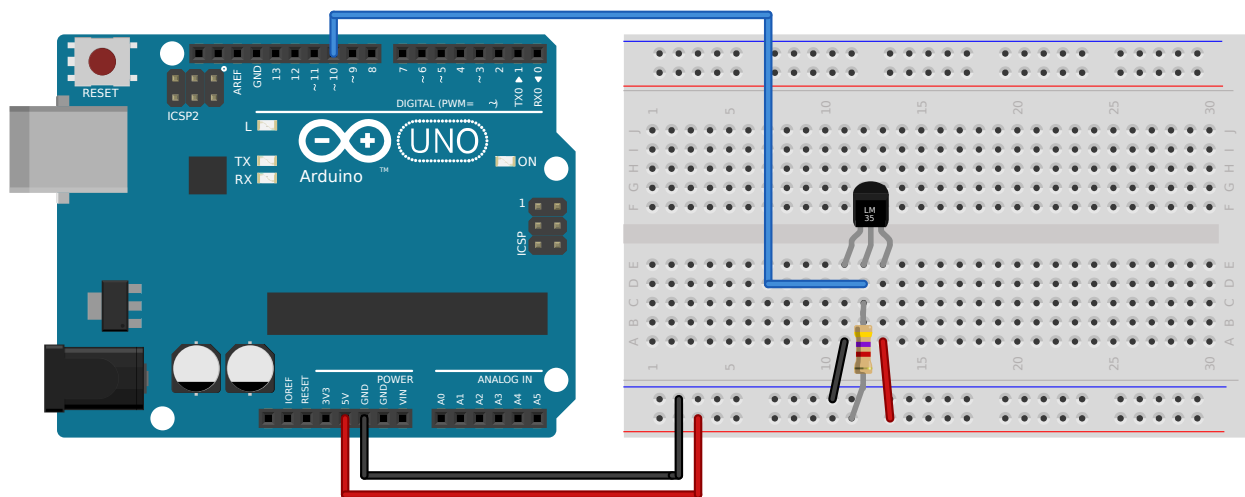
Opdracht 6 - One-wire communicatie met de temperatuur sensor

Er zijn veel IC's die intern al uitgebreide logica bevatten en zelf analoge waarden in digitale waarden omzetten. De gemeten waarden kunnen dan digitaal van de chip uitgelezen worden m.b.v. een protocol dat door de fabrikant gedefinieerd wordt. Vaak is voor de communicatie maar 1 pin nodig, dit heet dan ook one-wire communicatie. Natuurlijk is er dan ook nog een pin die aan de ground (0 V) ligt en vaak is er ook nog energie (5 V) nodig. Dit is ook het geval bij de temperatuur sensor die we gaan gebruiken. Deze is van het type DS18B20.

De figuur hiernaast geeft de functie van de 3 pinnen weer:



1. GND (0V)
2. DQ (data pin)
3. VDD (5V)



6a - Uitproberen

Vraag de code voor de temperatuursensor opdracht van een begeleider. Kopieer de directory naar je laptop en open de sketch.

Upload de sketch naar de arduino, open de seriële monitor en test of het programma werkt.

Bestudeer het programma en kijk of je in de sketch de plek kunt vinden waar de temperatuur uit de sensor gelezen wordt en hoe die daarna verwerkt en verstuurd wordt.

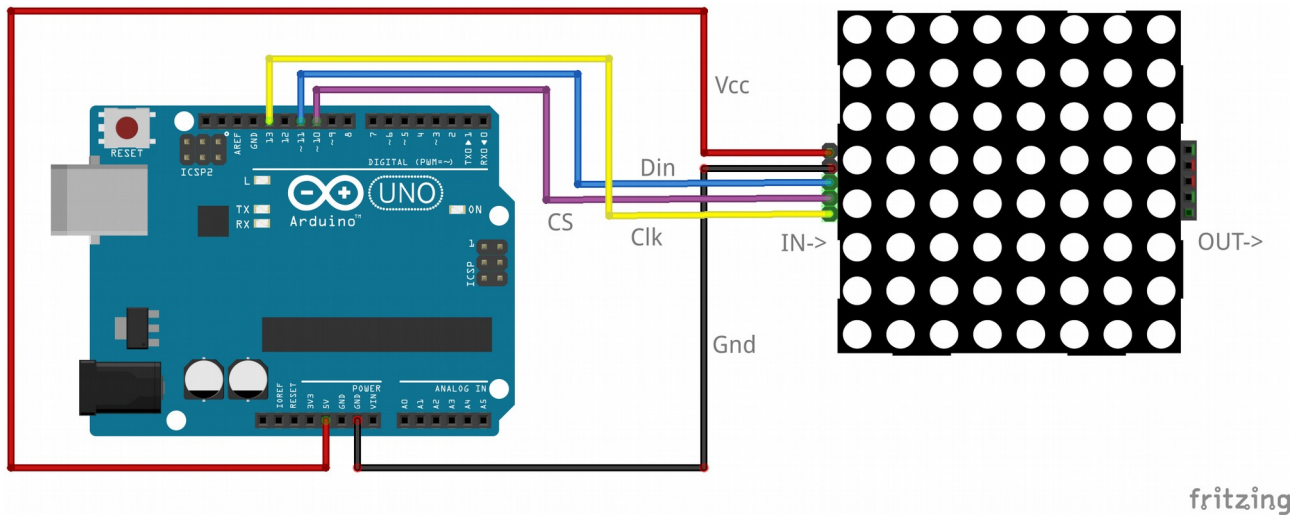
6b – Temperatuur sensor en RGB led

Maak een programma dat de temperatuur aangeeft d.m.v. de kleur van de RGB led. Gebruik de

volgende kleuren voor de temperatuur:

- blauw tussen 0 en 15
- groen tussen 15 en en 21
- oranje tussen 21 en 25
- rood boven 25 graden

Project 7 – 8x8 led display



Om het led display in elkaar te zetten, moet er eerst wat gesoldeerd worden. Als je hier geen ervaring mee hebt, vraag een van de begeleiders om daar bij te helpen.

- Soldeer de headers op het bordje; het ic en alle headers moeten aan dezelfde kant zitten. Van de haakse headers moet in ieder geval de input kant op het bord gezet worden.
- Plaats het led display op het bordje; onderaan zit in het midden van een van de zijden een uitstulping, die moet aan de kant waar FC-16 staat.

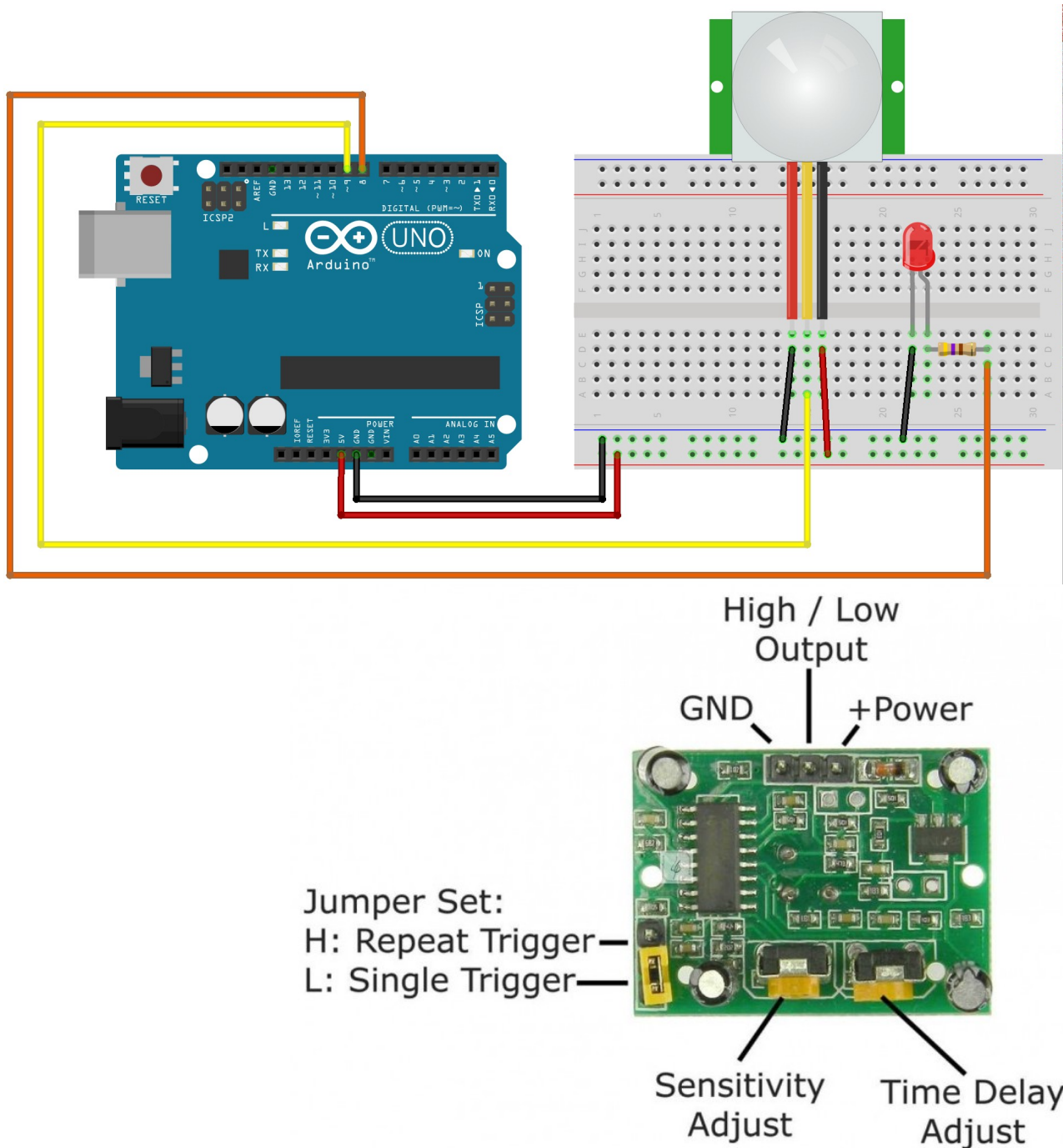
Het led display wordt aangestuurd door de MAXIM Max7219 chip die onder het display zit.

Sluit het display aan en laad de besturings code (module genaamd max7219) op de arduino.

Bestudeer de code.

Kun je het display ook iets anders laten doen? Bedenk zelf wat patronen om te laten zien. Je kunt bijvoorbeeld een patroon heen en weer laten schuiven, of rond laten draaien.

Project 8 – Bewegingssensor



Als bewegingssensor wordt de HC-SR501 PIR gebruikt. Dit is een sensor die met infrarood licht beweging detecteert. Er zitten een paar variabele weerstanden (genaamd potmeters) op waarmee de gevoeligheid en de bewegingsmeldingvertraging ingesteld kan worden.

De uitvoer van de sensor is hoog als er beweging is gedetecteerd en zo niet laag.

Schrijf een programma wat de led aan zet als er beweging gedetecteerd wordt.

Pas de potmeters aan en bekijk wat het resultaat is.

Project 9 – Berekenen vervangingsweerstand

Er is al gesproken over de wet van Ohm die geldt voor weerstanden: $V = I * R$.

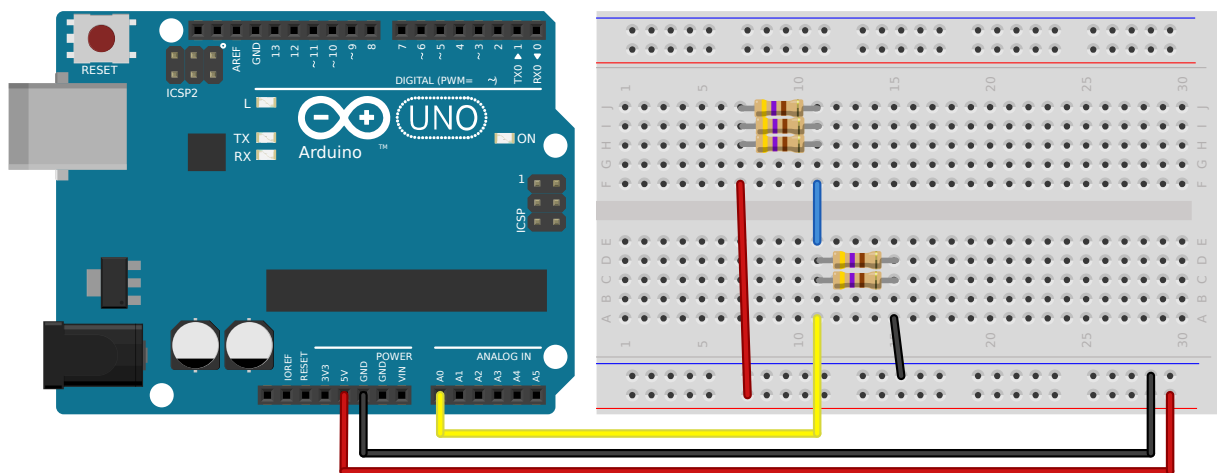
Hierin is:

- V de spanning over de weerstand,
- R de waarde van de weerstand en
- I de stroom door de weerstand.

Als je 2 (of meer) weerstanden in **serie** (achter elkaar) zet of als je de weerstanden **parallel** (naast elkaar) zet, kun je die vervangen door 1 weerstand waarvan de waarde de **vervangingswaarde** heeft. Dit betekent dat je de weerstanden dus kunt vervangen door 1 weerstand met de grootte van de vervangingsweerstand zonder dat de rest van de schakeling daar iets van merkt.

Voor weerstanden in serie (met bijv. waarden R1 en R2) is de vervangingswaarde gelijk aan R1 + R2. Voor R1, R2 en R3 in serie geldt dat de vervangingswaarde gelijk is aan R1 + R2 + R3.

Voor weerstanden parallel is de formule wat ingewikkelder. Hiervoor geldt dat als de weerstanden R1 en R2 parallel staan de vervangingswaarde gelijk is aan $(R1 * R2) / (R1 + R2)$. Als R1 en R2 dezelfde weerstandswaarde R hebben geldt dus dat de vervangingswaarde $(R * R) / (R + R) = (R * R) / (2 * R) = R / 2$. De vervangingsweerstand van tweede zelfde weerstanden parallel is dus de helft van die weerstand. Idem voor 3 parallel een derde, etc.



Bij deze opdracht is het de bedoeling dat je een aantal weerstanden parallel en in serie zet zoals in de figuur aangegeven.

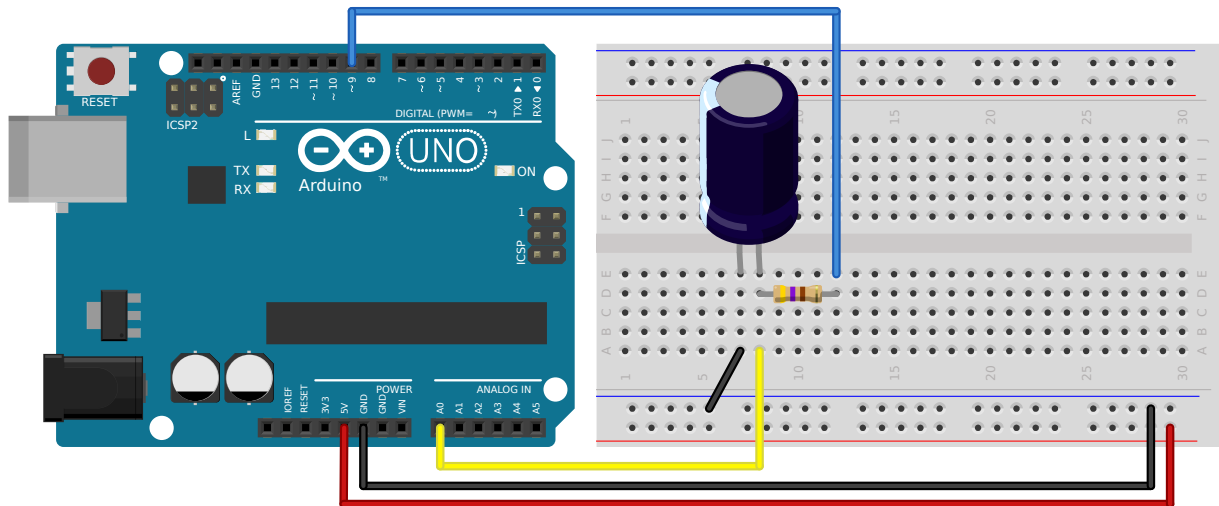
Door de vervangingsweerstand van het geheel te berekenen kun je de stroom erdoor bepalen.

Hiermee kun je spanningen berekenen over de delen die in serie staan.

Bereken eerst de waarde die je verwacht te meten op de analoge ingang. Schrijf daarna een programma die de waarde meet. Komen de waarden met elkaar overeen?

Project 10 – Condensator opladen

Opdracht 10a: Digitale aansturing



Bouw de bovenstaande schakelink en schrijf een programma dat:

- pin 9 digitaal 10 seconden hoog en daarna 10 sec laag maakt
- de spanning over de condensator (pin A0) analoog meet
- de gemeten waarden naar de seriële poort schrijft (met `Serial.println`).

Bekijk in de seriële monitor of met het grafische ScrollPlot programma wat de spanning doet. Kun je dat verklaren?

Als je andere weerstandswaarden gebruikt wat verandert er dan en waarom?

Opdracht 10b: Nu met PWM

Stuur pin 9 nu aan met PWM (`analogWrite`). Gebruik bijv. voor 10 seconden 25% (waarde 63), dan 10 seconden 50% (waarde 127) en dan 75% (waarde 191) .

Wat gebeurt er en waarom?