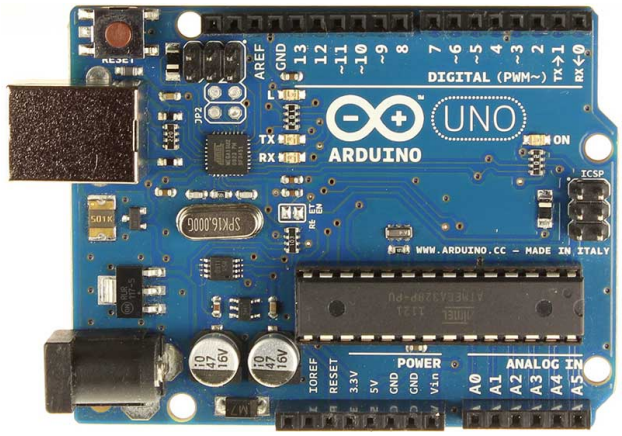


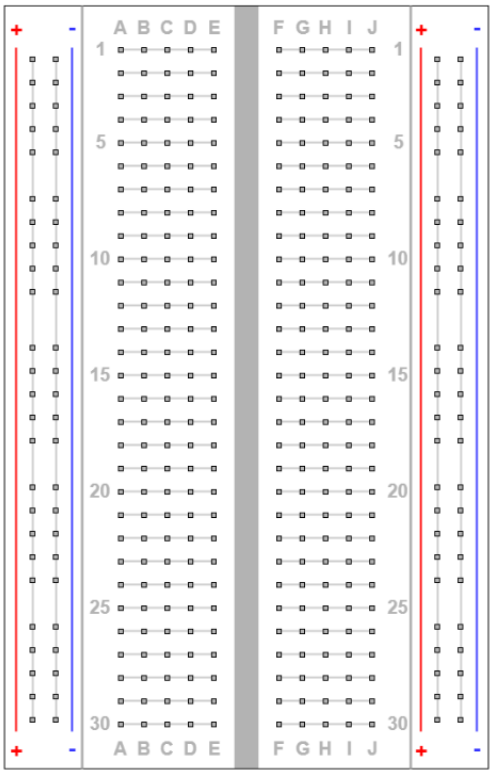
Een Arduino is een soort microcomputer. De Arduino sluit je aan op een computer waarbij je een programma (ook wel een sketch genoemd) stuurt naar de Arduino. De Arduino voert vervolgens je geschreven script uit en zorgt voor de uitvoer. Zo kun je een koelkast op een bepaalde temperatuur houden, een zelfrijdende robot aansturen, lichtsensoren maken, een lcd scherm op je shirt aansturen en ga zo maar door.

UITLEG EN THEORIE VAARDIGHEID

In deze module gaan we aan het werk met de Arduino en leren we de basismogelijkheden van een Arduino. Bij het werken met een Arduino heb je twee belangrijke onderdelen: De Arduino en een breadboard. De Arduino is de computer, met invoer en uitvoer mogelijkheden. Op het breadboard sluit je de elektronica aan die aangestuurd wordt door de Arduino. Je moet de Arduino, net als andere computers, programmeren om hem te laten doen wat je wilt. Bekijk deze introductievideo over de mogelijkheden van de Arduino: https://www.youtube.com/watch?v=meo_OyqEbiM.

Breadboard

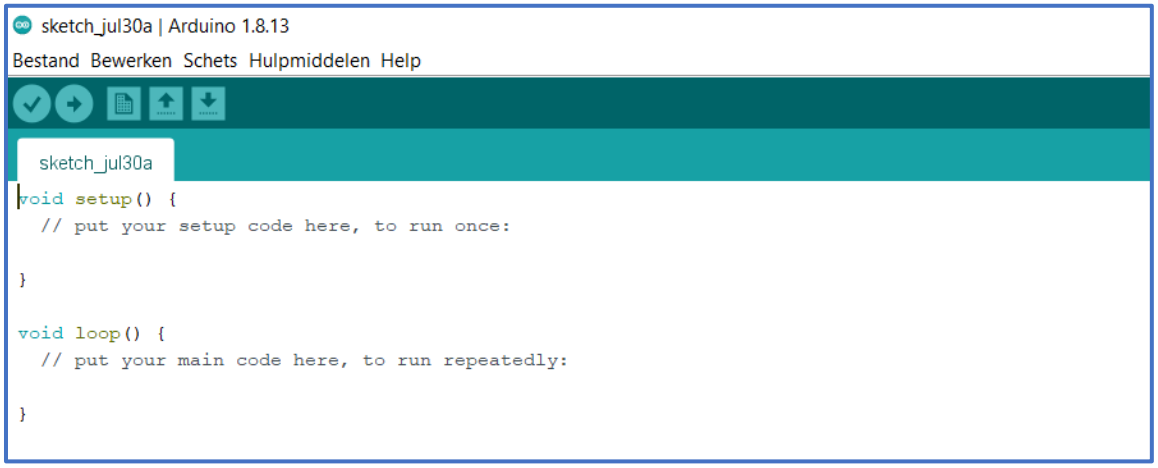
Het breadboard heeft aan beide zijdes twee kolommen die verbonden worden met de voeding. De + kant sluit je aan op de 5V uitgang of op een uitvoerpoort van de Arduino. De – kant sluit je aan op de GND (ground) van de Arduino. Alhoewel je de constante output niet altijd gebruikt, is het wel verstandig om deze altijd aan te sluiten.
Het breadboard heeft rijen en kolommen, zie hiernaast. De punten in een rij zijn met elkaar verbonden.
Dus als je kijkt naar de horizontale lijn 1 dan zijn daar de gaatjes A, B, C, D en E met elkaar verbonden.
Hetzelfde geldt voor de gaatjes F, G, H, I en J.
In die gaatjes kunnen we de aansluitdraadjes en vaak ook de pennetjes van de sensoren steken en zo gebruik maken van de verbindingen van het breadboard. Maar laten we hier niet te lang bij stil staan… we gaan aan de slag.



Maak nu **opdracht 1**

Programmeren deel 1

Arduino heeft een eigen programma dat gebaseerd is op de programmeertaal C++.
Heb je al ervaring met programmeren en/of met C++ (of java), dan is het programmeren heel makkelijk. Heb je geen ervaring met programmeren, het is veel minder moeilijk dan je misschien wel denkt! Je kunt het programma online gebruiken (je moet dan wel een account maken) of downloaden op https://www.arduino.cc/en/main/software. Wanneer je sketch heb gedownload en opent krijg je dit scherm te zien:



In **opdracht 1** heb je een simpele LED-schakeling gebouwd, die gaan we nu functioneel maken.
Open het het script “Blink” via: bestand/voorbeelden/basis/blink of examples/basics/blink.
Bovenaan het script Blink staat /* dit betekent dat alles na dit beginteken en voor het afsluitteken */ commentaar is. Hier zet je neer hoe het programma heet, wat het doet, wie het gemaakt heeft en wanneer je het voor het laatst hebt veranderd. Een tweede mogelijkheid om commentaar toe te voegen is met behulp van //. Alles op dezelfde regel na // is commentaar. Zo kun je bijhouden wat de code op die regel doet.

Voordat de code wordt uitgevoerd moet je zeggen wat er precies aangestuurd wordt. We geven aan dat in pin 13 iets zit en dat de Arduino daar een output (spanning) geeft als jij dat wilt. Alles wat tussen de accolades {} van de loop staat wordt continu herhaald. Eerst wordt pin 13 hoog (5,0 V) gemaakt met behulp van de code digitalWrite (digitalWrite kan alleen aan of uit). Daarna moet het programma 1000 ms wachten (delay) voordat de volgende regel code wordt uitgevoerd. De volgende regel code maakt pin 13 weer laag (0,0 V).

```
void setup(){
   pinMode(13,OUTPUT);
}
```

```
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Maak nu **opdracht 2**
Bekijk deze video’s als je vastloopt:
https://www.youtube.com/watch?v=KgMocAMAJvg
https://www.youtube.com/watch?v=b3f40MWFHiI

Programmeren deel 2 = uitleg Button

```
const int buttonPin = 2;
const int ledPin =  13;
```

Het script “Button” is uitgebreider dan “Blink”. We lopen stap voor stap door het script.

```
int buttonState = 0;
```

We hebben te maken met twee poorten waar iets moet gebeuren. Pin 2 is een invoer en pin 13 moet een uitvoer zijn. De pin verandert niet, dit is een constante (const). Het is een gehele waarde, een integer (int).

```
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
```

We geven pin 2 een herkenbare naam, pin 2 heet nu buttonPin. Pin 13 heeft de naam ledPin gekregen.
We willen straks weten wat de ‘staat’ van de knop is (ingedrukt of niet). Deze kan 1 of 0 zijn. We moeten even vertellen dat we de staat van de knop willen weten (aanmaken van een variabele) en deze straks willen vergelijken. Voordat we het script laten draaien is de buttonState gelijk aan 0.
Zoals gezegd, we moeten even vertellen dat de ledPin een OUTPUT is en de buttonPin een INPUT. Zo, dat weten ze nu dan ook…

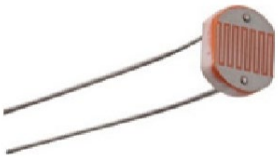
```
void loop() {
  buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
      digitalWrite(ledPin, LOW);
  }
}
```

Er moet iets gebeuren als de knop wordt ingedrukt. Aan het begin van de loop wordt de knop uitgelezen (met digitalRead die gecontroleerd of er 5,0 V spanning over de weerstand staat). Daarna volgt een if statement. Als (if) de knop is ingedrukt (buttonState == HIGH) dan moet de LED gaan branden (digitalWrite(ledPin, HIGH);). In alle andere gevallen (else), moet de LED niet branden (digitalWrite(ledPin, LOW);).
Het is een makkelijk if-statement. Je kunt ook meerdere voorwaarden stellen. Als je twee knoppen tegelijk in moet drukken voordat de LED gaat branden zet je && tussen de tekens (if buttonState1 == HIGH && buttonState 2 == HIGH){}).
Helaas blijft het lampje nu nog niet branden als je de knop hebt ingedrukt. Je kunt er wel zelf voor zorgen dat je met één knop het lampje aan en uit kan zetten. Vergeet niet de nieuwe variabele (int) te definiëren aan het begin!

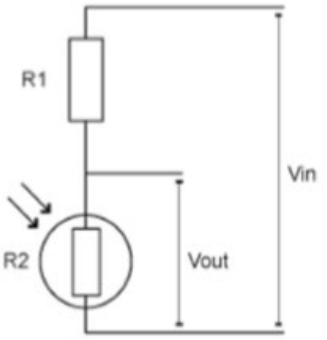
```
void loop() {
  buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH && state == LOW)
{
          state2 = HIGH;
  }
if (buttonState == HIGH && state == HIGH) {
          state2 = LOW;
  }
  state = state2;
  digitalWrite(ledPin,state);
  delay (100);
}
```

Maak nu **opdracht 3**

Sensoren = INPUT



Wanneer je alleen een knop en een LED gebruikt, ben je snel uitgekeken en heb je nog nauwelijks iets gedaan… We willen robots aansturen, racemachines niet tegen muren laten botsen, kassen automatisch aansturen en ga zo maar door. Daar hebben we sensoren voor nodig.
Vrijwel alle sensoren werken hetzelfde, de weerstandswaarde verandert bij een veranderende input (bijvoorbeeld licht, vocht of kracht). We gaan beginnen met een lichtsensor. Voor een lichtsensor hebben we een LDR nodig, zie de foto.
LDR staat voor Light Dependent Resistor (licht afhankelijke weerstand).
De weerstandswaarde wordt lager als de lichtintensiteit toeneemt.
We sluiten de LDR in serie aan met een weerstand met een constante waarde. Zo hebben we een spanningsdeler gemaakt. Een deel van de spanning staat over de LDR en een deel van de spanning staat over de weerstand. Het enige wat we moeten doen is de spanning over de LDR uitlezen en we weten hoe licht het is!

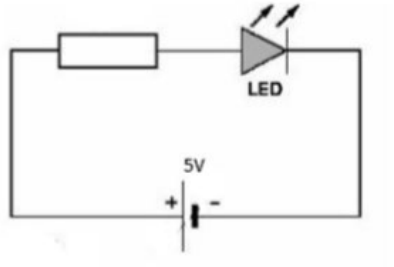


Als het lichter is, wordt de weerstandswaarde van de LDR kleiner waardoor er minder spanning over de LDR staat maar meer spanning over de Ohmse weerstand. De spanning over de LDR kun je meten met behulp van de ANALOG IN van de Arduino.
De spanning is dan een maat voor de gemeten lichtintensiteit. In welk gebied je sensor gevoelig moet zijn bepaalt de keuze voor een Ohmse weerstand. De LDR heeft een waarde tussen de 10 kΩ en 20 kΩ: dus kies een grote weerstand!

Maak nu **opdracht 4**
Bekijk deze video als je vastloopt of meer wilt leren over de LDR:
https://www.youtube.com/watch?v=HcoPGiV9SUU

OEFENINGEN

Opdracht 1 Het aansluiten van een LED



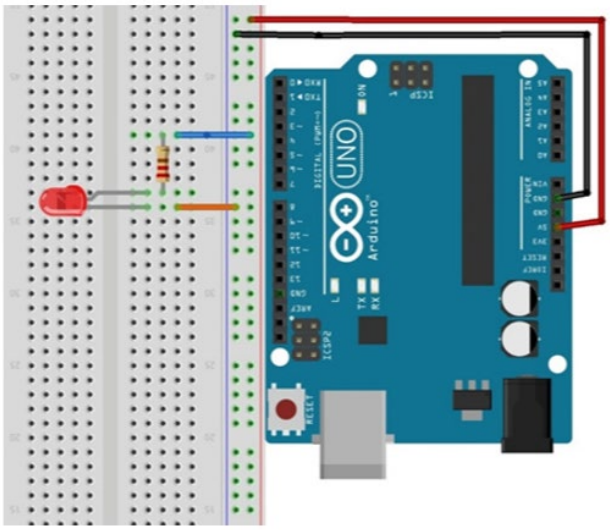
Een LED is een diode die licht uitzendt als er stroom door de LED gaat. De LED laat stroom maar door in één richting. De lange poot van de LED moet altijd op de + kant aangesloten worden, de korte poot op de – kant.
De stroom door een LED mag meestal maar 20 mA zijn. De spanning over de LED is dan ongeveer 2,0V, deze waardes verschillen een beetje per soort LED.
De Arduino levert een 5,0 V spanning. De LED moet dus in serie geschakeld worden met een weerstand. De weerstand moet dus minimaal 150 Ω zijn (R = U / I = 3,0 / 0,020 = 150 Ω).

a) Er is geen weerstand aanwezig van 150 Ω maar wel een van 220 Ω. Zoek de weerstand op met behulp van de kleurcode: de eerste ring moet rood zijn, de tweede ring ook en de derde ring bruin (22·10). Een andere mogelijkheid is rood, rood, zwart, zwart.

b) Sluit nu de 5V uitgang van de Arduino aan op de + kolom en de GND (ground) uitgang van de Arduino aan op de – kolom.

KLEURCODE VAN WEERSTANDEN

KLEUR	1e RING	2e RING	3e RING	MULTIPL.	TOL.
ZWART	0	0	0	1	
BRUIN	1	1	1	10	± 1%
ROOD	2	2	2	100	± 2%
ORANJE	3	3	3	1k	
GEEL	4	4	4	10k	
GROEN	5	5	5	100k	± 0,5%
BLAUW	6	6	6	1M	± 0,25%
VIOLET	7	7	7	10M	± 0,10%
GRIJS	8	8	8		± 0,05%
WIT	9	9	9		
GOUD				0,1	± 5%
ZILVER				0,01	± 10%
BLANK					± 20%

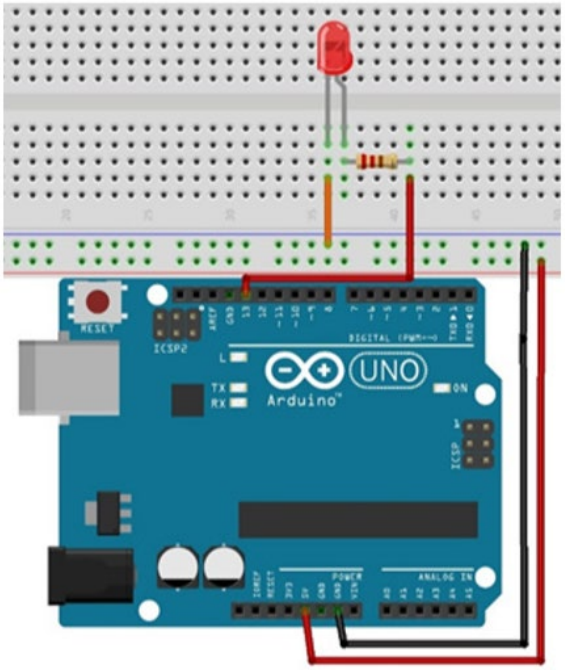


c) Sluit de LED en de weerstand in serie aan, zie de tekening.

d) Verbind de Arduino via de usb met de computer. Als je het goed hebt gedaan brandt de LED!

Maak een screen-shot, foto of video van je programma en werkende schakeling

Opdracht 2 Een knipperende LED



Nu kun je je led wel laten branden, maar daar heb je nog niet veel aan… Een volgende stap is de LED aansturen met behulp van een stukje code. Om de LED aan te sturen met behulp van code moet je een uitvoerpoort (bijvoorbeeld pin 13) van de Arduino verbinden met je LED. De 5V output heb je nu niet nodig, pin 13 levert nu de spanning. In de tekening zie je dat deze wel verbonden is, bij grotere projecten vergeet je snel de 5V, vandaar!

a) Bouw de opstelling die je ziet in de tekening en sluit de Arduino aan op de computer.

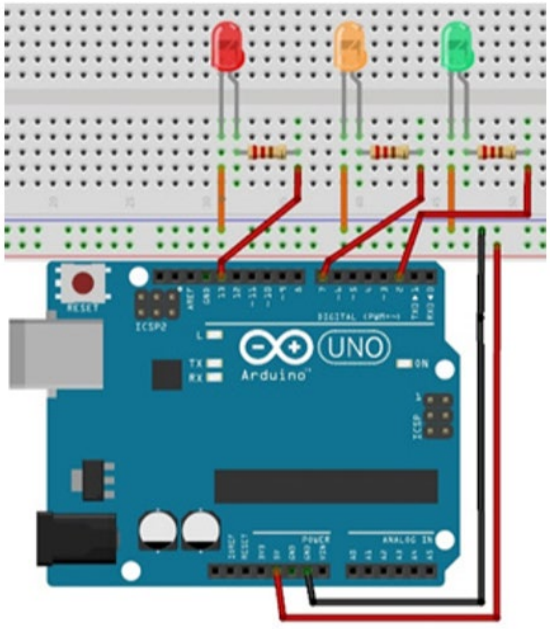
We willen de Arduino aansturen, daarvoor gebruiken we het Arduino programma.

b) Open het programma en open het script blink via: bestand/voorbeelden/basis/blink.

c) Controleer het script met het vinkje. Mocht het programma niet werken, dan geeft het onderaan een foutmelding weer

d) Upload het script naar je Arduino met het tekentje  . (snelcode: ctrl + u)

e) Beschrijf kort wat je ziet. Probeer wat je ziet te verklaren met behulp van de code.



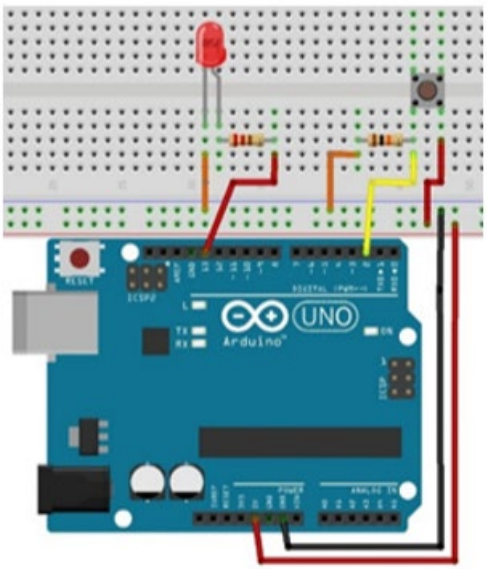
f) Verander het script zodat de LED sneller knippert.

g) Sluit drie verschillende LED lampjes aan, verander het script zodat ze om de beurt aan staan. Hint: je kunt de commando's in de void loop kopiëren plakken en aanpassen.

h) Verander het script en maak een verkeerslicht waarbij oranje maar even aan staat.

Maak een screen-shot, foto of video van je programma en werkende schakeling

Opdracht 3 Een aan en uit knop voor de LED



We kunnen nu de LED met behulp van de code aan en uit zetten en zelfs dimmen. Maar vaak wil je ook een schakeling handmatig aan en uit kunnen zetten. Daarvoor hebben we een drukknop nodig.

a) Bouw de schakeling die hiernaast staat.
Let op dat je een 1K of hogere weerstand gebruikt voor de knop zodat de te leveren stroom niet te groot is, een Arduino is namelijk slecht in het leveren van grote stroomsterktes.

Het idee is nu dat we pin 2 gebruiken als INPUT. Pin 2 meet de spanning op dat punt (vergelijkt deze met 0,0 V). Dit gebeurt met de code digitalRead(). Deze kan nu een hoog (5,0 V) of laag (0,0 V) signaal meten (met een analoge pin, A0 t/m A5, kunnen ook tussen gelegen waardes (10 bits) gemeten worden). Pin 2 meet alleen een spanning als de knop (button) ingedrukt wordt.

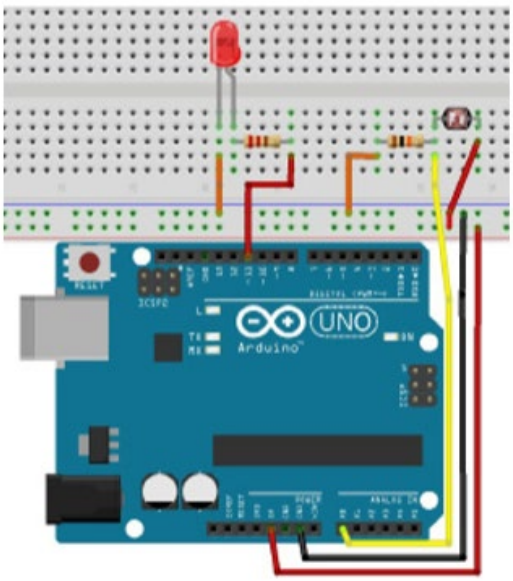
b) Open het script Button via voorbeelden/digitaal en upload het script.

c) Druk op de knop en controleer wanneer de LED uit is en wanneer deze aan is.

d) Pas het script aan zodat de functie van de knop precies omgedraaid wordt.

Maak een screen-shot, foto of video van je programma en werkende schakeling

Opdracht 4 Een lichtsensor



De schakeling lijkt erg op de schakeling van opdracht 3. Alleen nu gebruiken we een LDR en een ANALOG IN.

a) Bouw de schakeling.

b) Open in het script analogReadSerial: voorbeelden/basis/ Analogreadserial.

De belangrijkste code vind je hieronder:

```
void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(10);
}
```

De Arduino krijgt de opdracht om de analoge poort uit te lezen (analogRead(A0)). Deze waarde wordt gehangen aan de variabele sensorValue. Vervolgens willen we deze waarde weten. De waarde wordt dan ook geprint voor ons (Serial.println(sensorValue);), deze is te lezen met behulp van de seriële monitor (het loepje rechts bovenin). Voor de stabiliteit is het goed om een delay in te bouwen.

c) Upload het script naar de Arduino en lees de waarden uit met behulp van de Serial Monitor.

d) Bedek met je hand de LDR. Verandert de gegeven waarde?

e) Breid de schakeling verder uit zodat je de hele schakeling ook met een knop aan en uit kan zetten.

Maak een screen-shot, foto of video van je programma en werkende schakeling

WAT MOET JE KUNNEN?

Wat moet de leerling kunnen laten zien om de vaardigheid af te tekenen?

HANDIGE LINKS

https://maken.wikiwijs.nl/userfiles/9ea9aebf22a33f26ac82d00c51ade3b7a4966149.pdf

https://maken.wikiwijs.nl/135892/Cyclus_1___Arduino#!page-4957061