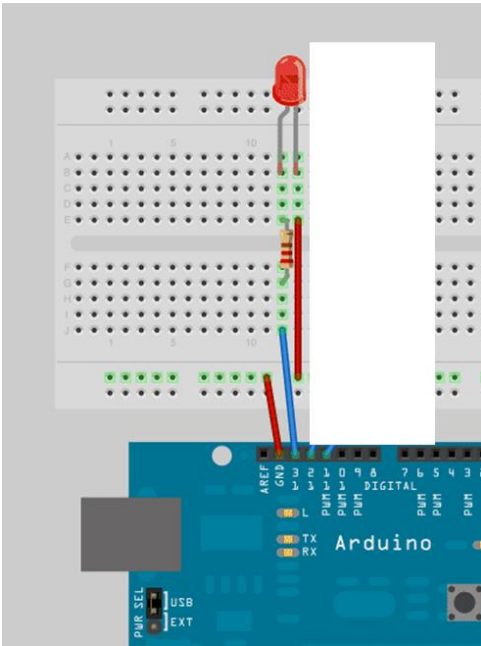


Workshop Arduino - deel 2

Inleiding

In het vorige deel van de workshop Arduino heb je voor het eerst kennis gemaakt met deze microcontroller. In dit deel, deel 2, ga je meer leren over de software: het schrijven van het programma dat door de Arduino uitgevoerd wordt.

1. Één LED



In de volgende opdracht gaan we drie LEDs op verschillende manieren laten knipperen. Daarvoor starten we met het programma 'Blink', dat je in deel 1 ook al een keer gezien hebt. Voordat we de volledige schakeling met 3 LEDs maken, maken we eerst de schakeling met één LED. Zie de tekening hiernaast, er wordt een weerstandje van 330 ohm (oranje-oranje-bruin) gebruikt.

Start de Arduino-software op op je laptop. Klik in de menubalk op *bestand, voorbeelden, 01.Basics* en dan *Blink*. De tekst van hierboven komt nu ook op je eigen scherm te staan. Sluit de USB-kabel aan en upload het programma naar de microcontroller.

We gaan nu kijken wat elke regel in het programmaatje betekent. Belangrijk om te weten is dat alle tekst die tussen */** en **/* staat of die op een regel na *//* staat niet bij het programmaatje hoort. Dit is alleen een toelichting, maar de Arduino doet er niks mee.

Eerst de twee grote blokken in het programmaatje:

Programmeercode	Betekent
<pre>void setup() { pinMode(13, OUTPUT); }</pre>	<i>Alles wat tussen de { en } staat achter 'void setup()' wordt alléén uitgevoerd bij de eerste keer.</i>
<pre>void loop() { digitalWrite(13, HIGH); delay(1000); digitalWrite(13, LOW); delay(1000); }</pre>	<i>Alles wat tussen de { en } staat achter 'void loop()' wordt telkens herhaald: wanneer het programmaatje onderaan is begint hij weer bij het begin van de loop.</i>

En nu de afzonderlijke opdrachten die je ziet:

Programmeercode	Betekent
<code>pinMode(13, OUTPUT);</code>	Zo 'vertel' je de Arduino dat poortje nummer 13 een uitgang (output) is.
<code>digitalWrite(13, HIGH);</code>	Stuur een hoog ('high') signaal naar poortje 13. Er komt dus een spanning (van 5 volt) op poortje 13 te staan. De LED gaat hierdoor aan.
<code>delay(1000);</code>	Wacht 1000 milliseconde (= 1 seconde) voordat je verder gaat met de volgende opdracht.
<code>digitalWrite(13, LOW);</code>	Stuur een laag ('low') signaal naar poortje 13. Er komt dus géén spanning (dus 0 volt) op poortje 13 te staan. De LED gaat hierdoor uit.
<code>delay(1000);</code>	Wacht 1000 milliseconde (= 1 seconde) voordat je verder gaat met de volgende opdracht.

Doordat het deel in de 'loop' steeds herhaald wordt, blijft het LEDje aan en uit gaan.

Opdrachten

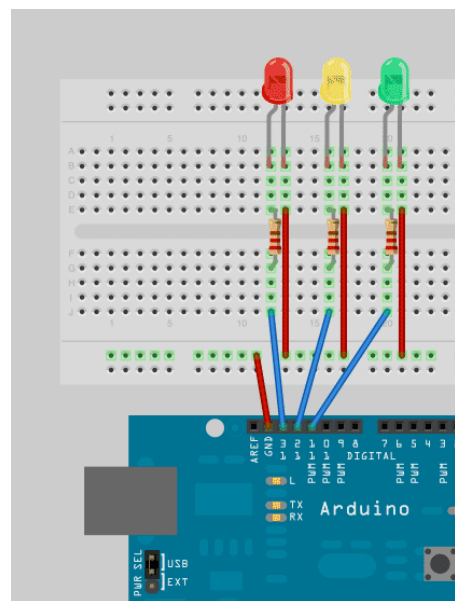
Blijf proberen tot het opdrachtje gelukt is. Loop je vast? Vraag gerust een docent of klasgenoot om hulp!

		Gelukt?
a.	Sluit het blauwe kabeltje (zie de tekening op de vorige pagina) aan op poort 12 in plaats van poort 13. Pas het programmaatje zó aan dat ook nu de LED knippert. (Vergeet niet om het programmaatje te uploaden.)	
b.	Pas het programmaatje zó aan dat het LEDje eerst twee seconden aan staat en dan een halve seconde uit (en daarna opnieuw).	
c.	Pas het programmaatje zó aan dat het LEDje eerst één seconde aan staat, dan één seconde uit, daarna drie seconden aan en dan drie seconden uit (en daarna alles opnieuw). Hint: hiervoor moet je zelf 4 nieuwe regels in je programmaatje typen.	

2. Drie LEDs

In deze opdracht ga je het programmaatje van opdracht 1 op verschillende manieren aanpassen om drie LEDs te laten knipperen. Maak daarvoor eerst de schakeling zoals op de tekening hiernaast. Er worden weerstandjes van 330 ohm (oranje-oranje-bruin) gebruikt. De kleur van de LEDjes is niet van belang. Open opnieuw het programma 'Blink' in de menubalk via *bestand*, *voorbeelden*, *01.Basics* en dan *Blink*.

We willen nu alle drie de LEDs tegelijkertijd aan en uit laten gaan. Daarvoor moet het programmaatje gewijzigd worden. Hieronder staat het programmaatje, met aangegeven wáár je een regel moet toevoegen. De eerste twee regels die je toe moet voegen heb je nodig om de Arduino te vertellen dat poort 12 en 11 óók uitgangen zijn. De middelste twee regels zijn nodig om de twee nieuwe lampjes ook aan te zetten, de onderste twee regels gebruik je om de twee nieuwe lampjes ook uit te zetten.



```

void setup() {
  pinMode(13, OUTPUT);
  //Hier een regel toevoegen!
  //Hier een regel toevoegen!
}
void loop() {
  digitalWrite(13, HIGH);
  //Hier een regel toevoegen!
  //Hier een regel toevoegen!
  delay(1000);
  digitalWrite(13, LOW);
  //Hier een regel toevoegen!
  //Hier een regel toevoegen!
  delay(1000);
}

```

Schrijf zelf de ontbrekende regels zodat alle drie de LEDs samen (tegelijktijd) gaan knipperen. Voeg ze toe in het programmaatje en upload het naar de Arduino.

Als dit gelukt is weet je hoe je een LED aan en uit kunt zetten. Nu kun je de volgende opdrachtjes proberen:

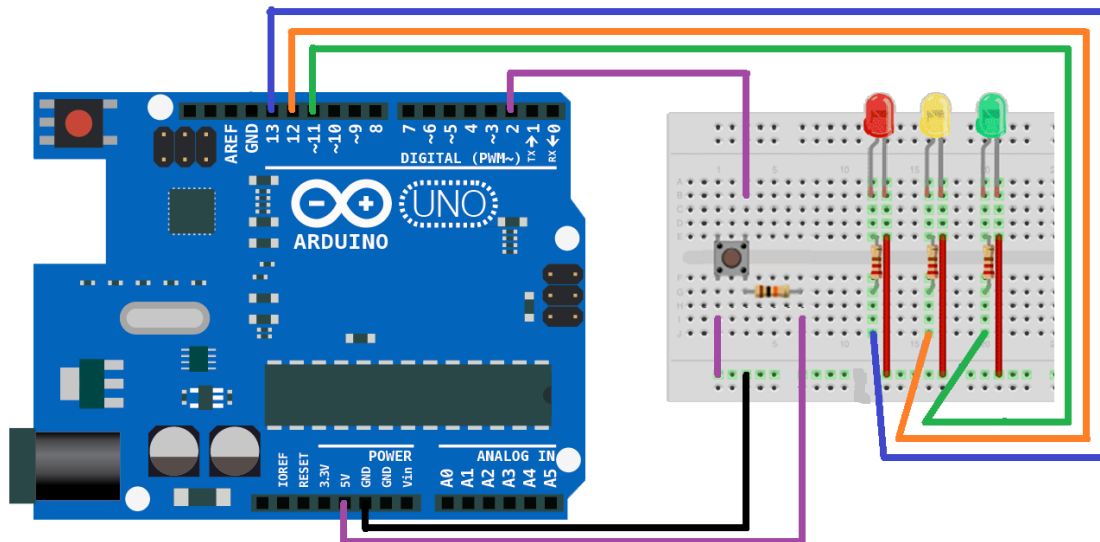
Opdrachten

		Gelukt?
a.	Maak een programmaatje dat ervoor zorgt dat de LEDs om de beurt aan staan (dus éérst alleen de eerste LED, dan na één seconde alleen de tweede LED, dan na één seconde alleen de derde LED en dat steeds opnieuw).	
b.	Maak een programmaatje dat het volgende doet: - eerst brandt alleen de eerste LED - na een seconde branden de eerste en tweede LED samen - na de volgende seconde branden alle drie de LEDs samen - na de volgende seconde staan alle LEDs uit - na een seconde begint alles weer opnieuw	
c.	Het eerste en derde LEDje knipperen samen, steeds één seconde aan en één seconde uit. Het middelste LEDje knippert sneller: steeds een halve seconde aan en een halve seconde uit.	

3. Verkeerslicht

We gaan, in stapjes, een verkeerslicht maken. Daarvoor combineren we de drukknop uit het vorige deel met de drie LEDs uit opdracht 2. Het is nu natuurlijk het mooiste wanneer je een rode, gele en groene LED gebruikt.

De schakeling die je nodig hebt ziet er uit als in de tekening op de volgende pagina. Let op: de weerstandjes zijn verschillend. Het weerstandje dat je nodig hebt voor de schakelaar is 10 kilo-ohm (bruin-zwart-oranje). Het rode licht zit op poort 13, het gele licht op poort 12 en het groene licht op poort 11.



Je gaat nu een nieuw programmaatje schrijven, ga naar 'bestand' → 'nieuw' om een leeg scherm te openen. Typ onderstaand programma in, en let op: het komt heel precies!

```
int buttonState = 0;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  buttonState = digitalRead(2);
  if (buttonState == HIGH) {
    digitalWrite(13, HIGH);
  }
  else {
    digitalWrite(13, LOW);
  }
}
```

Van de regels die je nog niet kent volgt hier de uitleg:

Programmeercode	Betekent
<code>pinMode(2, INPUT);</code>	Zo 'vertel' je de Arduino dat poortje nummer 2 een ingang (input) is, hier komt een signaal binnen.
<code>buttonState = digitalRead(2);</code>	Het digitale signaal dat van de drukknop af komt noemen we 'buttonState'. Als je het knopje indrukt is 'buttonState' gelijk aan HIGH, als je het knopje loslaat is 'buttonState' gelijk aan LOW.
<code>if (buttonState == HIGH) { digitalWrite(13, HIGH); }</code>	Als ('if') 'buttonState' HIGH is (dus het knopje is ingedrukt) dan moet poortje 13 HIGH worden (rode lampje brandt).
<code>else { digitalWrite(13, LOW); }</code>	En wanneer dat niet zo is ('else') dan moet poortje 13 LOW worden (rode lampje brandt niet).

Upload het programma naar de Arduino en test het. Als het goed is brandt het rode lampje als je op het knopje drukt.

Als je de volgende opdrachten goed uitvoert kom je uiteindelijk bij een verkeerslicht uit. Upload en test je programmaatje na elke opdracht!

Opdrachten

		Gelukt?
a.	Verander het programma zo dat in plaats van het rode lampje het groene lampje gaat branden als je het knopje indrukt.	
b.	Zorg ervoor dat, als je even kort op de knop drukt en dan weer los laat, het groene lampje 5 seconden blijft branden. Hiervoor moet je een regel toevoegen: delay(5000); Het is belangrijk dat je deze regel op de juiste plek in het programmaatje zet, bedenk zelf waar dat is.	
c.	Zorg ervoor dat, als het groene lampje 5 seconden gebrand heeft... - het groene lampje uit gaat. - het gele lampje aan gaat. - het gele lampje 2 seconden blijft branden. Let op: vergeet niet om de Arduino te 'vertellen' dat poort 12 nu ook een uitgang is!	
d.	Zorg ervoor dat, als het gele lampje 2 seconden gebrand heeft... - het gele lampje uit gaat. - het rode lampje aan gaat. Let op: vergeet niet om de Arduino te 'vertellen' dat poort 13 nu ook een uitgang is!	
e.	Het verkeerslicht is nog niet helemaal klaar, je moet er nog voor zorgen dat het rode licht blijft branden als er niks gebeurt. Daarvoor moet je twee dingen doen: - de opdracht die <u>na</u> 'else {' komt aanpassen (zodat het rode licht brandt als er niks gebeurt). - zorgen dat, als je op het knopje drukt, eerst het rode licht uitgezet wordt voordat de groene aan gaat. Hiervoor moet je een regel toevoegen <u>na</u> 'if (buttonState == HIGH) {'.	
f.	Tot slot gaat het verkeerslicht natuurlijk niet meteen op groen, maar pas na drie seconden nadat je op het knopje gedrukt hebt. Verwerk dat in het programma.	