Workshop Arduino

v0.3
Wouter Rusman
Maakplek

Inhoudsopgave

Inleiding electronica

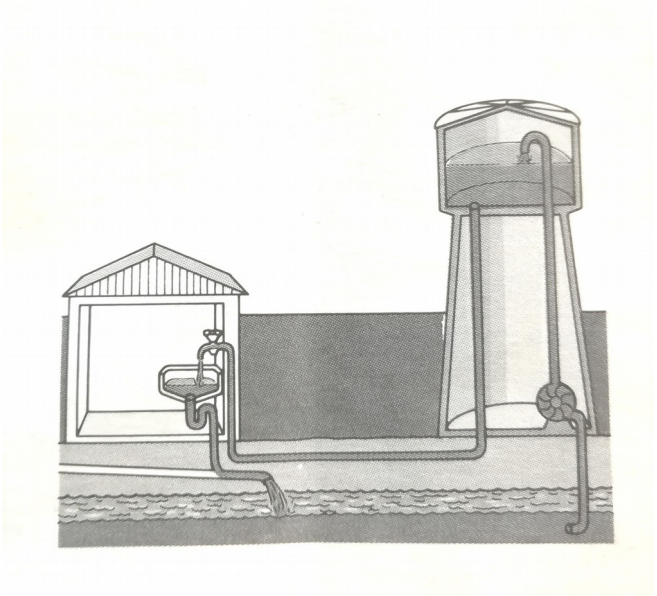
Stroom en spanning

Wanneer er wordt gesproken over electriciteit in het algemeen wordt vaak de term “stroom” genoemd en niet spanning, maar wat zijn stroom en spanning nou eigenlijk?

In de eerste uitleg die ik kreeg over stroom en spanning, werd een vergelijking gemaakt tussen electriciteit en water die eigenlijk heel logisch uitleg geeft over hoe spanning en stroom werken. Ik zal het proberen te verwoorden in mijn eigen woorden.

De vergelijking begint met watertoren die waterdruk geeft vergeleken met een batterij en een aangesloten lampje.

Bij de watertoren is er daadwerkelijk het fysieke hoogteverschil tussen de watertoren en de huizen op de grond waardoor het water door de leidingen naar de huizen wil stromen. Bij elektriciteit gaat het om een hoogteverschil in elektrisch potentiaal tussen twee punten, zoals tussen de pluspool en minpool van een batterij. Dit verschil wordt “spanning” genoemd en heeft de eenheid “voltage” of kortweg “volt” met de aanduiding “U”. De hoogte van het voltage is te vergelijken met hoe hoog de watertoren is; hoe hoger de toren, hoe hoger de waterdruk. De batterij kun je vergelijken met een pomp die water in de toren pompt.



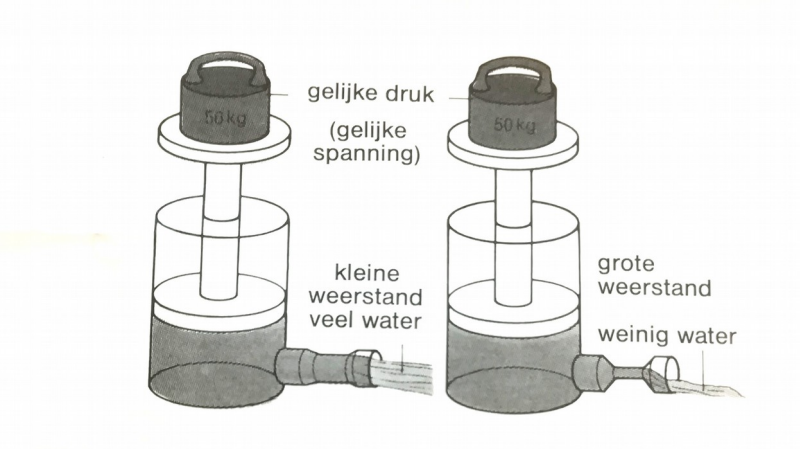
Electriciteit wil net zoals water ook graag stromen, uiteraard niet vanuit een toren naar huizen maar van een punt met een hoger elektrisch potentiaal naar een punt met een lager elektrisch potentiaal, bijvoorbeeld van de pluspool naar de minpool van een batterij. Deze stroom van elektronen tussen deze punten wordt “stroom” genoemd en heeft de eenheid “Ampere” en de aanduiding “I”.

De hoeveelheid stroom is te vergelijken met hoeveel water in een keer naar beneden kan lopen door de leidingen en hoeveel kracht er dus achter het water staat. Des te meer stroming of stroom, des te meer kracht, maar ook hoe eerder de toren leeg is. Ditzelfde geldt voor een batterij; bij meer stroom is de elektriciteit krachtiger maar is de batterij ook eerder leeg.

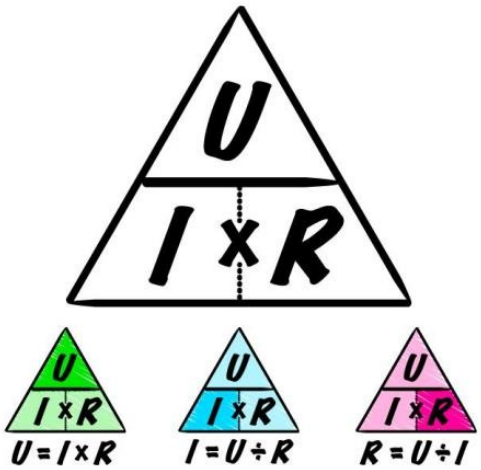
Te veel kracht in de waterstroom kan zorgen dat de leiding de stroming niet meer aankan en dat deze knapt. Dit kan ook bij elektriciteit gebeuren als er meer stroom door de draad wil dan mogelijk is. De draad zal dan beginnen te gloeien en uiteindelijk misschien zelfs doorbranden.

Weerstand

Doorgaand op het verhaal over spanning en stroom is ook weerstand uit te leggen door een vergelijking te maken met water.
Als in de leiding een versmalling wordt aangebracht kan er minder water door de leiding stromen, dit hangt samen met breedte van de versmalling en de waterdruk.



Zo kan er bij gebruik van een weerstand in een elektrische kring ook minder elektriciteit stromen tussen de twee punten afhankelijk van de weerstandsgrootte en het voltage. De stroom die kan lopen is te berekenen met de wet van Ohm.

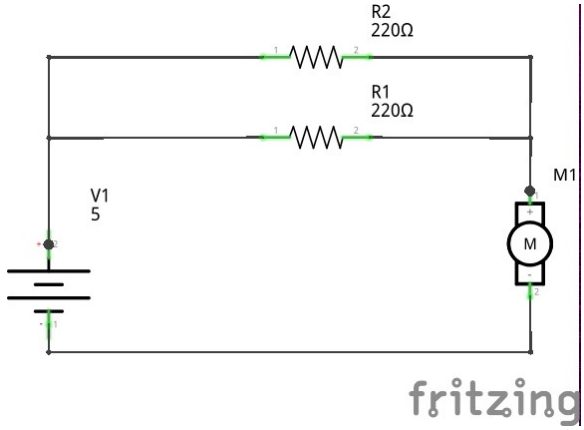


R is de weerstand in Ohm, I de stroom in Ampere en U de spanning in volt.

Als we bijvoorbeeld weten dat de spanning 5 volt is, en de weerstand 10 Ohm, dan is de stroom die door de weerstand kan lopen dus (I=U / R) 5 gedeeld door 10 = 0,5 Ampere. (ofwel 500 milli Ampere). Als we andersom willen dat er een stroom van maximaal 5 micro Ampere kan lopen door de weerstand dan kunnen we berekenen dat de weerstand (R=U / I) 5 gedeeld door 0,0005 = 10.000 Ohm ofwel 10 kilo Ohm moet zijn.
Met deze formules kun je berekenen wat je nodig hebt om bepaalde componenten beschermen tegen te grote stromen, of om de spanning die je wilt gebruiken te verlagen of stabiliseren.

Parallelle weerstanden

Weerstanden kunnen ook parallel aan elkaar worden gezet in een elektronisch circuit. Door de twee weerstanden kan allebei een deel van de totale stroom lopen, de stroom die totaal kan lopen is de som van de twee stromen die je volgens de wet van Ohm kan berekenen.



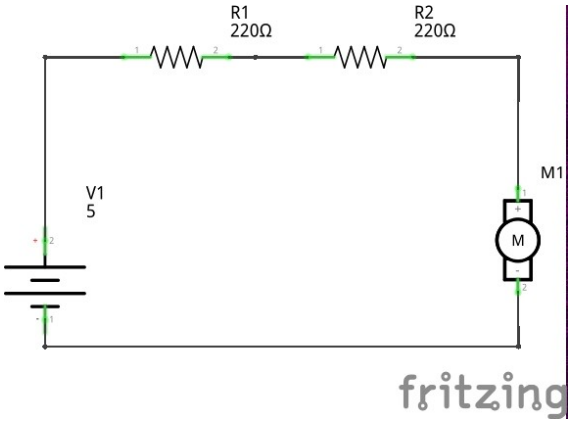
I = U / R
I = 5 / 220
I = 0,0227 A = 22 mA

Itotaal = 22 ma + 22ma = 44 mA

Bij een parallel schakeling van weerstanden staat op beide weerstanden dezelfde spanning omdat ze beide op hetzelfde punt zijn aangesloten. In schema hier boven zal op beide weerstanden 5 volt te meten zijn aan de uiteinden.

Seriële weerstanden

Weerstanden kunnen ook seriëel achter elkaar worden gezet in een elektronisch circuit. De weerstanden gelden beide in dezelfde “leiding” en worden daarom bij elkaar opgeteld.
De stroom kun je berekenen door in de wet van Ohm beide weerstanden bij elkaar op te tellen.



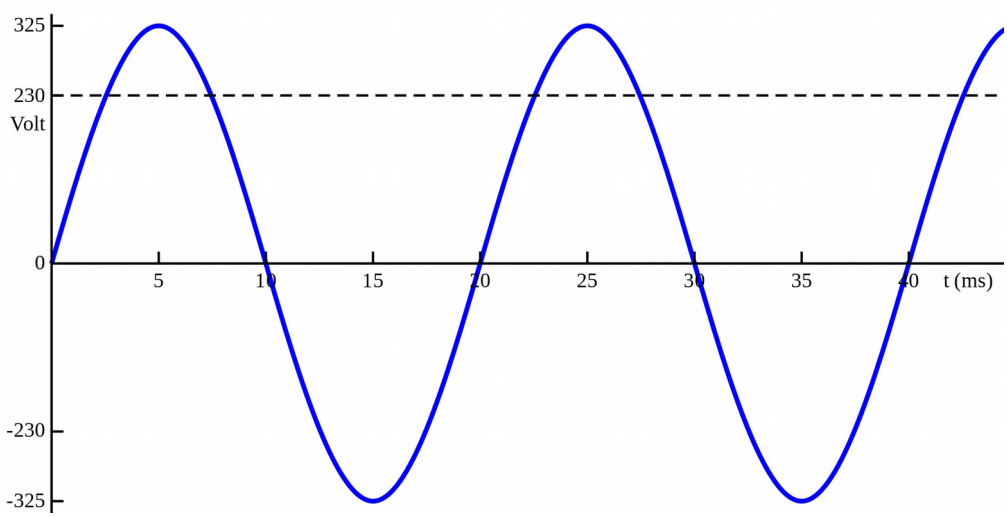
I = U / R
I = 5 / (220+220)
I = 0,0114 A = 11 mA

Bij een serie schakeling van weerstanden staat niet op beide weerstanden dezelfde spanning omdat ze na elkaar zijn aangesloten. De spanning zal zich verdelen over de weerstanden, er wordt dus ook wel gesproken van een spanningsdeler. In het schema hier boven zal (omdat de weerstanden gelijk zijn) op elke weerstand 2,5 V te meten zijn aan de uiteinden.

Analoge signalen

Een goede definitie van de term analoog is “met een eindig aantal traploze waarden in een continuüm”. In andere woorden betekent dit dat het analoog signaal voor onbepaalde tijd elke waarde binnen vastgestelde grenzen aan kan nemen. Voor de komst van de computer was de meeste elektronica analoog. De inmiddels ouderwetse radio, telefonie en ook televisie zijn allemaal voorbeelden van analoge technieken.

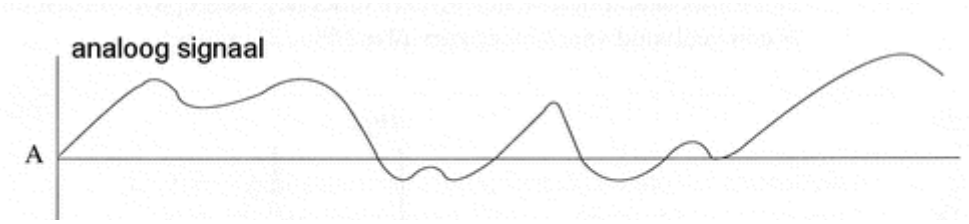
De meeste mensen zullen bij analoog een sinusgrafiek voor ogen hebben zoals de wisselspanning uit het stopcontact.



Dit signaal varieert oneindig met traploze waarden tussen -325 en +325 volt.

* De oplettenden zien dat de maximale waarden hoger zijn dan de 230v die iedereen gewend is, hoe komt dat? Door de gemiddelde spanning (RMS waarde) te berekenen kom je hierdoor op een gemiddeld spanningsverschil tussen beide punten van 230v.

Een analoog signaal kan echter zoals eerder aangegeven traploos elke willekeurige waarde tussen twee limieten aannemen. Het signaal kan er dus ook als volgt uit zien.



Dit zou bijvoorbeeld het signaal van een thermometer kunnen zijn die de temperatuur omzet in een spanning of de spanning die van een fietsdynamo af komt tijdens een ritje naar huis door de spits.

Digitale signalen

Digitale signalen hebben maar twee waarden; een 1 (hoog) of een 0 (laag). Maar wat bepaalt deze waarden en hoe meten we dit?

Om het lastig te maken zijn er verschillende soorten digitale elektronica in omloop met verschillende voorwaarden of een signaal een digitale “1” of een “0” is.

– TTL (Transistor-Transistor Logic)
TTL chips (Integrated Circuits ofwel IC’s) zijn de eerste vorm van digitale electronica componenten en gebruiken intern transistors. Deze ondertussen verouderde vorm van digitaal schakelen gebruikt relatief veel stroom en zet deze extra verbruikte energie om in warmte, twee minpunten van deze techniek dus. Ook de snelheid is beperkt, wat het dus ongeschikt maakt voor het gebruik met snellere processoren.
TTL IC’s beginnen meestal met de code 74XX (bijv. 7420) en gebruiken een voedingspanning van 5 volt. TTL heeft in zijn specificatie staan dat voor de in- en uitgangen andere waarden een digitale “0” of “1” vertegenwoordigen.

Waarde	1 – Hoog	0 – Laag
IC Ingang	>= 2 volt	=< 0,8 volt
IC Uitgang	>= 2,4 volt	=< 0,4 volt

Zoals je ziet zijn er marges ingebouwd tussen de uitgangs- en ingangsspanningen, dit is om verliezen in de verbinding tussen de IC’s op te vangen en het zo toch goed te laten functioneren.

– CMOS (*Complementary Metal Oxide Semiconductor logic)*
CMOS chips maken intern gebruik van de nieuwere FET (Field Effect Transistor) techniek wat het stroomverbruik, en dus ook warmteontwikkeling, vergeleken met de TTL IC’s aanzienlijk verminderd. Een bijkomend voordeel is het kunnen verlagen van de spanning naar 3,3v wat de warmteontwikkeling nog verder doet dalen en daardoor het schakelen op hogere frequenties en zo snellere processoren mogelijk maakt. CMOS IC’s kunnen dezelfde codering en functie hebben als TTL IC’s met de toevoeging “CD” voor de nummers (bijv. CD4011), maar vaak is de pinbezetting totaal anders waardoor ze niet 1-op-1 verwisseld kunnen worden met TTL IC’s.

– TTL HC (TTL Highspeed CMOS)
Vanwege de vraag naar TTL pin-compatible componenten met CMOS techniek is er een nieuwe serie op de markt gekomen welke wel pin-compatible was met TTL IC’s welke zo in oude apparatuur konden worden verwisseld met TTL IC’s. Deze serie kenmerkt zich door de “HC” codering tussen de eerste twee en latere cijfers (bijv. 74HC20). Deze IC’s kunnen op zowel 5 als 3,3 volt werken wat ze prima geschikt maakt om oude IC’s te vervangen door zuinigere nieuwe.

Wel moeten alle IC’s dan vervangen worden want bij een lagere voedingspanning heeft de CMOS techniek ook andere spanningen die gelden als digitale “0” en “1” wat door elkaar gebruiken weer lastig maakt:

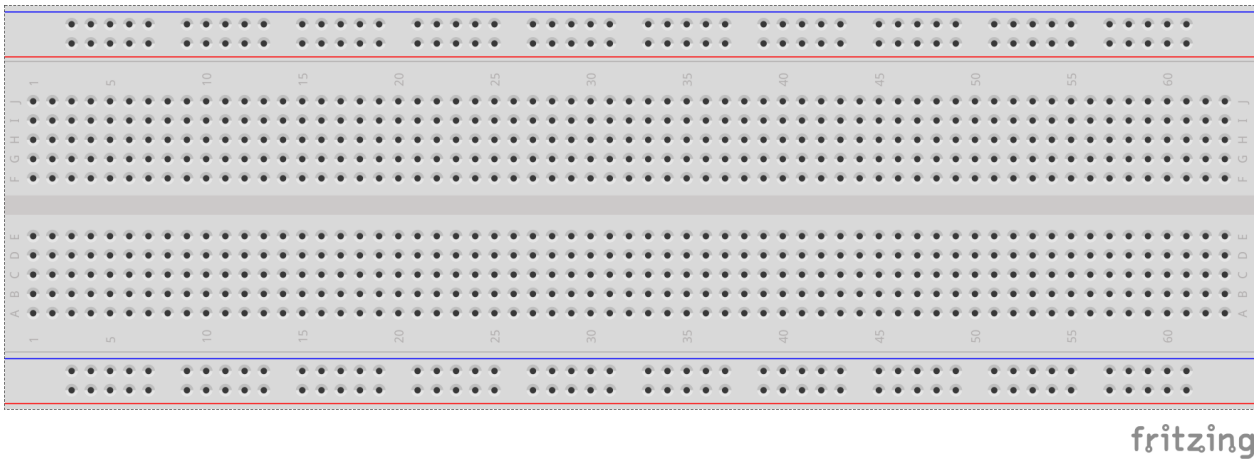
Waarde	1 – Hoog	0 – Laag
IC Ingang/Uitgang	>= 3,5 volt	=< 1,5 volt

Voor een gemixte situatie is de HCT serie uitgebracht (bijv. 74HCT20) welke zowel TTL als CMOS schakelniveaus respecteert en dus in combinatie met beide voltages goed werkt. De snelheid is daarentegen weer gelijk aan de TTL serie, dus in combinatie met hogere snelheden werkt dit weer niet goed.

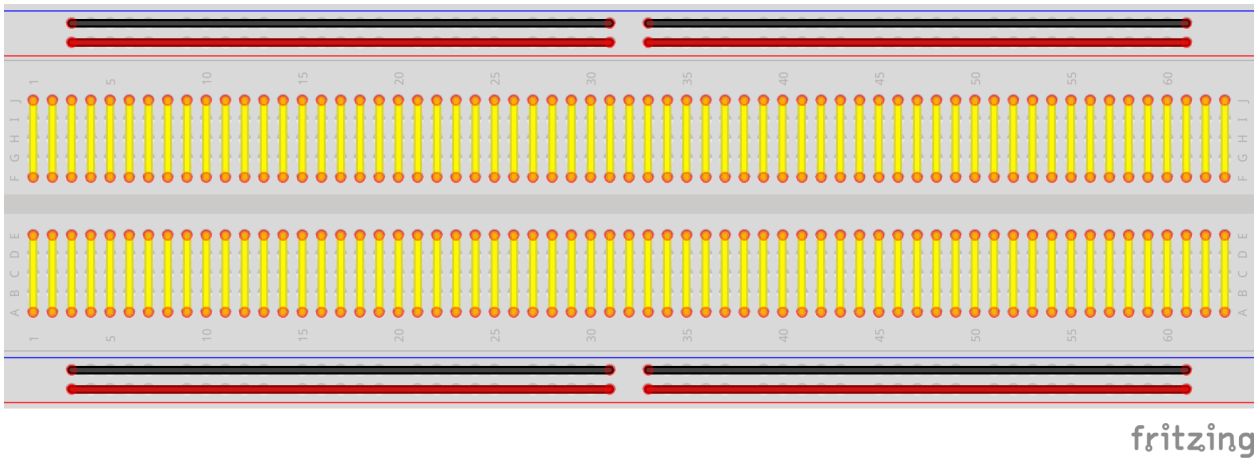
Beschrijving van de gebruikte onderdelen

Het breadboard

Het breadboard is een plastic bord met 830 insteek gaatjes waar je de pinnen van bijna alle micro-elektronica componenten in kunt steken zodat ze verbonden zijn met andere gaatjes waar pinnen van andere componenten in kunnen zitten. Op deze manier is snel een schakeling op te bouwen of aan te passen zonder te hoeven solderen.



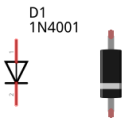
Hoe en welke gaatjes met elkaar zijn verbonden zie je hier.



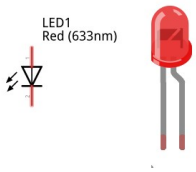
Alle gaatjes in het middelste vlak zijn vertikaal verbonden, de gaatjes in de buitenste rijen met rode en blauwe streep ernaast zijn horizontaal verbonden.
Bij sommige breadbords zijn deze buitenste rijen in het midden van elkaar gescheiden en zijn dus niet alle gaatje met elkaar verbonden. Dit is meestal te zien aan een onderbroken blauwe en rode streep naast de rij pinnen.

De L.E.D.

De term LED is eigenlijk de afkorting L.E.D. wat staat voor Light Emitting Diode.
Een diode is een elektronica component die de stroom maar in één bepaalde richting doorlaat en andersom blokkeert. In een schema ziet een diode er dus ook uit als een pijl met een streep.
Stroom kan altijd vanaf de anode naar de kathode lopen (met de pijlrichting mee) maar niet van de kathode naar de anode (tegen de streep). Een ezelsbruggetje dat hier ook voor wordt gebruikt is “KNAP”; Kathode Negatief, Anode Positief.



Bij de diode is de richting te herkennen aan de ring die op de diode is aangebracht, deze komt overeen met de streep in het schema.



Net zoals de diode heeft een LED ook een kathode en een anode, alleen zijn deze anders te herkennen. Als je van bovenaf naar de LED kijkt, zie je dat de ronde vorm aan één kant afgeplat is, dit komt overeen met de streep het schema. Ook is de kathode (bij een nieuwe LED) te herkennen aan de korte poot, dus kort = kathode. In het schema symbool hebben ze aan de diode een of twee kleine pijltjes aan toegevoegd om aan te geven dat hij licht uitstraalt.

Het soort licht dat een LED uitstraalt kan rood, wit, blauw of een tussenliggende kleur zijn, maar ook infrarood of ultraviolet. Door gebruik te maken van een High Energy LED en speciale lenzen kunnen zelfs laser LED’s worden gemaakt zoals gebruikt in CD, DVD en bluRay of kleinere laser graveer machines.

Een fout die vaker wordt gemaakt is dat bij het aansluiten van een LED alleen naar het voltage wordt gekeken dat de LED nodig heeft. Uiteraard heeft een LED wel een bepaalde werkspanning om te kunnen gloeien maar belangrijker is hoeveel stroom de LED te verwerken krijgt. Daarom kun je nooit een LED aansluiten zonder een weerstand er mee in serie te zetten. Aan welke poot van de led de weerstand zit is niet belangrijk. In onderstaande tabel is van de meest voorkomende LED’s de stroom (IF Typical) en het werk voltage (VF Typical) genoteerd, hiermee is dus ook de weerstand te berekenen die gebruikt moet worden bij een bepaald voltage.

3mm LEDs

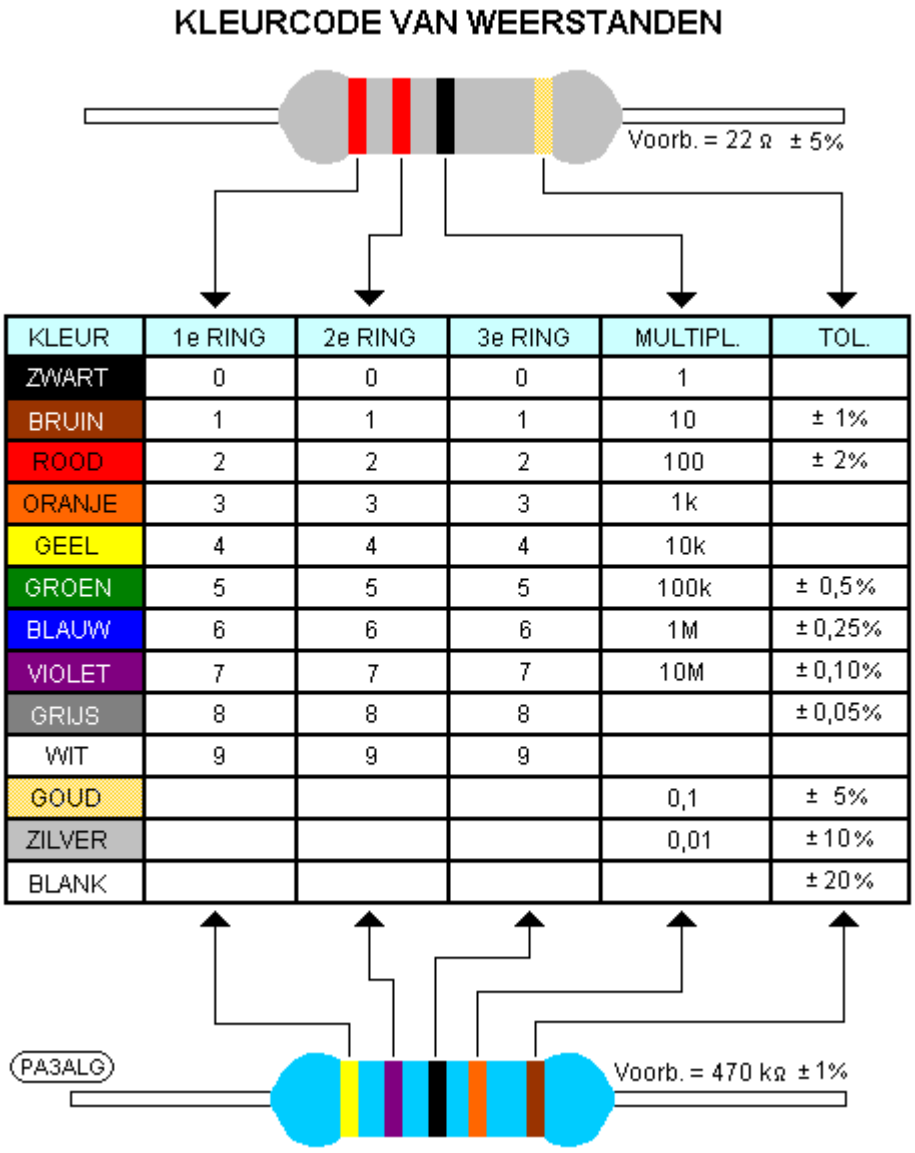
Optical and Electronic Characteristics								Max Continuous	Preferred Value Series Resistor (ohms) for		
Size (mm)	Viewing Angle (Degrees)	Colour	Lens	Wavelength, Chromatic (nm), Co-ordinates	IF Typical (mA)	VF Typical (V)	IV Typical (mcd)	IF Max (mA)	5VDC	9VDC	12VDC
3	45	Red	Diffused	650	15	2.3	40	15	180	470	680
3	15	Red	Waterclear	660	20	1.8	1500	30	160	360	510
3	20	Red	Waterclear	625	20	2.0	2100	50	150	360	510
3	15	Red	Waterclear	625	20	2.1	7000	30	150	360	510
3	45	Orange	Diffused	625	20	1.9	35	30	160	360	510
3	50	Yellow	Diffused	585	20	2.1	10	20	150	360	510
3	20	Yellow	Waterclear	588	20	2.0	3000	50	150	360	510
3	15	Yellow	Waterclear	588	20	2.2	6500	50	150	360	510
3	50	Green	Diffused	573	20	2.3	40	20	130	330	470
3	20	Green	Waterclear	568	20	2.1	500	30	150	360	510
3	20	Green	Waterclear	520	20	3.2	6000	20	91	300	470
3	15	Aqua	Waterclear	505	20	3.5	4000	30	75	270	430
3	15	Blue	Waterclear	465	20	3.3	1500	20	82	300	430
3	15	Blue	Waterclear	470	20	3.2	3700	30	91	300	430
3	15	White	Waterclear	0.31/0.32	20	3.2	1000	20	91	300	430
3	20	White	Waterclear	0.31/0.32	20	3.2	5000	30	91	300	430
3	15	White	Waterclear	0.31/0.32*	25	3.4	12000	25	62	220	360

Weerstanden

Weerstanden zijn meestal te herkennen aan hun gekleurde ringen, waarmee de weerstandswaarde wordt aangegeven. Er zal naast het groepje gekleurde ringen vaak ook apart een gekleurde, zilveren of gouden ring op de weerstand zijn gedrukt, hiermee wordt de nauwkeurigheid aangegeven.

Bij deze workshop maken we gebruik van 2 weerstandswaarden: 100 kOhm (100.000 Ohm) en 330 Ohm.

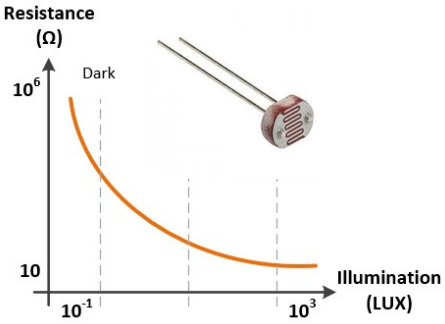
In onderstaande tabel wordt aangegeven hoe je de waarde van een weerstand kunt berekenen met behulp van de gekleurde ringen.



Een ezelsbruggetje om de volgorde van de ringen te onthouden kan zijn:

Zij	Bracht	Rozen	Op	Gerrits	Graf	Bij	Vies	Grauw	Weer
Zwart	Bruin	Rood	Oranje	Geel	Groen	Blauw	Violet	Grijs	Wit
0	1	2	3	4	5	6	7	8	9

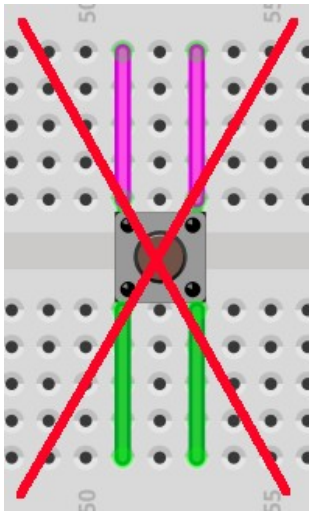
De LDR (light dependent resistor)

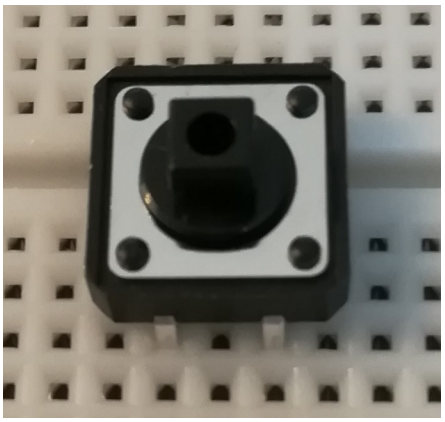


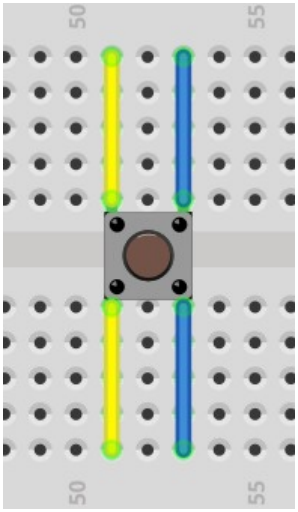
Naast weerstanden met een vaste waarde zijn er ook weerstanden die afhankelijk van de stand van een draaiknop of schuif (potentiometer) of van bijvoorbeeld temperatuur of lichtsterkte een andere waarde krijgen.
De LDR is een type weerstand waar de waarde afhankelijk is van lichtsterkte; deste meer licht er op valt, deste minder weerstand heeft de LDR.

De drukknop

De drukknop in deze workshop is een zogenaamd maak-contact, dit houdt in dat hij alleen de uiteinden verbind op het moment dat je hem indrukt. Er zitten 4 pinnen aan de drukknop waarvan twee aan elk uiteinde zijn verbonden. Je moet dus wel opletten hoe je de drukknop aansluit anders zal hij continue verbinding maken.







Het plaatje met de paarse en groene lijnen is hoe je het zou verwachten maar dit klopt dus niet!
Het plaatje met de blauwe en gele lijnen is hoe de pinnen van de drukknop wel zijn doorverbonden.
Zodra je op de knop drukt zijn alle pinnen met elkaar verbonden.

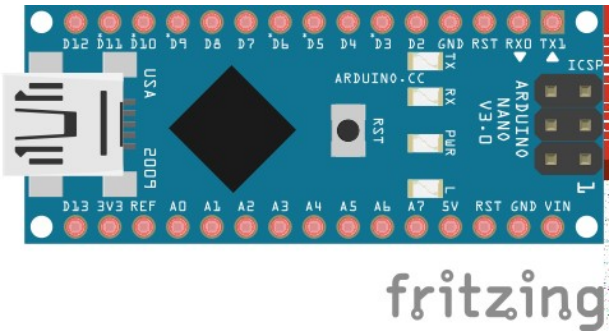
De Arduino

Het belangrijkste in deze workshop is uiteraard de arduino, maar wat is nou eigenlijk een arduino?

Een arduino is een mini-processor van de producent Atmel welke met een minimum aan externe componenten te gebruiken is. De processor heeft intern een stukje opslag voor het programma dat je er in wilt stoppen (opslag – ROM) en een stukje opslag voor tijdelijke variabelen van het programma dat je er in gestopt hebt (werkgeheugen – RAM).

Het “uploaden” van het programma dat je hebt gemaakt naar de arduino gebeurt door een seriëel naar TTL converter tussen je computer en de arduino. In sommige arduino modellen zit deze converter ingebouwd, in sommige moet je hiervoor een aparte converter aansluiten op specifieke pinnen van de arduino.

Elke arduino heeft meerdere in- en uitgangs pinnen (I/O pinnen) waarmee analoge en/of digitale signalen gemeten kunnen worden of uitgestuurd kunnen worden. Afhankelijk van het model arduino variëert het aantal I/O pinnen en de kracht van de processor die de pinnen aan kan sturen.
In deze workshop maken we gebruik van de arduino Nano.



Dit is een kleiner model arduino met een Atmega328 Processor, 14 Digitale I/O pinnen (D0 – D13) waarvan er 6 ook als analoge uitgang kunnen worden gebruikt (D3, D5, D6, D9, D10, D11) en 8 analoge input pinnen (A0 – A7).
Voor de gevanceerdere programma's zijn D2 en D3 te gebruiken als hardware interrupt.

Voor de meeste hobbyprojectjes zijn deze specificaties meer dan genoeg en door zijn kleine formaat is hij makkelijk in een breadboard te gebruiken of ergens in te bouwen. De seriëel – TTL converter zit ingebouwd dus met een mini-USB snoer is simpel een programma te uploaden.
De arduino Nano werkt op 5 volt, let dus op met sensoren die op 3,3 volt werken.
Op de Vin pin kan een spanning van 7-12 volt worden aangesloten, de ingebouwde spanningsregelaar zal hier zelf 5 volt van maken.

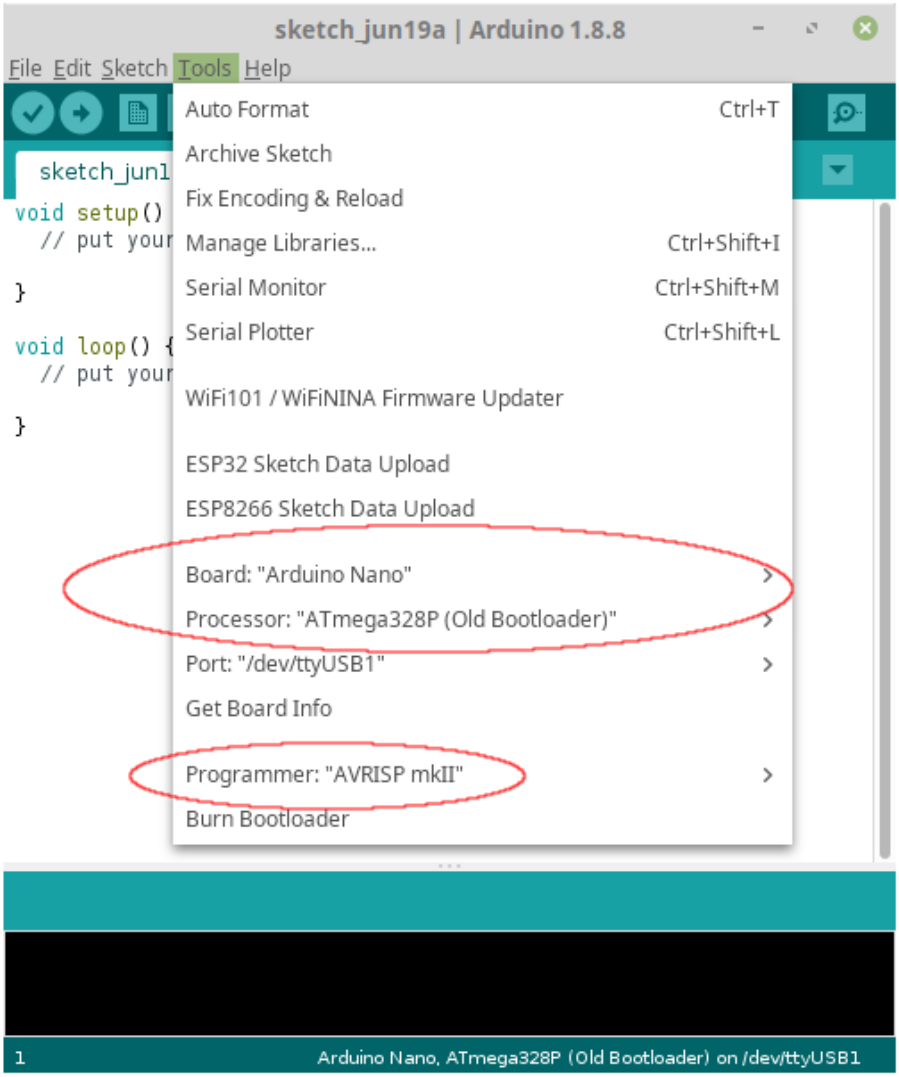
De analoge ingangen maken gebruik van een 10 bits Analoog naar digitaal converter (een digitale processor kan zelf geen analoge signalen lezen) welke het signaal tussen 0 – 5 volt dus vertaalt naar een waarde van 0 – 1024.

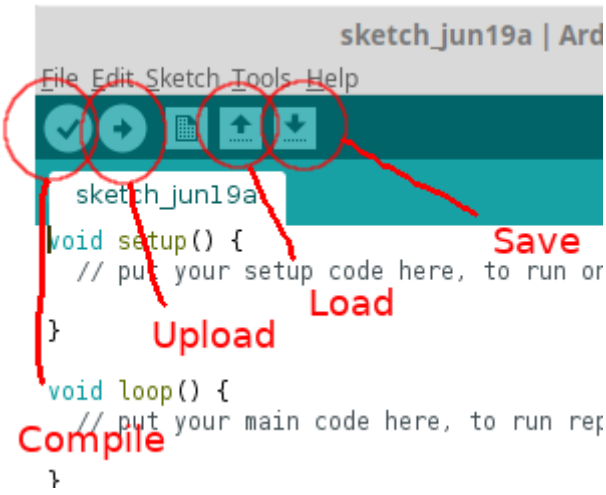
De analoge uitgangen zijn geen "echte" analoge uitgangen maar PWM (Pulse Width Modulation) signalen die afhankelijk van de waarde die wordt geschreven naar de pin (0-255) een gemiddelde spanning van 0 – 5 volt op de pingeven.

Het programma voor een arduino kun je maken in verschillende programma's beschikbaar voor windows, linux en mac. In deze workshop maken we gebruik van de Arduino IDE (Integrated Development Environment) die door de bedenkers van de arduino wordt ontwikkeld.

Na het starten van de arduino IDE zijn er enkele instellingen en knoppen belangrijk voordat we kunnen beginnen.

De Arduino's die we bij deze workshop gebruiken zijn klonen uit china en deze worden nog steeds voorzien van een oud type bootloader (firmware). Deze moet in de IDE dus goed gekozen worden.
De poort is afhankelijk van de computer en soms van de USB poort die wordt gebruikt.





Compile: compileert de code en geeft eventueel foutmeldingen als dit niet lukt.

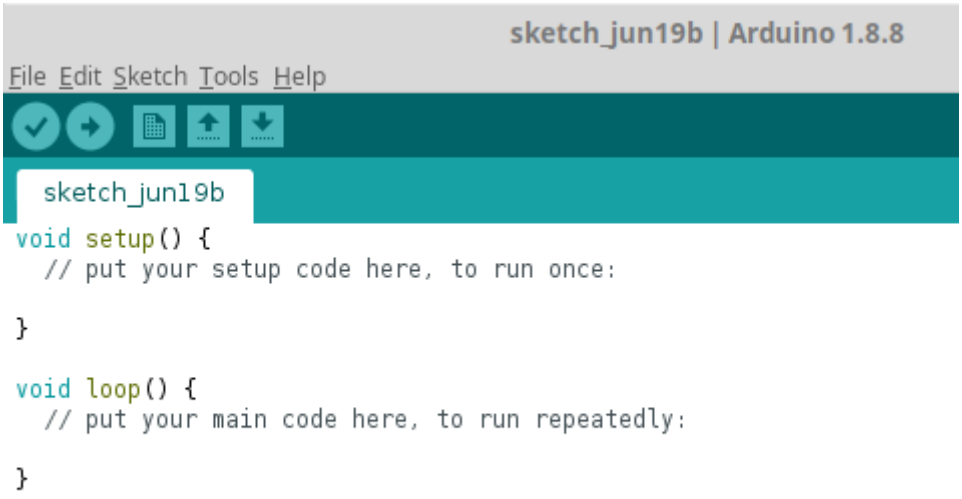
Upload: compileert de code en upload deze naar de arduino als dit geen fouten geeft.

Load – laad een arduino sketch (code) uit een bestand

Save – sla de arduino sketch op in een bestand.

Opbouw van een arduino sketch

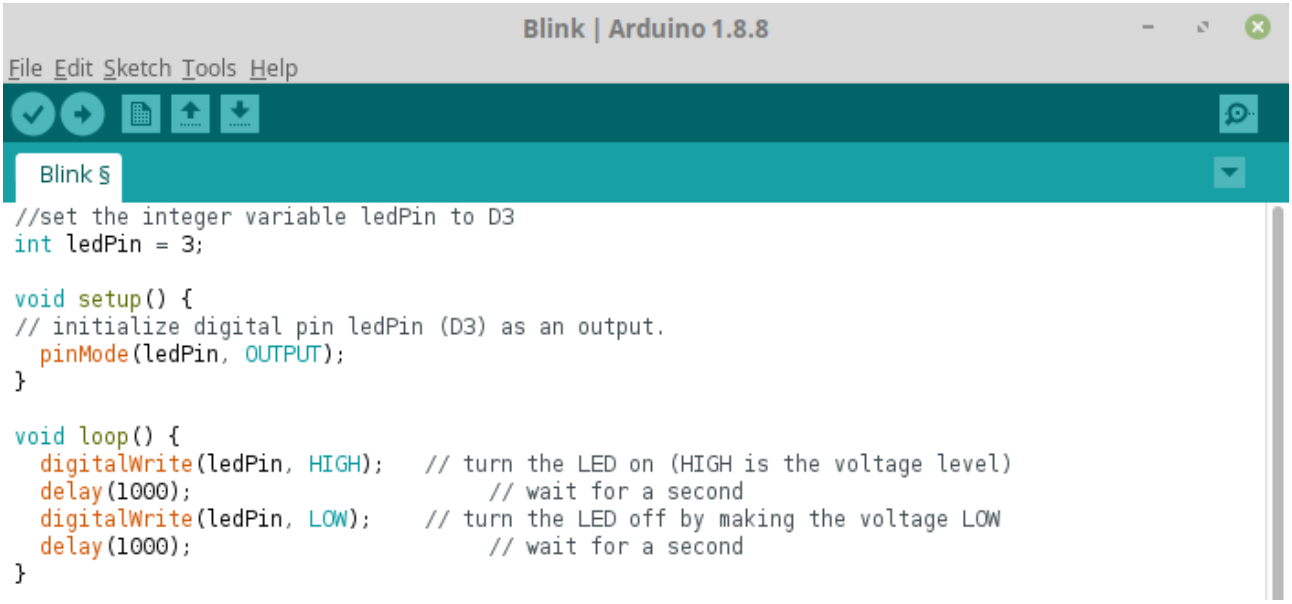
In een sketch voor een arduino zijn twee delen essentieel: de setup en de loop.



In de setup plaats je commando's of routines die eenmalig uitgevoerd moeten worden als de arduino opstart.

In de loop plaats je commando's die telkens weer in een lus moeten worden uitgevoerd.
Deze loop wordt oneindig uitgevoerd zodra de arduino aan staat.

Bij beide onderdelen zie je accolades ({ en }) staan, deze geven het begin en het eind aan van de subroutine. Elk commando dat in deze subroutine word gegeven sluit je af met een puntkomma.



In dit voorbeeld wordt in de setup pin D3 die is verbonden aan het ledje gedefiniëerd als OUTPUT. Daarna wordt in de loop telkens de pin "hoog" gemaakt, wordt 1000 milliseconden gepauzeerd, daarna wordt de pin "laag" gemaakt en wordt er weer 1000 milliseconden gepauzeerd. Daarna begint de loop opnieuw en wordt weer de pin "hoog" gemaakt, wordt 1000 milliseconden gepauzeerd enz. enz...

Variabelen

Tijdelijk een waarde opslaan in het geheugen zodat je de waarde later weer kunt gebruiken wordt een variabele genoemd. In het voorbeeld zie je ook dat een variabele “ledPin”wordt aangemaakt van het type int (integer) en dat hieraan ook meteen de waarde “3” wordt toegekend. Deze variabele wordt verderop gebruikt door dezelfde naam te typen als naam van een pin.

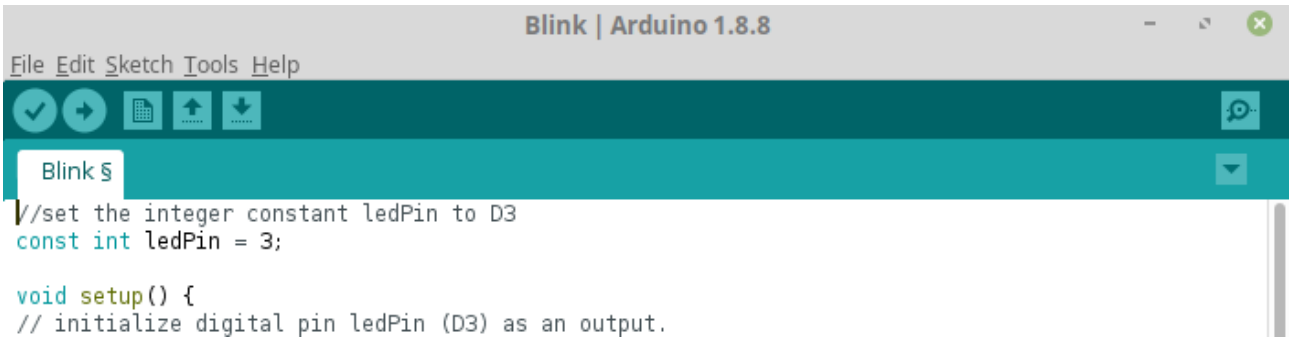
Variabelen kunnen op verschillende manieren worden gedeclareerd; programmawijd of specifiek in een functie of zelfs gedurende het programma. Programmawijd (zoals in het voorbeeld hier boven) heeft als voordeel dat ze overal in je programma zijn te gebruiken. Specifiek in een functie of gedurende het programma kun je ook variabelen aan maken, deze zijn dan alleen geldig binnen dat stuk code. Zorg ervoor dat je duidelijk hebt welke variabelen je gebruikt en welke namen ze je geeft zodat je niet per ongeluk een variabele gaat gebruiken die al in gebruik was voor een andere functie. Namen van variabelen zijn case sensitive, dus ledPin en LedPin zijn niet dezelfde variabele.

Afhankelijk van het soort gegevens dat je tijdelijk wilt opslaan zijn er diverse soorten variabelen beschikbaar:

char – karakter met enkele quotes (bijv. ‘A’) of een woord met dubbele quotes (bijv. “hello”)
byte – 8 bit waarde zonder negatief, 0 tot 255 (2^8)
int – 16 bit waarde met negatief, -32,768 tot 32,767 (- 2^15 tot 2^15)
word – 16 bit waarde zonder negatief, 0 tot 65,535 (2^16)
unsigned int – 16 bit waarde zonder negatief, 0 tot 65,535 (2^16)
short – 16 bit waarde met negatief, -32,768 tot 32,767 (- 2^15 tot 2^15)
long – 32 bit waarde met negatief, -2,147,483,648 tot 2,147,483,647 (- 2^31 tot 2^31)
unsigned long – 32 bit waarde zonder negatief, 0 tot 4,294,967,295 (2^32)
float – 32 bit waarden met negatief en decimalen, 3.4028235E+38 tot -3.4028235E+38
double – hetzelfde als float, alleen op de arduino Due is dit een 64 bit waarde.
String – een array (rij) van karakters, dit type variabele is niet gebonden aan een vaste hoeveelheid geheugenruimte. (bijv. “Hello world”)
Bool – (boolean) waarde van “YES” of “NO”

Constanten

Aan het begin van je programma kun je door gebruik van een constante een naam koppelen aan een waarde. Hierdoor zal deze gedurende het hele programma nooit meer kunnen veranderen.



Dit kun je gebruiken om bijvoorbeeld pinnummers een naam te geven zodat je niet hoeft te onthouden op welke pin ook alweer wat aangesloten is. Als je een andere arduino gaat gebruiken met andere pinnummers kun je makkelijk je code aanpassen, door maar op 1 plek de pinnummers van de constanten te hoeven wijzigen in plaats van door de hele code heen.

If..then..else

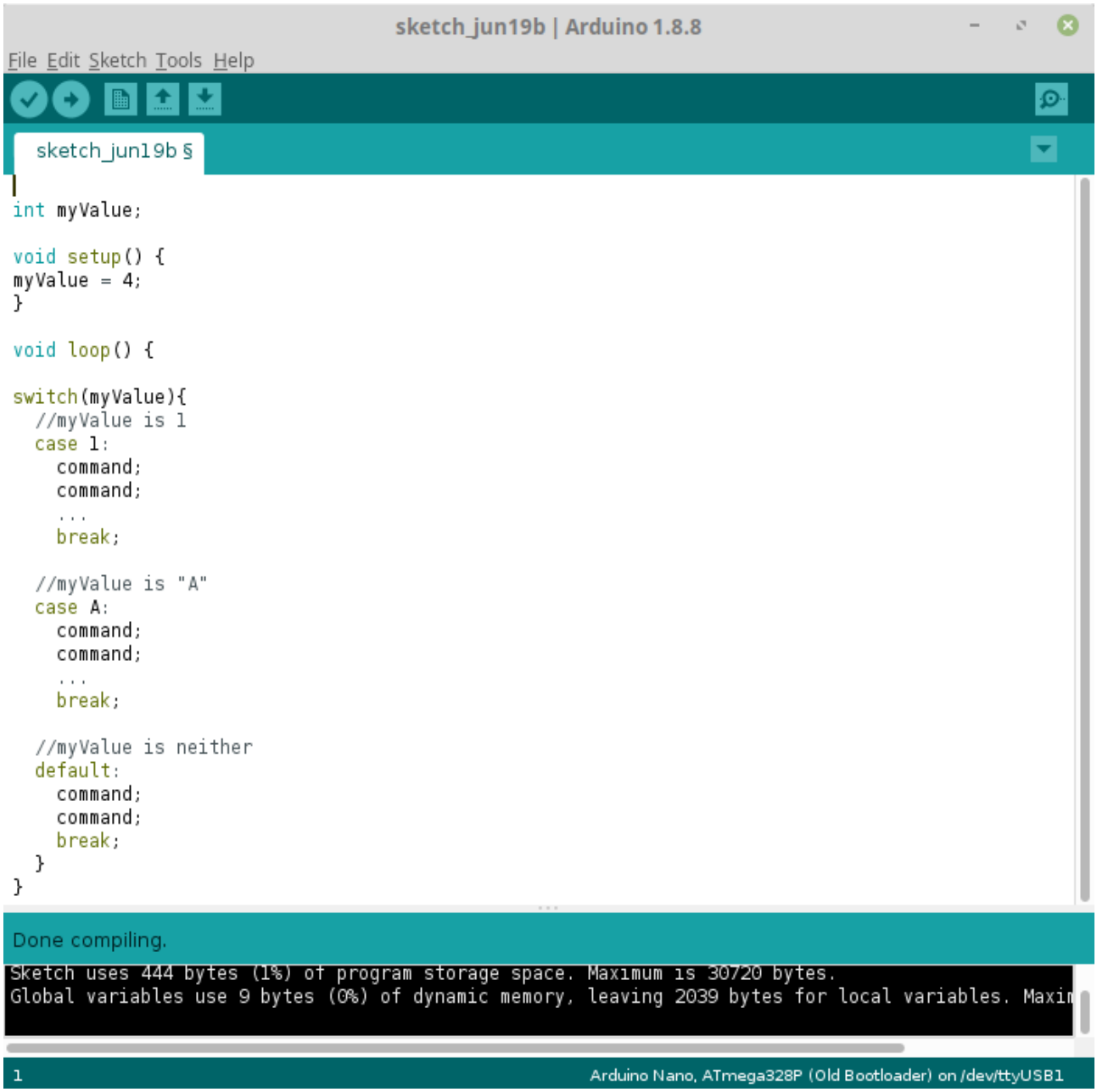
Om in je programma keuzes te kunnen maken op basis van waarden van variabelen kun je een if-loop gebruiken. Om waarden van variabelen en/of constanten met elkaar te vergelijken zijn er enkele comparators beschikbaar:

```
x == y (x is gelijk aan y)
x != y (x is niet gelijk aan y)
x <  y (x is kleiner dan y)
x >  y (x is groter dan y)
x <= y (x is kleiner dan of gelijk aan y)
x >= y (x is groter dan of gelijk aan y)
```

In dit voorbeeld zijn er twee vergelijkingen (“if” en “else if”) of een variabele een bepaalde waarde heeft, en een derde optie (“else”) voor als geen enkele vergelijking klopt.

Switch

Om een variabele sneller op meerdere waarden te kunnen testen kan in plaats van een oneindig “if”..”else if” scenario het commando “switch” gebruikt worden;



Zoals in het voorbeeld kan de vergelijking met zowel de typen integer als char worden uitgevoerd, en wordt optineel “default” uitgevoerd als geen van de vergelijkingen waar is.

While..do

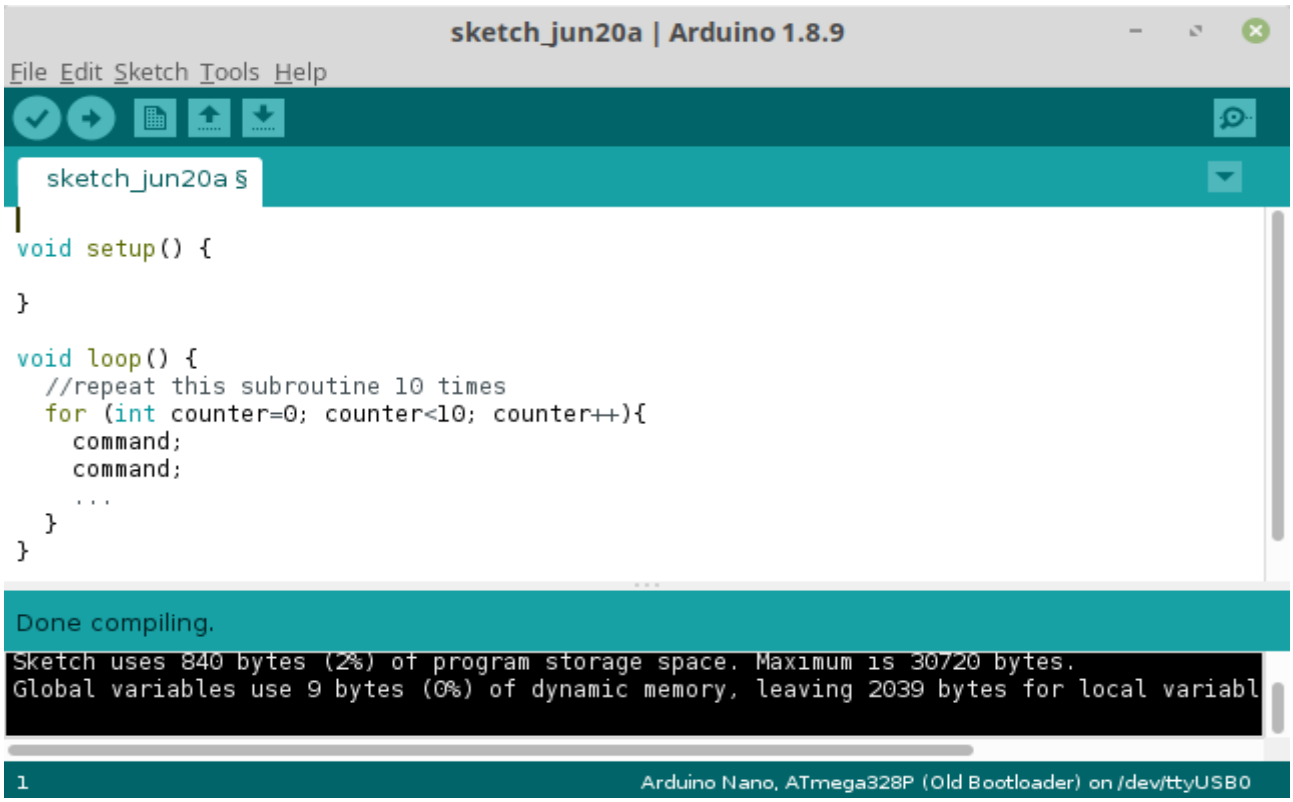
Om een subroutine te blijven herhalen zolang een vergelijking klopt, kun je een while-do loop gebruiken.

Dit voorbeeld herhaalt de while loop totdat myValue groter wordt dan 9. In de loop wordt de variabele myValue elke keer met 1 opgehoogd.

Door de if-loop die in deze while loop is gezet zal myValue echter nooit groot genoeg worden. De if-loop maakt namelijk myValue weer 0 zodra deze 6 is geworden. De while loop zal dus oneindig doorgaan.

For..do

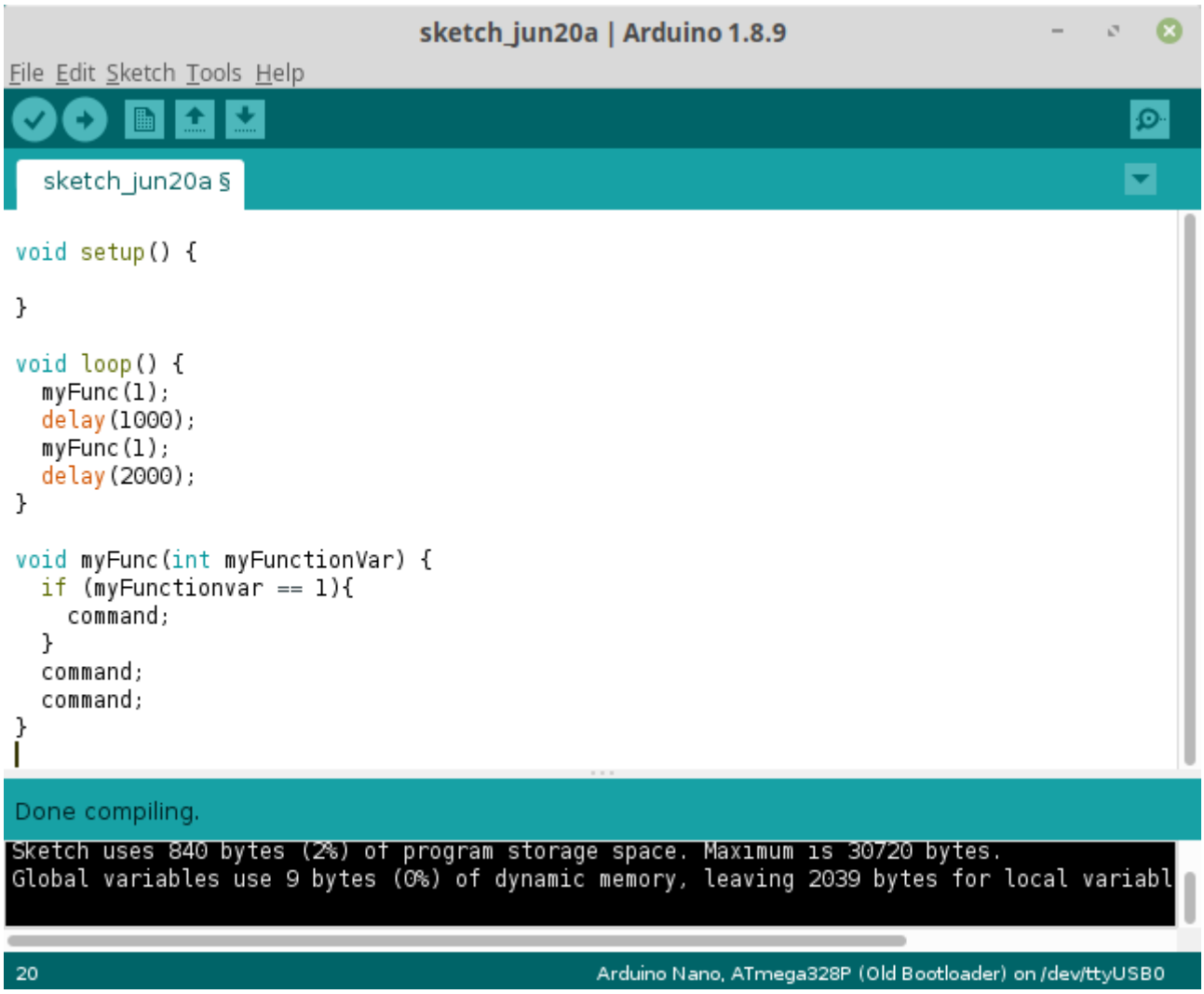
Om een subroutine een bepaald aantal keer uit te laten voeren kun je een for-loop gebruiken:



In dit voorbeeld wordt tijdens het aanroepen van de for loop de integer variabele “counter” aangemaakt en op 0 gezet. Daarna wordt de loop herhaald zolang de waarde van “counter” kleiner is dan 10 en wordt elke keer dat een loop wordt gestart de variabele “counter” met 1 opgehoogd.

Functies

Als een stuk code meerdere malen terugkomt in je sketch kun je er voor kiezen hier een functie van te maken. Deze functie wordt net zoals de setup en loop gedefinieerd met de naam van de functie, gevolgd door haakjes en accolades. In de haakjes kunnen eventuele parameters aan de functie worden meegegeven, maar zoals bij setup en loop het geval is kun je de haakjes ook leeg laten.



In dit voorbeeld wordt een functie “myFunc” aangemaakt met een integer functievariabele “myFunctionVar”. Vanuit de loop wordt de functie aangeroepen met een waarde die door de functie wordt vergeleken met “1”.

Alle commando’s, functies en variabelen zijn terug te vinden in de arduino reference te vinden op :

https://www.arduino.cc/reference/en/

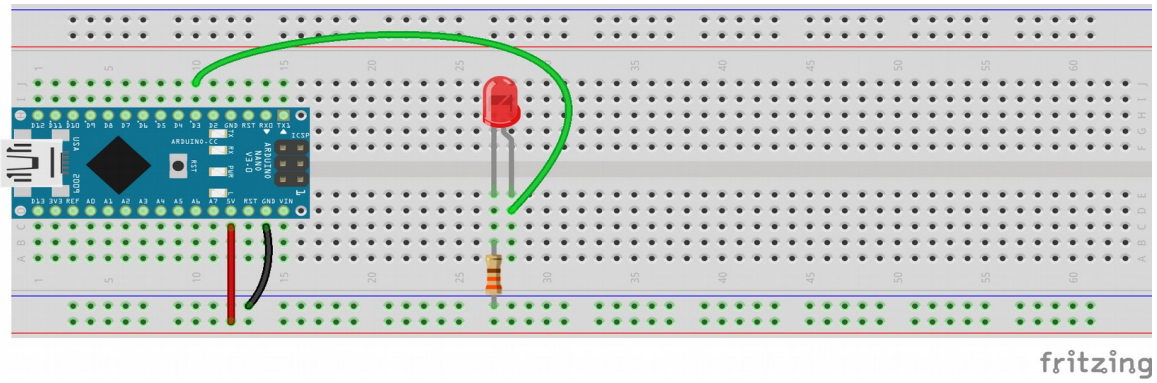
Ook zijn er diverse arduino cheat-sheets te vinden waarop de meest gebruikte functies staan.

De opdrachten

Digitale output naar een I/O pin

Deze opdracht laat zien hoe je een digitale pin “1” of “0” kunt maken. We gebruiken hiervoor pin D3.

Bouw het schema op zoals onderstaand plaatje, de gebruikte weerstand is die van 330 Ohm.



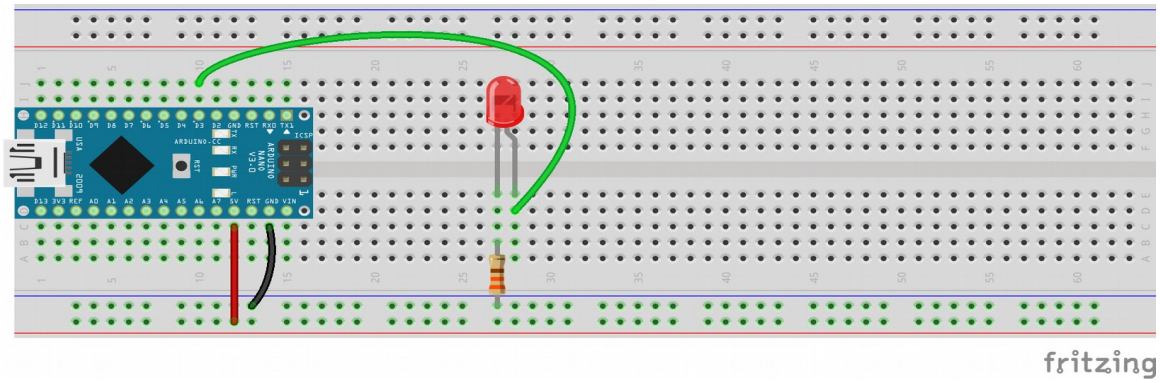
Deze sketch zal pin D3 om en om “1” en weer “0” maken met een pauze van 1 seconde.

– Probeer eens de LED sneller te laten knipperen.
– Probeer eens een variabele of constante te gebruiken voor het pinnummer.

Analoge output naar een I/O pin

Deze opdracht laat zien hoe je een I/O pin een analoge waarde kunt geven, we gebruiken hiervoor pin D3.

Het schema is hetzelfde als hierboven omdat we ook in deze opdracht de LED willen laten branden. De gebruikte weerstand is ook hier die van 330 Ohm.



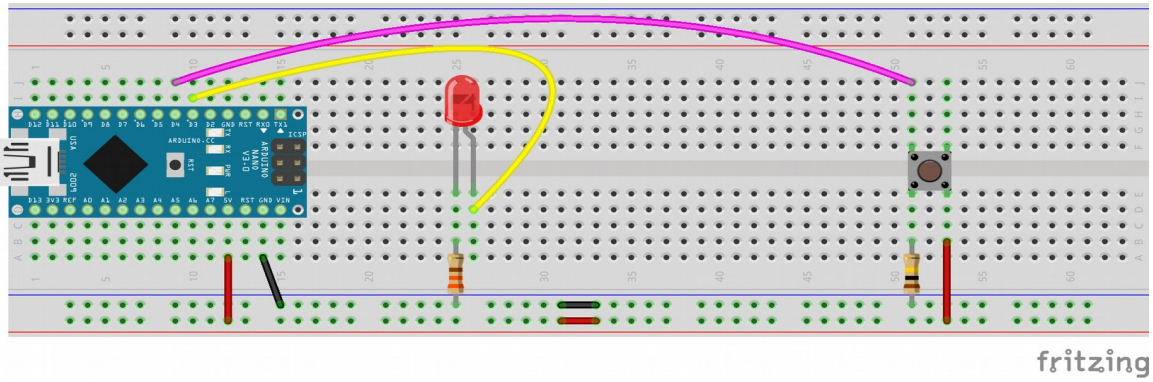
Deze sketch zal de LED steeds feller laten branden tot het maximum, daarna zal hij steeds meer dimmen totdat de LED uit is.

– probeer eens het dimmen en feller laten branden sneller of langzamer te laten verlopen.
– probeer eens de LED niet helemaal uit te laten gaan.
– bonuspunten voor de snelle snappers: probeer eens een tweede LED aan te sluiten die tegenovergesteld werkt.

Het lezen van een digitale “1” met een I/O pin (met pull-down)

Deze opdracht laat zien hoe je een digitale pin kunt lezen die naar 5 volt schakelt. We gebruiken hiervoor pin D4. We laten een LED branden als de drukknop is ingedrukt, deze is aangesloten op D3.

Bouw het schema op zoals onderstaand plaatje, de gebruikte weerstanden kun je door de kleurcode bepalen.



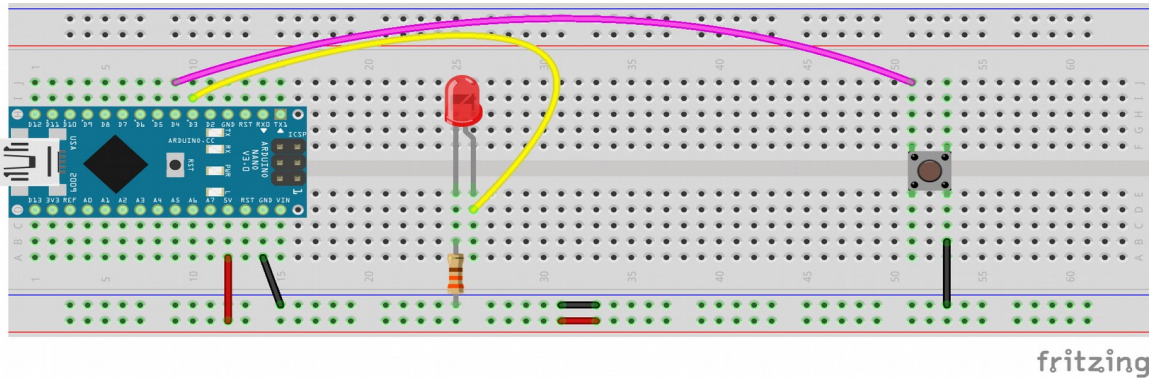
Deze sketch laat de LED branden zodra de drukknop is ingedrukt.

– probeer eens de LED te laten branden als de knop niet ingedrukt wordt.
– probeer eens de LED langer te laten branden dan de knop ingedrukt is.
– bonuspunten voor de snelle snappers : probeer eens een tweede LED aan te sluiten die tegenovergesteld werkt.

Het lezen van een digitale “0” met een I/O pin (met pull-up)

Deze opdracht laat zien hoe je een digitale pin kunt lezen die naar 0 volt (ground, GND) schakelt. We gebruiken hiervoor pin D4. We laten een LED branden als de drukknop is ingedrukt, deze is aangesloten op D3.

Bouw het schema op zoals onderstaand plaatje, de gebruikte weerstand kun je door de kleurcode bepalen.



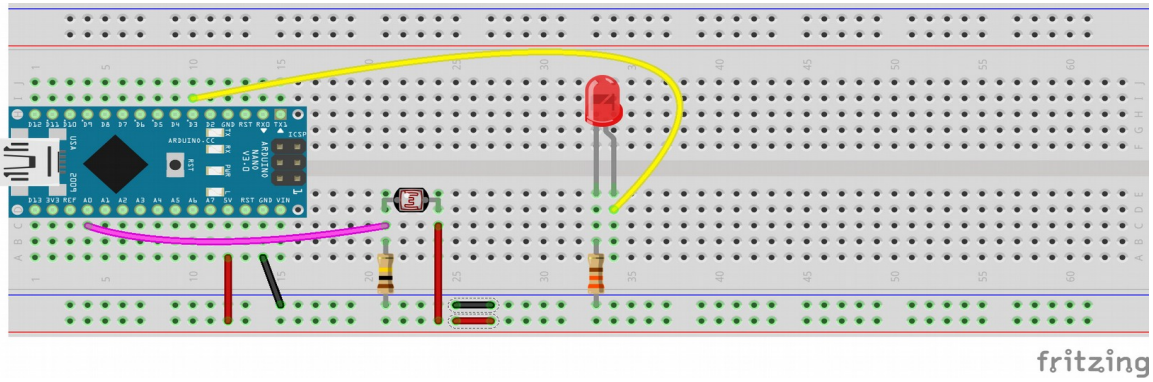
Deze sketch laat de LED branden zodra de drukknop is ingedrukt.

– probeer eens de LED te laten branden als de knop niet ingedrukt wordt.
– probeer eens de LED aan te laten gaan en niet meer uit totdat de arduino wordt gereset.
– bonuspunten voor de snelle snappers : probeer eens een de LED uit te laten gaan en pas weer aan te laten gaan nadat twee keer op de knop is gedrukt.

Het lezen van een analoge waarde met een I/O pin

Deze opdracht laat zien hoe je een analoge pin kunt lezen. We gebruiken hiervoor een LDR aangesloten op pin A0. We laten een LED branden als het donker wordt, de LED is aangesloten op D3.

Bouw het schema op zoals onderstaand plaatje, de gebruikte weerstand kun je door de kleurcode bepalen.



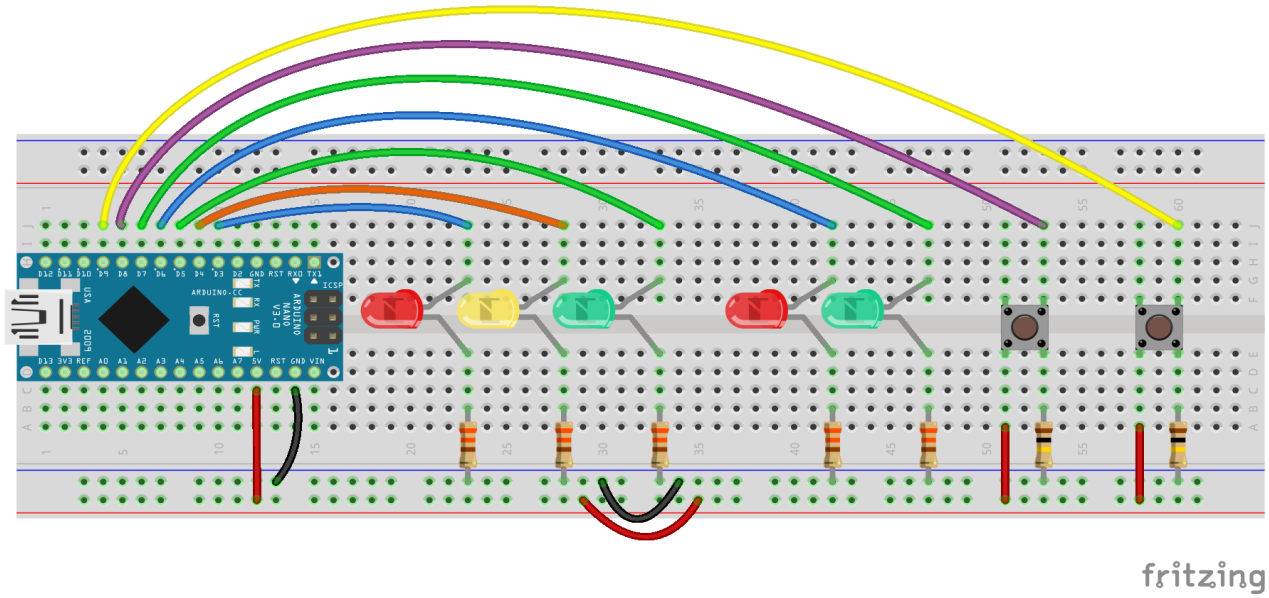
Deze sketch laat de LED branden zodra de LDR verduisterd wordt.

– probeer eens de LED normaal aan te laten zijn en uit te laten gaan als het donker wordt.
– probeer eens de LED bij meer of minder licht aan te laten gaan.
– bonuspunten voor de snelle snappers : probeer eens een vertraging in te bouwen en te zorgen dat de LED pas bij meer licht weer uit gaat dan dat hij aan ging (een zogenaamde schakelhysteresis).

Finale opdracht: het stoplicht.

Zorg dat de stoplichten voor de auto’s en de voetgangers samenwerken.
De drukknoppen kun je gebruiken als knopje voor de voetgangers en als de detectielus in de weg voor de auto’s

Succes!



Voor de snelle snappers: voeg een extra drukknop en twee blauwe LED’s toe en laat de politie door-rood-rijders bekeuren.