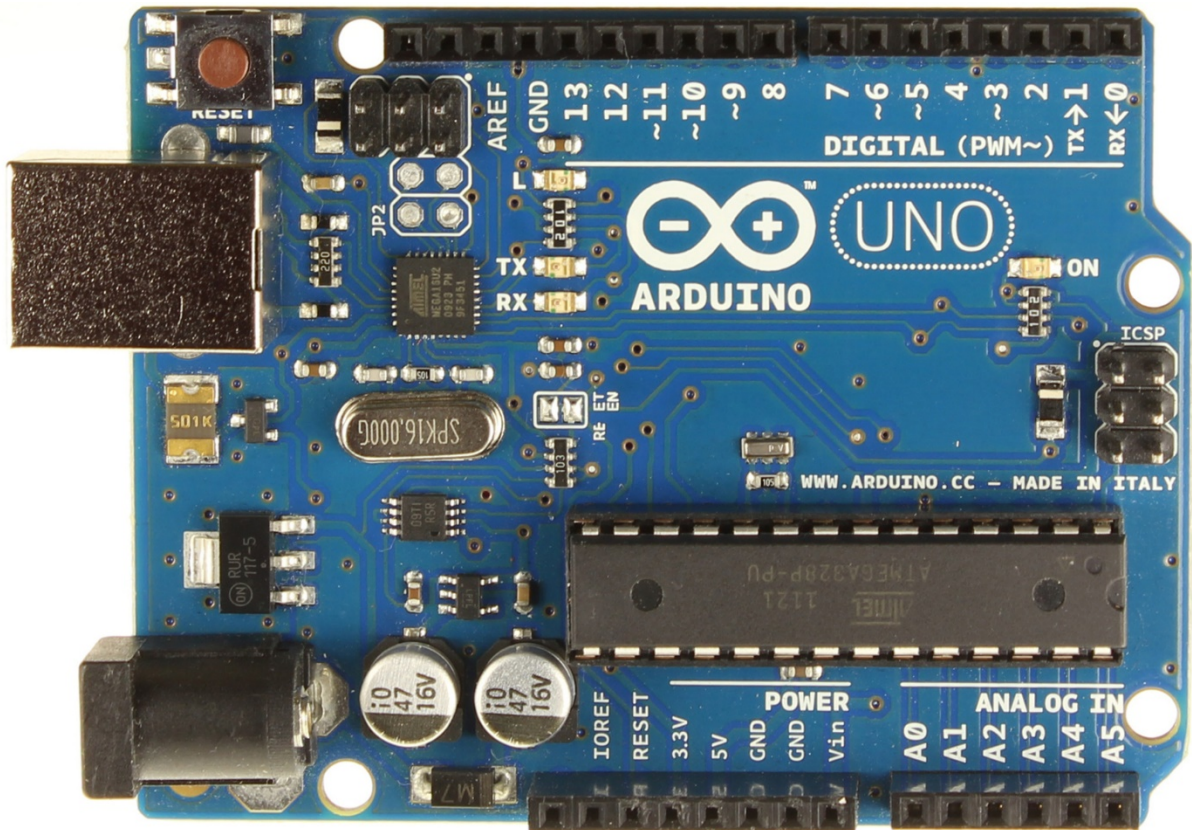


BOSWELL-BÈTA



Arduino

Vervolg cursus

C.F.J. Pols

Introductie

In het eerste deel hebben we stil gestaan bij een aantal basismogelijkheden van de Arduino. In dit deel gaan we kijken naar wat grotere projecten, diverse integrated circuits, inzet van de Arduino bij practica, lichtprojecten en nog veel meer. Deze module is verdeeld in verschillende stukken: projecten, sensoren en programmeren. Achterin vind je ook een aantal voorbeeld scripts behorende bij sensoren of opdrachten.

Als je er even over na denkt kan de Arduino alleen pinnen aan en uit zetten en spanning uitlezen. Het krachtige if-statement zorgt ervoor dat de Arduino nuttig ingezet kan worden bij de verschillende elektrische apparaten. In combinatie met de while-statement gebeurt dit precies op het moment dat jij dat wilt. Een for-loop en gebruik van arrays maakt je code weer korter en overzichtelijker. Door de schakelingen en codes langzaam en één voor één op te bouwen is het werken met de Arduino voor een ieder toegankelijk, klein beginnen en steeds verder uitbreiden. Je zult zien dat het redelijk eenvoudig is om daarna de verschillende onderdelen te combineren en zo het project te verwezenlijken dat jij in je hoofd hebt.

Belangrijk om vooraf te weten is dat de Arduino maar een beperkte stroomsterkte kan leveren. We moeten dus snel werken met transistoren waardoor het alsnog mogelijk is om apparaten die veel stroom vragen van elektriciteit te voorzien.

De module is geschreven voor zowel leerlingen als docenten die binnen de bèta vakken aan de slag willen met de Arduino. De inhoud is geschreven naar projecten waarin diverse sensoren gebruikt kunnen worden. Je kunt uiteraard ook een losse sensoren pakken en kijken hoe deze werken. Diverse voorbeeld codes kunnen achterin de module gevonden worden.

Freek Pols



Projecten

Inbrekersalarm

Wanneer je een dag weg van huis bent, wil je voor de zekerheid een inbrekersalarm hebben. Een alarm is echter duur en met de spullen die we hebben kunnen we dat ook zelf maken. Aan het alarm stellen we de volgende eisen:

- Het alarm moet af gaan na het detecteren van beweging of bij te veel geluid
- Het alarm moet pas na een halve minuut afgaan, zodat je bij thuiskomst het alarm kunt uitschakelen (met een drukknop) zonder dat het alarm afgaat
- Het alarm moet een pulserend lichtje zijn en een sirene
- Als er geen beweging of geluid meer gedetecteerd wordt, moet het alarm stoppen
- Een lampje moet blijven branden zodat je weet dat iemand een inbraakpoging heeft gedaan



Het programma van eisen is opgesteld. Maak een plan om stap voor stap het alarmsysteem te bouwen, te testen en vervolgens uit te breiden. Voer het plan uit.

Een leerlingteller

Is iedereen binnen? Je telt. Mist er één, telt nog een keer. En ja, je hebt er ergens een gemist. Is het niet handiger om aan je Arduino aan te geven hoeveel leerlingen je verwacht en dat de Arduino dan bijhoudt hoeveel leerlingen je lokaal binnen komen lopen?



Voor het bijhouden van het aantal leerlingen dat binnenkomt kun je een afstandssensor gebruiken, een laser gericht op een Idr (tip: plaats een rietje over de Idr zodat alleen het laserlicht opgevangen wordt, of een bewegingssensor. Let bij het programmeren op dat een code als 0101 twee leerlingen zijn, maar 01110 maar een leerling!

Parkeersensor

Een parkeersensor begint te piepen wanneer de afstand kleiner wordt dan 50 cm. Daarna zal de piep hoger worden of volgen de piepjes elkaar sneller op. Bij minder dan 7 cm moet de piep aanhoudend zijn.



Automatische lichten

In de winkel kun je nu lampen kopen die je aan en uit kunt zetten met een app op je telefoon. Die lampen zijn vaak wel duur. Gelukkig kunnen we dat ook prima zelf maken met de Arduino en een bluetooth module. Maak een lamp die je aan en uit zet met je telefoon. Breid de code uit zodat je ook de helderheid van de lamp kunt instellen met je telefoon. Start eenvoudig door te praten met je Arduino m.b.v. de seriële monitor.

Sensoren

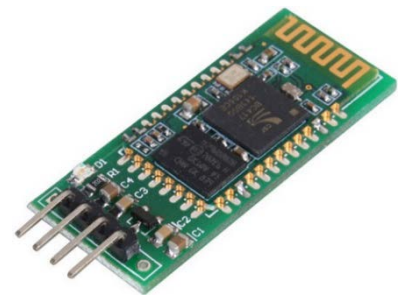
Praten met de Arduino

Via de seriële monitor kun je twee kanten op praten met je Arduino. Je kunt informatie opvragen maar ook informatie geven. Getallen, letters, of woorden invoeren is vrij eenvoudig. In het voorbeeld hiernaast wordt een getal gevraagd en gebruikt. Denk wel altijd aan het type variabele wat je wilt verkrijgen! In de voorbeeld code gebruiken we een integer.

```
if(Serial.available()) {  
  brightness = Serial.parseInt();  
  analogWrite(ledPin, brightness);  
}
```

Bluetooth module

Met de bluetooth module wordt het mogelijk om je Arduino aan te sturen met behulp van je telefoon. Je stuurt dan een stukje tekst (String) naar de Arduino. Er is niet veel code nodig om dit te verwezenlijken. Het lijkt op praten met de Arduino via de seriële monitor. Alleen heb je geen kabel meer nodig. Zo kun je een afstandbestuurbare auto maken.



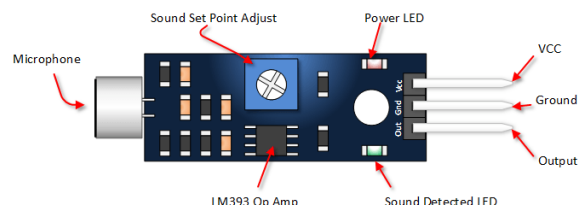
De basiscode voor communicatie met de bluetooth module staat hiernaast. Anders dan in het voorbeeld hierboven lezen we nu een karakter (ander datatype) uit. Er zijn wel mogelijkheden om waarden om te zetten, bijvoorbeeld via `blueToothVal.toInt()`. Deze code maakt van een karakter of string een integer.

```
if(Serial.available()){  
  blueToothVal=Serial.read();  
}
```

De aansluiting van pinnen staat op de module (let op: rx moet naar tx en tx naar rx!). Deze twee pinnen moeten pas verbonden worden met de Arduino als het programma is geüpload. Pin 0 en 1 van de Arduino worden immers ook gebruikt voor communicatie met de computer...

Geluidsensor

Een geluidsensor geeft een hoog signaal wanneer het geluidsniveau boven een bepaalde waarde komt. Deze waarde is in te stellen met de potmeter op de sensor. Je kunt helaas geen geluid opnemen via deze sensor.



Het uitlezen van de sensor is vrijwel identiek aan het uitlezen van een button. Je leest (digitaal) de output uit en krijgt een 0 of een 1 terug.

Timer

In veel projecten wordt gebruik gemaakt van een timer. Deze kun je maken m.b.v. de code `millis()`; deze code geeft aan hoeveel milliseconde zijn verstreken sinds de Arduino is aangezet. Maak van de variabelen die de tijd bijhouden wel het type long!

Een voorbeeldcode is hiernaast weergegeven. Deze toont elke seconde in de seriële monitor dat er een seconde voorbij is. Een volledige code vind je achterin de module.



```
Timer1 = millis();  
while(millis()-Timer1<1000){  
  Serial.print(i);  
  Serial.println(" s");  
  i = i + 1;  
}
```

Sensoren

Buzzer

In een piëzo buzzer zit een materiaal dat van vorm verandert als er een spanning over heen staat. Je kunt hem ook andersom gebruiken: verbuigt het materiaal dan wordt er een kleine spanning geleverd. Daarmee is dit materiaal ook prima te gebruiken om bijvoorbeeld regendruppels te detecteren. Je kunt er ook een goedkope speaker mee maken, zoals bij deze buzzer.



De Arduino kan één pin aansturen om een toon te maken, het is niet mogelijk meerdere tonen tegelijk te geven. Voor het maken van een toon kun je de tone functie gebruiken:

```
tone(pin,frequentie,duration);
```

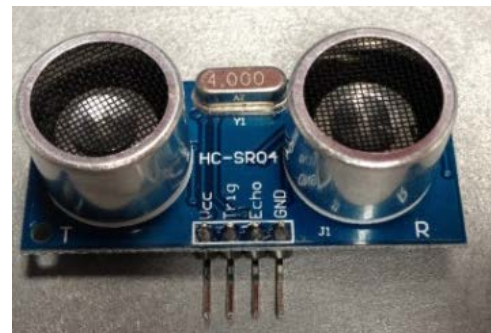
In deze functie is de derde optie, duration, optioneel. Met `noTone(pin)`; stop je het geluid weer.

Zoals gezegd kan de buzzer ook andersom gebruikt worden. Je leest hem dan uit via een analoge pin. Echter zullen de gemeten waarden zeer laag zijn.

Afstandssensor

De ultrasone afstandssensor werkt zoals een vleermuis afstanden bepaalt: Een puls wordt uitgezonden en de tijd tussen uitzenden van de puls en ontvangen van de reflectie wordt gemeten. Deze tijd wordt teruggegeven in microsecondes en kun je vervolgens omrekenen naar een afstand. In de Arduino voorbeelden zit het programma Ping (onder sensors). De afstandssensor die daar gebruikt wordt, werkt met drie pinnen waarbij de pinMode in het programma wordt omgedraaid. Dit programma kun je als basis gebruiken om de sensor met vier pinnen aan te sturen en uit te lezen. Wil je het liever meteen gebruiken? Dan staat de code hiernaast.

Let op! Het is pulsein met de i als hoofdletter.



```
digitalWrite(pingpin, LOW);  
delayMicroseconds(2);  
digitalWrite(pingpin, HIGH);  
delayMicroseconds(5);  
digitalWrite(pingpin, LOW);  
duration = pulseIn(echopin, HIGH);  
distance_cm = duration/20000*343;  
Serial.print(distance_cm);  
Serial.println(" cm");
```

Bewegingssensor

Als bewegingssensor kan een Passive InfraRed (PIR) gebruikt worden. Een PIR meet de stralingsenergie in het infraroodspectrum. Bij verandering van intensiteit, dus detectie van beweging, wordt het signaal hoog.

Het hoedje van de PIR kan opgetild worden zodat je weet welke functie elk van de pinnen heeft. Met de twee stelschroefjes kun je de gevoeligheid en de wachttijd tussen twee metingen instellen.



De PIR kijkt alle kanten op. Focussen op een bepaald gebied kan door een deel van de sensor af te dekken.

Projecten

Simon Says

Het spelletje Simon Says lijkt op het spelletje ik ga op reis en ik neem mee... Bij Simon Says gaan vier verschillende ledjes in een bepaalde volgorde knipperen (1234). Dezelfde volgorde moet ingedrukt worden met behulp van vier drukknoppen. Als de juiste volgorde is ingedrukt, wordt de serie verder uitgebreid (12343). Het spel gaat net zo lang door totdat de speler de foute volgorde indrukt. Het is natuurlijk het leukst wanneer de volgorde nooit hetzelfde is en dat de Arduino de volgorde voor je bepaalt.

Maak het spel!



Robotica

Veel scholen willen aan de slag met Robotica. De Arduino geeft de mogelijkheid om dit relatief goedkoop en snel te doen. Er zijn kant en klare auto's te koop voor ongeveer €100,- maar als je de onderdelen los besteld, dan heb je de auto zelf in elkaar gezet voor ongeveer €30,-.

We beginnen met het laten rijden van de Arduino op verschillende snelheden. Dit kun je later zelf uitbreiden door een afstandssensor te gebruiken die er voor zorgt dat de auto niet tegen een muur botst en een bluetoothmodule waarmee je de auto kunt aansturen vanaf je telefoon. Lees ook het stukje over functies programmeren! Dat is bij robotica erg handig.



Een thermometerdisplay

Systemen die de temperatuur, luchtvochtigheid en druk kunnen geven kunnen gekocht worden in de winkel. Je kunt ze ook zelf maken. En met een kleine uitbreiding van de spullen die je al hebt, kun je dit maken. Wanneer je er een mooi kastje om heen bouwt heb je je eigen weerstation gemaakt. Begin in deze opdracht met de temperatuur te tonen op een lcd display. Breid deze uit met een knop waarmee je bijvoorbeeld de hoeveelheid licht meet en deze toont op het display.



Een digitale meetlint

Met een afstandssensor en een lcd display kun je een eenvoudige digitale meetlint maken. Altijd handig als je snel een afstand moet meten die ligt tussen de 2 cm en 3,00 m, of bij het maken van een practicum waarbij de afstand bepaald moet worden (bijvoorbeeld als functie van de tijd). Bij dit laatste is het ook handig om je data op te slaan in een array en weg te schrijven naar een SD-module.

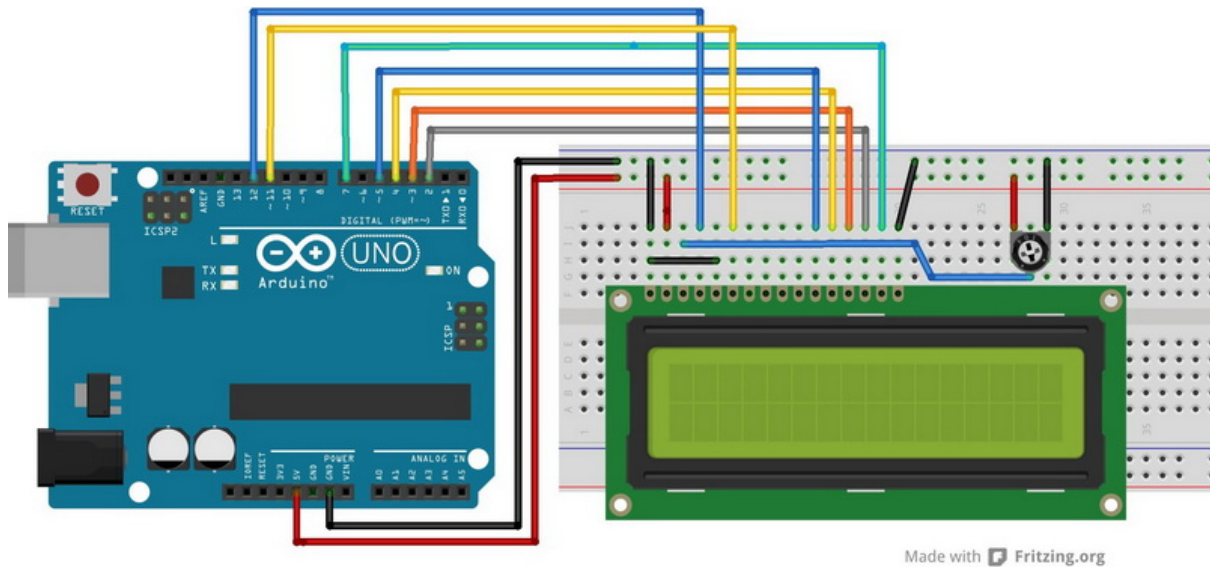


Sensoren

LCD module

De LCD module bestaat in meerdere varianten. Je hebt er een die gebruik maakt van een data en clock pin, deze heeft maar 2 pinnen nodig en is dus sneller in elkaar te zetten. Deze adapter kan er later ook opgezet worden (zoek bij bestellen op lcd1602 adapter of lcd i2c).

Voor de display die je nu hebt, moet je de onderstaande opstelling maken. Bij de voorbeeld programma's staat Hello World van Arduino, die kan gebruikt worden om de display aan te sturen. Let op! Pin 7 moet nog wel hoog gemaakt worden (deze mag ook direct naar de +5V). Met de potmeter regel je het contrast. Zie je niets, draai dan de potmeter totdat je zwarte letters ziet verschijnen.

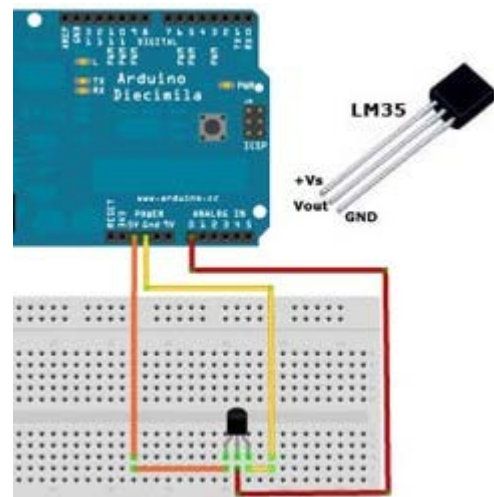


LM35

De LM35 is een van de vele temperatuursensoren die je goedkoop kunt krijgen. Volgens de datasheet van TI¹ heeft deze een bereik tussen -55°C en +150°C met een resolutie van 10 mV/°C. Bij kamertemperatuur is de nauwkeurigheid 0,25°C. Voor temperaturen onder de 2°C moet er extra weerstand in de schakeling opgenomen worden, zie wederom de datasheet voor verder informatie.

Voor het berekenen van de temperatuur kun je de volgende formule gebruiken: $\text{temp} = (5.0 * \text{analogRead}(\text{tempPin}) * 100.0) / 1024$;

De Arduino vergelijkt nu de spanning met een interne referentie van 5,0 V. Echter is de uitgelezen waarde altijd kleiner dan 1 V en is de resolutie niet zo hoog als mogelijk. Je kunt dit wijzigen door de interne referentie (1.1V) aan te zetten: `analogReference(INTERNAL)`; Het script voor de LM35 met hoge resolutie is in het boekje opgenomen, maar probeer eerst zelf de eenvoudige versie van de temperatuur uit te lezen!



¹ <http://www.ti.com/lit/ds/symlink/lm35.pdf>

Programmeren

Random

Voor bijvoorbeeld het maken van spelletjes met Arduino wil je random getallen gebruiken. De Arduino heeft ook de mogelijkheid voor een random getallen generator. Deze functie print willekeurig getallen. De functie Random is echter niet geheel random. Elke keer dat je de Arduino start zal dezelfde getallen reeks verschijnen. Om dit te voorkomen kun je gebruik maken van ruis, dat is immers altijd random. Door pin A0 uit te lezen (als er niets is aangesloten) kun je dit als input gebruiken om een reeks getallen random te krijgen. De code hiernaast kan daarvoor gebruikt worden.

```
randomSeed(analogRead(A0));
random(0,4);
```

Programmeren met Arrays

Een serie getallen kun je in een rij (array) zetten. Daarmee kan een stuk code overzichtelijker worden. Een voorbeeld is:

```
int ledgroen[] = {6,7,8,9};
```

Deze code kun je gebruiken om 4 ledjes aangesloten op pinnen 6,7,8 en 9 aan te sturen. De code: `digitalWrite(ledgroen[0],HIGH);` zal de led op pin 6 aan zetten.

```
#define arraysize 100
int rij[arraysize];

void setup() {
  for(int i = 0; i<arraysize; i++){
    rij[i] = random(0,4);
    i = i + 1;
  }
}
```

Je kunt een array ook langer maken in een loop. Wel moet de Arduino de totale lengte van de rij van te voren weten. Een mogelijke start voor je code van Simon says staat hiernaast.

Functies programmeren

Voor een code die vaak herhaald wordt kun je een functie aanmaken. Functies zijn kleine stukjes code die aangeroepen kunnen worden. In onderstaande code wordt gebruik gemaakt van een functie die een led aan en uit zet. Nevenstaande code kan gebruikt worden voor een verkeerslicht.

De functie heet aanuit en vraagt twee dingen, een pinnummer en een wachttijd. Deze worden in het script gebruikt. Bij het aanroepen van de functie kun je die twee waarden geven.

Bij robotica is het handig om te werken met functies, bijvoorbeeld voor vooruit/achteruit/bocht om.

```
void loop(){
  aanuit(13,100);
  aanuit(12,50);
  aanuit(11,100);
}

void aanuit(int pin, int tijd){
  digitalWrite(pin,HIGH);
  delay(tijd);
  digitalWrite(pin,LOW);
}
```

Switch case

Je kunt met Arduino een soort keuzemenu maken waar je doorheen kunt lopen. Dit wordt een switch case genoemd. Het idee is dat je een genummerd keuze menu maakt waarbij een variabele (hier var) aangeeft waar je in het keuze menu bent. Een eenvoudig voorbeeld van een case switch:

Je kunt eenvoudig switchen van menu door bijvoorbeeld een counter te maken. `var = counter%2` geeft nu de waarde 0 bij een even getal en 1 bij een oneven getal.

```
switch (var) {
  case 0:{
    digitalWrite(led,HIGH);
    break;
  }
  case 1:{
    digitalWrite(led,LOW);
    break;
  }
}
```

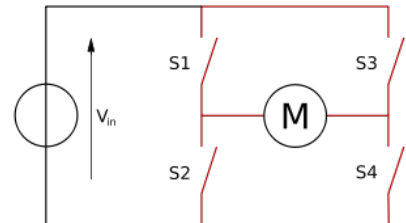

Sensoren

Hbrug

Een H-brug maakt het mogelijk om met DC spanning een motor toch twee kanten op te laten draaien. Om dit te doen wordt er gebruik gemaakt van transistoren. Je bestuurt met de Arduino de transistoren S1 t/m S4. Let op! S1 en S2 mogen nooit tegelijk omgezet worden. Dan ontstaat een kortsluitstroom. Hetzelfde geldt voor S3 en S4 (in onderstaande code MCL1 en MCL2).



Met de L298N is het mogelijk twee motoren aan te sturen. Je kunt een hogere (externe) bronspanning leveren. De 5V is nodig voor het aansturen van het interne circuit. Het PWM-sigitaal (voor het regelen van de snelheid) kan gedaan worden met de ENA. IN1 hoog zetten laat de motor de ene kant op draaien, IN2 de andere kant op. De drie pinnen er naast (IN3, IN4, ENB) doen hetzelfde maar dan voor de andere motor.



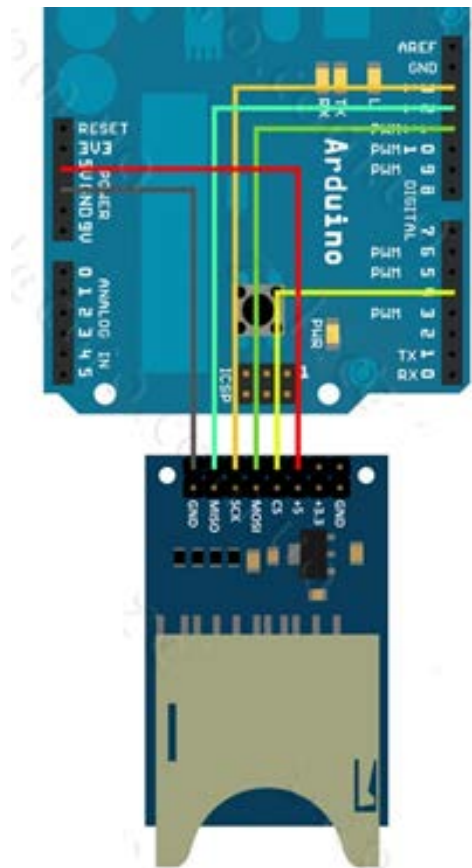
Een voorbeeld code voor de motor staat aan het einde van deze handleiding.

Dataloggen met een SD-kaart

Wanneer je de Arduino uit zet, is je verzamelde data weg. Wil je gebruik maken van je data dan kun je dit via een omweg wegschrijven naar bijvoorbeeld excel (voorbeelden zijn redelijk eenvoudig op internet te vinden). Een tweede mogelijkheid is om de data weg te schrijven naar een SD-kaart m.b.v. een SD module. Voordeel hiervan is dat de Arduino stand-alone kan draaien en dus op afgelegen plekken (dak van de school) kan liggen.

In de voorbeelden van Arduino staat de code voor datalogger (voorbeelden/SD/datalogger). In dat voorbeeld worden drie analoge pinnen uitgelezen. Er wordt een String aangemaakt en de uitgelezen waarden worden toegevoegd aan deze string. Per meting wordt die string weggeschreven naar de SD kaart.

In codes achterin staat een voorbeeld van een temperatuurmeting die elke minuut plaats vindt.



Projecten

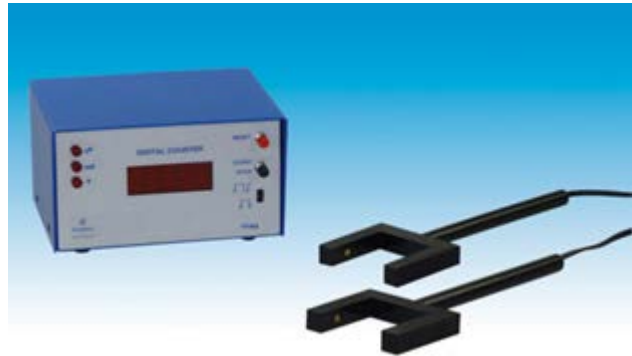
Piano

De combinatie van de buzzer met drukknopjes levert een instrument op. Elke drukknop komt overeen met een bepaalde frequentie. Met bijvoorbeeld de if functie kun je elke drukknop koppelen aan een bepaalde frequentie. Probeer maar eens een piano te maken!

Toon	Frequentie (Hz)
C	261,6
D	293,7
E	329,6
F	349,2
G	392
A	440
B	493,9

Een natuurkunde practicum

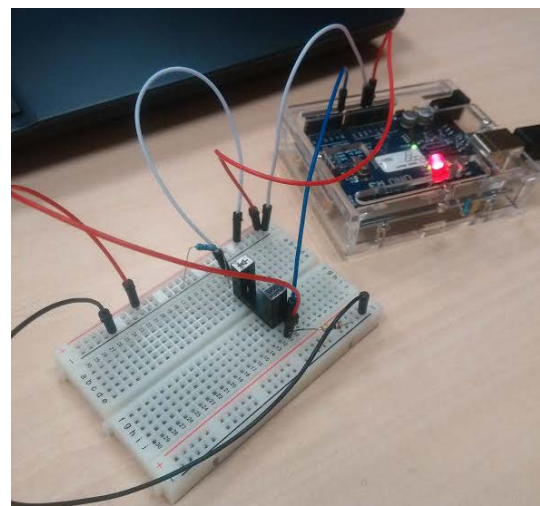
Een bekend natuurkundige practicum is de luchtrailsproef waarbij je gebruik maakt van twee lichtpoortjes en een timer. Het kastje inclusief twee lichtpoortjes kost ongeveer €190,-. Dezelfde soort opstelling kun je maken met behulp van de Arduino waardoor je de prijs reduceert met een factor 10. Je kunt zelfs meerdere fotopoortjes gebruiken waardoor de gehele beweging geregistreerd staat.



Op het moment dat de eerste lichtstraal wordt onderbroken start de timer. Op het moment dat de tweede lichtstraal wordt onderbroken stopt de timer. Het verschil in deze twee tijden (Δt) moet getoond worden, bijvoorbeeld met behulp van de seriële monitor, de lcd of m.b.v. een SD kaart. Met deze tijd kun je de gemiddelde snelheid uitrekenen (als je ook de afstand hebt gemeten, bijv. met de ultrasone geluidsensor).

In deze opdracht sla je de tijd op, op een sd-kaart en bouw je dus eigenlijk een datalogger. In de voorbeeldscripts van Arduino staat een apart stukje over SD. Maak voor het gebruik van deze proef gebruik van het script Datalogger.

Bouw de practicumopstelling en schrijf het script zodat de tijd opgeslagen wordt op de SD kaart.



In het lichtpoortje zit een fotodiode of fototransistor, zie het symbool hiernaast. Deze laat stroom door zodra er licht op de diode valt. Er zijn twee varianten op de markt: als module (iets duurder en drie pinnen) en los (H2010). Bij de module verbind je de 5V, Ground en de digitale uitgang met een digitale pin (waarde kan 0 en 1 zijn). Bij de losse H2010 verbind je de diode met een 220 Ω weerstand. De fototransistor verbind je in serie met een 1k Ω weerstand, zie foto. De deelspanning tussen de transistor en de weerstand lees je uit.

Sensoren

Hall sensor

Een Hall sensor is een sensor voor het meten van het magnetische veld. De sensor is ideaal om bijvoorbeeld snelheden te meten zoals dat gedaan wordt met een snelheidsmeter op je fiets. De sensor is gebaseerd op het Hall-effect: Elektronen gaan naar een zijde (van een draad) wanneer ze bewegen door een magnetisch veld. Hierdoor ontstaat er een spanningsverschil tussen de verschillende zijdes van de draad. Er zijn verschillende soorten Hall sensoren, de een geeft een hoog/laag signaal als er een bepaalde waarde overschreden wordt (soort treshold). De andere sensor meet daadwerkelijk de sterkte van het magnetische veld. In het pakket zit de eenvoudige versie.

Voor het werken met de Hall-sensor, plaats de sensor met de tekst zijde naar je toe in het breadboard. Plaats over de uiterste twee poten een 10 kΩ weerstand. Verbind de rechterzijde van de sensor met pin 2, de middelste pin met GND en de linkerzijde met 5V. Voor het programma kun je starten met analogReadSerial (en dan analogRead veranderen in digitalRead en A0 naar pin 2 wijzigen).

Een nadeel van deze wijze van uitlezen is dat je altijd bezig bent met kijken naar pin 2. Je kunt dit veranderen door gebruik maken van een interrupt (pin 2 en 3 zijn daar specifiek voor op de UNO). Zodra pin 2 iets meet, wordt het programma onderbroken om een functie uit te voeren. Zie de voorbeeld sketches onderaan voor een voorbeeld.

Let op! De magneet moet aan de bedrukte zijde langs gaan voor een detectie.

Een leuk project met de Hall-sensor is gemaakt door Rolf Hut:

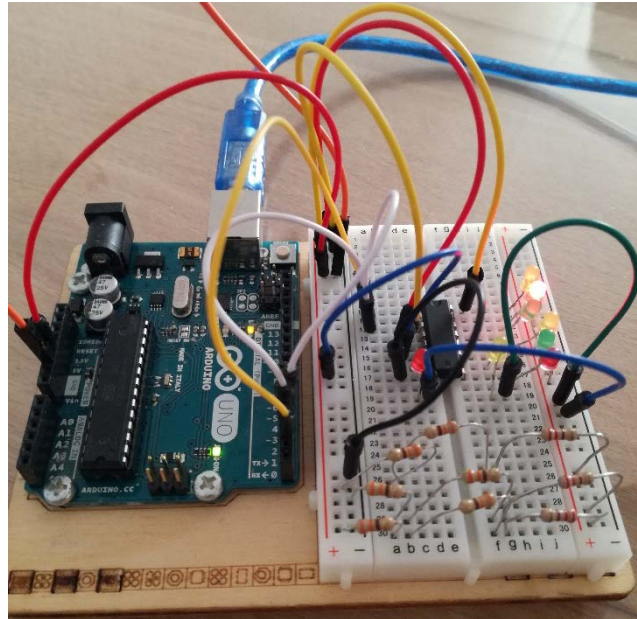
<http://www.instructables.com/id/Longboard-With-NeoPixel-LEDStrip-Reacting-to-Speed/>



IC's

Multiplexers

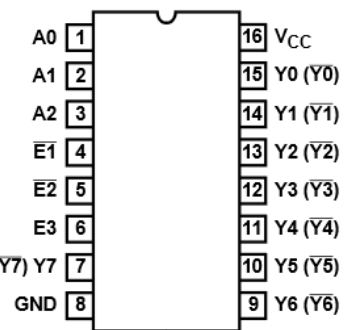
Bij grotere projecten heb je al snel ruimte te kort: je hebt maar een beperkt aantal pinnen. Een van de mogelijkheden die je hebt is dan een Arduino Mega te kopen, deze heeft veel meer pinnen uit. Echter moet je dan wel weer een Arduino kopen en heb je wellicht ruimtegebrek in je project (stel dat je t allemaal wilt wegwerken in een kastje). In principe zou het toch mogelijk zijn om met drie pinnen 8 ledjes aan te sturen: $2^3 = 8$. Dit is inderdaad waar en kan ook met een multiplexer. Een 3-8 line decoder (ja, je hebt ook 4-16 line decoders) is in staat om met drie pinnen 8 outputs te bedienen. In dit project gebruiken we de 74HC238E.



Verbind de VCC en E3 met de 5V van de Arduino. Verbind de GND, E1 en E2 met de GND van de Arduino. Door nu spanning op pin A0, A1, en/of A2 van het IC te zetten, zal een van de Y pinnen een stroom door laten. Dit volgens de waarheidstabel hieronder.

Laat je de ledjes heel snel achter elkaar aan en uit gaan, zonder delay er tussen, dan lijkt het er op dat de ledjes allemaal aan staan.

TOP VIEW



TRUTH TABLE 'HC238, 'HCT238

INPUTS						OUTPUT			
ENABLE			ADDRESS						
E3	E2	E1	A2	A1	A0	Y0	Y1	Y2	Y3
X	X	H	X	X	X	L	L	L	L
L	X	X	X	X	X	L	L	L	L
X	H	X	X	X	X	L	L	L	L
H	L	L	L	L	L	H	L	L	L
H	L	L	L	L	H	L	H	L	L
H	L	L	L	H	L	L	L	H	L
H	L	L	L	H	H	L	L	L	H
H	L	L	H	L	L	L	L	L	L
H	L	L	H	L	H	L	L	L	L
H	L	L	H	H	L	L	L	L	L
H	L	L	H	H	H	L	L	L	L

H = High Voltage Level, L = Low Voltage Level, X = Don't Care

IC's

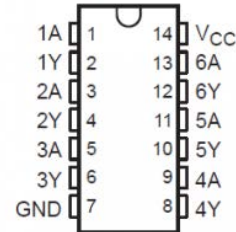
Inverter

Een inverter is wel bekend van het systeembord. De inverter draait het signaal om, een 1 wordt een 0 en een 0 wordt een 1. Dat is handig voor sommige projecten. Zo zijn er RGB-leds met een gezamenlijke kathode en met een gezamenlijke anode. Heb je net de verkeerde... dan kun je de werking als nog eenvoudig omdraaien m.b.v. een inverter.

Je voedt pin 14 met 5V (of een andere voeding, zie specs). Sluit de ground aan. 1A is een ingang, en 1Y de uitgang. Het signaal wordt omgedraaid: een 1 wordt een 0 of een 0 wordt een 1. Het wordt zelfs nog leuker! Ook met analogWrite kan het effect omgedraaid worden! Een analogWrite(240); brandt dus maar zwakjes. In combinatie met een multiplexer kan dit een krachtig IC zijn.

Wide Operating Voltage Range of 2 V to 6 V
Outputs Can Drive Up To 10 LSTTL Loads
Low Power Consumption, 20- μ A Max I_{CC}

SN54HC14... J OR W PACKAGE
SN74HC14... D, DB, N, NS, OR PW PACKAGE
(TOP VIEW)

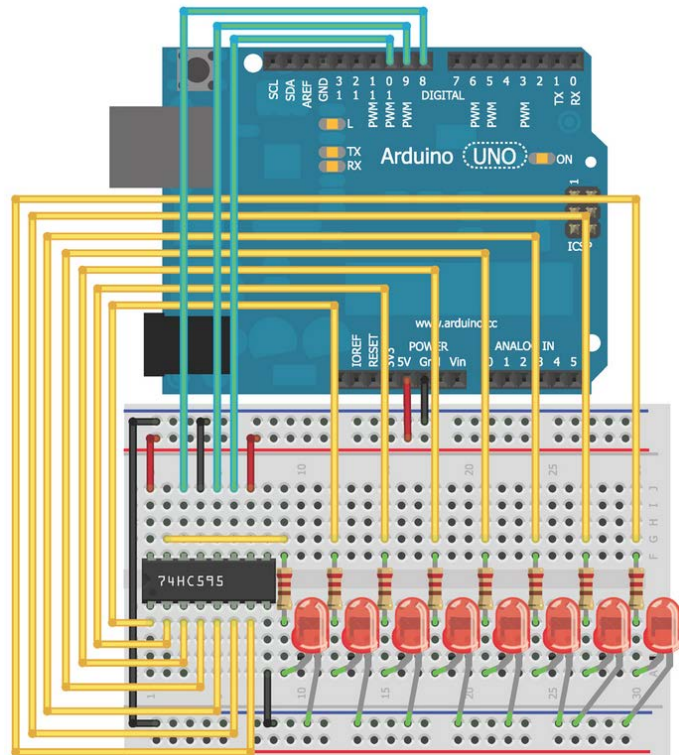


Shift register

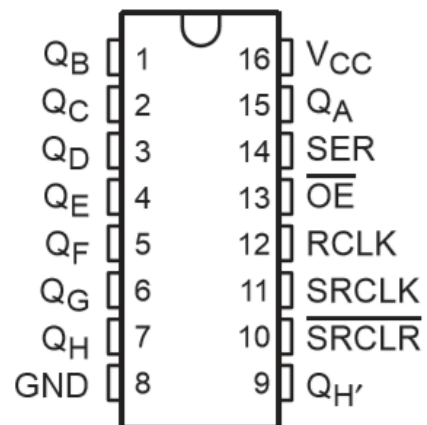
We hebben de 3-8 line decoder gezien. In plaats van een 3-8 line decoder kun je ook een shift register gebruiken. Ook een shift register drie ingangen en acht outputs. Alleen stuur je nu een (binair) code naar de shift register en wordt deze uitgevoerd. Je kunt shift registers door verbinden met elkaar en zo 8, 16, 24 (etc.) outputs hebben.

De 74HC595 shift register die wij gebruiken heeft drie belangrijke pinnen. Een clock (SRCLK), data (SER) en latch (RCLK).

Op het moment dat de latch laag is, kun je code sturen naar de shift register. Dit wordt gedaan met de shiftOut code. Deze stuurt een 0 of een 1 naar de eerste pin van de shift register, waarna de clock naar de volgende output pin gaat en er daar weer een 0 of een 1 wordt toegewezen. Dit kun je ook snel doen door een getal te sturen, zoals in het voorbeeld hiernaast. De shift register maakt er dan een binaire code van (je kunt ook zelf een binaire code sturen bijvoorbeeld: B10101010). Op het moment dat de latch hoog wordt, zullen de pinnen die een 1 hebben gekregen hoog worden.



74HC595 Shift Register



Programmeren

Simultaan werken

De Arduino kan geen twee dingen tegelijkertijd (er is wel een truc genaamd interrupts). Om toch twee dingen tegelijk uit te voeren kun je werken met timers. Daarbij is het belangrijk om geen delay te gebruiken in je script. Deze stopt namelijk het hele proces. In de voorbeeldscripts van Arduino vind je het programma Blinkwithoutdelay waarin het eenvoudige programma Blink is herschreven.

Bovenaan wordt een tijdsinterval gekozen. Als er een seconde voorbij is moet de LED wisselen van status. Dit wordt gedaan door de tijd op te vragen (millis()) en te vergelijken met de tijd van de vorige keer dat de led gewisseld is van status. Is de voorbij tijd nog niet groter dan het gekozen tijdsinterval wordt de rest van de loop afgedraaid (deze is in dit voorbeeld leeg).

Op deze wijze kun je de status van de LED wijzigen en de gehele tijd letten op bijvoorbeeld een drukknop en of deze wordt ingedrukt.

Het programma kan ook korter geschreven worden.

```
const int ledPin = 13;
int ledState = LOW;
unsigned long previousMillis = 0;
const long interval = 1000;

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

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }
    digitalWrite(ledPin, ledState);
  }
}
```

```
const int ledPin = 13;
int ledState = LOW;
const long interval = 1000;

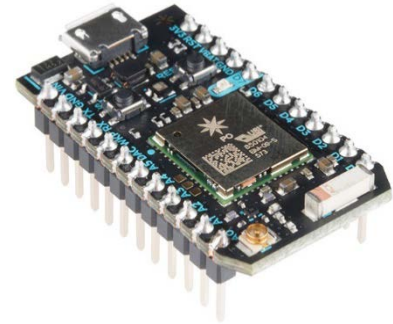
void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  if (millis() - previousMillis >= interval) {
    previousMillis = millis();
    digitalWrite(ledPin, !digitalRead(ledPin));
  }
}
```

Wat is er nog meer?

Photon

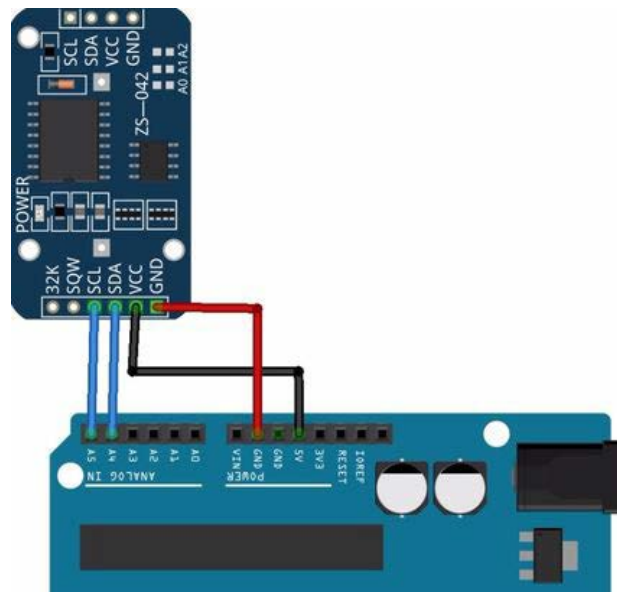
Naast de Arduino Uno zijn er meerdere microcontrollers die ongeveer hetzelfde kunnen maar allemaal hun eigen eigenschappen hebben. Zo is er de Arduino 101 met drie ingebouwde versnellingsmeter en bluetoothmodule. De nieuwste Arduino heeft ook standaard WiFi aan boord. Dit is wellicht afgekeken van de Photon van Particle. Deze heeft



een WiFi shield waarmee je dus niet meer via een usb een connectie legt tussen de computer en de Photon maar via het internet. Voordeel is dat je vanuit heel de wereld kunt inloggen op je Photon en dat je gebruik kunt maken van IFTTT (If this then that). Je stuurt dan een berichtje (bijvoorbeeld via twitter) naar je Photon dat je naar huis komt en dat de temperatuur dan een aangename 20°C moet zijn. Je Photon vervangt dan je thermostaat. De slimme thermostaat kan dit ook maar kost wel €200,- terwijl een Photon nog geen €25,- kost. Je kunt alsnog de Photon uitrusten met een Bluetoothmodule en vanaf je bank de thermostaat bedienen. In combinatie met de RTC (zie hieronder) kun je de gehele thermostaat programmeren en bedienen, veranderen wanneer je op vakantie gaat etc.

RTC

De Arduino kan een heleboel, maar erg nauwkeurig de tijd bijhouden (denk aan dagen lang), is toch lastig. De Arduino houdt alleen de tijd bij sinds deze is aangezet. Om netjes de tijd bij te houden heb je een Real-Time-Clock (RTC) nodig, bijvoorbeeld de ds3231.



Bij het gebruik van onbekende sensoren of elektronica, is het vaak handig om te zoeken naar een juiste library. Dit maakt het eenvoudig om de sensoren aan te sturen.

Deze libraries kun je apart installeren. Voor

de RTC gebruik ik zelf de library DS3231 verkrijgbaar via:

<http://www.rinkydinkelectronics.com/library.php?id=73> In een goede library vind je ook examples en een ReadMe waarin uitleg staat over de functies die aangemaakt zijn.

Bij een RTC moet je eenmaal de tijd instellen en vervolgens onthoudt de RTC de tijd (een zelfde soort systeem zit in je computer, waarbij de tijd goed staat tot je de batterij moet vervangen). Je kunt nu bijvoorbeeld een wekker maken of een wake-up light omdat je de precieze tijd weet! In de voorbeelden aan het eind van de module vind je een script waarbij de tijd elke minuut wordt geprint en er een alarm afgaat om 20:45h.

Wat is er nog meer?

Relais

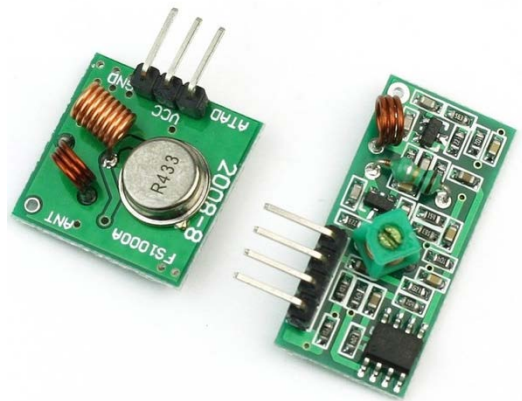
Voor het aansturen van elektronica in huis (domotica) kun je een relais gebruiken. Het sturen van een hoog signaal naar de Relais zorgt ervoor dat de schakelaar op het andere circuit gesloten wordt. De relais zijn te verkrijgen voor onder de euro per stuk. Advies hier is om de relais met ledjes te nemen. Deze gaan branden op het moment dat de schakelaar gesloten is. Voorzichtigheid blijft altijd geboden!



Afstand communicatie

Er zijn diverse mogelijkheden voor communiceren met Arduino's of communicatie tussen Arduino's. Voor wat grotere afstanden is er een 433 MHz zend en ontvangstmodule.

Volgens de specificaties moet deze een afstand van 200 m kunnen overbruggen.



4 digit 7 segment display

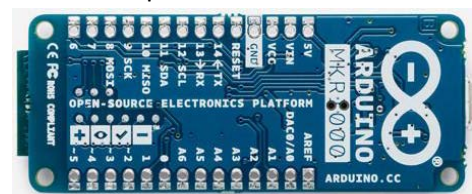
Het LED display waarop je vier digits kunt schrijven is er in twee varianten, eentje met I2C protocol die maar vier pinnen nodig heeft en de 'gewone' versie waarbij je 12 pinnen nodig hebt. Als module ben ik alleen een hele kleine versie tegengekomen. Van de 12 pins variant heb ik er een die 7x2,5cm groot is en dus prima gebruikt kan worden in combinatie met een RTC om een eigen digitale klok te maken! Het display kan ook gebruikt worden voor een escaperoombox:

<http://www.instructables.com/id/Escaperoom/> Op de volgende instructable vind je een prima beschrijving hoe de 12 pins variant aan te sluiten en welke library daarvoor gebruikt kan worden: <http://www.instructables.com/id/Using-a-4-digit-7-segment-display-with-arduino/>



Arduino 101 en Arduino MKR1000

De Arduino 101 is speciaal gebouwd voor onderwijs doeleinden. Deze Arduino heeft standaard drie versnellingsschakelaars en een bluetoothmodule aan board. Dat scheelt weer pinnen! De Arduino MKR1000 heeft standaard een WiFi shield en daarmee is het dus, net als bij de photon, eenvoudig mogelijk om verbinding te maken met het internet en je IOT projecten aan te pakken.



Wat is er nog meer?

Playstation

Leerlingen die thuis bijvoorbeeld met een playstation spelen, moeten de knop hiernaast herkennen. De knop heeft drie mogelijkheden: links/rechts, boven/beneden, en indrukken. Daarom zijn er ook 5 pinnen (incl gnd en 5V). Het uitlezen is eenvoudig als je weet wat er in zit: een drukknop en twee keer een potentiometer. In combinatie met bijvoorbeeld een RGB kun je hier leuke projecten mee doen!



Flow sensor

Mijn directe natuurkunde collega gebruikt de flow-sensor bij het bepalen van de wet van Darcy, geofysica practicum voor 5V. De sensor meet het aantal rotaties per seconde, daaruit rolt een frequentie die vervolgens omgezet kan worden naar een stroomsnelheid of debiet.



BMP 180

De BMP 180 is een luchtdruksensor met temperatuursensor. De sensor is prima te verwerken in een weerstation of bij het bepalen van de hoogte.



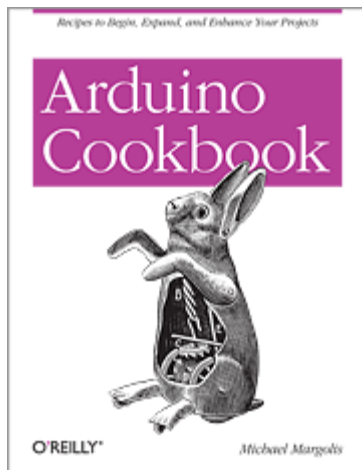
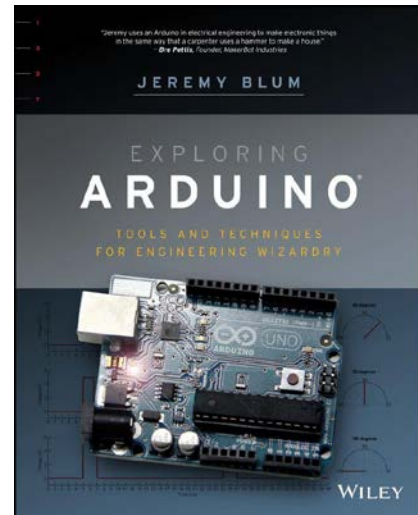
Laser

Voor €0,40 heb je een laser geschikt voor de Arduino. In combinatie met een LDR kun je deze prima gebruiken als tripwire of om tussen twee Arduino's te communiceren.



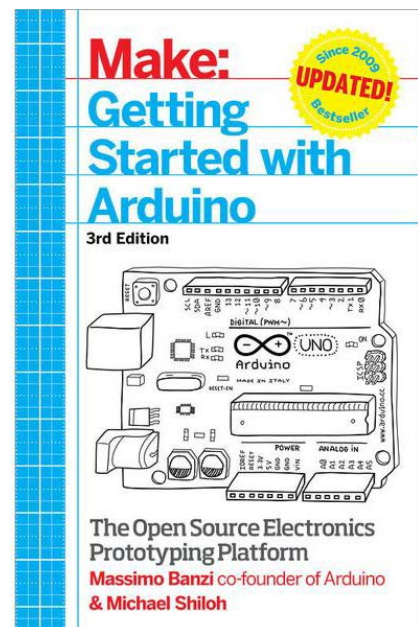
Bronnen

Jeremy Blum heeft een zeer uitgebreid boek over Arduino geschreven: *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. In het boek legt hij diverse programmeertechnieken uit, hij heeft voorbeeld codes en links naar websites, tutorials en instructiefilmpjes. Dit is de voornaamste bron waaruit ik mijn kennis heb.



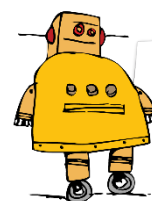
Arduino Cookbook is een mooie bron voor vele projecten en voorbeeld codes. De nieuwste versie is in januari 2017 uitgebracht. Met ruim 600 bladzijdes (18 hoofdstukken) is dit een mooi naslagwerk voor al je Arduino vragen.

Het magazijn MAKE heeft mooie uitgaven van boeken, waaronder *MAKE: getting started with Arduino*. Wil je op een andere manier de projecten uit het basisboek doen, met veel uitleg over de code die gebruikt wordt. Dan is dit boek een aanrader.



Het Arduino projects book zit in het starters pakket van Arduino. Het boek heeft veel kleine projecten met korte uitleg. Om compleet te zijn kun je deze aanschaffen, maar deze biedt m.i. niet veel meerwaarde t.o.v. de basiscursus.

Instructables.com is een website met heel veel ideeën en voorbeelden van gebruik van Arduino. Heb je een mooi project gemaakt, deel die dan op *instructables* zodat anderen er ook gebruik van kunnen maken.



instructables

Scripts

Praten via bluetooth

```
char blueToothVal;
char lastValue;

void setup()
{
  Serial.begin(9600);
  pinMode(11,OUTPUT);
}

void loop(){
  if(Serial.available()){
    blueToothVal=Serial.read();
  }

  if (blueToothVal=='n'){
    digitalWrite(11,HIGH);
    if (lastValue!='n')
      Serial.println(F("LED is on"));
    lastValue=blueToothVal;
  }

  if (blueToothVal=='p'){
    analogWrite(11,100);
    lastValue=blueToothVal;
  }

  else if (blueToothVal=='f'){
    digitalWrite(11,LOW);
    if (lastValue!='f')
      Serial.println(F("LED is off"));
    lastValue=blueToothVal;
  }
  delay(1000);
}
```

Praten met de Arduino

```
int ledPin = 9;

void setup() {
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if(Serial.available()) {
    int brightness = Serial.parseInt();
    analogWrite(ledPin, brightness);
  }
}
```

Praten met Arduino 2

```
int R = 9;           int Rwaarde = 0;
int G = 10;          int Gwaarde = 0;
int B = 11;          int Bwaarde = 0;

void setup() {
  pinMode(R,OUTPUT);
  pinMode(G,OUTPUT);
  pinMode(B,OUTPUT);
  Serial.begin(9600);
  Serial.println("Type R, G of B");
}

void loop() {
  while(!Serial.available() > 0) { }
  String RGB = Serial.readString();
  if(RGB == "R"){
    Serial.println(RGB);
    Serial.println("Type de waarde");
    while(!Serial.available() > 0) { }
    Rwaarde = Serial.parseInt();
    if(Rwaarde > 255){
      Serial.println("Sorry, mag niet");
    }
  }
  else{
    Serial.println(Rwaarde);
    analogWrite(R,Rwaarde);
  }
}
}
```

Afstandssensor

```
int echopin = 12;
int pingpin = 13;
long duration;
int distance_cm;

void setup() {
  pinMode(echopin,INPUT);
  pinMode(pingpin,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(pingpin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingpin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingpin, LOW);
  duration = pulseIn(echopin, HIGH);
  distance_cm = duration/20000*343;
  Serial.print(distance_cm);
  Serial.println(" cm");
  delay(500);
}
```

Timer

```
int i = 0;
int led = 13;
long Timer1 = 0;

void setup() {
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  Timer1 = millis();
  while(millis()-Timer1<1000){ }
  Serial.print(i);
  Serial.println(" s");
  i = i + 1;
  digitalWrite(led,!digitalRead(led));
}
```

Temperatuur naar SD

```
#include <SPI.h>
#include <SD.h>

const int chipSelect = 4;
float tempC;
int tempPin = 0;
int minuten = 0;
long timer = 0;

void setup() {
  Serial.begin(9600);
  analogReference(INTERNAL);
  while (!Serial) { ; }

  Serial.print("Initializing SD card...");

  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    return;
  }
  Serial.println("card initialized.");
}

void loop() {
  if(millis()-timer>60000){
    minuten = minuten + 1;
    timer = millis();
    String dataString = "";
    tempC = analogRead(tempPin) / 9.31;
    delay(1000);
    dataString += String(tempC);
    dataString += " ";
    dataString += String(minuten);

    File dataFile = SD.open("datalog.txt", FILE_WRITE);

    if (dataFile) {
      dataFile.println(dataString);
      dataFile.close();
      Serial.println(dataString);
    }

    else {
      Serial.println("error opening datalog.txt");
    }
  }
}
```

595

```
const int SER =8; //Serial Output to Shift Register
const int LATCH =9; //Shift Register Latch Pin
const int CLK =10; //Shift Register Clock Pin

//Sequence of LEDs
int seq[14] = {1,2,4,8,16,32,64,128,64,32,16,8,4,2};

void setup(){
  pinMode(SER, OUTPUT);
  pinMode(LATCH, OUTPUT);
  pinMode(CLK, OUTPUT);
}

void loop(){
  for (int i = 0; i < 14; i++){
    digitalWrite(LATCH, LOW); //Latch Low - start sending
    shiftOut(SER, CLK, MSBFIRST, seq[i]); //Shift Most Sig. Bit First
    digitalWrite(LATCH, HIGH); //Latch High - stop sending
    delay(100); //Animation Speed
  }
}
```

Hoge resolutie LM35

```
float tempC;
int reading;
int tempPin = 0;

void setup(){
  analogReference(INTERNAL);
  Serial.begin(9600);
}

void loop(){
  reading = analogRead(tempPin);
  tempC = reading / 9.31;
  Serial.println(tempC);
  delay(1000);
}
```

238E

```
int pin[] = {2,3,4};
int var = 0;
void setup() {
  for(int i = 0; i<3; i++){
    pinMode(pin[i],OUTPUT);
  }
  Serial.begin(9600);
}

void loop() {
  for(int i =0; i<3; i++){
    digitalWrite(pin[i],bitRead(var,i));
  }
  delay(250);
  var = var + 1;
}
```

Muur ontwijkende auto

//het programma is (te) eenvoudig geschreven, maar doet het wel! Er is maar een snelheid en dat is full speed!

```
const int pingPin = 13;  const int echoPin = 12;
const int MR=8;         const int ML=7;
long duration, inches;
long cm = 50;

void setup() {
  pinMode(pingPin, OUTPUT);  pinMode(echoPin, INPUT);
  pinMode(MR, OUTPUT);      pinMode(ML, OUTPUT);
  vooruit();
}

void loop()
{
  while(cm>10){
    vooruit();  afstand();    delay(10);
  }
  rem();      delay(500);    turn();      afstand();
}

long microsecondsToCentimeters(long microseconds){
  return microseconds / 29 / 2;
}

void afstand(){
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  cm = microsecondsToCentimeters(duration);
}

void vooruit(){
  digitalWrite(ML,HIGH); digitalWrite(MR,HIGH);
}

void rem(){
  digitalWrite(ML,LOW); digitalWrite(MR,LOW);
}

void turn(){
  digitalWrite(ML,LOW); digitalWrite(MR,HIGH);
  delay(1000);
}
```

RTC

```
#include <DS3231.h>
//http://www.rinkydinkelectronics.com/library
.php?id=73
DS3231 rtc(SDA, SCL); //Uno SDA: A4 SCL: A5

int alarmh = 20;
int alarmm = 45;
Time t;
int buzzer = 13;
int told;

void setup() {
  Serial.begin(115200);
  rtc.begin();
  pinMode(buzzer,OUTPUT);
  // rtc.setDOW(WEDNESDAY); // Set Day-
of-Week to WEDNESDAY
  // rtc.setTime(20, 20, 0); // Set the time to
20:20:00 (24hr format)
  // rtc.setDate(04, 01, 2016); // Set the date
to January 4th, 2016
  t = rtc.getTime();
  told = t.min;
}

void loop() {
  t = rtc.getTime();
  if(t.min == told+1){
    Serial.println(rtc.getTimeStr());
    told = t.min;
  }

  if(alarmh == t.hour && alarmm == t.min){
    tone(buzzer,440,100);
  }
}
```

```
const int sensor = 2;
int n = 0;
long timer = 0;

void setup(){
  Serial.begin(115200);
  pinMode(sensor,INPUT);
  attachInterrupt(0, counter, FALLING);
}

void loop(){
  timer = millis();
  while(millis()-timer<10000){
  }
  Serial.print("aantal detecties: ");
  Serial.print(n);
  Serial.print("\t");
  Serial.print("frequetie: ");
  Serial.print(n/10);
  Serial.println(" Hz");
  n = 0;
}

void counter(){
  n = n + 1;
}
```