

Arduino Document.

## Port Manipulatie

Omdat ik dus met lichten en geluid aan de slag wilde was het tijd om de code in te duiken en te kijken wat ik het beste kon doen met lichten. In eerste instantie wilde ik beginnen met de sensoren alleen had ik die nog niet tot m'n beschikking. Door verder te kijken dan de sensoren ben ik gaan denken over de output. Om lichtjes bijvoorbeeld apart te sturen die reageren op beweging zou dit voor teveel regels code zorgen als ik elk lampje individueel een zou aansturen en een opdracht gaf. Op dit gebied ging ik even op onderzoek en kwam ik de oplossing 'Port Manipulatie' tegen en ben gaan onderzoeken hoe ik dit kan gebruiken voor de Arduino. Port manipulatie zorgt ervoor dat je niet elk lampje hoeft aan te sturen maar dat je gewoon een gehele Port kan aansturen aan de hand van gegeven input.

Port Registers maken een lagere en snellere manipulatie van de i/o pinnen van de Arduino mogelijk. De chips die op het Arduino-bord worden gebruikt, hebben drie poorten:

- Port B (digital pin 8 to 13)
- Port C (analog input pins)
- Port D (digital pins 0 to 7)

Elke Port wordt bestuurd door 3 register die als variabelen in de Arduino te vinden zijn. Het DDR-register bepaalt of de pin een INPUT of OUTPUT is. Het Port-register bepaalt of de pin HIGH of LOW is en het Pin-register leest de toestand van INPUT-pinnen ingesteld op invoer met de standaard pinMode() code.

In dit geval gebruikte ik Port D aangezien op de Port D al mijn led lichtjes op aangesloten zijn. Port D maps alle digitale pins van 0 tot 7 op de Arduino. De DDR die je doorgeeft kan een 1 of 0 zijn, waarin de 1 voor HIGH staat en 0 voor LOW. Zonder Port manipulatie zou de code er zo uitzien om een lichtje aan te krijgen:

```
1. digitalWrite(blauwLeftPin, HIGH);
2.   delay(100);
3.   //digitalWrite(whiteBlueleftPin, LOW);
4.   //delay(100);
```

Dit zou echter voor een veel te lange code geven als er een sequens gebouwd moet worden van meerdere lampjes. Met behulp van Port D & DDR wordt het zo gedaan:

```
1. DDRD = B11111110; // sets Arduino pins 1 to 7 as outputs, pin 0 as input
2. DDRD = DDRD | B11111100; // this is safer as it sets pins 2 to 7 as outputs
3.                               // without changing the value of pins 0 & 1, which are RX & TX
4.
5. PORTD = B11000000;
6.   delay(100);
7. PORTD = B00000000;
8.   delay(100);
9. PORTD = B11000000;
10.  delay(100);
11. PORTD = B00000000;
```

```

12. delay(100);
13. PORTD = B11000000;
14. delay(100);
15. PORTD = B00000000;
16. delay(100);
17. PORTD = B11000000;
18. delay(100);
19. PORTD = B00000000;
20. delay(100);

```

Met behulp van Port D kan ik de gehele analoge rij aansturen met een regel code welke wel op HIGH moeten worden gezet en welke niet. Hiervan kan meerdere sequensen achter elkaar zetten waardoor een hele grote sequens krijgt. Dit is qua code veel voordeliger en netjes aangezien het niet veel ruimte meer inneemt.

```

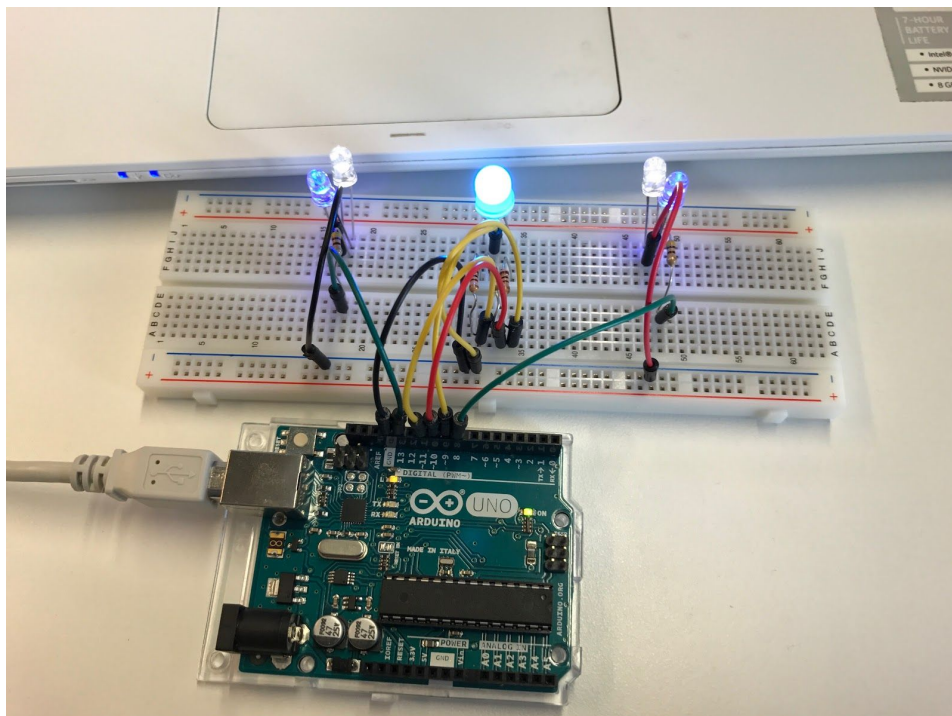
1. // Alle Pin aangeven
2. const int redPin = 11;
3. const int greenPin = 10;
4. const int bluePin = 9;
5. const int blauwLeftPin = 5;
6. const int witLeftPin = 4;
7. const int blauwRightPin = 6;
8. const int witRightPin = 7;
9. // Functie
10. unsigned int rgbColour[3];
11.
12. void setup() {
13.   // RGB LED moet uit beginnen.
14.   setColourRgb(0, 0, 0);
15.   pinMode(blauwLeftPin, OUTPUT);
16.   pinMode(witLeftPin, OUTPUT);
17.   pinMode(blauwRightPin, OUTPUT);
18.   pinMode(witRightPin, OUTPUT);
19. }
20.
21. void loop() {
22.
23.   DDRD = B11111110; // Pin 1 tot 7 Als Outputs, pin 0 als input
24.   DDRD = DDRD | B11111100; // Pin 2 tot 7 als Outputs
25.   // Hierdoor laat ik 0 en 1 met rust aangezien zij RX & TX zijn
26.
27.   //Port manipulatie welke pins aan en uit gaan tussen 7,6,5 en 4
28.   PORTD = B11000000;
29.   delay(100);
30.   PORTD = B00110000;
31.   delay(100);
32.   PORTD = B11000000;
33.   delay(100);
34.   PORTD = B00000000;
35.   delay(100);
36.   PORTD = B11000000;
37.   delay(100);
38.   PORTD = B00110000;

```

```

39. delay(100);
40. PORTD = B11000000;
41. delay(100);
42. PORTD = B00000000;
43. delay(100);
44.
45. // Start RGB LED met Red als kleur. Groen en blauw 0 waarde.
46. rgbColour[0] = 255;
47. rgbColour[1] = 0;
48. rgbColour[2] = 0;
49. rgbColour[3] = 0;
50.
51. // Increment & decrement tussen de kleuren.
52. for (int decColour = 0; decColour < 3; decColour += 1) {
53.     int incColour = decColour == 2 ? 0 : decColour + 1;
54.
55.     // cross-fade tussen de RGB kleuren.
56.     for (int i = 0; i < 255; i += 1) {
57.         rgbColour[decColour] -= 1;
58.         rgbColour[incColour] += 1;
59.
60.         setColourRgb(rgbColour[0], rgbColour[1], rgbColour[2]);
61.         delay(5);
62.     }
63. }
64.
65. //Het zetten van de Brightness voor elke pin.
66. void setColourRgb(unsigned int red, unsigned int green, unsigned int blue) {
67.     analogWrite(redPin, red);
68.     analogWrite(greenPin, green);
69.     analogWrite(bluePin, blue);
70. }

```



Nadat het was gelukt om lichten aan te sturen (en mijn enthousiasme nog hoog was) wilde ik meteen verder gaan in het proces dat ik 3 verschillende sensoren besteld heb (en andere arduino accessoires) om mee verder te experimenteren. Om uiteindelijk een interactieve installatie te maken zijn sensoren nodig om bepaalde dingen aan te sturen. In dit geval het licht. Het gaat om 3 sensoren waar ik mee geëxperimenteerde heb;

- PIR Sensor

*A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors.*

- IR Sensor

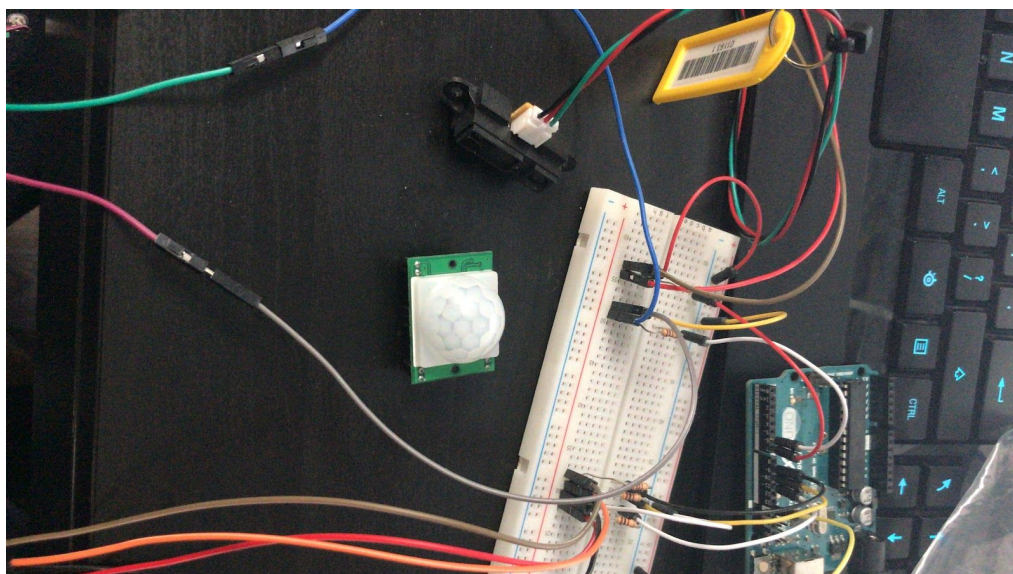
*IR detectors are little microchips with a photocell that are tuned to listen to infrared light. They are almost always used for remote control detection - every TV and DVD player has one of these in the front to listen for the IR signal from the clicker. Inside the remote control is a matching IR LED, which emits IR pulses to tell the TV to turn on, off or change channels. IR light is not visible to the human eye, which means it takes a little more work to test a setup.*

- Ultrasonic Sensor

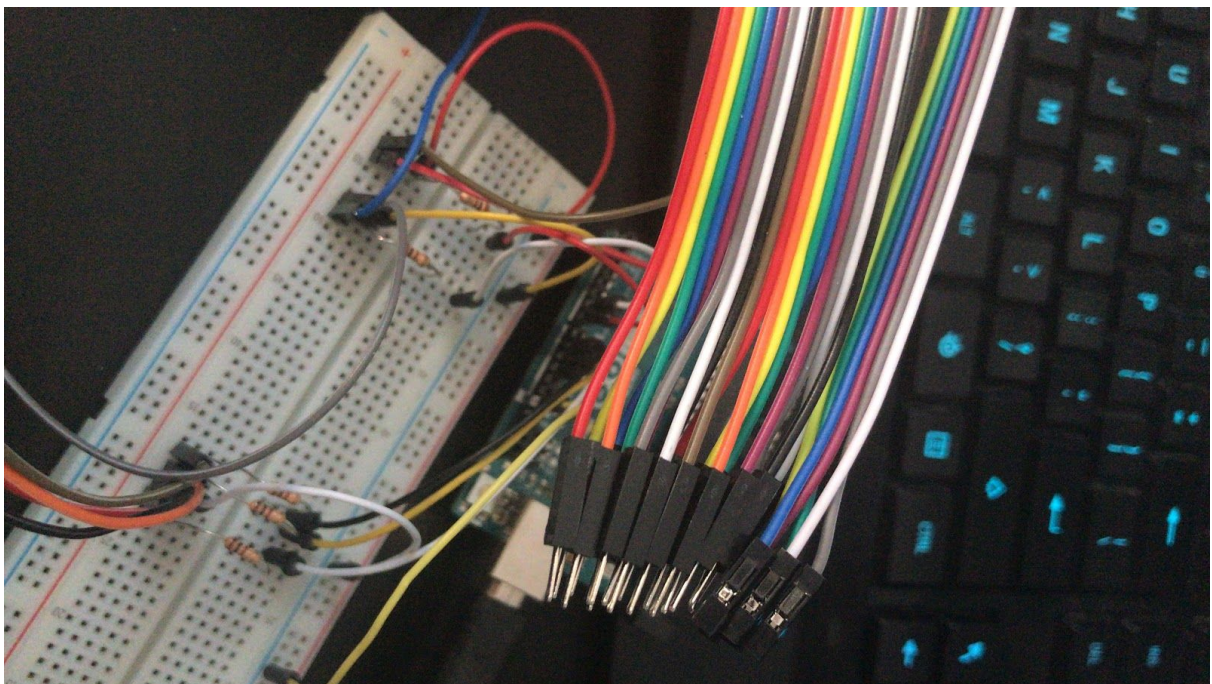
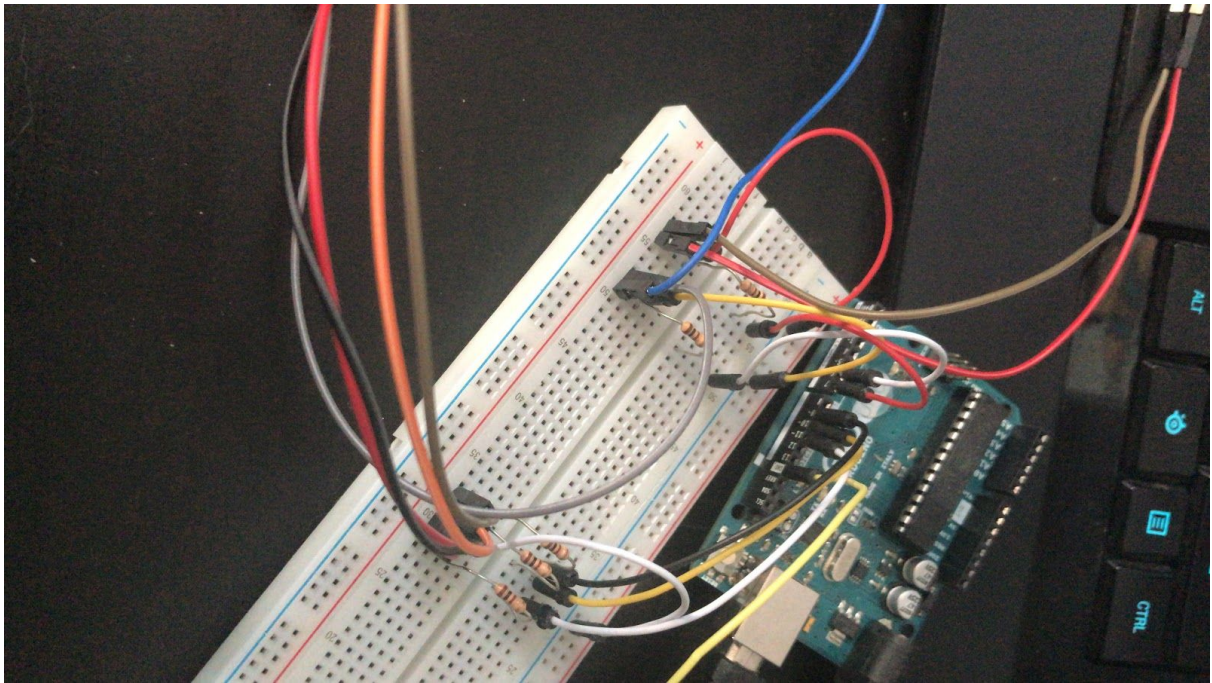
*Ultrasonic transducers or ultrasonic sensors are a type of acoustic sensor divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.*

Na het onderzoeken van deze sensoren ben ik erachter gekomen dat ik de IR sensor niet zal gebruiken aangezien ik niet van plan ben om direct met infrarood signalen te spelen. Zelf als 'speler' binnen de installatie is het niet zo dat jij een apparaat hebt/krijgt wat infrarood signalen zal uitstralen en opgepikt worden door de IR sensor.

PIR en Ultrasonic zijn wel 2 geschikte sensoren aangezien de een een motion detection sensor is en de ander een afstandsmeter. Met deze twee factoren kunnen bepaalde dingen aangestuurd worden, in dit geval bijvoorbeeld lichten.







### Conclusie

Na het succesvol aan te sluiten van de sensoren was het niet zo moeilijk om een connectie te maken met de LED's. Uit de twee gebruikte sensoren kan ik concluderen dat er veel gewerkt zal worden met de PIR sensor (motion sensor) aangezien het beweging detecteert. Voor een interactieve (dans) installatie is beweging cruciaal, dus het gebruik van PIR sensoren zal hier goed voor dienen.

Een volgende stap zou nu zijn hoe ik hiermee verder ga. Hoe zie ik dit in praktijk? Kan een interactieve installatie al gemaakt worden met 1 sensor? Of heb ik er meerdere nodig?

## Next step

Nu ik weet dat het mij is gelukt om een bewegingssensor en licht met elkaar te verbinden kan hier in principe al een prototype installatie van gemaakt worden. Voor een installatie hoeft het in mijn ogen niet al meteen grootschalig. Met 1 sensor kan je al veel doen. Zo is het mogelijk om lichten aan te sturen met beweging. Hier zelf kan al bijvoorbeeld een dans choreo gemaakt worden met licht.

De sensor zou eventueel ook webbased gelinkt worden en kunnen reageren met een 3D model. Wat je hier dan voor ogen kunt zien is dat er bijvoorbeeld een 'blob slijm' met je meebeweegt of reageert op je beweging. Echter weet ik op dit moment niet hoe ik dat zou moeten doen.

Als je kijkt naar de mogelijkheden van het connecten van sensoren naar digitaal is er best veel wat je kunt doen. Een daarvan is hierboven al gegeven; Het laten bewegen van een 3D object aan de hand van de gegeven sensoren. Maar onder andere concepten hoeft dit niet perse een 3D object te zijn. Dit kan net zo goed een webbased visualizer zijn die meebeweegt op beweging.