Inhoudsopgave

Inleiding 3

Voorbereiding 4

Interieur 4

Hardware 4

Arduino Uno (compatible) + USB-kabel (1) 5

PCB (2) 5

Dupont draden (3) 5

HM-10, Bluetooth 4.0 module (4) 6

RGB LED module (5) 6

Buzzer (6) 7

Servo Motor (7) 7

9V batterij + kabel (8) 7

“IoT Kijkdoos” app 8

Software 8

Arduino IDE installeren (Windows + Mac) 8

Arduino sketch openen 8

Sketch Uploaden 9

Debuggen 9

Challenges 10

Challenge I Broken Light 10

Challenge II Gradient 10

Challenge III Custom Sound 11

Challenge IV Special Effect 11

Functieblad 12

Inleiding

Beste deelnemer, allereerst welkom! Ordina heeft voor dit hands-on lab de IoT Kijkdoos ontworpen. Dit is de kijkdoos van vroeger, maar dankzij domotica klaar voor de 21ste eeuw!

Voor u ligt een pakket dat is uitgedeeld door het Ordina team. Gedurende dit hands-on lab gaat u met dit pakket uw eigen IoT Kijkdoos maken en besturen met de “IoT Kijkdoos” app. Als dat gelukt is gaat u de Arduino programmeren door vier challenges uit te voeren. Als iets niet lukt, schroom niet en vraag het Ordina team om u te helpen. Het volgende is vereist:

* Een telefoon of tablet (Android SDK 18/iOS 9 of hoger) met bluetooth.
* Een Laptop (Windows of Mac).

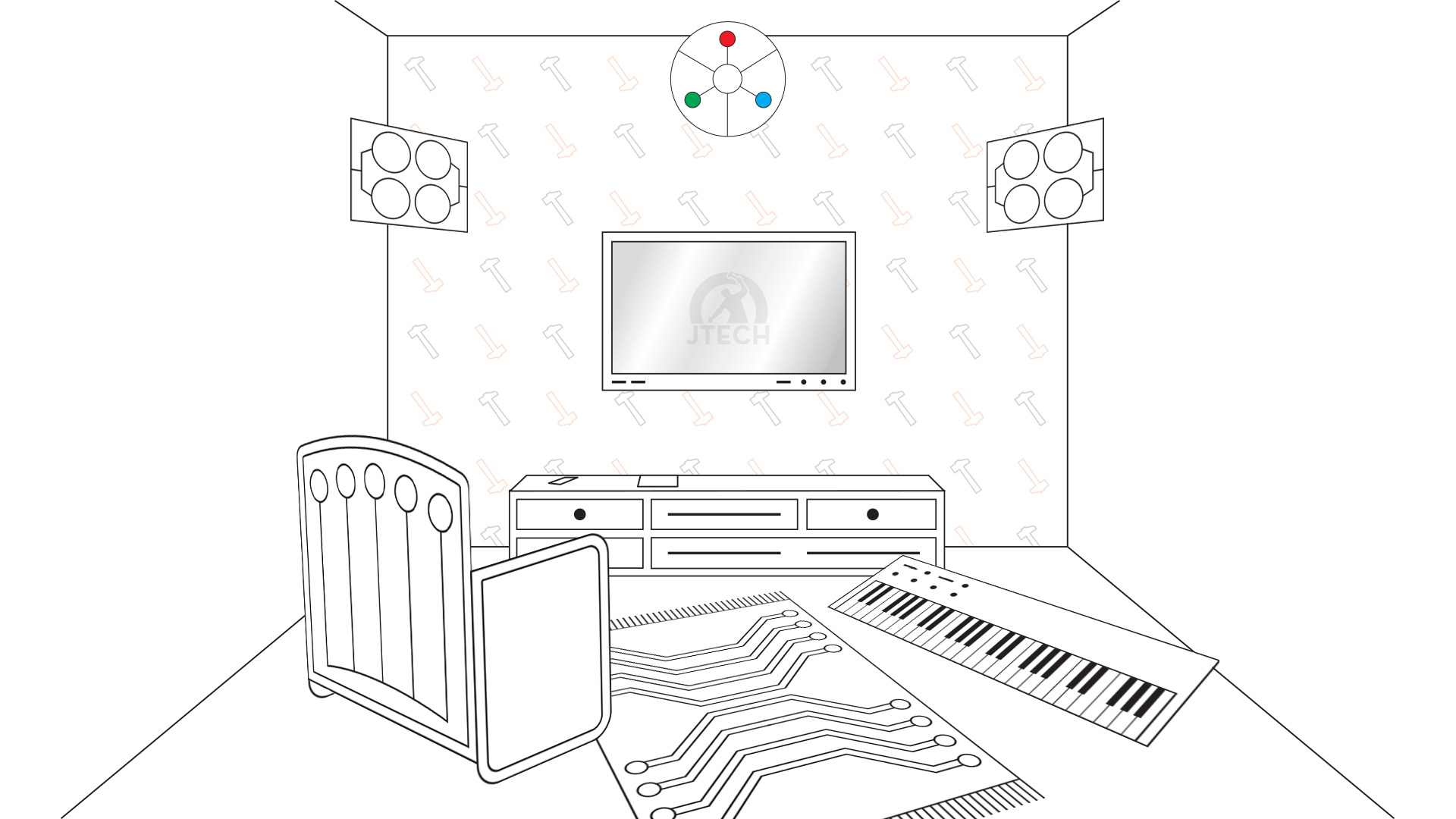
Voorbereiding

U kunt alvast de kartonnen doos in elkaar zetten en het kijkgat maken. Hiervoor gaan een aantal stanleymessen rond.

# Interieur

U heeft twee bladen papier gekregen. Het blad met de hamertjes op de achtergrond zal fungeren als dubbele wand. Hierachter zullen de elektronische componenten worden geplaatst, zodat ze niet te zien zijn in de kijkdoos. Op de plaats van de vier kruisjes moeten gaatjes gemaakt worden. De bovenste drie ter grootte van een LED en de onderste zo groot dat de servo tot aan de uitsteeksels erin past (zie “Hardware”). De dubbele wand kan vervolgens in de kijkdoos worden bevestigd. Let er wel op dat er voldoende ruimte is tussen de achterkant van de doos en de dubbele wand.

Op het andere blad is het interieur voor in de kijkdoos weergegeven. Knip de losse attributen uit en plak deze in de kijkdoos als in Afbeelding 1. Wacht met het plakken van de televisie totdat de servo in de dubbele wand is bevestigd, zodat de televisie zeker horizontaal hangt.



Afbeelding 1 interieur IoT Kijkdoos.

# Hardware

In Afbeelding 2 zijn de elektronische componenten weergegeven die u nodig heeft voor in de kijkdoos. Kijk goed of alles aanwezig is. Zo niet, meld dit dan even bij het Ordina team en dan krijgt u het alsnog. Onder Afbeelding 2 is per genummerd component een korte uitleg gegeven over wat het is en hoe het aangesloten dient te worden.



1

2

3

4

5

6

7

8

Afbeelding 2 Benodigde elektronische componenten.

## Arduino Uno (compatible) + USB-kabel (1)

De Arduino is een microcontroller. Het kan analoge input lezen en elektronische componenten aansturen d.m.v. digitale output. Deze Arduino is een compatible versie. Dat wil zeggen dat het gebaseerd is op de open-source hardware van de originele Arduino. De USB-kabel is voor het programmeren op de Arduino.

## PCB (2)

Het Printed Circuit Board (PCB) is het zwarte board met het witte Ordina/JTech-logo erop. Hierin zitten de elektrische circuits van de elektronische componenten verwerkt. Hierdoor worden het aantal losse draden gereduceerd en is er minder kans op hardware fouten. Dit board moet met de ijzeren pinnen op de zwarte headers van de Arduino worden geklikt.

## Dupont draden (3)

Met de Dupont draden koppelt u de elektronische componenten aan de PCB. Deze draden zitten aan elkaar, maar kunnen makkelijk gescheiden worden.

## HM-10, Bluetooth 4.0 module (4)

De bluetooth module zorgt voor de draadloze communicatie met de “IoT Kijkdoos” app. Deze module is direct aan te sluiten op de PCB, zie Tabel 1. De module bevat een rode LED die knippert wanneer er geen verbinding is en constant blijft als dit wel het geval is.

Op de module is een sticker geplakt met de naam van uw bluetooth module. Deze naam moet u in de app selecteren om uw IoT Kijkdoos te besturen. Dit wordt onder het kopje “Software” verder uitgelegd.

|  |  |
| --- | --- |
| Bluetooth | PCB (BT) |
| VCC | 3.3V |
| GND | GND |
| TX | D10 |
| RX | D11 |

Tabel 1 bevestiging van bluetooth module op de PCB.

## RGB LED module (5)

U heeft drie RGB LED modules gekregen. Een wordt er daadwerkelijk gewerkt als RGB LED en de andere twee als normale LED(aan/uit). Elke module heeft vier aansluitingen: R (rood), G (groen), B (bauw) en – (GND). Het is de bedoeling dat de module met de Dupont draden wordt bevestigd aan de PCB. Zie Tabel 2 voor de bevestiging van de normale LED en Tabel 3 voor de RGB LED.

|  |  |
| --- | --- |
| LED | PCB (LED 1+2) |
| R/G/B | D7 + D8 |
| - | GND |

Tabel 2 bevestiging van LED op de PCB.

|  |  |
| --- | --- |
| RGB LED | PCB (RGB) |
| R | D9 |
| G | D6 |
| B | D5 |
| - | GND |

Tabel 3 bevestiging RGB LED op de PCB.

## Buzzer (6)

Met de buzzer kunnen er tonen afgespeeld worden. Op de bovenkant van de buzzer staat een plusteken (+). Dit geeft aan dat de ijzeren pin aan die zijde de +5V kant is. De andere pin is GND. Zie Tabel 4 voor de juiste koppeling met de PCB. De ijzeren pootjes zijn wat aan de kleine kant, dus zorg ervoor dat de pootjes volledig in de zwarte headers van de PCB zijn gedrukt.

|  |  |
| --- | --- |
| Buzzer | PCB (BUZZER) |
| +5V | D4 |
| GND | GND |

Tabel 4 bevestiging buzzer op de PCB.

## Servo Motor (7)

De servo motor is een licht motortje en heeft een maximale hoekuitslag van 180°. Aan de servo zitten drie gekleurde draden. Deze moeten gekoppeld worden aan de drie ijzeren pinnen op de PCB. Zie Tabel 5 voor de juiste bevestiging.

|  |  |
| --- | --- |
| Servo | PCB (SERVO) |
| Oranje | D3 |
| Rood | 5V |
| Bruin | GND |

Tabel 5 bevestiging servo op de PCB.

## 9V batterij + kabel (8)

De 9V batterij en kabel zorgen ervoor dat de Arduino draadloos kan functioneren. Als de batterij leeg is kunt u de Arduino voeden door deze met de USB-kabel aan de computer te bevestigen.

# “IoT Kijkdoos” app

Ordina heeft voor dit hands-on lab de “IoT Kijkdoos” app gecreëerd. Hiermee kunnen de elektronische componenten in de IoT Kijkdoos draadloos bestuurd worden. Deze app is zowel in de Play Store als de App Store te downloaden door op de naam te zoeken of de QR-code in Afbeelding 3 of 4 te scannen.



Afbeelding 3 Android QR-code.



Afbeelding 4 iOS QR-code.

Als de app geopend is ziet u een lijst met alle IoT Kijkdozen in de buurt. Kies de naam die op de bluetooth module vermeld staat. Mocht de naam niet in de lijst voorkomen, kijk dan goed of de Arduino aan staat en dat de bluetooth module juist is aangesloten. Eenmaal gekoppeld kunt u de IoT Kijkdoos tot leven laten komen door op het interieur te klikken!

# Software

## Arduino IDE installeren (Windows + Mac)

* Ga naar de map “Arduino IDE” en kies het juiste besturingssysteem.
* Dubbelklik op het bestand en volg de instructies.

## Arduino sketch openen

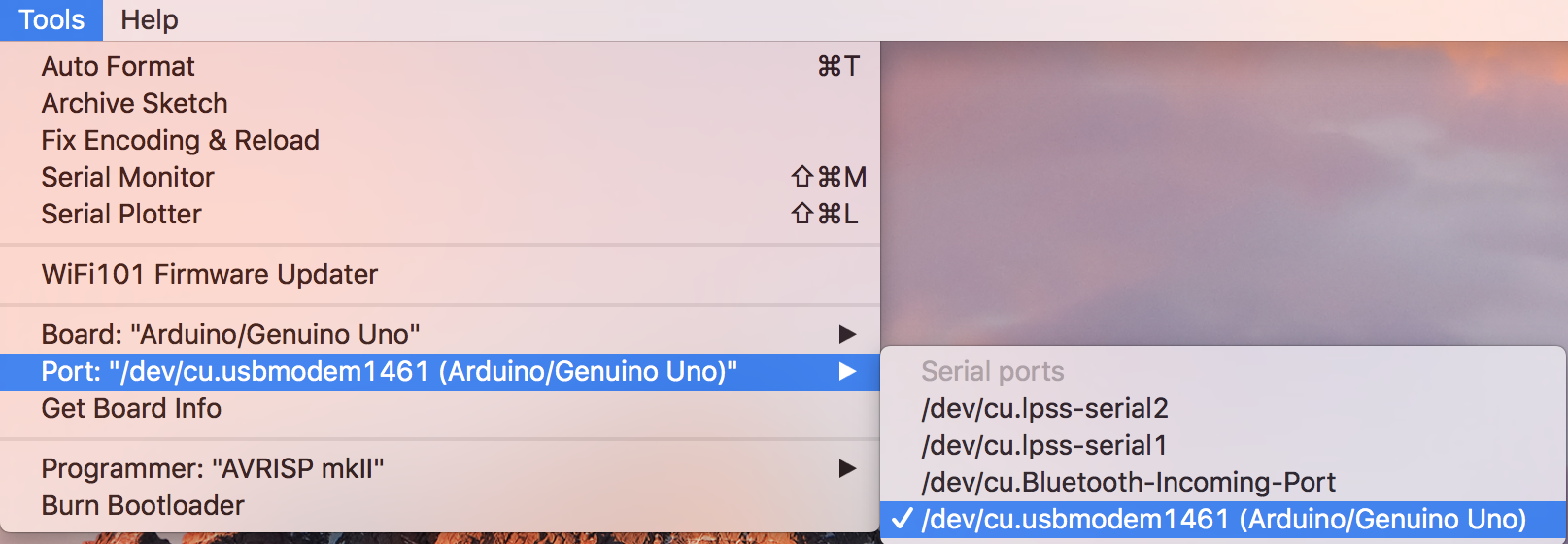
Ga naar de map “Arduino Sketch” en open het hoofdbestand “Ordina.ino”. Hierdoor wordt de Arduino IDE geopend en worden er meerdere tabbladen weergegeven. De bluetooth communicatie en elk elektronisch component heeft zijn eigen class. In het *.h* bestand staat de declaratie van class en in het *.cpp* bestand de implementatie hiervan. Het programma dat op de Arduino draait wordt een sketch genoemd.

## Sketch Uploaden

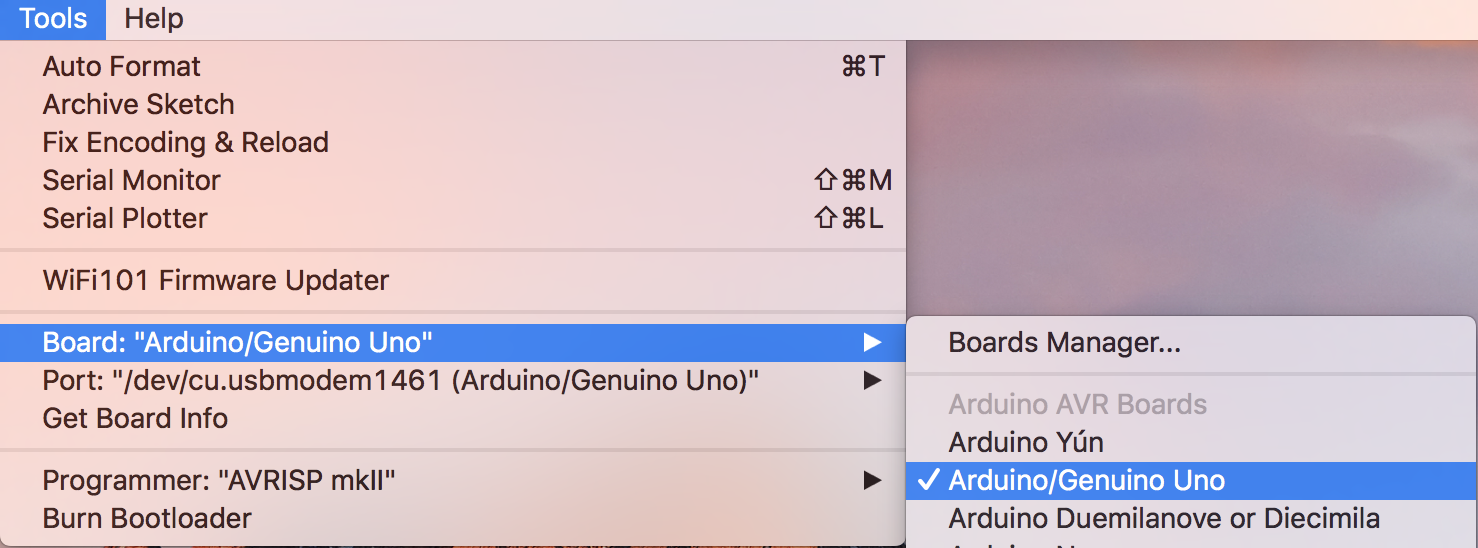
Als de code veranderd is moet deze naar de Arduino geüpload worden.

Voordat dit kan moet u eerst het volgende doen:

1. Koppel de Arduino aan de computer (USB-kabel).
2. Selecteer het juiste board (Afbeelding 5).
   * *Tools -> Board -> Arduino/Genuino Uno*
3. Selecteer de juiste poort (Afbeelding 6).
   * *Tools -> Port -> “USB naam” (Arduino/Genuino Uno)*
4. Controleer of de code geen error bevat.
   * *Klik linksboven op het vinkje*



Afbeelding 6 dejuiste poort selecteren.



Afbeelding 5 het juiste board selecteren.

Indien de code een error bevat dient deze eerst opgelost te worden. Als het niet lukt om dit zelf op te lossen, aarzel dan niet om iemand van het Ordina team aan te spreken. Als het bovenstaande is uitgevoerd bent u klaar om de sketch te uploaden. Klik linksboven op het pijltje en wacht tot het uploaden is voltooid.

## Debuggen

Stel dat u de code heeft veranderd en geüpload, maar er gaat iets niet goed en u weet niet waardoor het komt. Dan kunt u Serial.println()gebruiken om een waarde in de Serial Monitor te printen (zie functiekaart). Let er wel op dat de Serial Monitor geopend is tijdens het runnen van de code (*Tools -> Serial Monitor*).

Challenges

Gefeliciteerd! Het is u gelukt om een eigen IoT Kijkdoos te maken en te besturen. Ordina heeft voor u vier challenges bedacht om extra functionaliteit aan de IoT Kijkdoos te geven en te leren programmeren met Arduino. Alle functies die u voor de challenges nodig heeft zijn onderstreept en zijn nader toegelicht in het hoofdstuk “Functieblad”. U kunt de Arduino website raadplegen voor een duidelijkere uitleg en andere functies (www.arduino.cc). Succes met het uitvoeren van de challenges!

## Challenge I Broken Light

Zoals u gemerkt heeft is de lamp aan de rechterkant defect. Aan u de uitdaging om deze weer werkend te krijgen. Tijdens deze challenge kunt u gewend raken aan het veranderen en uploaden van de code. U kunt in het vorige hoofdstuk nogmaals het kopje “Sketch Uploaden” doornemen als dit niet meer duidelijk is. Neem tijdens deze challenge ook vooral de tijd om even door de gehele code te kijken.

*Tips:*

* De lamp kan gemaakt worden in het hoofdbestand (Ordina.ino).

## Challenge II Gradient

De kleur van de discobal kan veranderd worden door gebruik te maken van de slider in de app, maar hoe leuk is het om met één druk op de knop een kleurverloop te zien? Ga naar het bestand RGB.cpp, scroll naar de methode *RGB::showGradient()* en voeg tussen de accolades de code die zorgt voor het kleurverloop. U kunt hiervoor de array “discoStates” en de variabele “nStates” (lengte discoStates) gebruiken. Het is natuurlijk ook mogelijk om een eigen array met kleuren te creëren.

*Tips:*

* Gebruik de methode *RGB::writeColor(int rgbValue[3])* in RGB.h om de kleur van RGB LED te veranderden.
* De functie delay() kan worden gebruikt om tussen het veranderen van de kleuren een pauze te creëren.

## Challenge III Custom Sound

|  |  |
| --- | --- |
| Noot | Frequentie (Hz) |
| c | 262 |
| d | 294 |
| e | 330 |
| f | 349 |
| g | 392 |
| a | 440 |
| b | 494 |
| C | 523 |
| D | 587 |
| E | 659 |
| F | 699 |
| G | 784 |
| A | 880 |
| B | 989 |

Het is tijd om een eigen geluid te maken met de buzzer. Ga naar het bestand Buzzer.cpp en scroll naar de methode *Buzzer::customSound()*. Met de functie tone() kunt u de buzzer een bepaalde toon laten produceren.

Voor de muzikale deelnemers is hiernaast een tabel weergegeven met daarin de noten met bijbehorende frequenties. Experimenteren met andere frequenties kan natuurlijk ook. De minimale frequentie is 31 Hz en de maximale 65535 Hz.

*Tips:*

* De functie delay() kan worden gebruikt om een pauze tussen tonen te creëren.
* Bij herhalende tonen kunt u een for loop gebruiken om codeduplicatie tegen te gaan.

## Challenge IV Special Effect

Tabel 6 overzicht noten met bijbehorende frequentie (Hz).

Deze challenge is speciaal. Tot nu toe hebben wij voor u bepaald welke methode er geïmplementeerd moest worden, maar nu is de beurt aan uzelf! Creëer een eigen methode voor één van de vier componenten en zorg dat deze wordt aangeroepen als de character ‘i’ wordt ontvangen in het hoofdbestand. Laat iedereen versteld staan van uw ‘Special Effect’!

Functieblad

|  |  |  |
| --- | --- | --- |
| Syntax | Parameters | Uitleg |
| if (condition) { //Statements } | **condition:** de uitkomst van de conditie moet altijd waar of niet waar zijn. Als de uitkomst waar is worden de regels code binnen het code block uitgevoerd. | De *if loop* wordt gebruikt om een codeblok alleen uit te voeren als er aan de conditie wordt voldaan. |
| for (initialization; condition; increment) { //Statements }  Afbeelding De For loop | **int x = 0:** de variabele x wordt hier verklaard. Het getal 0 is de startwaarde van de *for loop*. Deze kan zelf bepaald worden.  **x < 100:** De conditie. Als x groter of gelijk is aan 100 dan wordt de loop verlaten. Deze waarde kan zelf bepaald worden.  **x++:** de toename (in dit geval). x wordt hier elke ronde met 1 verhoogd. | De *for loop* wordt gebruikt om een codeblok meerdere keren uit te voeren. In Afbeelding 1 wordt bijvoorbeeld het getal van 0 tot 99 geprint. |
| Serial.print(val)  Serial.println(val) | **val:** de waarde die geprint wordt in de Serial Monitor. Dit kan elke datatype zijn. | U kunt *Serial.println()* gebruiken om te debuggen. U gebruikt het bijvoorbeeld als u wil weten of een methode wordt aangeroepen of wat de waarde van een variabele is. Let er wel op dat u de Serial Monitor heeft geopend tijdens het runnen van de code (*Tools -> Serial Monitor*). |

|  |  |  |
| --- | --- | --- |
| Syntax | Parameters | Uitleg |
| tone(pin, frequency)  tone(pin, frequency, duration) | **pin**: de output pin van de buzzer.  **frequency**: de frequentie van de toon in hertz.  **duration**: de duur van de toon in milliseconden. Deze parameter is optioneel. | *tone()* wordt gebruikt om een toon te produceren met de buzzer. |
| delay(ms) | **ms:** de lengte van de pauze in milliseconden. | *delay()* wordt gebruikt om het programma voor x aantal milliseconden te pauzeren. |