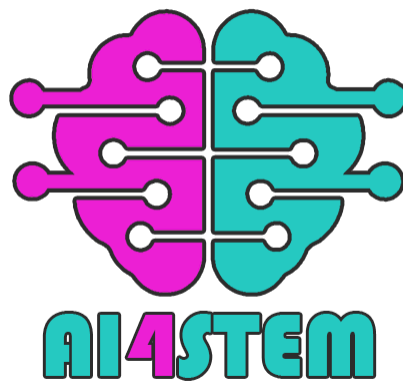




Medegefinancierd door  
de Europese Unie

Gefinancierd door de Europese Unie. De hier geuite ideeën en meningen komen echter uitsluitend voor rekening van de auteur(s) en geven niet noodzakelijkerwijs die van de Europese Unie of het Europese Uitvoerende Agentschap onderwijs en cultuur (EACEA) weer. Noch de Europese Unie, noch het EACEA kan ervoor aansprakelijk worden gesteld.

# IoT robotauto



Introductie van de 5 grote ideeën op het gebied van  
kunstmatige intelligentie met behulp van het internet der  
dingen in STEM-onderwijs  
T2.4 IoT-projectenontwerp en ontwikkeling van middelen

06.10.2023 | EDUMOTIVA  
PROJECT NUMBER: 2022-1-FR01-KA220-SCH-000085611

# AI4STEM IOT-projecten

## Project: IoT robotauto

### Copyright

© Copyright het AI4STEM Consortium

2022-1-FR01-KA220-SCH-000085611

Alle rechten voorbehouden.



AI4STEM IOT-projecten Project: IoT robotauto © 2023 bij [AI4STEM CONSORTIUM](#) is gelicentieerd onder [Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Internationaal](#)

## Inhoudsopgave

|  |    |
|--|----|
| 1 . Inleiding tot het project .....  | 4  |
| 1.1 De reikwijdte van het Project.....   | 4  |
| 1.2 De doelgroepen.....  | 5  |
| 1.3 Het doel van dit document .....  | 5  |
| 2. Woordenlijst van de eenheid.....  | 5  |
| 3. Introductie van de “DIY-robotauto die kan worden bestuurd en bestuurd met behulp van spraakopdrachten” .....                        | 6  |
| 3.1 Beschrijving .....   | 6  |
| 3.2 Leerdoelen & resultaten .....  | 7  |
| 3.3 Geschatte duur van de Eenheid.....   | 8  |
| 3.4 Activiteit 1 – Introductie van het grote idee van perceptie via IoT: .....   | 9  |
| 3.4.1 Beschrijving .....   | 9  |
| 3.4.2 Hardware .....   | 9  |
| 3.4.3 Installatie.....   | 9  |
| 3.4.4 Experiment 1 .....   | 18 |
| 3.5 Activiteit 2: Introductie van het idee van representatie en redeneren.....   | 20 |
| 3.5.1 Beschrijving .....   | 20 |
| 3.5.2 Een beslisboom maken .....   | 20 |
| 3.5.3 Ontwerpen en programmeren van de applicatie .....  | 21 |
| 3.5.4 Experiment 2 .....   | 30 |
| 3. 6 Activiteit 3: Introductie van het idee van leren door een model te trainen voor het herkennen van stemcommando's.....             | 32 |
| 3.6.1 Beschrijving .....   | 32 |
| 3.6.2 Personal Audio Classifier gebruiken om een model te trainen .....  | 32 |
| 3.6.3 Experiment 3 .....   | 37 |
| 3. 7 Activiteit 4: Introductie van het idee van natuurlijke interactie door een getraind model te integreren in een AI-toepassing..... | 39 |
| 3.7.1 Beschrijving .....   | 39 |
| 3.7.2 Integreren van het getrainde model in de AI-applicatie.....  | 39 |
| 3.7.3 Experiment 4 .....   | 50 |
| 3. 8 Activiteit 5: Introductie van het idee van maatschappelijke impact .....  | 52 |
| 3.8.1 Beschrijving .....   | 52 |

|  |    |
|--|----|
| 3. 9 Materiaal en hulpbronnen .....      | 53 |
| 3.10 De hardware voor de robotauto ..... | 54 |

## 1 . Inleiding tot het project

Het huidige project richt zich op de creatie van een doe-het-zelf-robotauto die kan worden bestuurd door middel van stemcommando's, terwijl hij tegelijkertijd een aantal gegevens kan verzamelen (zoals de temperatuur, het lichtniveau, de afstand, de versnelling enz.) die ertoe kunnen leiden tot bepaalde beslissingen om de prestaties ervan te optimaliseren. Deze projectgebaseerde leerinterventie zal zowel docenten als studenten helpen kennis te maken met de vakgebieden AI en IoT, door de lens van de 5 Grote Ideeën (namelijk Perceptie, Representatie en Redeneren, Leren, Natuurlijke Interactie en Maatschappelijke Impact). maar ook in het licht van de robotica door een reeks praktische en computergebaseerde praktijken en activiteiten te implementeren. Wat AI betreft, zullen ze kennis maken met spraakherkenning en hoe deze dienst kan worden gebruikt om door een robotartefact te navigeren. Daartoe zullen ze kennis maken met de processen van het ontwerpen en programmeren van een op AI gebaseerde applicatie met behulp van de MIT App Inventor-software, evenals met het trainen van een model om de binnenkomende informatie te classificeren met behulp van de Personal Audio Classificatie-omgeving. Wat robotica betreft, zullen ze leren hoe ze een robotauto kunnen bouwen met behulp van de BBC micro:bit- microcontroller en verschillende compatibele elektronische componenten, en hoe ze dit robotartefact kunnen programmeren met behulp van de op blokken gebaseerde software Makecode. Het project zal in 5 activiteiten worden verdeeld. Elk van deze activiteiten zal draaien rond een van de 5 Grote Ideeën. Tijdens deze activiteiten worden de studenten uitgenodigd om zich te concentreren op verschillende delen van het implementatieproces en zich bezig te houden met verschillende aspecten van AI en IoT. De activiteiten omvatten richtlijnen voor leerkrachten en verschillende voorgestelde taken voor leerlingen om te zorgen voor een soepele introductie en implementatie van alle bovengenoemde concepten en hun inherente aspecten, wat uiteindelijk zal leiden tot de verwerving en ontwikkeling van verschillende vaardigheden van de 21e eeuw, zoals creativiteit, kritisch denken, probleemoplossing en samenwerking.

### 1.1 De reikwijdte van het Project

Via dit project maken de studenten kennis met het vakgebied AI en de 5 Grote Ideeën, evenals het IoT-domein, in het licht van robotica. Met name door de 5 activiteiten en de bijbehorende taken, die draaien om de constructie en het programmeren van de robotauto, evenals het ontwerp en programmeren van de AI-toepassing, zullen de leerlingen de 5 Grote Ideeën beter begrijpen en zich verdiepen in enkele van de kernmechanismen van AI en IoT. Meer specifiek zullen de leerlingen in de eerste activiteit kennis maken met IoT door te leren hoe ze hun robotauto kunnen programmeren om een aantal omgevingsgegevens te verzamelen en te monitoren via een IoT-analyseplatformdienst (dwz ThingSpeak). Via de daarin opgenomen taken zullen ze kennis maken met het idee van perceptie. In de tweede activiteit maken leerlingen kennis met het idee van representatie en redeneren door te leren hoe de robotauto kan 'denken' en hoe gegevens op meerdere manieren kunnen worden weergegeven. Daartoe leren ze hoe ze beslisbomen en stroomdiagrammen kunnen maken, en deze informatie gebruiken om een applicatie te maken waarmee de robotauto kan worden genavigeerd met behulp van spraakopdrachten. Daarom zullen ze leren hoe ze de AI-service voor spraakherkenning kunnen gebruiken en de resultaten van de implementatie ervan kunnen evalueren. In de derde activiteit maken de leerlingen kennis met het idee van leren en, door gebruik te maken van een Machine Learning (ML)-tool (de Personal Audio Classifier), zullen ze vertrouwd raken met methoden die de robotauto kan leren uit data en via ml. In de vierde activiteit zullen

ze kennismaken met het idee van natuurlijke interactie, door het getrainde model (geproduceerd in de derde <sup>activiteit</sup>) te integreren in de gecreëerde applicatie en de resultaten te evalueren in het licht van een fysieke implementatie, waardoor ze zich bewust worden van de beperkingen van AI-systemen op het gebied van natuurlijke interactie. Ten slotte zullen ze in de vijfde <sup>activiteit</sup> kennismaken met het idee van maatschappelijke impact, door na te denken over hun hele ervaring en zich bewust te worden van de voordelen, nadelen en risico's achter het gebruik van AI en IoT in ons dagelijks leven. Het uiteindelijke doel van dit project is om het vertrouwen van leerlingen in alle bovengenoemde concepten en aspecten te vergroten en betekenisvolle leerervaringen te creëren die hen zullen helpen een aantal 21e-eeuwse vaardigheden te ontwikkelen, <sup>zoals</sup> creativiteit, kritisch denken, probleemoplossing en samenwerking.

## 1.2 De doelgroepen

Het project is bedoeld voor leerlingen tussen de 12 en 16 jaar oud. Studenten moeten enige ervaring hebben met blokgebaseerde programmeeromgevingen.

## 1.3 Het doel van dit document

Het doel van dit document is om docenten een aantal concrete ideeën en leeractiviteiten te bieden over hoe de concepten AI en IoT op betekenisvolle wijze kunnen worden geïntroduceerd en aan leerlingen kunnen worden onderwezen, door de lens van robotica en een reeks praktische taken.

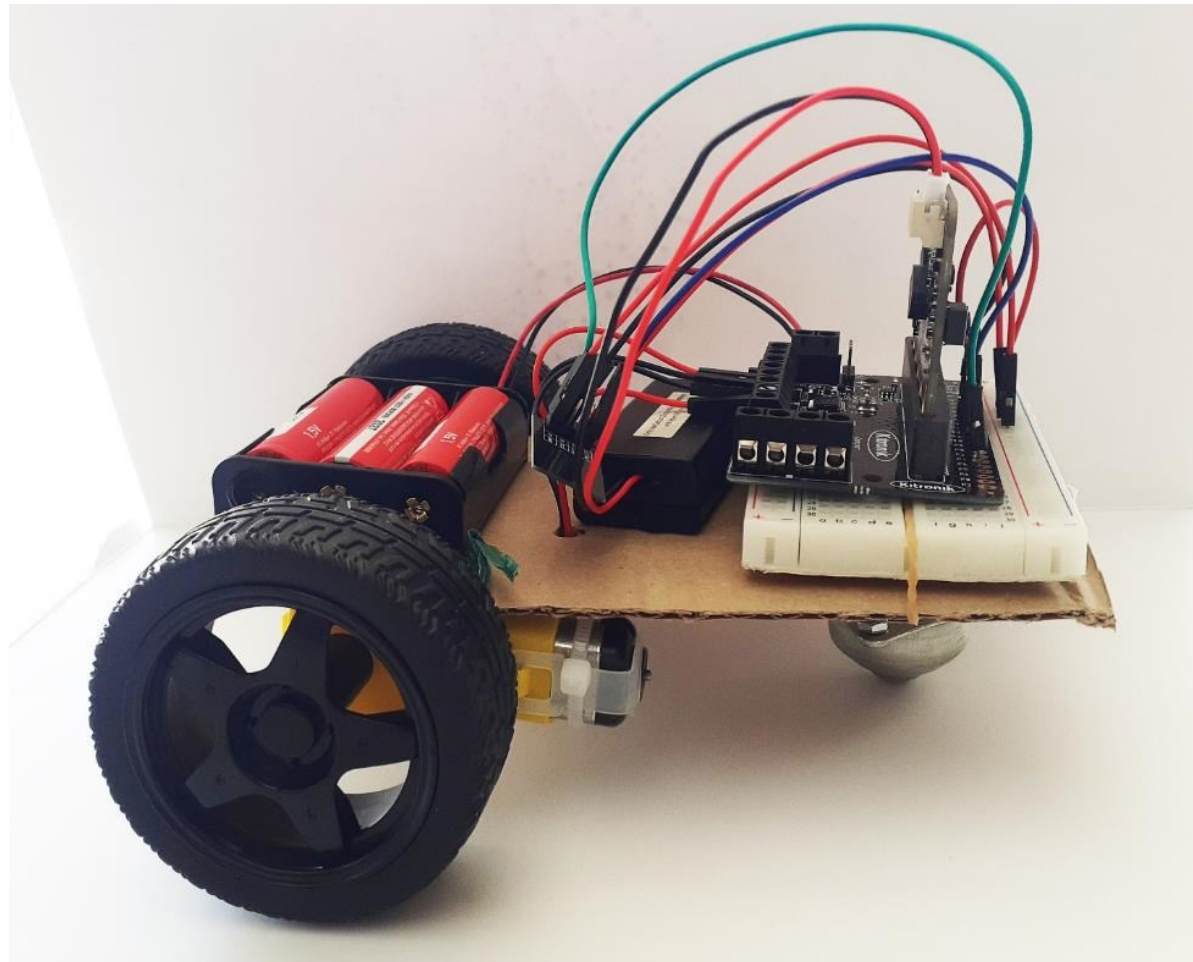
## 2. Woordenlijst van de eenheid

| Woord   | Definitie   |
|---|---|
| <b>DingSpreek</b>                                     | Een IoT-serviceplatform om gegevens te monitoren die zijn verzameld door de ESP8266 WiFi-module           |
| <b>MIT-app-uitvinder</b>                              | Software voor het maken van applicaties   |
| <b>Spraakherkenning</b>                               | Een AI-service in MIT App Inventor die spraak herkent en er een tekst van maakt                           |
| <b>Reageer-app of persoonlijke audioclassificator</b> | Een machine learning (ML) audioclassificatie en modeltrainingstool die compatibel is met MIT App Inventor |

### 3. Introductie van de “DIY-robotauto die kan worden bestuurd en bestuurd met behulp van spraakopdrachten”

#### 3.1 Beschrijving

Dit project zal leerlingen kennis laten maken met de 5 grote ideeën van AI en IoT in STEM-onderwijs door de creatie van een doe-het-zelf-robotauto ( *Figuur 1* ), geprogrammeerd om te worden genavigeerd met behulp van spraakopdrachten. Concreet zullen leerlingen worden aangemoedigd om de robotauto te ontwerpen en in elkaar te zetten door gebruik te maken van verschillende elektronische componenten en eenvoudige materialen, en om deze te programmeren met behulp van op blokken gebaseerde programmeeromgevingen. Ze zullen leren om in realtime gegevens te verzamelen en na te denken over de verkregen resultaten en hoe deze kunnen worden gebruikt om specifieke beslissingen te nemen over de prestaties van de robotauto. Daarnaast zullen ze worden aangemoedigd om een applicatie te ontwerpen en te programmeren die gebruik maakt van de Speech Recognition AI-service om verbale navigatie van de robotauto mogelijk te maken. Bovendien leren ze hoe ze een getraind model kunnen maken om de prestaties van de applicatie te optimaliseren.



*Figuur 1: De doe-het-zelf-robotauto*

Om door te gaan met de activiteiten in dit document, moet u eerst de instructies volgen die zijn opgenomen in het bestand “ *T2.4\_Creating\_the\_robotic\_car.pdf* ”, terwijl u uw leerlingen kunt aanmoedigen om enkele opwarmprogrammeringsactiviteiten uit te voeren, vergelijkbaar met die in het bestand “ *T2.4\_WarmUp\_programming\_activities\_for\_the\_robotic\_car.pdf* ”. Het wordt sterk aanbevolen dat u uw leerlingen in teams verdeelt en hen aanmoedigt om de verschillende aspecten van elke activiteit te bespreken. Om dit proces te vergemakkelijken, vindt u in dit document links naar materiaal dat u kan helpen elke stap van het project soepel te introduceren, evenals enkele voorgestelde vragen die u kunnen helpen een dialoog met uw leerlingen op gang te brengen over verschillende delen en aspecten van het project. het project.

## 3.2 Leerdoelen & resultaten

Na succesvolle afronding van dit onderdeel moeten de leerlingen in staat zijn om:

- Identificeer de aanwezigheid en het gebruik van AI-robotica in het dagelijks leven
- Bespreek en begrijp de rol van AI in robotica
- Verschillende aspecten van de integratie van AI in een robotproject door middel van spraakherkenning en spraakopdrachten uitleggen en bespreken
- Begrijp hoe AI-services zoals spraakherkenning werken
- Begrijp hoe beslissingsbomen of stroomdiagrammen mogelijke logische paden kunnen tonen en ook kunnen leiden tot beslissingen over operatoren die in een later stadium kunnen worden gebruikt
- Bespreek de rol van Machine Learning (ML) in AI-robotica en hoe ML kan worden gebruikt om een robotartefact te trainen om zijn omgeving waar te nemen
- Leg de basisconcepten van audioclassificatie uit
- Identificeer en bespreek de voordelen en risico's van het implementeren van spraakopdrachten tijdens het autorijden
- Identificeer en bespreek de voordelen en risico's van audioclassificatie
- Basisprogrammeerconstructies/concepten uitleggen die verband houden met de implementatie van spraak-naar-tekstmethoden
- Begrijp de belangrijkste concepten die ten grondslag liggen aan IoT en de implicaties van monitoringgegevens
- Denk na over de impact van datagestuurde beslissingen in het dagelijks leven
- Begrijp dat services zoals Spraakherkenning foutgevoelig zijn

Studenten zullen ook leren om:

- Construeer een robotartefact en maak circuits als onderdeel van een robotconstructie
- Bewaak gegevens met behulp van een IoT-analyseplatformservice
- Maak beslissingsbomen en stroomdiagrammen om een type informatie weer te geven
- Gebruik programmeeropdrachten in combinatie met AI-methoden om specifiek gedrag op een robotartefact aan te pakken
- Programmeer een robot die instructies moet krijgen met behulp van spraakopdrachten
- Geef uitdrukking aan hun ideeën door middel van programmeren
- Gebruik de Personal Audio Classifier ML-tool om verschillende geluiden te classificeren
- Evalueer de resultaten van een ML-tool
- Verbeteringen aanbrengen in een getraind model op basis van evaluatie

- Onderzoek factoren in de trainingsgegevens die tot vertekening kunnen leiden

### 3.3 Geschatte duur van de Eenheid

Dit is een vrij uitgebreid project, waarbij meerdere uren nodig zijn om alle aspecten goed aan te pakken. De volgende duur is indicatief en kan variëren afhankelijk van de leeftijd en het niveau van uw leerlingen.

**Activiteit 1:** 4 – 6 uur

**Activiteit 2:** 8 – 15 uur

**Activiteit 3:** 2 – 4 uur

**Activiteit 4:** 2 – 4 uur

**Activiteit 5:** 1 – 2 uur

## 3.4 Activiteit 1 – Introductie van het grote idee van perceptie via IoT:

### 3.4.1 Beschrijving

Deze activiteit is een leerinterventie als opwarmer om leerlingen kennis te laten maken met het idee van perceptie in het licht van IoT en robotica. Ze zullen onderzoeken hoe hun robotauto zijn omgeving kan waarnemen en waarnemen door gegevens te verzamelen en op te slaan, die later kunnen worden gebruikt om beslissingen te nemen over de prestaties van de auto. In het bijzonder zullen ze onderzoeken hoe sensoren die in het BBC micro:bit- bord zijn ingebouwd, of sensoren die op het bord kunnen worden aangesloten (bijvoorbeeld een ultrasone sensor), gegevens kunnen verzamelen en kunnen worden gemonitord met behulp van een IoT-analyseplatformdienst. De dienst die wordt gebruikt voor de behoeften van deze activiteit is ThingSpeak, een platform dat aggregatie, visualisatie en analyse van live datastromen in de cloud mogelijk maakt.

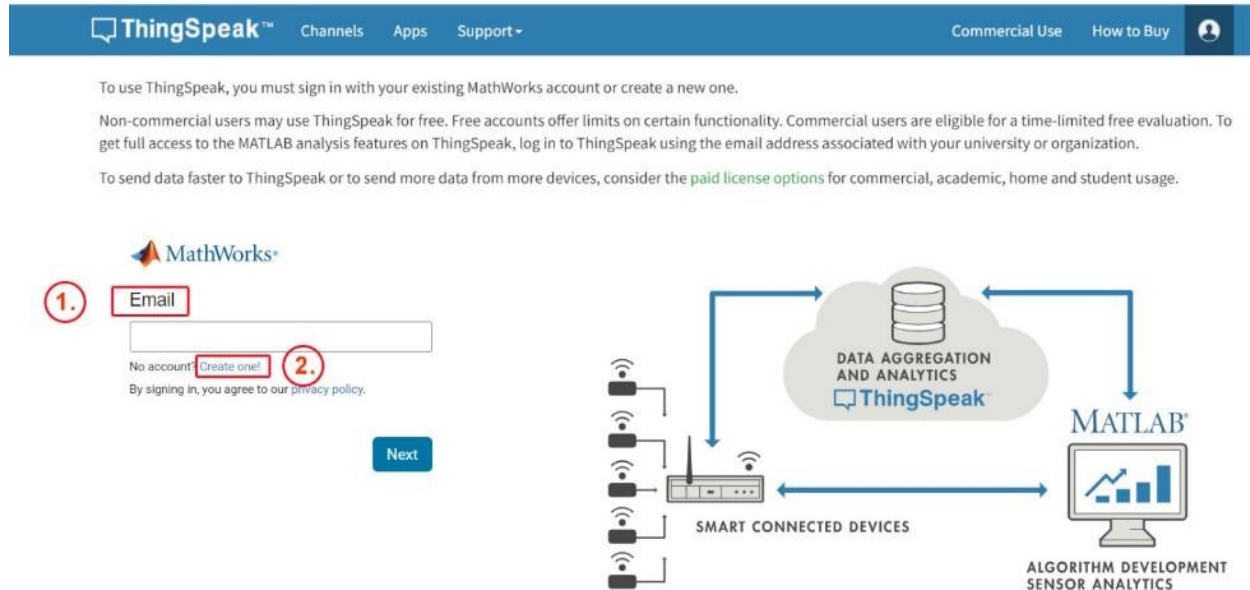
**Tip:** Het begrijpen van een dergelijke omgeving (dat wil zeggen ThingSpeak) kan een beetje uitdagend zijn voor uw leerlingen. Daarom kun je, afhankelijk van hun niveau, besluiten om deze activiteit na afloop van de activiteiten 2 tot en met 4 te introduceren, om zo beter te beseffen hoe IoT kan helpen de robotauto te optimaliseren. Als uw leerlingen jonger zijn dan 12 jaar, kunt u deze activiteit overslaan.

### 3.4.2 Hardware

Naast de BBC micro:bit is de voor deze activiteit benodigde hardware de ESP8266 Wi-Fi-module en optioneel enkele andere sensoren zoals een ultrasone sensor (bij voorkeur de HC-SR04).

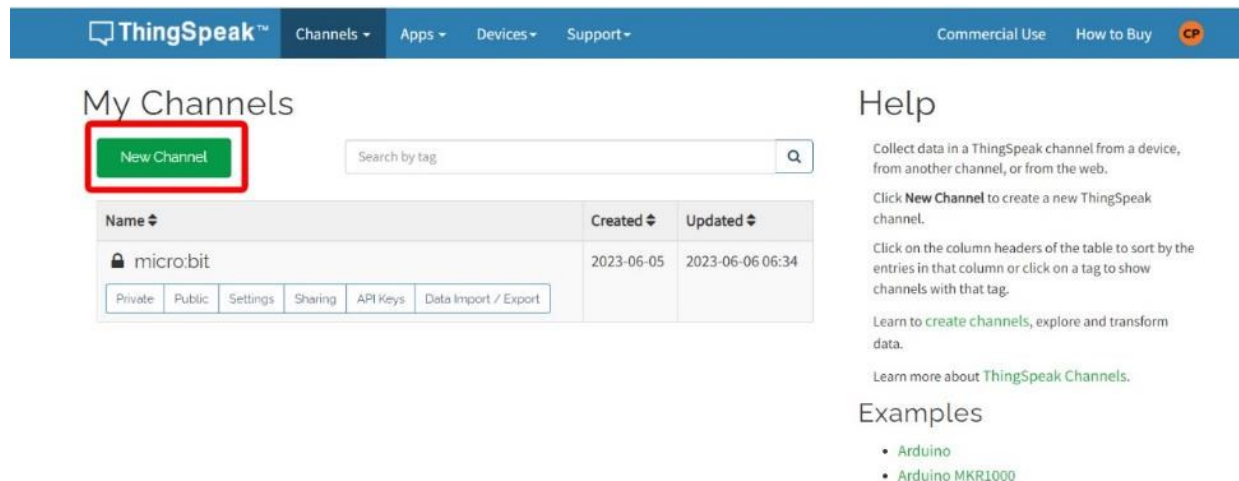
### 3.4.3 Installatie

Voordat u begint met bedrading en coderen, moet u een kanaal maken om de gegevens die u ontvangt te controleren. Om dit te doen, gaat u naar de ThingSpeak-website ( <https://thingspeak.com/> ) en logt u in op uw account door uw e-mailadres in te voeren ( **1** ), of maakt u een account aan door op “Create One” te klikken ( **2** ) optie ( *Figuur 2* ).



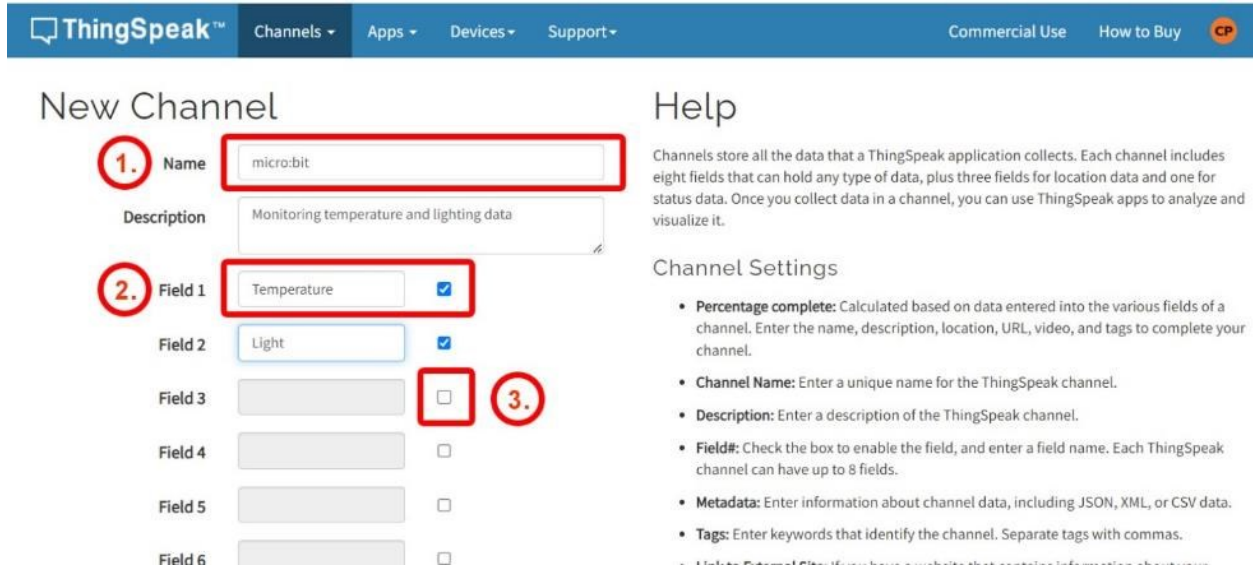
Figuur 2: Menu voor inloggen of het aanmaken van een nieuw account bij ThingSpeak

Ga vervolgens naar het menu Kanalen en klik op de knop “ **Nieuw kanaal** ” onder “Mijn kanalen” om een nieuw kanaal te maken om gegevens van uw sensoren op te nemen en weer te geven ( *Figuur 3* ).



Figuur 3: De knop Nieuw kanaal, gelegen in het menu Kanalen

De volgende stap is het creëren van uw nieuwe kanaal door de naam ( **1.** ) van uw kanaal (bijvoorbeeld microbit) en de parameters die u wilt monitoren in de daarvoor bestemde velden in te voeren ( **2.** ) ( *Figuur 4* ). In *Figuur 3* heeft de gebruiker bijvoorbeeld Temperatuur in Veld 1 en Licht in Veld 2 ingevoegd. U kunt meer velden activeren door op het selectievakje ( **3.** ) naast een Veld te klikken. Optioneel kun je een beschrijving schrijven, zodat je kunt onthouden welke parameters je kanaal controleert.



**ThingSpeak™** Channels Apps Devices Support Commercial Use How to Buy CP

## New Channel

**1.** Name: micro:bit

Description: Monitoring temperature and lighting data

**2.** Field 1: Temperature ☒

Field 2: Light ☒

Field 3: ☐ **3.**

Field 4: ☐

Field 5: ☐

Field 6: ☐

## Help

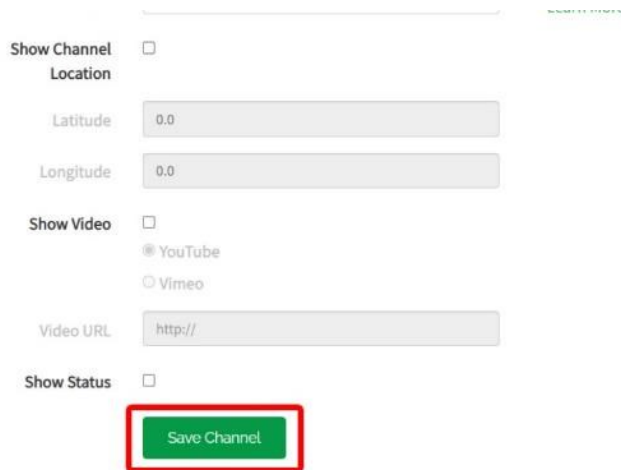
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

### Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.

Figuur 4: Een nieuw kanaal creëren

Zodra u alle parameters heeft opgegeven die u wilt monitoren, bladert u naar beneden op de pagina Nieuw kanaal en klikt u op de knop Kanaal opslaan ( Afbeelding 5 ) om alle ingevoegde informatie op te slaan.



Show Channel ☐

Location

Latitude: 0.0

Longitude: 0.0

Show Video ☐

☒ YouTube

☐ Vimeo

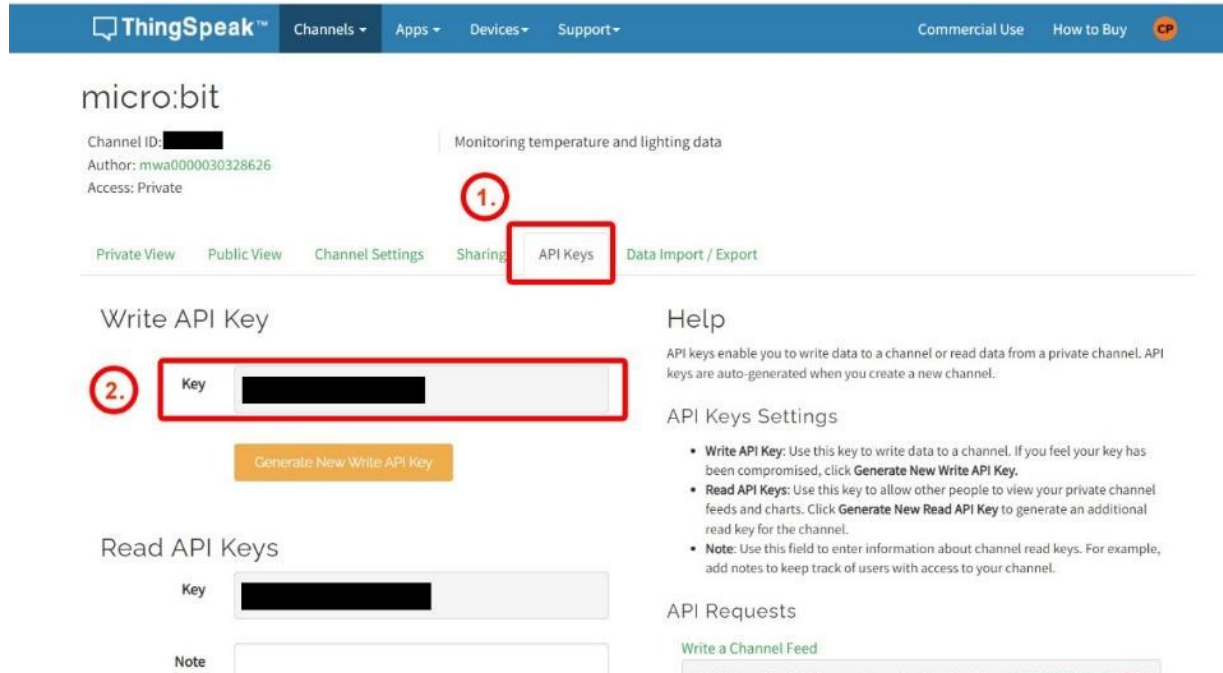
Video URL: http://

Show Status ☐

**Save Channel**

Figuur 5: Scroll naar beneden op de pagina Nieuw kanaal en klik op de knop Kanaal opslaan

Nadat u op Kanaal opslaan heeft gedrukt, wordt u automatisch doorgestuurd naar uw nieuwe kanaal. Klik op het menu API Keys ( **1.** ) en noteer de sleutel ( **2.** ) die verschijnt in het veld API Key schrijven ( **Figuur 6** ).



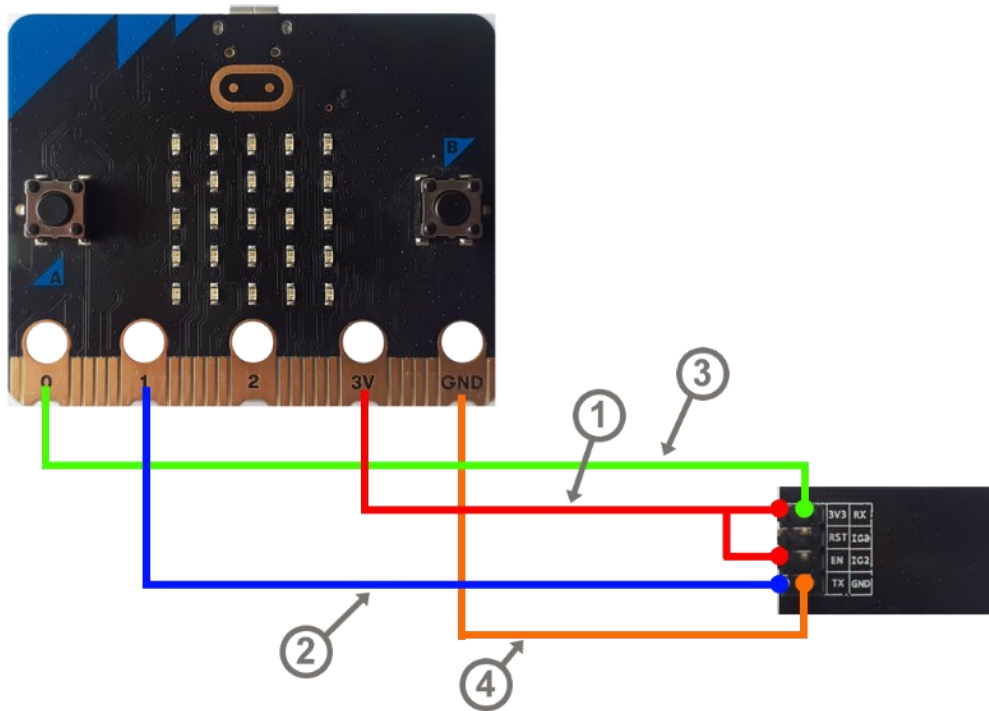
Figuur 6: De API-sleutel vinden

Deze sleutel wordt vervolgens ingevoegd in een van de opdrachtblokken die u gaat gebruiken in de Makecode-programmeeromgeving.

Nu u uw kanaal op de ThingSpeak-site heeft geregistreerd, kunt u doorgaan met de bedradings- en programmeerfase van de ESP8266 Wi-Fi-module.

#### 3.4.3.1 Bedrading

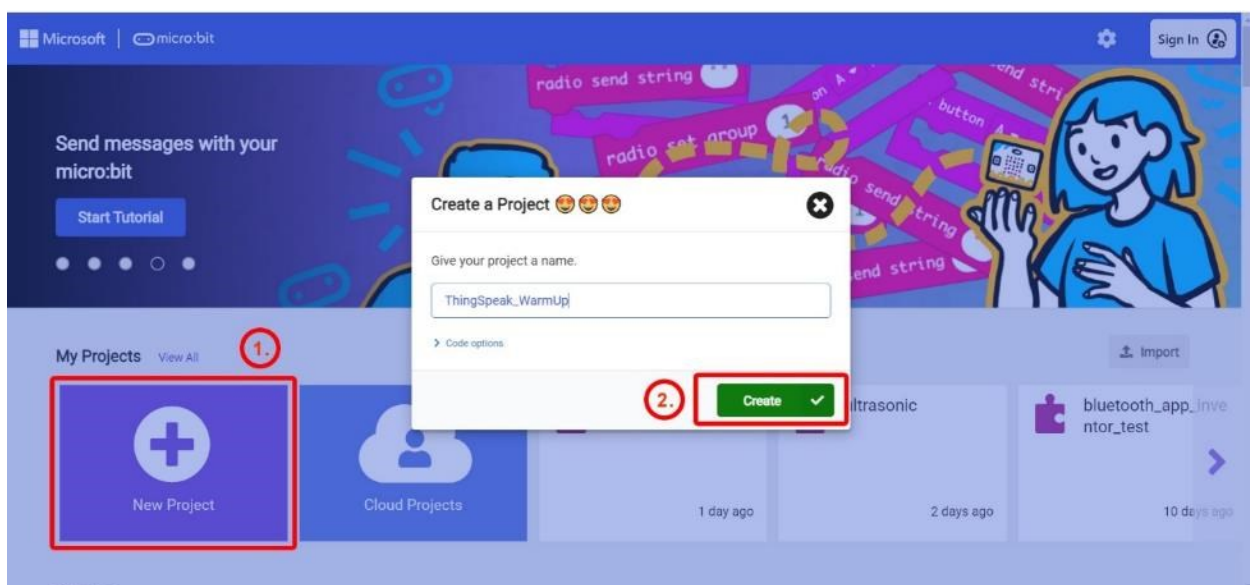
Figuur 7 laat zien hoe u de ESP8266 Wi-Fi-module aansluit op de BBC micro:bit- microcontroller. De ESP8266-module heeft 8 verschillende pinnen. Je moet er 5 op het micro:bit- bord aansluiten. In het bijzonder moet u de 3V3- en EN-pinnen ( 1 ) aansluiten op de voeding (3V), en de GND-pin ( 4 ) op de GND-pin van de micro:bit . Sluit ten slotte de TX-pin ( 2 ) en de RX-pin ( 3 ) aan op respectievelijk pin P1 en P0. Zodra het circuit voltooid is, sluit u de microbit aan op uw computer met behulp van een USB-kabel.



Figuur 7: ESP8266 WiFi-module aansluiten op de BBC micro:bit- microcontroller

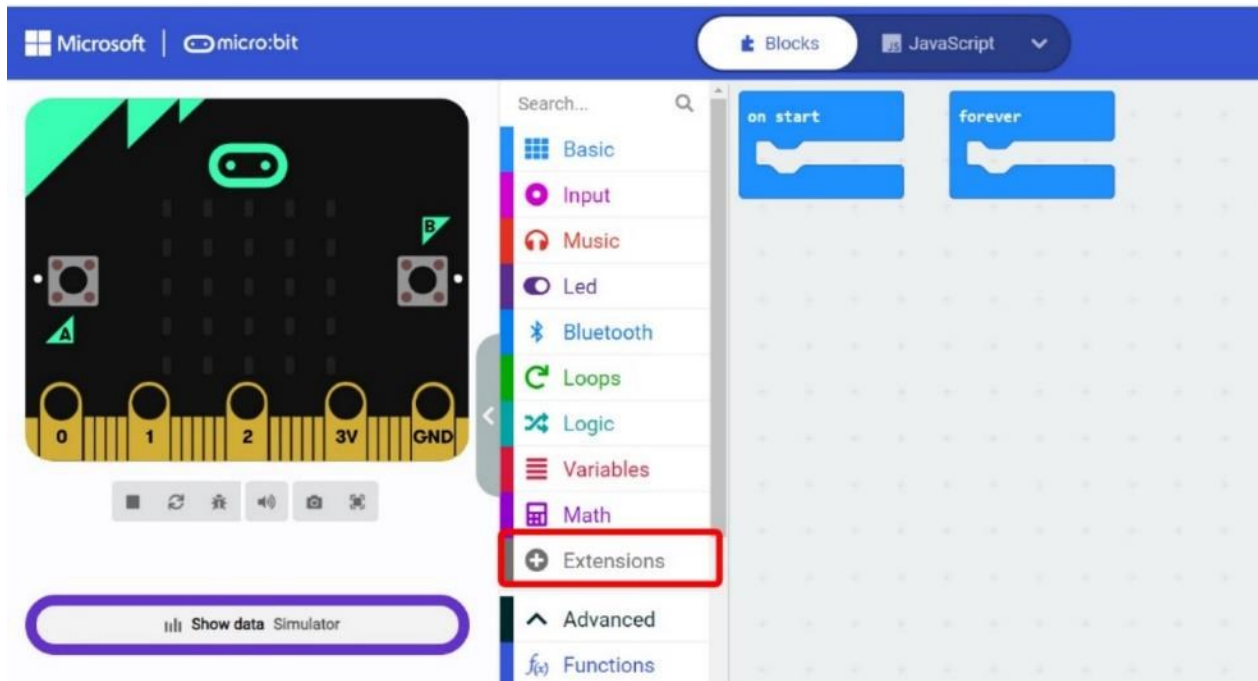
### 3.4.3.2 Code

De volgende stap is het openen van de Microsoft Makecode-blokgebaseerde programmeeromgeving ( <https://makecode.microbit.org/#> ) en het maken van een nieuw project. Klik op de startpagina van Makecode op het tabblad Nieuw project ( 1. ) en voer in het pop-upmenu een naam in voor uw project (bijvoorbeeld ThingSpeak\_WarmUp). Klik vervolgens op de knop Maken ( 2. ) ( Figuur 8 ).



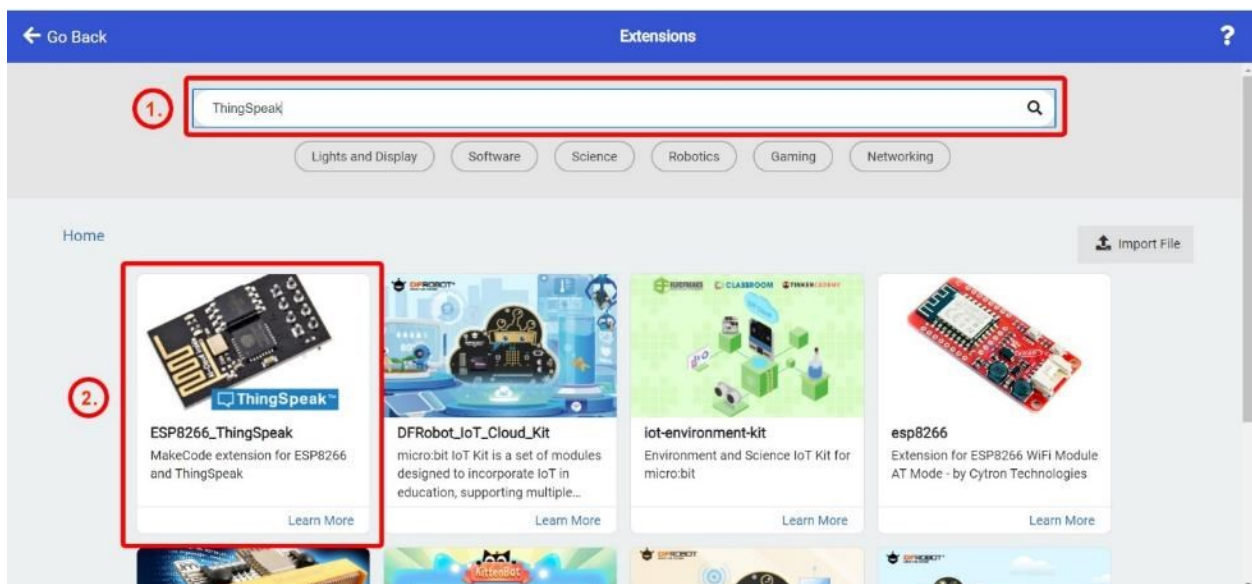
Figuur 8: Een nieuw project maken in de MakeCode-programmeeromgeving

U wordt automatisch doorgestuurd naar het gebied waar u uw code kunt samenstellen. Om de blokken voor het programmeren van de ESP8266 Wi-Fi-module toe te voegen, klikt u op het menu +Extensies (Figuur 9).



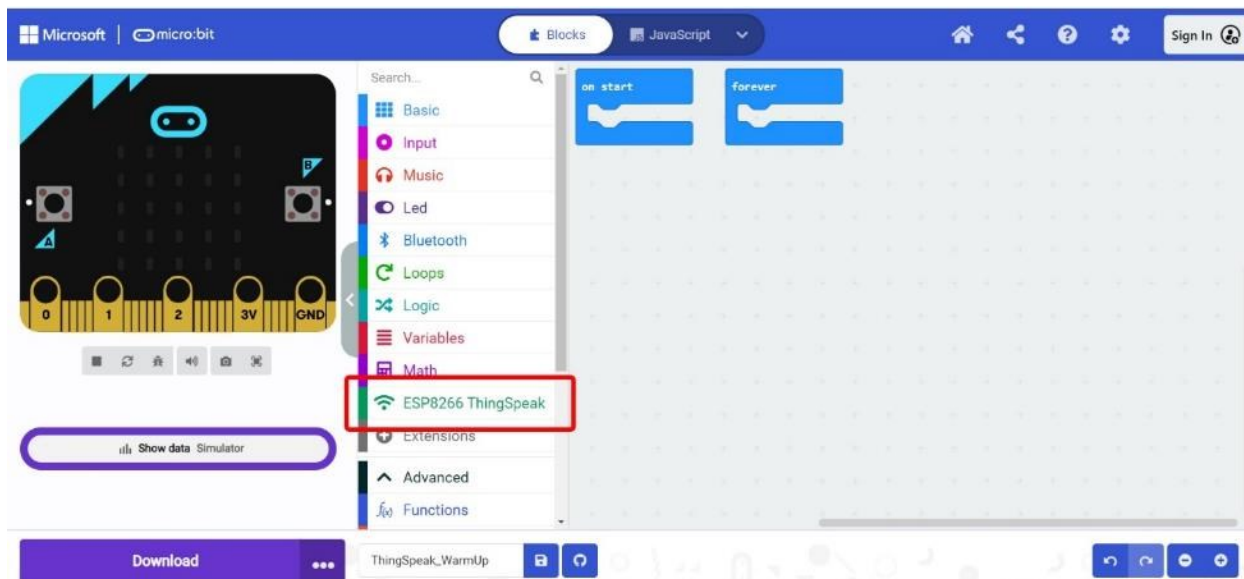
Figuur 9: Het menu +Extensies

Typ "ThingSpeak" in het veld "Zoek of voer project-URL in" (1.) en selecteer de overeenkomstige extensie (2.) (Figuur 10).



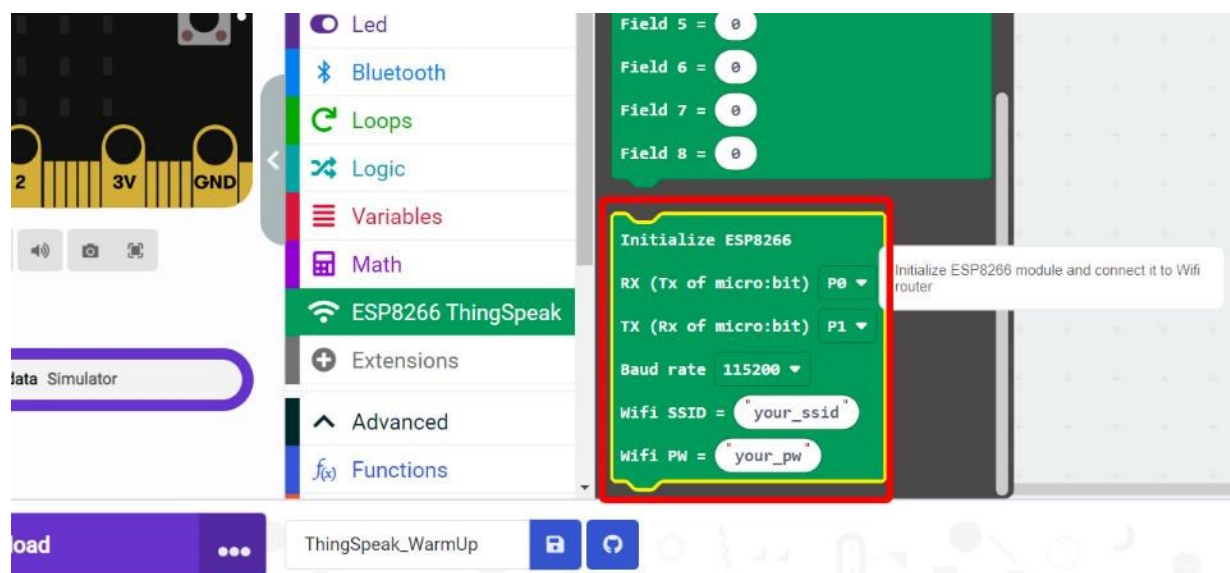
Figuur 10: De ESP8266\_ThingSpeak-extensie zoeken en selecteren

Een nieuw menu met blokken (dwz ESP8266 ThingSpeak) zal verschijnen in het opdrachtmenu ( *Figuur 11* ).



Figuur 11: Het nieuwe menu met commandoblok voor het programmeren van de ESP8266 WiFi-module

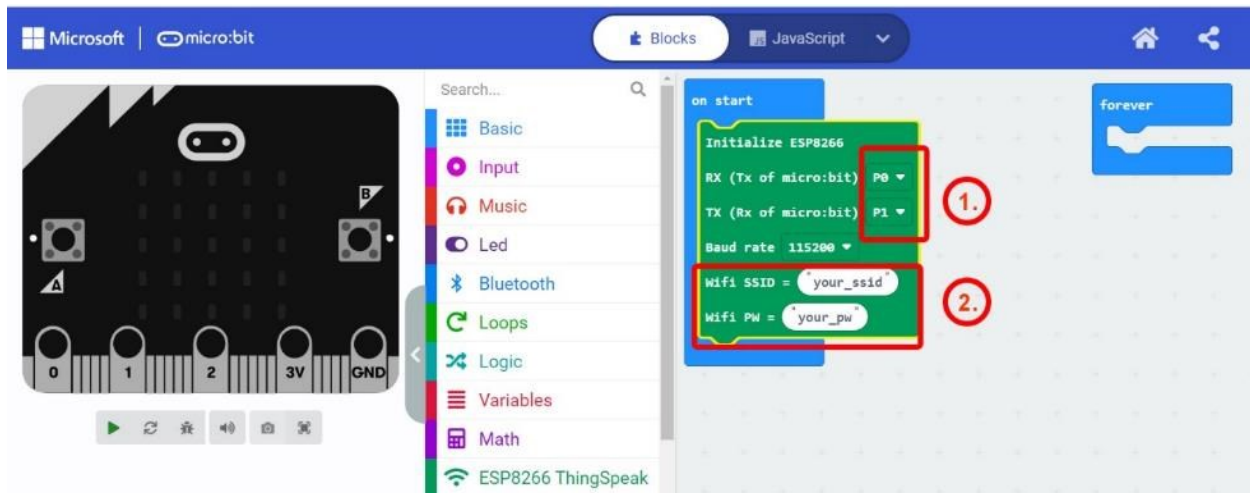
Eerst moet u de Wi-Fi-module initialiseren door de pinnen waarmee deze is verbonden aan te geven en informatie in te voeren van de Wi-Fi-router waarmee u verbinding gaat maken. Om dit te doen, klikt u op het ESP8266 ThingSpeak-menu en selecteert u **ESP8266 initialiseren** blokopdracht vanuit het zwevende menu ( *Figuur 12* ).



Figuur 12: Zoeken en selecteren van de opdracht ESP8266-blok initialiseren

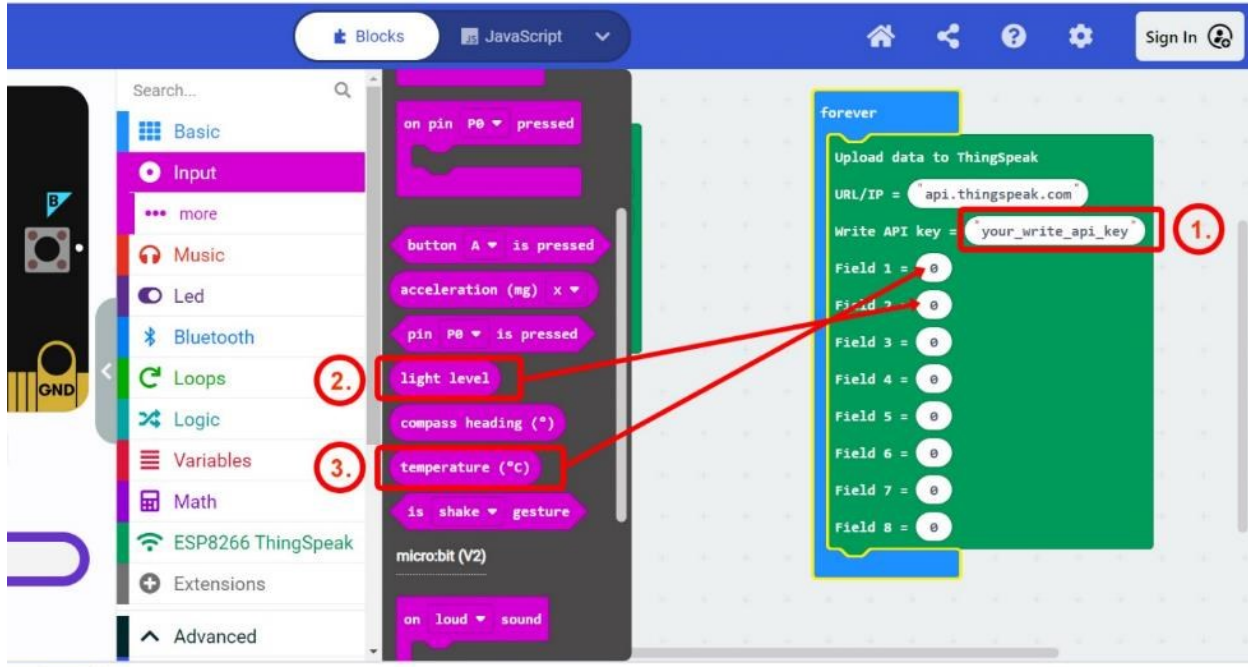
Sleep deze opdracht naar het blok 'aan start'. Geef in de RX- en TX-velden ( 1. ) de poorten op waarop u de overeenkomstige pinnen van de module (namelijk P0 en P1) hebt aangesloten ( *Figuur 13* ). Voer

vervolgens in de velden wifi SSID en wifiPW ( 2. ) de naam van uw wifi-router en het bijbehorende wachtwoord in. Hierdoor wordt de Wi-Fi-module met uw Wi-Fi-router verbonden.



Figuur 13: De pinnen aangeven dat ESP8266 is verbonden en informatie invoegen over de wifi-verbinding

De volgende stap is om de Wi-Fi-module in staat te stellen de gegevens die hij van microbit ontvangt, te uploaden naar het ThingSpeak-platform. Om dit te doen, sleept u de blokopdracht “**gegevens uploaden naar ThingSpeak**” van het **ESP8266 ThingSpeak-menu** naar het blok “**voor altijd**”. In het veld Write API key ( 1. ) ( Afbeelding 14 ), typt u de Write API key die u heeft ontvangen nadat u uw kanaal op het ThingSpeak-platform hebt aangemaakt ( Afbeelding 6 ). Om vervolgens de gegevens aan te geven die door het ThingSpeak-platform worden gemonitord (dwz temperatuur en licht), gaat u naar het menu “**Invoer**” en sleept u de “**temperatuur**” ( 3. ) en het “**lichtniveau**” ( 2 ) voer opdrachten in voor respectievelijk Veld 1 en Veld 2 ( Afbeelding 14 ).



Figuur 14: De API-sleutel invoegen en de parameters toevoegen die zullen worden gemonitord

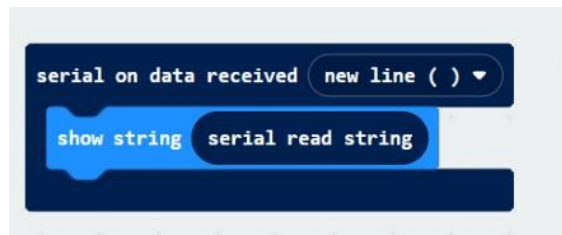
Daarna kunt u het script naar uw micro:bit downloaden door op de downloadknop te klikken.



Figuur 15: De knop Downloaden

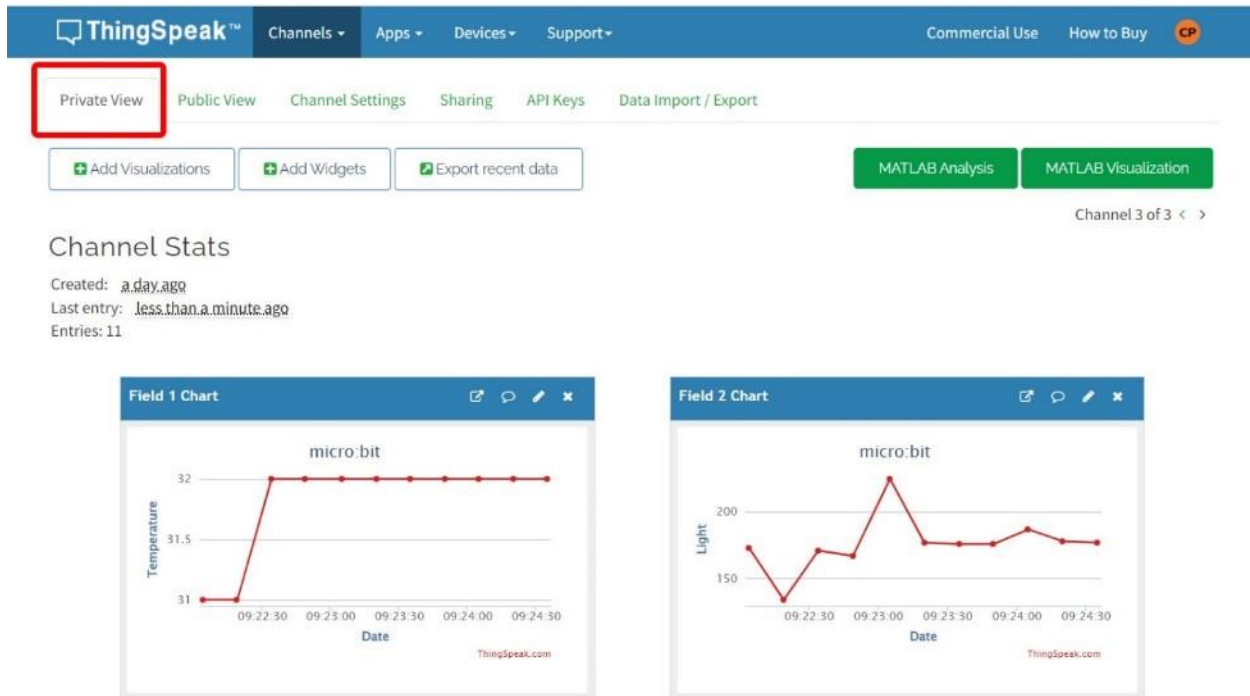
**Opmerking:** Houd er rekening mee dat u mogelijk een pauzeopdracht van 15 seconden moet toevoegen binnen de forever loop-opdracht, om een soepele overdracht van gegevens naar het ThingSpeak-platform mogelijk te maken.

Optioneel kunt u het script uit Figuur 16 aan uw code toevoegen om een deel van de ontvangen gegevens op het LED-scherm van uw microbit weer te geven. Deze commando's (dat wil zeggen: " **serieel bij ontvangen gegevens ...** " " **seriële leesreeks** ") bevinden zich in het menu **Seriële** opdrachten.



Figuur 16: Weergave van de ontvangen gegevens op het LED-scherm van microbit

Nadat u de code heeft gedownload, gaat u naar het ThingSpeak-platform en gebruikt u het tabblad Privéweergave om te zien hoe de door microbit ontvangen gegevens in de loop van de tijd veranderen (Figuur 17).



Figuur 17: Controleren hoe de ontvangen gegevens in de loop van de tijd veranderen

### 3.4.4 Experiment 1

Herhaal het bovengenoemde proces met uw leerlingen. Maak eerst enkele accounts aan op het ThingSpeak-platform en deel deze met uw studenten. Verdeel ze in teams van 2 of 3 en geef elk team een micro:bit- bord en een ESP8266 Wi-Fi-module. Moedig uw leerlingen aan om naar de Wi-Fi-module te kijken om de ingebouwde pinnen te verkennen, en vraag vervolgens elk team om de module op het micro:bit- bord aan te sluiten. Om dit proces te vergemakkelijken, kunt u hen de circuitkaart geven die zich in het bestand "Circuit\_card\_Arctivity1.pdf" bevindt.

Doorloop alle teams om er zeker van te zijn dat ze allemaal met succes het circuit hebben gecreëerd. Ga vervolgens verder met het programmeerproces. Geef elk team een van de accounts die u op het ThingSpeak-platform hebt aangemaakt, en vraag hen om in te loggen. Moedig hen aan om naar de verschillende tabbladen te kijken (dwz Privéweergave, API-sleutels enz.). Geef vervolgens elk team een kopie van het bestand "Half\_baked\_Activity1.pdf" en vraag hen om het script te maken om de Wi-Fi-module te programmeren om de temperatuur en het lichtniveau te controleren die door de micro:bit zijn geregistreerd. Vraag hen om het script naar hun micro:bit te downloaden en adviseer hen om de gegevens te observeren die op het ThingSpeak-platform worden gemonitord. Moedig ze aan een zaklamp te gebruiken en/of het circuit dicht bij koudere of warmere oppervlakken te plaatsen om te observeren hoe de bewaakte gegevens dynamisch veranderen.

Bespreek vervolgens enkele van de volgende kwesties met hen:

- Welke informatie wordt eruit gehaald?
- Heeft deze informatie invloed op de prestaties van de robotauto?
- Kan deze informatie leiden tot datagestuurde beslissingen om de prestaties van de robotauto te optimaliseren?
- Bedenk andere parameters die kunnen worden vastgelegd. Wat voor soort sensoren moet je op de robotauto aansluiten om deze parameters te registreren? Kunnen al deze parameters leiden tot significante datagestuurde beslissingen voor de prestaties van de robotauto?
- Denk na over AI-diensten die van dergelijke gegevens zouden kunnen profiteren. Hoe kunnen deze diensten deze parameters gebruiken om de informatie achter een dataset te extraheren en zo de prestaties van de robotauto te optimaliseren?

Door deze activiteit en de volgende bespreking zullen de leerlingen begrijpen:

- Hoe sensoren niet alleen kunnen worden gebruikt om een omgeving waar te nemen, maar ook om datagestuurde beslissingen te nemen.
- Dat de perceptie van een robotartefact gebaseerd is op verzamelde gegevens
- Het belang van het monitoren en evalueren van gegevens om optimale beslissingen te nemen voor de prestaties van een autonoom voertuig
- De gemonitorde gegevens kunnen door AI-diensten worden gebruikt om hun prestaties te optimaliseren.

## 3.5 Activiteit 2: Introductie van het idee van representatie en redeneren

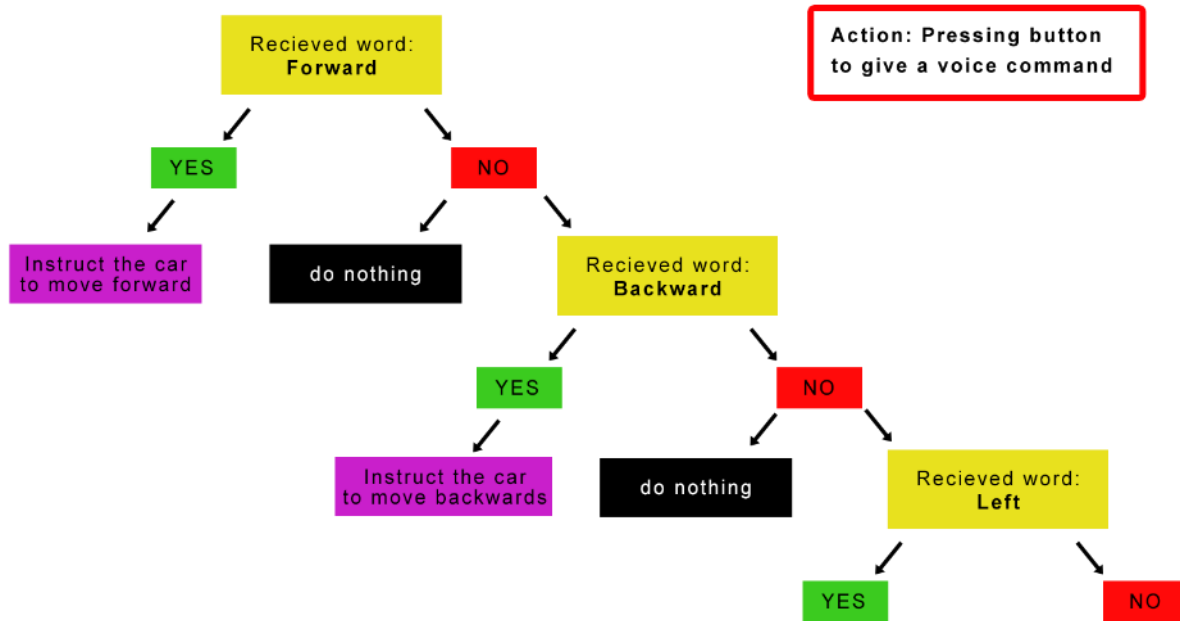
### 3.5.1 Beschrijving

Deze activiteit heeft tot doel leerlingen kennis te laten maken met het <sup>tweede</sup> grote idee, namelijk het idee van representatie en redeneren, en hen vertrouwd te maken met de manier waarop een robotartefact van gegevens kan 'leren'. Tijdens deze activiteit moeten de leerlingen in gedachten houden dat AI en intelligente agenten kunnen 'denken': **a)** door representaties van de wereld te construeren in de vorm van datastructuren, en **b)** door algoritmen te gebruiken om hen te helpen deze gegevens te begrijpen. structuren (redeneren). Daarom leren studenten hoe datastructuren kunnen worden weergegeven en hoe algoritmen kunnen worden gebruikt om specifieke informatie uit de representaties te extraheren. Daartoe zullen zij worden aangemoedigd om een aantal beslisbomen te creëren. Door dit proces zullen ze de criteria begrijpen voor het selecteren van een algoritme dat tot de best mogelijke oplossing zal leiden.

In het bijzonder zullen studenten leren hoe ze een applicatie kunnen programmeren waarmee de gebruiker door middel van spraakopdrachten door de robotauto kan navigeren. Hiervoor maken ze gebruik van de dienst Speech Recognition AI en maken ze kennis met de App Inventor-omgeving. Voordat ze de applicatie ontwikkelen, zullen ze een beslissingsboom maken die de beslissing weergeeft die hun robotauto moet nemen wanneer een stemcommando wordt ontvangen. Vervolgens beslissen ze welk logisch pad ze moeten volgen bij het programmeren van hun applicatie.

### 3.5.2 Een beslisboom maken

Om te beginnen met het maken van een beslisboom, moet je nadenken over hoe de AI-dienst zal worden geïmplementeerd en welke mogelijke acties kunnen worden geactiveerd. Het belangrijkste doel is om een applicatie te maken met een knop die spraakopdrachten opneemt wanneer erop wordt gedrukt. Op basis hiervan kunt u uw boom starten door ervan uit te gaan dat het ontvangen commando het woord "vooruit" bevat. Als dat zo is, moet de robotauto de opdracht krijgen om vooruit te rijden. Als er geen "vooruit" in staat, zijn er minstens twee verschillende logische paden: **a)** Het commando wordt helemaal niet herkend, dus de robotauto zal niet bewegen (of hij zal doorgaan met het uitvoeren van het vorige commando), **b)** het ontvangen commando bevat een ander richtingssleutelwoord, dus de robotauto zal dienovereenkomstig worden geïnstrueerd. Figuur 18 toont een indicatief voorbeeld van een dergelijke beslisboom. De boom is niet compleet. Er kunnen meer vertakkingen worden toegevoegd op basis van de verschillende ontvangen opdrachten.



Figuur 18: Een indicatieve beslissingsboom die de mogelijke acties weergeeft die kunnen worden geactiveerd wanneer de knop 'Druk om te spreken' wordt ingedrukt

Op basis van de beslisboom kunnen de leerlingen beslissen welke logische paden gebruikt moeten worden bij het programmeren. De beslissingsboom in Figuur 18 suggereert dat een reeks “if...then”-opdrachten moet worden gebruikt om de applicatie succesvol te programmeren.

### 3.5.3 Ontwerpen en programmeren van de applicatie

Na het analyseren van de mogelijke acties die kunnen worden geactiveerd, is de volgende stap het maken van de applicatie die de dienst Spraakherkenner AI gebruikt, zodat de gebruiker de robotauto kan besturen met behulp van spraakopdrachten. Voordat u begint met het maken van de applicatie, wordt het ten eerste aanbevolen dat u samen met uw leerlingen de opwarmactiviteit uit het bestand “2.4\_App\_Inventor\_Warm\_Up.pdf” uitvoert, en verder werkt aan hetzelfde “.aia”-project. Als alternatief kunt u het projectbestand “Remote\_control\_Microbit.aia” openen en eraan werken. In dat geval wordt u ten eerste aangeraden het project onder een nieuwe naam op te slaan.

**Opmerking:** als u het bovengenoemde .aia-bestand gebruikt, moet u er in plaats van de App Inventor-opwarmactiviteit ook aan te denken uw leerlingen te adviseren het script te downloaden dat wordt beschreven in het bestand “T2.4\_Programming\_the\_robotic\_car.pdf” naar de robotauto. .

#### Ontwerpproces

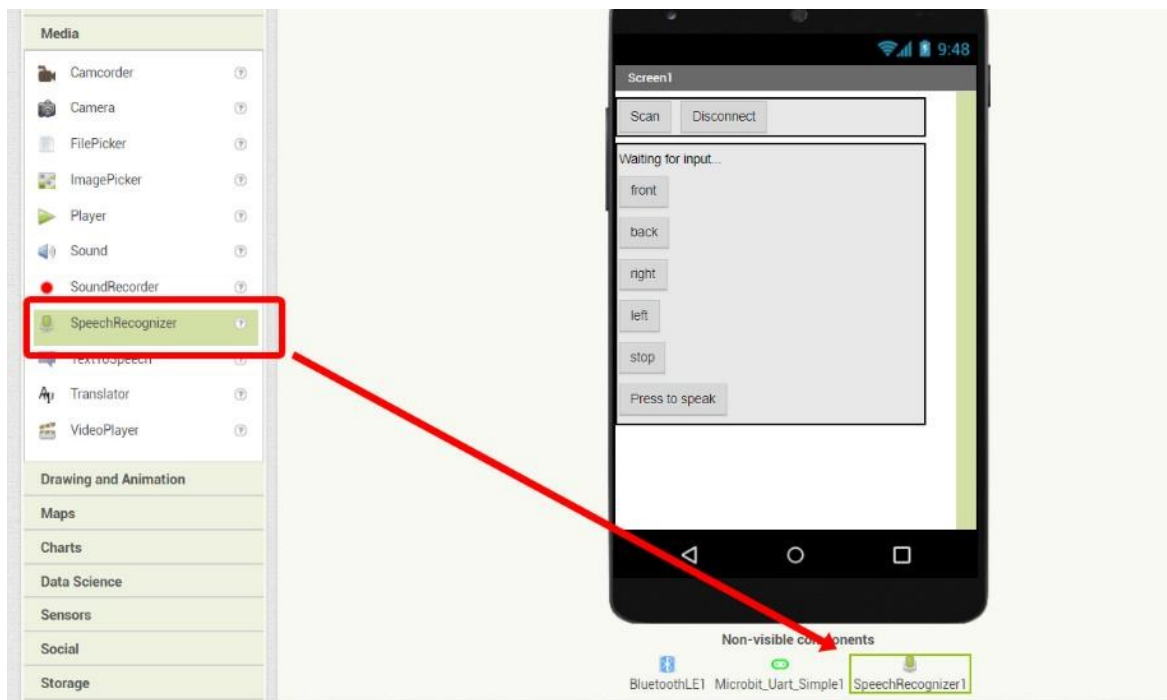
Eerst moet u een nieuwe knop toevoegen die de Spraakherkenning AI-service activeert. Sleep een knopitem en zet het neer onder de navigatieknoppen ( 1 ), hernoem het naar btn\_speak ( 2 ), terwijl de tekst ervan wordt gewijzigd in “Druk om te spreken” ( 3 ) ( Figuur 19 ).



Figuur 19: Een nieuwe knop toevoegen (1) en hernoemen naar btn\_speak (2), terwijl de tekst wordt gewijzigd in Druk om te spreken (3)

Wanneer u op de knop “ **Druk om te spreken**” drukt, begint de applicatie uw stem te detecteren. Vervolgens gebruikt het standaard de spraak-naar-tekst AI-service om de ontvangen spraakopdrachten om te zetten in tekst.

Om deze service te gebruiken, moet u de component “SpeechRecogniser” toevoegen. Zoek het op het tabblad “Media” en sleep het naar het scherm ( Figuur 20 ).



Figuur 20: De component SpeechRecognizer slepen en neerzetten naar het ontwerpgebied

**Opmerking:** Net als de componenten Bluetooth LE en Microbit\_Uart\_Simple is de SpeechRecognizer een niet-zichtbare component en verschijnt daarom niet op het scherm, maar in het gebied onder het schermvoorbeeld.

## Programmeerproces

Deze programmeeroplossing zal uw applicatie instrueren om een stemcommando te herkennen, telkens wanneer de “Press to Speak”-knop wordt ingedrukt.

Ga naar het menu Blokkeren en selecteer de component “btn\_speak”. Selecteer in het zwevende menu de **"wanneer btn\_speak.Click do"** gebeurtenishandler en sleep deze naar het gebied voor het samenstellen van het script. Klik vervolgens op de component Spraakherkenner en selecteer in het zwevende menu de **"roep SpeechRecognizer.GetText"** opdracht en plaats deze in de handlergebeurtenis.



Dit blok programmeert de knop btn\_speak om **"de tekst op te halen"** die SpeechRecognizer detecteert, elke keer dat de knop “Druk om te spreken” wordt ingedrukt. Tegelijkertijd wordt de spraak-naar-tekst AI-service geactiveerd en wordt de stem van de gebruiker omgezet in tekst, waardoor de robotauto de geïnstrueerde beweging kan uitvoeren (dwz vooruit rijden als het ontvangen stemcommando “voor” is, enz.)

**Let op:** de microfoon op het smartapparaat van de gebruiker wordt gebruikt voor het vastleggen en opnemen van spraak.

## Reageren op stemcommando's

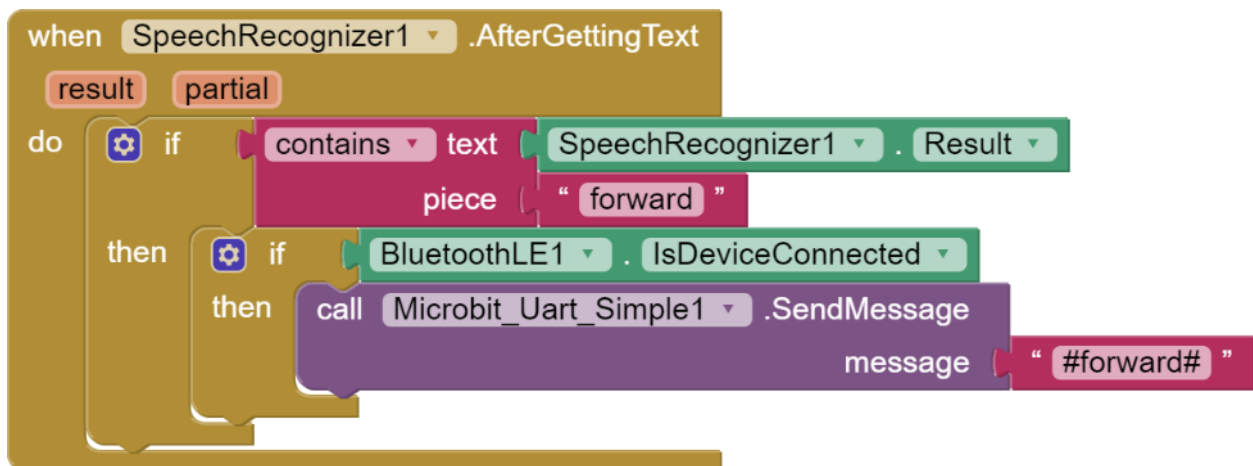
Via bovenstaande programmeermethode – en door het implementeren van de spraak-naar-tekst AI-dienst – hebben we een manier gevonden om een stemcommando als tekst op te slaan. Deze tekst wordt gebruikt als invoer om de robotauto verschillende bewegingen te laten uitvoeren. Hiervoor worden de binnenkomende spraakopdrachten gefilterd en gezocht op specifieke trefwoorden die op ons hoesje van toepassing zijn, namelijk voorkant, achterkant etc. Op deze manier kan de applicatie overeenkomen met een aantal verschillende spraakopdrachten. In de volgende tabel wordt dit concept kort uitgelegd:

| <i>door de Spraakherkenner ontvangen zin het woord bevat:</i> | <i>Stuur vervolgens het bericht:</i> | <i>En de auto zal:</i> |
|---|--------------------------------------|------------------------|
| "vooruit"   | #vooruit#                            | ga vooruit             |
| "achteruit"   | #achteruit#                          | Achteruit gaan         |
| "links"   | #links#                              | sla linksaf            |
| "rechts"  | #rechts#                             | ga rechtsaf            |
| "stop"  | #stop#                               | stop                   |

Zinnen die lijken op de volgende hebben hetzelfde effect op de beweging van de auto:

**Vooruit:** "Ga vooruit ", "Ga vooruit , alstublieft", "Ik wil dat je vooruit gaat ", enz...

Het volgende script instrueert de applicatie om het bericht #front# naar de robotauto te sturen, wanneer een opgenomen spraakcommando het woord "vooruit" bevat. Wanneer het bericht #front# vervolgens door de robotauto wordt ontvangen (gebaseerd op de code die al vanuit de Makecode-omgeving naar de robotauto is geüpload), beweegt de robotauto vooruit.



Hieronder wordt de functionaliteit van elk commandoblok in meer detail uitgelegd:

**When\_SpeechRecogniser.AfterGettingText** : dit blok ontvangt het tekstresultaat nadat het opgenomen spraakcommando naar tekst is geconverteerd. Dus wanneer de spraakopdracht van de gebruiker door de SpeechRecogniser naar tekst wordt geconverteerd, zal dit blok de volgende blokken met opdrachten **uitvoeren / uitvoeren**:

**Als** SpeechRecogniser.Result , namelijk de tekst die wordt geretourneerd als resultaat van het SpeechRecogniser-proces, **\_text bevat (), stukje “forward”** (dat wil zeggen, de ontvangen tekst bevat het woord “forward”)

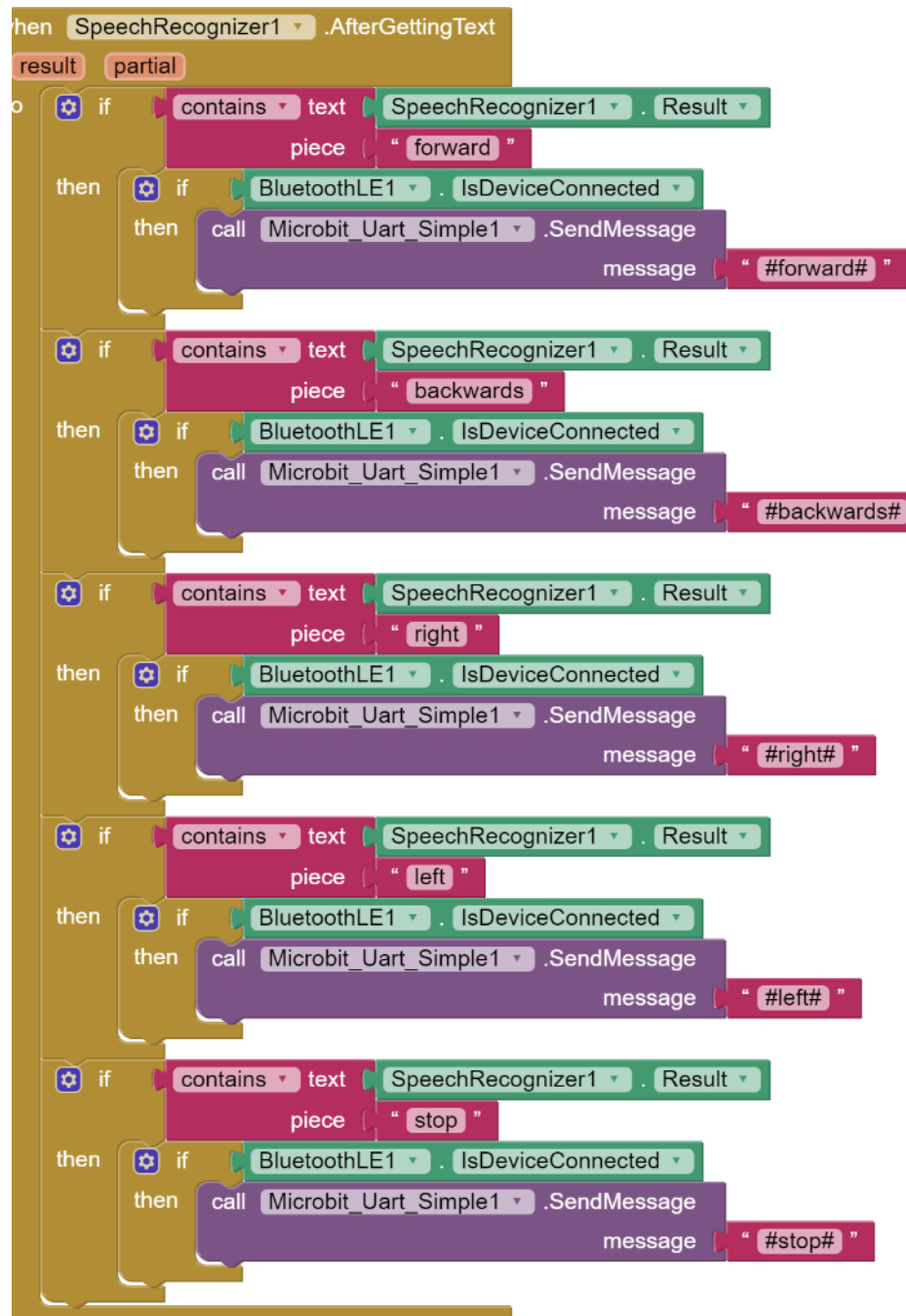
**Dan als BluetoothLE1** is **verbonden** naar het **gewenste** apparaat, de **Microbit\_Uart\_Simple1.SendMessage** wordt **aangeropen** en verzendt het bericht **#forward# ( tekst “vooruit” )** naar de robotauto. Als gevolg hiervan gaat de robotauto vooruit.

Om het in lijn te brengen met de geproduceerde beslisboom:

“ **Wanneer de SpraakHerkenner** heeft het spraakcommando van de gebruiker omgezet naar **tekst** , controleer **als het resultaat van de SpeechRecogniser** (namelijk de tekst) **bevat** het woord “ **vooruit** “. **Als** dat doet het **dan ook versturen** het bericht “ **voorkant** ” via de **Microbit\_Uart\_Simple** ”.

De rest van de spraakcommando's zijn op dezelfde manier te herkennen. Daarom zou het bovenstaande script moeten worden uitgebreid door nog vier “**als...dan**” -voorwaarden toe te voegen, en de bijbehorende **tekststukken** en **tekstberichten** , die onze robotauto zullen instrueren om een specifieke beweging uit te voeren (dat wil zeggen **achteruit** en het bericht “**#backwards #**” voor achteruit rijden, **naar rechts** en bericht “**#right#**” voor rechtsaf draaien enz.).

Het volledige script wordt weergegeven in de volgende afbeelding ( *Figuur 21* ):



Figuur 21: Het volledige script voor de component SpeechRecognizer na het ophalen van tekst

## Beperkingen – op weg naar een optimale programmeeroplossing

De bovenstaande programmeeroplossing heeft één groot nadeel; het vereist dat de gebruiker elke keer dat hij een spraakopdracht wil geven, op de knop “Druk om te spreken” drukt. Dit maakt het hele proces minder intuïtief in termen van rijgedrag en kan ook het aanvankelijke enthousiasme van uw leerlingen verminderen, omdat deze oplossing wellicht erg lijkt op het proces van afstandsbediening (dat wil zeggen, altijd op een knop drukken om één stemcommando te geven).

Om dit nadeel te ondervangen wordt ook een optimale programmeeroplossing voorgesteld, waarbij de gebruiker aan het begin van het proces slechts één keer op de knop “Druk om te spreken” hoeft te drukken, waarna de applicatie elke keer zal proberen de stem van de gebruiker te horen. 2 seconden.

**Let op:** dit is een iets geavanceerdere oplossing, dus afhankelijk van het niveau van uw leerlingen kunt u dit onderdeel overslaan en deze leeractiviteit hier beëindigen.

### Een kloksensor toevoegen

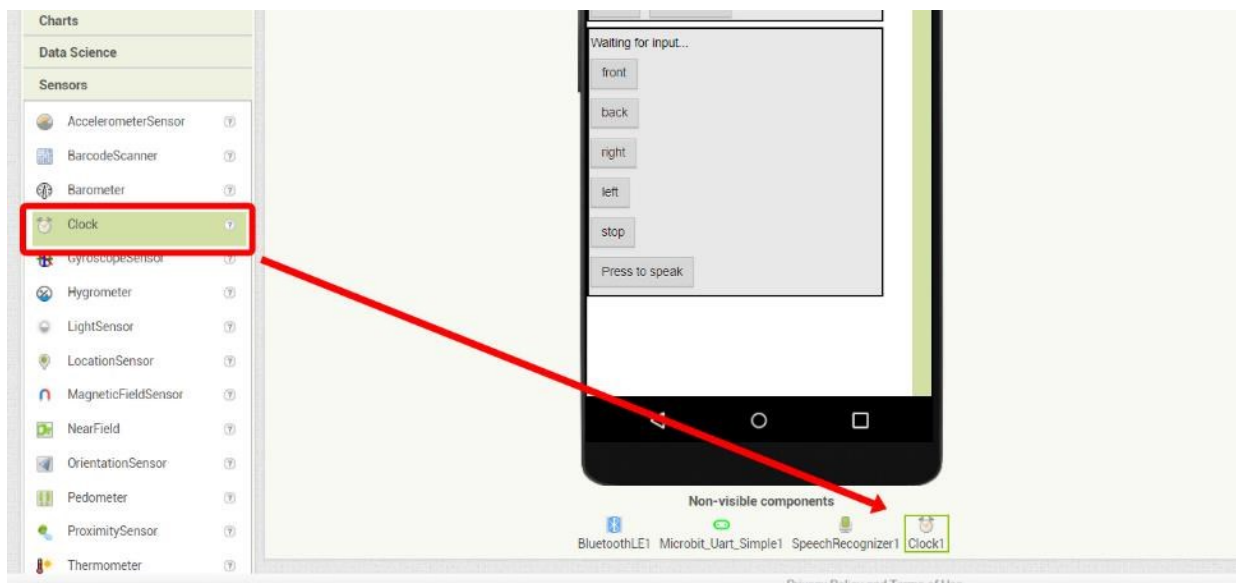
In deze oplossing voegen we een kloksensor toe die een afteltimer voor de applicatie instelt. Wanneer de knop “Druk om te spreken” wordt ingedrukt, wordt de timer geactiveerd/ontstoken en begint het aftellen voor een bepaald tijdsinterval. Gedurende dit tijdsinterval luistert de applicatie naar een binnenkomend spraakcommando. Als de applicatie een geldig stemcommando herkent, zal deze de robotauto de opdracht geven de bijbehorende beweging uit te voeren. Anders blijft het zoeken totdat het een geldig commando hoort. Wanneer het aftellen eindigt, wordt de timer automatisch opnieuw ingesteld en begint een nieuwe aftelling (voor hetzelfde tijdsinterval).

Daarom hoeft de gebruiker slechts één keer op de knop “Druk om te spreken” te drukken om het hele proces te initialiseren, rekening houdend met het feit dat de applicatie tussen tijdsintervallen slechts één geldig stemcommando kan ontvangen.

Hier ziet u hoe deze oplossing kan worden geïmplementeerd.

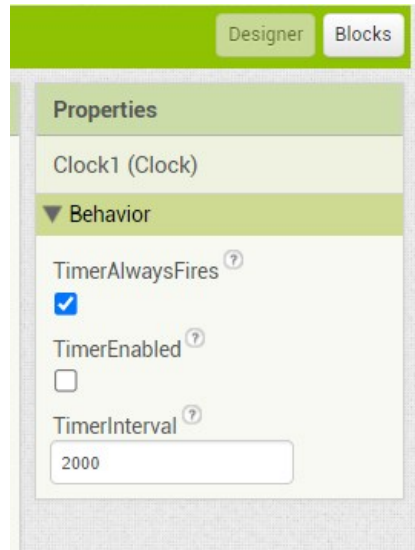
Eerst moet u de kloksensor aan de applicatie toevoegen. Ga in het ontwerpmenu naar het tabblad “Sensoren” en sleep de “Klok”-sensor naar het scherm ( *Figuur 22* ).

**Let op:** De Klok is ook een niet-zichtbaar item en verschijnt dus niet in het schermvoorbeeld



Figuur 22: Het item Klok toevoegen aan de applicatie

Selecteer de component Klok in de componentenlijst en stel in het menu Eigenschappen de volgende parameters in: a) vink het vakje onder TimerAlwaysFires aan om de timer van de klok in te schakelen elke keer dat het aftellen eindigt, b) in het veld TimerInterval de duur van de timer, door handmatig een numerieke waarde in milliseconden in te voeren (in het voorbeeld 2000) ( Afbeelding 23 ).



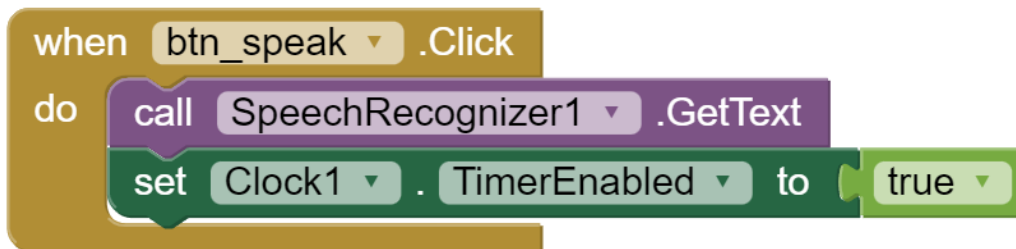
Figuur 23: De eigenschappen van de klokcomponent instellen

**Opmerking 1:** de duur van een interval wordt gemeten in milliseconden. 1 seconde is gelijk aan 1000 milliseconden. Daarom is 2000 ms gelijk aan 2 seconden.

**Opmerking 2:** het vakje onder TimerEnabled is niet aangevinkt, omdat we niet willen dat de Timer wordt ingeschakeld wanneer onze applicatie wordt geïnitieerd. We willen dat het wordt ingeschakeld wanneer de knop 'Druk om te spreken' wordt ingedrukt.

De volgende stap is het programmeren van dit nieuwe onderdeel.

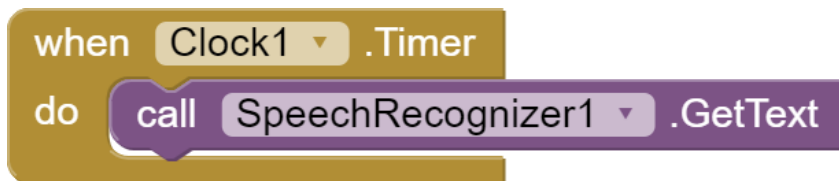
Voeg in het menu Blokken de opdracht “ **set Clock1.TimerEnabled to** ” toe binnen de gebeurtenishandler “ **When btn\_speak.Click** ” en onder de opdracht “ **call SpeechRecognizer.GetText** ” . Knip dan een “**WAAR**” voorwaarde aan de rechterkant van de opdracht. Via dit commando wordt de timer van de klok ingeschakeld na de eerste activering van de SpeechRecognizer.



Nu we de timer voor de eerste keer hebben geactiveerd/aangestoken (na het indrukken van de “Spreek”-knop), is de volgende stap het opnieuw activeren van de SpeechRecognizer elke keer dat het aftellen eindigt. De duur van de timer wordt tijdens dit proces automatisch verlengd.

Om dit te doen, selecteert u de component Klok en selecteert u in het zwevende menu de gebeurtenishandler " **When Clock1.Timer** ". Plaats vervolgens de “**roep SpeechRecognizer1.GetText**” commando binnen de handler.

Via dit script instrueren we onze applicatie om “ **de SpeechRecogniser op te roepen om de tekst op te halen**” van wat het hoort, elke keer dat de kloktimer wordt geactiveerd.



**Uitleg van de volledige code: Wanneer de gebruiker voor de eerste keer op de “spreek”-knop klikt , wordt de SpeechRecogniser aangeroepen en vervolgens wordt de kloktimer ingeschakeld. Als de timer is ingeschakeld, wordt deze elke 2 seconden geactiveerd. Elke keer dat hij vuurt, wordt de SpeechRecogniser gebeld, wachtend op het ontvangen van een hoorbare/verbale input (in ons geval een stemcommando).**

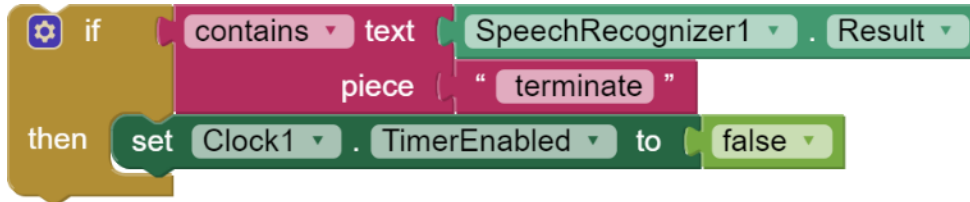
U hebt dus een applicatie gemaakt waarmee de gebruiker slechts één keer, in het begin, op de knop “Druk om te spreken” kan klikken en vervolgens automatisch elke 2 seconden de SpeechRecogniser belt om te controleren op nieuwe spraakopdrachten.

Een extra commando toevoegen om de applicatie te beëindigen

Door de component Klok toe te voegen, hebt u een intuïtievere toepassing gecreëerd. De applicatie blijft echter voor altijd actief. Dit komt omdat de timer van de klok voortdurend en elke 2 seconden afgaat, waardoor een oneindige lus ontstaat, waarin de SpeechRecogniser voortdurend op zoek is naar geldige spraakopdrachten.

Om dit probleem op te lossen, moet je wat extra codeblokken toevoegen aan de applicatie. Concreet moet u aan het script in Figuur 20 het volgende blok met opdrachten toevoegen.

in het bijzonder nog een “if...then” -voorwaarde toe aan het einde van het script. Plaats binnen het if-gedeelte een blok met opdrachten waarmee SpeechRecognizer het tekststuk “termineren” kan identificeren . Plaats in het toenmalige gedeelte een commandoblok dat de kloksensor uitschakelt, namelijk de “zet Clock1.TimerEnabled in op” “false” blokken .



Door deze extra codeblokken wordt de kloktimer uitgeschakeld wanneer de gebruiker het woord beëindigen zegt. De lus wordt dus beëindigd en de SpeechRecognizer stopt met zoeken naar geldige spraakopdrachten.

**Tip:** Afhankelijk van het niveau van uw leerlingen kunt u ze aanmoedigen om hun eigen oplossing voor het oneindige lusprobleem te vinden. Op deze manier kunnen ze beter begrijpen hoe de kloksensor werkt en hebben ze meer zelfvertrouwen over het hele proces.

### 3.5.4 Experiment 2

Deze <sup>tweede</sup> activiteit is behoorlijk uitgebreid, dus je moet een paar uur uittrekken om alle concepten en hun inherente aspecten te behandelen.

Om het tweede idee te introduceren, kun je niet alleen je leerlingen aanmoedigen om beslissingsbomen of stroomdiagrammen te ontwerpen die mogelijke acties of gedragingen weergeven die de robotauto kan aannemen, maar je kunt ook een dialoog op gang brengen door een aantal van de volgende **vragen te stellen** :

- Wat voor soort representaties moet u maken om de gecodeerde informatie beter te illustreren?
- Welke logische paden of operatoren moet u gebruiken om een intelligente agent te helpen de best mogelijke actie te ondernemen?
- Wat is de beste manier om de resultaten van de perceptie van de AI te gebruiken? (denk bijvoorbeeld op basis van de reactie van de robotauto aan gevallen waarin dergelijke technologie van toegevoegde waarde zou zijn)

Door deze activiteit en de discussie die volgt, zullen de leerlingen het volgende kunnen begrijpen:

- Hoe beslissingsbomen en stroomdiagrammen mogelijke logische paden kunnen onthullen, die ook kunnen leiden tot beslissingen over welke operators moeten worden gebruikt
- Begrijp hoe een intelligente agent het best mogelijke gedrag of de best mogelijke prestaties kan aannemen
- Begrijp hoe intelligente agenten hun omgeving kunnen waarnemen en kunnen reageren op binnenkomende informatie

Voor het ontwerpen en maken van de applicatie kunt u gebruik maken van het bestand “Students\_Worksheet\_for\_Activity\_2.pdf” en uw leerlingen discreet door het proces begeleiden en tips geven waar nodig. Je kunt ze ook aanmoedigen om met verschillende stemcommando's (en dus **tekststukken**) te experimenteren en de resultaten te observeren. Je kunt ze hiervoor adviseren om een tabel te maken – zoals hieronder – waarin ze vastleggen welke stemcommando's succesvol zijn en welke niet om de robotauto in een bepaalde richting te laten bewegen.

| Spraakopdracht | Succes | Mislukking |
|----------------|--------|------------|
| Ga vooruit     | √      |            |
| Ga naar voren  |        | √          |
| ....           |        |            |

Voor de geoptimaliseerde oplossing (met de toevoeging van het item Klok) kunt u uw leerlingen aanmoedigen om verschillende tijdsduren in te stellen voor Timer Interval en hun observaties over de functionaliteit van de applicatie vast te leggen. Je kunt ze ook voorstellen een tabel te maken (zoals hieronder) om hun observaties te ordenen. Bovendien kunt u elk team aanmoedigen een ander tijdsinterval te gebruiken en de resultaten in parallelle demonstraties te vergelijken.

| TimerInterval | Observatie                       |
|---------------|----------------------------------|
| 500 ms        | De applicatie werkt erg snel     |
| 2000 ms       | De applicatie werkt naar behoren |
| ....          | ....                             |

## 3. 6 Activiteit 3: Introductie van het idee van leren door een model te trainen voor het herkennen van stemcommando's

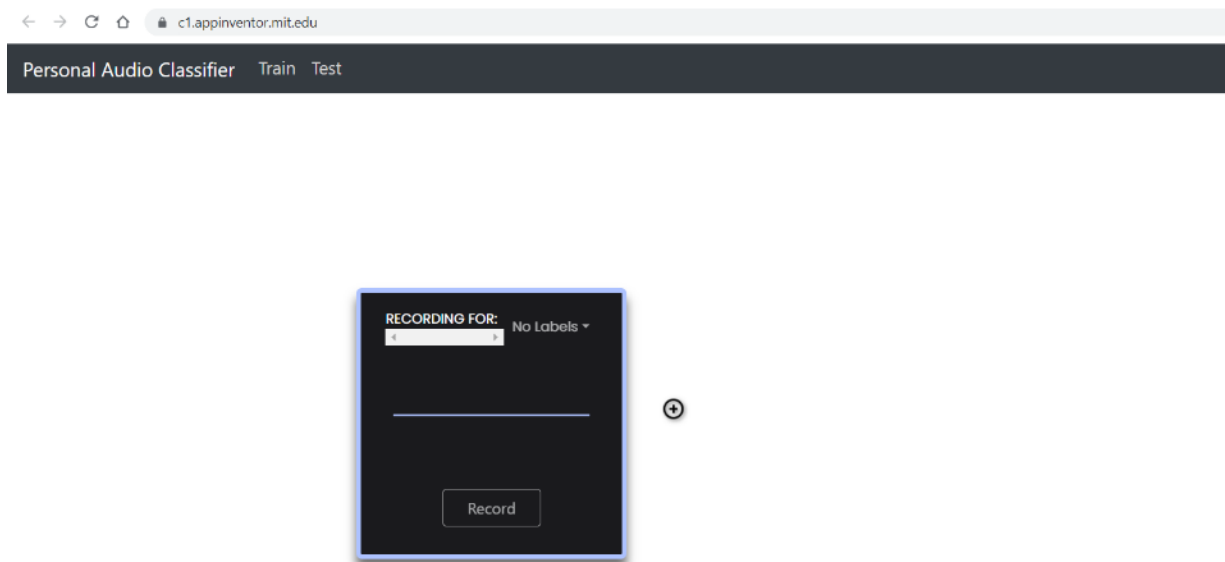
### 3.6.1 Beschrijving

In deze activiteit maken de leerlingen kennis met het derde <sup>Grote</sup> Idee, namelijk Leren, door een model te trainen om specifieke stemcommando's te herkennen. Door deze activiteit zullen ze de rol van Machine Learning (ML) en Machine Learning-algoritmen begrijpen bij het helpen van computers bij het leren. Ze zullen in het bijzonder leren hoe ze de Personal Audio Classifier ML-tool kunnen gebruiken om een model te trainen dat spraakopdrachten classificeert volgens specifieke criteria.

### 3.6.2 Personal Audio Classifier gebruiken om een model te trainen

In de vorige activiteit leer je hoe je de spraakherkenningsservice kunt gebruiken om spraakopdrachten op te nemen en deze in tekst om te zetten, zodat je door de robotauto kunt navigeren. In deze activiteit leer je hoe je een model traint om een aantal stemcommando's waar te nemen en deze te classificeren volgens specifieke criteria. Hiervoor maak je gebruik van de Personal Audio Classifier (PAC) trainingsomgeving ( <https://c1.appinventor.mit.edu/> ).

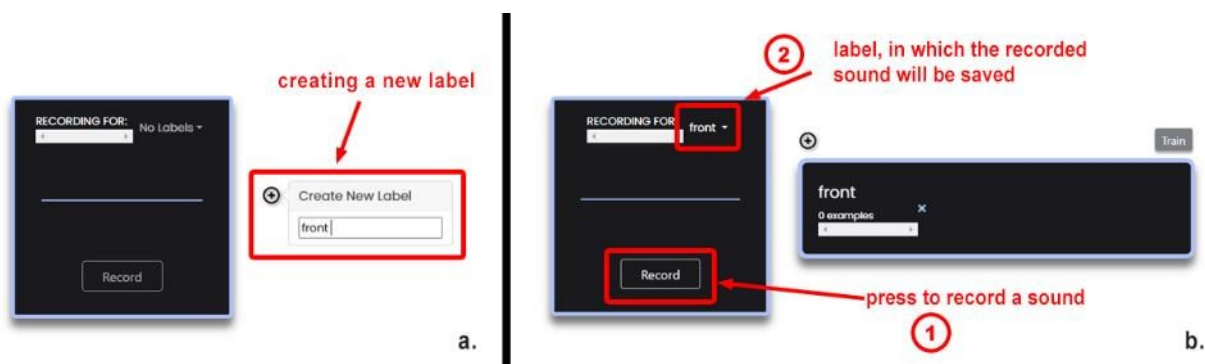
Figuur 24 toont de PAC-trainingsomgeving. Zoals je ziet is het model leeg. Er zijn geen categorieën, geen labels en geen geclassificeerde geluiden. U zult daarom een aantal categorieën moeten aanmaken. Elk van deze categorieën zal verschillende opgenomen audiofragmenten bevatten (bijvoorbeeld verschillende commando's die kunnen worden geïdentificeerd zoals voor, achter enz.), die zullen worden weergegeven door een gemeenschappelijk label (bijvoorbeeld voor, achter enz.).



Figuur 24: Personal Audio Classifier (PAC) trainingsomgeving

**Let op:** Zorg ervoor dat uw apparaat (PC, laptop etc.) is uitgerust met een microfoon/luidspreker. Anders kunt u geen geluiden opnemen.

De eerste stap bij het creëren van een categorie met veel voorkomende geluiden is het maken van een nieuw label. Klik op de knop + en typ in het zwevende menu 'Nieuw label maken' dat verschijnt de naam van de categorie die u wilt maken (dwz 'voorkant' in het voorbeeld in *Figuur 25a* ). Druk op de “Enter”-toets en er verschijnt een nieuw venster met de titel van het label dat u hebt gemaakt ( *Figuur 25b* ). Dit is de eerste categorie geluiden die u gaat vullen met audiofragmenten. Om dit te doen, drukt u op de knop Opnemen ( *Figuur 25b* ) ( 1 ) om te beginnen met het opnemen van de audio die door uw microfoon wordt ontvangen (voor meer informatie, zie de “ **Tips bij opname** ” sectie). Zorg ervoor dat u voor de juiste categorie opneemt, door de naam van het label te controleren dat verschijnt naast het gedeelte 'Opnemen voor' ( *Afbeelding 25b* ) ( 2 ).



Figuur 25: een. Een nieuw etiket aanmaken; B. Beginnen met het opnemen van audiofragmenten om de categorie “voorkant” te vullen

Herhaal hetzelfde proces om alle verschillende categorieën spraakopdrachten te maken die u door uw toepassing wilt laten herkennen en die de robotauto moet uitvoeren (bijvoorbeeld voor, achter, rechts enz.).

**Opmerking:** Het is belangrijk om de exacte naam van elk label/categorie te onthouden, zodat de applicatie deze categorieën met succes kan ophalen. Moedig uw leerlingen aan om een tabel te maken waarin ze de naam van elk label en een korte beschrijving van de daarin opgeslagen geluiden noteren.

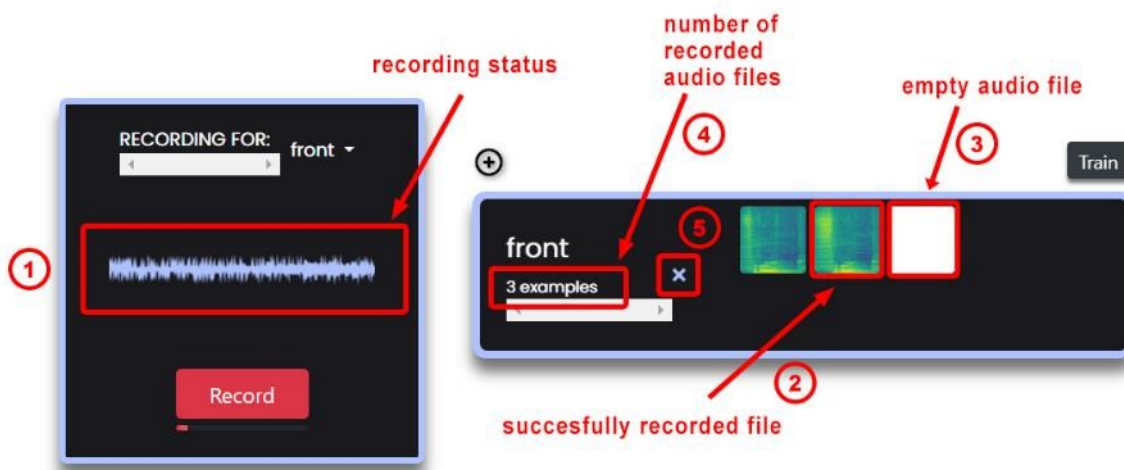
| Etiket/Categorie | Resultaat van spraakopdracht         |
|------------------|--------------------------------------|
| voorkant         | Verplaats de robotauto naar voren    |
| klap             | Verplaats de robotauto achteruit     |
| links            | Laat de robotauto naar links draaien |
| ...              | ...                                  |

**Tips voor het opnemen ( *Figuur 26* ):**

Wanneer u op de opnameknop drukt, wordt het opnameproces gedurende ongeveer 1 seconde geactiveerd. Gedurende deze tijd neemt uw microfoon elk geluid op dat hoorbaar is. Vanwege deze tijdslimiet wordt aanbevolen dat u korte spraakopdrachten opneemt (bijvoorbeeld “voor” in plaats van “vooruit”).

U kunt de opnamestatus controleren door naar de blauwe lijn **(1)** boven de opnameknop te kijken. Als de rechte lijn in een golfvorm verandert, werkt uw microfoon goed. Elk opgenomen geluid wordt opgeslagen als een nieuw audiobestand binnen de geselecteerde categorie (“voorkant” in het voorbeeld). Als uw microfoon met succes een geluid heeft opgenomen, wordt dit omgezet in een audiobestand en verschijnt er een gekleurd spectrogrampictogram **(2)**. Als uw microfoon geen geluid heeft opgevangen, verschijnt er een wit vakje, wat aangeeft dat het opgeslagen audiobestand leeg is **(3)**. Als u een audiobestand wilt verwijderen dat al is opgeslagen, beweegt u uw cursor over het overeenkomstige audiobestand. Er verschijnt een X-markering in de rechterbovenhoek. Klik erop en het audiobestand wordt verwijderd.

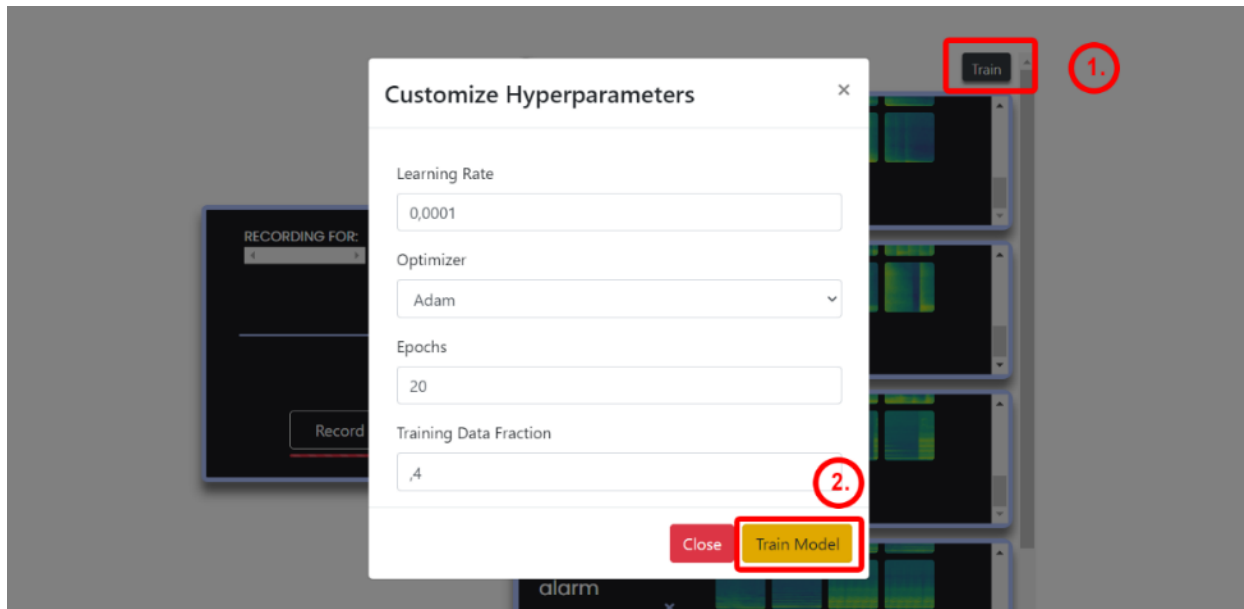
Om een redelijk betrouwbaar getraind model te maken, moet u in elke categorie minimaal 5 tot 6 audiobestandsvoorbeelden/samples opnemen. Het aantal reeds opgenomen audiobestanden wordt aangegeven in het veld onder de hoofdtitel van elke categorie **(4)**. Als je een hele categorie wilt verwijderen, druk je op de X-knop **(5)** naast het veld ‘voorbeelden’.



Figuur 26: Informatie over het opnameproces en de opgenomen audiobestanden

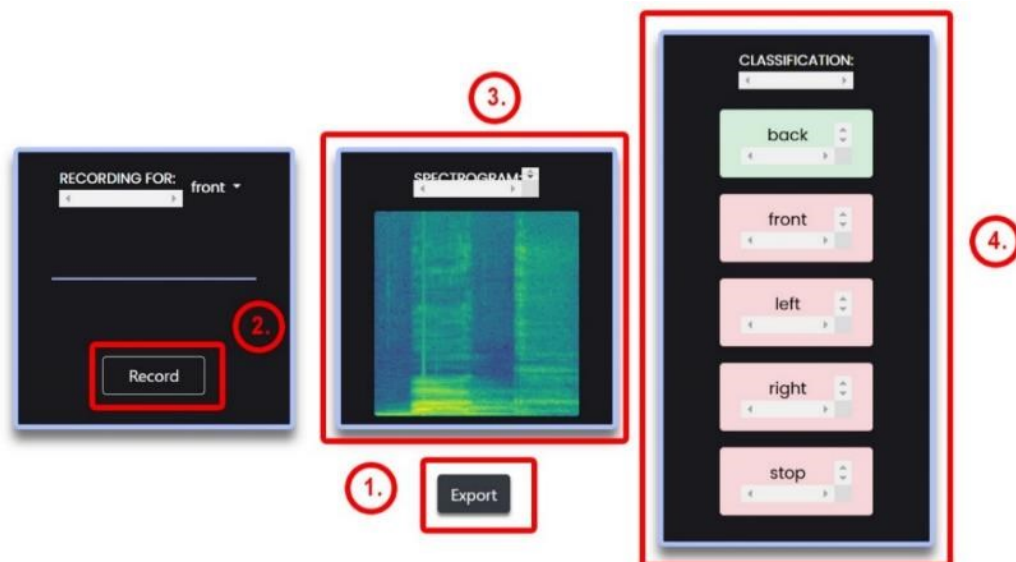
### Het model trainen, testen en exporteren

Zodra alle labels/categorieën zijn gemaakt, met ten minste 5 tot 6 geluiden opgenomen in elke categorie, kunt u doorgaan met het trainen van het model ( Afbeelding 27 ). Klik hiervoor op de knop Trein **(1)**. Er verschijnt een nieuw venster waarin u enkele parameters kunt aanpassen. Behoud de standaardparameters en klik op de knop Treinmodel **(2)** om het trainingsproces te starten.



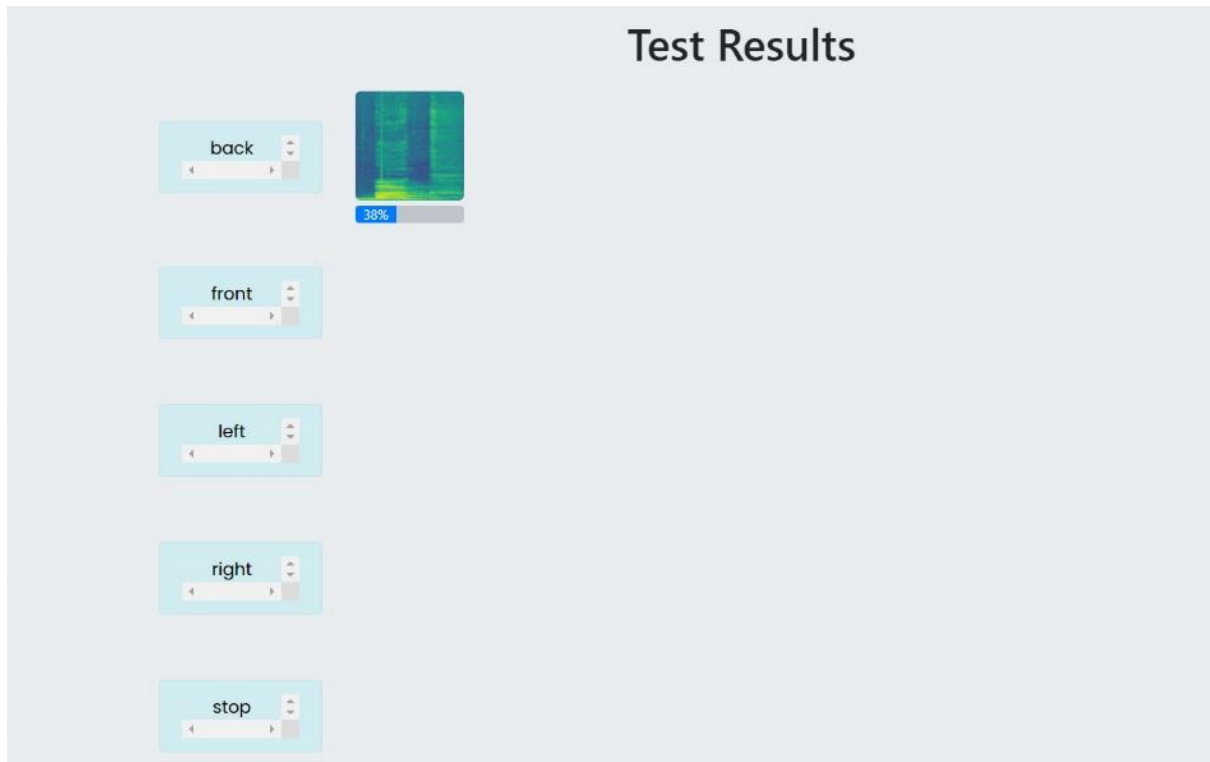
Figuur 27: Het model trainen

Wacht een paar minuten totdat de training is voltooid. Voordat u het getrainde model exporteert, wordt u ten eerste aangeraden het gemaakte getrainde model te testen door op de knop Opnemen **(2)** te klikken, een geluid op te nemen (druk bijvoorbeeld op opnemen terwijl u 'terug' zegt) en kijk of het opgenomen geluid wordt herkend en geclassificeerd in de overeenkomstige categorie. Dit kunt u doen door aan te vinken welke van de categorieën in het Classificatievenster **(4)** groen wordt. Het spectrogram van het opgenomen geluid **(3)** verschijnt ook, wat bevestigt dat het binnenkomende geluid met succes is opgenomen.



Figuur 28: Het getrainde model testen en exporteren

U kunt ook naar beneden scrollen op de pagina om het betrouwbaarheidsniveau van de voorspelling te bekijken. In Figuur 29 wordt u bijvoorbeeld geïnformeerd dat het betrouwbaarheidsniveau van de “terug”-voorspelling 38% is.



Figuur 29: Het betrouwbaarheidsniveau van de voorspelling voor de binnenkomende opname

Moedig uw leerlingen aan om verschillende tests uit te voeren met het getrainde model. Als er te veel fouten worden gevonden, adviseer hen dan om op het tabblad “trein” te klikken ( Figuur 30 ) om wijzigingen aan te brengen in de geregistreerde gegevens.



Figuur 30: Klik op het tabblad Train om terug te keren naar de opnamemodus

Als het getrainde model goed werkt, klikt u op de knop Exporteren **(1)** ( *Afbeelding 28* ) om het getrainde model lokaal op uw computer op te slaan als een *.mdl* -bestand.

**Opmerking:** het bestand krijgt automatisch de naam *model.mdl* wanneer het wordt gedownload, maar u kunt dit bestand (na het downloadproces) handmatig hernoemen naar iets betekenisvol (bijvoorbeeld *voice\_commands\_model.mdl* enz.).

**Belangrijke opmerking:** helaas staat de huidige versie van de Personal Audio Classifier-omgeving geen wijzigingen toe aan het getrainde model, na het verlaten van de pagina. Trek daarom voldoende tijd uit voor deze activiteit tijdens een les-/maakuur.

In de vierde <sup>activiteit</sup> leer je hoe je dit model kunt integreren in de applicatie die je al hebt ontworpen in App Inventor, en hoe je het geëxporteerde model kunt gebruiken om de binnenkomende spraakopdrachten te classificeren om de robotauto te instrueren dienvooreenkomstig te bewegen.

### 3.6.3 Experiment 3

In het vorige gedeelte zijn enkele tips gegeven over hoe u deze activiteit bij uw leerling kunt introduceren. Het eerste wat we moeten doen is hen uitleggen wat het doel van deze activiteit is (dwz leren hoe je een ML-tool kunt gebruiken om een model te trainen dat in een later stadium door de robotauto kan worden gebruikt). Je kunt ze aanmoedigen om verschillende scenario's te bedenken waarin audioclassificatie de robotauto zou kunnen helpen om van zijn omgeving te leren, maar als uitgangspunt kun je ze aanmoedigen om na te denken over hoe ze een model kunnen trainen om verschillende stemcommando's te classificeren.

Moedig ze daartoe aan, voordat ze de Personal Audio Classifier-tool openen, om op te schrijven welke spraakopdrachten ze willen opnemen en wat het verwachte resultaat van elk van deze spraakopdrachten is (bijvoorbeeld wat ze van de robotauto verwachten). , als het spraakcommando “goed” is). Om dit proces te vergemakkelijken, kunt u hen adviseren een tabel zoals hieronder te maken en hun ideeën op te schrijven.

| Spraakopdracht | Resultaat |
|----------------|-----------|
|                |           |
|                |           |
|                |           |
|                |           |

Vraag hen vervolgens om de Personal Audio Classifier-tool te openen en kort uit te leggen hoe ze deze kunnen gebruiken. Moedig ze aan om een aantal labels te maken en in elke categorie enkele audiofragmenten op te nemen. Enkele tips die je ze kunt geven zijn:

- om zoveel labels/categorieën te creëren als de bewegingen die de robotauto zal uitvoeren
- experimenteer met verschillende accenten of probeer verschillende delen van een opgenomen woord te benadrukken
- noteer de exacte naam die ze aan elk label/categorie geven. Dit is belangrijk voor de volgende activiteit (dwz de 4e <sup>activiteit</sup>)

Zodra ze alle categorieën hebben gemaakt, vraag je ze om het model te trainen en te testen. Adviseer hen bij het testen ook het betrouwbaarheidsniveau voor elke voorspelling te controleren. U kunt ze aanmoedigen om in een tabel als de volgende te noteren of het opgenomen stemcommando correct is waargenomen en geclassificeerd en wat het betrouwbaarheidsniveau van elke voorspelling is.

| Geluid | Succesvol geclassificeerd | Betrouwbaarheidsniveau (%) |
|--------|---------------------------|----------------------------|
|        | JA NEE                    |                            |
|        | JA NEE                    |                            |
|        | JA NEE                    |                            |
|        | JA NEE                    |                            |

Als ze tevreden zijn met het resultaat, vraag ze dan om het model te exporteren en zo lokaal op hun computer op te slaan.

Tijdens deze activiteit kun je met hen de verschillende aspecten van dit proces bespreken. Om een dialoog met hen op gang te brengen, kunt u een aantal van de volgende **vragen stellen** :

- Wat kan een ML-tool doen?
- Met welke parameters moet u rekening houden bij het trainen van uw model?
- Is het belangrijk om een getraind model te evalueren voordat het in een AI-toepassing wordt gebruikt?
- Hoe kan een bevooroordeeld getraind model, dat spraakopdrachten classificeert, zelfrijdende auto's beïnvloeden?
- Hoe kunnen we bevooroordeelde getrainde modellen vermijden?

Door deze activiteit leren de leerlingen:

- Hoe u de Personal Audio Classifier ML-tool gebruikt, of vergelijkbaar met deze ML-tools

- Hoe je een model traint op basis van een geplande classificatie
- Hoe u een getraind model kunt testen en evalueren
- Hoe bevooroordeelde gegevens de nauwkeurigheid van een getraind model kunnen beïnvloeden

### 3. 7 Activiteit 4: Introductie van het idee van natuurlijke interactie door een getraind model te integreren in een AI-toepassing

#### 3.7.1 Beschrijving

In deze 4e <sup>activiteit</sup> leren de leerlingen hoe ze het getrainde model, geproduceerd in de context van de 3e activiteit, kunnen integreren <sup>in</sup> de applicatie die in de 2e activiteit is gemaakt om te observeren hoe de prestaties van de robotauto kunnen worden verbeterd. wordt beïnvloed wanneer een getraind model in de toepassing wordt geïntegreerd. Op deze manier zullen ze zich bewust worden van hoe AI-systemen gevoelig kunnen zijn voor fouten als gevolg van de beperkingen van AI in de natuurlijke interactie.

#### 3.7.2 Integreren van het getrainde model in de AI-applicatie

Voor de behoeften van deze activiteit kunt u doorgaan met werken aan de applicatie die u in Activiteit 2 hebt gemaakt, of u kunt het bestand “ *Robotic\_car\_SpeechRecognizer.aia* ” gebruiken. Als alternatief kunt u overwegen een nieuw *.aia* -project te maken , maar het wordt ten zeerste aanbevolen om door te gaan met het vorige bestand, omdat het voor leerlingen gemakkelijker zou zijn om de verschillen in de prestaties van de robotauto te vergelijken als ze de Spraakherkenner gebruiken. en bij gebruik van de Personal Audio Classifier.

Het toevoegen van wat extra componenten

Eerst moet u wat extra componenten toevoegen aan het applicatiescherm (in het ontwerpermenu). Concreet moet u **a)** een WebViewer-component toevoegen , **b)** de Personal Audio Classifier-extensie en **c)** een extra label waar de resultaten van de classificatie worden weergegeven.

**a)** Met de WebViewer-component kunnen toepassingen een URL hosten en de gebruiker omleiden naar een specifieke webpagina. Voor deze activiteit wordt dit onderdeel toegevoegd zodat de Personal Audio Classifier de database kan laden die is ingebed in het getrainde model.

Om dit onderdeel te vinden, gaat u naar het submenu Gebruikersinterface en sleept u het naar het scherm, onder de lay-out Horizontale opstelling (Afbeelding 31).



Figuur 31: Het WebView-onderdeel toevoegen

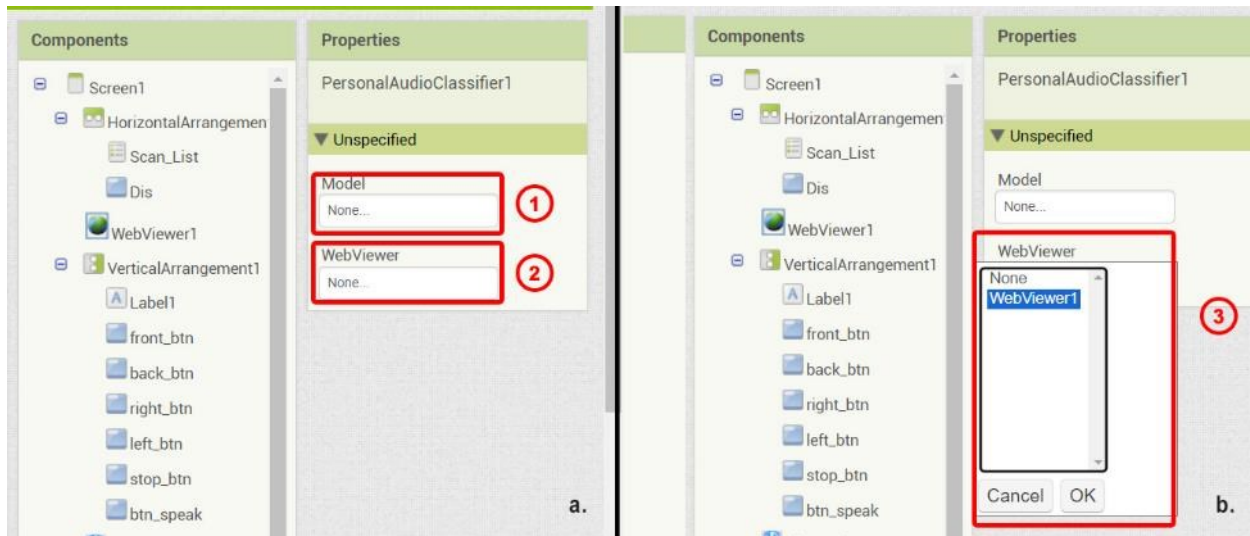
b) De volgende stap is het toevoegen van de Personal Audio Classifier-component, waardoor de applicatie het getrainde model kan gebruiken dat eerder door de PAC-trainingsomgeving is gemaakt. Om dit te doen, moet u deze extensie eerst lokaal naar uw computer downloaden. Ga naar deze link: <https://mit-cml.github.io/extensions/> en sla het overeenkomstige .aix-bestand op uw lokale schijf op ( Figuur 32 ). Voeg vervolgens deze extensie toe aan de App Inventor-omgeving op dezelfde manier als waarop je de extensies voor Bluetooth en micro:bit hebt toegevoegd .

| MIT APP INVENTOR        |  |                  |          |   |                            |
|-------------------------|--|------------------|----------|---|----------------------------|
| Supported:              |  |                  |          |   |                            |
| Name                    | Description  | Author           | Version  | Download .aix File                          | Source Code                |
| BluetoothLE             | Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information. | MIT App Inventor | 20230728 | <a href="#">BluetoothLE.aix</a>             | <a href="#">Via GitHub</a> |
| FaceMeshExtension       | Estimate face landmarks with this extension.   | MIT App Inventor | 20210414 | <a href="#">Facemesh.aix</a>                | <a href="#">Via GitHub</a> |
| LookExtension           | Adds object recognition using a neural network compiled into the extension.  | MIT App Inventor | 20181124 | <a href="#">LookExtension.aix</a>           | <a href="#">Via GitHub</a> |
| Microbit                | Communicate with micro:bit devices using Bluetooth low energy (needs BluetoothLE extension above).   | MIT App Inventor | 20200518 | <a href="#">Microbit.aix</a>                | <a href="#">Via GitHub</a> |
| PersonalAudioClassifier | Use your own neural network classifier to recognize sounds with this extension.  | MIT App Inventor | 20200904 | <a href="#">PersonalAudioClassifier.aix</a> | <a href="#">Via GitHub</a> |
| PersonalImageClassifier | Use your own neural network classifier to recognize images with this extension.  | MIT App Inventor | 20210315 | <a href="#">PersonalImageClassifier.aix</a> | <a href="#">Via GitHub</a> |
| PosenetExtension        | Estimate pose with this extension.   | MIT App Inventor | 20200226 | <a href="#">Posenet.aix</a>                 | <a href="#">Via GitHub</a> |
| TeachableMachine        | Use vision models trained in TeachableMachine with your device's camera.   | MIT App Inventor | 1        | <a href="#">TeachableMachine.aix</a>        | <a href="#">Via GitHub</a> |

Figuur 32: Download de Personal Audio Classifier-extensie door op PersonalAudioClassifier.aix te klikken

Na het importeren van de PersonalAudioClassifier-extensie sleept u deze naar het scherm van de ontworpen applicatie. PersonalAudioClassifier is ook een niet-zichtbare component en verschijnt daarom in de sectie niet-zichtbare componenten.

vervolgens de component PersonalAudioClassifier in het menu Componenten om de eigenschappen ervan te wijzigen ( *Afbeelding 33* ). Klik met name op het veld "Geen..." onder zowel de eigenschappen Model (1) als de WebViewer (2) om het .mdl- getrainde modelbestand (dat u eerder in de PAC-trainingsomgeving hebt gemaakt) dienovereenkomstig te uploaden en de component WebViewer1 te selecteren uit de zwevende lijst (3) .



Figuur 33: een. Het wijzigen van de eigenschappen van PersonalAudioClassifier; B. Upload het bestand van het getrainde model en selecteer de component WebViewer1

c) De laatste stap is het toevoegen van nog een label, dat de resultaten van de classificatie weergeeft, telkens wanneer een nieuw geluid door de applicatie wordt opgenomen.

Sleep een labelcomponent naar het scherm, onder de WebViewer-component en vanuit het menu Eigenschappen, wijzig de tekst van het label in 'In afwachting van classificatie' of iets dergelijks. U kunt de labelcomponent ook hernoemen – vanuit het menu Componenten – naar iets betekenisvol, zoals Classification\_label ( *Afbeelding 34* ).



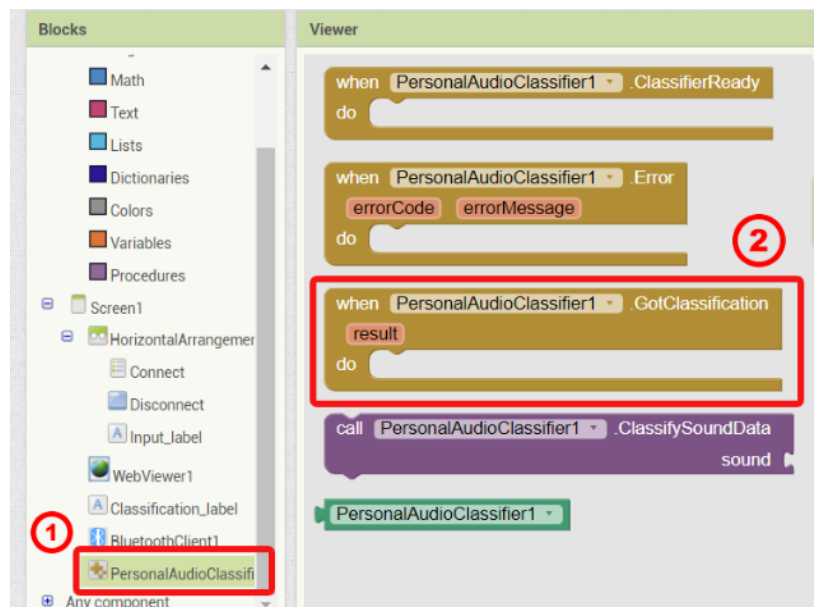
Figuur 34: Een tweede label toevoegen voor het weergeven van de classificatieresultaten

Na deze stap kunt u de nieuwe componenten programmeren.

### Programmeren van de PersonalAudioClassifier-component

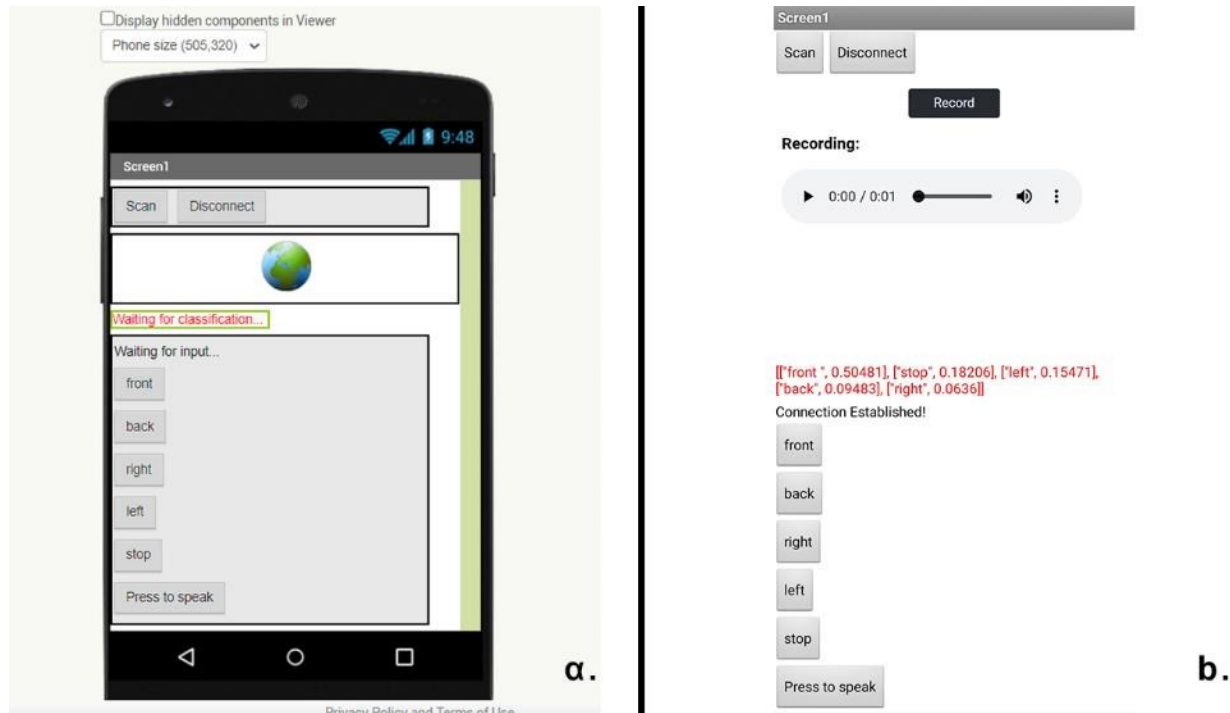
De volgende stap is het programmeren van de PersonalAudioClassifier-component en specifiek de functionaliteit ervan nadat de audioclassificatie van een binnenkomend geluid is voltooid.

Voor deze programmeerstap is het gebeurtenisblokcommando **“when PersonalAudioClassifier 1 .GotClassification ... result ... do” (2)** in het zwevende menu van de PersonalAudioClassifier-component **(1)** moet worden geïmplementeerd ( *Figuur 35* ). Sleep dit gebeurtenisblokcommando naar het gebied waar de code is samengesteld.



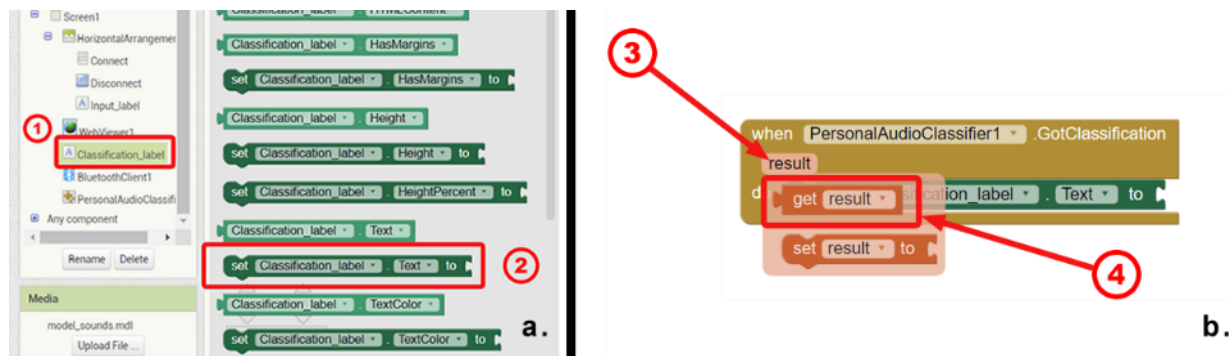
Figuur 35: Het vinden van de gebeurtenisblokopdracht wanneer PersonalAudioClassifier 1 .GotClassification ...result ..do..”

De component Personal Audio Classifier voegt een opnameknop toe aan de applicatie. Deze knop is niet zichtbaar in het ontwerpgebied. Het zal alleen zichtbaar zijn op het smart device (nadat het ontwerp en de programmering van de applicatie is voltooid, en zodra de applicatie is gebouwd en geïnstalleerd op een smart device) ( *Figuur 36* ). Het bovenstaande Gebeurtenisblok-commando bepaalt wat de toepassing moet doen met de resultaten van de opname, nadat het binnenkomende spraakcommando is geclassificeerd (dwz de robotauto opdracht geven dienovereenkomstig te bewegen).



Figuur 36: een. de applicatie zoals weergegeven in de ontwerperweergave; B. voorbeeld van de applicatie wanneer deze op een smartapparaat is geïnstalleerd

Eerst moet u een opdracht toevoegen waarmee de component Classification\_label de resultaten van de audioclassificatie kan weergeven. Om dit te doen, selecteert u de component Classification\_label (1) en sleept en plaatst u vanuit het zwevende menu de **set Classification\_label. Tekst naar** -blokopdracht (2) binnen de gebeurtenishandleropdracht (Figuur 37a). Beweeg vervolgens de cursor over het **resultaat** (3) veld, en sleep en klik op het **“haal resultaat”** (4) blokopdracht naar de **“set Classification\_label”** . Tekst naar” -opdracht (Figuur 37b).

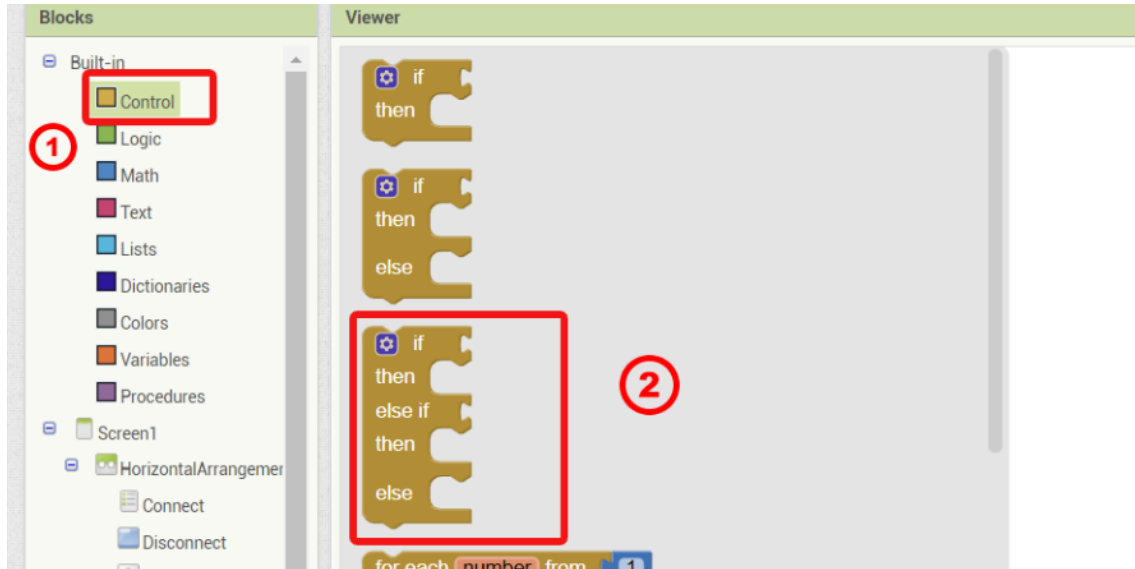


Figuur 37: een. Het vinden van de blokopdracht "set... Text to"; B. het vinden van het blokcommando "set result".

Door dit proces zal de tekst 'Wachten op classificatie' in Classification\_label ( zie Figuur 34 ) veranderen in de resultaten van de audioclassificatie, gebaseerd op het classificatiemodel dat we hebben geüpload, gevolgd door het betrouwbaarheidsniveau (bijvoorbeeld als Als het stemcommando aan de voorkant

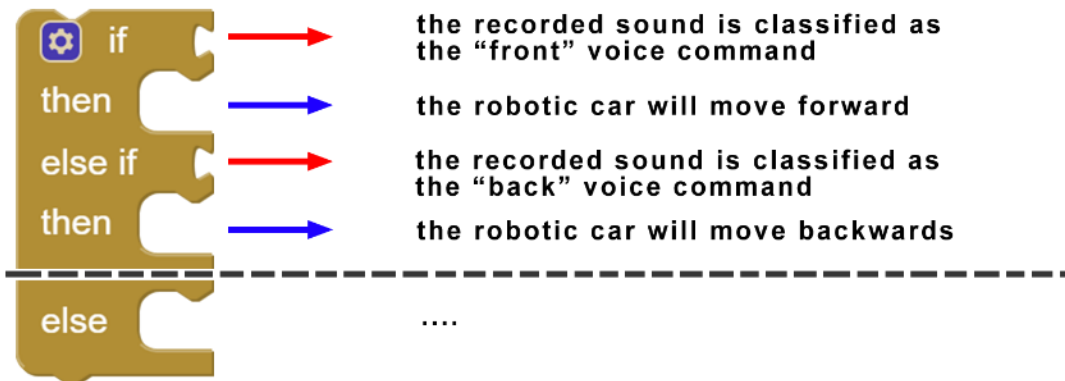
wordt opgenomen , kan de tekst van het label mogelijk veranderen in iets zoals weergegeven in Figuur 36b, dat wil zeggen: “ [voor, 0,50], [stop, 0,18], [links, 0,15], enz. ” ).

Vervolgens onder de " **set Classification\_label** ". **Tekst naar**" commando, plaats een "**als dan...anders als dan...anders**" (2) voorwaarde, te vinden in het menu Bediening (1) ( *Figuur 38* ), om te bepalen wat de applicatie zal doen op basis van de resultaten verkregen uit het classificatieproces.



Figuur 38: De voorwaarde 'als dan...anders als dan...anders' vinden

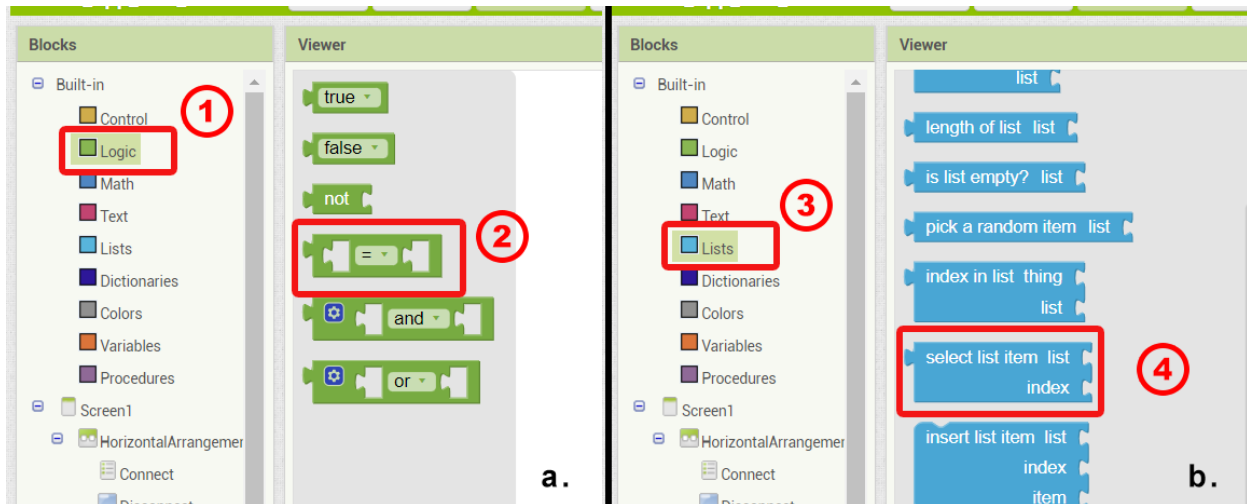
Het volgende diagram toont de parameters die we moeten invoegen in de voorwaarde “ **if then... else if then...else** ”.



In het bijzonder hebben we een reeks commando's nodig die herkennen en controleren welke geclassificeerde categorie/label wordt aangeroepen. Deze opdrachten worden in de instructie “if” of “else if” geplaatst. We hebben ook nog wat commando's nodig die de robotauto zullen instrueren om de juiste

beweging uit te voeren, volgens de resultaten van de classificatie. Deze commando's worden in de "then"-instructie geplaatst.

Om te controleren welke geclassificeerde categorie/label wordt aangeroepen, wordt een gelijk (=) blok (2) uit het logische menu (1) (Figuur 39a) en een "select list item list... index..." blok met opdrachten (4) geplaatst, in het Lijsten-menu (3) (Figuur 39b) moet worden gebruikt.



Figuur 39: a. Het gelijke (=) blok vinden; B. Het blok "selecteer lijstitemlijstindex" zoeken

Plaats eerst het equal (=) blok binnen de if-instructie, zoals aangegeven in het volgende blok met opdrachten.



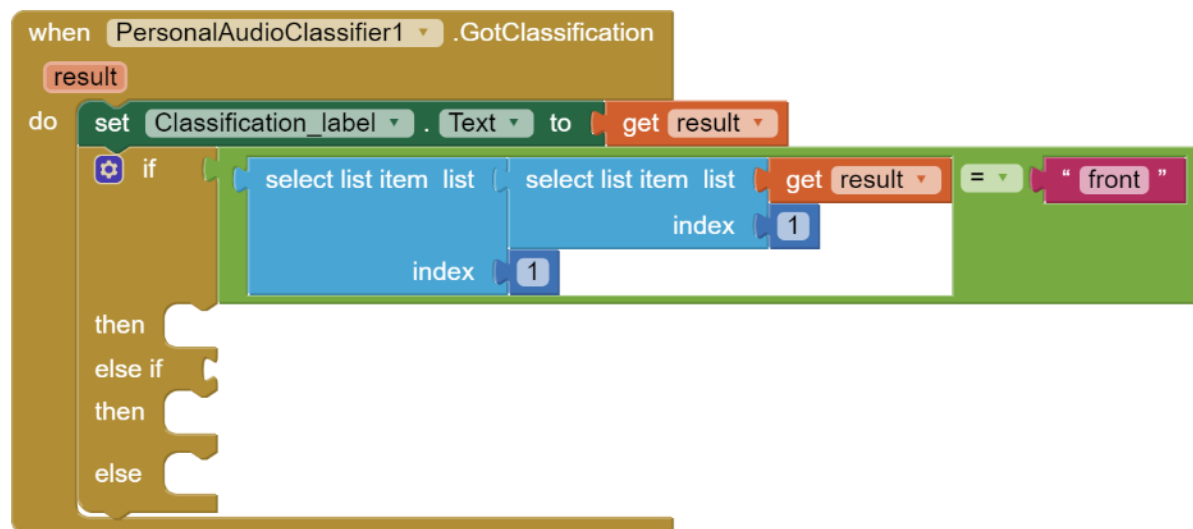
Het geretourneerde resultaat is feitelijk een **lijst** met de classificaties uit het geüploade model. De lijst bevat vijf sublijsten. Elke sublijst bevat een van de labels/categorieën (dwz 'voorkant', 'achterkant' enz.) waarvan de classifier denkt dat deze overeenkomt, gevolgd door het betrouwbaarheidsniveau. In het

bovengenoemde indicatieve voorbeeld: *[voor, 0,50], [stop, 0,18], [links, 0,15], enz.*, is de classifier er 50% zeker van dat het ontvangen commando “voor” is, en 18% zeker dat het ontvangen commando “voor” is. “stop” enz. We moeten het eerste item uit de eerste sublijst eruit halen (“terug” in het bovengenoemde geval) en testen of dit de categorie is waartoe het binnenkomende geluid behoort.

Om dit te doen, **zullen er twee commandoblokken** gebruikt worden.



Om te beginnen halen we het eerste item (index) uit het **get-resultaat** (onze hoofdlijst). Dan wordt dit item onze nieuwe hoofdlijst waaruit we opnieuw het eerste item (index) zullen extraheren, dat in ons voorbeeld “front” is (getypt in een **tekstinvoerblok**). Het eerste lijstitem bevindt zich altijd op index 1 en wordt in ons geval in een **basisnummerblok** getypt.

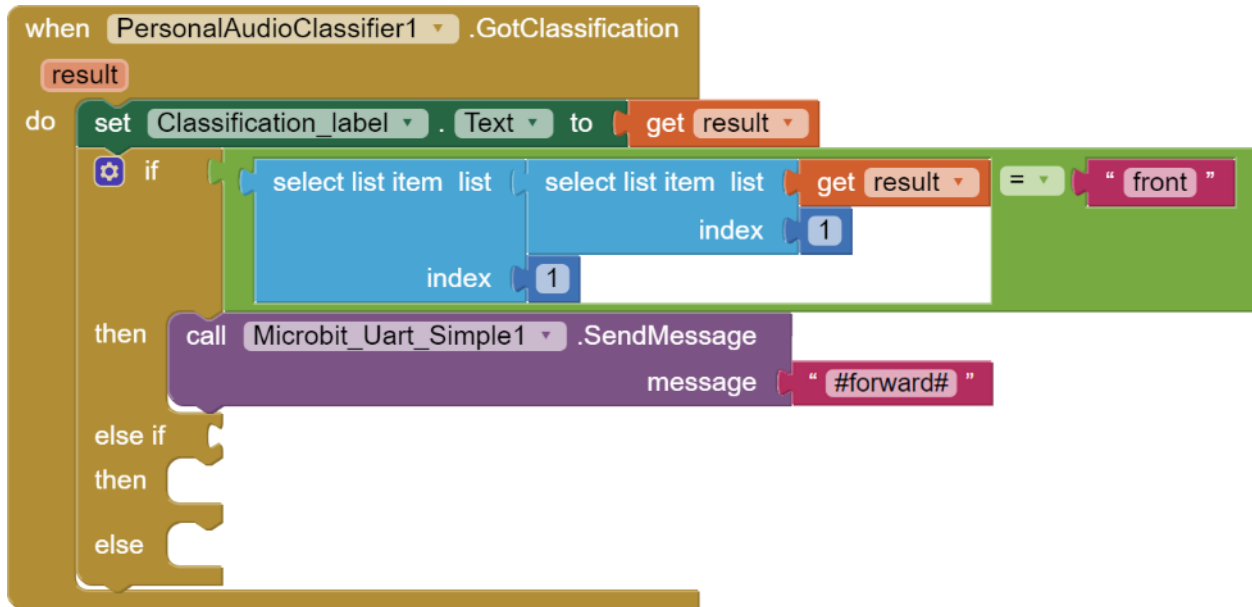


**Opmerking:** het basisnummerblok bevindt zich in het **Math-** menu, terwijl het tekstinvoerblok in het **Tekst-** menu staat.

De volgende stap is om de robotauto opdracht te geven vooruit te rijden. Om dit te doen, moeten we de applicatie de opdracht geven om (via Bluetooth) het corresponderende bericht (dat wil zeggen #forward#) naar de robotauto te verzenden (dat wil zeggen, zoals aangegeven in het Makecode-script).

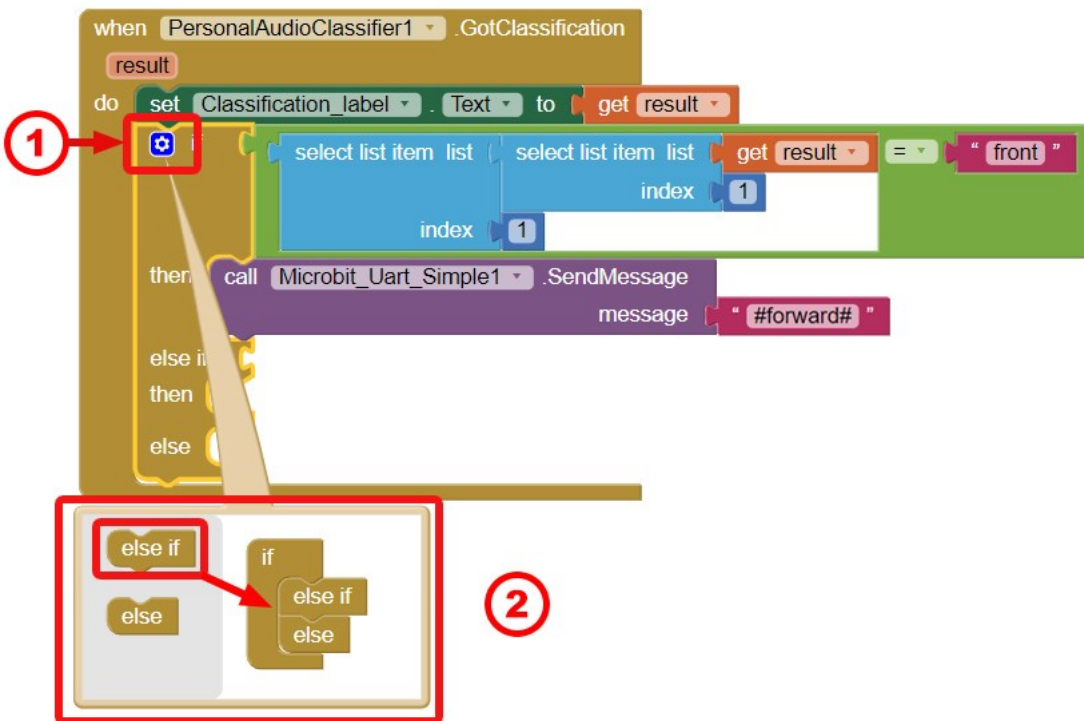
Plaats daarom binnen de “then”-instructie een “**call Microbit\_Uart\_Simple 1 . Bericht verzenden**”-blok. Bevestig vervolgens een **tekstinvoerblok** “” en typ het woord “**#forward#**”.

Dit is hoe de code eruit zal zien na het voltooien van de eerste twee instructies van de voorwaarde “**if then.. else if then...else**”.



Herhaal hetzelfde proces voor de andere opdrachten om uw toepassing te programmeren en de robotauto te instrueren andere bewegingen uit te voeren, afhankelijk van het geclassificeerde binnenkomende geluid.

**Opmerking :** als u meer “else if”-voorwaarden wilt toevoegen, klikt u op het blauwe tandwiel (1) naast de “if”-instructie en sleept u vanuit het zwevende menu (2) zoveel nieuwe “else if”-voorwaarden als u wilt. behoefte ( *Figuur 40* ).



Figuur 40: Meer anders toevoegen als voorwaarden

**Opmerking:** binnen elk tekstinvoerblok is het van cruciaal belang om **a)** hetzelfde label te gebruiken dat u hebt gebruikt tijdens de productie van het classificatiemodel, **b)** hetzelfde bericht in te voegen dat u in het Makecode-script hebt gedeclareerd.

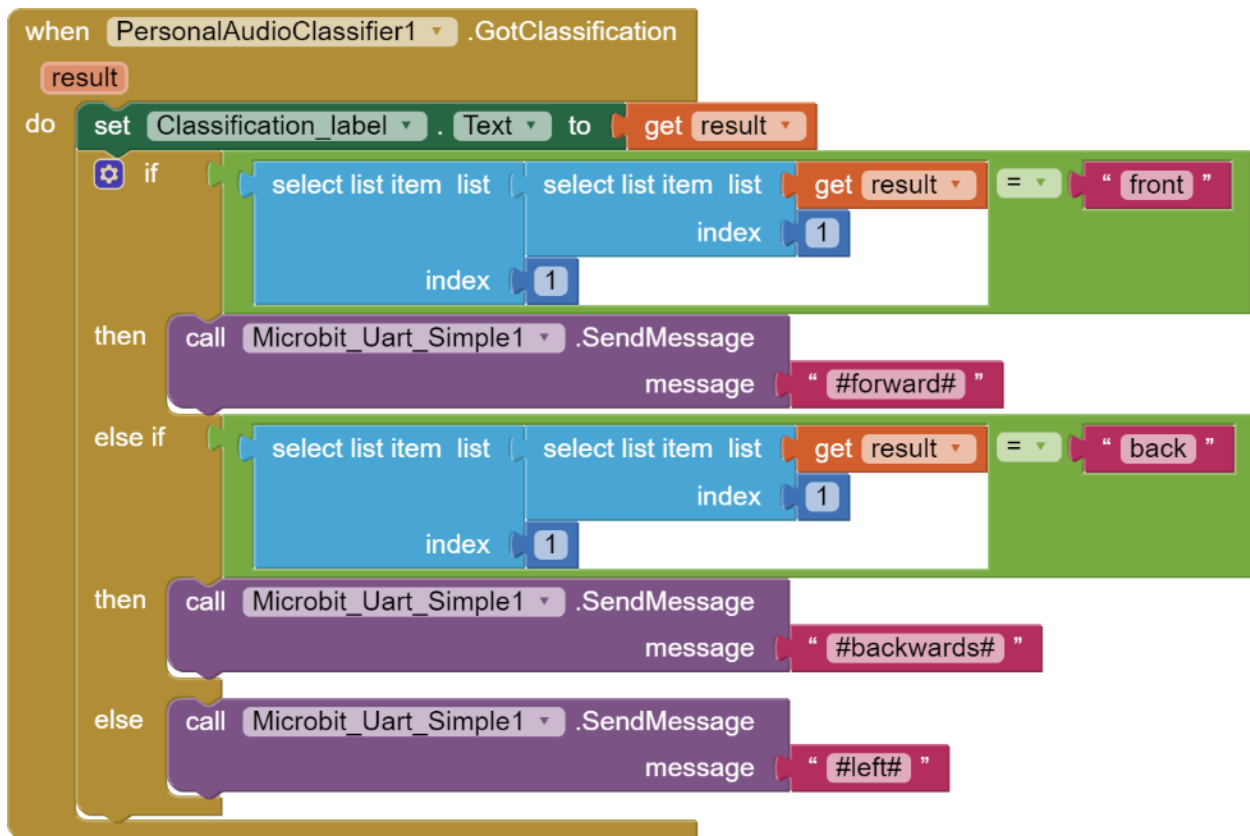
De volgende afbeelding toont het volledige script voor het voorbeeld van de classificatie die in dit document is gemaakt.



Wanneer u alle bovengenoemde stappen heeft voltooid, is de applicatie klaar om te worden geüpload en geïnstalleerd op uw smartapparaat. Ga naar het Build-menu en selecteer “Android App (.apk)” in het vervolgkeuzemenu om het proces van de productie van .apk-bestanden te starten. Dit kan een paar

minuten duren. Installeer vervolgens de applicatie op uw smartapparaat via de MIT AI2 Companion-applicatie.

**Belangrijke opmerking:** In sommige gevallen (vooral als het getrainde model te veel labels/categorieën heeft), werkt het getrainde model mogelijk niet goed wanneer het in de toepassing wordt geïntegreerd, wat tot valse resultaten leidt. Als uw leerlingen met een dergelijk probleem worden geconfronteerd, moedig ze dan aan om te experimenteren met een **kleiner** getraind model (met slechts 2 of 3 labels) en alternatieve scenario's te bedenken waarin deze AI-service nuttiger zou zijn (bijvoorbeeld het starten en stoppen van de auto). In het geval van een eenvoudiger getraind model zou het PersonalAudioClassifier-script er uit kunnen zien zoals weergegeven in Afbeelding 41.



Figuur 41: Een script voor het programmeren van Personal Audio Classifier wanneer het getrainde model slechts twee categorieën bevatte, zoals voor en achter.

### 3.7.3 Experiment 4

Om deze activiteit soepel bij uw leerlingen te introduceren, kunt u het bijbehorende werkbladdocument gebruiken (*Students\_worksheet\_for\_Activity\_4.pdf*). Voordat u dit doet, moet u ervoor zorgen dat ze het doel van deze activiteit begrijpen (dwz een getraind model evalueren door het in een applicatie te integreren en de beperkingen van AI ontdekken).

Nadat uw leerlingen de applicatie hebben gemaakt, moedigt u elk team aan om de applicatie van andere teams te testen. Moedig ze vervolgens aan om hun ervaringen in de plenaire vergadering te delen.

Enkele **vragen** die u kunt stellen om de dialoog op gang te brengen zijn:

- Werkte uw aanvraag correct?
- Denk je dat verschillende stemmen of verschillende uitspraken het resultaat beïnvloeden?
- Kunt u een parameter bedenken die tot een storing kan leiden?
- Wat zijn de risico's of gevaren van het gebruik van een getraind model dat fouten bevat?
- Soms kan een getraind model met te veel categorieën tot storingen leiden. Kunt u een alternatief scenario voor uw robotauto bedenken waarbij een kleiner getraind model (met minder labels/categorieën) nodig zou zijn?

Door deze activiteit zal uw leerling begrijpen dat:

- Er moet nog veel werk worden verzet voordat een AI-systeem op een meer natuurlijke manier met elkaar kan communiceren
- Verschillende uitspraken of stemtonen kunnen tot storingen leiden
- Er zijn ernstige risico's verbonden aan het gebruik van een AI-systeem in het dagelijks leven, als de dataset niet goed is getraind of vooroordelen bevat

## 3. 8 Activiteit 5: Introductie van het idee van maatschappelijke impact

### 3.8.1 Beschrijving

In deze activiteit maken de leerlingen kennis met het vijfde <sup>Grote</sup> Idee, namelijk Maatschappelijke Impact, door te reflecteren op de ervaringen die ze hebben opgedaan tijdens het uitvoeren van de voorgaande vier activiteiten. In het bijzonder zullen zij worden aangemoedigd om na te denken over de voordelen, de nadelen en de inherente risico's van het gebruik van AI-diensten en -instrumenten, evenals van het monitoren van gegevens en het nemen van beslissingen op basis van specifieke datasets. Deze activiteit kan afzonderlijk worden uitgevoerd of worden gecombineerd met de voorgaande vier. Op deze manier zullen ze zich bewust worden van verschillende ethische beslissingen waarmee rekening moet worden gehouden bij het ontwerpen en gebruiken van AI- en IoT-diensten en -technologieën.

Een aantal vragen over ethische kwesties zijn al in eerdere activiteiten aan bod gekomen. Hier zijn enkele aanvullende voorbeelden om de dialoog in deze richting op gang te brengen:

1) U kunt uw leerlingen aanmoedigen om een scenario te bedenken waarin de robotauto gevoeliger gegevens verzamelt (bijvoorbeeld afbeeldingen van mensen zodat de auto voetgangers kan herkennen) en deze gegevens naar de cloud of naar andere diensten verzendt.

- Wat zijn de voor- en nadelen van deze technologie?
- Met welke parameters moet rekening worden gehouden met betrekking tot de beveiliging van deze gegevens?

2) Moedig je leerlingen aan om na te denken over hoe een bevooroordeeld model de manier kan beïnvloeden waarop een intelligente robot 'denkt', wat kan leiden tot de constructie van bevooroordeelde representaties van de wereld. Moedig ze aan zich een scenario voor te stellen waarin een robotauto zonder bestuurder, die wordt gebruikt om vastzittende mensen te redden, is getraind in het herkennen van audio-noodsignalen die hij ontvangt, maar niet goed is getraind. Wat zouden de gevolgen zijn van deze verkeerde voorstelling van de werkelijkheid?

3) Stel je een geval voor waarin een robotauto wordt getraind om alleen mannenstemmen en alleen baritonmannenstemmen te herkennen. Hoe kan zo'n bevooroordeeld getraind model de levens van verschillende andere mensen beïnvloeden?

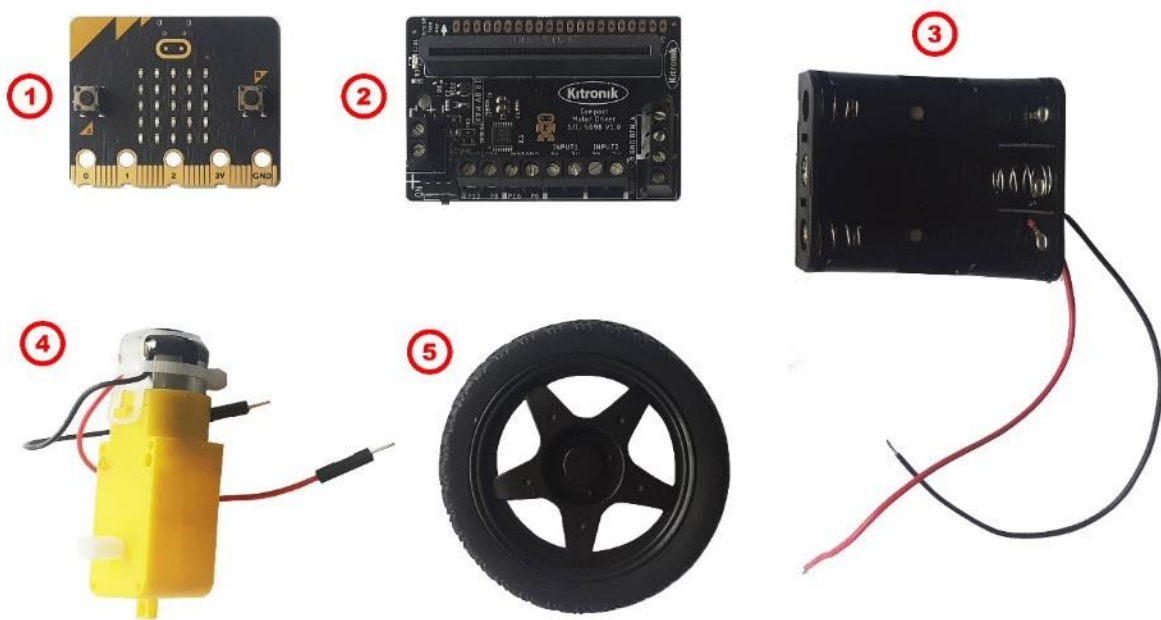
### 3. 9 Materiaal en hulpbronnen

| Type bron    | Titel                                      | Onderwerp  | Koppeling |
|--------------|--|--|-----------|
| PDF bestand  | T2.4_Het creëren van_de_robotauto          | Richtlijnen voor het maken van de robotauto  |           |
| PDF bestand  | T2.4_WarmUp_activiteiten_voor_de_robotauto | Programmeeractiviteiten opwarmen om vertrouwd te raken met de commando's voor de robotauto   |           |
| PDF bestand  | Circuitkaart_Activiteit1                   | Materiaal om leerlingen te helpen bij het maken van het circuit voor de eerste activiteit  |           |
| PDF bestand  | Halfbakken_Activiteit1                     | Document met een halfbakken oplossing van de code voor de 1 <sup>e</sup> activiteit  |           |
| PDF bestand  | T2.4_App_Inventor_Warm_Up                  | Een opwarmactiviteit om vertrouwd te raken met MIT App Inventor en een applicatie te maken voor het op afstand besturen van de robotauto                                       |           |
| .aia-bestand | Afstandsbediening_Microbit                 | Een App Inventor-bestand voor gebruik van het bestand dat is geproduceerd in het kader van de App Inventor Warm Up-activiteit en kan worden gebruikt voor de tweede activiteit |           |
| PDF bestand  | T2.4_Programmeren_de_robotauto             | Een document met richtlijnen voor het maken van het script om de robotauto te programmeren om de door de ontworpen applicatie ontvangen orders op te volgen                    |           |
| .aia-bestand | Robotic_Car_SpeechRecognizer               | Een App Inventor-bestand dat kan worden gebruikt voor de 4 <sup>e</sup> activiteit   |           |
| PDF bestand  | Studenten_Werkblad_voor_Activiteit_2       | Werkblad voor leerlingen om hen te   |           |

|             |                                      |  |  |
|-------------|--------------------------------------|--|--|
|             |                                      | helpen bij het implementeren van de tweede activiteit                                |  |
| PDF bestand | Studenten_Werkblad_voor_Activiteit_4 | Werkblad voor leerlingen om hen te helpen bij het implementeren van de 4e activiteit |  |

### 3.10 De hardware voor de robotauto

Figuur 42 toont de elektronische basiscomponenten die je nodig hebt voor het maken van de robotauto. In het bijzonder hebt u een BBC micro:bit- microcontroller ( **1** ), een Kitronik Compact Motor Driver ( **2** ), een 3AA (of 4AA) batterijhouder (bij voorkeur met vooraf aangesloten draden) ( **3** ), 2 DC-reductiemotoren ( bij voorkeur met vooraf bevestigde draden) ( **4** ) en 2 wielen ( **5** ).



Figuur 42: De elektronische componenten die nodig zijn voor het maken van de robotauto