



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.

Inleiding tot MIT App Inventor - het maken van een applicatie voor het op afstand besturen van de doe-het-zelf 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

1.1 Inleiding.....	3
1.2 Plannen van het ontwerp van de applicatie	3
1.3 Het ontwerpen van de applicatie.....	3
Het maken van de lay-out voor het plaatsen/schikken van de knoppen	5
1.4 Programmeren van de applicatie.....	15
Codering van de knop Verbinding verbreken (dwz Dis).	19
Codering van de knoppen voor het navigeren door de robotauto	20
1.5 Het bouwen van de applicatie	23
1.6 De applicatie koppelen met de robotauto.....	24

1. Een applicatie maken voor het op afstand besturen van de doe-het-zelf-robotauto

1.1 Inleiding

Dit document biedt een opwarmactiviteit waarmee u uw leerlingen kennis kunt laten maken met de MIT App Inventor-omgeving. Door deze activiteit leren de leerlingen hoe ze een applicatie kunnen maken waarmee ze de robotauto op afstand kunnen besturen, via een slim apparaat. Daarom, en met het oog op dit doel, zullen ze leren hoe ze de interface van de applicatie moeten ontwerpen en hoe ze de daarin opgenomen items kunnen programmeren. Voorafgaand aan deze activiteit wordt u ten zeerste aangeraden uw leerlingen te instrueren om het script te maken dat wordt beschreven in het bestand “T2.4_Programming_the_robotic_car.pdf” of om het naar de robotauto te downloaden met behulp van het bijbehorende .hex-bestand.

1.2 Plannen van het ontwerp van de applicatie

Voordat u verder gaat met het ontwerp van de applicatie, is het van cruciaal belang om op de hoogte te zijn van de componenten die moeten worden opgenomen. Een van de belangrijkste doelen is het vinden van een manier om verbinding en communicatie tot stand te brengen tussen ons slimme apparaat en onze robotauto. Om dat te doen, zullen we een aantal berichten instellen die door ons slimme apparaat worden verzonden/verzonden en via Bluetooth door onze robotauto worden ontvangen. Elke keer dat er een bericht wordt verzonden/verzonden door het slimme apparaat en wordt ontvangen door het micro:bit-bord, zal de robotauto op een andere manier corresponderen.

Met dit in gedachten moeten we een applicatie ontwerpen waarmee we:

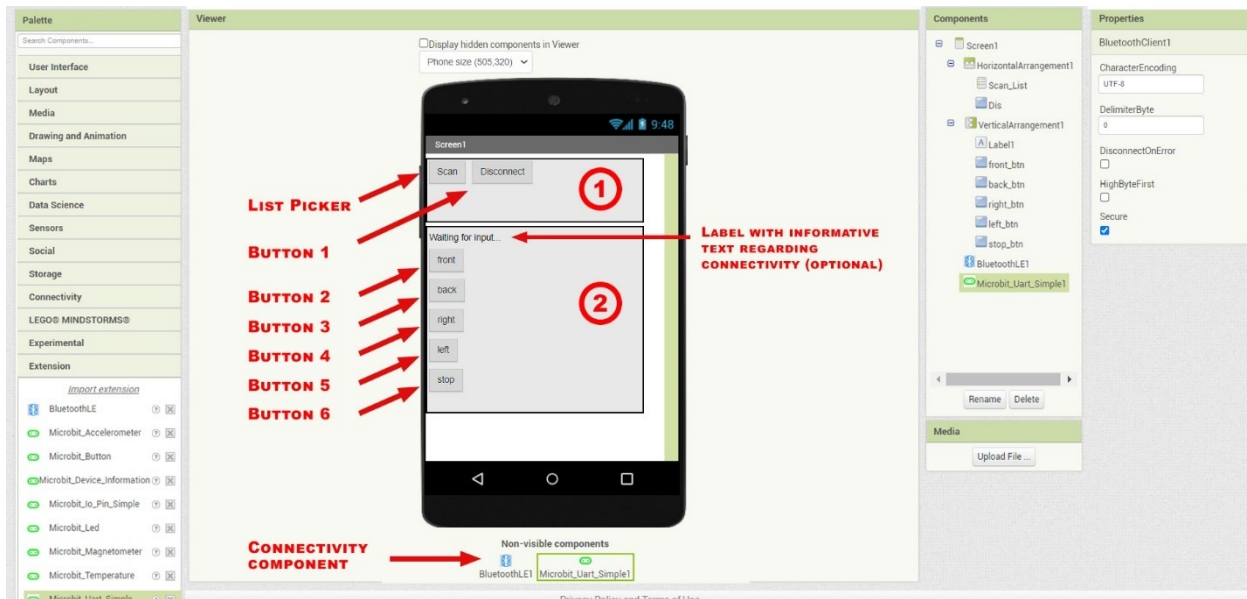
- controleer de beweging van onze robotauto. Om dat te doen hebben we 5 knoppen nodig: 4 daarvan zullen onze robotauto vooruit, achteruit, rechts en links bewegen, en 1 die elke uitvoerende beweging stopt.
- toestaan dat onze applicatie wordt verbonden via de Bluetooth van ons apparaat
- sta toe dat onze applicatie beschikbare Bluetooth-apparaten scant/zoekt en verbinding maakt met een gekozen apparaat
- ontkoppel onze applicatie van het verbonden Bluetooth-apparaat
- informeer ons optioneel over de huidige status van de connectiviteit

Al het bovenstaande helpt ons een concreet idee te creëren over de uitkomst van het ontwerpproces

1.3 Het ontwerpen van de applicatie

Ontwerp is een vrij vrij proces en grotendeels gebaseerd op de esthetiek van de maker. De volgende instructies zijn indicatief en presenteren een nogal vereenvoudigde versie van het uiterlijk van de interface die onze applicatie kan hebben.

Figuur 1 toont een voorbeeld van de interface die we zullen creëren op basis van de behoeften die we in de vorige sectie hebben vastgelegd.



Figuur 1: Een voorbeeld van de interface die u gaat ontwerpen

Om alle componenten op ons scherm beter te organiseren, willen we twee lay-outs maken. Eén die de Scan/zoek-knop en de knop Verbinding verbreken zal hosten en ons in staat zal stellen deze op een rij te rangschikken (dwz naast elkaar) **(1)**, en één die de navigatieknoppen zal hosten en ons in staat zal stellen deze te ordenen ze in een kolom (dwz onder elkaar) **(2)**.

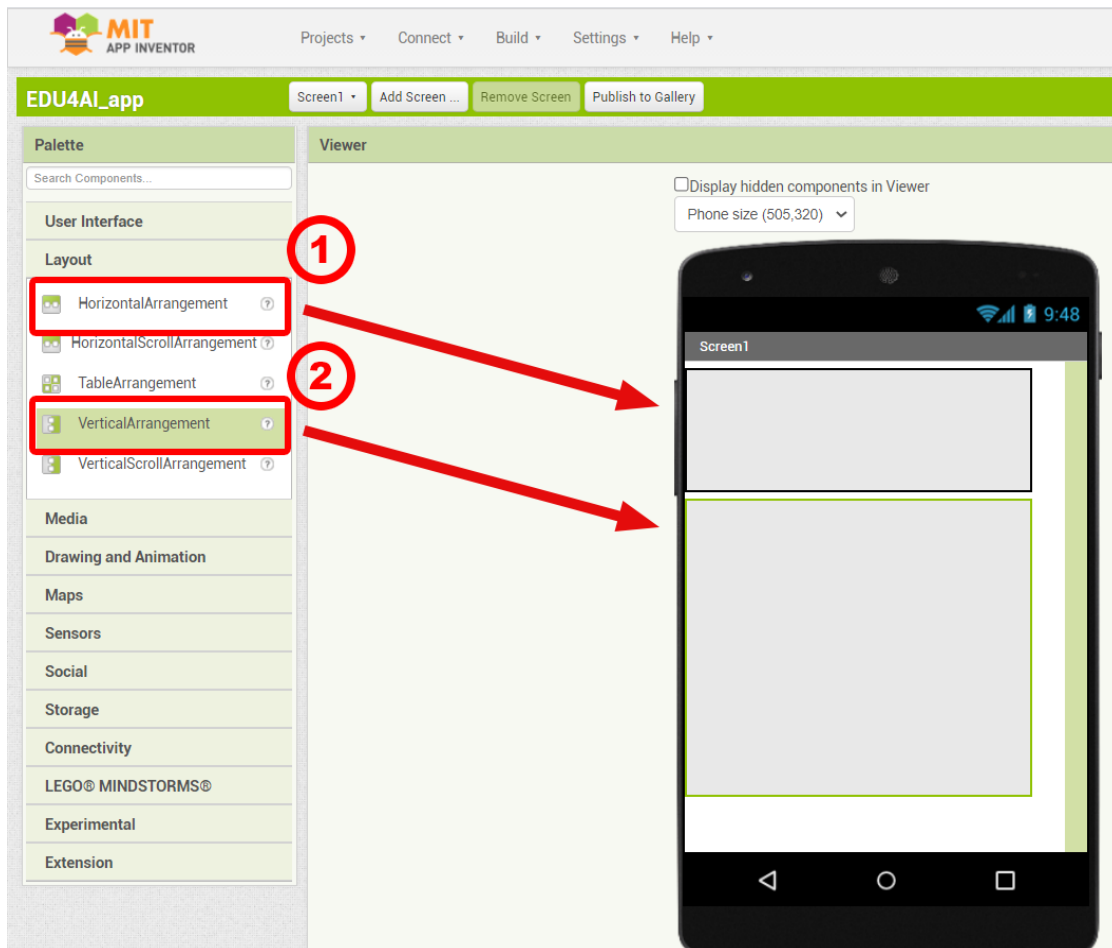
Laten we de functionaliteit van elke knop eens nader bekijken:

- 1) **Scannen:** De knop “Scannen” zou een lijst moeten openen met alle beschikbare Bluetooth Low Energy-apparaten in de omgeving. Uit die lijst moet de gebruiker het Bluetooth-adres van Micro:bit kiezen. Vervolgens wordt de verbinding automatisch tot stand gebracht. Deze knop verschilt omdat deze de gebruiker doorverwijst naar een lijst met alle beschikbare Bluetooth-verbindingen. Om deze functie in te schakelen, zullen we een knop 'Lijstkiezer' toevoegen, die verderop in deze richtlijn verder wordt beschreven.
- 2) **Verbinding verbreken:** Wanneer de knop “Verbinding verbreken” wordt ingedrukt, wordt de verbinding tussen de micro:bit en het smartapparaat van de gebruiker gedeactiveerd.
- 3) **Navigatieknoppen (voorkant, achterkant, enz.):** Wanneer een van deze knoppen wordt ingedrukt, beweegt onze robotauto in de overeenkomstige richting.

Tip: Tijdens deze fase kun je je leerlingen adviseren/aanmoedigen om een schets of diagram te maken van de interface en de daarin opgenomen componenten, of/en een lijst met alle benodigde items. Op deze manier zullen zij de stappen naar de realisatie van de huidige opgave beter kunnen organiseren.

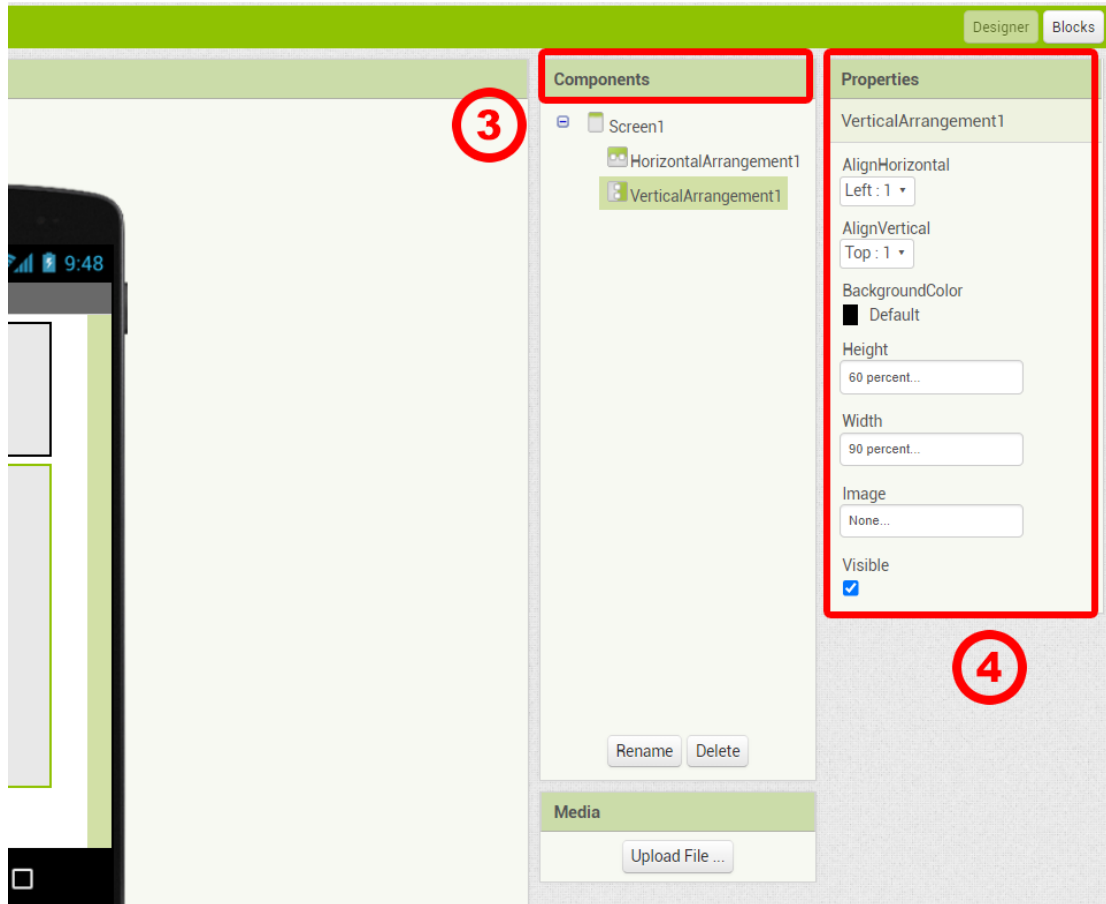
Het maken van de lay-out voor het plaatsen/schikken van de knoppen

De eerste stap op weg naar het maken van onze applicatie is het instellen van de twee lay-outs waarin alle benodigde knoppen en labels worden gehost en gerangschikt. Om dat te doen zullen we twee lay-outitems toevoegen, namelijk een horizontale lay-out die de knoppen ListPicker en de Disconnect-knoppen zal hosten, en een verticale lay-out die de navigatieknoppen zal hosten. Daarom slepen we vanaf het tabblad “Lay-out” de componenten “ HorizontalArrangement ” (1) en “ VerticalArrangement ” (2) en laten we ze op het scherm vallen (*Figuur 2*).



Figuur 2: De twee lay-outitems slepen en neerzetten op het scherm

(4) een aantal eigenschappen aanpassen, zoals de hoogte en de breedte . Om dat te doen moeten we het overeenkomstige onderdeel selecteren uit de lijst Componenten (3) . In het voorbeeld in Figuur 3 hebben we de breedte van de component “ HorizontalArrangement ” ingesteld op 90 procent, en de breedte en de hoogte van de component “ VerticalArrangement ” op respectievelijk 60 en 90 procent.



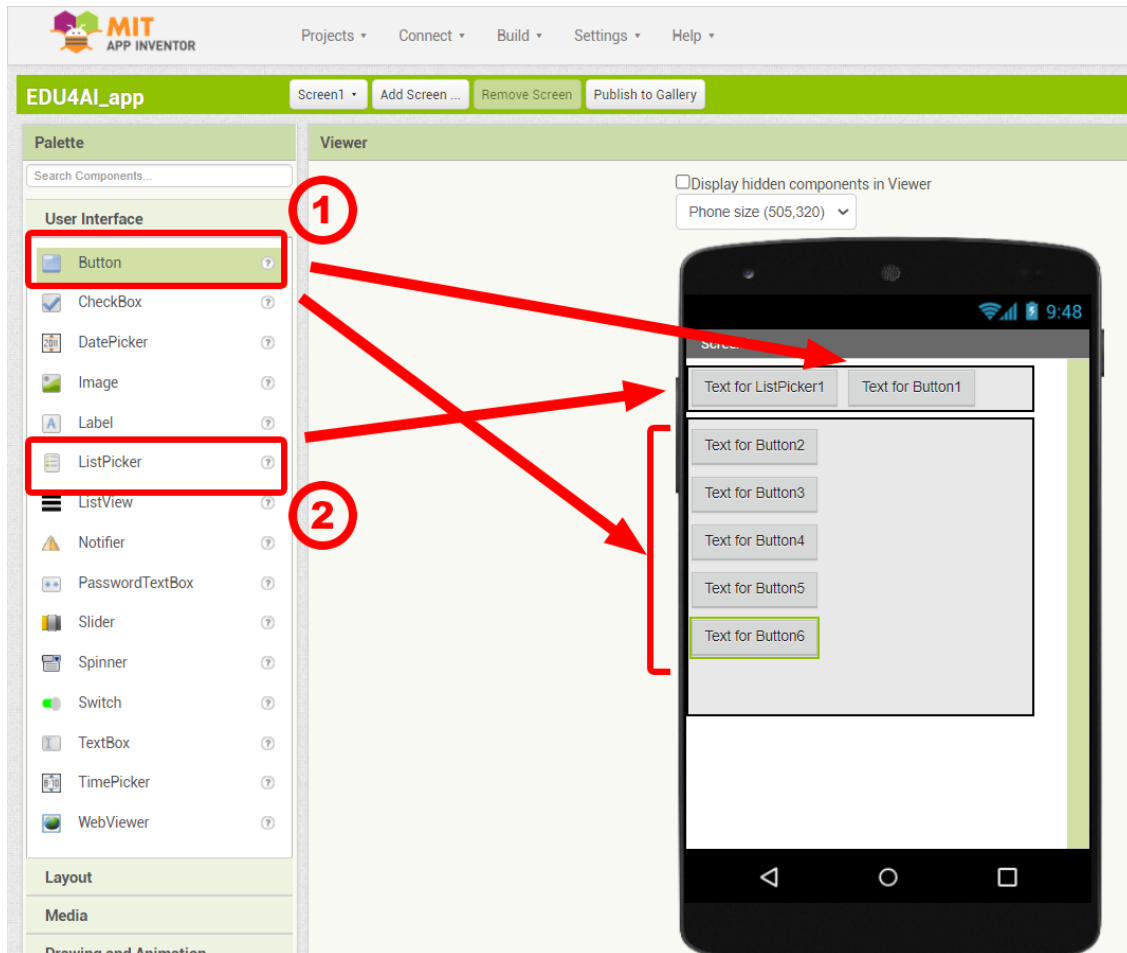
Figuur 3: De eigenschappen van elk onderdeel aanpassen

Tip: Als u wilt, kunt u de bovengenoemde eigenschappen herzien en wijzigen nadat u de knoppen in de lay-outcomponenten hebt toegevoegd.

ListPicker en de knoppen toevoegen

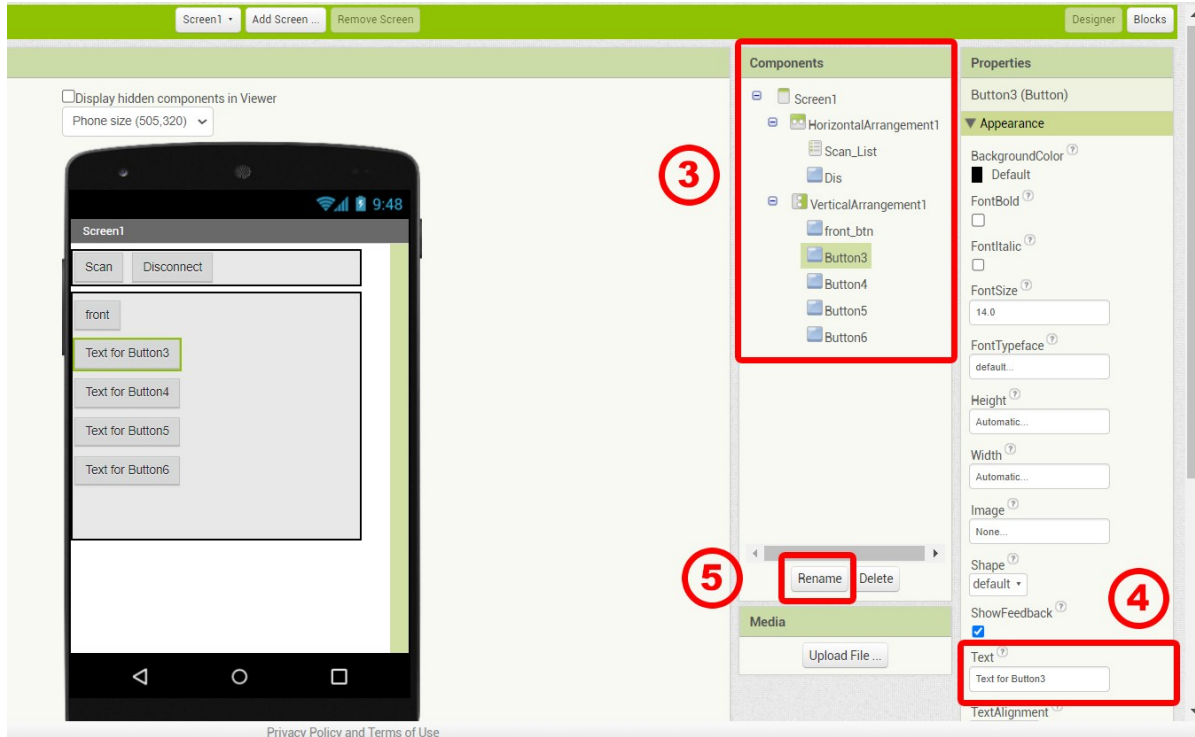
De volgende stap is het toevoegen van de ListPicker en de knoppen op het scherm. Om dat te doen gaan we naar het submenu “Gebruikersinterface” en vanuit het gedeelte “Palet” slepen we 7 items naar het scherm, namelijk een “Lijstkiezer” en 6 knoppen. De ListPicker wordt gebruikt om alle beschikbare Bluetooth-apparaten te zoeken/scannen en te onthullen.

Concreet slepen we het item “ListPicker” (2) en één knop (1) (dat wil zeggen de knop voor het verbreken van de verbinding) naar de lay-out “HorizontalArrangement”, en daarna slepen we nog 5 knoppen (1) (dat wil zeggen de navigatieknoppen) en zet ze neer in de lay-out “VerticalArrangement” (Figuur 4).



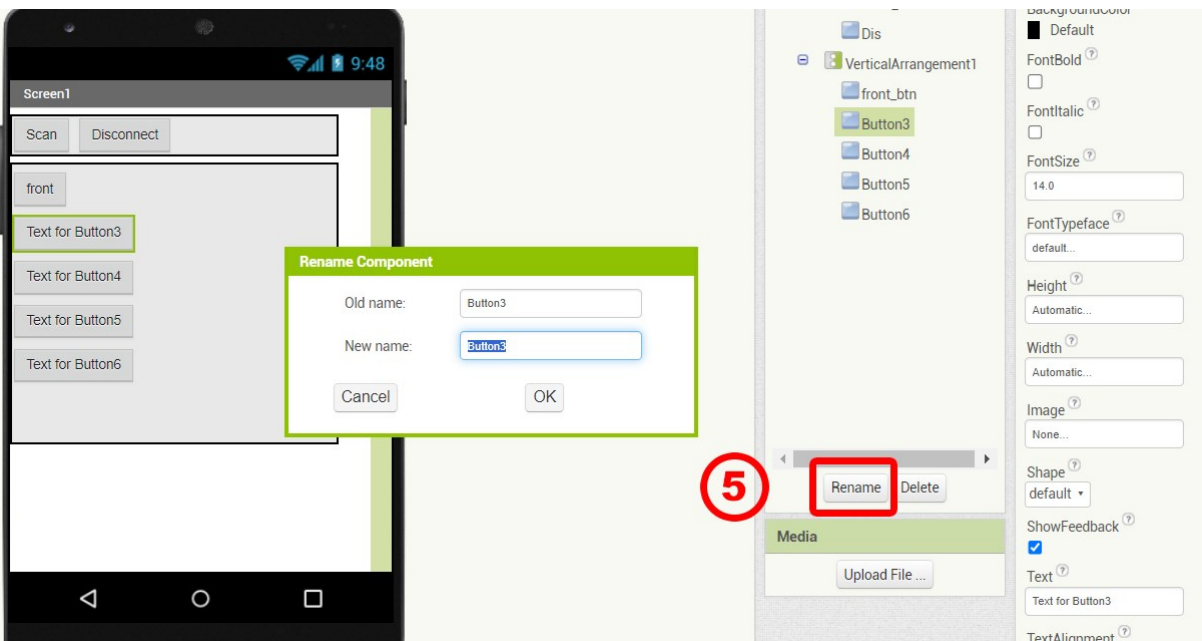
Figuur 4: Een ListPicker- item en 6 knoppen op het scherm toevoegen

Om het uiterlijk van onze interface te optimaliseren, kunnen we de naam van elke knop wijzigen via het tabblad “Tekst” (4) in het menu “Eigenschappen”. Om dat te doen, moeten we het overeenkomstige item selecteren uit de lijst “Componenten” (3) (knop 3 in ons voorbeeld) en vervolgens de naam handmatig wijzigen (Afbeelding 5).



Figuur 5: De inhoud van de tekst van de knoppen wijzigen

Een andere goede gewoonte is om de namen van de knoppen te wijzigen. Dit zal later ook de codeerfase vergemakkelijken. U kunt dat doen door op de knop “Hernoemen” (5) te klikken en door de nieuwe naam in te voegen in het tekstvak “Nieuwe naam” in het pop-upmenu (Figuur 5, Figuur 6).



Figuur 6: De naam van het knoponderdeel wijzigen

Tip: De namen van de knoppen zijn indicatief en hebben geen invloed op de functionaliteit ervan. Het is echter een goede gewoonte om namen te gebruiken die betekenisvol zijn voor het hele proces (geef bijvoorbeeld de naam “ front_btn ” aan de knop waarmee de auto vooruit beweegt, “ back_btn ” aan degene die de auto achteruit beweegt, enz.). . In ons voorbeeld hernoemen we de componenten als volgt:

Onderdeel	Tekst naam (gewijzigd van het tabblad Tekst in eigenschappen)	Nieuwe naam (gewijzigd van Naam wijzigen in componentenlijst)
Lijstkiezer	Scannen	Scan_Lijst
Knop1	Loskoppelen	Dis
Knop 2	voorkant	front_btn
Knop 3	rug	terug_btn
Knop 4	rechts	rechts_btn
Knop5	links	links_btn
Knop6	stop	stop_btn

Belangrijke opmerking : gebruik niet hetzelfde woord voor tekstnaam en knopnaam, omdat dit een storing in App Inventor veroorzaakt, waardoor het niet meer mogelijk is de applicatie te bouwen.

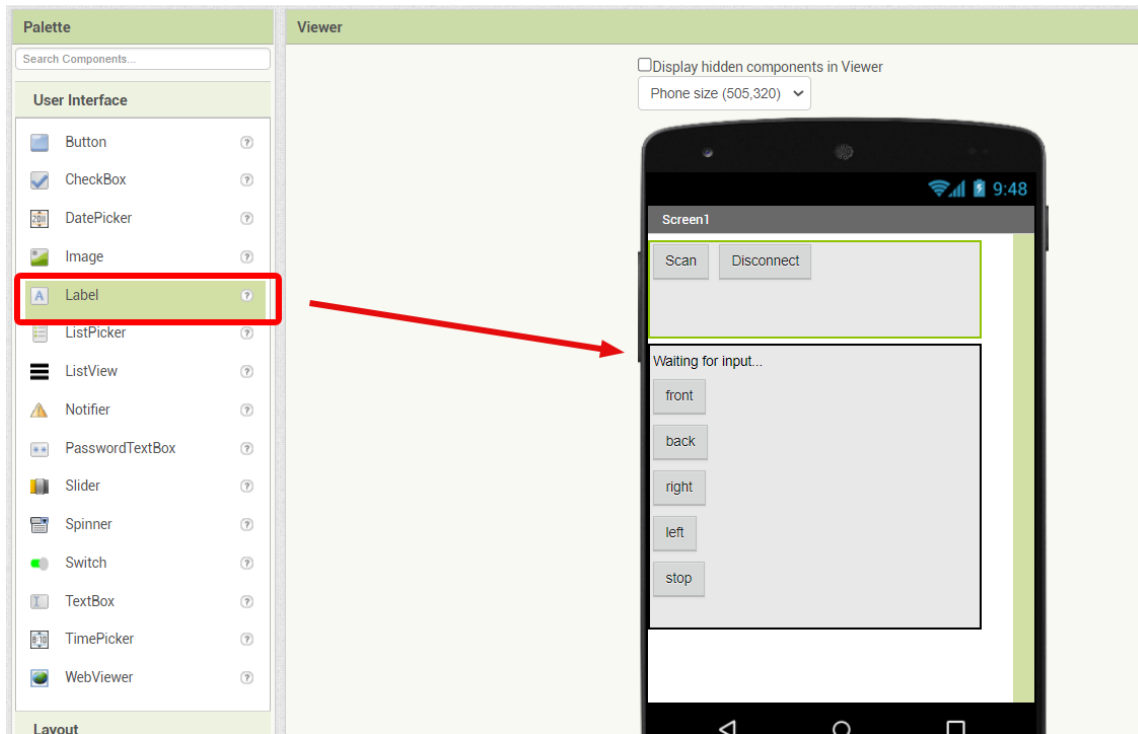
Samenvattend zal de ListPicker “Scan” zoeken naar beschikbare Bluetooth-apparaten, terwijl de knop “Verbinding verbreken” de verbinding tussen ons smartapparaat en de Bluetooth-module verbreekt. De overige knoppen worden gebruikt om door onze robotauto te navigeren.

Tip: Houd er rekening mee dat bovengenoemde richtlijnen indicatief zijn. Experimenteer gerust met verschillende vormen en kleuren voor elke knop, of zelfs met verschillende arrangementen, waardoor een uniekere en visueel aantrekkelijker interface ontstaat.

Een tekstmelding toevoegen

De volgende stap is het toevoegen van een tekstlabel aan ons scherm. Deze stap is niet verplicht, maar wel erg handig, omdat deze ons informeert of de verbinding met ons smartapparaat tot stand is gebracht of niet. Zoek “Label” in het submenu “Gebruikersinterface” en sleep het naar het scherm (*Figuur 7*).

Wijzig de tekstinhoud in 'Wachten op invoer' (of iets soortgelijks naar uw voorkeur, op dezelfde manier waarop u de tekstinhoud voor de overige componenten hebt gewijzigd).



Figuur 7: Een label toevoegen voor informatieve doeleinden met betrekking tot connectiviteit

Extensies toevoegen

De volgende stap is het toevoegen van enkele componenten die de verbinding tussen onze applicatie en de robotauto mogelijk maken. Om dat te doen, moeten we de volgende extensies aan onze applicatie toevoegen; de BluetoothLE- extensie en de Microbit_Uart_Simple . De eerste wordt gebruikt voor het tot stand brengen van de Bluetooth-verbinding tussen ons smartapparaat en de Micro:bit, terwijl de tweede voor het verzenden van de juiste berichten nadat de verbinding tot stand is gebracht.

Om deze extensies te kunnen gebruiken, moet u ze lokaal naar uw computer downloaden. Om dat te doen, klikt u hier <https://mit-cml.github.io/extensions/> en downloadt u het *BluetoothLE.aix*- bestand en het *Microbit.aix*- bestand naar uw computer (Figuur 8).

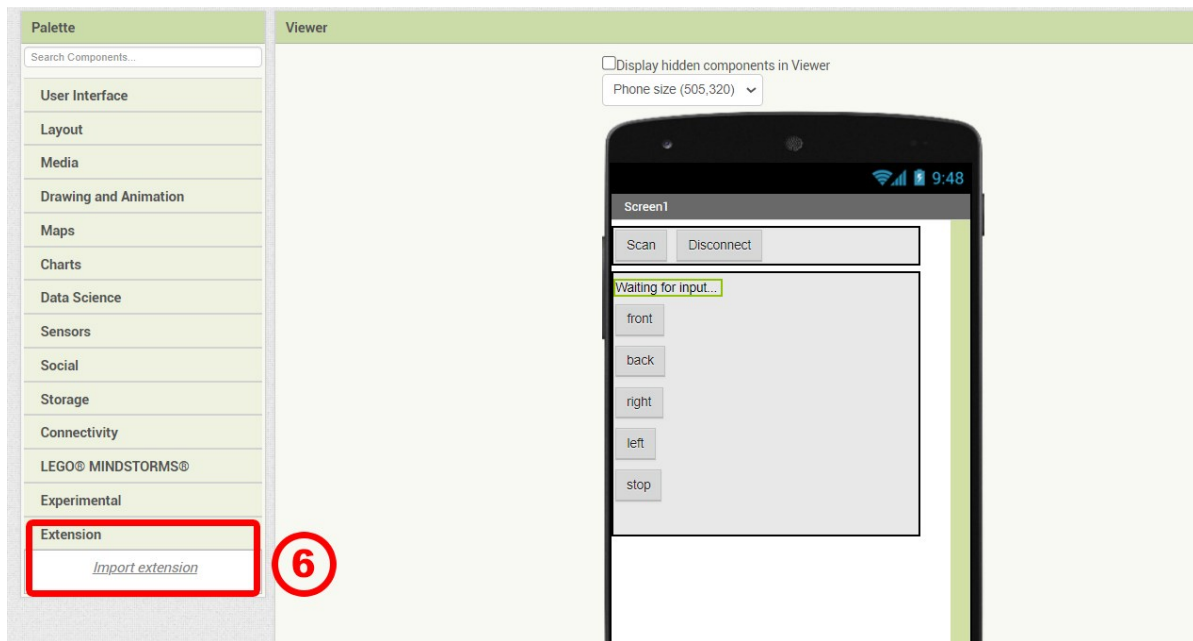
MIT APP INVENTOR Home Directory Documentation

Supported:

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

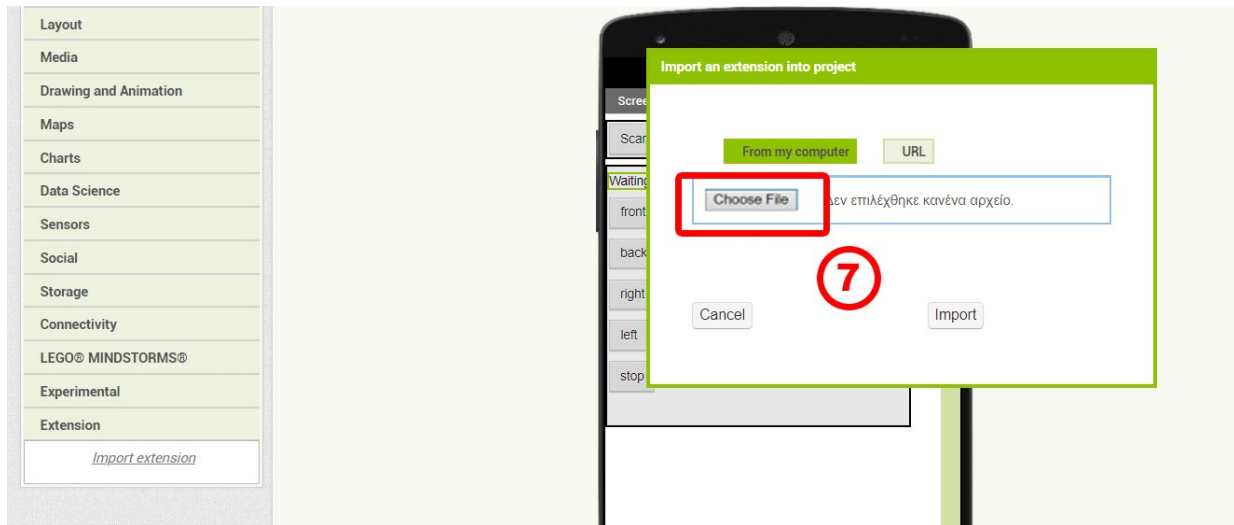
Figuur 8: Zoeken naar de te downloaden extensies

Na het downloaden van de extensies keert u terug naar App Inventor. Klik in de sectie Palet op het tabblad Extensie en klik vervolgens op de selectie Extensie importeren (6) (Figuur 9)



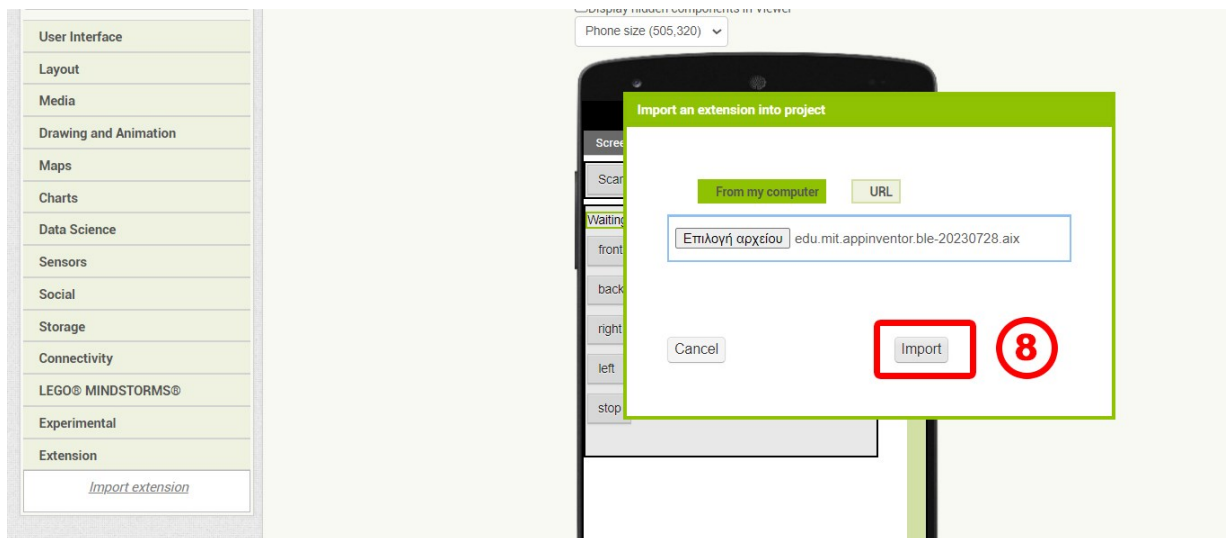
Figuur 9: De selectie van de importextensie

Klik in het pop-upmenu op de knop *Bestand kiezen* (7) om in uw lokale map te zoeken en de gedownloade extensie te selecteren (Figuur 10). Zorg ervoor dat " *Vanaf mijn computer* ", gelegen boven de knop *Bestand kiezen* , is geselecteerd.



Figuur 10: Klik op de knop Bestand kiezen om naar het extensiebestand in uw lokale map te zoeken

Nadat u het extensiebestand hebt gevonden en geselecteerd, klikt u op de knop Importeren (8) (Figuur 11).



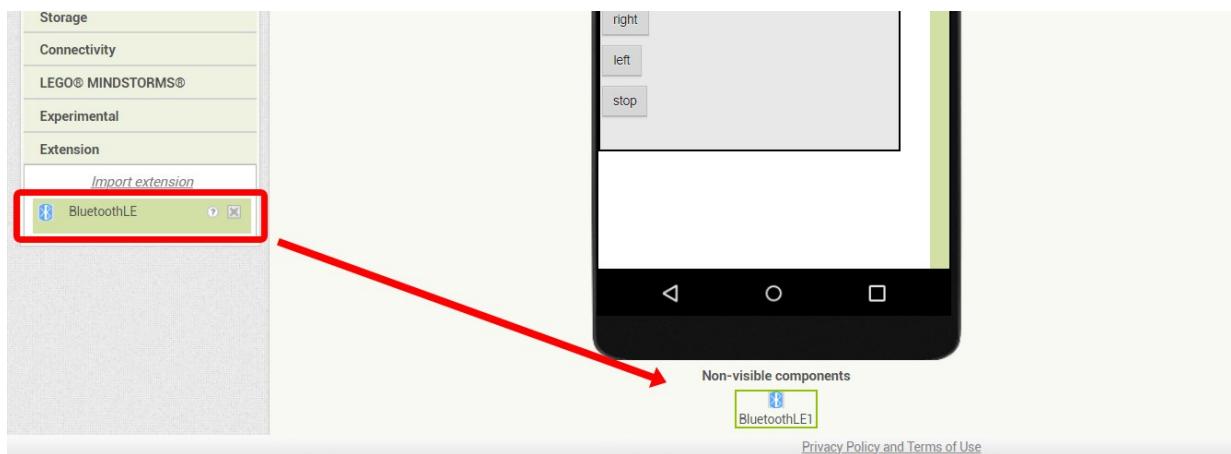
Figuur 11: Klik op de knop Importeren om de geselecteerde extensie te importeren

Na een tijdje zal de nieuwe extensie verschijnen onder Extensie importeren op het tabblad Extensie (in Afbeelding 12 is bijvoorbeeld de BluetoothLE- extensie geïmporteerd)



Figuur 12: De BluetoothLE- extensie is geïmporteerd

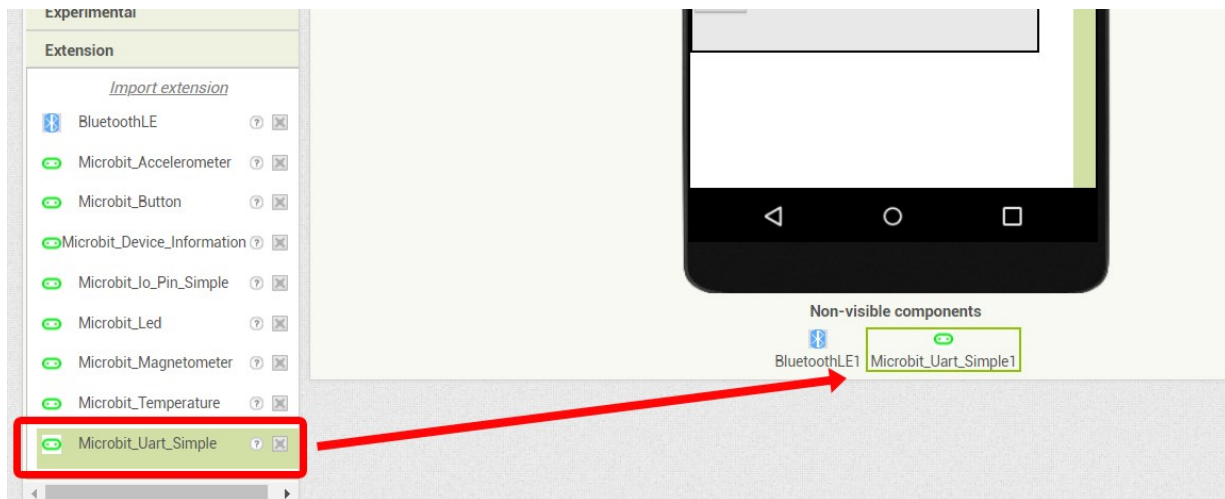
Om deze extensie aan de ontworpen applicatie toe te voegen, sleept u deze naar het ontwerpgebied (Figuur 13). De verlengstukken zijn normaal gesproken niet-zichtbare componenten. Daarom verschijnen deze componenten onder het ontwerpgebied, in het gedeelte 'Niet-zichtbare componenten'.



Figuur 13: De BluetoothLE- extensie naar het ontwerpgebied slepen en neerzetten

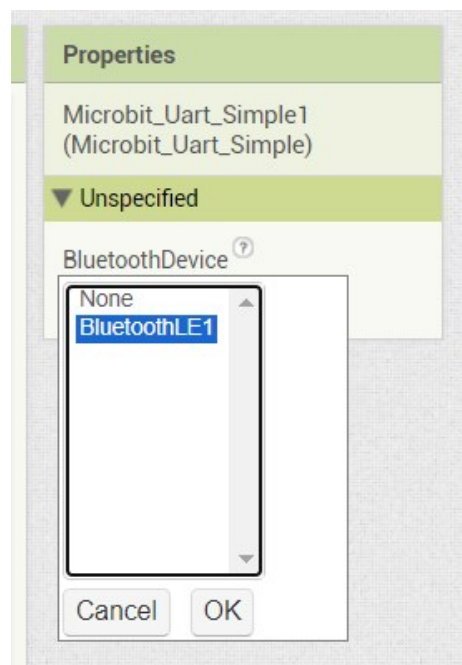
Herhaal hetzelfde proces om de Microbit_Uart_Simple- extensie te importeren.

Belangrijke notitie: na het kiezen en importeren van het *Microbit.aix*- bestand zult u merken dat er verschillende extensies verschijnen onder het tabblad Extensie. Voor deze activiteit hoeft u alleen de Microbit_Uart_Simple- extensie naar het ontwerpgebied te slepen en neer te zetten (Afbeelding 14).



Figuur 14: Alleen de Microbit_Uart_Simple is nodig om te slepen en neer te zetten

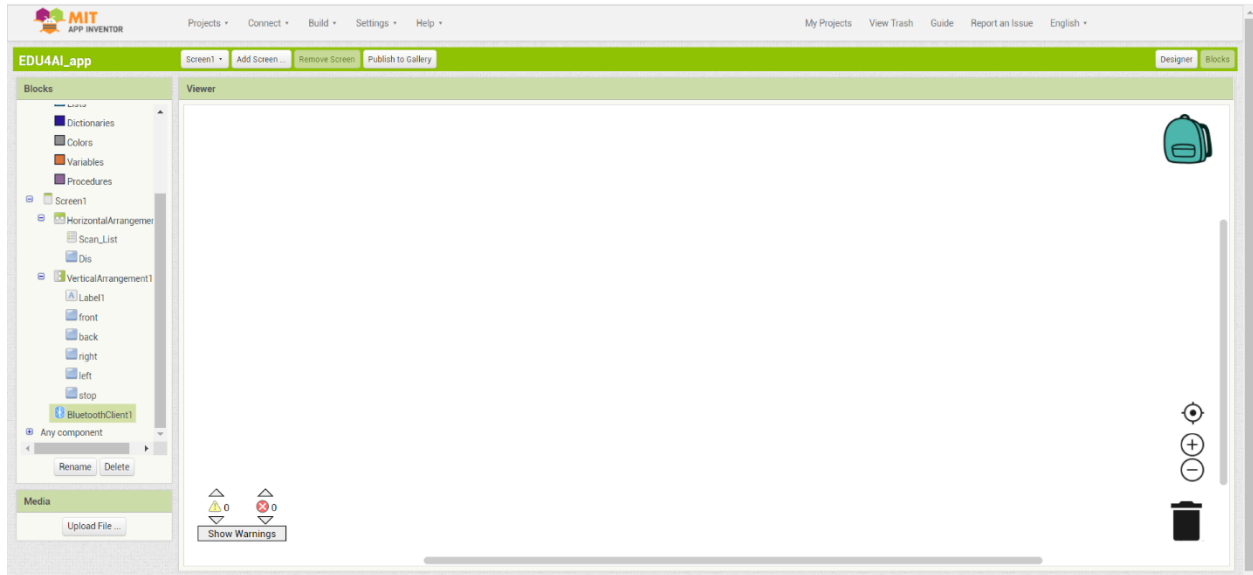
Na het importeren van de Microbit_Uart_Simple- extensie gaat u naar het menu Eigenschappen en selecteert u in het Bluetooth-apparaatveld BluetoothLE1 in het zwevende menu.



Nadat we alle benodigde uitbreidingen hebben toegevoegd, zijn we klaar om verder te gaan met het programmeren van de applicatie.

1.4 Programmeren van de applicatie

Om onze applicatie te laten werken, moeten we bepalen hoe de toegevoegde componenten zullen functioneren door te specificeren welke acties ze zullen activeren wanneer ze worden ingedrukt (op het scherm van onze applicatie). Om dat te doen moeten we naar het menu Blokken gaan en onze code in de viewerruimte creëren (Afbeelding 21) door de geschikte blokken met opdrachten te slepen en samen te stellen.



Figuur 15: Blokmenu van App Inventor

Codering van de ListPicker- component (dwz Scan_List).

We beginnen met het coderen van de ListPicker- component, of de “ Scan_List ”-component, zoals genoemd in ons voorbeeld. Het volgende diagram toont de bewerkingen/acties die we willen uitvoeren wanneer op de ListPicker- component wordt gedrukt, en de bijbehorende **gebeurtenis** commando dat voor dit doel moet worden toegepast. Laten we dus eens kijken hoe deze codes zullen worden gestructureerd.



Vóór het plukken commando

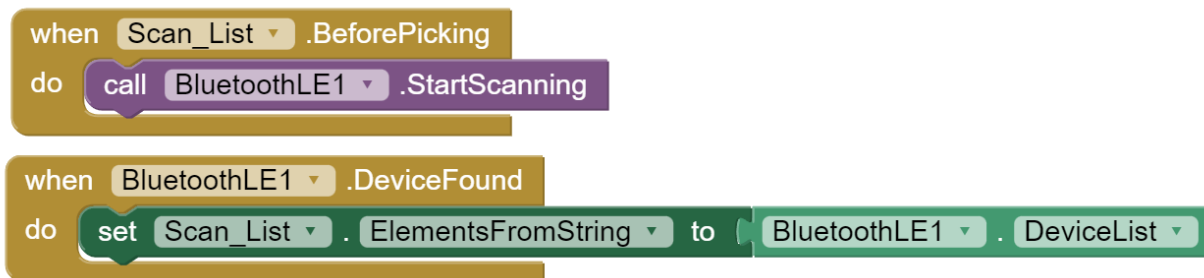
Klik op het item “ Scan_List ” (1) en selecteer in het zwevende menu het gebeurteniscommando “When Scan_List Vóór het plukken ” (2) (Figuur 16).

Wanneer Scan_List wordt ingedrukt, zou er een lijst met beschikbare Bluetooth-apparaten moeten verschijnen. “ BeforePicking ” betekent dat we nog geen van de weergegeven Bluetooth-apparaten hebben gekozen.

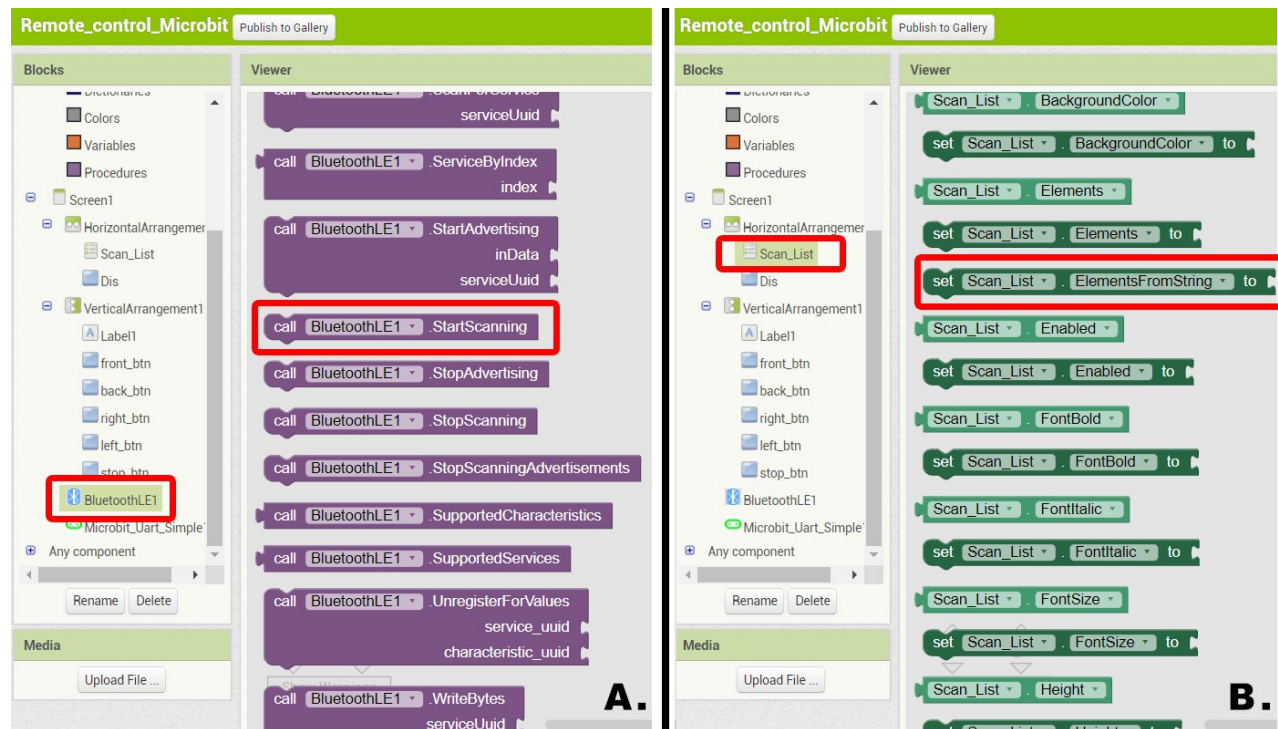


Figuur 16: Klikken op het item Scanlijst en selecteren van de benodigde opdracht in het zwevende menu

Om elementen uit onze lijst te kunnen kiezen, moet de lijst in eerste instantie gevuld zijn met alle beschikbare Bluetooth-apparaten. (De lijst is in het begin leeg, dus voordat we iets kiezen, moet deze eerst gevuld worden). Om dat te doen, moeten we beginnen met scannen naar BLE-apparaten. Vervolgens wordt de lijst gevuld met elk beschikbaar apparaat dat wordt gevonden:



Opmerking: De blokken met opdrachten die in het bovenstaande script worden gebruikt, kunnen op dezelfde manier worden gevonden door op het overeenkomstige item te klikken (Scan_List of Bluetooth LE1) en door de relevante blokken te vinden in het overeenkomstige zwevende menu (enkele voorbeelden worden weergegeven in *Figuur 17*).

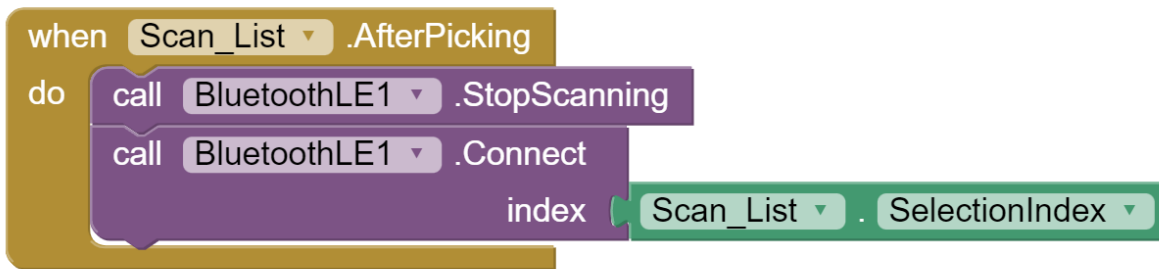


Figuur 17: Enkele blokopdrachten vinden

Na het plukken commando

Nu moeten we bepalen wat er gebeurt als de gebruiker de micro:bit's selecteert Bluetooth-adres om verbinding te maken tussen de applicatie en de robotauto. Om dat te doen, hebben we het gebeurteniscommando **"When Scan_List AfterPicking"**.

Binnen in de **gebeurtenis commando** we zullen enkele opdrachten plaatsen die de actie bepalen die zal worden uitgevoerd nadat we een Bluetooth-adres uit de lijst hebben gekozen. Daarom zullen we de opdracht **"call BluetoothLE1.StopScanning"** gebruiken , zodat ons apparaat stopt met zoeken naar andere Bluetooth LE-apparaten. Vervolgens voegen we de opdracht **"call BluetoothLE1.Connect"** toe , en aan de rechterkant klikken we de opdracht **" Scan_List.SelectionIndex "** uit , zodat de applicatie kan worden verbonden met het geselecteerde Bluetooth-apparaat.



Het label coderen (dwz Label1)

Om er zeker van te zijn dat de Bluetooth-verbinding tot stand is gebracht, zullen we de applicatie zo programmeren dat hij een relevante melding verzendt. Dit wordt gedaan door het Label-item te programmeren en door de volgende codeblokken te gebruiken:

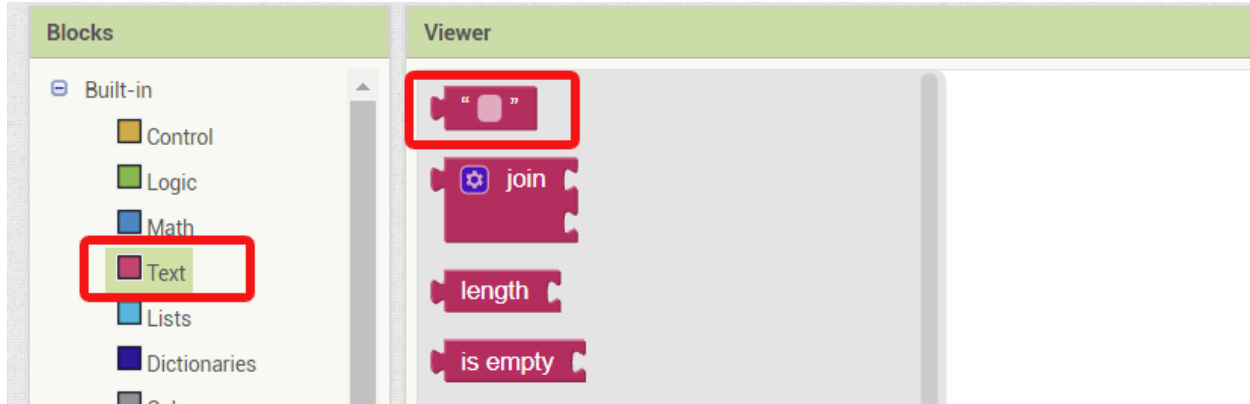


In het bijzonder gebruiken we de optie "set Label1.Text to" commando en aan de rechterkant klikken we een **tekstblok** "", waarin we "Verbinding tot stand gebracht" typen.

Als de verbinding tot stand is gebracht, verschijnt de melding 'Verbinding tot stand gebracht' op het scherm, ter vervanging van het bericht 'Wachten op invoer...' (Figuur 7). Als de verbinding niet succesvol tot stand is gebracht, verandert het bericht "Wachten op invoer..." niet.

Dus nadat de selectie van het Bluetooth-apparaat is gemaakt (namelijk de Bluetooth van de micro:bit), zal de BluetoothLE- component proberen verbinding te maken met het voorkeursadres (dwz de micro:bit die de robotauto gebruikt). Als de poging succesvol is, verandert de tekst van het label "Wachten op invoer..." in "Verbinding tot stand gebracht!".

Opmerking: het tekstinvoerblok is te vinden in het zwevende menu van het tabblad "Tekst" (Figuur 18).



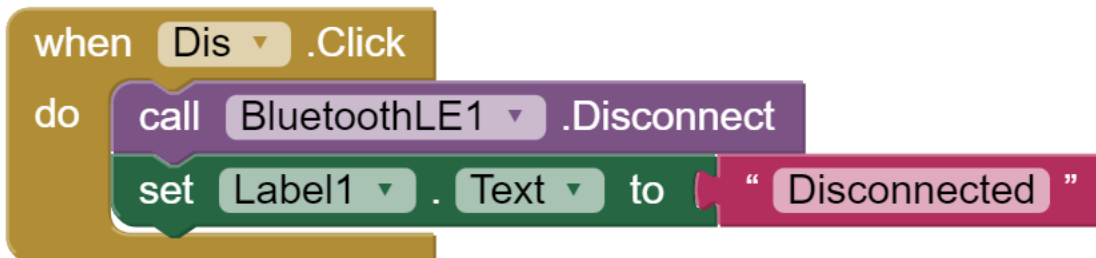
Figuur 18: Het tekstinvoerblok vinden

Codering van de knop Verbinding verbreken (dwz Dis).

In deze stap programmeren we de BluetoothLE- component om de verbinding te verbreken wanneer de knop "Verbinding verbreken" wordt ingedrukt. De tekst van het label zal ook veranderen in "Verbindingstime-out".

Voor deze coderingsstap hebben we het gebeurteniscommando "When Dis .Click...do" nodig .

Dus, via het volgende script, **wanneer** de knop "Verbinding verbreken" wordt ingedrukt, krijgt de applicatie de opdracht om **de BluetoothLE te bellen** en **koppel** deze los van het vooraf geselecteerde apparaat. We hebben ook **de tekst van het label ingesteld op "Disconnected"** om de gebruiker te informeren dat de verbinding is verbroken.



Let op: Het script voor het activeren van de knoppen "Scannen" en "Verbinding verbreken" kan lastig zijn voor uw leerlingen. Daarom kun je ze, afhankelijk van hun niveau, deze delen van het script aanreiken en ze het volgende deel van het script grondig leren.

Codering van de knoppen voor het navigeren door de robotauto

De laatste stap is het programmeren van de navigatieknoppen. Zoals eerder in dit document vermeld, verzendt ons slimme apparaat (via Bluetooth) een specifiek bericht naar het micro:bit-bord van onze robotauto wanneer er op een navigatieknop wordt gedrukt. Wanneer dit bericht wordt ontvangen, zal de robotauto zich dienovereenkomstig gedragen, door een specifieke beweging uit te voeren (dwz vooruit, achteruit enz.). De volgende tabel laat zien welk bericht wordt verzonden wanneer een specifieke knop wordt ingedrukt:

<i>Knop die wordt ingedrukt</i>	<i>Woord dat wordt verzonden</i>
"voorkant"	"#vooruit#"
"rug"	"#achteruit#"
"rechts"	"#rechts#"
"links"	"#links#"
"stop"	"#stop#"

Belangrijke opmerking : elk bericht dat we willen verzenden moet beginnen en eindigen met het symbool “#” , zodat de micro:bit de grenzen van het bericht kan onderscheiden .

Wanneer we bijvoorbeeld het woord “#forward#” verzenden, zal de micro:bit de gelijkstroommotoren vooruit laten draaien.

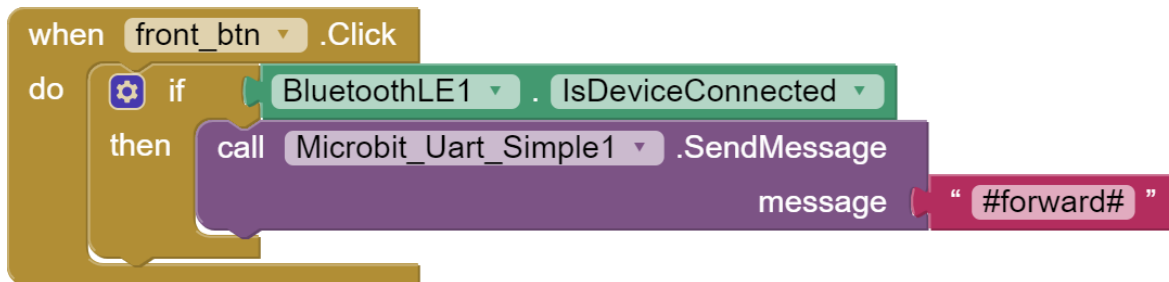
Voor dit deel van het script en voor elk van de knoppen hebben we het gebeurteniscommando “**When .Click...do**” en een “ **if..then** ” nodig. voorwaarde.

Opmerking: de “als-dan”-blokvoorwaarde kunt u vinden onder het tabblad “Controle” (*Figuur 19*).



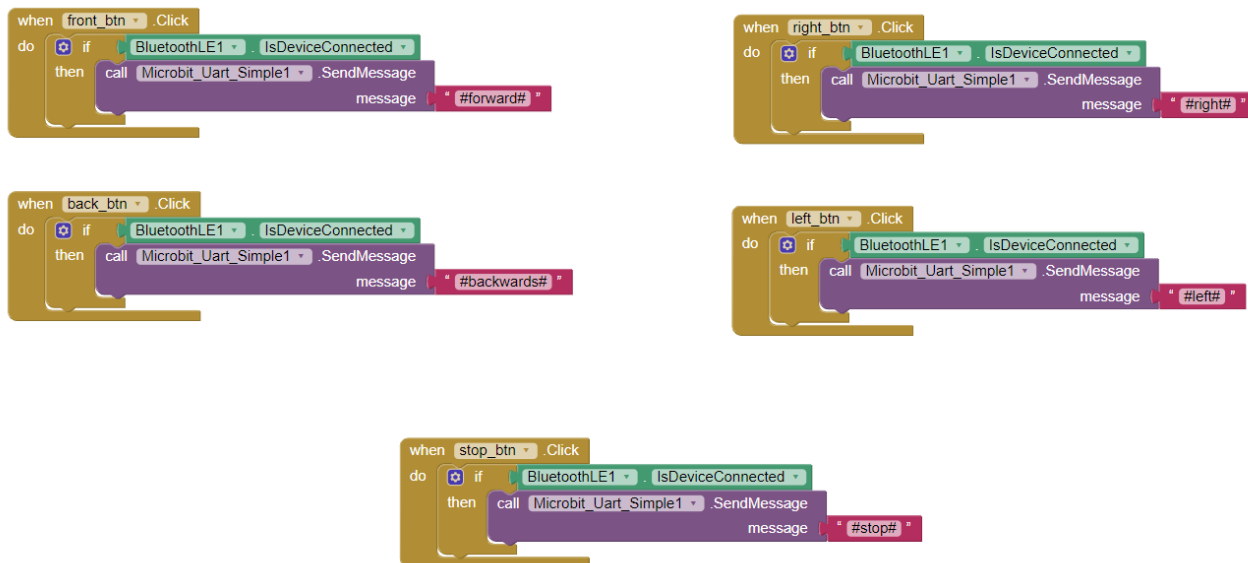
Figuur 19: De “als...dan”-blokvoorwaarde vinden

Via het volgende blok met opdrachten instrueren we onze applicatie dus om te controleren of de **BluetoothLE**- component **verbinding heeft gemaakt met** het gewenste **apparaat** , en **als** dat het geval is, **dan** het bijbehorende **bericht** (“**#forward#**” in dit voorbeeld) wordt **verzonden** door te **bellen** de **Microbit_Uart_Simple** bestanddeel.



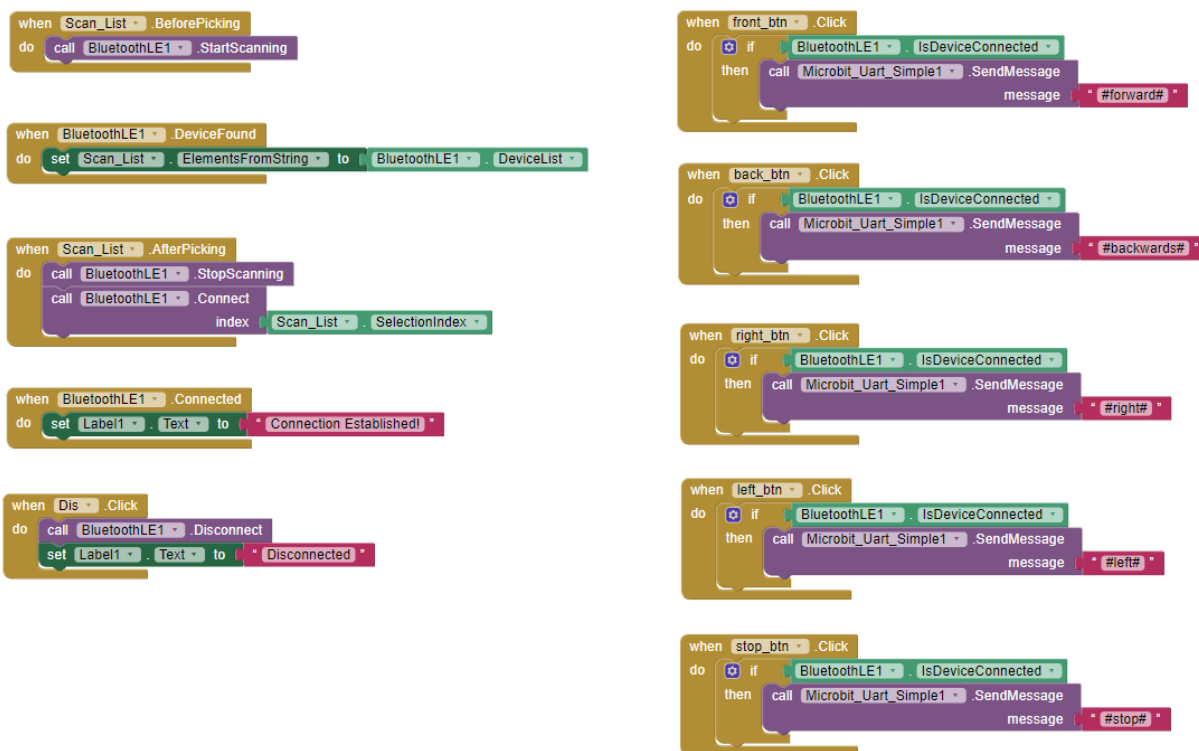
Opmerking: de “vooruit”-waarde, geschreven in de tekst, is een waarde die is gedeclareerd in het script dat is gemaakt in de MakeCode- programmeeromgeving en is toegewezen aan de “vooruitgaan”-beweging.

Herhaal hetzelfde proces voor de andere vier knoppen. Het resultaat moet vergelijkbaar zijn met het resultaat dat is afgebeeld in *Figuur 20* .



Figuur 20: De scripts maken voor alle knoppen van de applicatie

Wanneer u alle bovengenoemde stappen heeft voltooid, zou het volledige script eruit moeten zien zoals afgebeeld in *Figuur 21*.



Figuur 21: Het volledige script voor deze opwarmactiviteit

Nu de applicatie klaar is, kunt u de applicatie bouwen en uploaden naar uw smart device.

1.5 Het bouwen van de applicatie

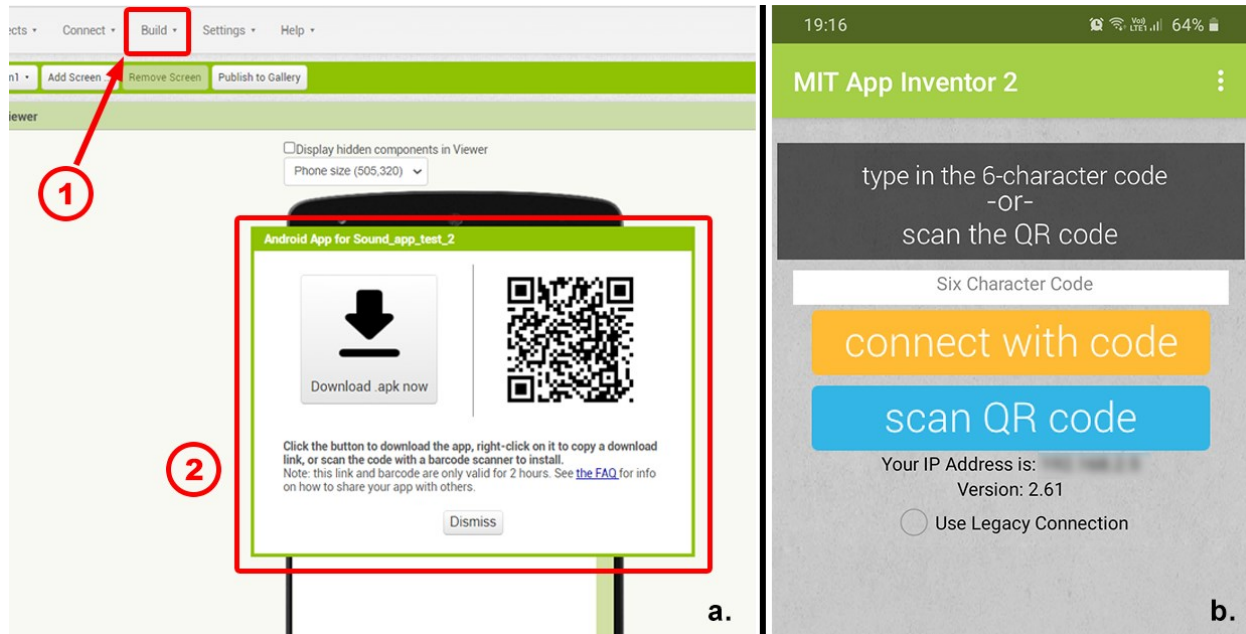
Wanneer u alle bovengenoemde stappen heeft voltooid (en de volledige code vergelijkbaar is met die weergegeven in *Figuur 21*), is de applicatie klaar om te worden geüpload en geïnstalleerd op uw smartapparaat.

Ga dus naar het *Build-menu* **(1)** (*Figuur 22a*) en selecteer “ *Android-app (.apk)* ” in het vervolgkeuzemenu om het proces van het . productie van apk- bestanden. Dit kan een paar minuten duren.

Wanneer het bouwproces is voltooid, verschijnt er een nieuw venster **(2)** (*Figuur 22a*). U kunt ervoor kiezen om het geproduceerde . apk- bestand, of u kunt met uw smartapparaat de ingebedde QR-code scannen via de **MIT AI2 Companion-app** (*Afbeelding 22b*), die u eerder had moeten downloaden en installeren op uw smartapparaat .

Opmerking 1 : MIT AI2 Companion is een applicatie/service die (gratis) beschikbaar is en u kunt deze vinden in de “Play Store”-service van uw smartapparaat. Deze applicatie fungeert als bemiddelaar en vergemakkelijkt de succesvolle installatie van het geproduceerde . apk- bestand naar uw smartapparaat. Deze applicatie is alleen beschikbaar voor Android-apparaten.

Opmerking 2: Tijdens de installatie van het . apk- bestand, kunnen er een aantal meldingen verschijnen over de veiligheid/beveiliging van dit bestand. Negeer ze allemaal en vraag uw apparaat om door te gaan met het installatieproces.



Figuur 22: een. QR-code geproduceerd na voltooiing van het bouwproces; B. Schermafbeelding van de MIT AI2 Companion-interface

Nadat u de applicatie met succes op uw smartapparaat hebt geïnstalleerd, kunt u testen of deze goed werkt. **Houd er rekening mee** dat u het script dat is geproduceerd in de Makecode-programmeeromgeving naar uw robotauto moet hebben gedownload.

1.6 De applicatie koppelen met de robotauto

Zoals hierboven vermeld, moet u, om de applicatie met de robotauto te verbinden, op de knop "Scannen" drukken en het Bluetooth-adres van de micro:bit uit de lijst selecteren. Het is echter waarschijnlijk dat de micro:bit niet zal worden opgenomen in de lijst met beschikbare Bluetooth-apparaten. Om dit probleem op te lossen, gaat u terug naar het hoofdmenu van de applicatie en **activeert u** de "vliegtuigmodus" op uw smartapparaat gedurende een paar seconden. **Deactiveer het** vervolgens en druk opnieuw op de Scan-knop. Het Bluetooth-adres van Micro:bit is nu beschikbaar.

Let op: Zorg ervoor dat "Locatie" ook is geactiveerd op uw smartapparaat