










Module 5: Programmeren

 Doelgroep Vmbo leerjaar 3/4	 Aansluitend keuzedeel Robotica
 Duur 600 minuten	

Tijdsindicatie per onderdeel

Blok 1: Introductie in programmeertalen	90 min.	
Blok 2: Installatie en configuratie robot	110 min.	
Blok 3: Blocks programmeren	300 min.	
Blok 4: Onbot java	300 min.	

Blok 1: Introductie in programmeertalen



Wat gaan we doen?

1. Kennismaken met programmertalen
2. Blocks, onbot Java en Android studio
3. Aan de slag met computational thinking



10 min.
20 min.
20 min.

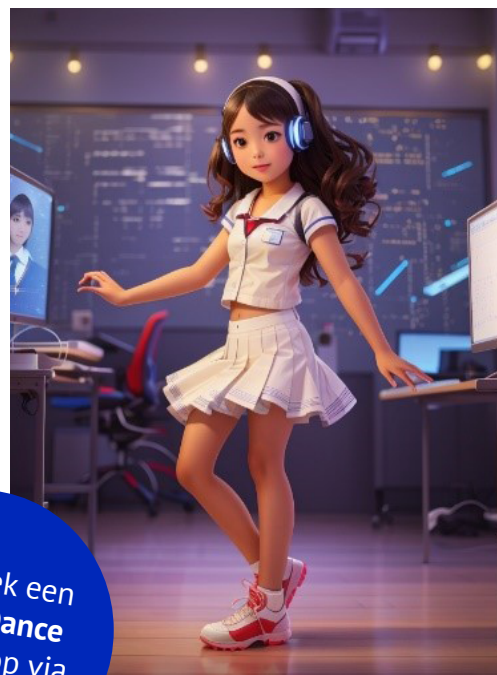
Stap 1: Kennismaken met programmeertalen

In dit blok maak je een start met het ontdekken wat een programmeertaal is. Je leert daarna hoe je een programmeertaal kunt gebruiken om een robot te laten bewegen. Programmeren is het geven van instructies of opdrachten aan een computer. De computer kan deze instructies uitvoeren. De persoon die instructies geeft aan een computer, staat ook wel bekend als een programmeur of developer.

Als je gaat programmeren maak je gebruik van een **programmeertaal**. Dit is een taal waarin de opdrachten die een computer moet uitvoeren, worden geschreven. Er zijn heel veel programmeertalen. De drie bekendste programmeertalen zijn Python, C en Java.

Bij een programmeertaal zijn precieze en volledige instructies erg belangrijk. Als iets niet precies beschreven is of als stappen zijn overgeslagen werkt het programma niet.

- 1 Je gaat een dans programmeren met enkele klasgenoten. Je gaat hierbij kijken of je precies kunt programmeren.
 - Een persoon is de robot en gaat naar de gang. De anderen (de programmeurs) schrijven (of tekenen) een **serie van 10** dansmoves.
 - De 'robot' komt terug en gaat het programma uitvoeren.
 - De programmeurs kijken of de robot het programma goed doet. Ze moeten hem anders **debuggen** (de fout eruit halen).



Tip! Zoek een **Robot Dance** tutorial op via Youtube.

De stappen die je hebt gebruikt om jouw klasgenoot een dansje te laten doen is vergelijkbaar met een programmeertaal. Het gaat om heldere en duidelijke stappen die stap voor stap gedaan moeten worden. **FIRST**[®] raadt aan om te beginnen met de Blocks programmeertaal.

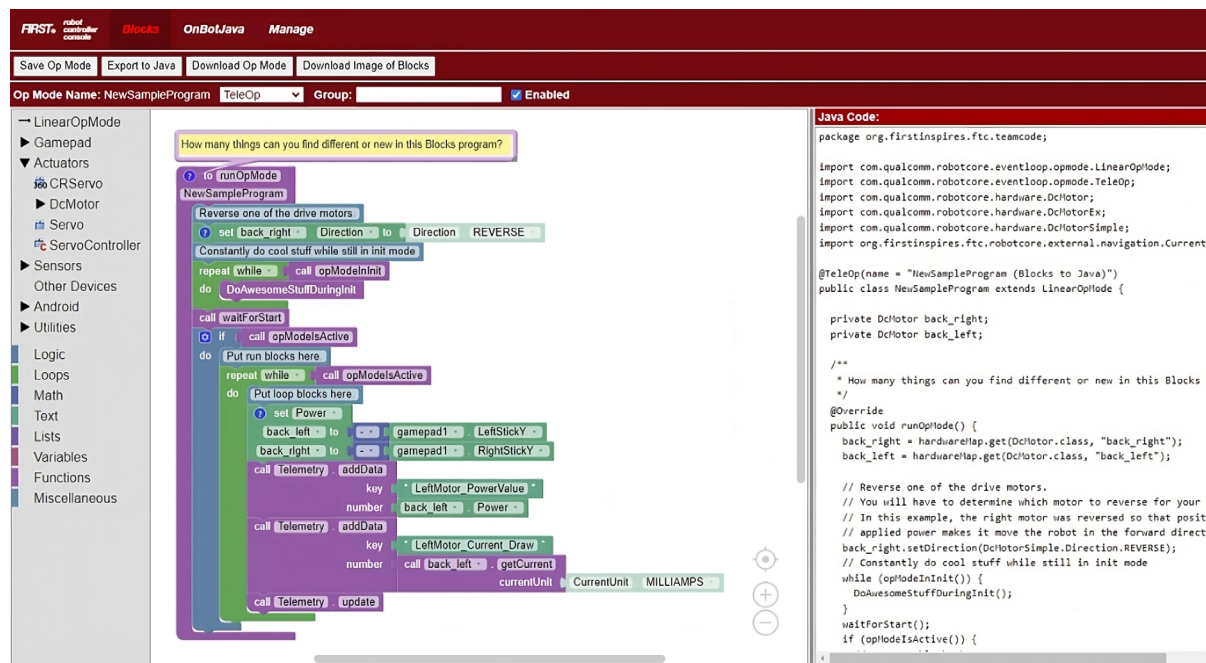
Stap 2: Kennismaken met Blocks en onbot Java

Met Blocks is het mogelijk om op een grafische manier te programmeren. Door middel van blokken geef je opdrachten aan een computer.

Je ziet hieronder een voorbeeld van Blocks Code in de console.

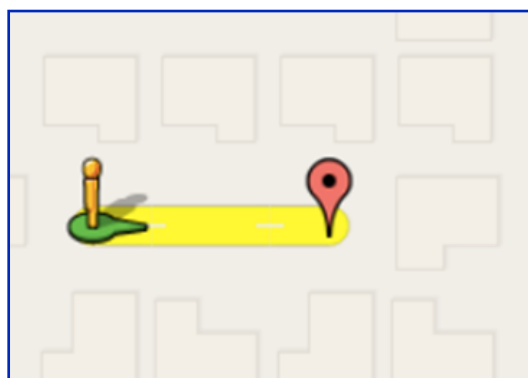
Rechts zie je een tekstcode. Dat is de Java Code. Door middel van regels tekst geef je instructies aan de computer.

Dit lijkt in het begin lastig. Maar het is wel heel makkelijk aan te passen. Ook kan je veel code online vinden en makkelijk kopiëren.



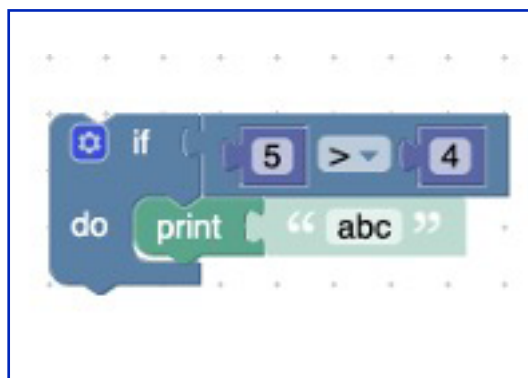
- 1 Laten we eerst simpel beginnen en kijken hoe Blocks werkt. Dit doen we aan de hand van een spelletje genaamd Blockly. Hierin moet je elke keer het poppetje naar het eindpunt laten lopen. Probeer **minimaal 2** levels te halen.

Ga naar de site om het spel te spelen:
<https://blockly.games/maze?lang=nl>



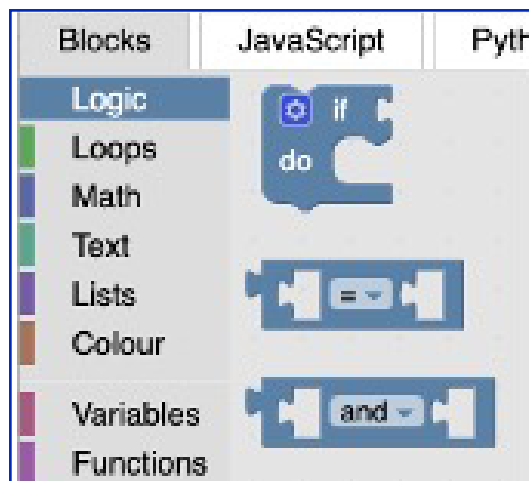
- 2 We gaan nu oefenen met Blocks instructies te geven.

Ga naar:
<https://blockly-demo.appspot.com/static/demos/code/index.html>



Stap 1

Ga eerst naar het “Blocks” gedeelte en klik op “Logic”.



Stap 2

Sleep het blok zoals vakje hiernaast naar het middelste gedeelte met de stippen.



Stap 3

Sleep het blok zoals vakje hiernaast in “Logic” achter de “if” van het vorige blok en zorg dat het past:



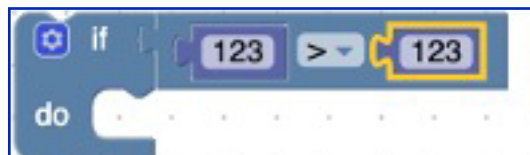
Stap 4

Verander = naar >



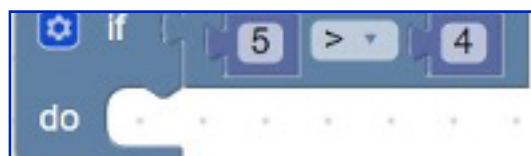
Stap 5

Klik op "Math" en sleep het blok zoals vakje hiernaast in de gaten van het vorige blok om getallen in te vullen.

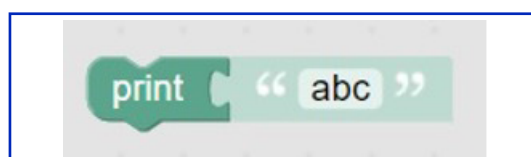


Stap 6

Verander het linkse cijfer naar 5 en het rechtse naar 4. Dit komt er zo uit te zien:

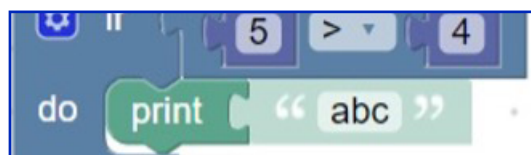


Tot nu toe zegt deze instructie in Blockly het volgende: Als 5 groter is dan (>) 4, doe dan dit... "do" is echter nog leeg, maar hier gaan we verandering in brengen.



Stap 7

Klik op "Text" en sleep dit blok rechts naast "do":



Stap 8

klik op de Play knop.

5 is groter dan 4, dus als je dit programma uitvoert zal je een pop-op op je scherm zien met de tekst "abc".



Stap 9

Maak een Print screen van je resultaat en sla die op in je portfolio.

Stap 3: Aan de slag met computational thinking

Computational thinking is het logisch benaderen van problemen. Waarbij oplossingen worden weergegeven als computer stappen en algoritmen. Een algoritme is een set instructies.

De volgende stappen zijn daarbij van belang:

- Probleem in kleine stukjes opdelen
- Patronen herkennen
- Filteren van informatie
- Oplossingen bedenken en uitproberen
- De oplossingen algemeen maken

Bij het programmeren van een FTC robot is het belangrijk dat je de verschillende opdrachten die de robot moet uitvoeren in kleine delen gaat uitdenken.

Bedenk elke keer:

1. Wat moet de robot precies doen?
2. In welke volgorde?
3. Hoe geef ik instructies met code?
4. Hoe zorg ik voor zo min mogelijk code?

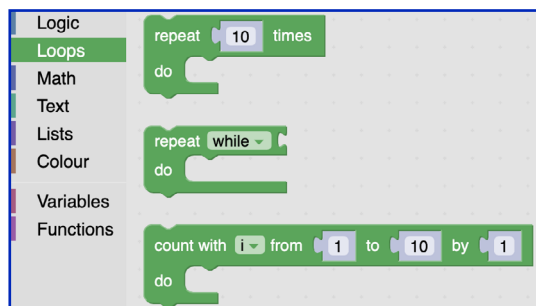


Als je gaat programmeren kom je **Loops**, **if-Statements** en **variables** tegen.

Loops

Met een loop vertel je de computer (of jezelf) om een bepaalde taak te herhalen.

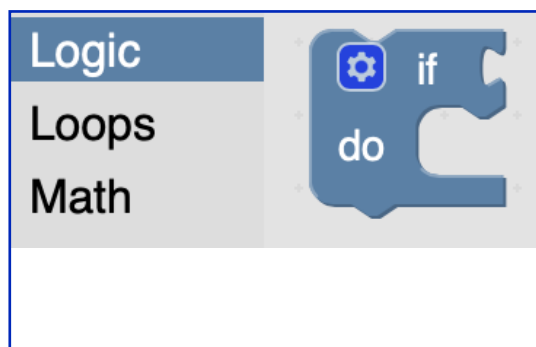
Je kunt bijvoorbeeld iets plaatsen in het “repeat 10 times” block. Dan wordt het 10 keer herhaald. Het getal kan je natuurlijk eenvoudig aanpassen.



If-statements

Een 'if statement' in programmeren werkt een beetje zoals het nemen van beslissingen. Het laat de computer bepaalde acties uitvoeren op basis van voorwaarden. Als de voorwaarde waar is, dan doet de computer iets. Als de voorwaarde niet waar is, doet de computer iets anders of helemaal niets.

Als dan gebeurt dit ... (If ... then...)



- 3 Je ziet hiernaast een stuk Scratch code. Dit is net als Blocks een grafische programmeertaal. Je ziet hier niet staan "If...do.." maar "Als..dan..". De werking is hetzelfde.

Welke actie voert de robot uit als het regent?

```
to regenCheck
  call Telemetry . addLine
  text "Maak je klaar"
  if hetRegent = true
  do
    call Telemetry . addLine
    text "Gebruik een paraplu"
```

- 4 Benoem het if-statement in onderstaande code:

```
to runOpMode
  MyFIRSTOpMode
  Put initialization blocks here.
  call waitForStart
  if call opModelsActive
  do
    Put run blocks here.
    set LeftDrive . Power to 1
    repeat while call opModelsActive
    do
      Put loop blocks here.
      call Telemetry . update
```

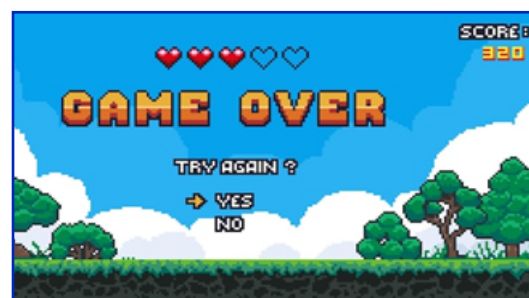
Variabelen

Soms heb je een bepaalde functie of waarde in een programmacode meer dan één keer nodig. In dat geval kun je – i.p.v. die functie of waarde steeds opnieuw op te schrijven – de data opslaan in een variabele en dan gewoon de naam van de variabele gebruiken.

```
Create variable...
set score to
change score by 1
score
```

- 5 In dit spel zitten bepaalde waarden die elke keer kunnen veranderen.

Noem twee waarden of variabelen.



Blok 2: Robot installeren en configureren



Wat gaan we doen?

1. Materiaal ordenen
2. Robot verbinden
3. Connectie maken met een device
4. Configuratie bestand maken



10 min.
15 min.
25 min.
50 min.

In dit blok ga je aan de slag met het installeren en configureren van de robot. Er wordt uitgegaan van het gebruik van de Control Hub en een Driver Hub.

Stap 1: Materiaal ordenen

- 1 Zet je robot klaar en leg daarnaast alle spullen klaar zoals de afbeelding hieronder en maak een foto voor je portfolio.

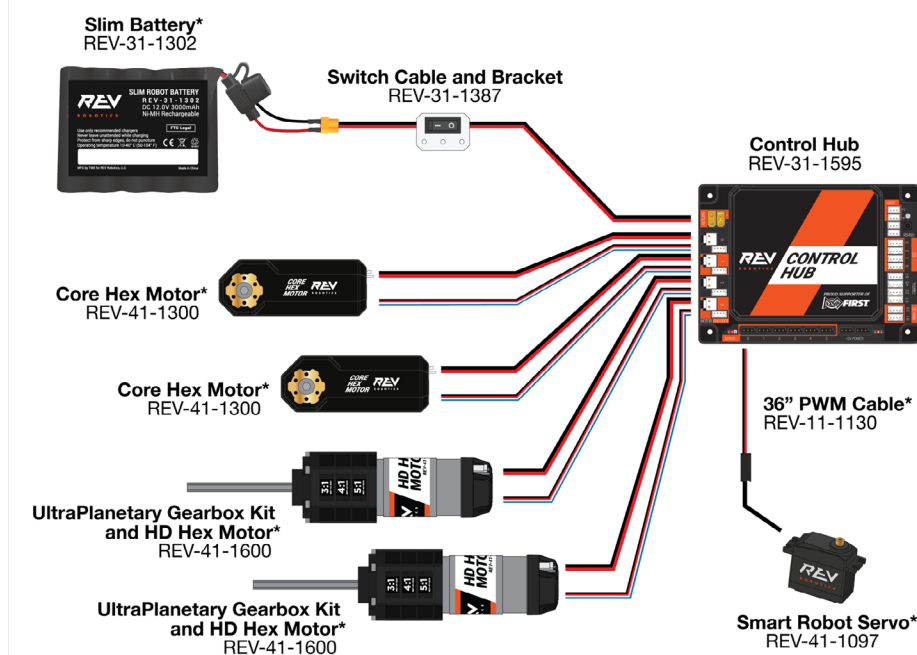


Let op!

Vergeet niet je robot ook op de foto te zetten!

Stap 2: Robot verbinden

- 1 Verbind de verschillende onderdelen met elkaar zoals aangegeven op onderstaande afbeelding. Bespreek met je docent of alles goed zit aangesloten.



- 2 Sluit de twee controllers aan op de **driver station**.

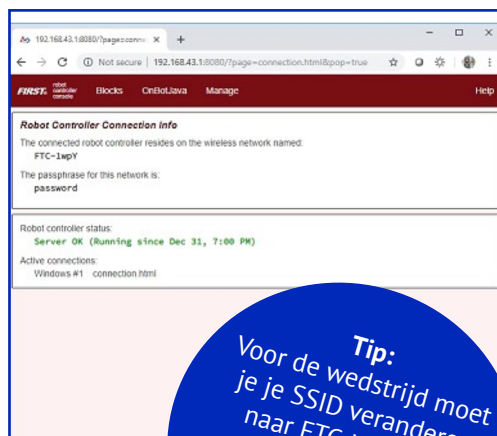


Stap 3: Connectie maken met een device

- 1 Je gaat de verschillende systemen met elkaar verbinden.
 1. Zet een computer klaar met wifi functionaliteit.
 2. Schakel de Control Hub in. (Duurt ongeveer 40 seconden.)
 3. Koppel de Control Hub met een computer via wifi:
Open een webbrowser (Chrome, Firefox, Edge) en ga naar "192.168.43.1:8080" via de adresbalk om bij de Robot Control Console te komen.

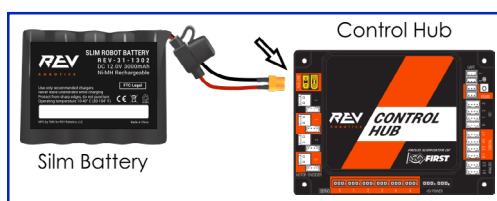
Tip: Standaard heeft de Control Hub een naam die begint met de zin "FTC" of "FIRST" gevolgd door vier willekeurig toegewezen tekens. Het standaard wachtwoord voor het netwerk is "password".

Vanuit de console kun je de naam van de Control Hub en het wachtwoord veranderen. Je kunt hier ook het apparaat programmeren. Eventuele updates aan het besturingssysteem of de software kun je hier ook doen.



Tip:
Voor de wedstrijd moet je je SSID veranderen naar FTC. Verander daarna het teamnummer naar jouw nummer van FIRST.

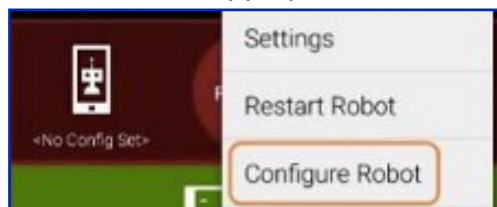
4. Verbindt de Driver Station met de Control Hub. Bekijk de [video](#)¹ om te zien hoe je dit kunt doen.



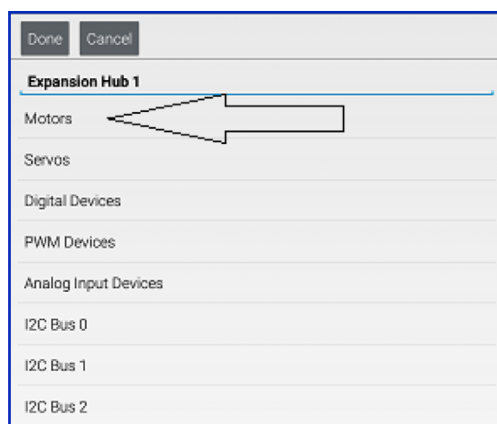
Stap 4: Configuratie bestand maken

- 1 Om de Control Hub te kunnen gebruiken hebben we een configuratie bestand nodig, volg onderstaande stappen om deze te maken.

1. Klik op de 3 stippen in de rechterbovenhoek van de FTC-app op de Driver station.
2. Selecteer 'Configure Robot'.
3. Maak een nieuw configuratie bestand.

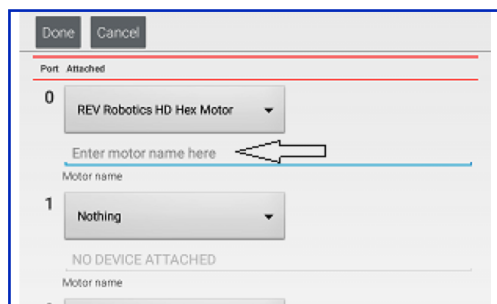


4. Als het configuratiescherm opstart, voert de robot controller app een scan uit om te kijken wat al is verbonden met de Robot controller. Na de scan zie je de Control Hub. Vanuit de Control Hub kan je de verschillende onderdelen een naam geven. Dit is nodig om later in de code de onderdelen te kunnen gebruiken.



¹ <https://docs.revrobotics.com/duo-control/control-hub-gs/driver-station-pairing-to-control-hub>

- Selecteer de onderdelen.
Je ziet op poort 0 een HD Hex motor, kijk goed hoe je deze zelf hebt aangesloten en zorg dat je de motoren op de juiste poort zet. Geef de onderdelen een duidelijke naam. Gebruik voor motoren: leftDrive en rightDrive.



2 Je ziet de motoren hieronder in de code terugkomen. Zie jij waar?

Hieronder zie je een voorbeeld van configuratie instellingen:

Port Type	Port Number	Device type	Name
Motor	0	HdHexMotor	leftDrive
Motor	1	HdHexMotor	rightDrive
Motor	2	CoreHexMotor	arm
Motor	3	CoreHexMotor	intake
Servo	0	Servo	servo
12C	0	IMU	IMU

Blok 3: Programmeren in Blocks



Wat gaan we doen?

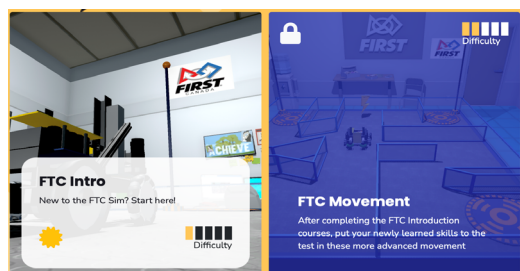
- | | |
|--|----------|
| 1. Aan de slag met een simulator | 30 min. |
| 2. TeleOp modes schrijven in Blocks | 20 min. |
| 3. Contollers bedienen | 30 min. |
| 4. Onderstel testen | 60 min. |
| 5. Een TeleOpMode uitvoeren | 150 min. |
| 6. Autonome modus programmeren in Blocks | 150 min. |



In dit blok ga je aan de slag met het programmeren van de *FIRST* Tech Challenge robot. Je gaat **TeleOp** en **Autonomous** modes schrijven. Je start met een simulator.

Stap 1: Aan de slag met een simulator

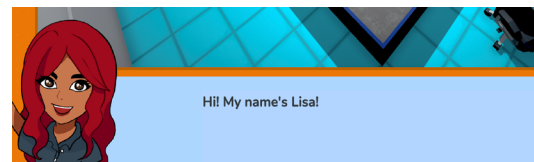
- Ga naar <https://ftcsim.org> en start met de FTC intro course.
- Start met FTC movement (zie afbeelding). Voor de laatste lessen moet je jezelf eenmalig aanmelden.



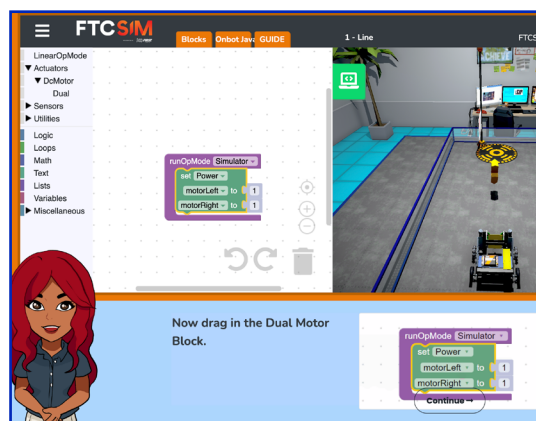
In Blocks heb je verschillende **OpModes**. Dit zijn zogenaamde Operational Modes. Dit zijn manieren om je robot te programmeren. Je ziet hiernaast een voorbeeld van de Simulator modus. Als je straks daadwerkelijk aan de slag gaat met je robot heb je de **TeleOp** Mode en de **Autonomous Mode**.



- Klik rechtsboven op help. Het karakter Lisa helpt je dan door de verschillende onderdelen.
- Begin bij het toevoegen van de motor. Probeer daarna de verschillende onderdelen uit te voeren.



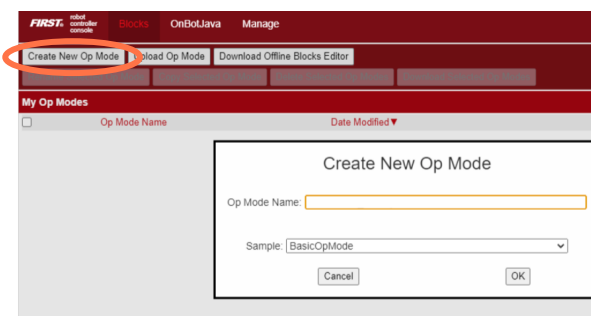
Als je klaar bent met de verschillende onderdelen in de Sim kan je starten met het programmeren van de 'echte' robot.



Stap 2: TeleOp Mode schrijven in Blocks

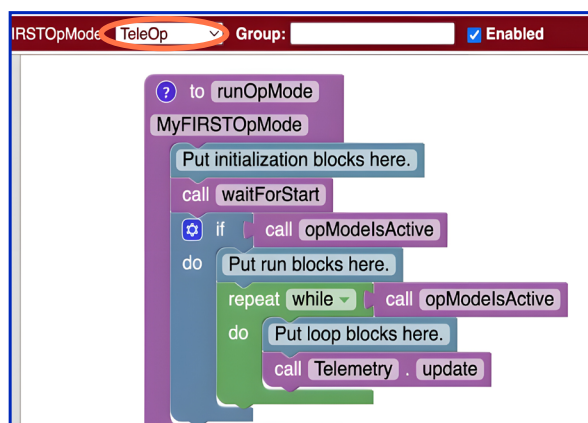
Bij TeleOp Modes gebruik je de controllers om de robot te bewegen.

- Open een webbrowser (Chrome, Firefox, Edge) en ga naar "192.168.43.1:8080" via de adresbalk om bij de Robot Control Console te komen.
- Klik op Blocks.
- Klik op Create New Op Mode.
- Geef de Mode een naam.



Je ziet dat er meerdere blokken te zien zijn die je niet in de simulator zag.

Alle programmeerblokken die na de opmerking "Put initialization blocks here" worden geplaatst (en vóór het blok "call MyFIRSTOpMode") worden door de robot uitgevoerd wanneer de Op Mode wordt geselecteerd en ip INIT wordt geklikt.



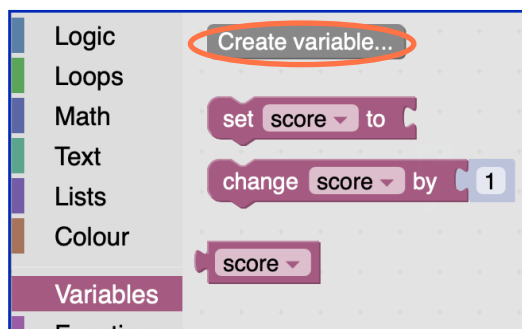
Stap 3: Controllers bedienen

Bij de FTC-wedstrijden zijn er twee (PS4 of f310) controllers aangesloten op een Driver Hub, zoals de afbeelding laat zien. We gaan de controllers klaarmaken om te gebruiken.

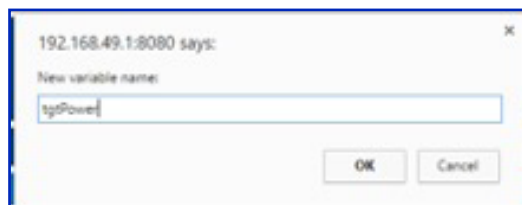
- Selecteer aan de linkerkant van het scherm de categorie Variables om de lijst met Block taken weer te geven die worden gebruikt om variabelen binnen de OpMode te maken en of te wijzigen.



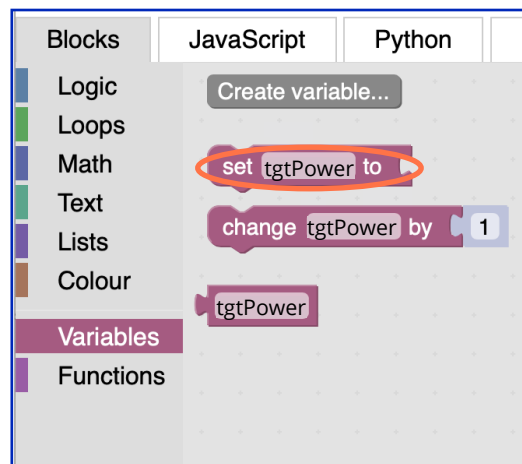
- Selecteer 'Create Variable...' om het motorvermogen voor onze OpMode te creëren.



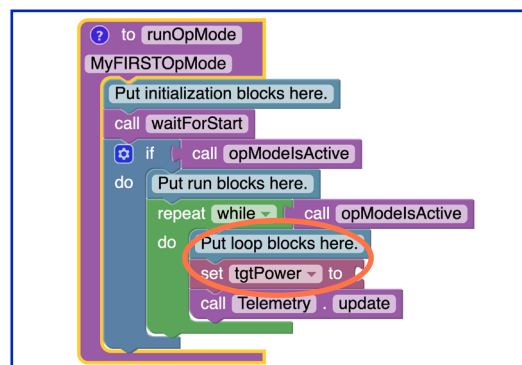
- Typ in het volgende blokje wat verschijnt onder New Variabel Name: tgtPower. Druk op 'OK'.



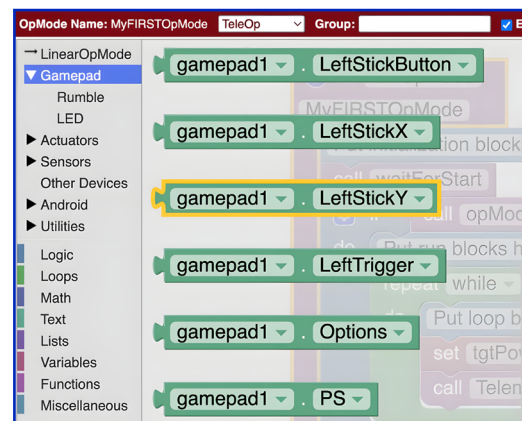
- Klik op de "Variables" block categorie en selecteer "tgtPower" block. Als je dat hebt gedaan verschijnen er meer Blocks blokken zoals het plaatje hiernaast laat zien.



- Selecteer het blok "set tgtPower to" en sleep dit onder het blok genaamd "Put loop blocks here". Het blok "set tgtPower to" zou moeten oplichten en op zijn plaats moeten klikken.



- klik op de categorie "Gamepad" van de programmeerblokken en selecteer het blok "gamepad1.LeftStickY" uit de lijst met beschikbare blokken.







- Sleep “gamepad1.LeftStickY” rechts naast “set tgtPower” to block.
- Controleer of je code overeenkomt. Deze set blokken loopt voortdurend door (de loop is: repeat while).

```

to runOpMode
  MyFIRSTOpMode
  Put initialization blocks here.
  call waitForStart
  if call opModelsActive
  do
    Put run blocks here.
    repeat while call opModelsActive
    do
      Put loop blocks here.
      set tgtPower to gamepad1 . LeftStickY
  
```

De waarde van de linker joystick van gamepad #1 (de y-positie) en stelt de variabele tgtPower in op de Y-waarde van de linker joystick. In onderstaande tabel zie je verschillende waarden van de PS4 controller:

- Je ziet dat bij Y waarde de joystick omhooggaat (y-waarde = 1) of naar beneden (y-waarde = -1).
- Bij rightmotor en leftmotor zie je het getal 1. Dit betekent dus vooruit. Of het getal -1 dat is dus achteruit.
- Je zag deze waarden in de sim.

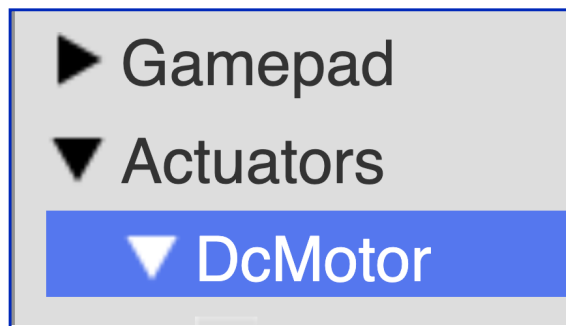
Joystick Direction	(x , y)	leftmotor	rightmotor
	(0,1)	1	1
	(0,-1)	-1	-1
	(-1,0)	1	-1
	(1,0)	-1	1

Let op!
 De waarden zijn afhankelijk van de controller die wordt gebruikt. Bovenstaande waarden zijn correct voor PS4 controllers. De waarden van de y-as zijn omgekeerd voor F310 controllers. Bij de F310 controller wordt iedere 1 dus een -1.

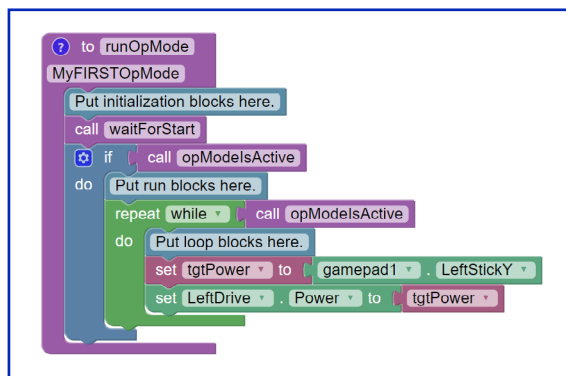
- Klik vervolgens op de “Actuators” categorie in Blocks

Tip:

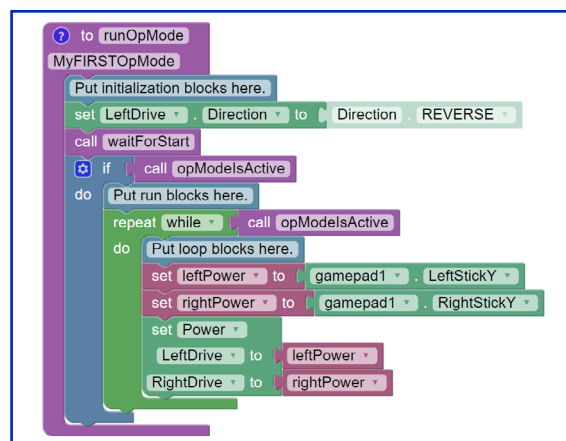
In het voorbeeld zijn de motoren niet als LeftDrive en RightDrive opgeslagen maar als motor1. Controleer de naam van de motoren die horen bij jouw file.



- Selecteer de “set motor1.Power to 1” programming block.
- Klik op de “Variables” block categorie en selecteer “tgtPower” block.
- Sleep de “set motor1.Power to 1” block onder de “set tgtPower to” block.
- Klik op de “Variables” block categorie en selecteer “tgtPower” block.
- sleep de “tgtPower” block naar “set motor1.Power to” block.



- Klik op “Save Op Mode” als je klaar bent met het configureren van je block. Je ziet hieronder een voorbeeld van een drive code waarbij de motoren anders genaamd zijn. Herken je ook het REVERSE block uit de simulator?



Stap 4: Onderstel testen

In blok 4 heb je één van de grijpers/verzamelaars die je hebt gemaakt bevestigd aan je robot. De opdracht voor de robot is om voorwerpen op te pakken en in een doos te doen.

1 Gebruik programma : Testen van de drivetrain. Met het testprogramma TeleOp kan je de robot besturen met de controllers.

- De leftdrive is in dit voorbeeld de linkermotor.
- De rightdrive is in dit voorbeeld de rechtermotor.
- De gripper is de grijper. (blok 2)
- De arm is de hefboom. (blok 3)

2 Beantwoord de vragen in je learning journal.

• **Wendbaarheid:**

1. Kan je robot makkelijk draaien?
2. Kan je robot snel draaien?
3. Is je robot makkelijk te besturen

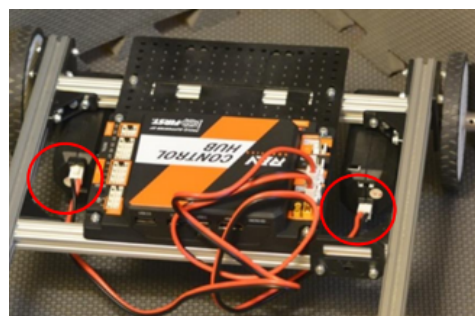
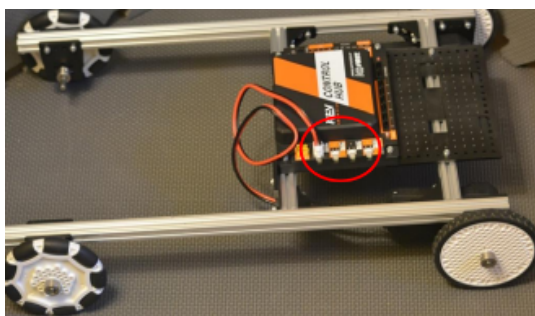
• **Vormen:**

1. Kan de robot een volle cirkel, 360 graden op de plaats draaien?
2. Kan de robot een rondje om een bekertje heen rijden?
3. Kan de robot op zijn plek een hoek van 90 graden naar links draaien? Noteer waar het draaipunt van de robot ligt.
4. kan je de robot een vierkant laten rijden? Of een driehoek?

• **Terrein:**

Beantwoord onderstaande vragen voor verschillende voorwerpen zoals een potlood, dun schrift of een boek.

1. Leg het voorwerp neer (en houd het vast).
Kan de robot over het voorwerp heen rijden?
2. Maakt het verschil of het voorwerp voor of achter de robot ligt?
3. Maakt het verschil of het voorwerp tegen de wielen aan ligt bij het begin of dat de robot een aanloop neemt?
4. Maakt het uit hoe snel de robot gaat?



Kijk hier voor meer informatie: https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/robot-wiring-guide.pdf

Je hebt nu geprogrammeerd dat je met de joystick de verschillende motoren kan bedienen. Hieronder zie je een code waarbij ook de arm en de intake zijn geprogrammeerd:

```

? to runOpMode
MyFIRSTOpMode
  Put initialization blocks here.
  set LeftDrive . Direction to Direction . REVERSE
  call waitForStart
  if call opModelsActive
  do Put run blocks here.
  repeat while call opModelsActive
  do Rijden
  set leftPower to gamepad1 . LeftStickY + gamepad1 . RightStickX
  set rightPower to gamepad1 . LeftStickY - gamepad1 . RightStickX
  set Power
  LeftDrive to leftPower
  RightDrive to rightPower
  Arm beweging
  set arm . Power to gamepad2 . LeftStickY
  Gripper beweging (met servo)
  set gripper . Position to gamepad2 . RightTrigger
  
```

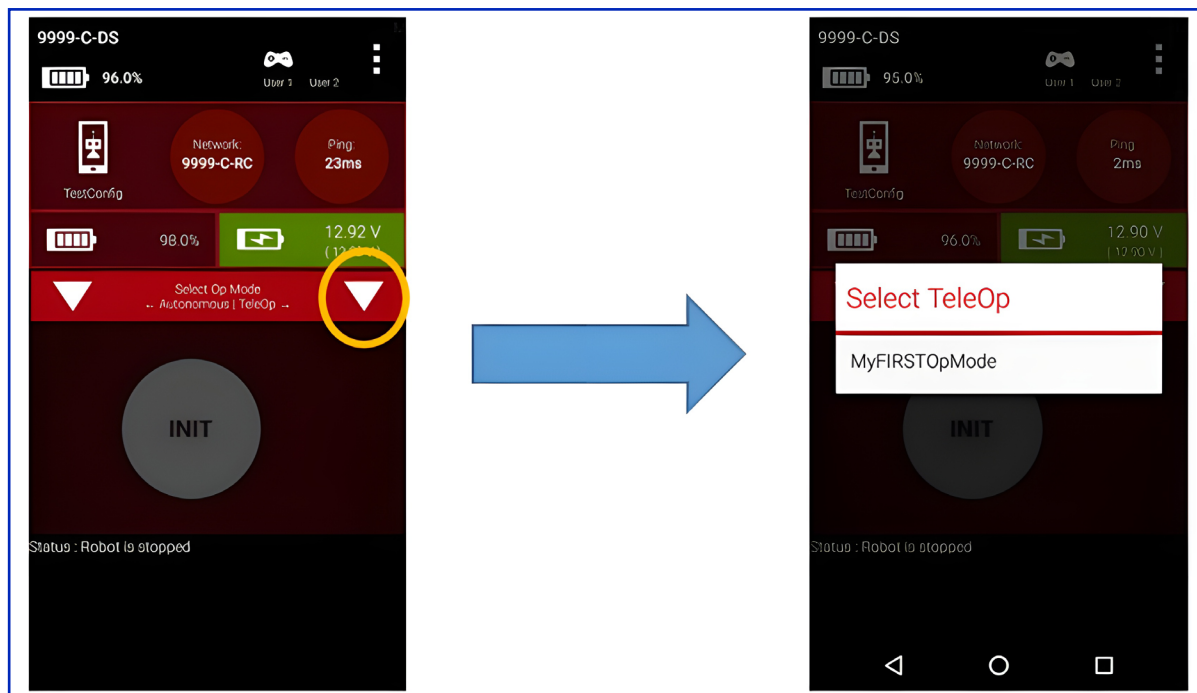
Stap 5: TeleOpMode uitvoeren

As de Control Hub en de driver station goed zijn verbonden dan zou de driver station app er ongeveer zoals het plaatje hier uit moeten zien.

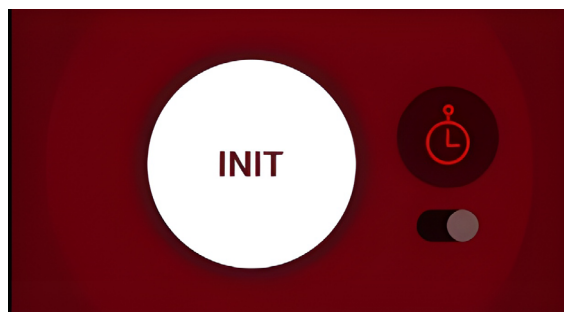
Het linker driehoekje staat voor Autonomous.

Het rechter driehoekje staat voor de TeleOp mode.

- Klik op de driehoek van “TeleOp” aan de rechterkant van het Driver Station scherm om de beschikbare OpModes te laten zien
- Selecteer de OpMode die jij ziet staan.



- Druk op de “INIT” knop om de OpMode te starten. Druk vervolgens op de play knop.
- Je kunt de controllers nu uitproberen.



Stap 6: Autonome modus programmeren in Blocks

Naast de TeleOp Mode kun je de robot ook autonoom laten bewegen. Je gebruik daarvoor de Autonomous mode.

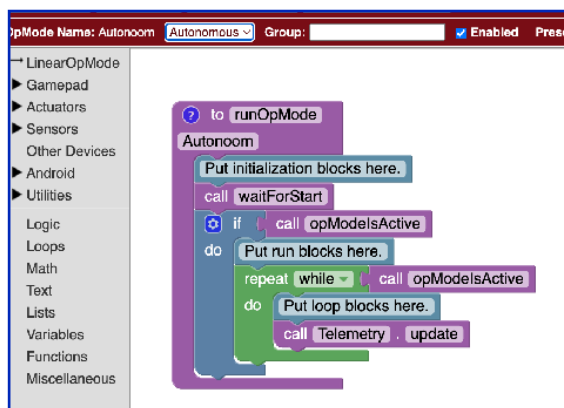
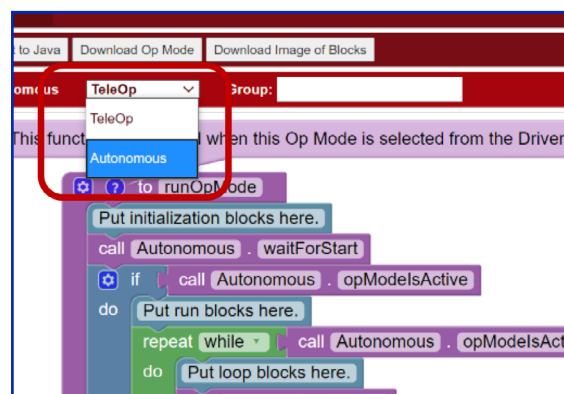
- Maak een nieuwe OpMode aan en kies Autonomous

In de simulator heb je geoefend met vooruit en achteruitrijden. Ook heb je geleerd een bocht te maken.

Je gaat deze code nu zelf in het blok zetten.

- Plaats vanuit de Actuators de motor in het block/code.
- Zorg voor het Reverse blok.
- Plaats de code in het if..do..blok.
- Zorg dat de robot automatisch vooruit en automatisch achteruitgaat.
- Zorg er ook voor dat de robot een bocht maakt.

Je ziet hiernaast nog een keer de code uit de simulator. Let op dat de runOpMode hier dus op Simulator staat en niet op Autonomous.



Blok 4: OnBot Java



Wat gaan we doen?

1. Programmeren in OnBot Java
2. OpMode schrijven in OnBot Java
3. Autonomous mode schrijven in OnBot Java
4. Het omzetten van encoder-tikken naar een afstand



30 min.
40 min.
40 min.
40 min.

Stap 1: Programmeren in OnBot Java

Naast Blocks kan je ook gebruik maken van OnBotJava om je robot te programmeren, het verschil hierin is de OnBotJava niet met blokken werkt maar met tekst.

- Klik bovenin op OnBotJava.
- Klik op + voor een nieuw bestand.
- Noem het bestand "MyFIRSTJavaOpMode" en klik op "OK".

```

31 @TeleOp
32 ~ public class MyFIRSTJavaOpMode extends LinearOpMode {
33     private DcMotor leftDrive;
34     private DcMotor rightDrive;
35     private DcMotor arm;
36     private Servo gripper;
37     private IMU imu;
38
39
40     @Override
41 ~ public void runOpMode() {
42         control_Hub = hardwareMap.get(Blinker.class, "Control Hub");
43         leftDrive = hardwareMap.get(DcMotor.class, "LeftDrive");
44         rightDrive = hardwareMap.get(DcMotor.class, "RightDrive");
45         arm = hardwareMap.get(DcMotor.class, "arm");
46         gripper = hardwareMap.get(Servo.class, "gripper");
47         imu = hardwareMap.get(IMU.class, "imu");
48
49         telemetry.addData("Status", "Initialized");
50         telemetry.update();
51         // Wait for the game to start (driver presses PLAY)
52         waitForStart();
53     }

```

Stap 2: OpMode schrijven in OnBot Java

In het vorige blok heb je gezien hoe je de joysticks kan gebruiken voor het besturen van je robot. Je noemt dit ook wel een Arcade-aandrijving.

Om de controllers te kunnen gebruiken in OnbotJava voer je de volgende stappen uit:

- Schrijf de code om de controllers in te stellen.

```

// run until the end of the match (driver presses STOP)
double tgtPower = 0;

while (opModeIsActive()) {
    tgtPower = gamepad1.left_stick_y;
    leftDrive.setPower(tgtPower);
    telemetry.addData("Target Power", tgtPower);
    telemetry.addData("Motor Power", leftDrive.getPower());
    telemetry.addData("Status", "Running");
    telemetry.update();
}

```

Deze code loopt voortdurend door en leest de waarde van de linker joystick van gamepad #1 (de y-positie) en stelt de variabele `tgtPower` in op de Y-waarde van de linker joystick (PS4 of F310). Merk op dat voor de F310-gamepads de Y-waarde van de joysticks varieert van -1 wanneer de joystick in de bovenste positie staat tot +1 wanneer de joystick in de onderste positie staat.

- Klik op “[Save Op Mode](#)” als je klaar bent met het configureren van je block. Je kunt de `OpMode` nu op dezelfde manier uitvoeren als je gewend bent. Het grote voordeel van werken met OnBot Java is dat je snel code kan kopiëren.
- In dit voorbeeld ga je een arcade-aandrijving programmeren met behulp van twee joysticks.

Dit type aandrijving wordt “[split arcade](#)” genoemd. Bij split arcade-aandrijving zal de linker joystick de voor- en achterwaartse beweging van de robot regelen, en de rechter joystick zal de draaibeweging regelen.

- Open het `.java` bestand die de docent met je heeft gedeeld en kopieer de tekst in je ‘[MyFIRSTJavaOpMode](#)’.

Let op!

Zorg dat de `deviceName` in regel 51 en 52 overeenkomt naar de `deviceName` uit het configuratiebestand:

```
motorLeft = hardwareMap.get(DcMotor.class, deviceName: "left_drive");  
motorRight = hardwareMap.get(DcMotor.class, deviceName: "right_drive");
```

Stap 3: Autonomous mode schrijven in OnBot Java

Tijdens de wedstrijd zijn de eerste 30 seconden autonoom. Vervolgens ga je de robot handmatig besturen. Om een autonomous mode in Java te programmeren doe je het volgende:

- Je klikt bovenin op `OnBotJava`.
- Klik op + voor een nieuwe file.
- Noem de file “[MyFIRSTJavaAutonomousMode](#)” en klik op “OK”.
- Neem de code op de volgende pagina over:

```

package org.firstinspires.ftc.teamcode;
package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import com.qualcomm.robotcore.eventloop.opmode.Autonomous;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.hardware.DcMotorSimple;

@Autonomous //sets the op mode as an autonomous op mode
public class MyFIRSTJavaAutonomousMode extends LinearOpMode {
    private DcMotor leftmotor;
    private DcMotor rightmotor;

    @Override
    public void runOpMode() {
        leftmotor = hardwareMap.get(DcMotor.class, "LeftDrive");
        rightmotor = hardwareMap.get(DcMotor.class, "RightDrive");

        // Wait for the game to start (driver presses PLAY)
        waitForStart();
        // run until the end of the match (driver presses STOP)
        while (opModeIsActive()){

        }
    }
}

```

Je hebt nu de basiscode. Je hebt geleerd dat je de richting van een van de motoren moet omkeren zodat beide motoren in dezelfde richting bewegen.

- Voeg de volgende regel toe aan de code (omdat motors gespiegeld gemonteert zijn moet je de motor omdraaien met reverse):

```
rightmotor.setDirection(DcMotor.Direction.REVERSE);
```

Plaats de volgende de regel onder bovenstaande regel:

```
rightmotor = hardwareMap.get(DcMotor.class, "RightDrive");
```

- De eerste stap is het instellen van de doelpositie. Voeg de volgende regels toe:

```
leftmotor.setTargetPosition(1000);
rightmotor.setTargetPosition(1000);
```

Plaats de regels na:

```
waitForStart();
```

Om een doelpositie te krijgen die overeenkomt met een doelf afstand zijn berekeningen nodig, die later zullen worden behandeld. Stel voorlopig de doelpositie in op 1000 tikken zoals je kunt zien op de volgende pagina.

```
waitForStart();  
  
leftmotor.  
setTargetPosition(1000);  
rightmotor.  
setTargetPosition(1000);  
  
while (opModeIsActive()){  
  
}
```

De volgende stap is om beide motoren in de RUN_TO_POSITION-modus te zetten.

- Voeg de volgende regels toe aan de code onder de code voor het instellen van de doelpositie:

```
leftmotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);  
rightmotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);
```

De code wordt dan:

```
waitForStart();  
  
leftmotor.setTargetPosition(1000);  
rightmotor.setTargetPosition(1000);  
  
leftmotor.setMode(DcMotor.RunMode.RUN_TO_  
POSITION);  
rightmotor.setMode(DcMotor.RunMode.RUN_TO_  
POSITION);  
  
while (opModeIsActive()){  
  
}
```

Het belangrijkste van het driedelige proces is om een doel in te stellen, de robot te vertellen om naar dat doel te bewegen, en met welke snelheid de robot dat doel moet bereiken.

- Voeg de regels toe om de kracht van beide motoren in te stellen op 80%.

```
waitForStart();

leftmotor.setTargetPosition(1000);
rightmotor.setTargetPosition(1000);

leftmotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);
rightmotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);

leftmotor.setPower(0.8);
rightmotor.setPower(0.8);

while (opModeIsActive()){

}
```

Nu alle drie de RUN_TO_POSITION-stappen aan de code zijn toegevoegd, kan de code worden getest.

- Test de robot twee keer.

Wat valt op? De tweede keer werkt de robot niet.

Alle encoderaansluitingen beginnen op 0 tikken beginnen wanneer de Control Hub wordt ingeschakeld. Omdat je de Control Hub niet hebt uitgeschakeld tussen de runs, waren de motoren de tweede keer dat je de op-modus hebt uitgevoerd al bij de doelpositie, of daaromheen. Wanneer je een code uitvoert, wil je ervoor zorgen dat bepaalde variabelen in een bekende toestand beginnen. Voor de encoder-tikken kan dit worden bereikt door de modus in te stellen op: **STOP_AND_RESET_ENCODER**

- Voeg dit blok toe aan de op-modus in het initialisatiegedeelte. Telkens wanneer de op-modus wordt geïnitieerd, worden de encoder-tikken gereset naar nul. Je krijgt dan de volgende code:

```
public void runOpMode() {
    leftmotor = hardwareMap.get(DcMotor.class, "leftmotor");
    rightmotor = hardwareMap.get(DcMotor.class, "rightmotor");

    rightmotor.setDirection(DcMotor.Direction.REVERSE);

    leftmotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    rightmotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);

    waitForStart();
}
```

Stap 4: Het omzetten van encoder-tikken naar een afstand

Je hebt nu de basis om run_to_position te gebruiken.

Door de plaatsing van de volgende regels wordt de doelpositie ingesteld op 1000 tikken:

```
leftmotor.setTargetPosition(1000);
rightmotor.setTargetPosition(1000);
```

Hoe programmeer je als je wilt dat je robot 1 meter aflegt ten opzichte van zijn startpositie? In plaats van te proberen de de robot aflegt te meten of te schatten, kunnen de encoder tikken worden omgezet van het aantal tikken per omwenteling van de encoder naar hoeveel encoder tikken nodig zijn om de robot eenheid van afstand, zoals een millimeter of een meter te verplaatsen. Als je het aantal tikken per eenheid van maat weet, kun je een specifieke afstand instellen.

Tip:
Kijk op de website bij de specificaties van je encoder om het aantal tikken per omwenteling te vinden.

Je volgt de volgende stappen:

1. Uitzoeken hoeveel tikken de encoder per omwenteling genereert.
2. Uitzoeken wat de afstand per omwenteling oftewel de omtrek van het wiel is:
Omtrek wiel = $\pi \times \text{diameter}$
3. Afstand bereken per encoder tik: $\text{Afstand per tik} = \frac{\text{omtrek wiel}}{\text{encoder tikken per omwenteling}}$
4. Het aantal tikken voor een bepaalde afstand berekenen:
 $\text{aantal tikken} = \frac{\text{Gewenste afstand}}{\text{Afstand per tik}}$

Bijvoorbeeld:

Stel je weet het volgende:

Aantal tikken per omwenteling	Diameter wiel	Afstand per omwenteling	Afstand die je wil dat de robot aflegt
500	10 cm	Omtrek wiel	1 meter (100 cm)

1. Je hebt gevonden dat de encoder 500 tikken per omwenteling genereert.

2. Je hebt gevonden of gemeten dat de diameter van je wiel 10 cm is. Je berekent vervolgens de omtrek van het wiel:
omtrek wiel = $\pi \times \text{diameter} = \pi \times 10 = 31.42$.

3. $\text{Afstand per tik} = \frac{\text{omtrek wiel}}{\text{encoder tikken per omwenteling}} = \frac{31.42}{500} = 0.06284$

4. $\text{Aantal tikken voor bepaalde afstand} = \frac{\text{Gewenste afstand}}{\text{Afstand per tik}} = \frac{100}{0.06284} = 1592$

Dus, als de encoder 1592 tikken registreert verplaatst de robot zich 1 meter.

Veel succes met het programmeren van je robot!