

MikroBOTIK

KIT PEMBELAJARAN ROBOTIK ALAF BARU



- Robot pembelajaran dengan spesifikasi pertandingan.
- Pergerakan berautonomi mengikut garisan.
- Pergerakan bebas dengan kawalan 'Bluetooth'.
- Pengkodan grafik yang mudah dan seronok.



Contents

Element of Robotics.....	1
What is Electronic Hardware?.....	2
What is Software Coding?.....	3
Path Finding Robot.....	4
Contents in the box.....	5
"Mikrobotik" Pathfinding Robot.....	6
Arduino Nano Microcontroller.....	7
LiPo Battery.....	8
Low Battery Indicator.....	9
Installation of mBlock v5.....	10
Steps for Adding Mikrobotik.....	12
Calibration Process.....	15
Autonomous Robot PID Algorithm.....	21

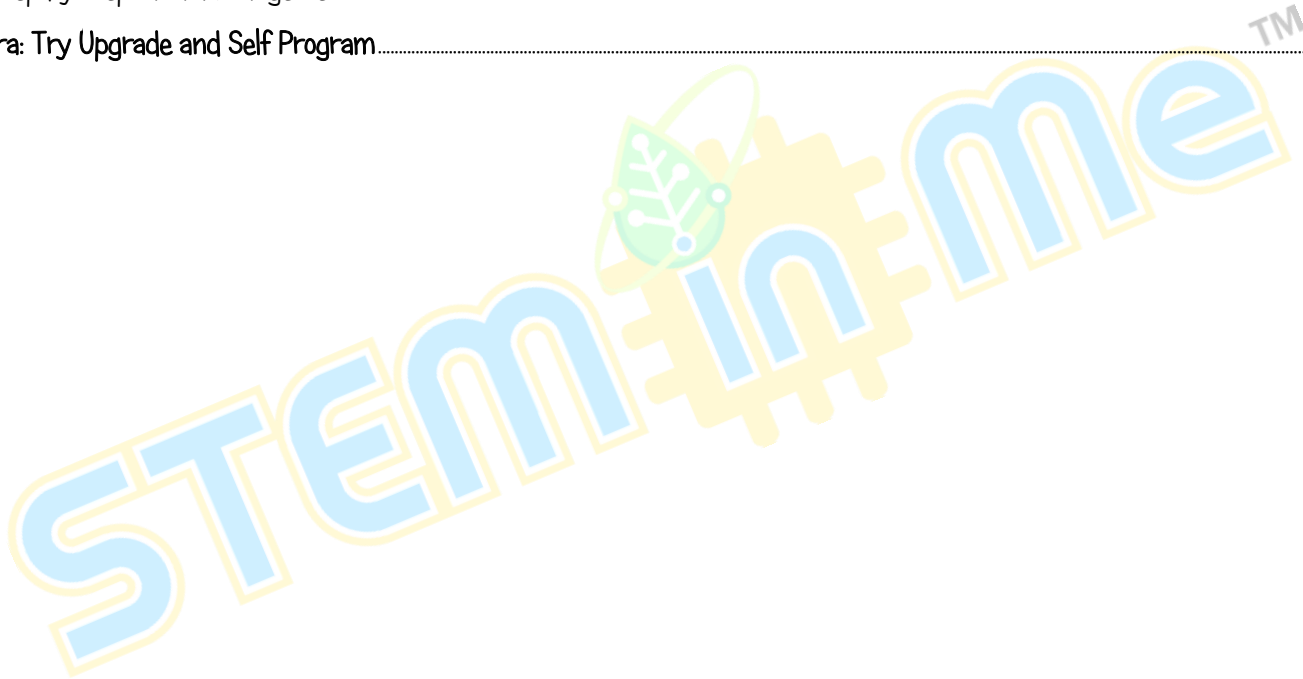
Type of Tracks.....	22
Type of Junctions.....	23
Objective 1: Vroom Vroom.....	24
Introduction to Buzzer.....	24
Step by Step block arrangement.....	24
Challenge.....	26
Objective 2: Please Switch On The Lights!.....	27
Introduction to Light-Emitting Diode (LED).....	27
Step by Step block arrangement.....	27
Challenge.....	29
Objective 3: Our Adventure Begin (Free Movement).....	30
Introduction to Motors.....	30
Introduction to Basic of Free Movement.....	31
Step by Step block arrangement.....	34
Challenge.....	40
Objective 4: Let's Follow The Line!.....	41

Introduction to Line Detector.....	41
Introduction to Line Tracer Time and its Mechanism.....	41
Step by Step block arrangement:.....	42
Challenge.....	46
Objective 5: What To Do When Meeting Junction?	47
Introduction to <i>Path Finder</i> and its Mechanism.....	47
Step by Step Block Arrangement.....	48
Challenge.....	53
Objective 6: What Else Can Be Done When Meeting Junction?	55
Introduction to <i>Path Finder Tank</i> and its Mechanism.....	55
Step by Step block arrangement:.....	56
Challenge.....	61
Objective 7: Wrong way? Make U-turn	63
Introduction to <i>Turn At Centre</i> and its Mechanism.....	63
Step by Step blocks arrangement.....	64
Challenge.....	66

Objective 8: Let's Control Mikrobotik	67
Introduction to Bluetooth and its Mechanism.....	67
Step by Step block arrangement:.....	68
Mikrobotik Mobile Apps.....	73
Challenge.....	75
Objective 9: We Need Area Patrol!	76
Introduction to Movement and Its Mechanisms.....	76
Step by Step Block Arrangement.....	79
Challenge!!.....	82
Objective 10: Let's Find Hidden Treasures	84
Introduction to Push Button.....	85
Introduction to Movement and Its Mechanisms.....	85
Step by Step Block Arrangement:.....	90
Challenge!!.....	94
Objective 11: Recycling Material Separation	95
Introduction to Single Gripper and Its Mechanism.....	96

Arrange Strategy and Movement Techniques.....	100
Step by Step block arrangement:.....	105
Extra Information.....	110
Challenge!!.....	111
Objective 12: Efficient Space Storage.....	112
Introduction to Dual Gripper and Its Mechanism.....	113
Installation of Dual Grippers on the robot.....	114
Arrange Strategy and Movement Techniques.....	117
Step by Step block arrangement:.....	123
Challenge!!.....	131
Objective 13: Stop! Obstacles in front.....	132
Introduction to Ultrasonic and Its Mechanism.....	132
Installation of Ultrasonic Sensor on the robot.....	133
Arrange Strategy and Movement Techniques.....	135
Step by Step block arrangement:.....	136
Challenge!!.....	140

Project 14: Lines of different colours? Let's finish it!.....	141
Introduction to Movement and its Mechanism	141
Step by Step Block Arrangement.....	145
Extra: Try Upgrade and Self Program.....	151





Mechanical Structure

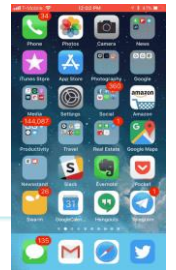


Mechanical Movement

Element of Robotics



Electronic Hardware



Software Coding

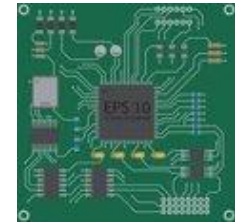
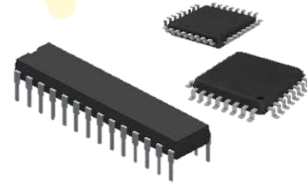
What is Electronic Hardware?



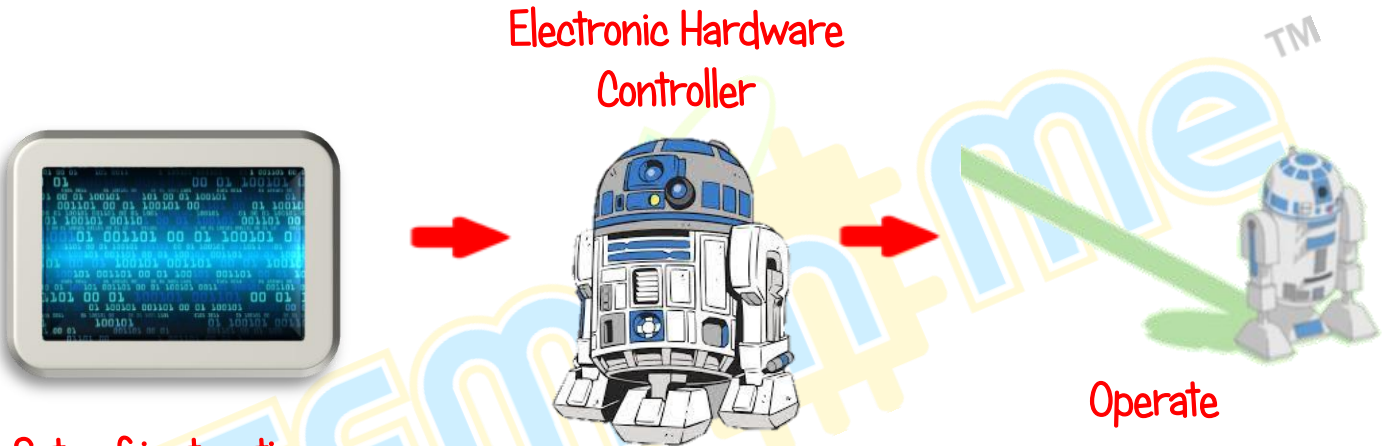
Detects and senses the surrounding



Controls or reacts to surrounding



What is Software Coding?



Sets of instruction written in specific language

Electronic Hardware Controller

Operate





Path Finding Robot

Robot designed and built specifically to detect and autonomously follow white and black line. Besides, robot also designed for other functions such as obstacle detector and moving small objects.

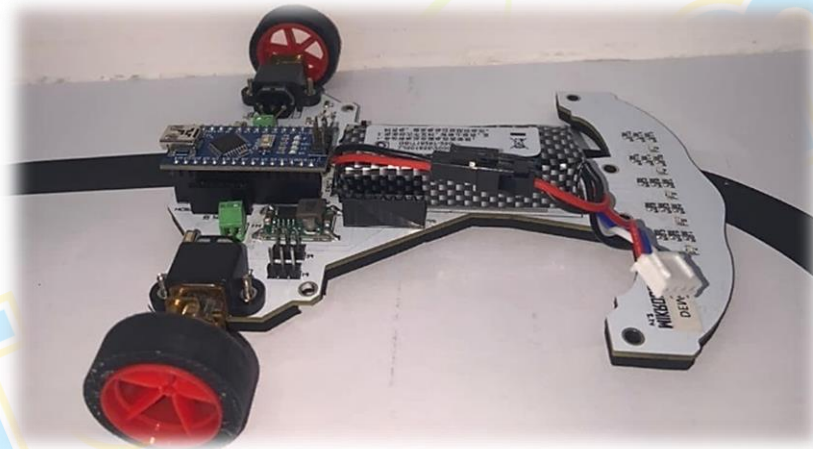


Figure 1: "MikroBotik" Pathfinding Robot



Contents in the box

- 1x Usb Cable
- 1x Charger
- 1x Mikrobotik
- 1x Bluetooth Module
- 1x Mikrobotik Track

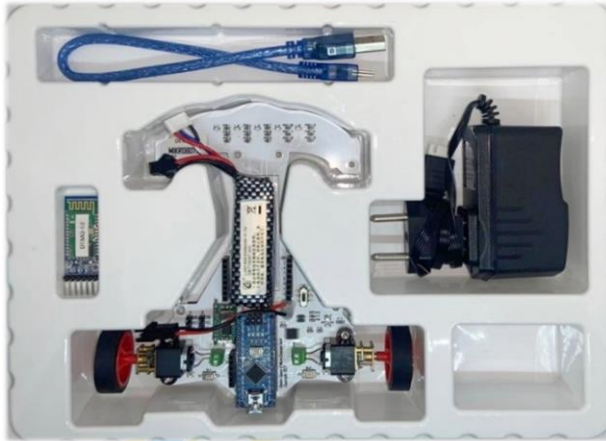


Figure 2: Mikrobotik Set

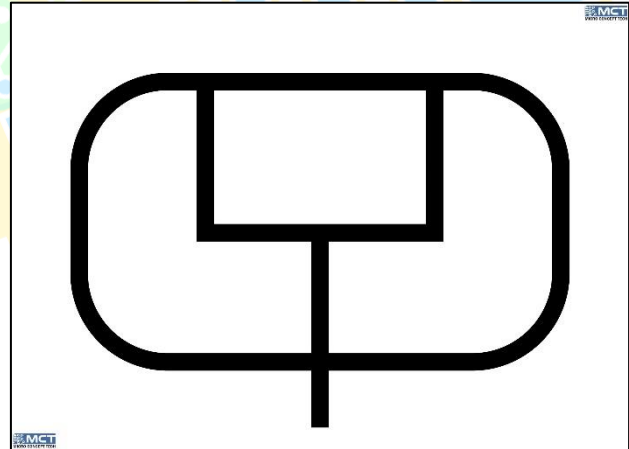
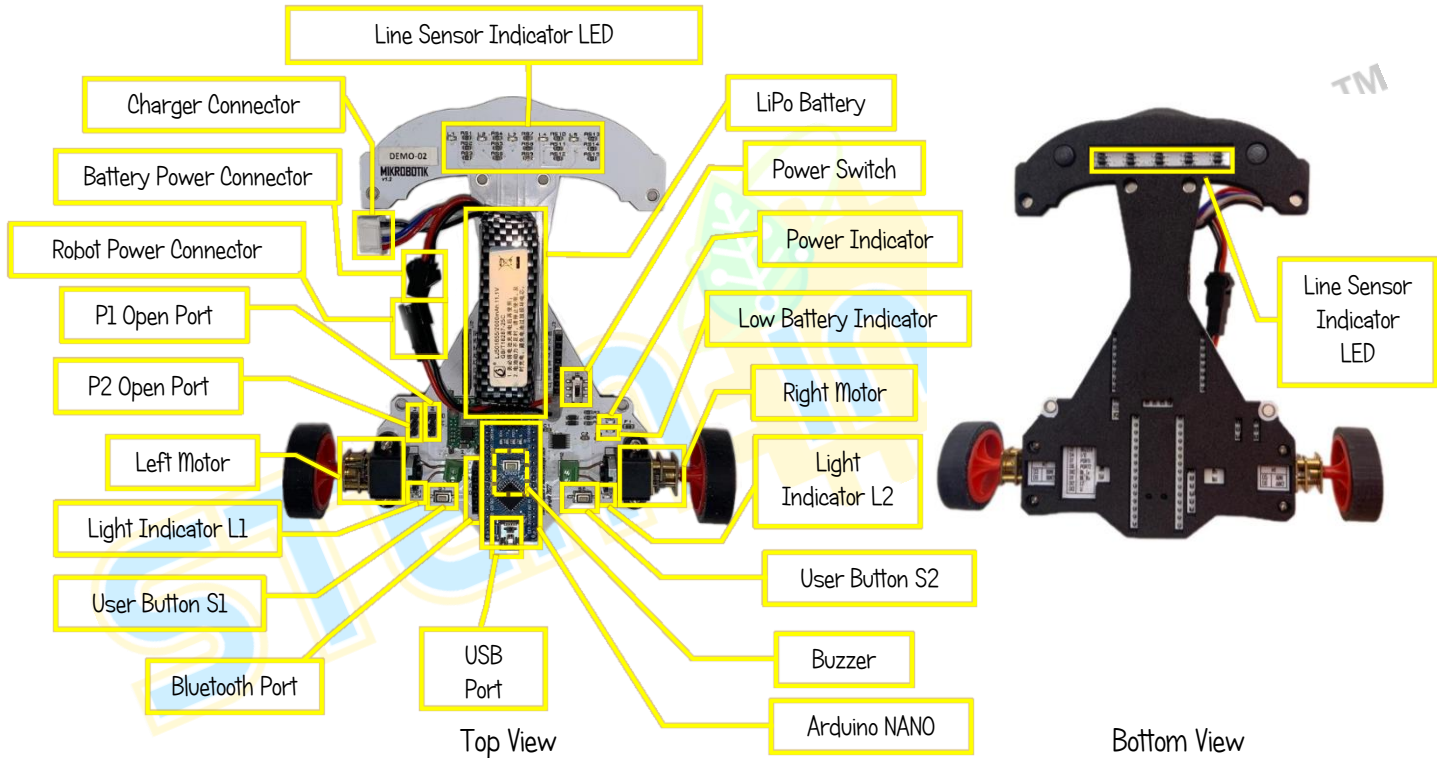


Figure 3: Mikrobotik Track

"Mikrobotik" Pathfinding Robot



Arduino Nano Microcontroller

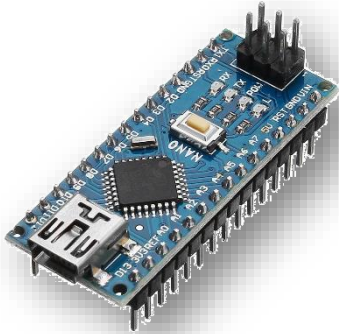
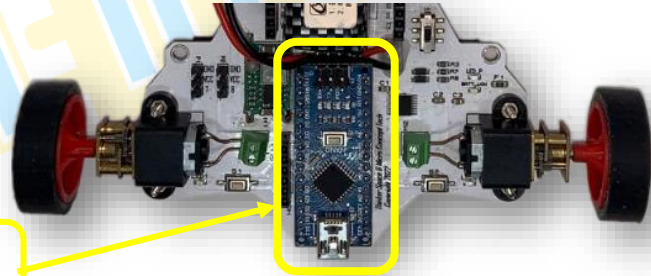


Figure 4: Arduino Nano Atmega328p

A microcontroller is a device that handles core functions such as controlling the use of other electronic hardware connected to it, analysing data and executing logic.

Mikrobotik uses an Arduino Nano microcontroller that acts as the brain to control the entire hardware and movement of the robot.



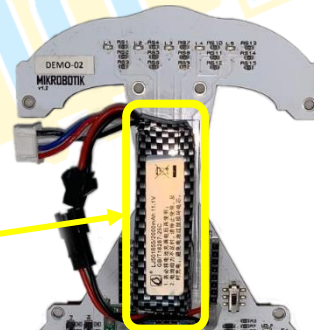
Arduino Nano microcontroller on Mikrobotik

LiPo Battery



Lithium Polymer (LiPo) batteries are rechargeable lithium-ion battery technology that uses a polymer electrolyte instead of a liquid electrolyte. They function by providing a higher specific energy than other types of lithium batteries and are used in applications where weight is a critical feature.

MikroBOTIK utilises an 11.1V LiPo battery to ensure that maximum speed can be achieved for movement.



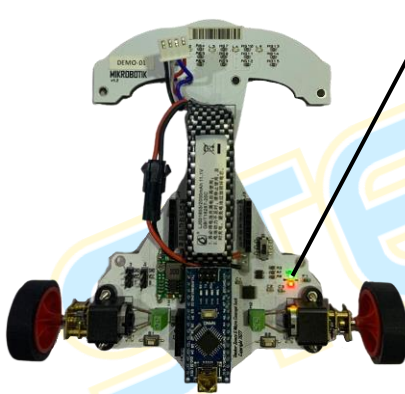
LiPo battery in MikroBOTIK

Low Battery Indicator

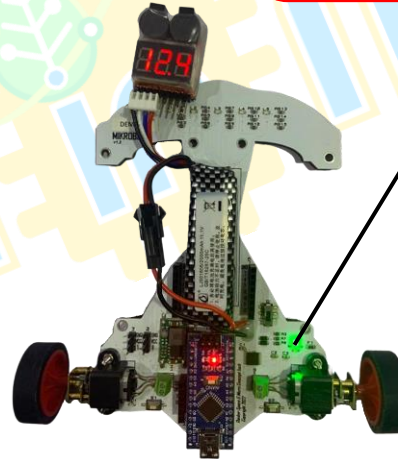
Low battery indicator will light up red colour.
The lower the voltage value in battery, the brighter the indicator.
Minimum voltage operated: 11.0 V (Low battery indicator light at maximum bright)



User need to stop using Mikrobotik and start charging when low battery indicator lights at maximum.



Battery indicator when battery is low.



Battery indicator when battery is full.



Installation of mBlock v5

Step 1

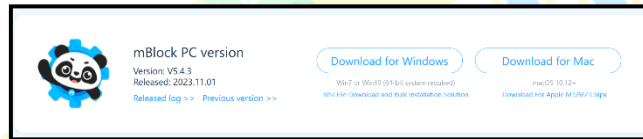
mBlock v5 software can be obtained from:

Link: <https://mblock.makeblock.com/en-us/download/> @ QR:



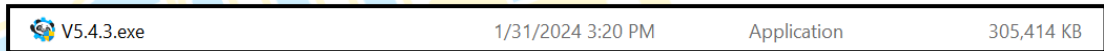
Step 2

Download the latest version of mBlock v5 based on the computer operating system.



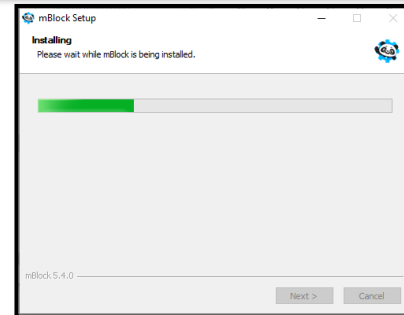
Step 3

Click mBlock v5 on your download location.



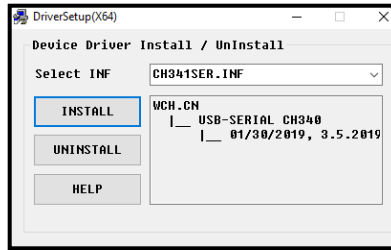
Step 4

Wait until mBlock v5 installation is complete.



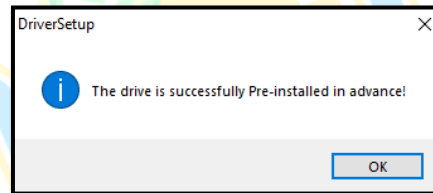
Step 5

Click *INSTALL*:



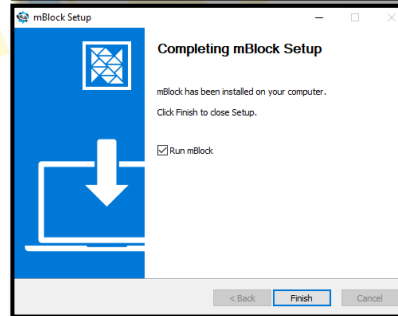
Step 6

Click OK and exit



Step 7

Tick *Run mBlock*.
Click *Finish*.



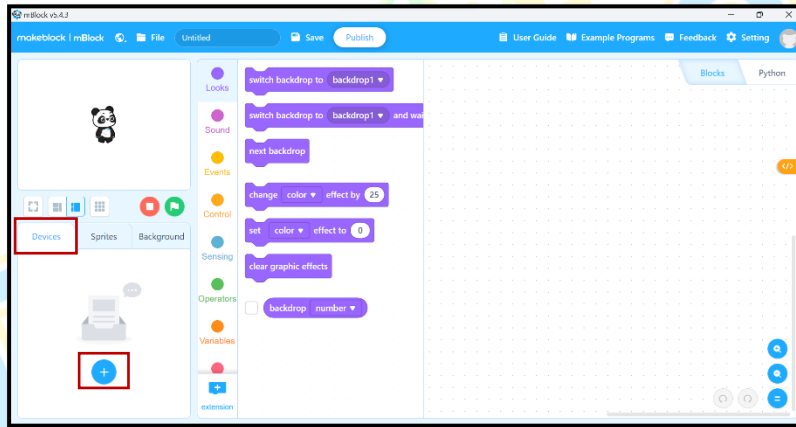


Steps for Adding Mikrobotik



Step 1 Open mBlock v5.4.3

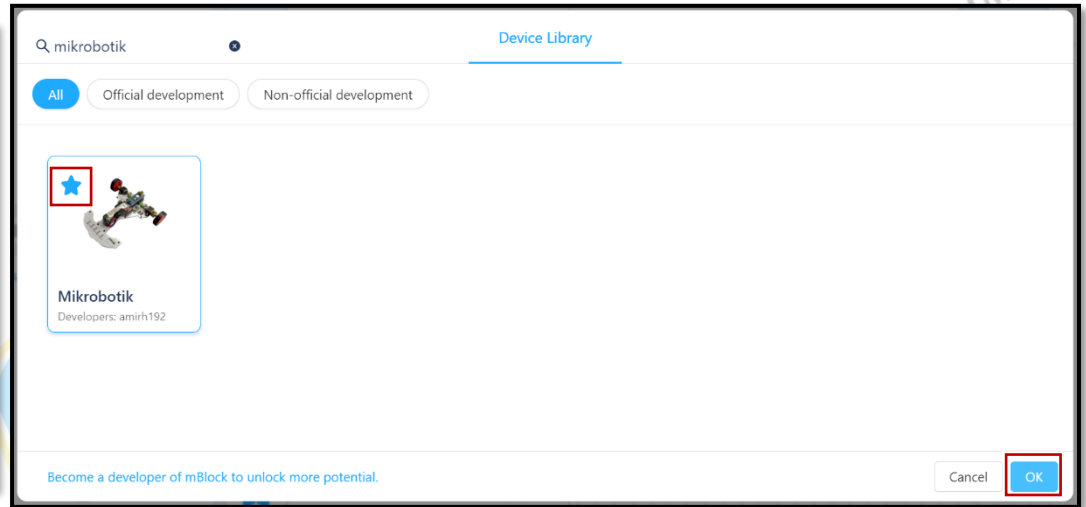
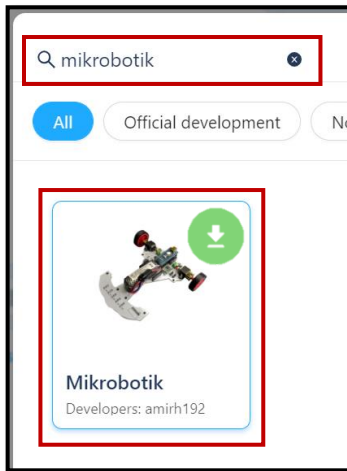
Step 2 On your mBlock, choose *Devices* and click *Add*





Step 3

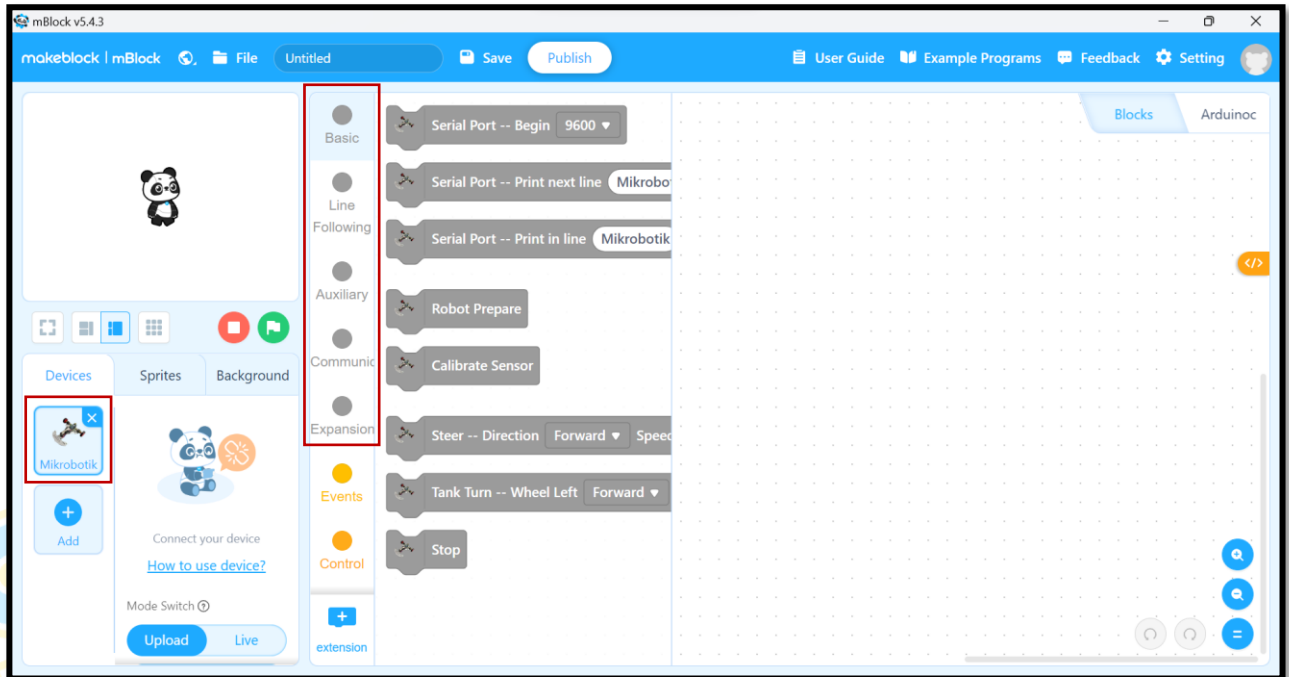
A new pop-up will appear once you click *Add*, search 'Mikrobotik', download and click 'Set as mostly used device' and click *OK*. Make sure that your laptop or computer has internet access.





Step 4

Mikrobotik has been added at *Devices*. Mikrobotik library will appear at the block drawer
You can start coding now.





Calibration Process

The calibration process is an important process for the robot to identify between the white line and the black line. The calibration process for this Mikrobotik robot can be done both manually and automatically. This process is done before the robot can follow the line and complete the circuit.

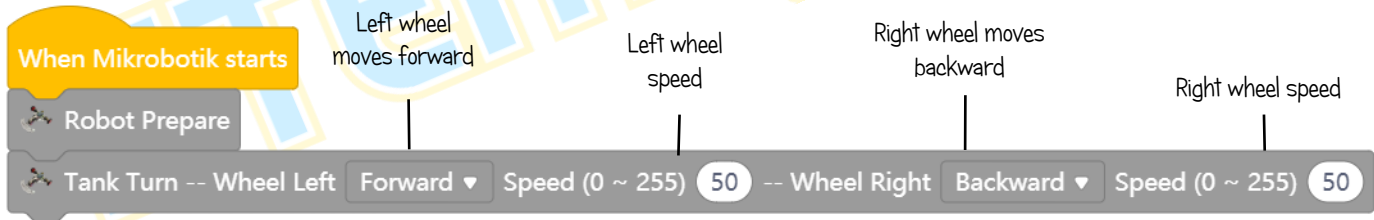
Block Arrangement (Automatic Calibration):

Step 1 Insert block *When Mikrobotik Starts* and combine with block *Prepare*.

When Mikrobotik starts

Robot Prepare

Step 2 Next, combine block *Tank Turn (Wheel Left-Forward, Speed-50, Wheel Right-Backward, Speed-50)* under *Robot Prepare*.



Step 3

Furthermore, combine block *Repeat* with block *Calibrate Sensor*.
Combine these blocks with blocks in Step 2.

Step 4

Lastly, drag *Stop* and block *Wait (1 second)* and put under block *Repeat*.

The image shows a Scratch script for Mikrobotik. It starts with a yellow 'When Mikrobotik starts' block. Below it are three grey blocks: 'Robot Prepare', 'Tank Turn -- Wheel Left Forward ▾ Speed (0 ~ 255) 50 -- Wheel Right Backward ▾ Speed (0 ~ 255) 50', and 'Calibrate Sensor'. A blue 'repeat 150' block is placed over the 'Calibrate Sensor' block. Inside the 'repeat' block, there is a grey 'Stop' block followed by an orange 'wait 1 seconds' block. Two lines with arrows point from text annotations to the 'Calibrate Sensor' and 'wait 1 seconds' blocks.

When Mikrobotik starts

Robot Prepare

Tank Turn -- Wheel Left Forward ▾ Speed (0 ~ 255) 50 -- Wheel Right Backward ▾ Speed (0 ~ 255) 50

repeat 150

Calibrate Sensor

Stop

wait 1 seconds

Mikrobotik performs calibration process while moving for few seconds

Mikrobotik stops for 1 seconds

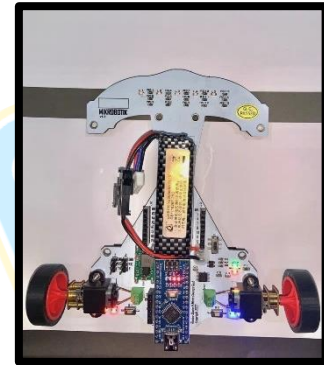
Steps for Automatic Calibration Process

Step 1

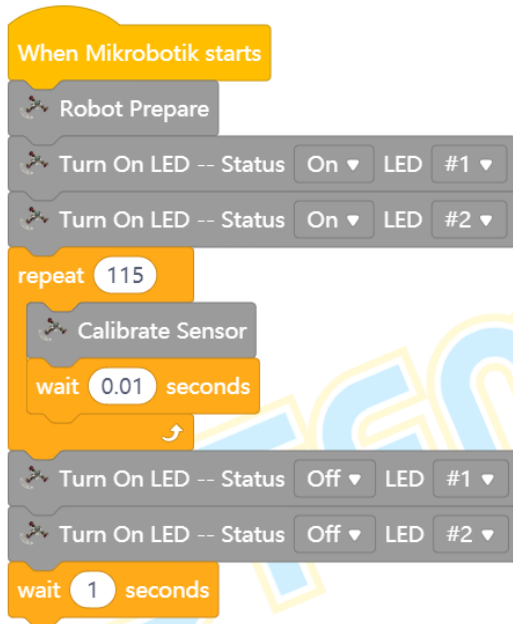
Put Mikrobotik on the track.
Make sure all sensors starting from TR1 (LED L1) until TR5 (LED L5) were put on black line.

Step 2

Switch on the power switch for Mikrobotik.
The robot will automatically rotate to carry out the calibration process.



Block Arrangement (Manual Calibration):

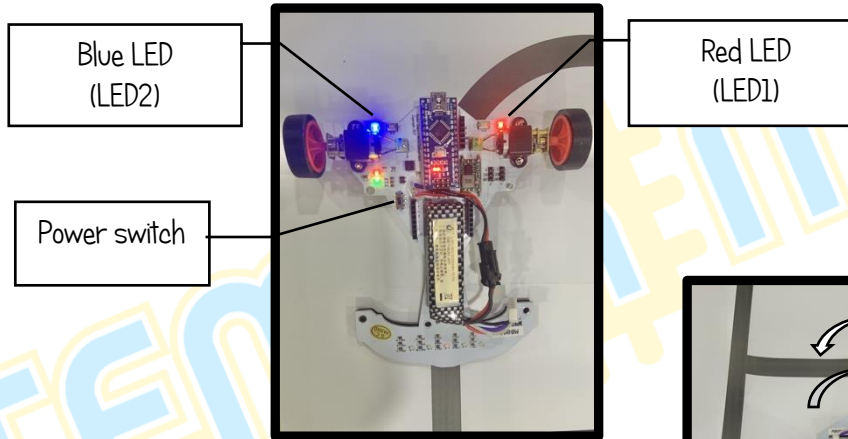


Steps for Manual Calibration Process

Step 1

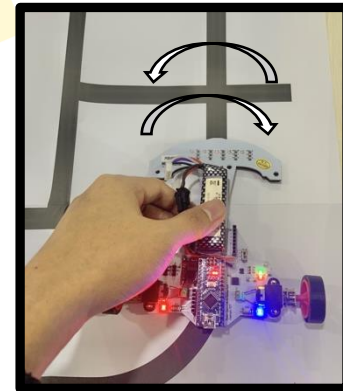
Switch on Mikrobotik.

LED1 with red colour and LED2 with blue colour will light up.



Step 2

Move all sensors starting from sensor labelled with TR1 (LED L1) until TR5 (LED L5) and bring back to TR1 (LED L1).

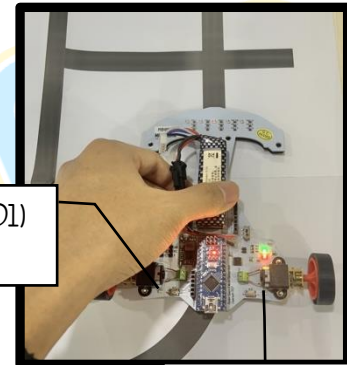


Step 3 Repeat movement in Step 2 until both LED (LED1 and LED2) light off.



Make sure all the sensors can detect the black line. The LED on the sensors will light up if the sensor detects a black line. For example, LED L1 will light up if sensor TR1 detects a black line on the circuit.

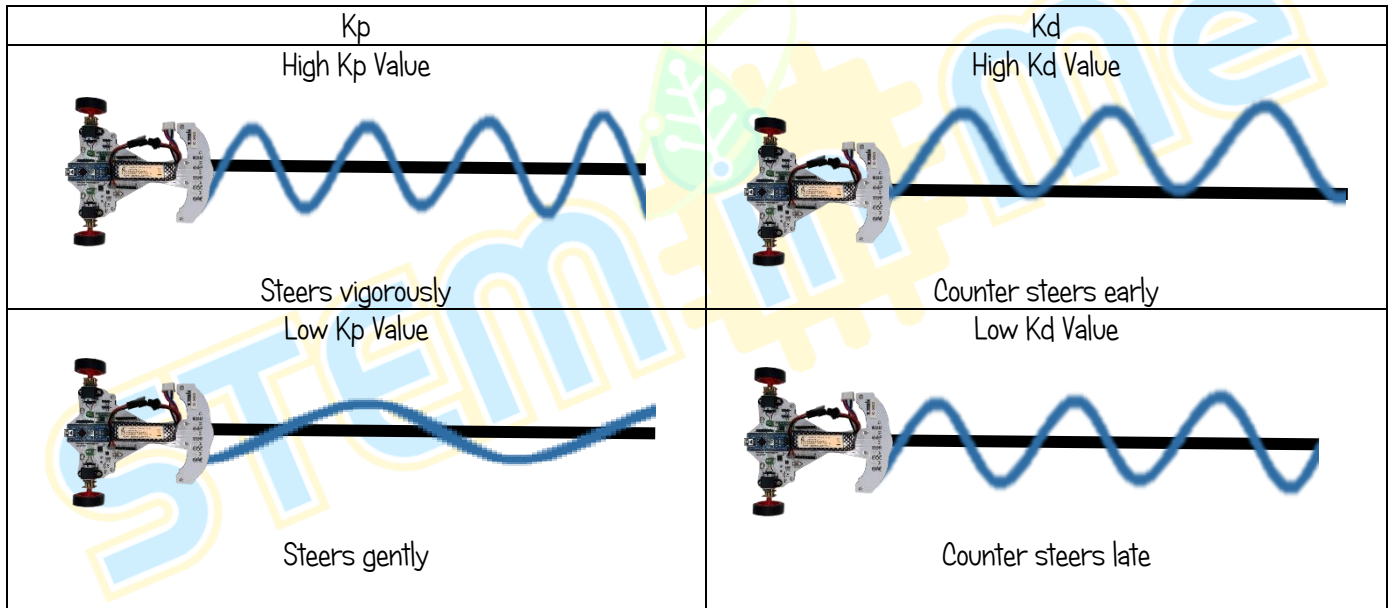
Red LED (LED1)
lights off.



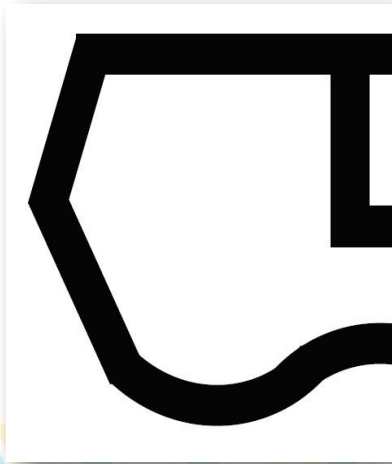
Blue LED (LED2)
lights off

Autonomous Robot PID Algorithm

PID algorithm is a control strategy suitable to assist determine the direction and speed of the robot such that it autonomously drive and follows the line as close and fast possible. PID algorithm will ensure the robot does not overshoot from the line track when turning and moves straight along the line.



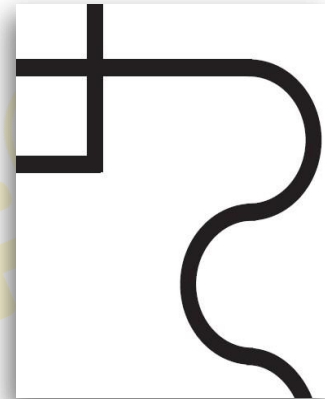
Type of Tracks



Black Line
(aprox 20mm)

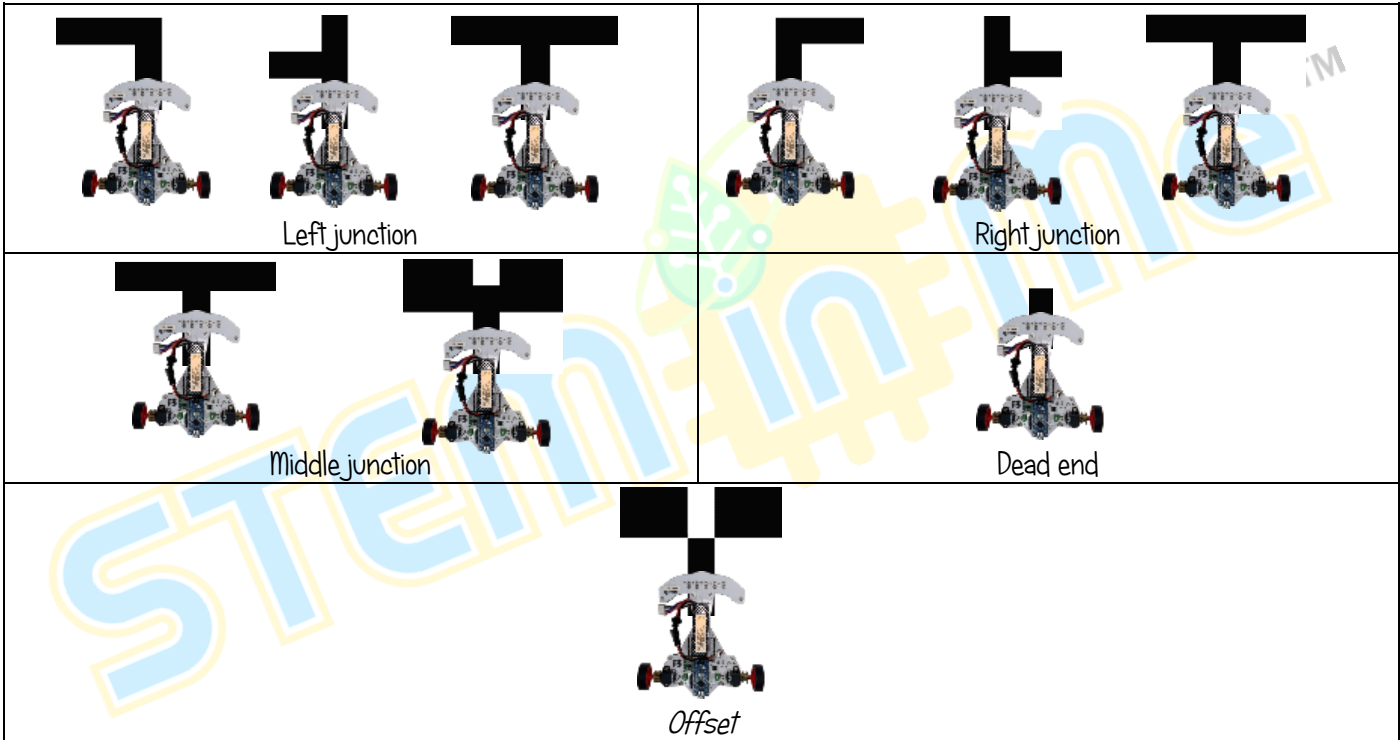


White Line
(aprox 20mm)



Black Thin Line
(aprox 10mm)

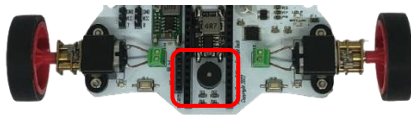
Type of Junctions



Objective 1: Vroom Vroom

The robot will use a buzzer to produce a simple sound. It can only produce one tone at a time. This code block can be used to produce different tones to create an interesting sound pattern.

Introduction to Buzzer



Buzzer is a type of sound device that converts audio models into sound signals. It is usually used for alarms.

Step by Step block arrangement:

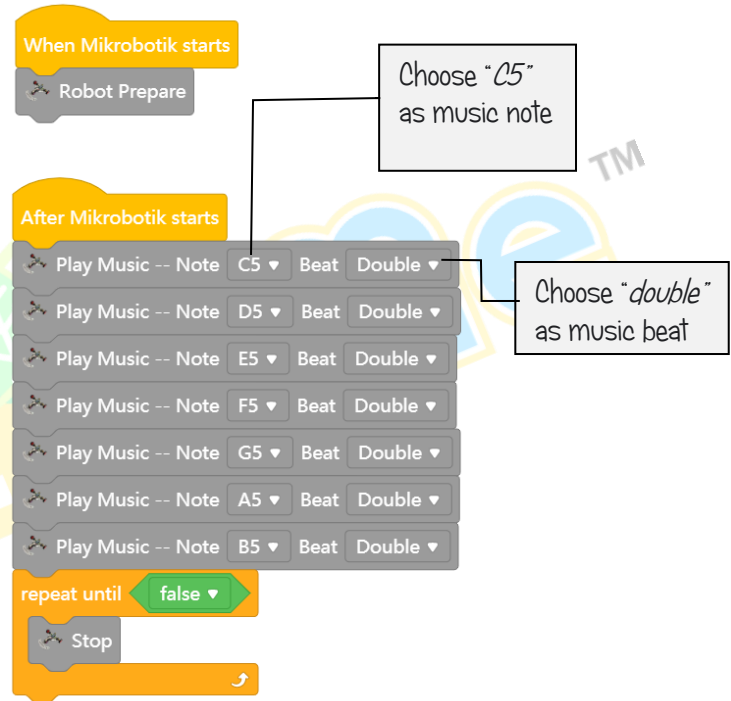
Step 1 Step 1 Combine block *When Mikrobotik starts* with block *Robot Prepare*.



This block is to prepare robot with specific *library* and to configure pin numbers and pot numbers in and out for each sensor and output attached to the robot.

Step 2

Next, combine block *After Mikrobotik starts* with block *Play Music (Note-C5, Beat-Double)*. (Note-D5, Beat-Double). (Note-E5, Beat-Double). (Note-F5, Beat-Double). (Note-G5, Beat-Double). (Note-A5, Beat-Double). (Note-B5, Beat-Double)



Step 3

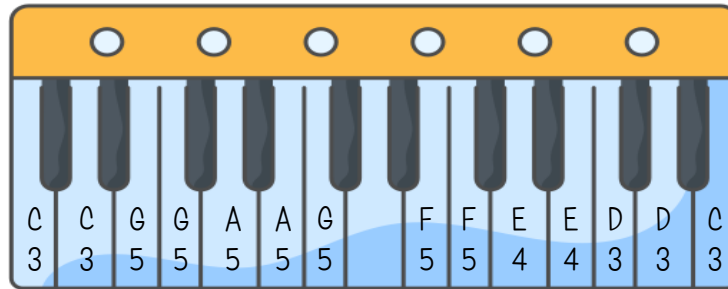
Then, combine block *repeat until (false)* with block *stop*. Combine these blocks with blocks in Step 2.

Step 4

Once the program is uploaded, the robot will produce the sound or tone you have entered.

Challenge!!

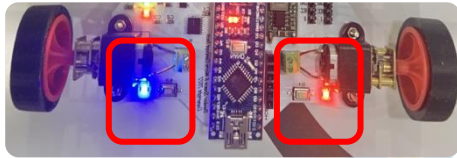
In this challenge, you have to enter the provided musical notes and try to guess the name of the music.



Objective 2: Please Switch On The Lights!

Light-Emitting Diode (LED) on robot used as indicators. LEDs on the robot can be seen at power indicator, low battery indicator, LED1, LED2, Arduino NANO and line sensors.

Introduction to Light-Emitting Diode (LED)



Light-Emitting Diode or LED functions to convert electric current into light and emit light. Used as an application for indicator and light source.

Step by Step block arrangement:

Step 1

Combine block *When Mikrobotik starts* with block *Robot Prepare*

When Mikrobotik starts

Robot Prepare



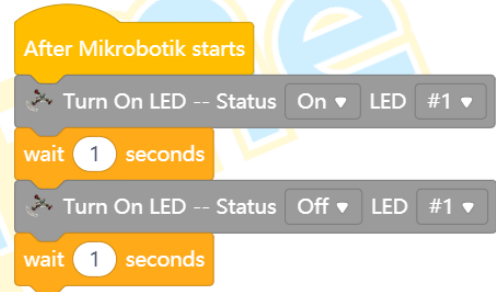
Step 2

Combine block *After Mikrobotik starts* with block *Turn On LED* with choice of *Status On* and LED #1 and block *wait 1 second*. Put the blocks under the block in Step 1. This program will light up the LED.



Step 3

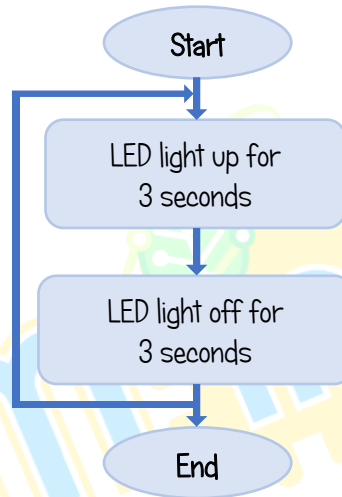
Add another block *Turn On LED* with choice of *Status Off* and LED #1 with block *wait 1 second* and combine with block in Step 2 to switch off the LED.



Step 4

Lastly, upload the program. After the program uploaded, LED 1 will light up in one second and light off in one second. The program will continue to run until the robot is turned off by the user.

Challenge!!

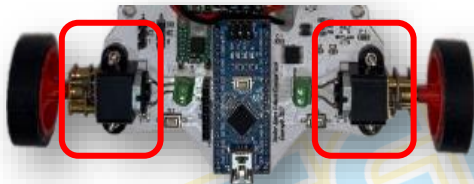


Program above show LED will light up in three second and light off in three second. The program will continue to run until the Mikrobotik is turned off by the user.

Objective 3: Our Adventure Begin (Free Movement)

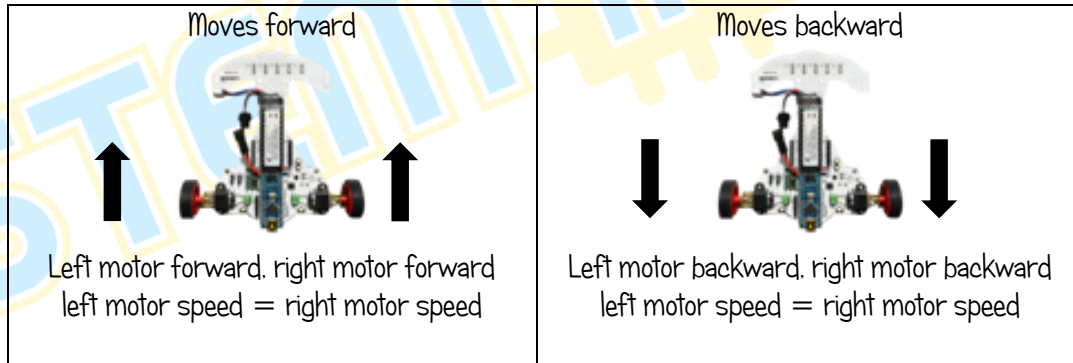
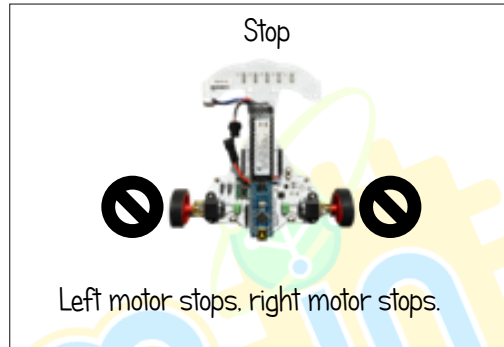
The robot is moved using the code block "tank turn" to move without following the line. This code block is suitable for solving maze circuits (labyrinth). The robot will move depending on the speed and direction of the left and right motors set by user.



Introduction to Motors





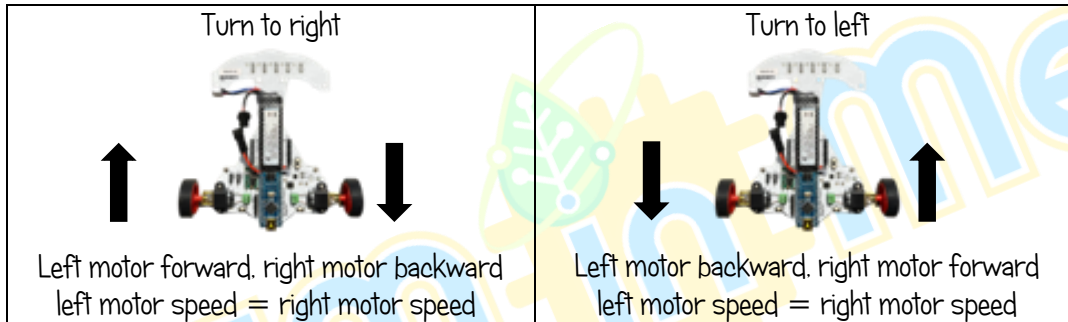
There are two motors on MikroBOTIK that can be controlled separately. can rotate clockwise and counter-clockwise continuously. These motors can also be used to move or drive the project. The speed and duration can also be set.

Introduction to Basic of Free Movement:



<p>Steer to right</p>  <p>Left motor forward, right motor forward. Left motor speed > Right motor speed</p>	<p>Steer to left</p>  <p>Left motor forward, right motor forward. Left motor speed < Right motor speed</p>
---	---

<p>Sharp steer to right</p> 	<p>Sharp steer to left</p> 
---	---



Step by Step block arrangement

i) Move forwards

Step 1 Combine block *When Mikrobotik starts* with block *Robot Prepare*.

When Mikrobotik starts

Robot Prepare

Step 2 Combine block *After Mikrobotik starts* with block *Tank Turn (Wheel Left -Forward. Speed-100. Wheel Right-Forward. Speed-100)*, block *wait(3 seconds)* and block *stop*. Drag those blocks and put it under the blocks in Step 1.
Left motor and right motor will move with the same speed.

When Mikrobotik starts

Robot Prepare

Left wheel moves forward

Left wheel speed

Right wheel moves forward

Right wheel moves

After Mikrobotik starts

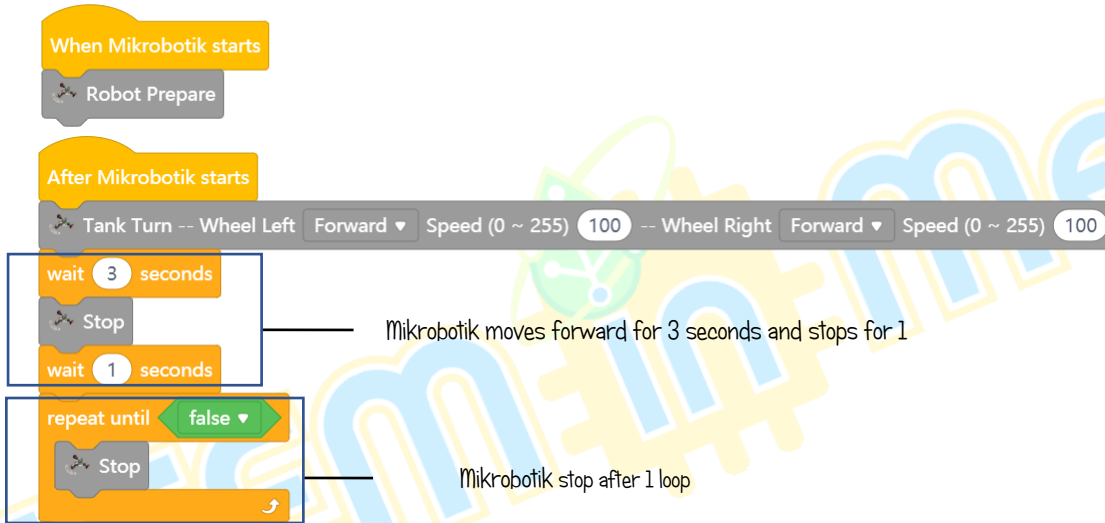
Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 100 -- Wheel Right Forward Speed (0 ~ 255) 100

wait 3 seconds

Stop

Step 3

Lastly, combine block *wait*, block *repeat until (false)*, and block *stop* and combine those blocks with blocks in Step 2.



Step 4

After the program uploaded, Mikrobotik will move forwards for duration of 3 seconds.



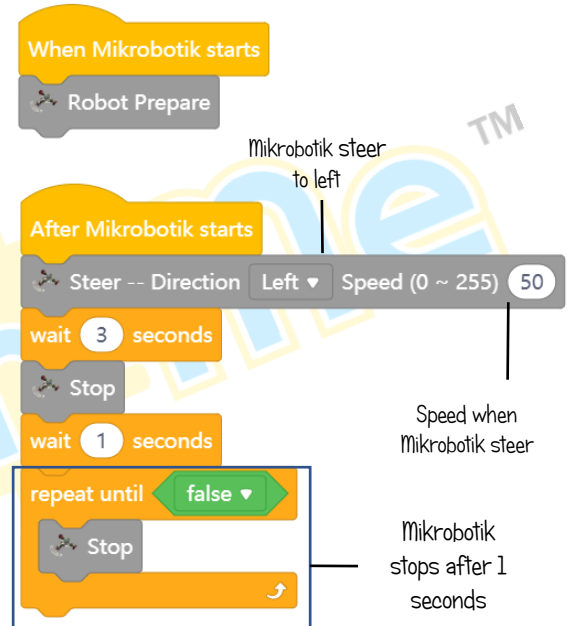
Step to make the robot move backwards is the same as step to move forwards. You only need to change the direction of the left wheel to backward and the direction of the right wheel to backward.

ii) Steer to left

Step 1 Combine block *When Mikrobotik starts* with block *Robot Prepare*.

Step 2 Combine block *After Mikrobotik starts* with block *Steer (Direction -Left. Speed-50)*. Then, add block *wait (3 seconds)* and block *stop*. Left motor will stop and right motor will move forwards with the set up speed.

Step 3 Lastly, combine block *wait (1 second)*, *repeat until (false)* and block *stop* and combine those blocks with blocks in Step 2.



Step 4

After the program uploaded, Mikrobotik will steer to left for duration of 3 seconds.



Step for steer to the right is similar to the step for steer to the left. You just need to change the direction to right.

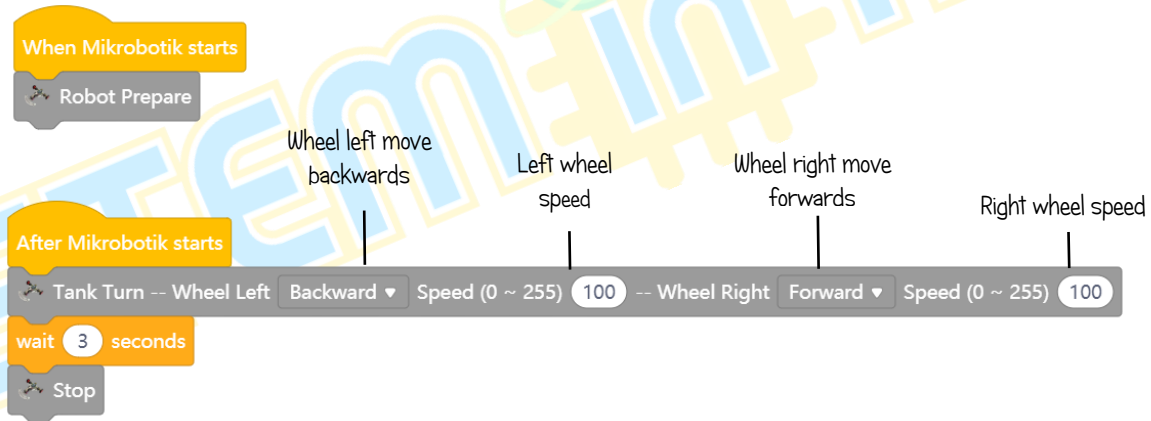
STEM in me™

iii) Turns left

Step 1 Combine block *When Mikrobotik starts* with block *Robot Prepare*.

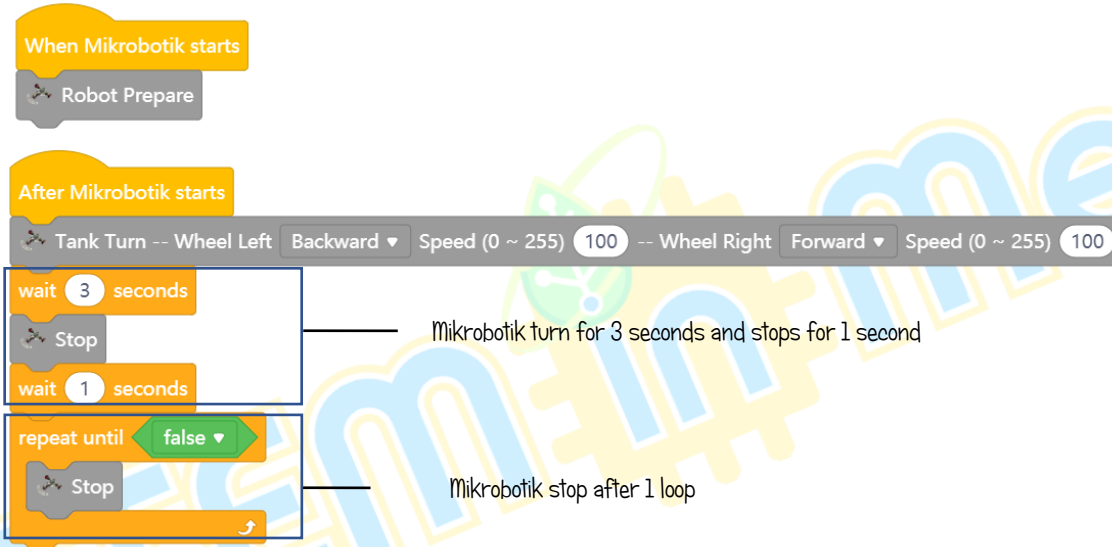


Step 2 Combine block *After Mikrobotik starts* with block *Tank Turn (Wheel Left -Backward, Speed-100, Wheel Right-Forward, Speed-100)*, block *wait (3 seconds)* and block *stop*. Drag those blocks to under block in Step 1. The left motor will move backward and the right motor will move forward with the same speed.



Step 3

Lastly, combine block *wait (1 second)*, *repeat until (false)* and block *stop*. Then, combine those blocks with blocks in Step 2.



Step 4

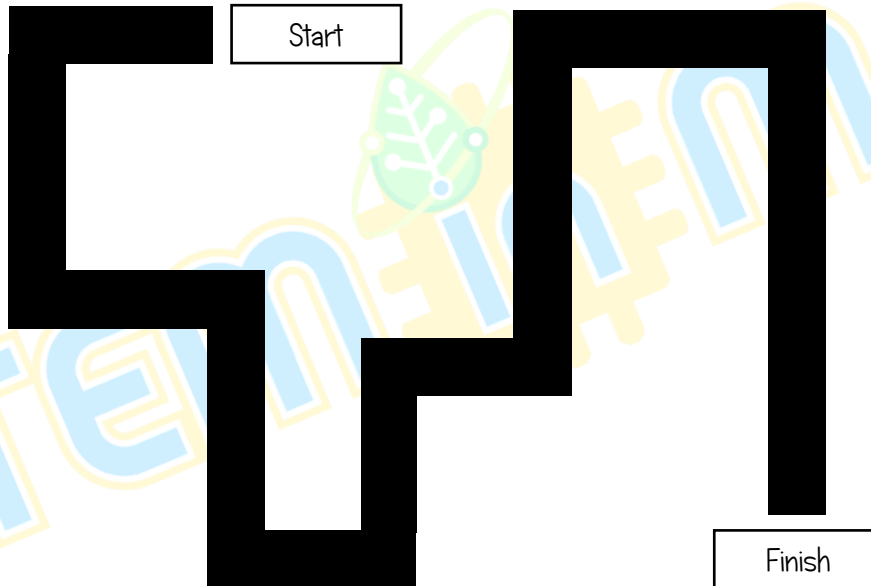
After program uploaded, Mikrobotik turns left for 3 seconds and stops.



Step for turn to the right is similar to the step for turn to the left. You just need to change the wheel left direction to forward and wheel left direction to backward.

Challenge!!

In this Challenge, you need to ensure that Mikrobotik moves along the path that has been prepared by applying the knowledge that has been learned.



Objective 4: Let's Follow The Line!

The robot will follow the line (Black or White) continuously. The robot will always move even if it meets a left or right junction.

Introduction to Line Detector



The line detector will emit infrared light and detect black or white surfaces. The analogue reading value will be high if a black surface is detected while the analogue reading value will be low when a white surface is detected.

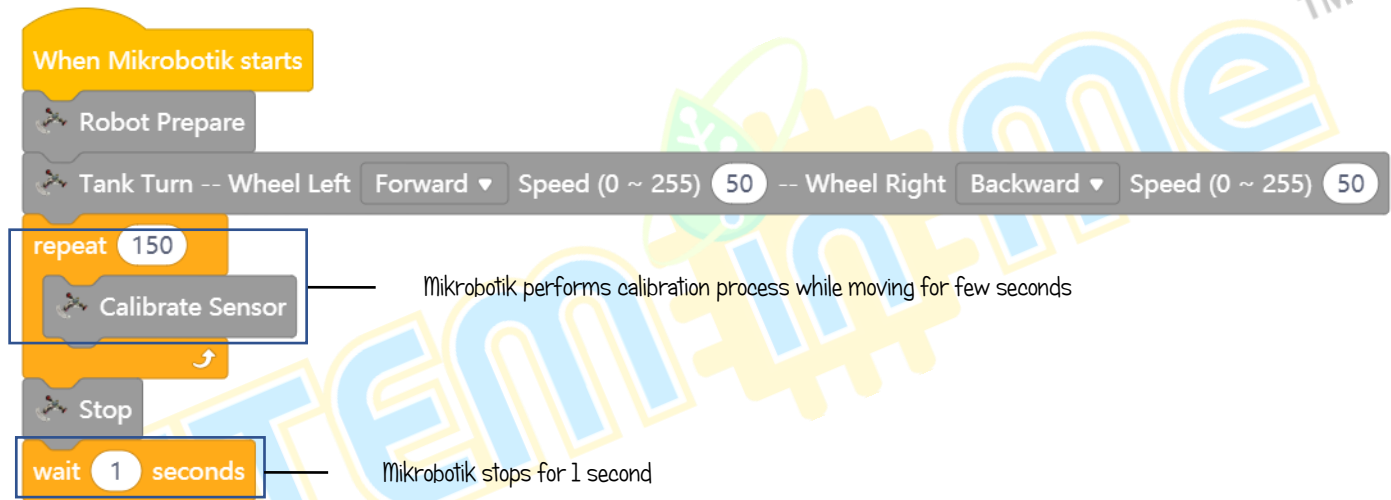
Introduction to Line Tracer Time and its Mechanism

Line Tracer Time used to make Mikrobotik follow along line with Black or White colour until it reach maximum time period (in ms).

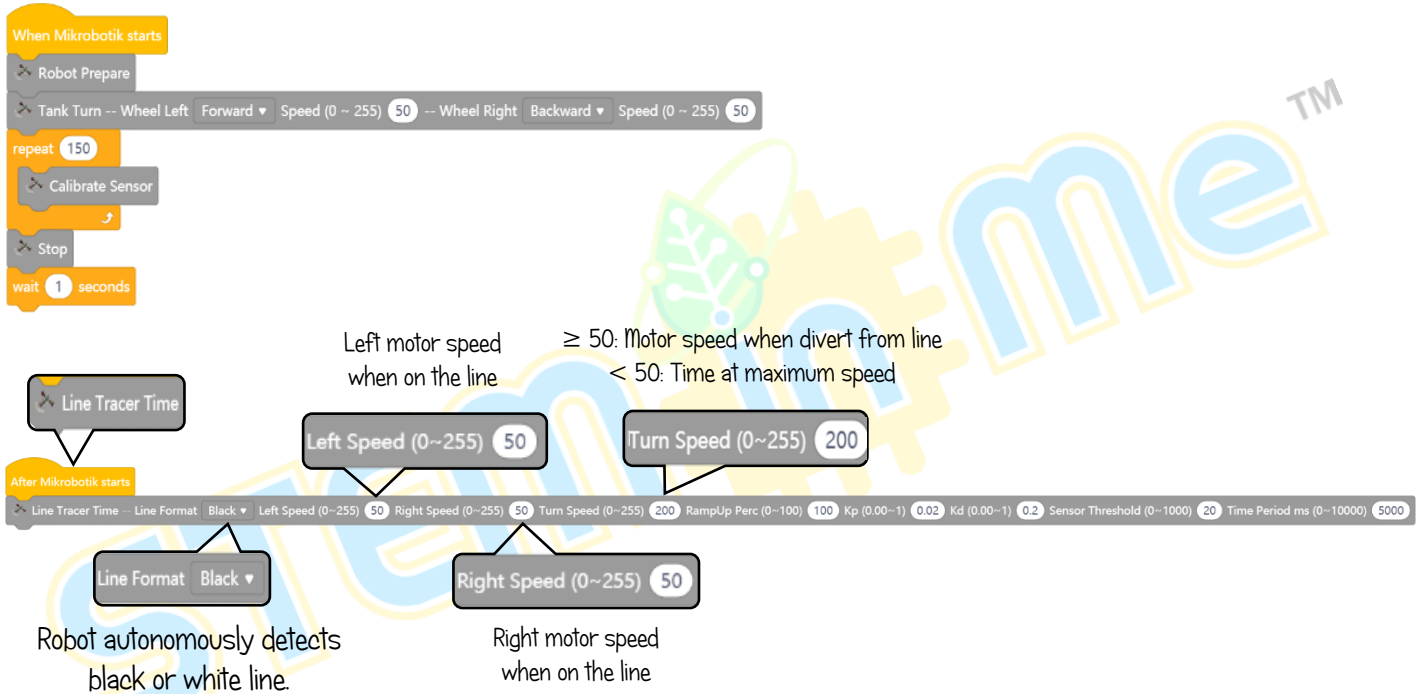
When Mikrobotik reaches the maximum time period, Mikrobotik will stop. Mikrobotik will move continuously without making turn when meeting left junction, right junction and middle junction.

Step by Step block arrangement:

Step 1 Prepare the arrangement blocks for automatic calibration.



Step 2 Then, insert block *After Mikrobotik Starts* and combine with block *Line Tracer Time*.



Continuation

After Mikrobotik starts

Line Tracer Time | Line Format | Black | Left Speed (0~255) 50 | Right Speed (0~255) 50 | Turn Speed (0~255) 200 | RampUp Perc (0~100) 100 | Kp (0.00~1) 0.02 | Kd (0.00~1) 0.2 | Sensor Threshold (0~1000) 20 | Time Period ms (0~10000) 5000

Control Kp value
Kp (0.00~1) 0.02

Line sensitivity
(higher values mean more sensitive)
Sensor Threshold (0~1000) 20

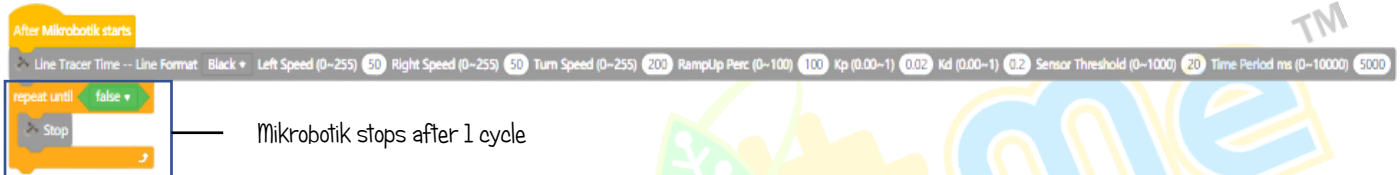
RampUp Perc (0~100) 100
Acceleration percentage

Kd (0.00~1) 0.2
Line sensitivity
(higher values mean more sensitive)

Time Period ms (0~10000) 5000
Maximum time (ms) for robot move forward

Step 3

Combine the "repeat until (false)" block and the "stop" block. Then, integrate these blocks with the block in Step 2.



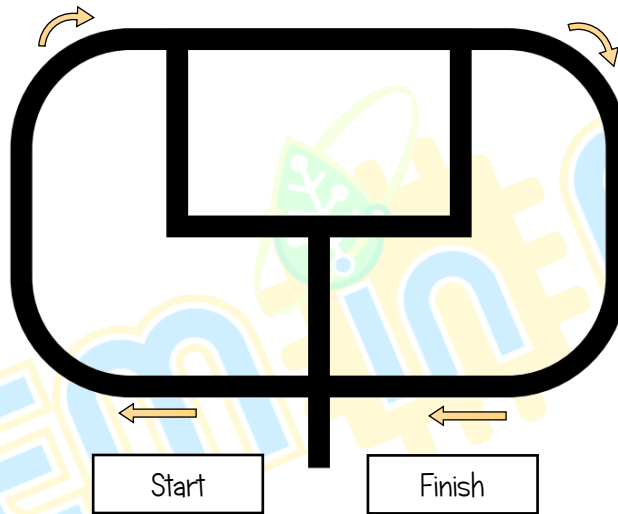
The screenshot shows a code editor interface for MikroBotik. At the top, there is a header bar with various settings: Line Tracer Time, Line Format (Black), Left Speed (50), Right Speed (50), Turn Speed (200), RampUp Perc (100), Kp (0.02), Kd (0.2), Sensor Threshold (20), and Time Period ms (5000). Below the header, a code block is visible. It starts with a yellow 'After MikroBotik starts' block, followed by a 'repeat until false' block. Inside the 'repeat until false' block, there is a 'stop' block. A line points from the 'stop' block to the text 'MikroBotik stops after 1 cycle'.

Step 4

After uploading the code, MikroBotik will start moving forward temporarily. Perform the calibration process on the line sensor. After that, MikroBotik will follow the line of either Black or White until it reaches the maximum time period (in ms).

Challenge!!

Apply *Line Tracer Time* to solve the track below.



Objective 5: What To Do When Meeting Junction?

Robot will move autonomously and can decide whether to turn left, turn right or stop at the junction. The technique used to know as *Steer Turn Method*.

Introduction to *Path Finder* and its Mechanism

Path Finder is used to move the Mikrobotik autonomously follow over a white or black line until the Mikrobotik finds a junction (right or left or middle or dead end or offset).

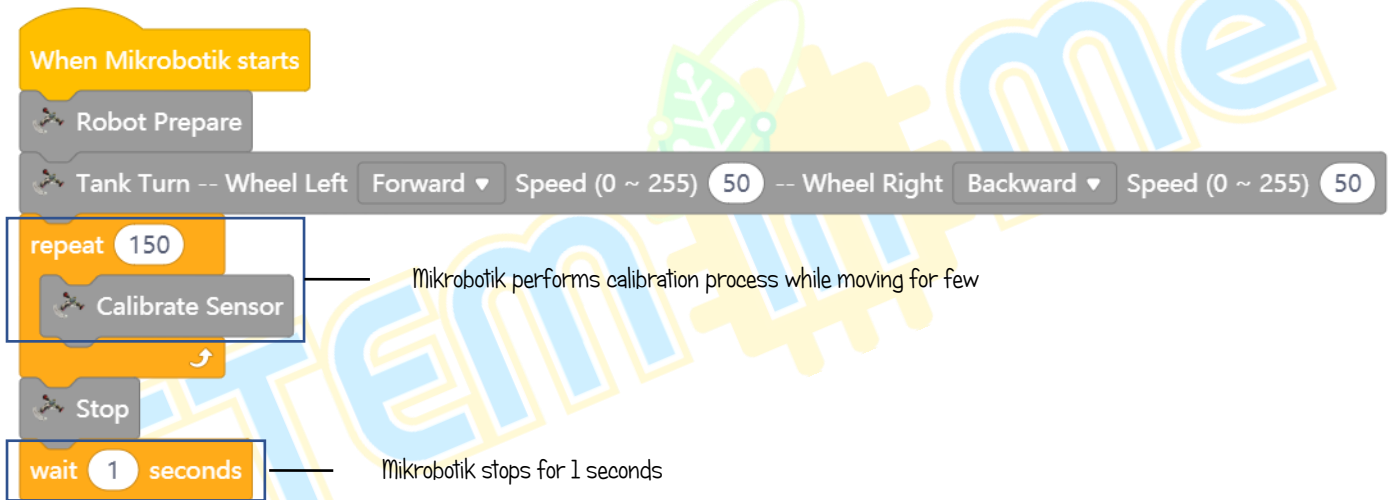
At the junction, Mikrobotik will act to turn (left or right or stop) for a set period of time or until the robot finds the next line and will stop.

Robot will turn by using *Steer Turn* method.

Step by Step Block Arrangement

Step 1

Prepare the blocks arrangement for automatic calibration.



Step 2

Lastly, combine block *After Mikrobotik Starts* and block *Path Finder (Line Format-Black Junction-Left, Action-Turn Left, Left Speed-50, Right Speed-50, Turn Speed-200, RampUp Perc-100, Kp-0.02, Kd-0.2, Sensor Threshold-20, Junction Speed-50, Forward Delay-50, Turn Period ms-300)*.

When Mikrobotik starts

- Robot Prepare
- Tank Turn -- Wheel Left Forward ▾ Speed (0 ~ 255) 50 -- Wheel Right Backward ▾ Speed (0 ~ 255) 50
- repeat 150
 - Calibrate Sensor
- Stop
- wait 1 seconds

Robot autonomously detects black or white line.

Robot action when meeting junction

Line Format Black ▾

Action Turn Left ▾

Path Finder

Junction Left ▾

Left Speed (0~255) 50

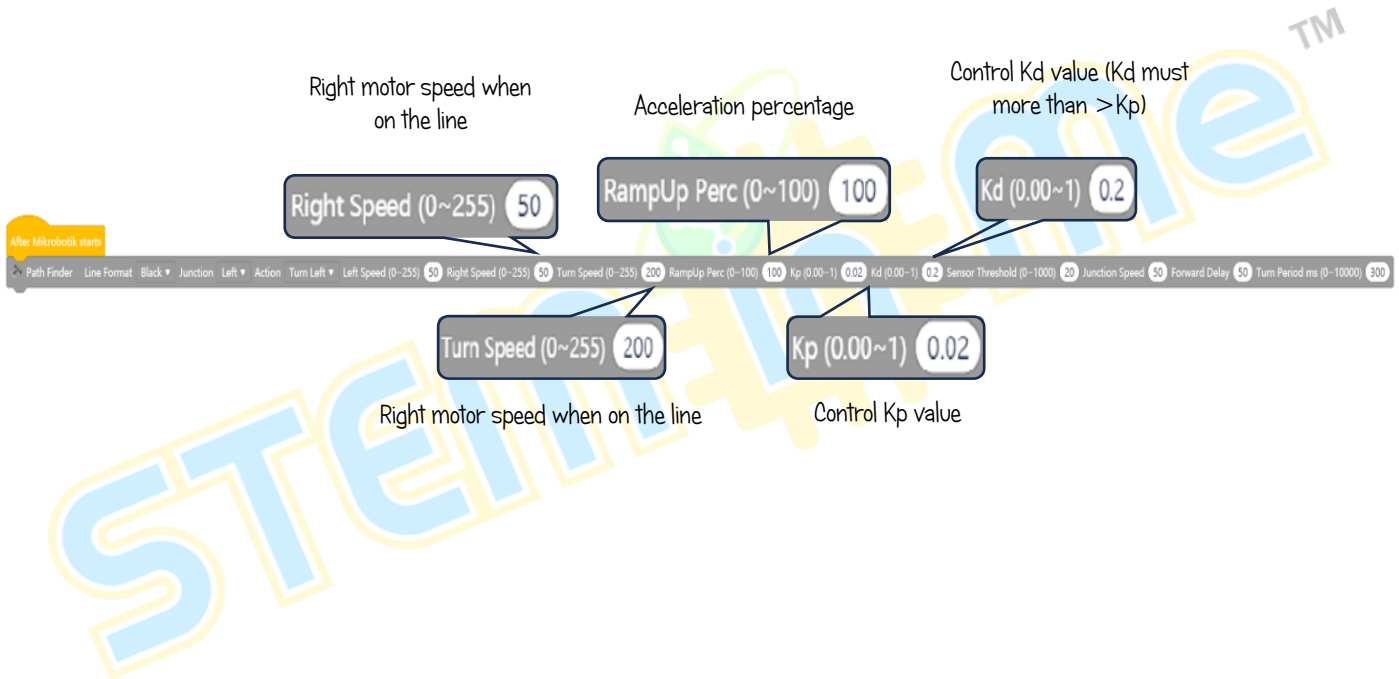
Robot tracking and move forward until meeting

Left motor speed when on the line

After Mikrobotik starts

Path Finder Line Format Black ▾ Junction Left ▾ Action Turn Left ▾ Left Speed (0-255) 50 Right Speed (0-255) 50 Turn Speed (0-255) 200 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 50 Forward Delay 50 Turn Period ms (0-10000) 300

Continuation



Continuation

After Mikrobotik starts

Path Finder Line Format: Black ▾ Junction Left ▾ Action Turn Left ▾ Left Speed (0~255) 50 Right Speed (0~255) 50 Turn Speed (0~255) 200 RampUp Parc (0~100) 100 Kp (0.00~1) 0.02 Kd (0.00~1) 0.2 Sensor Threshold (0~1000) 20 Junction Speed 50 Forward Delay 50 Turn Period ms (0~10000) 300

Turning speed at junction (higher value, sharper turns)

Time for robot move forward after meeting junction and make action

Junction Speed 50

Forward Delay 50

Sensor Threshold (0~1000) 20

Turn Period ms (0~10000) 300

Line sensitivity (higher values mean more sensitive)

On time (ms) for robot to turn or till find line (0)

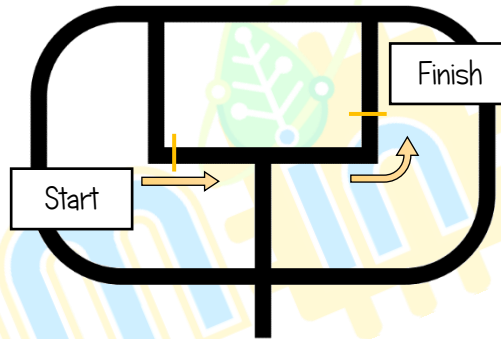
Step 3 Combine the "repeat until (false)" block and the "stop" block. Then, integrate these blocks with the block in Step 2.



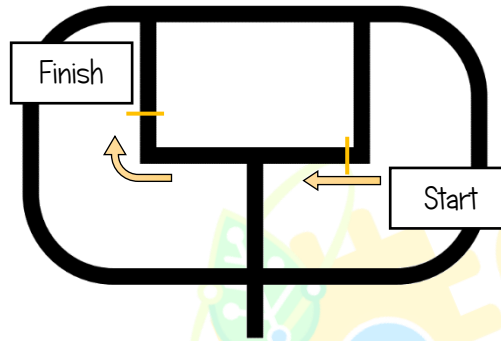
Step 4 After uploading the code, turn on the Mikrobotik switch and perform the calibration process. After that, Mikrobotik will follow the black line and if the robot meets a left junction, Mikrobotik will move forward and then turn to enter the left junction until Mikrobotik meets another line.

Challenge!!

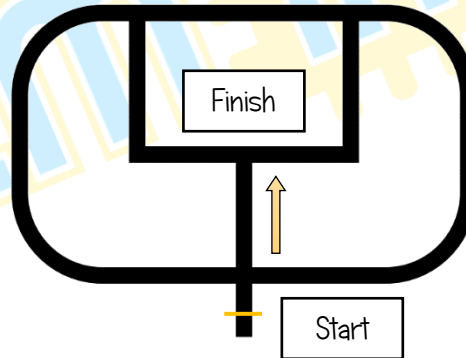
- i) *Path Finder* Left Junction, steer left at left junction.



ii) *Path Finder Right Junction*, steer right at right junction.



iii) *Path Finder Middle Junction*, stop.



Objective 6: What Else Can Be Done When Meeting Junction?

Robot will move autonomously and can decide whether to turn left, turn right or stop at the intersection. The technique is known as *Tank Turn Method*.

Introduction to *Path Finder Tank* and its Mechanism

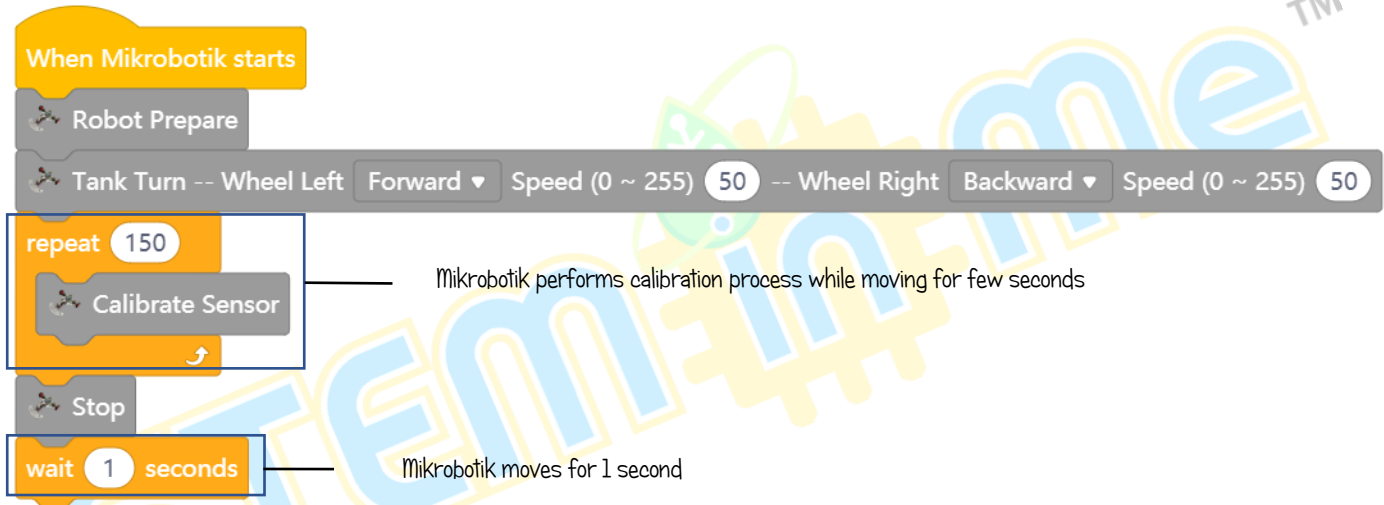
Mikrobotik travel autonomously on lines (Black or White or Thin Black or Thin White) until meet an intersection (Left or Right or Middle or Dead End or Offset).

At the junction, Mikrobotik will act (Turn left or Turn right or Stop) for at least the Minimum Turn Period (*Min Turn Period*) and continue turning until it detects the line and stops.

Mikrobotik will turn by using *Tank Turn Method*.

Step by Step block arrangement:

Step 1 Prepare the blocks arrangement for automatic calibration.



Step 2

Combine block *After Mikrobotik starts* with block *Path Finder Tank* (Line Format- Black, Junction- Right, Action-Turn Left, Left Speed-50, Right Speed-50, Turn Speed-200, RampUp Perc-100, Kp-0.02, Kd-0.2, Sensor Threshold-20, Junction Speed-50, Forward Delay-50, Min Turn Period ms-200).

When Mikrobotik starts

- Robot Prepare
- Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 50 -- Wheel Right Backward Speed (0 ~ 255) 50
- repeat 150
 - Calibrate Sensor
- Stop
- wait 1 seconds

Robot autonomously detects black or white line.

Robot action when meeting junction

Line Format Black

Action Turn Left

Path Finder Tank

Junction Left

Left Speed (0~255) 50

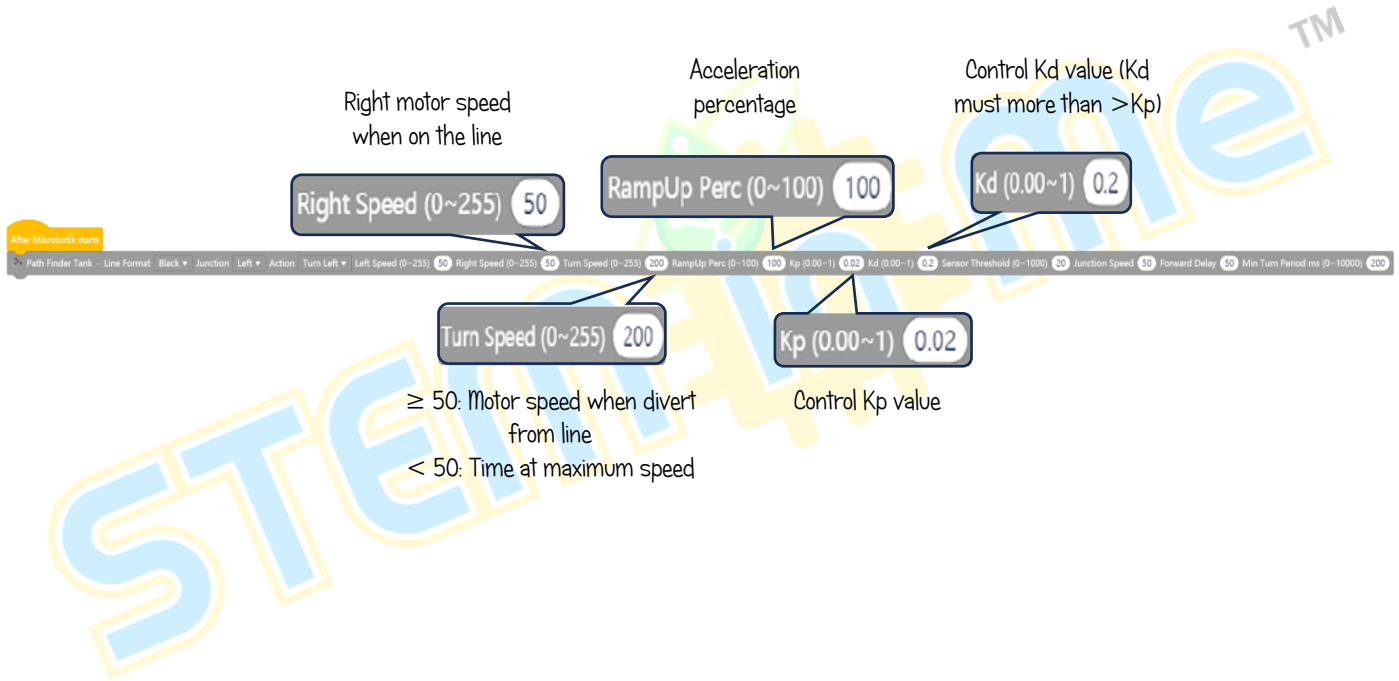
After Mikrobotik starts

Path Finder Tank - Line Format Black - Junction Left - Action Turn Left - Left Speed (0-255) 50 - Right Speed (0-255) 50 - Turn Speed (0-255) 200 - RampUp Perc (0-100) 100 - Kp (0.00-1) 0.02 - Kd (0.00-1) 0.2 - Sensor Threshold (0-1000) 20 - Junction Speed 50 - Forward Delay 50 - Min Turn Period ms (0-10000) 200

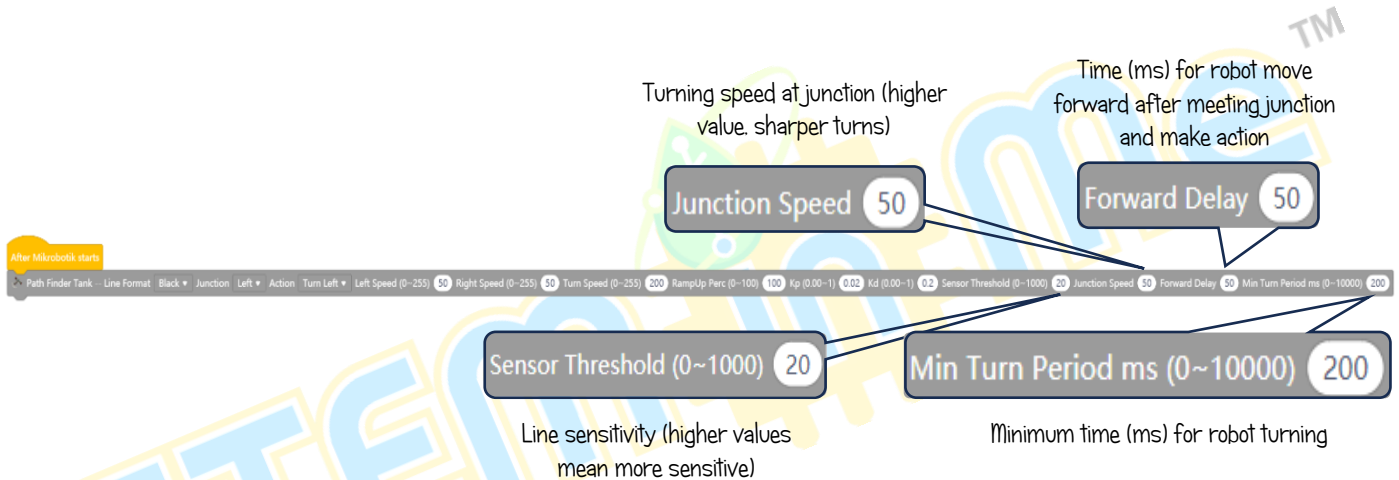
Robot tracking and move forward until meeting junction

Left motor speed when on the line

Continuation



Continuation



Step 3

Combine the "repeat until (false)" block and the "stop" block. Then, integrate these blocks with the block in Step 2.

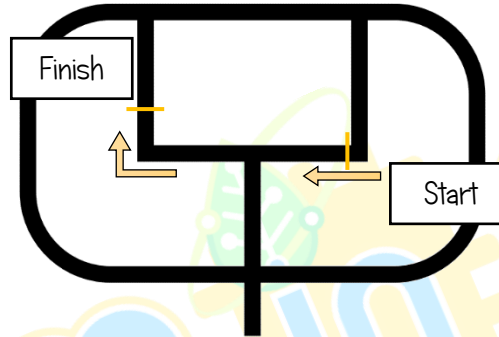


Step 4

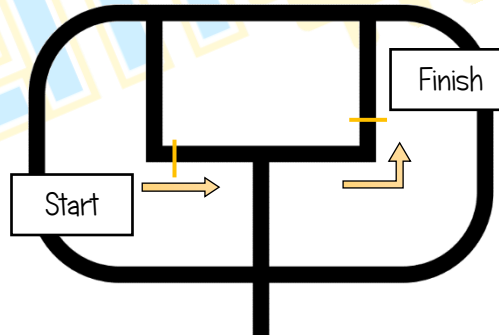
After uploading the code, turn on the Mikrobotik switch and perform the calibration process. After that, Mikrobotik will follow the black line and if the robot finds a left junction, Mikrobotik will move forward and then turn for at least the Minimum Turn Period (*Min Turn Period*) and continue turning until it detects the line and stops.

Challenge!!

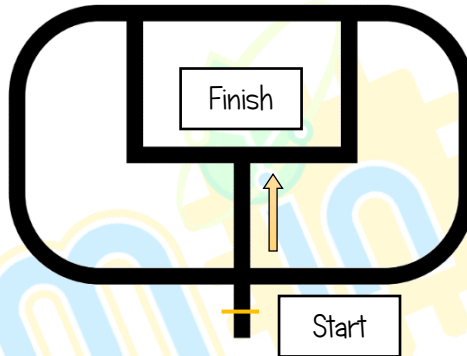
- i) *Path Finder Tank* Right Junction, turn at right junction



- ii) *Path Finder Tank* Left Junction, turn at left junction



iii) *Path Finder Tank* Middle Junction. stop.



Objective 7: Wrong way? Make U-turn

Mikrobotik can make a U-turn on the line it passes through 180 degrees on its axis and turn left or right during the Minimum Turn Duration (Min Turn Period) and continue until it meets the line (Black or White).

Introduction to *Turn At Centre* and its Mechanism

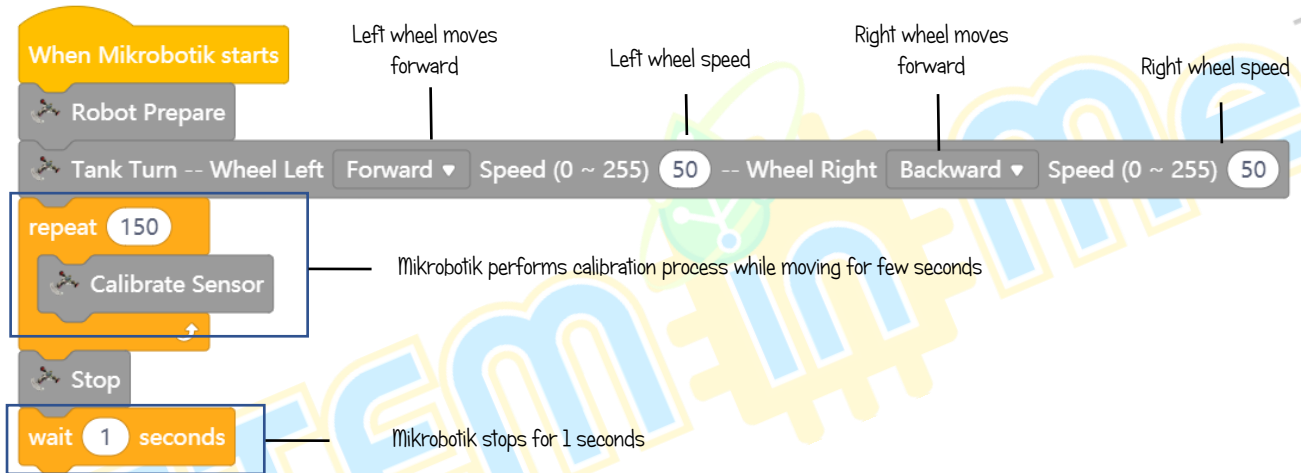
Mikrobotik will make a tank turn in the direction (left or right) for the Minimum Turn Duration (Min Turn Period) until the robot finds the line and finally stops.

This technique is useful for making U-turns.

Step by Step blocks arrangement:

Step 1

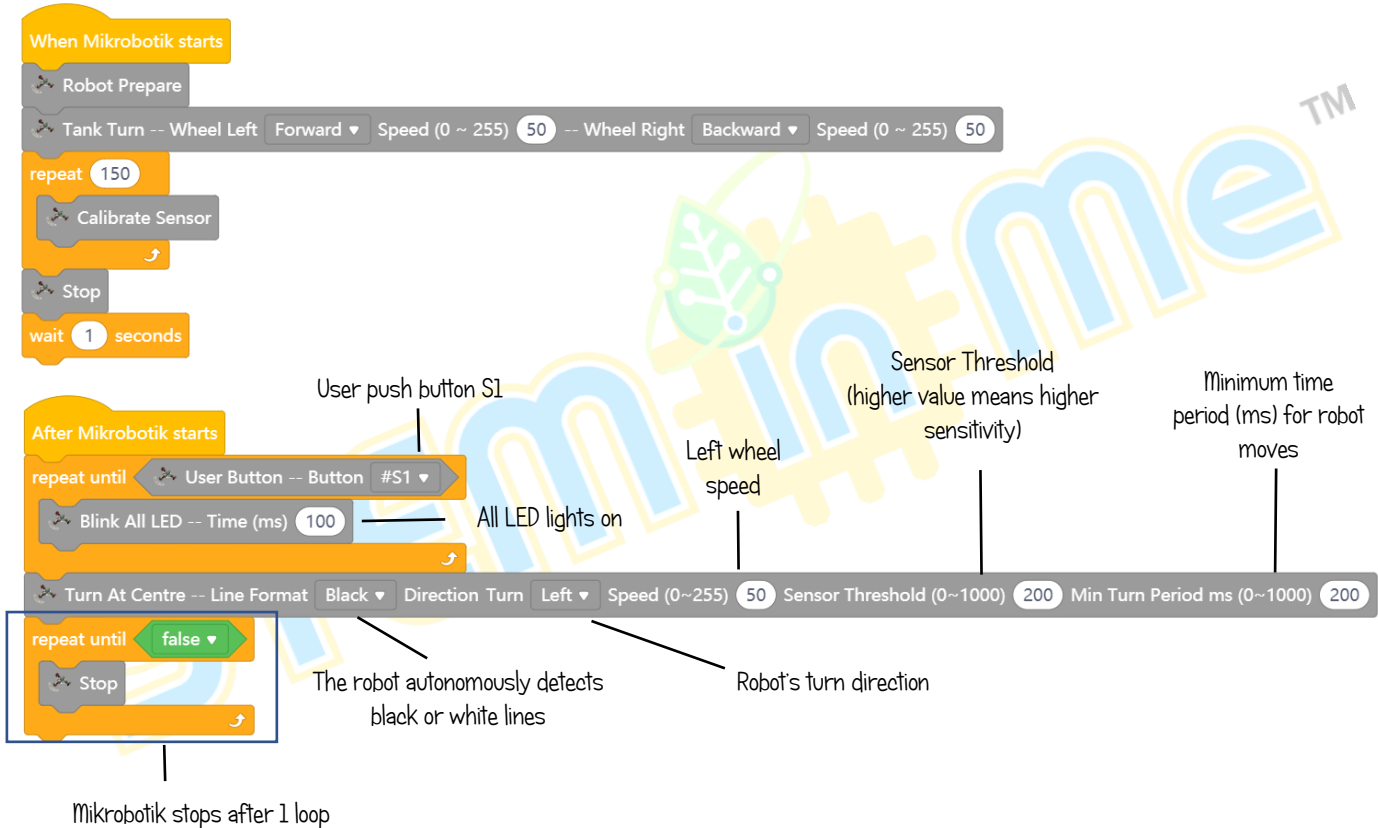
Prepare the arrangement blocks for automatic calibration.



Step 2

Combine block *Find Lind* (Line Format- Black, Direction-Forward, Left Speed-100, Right Speed-100, RampUp Perc-100, Sensor Threshold-20, Forward Delay-0) with block *wait* (1 second). Combine those blocks with blocks in Step 4.

Continuation



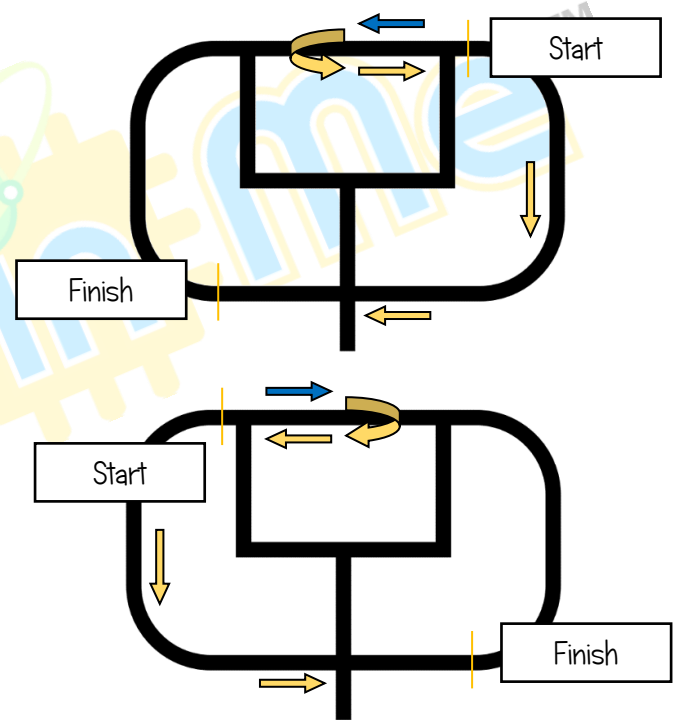
Step 3

After uploading the code, turn on the Mikrobotik switch and perform the calibration process. After that, Mikrobotik will make a U-turn according to the set direction and will stop after detect the black line.

Challenge!!

- i) U-turn at the middle, left direction and move forward.
(blue arrow to yellow arrow)

- ii) U-turn at the middle, left direction and move forward.
(blue arrow to yellow arrow)



Objective 8: Let's Control Mikrobotik

Bluetooth is a short distance wireless technology used for data exchange between fixed and mobile devices in close range. Besides, it can build private network area. Bluetooth allows Mikrobotik to exchange the desired data with other devices directly.

Introduction to Bluetooth and its Mechanism



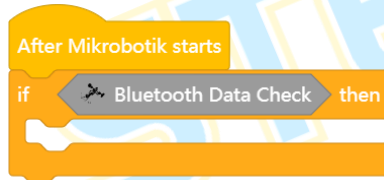
Mikrobotik can be controlled in close range using the Bluetooth approach as it can be found and controlled easily. The Bluetooth module is inserted into the port provided. This Bluetooth module contains 4 pins which are RXD, TXD, GND, and VCC.

Step by Step block arrangement:

Step 1 Insert block *When Mikrobotik Starts* and combine with block *Robot Prepare*

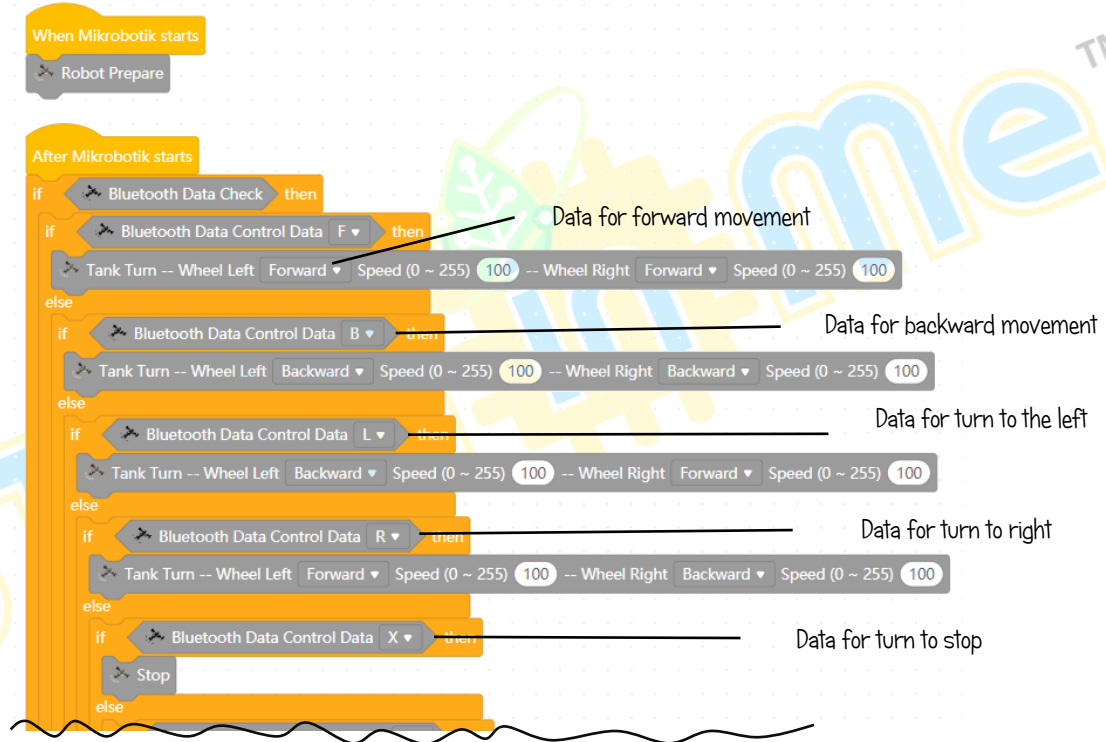


Step 2 Combine block *After Mikrobotik starts* with block *if* after combined with block *Bluetooth Data Check*. Put those blocks under the block in Step 1.



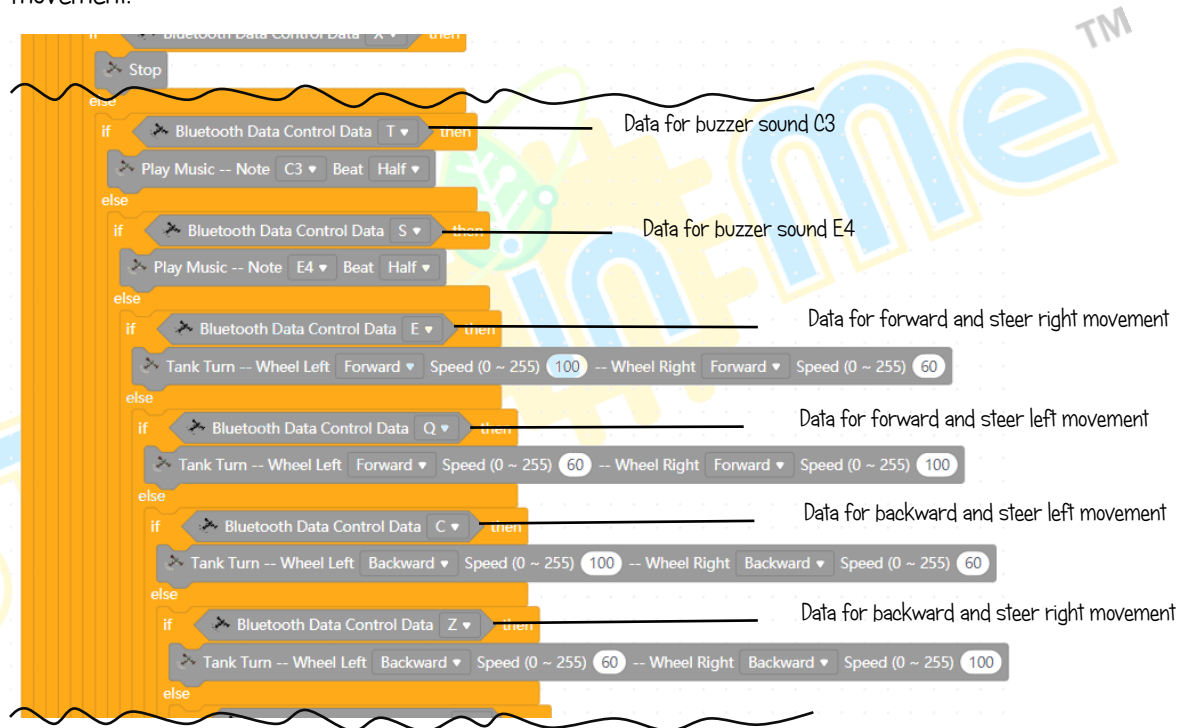
Step 3

Under the block *After Mikrobotik starts*, combine 5 blocks *Bluetooth Data Control Data (F, B, L, R, X)* with 5 blocks *if* and under the block *then* combine with 5 blocks *Tank Turn* to get forward, backward, turn to left, turn to right and stop movement.



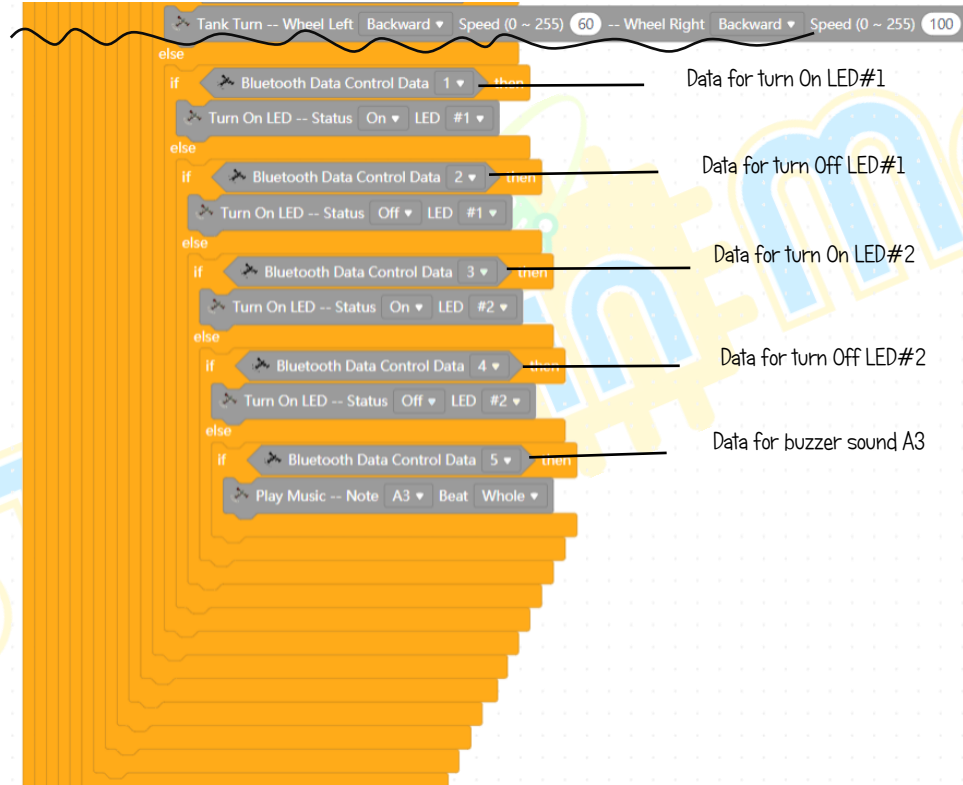
Step 4

For the next block, combine 6 blocks *Bluetooth Data Control Data (T, S, E, Q, C, Z)* with 6 blocks *if* and under the block *then* combine with 2 blocks *Play Music (Note-C3, Beat Half)*5 and 4 blocks *Tank Turn* to get forward and steer right, forward and steer left, backward and steer left, backward and steer right movement.



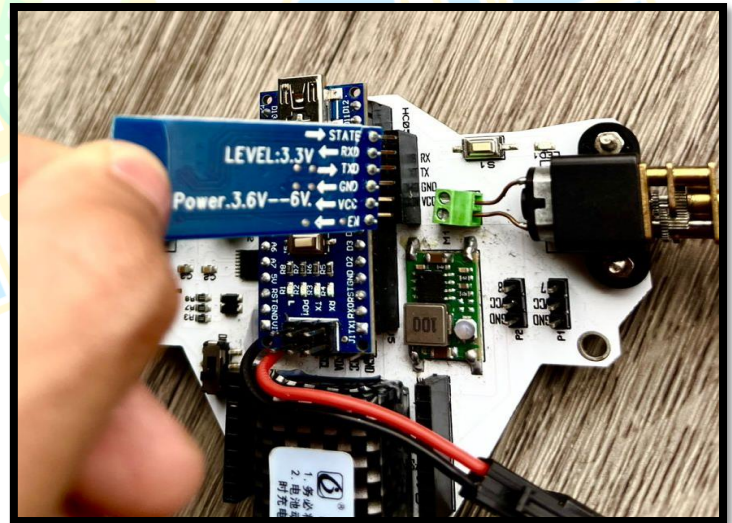
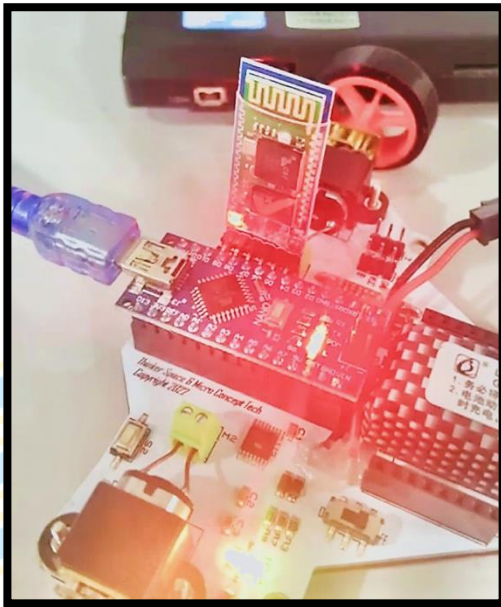
Step 5

For the next block, combine 5 blocks *Bluetooth Data Control Data* (1, 2, 3, 4, 5) with 5 blocks *if* and under the block *then* combine with 2 blocks *Turn On LED* (#1 On, #1 Off, #2 On, #2 Off) and 1 block *Play Music* (Note-A3, Beat Whole).



Step 6

After uploading the code, pair the Bluetooth module on the Mikrobotik and match it with the device. Mikrobotik is ready to be controlled by the device. Make sure all Bluetooth pins are connected to the Bluetooth port (RXD-RX, TXD-TX, GND-GND, VCC-VCC)

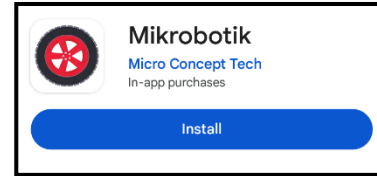




Mikrobotik Mobile Apps

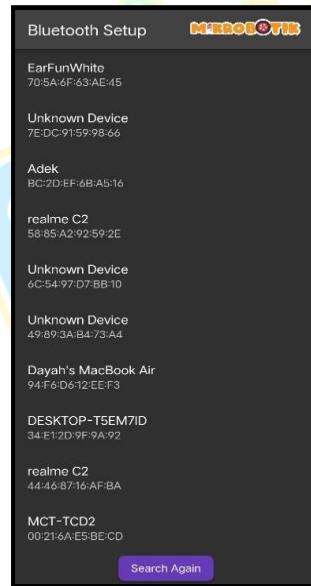
Step 1

Download and install Mikrobotik Apps at Google Play Store.



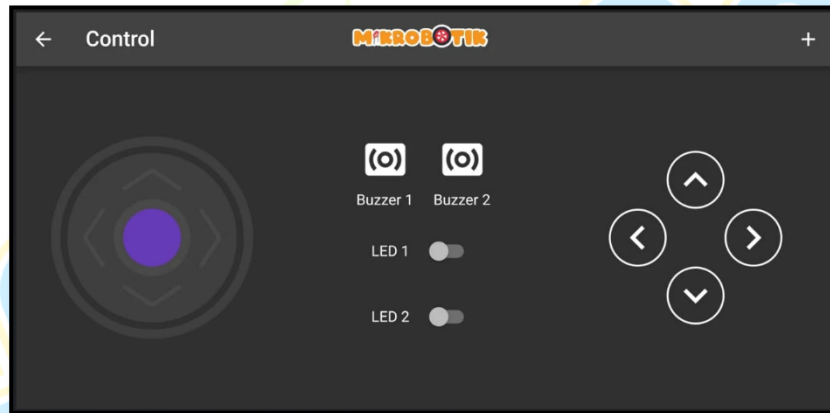
Step 2

Open application. At "Bluetooth Setup", choose based on the serial number on the Bluetooth Module



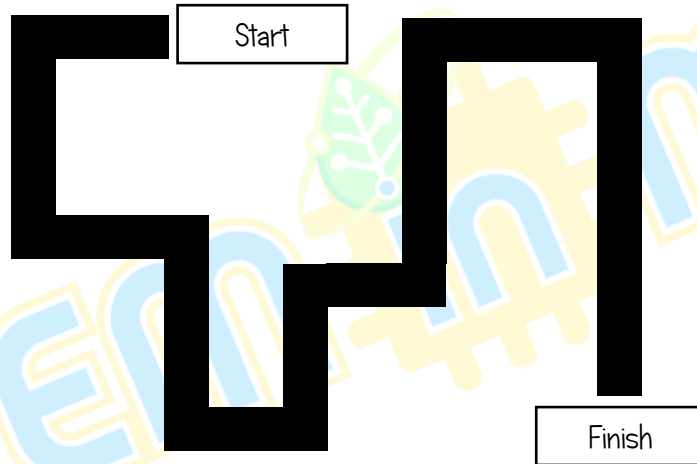
Step 3

Click "Control". Now Mikrobotik can be controlled in free movement.



Challenge!!

In this challenge, you need to make sure that Mikrobotik moves along the path provided by using a device that has been paired with a Bluetooth module on Mikrobotik.



Objective 9: We Need Area Patrol!

Sometimes a robot needs to use more than one block to complete a task such as "area patrol robot". To patrol an area, the robot needs to move along a line at varying speeds and for a certain distance or time. In addition, while following the line, the robot has to make a turn in the opposite direction.

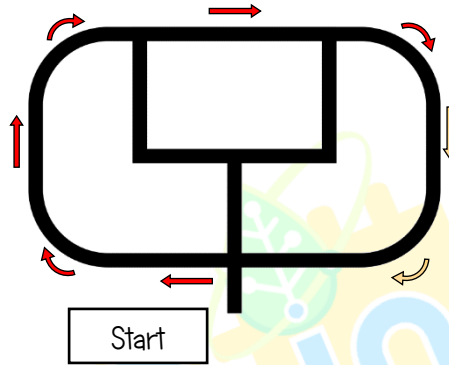
Introduction to Movement and Its Mechanisms

The technique used is to combine several *Line Tracer Time* blocks and *Turn at Centre* blocks.

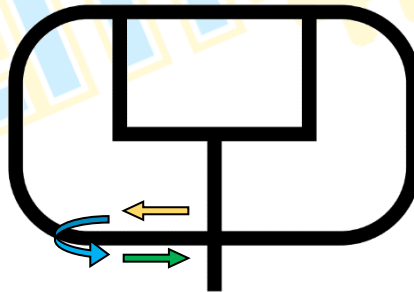
The robot moves autonomously following the line using the *Line Tracer Time* block with high speed and then with low speed for a certain time. After finishing moving, the robot makes a turn in the opposite direction using the *Turn at Centre* block. Finally, the robot moves again autonomously by using the *Line Tracer Time* block with high speed.

Here is a sketch of the movement of the "area patrol robot" with a set time and speed and make a turn to complete the task.

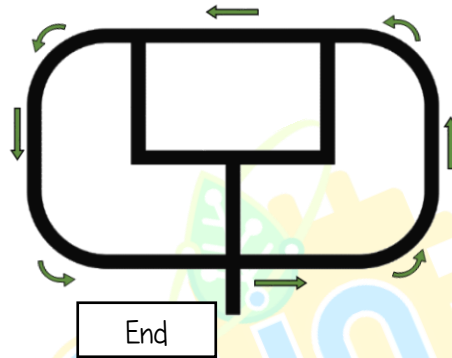
- i) The robot moves autonomously using *Line Tracer Time* with high speed for 3 seconds (red arrow) and then with low speed for 3 seconds (yellow arrow).



- ii) The robot makes a turn in the middle, towards the left. (yellow arrow direction to green arrow) using *Turn At Centre*.

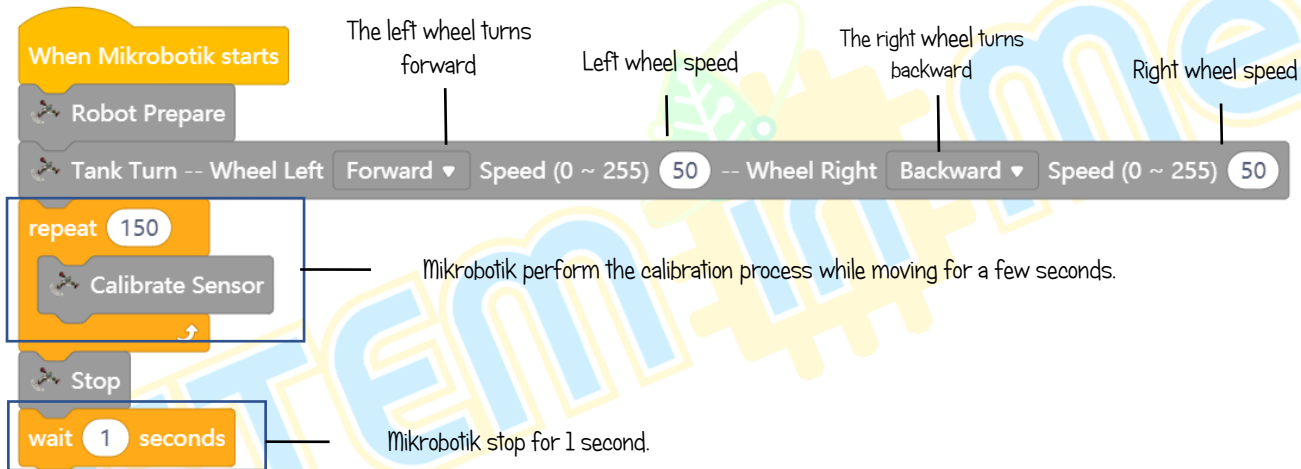


- i) The robot moves autonomously to the end point using *Line Tracer Time* at high speed for 4 seconds.



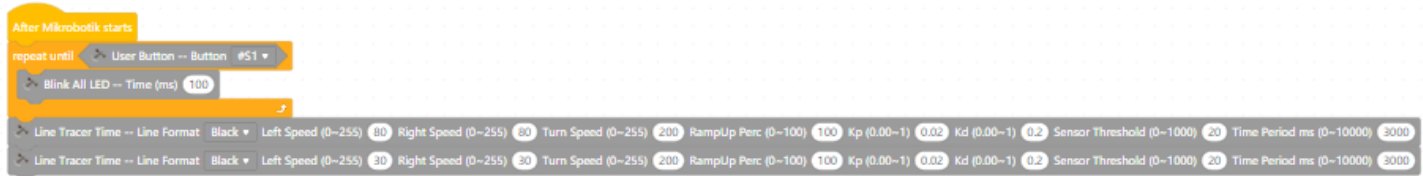
Step by Step Block Arrangement

Step 1 Set up an automatic calibration block arrangement.



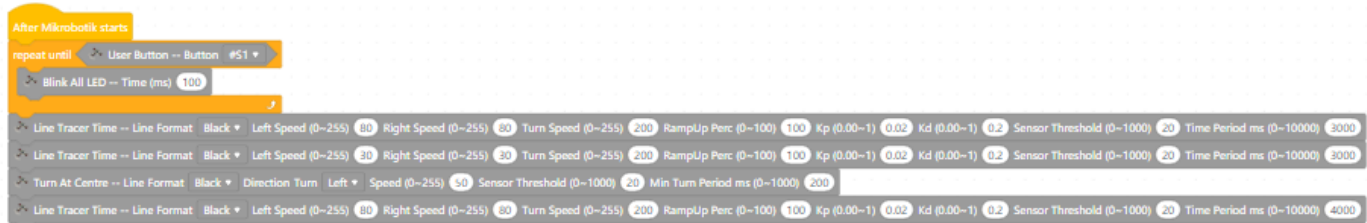
Step 2

Combine the *After Mikrobotik starts* block with the *repeat until* block. After that, add a *Line Tracer Time* block by setting the speed to 80 within 3 seconds. Then, add another *Line Tracer Time* block by setting the speed to 30 within 3 seconds.



Step 3

Add a *Turn at Centre* block and set it to make a left turn. Add another *Line Tracer Time* by setting the speed to 80 for 3 seconds. Then, add another *Line Tracer Time* block by setting the speed to 80 in 4 seconds.





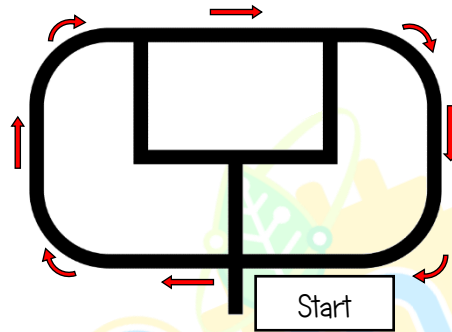
Step 4

After uploading the code, turn on the Mikrobotik switch and perform the calibration process. After that, Mikrobotik will make all movements according to the set time and will stop after the set time.

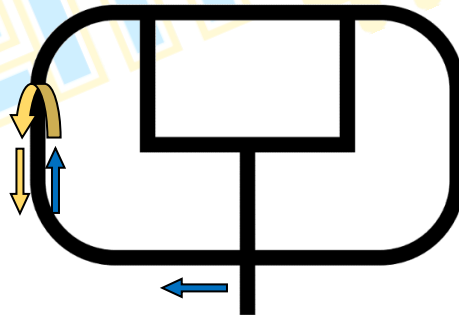


Challenge!!

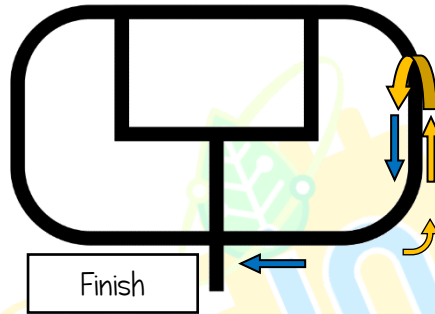
- i. The robot moves autonomously for one round.



- ii. Take a U-turn at the middle.



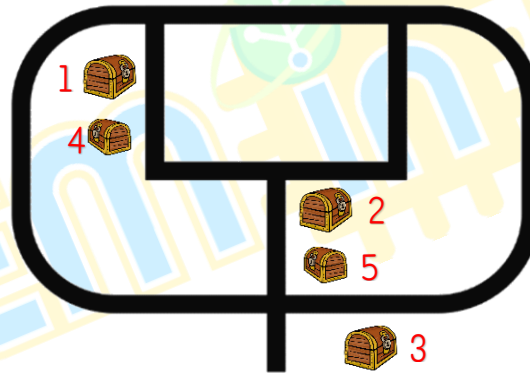
iii. Take another U-turn and finish.



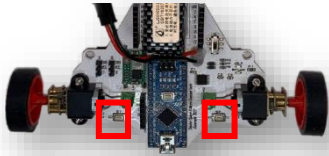
STEMtime™

Objective 10: Let's Find Hidden Treasures.

Sometimes the robot needs to use more than one block to complete a task such as "robot looking for hidden treasure". In order to get all five hidden treasures, the robot needs to go through many intersections, among which are left junctions, right junctions and 3-way junctions. Sometimes the robot needs to turn at different speeds to enter the junction.



Introduction to Push Button



A push button is a type of switch that functions to control a machine directly through the touch of a hand or finger from the user or the surface of a component. MikroBotik has push buttons S1 and S2. The analogue reading value will be less than 400 when S1 is pressed while the analogue reading will be less than 500 when S2 is pressed.

Introduction to Movement and Its Mechanisms.

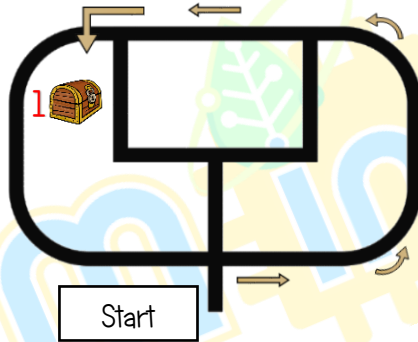
The technique used is to combine several *Path Finder* and *Path Finder Tank* blocks.

By using the *Path Finder* or *Path Finder Tank* block, the robot will move autonomously following the black or white line until it finds an intersection and then the robot will turn towards the specified intersection.

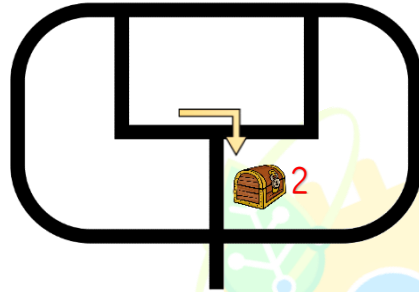
To get the first hidden treasure, the robot moves using the *Path Finder* until it finds a left intersection and turns to the left. Then, the robot continues moving using the *Path Finder* until it finds a right intersection and turns to the right for the second hidden treasure. Next, the robot continues moving using the *Path Finder* until it finds a 3-way junction and turns left for the third hidden treasure. After that, to get the fourth haunted treasure the robot needs to use the *Path Finder Tank* until it meets the left intersection and turn left and finally to get the last haunted treasure the robot needs to use the *Path Finder Tank* until it finds the right intersection and turns right.

Here is a sketch of the movement of the robot searching for hidden treasures by going through different intersections to complete the task.

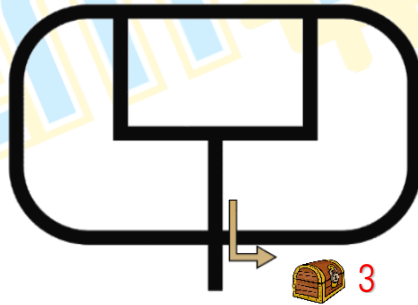
- i) The robot moves from the starting point to find the left junction and turns using the *Path Finder* to the left to pick up the first hidden treasure.



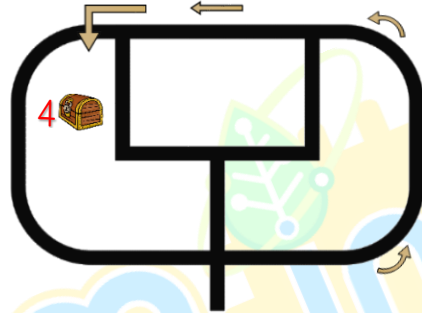
- ii) The robot moves to find the right junction and turns using the *Path Finder* round to the right to pick up the second hidden treasure.



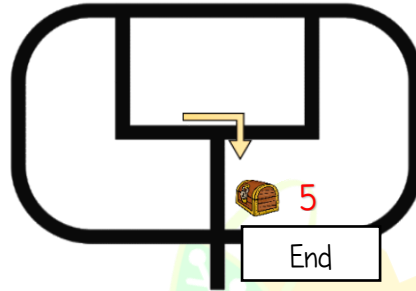
- iii) The robot moves to find the 3-way intersection and turns using the *Path Finder* to the left to pick up the third hidden treasure.



- iv) The robot moves again to find the left junction and turns using the *Path Finder Tank* to the left to pick up the fourth hidden treasure.



- v) The robot continues to move looking for the right junction and turns using the *Path Finder Tank* to the right to take the last hidden treasure and then stops.

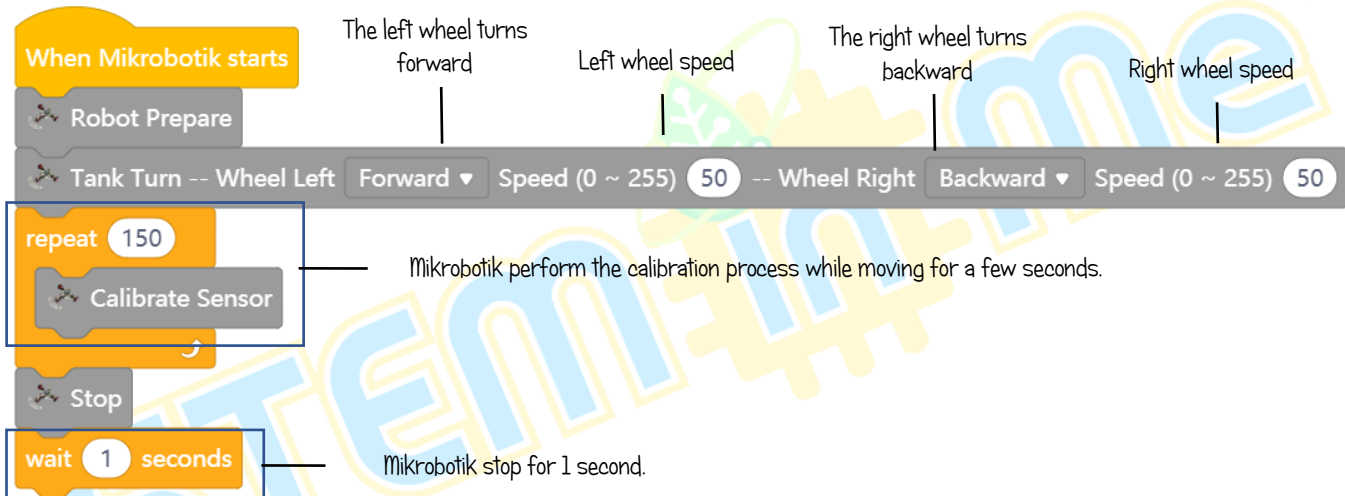


STEM in me™

Step by Step Block Arrangement:

Step 1

Set up an automatic calibration block arrangement.



Step 2

Combine the *After Mikrobotik starts* block with the *repeat until* block. After that, add the *Path Finder* block and set it (*Junction* – *Left*; *Action* – *Turn Left*; *Speed* – *60*; *Turn Speed* – *200*; *Junction Speed* – *200*; *Forward Delay* – *400* and *Turn Period* – *400*”).



Step 3

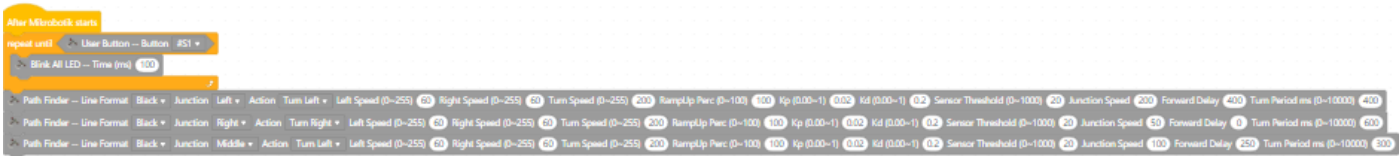
Add a new *Path Finder* block and set the value to (*Junction* – *Right*; *Action* – *Turn Right*; *Speed* – *60*; *Turn Speed* – *200*; *Junction Speed* – *50*; *Forward Delay* – *0* and *Turn Period* – *600*”).



Step 4

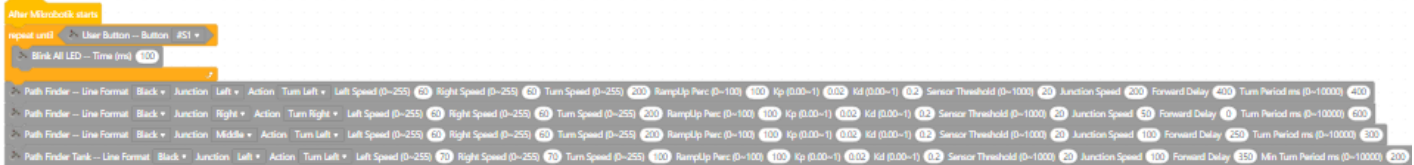
Add a new *Path Finder* block and set the value to (*Junction* – *“Middle”*; *Action* – *“Turn Left”*; *Speed* – *“60”*; *Turn Speed* – *“200”*; *Junction Speed* – *“100”*; *Forward Delay* – *“250”* and *Turn Period* – *“300”*).

TMI



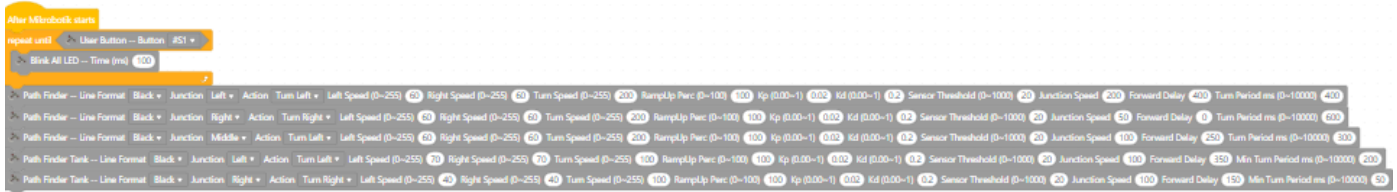
Step 5

Add a *Path Finder Tank* block and set the value to (*Junction* – *“Left”*; *Action* – *“Turn Left”*; *Speed* – *“70”*; *Turn Speed* – *“100”*; *Junction Speed* – *“100”*; *Forward Delay* – *“350”* and *Min Turn Period* – *“200”*).



Step 6

Add a new *Path Finder Tank* block and set the value to (*Junction – “Right”, Action – Turn Right”, Speed – “40”, Turn Speed – “100”, Junction Speed – “100”, Forward Delay – “150” and Min Turn Period – “50”*).

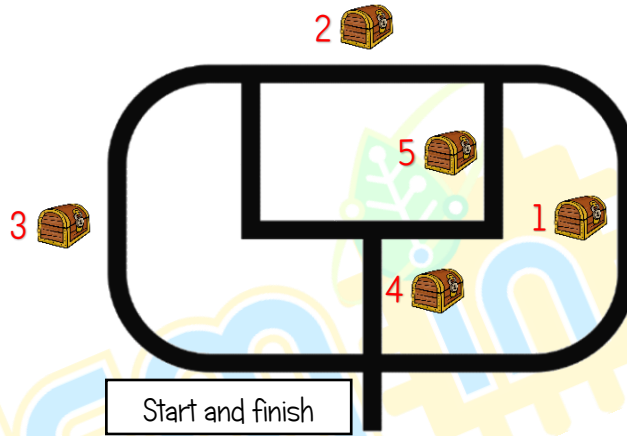


Step 7

After uploading the code, turn on the Mikrobotik switch and perform the calibration process. After that, Mikrobotik will move to find the designated intersection and make a circle until it finds the hidden treasure.

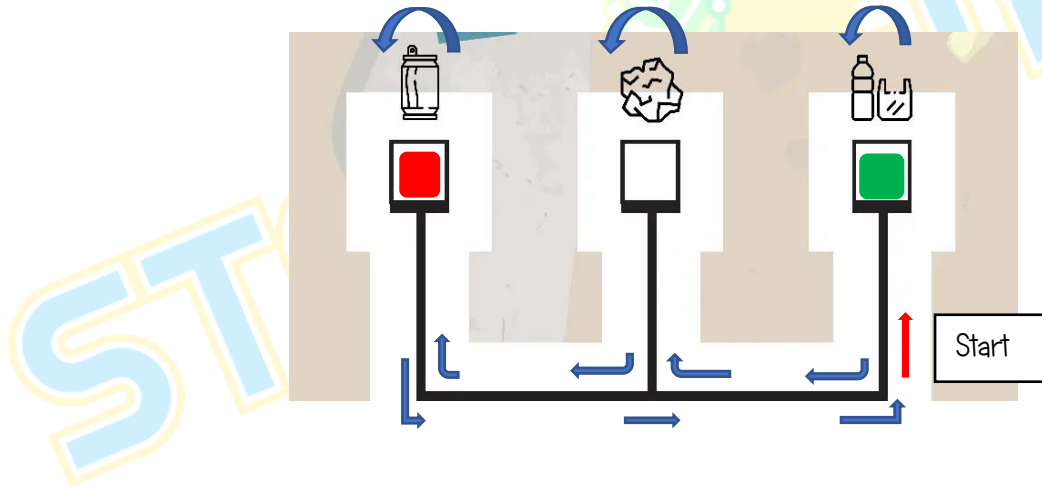
Challenge!!

Get all five hidden treasures and stop at finish.

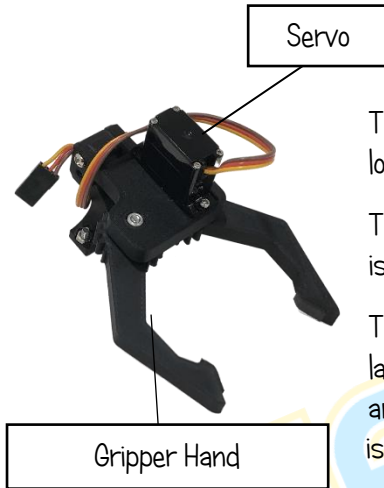


Objective II: Recycling Material Separation.

This objective focuses on the effort of material separation for recycling using robots. To aid in recycling efforts, we can employ robots to move recyclable objects from one location to a specific designated area. For this purpose, the robot utilizes a gripper tool to grasp the recyclable object, then the robot moves from one location to the designated area, and finally releases the object.



Introduction to Single Gripper and Its Mechanism



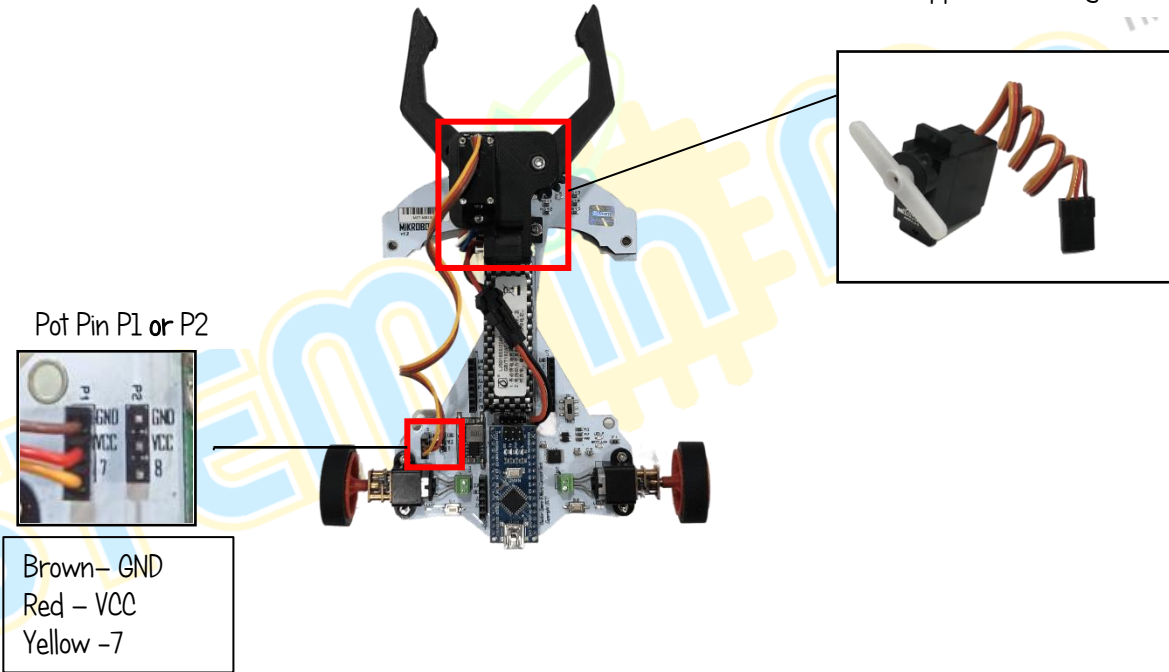
The single gripper is used in MikroBOTIK to grasp objects and move (drag) them to the desired location. The single gripper consists of one servo motor and one mechanical gripper hand.

The single gripper is mounted on MikroBOTIK using screws and nuts, while the servo motor wire is connected to the existing servo pot pins labelled "P1" or "P2" on MikroBOTIK.

The width of the gripper hand's opening can be adjusted by setting the servo motor's angle. A larger servo angle results in a smaller gripper hand opening. Typically, when the servo motor angle is set to 0 degrees, the gripper hand is in its widest position. When the servo motor angle is set to 180 degrees, the gripper hand is in its smallest or fully closed position.

Installation of the Single Gripper on the robot

Servo Motor
(Gripper Width Angle)

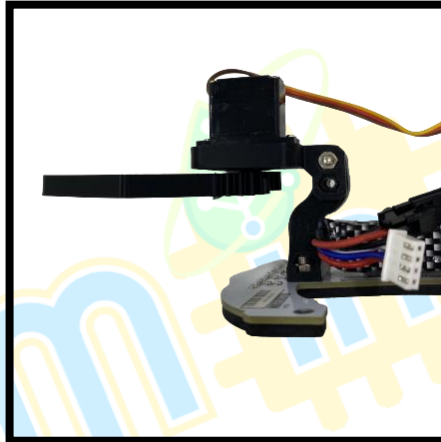


Here is the sequence of steps for installing a *Single Gripper* on Mikrobotik:

- 1- Loosen the screws and nuts on the *Single Grippers*. There are two screw holes for the *Single Grippers* under the Mikrobotik robot. Insert the screws and nuts, then tighten them.



2- Ensure the position of the *Single Gripper* after installation is like this.



Arrange Strategy and Movement Techniques.

The technique used is by combining several blocks: *Path Finder*, *Gripper Servo Port*, and *Turn at Centre*.

Using the *Path Finder* block, the robot will autonomously move following black or white lines until it encounters the first object (green).

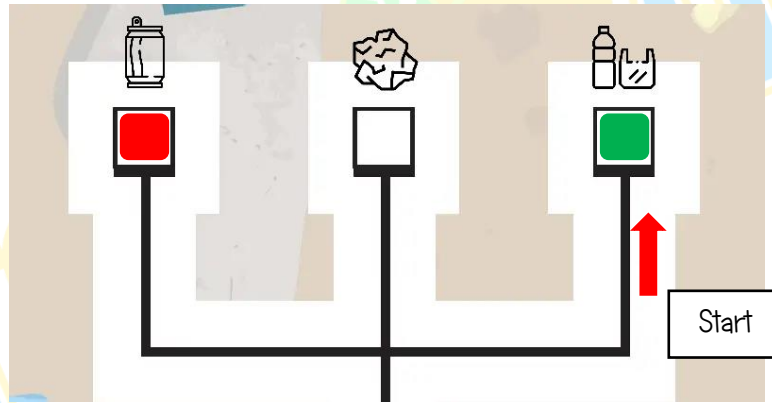
To obtain the first object (green), the robot will grasp it using the *Gripper Servo Port* block. Then, the robot will rotate using the *Turn at Centre* block and continue its movement using several *Path Finder* blocks until it reaches a left-right junction in a designated area. It will stop to place the object into the designated area using the *Gripper Servo Port*.

Next, the robot will rotate using the *Turn at Centre* block and continue its movement using several *Path Finder* blocks until it reaches a left-right junction in the designated area. It will stop to pick up the second object (red) using the *Gripper Servo Port*.

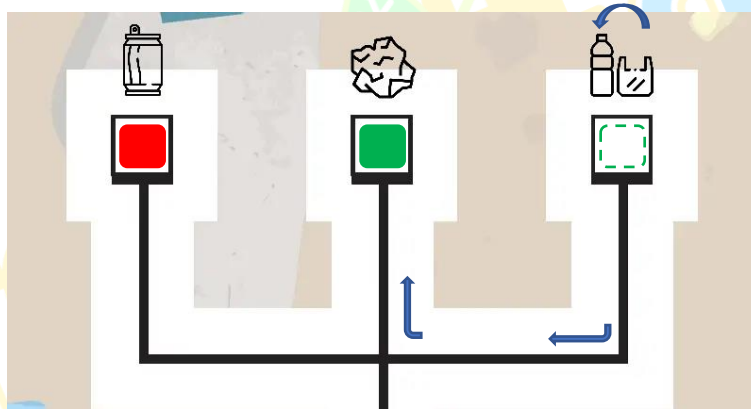
Afterward, the robot will rotate using the *Turn at Centre* block and move using several *Path Finder* blocks until it reaches a left-right junction in the designated area. It will then place the second object into the designated area using the *Gripper Servo Port*.

Here is a provided sketch of the robot's movements for individually picking up objects and arranging them in designated spaces.

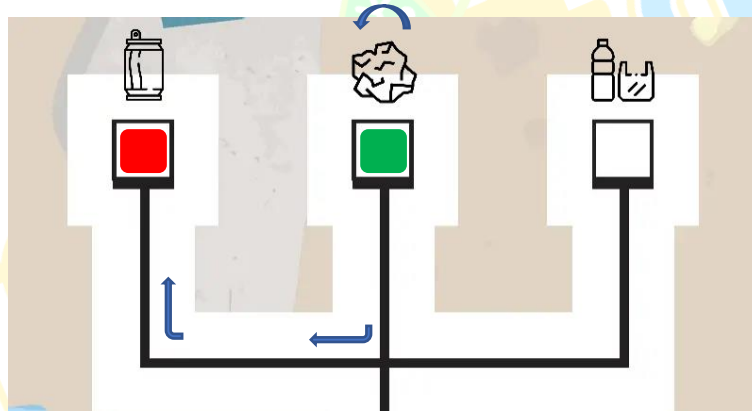
- i) Using the Path Finder block, the robot moves from the starting point until it reaches a left-right junction and stops. Then, the robot grasps the first object (green) using the Gripper Servo Port block (Gripper) with a large angle setting (small gripper opening).



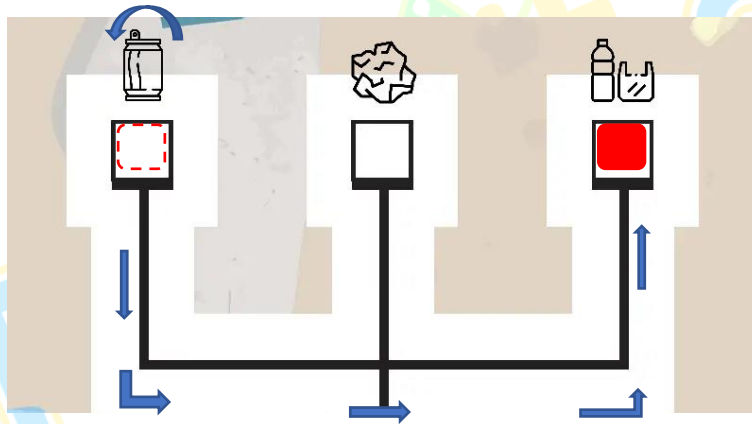
- ii) The robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a right junction and turns to the right. Subsequently, the robot moves using the *Path Finder* block until it reaches a left-right junction and turns to the right. After that, the robot moves using the *Path Finder* block until it reaches a left-right junction and stops. Finally, it releases the second object (red) into a designated space using the *Gripper Servo Port* block with a small angle setting (large gripper opening).



- iii) After releasing the first object, the robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a left-right junction and turns to the right. After that, the robot moves using the *Path Finder* block until it reaches a right junction and turns to the right, then stops. Next, the robot grasps the second object (red) using the *Gripper Servo Port* block (Gripper) with a large angle setting (small gripper opening).



- iv) The robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a left junction and turns to the left. Subsequently, the robot moves using the *Path Finder* block until it reaches another left junction and turns left again. After that, the robot moves using the *Path Finder* block until it reaches a left-right junction and stops. Finally, it releases the second object (red) into a designated space using the *Gripper Servo Port* block with a small angle setting (large gripper opening).



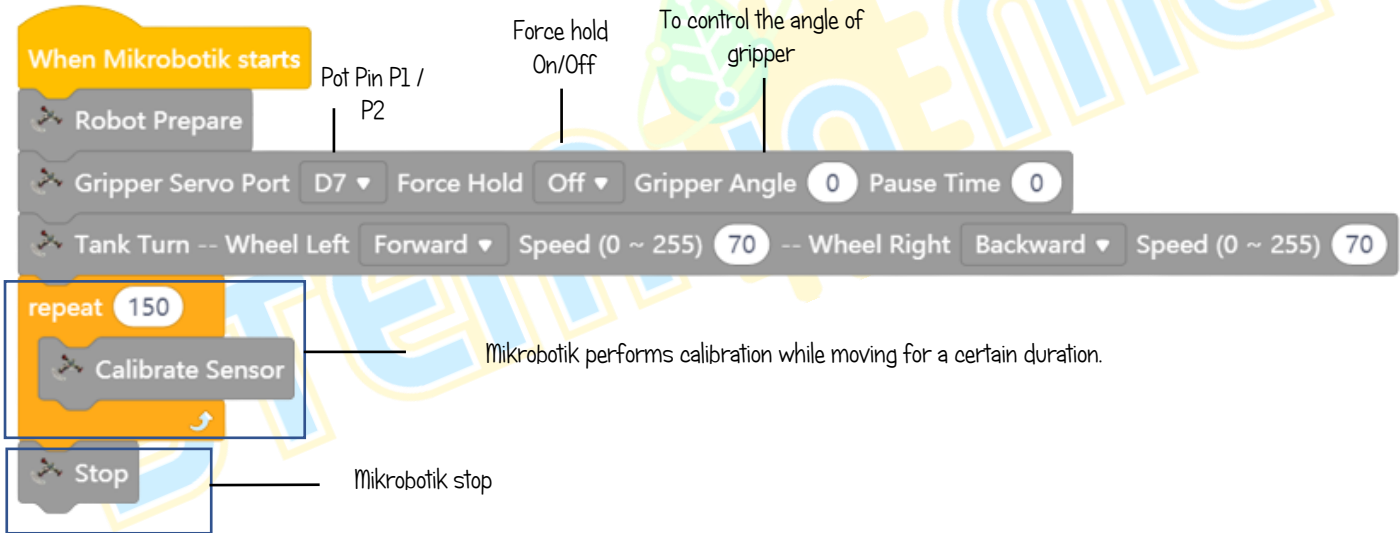
Step by Step block arrangement:

Step 1

Connect the wire of the *single gripper* servo motor to the pot labelled P1 or P2.

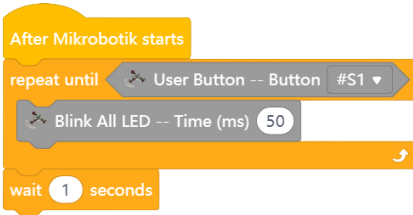
Step 2

Prepare the blocks arrangement for automatic calibration.



Step 3

Combine the *After Mikrobotik starts* block with the *repeat until* block and add a *wait 1 second* block.



Step 4

Add a *Path Finder* block and configure it with the following settings (*Juction* – *“Middle”*; *Action* – *“Stop”*; *Speed* – *“50”*; *Turn Speed* – *“50”*; *Junction Speed* – *“50”*; *Forward Delay* – *“0”* dan *Turn Period* – *“0”*). Then, add a *Gripper* block and configure it with these settings (*Gripper Servo Port* – *“D7”*; *Force hold* – *“On”*; *Gripper Angle* – *“133”*; *Pause* – *“0”*). Finally, include a *Turn At Centre* block with these settings (*Direction Turn* – *“Left”*; *Speed* – *“50”*; *MinTurn Period* – *“600”*).



Step 5

Add a *Path Finder* block and configure it with the following settings (*Junction* – [*Right*: *Middle*: *Middle*]; *Action* – [*Turn Right*: *Turn Right*: *Stop*]; *Speed* – [*100*: *100*: *50*]; *Turn Speed* – [*100*: *100*: *50*]; *Junction Speed* – [*100*: *100*: *50*]; *Forward Delay* – [*50*: *50*: *0*] dan *Turn Period* – [*600*: *600*: *0*]). Then, add a *Gripper* block and configure it with these settings (*Gripper Servo Port* – *D7*, *Force hold* – *Off*; *Gripper Angle* – *20*; *Pause* – *500*). Finally, include a *Turn At Centre* block with these settings (*Direction Turn* – *Left*; *Speed* – *50*; *Min Turn Period* – *300*).

The screenshot shows a Scratch script with the following blocks and their configurations:

- repeat until**: User Button - Button #51
- blink all LED**: Time (ms) 50
- wait**: 1 seconds
- Path Finder**: Line Format: Black, Junction: Middle, Action: Stop, Left Speed (0-255): 50, Right Speed (0-255): 50, Turn Speed (0-255): 50, RampUp Perc (0-100): 100, Kp (0.00-1): 0.02, Kd (0.00-1): 0.2, Sensor Threshold (0-1000): 20, Junction Speed: 50, Forward Delay: 0, Turn Period ms (0-10000): 0
- Gripper Servo Port**: D7, Force Hold: On, Gripper Angle: 133, Pause Time: 0
- Turn At Centre**: Line Format: Black, Direction Turn: Left, Speed (0-255): 50, Sensor Threshold (0-1000): 20, Min Turn Period ms (0-1000): 600
- Path Finder**: Line Format: Black, Junction: Right, Action: Turn Right, Left Speed (0-255): 100, Right Speed (0-255): 100, Turn Speed (0-255): 100, RampUp Perc (0-100): 100, Kp (0.00-1): 0.02, Kd (0.00-1): 0.2, Sensor Threshold (0-1000): 20, Junction Speed: 100, Forward Delay: 50, Turn Period ms (0-10000): 600
- Path Finder**: Line Format: Black, Junction: Middle, Action: Turn Right, Left Speed (0-255): 100, Right Speed (0-255): 100, Turn Speed (0-255): 100, RampUp Perc (0-100): 100, Kp (0.00-1): 0.02, Kd (0.00-1): 0.2, Sensor Threshold (0-1000): 20, Junction Speed: 100, Forward Delay: 50, Turn Period ms (0-10000): 600
- Path Finder**: Line Format: Black, Junction: Middle, Action: Stop, Left Speed (0-255): 50, Right Speed (0-255): 50, Turn Speed (0-255): 50, RampUp Perc (0-100): 100, Kp (0.00-1): 0.02, Kd (0.00-1): 0.2, Sensor Threshold (0-1000): 20, Junction Speed: 50, Forward Delay: 0, Turn Period ms (0-10000): 0
- Gripper Servo Port**: D7, Force Hold: Off, Gripper Angle: 20, Pause Time: 500
- Turn At Centre**: Line Format: Black, Direction Turn: Left, Speed (0-255): 50, Sensor Threshold (0-1000): 20, Min Turn Period ms (0-1000): 300

Step 6

Add 3 *Path Finder* blocks and configure them in the following sequence (*Junction* – [*“Middle”*; *“Right”*; *“Middle”*]; *Action* – [*“Turn Right”*; *Turn Right”*; *“Stop”*]; *Speed* – [*“100”*; *“100”*; *“50”*]; *Turn Speed* – [*“100”*; *“100”*; *“50”*]; *Junction Speed* – [*“100”*; *“100”*; *“50”*]; *Forward Delay* – [*“50”*; *“50”*; *“0”*] dan *Turn Period* – [*“600”*; *“600”*; *“0”*]). Add a *Gripper* block and configure it with the following settings (*Gripper Servo Port* – *“D7”*; *Force hold* – *“On”*; *Gripper Angle* – *“133”*; *Pause* – *“0”*). Then, add a *Turn At Centre* block with the following settings (*Direction Turn* – *“Left”*; *Speed* – *“50”*; *Min Turn Period* – *“300”*).

Step 7

Add 4 *Path Finder* blocks in the following sequence (*Junction* – [*Left*: *Middle*: *Left*: *Middle*]. *Action* – [*Turn Left*: *Stop*: *Turn Left*: *Stop*]. *Speed* – [*100*: *100*: *50*]. *Turn Speed* – [*100*: *100*: *50*]. *Junction Speed* – [*100*: *100*: *100*: *50*]. *Forward Delay* – [*50*: *50*: *50*: *0*] dan *Turn Period* – [*550*: *0*: *600*: *0*]). Next, add a *Gripper* block and configure it with the following settings (*Gripper Servo Port* – *D7*, *Force hold* – *Off*, *Gripper Angle* – *20*, *Pause* – *550*).

The screenshot shows a sequence of programming blocks in a MikroBotik environment. The blocks are as follows:

- repeat until**: User Button - Button #51
- Blink All LED**: Time (ms) 50
- wait**: 1 seconds
- Path Finder**: Line Format Black, Junction Middle, Action Stop, Left Speed (0-255) 50, Right Speed (0-255) 50, Turn Speed (0-255) 50, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 50, Forward Delay 0, Turn Period ms (0-10000) 0
- Gripper Servo Port**: D7, Force Hold On, Gripper Angle 133, Pause Time 0
- Turn At Centre**: Line Format Black, Direction Turn Left, Speed (0-255) 50, Sensor Threshold (0-1000) 20, Min Turn Period ms (0-1000) 600
- Path Finder**: Line Format Black, Junction Right, Action Turn Right, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 600
- Path Finder**: Line Format Black, Junction Middle, Action Turn Right, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 600
- Path Finder**: Line Format Black, Junction Middle, Action Stop, Left Speed (0-255) 50, Right Speed (0-255) 50, Turn Speed (0-255) 50, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 50, Forward Delay 0, Turn Period ms (0-10000) 0
- Gripper Servo Port**: D7, Force Hold Off, Gripper Angle 20, Pause Time 500
- Turn At Centre**: Line Format Black, Direction Turn Left, Speed (0-255) 50, Sensor Threshold (0-1000) 20, Min Turn Period ms (0-1000) 300
- Path Finder**: Line Format Black, Junction Middle, Action Turn Right, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 600
- Path Finder**: Line Format Black, Junction Right, Action Turn Right, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 600
- Path Finder**: Line Format Black, Junction Middle, Action Stop, Left Speed (0-255) 50, Right Speed (0-255) 50, Turn Speed (0-255) 50, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 50, Forward Delay 0, Turn Period ms (0-10000) 0
- Gripper Servo Port**: D7, Force Hold On, Gripper Angle 133, Pause Time 0
- Turn At Centre**: Line Format Black, Direction Turn Left, Speed (0-255) 50, Sensor Threshold (0-1000) 20, Min Turn Period ms (0-1000) 300
- Path Finder**: Line Format Black, Junction Left, Action Turn Left, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 550
- Path Finder**: Line Format Black, Junction Middle, Action Stop, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 0
- Path Finder**: Line Format Black, Junction Left, Action Turn Left, Left Speed (0-255) 100, Right Speed (0-255) 100, Turn Speed (0-255) 100, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 100, Forward Delay 50, Turn Period ms (0-10000) 600
- Path Finder**: Line Format Black, Junction Middle, Action Stop, Left Speed (0-255) 50, Right Speed (0-255) 50, Turn Speed (0-255) 50, RampUp Perc (0-100) 100, Kp (0.00-1) 0.02, Kd (0.00-1) 0.2, Sensor Threshold (0-1000) 20, Junction Speed 50, Forward Delay 0, Turn Period ms (0-10000) 0
- Gripper Servo Port**: D7, Force Hold Off, Gripper Angle 20, Pause Time 500

Extra Information

Force Hold (Daya Cengkaman)

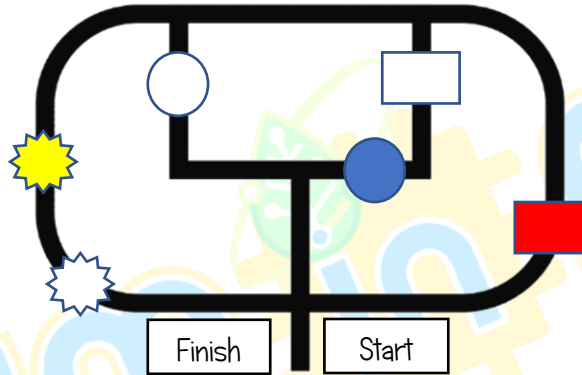
- When the force hold is turned on (ON), the gripper will maintain its gripper width angle and prevent it from being changed.
- When the force hold is turned off (OFF), the gripper will no longer maintain its gripper width angle, and it can be adjusted freely.

Pause Time (Tempoh Jeda)

- The pause time function can only be used when the force hold is turned off (OFF).
- The pause time function is used to activate the force hold for a specified duration. Once the pause time expires, the force hold will be turned off.

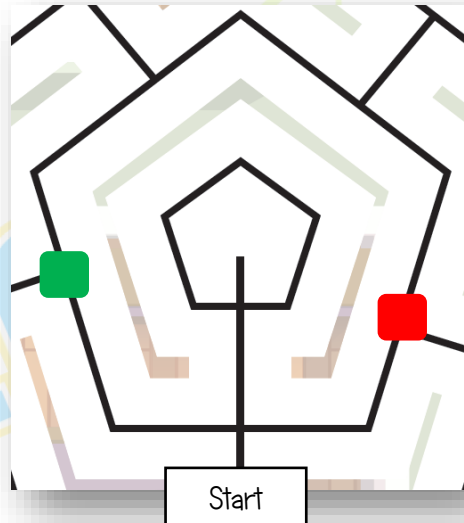
Challenge!!

Move recyclable objects from one location to a specific designated area.

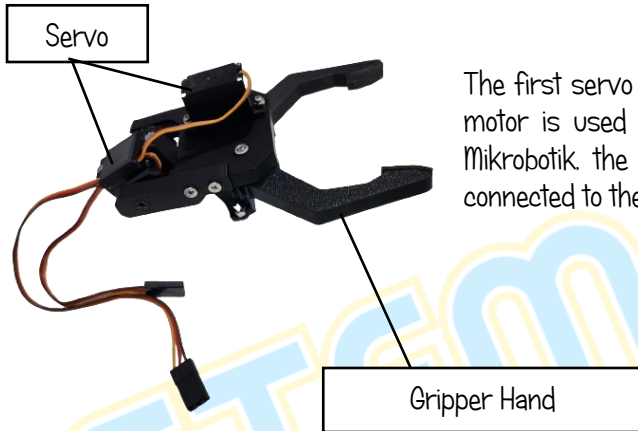


Objective 12: Efficient Space Storage

Sometimes, we need to arrange a large number of objects in a sequential manner to fully utilise space. This is where a dual servo gripper comes into play. Just as its name suggests, it is equipped with two servo motors.

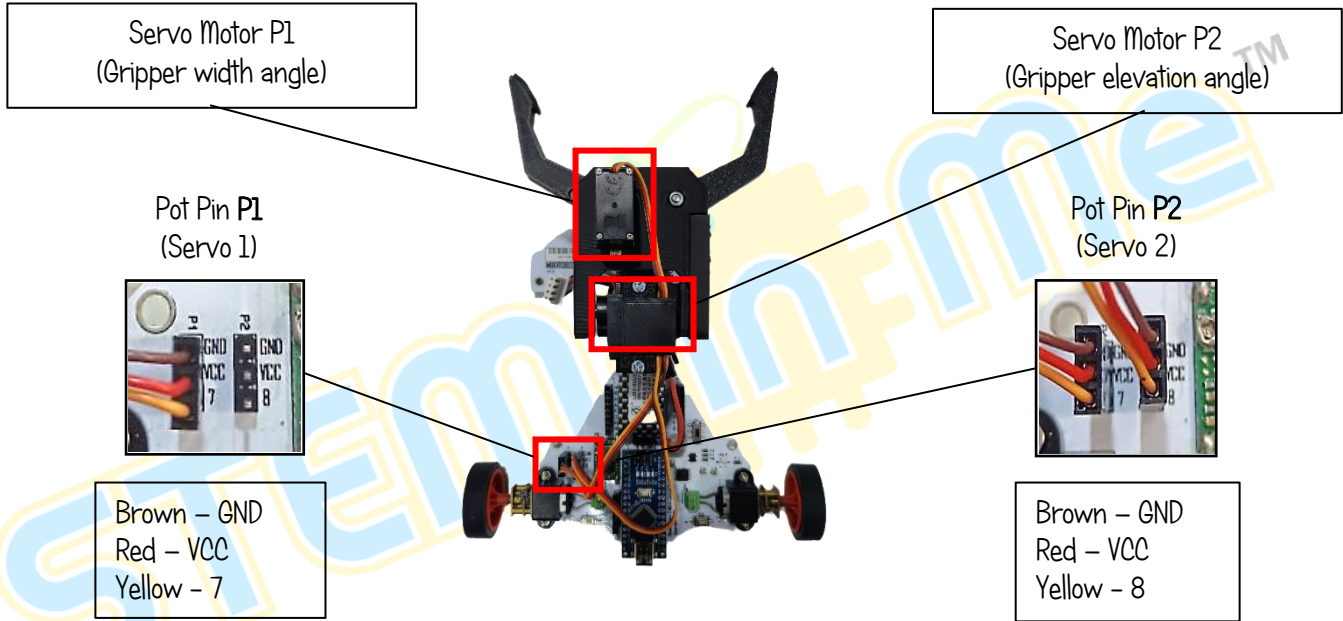


Introduction to Dual Gripper and Its Mechanism



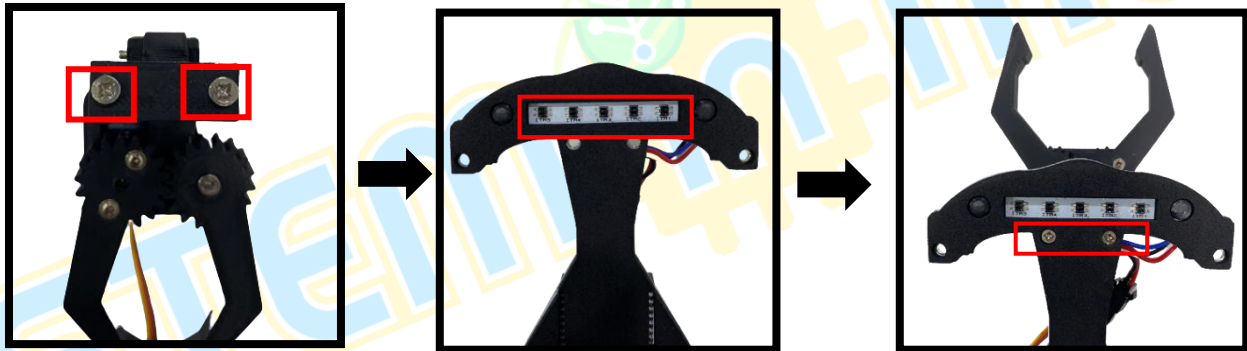
The first servo motor is used to control the gripper's width, while the second servo motor is used to control the gripper's lifting angle. Therefore, when installed on Mikrobotik, the robot will be able to grasp and lift objects. The servo wires are connected to the existing servo header pins labelled "P1" and "P2."

Installation of Dual Grippers on the robot

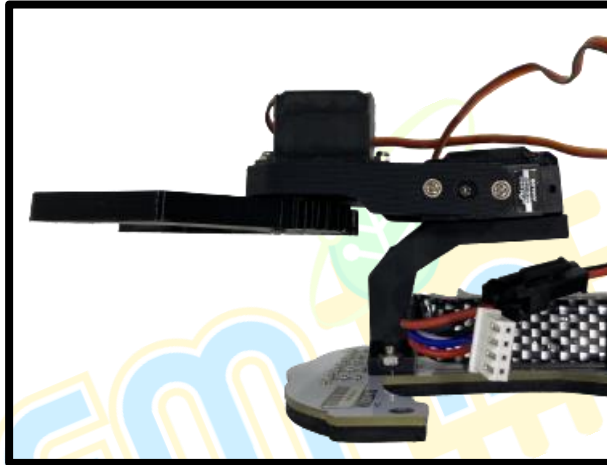


Here is the sequence of steps for installing a Dual Gripper on Mikrobotik:

- 1- Loosen the screws and nuts on the *Dual Gripper*. Below the Mikrobotik robot, there are two screw holes for the *Dual Gripper*. Insert the screws and nuts, then tighten them.



2- Ensure the position of the *Dual Gripper* after installation is like this.



Arrange Strategy and Movement Techniques.

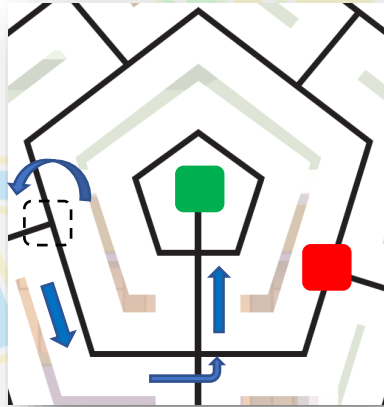
The technique used involves combining several blocks, including Path Finder, Line Tracer Time, Gripper Servo Port, and Turn at Centre.

By using the Path Finder and Line Tracer Time blocks, the robot will autonomously move along black or white lines until it encounters the first object.

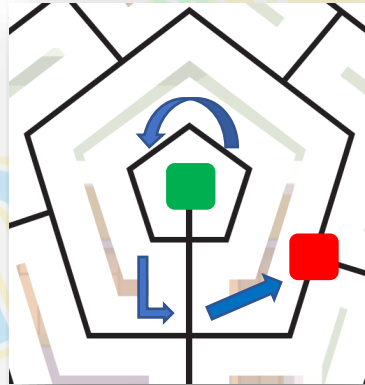
To obtain the first object (green), the robot will grasp it using the Gripper Servo Port block. Then, the robot will rotate using the Turn at Centre block and continue its movement using the Path Finder block until it reaches a left junction, where the robot will turn left. Subsequently, the robot will keep moving using the Path Finder block until it encounters a left-right junction, at which point the robot will turn left. Next, the robot will continue its movement using the Path Finder block until it reaches a dead-end and stops to place the object into a designated space using the Gripper Servo Port block. Following this, the robot will rotate using the *Turn at Centre* block and continue its movement using several *Path Finder* blocks until it encounters a left-right junction. The robot will then turn left and move using *Line Tracer Time* to pick up the second object (red) using the *Gripper Servo Port* block. After that, the robot will rotate using the *Turn at Centre* block and move using several *Path Finder* blocks until it reaches a right junction and then a right-left junction. The robot will turn right.

Subsequently, the robot will pick up the object using the *Gripper Servo Port* block and move using the *Path Finder* block until it reaches a dead-end, where it will stop to place the second object (red) sequentially into a designated space using the *Gripper Servo Port* block. Finally, the robot will rotate using the *Turn at Centre* block and move using the *Path Finder* block until it encounters a left-right junction and stops.

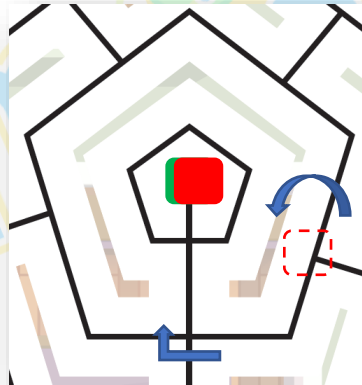
- ii) The robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a left junction and turns left. Next, the robot moves using the *Path Finder* block until it encounters a left-right junction, where it turns left. After that, the robot moves using the *Path Finder* block until it reaches a dead-end and stops. It then releases the first object (green) into a designated space using the *Gripper Servo Port* block with a small angle setting (large gripper opening).



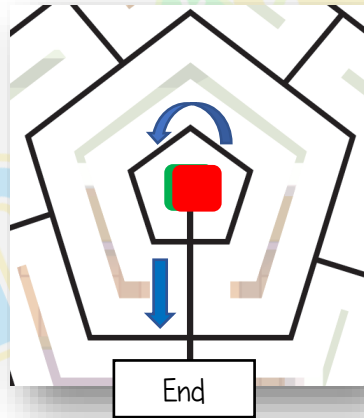
- iii) After releasing the first object (green), the robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a left-right junction and turns left. Subsequently, the robot continues to move using the *Path Finder* block until it encounters a left junction, where it turns left. Next, it uses the *Line Tracer Time* block and grasps the second object (red) using the *Gripper Servo Port* block (*Gripper*) with a large angle setting (small gripper opening).



- iv) The robot rotates to face the opposite direction using the *Turn at Centre* block. Then, the robot moves using the *Path Finder* block until it reaches a right junction and turns right. Next, the robot continues to move using the *Path Finder* block until it encounters a left-right junction, where it turns right and stops. It then lifts the second object (red) using the *Gripper Servo Port* block with a large angle setting (hand height). Afterward, the robot moves using the *Path Finder* block and proceeds to a dead-end junction where it stops. Following this, the robot sequentially places and releases the second object (red) into a designated space using the *Gripper Servo Port* block with both a large angle setting (hand height) and a small angle setting (large gripper opening).



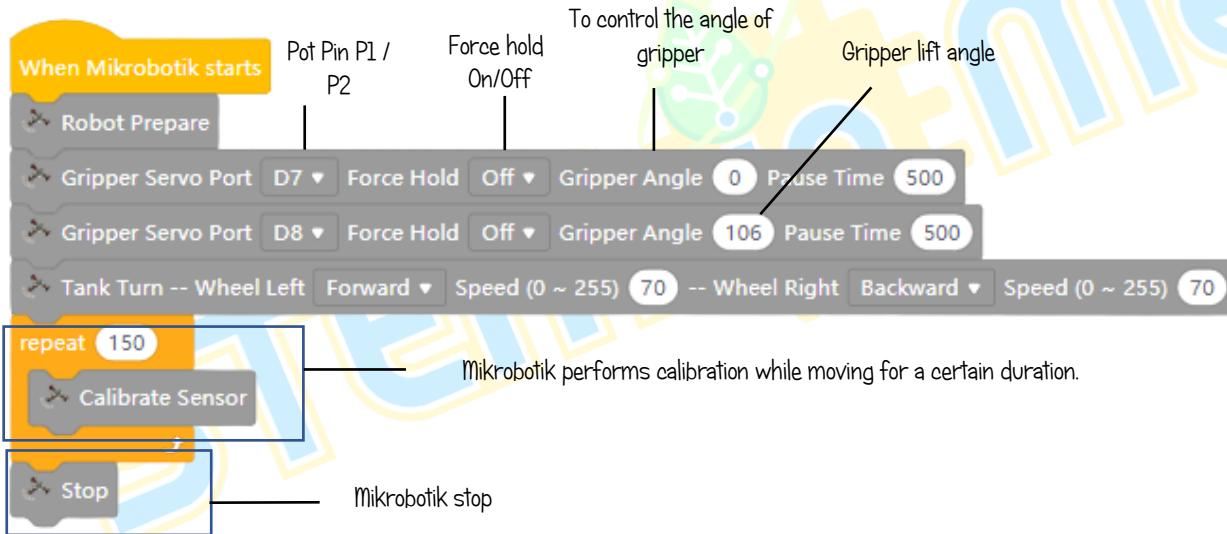
- v) The robot rotates to face the opposite direction using the *Turn at Centre* block and moves using the *Path Finder* block until it reaches a left-right junction and then stops.



Step by Step block arrangement:

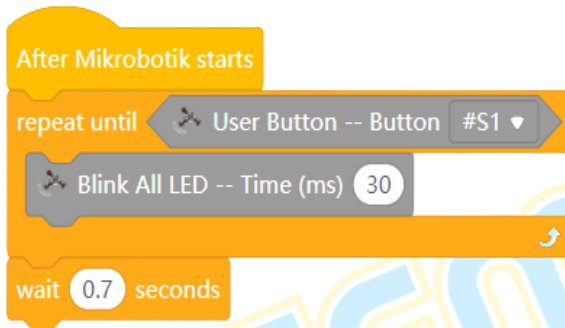
Step 1 Connect the wire of the *servo motor* to the pot labelled P1 and P2.

Step 2 Prepare the blocks arrangement for automatic calibration.



Step 3

Combine the *After Mikrobotik starts* block with a *repeat until* block, and insert a *wait 0.7 seconds* block within the *repeat until* block.



Step 4

Add 2 *Path Finder* blocks and set the values as follows (*Junction* – [*“Middle”*; *“Right”*], *Action* – [*“Turn Left”*; *“Turn Right”*], *Speed* – [*“100”*; *“100”*], *Turn Speed* – [*“100”*; *“100”*], *Junction Speed* – [*“100”*; *“100”*], *Forward Delay* – [*“70”*; *“70”*] dan *Turn Period* – [*“550”*; *“400”*]). Next, add a *Line Tracer Time* block and configure it with the following values (*Speed* – *“50”*, *Turn Speed* – *“50”*, *Time Period* – *“1100”*). Then, include a *Gripper* block with these settings (*Gripper Servo Port* – *“D7”*, *Force hold* – *“On”*, *Gripper Angle* – *“95”*, *Pause* – *“0”*). Finally, add a *Turn At Centre* block with the following parameters (*Direction Turn* – *“Left”*, *Speed* – *“80”*, *Min Turn Period* – *“600”*).



```

After Mikrobotik starts
repeat until User Button -- Button #51
  blink All LED -- Time (ms) 50
wait 0.7 seconds
Path Finder -- Line Format Black -- Junction Middle -- Action Turn Left -- Left Speed (0-255) 100 -- Right Speed (0-255) 100 -- Turn Speed (0-255) 100 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 100 -- Forward Delay 70 -- Turn Period ms (0-10000) 550
Path Finder -- Line Format Black -- Junction Right -- Action Turn Right -- Left Speed (0-255) 100 -- Right Speed (0-255) 100 -- Turn Speed (0-255) 100 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 100 -- Forward Delay 70 -- Turn Period ms (0-10000) 400
Line Tracer Time -- Line Format Black -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 50 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Time Period ms (0-10000) 1100
Gripper Servo Port D7 -- Force Hold On -- Gripper Angle 95 -- Pause Time 0
Turn At Centre -- Line Format Black -- Direction Turn Left -- Speed (0-255) 80 -- Sensor Threshold (0-1000) 20 -- Min Turn Period ms (0-1000) 600
  
```

Step 5

Add 3 *Path Finder* blocks and set the values in the following order (*Junction* – [*Left*: *Middle*: *DeadEnd*]). *Action* – [*Turn Left*: *Turn Left*: *Stop*]. *Speed* – [*100*: *100*: *70*]. *Turn Speed* – [*100*: *100*: *70*]. *Junction Speed* – [*100*: *100*: *70*]. *Forward Delay* – [*70*: *70*: *0*] dan *Turn Period* – [*400*: *700*: *0*]). Then, add 2 *Gripper* blocks and configure them with the following values (*Gripper Servo Port* – [*D7*: *D8*]. *Force hold* – [*Off*: *Off*]. *Gripper Angle* – [*0*: *106*]. *Pause* – [*0*: *500*]). Finally, include a *Turn At Centre* block with these settings (*Direction Turn* – *Left*. *Speed* – *80*. *MinTurn Period* – *600*).

Step 6

Add 2 Path Finder blocks and set the values in the following sequence (*Junction* – [*“Middle”*; *“Left”*]; *Action* – [*“Turn Left”*; *Turn Left*]; *Speed* – [*“100”*; *100*]; *Turn Speed* – [*“100”*; *100*]; *Junction Speed* – [*“100”*; *100*]; *Forward Delay* – [*“70”*; *70*] dan *Turn Period* – [*“800”*; *400*]). Add a *Line Tracer Time* block and set the values as follows (*Speed* – *“50”*; *Turn Speed* – *“50”*; *Time Period* – *“350”*). Add a *Gripper* block and configure it with the following values (*Gripper Servo Port* – *“D7”*; *Force hold* – *“On”*; *Gripper Angle* – *“100”*; *Pause* – *“0”*), followed by a *Turn At Centre* block. (*Direction Turn* – *“Left”*; *Speed* – *“80”*; *MinTurn Period* – *“800”*).

After MikroBotik starts

repeat until User Button -- Button #51

Blink All LED -- Time (ms) 30

wait 0.7 seconds

The screenshot shows a sequence of programming blocks with the following configurations:

- Path Finder** (Line Format: Black, Junction: Middle, Action: Turn Left): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 530
- Path Finder** (Line Format: Black, Junction: Right, Action: Turn Right): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 400
- Line Tracer Time** (Line Format: Black): Left Speed (50), Right Speed (50), Turn Speed (50), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Time Period ms (0-10000), 1100
- Gripper Servo Port** (D7): Force Hold (On), Gripper Angle (95), Pause Time (0)
- Turn At Centre** (Line Format: Black, Direction Turn: Left): Speed (80), Sensor Threshold (0-1000), 20, Min Turn Period ms (0-1000), 600
- Path Finder** (Line Format: Black, Junction: Left, Action: Turn Left): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 400
- Path Finder** (Line Format: Black, Junction: Middle, Action: Turn Left): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 700
- Path Finder** (Line Format: Black, Junction: DeadEnd, Action: Stop): Left Speed (70), Right Speed (70), Turn Speed (70), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (70), Forward Delay (0), Turn Period ms (0-10000), 0
- Gripper Servo Port** (D7): Force Hold (Off), Gripper Angle (0), Pause Time (0)
- Gripper Servo Port** (D8): Force Hold (Off), Gripper Angle (100), Pause Time (500)
- Turn At Centre** (Line Format: Black, Direction Turn: Left): Speed (80), Sensor Threshold (0-1000), 20, Min Turn Period ms (0-1000), 600
- Path Finder** (Line Format: Black, Junction: Middle, Action: Turn Left): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 600
- Path Finder** (Line Format: Black, Junction: Left, Action: Turn Left): Left Speed (100), Right Speed (100), Turn Speed (100), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Junction Speed (100), Forward Delay (70), Turn Period ms (0-10000), 400
- Line Tracer Time** (Line Format: Black): Left Speed (50), Right Speed (50), Turn Speed (50), RampUp Perc (100), Kp (0.00-1), Kd (0.00-1), Sensor Threshold (0-1000), Time Period ms (0-10000), 350
- Gripper Servo Port** (D7): Force Hold (On), Gripper Angle (100), Pause Time (0)
- Turn At Centre** (Line Format: Black, Direction Turn: Left): Speed (80), Sensor Threshold (0-1000), 20, Min Turn Period ms (0-1000), 800

Step 7

Add 2 *Path Finder* blocks and set the values in the following sequence (*Junction* – [*Right*: *Middle*]. *Action* – [*Turn Right*: *Turn Right*]. *Speed* – [*100*: *100*]. *Turn Speed* – [*100*: *100*]. *Junction Speed* – [*100*: *100*]. *Forward Delay* – [*100*: *100*] dan *Turn Period* – [*400*: *700*]). Next, add a *Gripper* block and configure it with the following settings (*Gripper Servo Port* – *D8*. *Force hold* – *On*. *Gripper Angle* – *70*. *Pause* – *500*).

```

After Mikrobotik starts
repeat until User Button - Button #51
  Blink All LED - Time (ms) 30
  wait 0.7 seconds
  Path Finder - Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 550
  Path Finder - Line Format Black Junction Right Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Line Tracer Time - Line Format Black Left Speed (0-255) 50 Right Speed (0-255) 50 Turn Speed (0-255) 50 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Time Period ms (0-10000) 1100
  Gripper Servo Port D7 Force Hold On Gripper Angle 95 Pause Time 0
  Turn At Centre - Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder - Line Format Black Junction Left Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Path Finder - Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 700
  Path Finder - Line Format Black Junction DeadEnd Action Stop Left Speed (0-255) 70 Right Speed (0-255) 70 Turn Speed (0-255) 70 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 70 Forward Delay 0 Turn Period ms (0-10000) 0
  Gripper Servo Port D7 Force Hold Off Gripper Angle 0 Pause Time 0
  Gripper Servo Port D8 Force Hold Off Gripper Angle 106 Pause Time 500
  Turn At Centre - Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder - Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 800
  Path Finder - Line Format Black Junction Left Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Line Tracer Time - Line Format Black Left Speed (0-255) 50 Right Speed (0-255) 50 Turn Speed (0-255) 50 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Time Period ms (0-10000) 350
  Gripper Servo Port D7 Force Hold On Gripper Angle 100 Pause Time 0
  Turn At Centre - Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder - Line Format Black Junction Right Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 100 Turn Period ms (0-10000) 400
  Path Finder - Line Format Black Junction Middle Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 100 Turn Period ms (0-10000) 700
  Gripper Servo Port D8 Force Hold On Gripper Angle 70 Pause Time 500
  
```


Step 8

Add a *Path Finder* block and set the values as follows (*Junction* – *DeadEnd*: *Action* – *Stop*: *Speed* – *70*; *Turn Speed* – *70*; *Junction Speed* – *70*]. *Forward Delay* – *0* dan *Turn Period* – *0*). Then, add 2 *Gripper* blocks and configure them with the following settings (*Gripper Servo Port* – [*D8*: *D7*]. *Force hold* – [*Off*: *Off*]. *Gripper Angle* – [*90*: *0*]. *Pause* – [*500*: *0*]).

After MikroBotik starts

```

repeat until User Button Button #51
  Blink All LED Time (ms) 30
wait 0.7 seconds
  Path Finder Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 550
  Path Finder Line Format Black Junction Right Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Line Tracer Time Line Format Black Left Speed (0-255) 50 Right Speed (0-255) 50 Turn Speed (0-255) 50 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Time Period ms (0-10000) 1100
  Gripper Servo Port D7 Force Hold On Gripper Angle 95 Pause Time 0
  Turn At Centre Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder Line Format Black Junction Left Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Path Finder Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 700
  Path Finder Line Format Black Junction DeadEnd Action Stop Left Speed (0-255) 70 Right Speed (0-255) 70 Turn Speed (0-255) 70 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 70 Forward Delay 0 Turn Period ms (0-10000) 0
  Gripper Servo Port D7 Force Hold Off Gripper Angle 0 Pause Time 0
  Gripper Servo Port D8 Force Hold Off Gripper Angle 108 Pause Time 500
  Turn At Centre Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder Line Format Black Junction Middle Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 600
  Path Finder Line Format Black Junction Left Action Turn Left Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 70 Turn Period ms (0-10000) 400
  Line Tracer Time Line Format Black Left Speed (0-255) 50 Right Speed (0-255) 50 Turn Speed (0-255) 50 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Time Period ms (0-10000) 350
  Gripper Servo Port D7 Force Hold On Gripper Angle 100 Pause Time 0
  Turn At Centre Line Format Black Direction Turn Left Speed (0-255) 80 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 600
  Path Finder Line Format Black Junction Right Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 100 Turn Period ms (0-10000) 400
  Path Finder Line Format Black Junction Middle Action Turn Right Left Speed (0-255) 100 Right Speed (0-255) 100 Turn Speed (0-255) 100 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 100 Forward Delay 100 Turn Period ms (0-10000) 700
  Gripper Servo Port D8 Force Hold On Gripper Angle 70 Pause Time 500
  Path Finder Line Format Black Junction DeadEnd Action Stop Left Speed (0-255) 70 Right Speed (0-255) 70 Turn Speed (0-255) 70 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Junction Speed 70 Forward Delay 0 Turn Period ms (0-10000) 0
  Gripper Servo Port D8 Force Hold Off Gripper Angle 90 Pause Time 500
  Gripper Servo Port D7 Force Hold Off Gripper Angle 0 Pause Time 0
  
```

Step 9

Add a *Turn At Centre* block with the following settings (*Direction Turn* – “*Left*”. *Speed* – “*80*”. *Min Turn Period* – “*600*”). Then, add a *Path Finder* block and configure it with the following values (*Junction* – “*Middle*”. *Action* – “*Stop*”. *Speed* – “*100*”. *Turn Speed* – “*100*”. *Junction Speed* – “*100*”. *Forward Delay* – “*70*” and *Min Turn Period* – “*100*”).

After MikroBotik starts

repeat until User Button -- Button #51

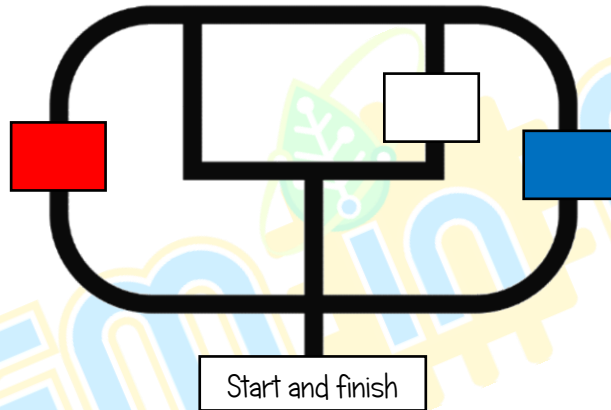
Blink All LED -- Time (ms) 85

wait 0.7 seconds

Path Finder	Line Format	Black	Junction	Middle	Action	Turn Left	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	550
Path Finder	Line Format	Black	Junction	Right	Action	Turn Right	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	400
Line Tracer Time	Line Format	Black	Left Speed (0-255)	50	Right Speed (0-255)	50	Turn Speed (0-255)	50	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Time Period ms (0-10000)	1100								
Gripper Servo Port	D7	Force Hold	On	Gripper Angle	95	Pause Time	0																			
Turn At Centre	Line Format	Black	Direction Turn	Left	Speed (0-255)	80	Sensor Threshold (0-1000)	20	Min Turn Period ms (0-1000)	600																
Path Finder	Line Format	Black	Junction	Left	Action	Turn Left	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	400
Path Finder	Line Format	Black	Junction	Middle	Action	Turn Left	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	700
Path Finder	Line Format	Black	Junction	DeadEnd	Action	Stop	Left Speed (0-255)	70	Right Speed (0-255)	70	Turn Speed (0-255)	70	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	70	Forward Delay	0	Turn Period ms (0-10000)	0
Gripper Servo Port	D7	Force Hold	Off	Gripper Angle	0	Pause Time	0																			
Gripper Servo Port	D8	Force Hold	Off	Gripper Angle	106	Pause Time	500																			
Turn At Centre	Line Format	Black	Direction Turn	Left	Speed (0-255)	80	Sensor Threshold (0-1000)	20	Min Turn Period ms (0-1000)	600																
Path Finder	Line Format	Black	Junction	Middle	Action	Turn Left	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	600
Path Finder	Line Format	Black	Junction	Left	Action	Turn Left	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	400
Line Tracer Time	Line Format	Black	Left Speed (0-255)	50	Right Speed (0-255)	50	Turn Speed (0-255)	50	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Time Period ms (0-10000)	350								
Gripper Servo Port	D7	Force Hold	On	Gripper Angle	100	Pause Time	0																			
Turn At Centre	Line Format	Black	Direction Turn	Left	Speed (0-255)	80	Sensor Threshold (0-1000)	20	Min Turn Period ms (0-1000)	800																
Path Finder	Line Format	Black	Junction	Right	Action	Turn Right	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	100	Turn Period ms (0-10000)	400
Path Finder	Line Format	Black	Junction	Middle	Action	Turn Right	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	100	Turn Period ms (0-10000)	700
Gripper Servo Port	D8	Force Hold	On	Gripper Angle	70	Pause Time	500																			
Path Finder	Line Format	Black	Junction	DeadEnd	Action	Stop	Left Speed (0-255)	70	Right Speed (0-255)	70	Turn Speed (0-255)	70	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	70	Forward Delay	0	Turn Period ms (0-10000)	0
Gripper Servo Port	D8	Force Hold	Off	Gripper Angle	80	Pause Time	500																			
Gripper Servo Port	D7	Force Hold	Off	Gripper Angle	0	Pause Time	0																			
Turn At Centre	Line Format	Black	Direction Turn	Left	Speed (0-255)	80	Sensor Threshold (0-1000)	20	Min Turn Period ms (0-1000)	600																
Path Finder	Line Format	Black	Junction	Middle	Action	Stop	Left Speed (0-255)	100	Right Speed (0-255)	100	Turn Speed (0-255)	100	RampUp Perc (0-100)	100	Kp (0.00-1)	0.02	Kd (0.00-1)	0.2	Sensor Threshold (0-1000)	20	Junction Speed	100	Forward Delay	70	Turn Period ms (0-10000)	100

Challenge!!

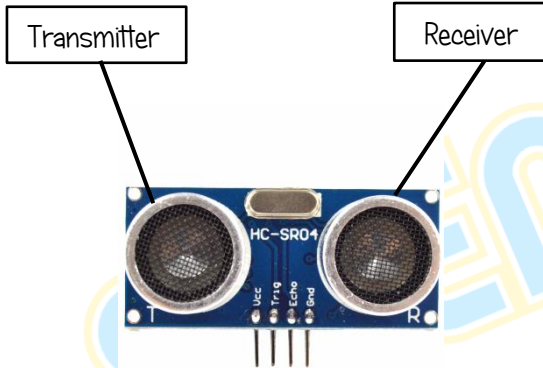
Move the blue and red object into the designed space.



Objective 13: Stop! Obstacles in front

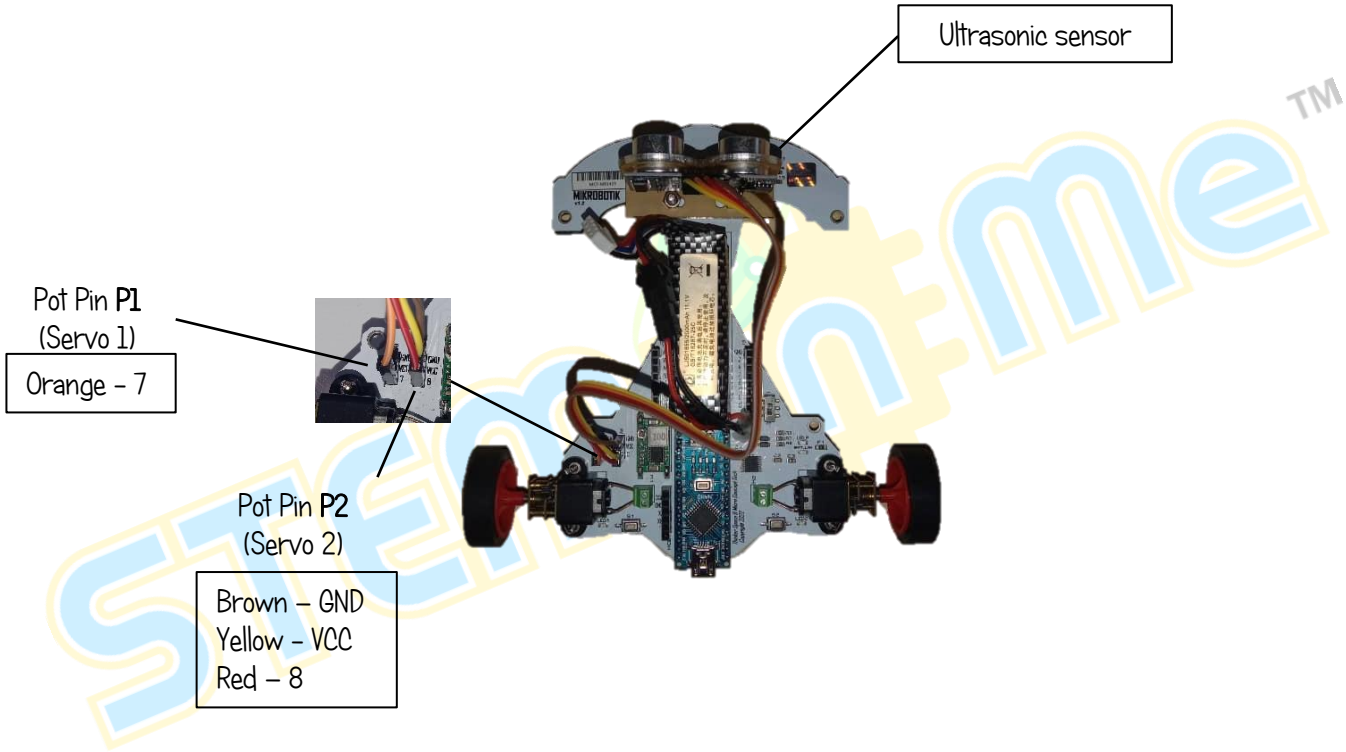
Sometimes there are obstacles along the Robot's pathway, we must make sure that our robot does not crash the obstacle. Here is where we will use the new component – Ultrasonic Sensor. The objective focuses on the robot to stop when there is an obstacle. For this purpose, the robot utilizes an ultrasonic sensor to detect any obstacles.

Introduction to Ultrasonic and Its Mechanism



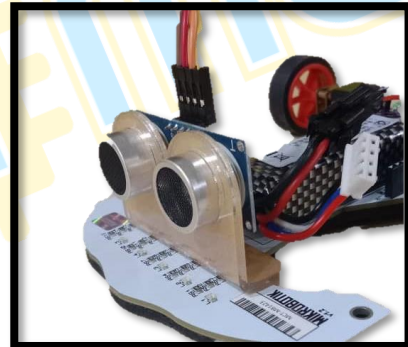
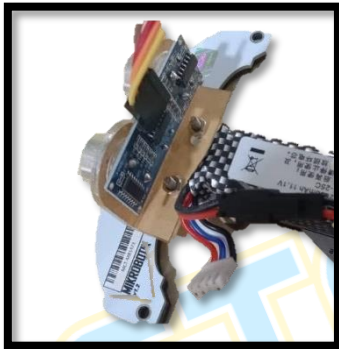
An ultrasonic sensor uses ultrasonic sound waves to measure an object's distance and then transforms the reflected sound waves into an electrical signal. There are two main components of ultrasonic sensors. The transmitter uses piezoelectric crystals to generate sound waves towards the obstacles, and the receiver will receive the reflected sound waves after it has travelled from the obstacles. The ultrasonic sensor will calculate the obstacle's distance based on the received echo.

Installation of Ultrasonic Sensor on the robot



Here is the sequence of steps for installing an Ultrasonic Sensor on Mikrobotik:

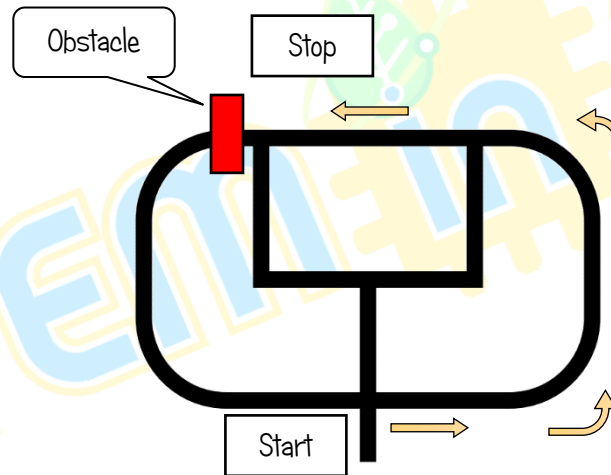
- 1- Loosen the screws and nuts on the Ultrasonic bracket. Below the Mikrobotik robot, there are two screw holes for the *Ultrasonic bracket*. Insert the screws and nuts, then tighten them.
- 2- Ensure the position of the *Ultrasonic Sensor* after installation is like this.



Arrange Strategy and Movement Techniques.

The technique used involves combining several blocks, including the *Line Tracer Obstruct*.

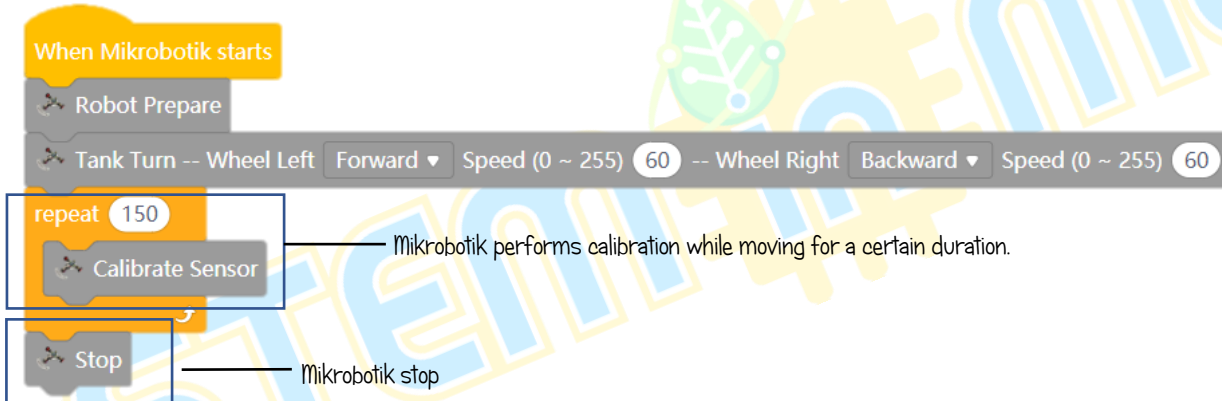
By using the *Line Tracer Obstruct* block, the robot will move autonomously along a black or white line until it encounters a solid obstacle and the robot will stop at a specific distance.



Step by Step block arrangement:

Step 1 Connect the wire of the *servo motor* to the pot labelled P1 and P2.

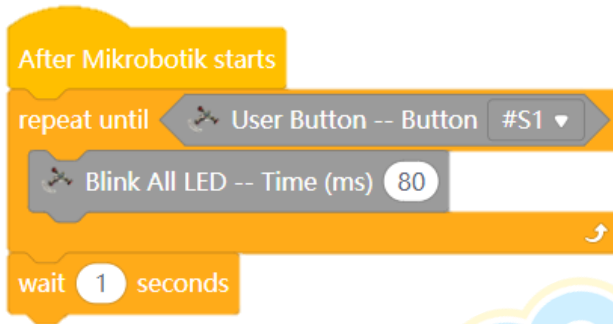
Step 2 Prepare the blocks arrangement for automatic calibration.



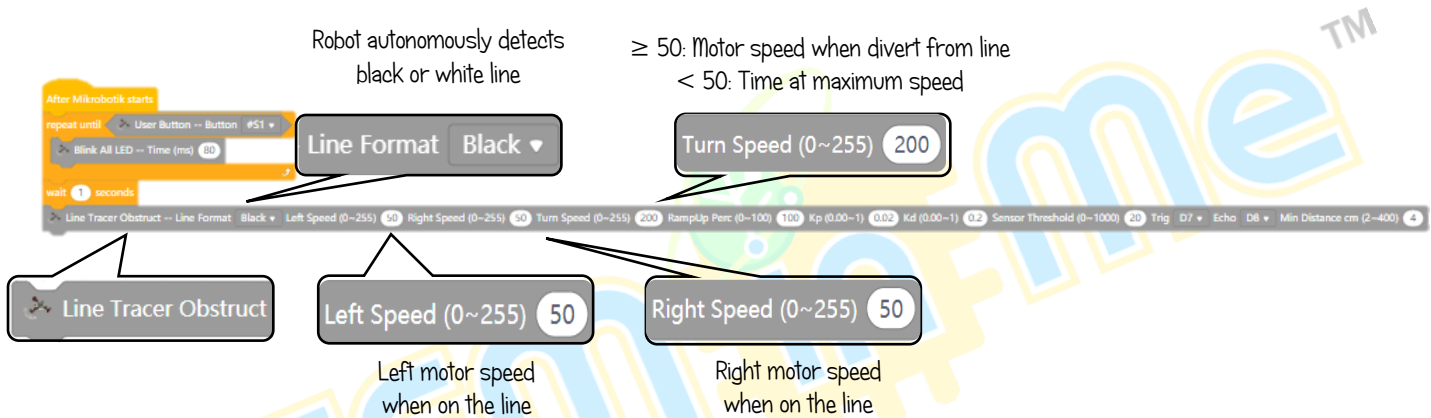


Step 3

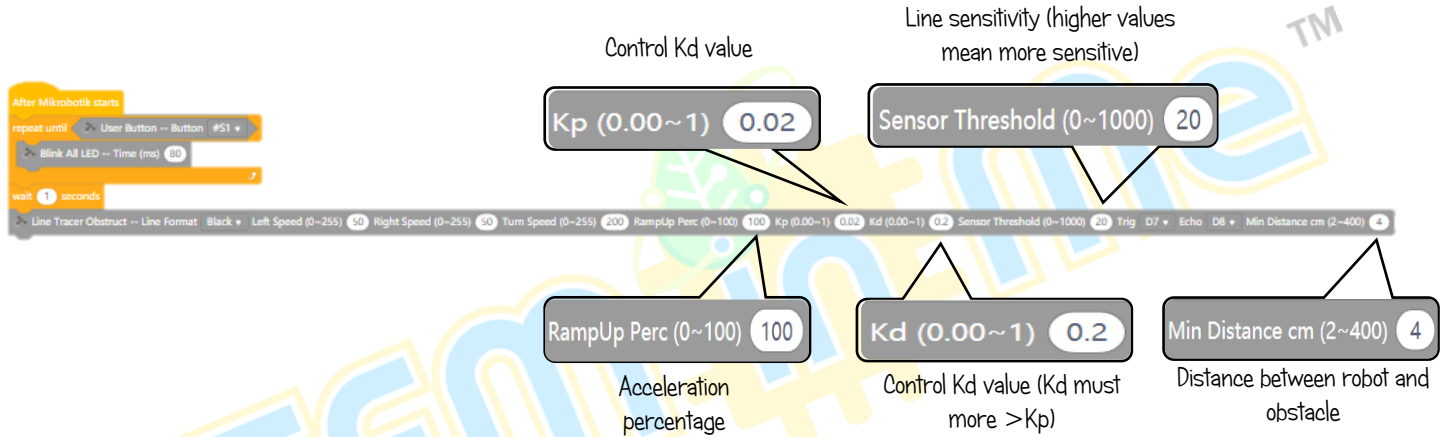
Combine the *After Mikrobotik starts* block with a *repeat until* block, and insert a *wait 1 seconds* block within the *repeat until* block.



Step 4 Add 1 *Line Tracer Obstruct* block and set the values as follows:

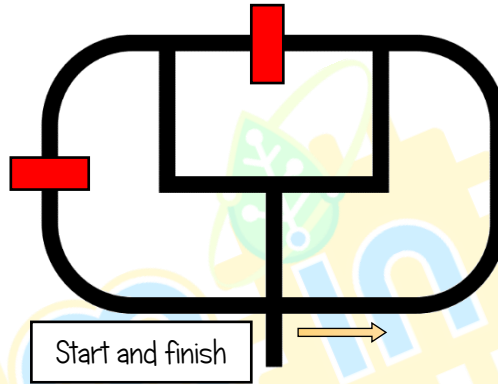


Continuation



Challenge!!

Stop at the first obstacle, then move and stop at the second obstacle then finish.



Project 14: Lines of different colours? Let's finish it!

There are situations where the robot needs to go through lines with different colours in a circuit. The movement of the robot will be applied using the concept of changing from a black line to a white line or vice versa. To implement this concept, the robot will use the Path Finder block with the option of middle junction and dead end.

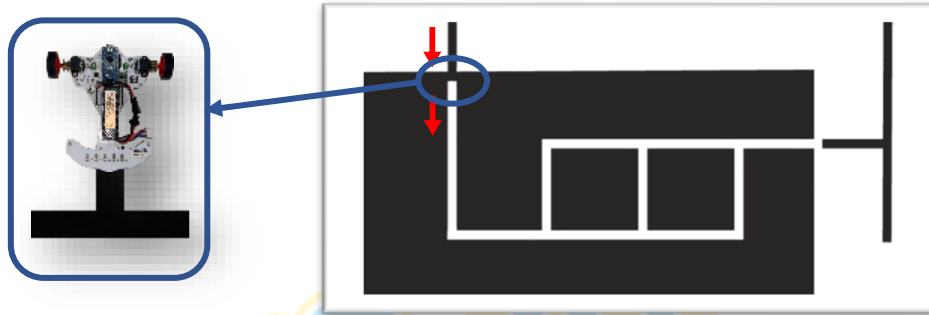
Introduction to Movement and its Mechanism

The technique used is to combine several Path Finder blocks.

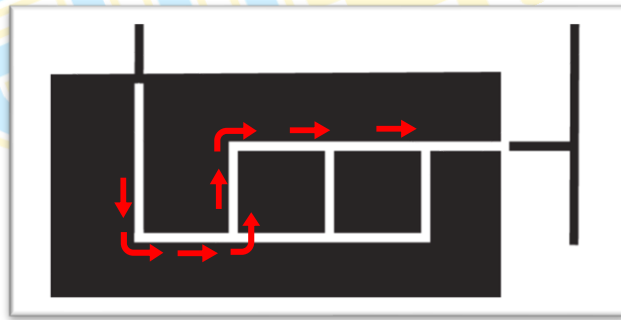
The robot moves autonomously following the line by using the Path Finder block with Line Format according to the colour of the line either black or white. Usually, the place where the colour of the line changes will be distinguished by a middle junction or a dead end. Therefore, the robot needs to identify the type of intersection to go through when the black line changes to white or vice versa.

Here is a sketch of the robot's movements to complete the task.

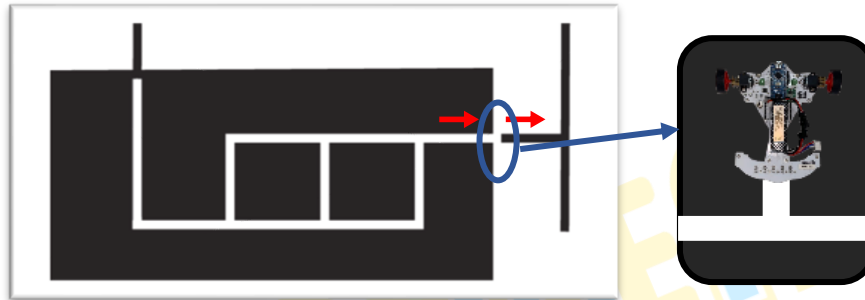
- i) Robot moves autonomously by using the *Path Finder* following the black line with the middle junction option. This is because the place where the line colour changes on the circuit is the middle junction as shown. Then the robot will move along the white line.



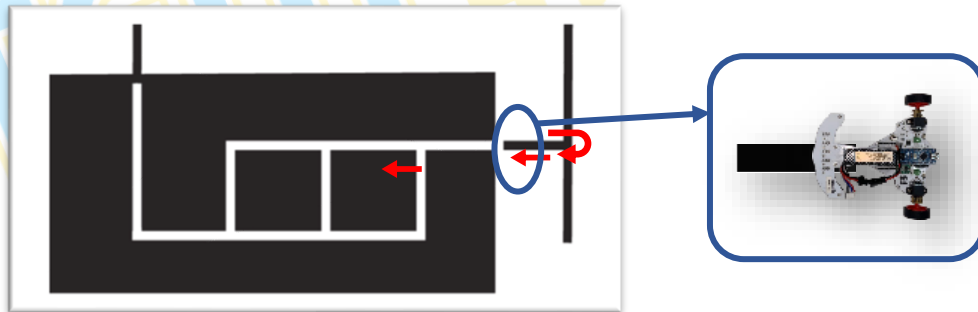
- ii) Robot turns left when at a left intersection and turns right when at a right intersection by using the *Path Finder* block.



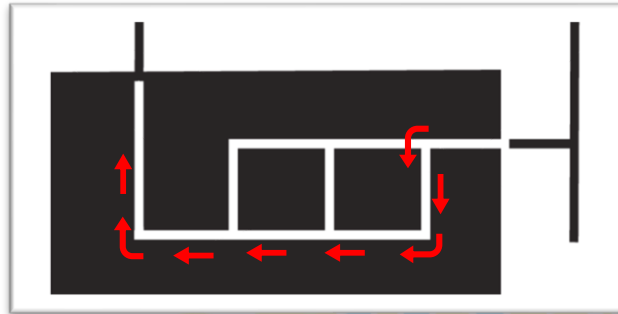
- iii) Robot moves autonomously by using the *Path Finder* following the white line with the middle junction option. This is because the place where the line colour changes on the circuit is the middle junction as shown. Then the robot will move along the black line.



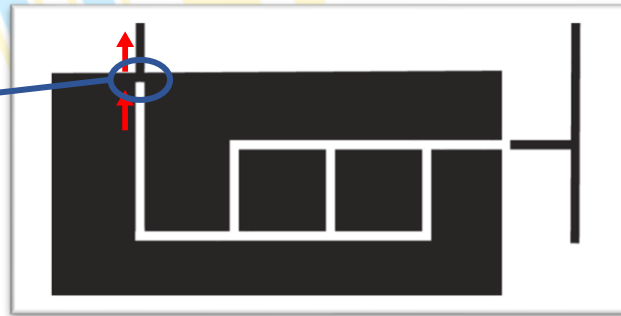
- iv) Robot makes a turn using the *Path Finder Tank* block. Then, the robot moves along the black line using the *Path Finder* block with the *Dead End* option. This is because the place where the line colour changes on the circuit is a dead end as shown. Then the robot will move along the white line.



- v) Robot will turn left at the left intersection and turn right at the right intersection following the white line by using the *Path Finder* block.



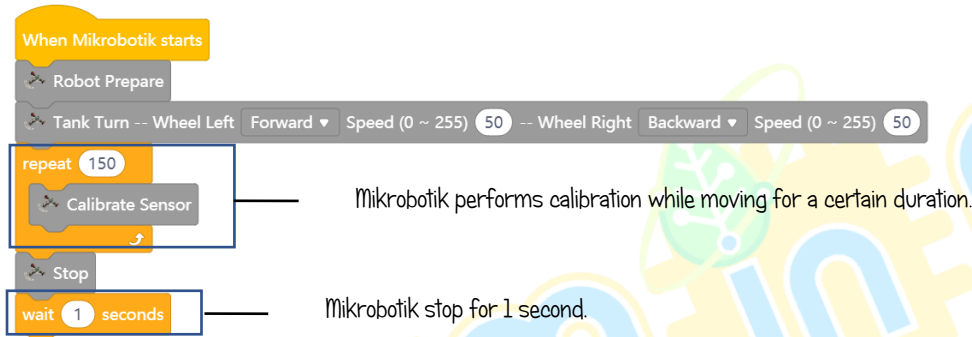
- vi) Robot will move along the white line using the *Path Finder* block with the *Dead End* option. This is because the place where the line colour changes on the circuit is a dead end as shown. Then the robot will move along the black line.



Step by Step Block Arrangement:

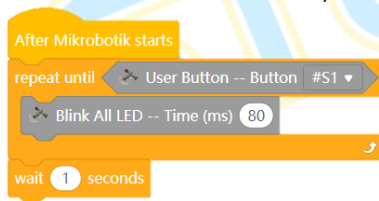
Step 1

Prepare the blocks arrangement for automatic calibration.



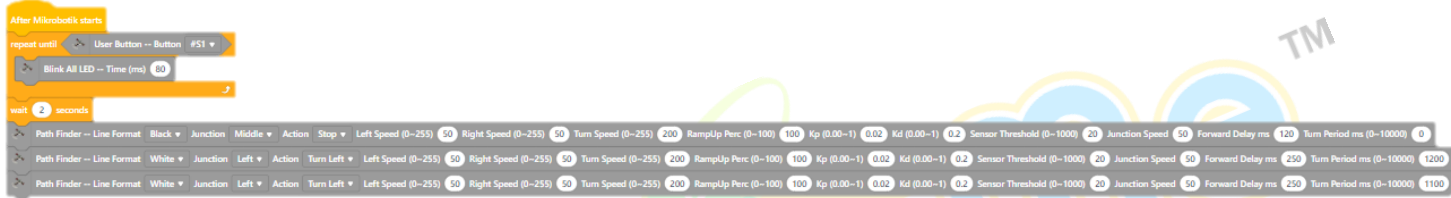
Step 2

Combine the *After Mikrobotik starts* block with the *repeat until* block and enter the *wait 1 second* block into the *repeat until* block.



Step 3

Add 3 *Path Finder* blocks and set *Line Format* to **Black, White, White**. *Junction* to **Middle, Left, Left**. *Action* to **Stop, Turn Left, Turn Left**. *Forward Delay* **120, 250, 250** and *Turn Period* **0, 1200** and **1100**.



NOTES

When the color of the line changes from black to white or vice versa, the addition of the Forward Delay value needs to be done so that the robot will pass the color of the initial line to detect the color of the next line.

Step 4

Add 4 *Path Finder* blocks and set Line Format to **White**. *Junction* to **Right, Right, Right, Middle**. *Action* to **Turn Right, Stop, Stop, Stop**. *Forward Delay* to **230, 120, 120, 120** and *Turn Period* to **1270, 0, 0, 0**.

The screenshot shows a Scratch script with the following blocks:

- After MikroBotik starts
- repeat until (User Button -- Button #51)
- Blink All LED -- Time (ms) 80
- wait 2 seconds
- Path Finder -- Line Format: White, Junction: Right, Action: Turn Right, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, 0.02, Kd: 0.00-1, 0.2, Sensor Threshold: 0-1000, 20, Junction Speed: 50, Forward Delay ms: 230, Turn Period ms: 1270
- Path Finder -- Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, 0.02, Kd: 0.00-1, 0.2, Sensor Threshold: 0-1000, 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0
- Path Finder -- Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, 0.02, Kd: 0.00-1, 0.2, Sensor Threshold: 0-1000, 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0
- Path Finder -- Line Format: White, Junction: Middle, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, 0.02, Kd: 0.00-1, 0.2, Sensor Threshold: 0-1000, 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0



Step 5

Add a *Path Finder Tank* block with *Line Format Black*, *Junction Middle*, *Action Turn Right*, *Forward Delay 500* and *Min Turn Period 900*. Then, add a *Path Finder* block with *Line Format Black*, *Junction DeadEnd*, *Action Stop*, *Forward Delay 120* and *Turn Period 0*.

The script consists of the following blocks:

- After MikroBotik starts
- repeat until User Button -> Button #51
- Blink All LED -> Time (ms) 60
- wait 2 seconds
- Path Finder -- Line Format Black -- Junction Middle -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Left -- Action Turn Left -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 250 -- Turn Period ms (0-10000) 1200
- Path Finder -- Line Format White -- Junction Right -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 230 -- Turn Period ms (0-10000) 1270
- Path Finder -- Line Format White -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Middle -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder Tank -- Line Format Black -- Junction Middle -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 500 -- Min Turn Period ms (0-10000) 900
- Path Finder -- Line Format Black -- Junction DeadEnd -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0

Step 6

Add 4 *Path Finder* blocks and set *Line Format* to **White**. *Junction* to **Left**, **Right**, **Right** and **Right**. *Action* to **Turn Left**, **Turn Right**, **Stop** and **Stop**. *Forward Delay* to 250, 230, 120, 120 and *Min Turn Period* to 1100, 1270, 0 and 0.

The screenshot shows a Scratch script starting with a 'wait 2 seconds' block, followed by a 'repeat until' loop containing a 'Blink All LED' block. The main script consists of 12 'Path Finder' blocks with the following settings:

- Block 1: Line Format: Black, Junction: Middle, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.
- Block 2: Line Format: White, Junction: Left, Action: Turn Left, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 250, Turn Period ms: 1200.
- Block 3: Line Format: White, Junction: Left, Action: Turn Left, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 250, Turn Period ms: 1100.
- Block 4: Line Format: White, Junction: Right, Action: Turn Right, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 230, Turn Period ms: 1270.
- Block 5: Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.
- Block 6: Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.
- Block 7: Line Format: Black, Junction: Middle, Action: Turn Right, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 900.
- Block 8: Line Format: Black, Junction: Middle, Action: Turn Right, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 500, Min Turn Period ms: 10000, 900.
- Block 9: Line Format: Black, Junction: DeadEnd, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.
- Block 10: Line Format: White, Junction: Left, Action: Turn Left, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 250, Turn Period ms: 1100.
- Block 11: Line Format: White, Junction: Right, Action: Turn Right, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 230, Turn Period ms: 1270.
- Block 12: Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.
- Block 13: Line Format: White, Junction: Right, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 200, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay ms: 120, Turn Period ms: 0.



Step 7

Add 3 *Path Finder* blocks with *Line Format* White, White, Black. *Junction* Right, DeadEnd, Right. *Action* Turn Right, Stop, Stop, *Forward Delay* 230, 300, 300 and *Turn Period* 1270, 0, 0.

The screenshot shows a Scratch script with the following blocks:

- After Mikrobotik starts
- repeat until (User Button -- Button #51)
- Blink All LED -- Time (ms) 80
- wait 2 seconds
- Path Finder -- Line Format Black -- Junction Middle -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Left -- Action Turn Left -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 250 -- Turn Period ms (0-10000) 1200
- Path Finder -- Line Format White -- Junction Left -- Action Turn Left -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 250 -- Turn Period ms (0-10000) 1100
- Path Finder -- Line Format White -- Junction Right -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 230 -- Turn Period ms (0-10000) 1270
- Path Finder -- Line Format White -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Middle -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder Tank -- Line Format Black -- Junction Middle -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 500 -- Min Turn Period ms (0-10000) 900
- Path Finder -- Line Format White -- Junction DeadEnd -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Left -- Action Turn Left -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 250 -- Turn Period ms (0-10000) 1100
- Path Finder -- Line Format White -- Junction Right -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 230 -- Turn Period ms (0-10000) 1270
- Path Finder -- Line Format White -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 120 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format White -- Junction Right -- Action Turn Right -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 230 -- Turn Period ms (0-10000) 1270
- Path Finder -- Line Format White -- Junction DeadEnd -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 300 -- Turn Period ms (0-10000) 0
- Path Finder -- Line Format Black -- Junction Right -- Action Stop -- Left Speed (0-255) 50 -- Right Speed (0-255) 50 -- Turn Speed (0-255) 200 -- RampUp Perc (0-100) 100 -- Kp (0.00-1) 0.02 -- Kd (0.00-1) 0.2 -- Sensor Threshold (0-1000) 20 -- Junction Speed 50 -- Forward Delay ms 300 -- Turn Period ms (0-10000) 0



Extra: Try Upgrade and Self Program

DEVICE PORT	ARDUINO NANO PIN	PERIPHERALS	ADDITIONAL INFO
ITR1	A6	Line Detection Sensor – Outer Left	ITR8307
ITR2	A3	Line Detection Sensor – Inside Left	ITR8307
ITR3	A2	Line Detection Sensor – Middle	ITR8307
ITR4	A1	Line Detection Sensor – Inside Right	ITR8307
ITR5	A0	Line Detection Sensor – Outer Right	ITR8307
S1	A7	User Switch S1	Value < 100
S2	A7	User Switch S2	Value ≥ 100 & < 400
BUZZER	D2	Buzzer	
LED1	D13	Indicator Light L1	
LED2	D12	Indicator Light L2	
M1 – AIN1	D5	Left Motor – Bridge A Input 1	DRV8833 Dual H-Bridge Motor Driver
M1 – AIN2	D6	Left Motor – Bridge A Input 2	DRV8833 Dual H-Bridge Motor Driver
M2 – BIN1	D3	Right motor – Bridge B Input 1	DRV8833 Dual H-Bridge Motor Driver
M2 – BIN2	D9	Right motor – Bridge B Input 2	DRV8833 Dual H-Bridge Motor Driver
P1	D7	Open Pot P1	
P2	D8	Open Pot P2	
BT – TX	D10	Bluetooth Pot TX	
BT – RX	D11	Bluetooth Pot RX	

MERAKYATKAN TEKNOLOGI

- Industry 4WRD
- Pemikiran Kreatif
- Pembudayaan Inovasi
- Kesejahteraan Hidup
- Kelestarian Alam
- Pembelajaran
Menyeronokkan

PENGLUAR:

MICRO CONCEPT TECH SDN BHD
1230153-W

No. 5-5, Pusat Dagangan Shah Alam,
Persiaran Damai, Seksyen 11,
40100 Shah Alam, Selangor, Malaysia

  @steminme



 <http://www.microconcept.com.my>

 steminme@microconcept.com.my