

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.



Isi Kandungan

Elemen Pada Robotik.....	1
Apa itu perkakasan elektronik?.....	2
Apa itu pengaturcaraan perisian?.....	3
Robot Berautonomi.....	4
Kandungan di dalam kotak.....	5
"Mikrobotik" Robot Berautonomi.....	6
Mikropengawal Arduino Nano.....	7
Bateri LiPo.....	8
Indikator Bateri Rendah.....	9
Pemasangan perisian mBlock v5.....	10
Cara untuk menambah Mikrobotik.....	12
Proses kalibrasi.....	13
Algoritma PID Robot Berautonomi.....	19

Apa jenis-jenis Litar?	20
Jenis-jenis Persimpangan	21
Objektif 1: Vroom Vroom	22
Pengenalan Mudah Pembaz	22
Langkah-langkah susunan blok	22
Cabaran !!	24
Objektif 2: Tolong Hidupkan Lampul	25
Pengenalan Mudah Diod Pemancar Cahaya (LED)	25
Langkah-langkah susunan blok	25
Cabaran!!	27
Objektif 3: Mulakan Pengembaraan Kita (Pergerakan Bebas)	28
Pengenalan kepada Motor	28
Pengenalan Mudah Pergerakan Asas Robot	29
Langkah-langkah susunan blok	32
Cabaran!!	37

Objektif 4: Ayuh Ikuti Garisan Itu	38
Pengenalan mudah Pengesanan Garisan.....	38
Pengenalan <i>Line Tracer Time</i> dan Mekanismanya.....	38
Langkah-langkah susunan blok.....	39
Cabaran!!.....	43
Objektif 5: Apa Yang Perlu Dilakukan Ketika Di Persimpangan?	44
Pengenalan <i>Path Finder</i> dan Mekanismanya.	44
Langkah-langkah susunan blok.....	45
Cabaran!!.....	50
Objektif 6: Apa Lagi Boleh Dilakukan Ketika Di Persimpangan?	52
Pengenalan <i>Path Finder Tank</i> dan Mekanismanya.	52
Langkah-langkah susunan blok.....	53
Cabaran!!.....	58
Objektif 7: Salah Jalan? Buat Pusingan-U	60
Pengenalan <i>Turn at Centre</i> dan Mekanismanya.....	60

Langkah-langkah susunan blok.....	61
Cabaran!!	63
Objektif 8: Ayuh kawal Mikrobotik.....	64
Pengenalan Bluetooth dan Mekanismanya.....	64
Langkah-langkah susunan blok.....	65
Penggunaan Peranti Pintar Mikrobotik.....	70
Cabaran!!	71
Objektif 9: Kita Perlukan Peronda Kawasan !.....	72
Susun Atur Stretegi dan Teknik Pergerakan.....	72
Langkah-langkah susunan blok.....	75
Objektif 10: Mari Mencari Harta Tersembunyi.....	77
Pengenalan Butang Tekan / Suis dan Mekanismanya.....	78
Susun Atur Stretegi dan Teknik Pergerakan.....	78
Langkah-langkah susunan blok.....	82
Objektif 11: Pengasingan Bahan Kitar Semula.....	86

Pengenalan Pencengkam Tunggal dan Mekanismanya.....	87
Susun Atur Strategi dan Teknik Pergerakan.....	91
Langkah-langkah susunan blok.....	96
Maklumat Tambahan.....	101
Objektif 12: Penyimpanan Cekap Ruang.....	102
Pengenalan Pencengkam Berganda dan Mekanismanya.....	103
Pemasangan Pencengkam Berganda pada robot.....	104
Susun Atur Strategi dan Teknik Pergerakan.....	107
Langkah-langkah susunan blok.....	113
Tambahan: Cuba Naik Taraf dan Pengatucaraan Sendiri.....	121

MIKROBOTIK



Struktur Mekanikal



Pergerakan mekanikal

Elemen Pada Robotik



Perkakasan Elektronik



Pengaturcaraan perisian

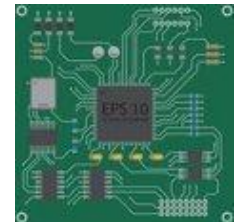
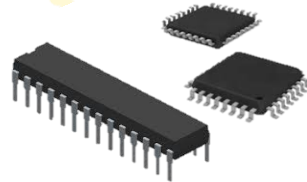
Apa itu perkakasan elektronik?



Mengesan
dan merasa
pada
persekitaran



Mengawal atau
bertindak balas
pada
persekitaran



Apa itu pengaturcaraan perisian?

Pengawal Perkakasan Elektronik



Set arahan ditulis menggunakan bahasa tertentu

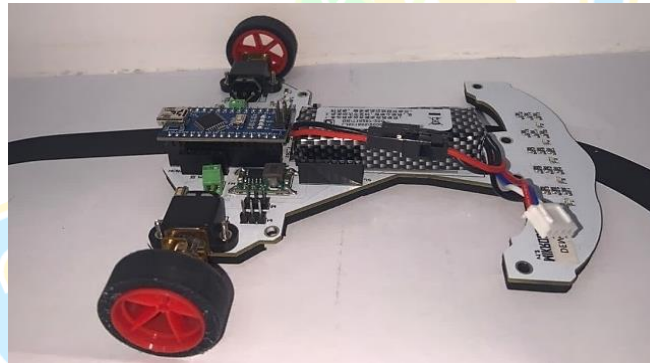


Berfungsi



Robot Berautonomi

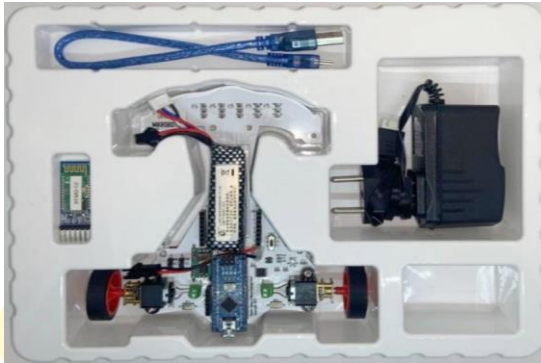
Robot direka dan dibina khas untuk mengesan dan bergerak secara automatik atau berautonomi mengikut garis putih dan hitam. Selain itu, robot juga direka untuk fungsi lain. Sebagai contoh, mengesan halangan dan menggerakkan objek.



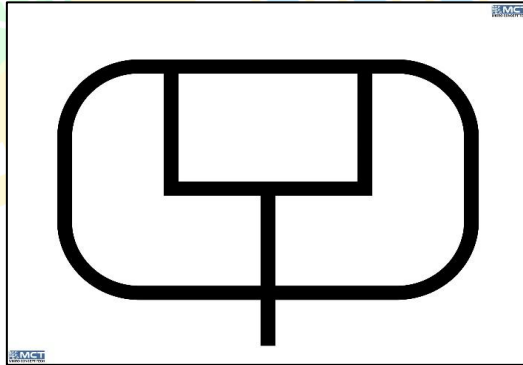
Gambar 1: "Mikrobotik" Robot Berautonomi

Kandungan di dalam kotak

- 1x kabel USB
- 1x Pengecas
- 1x Mikrobotik
- 1x Modul Bluetooth
- 1x Litar

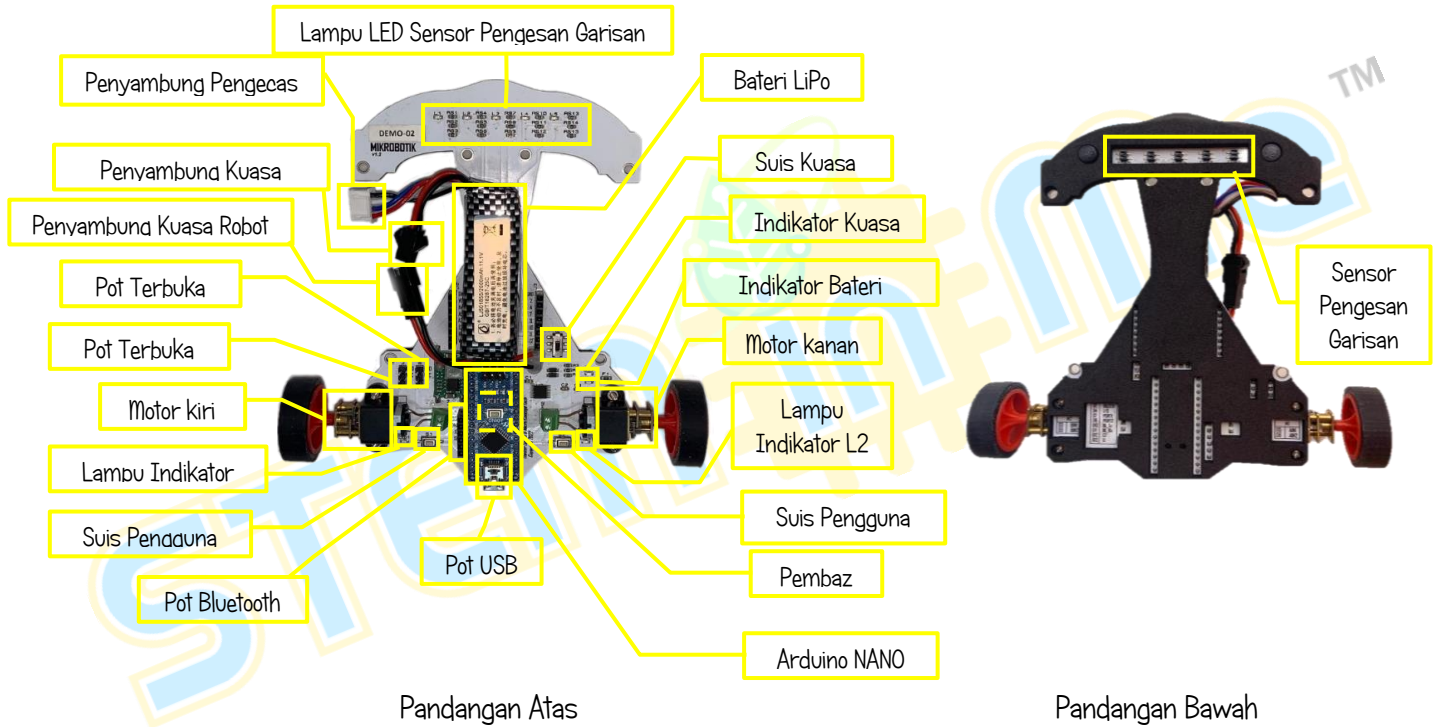


Gambar 2: Set Mikrobotik

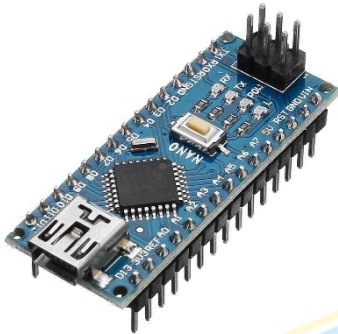


Gambar 3: Litar Mikrobotik

"Mikrobotik" Robot Berautonomi

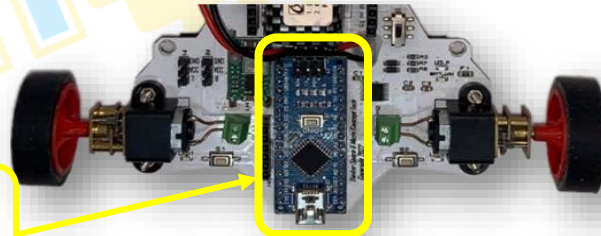


Mikropengawal Arduino Nano



Mikropengawal adalah peranti yang mengendalikan fungsi teras seperti mengawal penggunaan perkakasan elektronik lain yang bersambung dengannya. menganalisis data dan melaksanakan logik.

Mikrobotik menggunakan mikropengawal Arduino Nano yang berperanan sebagai otak untuk mengawal seluruh perkakasan dan pergerakan robot.



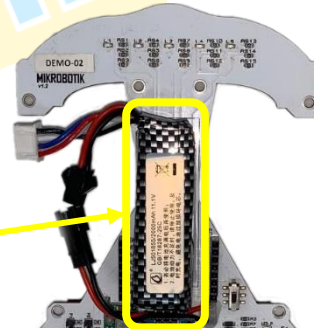
Mikropengawal Arduino Nano pada Mikrobotik

Bateri LiPo



Bateri polimer litium (LiPo) adalah baterai cas semula teknologi lithium-ion yang menggunakan elektrolit polimer berbanding elektrolit cecair. Ianya berfungsi dengan menyediakan tenaga spesifik yang lebih tinggi daripada jenis bateri litium yang lain dan digunakan dalam aplikasi dimana berat adalah ciri yang penting.

Mikrobotik menggunakan bateri LiPo 11.1V untuk memastikan pergerakan dengan tahap kelajuan maksima dapat dicapai.



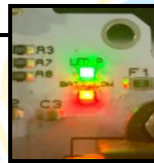
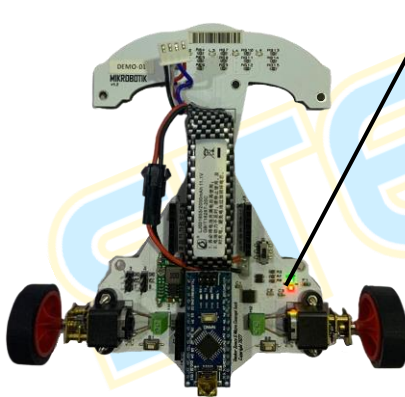
Bateri LiPo pada Mikrobotik

Indikator Bateri Rendah

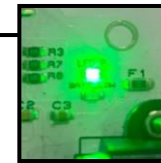
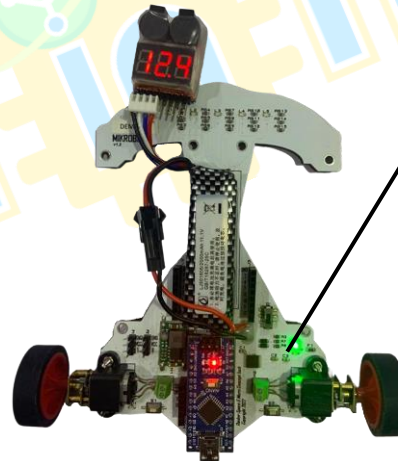
Indikator bateri rendah akan menyala warna merah.
Semakin rendah nilai voltan dalam bateri, semakin terang indikator menyala.
Voltan operasi minimum: 11.0 V (Indikator bateri rendah di kecerahan maksima)



Pengguna perlu berhenti menggunakan Mikrobotik dan perlu mengecas Mikrobotik apabila indikator bateri rendah kecerahan maksima.



Indikator bateri rendah ketika bateri rendah.



Indikator bateri rendah ketika bateri penuh.

Pemasangan perisian mBlock v5

Langkah 1

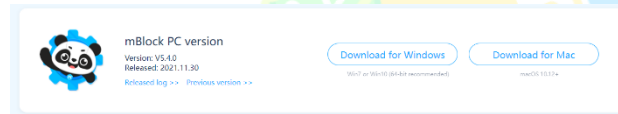
Perisian mBlock v5 boleh didapatkan daripada:

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



Langkah 2

Muat turun versi terkini mBlock v5 berdasarkan sistem pengendalian komputer.



Langkah 3

Klik mBlock v5 pada lokasi muat turun anda.



V5.4.0

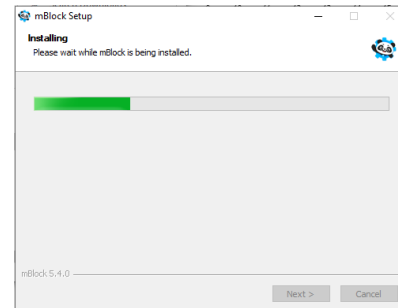
22/9/2022 4:33 PM

Application

251,230 KB

Langkah 4

Tunggu sehingga pemasangan mBlock v5 selesai.

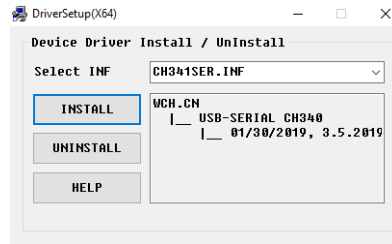


© 2023 Copyright micro concept tech

Version 1.3

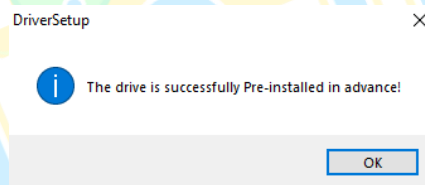
Langkah 5

Klik *INSTALL*:



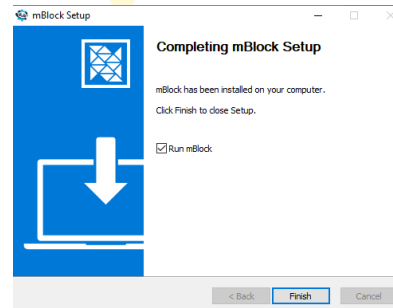
Langkah 6

Klik OK dan keluar



Langkah 7

Tandakan *Run mBlock*.
Klik *Finish*.



Cara untuk menambah Mikrobotik

Langkah 1

Perisian Mikrobotik boleh didapatkan daripada:

Link: <https://www.microconcept.com.my/stem-robotic/download/>

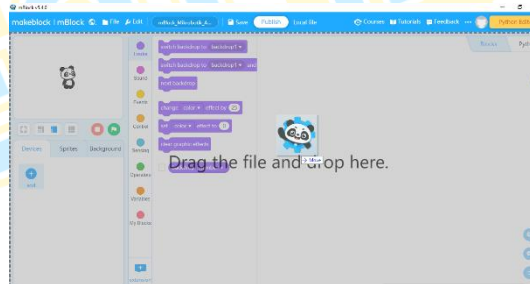
Langkah 2

Buka mBlock v5



Langkah 3

Pergi ke fail mikrobotik.mext dan seret ke dalam mBlock v5.



Langkah 4

Sekarang, anda boleh menikmati menggunakan mBlock v5!

Proses kalibrasi

Proses kalibrasi adalah proses penting untuk robot mengenali/pasti diantara garis putih dan garis hitam. Proses kalibrasi untuk robot Mikrobotik ini boleh dilakukan secara manual mahupun secara automatik. Proses ini perlu dilakukan setiap kali sebelum robot boleh bergerak secara berautonomi mengikuti garis dan menyelesaikan litar.

Susunan blok (Kalibrasi Automatik):

Langkah 1 Masukkan blok *When Mikrobotik Starts* dan gabungkan dengan blok *Prepare*.

When Mikrobotik starts

Robot Prepare

Langkah 2 Seterusnya, gabungkan blok ulang dengan blok *Tank Turn (Wheel Left-Forward, Speed-50, Wheel Right-Backward, Speed-50)* di bawah blok *Robot Prepare*.

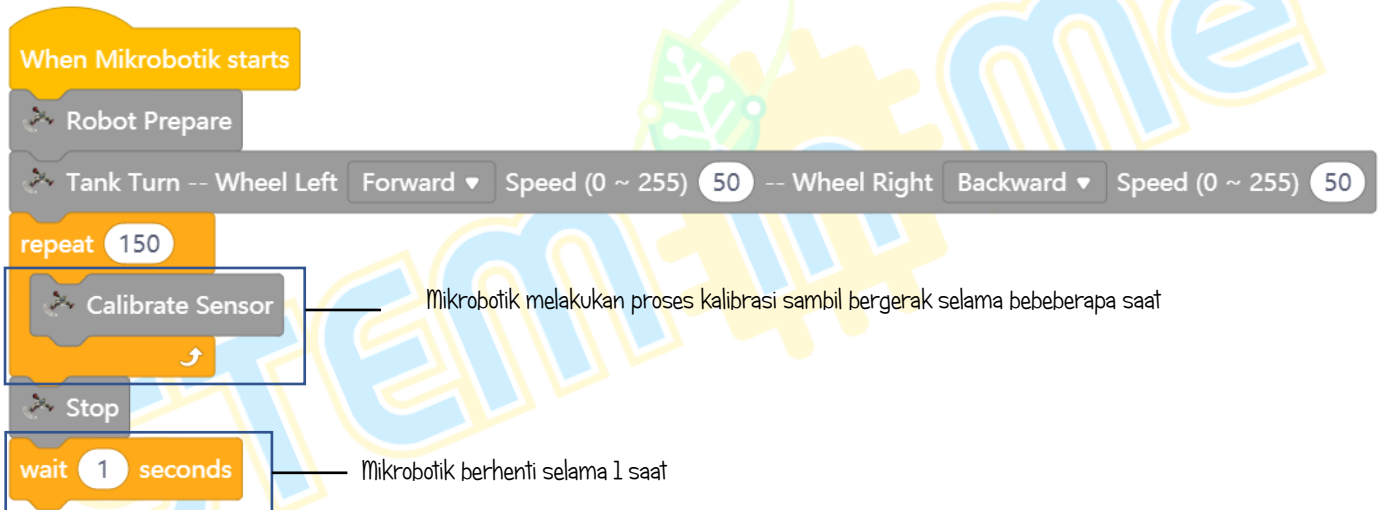


Langkah 3

Selepas itu, gabungkan blok *Repeat* dengan blok *Calibrate Sensor*.
Gabungkan blok ini dengan blok di Langkah 2.

Langkah 4

Kemudian, masukkan blok *Stop* dan blok *Wait (1 second)* di bawah blok *Repeat*.



The image shows a Scratch script for a Mikrobotik project. The script starts with a 'When Mikrobotik starts' event block. This is followed by a 'Robot Prepare' block. Then, a 'Tank Turn -- Wheel Left' block is set to 'Forward' with a speed of 50, and a 'Wheel Right' block is set to 'Backward' with a speed of 50. A 'repeat' block is set to 150 iterations. Inside the repeat loop, there is a 'Calibrate Sensor' block, a 'Stop' block, and a 'wait 1 seconds' block. Two callout boxes provide explanations: one for the 'Calibrate Sensor' block stating 'Mikrobotik melakukan proses kalibrasi sambil bergerak selama beberapa saat' (Mikrobotik performs the calibration process while moving for a few seconds), and another for the 'wait 1 seconds' block stating 'Mikrobotik berhenti selama 1 saat' (Mikrobotik stops for 1 second).

Langkah Proses Kalibrasi Automatik

Langkah 1

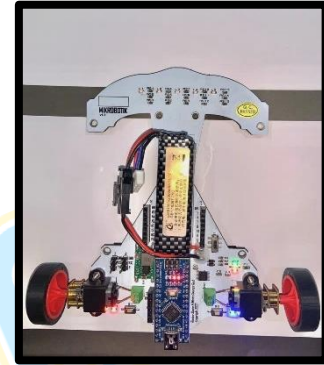
Letakkan Mikrobotik di atas litar.

Pastikan semua pengesan berlabel TR1 (LED L1) hingga ke TR5 (LED L5) berada di atas garisan hitam.

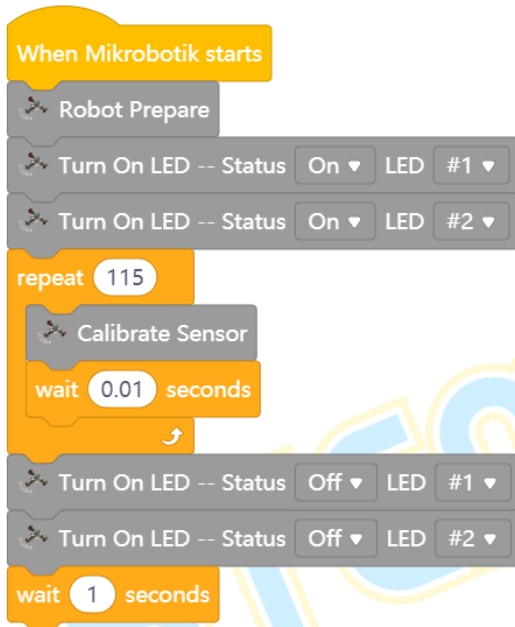
Langkah 2

Hidupkan suis Mikrobotik.

Lampu LED1 berwarna merah dan LED2 berwarna biru akan menyala. Robot akan berpusing secara automatik untuk menjalankan proses kalibrasi.



Susunan blok (Kalibrasi Manual):



```
When Mikrobotik starts
  Robot Prepare
  Turn On LED -- Status On LED #1
  Turn On LED -- Status On LED #2
  repeat 115
    Calibrate Sensor
    wait 0.01 seconds
  Turn On LED -- Status Off LED #1
  Turn On LED -- Status Off LED #2
  wait 1 seconds
```

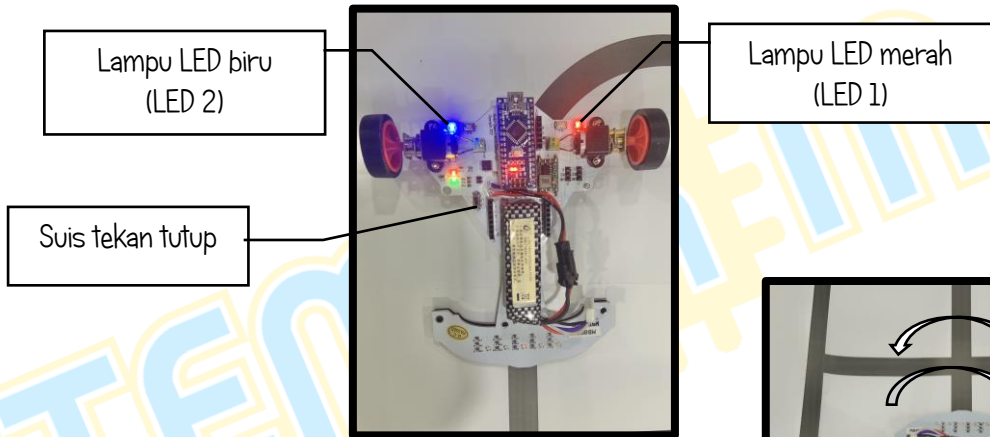
The image shows a Scratch script for manual calibration. It starts with a 'When Mikrobotik starts' event block. The script then performs the following steps: 'Robot Prepare', 'Turn On LED -- Status On LED #1', and 'Turn On LED -- Status On LED #2'. A 'repeat' loop with a count of 115 contains a 'Calibrate Sensor' block followed by a 'wait 0.01 seconds' block. After the loop, the script turns the LEDs off with 'Turn On LED -- Status Off LED #1' and 'Turn On LED -- Status Off LED #2', and finally waits for 1 second.

Langkah Proses Kalibrasi Manual

Langkah 1

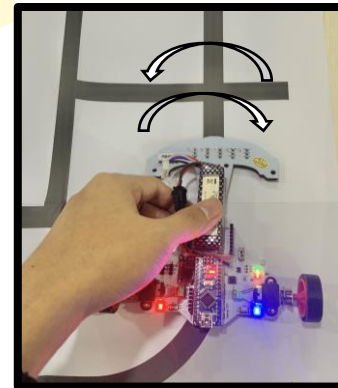
Hidupkan suis Mikrobotik.

Lampu LED1 berwarna merah dan LED2 berwarna biru akan menyala.



Langkah 2

Gerakkan semua pengesan bermula daripada pengesan berlabel TR1 (LED L1) hingga ke TR5 (LED L5) dan kembali semula ke TR1.



Langkah 3

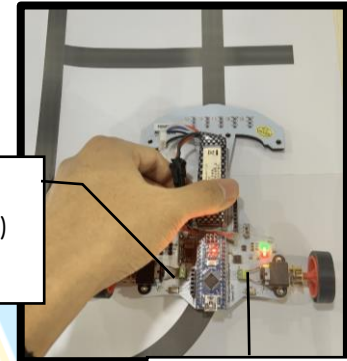
Ulangi pergerakan di Langkah 2 sehingga lampu LED1 dan LED2 terpadam



Pastikan semua pengesan dapat mengesan garisan hitam dengan cara LED pada pengesan tersebut akan menyala jika pengesan tersebut mengesan garisan hitam. Contohnya LED L1 akan menyala jika pengesan TRI mengesan garisan hitam pada litar.

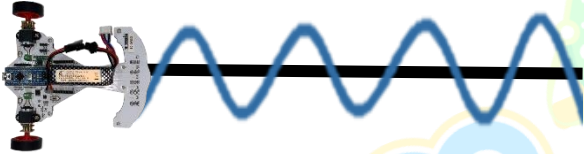


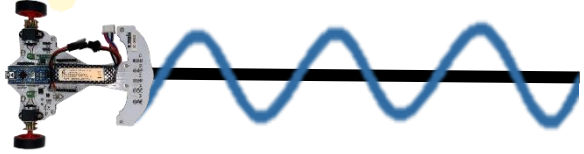
Lampu LED merah (LED 1) terpadam

Lampu LED biru (LED 2) terpadam

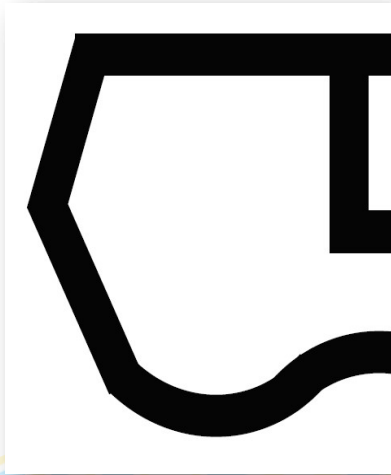


Algoritma PID Robot Berautonomi

Algoritma PID adalah strategi kawalan robot berautonomi yang sesuai untuk membantu menentukan arah kemudi dan kelajuan robot yang bergerak secara automatik mengikut garisan. Algoritma PID akan memastikan robot tidak tersasar dari litar ketika membelok dan bergerak lurus mengikut garisan.

<p style="text-align: center;">K_p Nilai K_p Tinggi</p>  <p style="text-align: center;">Kemudi terlalu bersungguh</p>	<p style="text-align: center;">K_d Nilai K_d Tinggi</p>  <p style="text-align: center;">Kemudi balas awal</p>
<p style="text-align: center;">Nilai K_p Rendah</p>  <p style="text-align: center;">Kemudi terlalu lemah</p>	<p style="text-align: center;">Nilai K_d Rendah</p>  <p style="text-align: center;">Kemudi balas lewat</p>

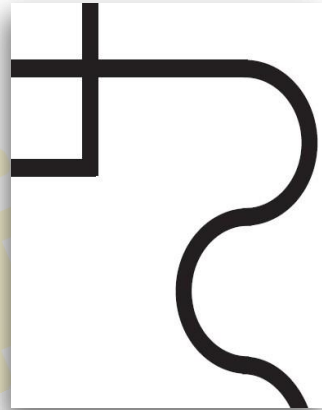
Apa jenis-jenis Litar?



Garisan Hitam
(Anggaran 20mm)

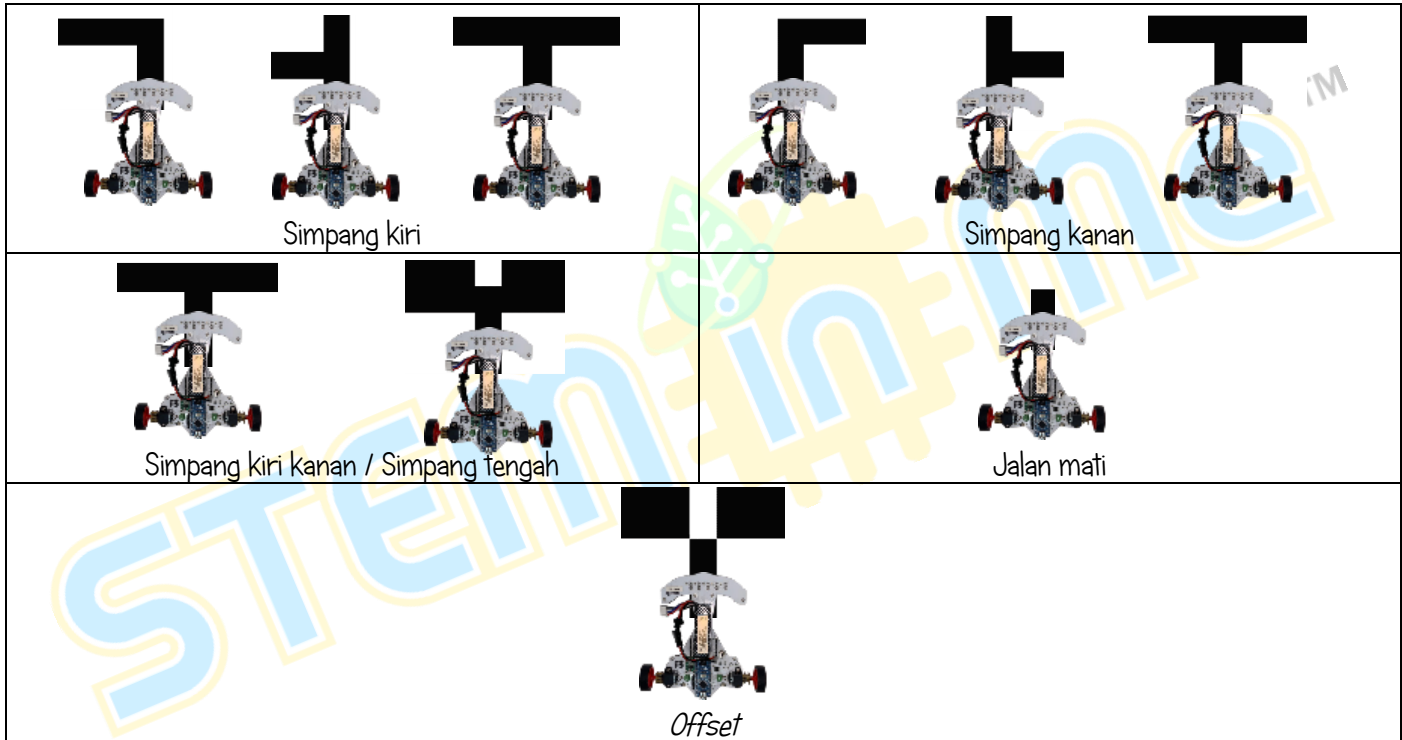


Garisan Putih
(Anggaran 20mm)



Garisan Hitam Nipis
(Anggaran 10mm)

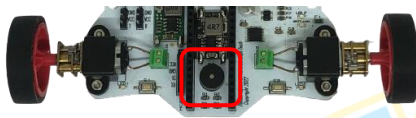
Jenis-jenis Persimpangan



Objektif 1: Vroom Vroom

Robot akan menggunakan pembaz untuk menghasilkan bunyi ringkas. Ia hanya boleh menghasilkan satu nada pada satu masa. Kod blok ini boleh digunakan untuk menghasilkan nada yang berbeza bagi mencipta satu corak bunyi yang menarik.

Pengenalan Mudah Pembaz



Pembaz ialah sejenis peranti suara yang menukar model audio kepada isyarat bunyi. Ia biasanya digunakan untuk penggera.

Langkah-langkah susunan blok

Langkah 1

Gabungkan blok *When Mikrobotik starts* dengan blok *Robot Prepare*.

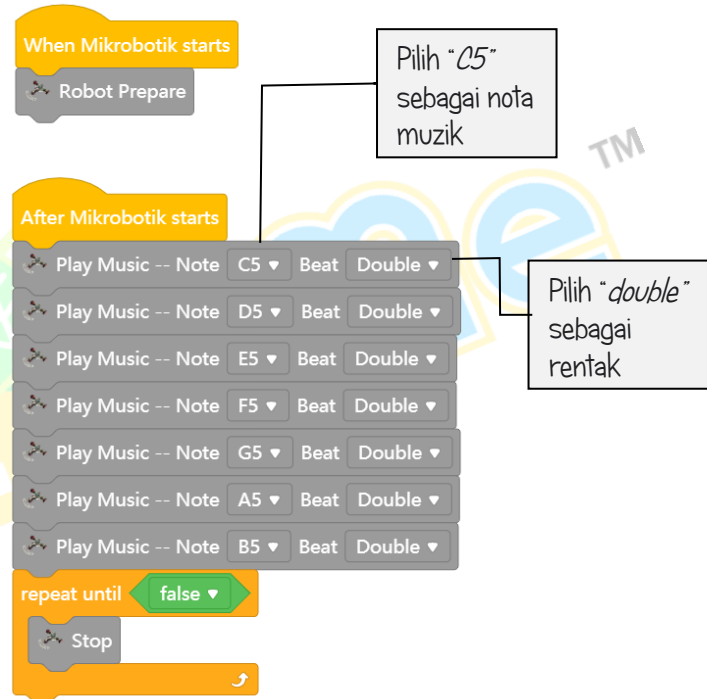
When Mikrobotik starts

Robot Prepare

Blok ini adalah untuk menyediakan robot dengan *library* tertentu dan untuk mengkonfigurasi nombor pin dan nombor pot keluar masuk untuk setiap sensor dan keluaran yang dipasang pada robot.

Langkah 2

Seterusnya, gabungkan blok *After Mikrobotik starts* dengan blok *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)*



Langkah 3

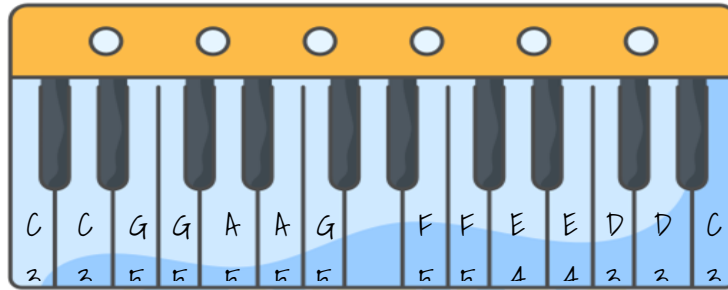
Seterusnya, gabungkan blok *repeat until (false)* dengan blok *stop*. Gabungkan blok tersebut dengan blok di Langkah 2.

Langkah 4

Setelah program dimuat naik, robot akan menghasilkan bunyi atau nada yang anda telah masukkan..

Cabaran !!

Dalam cabaran ini, anda perlu memasukkan not muzik yang disediakan dan cuba untuk meneka nama muzik yang dihasilkan.



Objektif 2: Tolong Hidupkan Lampu!

Diod Pemancar Cahaya (LED) pada robot digunakan sebagai penanda. LED pada robot boleh dilihat pada indikator kuasa, indikator bateri rendah, lampu indicator L1 and L2, Arduino NANO dan LED sensor pengesanan garisan.

Pengenalan Mudah Diod Pemancar Cahaya (LED)



Diod Pemancar Cahaya atau LED berfungsi menukarkan arus elektrik kepada cahaya dan memancarkan cahaya. Digunakan sebagai aplikasi bagi indikator dan sumber cahaya.

Langkah-langkah susunan blok

Langkah 1

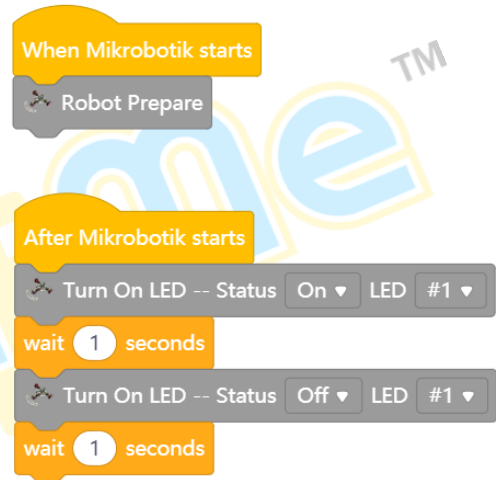
Gabungkan blok *When Mikrobotik starts* dengan blok *Robot Prepare*.

When Mikrobotik starts

Robot Prepare

Langkah 2

Gabungkan blok *After Mikrobotik starts* dengan blok *Turn On LED* dengan pilihan *Status On* dan LED #1 dan blok *wait 1 second*. Letakkan blok tersebut di bawah blok di Langkah 1. Program ini akan menyalakan lampu LED.



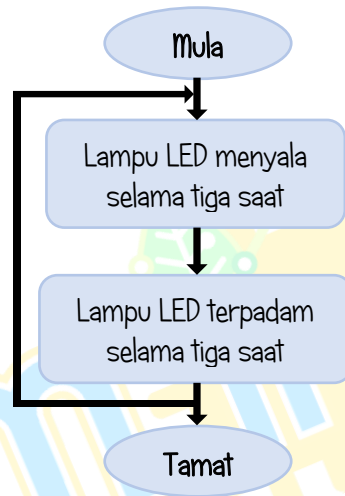
Langkah 3

Tambah satu lagi blok *Turn On LED* dengan pilihan *Status Off* dan LED#1 dengan blok *wait 1 second* dan gabungkan dengan blok di Langkah 2 untuk memadamkan lampu LED.

Langkah 4

Yang terakhir, muat naik program tersebut. Setelah program dimuat naik, LED 1 akan menyala dalam masa satu saat dan akan terpadam dalam masa satu saat. Program ini akan terus berjalan sehingga robot dimatikan oleh pengguna.

Cabaran!!

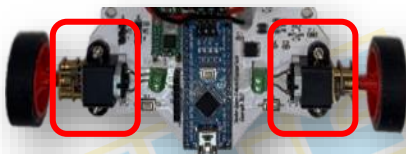


Program di atas akan menyala selama tiga saat dan akan terpadam selama tiga saat. Program ini akan berterusan sehingga Mikrobotik dimatikan.

Objektif 3: Mulakan Pengembaraan Kita (Pergerakan Bebas)

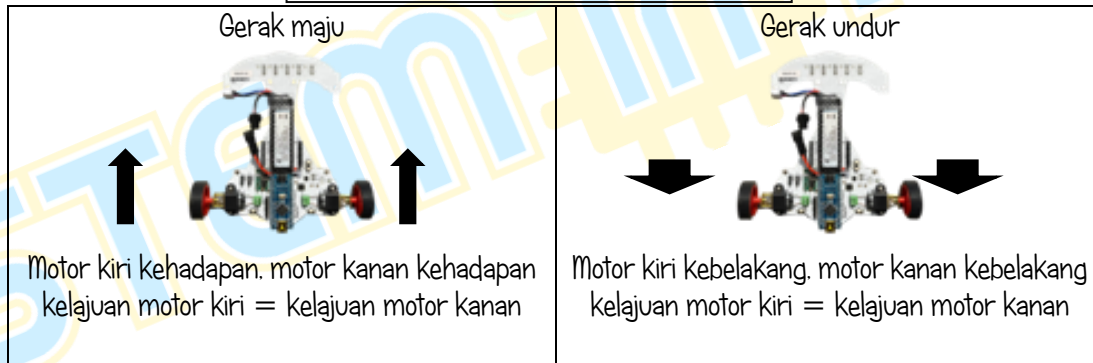
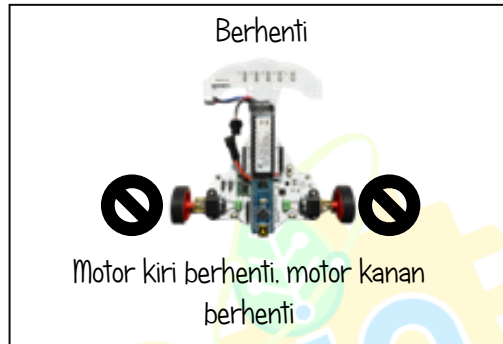
Robot digerakkan menggunakan kod blok "tank turn" untuk bergerak tanpa mengikuti garisan. Kod blok ini sesuai digunakan untuk menyelesaikan litar labirin (*maze*). Robot akan bergerak bergantung kepada kelajuan serta arah motor kiri dan kanan yang ditetapkan.




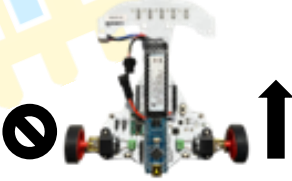
Pengenalan kepada Motor

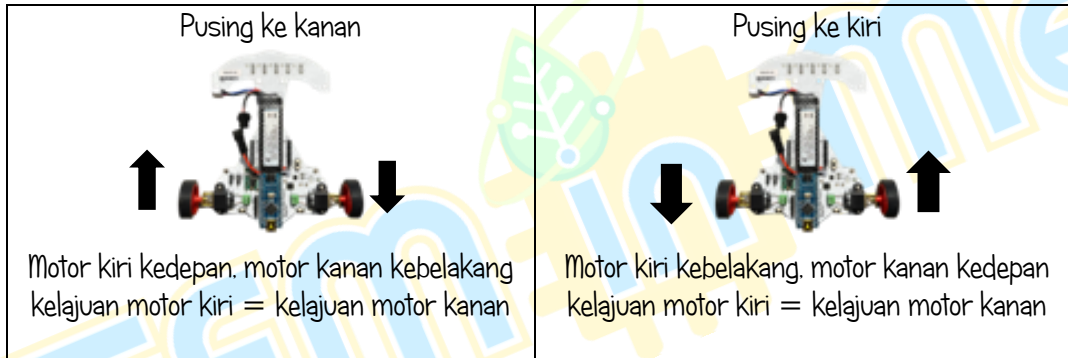


Terdapat 2 motor pada Mikrobotik boleh dikawal secara berasingan. boleh berputar mengikut putaran arah jam dan lawan jam secara berterusan. Motor ini juga boleh digunakan untuk menggerakkan atau memacu projek. Kelajuan dan jangka masa juga boleh ditetapkan.

Pengenalan Mudah Pergerakan Asas Robot



<p>Belok ke kanan</p>  <p>Motor kiri kedepan. motor kanan ke hadapan Kelajuan motor kiri > Kelajuan motor kanan</p>	<p>Belok ke kiri</p>  <p>Motor kiri ke hadapan. motor kanan ke hadapan Kelajuan motor kiri < Kelajuan motor kanan</p>
<p>Belok tajam ke kanan</p>  <p>Motor kiri ke hadapan. motor kanan berhenti</p>	<p>Belok tajam ke kiri</p>  <p>Motor kiri berhenti. motor kanan ke hadapan</p>



Langkah-langkah susunan blok

i) Maju

Langkah 1

Gabungkan blok *When Mikrobotik starts* dengan blok *Robot Prepare*.

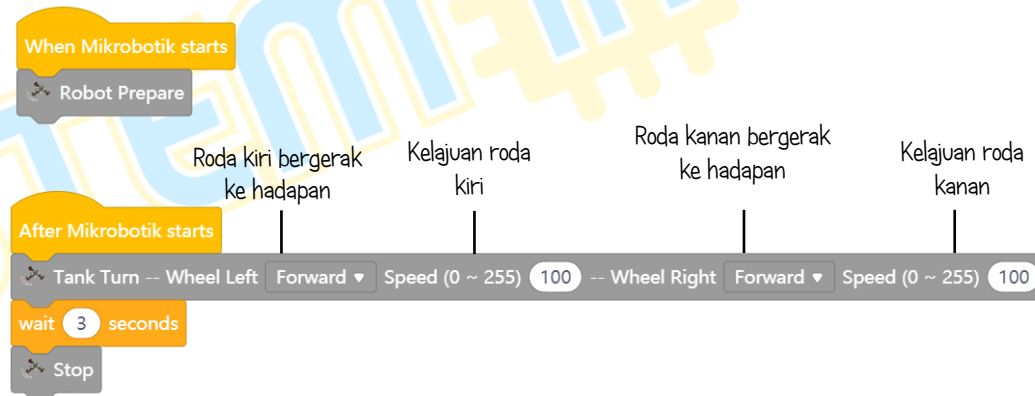
When Mikrobotik starts

Robot Prepare

Langkah 2

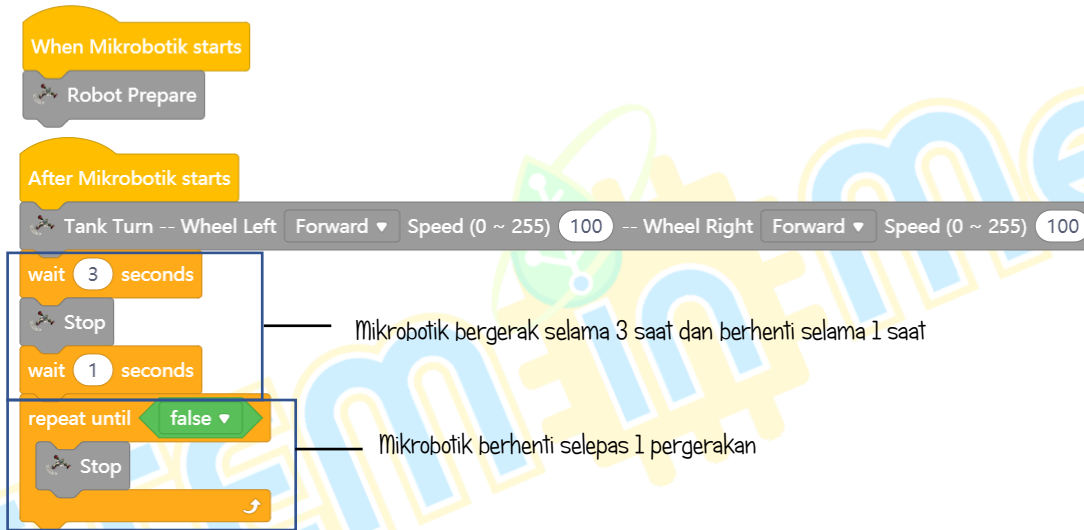
Gabungkan blok *After Mikrobotik starts* dengan blok *Tank Turn (Wheel Left -Forward, Speed-100, Wheel Right-Forward, Speed-100)*, blok *wait (3 seconds)* dan blok *stop*. Letakkan blok tersebut di bawah blok di Langkah 1.

Motor kiri dan kanan akan bergerak ke hadapan dengan kelajuan yang sama.



Langkah 3

Akhirnya, gabungkan blok *wait*, blok *repeat until (false)*, dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.



Langkah 4

Selepas memuat naik program, Mikrobotik akan maju ke hadapan selama 3 saat dan berhenti.



Langkah untuk undur ke belakang adalah serupa dengan langkah untuk maju ke hadapan. Anda hanya perlu menukar arah wheel left kepada backward dan arah wheel right kepada backward.

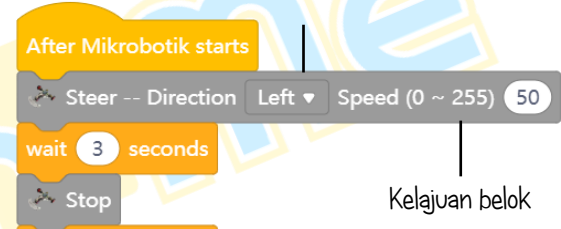
ii) Belok ke kiri

Langkah 1 Gabungkan blok *When Mikrobotik starts* dengan blok *Robot Prepare*.



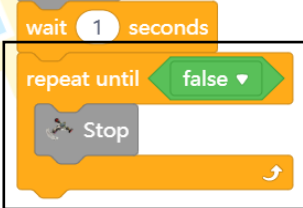
Mikrobotik belok ke kiri

Langkah 2 Gabungkan blok *After Mikrobotik starts* dengan blok *Steer (Direction -Left, Speed-50)*. Kemudian, tambahkan blok *wait (3 seconds)* dan blok *stop*. Motor kiri akan berhenti dan motor kanan akan bergerak ke hadapan dengan kelajuan yang ditetapkan.



Kelajuan belok

Langkah 3 Akhirnya, gabungkan blok *wait (1 second), repeat until (false)* dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.



Mikrobotik berhenti selepas 1 pusingan

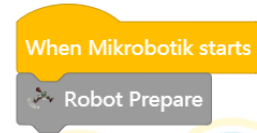
Langkah 4 Selepas memuat naik program. Mikrobotik belok ke kiri selama 3 saat dan berhenti.



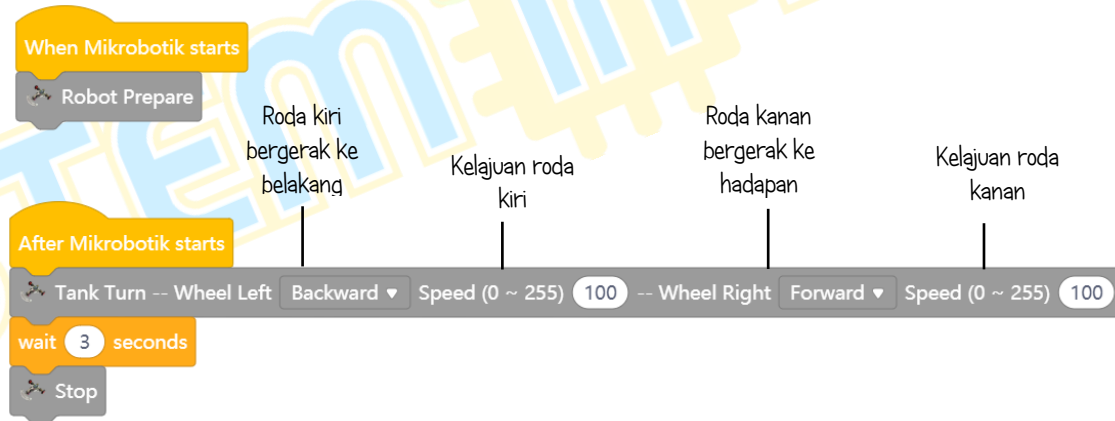
Langkah untuk belok ke kanan adalah serupa dengan langkah untuk belok ke kiri. Anda hanya perlu menukar *Direction* kepada right.

iii) Pusing ke kiri

Langkah 1 Gabungkan blok *When Mikrobotik starts* dengan blok *Robot Prepare*.



Langkah 2 Gabungkan blok *After Mikrobotik starts* dengan blok *Tank Turn (Wheel Left-Backward, Speed-100, Wheel Right-Forward, Speed-100)*, blok *wait (3 seconds)* dan blok *stop*. Letakkan blok tersebut di bawah blok di Langkah 1. Motor kiri akan bergerak ke belakang dan motor kanan akan bergerak ke hadapan dengan kelajuan yang sama.



Langkah 3

Akhirnya, gabungkan blok *wait (1 second)*, *repeat until (false)* dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.

The image shows a Scratch script for Mikrobotik. It starts with a yellow 'When Mikrobotik starts' block containing a grey 'Robot Prepare' block. Below this is an orange 'After Mikrobotik starts' block containing a grey 'Tank Turn -- Wheel Left' block with 'Backward' selected, a 'Speed (0 ~ 255)' block with '100', a grey 'Wheel Right' block with 'Forward' selected, and another 'Speed (0 ~ 255)' block with '100'. Below the tank turn block is a blue 'wait 3 seconds' block, followed by a grey 'Stop' block. Below the stop block is another blue 'wait 1 seconds' block, followed by an orange 'repeat until false' block containing a grey 'Stop' block. Two lines with arrows point from the text to the script: one points to the 'wait 3 seconds' and 'Stop' blocks, and the other points to the 'repeat until false' and 'Stop' blocks.

Mikrobotik bergerak selama 3 saat dan berhenti selama 1 saat

Mikrobotik berhenti selepas 1 pusingan

Langkah 4

Selepas memuat naik program. Mikrobotik pusing ke kiri selama 3 saat dan berhenti.



Langkah untuk pusingan ke kanan adalah serupa dengan langkah untuk pusingan ke kiri. Anda hanya perlu menukar arah wheel left kepada backward dan arah wheel right kepada forward.

Objektif 4: Ayuh Ikuti Garisan Itu

Robot akan bergerak secara berautonomi mengikuti garisan (Hitam atau Putih) secara berterusan. Robot akan sentiasa bergerak walaupun ia bertemu dengan simpang kiri atau simpang kanan.

Pengenalan mudah Pengesan Garisan



Terdapat 5 pengesan garisan akan memancarkan cahaya infrared dan mengesan permukaan berwarna hitam atau putih. Nilai bacaan analog akan tinggi jika permukaan hitam dikesan manakal bacaan analog akan rendah apabila permukaan putih dikesan.

Pengenalan *Line Tracer Time* dan Mekanismanya

Line Tracer Time digunakan untuk Mikrobotik bergerak secara berautonomi mengikuti garisan sama ada Hitam atau Putih sehingga mencapai tempoh masa maksimum (dalam ms).

Apabila Mikrobotik mencapai tempoh masa maksimum, Mikrobotik akan berhenti. Mikrobotik akan bergerak secara berterusan tanpa melalui persimpangan kiri, persimpangan kanan dan persimpangan tengah.

Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.

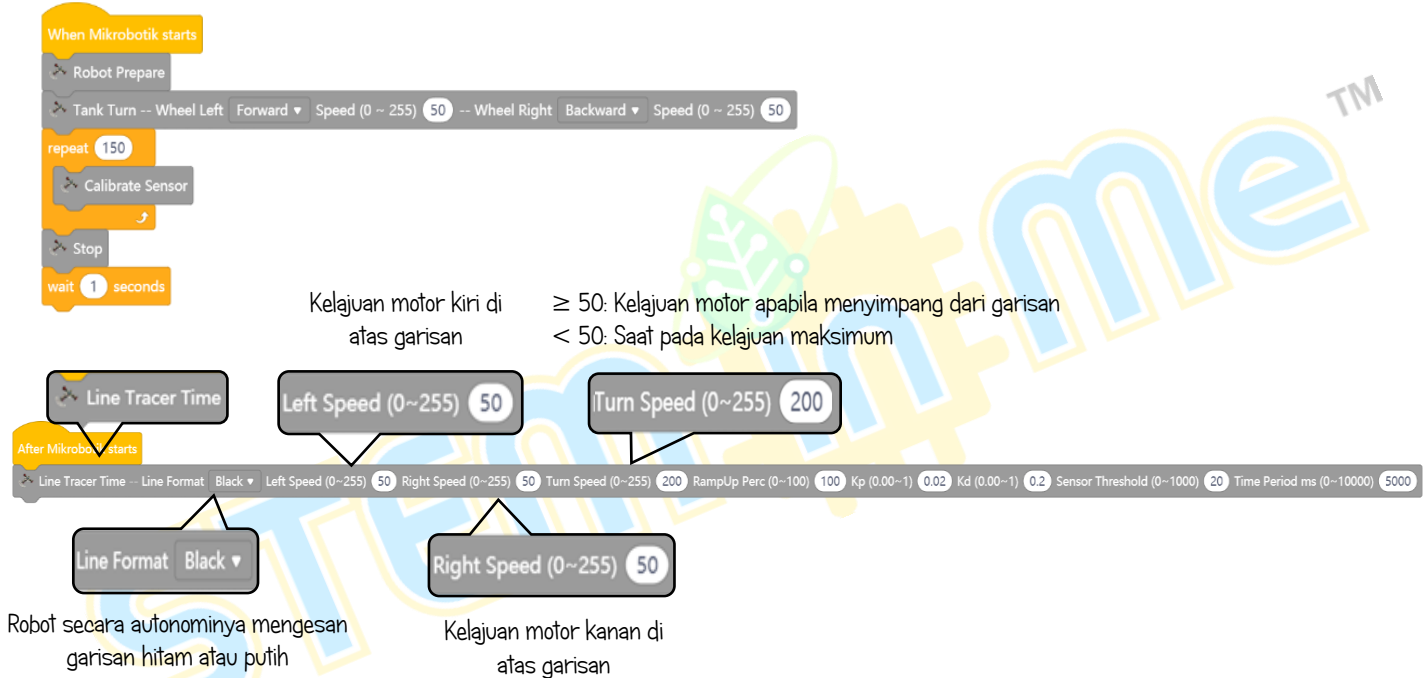
The image shows a Scratch script for a robot's automatic calibration process. The script starts with a yellow 'When Mikrobotik starts' block. This is followed by a grey 'Robot Prepare' block. Then, a grey 'Tank Turn -- Wheel Left' block is set to 'Forward' with a speed of 50, and the 'Wheel Right' block is set to 'Backward' with a speed of 50. Below this is an orange 'repeat' block set to 150 iterations, containing a grey 'Calibrate Sensor' block. This is followed by a grey 'Stop' block and an orange 'wait' block set to 1 second. Two blue boxes highlight the 'repeat' and 'wait' blocks, with lines pointing to explanatory text.

Mikrobotik melakukan proses kalibrasi sambil bergerak selama beberapa saat

Mikrobotik berhenti selama 1 saat

Langkah 2

Seterusnya, masukkan blok *After Mikrobotik Starts* dan gabungkannya dengan blok *Line Tracer Time*.



Sambungan

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

Mengawal nilai Kp (lebih tinggi nilai bermaksud lebih sensitif)

Kepekaan mengesan garisan (lebih tinggi nilai bermaksud lebih sensitif)

RampUp Perc (0~100) 100

Peratusan pecutan

Kd (0.00~1) 0.2

Mengawal nilai Kd (Kd mesti lebih > Kp)

Sensor Threshold (0~1000) 20

Masa maksimum (ms) untuk robot bergerak ke hadapan

Time Period ms (0~10000) 5000

Langkah 3

Akhirnya, gabungkan blok *repeat until (false)* dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.

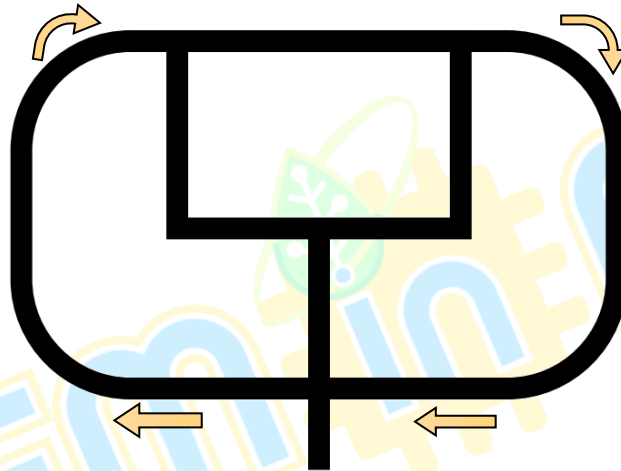


Langkah 4

Selepas memuat naik kod, Mikrobotik akan mula bergerak ke hadapan buat sementara waktu. Lakukan proses kalibrasi pada pengesanan garisan. Selepas itu, Mikrobotik akan mengikuti garisan sama ada Hitam atau Putih sehingga mencapai tempoh masa maksimum (dalam ms).

Cabaran!!

Gunakan *Line Tracer Time* untuk menyelesaikan litar di bawah.



Mula

Tamat

Objektif 5: Apa Yang Perlu Dilakukan Ketika Di Persimpangan?

Robot akan bergerak secara berautonomi dan membuat keputusan sama ada perlu belok kiri, belok kanan ataupun berhenti di persimpangan. Teknik yang digunakan adalah dengan menggunakan pergerakan belok (*Steer Turn Method*).

Pengenalan *Path Finder* dan Mekanismanya.

Path Finder digunakan untuk menggerakkan Mikrobotik bergerak secara berautonomi mengikuti garisan putih atau hitam sehingga Mikrobotik menemui persimpangan (kanan atau kiri atau tengah atau jalan mati atau *offset*).

Di persimpangan, Mikrobotik akan bertindak untuk belok (kiri atau kanan atau berhenti) untuk tempoh yang ditetapkan atau sehingga robot menjumpai garisan seterusnya dan akan berhenti.

Robot akan belok dengan menggunakan pergerakan belok *Steer Turn Method*.

Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



Langkah 2

Akhirnya, gabungkan blok *After Mikrobotik Starts* dan blok *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 secara autonominya mengesan garisan hitam atau putih

Aksi robot apabila menemui persimpangan

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

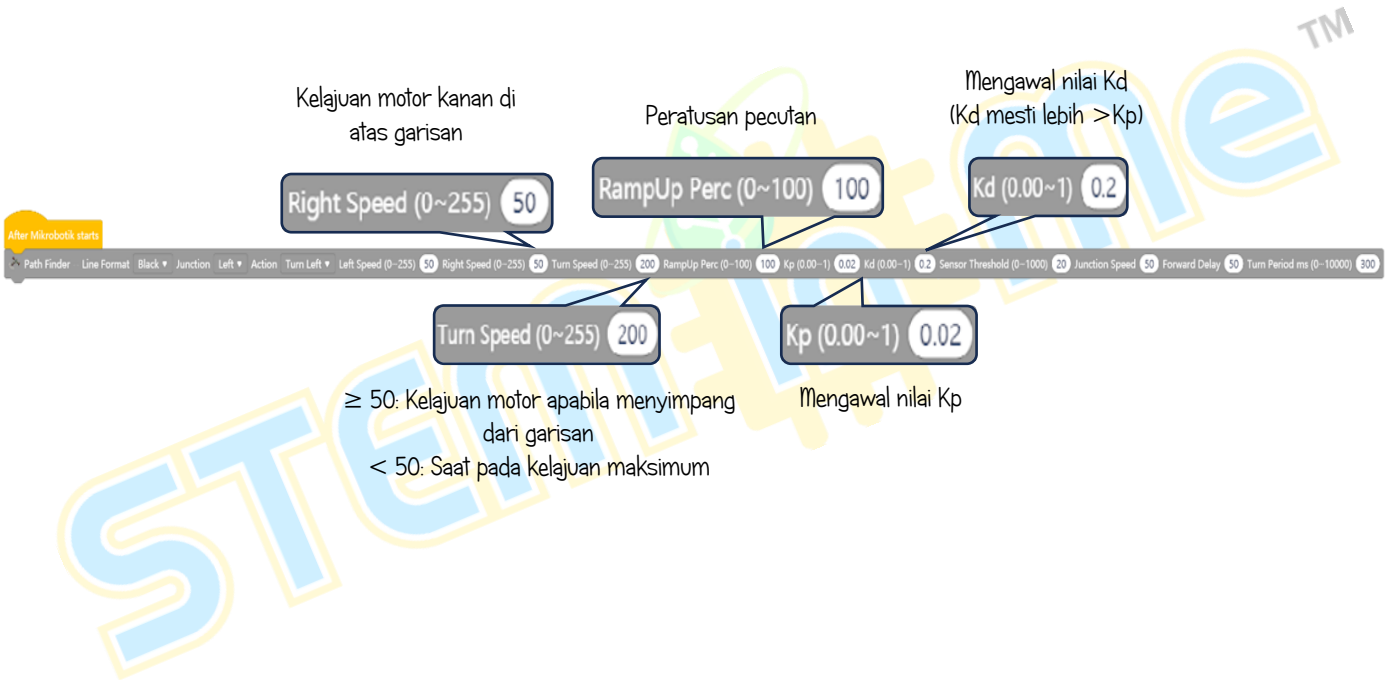
Callouts:

- Line Format Black
- Action Turn Left
- Path Finder
- Junction Left
- Left Speed (0~255) 50

Robot menjejak ke hadapan sehingga menemui persimpangan

Kelajuan motor kiri di atas garisan

Sambungan



Sambungan

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

Kelajuan membelok di persimpangan (lebih tinggi nilai, lebih tajam belokan)

Masa untuk robot terus bergerak ke hadapan sebelum membelok

Junction Speed 50

Forward Delay 50

Sensor Threshold (0~1000) 20

Turn Period ms (0~10000) 300

Kepekaan garisan (lebih tinggi nilai bermaksud lebih sensitif)

Masa tepat (ms) robot membelok atau sehingga menjumpai garisan (n)

Langkah 3

Akhirnya, gabungkan blok *repeat until (false)* dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.

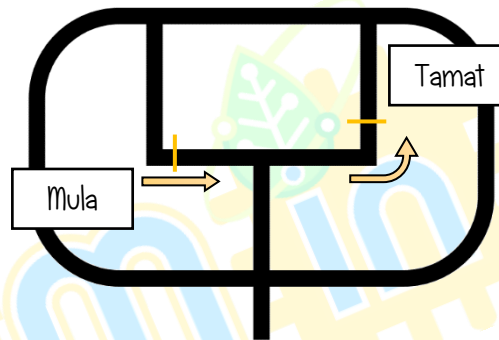


Langkah 4

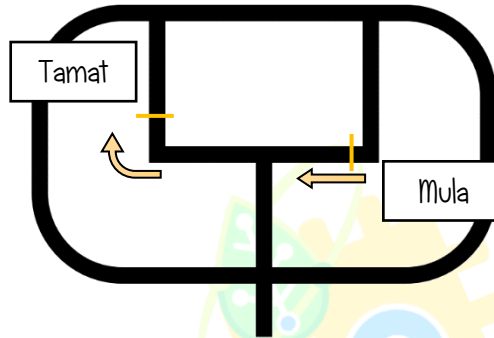
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi. Selepas itu, Mikrobotik akan mengikuti garisan hitam dan jika robot menemui persimpangan kiri, Mikrobotik akan bergerak ke hadapan dan kemudian belok memasuki simpang kiri sehingga Mikrobotik menemui garisan lain.

Cabaran!!

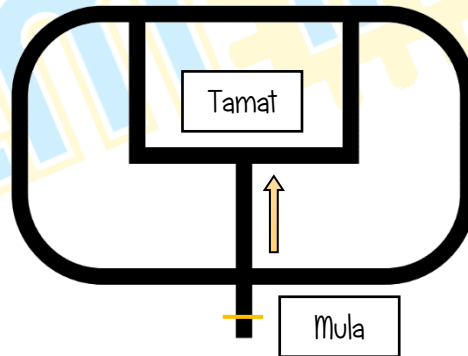
- i) *Path Finder* persimpangan kiri, belok di persimpangan kiri



ii) *Path Finder* persimpangan kanan. belok di persimpangan kanan



iii) *Path Finder* persimpangan tengah. berhenti



Objektif 6: Apa Lagi Boleh Dilakukan Ketika Di Persimpangan?

Robot akan bergerak secara berautonomi dan membuat keputusan sama ada perlu pusing kiri, pusing kanan ataupun berhenti di persimpangan. Teknik yang digunakan adalah dengan menggunakan pergerakan pusing (*Tank Turn Method*).

Pengenalan *Path Finder Tank* dan Mekanismanya.

Mikrobotik bergerak secara berautonomi mengikut garisan (Hitam atau Putih atau Hitam Nipis atau Putih Nipis) hingga menemui persimpangan (Kiri atau Kanan atau Tengah atau Jalan Mati atau *Offset*).

Di persimpangan, Mikrobotik akan bertindak (Pusing ke kiri atau Pusing ke kanan atau berhenti) untuk sekurang-kurangnya Durasi Minimum Pusingan (*Min Turn Period*) dan berterusan berpusing sehingga mengesan garisan dan berhenti.

Mikrobotik akan berpusing menggunakan pergerakan pusing (*Tank Turn Method*).

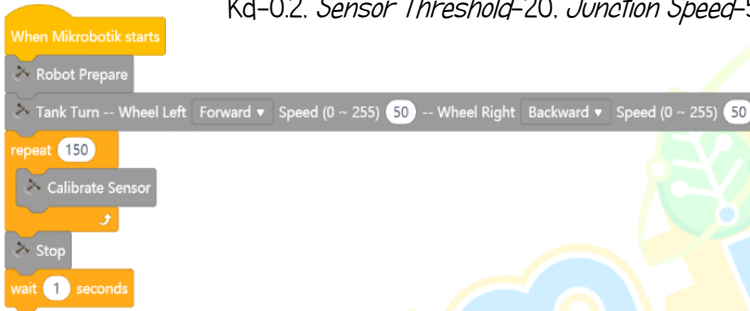
Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



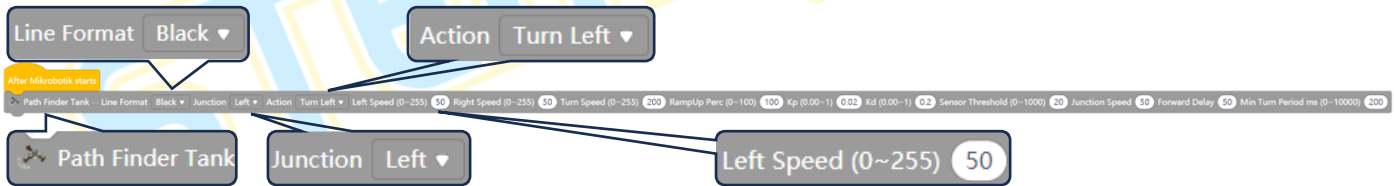
Langkah 2

Gabungkan blok *After Mikrobotik starts* dengan blok *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).



Robot secara autonominya mengesan garisan hitam atau putih

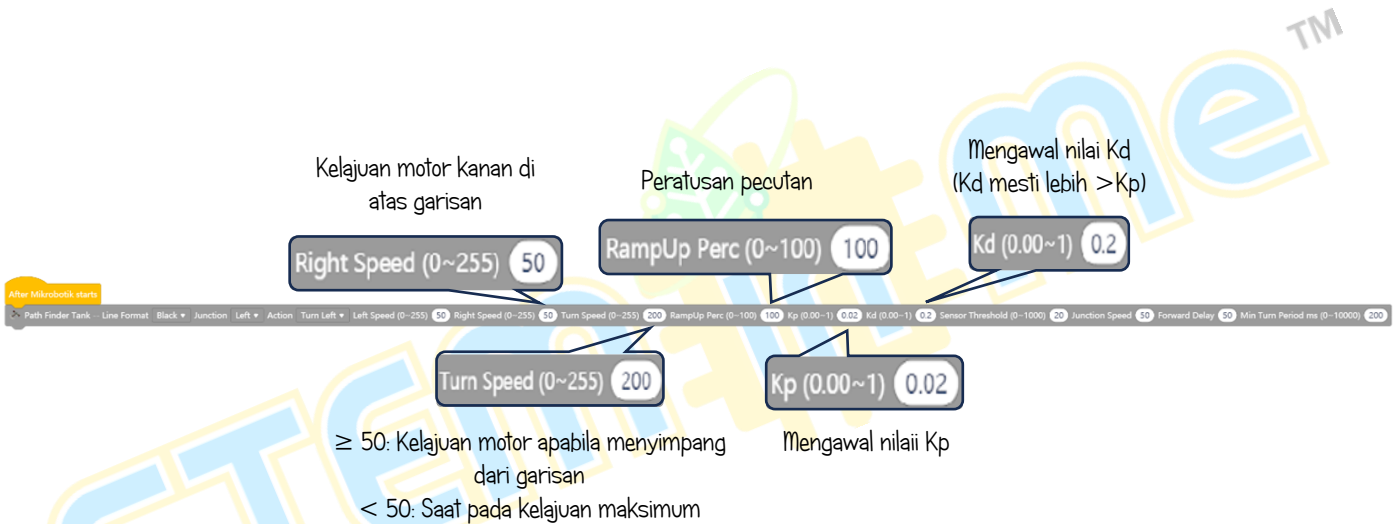
Aksi robot apabila menemui persimpangan



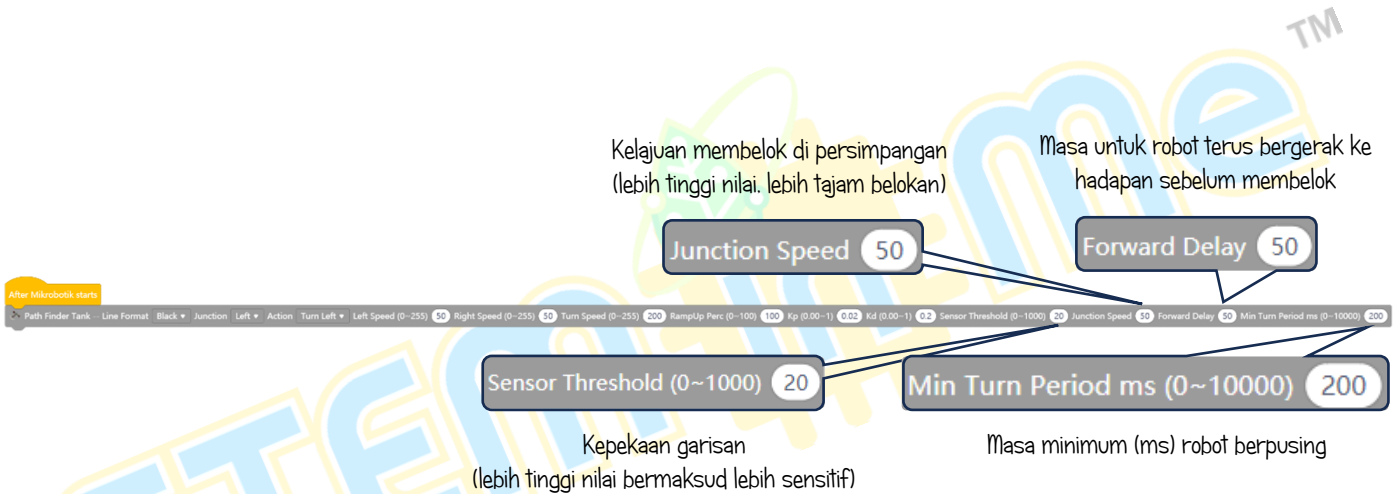
Robot menjejak ke hadapan sehingga menemui persimpangan

Kelajuan motor kiri di atas garisan

Sambungan



Sambungan



Langkah 3

Akhirnya, gabungkan blok *repeat until (false)* dan blok *stop*. Dan gabungkan blok tersebut dengan blok di Langkah 2.

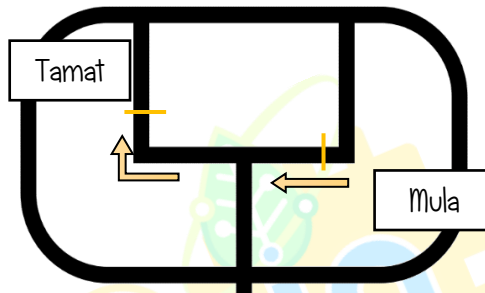


Langkah 4

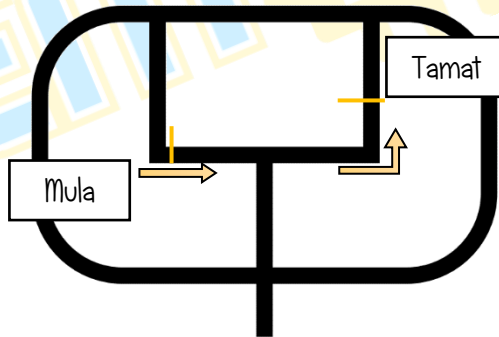
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi. Selepas itu, Mikrobotik akan mengikuti garisan hitam dan jika robot menemui persimpangan kiri, Mikrobotik akan bergerak ke hadapan dan kemudian pusing untuk sekurang-kurangnya Durasi Minimum Pusingan (*Min Turn Period*) dan berterusan berpusing sehingga mengesan garisan dan berhenti.

Cabaran!!

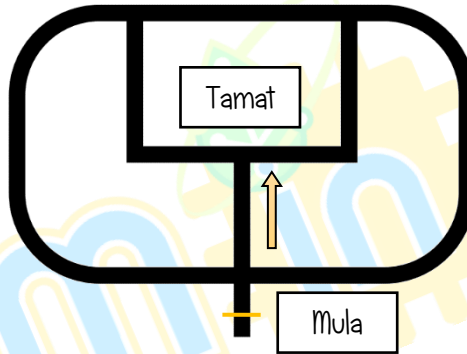
- i) *Path Finder Tank* persimpangan kanan, pusing di persimpangan kanan



- ii) *Path Finder Tank* persimpangan kiri, pusing di persimpangan kiri



iii) *Path Finder Tank* persimpangan tengah. berhenti.



Objektif 7: Salah Jalan? Buat Pusingan-U

Mikrobotik boleh membuat pusingan-U pada garisan yang dilaluinya pada paksi robot dan berpusing mengikut arah kiri atau kanan selama Durasi Pusingan Minimum (Min Turn Period) dan bersambung sehingga bertemu garisan (Hitam atau Putih)

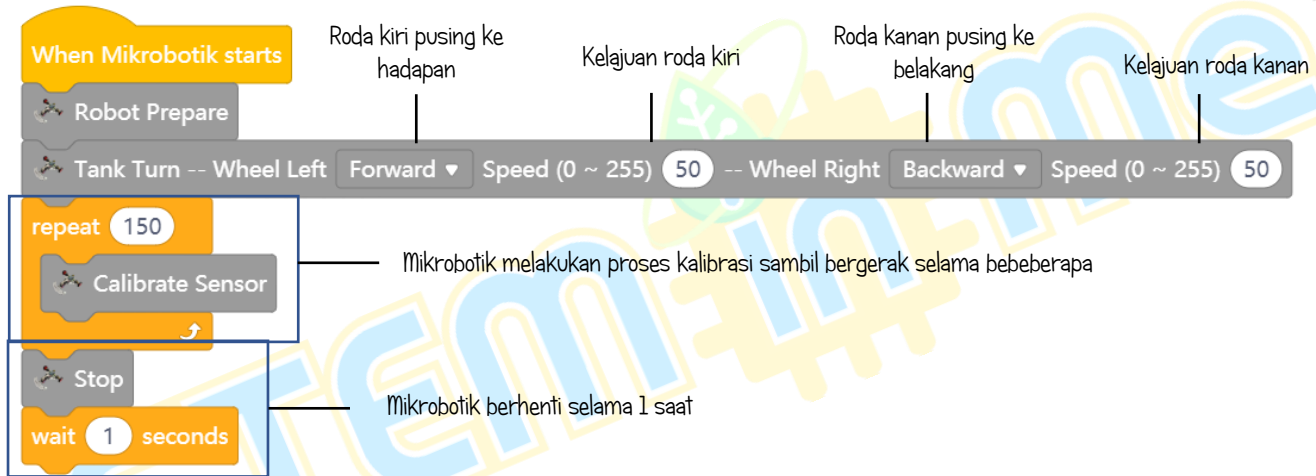
Pengenalan *Turn at Centre* dan Mekanismanya

Mikrobotik akan membuat pergerakan pusing (*tank turn*) ke arah (kiri atau kanan) untuk Durasi Pusingan Minimum (*Min Turn Period*) sehingga robot menemui garisan dan akhirnya berhenti.

Teknik ini berguna untuk membuat pusingan-U.

Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



Langkah 2 Gabungkan blok *Find Lind (Line Format- Black. Direction-Forward. Left Speed-100. Right Speed-100. RampUp Perc-100. Sensor Threshold-20. Forward Delay-0)* dengan blok *wait (1 second)*. Gabungkan blok tersebut dengan blok di Langkah 4.

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
  
```

Pengguna menekan butang S1

After Mikrobotik starts

```

repeat until User Button -- Button #S1
  Blink All LED -- Time (ms) 100
  Turn At Centre -- Line Format Black Direction Turn Left Speed (0~255) 50 Sensor Threshold (0~1000) 200 Min Turn Period ms (0~1000) 200
  
```

Kelajuan roda

Kepekaan sisihan garisan
(lebih tinggi nilai bermaksud
lebih sensitif)

Masa minimum (ms)
robot berpusing

Semua lampu LED

```

repeat until false
  Stop
  
```

Robot secara autonominya
mengesan garisan hitam atau
putih

Arah pusingan robot

Mikrobotik berhenti selepas 1 pusingan

Langkah 3

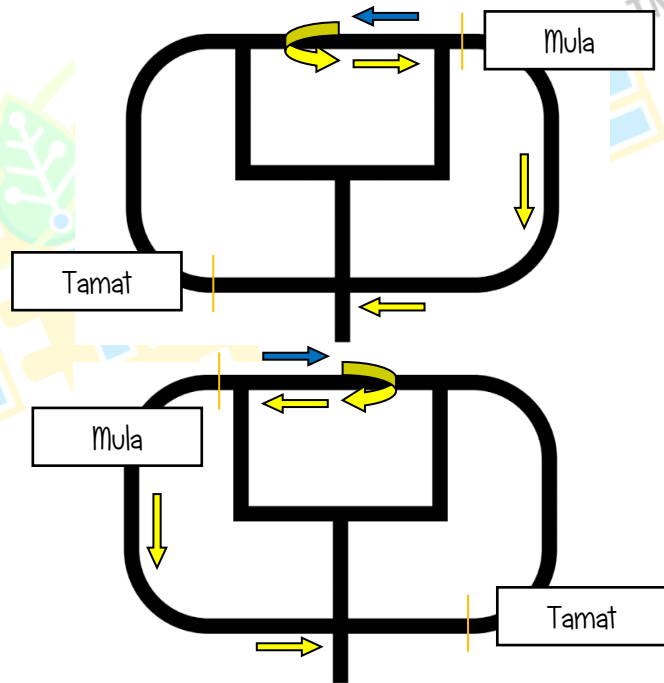
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi.

Selepas itu, Mikrobotik akan membuat pusingan-U mengikut arah yang ditetapkan dan akan berhenti setelah dapat mengesan garisan hitam.

Cabaran!!

- i) Pusing di tengah, arah kiri.
(anak panah biru ke anak panah kuning)

- ii) Pusing di tengah, arah kanan.
(anak panah biru ke anak panah kuning)



Objektif 8: Ayuh kawal Mikrobotik

Bluetooth ialah teknologi tanpa wayar jarak dekat yang digunakan untuk bertukar-tukar data antara peranti tetap dan mudah alih dalam jarak dekat dan membina rangkaian kawasan peribadi. Bluetooth membolehkan Mikrobotik bertukar data yang dikehendaki dengan peranti lain secara langsung.

Pengenalan Bluetooth dan Mekanismanya



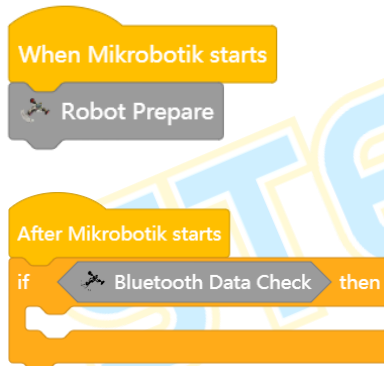
Mikrobotik boleh dikawal dalam jarak dekat menggunakan pendekatan Bluetooth kerana ia senang didapati dan senang mengawalinya. Modul Bluetooth itu dimasukkan pada pot yang disediakan pada Mikrobotik. Modul Bluetooth ini mengandungi 4 kaki. RXD, TXD, GND DAN VCC.

Langkah-langkah susunan blok

Langkah 1 Masukkan blok *When Mikrobotik Starts* dan gabungkan dengan blok *Robot Prepare*

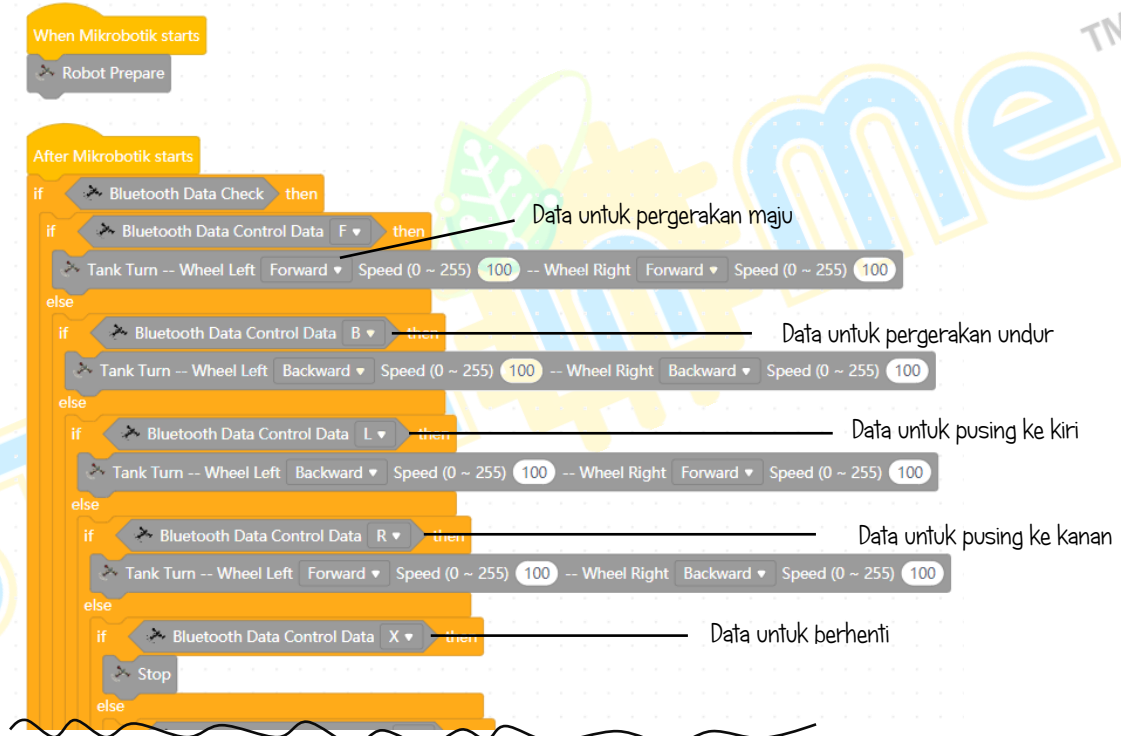


Langkah 2 Gabungkan blok *After Mikrobotik starts* dengan blok *if* yang telah digabungkan dengan blok *Bluetooth Data Check*. Letakkan blok tersebut di bawah Blok di Langkah 1.



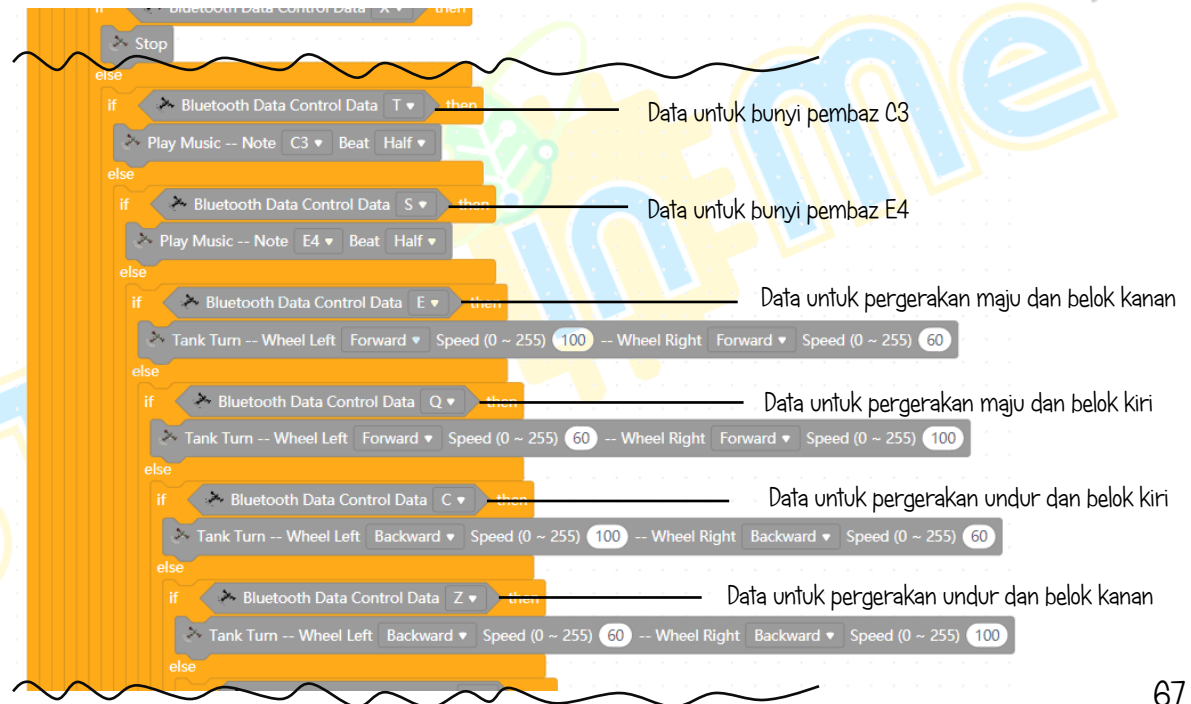
Langkah 3

Di bawah blok *After Mikrobotik starts*, gabungkan 5 blok *Bluetooth Data Control Data (F, B, L, R, X)* dengan 5 blok *if* dan di bawah blok *then* gabungkan dengan 5 blok *Tank Turn* untuk mendapat pergerakan maju, undur, pusing ke kiri, pusing ke kanan dan berhenti.



Langkah 4

Untuk blok seterusnya, gabungkan 6 blok *Bluetooth Data Control Data (T, S, E, Q, C, Z)* dengan 6 blok *if* dan di bawah blok *then* gabungkan dengan 2 blok *Play Music (Note-C3, Beat Half dan Note-E4, Beat Half)* dan 4 blok *Tank Turn* untuk mendapat pergerakan maju dan belok kanan, maju dan belok kiri, undur dan belok kiri dan undur dan belok kanan.



Langkah 5

Untuk blok seterusnya, gabungkan 5 blok *Bluetooth Data Control Data* (1, 2, 3, 4, 5) dengan 5 blok *if* dan di bawah blok *then* gabungkan dengan 4 blok *Turn On LED* (#1 On, #1 Off, #2 On, #2 Off) and 1 blok *Play Music* (Note-A3, Beat Whole).

The image shows a Scratch script with the following structure:

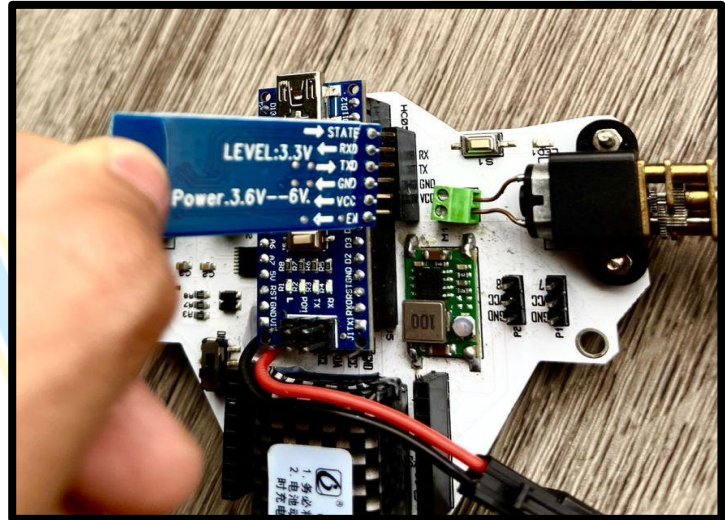
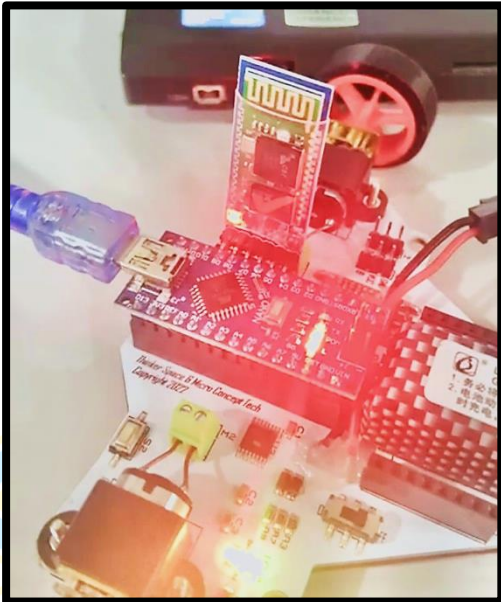
- Block 1:** *Bluetooth Data Control Data* (1) → *if* → *then* → *Turn On LED* (Status: On, LED: #1)
- Block 2:** *Bluetooth Data Control Data* (2) → *if* → *then* → *Turn On LED* (Status: Off, LED: #1)
- Block 3:** *Bluetooth Data Control Data* (3) → *if* → *then* → *Turn On LED* (Status: On, LED: #2)
- Block 4:** *Bluetooth Data Control Data* (4) → *if* → *then* → *Turn On LED* (Status: Off, LED: #2)
- Block 5:** *Bluetooth Data Control Data* (5) → *if* → *then* → *Play Music* (Note: A3, Beat: Whole)

Annotations on the right side of the code blocks:

- Data untuk nyalakan LED #1
- Data untuk padamkan LED #1
- Data untuk nyalakan LED #2
- Data untuk padamkan LED #2
- Data untuk bunyi pembaz A3

Langkah 6

Selepas memuat naik kod, pasangkan modul Bluetooth pada Mikrobotik dan padankan dengan peranti anda dan Mikrobotik akan bersedia untuk dikawal oleh peranti. Pastikan semua PIN pada Bluetooth disambungkan pada pot Bluetooth (RXD-RX, TXD-TX, GND-GND, VCC-VCC)



Penggunaan Peranti Pintar Mikrobotik

Langkah 1

Muat turun aplikasi Mikrobotik daripada:

<https://www.microconcept.com.my/stem-robotic/download/>



MIKROBOTIK

Langkah 2

Buka aplikasi dan klik pada "Bluetooth connection". Pilih berdasarkan nombor siri pada modul Bluetooth.



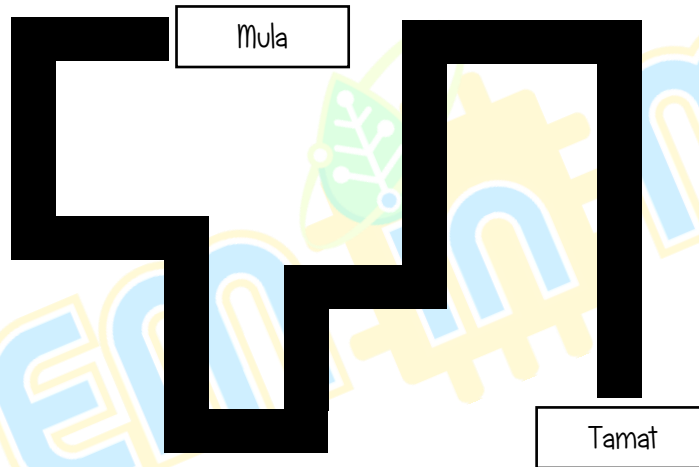
Langkah 3

Pastikan tertera "Connected". Anda sekarang boleh mengawal Mikrobotik secara pergerakan bebas.



Cabaran!!

Dalam cabaran kali ini, anda perlu memastikan Mikrobotik bergerak mengikut laluan yang telah disediakan dengan menggunakan peranti yang telah dipadankan dengan modul Bluetooth pada Mikrobotik.



Objektif 9: Kita Perlukan Peronda Kawasan !

Adakalanya robot perlu menggunakan lebih daripada satu blok untuk melengkapkan tugas seperti “robot peronda kawasan”. Untuk meronda sesuatu kawasan, robot perlu bergerak mengikut garisan dengan kelajuan yang berbeza-beza dan untuk jarak atau masa yang tertentu. Selain itu, sewaktu mengikut garisan, robot perlu membuat pusingan ke arah bertentangan.

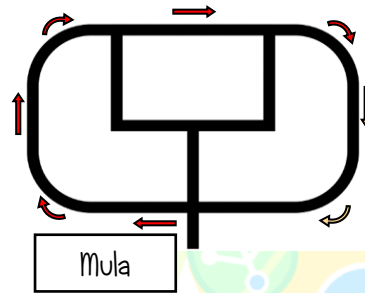
Susun Atur Strategi dan Teknik Pergerakan

Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Line Tracer Time* dan *Turn at Centre*.

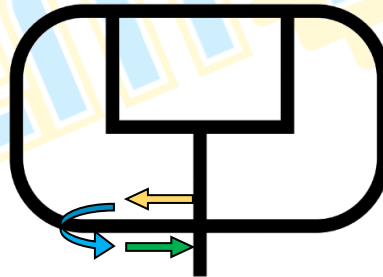
Robot bergerak secara berautonomi mengikut garisan dengan menggunakan blok *Line Tracer Time* dengan kelajuan yang tinggi dan kemudian dengan kelajuan rendah untuk masa yang tertentu. Selepas selesai berbergerak, robot membuat pusingan ke arah bertentangan dengan menggunakan blok *Turn AT Centre*. Akhirnya robot bergerak semula secara berautonomi dengan menggunakan blok *Line Tracer Time* dengan kelajuan yang tinggi.

Berikut disediakan lakaran pergerakan “robot peronda kawasan” dengan masa dan kelajuan yang ditetapkan dan membuat pusingan untuk melengkapkan tugas.

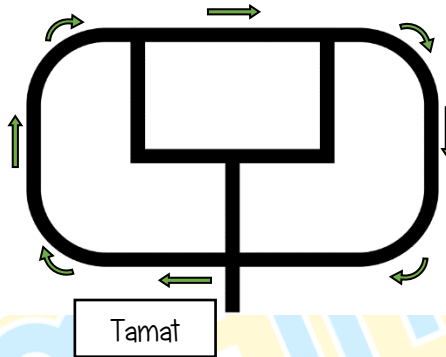
- i) Robot bergerak secara berautonomi dengan menggunakan *Line Tracer Time* dengan kelajuan tinggi selama 3 saat (anak panah merah) dan kemudian dengan kelajuan rendah selama 3 saat (anak panah kuning)



- ii) Robot membuat pusingan di tengah. ke arah kiri. (arah anak panah kuning ke anak panah hijau) menggunakan *Turn At Centre*.

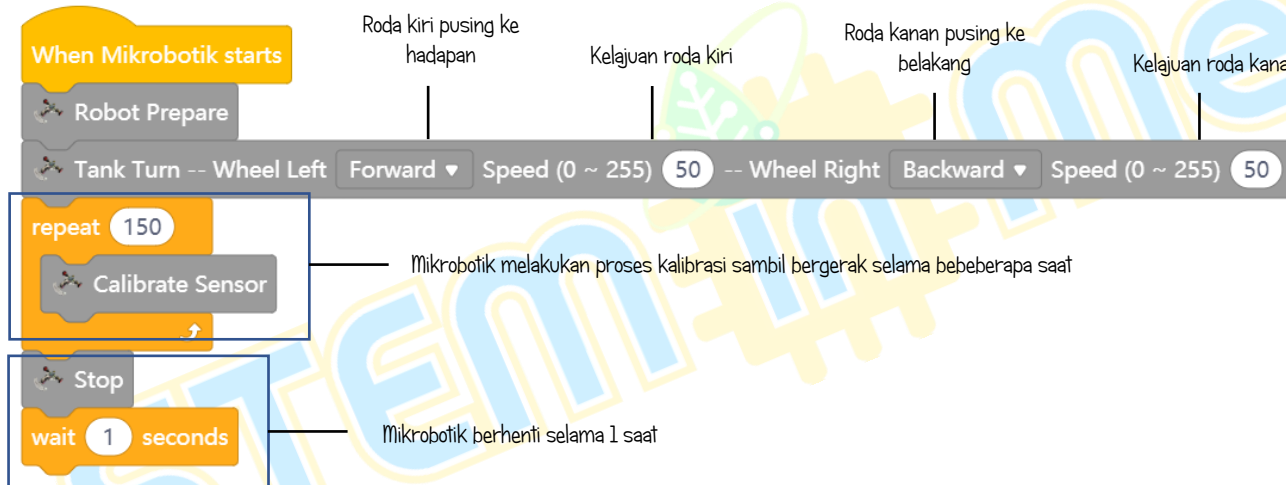


- iii) Robot bergerak secara berautonomi ke tempat tamat menggunakan *Line Tracer Time* dengan kelajuan yang tinggi selama 4 saat.



Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



Langkah 2 Gabungkan blok *After Mikrobotik starts* dengan blok *repeat until*. Selepas itu, tambahkan blok *Line Tracer Time* dengan tetapkan kelajuan kepada 80 dalam masa 3 saat. Kemudian, tambahkan lagi *blok Line Tracer Time* dengan menetapkan kelajuan kepada 30 dalam masa 3 saat.

```

After Mikrobotik starts
repeat until User Button -- Button #51
  Blink All LED -- Time (ms) 100
  Line Tracer Time -- Line Format Black Left Speed (0-255) 80 Right Speed (0-255) 80 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) 3000
  Line Tracer Time -- Line Format Black Left Speed (0-255) 30 Right Speed (0-255) 30 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) 3000
  
```

Langkah 3

Tambahkan blok *Turn At Centre* dan tetapkan untuk membuat pusingan kearah kiri. Tambahkan satu lagi *Line Tracer Time* dengan tetapkan kelajuan kepada 80 dalam masa 3 saat. Kemudian, tambahkan lagi blok *Line Tracer Time* dengan menetapkan kelajuan kepada 80 dalam masa 4 saat.

```

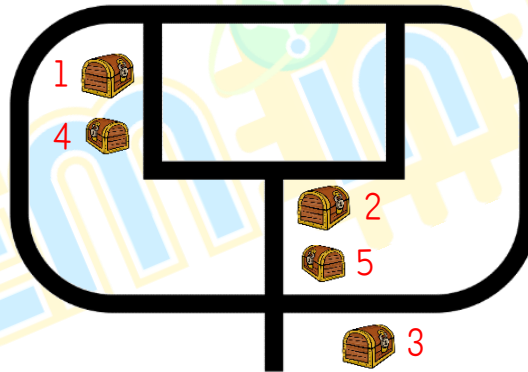
After Mikrobotik starts
repeat until User Button -- Button #51
  Blink All LED -- Time (ms) 100
  Line Tracer Time -- Line Format Black Left Speed (0-255) 80 Right Speed (0-255) 80 Turn Speed (0-255) 200 RampUp Perc (0-100) 100 Kp (0.00-1) 0.02 Kd (0.00-1) 0.2 Sensor Threshold (0-1000) 20 Time Period ms (0-10000) 3000
  Line Tracer Time -- Line Format Black Left Speed (0-255) 30 Right Speed (0-255) 30 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) 3000
  Turn At Centre -- Line Format Black Direction Turn Left Speed (0-255) 50 Sensor Threshold (0-1000) 20 Min Turn Period ms (0-1000) 200
  Line Tracer Time -- Line Format Black Left Speed (0-255) 80 Right Speed (0-255) 80 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) 4000
  
```

Langkah 4

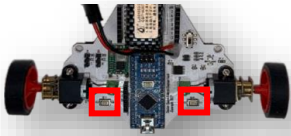
Selepas memuat naik kod, hiduapkan suis Mikrobotik dan lakukan proses kalibrasi. Selepas itu, Mikrobotik akan membuat semua pergerakan mengikut masa yang ditetapkan dan akan berhenti setelah tamat masa yang ditetapkan.

Objektif 10: Mari Mencari Harta Tersembunyi.

Adakalanya robot perlu menggunakan lebih daripada satu blok untuk melengkapkan tugas seperti "robot mencari harta tersembunyi". Untuk mendapatkan kesemua lima harta tersembunyi, robot perlu lalui banyak persimpangan yang antaranya ialah simpang kiri, simpang kanan dan simpang tiga. Adakalanya robot perlu membelok dengan kelajuan yang berbeza untuk memasuki simpang.



Pengenalan Butang Tekan / Suis dan Mekanismanya



Butang tekan adalah sejenis suis yang berfungsi mengawal sesuatu mesin secara langsung melalui sentuhan tangan atau jari daripada pengguna atau permukaan komponen. Mikrobotik mempunyai butang gtekan S1 dan S2. Nilai bacaan analog akan kurang daripada 400 apabila S1 ditekan manakala bacaan analog akan kurang daripada 500 apabila S2 ditekan.

Susun Atur Strategi dan Teknik Pergerakan.

Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder* dan *Path Finder Tank*

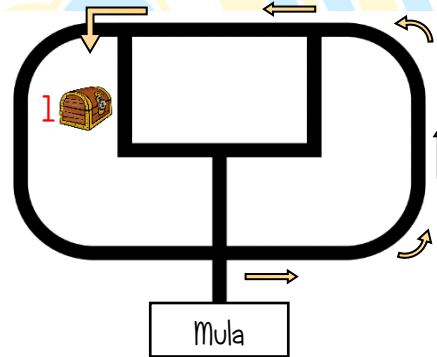
Dengan menggunakan blok *Path Finder* atau *Path Finder Tank* robot akan bergerak secara berautonomi mengikuti garisan hitam atau putih sehingga menemui persimpangan dan kemudian robot akan berpusing ke arah simpang yang ditentukan.

Untuk mendapatkan harta tersembunyi pertama, robot bergerak menggunakan *Path Finder* sehingga menemui persimpangan kiri dan berpusing ke arah kiri. Kemudian, robot meneruskan pergerakan menggunakan *Path Finder* sehingga menemui persimpangan kanan dan berpusing ke arah kanan untuk harta tersembunyi kedua. Seterusnya, robot meneruskan pergerakan menggunakan *Path Finder* sehingga menemui simpang tiga dan berpusing ke arah kiri untuk harta tersembunyi

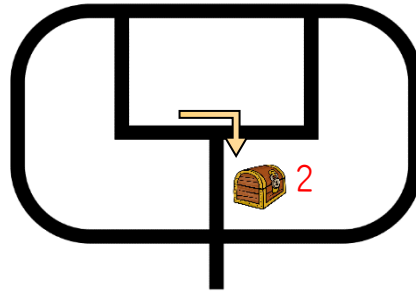
ketiga. Setelah itu, untuk mendapatkan harta tersumbunyi keempat robot perlu menggunakan *Path Finder Tank* sehingga menemui persimpangan kiri dan berpusing ke arah kiri dan akhir sekali untuk mendapatkan harta tersumbunyi terakhir robot perlu menggunakan *Path Finder Tank* sehingga menemui persimpangan kanan dan berpusing ke arah kanan

Berikut disediakan lakaran pergerakan robot mencari harta tersumbunyi dengan melalui simpang - simpang yang berbeza untuk melengkapkan tugas.

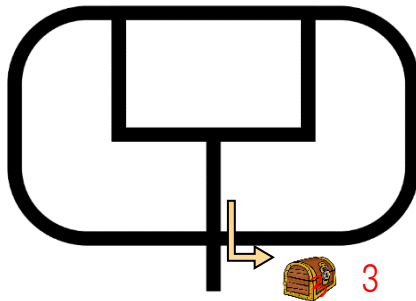
- i) Robot bergerak dari titik mula untuk mencari simpang kiri dan membelok menggunakan *Path Finder* ke arah kiri untuk mengambil harta tersumbunyi pertama.



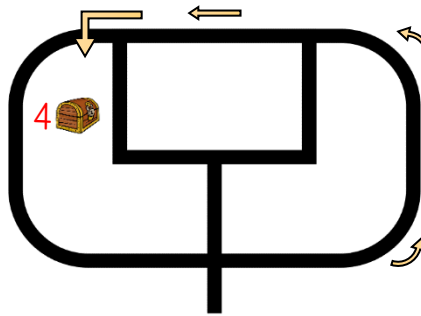
- ii) Robot bergerak mencari simpang kanan dan membelok menggunakan *Path Finder* pusingan ke arah kanan untuk mengambil harta tersembunyi yang kedua.



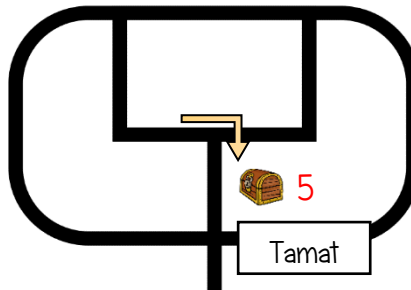
- iii) Robot bergerak mencari simpang tiga dan membelok menggunakan *Path Finder* ke arah kiri untuk mengambil harta yang tersembunyi ketiga.



- iv) Robot bergerak sekali lagi untuk mencari simpang kiri dan membelok menggunakan *Path Finder Tank* ke arah kiri untuk mengambil harta tersembunyi keempat.

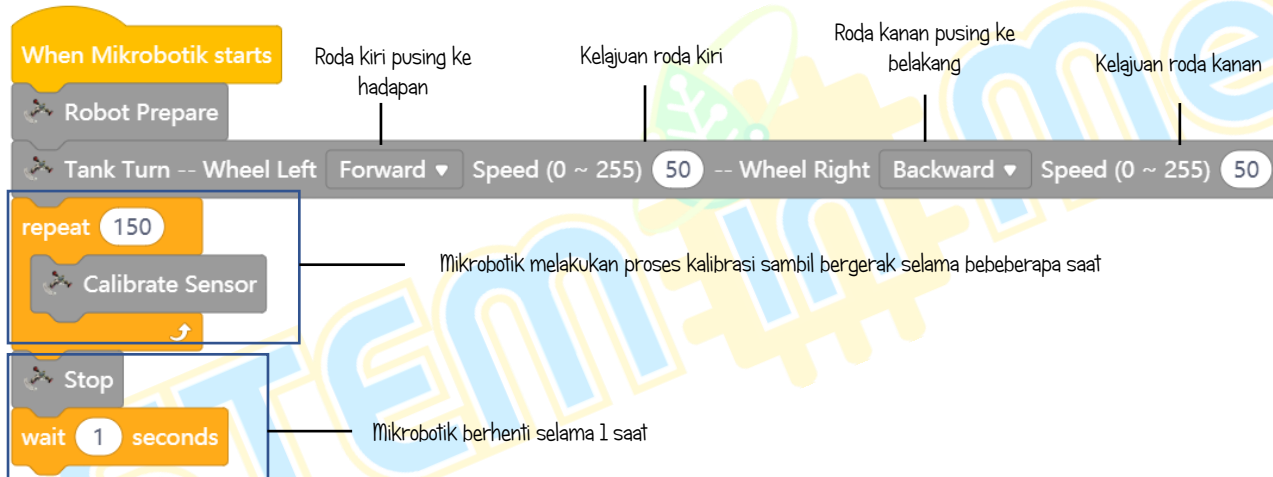


- v) Robot teruskan bergerak mencari simpang kanan dan membelok menggunakan *Path Finder Tank* ke arah kanan untuk mengambil harta tersembunyi terakhir dan lalu berhenti.



Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



Langkah 2

Gabungkan blok *After Mikrobotik starts* dengan blok *repeat until*. Selepas itu, tambahkan blok *Path Finder* Dan tetapkan (*Junction* – **“Left”**; *Action* – **“Turn Left”**; *Speed* – **“60”**; *Turn Speed* – **“200”**; *Junction Speed* – **“200”**; *Forward Delay* – **“400”** and *Turn Period* – **“400”**).



Langkah 3

Tambahkan blok *Path Finder* yang Baharu dan tetapkan nilai kepada (*Junction* – **“Right”**; *Action* – **“Turn Right”**; *Speed* – **“60”**; *Turn Speed* – **“200”**; *Junction Speed* – **“50”**; *Forward Delay* – **“0”** dan *Turn Period* – **“600”**).



Langkah 4

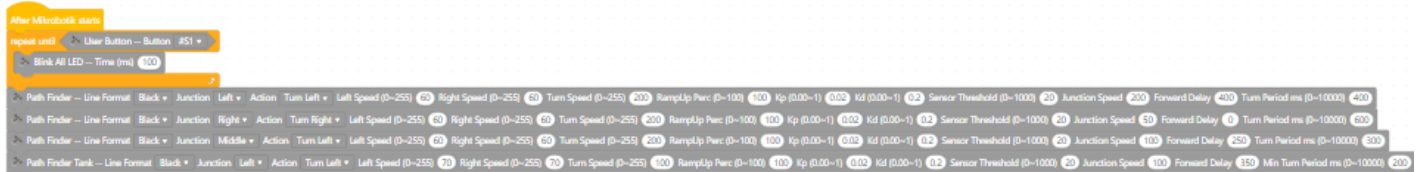
Tambahkan blok *Path Finder* yang Baru dan tetapkan nilai kepada (*Junction – “Middle”: Action – “Turn Left”. Speed – “60”. Turn Speed – “200”. Junction Speed – “100”. Forward Delay – “250” dan Turn Period – “300”*).

TM



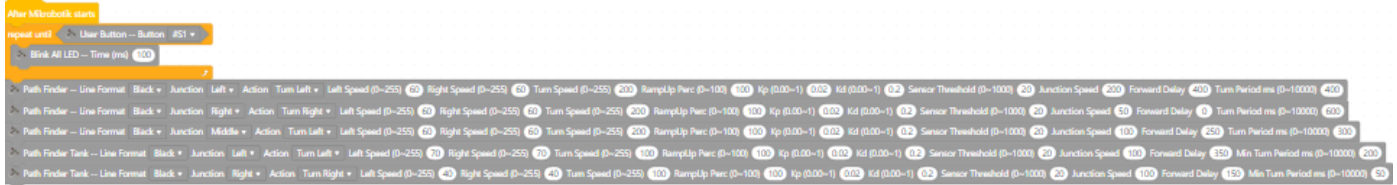
Langkah 5

Tambahkan blok *Path Finder Tank* dan tetapkan nilai kepada (*Junction – “Left”: Action – “Turn Left”. Speed – “70”. Turn Speed – “100”. Junction Speed – “100”. Forward Delay – “350” dan Min Turn Period – “200”*).



Langkah 6

Tambahkan blok *Path Finder Tank* yang Baru dan tetapkan nilai kepada (*Junction* – **“Right”**; *Action* – **“Turn Right”**; *Speed* – **“40”**; *Turn Speed* – **“100”**; *Junction Speed* – **“100”**; *Forward Delay* – **“150”** dan *Min Turn Period* – **“50”**).



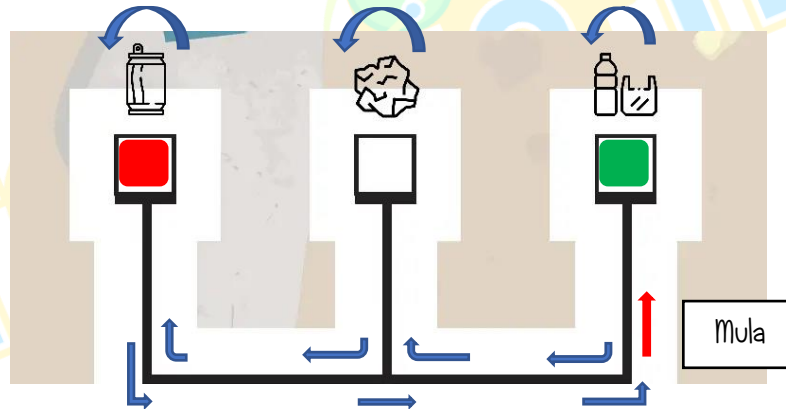
Langkah 7

Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi.

Selepas itu, Mikrobotik akan bergerak mencari persimpangan yang ditetapkan dan membuat pusingan sehingga menjumpai harta yang tersembunyi.

Objektif 11: Pengasingan Bahan Kitar Semula.

Objektif ini menumpukan kepada usaha pengasingan bahan kitar semula dengan menggunakan robot. Untuk membantu usaha kitar semula, kita boleh menggunakan bantuan robot untuk mengalihkan objek bahan kitar semula dari satu tempat ke satu tempat khusus yang tertentu. Untuk tujuan ini, robot menggunakan alat pencengkam untuk memegang objek kitar semula kemudian robot bergerak dari satu tempat ke satu tempat khusus tersebut dan akhirnya melepaskan objek.



Pengenalan Pencengkam Tunggal dan Mekanismanya

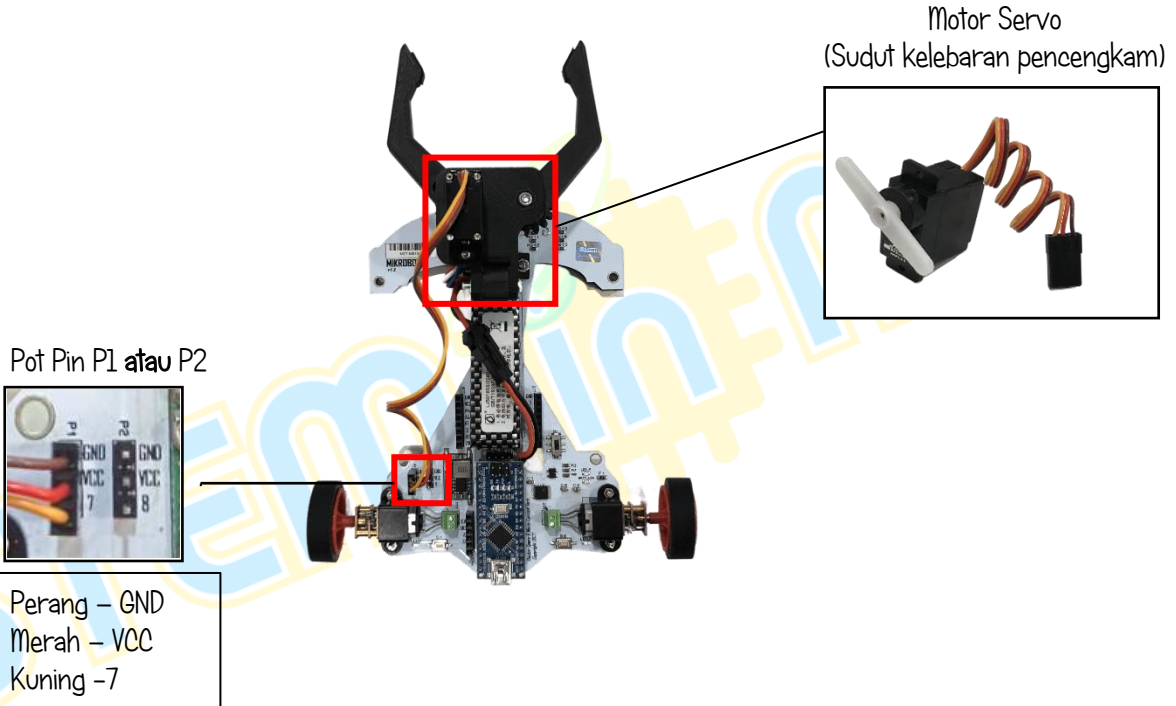


Pencengkam tunggal digunakan dalam Mikrobotik untuk mencengkam objek dan mengalihkan (secara menyeret) objek ke lokasi yang dikehendaki. Pencengkam tunggal terdiri daripada satu motor servo dan satu tangan pencengkam mekanikal.

Pencengkam tunggal dipasang pada Mikrobotik menggunakan skru dan nat manakala wayar motor servo disambungkan pada pin pot servo sedia ada pada Mikrobotik berlabel "P1" atau "P2".

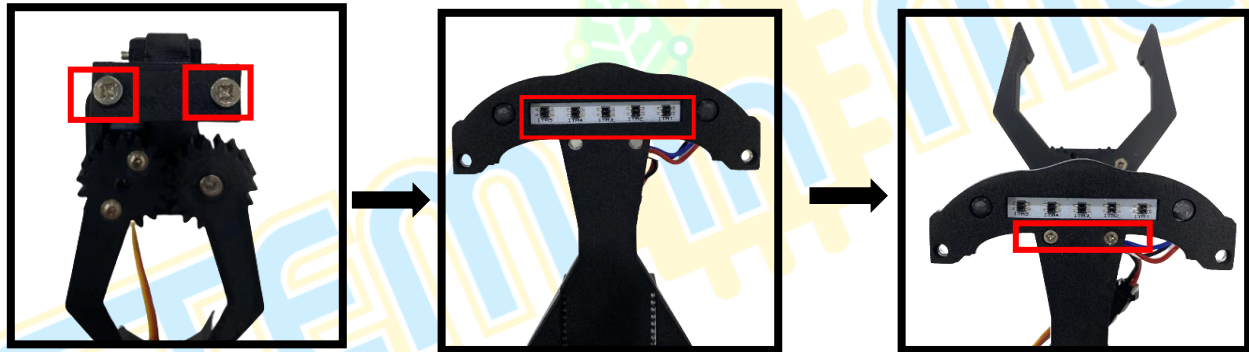
Lebar bukaan tangan pencengkam boleh dilaraskan dengan menetapkan sudut motor servo. Lebih besar sudut servo, lebih kecil bukaan tangan pencengkam. Biasanya apabila sudut motor servo ditetapkan pada 0 darjah, bukaan tangan pencengkam adalah pada keadaan paling luas. Apabila sudut motor servo ditetapkan pada 180 darjah, bukaan tangan pencengkam adalah pada keadaan paling kecil ataupun tertutup sepenuhnya.

Pemasangan Pencengkam Tunggal pada robot.

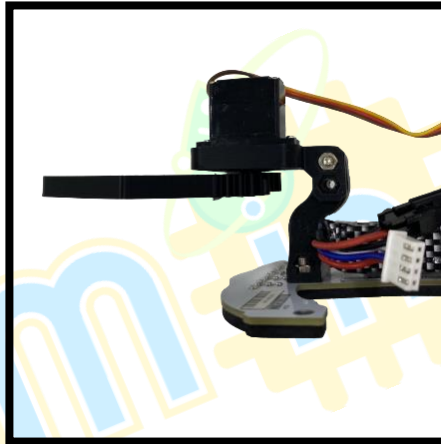


Berikut merupakan susunan langkah pemasangan Pencengkam Tunggal pada Mikrobotik:

- 1- Longgarkan skru dan nat pada Pencengkam Tunggal. Dibawah robot Mikrobotik terdapat dua lubang skru untuk Pencengkam Tunggal. Masukkan skru dan nat kemudian ketatkan.



2- Pastikan posisi Pencengkam Tunggal selepas pemasangan adalah seperti ini.



Susun Atur Strategi dan Teknik Pergerakan.

Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder*, *Gripper Servo Port* dan *Turn at Centre*.

Dengan menggunakan blok *Path Finder*, robot akan bergerak secara berautonomi mengikuti garisan hitam atau putih sehingga menemui objek yang pertama.

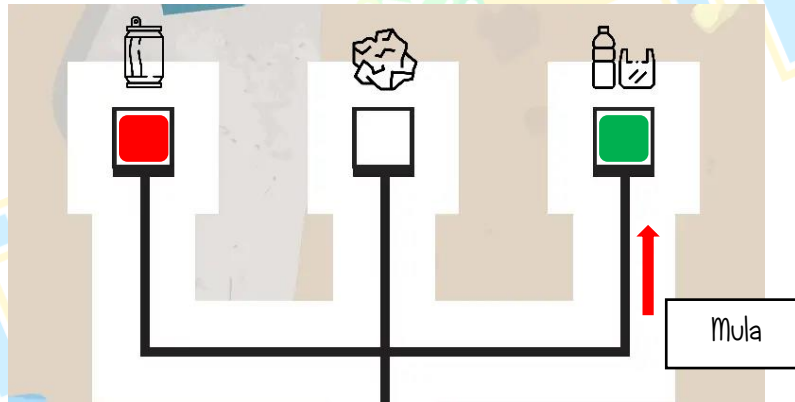
Untuk mendapatkan objek pertama (hijau), robot mencengkam objek itu menggunakan blok *Gripper Servo Port*. Kemudian, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kiri kanan di ruang khas dan berhenti untuk meletakkan objek ke dalam ruang khas menggunakan *Gripper Servo Port*.

Seterusnya, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kiri kanan di ruang khas dan berhenti untuk mengambil objek kedua (merah) menggunakan blok *Gripper Servo Port*.

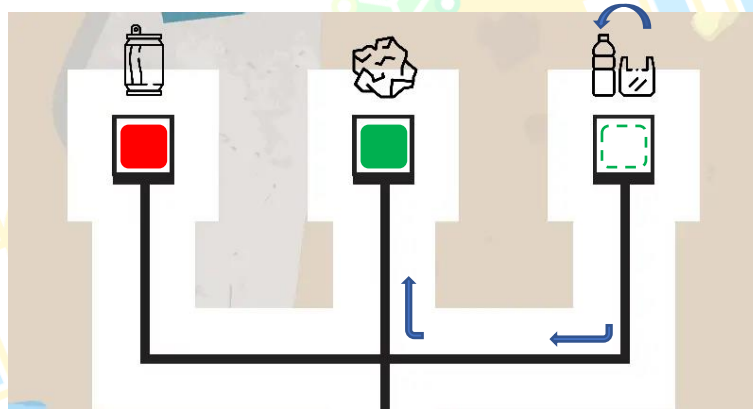
Setelah itu, robot berpusing menggunakan blok *Turn at Centre* dan bergerak menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kiri kanan di ruang khas lalu meletakkan objek kedua di dalam ruang khas menggunakan blok *Gripper Servo Port*.

Berikut disediakan lakaran pergerakan robot mengangkat objek berasingan untuk disusun mengikut ruang yang dikhaskan.

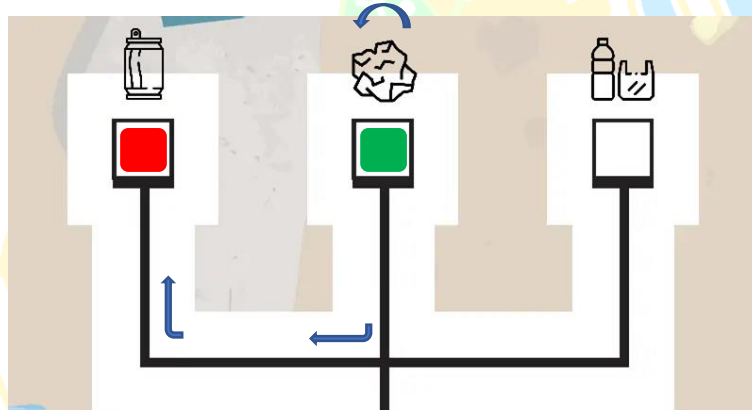
- i) Menggunakan blok *Path Finder*, robot bergerak dari titik mula sehingga bertemu simpang kiri kanan dan berhenti. Kemudian, robot mencengkam objek pertama (hijau) menggunakan blok *Gripper Servo Port* (*Pencengkam*) dengan tetapan sudut yang besar (bukan tangan kecil).



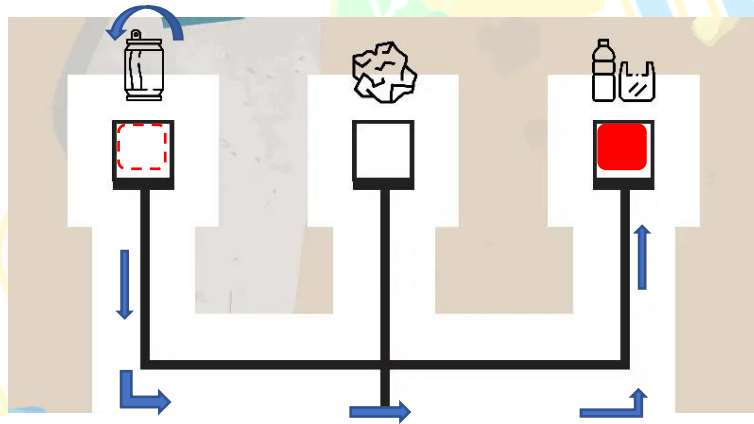
- ii) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre*. kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kanan dan pusing ke kanan. Seterusnya robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kanan. Selepas itu, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan berhenti. Akhir sekali melepaskan objek kedua (merah) itu ke dalam satu ruang disediakan menggunakan blok *Gripper Servo Port* dengan tetapan sudut yang kecil (bukan tangan besar).



- iii) Selepas melepaskan objek pertama, robot berpusing memnghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kanan. Selepas itu, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kanan dan pusing ke kanan lalu berhenti. Kemudian, robot mencengkam objek kedua (merah) menggunakan blok *Gripper Servo Port* (Pencengkam) dengan tetapan sudut besar (bukaan tangan kecil).



- i) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri dan pusing ke kiri. Seterusnya, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri dan pusing kiri. Selepas itu, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan berhenti. Akhir sekali, melepaskan objek kedua (merah) itu ke dalam satu ruang disediakan menggunakan blok *Gripper Servo Port* dengan tetapan sudut yang kecil (bukaan tangan besar).



Langkah-langkah susunan blok

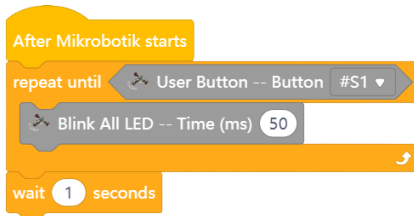
Langkah 1 Ambil maklum wayar motor servo pencengkam tunggal disambungkan ke pot P1 atau P2.

Langkah 2 Sediakan susunan blok kalibrasi automatik.



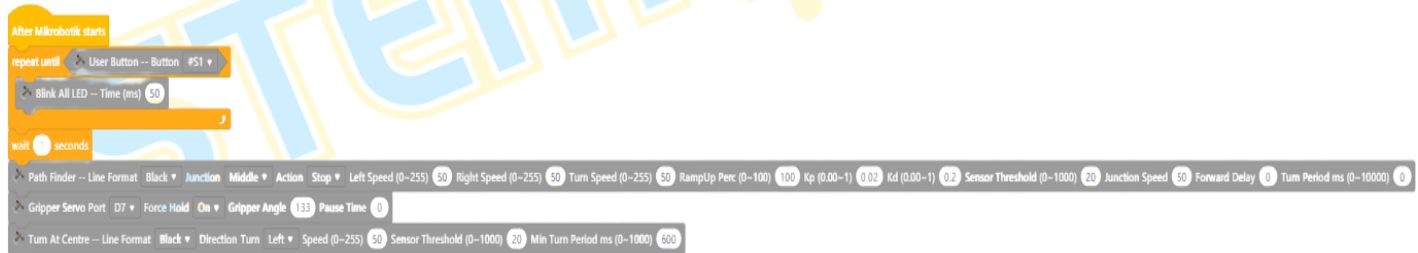
Langkah 3

Gabungkan blok *After Mikrobotik starts* dengan blok *repeat until* dan *wait 1 seconds*.



Langkah 4

Tambahkan blok *Path Finder* dan tetapkan (*Juction – "Middle": Action – "Stop": Speed – "50": Turn Speed – "50": Junction Speed – "50": Forward Delay – "0"* dan *Turn Period – "0"*). Tambahkan blok *Gripper* dan tetapkan (*Gripper Servo Port – "D7": Force hold – "On": Gripper Angle – "133": Pause – "0"*). diikuti dengan blok *Turn At Centre* (*Direction Turn – "Left": Speed – "50": MinTurn Period – "600"*).



Langkah 5

Tambahkan 3 blok *Path Finder* dan tetapkan mengikut turutan blok (*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*]). Tambahkan blok *Gripper* dan tetapkan (*Gripper Servo Port* – *D7*; *Force hold* – *Off*; *Gripper Angle* – *20*; *Pause* – *500*), diikuti dengan blok *Turn At Centre* (*Direction Turn* – *Left*; *Speed* – *50*; *Min Turn Period* – *300*).

Langkah 6

Tambahkan 3 blok *Path Finder* dan tetapkan mengikut turutan blok (*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”*]). Tambahkan blok *Gripper* dan tetapkan (*Gripper Servo Port* – *“D7”*; *Force hold* – *“On”*; *Gripper Angle* – *“133”*; *Pause* – *“0”*), diikuti dengan blok *Turn At Centre* (*Direction Turn* – *“Left”*; *Speed* – *“50”*; *Min Turn Period* – *“300”*).

Langkah 7

Tambahkan 4 blok *Path Finder* dan tetapkan mengikut turutan blok (*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*]). Tambahkan blok *Gripper* dan tetapkan (*Gripper Servo Port* – *D7*, *Force hold* – *Off*, *Gripper Angle* – *20*, *Pause* – *550*).

Maklumat Tambahan

Force Hold (Daya Cengkaman)

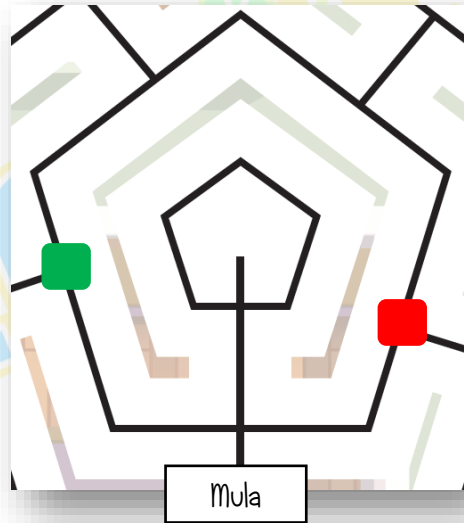
- Apabila daya cengkaman dihidupkan (ON) pencengkam akan menahan sudut kelebarannya daripada diubah.
- Apabila daya cengkaman dimatikan (OFF), pencengkam tidak akan menahan sudut kelebarannya daripada diubah.

Pause Time (Tempoh Jeda)

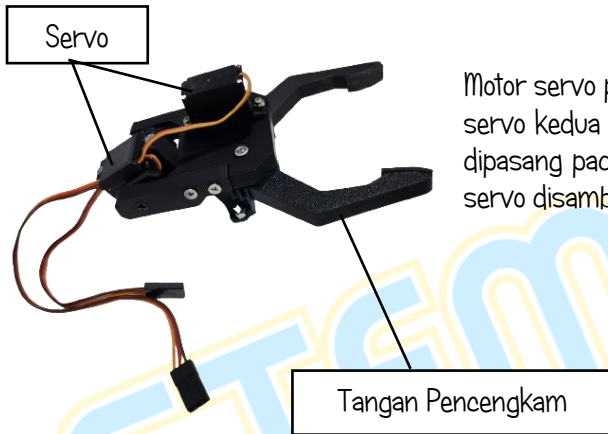
- Tempoh jeda hanya boleh digunapakai apabila daya cengkaman dimatikan (OFF).
- Fungsi waktu jeda digunakan untuk menghidupkan daya cengkaman mengikut nilai jeda yang ditetapkan. Setelah tempoh jeda tamat, daya cengkaman akan dimatikan.

Objektif 12: Penyimpanan Cekap Ruang.

Adakalanya kita perlu menyusun objek yang banyak secara bertindan agar ruang itu dapat digunakan dengan sepenuhnya. Di sinilah pencengkam servo berganda digunakan. Sama seperti namanya, ia dilengkapi dengan dua motor servo.

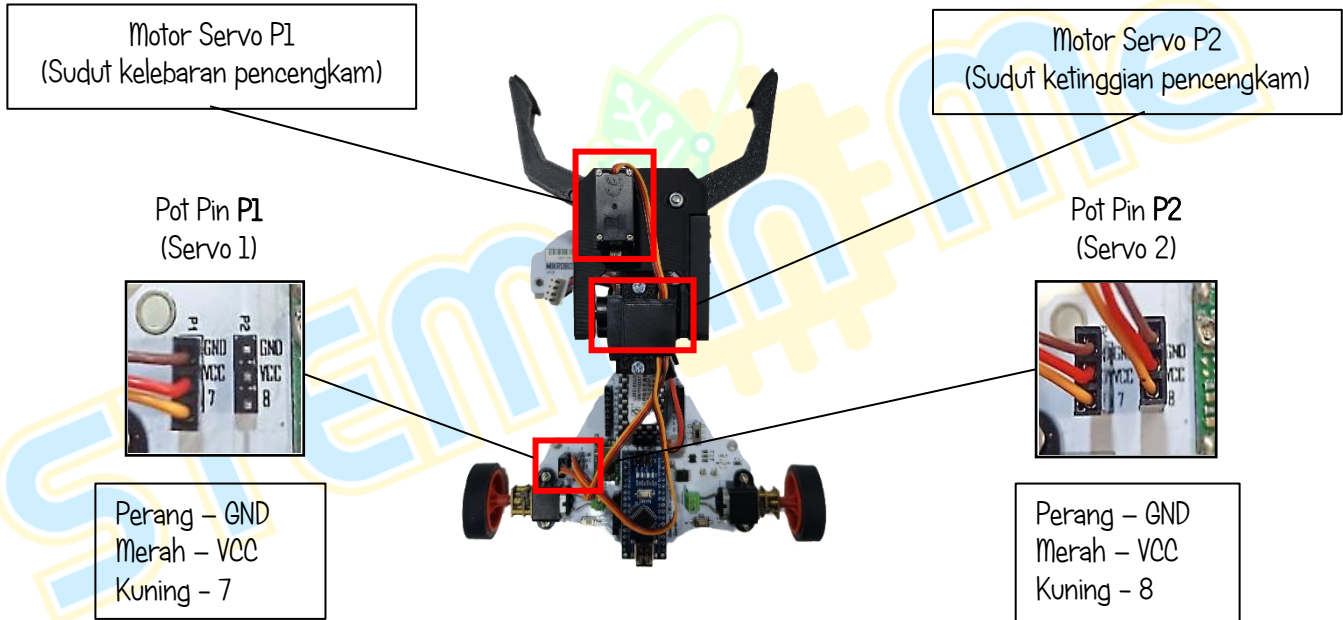


Pengenalan Pencengkam Berganda dan Mekanismenya



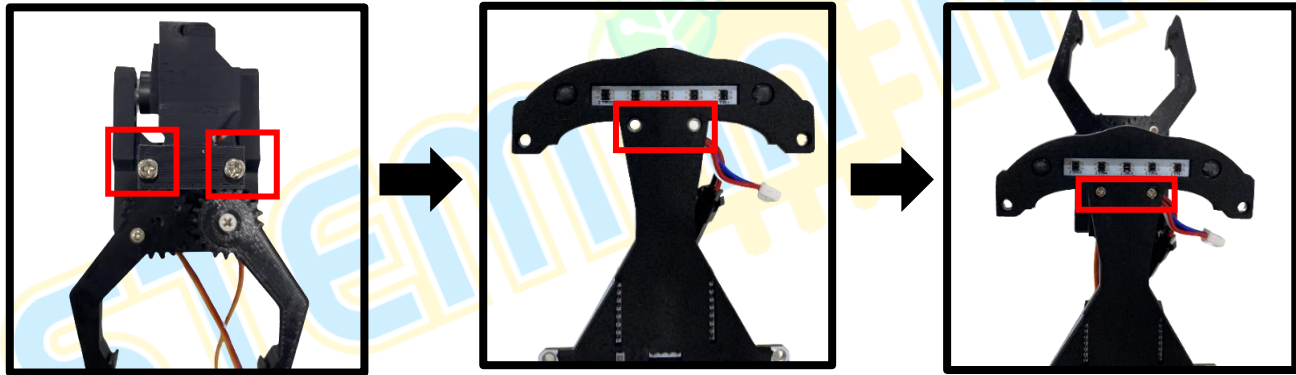
Motor servo pertama digunakan untuk mengawal lebar pencengkam. Manakala motor servo kedua digunakan untuk mengawal sudut angkat pencengkam. Oleh itu, apabila dipasang pada Mikrobotik, robot akan dapat mmbawa dan mengangkat objek. Wayar servo disambungkan ke pin pengepala servo sedia ada berlabel "P1" dan "P2".

Pemasangan Pencengkam Berganda pada robot.

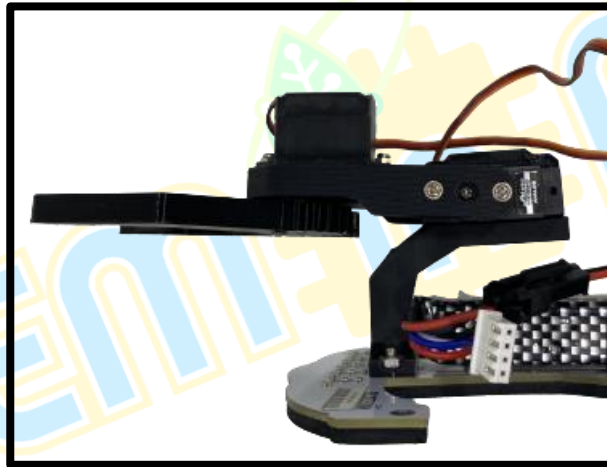


Berikut merupakan susunan langkah pemasangan Pencengkam Berganda pada Mikrobotik:

- 1- Longgarkan skru dan nat pada Pencengkam Berganda. Dibawah robot Mikrobotik terdapat dua lubang skru untuk Pencengkam Berganda. Masukkan skru dan nat kemudian ketatkan.



2- Pastikan posisi Pencengkam Berganda selepas pemasangan adalah seperti ini.



Susun Atur Strategi dan Teknik Pergerakan.

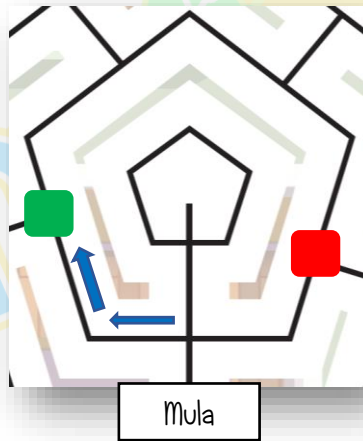
Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder*, *Line Tracer Time*, *Gripper Servo Port* dan *Turn at Centre*.

Dengan menggunakan blok *Path Finder* dan *Line Tracer Time* robot akan bergerak secara berautonomi mengikuti garisan hitam atau putih sehingga menemui objek yang pertama.

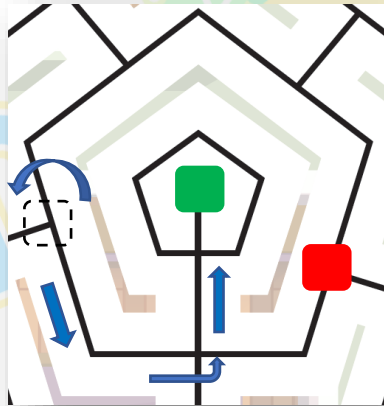
Untuk mendapatkan objek pertama (hijau), robot mencengkam objek itu menggunakan blok *Gripper Servo Port*. Kemudian, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan blok *Path Finder* sehingga menemui persimpangan kiri dan robot akan membelok ke kiri. Kemudian robot akan terus bergerak menggunakan blok *Path Finder* sehingga menjumpai simpang kiri kanan. robot akan pusing ke kiri. Seterusnya robot meneruskan pergerakan menggunakan blok *Path Finder* sehingga menemui simpang mati lalu berhenti untuk meletakkan objek ke dalam ruang khas menggunakan blok *Gripper Servo Port*. Seterusnya, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan beberapa blok *Path Finder* sehingga menemui simpang kiri kanan. kemudian robot pusing ke kiri lalu bergerak menggunakan *Line Tracer Time* untuk mengambil objek kedua (merah) menggunakan blok *Gripper Servo Port*. Setelah itu, robot berpusing menggunakan blok *Turn at Centre* dan bergerak menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kanan dan kiri kanan dan robot pusing ke kanan. Seterusnya robot mengangkat objek menggunakan blok *Gripper Servo Port* dan bergerak menggunakan blok *Path Finder* sehingga menemui simpang mati lalu berhenti untuk meletakkan objek kedua (merah) secara bertindan di dalam ruang khas menggunakan blok *Gripper Servo Port*. Akhir sekali, robot berpusing menggunakan blok *Turn at Centre* dan bergerak menggunakan blok *Path Finder* sehingga menemui sehingga menemui persimpangan kiri kanan lalu berhenti.

Berikut disediakan lakaran pergerakan robot mengangkat objek berasingan untuk disusun secara bertindan dalam satu ruang yang sama.

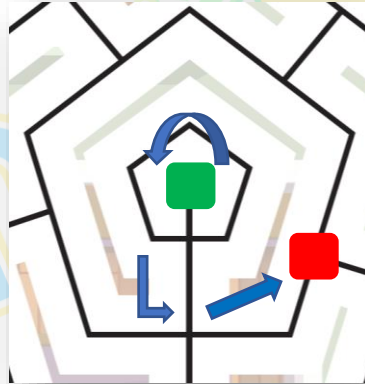
- i) Menggunakan blok *Path Finder*: robot bergerak dari titik mula sehingga bertemu simpang kiri kanan dan pusing ke kiri. Kemudian robot bergerak menggunakan blok *Path Finder* dan menemui persimpangan kanan lalu pusing ke kanan kemudian menggunakan blok *Line Tracer Time* dan mencengkam objek pertama (hijau) menggunakan blok *Gripper Servo Port (Pencengkam)* dengan tetapan sudut yang besar (bukaan tangan kecil).



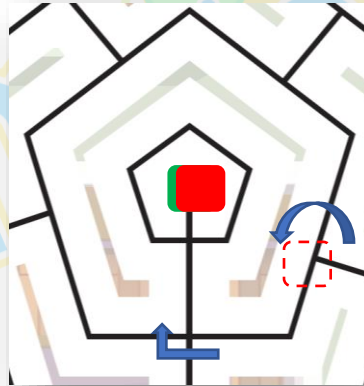
- ii) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri dan pusing ke kiri. Seterusnya robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kiri kemudian bergerak menggunakan blok *Path Finder* sehingga bertemu simpang mati lalu berhenti kemudian melepaskan objek pertama (hijau) itu ke dalam satu ruang disediakan menggunakan blok *Gripper Servo Port* dengan tetapan sudut yang kecil (bukaan tangan besar).



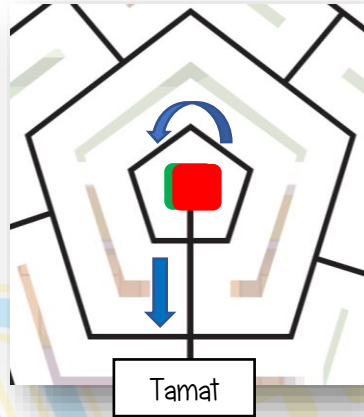
- iii) Selepas melepaskan objek pertama (hijau). robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kiri. Seterusnya. robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri dan pusing ke kiri kemudian menggunakan blok *Line Tracer Time* dan mencengkam objek kedua (merah) menggunakan blok *Gripper Servo Port* (Pencengkam) dengan tetapan sudut besar (bukakan tangan kecil).



- iv) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kanan dan pusing ke kanan. Seterusnya, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan kemudian pusing ke kanan dan berhenti lalu mengangkat objek kedua (merah) menggunakan blok *Gripper Servo Port* dengan tetapan sudut besar (ketinggian tangan). Setelah itu, robot bergerak menggunakan blok *Path Finder* dan bergerak ke persimpangan mati lalu berhenti. Selepas itu, robot meletakkan dan melepaskan objek kedua (merah) secara bertindan menggunakan blok *Gripper Servo Port* dengan tetapan sudut besar (ketinggian tangan) dan tetapan sudut kecil (bukaan tangan besar).



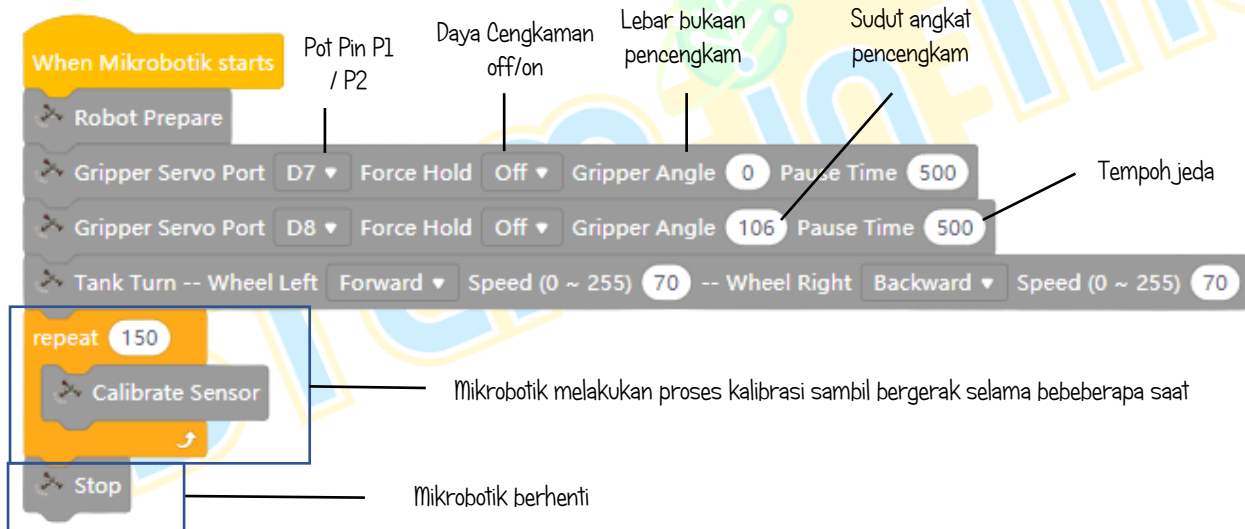
- v) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* dan bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan lalu berhenti.



Langkah-langkah susunan blok

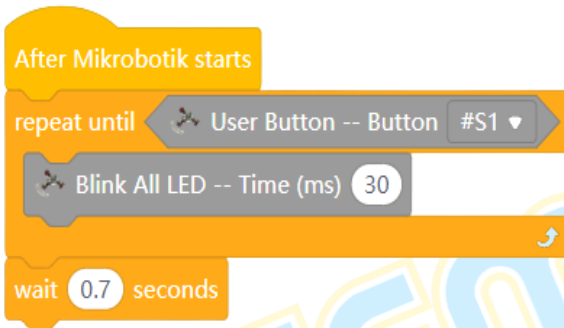
Langkah 1 Pastikan wayar motor servo disambungkan ke pin pengepala servo P1 dan P2.

Langkah 2 Sediakan susunan blok kalibrasi automatik.



Langkah 3

Gabungkan blok *After Mikrobotik starts* dengan blok *repeat until* dan *wait 0.7 seconds*.



Langkah 4

Tambahkan 2 blok *Path Finder* dan tetapkan nilai (*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”*]). Tambahkan blok *Line Tracer Time* dan tetapkan nilai (*Speed* – *“50”*, *Turn Speed* – *“50”*, *Time Period* – *“1100”*). Tambahkan blok *Gripper* dan tetapkan nilai (*Gripper Servo Port* – *“D7”*, *Force hold* – *“On”*, *Gripper Angle* – *“95”*, *Pause* – *“0”*), diikuti dengan blok *Turn At Centre* (*Direction Turn* – *“Left”*, *Speed* – *“80”*, *Min Turn Period* – *“600”*).

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

- 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 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: 50, Right Speed: 50, Turn Speed: 50, RampUp Perc: 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: 80, Sensor Threshold: 0-1000, 20, Min Turn Period ms: 0-1000, 600)

Langkah 5

Tambahkan 3 blok *Path Finder* dan tetapkan nilai mengikut turutan blok (*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”*]). Tambahkan 2 blok *Gripper* dan tetapkan nilai (*Gripper Servo Port* – [*“D7”*; *“D8”*], *Force hold* – [*“Off”*; *“Off”*], *Gripper Angle* – [*“0”*; *“106”*], *Pause* – [*“0”*; *“500”*]), diikuti dengan blok *Turn At Centre* (*Direction Turn* – *“Left”*, *Speed* – *“80”*, *MinTurn Period* – *“600”*).

Langkah 6

Tambahkan 2 blok *Path Finder* dan tetapkan nilai mengikut turutan blok (*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* – [*“600”*; *“400”*]). Tambahkan blok *Line Tracer Time* dan tetapkan nilai (*Speed* – *“50”*; *Turn Speed* – *“50”*; *Time Period* – *“350”*). Tambahkan blok *Gripper* dan tetapkan nilai (*Gripper Servo Port* – *“D7”*; *Force hold* – *“On”*; *Gripper Angle* – *“100”*; *Pause* – *“0”*). diikuti dengan blok *Turn At Centre* (*Direction Turn* – *“Left”*; *Speed* – *“80”*; *Min Turn 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 parameters:

- Path Finder** (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, 550)
- Path Finder** (Junction: Right, Action: Turn Right, Left Speed: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 0-100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 0-1000, 20, Junction Speed: 100, Forward Delay: 70, Turn Period ms: 0-10000, 400)
- Line Tracer Time** (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** (Direction Turn: Left, Speed: 0-255, 80, Sensor Threshold: 0-1000, 20, Min Turn Period ms: 0-1000, 600)
- Path Finder** (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** (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** (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: 100, Pause Time: 500)
- Turn At Centre** (Direction Turn: Left, Speed: 0-255, 80, Sensor Threshold: 0-1000, 20, Min Turn Period ms: 0-1000, 600)
- Path Finder** (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, 500)
- Path Finder** (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** (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** (Direction Turn: Left, Speed: 0-255, 80, Sensor Threshold: 0-1000, 20, Min Turn Period ms: 0-1000, 800)

Langkah 7

Tambahkan 2 blok *Path Finder* dan tetapkan nilai mengikut turutan blok (*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*]). Tambahkan blok *Gripper* dan tetapkan nilai (*Gripper Servo Port* – *D8*; *Force hold* – *On*; *Gripper Angle* – *70*; *Pause* – *500*).

Langkah 8

Tambahkan blok *Path Finder* dan tetapkan nilai (*Junction* – *DeadEnd*: *Action* – *Stop*: *Speed* – *70*: *Turn Speed* – *70*: *Junction Speed* – *70*]. *Forward Delay* – *0* dan *Turn Period* – *0*). Tambahkan 2 blok *Gripper* dan tetapkan nilai (*Gripper Servo Port* – [*D8*: *D7*]. *Force hold* – [*Off*: *Off*]. *Gripper Angle* – [*90*: *0*]. *Pause* – [*500*: *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) 20)
- wait** (0.7 seconds)
- 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: 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, 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: 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: 80, Sensor Threshold: 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: 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, 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: 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: 100, Pause Time: 500)
- Turn At Centre** (Line Format: Black, Direction: Turn Left, Speed: 80, Sensor Threshold: 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: 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, 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: 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: 80, Sensor Threshold: 20, Min Turn Period ms: 0-1000, 800)
- 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: 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: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 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: 70, Right Speed: 70, Turn Speed: 70, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 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)

Langkah 9

Tambahkan blok *Turn At Centre* (*Direction Turn* – “*Left*”. *Speed* – “*80*”. *Min Turn Period* – “*600*”).
Kemudian tambahkan blok *Path Finder* dan tetapkan nilai kepada (*Junction* – “*Middle*”. *Action* – “*Stop*”.
Speed – “*100*”. *Turn Speed* – “*100*”. *Junction Speed* – “*100*”. *Forward Delay* – “*70*” dan *Min Turn Period*
– “*100*”).

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 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) 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) 450

▶ 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 500 Pause Time 0

▶ 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

Tambahan: Cuba Naik Taraf dan Pengaturcaraan Sendiri

POT PERANTI	PIN ARDUINO NANO	PERANTI	MAKLUMAT TAMBAHAN
ITR1	A6	Sensor Pengesanan Garisan – Kiri Luar	ITR8307
ITR2	A3	Sensor Pengesanan Garisan – Kiri Dalam	ITR8307
ITR3	A2	Sensor Pengesanan Garisan – Tengah	ITR8307
ITR4	A1	Sensor Pengesanan Garisan – Kanan Dalam	ITR8307
ITR5	A0	Sensor Pengesanan Garisan – Kanan Luar	ITR8307
S1	A7	Suis Pengguna S1	Nilai bacaan < 100
S2	A7	Suis Pengguna S2	Nilai bacaan ≥ 100 & < 400
BUZZER	D2	Pembaz	
LED1	D13	Lampu Indikator L1	
LED2	D12	Lampu Indikator L2	
M1 – AIN1	D5	Motor Kiri – Bridge A Input 1	DRV8833 Dual H-Bridge Motor Driver
M1 – AIN2	D6	Motor Kiri – Bridge A Input 2	DRV8833 Dual H-Bridge Motor Driver
M2 – BIN1	D3	Motor Kanan – Bridge B Input 1	DRV8833 Dual H-Bridge Motor Driver
M2 – BIN2	D9	Motor Kanan – Bridge B Input 2	DRV8833 Dual H-Bridge Motor Driver
P1	D7	Pot Terbuka P1	
P2	D8	Pot Terbuka P2	
BT – TX	D10	Pot Bluetooth TX	
BT – RX	D11	Pot Bluetooth 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