

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.....	15
Algoritma PID Robot Berautonomi.....	21
Apa jenis-jenis Litar?.....	22
Jenis-jenis Persimpangan.....	23

Objektif 1: Vroom Vroom	24
Pengenalan Mudah Pembaz.....	24
Langkah-langkah susunan blok.....	24
Cabaran !!.....	26
Objektif 2: Tolong Hidupkan Lampu!	27
Pengenalan Mudah Diod Pemancar Cahaya (LED).....	27
Langkah-langkah susunan blok.....	27
Cabaran!!.....	29
Objektif 3: Mulakan Pengembaraan Kita (Pergerakan Bebas)	30
Pengenalan kepada Motor.....	30
Pengenalan Mudah Pergerakan Asas Robot.....	31
Langkah-langkah susunan blok.....	34
Cabaran!!.....	39
Objektif 4: Ayuh Ikuti Garisan Itu	40
Pengenalan mudah Pengesan Garisan.....	40
Pengenalan <i>Line Tracer Time</i> dan Mekanismanya.....	40

Langkah-langkah susunan blok.....	41
Cabaran!!.....	45
Objektif 5: Apa Yang Perlu Dilakukan Ketika Di Persimpangan?	46
Pengenalan <i>Path Finder</i> dan Mekanismanya.	46
Langkah-langkah susunan blok.....	47
Cabaran!!.....	52
Objektif 6: Apa Lagi Boleh Dilakukan Ketika Di Persimpangan?	54
Pengenalan <i>Path Finder Tank</i> dan Mekanismanya.	54
Langkah-langkah susunan blok.....	55
Cabaran!!.....	60
Objektif 7: Salah Jalan? Buat Pusingan-U	62
Pengenalan <i>Turn at Centre</i> dan Mekanismanya.....	62
Langkah-langkah susunan blok.....	63
Cabaran!!.....	65
Objektif 8: Ayuh kawal Mikrobotik	66
Pengenalan Bluetooth dan Mekanismanya.....	66

Langkah-langkah susunan blok.....	67
Penggunaan Peranti Pintar Mikrobotik.....	72
Cabaran!!.....	74
Objektif 9: Kita Perlukan Peronda Kawasan !	75
Susun Atur Stretegi dan Teknik Pergerakan.....	75
Langkah-langkah susunan blok.....	78
Cabaran!!.....	81
Objektif 10: Mari Mencari Harta Tersembunyi.	83
Pengenalan Butang Tekan / Suis dan Mekanismanya.....	84
Susun Atur Stretegi dan Teknik Pergerakan.....	84
Langkah-langkah susunan blok.....	88
Cabaran!!.....	92
Objektif 11: Pengasingan Bahan Kitar Semula.	93
Pengenalan Pencengkam Tunggal dan Mekanismanya.....	94
Susun Atur Strategi dan Teknik Pergerakan.....	98
Langkah-langkah susunan blok.....	103

Maklumat Tambahan.....	108
Cabaran!!.....	109
Objektif 12: Penyimpanan Cekap Ruang.....	110
Pengenalan Pencengkam Berganda dan Mekanismanya.....	111
Pemasangan Pencengkam Berganda pada robot.....	112
Susun Atur Strategi dan Teknik Pergerakan.....	115
Langkah-langkah susunan blok.....	121
Cabaran!!.....	129
Objektif 13: Berhenti! Halangan di Hadapan.....	130
Pengenalan sensor ultrasonik dan Mekanismanya.....	130
Pemasangan sensor ultrasonik pada robot.....	131
Susun Atur Strategi dan Teknik Pergerakan.....	133
Langkah-langkah susunan blok.....	134
Cabaran!!.....	138
Objektif 14: Garisan berbeza warna? Jom selesaikan!.....	139
Pengenalan Pergerakan dan Mekanismanya.....	139

Langkah-langkah susunan blok.....	143
Objektif 15: Pengecaman Bahan Kitar Semula Automatik.....	149
Pengenalan ESP32-Camera dan Mekanismenya.....	149
Menambah peranti ESP32-CAM pada Arduino IDE.....	151
Pengumpulan Data menggunakan ESP32-CAM.....	153
Penggunaan EdgeImpulse untuk Membangun Model Kecerdasan Buatan (AI).....	162
Muat turun data ke dalam ESP32-CAM.....	181
Langkah – langkah susunan blok.....	185
Tambahan: Cuba Naik Taraf dan Pengaturcaraan Sendiri.....	190

MIKROBOTIK



Struktur Mekanikal

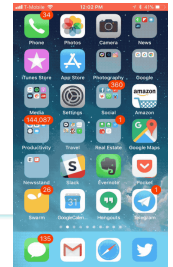


Pergerakan mekanikal

Elemen Pada Robotik



Perkakasan Elektronik



Pengaturcaraan perisian

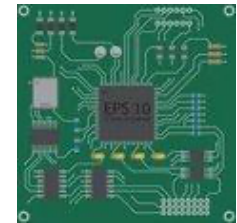
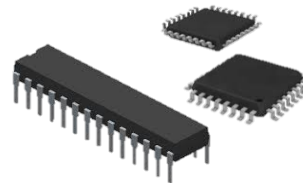
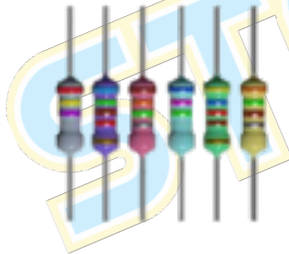
Apa itu perkakasan elektronik?



Mengesan dan merasa pada persekitaran



Mengawal atau bertindak balas pada persekitaran

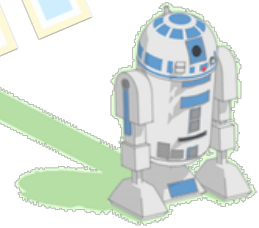


Apa itu pengaturcaraaan perisian?

Pengawal Perkakasan
Elektronik



Set arahan ditulis
menggunakan
bahasa tertentu

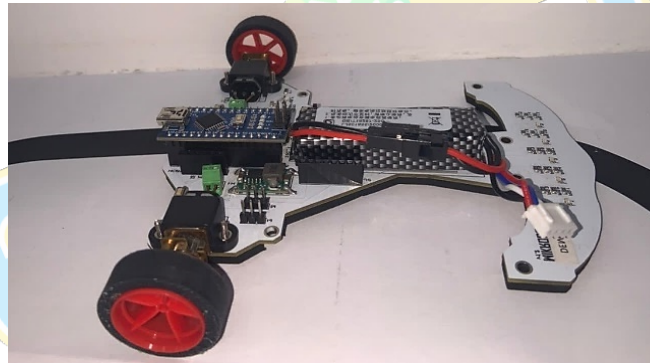


Berfungsi



Robot Berautonomi

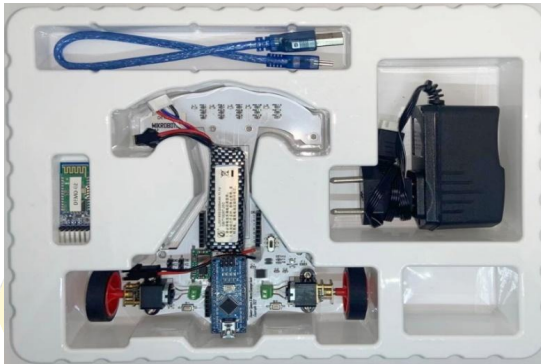
Robot direka dan dibina khas untuk mengesan dan bergerak secara automatik atau berautonomi mengikut garis putih dan hitam. Selain itu, robot juga direka untuk fungsi lain. Sebagai contoh, mengesan halangan dan menggerakkan objek.



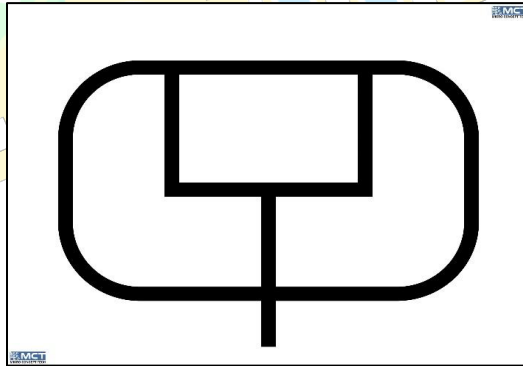
Gambar 1: "Mikrobotik" Robot Berautonomi

Kandungan di dalam kotak

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

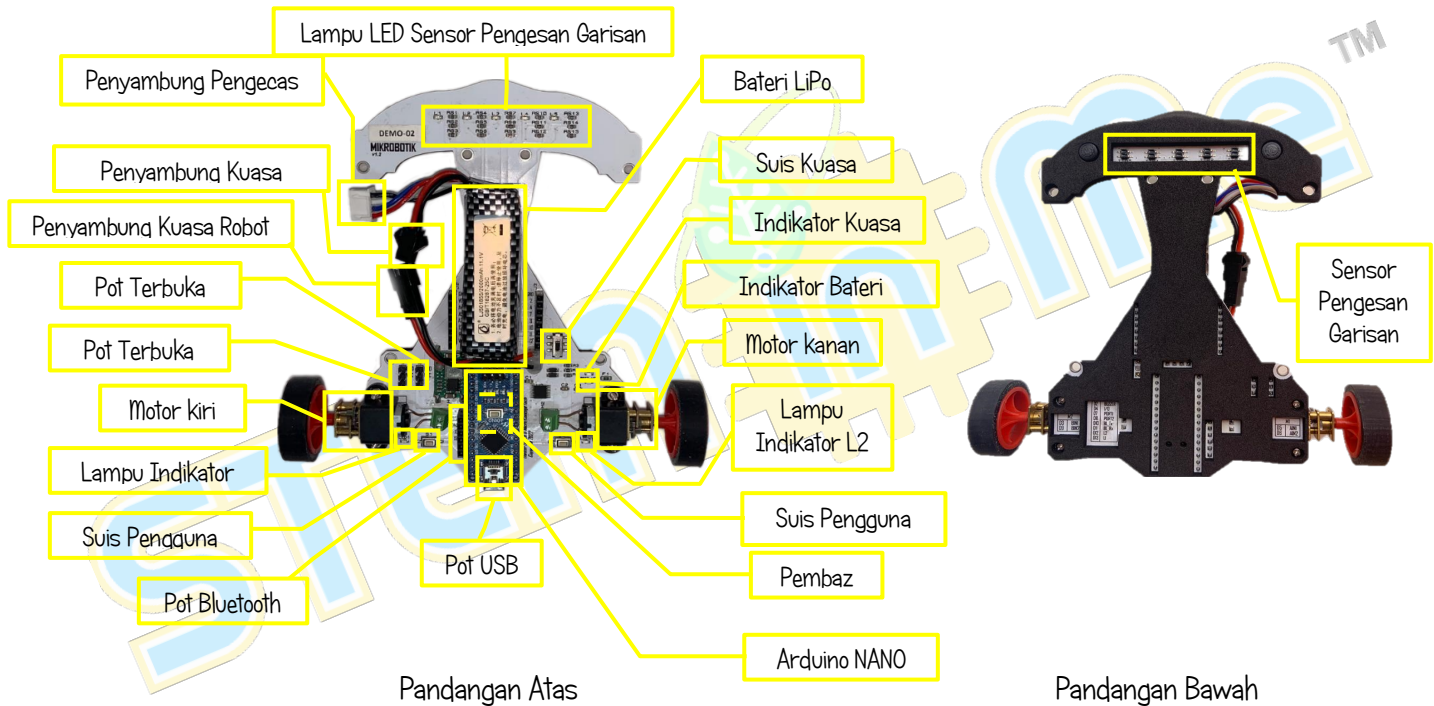


Gambar 2: Set Mikrobotik

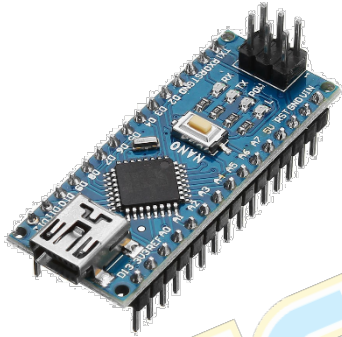


Gambar 3: Litar Mikrobotik

“Mikrobotik” Robot Berautonomi

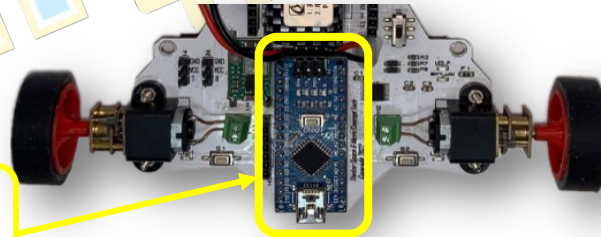


Mikropengawal Arduino Nano



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.



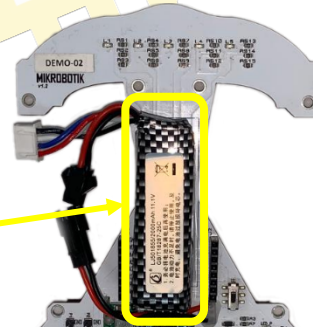
Mikropengawal Arduino Nano pada Mikrobotik

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.



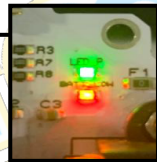
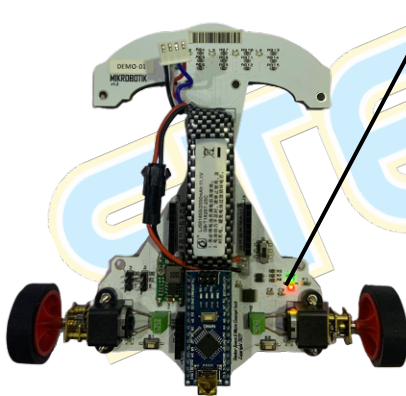
Bateri LiPo pada Mikrobotik

Indikator Bateri Rendah

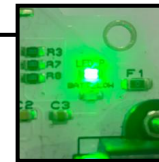
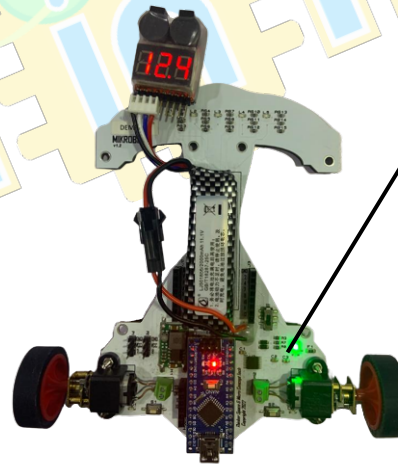
Indikator bateri rendah akan menyala warna merah.
Semakin rendah nilai voltan dalam bateri, semakin terang indikator menyala.
Voltan operasi minimum: 11.0 V (Indikator bateri rendah di kecerahan maksima)



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



Indikator bateri rendah ketika bateri rendah.



Indikator bateri rendah ketika bateri penuh.

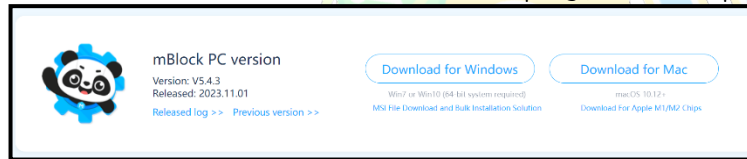
Pemasangan perisian mBlock v5

Langkah 1 Perisian mBlock v5 boleh didapatkan daripada:

Link: <https://mblock.cc/pages/downloads> @ QR:



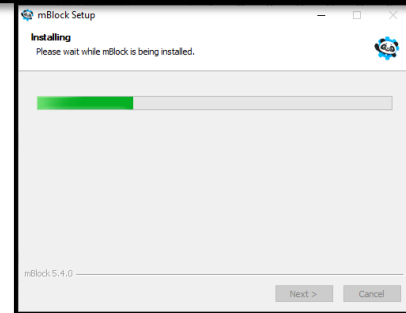
Langkah 2 Muat turun versi terkini mBlock v5 berdasarkan sistem pengendalian komputer.



Langkah 3 Klik mBlock v5 pada lokasi muat turun anda.

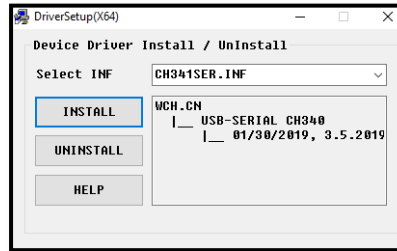


Langkah 4 Tunggu sehingga pemasangan mBlock v5 selesai.



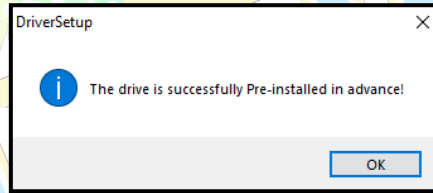
Langkah 5

Klik *INSTALL*.



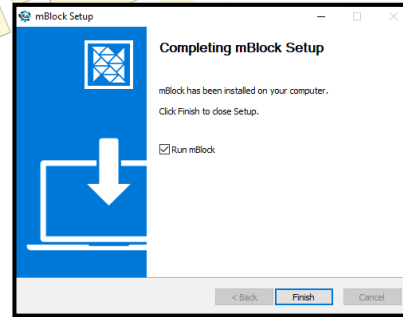
Langkah 6

Klik OK dan keluar



Langkah 7

Tandakan *Run mBlock*.
Klik *Finish*.

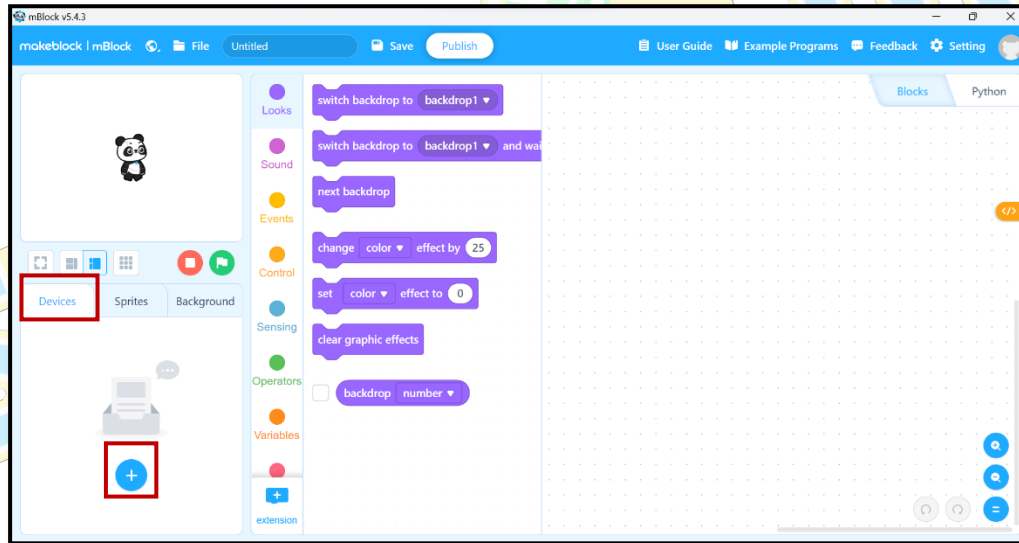


Cara untuk menambah Mikrobotik

Langkah 1 Buka mBlock v5.4.3

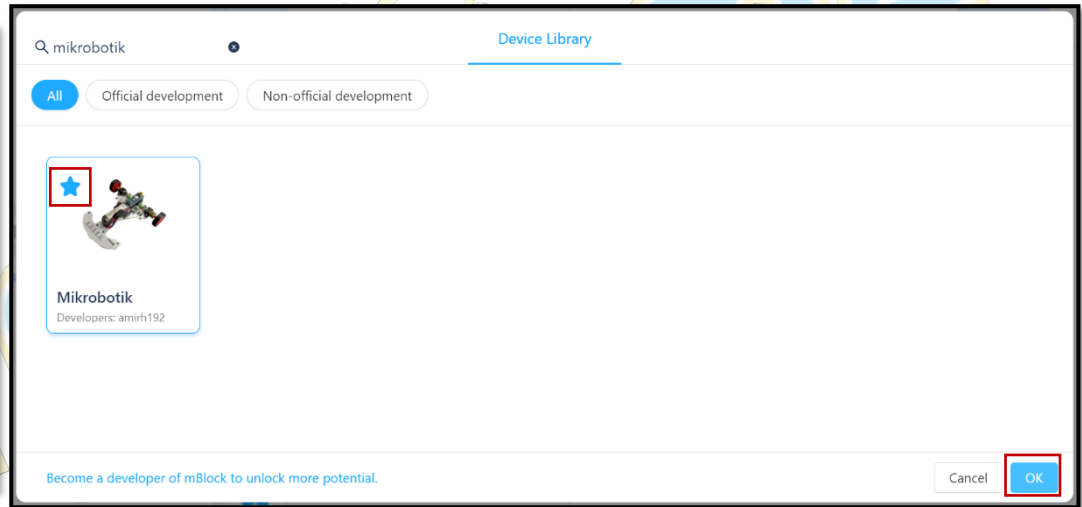
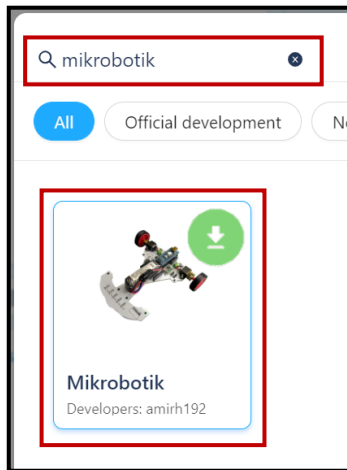


Langkah 2 Pada paparan mBlock, pilih *Devices* dan klik *Add*.



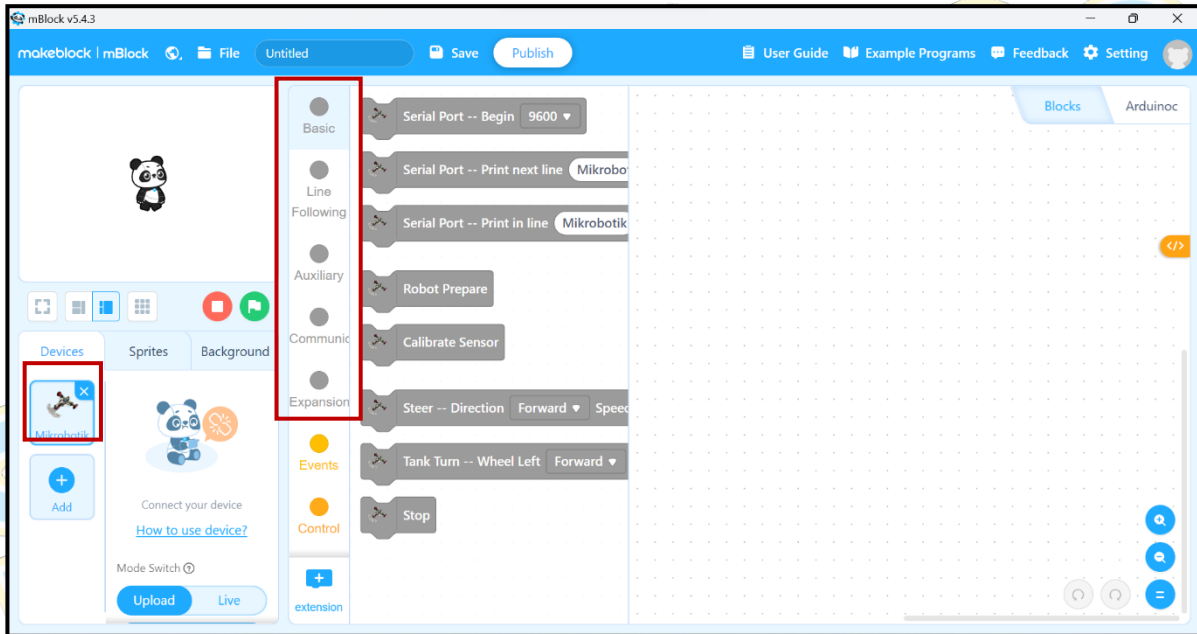
Langkah 3

Pada paparan mBlock, tulis 'Mikrobotik' pada ruangan *Search*. Muat turun Mikrobotik dan *Set as mostly used device* dan klik *OK*. Sila pastikan laptop atau komputer anda mempunyai akses internet



Langkah 4

Mikrobotik akan ditambah di bahagian *Devices*. *Library* Mikrobotik akan dipaparkan pada ruangan blok. Sekarang, anda boleh mulakan koding.



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 garisan dan menyelesaikan litar.

Susunan blok (Kalibrasi Automatik):



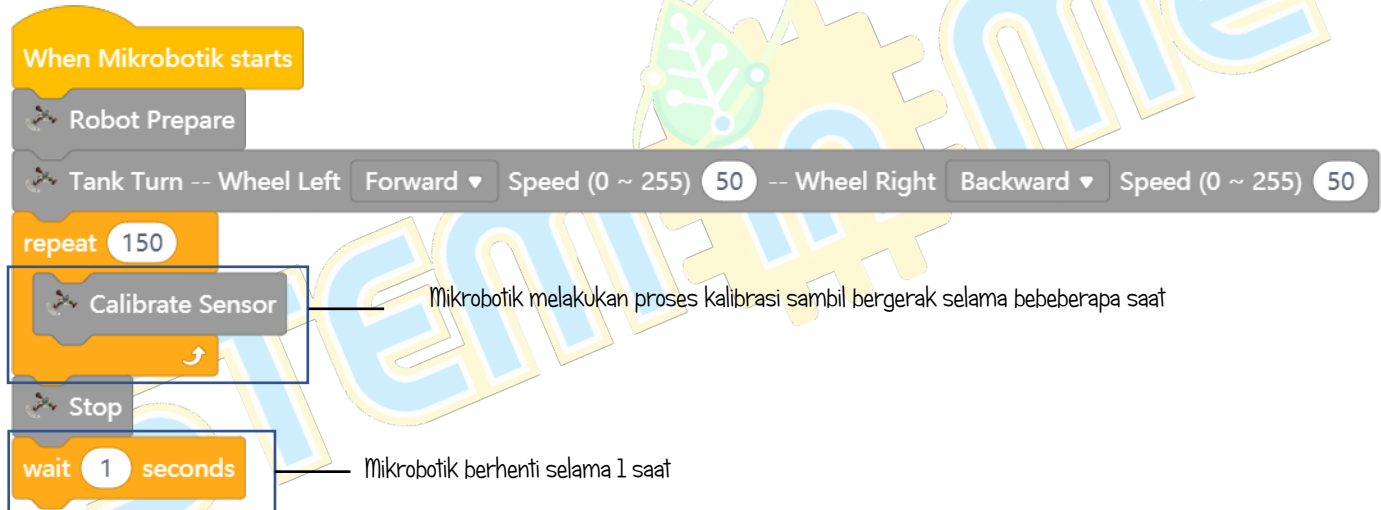
MIKROBOTIK

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 robot named Mikrobotik. The script starts with a yellow 'When Mikrobotik starts' block. Below it is a grey 'Robot Prepare' block. The next block is a grey 'Tank Turn -- Wheel Left' block with 'Forward' selected in a dropdown menu, a speed of 50, and 'Wheel Right' selected in another dropdown menu with 'Backward' selected in its dropdown menu and a speed of 50. This is followed by an orange 'repeat' block set to 150 iterations. Inside the repeat loop, there are three blocks: a grey 'Calibrate Sensor' block, a grey 'Stop' block, and an orange 'wait 1 seconds' block. Two lines of text with arrows point to the 'Calibrate Sensor' and 'wait 1 seconds' blocks, providing explanations in Indonesian.

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 melakukan proses kalibrasi sambil bergerak selama beberapa saat

Mikrobotik berhenti selama 1 saat

Langkah Proses Kalibrasi Automatik

Langkah 1

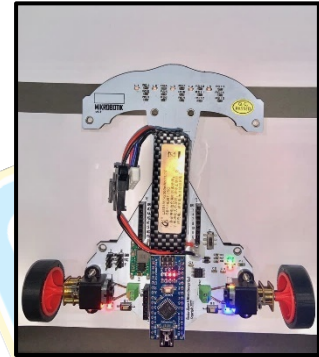
Letakkan Mikrobotik di atas litar.

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

Langkah 2

Hidupkan suis Mikrobotik.

Robot akan berpusing secara automatik untuk menjalankan proses kalibrasi.



STEM in

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
```

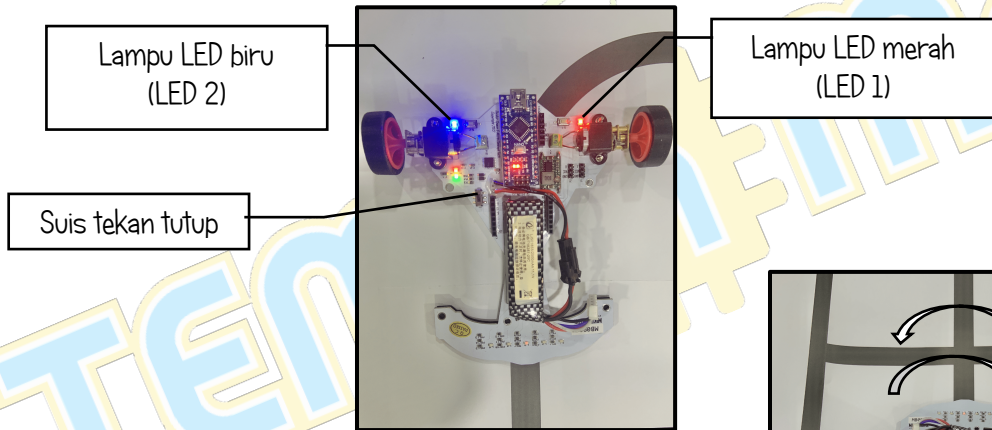


Langkah Proses Kalibrasi Manual

Langkah 1

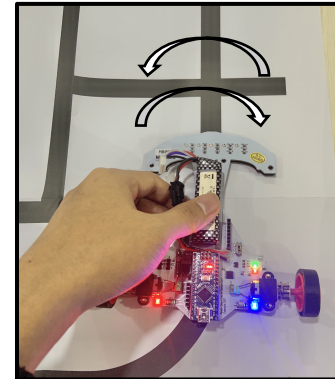
Hidupkan suis Mikrobotik.

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.

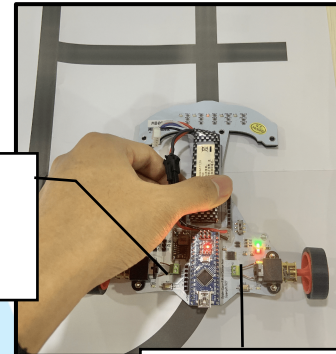


Langkah 3 Ulangi pergerakan di Langkah 2 sehingga lampu LED1 dan LED2 terpadam



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.

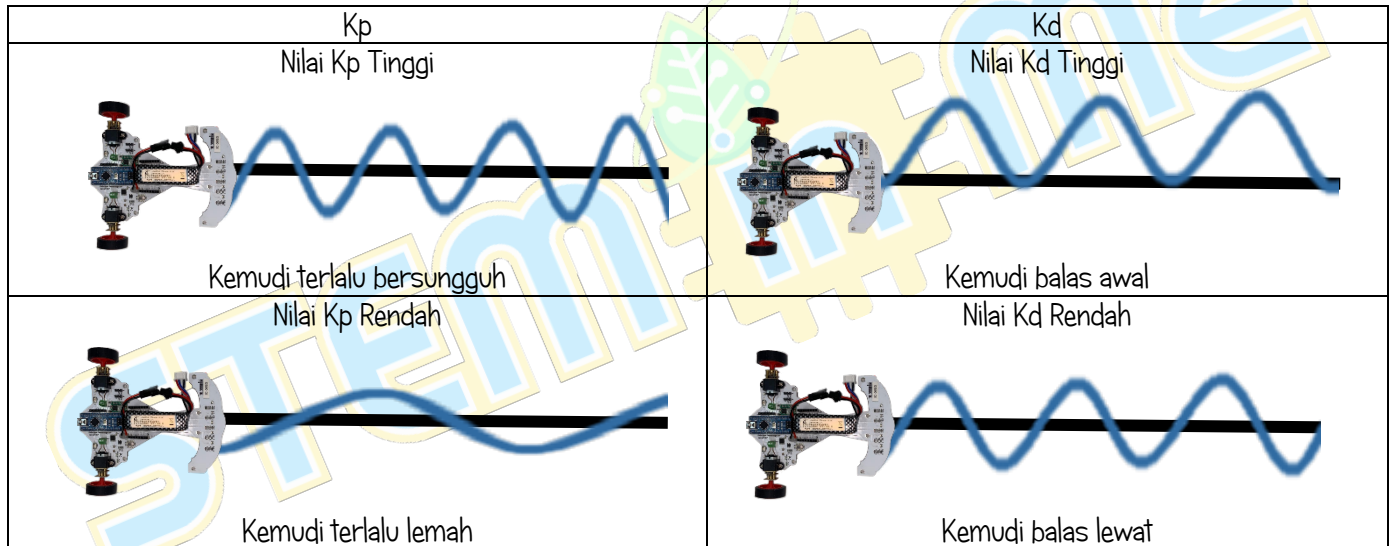
Lampu LED merah (LED 1) terpadam



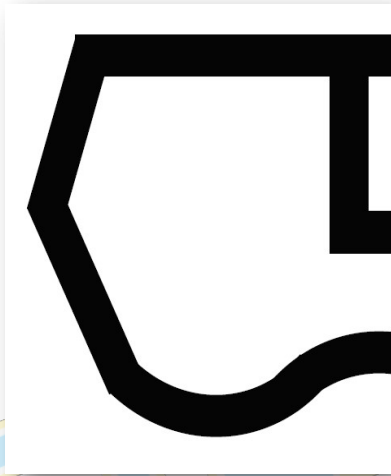
Lampu LED biru (LED 2) terpadam

Algoritma PID Robot Berautonomi

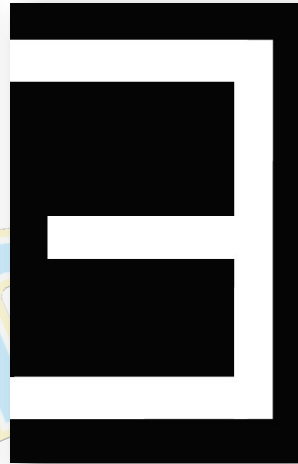
Algoritma PID adalah strategi kawalan robot berautonomi yang sesuai untuk membantu menentukan arah kemudi dan kelajuan robot yang bergerak secara automatik mengikut garisan. Algoritma PID akan memastikan robot tidak tersasar dari litar ketika membelok dan bergerak lurus mengikut garisan.



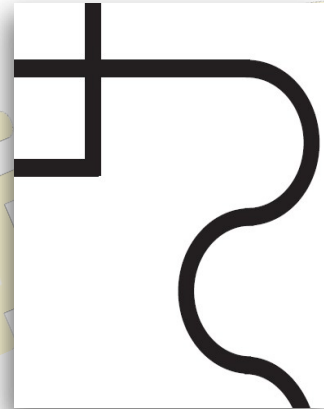
Apa jenis-jenis Litar?



Garisan Hitam
(Anggaran 20mm)

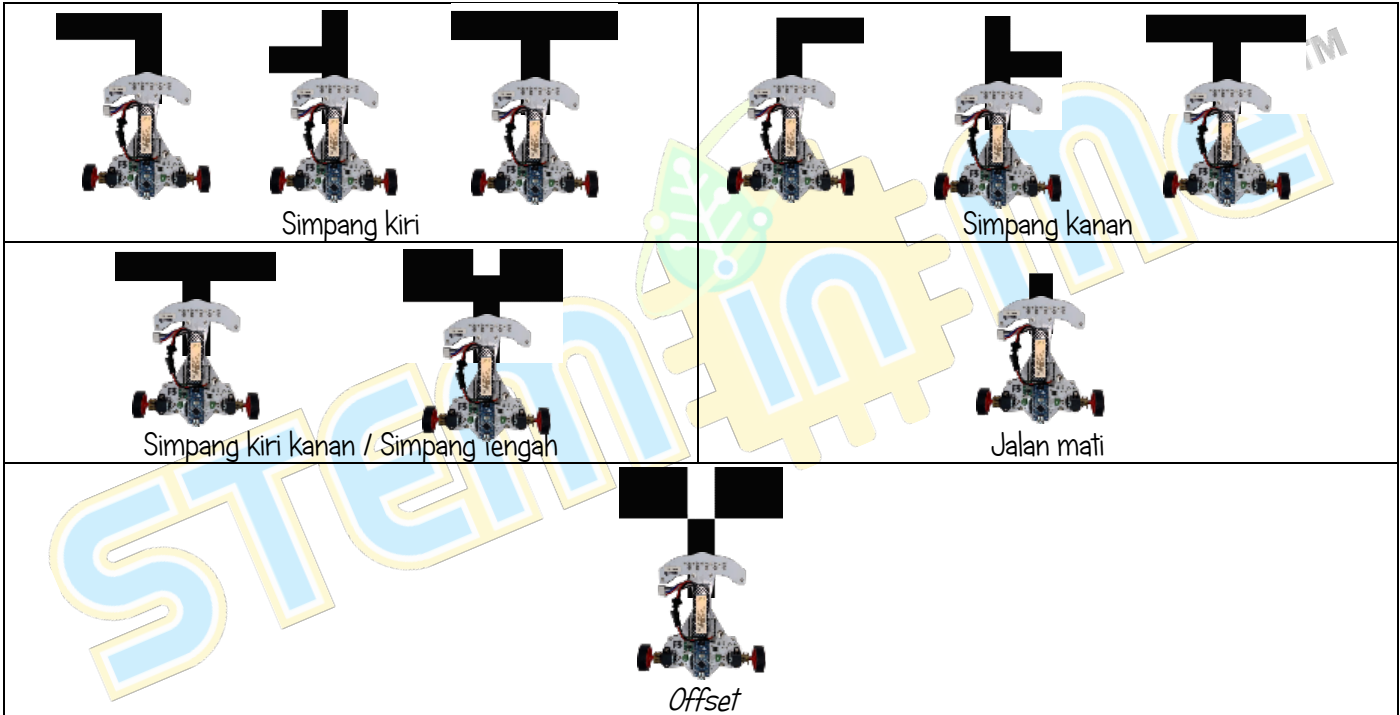


Garisan Putih
(Anggaran 20mm)



Garisan Hitam Nipis
(Anggaran 10mm)

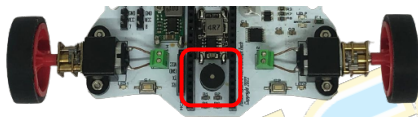
Jenis-jenis Persimpangan



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*.

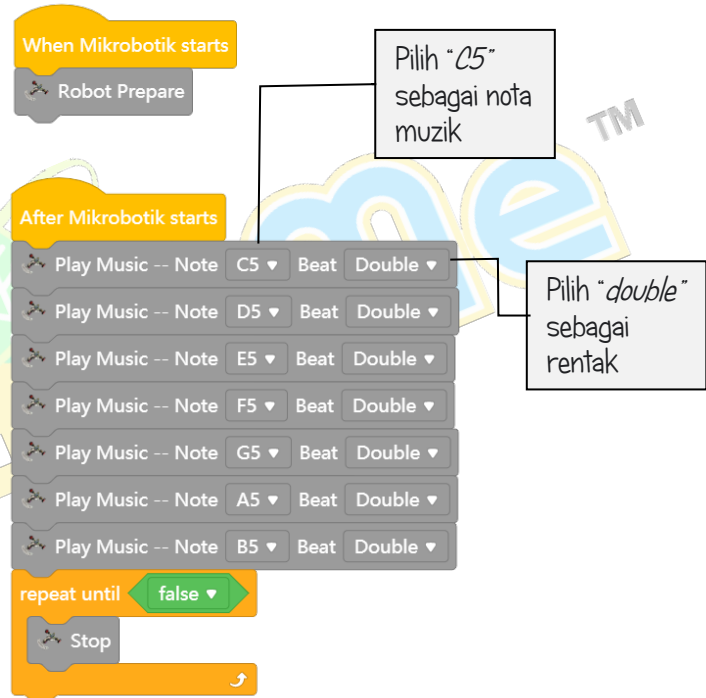
When Mikrobotik starts

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

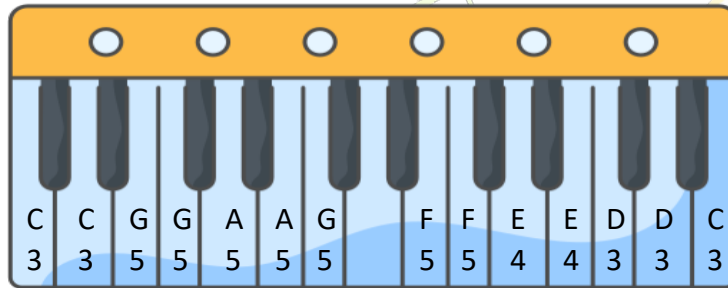
Seterusnya, gabungkan blok *repeat until (false)* dengan blok *stop*. Gabungkan blok tersebut dengan blok di Langkah 2.

Langkah 4

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

Cabaran !!

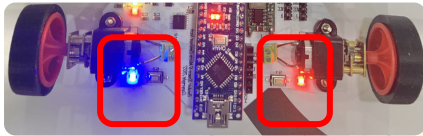
Dalam cabaran ini, anda perlu memasukkan not muzik yang disediakan dan cuba untuk meneka nama muzik yang dihasilkan.



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.

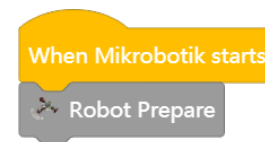
Pengenalan Mudah Diod Pemancar Cahaya (LED)



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

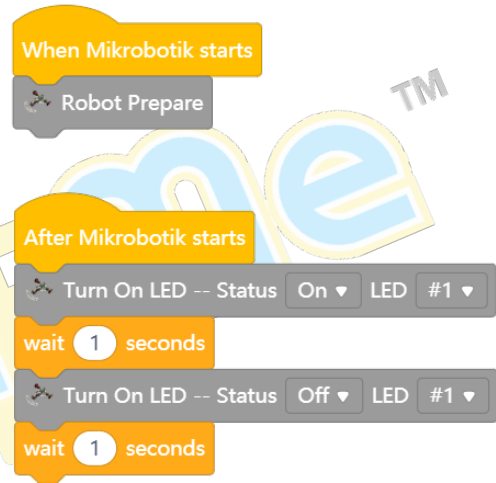
Langkah-langkah susunan blok

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



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.



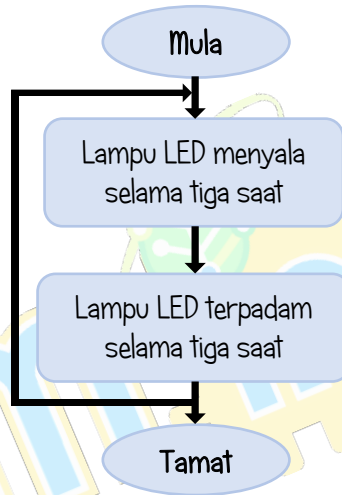
Langkah 3

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 mematikan 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!!

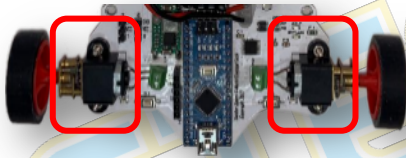


Program di atas akan menyala selama tiga saat dan akan terpadam selama tiga saat. Program ini akan berterusan sehingga Mikrobotik dimatikan.

Objektif 3: Mulakan Pengembaraan Kita (Pergerakan Bebas)

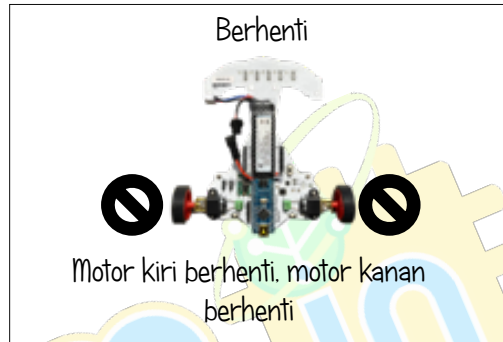
Robot digerakkan menggunakan kod blok "*tank turn*" untuk bergerak tanpa mengikuti garisan. Kod blok ini sesuai digunakan untuk menyelesaikan litar labirin (*maze*). Robot akan bergerak bergantung kepada kelajuan serta arah motor kiri dan kanan yang ditetapkan.


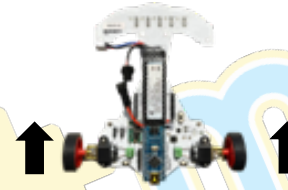
Pengenalan kepada Motor

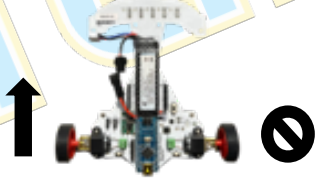



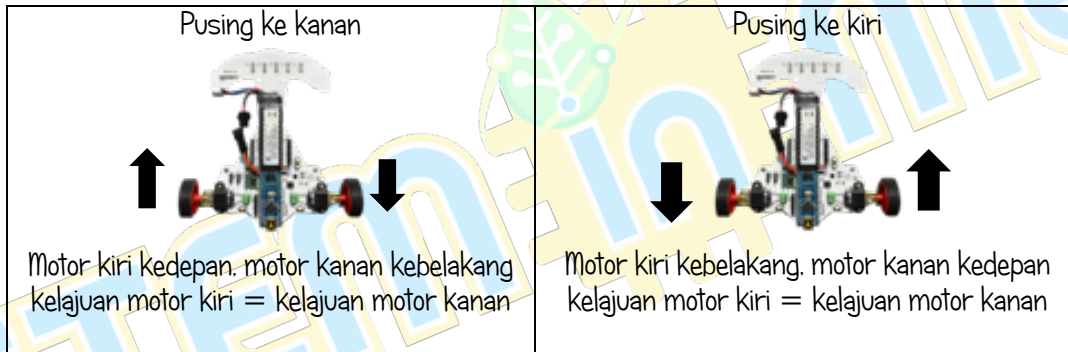
Terdapat 2 motor pada MikroBotik boleh dikawal secara berasingan. boleh berputar mengikut putaran arah jam dan lawan jam secara berterusan. Motor ini juga boleh digunakan untuk menggerakkan atau memacu projek. Kelajuan dan jangka masa juga boleh ditetapkan.

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*.

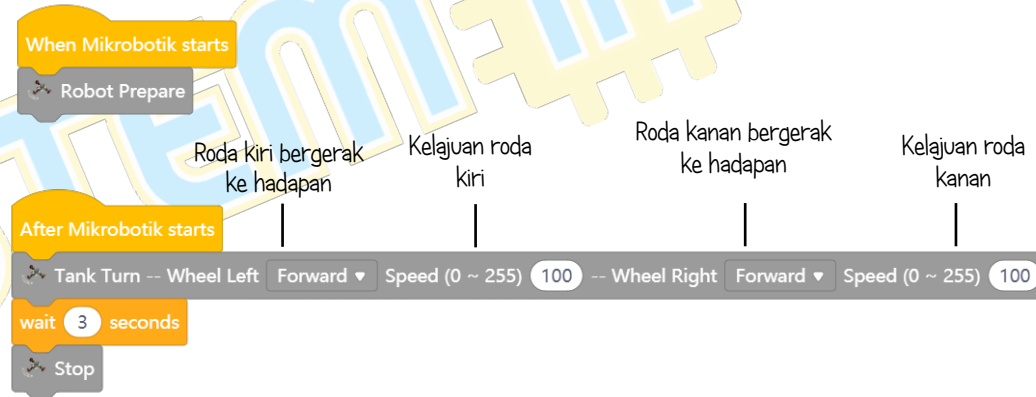
When Mikrobotik starts

Robot Prepare

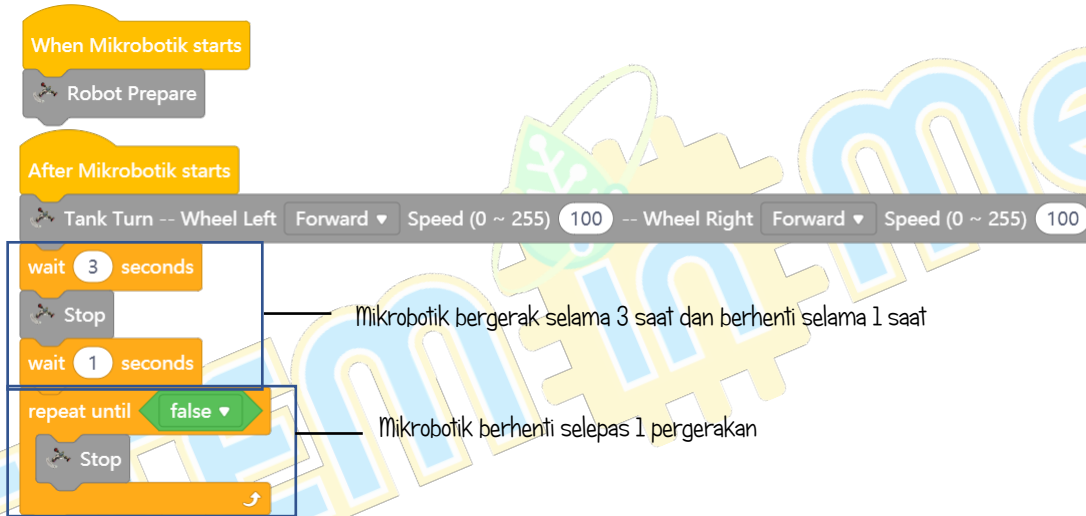
Langkah 2

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.

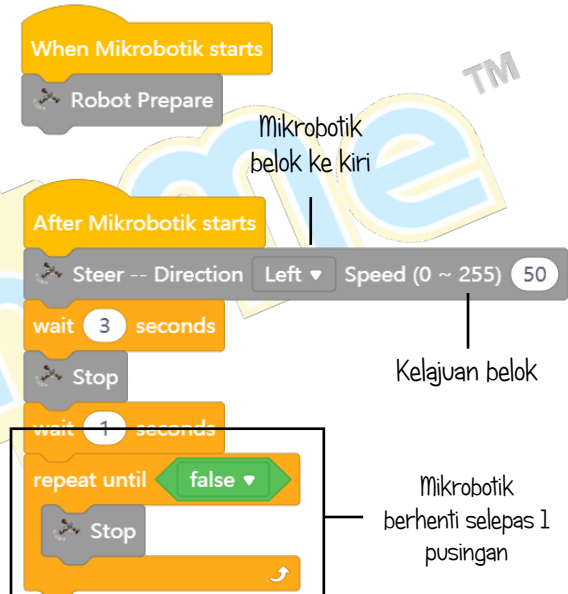
ii) Belok ke kiri

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

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.

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

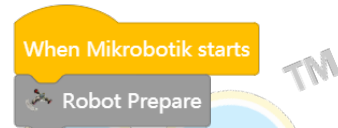
Langkah 4 Selepas memuat naik program. Mikrobotik belok ke kiri selama 3 saat dan berhenti.



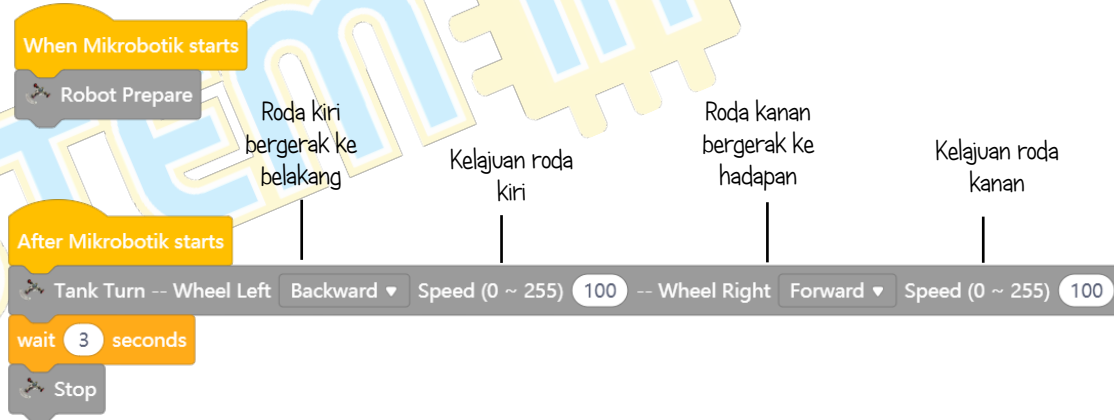
Langkah untuk belok ke kanan adalah serupa dengan langkah untuk belok ke kiri. Anda hanya perlu menukar *Direction* kepada right.

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.

When Mikrobotik starts

Robot Prepare

After Mikrobotik starts

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

wait 3 seconds

Stop

wait 1 seconds

repeat until false

Stop

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 diagram shows a sequence of blocks for an automatic calibration routine:

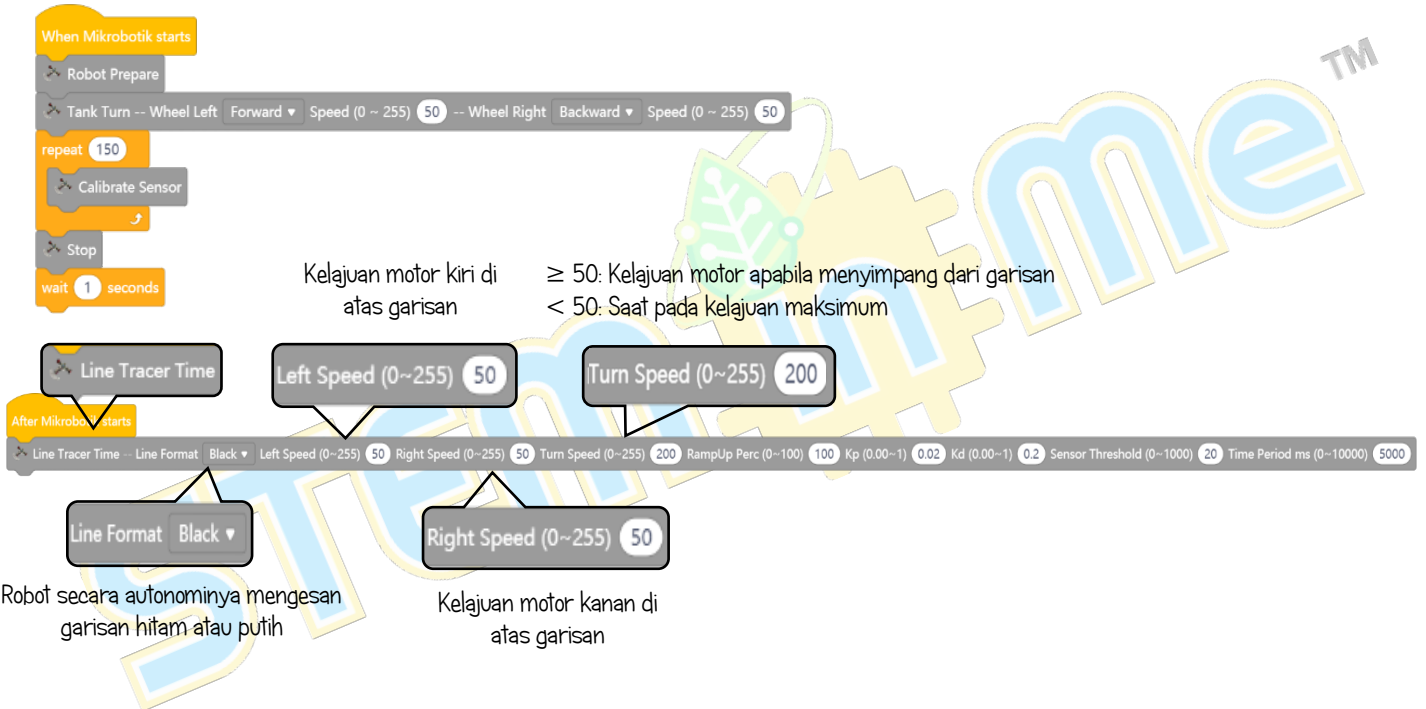
- When Mikrobotik starts** (Yellow block)
- Robot Prepare** (Grey block)
- Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 50 -- Wheel Right Backward Speed (0 ~ 255) 50** (Grey block)
- repeat 150** (Orange block) containing:
 - Calibrate Sensor** (Grey block)
- Stop** (Grey block)
- wait 1 seconds** (Orange block)

Annotations:

- A line points from the **Calibrate Sensor** block to the text: "Mikrobotik melakukan proses kalibrasi sambil bergerak selama beberapa saat"
- A line points from the **wait 1 seconds** block to the text: "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.



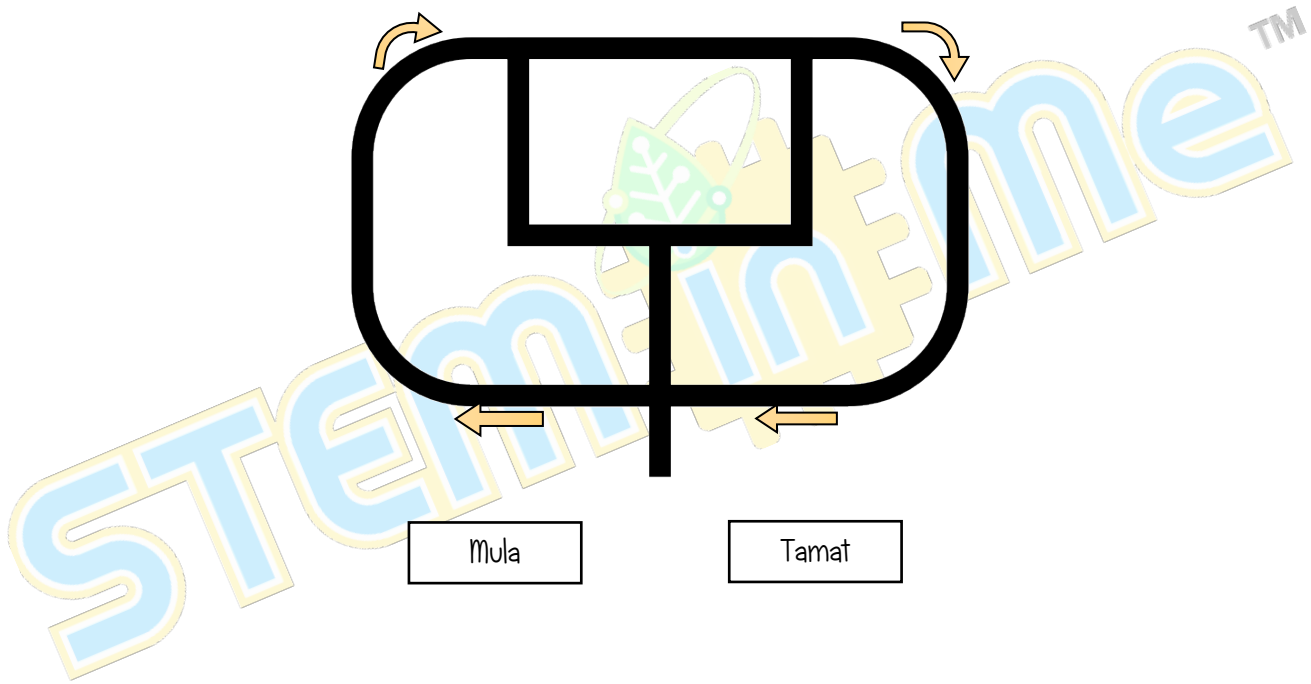
The screenshot shows a programming environment for a microbot. At the top, there is a status bar with various parameters: Line Tracer Time, Line Format (Black), Left Speed (0-255) set to 50, Right Speed (0-255) set to 50, Turn Speed (0-255) set to 200, RampUp Perc (0-100) set to 100, Kp (0.00-1) set to 0.02, Kd (0.00-1) set to 0.2, Sensor Threshold (0-1000) set to 30, and Time Period ms (0-10000) set to 5000. Below the status bar, a code block is visible, consisting of a 'repeat until false' block with a dropdown arrow, and a 'stop' block connected to its bottom. A line points from the text 'Mikrobotik berhenti selepas 1 pusingan' to the 'stop' block.

Langkah 4

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

Cabaran!!

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



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

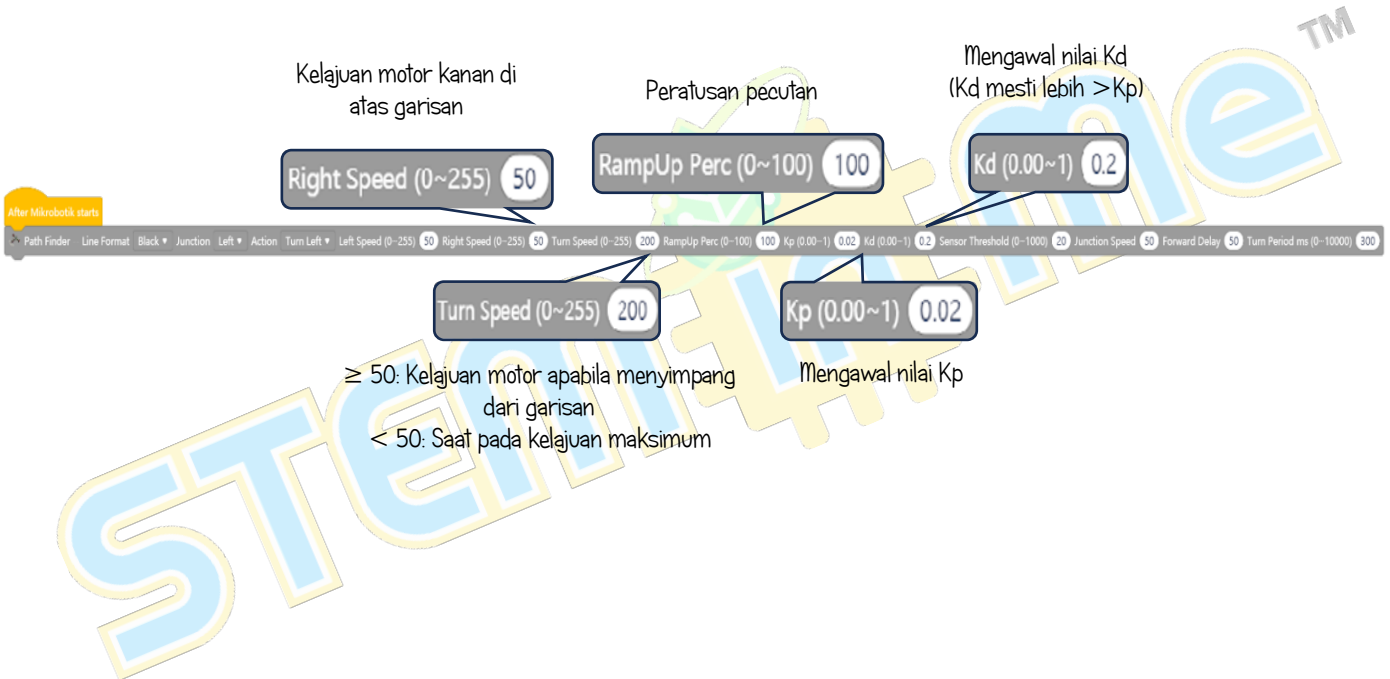
After Mikrobotik starts

- 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

Robot menjejak ke hadapan sehingga menemui persimpangan

Kelajuan motor kiri di atas garisan

Sambungan



Sambungan

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

Masa untuk robot terus bergerak ke hadapan sebelum membelok

Kepekaan garisan (lebih tinggi nilai bermaksud lebih sensitif)

Masa tepat (ms) robot membelok atau sehingga menjumpai narisian (0)

Alter 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

Junction Speed 50

Forward Delay 50

Sensor Threshold (0~1000) 20

Turn Period ms (0~10000) 300

Langkah 3

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

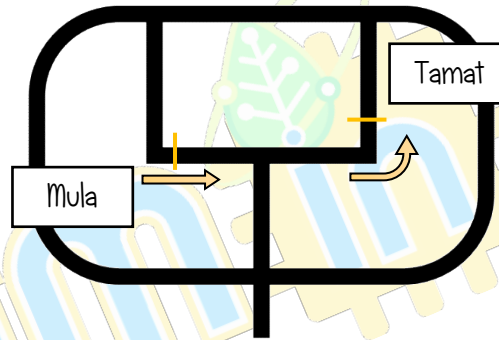


Langkah 4

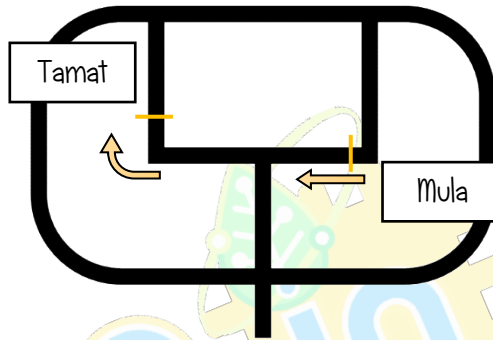
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi. Selepas itu, Mikrobotik akan mengikuti garisan hitam dan jika robot menemui persimpangan kiri, Mikrobotik akan bergerak ke hadapan dan kemudian belok memasuki simpang kiri sehingga Mikrobotik menemui garisan lain.

Cabaran!!

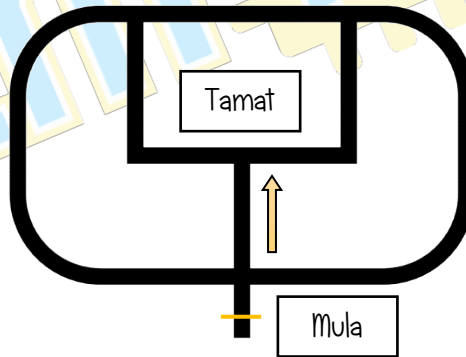
- i) *Path Finder* persimpangan kiri. belok di persimpangan kiri



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 mengikuti 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*).

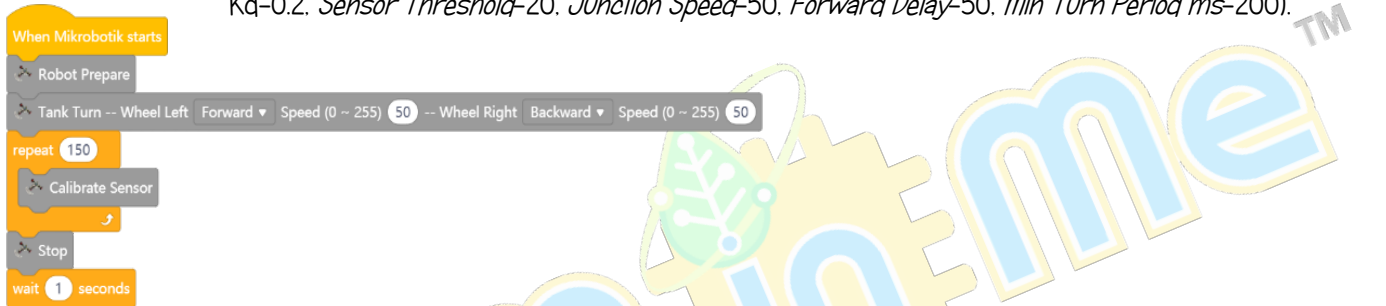
Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



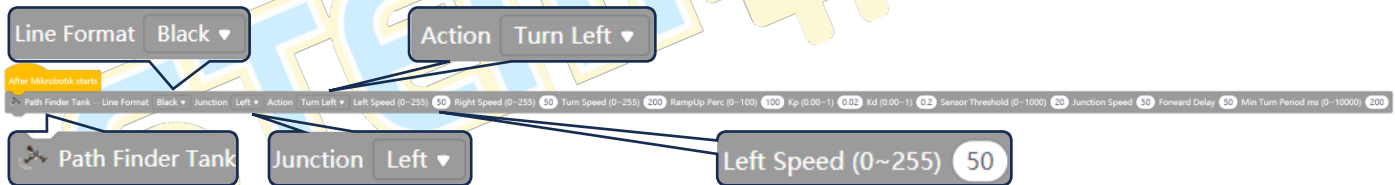
Langkah 2

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

Alter Mikrobook 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

Kelajuan motor kanan di atas garisan

Peratusan pecutan

Mengawal nilai Kd (Kd mesti lebih >Kp)

≥ 50: Kelajuan motor apabila menyimpang dari garisan
 < 50: Saat pada kelajuan maksimum

Mengawal nilai Kp

Sambungan

The screenshot shows a sequence of actions in a programming environment. The actions are: 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, and Min Turn Period ms (0-10000) 200. Callouts provide explanations for Junction Speed, Forward Delay, Sensor Threshold, and Min Turn Period.

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

Min Turn Period ms (0~10000) 200

Kepekaan garisan
(lebih tinggi nilai bermaksud lebih sensitif)

Masa minimum (ms) robot berpusing

Langkah 3

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

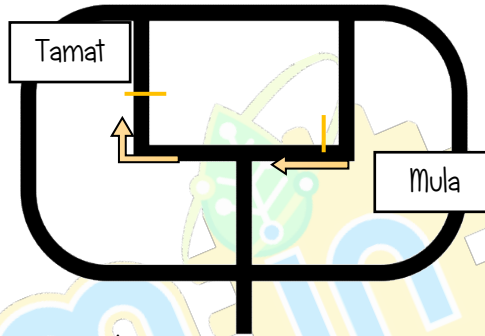


Langkah 4

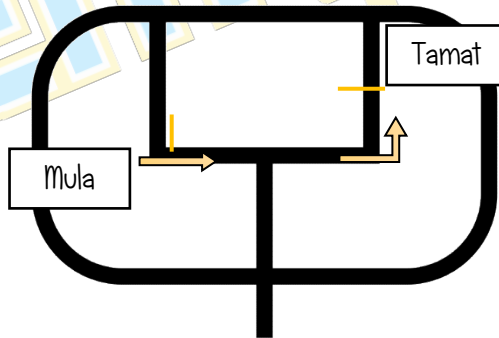
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi. Selepas itu, Mikrobotik akan mengikuti garisan hitam dan jika robot menemui persimpangan kiri, Mikrobotik akan bergerak ke hadapan dan kemudian pusing untuk sekurang-kurangnya Durasi Minimum Pusingan (*Min Turn Period*) dan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

Cabaran!!

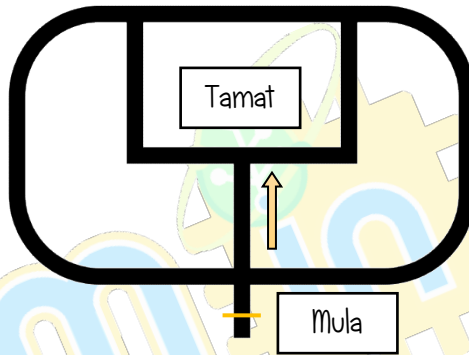
- i) *Path Finder Tank* persimpangan kanan. pusing di persimpangan kanan



- 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.

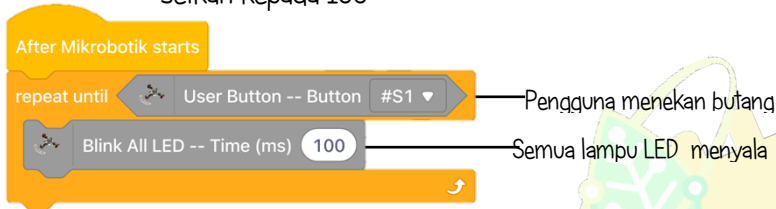
Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



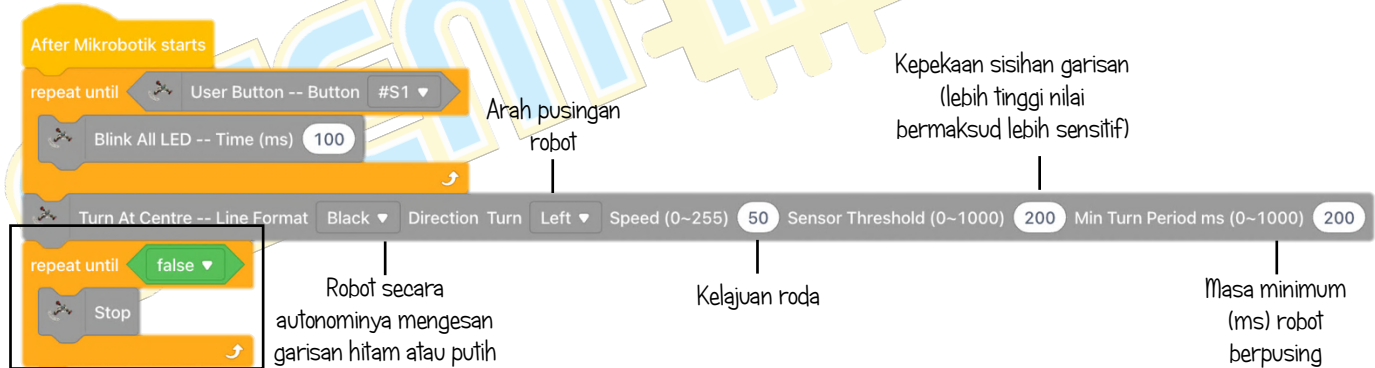
Langkah 2

Seretkan blok *After Mikrobotik starts* dan gabungkan blok *repeat until*. Seterusnya masukkan blok *Used Button – Button* dan setkan kepada #S. Seterusnya gabungkan blok *Blink All LED – Time (ms)* dan setkan kepada 100



Langkah 3

Gabungkan blok *Turn At Centre* setkan *Line Format – Black*, *Direction Turn – Left*, *Speed – 50*, *Sensor Threshold – 200*, *Min Turn Period – 200*. Seterusnya, gabungkan blok *Repeat until (False)* dan blok *Stop*



Mikrobotik berhenti selepas 1 pusingan

Langkah 3

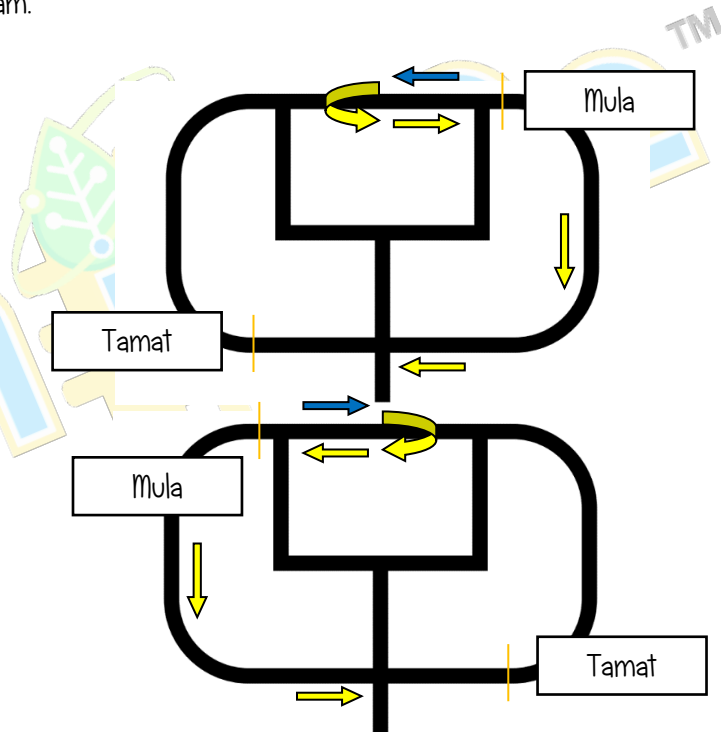
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi.

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.

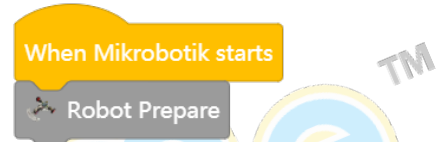
Pengenalan Bluetooth dan Mekanismanya



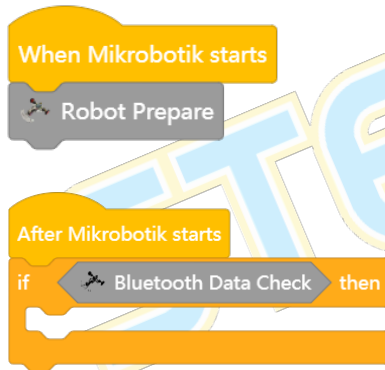
Mikrobotik boleh dikawal dalam jarak dekat menggunakan pendekatan Bluetooth kerana ia senang didapati dan senang mengawalinya. Modul Bluetooth itu dimasukkan pada pot yang disediakan pada Mikrobotik. Modul Bluetooth ini mengandungi 4 kaki. RXD, TXD, GND DAN VCC.

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.

When Mikrobotik starts

- Robot Prepare

After Mikrobotik starts

- if Bluetooth Data Check then
 - if Bluetooth Data Control Data F then
 - Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 100 -- Wheel Right Forward Speed (0 ~ 255) 100
 Data untuk pergerakan maju
 - else
 - if Bluetooth Data Control Data B then
 - Tank Turn -- Wheel Left Backward Speed (0 ~ 255) -100 -- Wheel Right Backward Speed (0 ~ 255) 100
 Data untuk pergerakan undur
 - else
 - if Bluetooth Data Control Data L then
 - Tank Turn -- Wheel Left Backward Speed (0 ~ 255) 100 -- Wheel Right Forward Speed (0 ~ 255) 100
 Data untuk pusing ke kiri
 - else
 - if Bluetooth Data Control Data R then
 - Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 100 -- Wheel Right Backward Speed (0 ~ 255) 100
 Data untuk pusing ke kanan
 - else
 - if Bluetooth Data Control Data X then
 - Stop
 Data untuk 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.

The script consists of the following blocks:

- Stop**
- if** Bluetooth Data Control Data **T** **then**
 - Play Music -- Note C3 Beat Half**
 - else**
 - if** Bluetooth Data Control Data **S** **then**
 - Play Music -- Note E4 Beat Half**
 - else**
 - if** Bluetooth Data Control Data **E** **then**
 - Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 100 -- Wheel Right Forward Speed (0 ~ 255) 60**
 - else**
 - if** Bluetooth Data Control Data **Q** **then**
 - Tank Turn -- Wheel Left Forward Speed (0 ~ 255) 60 -- Wheel Right Forward Speed (0 ~ 255) 100**
 - else**
 - if** Bluetooth Data Control Data **C** **then**
 - Tank Turn -- Wheel Left Backward Speed (0 ~ 255) 100 -- Wheel Right Backward Speed (0 ~ 255) 60**
 - else**
 - if** Bluetooth Data Control Data **Z** **then**
 - Tank Turn -- Wheel Left Backward Speed (0 ~ 255) 60 -- Wheel Right Backward Speed (0 ~ 255) 100**

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:

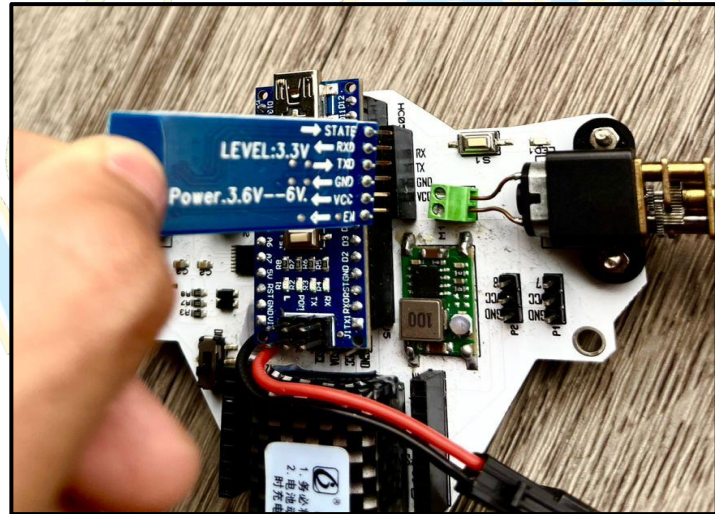
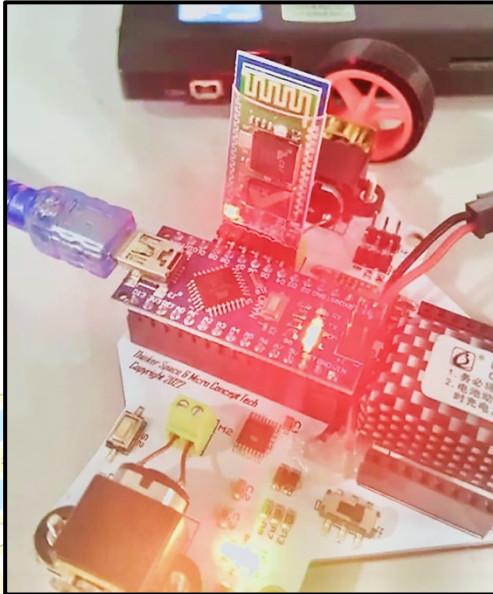
- else** block (parent)
- if** block (child of else)
 - Bluetooth Data Control Data** 1
 - then** block
 - Turn On LED** -- Status: On, LED: #1
- else** block (parent)
- if** block (child of else)
 - Bluetooth Data Control Data** 2
 - then** block
 - Turn On LED** -- Status: Off, LED: #1
- else** block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

Annotations on the right side of the script:

- 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, pasangankan 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

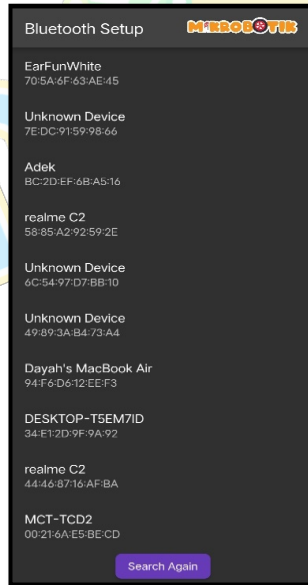
Langkah 1

Muat turun aplikasi Mikrobotik di Google Play Store: Mikrobotik



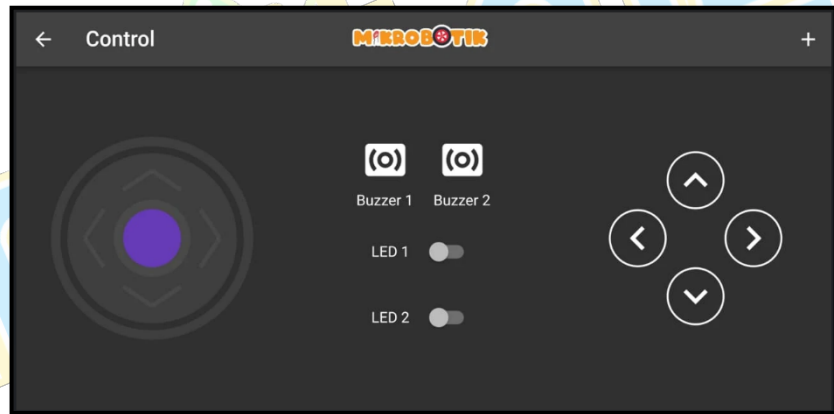
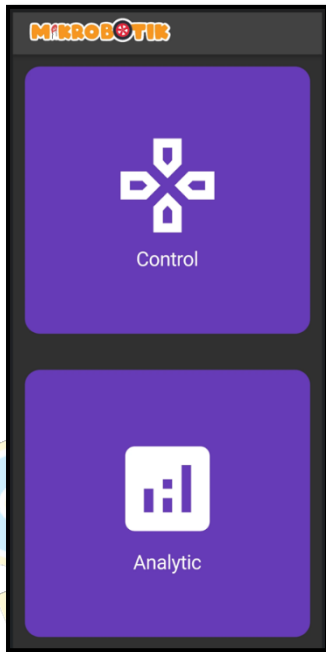
Langkah 2

Buka aplikasi. Pada "Bluetooth Setup", pilih berdasarkan nombor siri pada modul Bluetooth.



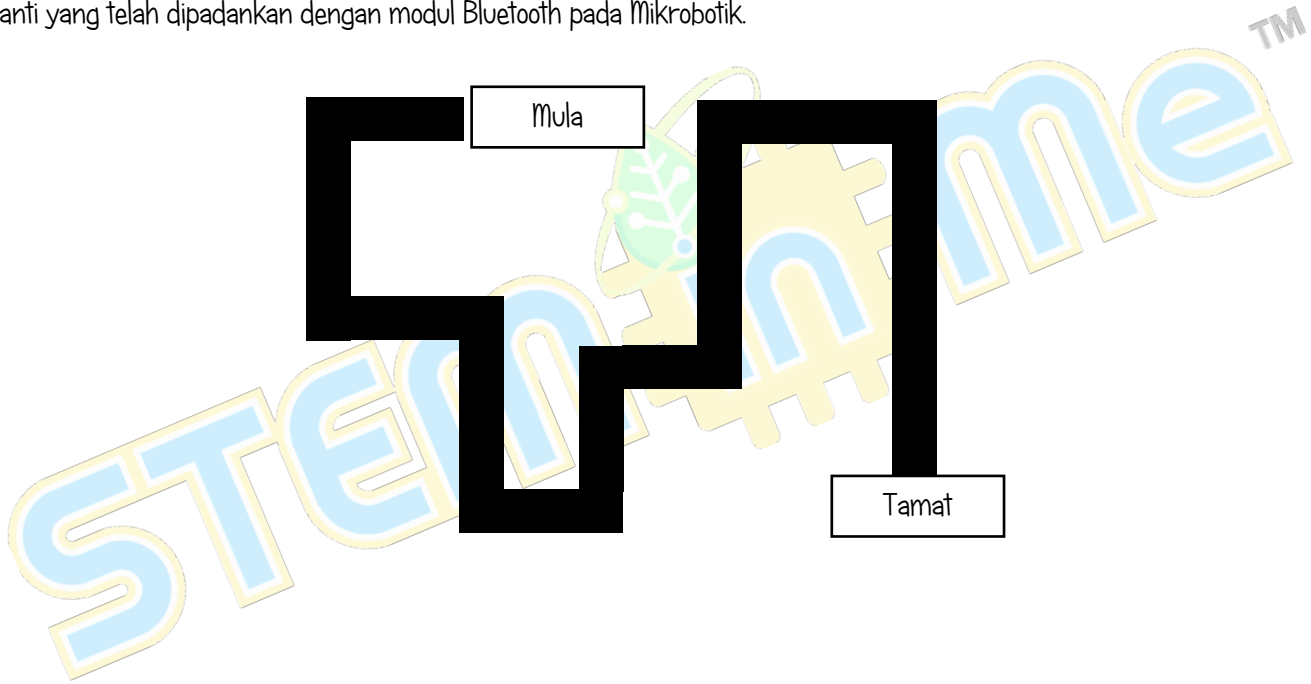
Langkah 3

Tekan "Control". 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.

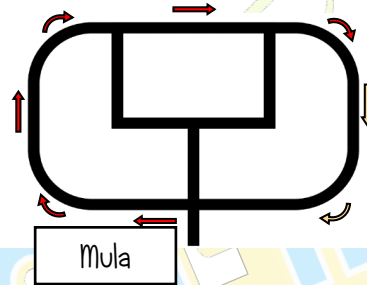
Susun Atur Strategi dan Teknik Pergerakan

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

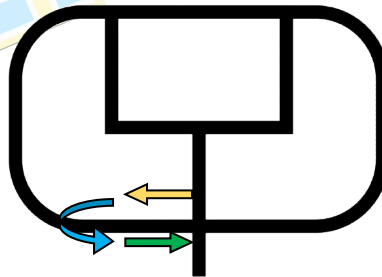
Robot bergerak secara berautonomi mengikut garisan dengan menggunakan blok *Line Tracer Time* dengan kelajuan yang tinggi dan kemudian dengan kelajuan rendah untuk masa yang tertentu. Selepas selesai berbegerak, robot membuat pusingan ke arah bertentangan dengan menggunakan blok *Turn AT Centre*. Akhirnya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

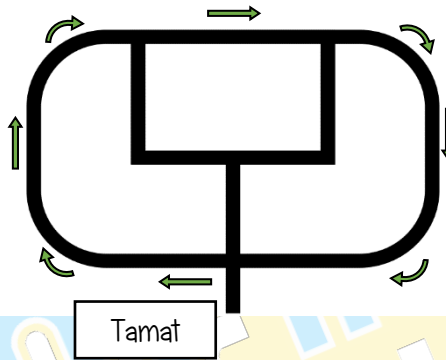
- i) Robot bergerak secara berautonomi dengan menggunakan *Line Tracer Time* dengan kelajuan tinggi selama 3 saat (anak panah merah) dan kemudian dengan kelajuan rendah selama 3 saat (anak panah kuning)



- 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.



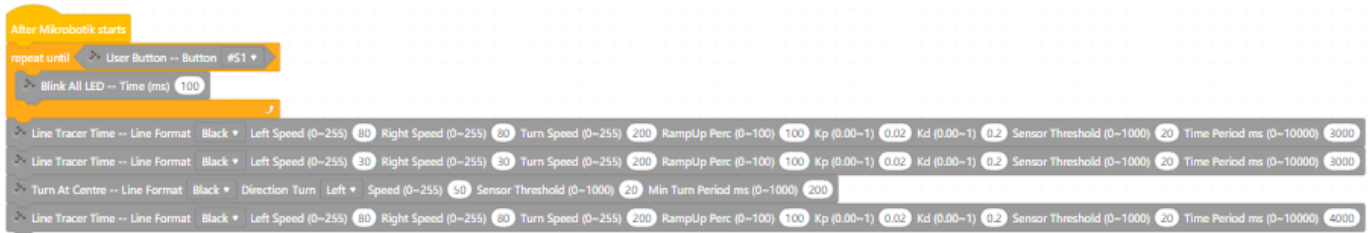
Langkah 2

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.



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.



Langkah 4

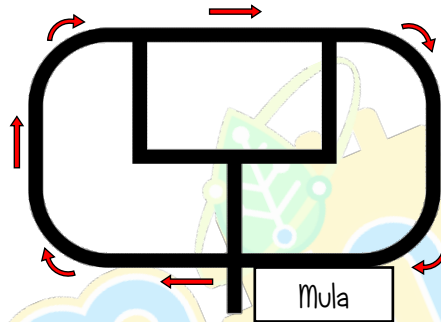
Selepas memuat naik kod, hidupkan suis Mikrobotik dan lakukan proses kalibrasi.

Selepas itu, Mikrobotik akan membuat semua pergerakan mengikut masa yang ditetapkan dan akan berhenti setelah tamat masa yang ditetapkan.

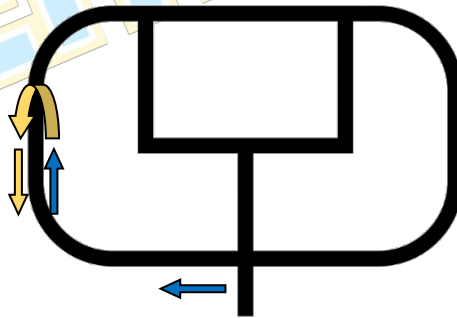


Cabaran!!

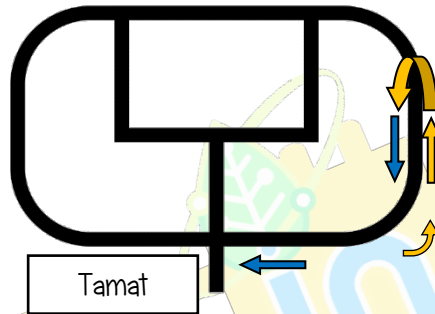
i. Robot bergerak satu pusingan



ii. Membuat pusingan di tengah

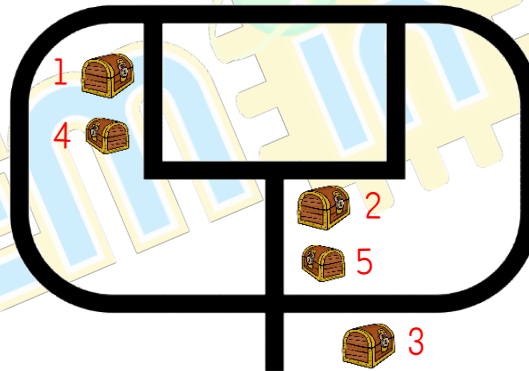


iii. Membuat pusingan lagi dan tamat

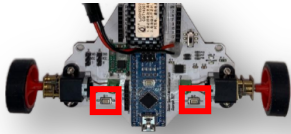


Objektif 10: Mari Mencari Harta Tersembunyi.

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.

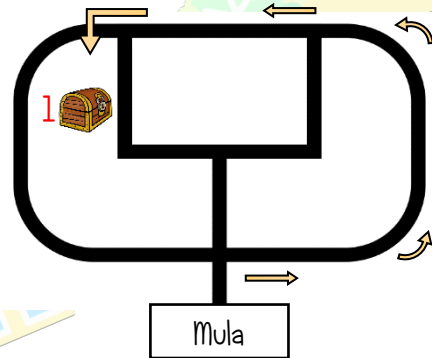
Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder* dan *Path Finder Tank*.

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.

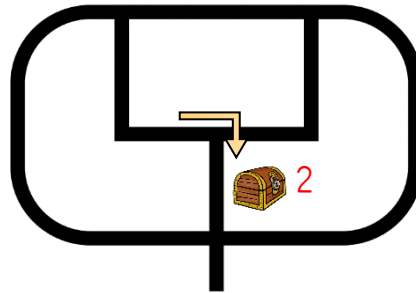
Untuk mendapatkan harta tersembunyi pertama, robot bergerak menggunakan *Path Finder* sehingga menemui persimpangan kiri dan berpusing ke arah kiri. Kemudian, robot meneruskan pergerakan menggunakan *Path Finder* sehingga menemui persimpangan kanan dan berpusing ke arah kanan untuk harta tersembunyi 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 tersembunyi dengan melalui simpang - simpang yang berbeza untuk melengkapkan tugas.

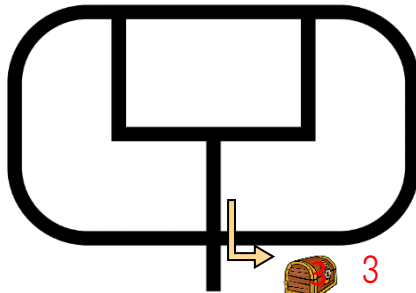
- i) Robot bergerak dari titik mula untuk mencari simpang kiri dan membelok menggunakan *Path Finder* ke arah kiri untuk mengambil harta tersembunyi pertama.



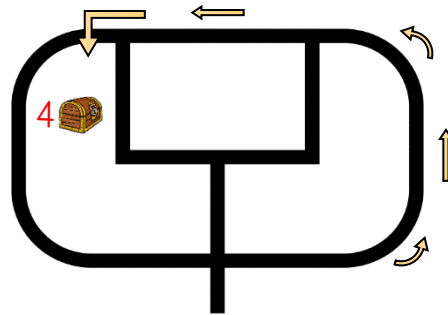
- ii) Robot bergerak mencari simpang kanan dan membelok menggunakan *Path Finder* pusingan ke arah kanan untuk mengambil harta tersembunyi yang kedua.



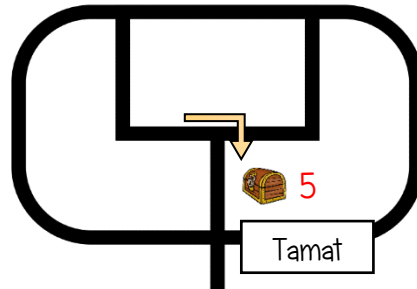
- iii) Robot bergerak mencari simpang tiga dan membelok menggunakan *Path Finder* ke arah kiri untuk mengambil harta yang tersembunyi ketiga.



- 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”**).



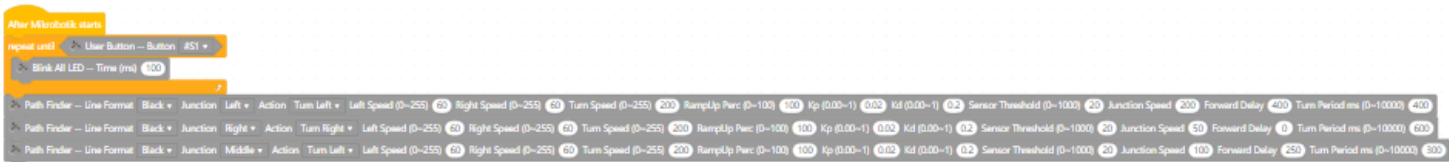
Langkah 3

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”**).



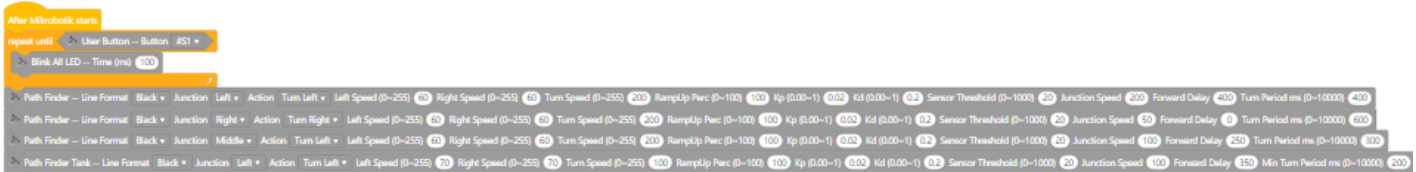
Langkah 4

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”*).



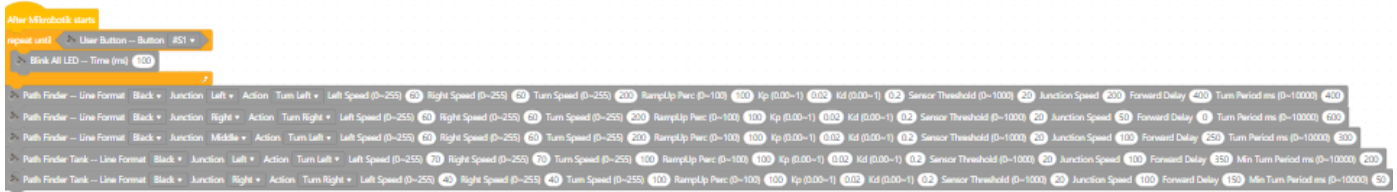
Langkah 5

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”*).

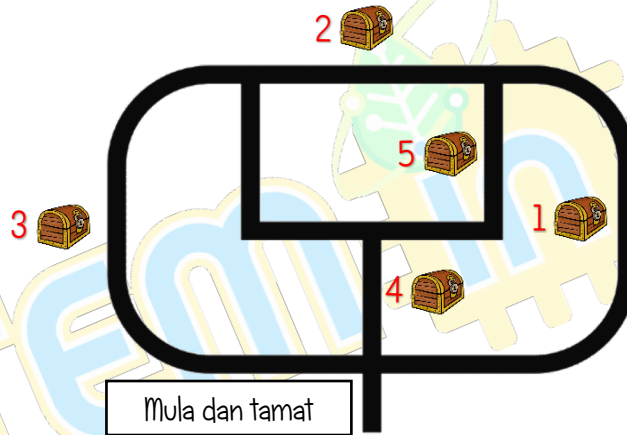


Langkah 7

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.

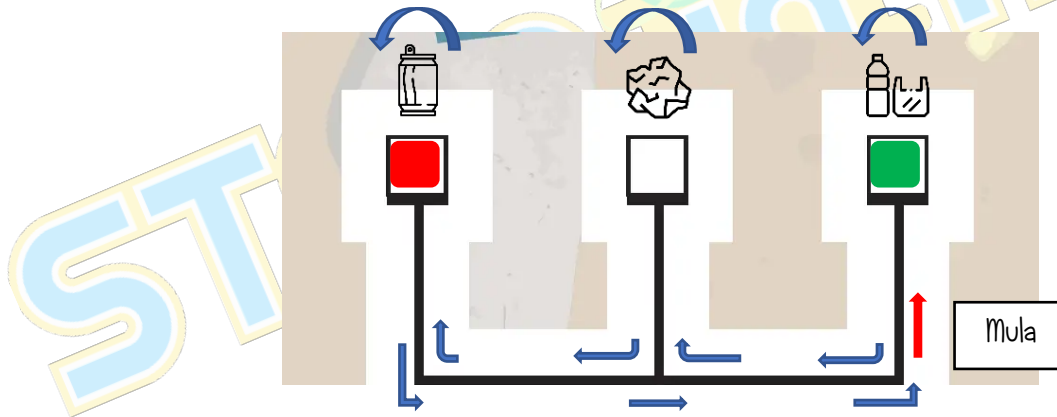
Cabaran!!

Dapatkan semua harta dan tamat.



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

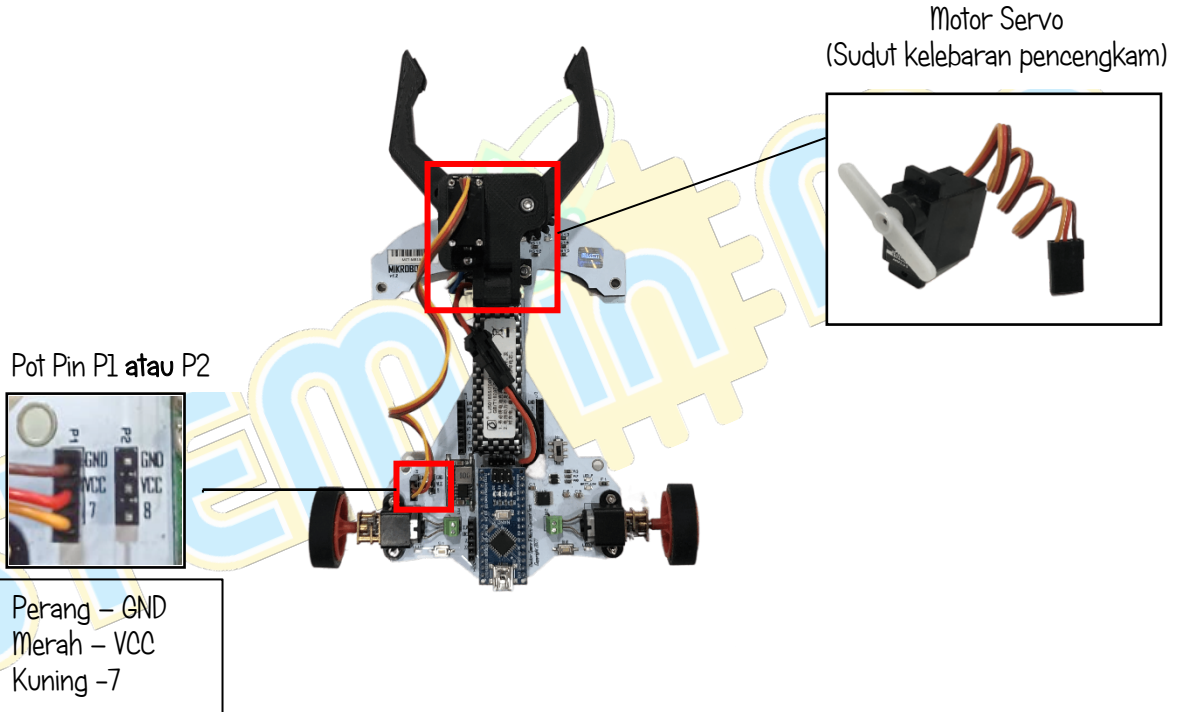


Pencengkam tunggal digunakan dalam Mikrobotik untuk mencengkam objek dan mengalihkan (secara menyeret) objek ke lokasi yang dikehendaki. Pencengkam tunggal terdiri daripada satu motor servo dan satu tangan pencengkam mekanikal.

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

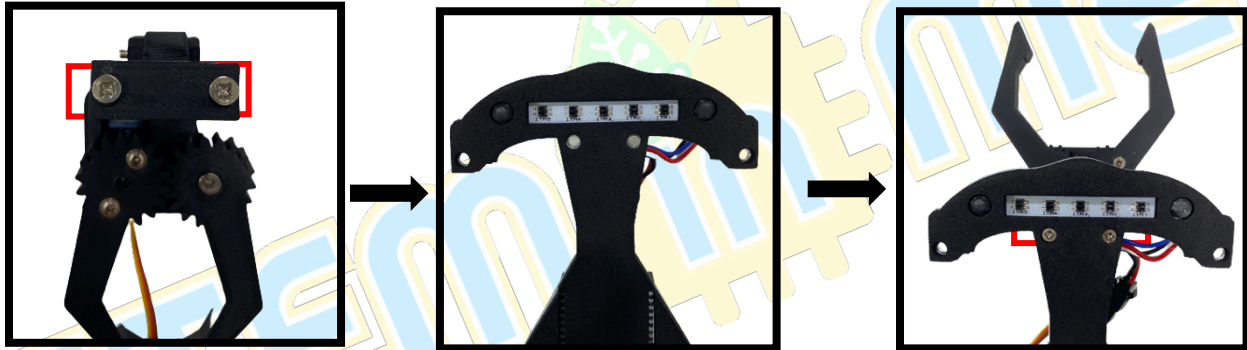
Lebar bukaan tangan pencengkam boleh dilaraskan dengan menetapkan sudut motor servo. Lebih besar sudut servo, lebih kecil bukaan tangan pencengkam. Biasanya apabila sudut motor servo ditetapkan pada 0 darjah, bukaan tangan pencengkam adalah pada keadaan paling luas. Apabila sudut motor servo ditetapkan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

Pemasangan Pencengkam Tunggal pada robot.

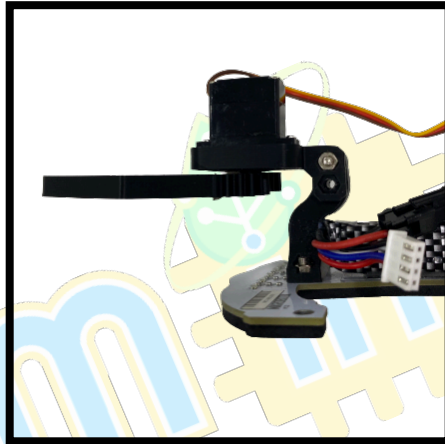


Berikut merupakan susunan langkah pemasangan Pencengkam Tunggal pada Mikrobotik:

- 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.

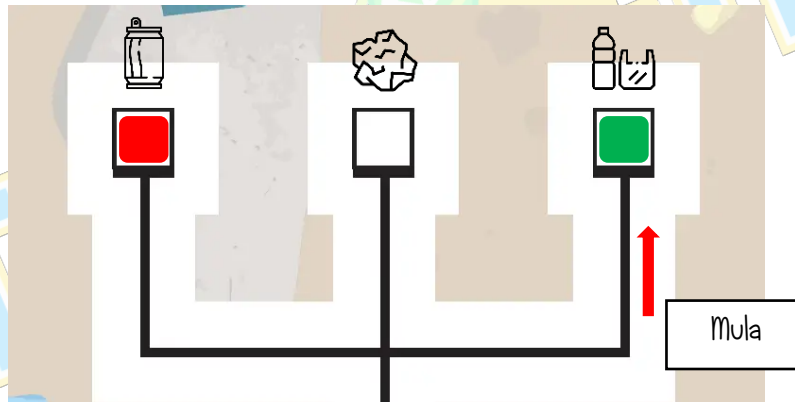
Untuk mendapatkan objek pertama (hijau), robot mencengkam objek itu menggunakan blok *Gripper Servo Port*. Kemudian, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kiri kanan di ruang khas dan berhenti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

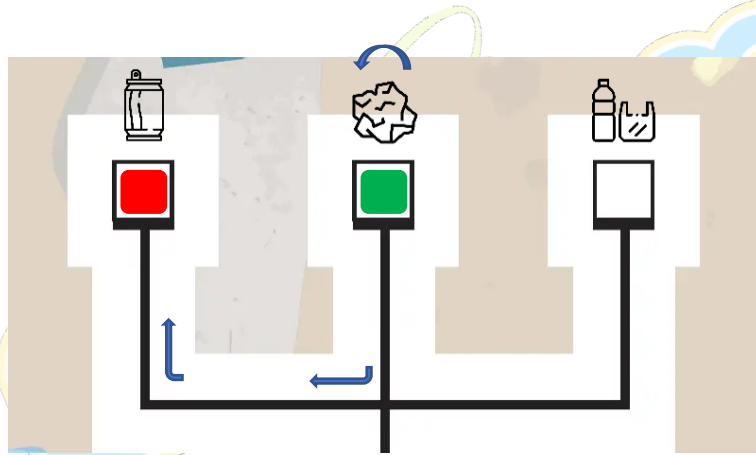
Setelah itu, robot berpusing menggunakan blok *Turn at Centre* dan bergerak menggunakan beberapa blok *Path Finder* sehingga menemui persimpangan kiri kanan di ruang khas lalu meletakkan objek kedua di dalam ruang khas menggunakan blok *Gripper Servo Port*.

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

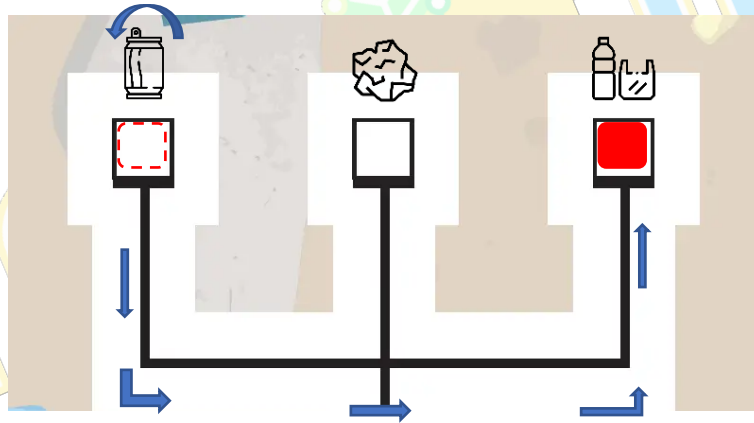
- i) Menggunakan blok *Path Finder*, robot bergerak dari titik mula sehingga bertemu simpang kiri kanan dan berhenti. Kemudian, robot mencengkam objek pertama (hijau) menggunakan blok Gripper Servo Port (Pencengkam) dengan tetapan sudut yang besar (bukaan tangan kecil).



- 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 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 kecil (bukaan tangan besar).
- iii) Selepas melepaskan objek pertama, 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 kanan. Selepas itu, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kanan menghadap 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).



- iv) 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 (*Junction* – **"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"**; *Min Turn 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”*. *MinTurn 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*).

The screenshot shows a sequence of programming blocks for a robot's path-finding task:

- After Mikrobolic starts**
- repeat until** (User Button - Button #S1)
- Blink All LED** (Time (ms): 50)
- wait** (1 seconds)
- Path Finder** (Line Format: Black, Junction: Middle, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 50, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, Sensor Threshold: 20, Junction Speed: 50, Forward Delay: 0, Turn Period ms: 0-10000)
- Gripper Servo Port** (D7, Force Hold: On, Gripper Angle: 133, Pause Time: 0)
- Turn At Centre** (Line Format: Black, Direction: Turn Left, Speed: 50, Sensor Threshold: 0-1000, Min Turn Period ms: 0-1000, 600)
- 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, 20, Junction Speed: 100, Forward Delay: 50, Turn Period ms: 0-10000, 600)
- Path Finder** (Line Format: Black, Junction: Middle, Action: Stop, Left Speed: 50, Right Speed: 50, Turn Speed: 50, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, 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: 50, Right Speed: 50, Turn Speed: 50, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, 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: 50, Sensor Threshold: 0-1000, Min Turn Period ms: 0-1000, 300)
- 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: 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, Right Speed: 0-255, Turn Speed: 0-255, RampUp Perc: 0-100, Kp: 0.00-1, Kd: 0.00-1, 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: 50, Right Speed: 50, Turn Speed: 50, RampUp Perc: 0-100, Kp: 0.00-1, Kd: 0.00-1, 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: 50, Sensor Threshold: 0-1000, Min Turn Period ms: 0-1000, 300)
- 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, 20, Junction Speed: 100, Forward Delay: 50, Turn Period ms: 0-10000, 550)
- Path Finder** (Line Format: Black, Junction: Middle, Action: Stop, Left Speed: 100, Right Speed: 100, Turn Speed: 100, RampUp Perc: 100, Kp: 0.00-1, Kd: 0.00-1, 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: 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: 50, Turn Period ms: 0-10000, 600)
- Path Finder** (Line Format: Black, Junction: Middle, Action: Stop, Left Speed: 0-255, Right Speed: 0-255, Turn Speed: 0-255, RampUp Perc: 0-100, Kp: 0.00-1, Kd: 0.00-1, 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)

Maklumat Tambahan

Force Hold (Daya Cengkaman)

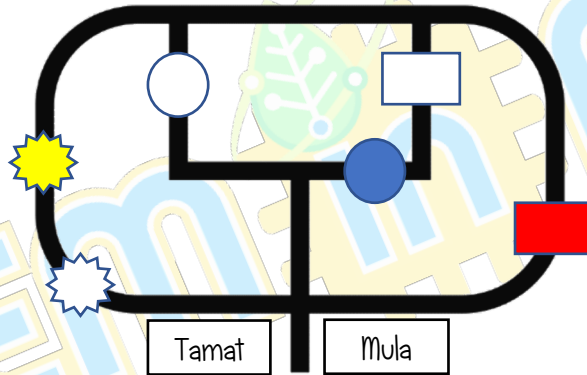
- Apabila daya cengkaman dihidupkan (ON) pencengkam akan menahan sudut kelebarannya daripada diubah.
- Apabila daya cengkaman dimatikan (OFF), pencengkam tidak akan menahan sudut kelebarannya daripada diubah.

Pause Time (Tempoh Jeda)

- Tempoh jeda hanya boleh digunakan 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.

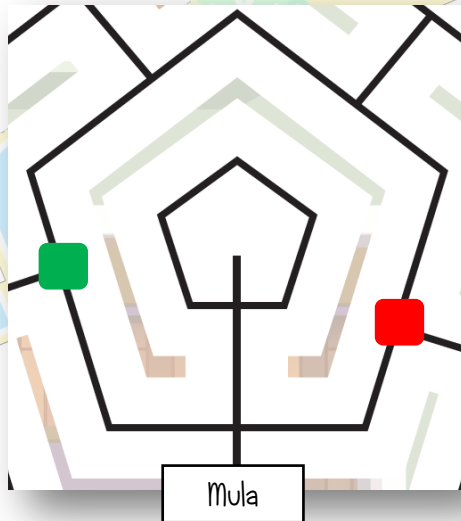
Cabaran!!

Alihkan objek bahan kitar semula dari satu tempat ke satu tempat khusus yang ditentukan.

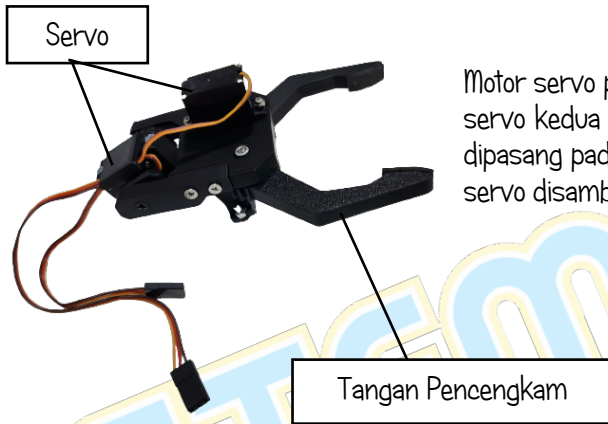


Objektif 12: Penyimpanan Cepak Ruang.

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 membawa dan mengangkat objek. Wayar servo disambungkan ke pin pengepala servo sedia ada berlabel "P1" dan "P2".

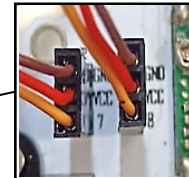
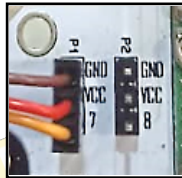
Pemasangan Pencengkam Berganda pada robot.

Motor Servo P1
(Sudut kelebaran pencengkam)

Motor Servo P2
(Sudut ketinggian pencengkam)

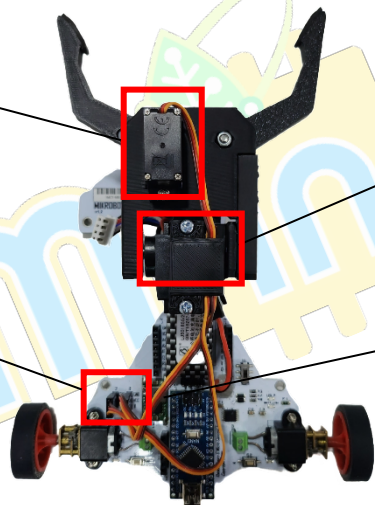
Pot Pin P1
(Servo 1)

Pot Pin P2
(Servo 2)



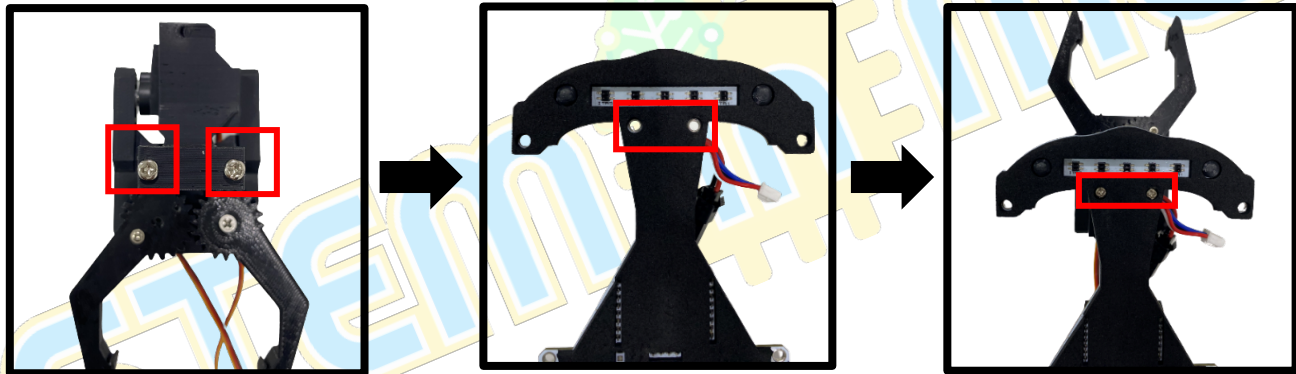
Perang - GND
Merah - VCC
Kuning - 7

Perang - GND
Merah - VCC
Kuning - 8

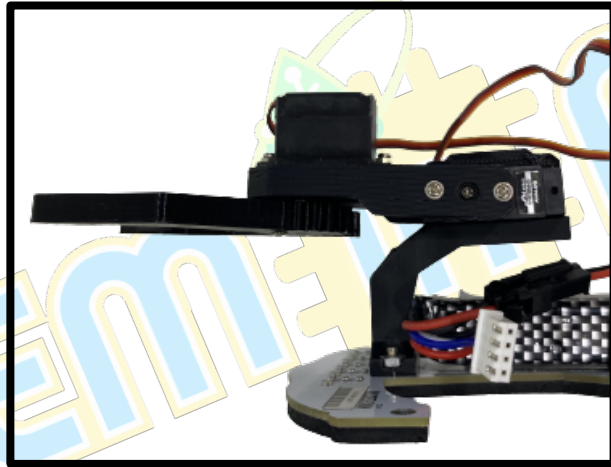


Berikut merupakan susunan langkah pemasangan Pencengkam Berganda pada Mikrobotik:

- 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.

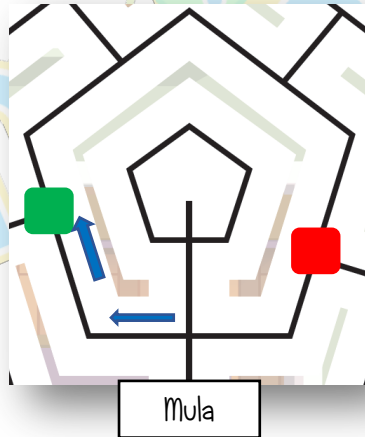
Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder*, *Line Tracer Time*, *Gripper Servo Port* dan *Turn at Centre*.

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

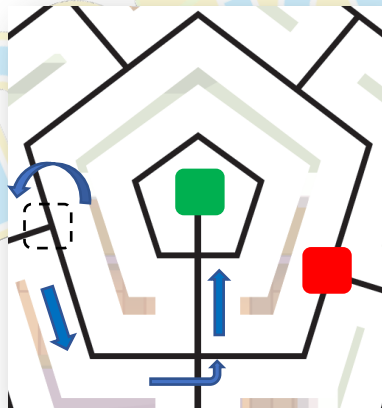
Untuk mendapatkan objek pertama (hijau), robot mencengkam objek itu menggunakan blok *Gripper Servo Port*. Kemudian, robot berpusing menggunakan blok *Turn at Centre* dan meneruskan pergerakan menggunakan blok *Path Finder* sehingga menemui persimpangan kiri dan robot akan membelok ke kiri. Kemudian 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 persimpangan kiri kanan lalu berhenti.

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

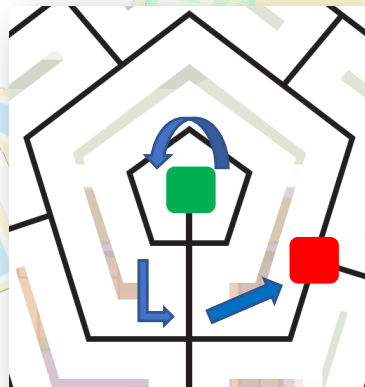
- i) Menggunakan blok *Path Finder*. Robot bergerak dari titik mula sehingga bertemu simpang kiri kanan dan pusing ke kiri. Kemudian robot bergerak menggunakan blok *Path Finder* dan menemui persimpangan kanan lalu pusing ke kanan kemudian menggunakan blok *Line Tracer Time* dan mencengkam objek pertama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



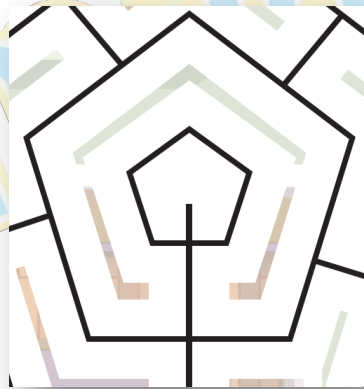
- ii) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri dan pusing ke kiri. Seterusnya, robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing 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) .



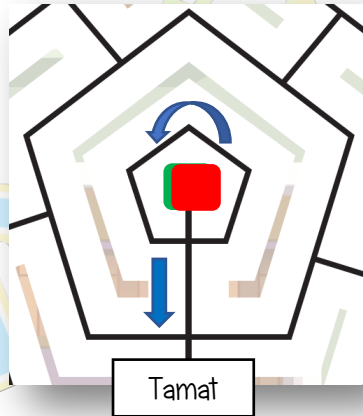
- iii) Selepas melepaskan objek pertama (hijau), robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* kemudian robot bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan dan pusing ke kiri. Seterusnya, robot bergerak menggunakan blok *Path Finder* 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 (bukaan 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).



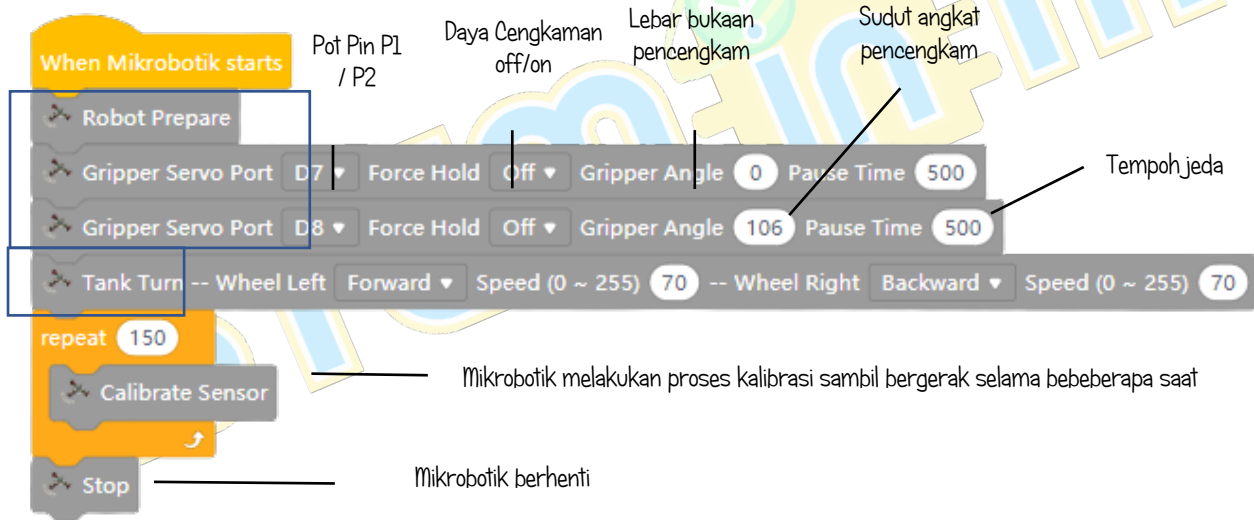
- i) Robot berpusing menghadap arah bertentangan menggunakan blok *Turn at Centre* dan bergerak menggunakan blok *Path Finder* sehingga bertemu persimpangan kiri kanan lalu berhenti.



Langkah-langkah susunan blok

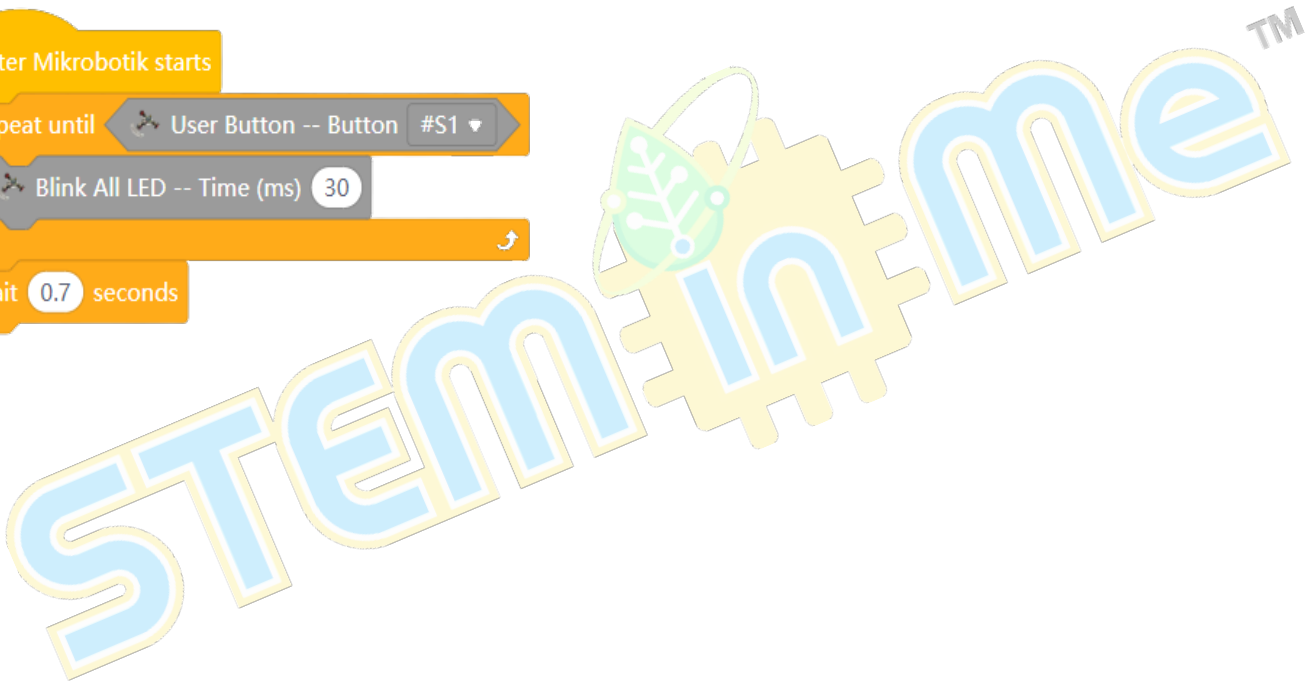
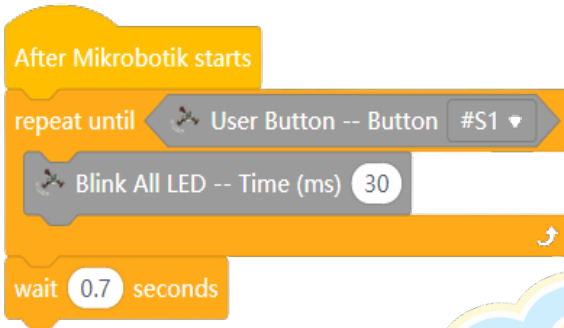
Langkah 1 Pastikan wayar motor servo disambungkan ke pin pengepala servo P1 dan P2.

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”*).

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”*).

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 Servo Port* dan tetapkan nilai (*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) 20

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) 530

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 100 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 90 Pause Time 500

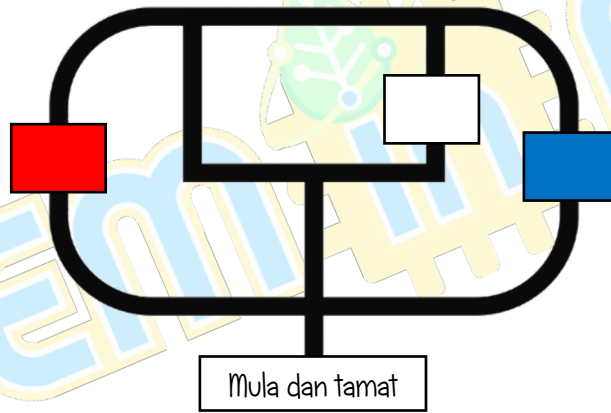
Gripper Servo Port D7 -- Force Hold Off -- Gripper Angle 0 Pause Time 0

Langkah 9

Tambahkan blok *Turn At Centre* (*Direction Turn* – “Left”. *Speed* – “80”. *MinTurn 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”).

Cabaran!!

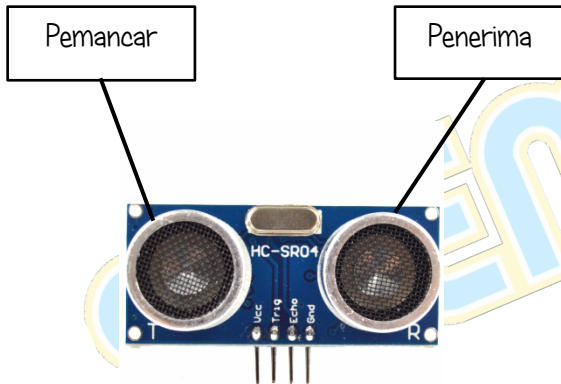
Alihkan blok merah dan biru kedalam ruang yang disediakan.



Objektif 13: Berhenti! Halangan di Hadapan

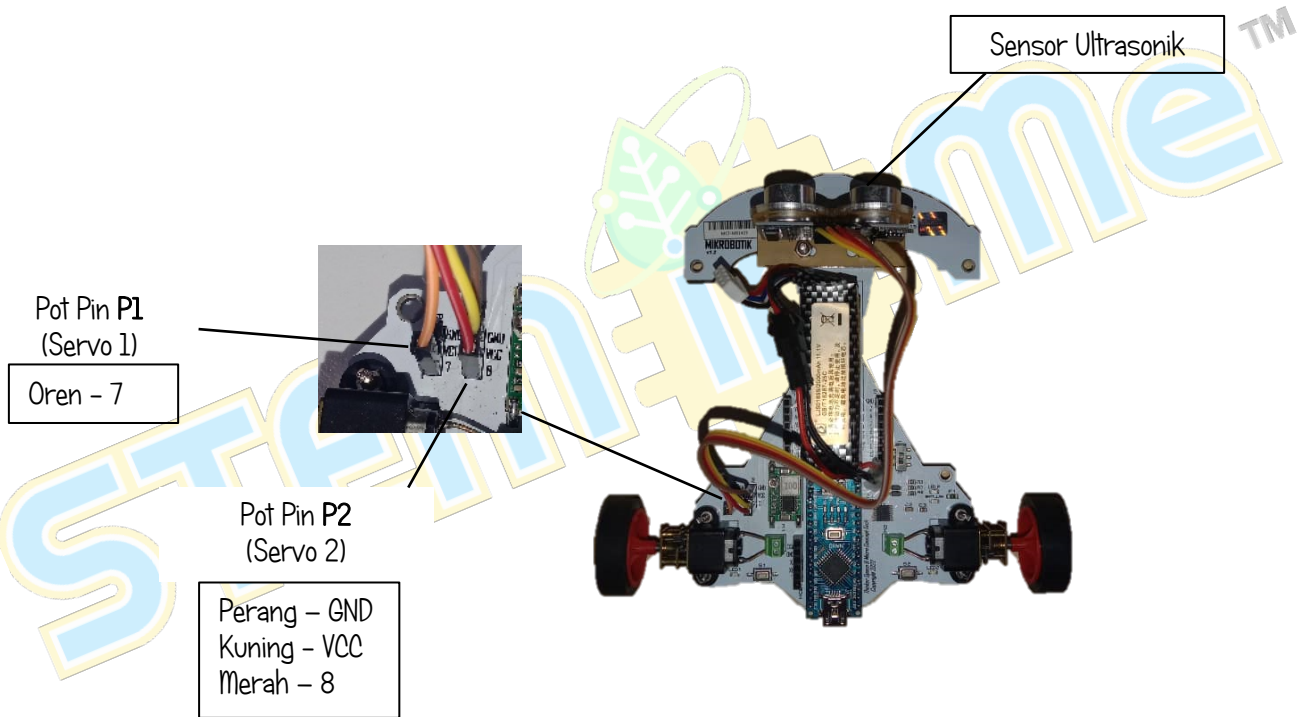
Kadang-kadang terdapat halangan di sepanjang laluan Robot. Kita mesti memastikan bahawa robot kita tidak melanggar halangan. Di sinilah kami akan menggunakan komponen baharu – Penderia Ultrasonik. Objektif tertumpu kepada robot untuk berhenti apabila terdapat halangan. Untuk tujuan ini, robot menggunakan sensor ultrasonik untuk mengesan sebarang halangan.

Pengenalan sensor ultrasonik dan Mekanismanya



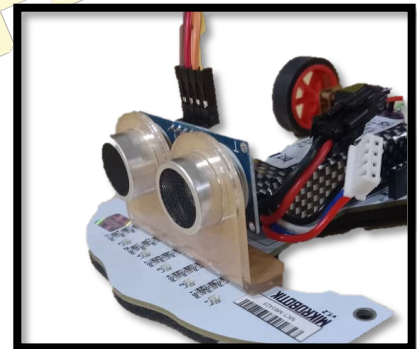
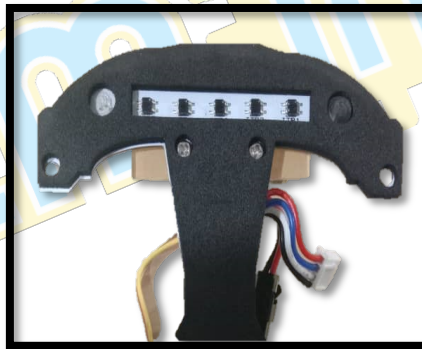
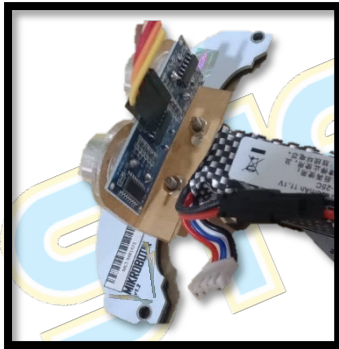
Penderia ultrasonik menggunakan gelombang bunyi ultrasonik untuk mengukur jarak objek dan kemudian mengubah gelombang bunyi yang dipantulkan kepada isyarat elektrik. Terdapat dua komponen utama penderia ultrasonik. Pemancar menggunakan kristal piezoelektrik untuk menjana gelombang bunyi ke arah halangan, dan penerima akan menerima gelombang bunyi yang dipantulkan selepas ia bergerak dari halangan. Penderia ultrasonik akan mengira jarak halangan berdasarkan gema yang diterima.

Pemasangan sensor ultrasonik pada robot



Berikut merupakan susunan langkah pemasangan Ultrasonik pada Mikrobotik:

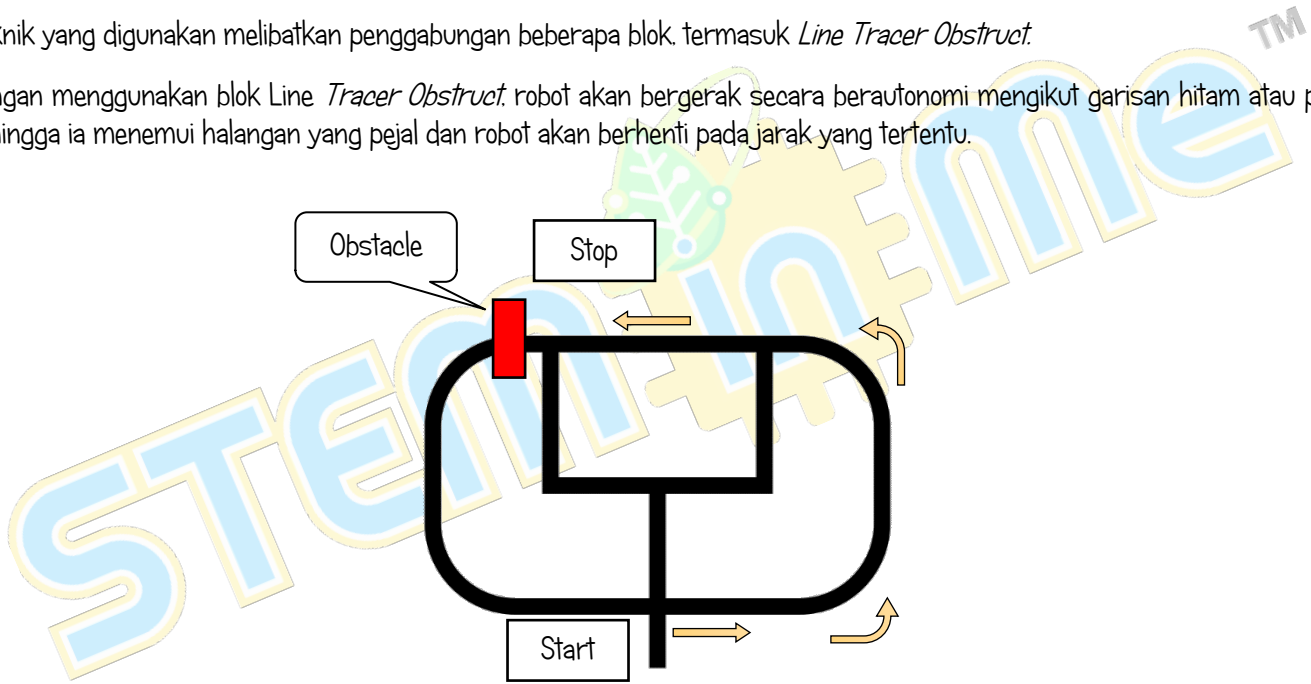
- 1- Longgarkan skru dan nat pada Pendakap Ultrasonik. Dibawah robot Mikrobotik terdapat dua lubang skru untuk Pendakap Ultrasonik. Masukkan skru dan nat kemudian ketatkan.
- 2- Pastikan posisi Sensor Ultrasonik selepas pemasangan adalah seperti ini.



Susun Atur Strategi dan Teknik Pergerakan

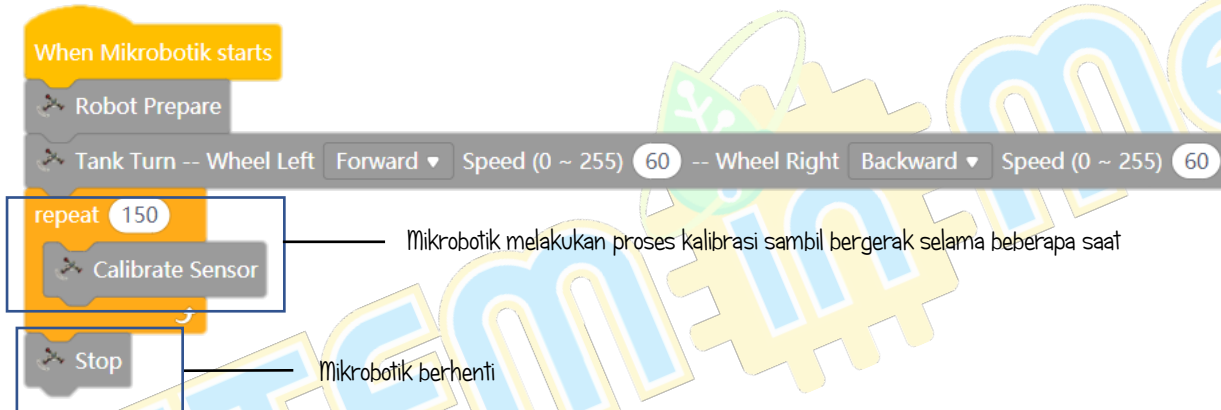
Teknik yang digunakan melibatkan penggabungan beberapa blok, termasuk *Line Tracer Obstruct*.

Dengan menggunakan blok *Line Tracer Obstruct*, robot akan bergerak secara berautonomi mengikuti garisan hitam atau putih sehingga ia menemui halangan yang pejal dan robot akan berhenti pada jarak yang tertentu.



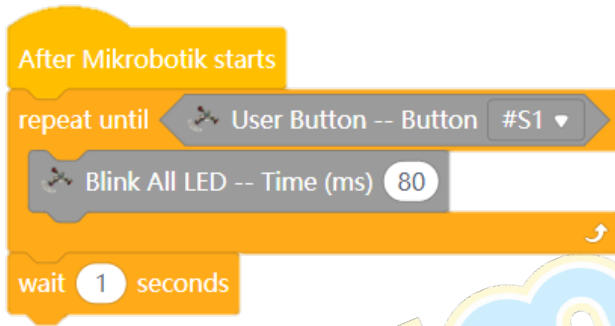
Langkah-langkah susunan blok

Langkah 1 Sediakan susunan blok kalibrasi automatik.



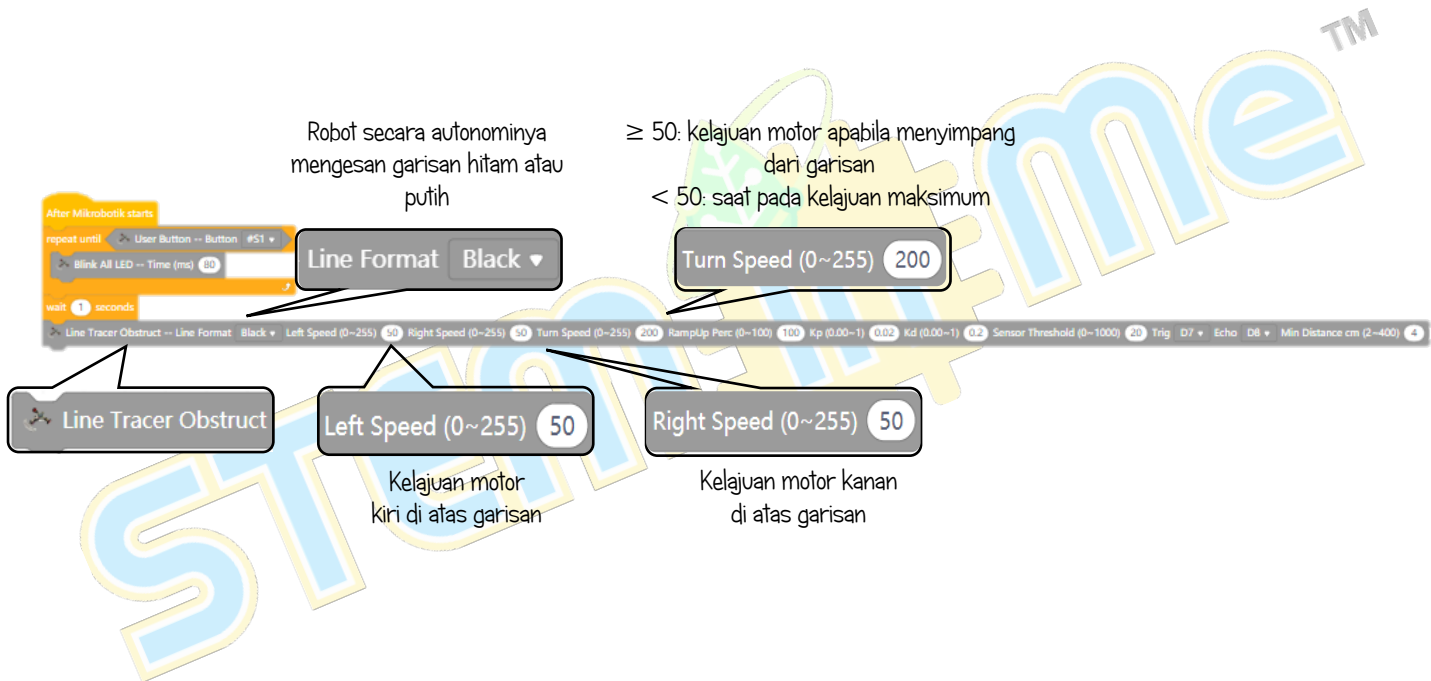
Langkah 2

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



STEM in me™

Langkah 3 Tambah 1 Line Tracer Obstruct block dan tetapkan nilai seperti di bawah:



Sambungan

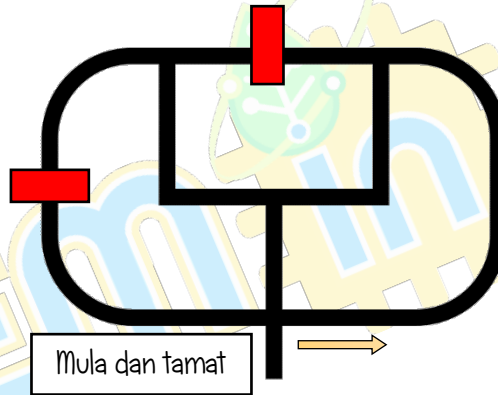
The image shows a screenshot of the MikroBotik programming environment. The main interface displays a sequence of commands for a line tracer robot. Callouts provide detailed explanations for several key parameters:

- Kp (0.00~1): 0.02** - Mengawal nilai Kp (Controlling the Kp value)
- Sensor Threshold (0~1000): 20** - Sensitiviti garisan (semakin tinggi nilai, semakin sensitif) (Line sensitivity (the higher the value, the more sensitive))
- RampUp Perc (0~100): 100** - Peratusan pecutan (Acceleration percentage)
- Kd (0.00~1): 0.2** - Mengawal nilai Kd (Kd mesti lebih >Kp) (Controlling the Kd value (Kd must be greater than Kp))
- Min Distance cm (2~400): 4** - Jarak antara robot dan halangan (Distance between robot and obstacle)

The background of the interface features a large, stylized watermark that reads "STEM MINDS" in blue and yellow.

Cabaran!!

Berhenti pada halangan pertama, kemudian bergerak dan berhenti pada halangan kedua dan tamat.



Objektif 14: Garisan berbeza warna? Jom selesaikan!

Terdapat situasi di mana robot perlu melalui garisan dengan warna yang berbeza dalam suatu litar. Pergerakan robot akan diaplikasikan menggunakan konsep pertukaran dari garisan hitam ke garisan putih atau sebaliknya. Bagi melaksanakan konsep ini, robot akan menggunakan blok *Path Finder* dengan pilihan persimpangan *middle junction* dan *dead end*.

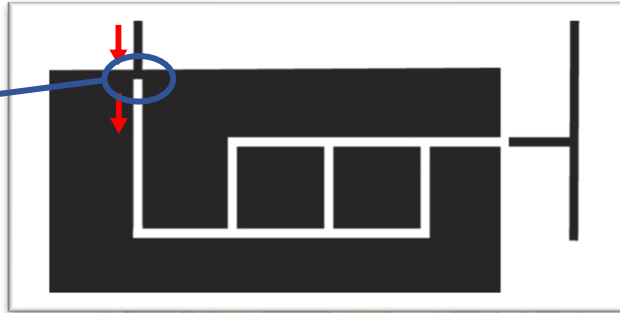
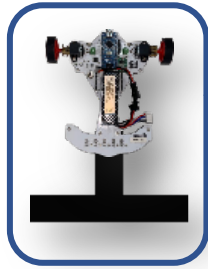
Pengenalan Pergerakan dan Mekanismenya

Teknik yang digunakan adalah dengan menggabungkan beberapa blok *Path Finder*.

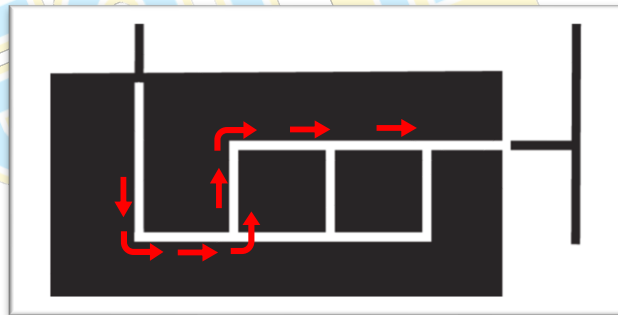
Robot bergerak secara berautonomi mengikuti garisan dengan menggunakan blok *Path Finder* tetapi *Line Format* mengikut warna garisan sama ada hitam atau putih. Kebiasaannya, tempat pertukaran warna garisan akan dibezakan dengan persimpangan tengah atau jalan mati. Oleh itu, robot perlu mengenal pasti jenis persimpangan yang akan dilalui ketika pertukaran garisan hitam ke putih atau sebaliknya.

Berikut disediakan lakaran pergerakan robot untuk melengkapkan tugas.

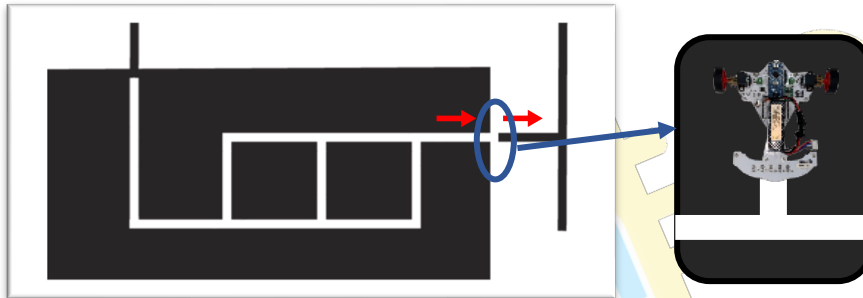
- i) Robot bergerak secara berautonomi dengan menggunakan *Path Finder* mengikuti garisan hitam dengan pilihan *middle junction*. Ini kerana tempat pertukaran warna garisan pada litar adalah persimpangan tengah seperti yang ditunjukkan. Kemudian robot akan bergerak mengikuti garisan putih.



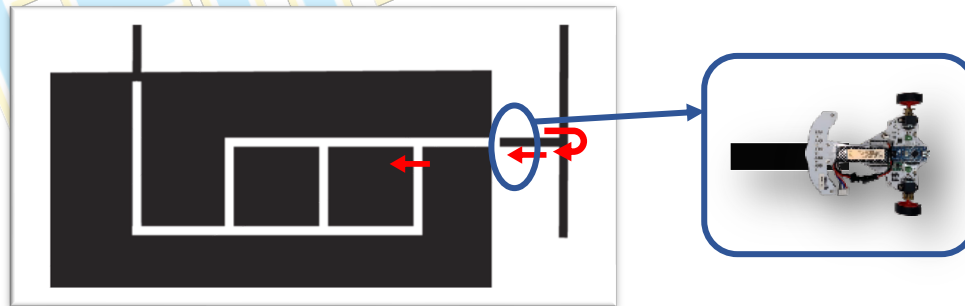
- ii) Robot pusing ke kiri ketika di persimpangan kiri dan pusing kanan ketika di persimpangan kanan dengan menggunakan blok *Path Finder*.



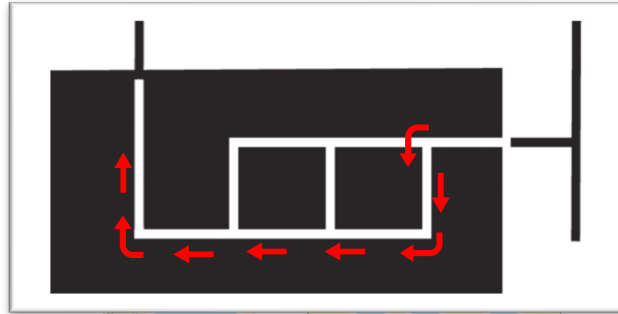
- iii) Robot bergerak secara berautonomi dengan menggunakan *Path Finder* mengikuti garisan putih dengan pilihan *middle junction*. Ini kerana tempat pertukaran warna garisan pada litar adalah persimpangan tengah seperti yang ditunjukkan. Kemudian robot akan bergerak mengikuti garisan hitam.



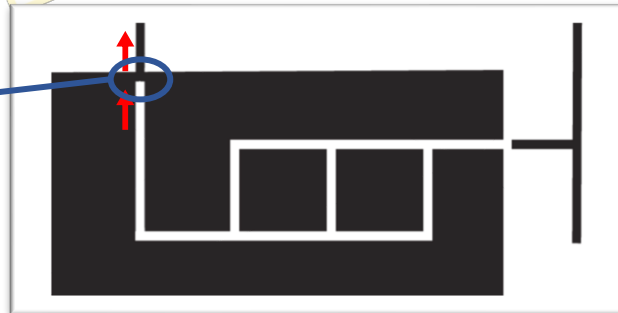
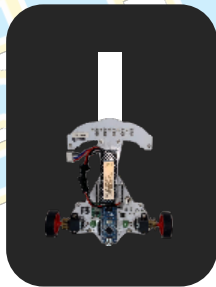
- iv) Robot melakukan pusingan dengan menggunakan blok *Path Finder Tank*. Kemudian, robot bergerak mengikuti garisan hitam menggunakan blok *Path Finder* dengan pilihan *dead end*. Ini kerana tempat pertukaran warna garisan pada litar adalah *dead end* seperti yang ditunjukkan. Kemudian robot akan bergerak mengikuti garisan putih.



- i) Robot akan pusing ke kiri di persimpangan kiri dan pusing ke kanan di persimpangan kanan mengikuti garisan putih dengan menggunakan blok *Path Finder*.

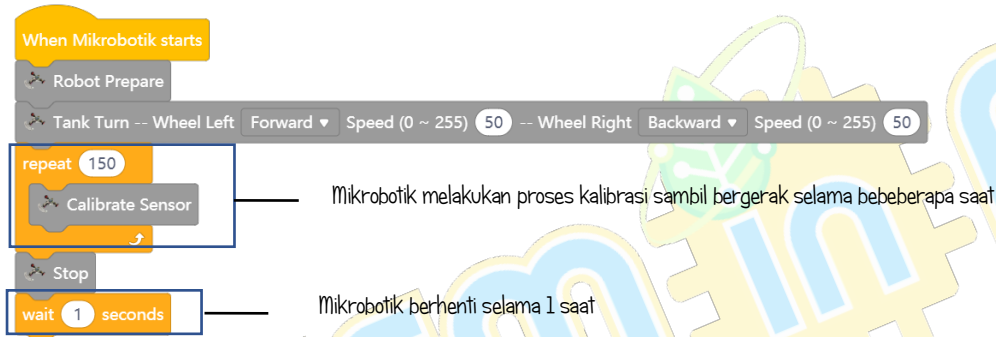


- ii) Robot akan bergerak mengikuti garisan putih menggunakan blok *Path Finder* dengan pilihan *dead end*. Ini kerana tempat pertukaran warna garisan pada litar adalah *dead end* seperti yang ditunjukkan. Kemudian robot akan bergerak mengikuti garisan hitam.

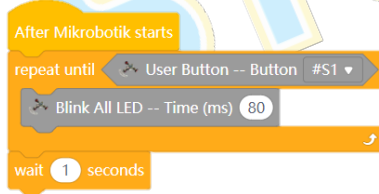


Langkah-langkah susunan blok:

Langkah 1 Sediakan susunan blok kalibrasi automatik.

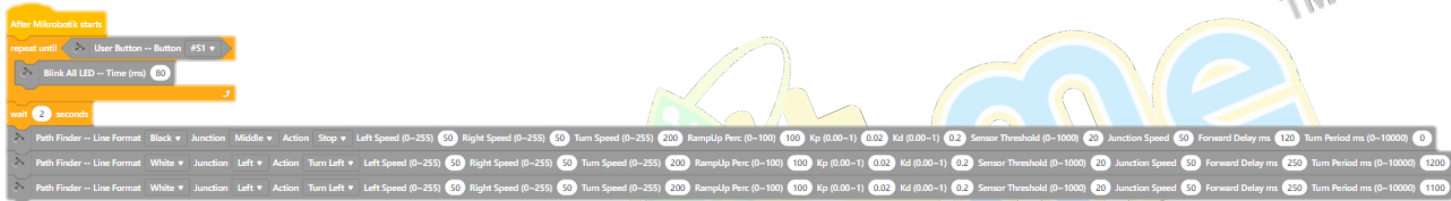


Langkah 2 Gabungkan blok *After Mikrobotik starts* dengan blok *repeat until* dan masukkan blok *wait 1 seconds* Ke dalam blok *repeat until*.



Langkah 3

Tambahkan blok 3 *Path Finder* dan tetapkan *Line Format* kepada *Black*, *White*, *White*, *Junction* kepada *Middle*, *Left*, *Left*, *Action* kepada *Stop*, *Turn Left*, *Turn Left*, *Forward Delay* 120, 250, 250 dan *Turn Period* 0, 1200 dan 1100.

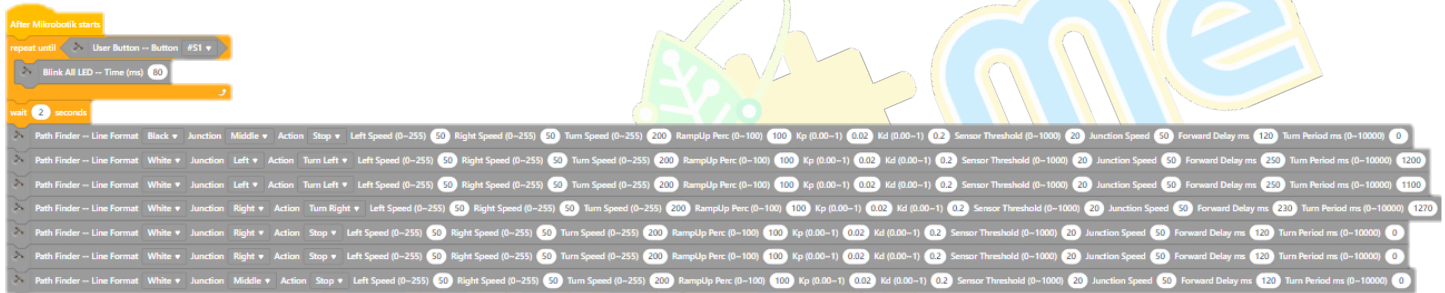


NOTA KAKI

Ketika pertukaran warna garisan berlaku dari garisan hitam ke putih atau sebaliknya, penambahan nilai *Forward Delay* perlu dilakukan supaya robot akan melepasi warna garisan awal untuk mengesan warna garisan yang seterusnya.

Langkah 4

Tambahkan blok 4 *Path Finder* dan tetapkan *Line Format* kepada *White*. *Junction* kepada *Right*. *Right*. *Right*. *Middle*. *Action* kepada *Turn Right*. *Stop*. *Stop*. *Stop*. *Forward Delay* kepada *230*. *120*. *120*. *120* dan *Turn Period* kepada *1270*. *0*. *0*. *0*.



The screenshot shows a sequence of programming blocks in a Scratch-like environment. The blocks are as follows:

- Block 1:** *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*
- Block 2:** *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*
- Block 3:** *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*
- Block 4:** *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*
- Block 5:** *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*
- Block 6:** *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*
- Block 7:** *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*

Langkah 5

Tambahkan blok *Path Finder Tank* dengan *Line Format Black*, *Junction Middle*, *Action Turn Right*, *Forward Delay 500* dan *Min Turn Period 900*. Kemudian, tambahkan blok *Path Finder* dengan *Line Format Black*, *Junction DeadEnd*, *Action Stop*, *Forward Delay 120* dan *turn period 0*.

The screenshot shows a sequence of programming blocks in a Scratch-like environment. The blocks are as follows:

- wait 2 seconds**
- repeat until** (User Button → Button #51)
 - Blink All LED → Time (ms)** (5)
- 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** (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

Langkah 6

Tambahkan 4 blok *Path Finder* dan tetapkan *Line Format* kepada *White*. *Junction* kepada *Left*. *Right*. *Right* dan *Right*. *Action* kepada *Turn Left*. *Turn Right*. *Stop* dan *Stop*. *Forward Delay* kepada *250*. *230*. *120*. *120* dan *Turn Period* kepada *1100*. *1270*. *0* dan *0*.

The image shows a Scratch script starting with a 'wait 2 seconds' block, followed by a 'repeat until' loop containing a 'Blink All LED' block. Below the loop are ten '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: 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 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: 500, Min Turn Period ms: 900.
- Block 8: 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 9: 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 10: 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 11: 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 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.

Langkah 7

Tambahkan 3 blok *Path Finder* dengan *Line Format White, White, Black, Junction Right, DeadEnd, Right, Action Turn Right, Stop, Stop, Forward Delay 230, 300, 300* dan *Turn Period 1270, 0, 0*.

The image shows a Scratch script editor with the following blocks:

- After Mikrobolik 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 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
- 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 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

Objektif 15: Pengecaman Bahan Kitar Semula Automatik

Memisahkan bahan kitar semula secara manual mengambil masa yang lama dan sering berlaku kesilapan. Ini membuatkan proses kitar semula kurang efektif dan tidak efisien. Dengan menggunakan kamera AI, proses pengecaman bahan kitar semula boleh dilakukan secara automatik. Kamera ini boleh mengenal pasti jenis bahan seperti plastik, kertas, logam, dan kaca dengan cepat dan tepat. Teknologi ini membantu mempercepatkan proses kitar semula, mengurangkan kesilapan, dan mengurangkan keperluan tenaga kerja manusia. Hasilnya, lebih banyak bahan dapat dikitar semula dengan cara yang lebih baik untuk alam sekitar.

Pengenalan ESP32-Camera dan Mekanismenya

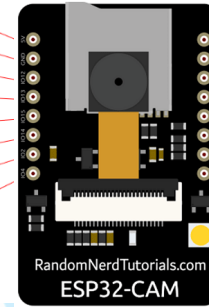
ESP32-CAM adalah modul kamera kecil yang menggunakan mikrokontroler ESP32. Ia mempunyai Wi-Fi, Bluetooth, dan kamera OV2640 dengan resolusi 2MP. Modul ini sesuai untuk aplikasi seperti pemantauan, penstriman video, dan IoT. Dengan slot MicroSD, ia boleh menyimpan imej dan video secara tempatan. ESP32-CAM boleh diprogramkan menggunakan Arduino IDE dan digunakan untuk pengesanan wajah, keselamatan, atau robotik. Sambungan Wi-Fi membolehkan pemindahan data jarak jauh, dan GPIO-nya boleh mengawal peranti lain. Ia memerlukan bekalan kuasa 5V dan mempunyai fungsi pemrosesan data secara dalaman.



Berikut merupakan sambungan pin ESP32-CAM ke port Mikrobotik.

Port pada Mikrobotik	Pin ESP32-CAM
GND	GND
VCC	5V
4	IO#1 (UOTXD)
7	IO#3 (UORXD)

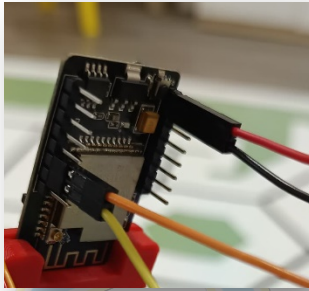
Power In: 5V
GND
IO#12
IO#13
IO#15
IO#14
IO#2
IO#4



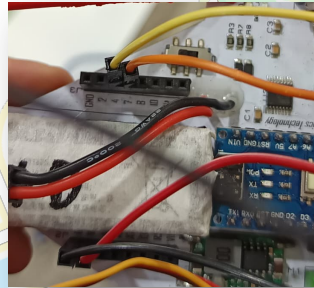
Power In: 3.3V
IO#16
IO#0
GND
VCC
IO#3 (UORXD)
IO#1 (UOTXD)
GND

RandomNerdTutorials.com
ESP32-CAM

Gambar 1: Pin pada ESP32-CAM



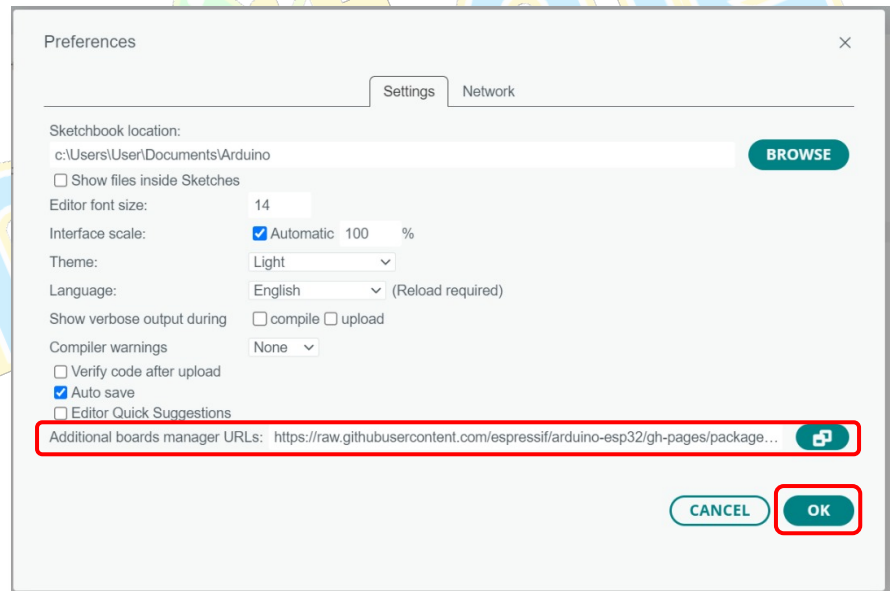
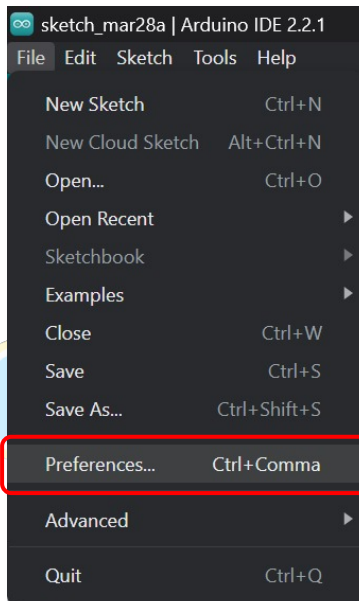
Gambar 2: penyambungan pada ESP32-CAM



Gambar 3: Penyambungan pada Mikrobotik

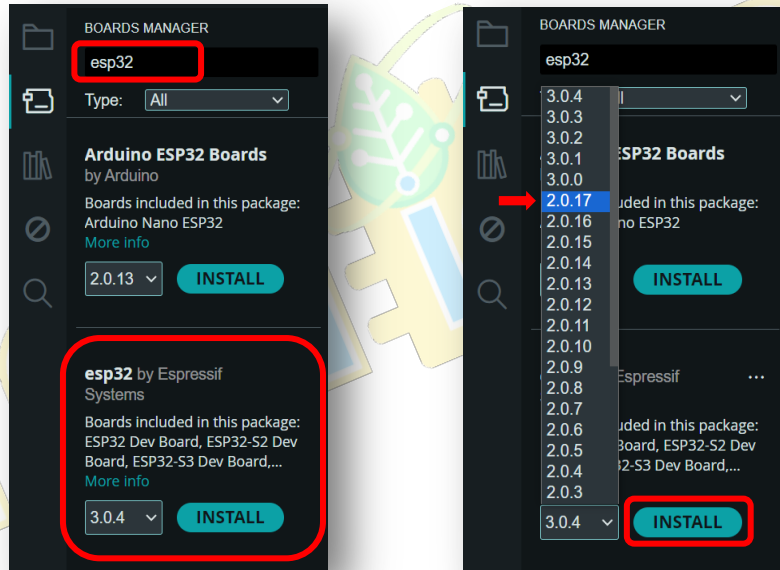
Menambah peranti ESP32-CAM pada Arduino IDE

Langkah 1 Buka Arduino IDE dan klik *File* di bahagian atas kiri. Cari dan klik *Preferences*. Di bahagian *Additional Boards Manager URLs*, masukkan URL: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json dan klik *OK*



Langkah 2

Seterusnya klik simbol fail di bahagian kiri aplikasi dan cari "esp32" seperti yang ditunjukkan. Pilih *esp32 by Espressif Systems*. Pastikan versi yang dipilih adalah 2.0.17 dan tekan install.

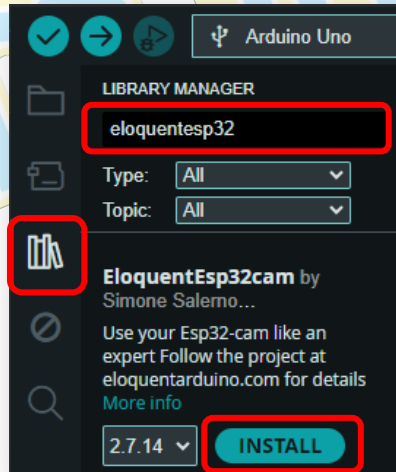


Pengumpulan Data menggunakan ESP32-CAM

Untuk melatih model AI dengan berkesan, data yang berkualiti dan relevan adalah diperlukan. Data yang diperlukan adalah dalam bentuk imej yang ditangkap oleh ESP32-CAM. Model AI memerlukan pelbagai contoh data untuk belajar mengenali corak dan berfungsi dengan baik. Dengan mengumpul data juga membolehkan kita menguji dan memperbaiki model AI berdasarkan data sebenar yang dikumpul untuk meningkatkan prestasi model.

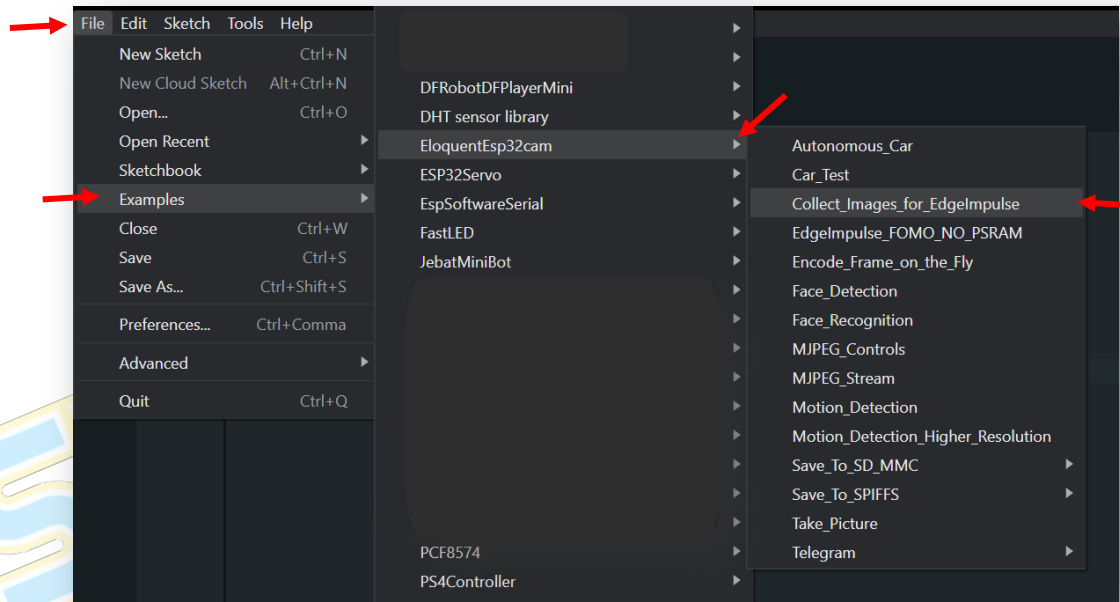
Langkah 1

Buka Arduino IDE dan klik simbol buku pada bahagian kiri aplikasi. Kemudian taip "eloquentesp32" pada ruangan *search*. Pastikan library yang muncul adalah *EloquentEsp32cam* by *Simone Salerno*. Kemudian tekan install.



Langkah 2

Klik pada menu *file* di bahagian kiri atas aplikasi. Kemudian pilih examples. Cari pilihan EloquentEsp32cam dan tekan pada menu Collect_Images_for_EdgeImpulse.



Langkah 3

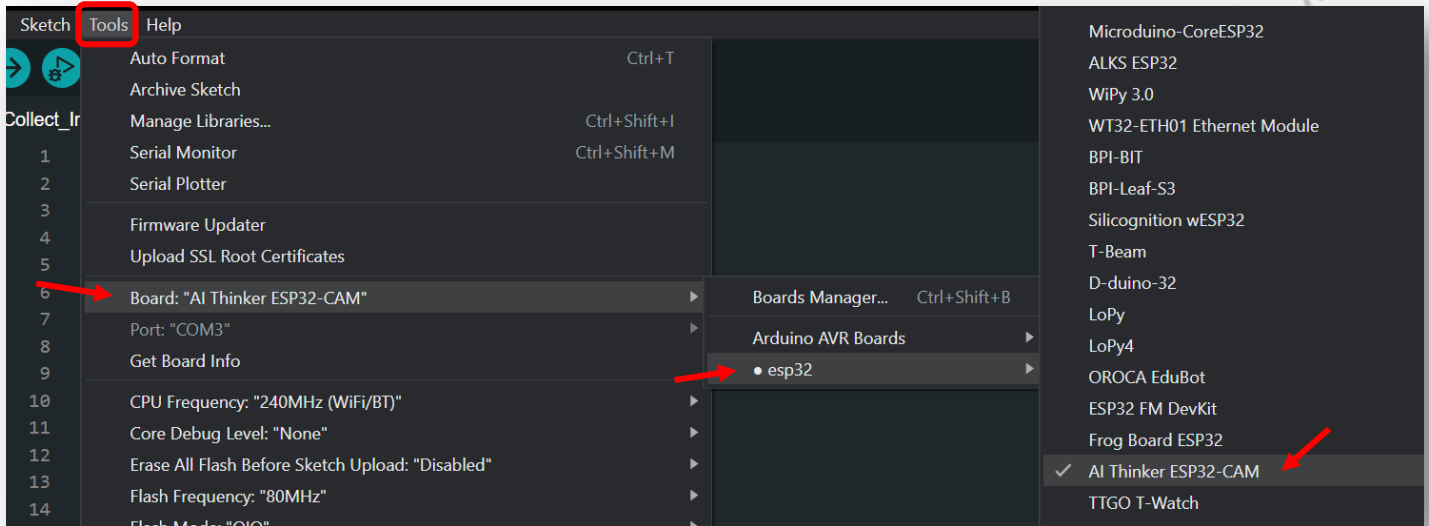
Kemudian pada coding yang dipaparkan, tukarkan Wifi SSID dan PASS pada line 15 dan 16 kepada nama dan password wifi yang sedang digunakan. Tukar "wroom_s3" kepada "aithinker" pada line 36. Ia ditukar mengikut jenis model camera yang digunakan.

```
Collect_Images_for_EdgeImpulse.ino
13 // the camera at http://{HOSTNAME}.local
14
15 #define WIFI_SSID "SSID"
16 #define WIFI_PASS "PASSWORD"
17 #define HOSTNAME "esp32cam"
18
19
20 #include <eloquent_esp32cam.h>
21 #include <eloquent_esp32cam/extra/esp32/wifi/sta.h>
22 #include <eloquent_esp32cam/viz/image_collection.h>
23
24 using eloq::camera;
25 using eloq::wifi;
26 using eloq::viz::collectionServer;
27
28
29 void setup() {
30     delay(3000);
31     Serial.begin(115200);
32     Serial.println("__IMAGE COLLECTION SERVER__");
33
34     // camera settings
35     // replace with your own model!
36     camera.pinout.wroom_s3();
37     camera.brownout.disable();
38     // Edge Impulse models work on square images
39     // face resolution is 240x240
40     camera.resolution.face();
41     camera.quality.high();
42
```

```
Collect_Images_for_EdgeImpulse.ino
13 // the camera at http://{HOSTNAME}.local
14
15 #define WIFI_SSID "StemInMe"
16 #define WIFI_PASS "1234567890"
17 #define HOSTNAME "esp32cam"
18
19
20 #include <eloquent_esp32cam.h>
21 #include <eloquent_esp32cam/extra/esp32/wifi/sta.h>
22 #include <eloquent_esp32cam/viz/image_collection.h>
23
24 using eloq::camera;
25 using eloq::wifi;
26 using eloq::viz::collectionServer;
27
28
29 void setup() {
30     delay(3000);
31     Serial.begin(115200);
32     Serial.println("__IMAGE COLLECTION SERVER__");
33
34     // camera settings
35     // replace with your own model!
36     camera.pinout.aithinker();
37     camera.brownout.disable();
38     // Edge Impulse models work on square images
39     // face resolution is 240x240
40     camera.resolution.face();
41     camera.quality.high();
42
```

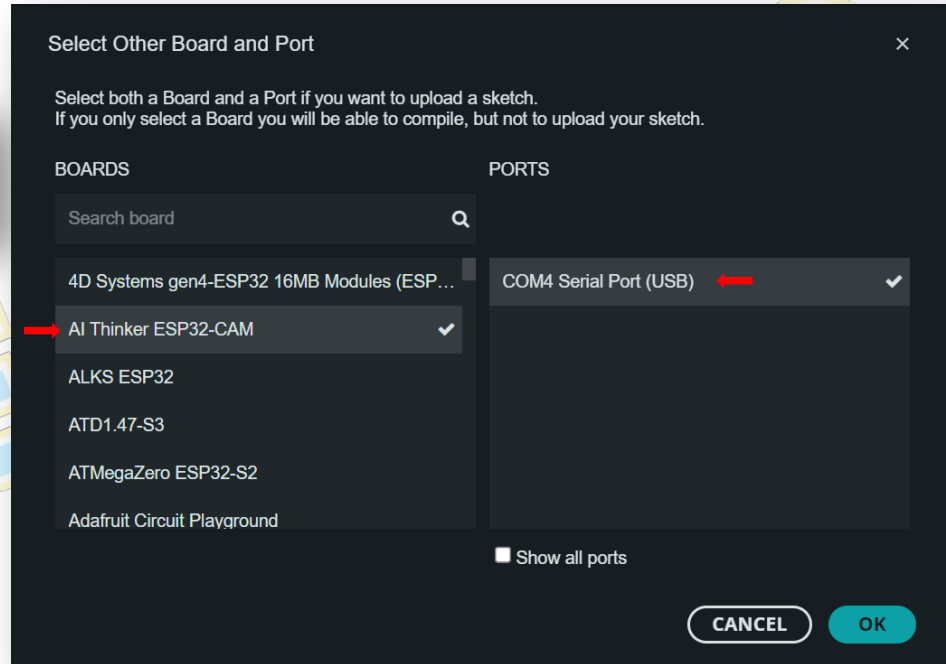
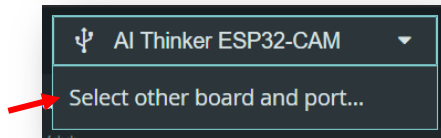
Langkah 4

Klik pada menu Tools dan pilih menu Board. Kemudian tekan pada esp32 dan cari AI Thinker ESP32-CAM.



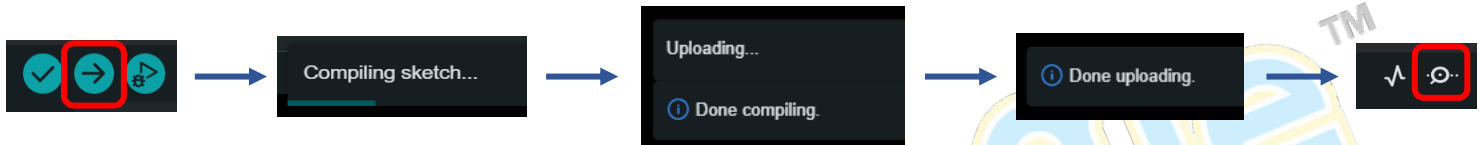
Langkah 5

Tekan pada *AI Thinker ESP32-CAM* dan pilih *select other board and port*. Pastikan *board* dan *port* dipilih dengan betul. Kemudian klik ok.



Langkah 6

Tekan pada simbol *arrow* di bahagian kiri atas untuk *compile* dan *upload*. Setelah selesai *upload*, tekan pada simbol kanta pembesar di bahagian kanan atas untuk membuka *serial monitor*.



Langkah 7

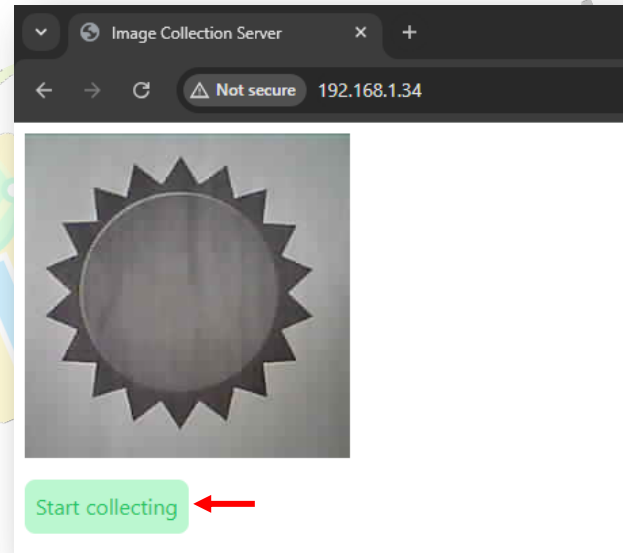
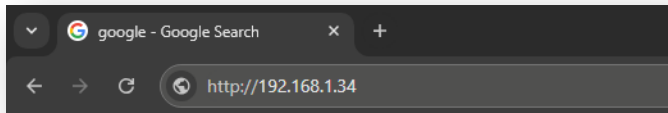
Pastikan nilai baud yang ditetapkan adalah 115200 baud. Kemudian pada paparan serial monitor, salin alamat IP seperti yang ditunjukkan.

```

Output Serial Monitor x
Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM4')
New Line 115200 baud
16:46:31.503 -> ets Jul 29 2019 12:21:46
16:46:31.503 ->
16:46:31.503 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
16:46:31.504 -> configsip: 0, SPIWP:0xee
16:46:31.504 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
16:46:31.504 -> mode:DIO, clock div:1
16:46:31.504 -> load:0x3fff0030,len:1344
16:46:31.504 -> load:0x40078000,len:13964
16:46:31.504 -> load:0x40080400,len:3600
16:46:31.504 -> entry 0x400805f0
16:46:35.235 -> __IMAGE_COLLECTION_SERVER__
16:46:40.075 -> Camera OK
16:46:40.075 -> WiFi OK
16:46:40.075 -> Image Collection Server OK
16:46:40.075 -> Image Collection Server is available at http://192.168.1.34
Ln 25, Col 18 AI Thinker ESP32-CAM on COM4 2
  
```

Langkah 8

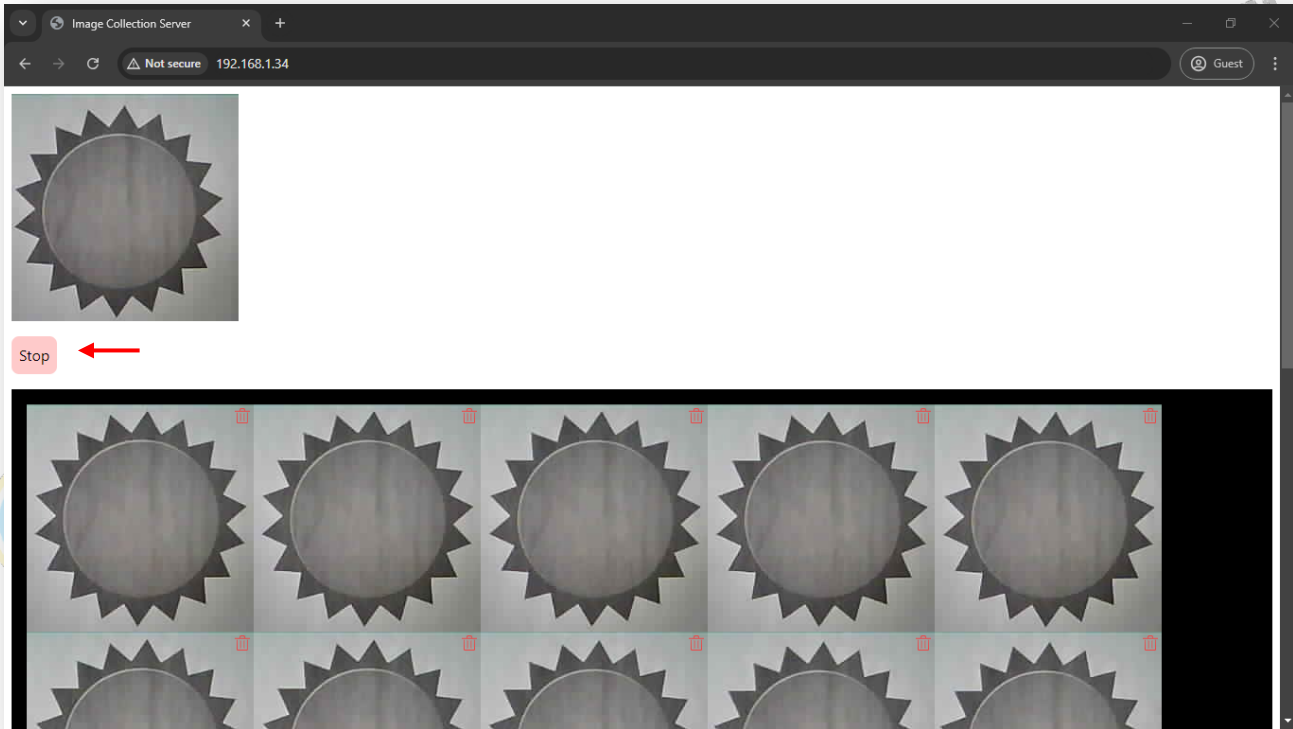
Tampil alamat IP pada carian *Google Search* dan laman *Image Collection Server* akan muncul. Halakan kamera pada data imej yang ingin dikumpul seperti yang ditunjukkan dan tekan *Start collecting*.



STEM

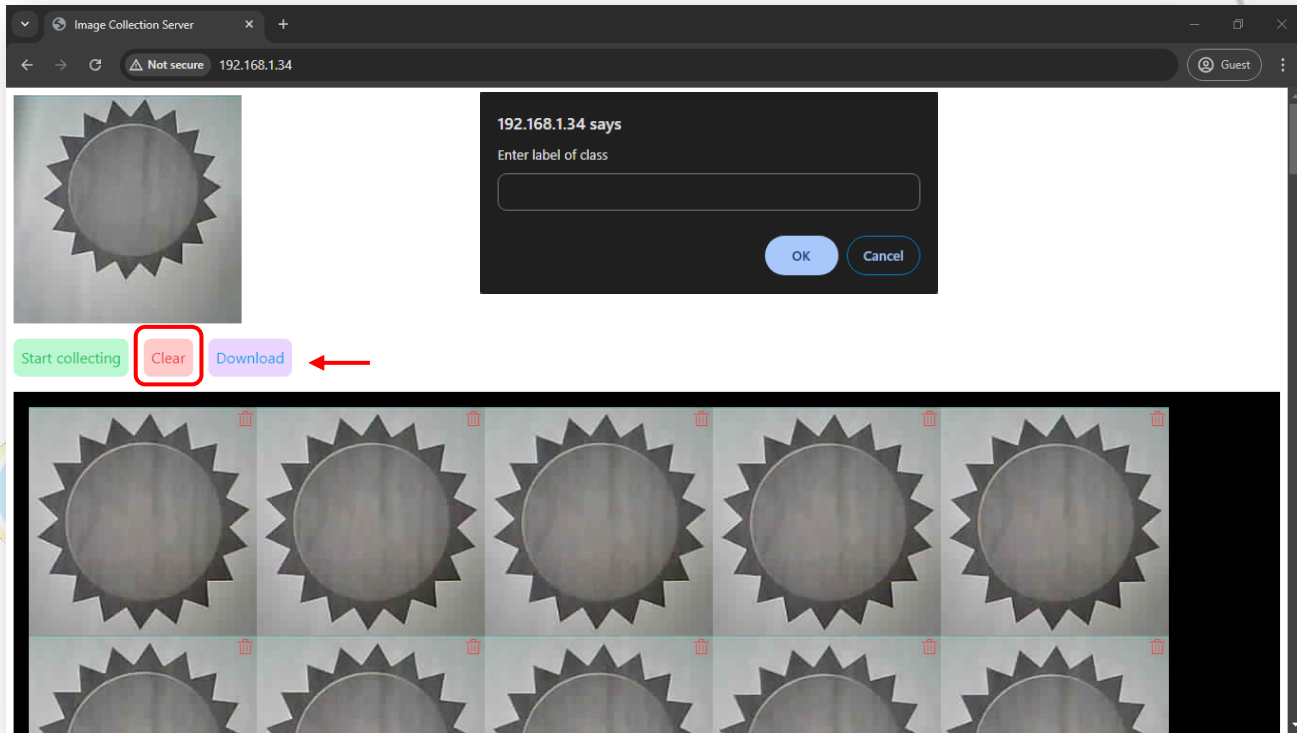
Langkah 9

Imej data yang dikumpul akan dipaparkan pada skrin. Lebih banyak data imej yang dikumpul, lebih tepat ia berfungsi. setelah cukup data imej yang ingin dikumpul, tekan *Stop*.



Langkah 10

Kemudian tekan *Download* untuk muat turun semua data imej yang telah dikumpulkan. Masukkan nama kelas untuk data imej dan tekan *OK*. Untuk mengumpul data imej berlainan kelas, tekan *Clear* untuk mengosongkan data lama dan ulang dari Langkah 8.

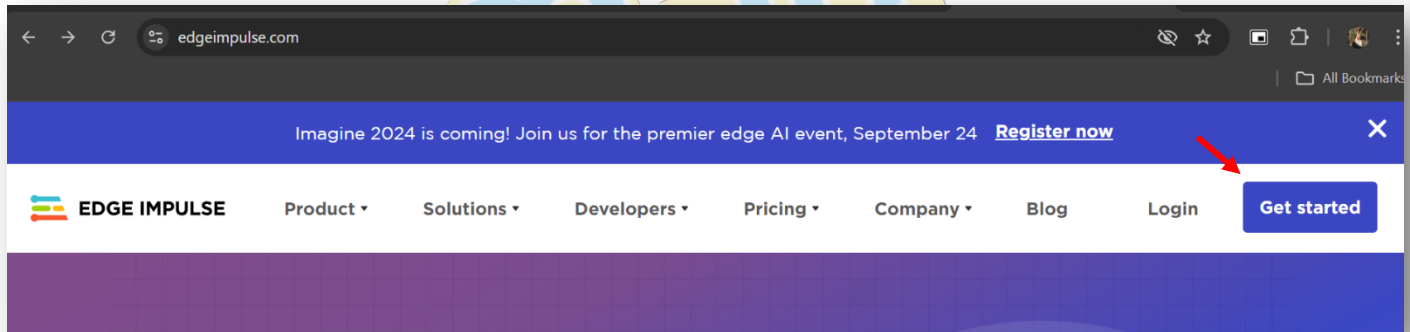


Penggunaan EdgeImpulse untuk Membangun Model Kecerdasan Buatan (AI)

Edge Impulse adalah platform yang memudahkan pembangun dalam mencipta dan menggunakan model kecerdasan buatan (AI) untuk peranti kecil seperti sensor, mikropengawal, dan kamera. Platform ini menyokong pengumpulan data, latihan model AI, dan penggunaan model pada peranti dengan kuasa rendah. Ia amat berguna untuk aplikasi peranti pintar dan projek Internet of Things (IoT).

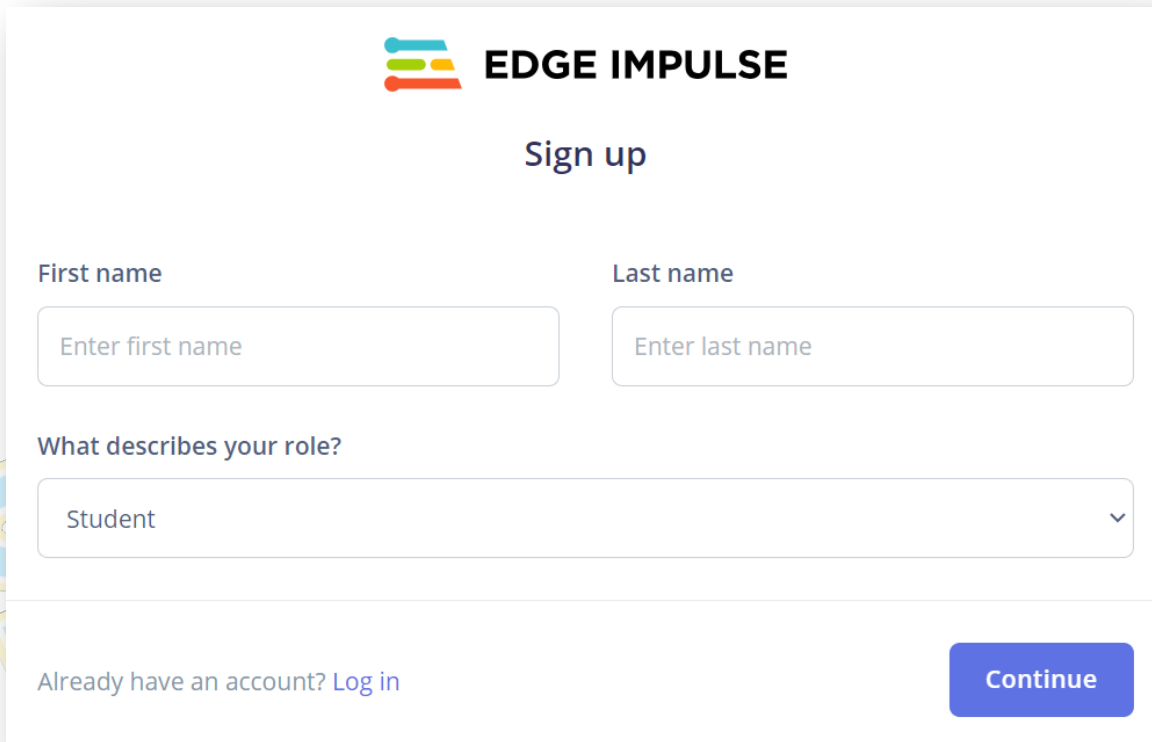
Langkah 1

Cari *Edge Impulse* pada carian *Google Search* dan tekan *Get Started*.



Langkah 2

Masukkan nama dan pastikan *role* yang ditetapkan adalah *Student*.



EDGE IMPULSE

Sign up

First name

Last name

What describes your role?

Already have an account? [Log in](#)

[Continue](#)

Langkah 3

Kemudian masukkan maklumat yang diperlukan dan klik *Sign up*. Setelah muncul *Sign up successful*. Tekan pada *Click here to build your first ML model!*

Sign up

First name

Last name

Email

Username

Password

I accept the [Privacy Policy](#), [Community Terms of Service](#), and [Responsible AI License](#).

Already have an account? [Log in](#)

Sign up

Sign up successful!

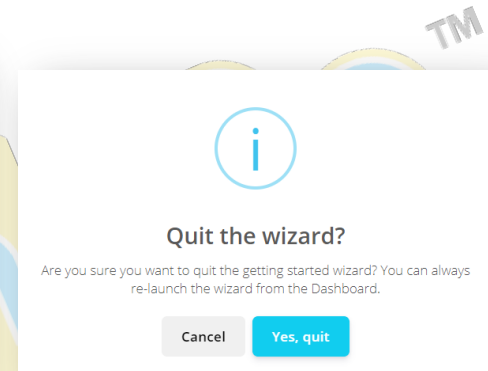
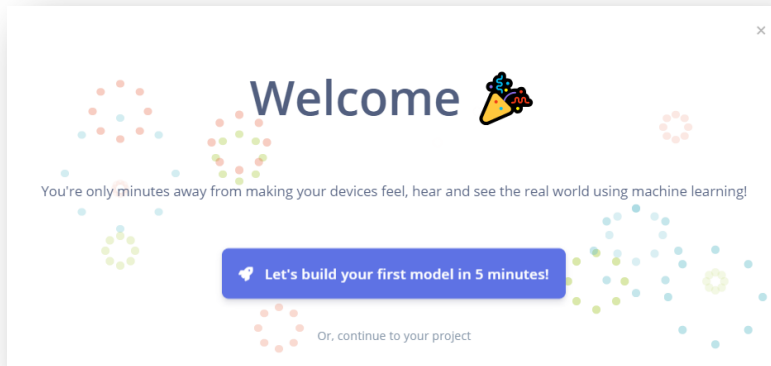
You have successfully signed up for Edge Impulse.

[Click here to build your first ML model!](#)

© 2024 Edgelpulse Inc. All rights reserved

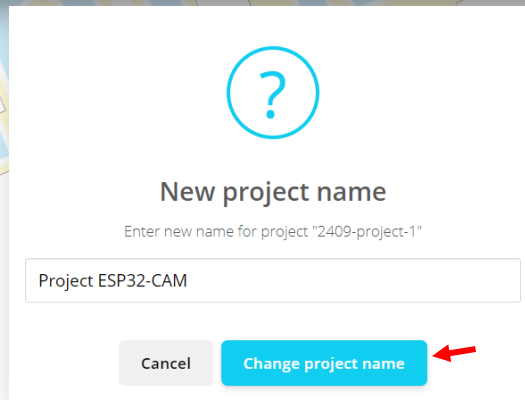
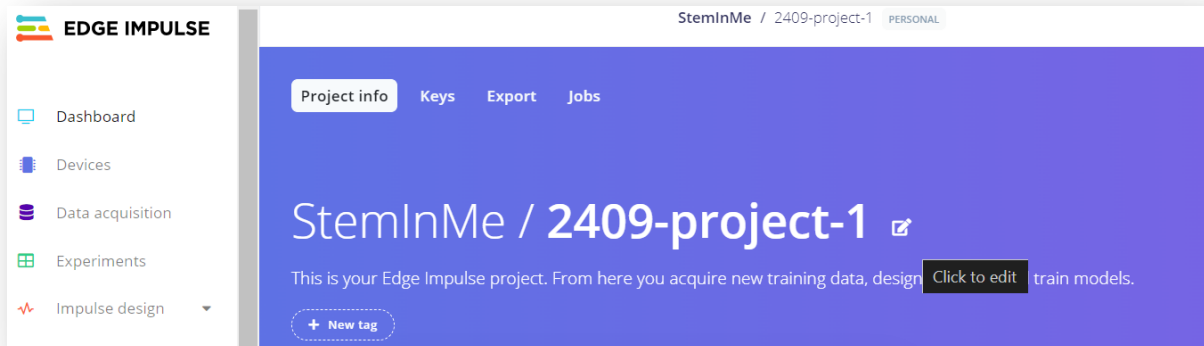
Langkah 4

Tekan *continue to your project*. Setelah muncul *Quit the wizard?* tekan *Yes, quit*.



Langkah 5

Pada *dashboard* Edge Impulse, tekan logo pensel untuk mengubah nama projek. Kemudian namakan projek seperti yang ditunjukkan dan tekan *change project name*.



Langkah 6

Kemudian pada *dashboard*, tekan pada *Add existing data* untuk menambahkan imej data yang telah dikumpulkan sebelum ini.

EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Experiments
- Impulse design
 - Create impulse
 - Retrain model

Upgrade Plan
Get access to higher job limits, collaborators and a

StemInMe / Project ESP32-CAM

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

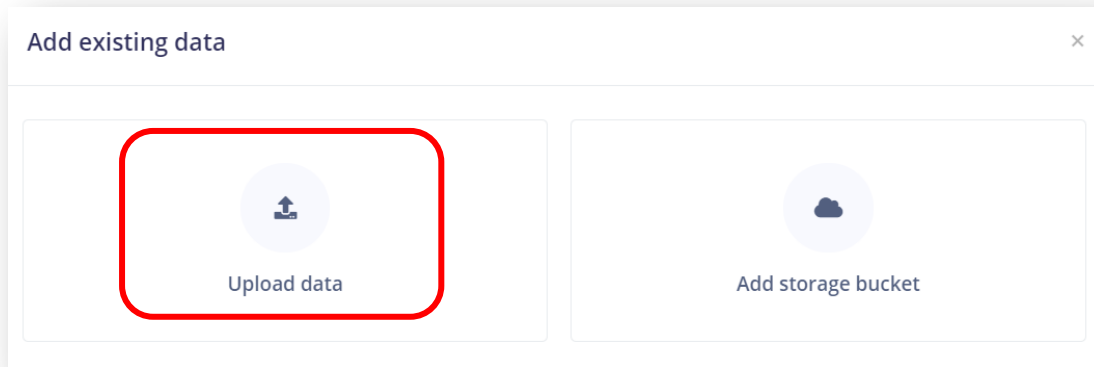
+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

- Add existing data**
- Collect new data
- Upload your model

Langkah 7 Tekan pada *Upload data*.



Langkah 8

Pastikan menu yang dipilih seperti yang ditunjukkan di bawah. Di bahagian *Enter label*, nama yang dimasukkan perlu mewakili setiap kelas data dan klik *Upload data*. Kemudian tekan *No* pada pada soalan *Are you building an object detection project?*

Upload mode

Select individual files ?

Select a folder ?

Select files

Choose Files 360 files

Upload into category

Automatically split between training and testing ?

Training

Testing

Label

Infer from filename ?

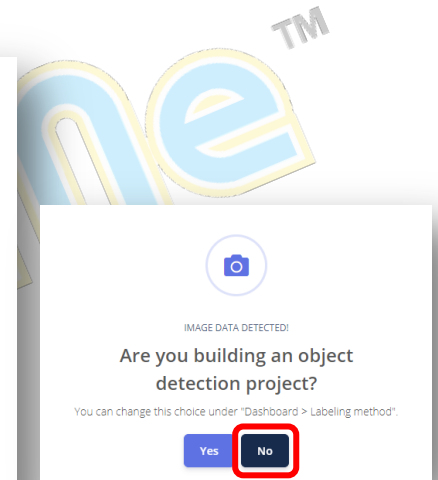
Leave data unlabeled ?

Enter label:

sun

< Back

Upload data



Langkah 9

Setelah imej data berjaya dimuat naik. *Job completed* akan dipaparkan. Ulang langkah 7 untuk setiap kelas data seterusnya. Pastikan nama label diubah mengikut nama kelas.

Upload mode

Select individual files ?

Select a folder ?

Select files

No file chosen

Upload into category

Automatically split between training and testing ?

Training

Testing

Label

Infer from filename ?

Leave data unlabeled ?

Enter label:

Upload output

```
[343/360] Uploading sun_1721880317743.jpg OK
[344/360] Uploading sun_1721880316250.jpg OK
[345/360] Uploading sun_1721880318746.jpg OK
[346/360] Uploading sun_1721880317256.jpg OK
[347/360] Uploading sun_1721880316497.jpg OK
[348/360] Uploading sun_1721880317495.jpg OK
[349/360] Uploading sun_1721880318497.jpg OK
[350/360] Uploading sun_1721880317992.jpg OK
[351/360] Uploading sun_1721880316999.jpg OK
[352/360] Uploading sun_1721880318249.jpg OK
[353/360] Uploading sun_1721880318992.jpg OK
[354/360] Uploading sun_1721880319242.jpg OK
[355/360] Uploading sun_1721880319756.jpg OK
[356/360] Uploading sun_1721880319493.jpg OK
[357/360] Uploading sun_1721880319993.jpg OK
[358/360] Uploading sun_1721880320242.jpg OK
[359/360] Uploading sun_1721880320502.jpg OK
[360/360] Uploading sun_1721880320745.jpg OK
```

Done. Files uploaded successful: 359. Files that failed to upload: 1.

Job completed ←

← Back Upload data

Langkah 10

Di bahagian kanan atas. Klik pada *target* seperti yang ditunjukkan. Kemudian tetapkan *target device* kepada *Espressif ESP-EYE (ESP32 240MHz)* dan tekan *save*.

Target: Cortex-M4F 80MHz

S

Configure your target device and application budget

Target device

Define your target device requirements to inform model optimizations and performance calculations. No device yet? Use the default settings which you can change at any time.

Target device



Espressif ESP-EYE (ESP32 240MHz)

Processor family

ESP32

Clock rate ⓘ

240

| MHz

Max

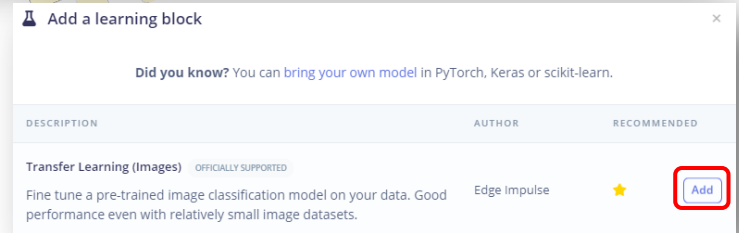
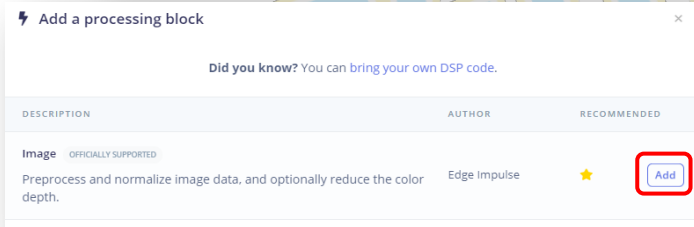
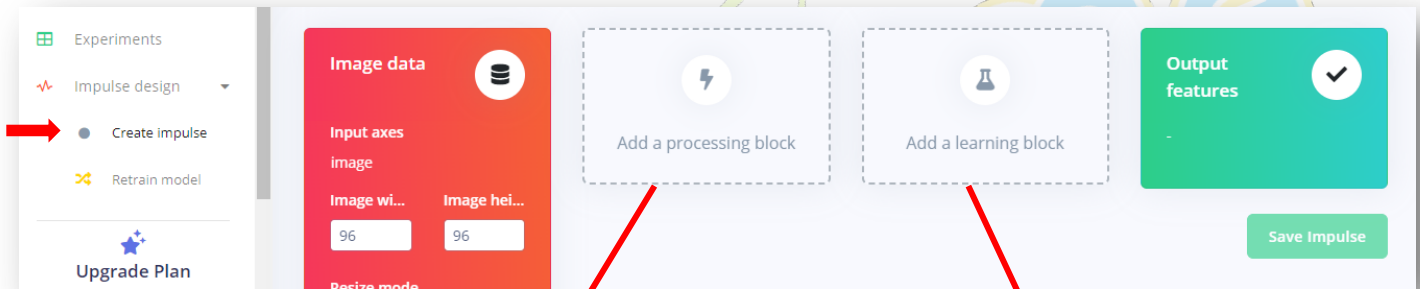
Custom device name (optional) ⓘ

Cancel

Save

Langkah 11

Pada menu di bahagian kiri, klik pada *create impulse* di dalam *impulse design*. Kemudian klik pada *Add a processing block* pilih *Image* dan klik *Add*. Seterusnya klik pada *Add a learning block*, pilih *Transfer Learning (Images)* dan klik *Add*.



Langkah 12

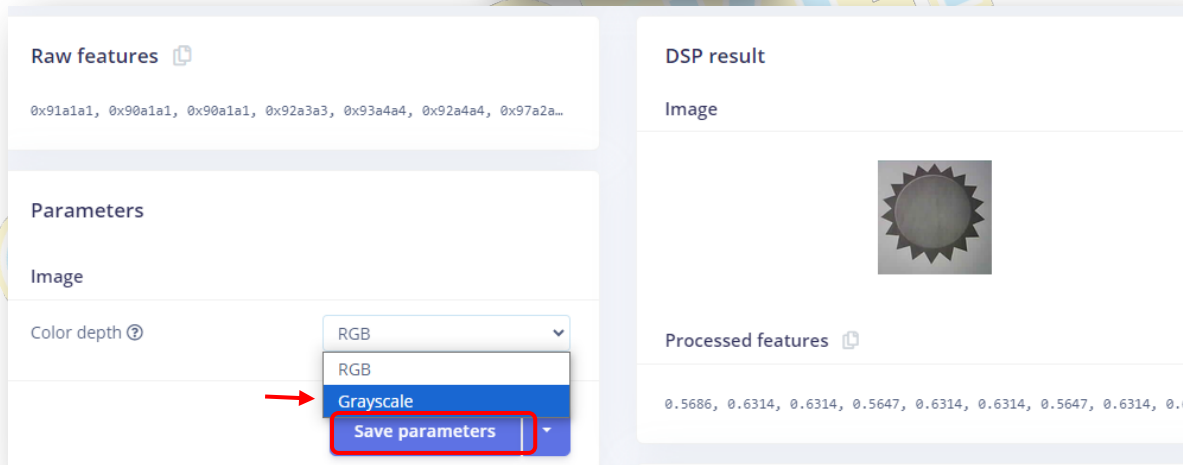
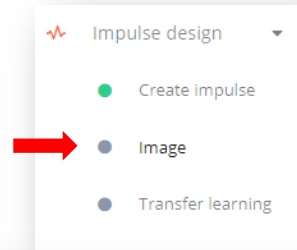
Klik pada *Save Impulse*

The screenshot displays the MikroBOTIK interface with four main panels:

- Image data (Red Panel):** Contains settings for input axes. The input is named 'image'. Both width and height are set to 96. The resize mode is set to 'Fit s'.
- Image (White Panel):** Shows the name 'Image' and the input axes configuration, which includes one axis named 'image'.
- Transfer Learning (Images) (Purple Panel):** Shows the name 'Transfer learning', the input features list containing 'Image' (checked), and the output features list containing '1 (sun)'.
- Output features (Green Panel):** Shows the output features list containing '1 (sun)'. The 'Save Impulse' button is highlighted with a red rectangular border.

Langkah 13

Tekan pada menu *Image* dan tukar *color depth* kepada *grayscale*. Ia bergantung kepada data imej yang dikumpul. Kemudian klik *Save parameters*. Data imej akan dipaparkan pada *DSP result*.



Langkah 14

Kemudian halaman *Training set* akan muncul. Tekan *Generate features* dan *Job completed (success)* akan muncul setelah selesai seperti yang ditunjukkan di bawah.

The screenshot displays the MIKROBOTIK interface with the following components:

- Training set:**
 - Data in training set: 1,040 items
 - Classes: 3 (sun, water, wind)
 - A red-bordered button labeled "Generate features" is highlighted.
- Feature generation output:**
 - Shows a list of logs including "Scheduling job in cluster...", "Container image pulled!", "Job started", "Reducing dimensions for visualizations...", "UMAP(verbose=True)", "Tue Sep 17 09:10:34 2024 Construct fuzzy simplicial set", "Tue Sep 17 09:10:36 2024 Finding Nearest Neighbors", "Tue Sep 17 09:10:38 2024 Finished Nearest Neighbor Search", "Tue Sep 17 09:10:39 2024 Construct embedding", "Epochs completed: 100% 500/500 [00:02:00:00, 166.95it/s]", "Tue Sep 17 09:10:43 2024 Finished embedding", "Writing output files...", "Writing output files OK", "Reducing dimensions for visualizations OK (took 12000ms.)", and "Job completed (success)" with a red arrow pointing to it.
- Feature explorer:**
 - Legend: sun (blue), water (orange), wind (green).
 - Visualizes data points clustered by class on a grid.
- On-device performance:**
 - PROCESSING TIME: 15 ms.
 - PEAK RAM USAGE: 4 KB.

Langkah 15

Pilih menu *Transfer learning* kemudian klik *Save & train* pada bahagian *Neural Network settings*. Data akan muncul seperti yang ditunjukkan.

Transfer learning menu:

- Create impulse
- Image
- Transfer learning** (highlighted with a red arrow)
- Retrain model

Save & train button: (highlighted with a red box)

Training output:

```

Calculating performance metrics...
Calculating inferring time...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Calculating inferring time OK
Calculating float32 accuracy...
Calculating inferring time OK
Calculating float32 accuracy...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Calculating int8 accuracy...

Model training complete

Model training complete

Job completed (success)
    
```

Metrics (validation set):

METRIC	VALUE
Area under ROC Curve	1.00
Weighted average Precision	1.00
Weighted average Recall	1.00
Weighted average F1 score	1.00

Data explorer (full training set):

- sun - correct
- water - correct
- wind - correct

Model: Model version: Quantized (int8)

Last training performance (validation set):

- ACCURACY: 100.0%
- LOSS: 0.00

Confusion matrix (validation set):

	SUN	WATER	WIND
SUN	100%	0%	0%
WATER	0%	100%	0%
WIND	0%	0%	100%
F1 SCORE	1.00	1.00	1.00

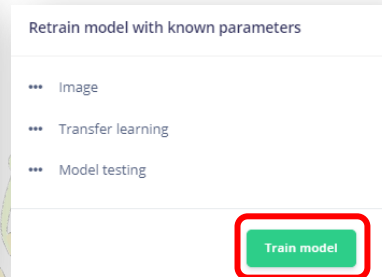
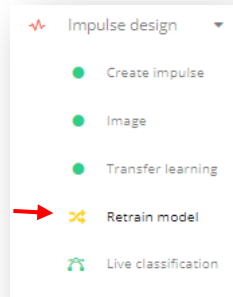
On-device performance:

- INFERRING TIME: 1690 ms.
- PEAK RAM USAGE: 334.6K
- FLASH USAGE: 585.2K

Engine: EON™ Compiler

Langkah 16

Pilih menu *Retrain model* dan klik pada *Train model*. Kemudian *Build output* akan memaparkan *Job completed (success)* dan semua parameter akan bertanda ✓.



Langkah 17

Klik pada menu *Model testing* dan tekan *Classify all* pada bahagian *Test data*. *Job completed (success)* akan dipaparkan pada *Model testing output*.

The screenshot shows the MCT interface. On the left, a sidebar menu has 'Model testing' selected with a red arrow. The main area shows the 'Test data' section with a 'Classify all' button highlighted in a red box. Below this is a table with the following data:

SAMPLE NAME	EXPECTED OUTCOME	ACCURACY	RESULT
wind_1723425...	wind	100%	1 wind
wind_1723425...	wind	100%	1 wind

Below the table is a 'Model testing output' window with the following log:

```

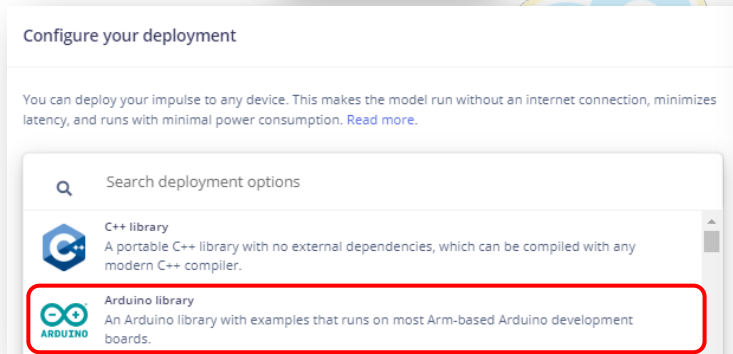
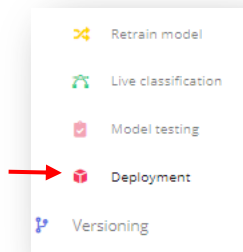
Classifying data for Transfer learning...
Classifying data for float32 model...
Scheduling job in cluster...
Container image pulled!
Job started
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Classifying data for Transfer learning OK

Generating model testing summary...
Finished generating model testing summary

Job completed (success)
    
```

Langkah 18

Klik pada menu *Deployment* dan pastikan *configure your deployment* adalah jenis *Arduino library*. kemudian pastikan pilihan model seperti yang ditunjukkan di bawah dan tekan *build*.



MODEL OPTIMIZATIONS
Model optimizations can increase on-device performance but may reduce accuracy.

EON™ Compiler
Same accuracy, 17% less RAM, 14% less ROM.

Quantized (int8) Selected ✓

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	15 ms.	1,690 ms.	1,705 ms.
RAM	4.0K	334.6K	334.6K
FLASH	-	585.2K	-
ACCURACY			-

Unoptimized (float32) Select

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	15 ms.	4,107 ms.	4,122 ms.
RAM	4.0K	893.7K	893.7K
FLASH	-	1.6M	-
ACCURACY			100.00%

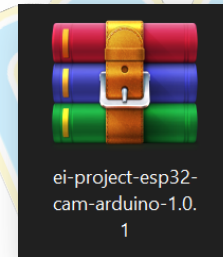
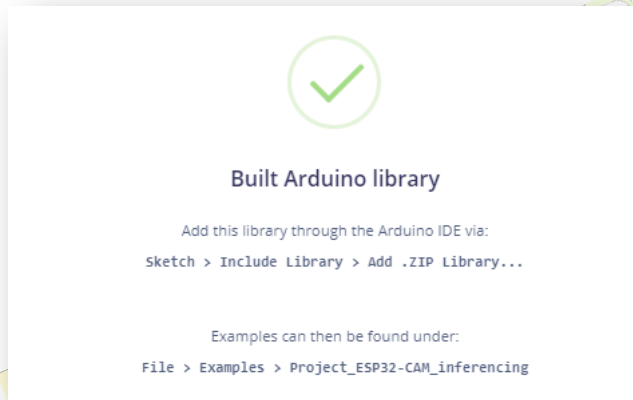
To compare model accuracy, run model testing for all available optimizations. **Run model testing**

Estimate for Espressif ESP-EYE (ESP32 240MHz) - [Change target](#)

Build

Langkah 19

Paparan seperti di bawah akan muncul dan fail Model Kecerdasan Buatan (AI) akan disimpan dalam *Download*.



Fail Model Kecerdasan
Buatan (AI)

Muat turun data ke dalam ESP32-CAM

Langkah 1

Klik pada laman: <https://www.microconcept.com.my/download/>. Cari dan klik pada *Mikrobotik ESP32CAM Arduino Edge Impulse v2* untuk muat turun templat pengaturcaraan. Kemudian, *Extract files*. Buka folder dan klik pada fail template pengaturcaraan *Mikrobotik_ESP32CAM_EdgeImpulse_v2.ino*. Buka template pengaturcaraan Arduino di dalamnya.

We offer a vibrant bilingual experience in English and Malay.

- Two Languages, Your Choice!

- Mikrobotik Modul eBook v1.4 English
- Mikrobotik Modul eBook v1.5 Bahasa Malaysia

- Dive deeper into learning with our bonus content!

- Mikrobotik Track Examples
- Mikrobotik mBlock Project v6
- Mikrobotik mBlock Troubleshooting v2
- Mikrobotik ESP32CAM Arduino Edge Impulse v2**



Mikrobotik_ESP32CAM_EdgeImpulse_v2.zip

- Open with WinRAR
- Extract files...
- Extract to "Mikrobotik_ESP32CAM_EdgeImpulse_v2\"
- Extract Here



Mikrobotik_ESP32CAM_EdgeImpulse_v2

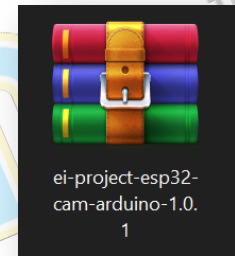
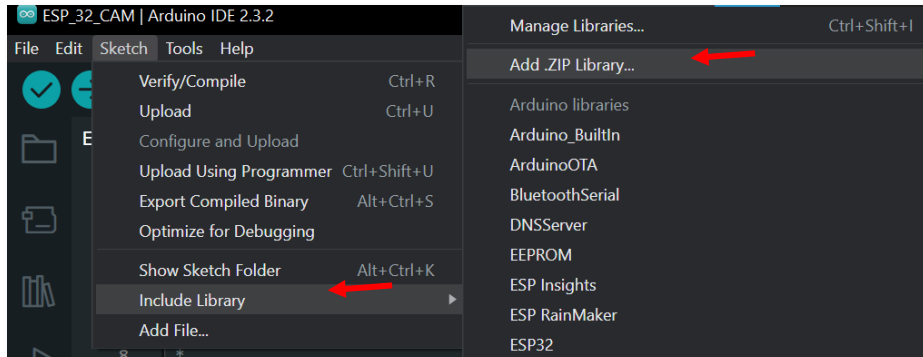


Mikrobotik_ESP32CAM_EdgeImpulse_v2.ino

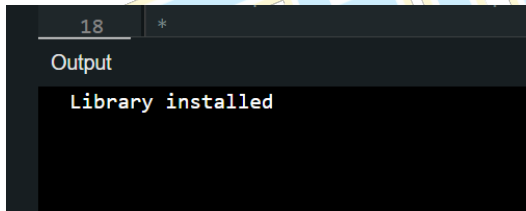
Templat Pengaturcaraan

Langkah 2

Klik pada menu *Sketch*, pilih *include Library* dan *Add .ZIP Library...*. Tambahkan fail *Model Kecerdasan Buatan AI* yang telah dimuat turun. *Library installed* akan dipaparkan pada bahagian *Output*.

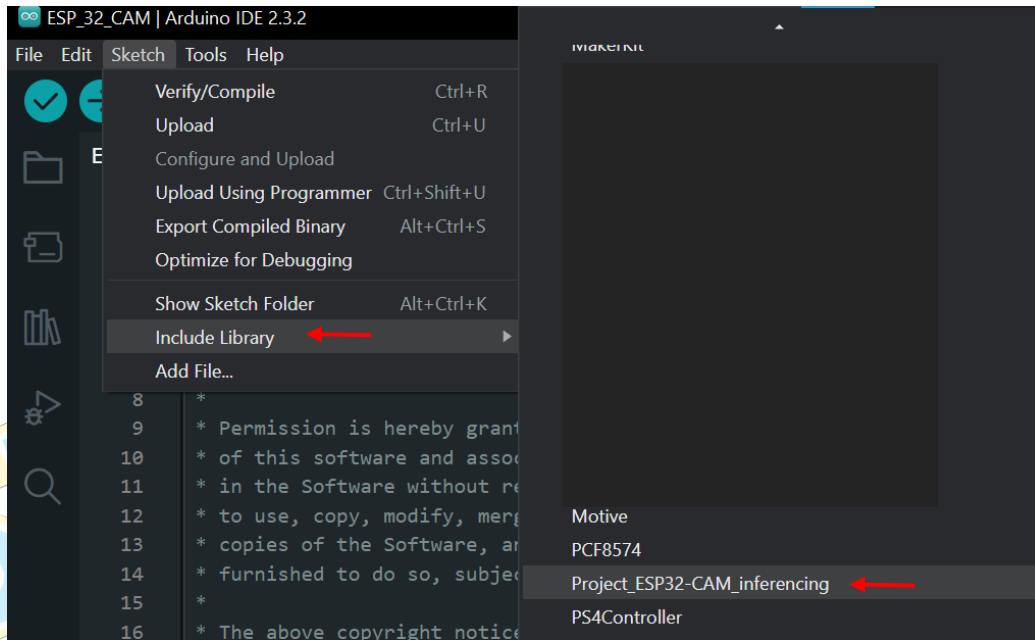


Fail Model Kecerdasan Buatan (AI)



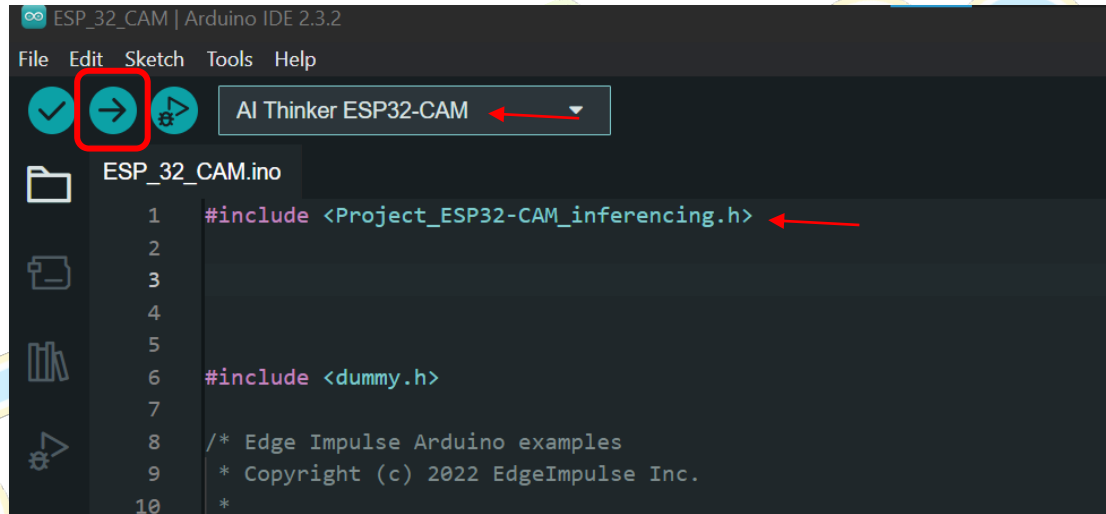
Langkah 3

Klik pada menu *Sketch*, pilih *Include Library* dan pilih *Project_ESP32-CAM_inferencing*.



Langkah 4

Nama library akan muncul pada bahagian atas coding. Kemudian sambungkan ESP32-CAM dan klik pada ikon anak panah. Pastikan *board* yang dipilih adalah *AI Thinker ESP32-CAM* dan port dipilih dengan betul.



Langkah – langkah susunan blok

Langkah 1

Seretkan blok *When Mikrobotik starts* dan blok *Robot Prepare*.



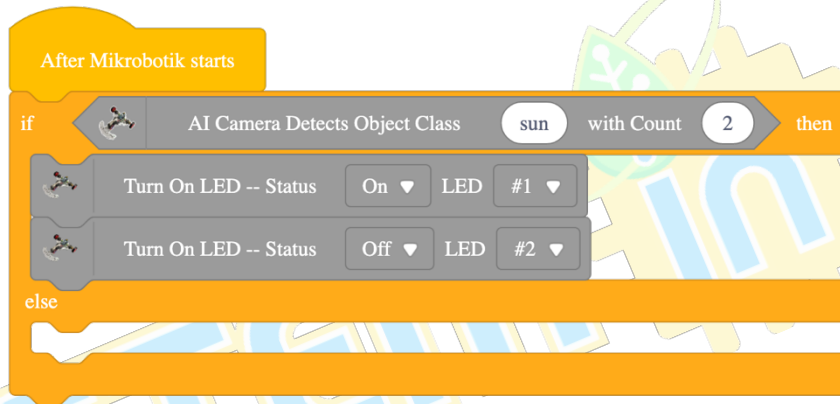
Langkah 2

Seretkan blok *After Mikrobotik starts* dan blok *if then else*.



Langkah 3

Kemudian seretkan blok *AI Camera Detect Object Class* dari menu *camera* ke dalam ruang heksagon. Seretkan 2 blok *Turn On LED – Status* ke dalam blok *if then* dan tetapkan *On* untuk *LED 1* dan *Off* untuk *LED 2*.



Object class adalah nama kelas yang digunakan dalam pemrosesan data dalam Edge Impulse. Nama yang digunakan mestilah sama supaya data dapat dikesan.

Count mewakili berapa kali pengesanan yang perlu dilakukan untuk memastikan data yang dikesan adalah benar. Semakin tinggi nilai count, semakin tepat data yang dikesan.

Langkah 4

Duplicate blok *if then else* di atas dan masukkan ke dalam ruangan *else*. Tukarkan *Object Class* kepada *water*. Tetapkan *status Off* untuk *LED 1* dan *On* untuk *LED 2*.

```

    After Mikrobotik starts
    if AI Camera Detects Object Class sun with Count 2 then
        Turn On LED -- Status On LED #1
        Turn On LED -- Status Off LED #2
    else
        if AI Camera Detects Object Class water with Count 2 then
            Turn On LED -- Status Off LED #1
            Turn On LED -- Status On LED #2
        else
    
```

Langkah 5

Ulang Langkah 4 dan tukarkan *Object Class* kepada *wind*. Tetapkan *status On* untuk kedua-dua *LED*.

```

After Mikrobotik starts
if AI Camera Detects Object Class sun with Count 2 then
  Turn On LED -- Status On LED #1
  Turn On LED -- Status Off LED #2
else
if AI Camera Detects Object Class water with Count 2 then
  Turn On LED -- Status Off LED #1
  Turn On LED -- Status On LED #2
else
if AI Camera Detects Object Class wind with Count 2 then
  Turn On LED -- Status On LED #1
  Turn On LED -- Status On LED #2
else
  
```

Langkah 6

Kemudian seretkan 2 blok *Turn On LED – Status* dan blok *Play Music*. Tetapkan *status Off* untuk kedua-dua LED. Tetapkan *Note* kepada *D5* dan *Beat* kepada *Half*.

```

    When Mikrobotik starts
      Robot Prepare

    After Mikrobotik starts
      if AI Camera Detects Object Class sun with Count 2 then
        Turn On LED -- Status On LED #1
        Turn On LED -- Status Off LED #2
      else
        if AI Camera Detects Object Class water with Count 2 then
          Turn On LED -- Status Off LED #1
          Turn On LED -- Status On LED #2
        else
          if AI Camera Detects Object Class wind with Count 2 then
            Turn On LED -- Status On LED #1
            Turn On LED -- Status On LED #2
          else
            Turn On LED -- Status Off LED #1
            Turn On LED -- Status Off LED #2
            Play Music -- Note D5 Beat Half
  
```

Tambahan: Cuba Naik Taraf dan Pengaturcaraan Sendiri

POT PERANTI	PIN ARDUINO NANO	PERANTI	MAKLUMAT TAMBAHAN
ITR1	A6	Sensor Pengesan Garisan – Kiri Luar	ITR8307
ITR2	A3	Sensor Pengesan Garisan – Kiri Dalam	ITR8307
ITR3	A2	Sensor Pengesan Garisan – Tengah	ITR8307
ITR4	A1	Sensor Pengesan Garisan – Kanan Dalam	ITR8307
ITR5	A0	Sensor Pengesan 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