

Explore STEM & Coding with

EDU:BIT

on start

start melody power up ▾ repeating once ▾

set all RGB pixels to 

forever

if IR sensor triggered then

Set servo S1 ▾ position to 40 degrees

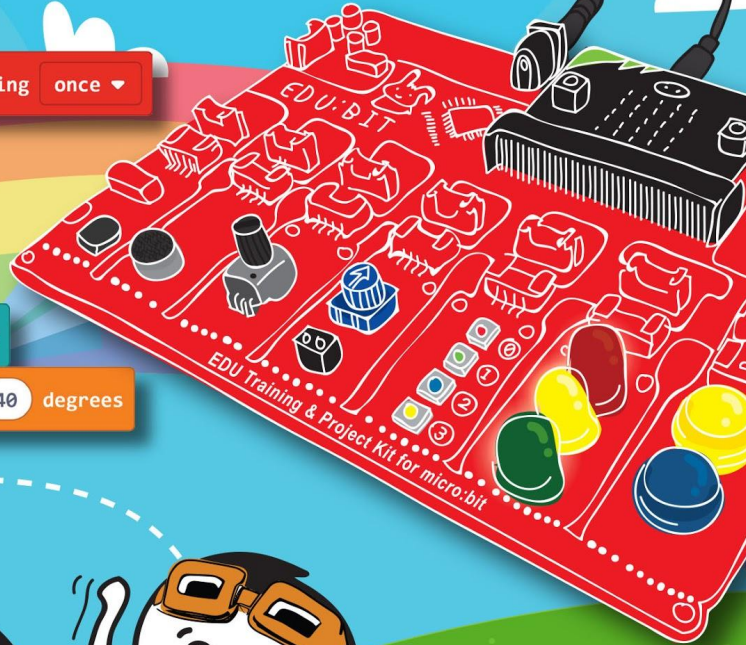
show leds



else

Set servo S1 ▾ position to 20 degrees

show arrow East



Pesanan daripada rero EDUteam @ Cytron

_____ yang dikasihi,
(Nama)

Pernakah anda dengar mengenai micro:bit? Ia adalah sebuah papan litar mini boleh atur cara yang direka cipta di United Kingdom dan telah diagihkan ke seluruh dunia bagi tujuan mencetus minat kanak-kanak kepada pembelajaran pengekodan secara seronok dan mudah difahami.

Jurutera kami di Cytron telah berusaha dan berjaya mereka bentuk papan litar EDU:BIT bagi memudahkan anda belajar pengekodan secara tuntas. Di antara komponen-komponen utama EDU:BIT adalah **Music Bit** yang dilengkapi dengan pembaz piezo dan bicu audio (*audio jack*) bagi membolehkan anda memainkan muzik, **Sound Bit** pula boleh mengesan bunyi, **Potentio Bit** sebagai pengawal isyarat analog, **IR Bit** bagi mengesan objek, **RGB Bit** membolehkan paparan cahaya berwarna-warni menggunakan 4 diod pancaran cahaya RGB, **Traffic Light Bit** pula dilengkapi dengan 3 diod pancaran cahaya berwarna merah, kuning dan hijau dan akhir sekali, **Button Bit** adalah butang input micro:bit berskala besar. Tambahan pula, kit ini juga dilengkapi dengan sebuah **motor DC** dan **motor servo**. Teruja tak?

Jom, kita mula belajar. Pada muka surat seterusnya, kita akan tinjau dan mereka cipta semula beberapa permainan klasik kanak-kanak seperti, *Rock Paper Scissors*, *Snakes and Ladders*, *Tag*, *You're It!*, *Talent Time Show*, *Twister*, *Simon Says* dan banyak lagi permainan yang menyeronokkan. Setelah berjaya mengekodkan atur cara, anda boleh bermain bersama rakan-rakan permainan klasik tersebut. Selain itu, anda digalakkan mengubah suai dan menambah baik atur cara yang dikod supaya lebih menyeronokkan dan unik.

Pada bahagian akhir setiap bab, kami juga telah menyertai beberapa aktiviti cabaran bagi menguji kefahaman anda mereka bentuk aplikasi untuk bilik darjah. Mari kita cuba dan jika anda menghadapi masalah, kami sentiasa akan bersedia membantu.

Adakah anda bersedia? Jom, kita mulakan pengembaraan dan pembelajaran yang menyeronokkan.

Adam & Anna



Kenali STEM & Pengekodaan dengan Kit EDU:BIT

Pengarang: Cheryl Ng, SC Lim & Adrian Teo

Ilustrasi oleh Suhana Oazmi

Diterjemahkan oleh

Muallim Muhammad Fadhil, SRI Hira Klang

Cikgu Mohammed Ariffin Zainal Abidin, ProjectMika London

Cikgu Ardian Hamid, ProjectMika Kuala Lumpur

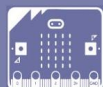
Cikgu Lee Setz Tsin, SJKC Poi Yuk, Sarawak

Cikgu Rizali Afiq bin Razak, Maktab Mahmud Alor Setar, Kedah

Diterbitkan oleh



Senarai Kandungan



Bab 1: Hello, World! (Papan Matriks LED pada micro:bit)
On start dan Forever

1 - 11



Bab 2: Jom Bermain Batu, Gunting, Kertas! (Button Bit)
Pemboleh ubah dan pengaturcaraan tetapan peristiwa

12 - 25



Bab 3: Jom Bermain Muzik~ (Music Bit)
Fungsi di dalam pengaturcaraan

26 - 38



Bab 4: Win, Lose or Draw~ (Traffic Light Bit)
Output digital

39 - 47



Bab 5: Jom Bermain Dadu Digital IR (IR Bit)
Input digital, *arrays* dan *while loops*

48 - 60



Bab 6: Tag, You're It! (Potentio Bit)
Input analog, pengaturcaraan bersyarat

61 - 74



Bab 7: Jom kita dengar tepukan! (Sound Bit)
Penukaran mod di dalam program

75 - 87



Bab 8: Jom Putar! (DC Motor)
Mengawal arah putaran dan kelajuan motor DC

88 - 95



Bab 9: Tendangan Penalti.. Gol!!! (Motor Servo)
Kawalan posisi sudut motor servo

96 - 104



Bab 10: Mastermind, bolehkah anda pecahkan kod? (RGB Bit)
Model warna RGB

105 - 113

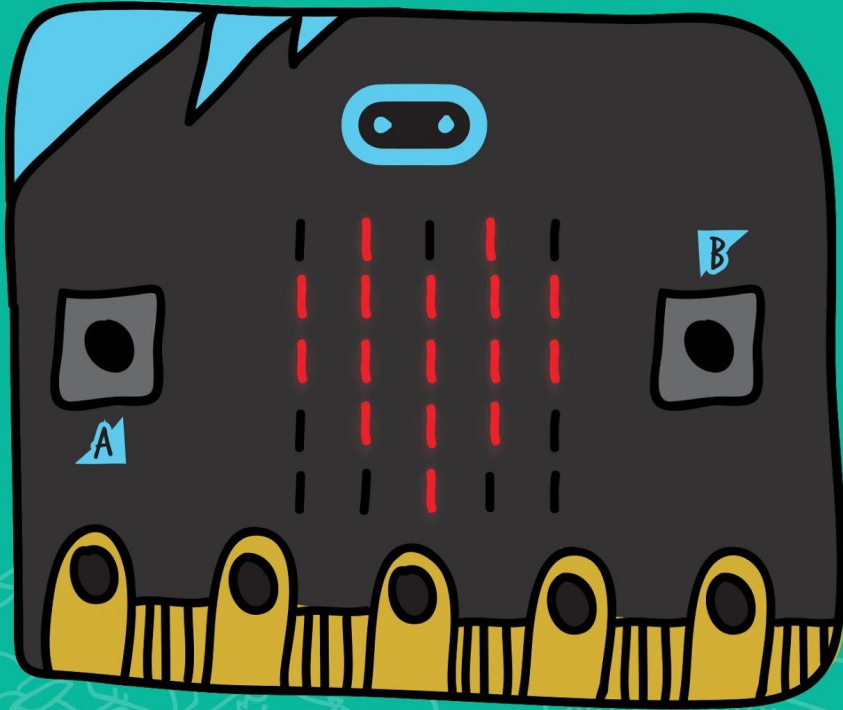


Bab Bonus: Simon Says dengan LED.
Komunikasi Radio

114 - 124

> Hello, World! _

Paparan Matriks LED pada micro:bit

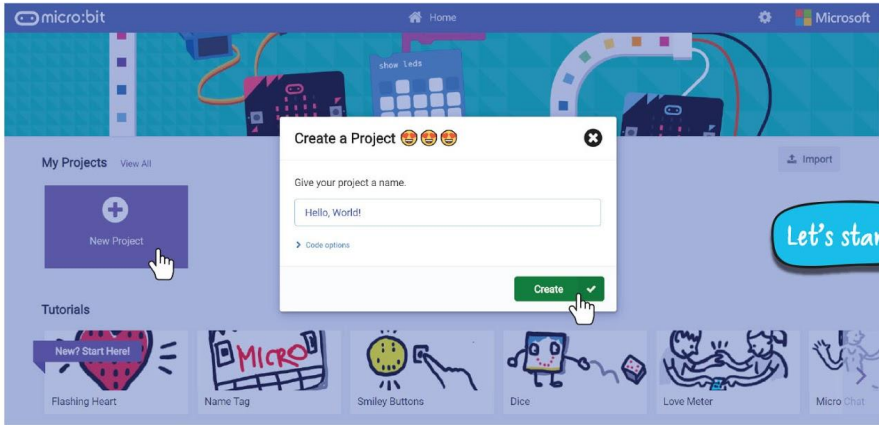


Imbas saya!

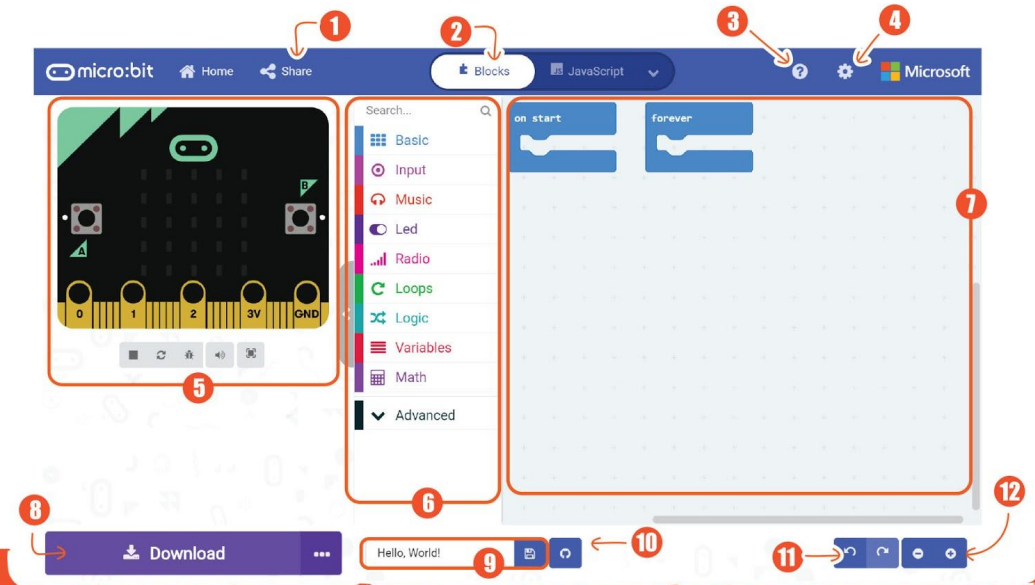
link.cytron.io/edubit-chapter-1

JOM KITA KOD!

Langkah 1 Layari laman sesawang <https://makecode.microbit.org>. Klik pada paparan **New Project**. Taipkan nama projek dan klik **Create**.

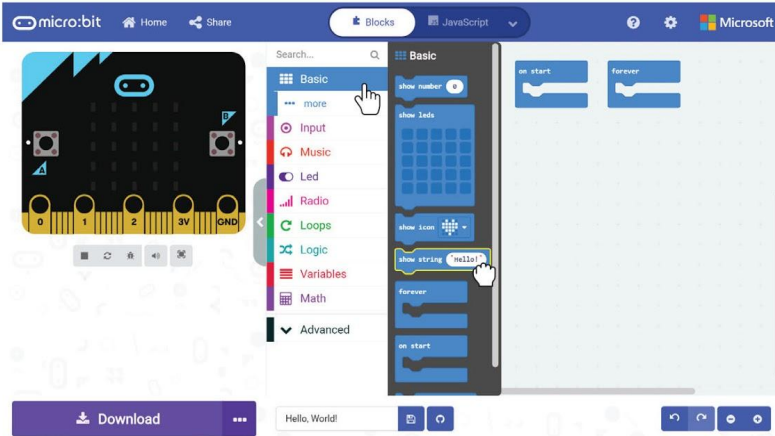


Laman sesawang Microsoft MakeCode Editor akan membolehkan anda mengekodkan atur cara EDU:BIT dengan menggunakan kaedah tarik dan lepas.

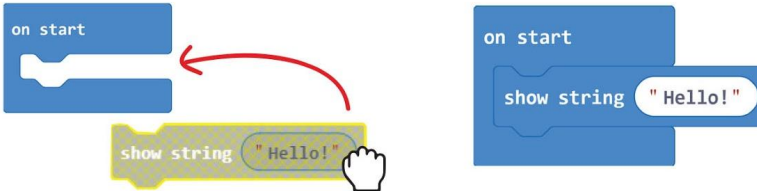


1. Terbit dan kongsi projek anda
2. Pilih untuk pengekod blok, bahasa atur cara JavaScript atau Python.
3. Paparan bantuan.
4. Ubah penentu, tambah komponen dan pasangan peranti.
5. Simulator - Paparan simulasi atur cara pada micro:bit.
6. *Toolbox* / Kotak Kategori Blok - Klik pada setiap kategori bagi melihat blok pengekodan yang tersedia. Setiap kategori blok akan mempunyai warna yang sama.
7. Ruang Kerja Pengekodan - Tempat atur cara dikod menggunakan blok dengan kaedah tarik dan lepas.
8. Muat Turun atur cara yang telah dikod kepada micro:bit
9. Nama dan simpan atur cara pada komputer
10. Bina dan kemas kini repositori Github
11. Buat asal / buat semula
12. Zum masuk, zum keluar

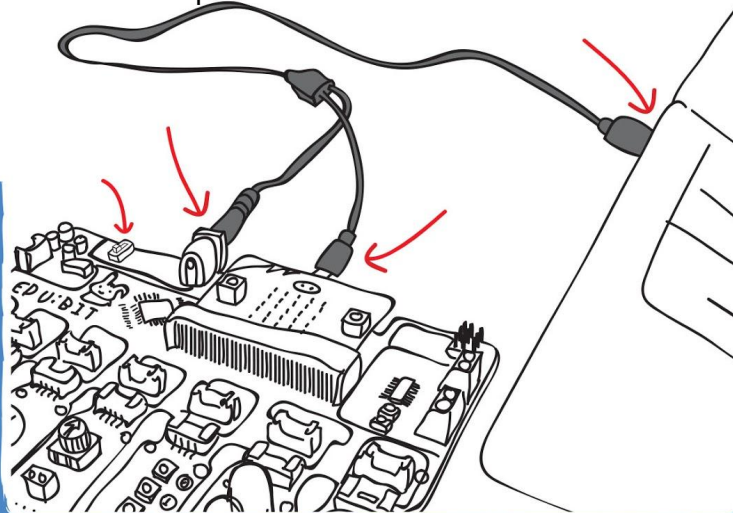
Langkah 2 Klik [Basic] dan kemudian pilih blok [show string].



Langkah 3 Klik dan masukkan blok [show string] ke dalam slot [on start].

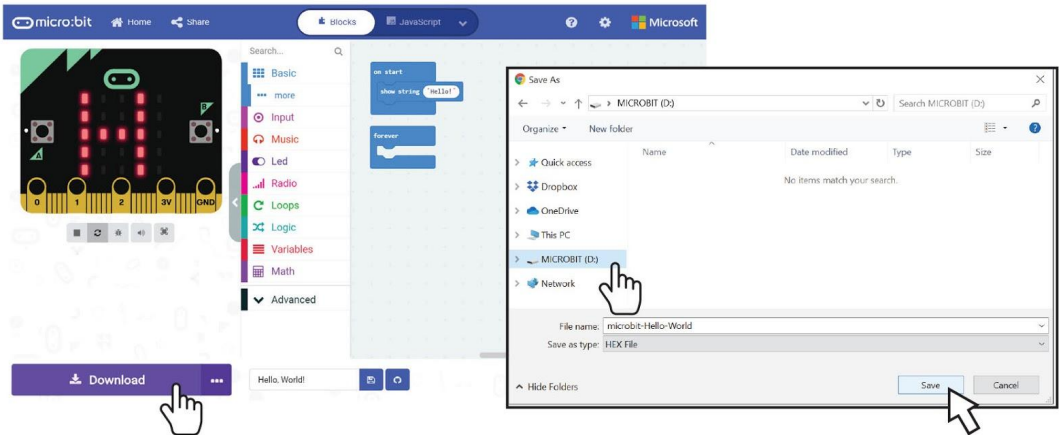


Langkah 4 Sambungkan kabel USB kepada komputer dan EDU:BIT anda seperti yang ditunjukkan di bawah. Jangan lupa untuk menghidupkan kuasa EDU:BIT dengan memetik suis ke posisi ON.





Langkah 5 Klik butang [**Download**]. Pada paparan tettingkap pop-up, pilih untuk memuat turun projek kepada pemacu MICROBIT. Selepas paparan menyatakan *Download completed*, tutup tettingkap tersebut.

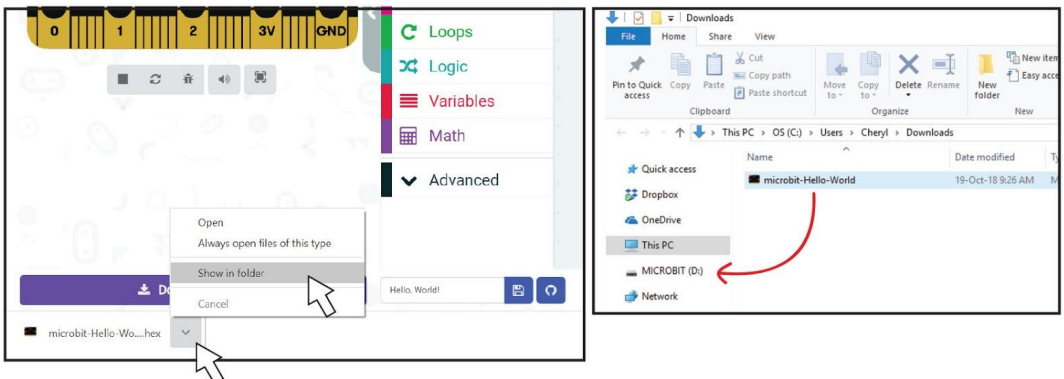


Flashing adalah proses pemindahan kod kepada micro:bit. LED berwarna oren di bahagian belakang micro:bit akan berkelip-kelip semasa proses ini sedang dilaksanakan. Setelah proses ini lengkap, kod pada micro:bit akan dimulakan secara automatik.

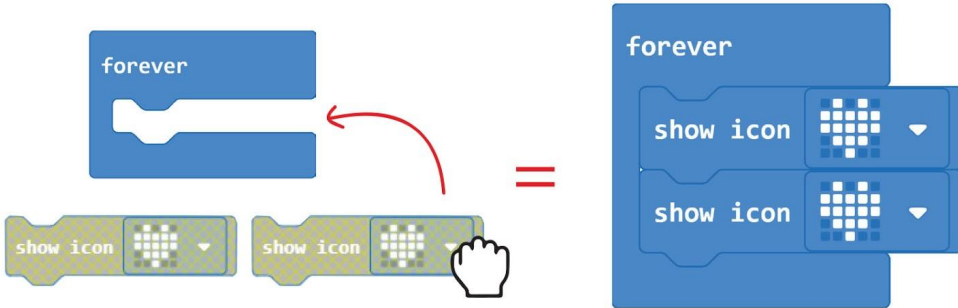


PERHATIAN !

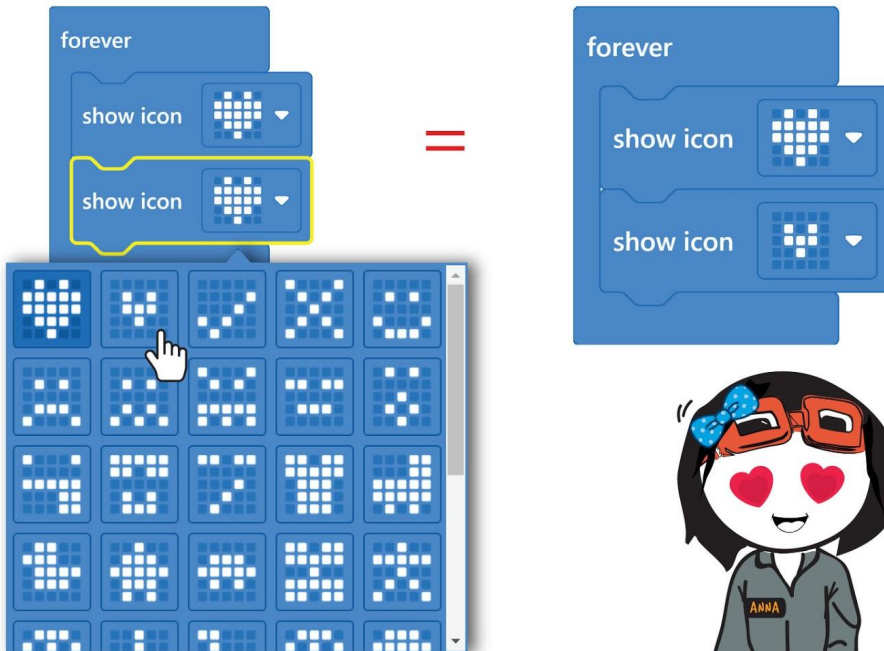
Jika tettingkap *pop-up* tidak kelihatan, ini bermakna fail kod tersebut telah dimuat turun secara automatik ke lokasi yang telah ditetapkan pada pelayar Internet anda. Klik butang kanan pada tetikus pada fail .hex pada bahagian bawah tettingkap dan pilih *Show in folder*. Klik pada fail "microbit-xxx.hex" dan kemudian tarik dan lepas pada pemacu MICROBIT, sama seperti proses memindahkan fail kepada pemacu kilat.



Langkah 6 Klik [**Basic**] dan kemudian pilih blok [**show icon**]. Ulang semula untuk menambah satu lagi blok [**show icon**]. Klik dan masukkan blok [**show icon**] ke dalam slot [**forever**].



Langkah 6 Klik butang kiri pada tetikus bagi ikon blok kedua [**show icon**] dan pilih corak 'small heart' pada tetingskap *pop-up*. Pindahkan kod tersebut ke dalam EDU:BIT anda.



Ini adalah cinta pandangan pertama. Adakah anda nampak animasi denyutan hati?

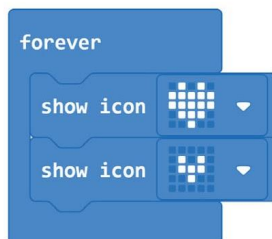
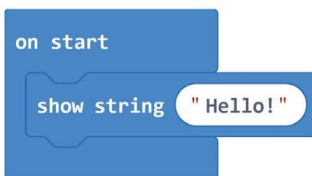
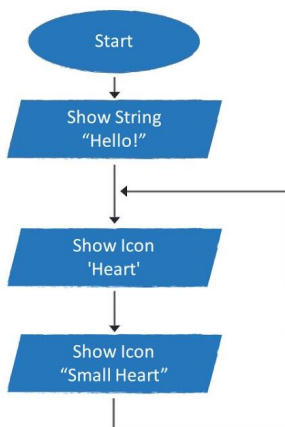


TAHUKAH

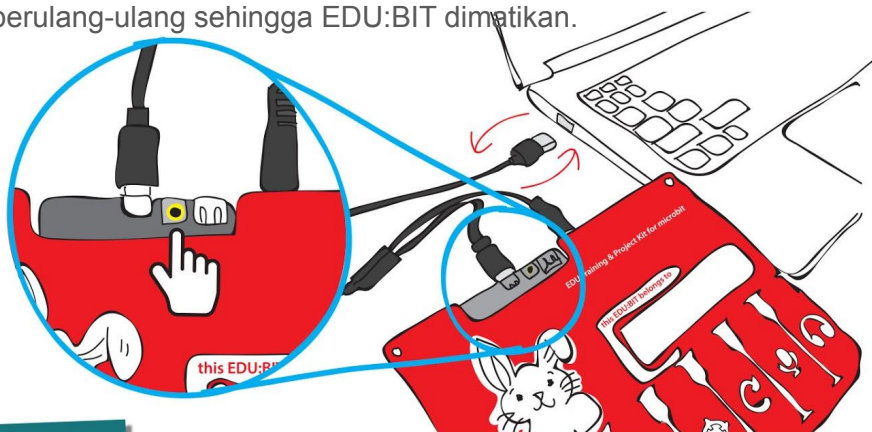
ANDA??



Perhatikan bahawa teks "Hello!" hanya akan dipamerkan secara menatal (*scrolling*) sekali sahaja dan sebaliknya, ikon denyutan hati akan berulang-ulang. Kenapa?



Kod dalam blok [on start] akan dilaksanakan sekali sahaja apabila aplikasi mula-mula dihidupkan. Manakala kod dalam blok [forever] akan dilaksanakan secara berulang-ulang sehingga EDU:BIT dimatikan.

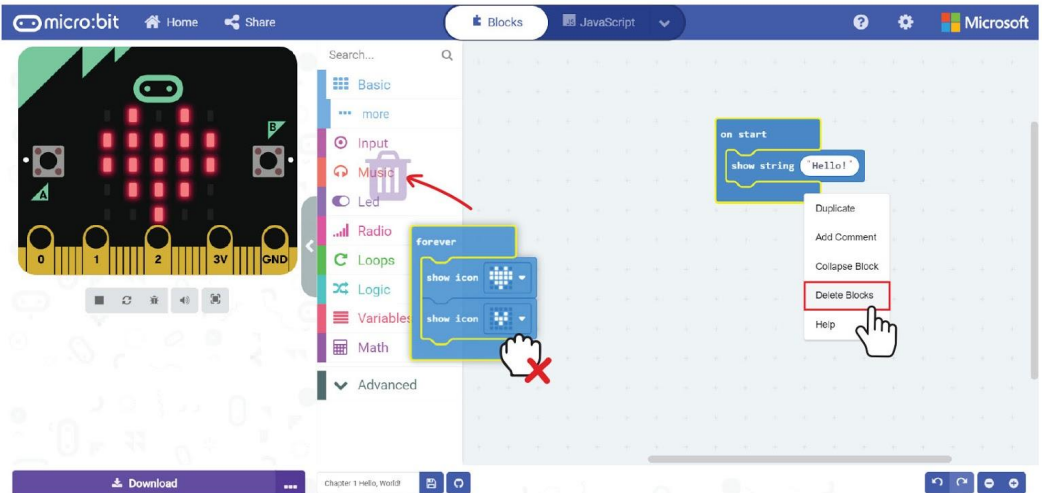


PERHATIAN !

Jika anda ingin melaksanakan semula program, anda hanya perlu tekan butang *RESET* di belakang micro:bit atau cabut dan pasang semula kabel USB pada komputer.

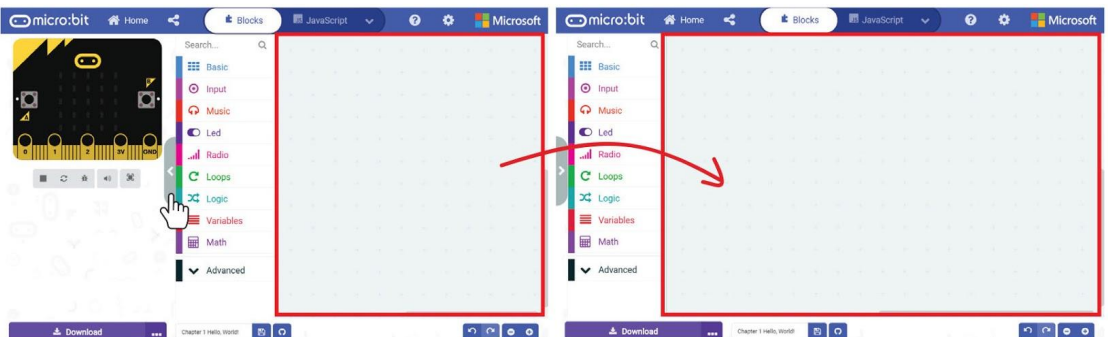
Tip Pantas #1!

Cara mudah untuk menghapuskan blok adalah dengan klik dan tarik blok tersebut ke bahagian 'Toolbox'. Lepaskan blok tersebut setelah ikon tong sampah kelihatan. Selain itu, anda juga boleh menekan butang kiri tetikus pada blok yang ingin dihapuskan dan kemudian memilih 'Delete Block'.



Tip Pantas #2!

Jika anda tidak memerlukan simulator, klik pada tab seperti ditunjukkan di bawah bagi menyembunyikan tingkap simulator supaya anda mempunyai ruang kerja yang lebih besar.





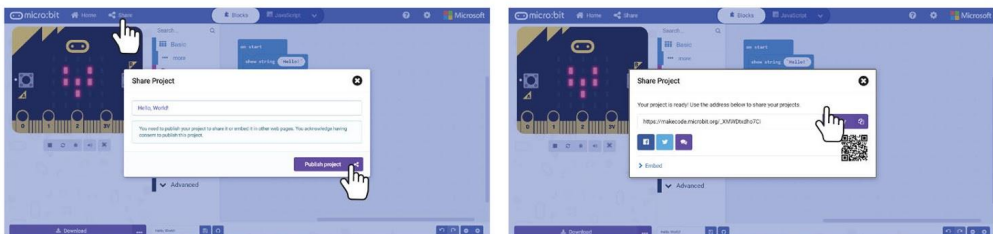
Tip Pantas #3!

Blok-blok pada kluster yang sama boleh dikumpulkan dengan menekan butang kanan tetikus dan pilih 'Collapse Block'. Bagi mempamerkan semula kluster blok-blok tersebut, klik pada ikon (v).



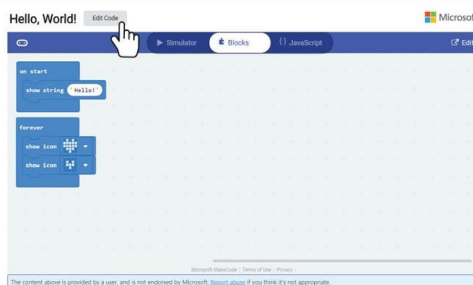
Tip Pantas #4!

Anda boleh kongsi kod projek anda kepada guru dan rakan-rakan dengan menerbitkan projek tersebut dan kemudian menghantar alamat URL projek anda kepada mereka. Klik butang [share] dan seterusnya klik [Publish project] pada tettingkap *pop-up*. Setelah projek diterbitkan, tettingkap *pop-up* akan mempamerkan alamat URL projek tersebut. Klik [Copy] dan kongsikan alamat URL tersebut.



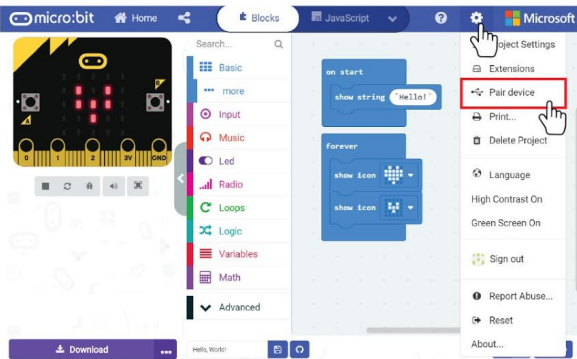
Tip Pantas #5!

Guru dan rakan anda akan dapat melihat laman web seperti rajah di bawah apabila mereka membuka alamat URL projek anda. Mereka boleh menekan butang [Edit Code] untuk melihat dan mengedit kod anda.



Tip Pantas #6!

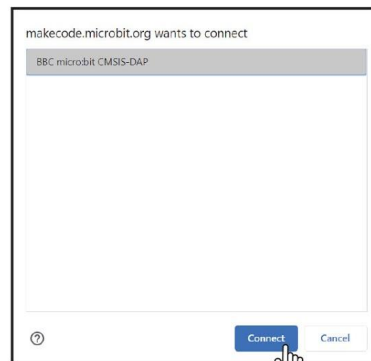
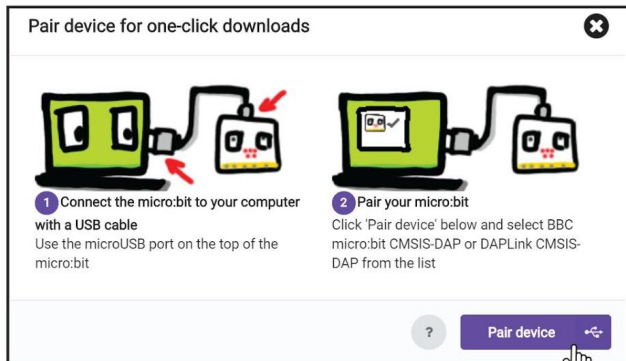
Tahukah anda bahawa anda boleh memuat turun terus kepada peranti pemasangan dengan satu klik? Bagi membolehkan kaedah ini, anda perlu menekan butang ikon gandar dan pilih 'Pair device'.



PERHATIAN !

Anda perlu memastikan bahawa micro:bit anda mempunyai perisian tegar (*firmware*) terkini dan juga pelayar Internet Microsoft Edge baru atau Google Chrome. Arahan terperinci untuk mengemaskini perisian tegar micro:bit boleh didapati pada alamat URL: <https://microbit.org/get-started/user-guide/firmware/>

Pastikan EDU:BIT anda disambungkan kepada PC dan klik butang [**Pair device**] pada tettingkap *pop-up*. Seterusnya, pilih BBC micro:bit CMSIS-DAP atau DAPLink CMSIS-DAP dan klik [**Connect**].



Setelah peranti dipasangkan, anda boleh terus pindahkan kod projek anda ke dalam EDU:BIT apabila butang [**Download**] ditekan. Mari kita cuba!

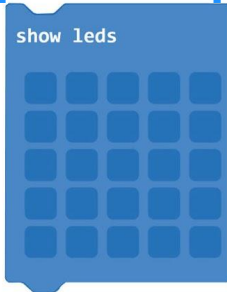
Jika anda mengalami masalah memasang peranti, anda boleh cuba melayari alamat URL berikut untuk penerangan lanjut: <https://makecode.microbit.org/device/usb/webusb/troubleshoot>





Terokai Blok-Blok Lain

#1 Gunakan blok [**show leds**] untuk reka corak ikon anda sendiri dan blok [**show number**] untuk memaparkan nombor.



Tahukah

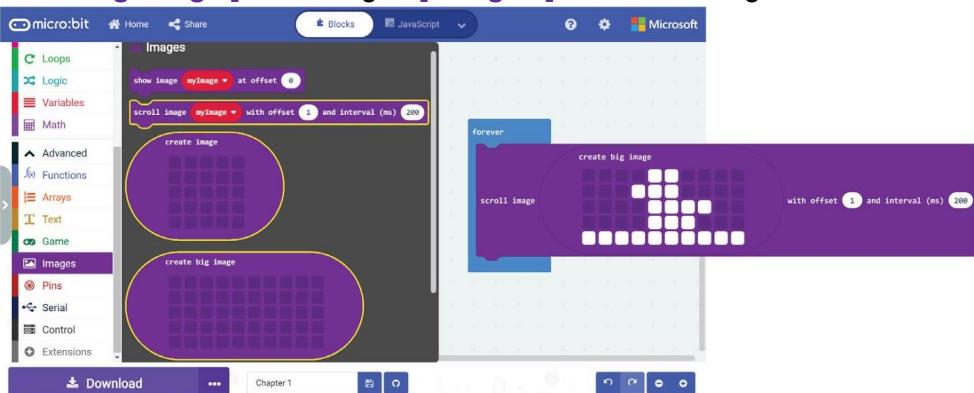
anda?

1000 milisaat (ms) = 1 saat

#2 Tambahkan blok [**pause**] untuk memperlahankan program anda. Jeda masa akan menghentikan sementara program mengikut ketetapan milisaat yang telah disetkan.



#3 Jika anda ingin paparan corak secara menatal (*scrolling*) pada matriks LED, anda boleh menggunakan blok [**scroll image _ with offset _ and interval (ms) _**] dan dipasangkan berserta blok [**create image**] atau blok [**create big image**] dari kategori [**Images**] di bawah kategori **Advanced**.



Apabila anda melaksanakan program seperti rajah di atas, anda akan dapat melihat anak itik menatal berulang kali pada skrin paparan LED micro:bit.

CABARAN APLIKASI

Programkan EDU:BIT untuk berfungsi sebagai sebuah papan notis digital.

Pada Mulanya
[On Start]

Paparkan animasi ringkas untuk menarik perhatian;
kemudian paparkan secara menatal nama kelas anda.

Sentiasa
[Forever]

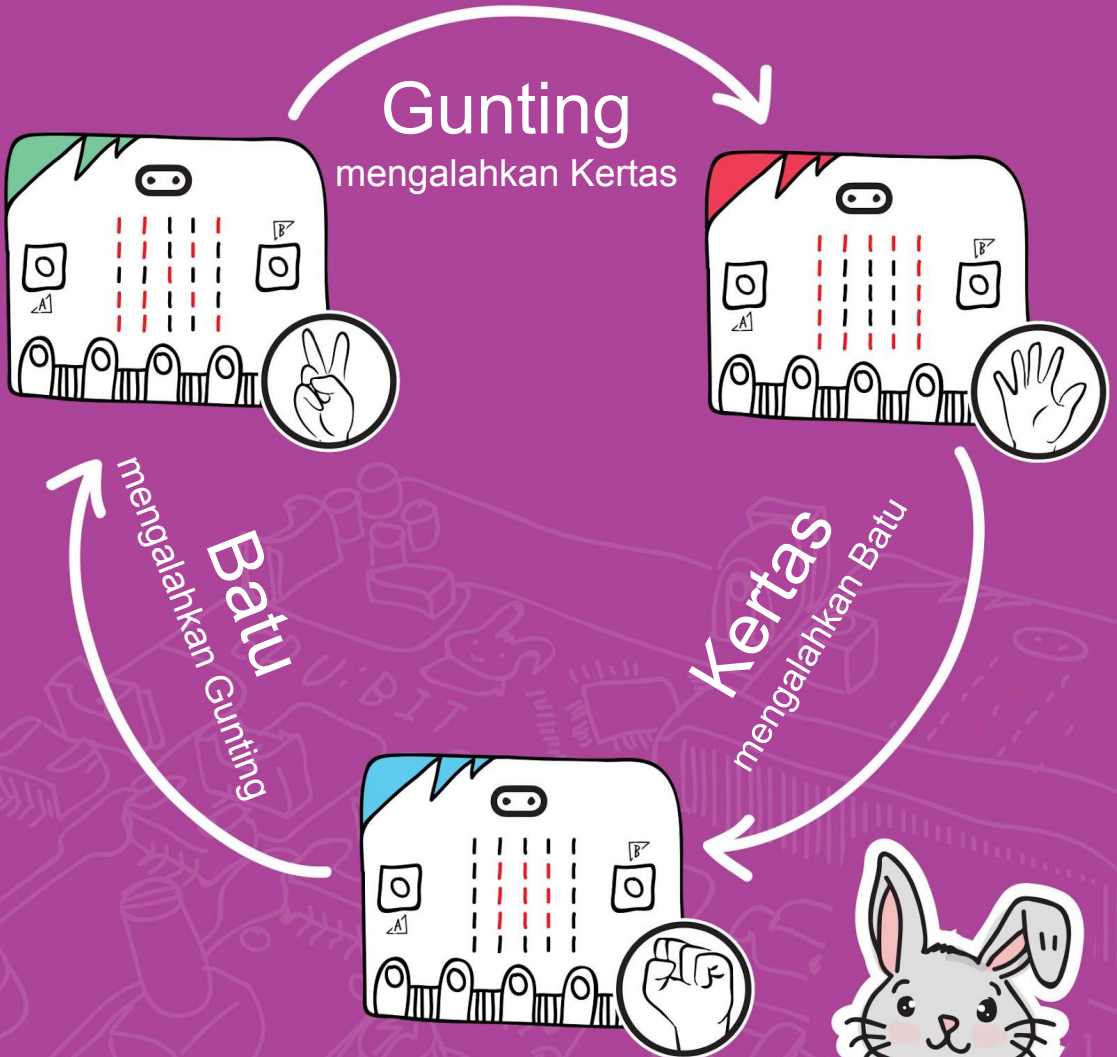
Paparkan secara menatal tarikh hari ini dan maklumat
penting kepada kelas.



Tahniah, anda telah berjaya!
Jom kita mulakan Bab 2!

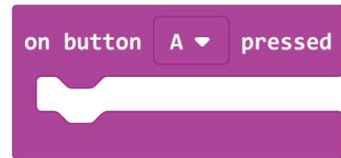
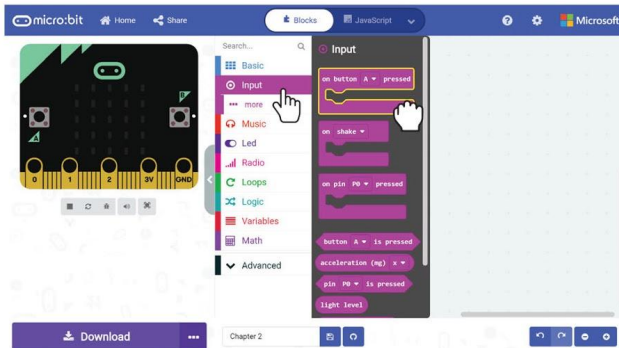
Jom Bermain Batu, Gunting, Kertas!

Butang micro:bit dan Button Bit

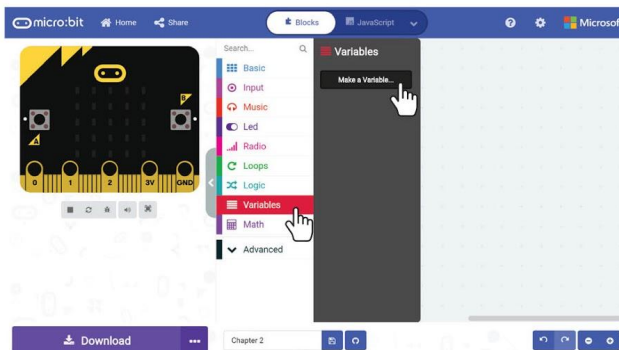


JOM KITA KOD!

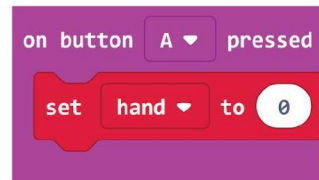
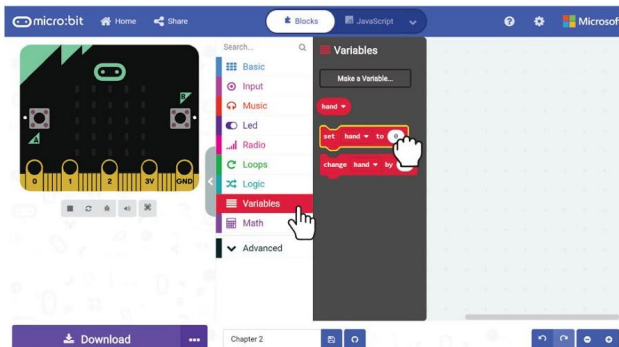
Langkah 1 Layari alamat tapak web <https://makecode.microbit.org> (atau tekan ikon 'Home' jika anda telah berada pada *MakeCode Editor*) dan mulakan projek baharu. Klik kategori [**Input**] and pilih blok [**on button _ pressed**].



Langkah 2 Klik kategori [**Variables**] dan kemudian pilih [**Make a Variable**]. Taipkan 'hand' pada tetitangkap *pop-up* dan tekan OK.

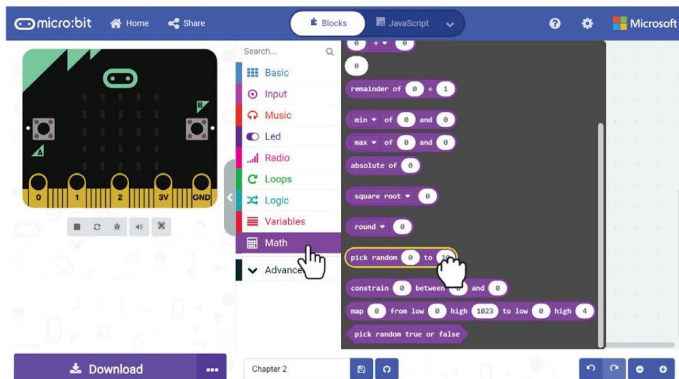


Langkah 3 Klik kategori [**Variables**] dan kemudian pilih blok [**set _ to _**]. Masukkan blok tersebut ke dalam slot [**on button A pressed**].

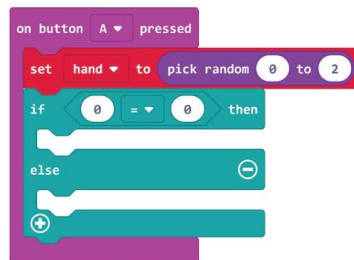
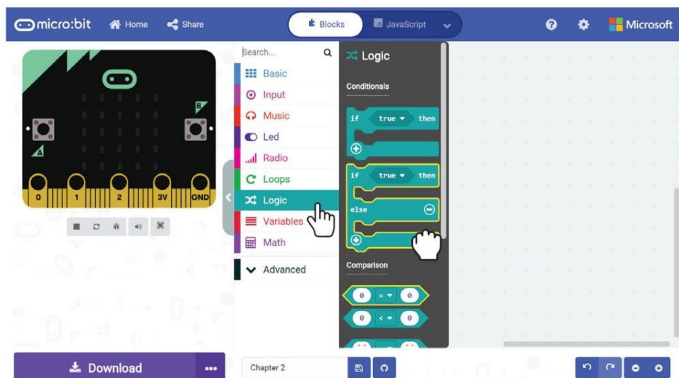




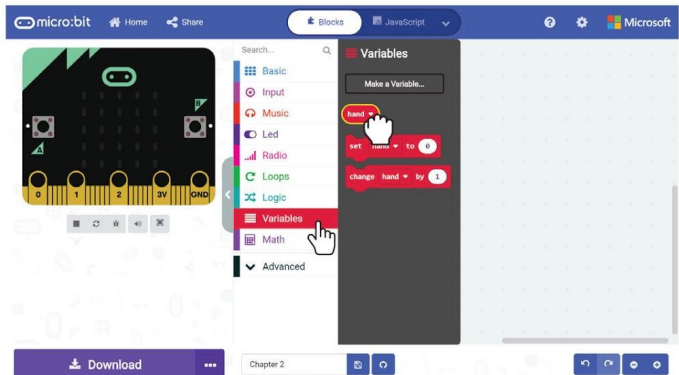
Langkah 4 Klik kategori [**Math**] dan kemudian pilih blok [**pick random _ to _**]. Tukarkan nombor 10 kepada 2.



Langkah 5 Klik kategori [**Logic**] dan kemudian pilih blok [**if - then - else**] dan blok perbandingan [**=**]. Masukkan blok perbandingan tersebut di dalam slot *if*.

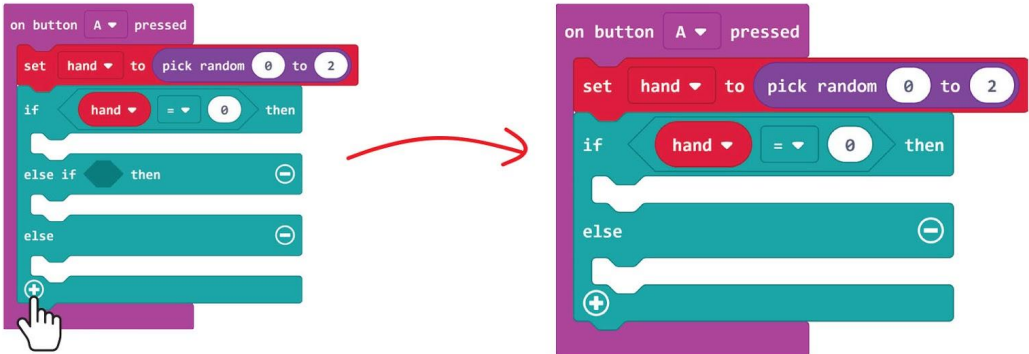


Langkah 6 Klik kategori [**Variables**] dan kemudian pilih blok pembolehkan [**hand**]. Masukkan blok tersebut pada slot pertama blok perbandingan.

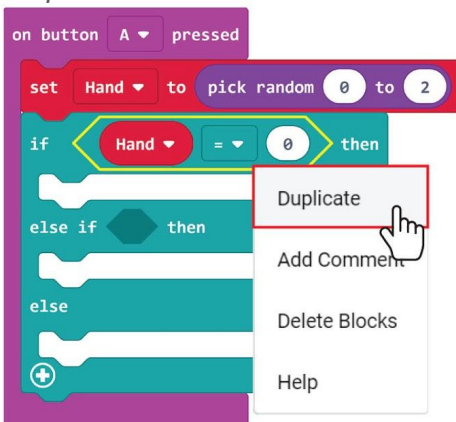


Bab 2 : Jom Bermain Batu, Gunting, Kertas !

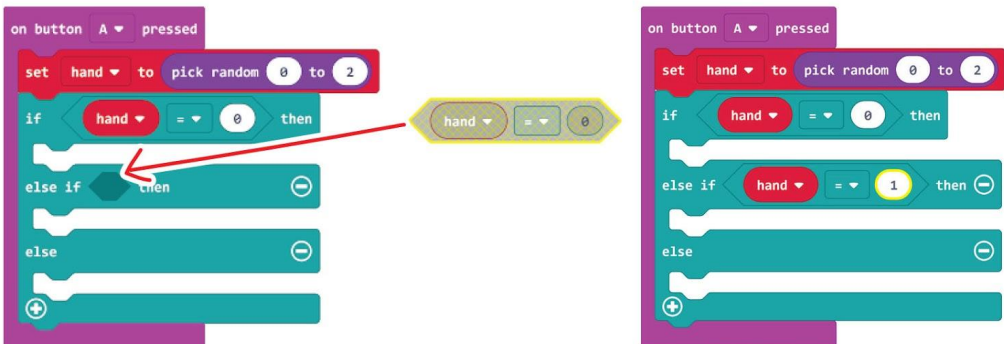
Langkah 7 Klik ikon [+] untuk menambah kondisi [if else] pada blok 'if'.



Langkah 8 Tekan butang tetikus kanan pada blok perbandingan dan pilih *Duplicate*.



Langkah 9 Masukkan blok yang telah disalin tadi kepada slot 'else-if' dan kemudiannya tukarkan nombor 0 kepada 1.





Langkah 10 Tambahkan blok [**Basic**] : [**show leds**] kepada slot 'if', 'else if' dan 'else'. Klik kotak pada [**show leds**] untuk mereka corak seperti rajah di bawah.

Jom Kita
Cuba!



```
on button A pressed
set hand to pick random 0 to 2
if hand = 0 then
  show leds
else if hand = 1 then
  show leds
else
  show leds
```

Muat turun kod anda kepada EDU:BIT dan sekarang anda boleh bermain permainan klasik "Rock, Paper, Scissors" dengan rakan-rakan yang lain. Apabila anda tekan butang A, pada micro:bit atau butang kuning, matriks LED akan memaparkan ikon "Rock", "Paper" atau "Scissors" secara rawak.

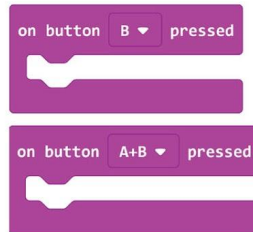


Peringatan: Jika anda ingin menyimpan projek ini, pastikan simpan pada folder di dalam komputer anda dengan menekan butang save.

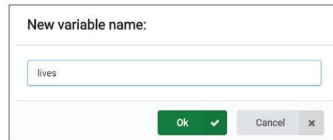
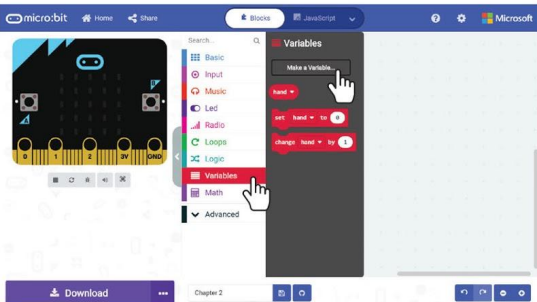


Jom kita ubah suai projek ini dengan memberikan pemain 3 nyawa permainan. Bagi membolehkan situasi tersebut, anda perlu buat satu lagi pemboleh ubah iaitu *Lives* dan kemudiannya menambah blok-blok seperti rajah di bawah.

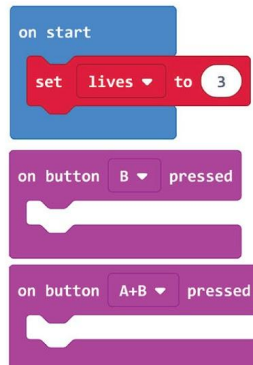
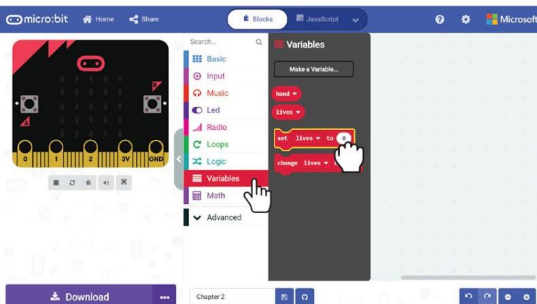
Langkah 11 Klik kategori [**Input**] dan pilih blok [**on button _ pressed**]. Salinkan blok tersebut dan tukarkan tetapan kepada 'button B' dan 'button A+B' pada kedua-dua blok tersebut.



Langkah 12 Klik kategori [**Variables**] dan kemudian pilih [**Make a Variable**]. Taipkan *Lives* pada tettingkap *pop-up* dan tekan OK.

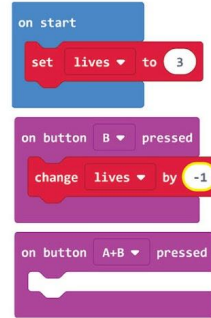
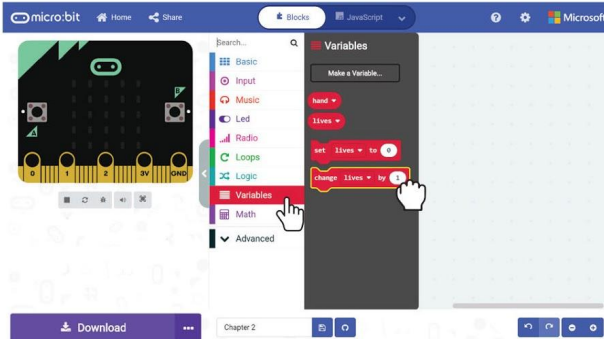


Langkah 13 Klik kategori [**Variables**] dan kemudian pilih blok [**set _ to _**]. Masukkan blok tersebut pada slot [**Basic**] : [**on start**]. Setkan pemboleh ubah kepada *Lives* dan tukar nilai kepada 3

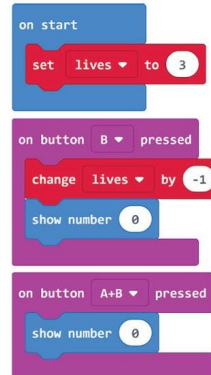
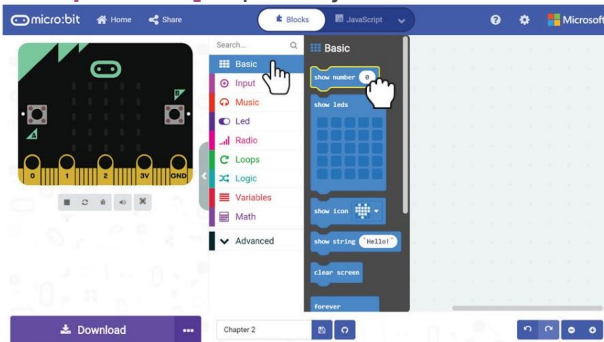




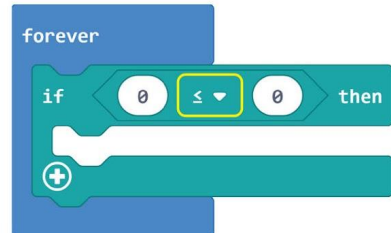
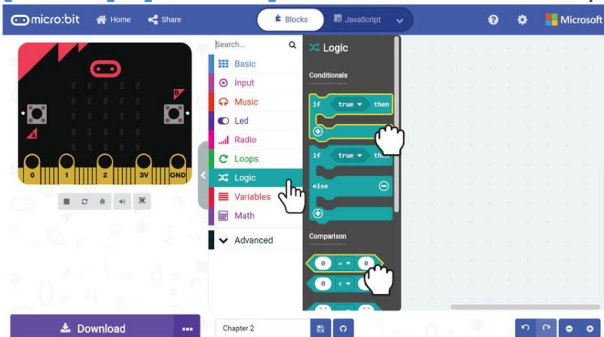
Langkah 14 Klik kategori [**Variables**] dan pilih blok [**change _ by _**]. Masukkan blok tersebut ke dalam slot [**on button B pressed**]. Setkan pemboleh ubah kepada *Lives* dan tukarkan kepada nilai -1



Langkah 15 Klik kategori [**Basic**] dan pilih blok [**show number**]. Salin blok tersebut dan masukkan ke dalam slot [**on button B pressed**] dan [**on button A+B pressed**] seperti rajah.

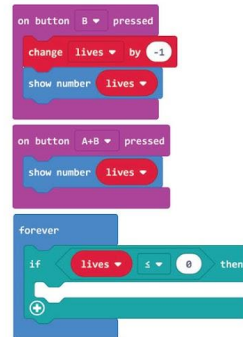
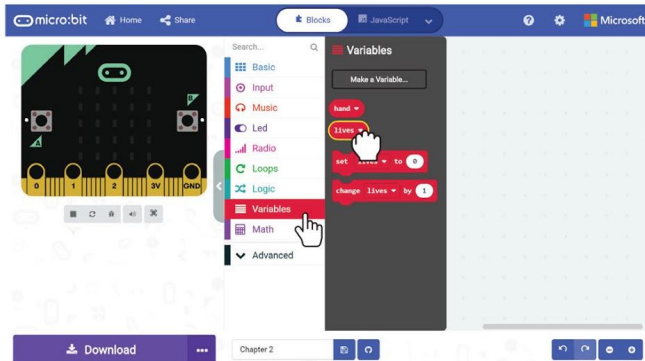


Langkah 16 Klik kategori [**Logic**] dan kemudian pilih blok [**if - then**] dan blok perbandingan [**_ = _**]. Masukkan blok perbandingan tersebut di dalam slot [**Basic**] : [**forever**] dan tukarkan simbol kepada ' \leq '

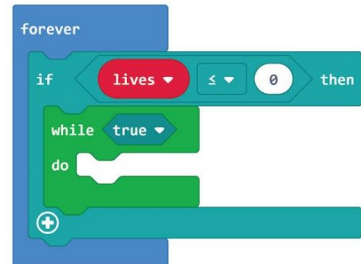
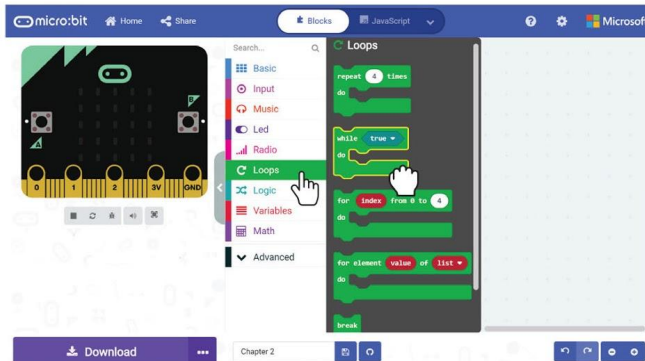


Bab 2 : Jom Bermain Batu, Gunting, Kertas !

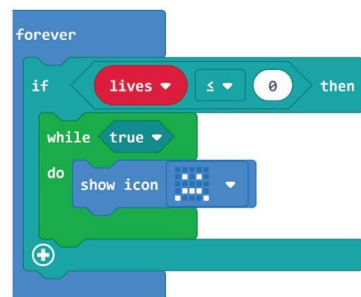
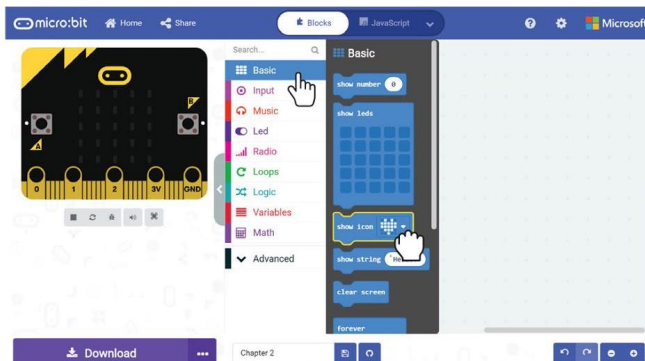
Langkah 17 Klik kategori kategori [**Variables**] dan pilih blok [**lives**]. Salin blok tersebut dan masukkan ke dalam kedua-dua slot [**show number**] dan pada slot kiri pada blok perbandingan [**_ ≤ _**].



Langkah 18 Klik kategori [**Loops**] dan pilih blok [**while _ do**]. Masukkan ke dalam slot bagi blok [**if-then**].



Langkah 19 Klik kategori [**Basic**] dan pilih blok [**show icon**]. Masukkan blok tersebut ke dalam slot bagi blok [**while _ do**] dan tukarkan icon kepada *sad face*.





Langkah 20 Berikut adalah kod projek yang lengkap. *Flash* kod anda kepada EDU:BIT dan bermain bersama rakan-rakan agar dapat menentukan siapa Raja atau Pemaisuri permainan Batu, Gunting, Kertas!

```
on button A pressed
  set hand to pick random 0 to 2
  if hand = 0 then
    show leds
  else if hand = 1 then
    show leds
  else
    show leds
  +

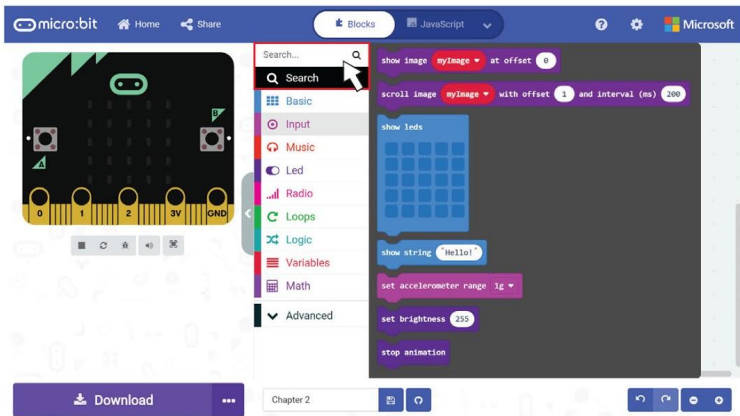
on start
  set lives to 3

on button B pressed
  change lives by -1
  show number lives

on button A+B pressed
  show number lives

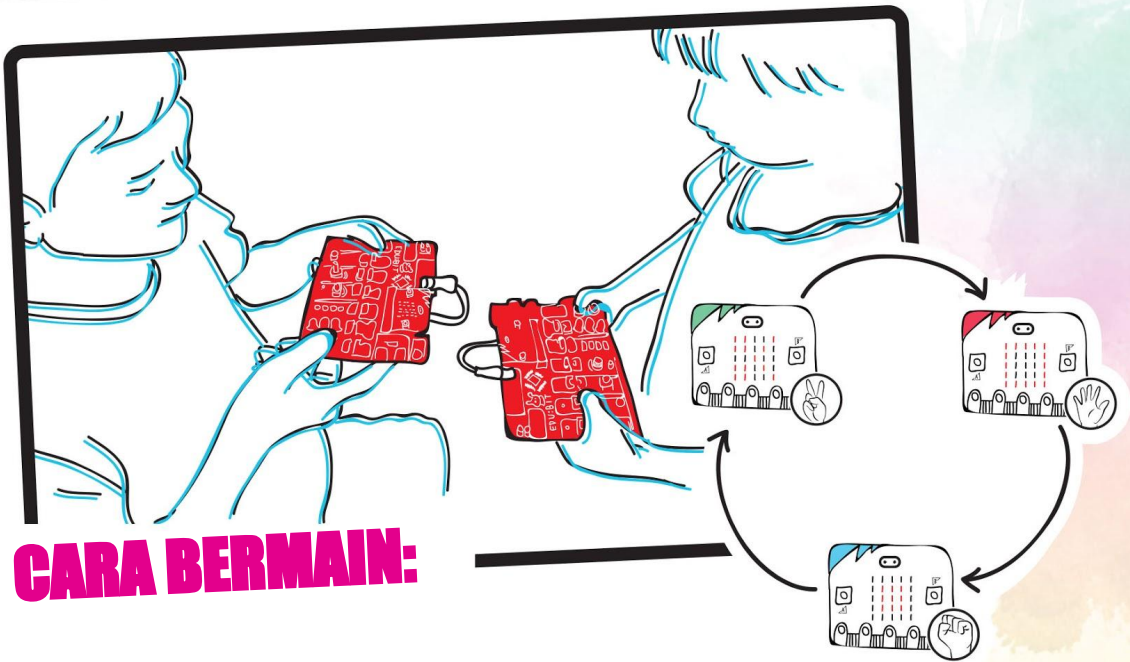
forever
  if lives <= 0 then
    while true
      do show icon
    +
```

Ini adalah tip yang amat berguna - Kesemua blok pengekod dilabelkan mengikut warna. Cari blok yang diperlukan pada kategori yang mempunyai warna yang sama. Secara alternatifnya, anda juga boleh taip pada *search box* untuk mencari blok yang diperlukan.



Jom Kita Main

'Batu, Gunting, Kertas!' - Edisi yang telah diubahsuai



CARA BERMAIN:

- Berdiri berhadapan lawan anda. Apabila kedua-dua pemain bersedia, tekan butang kuning (*Button A*) untuk memaparkan ikon Batu, Gunting dan Kertas.
- Banding dan tentukan siapa pemenang
- Pemain yang kalah perlu menekan butang biru (*button B*) pada EDU:BIT sekali untuk menolak satu nyawa permainan.
- Tekan kedua-dua butang kuning dan biru (*Button A+B*) serentak untuk memaparkan bilangan nyawa permainan yang masih tinggal.
- Jika pemain kalah 3 kali, permainan akan tamat dan EDU:BIT pemain tersebut akan memaparkan ikon *sad face*.

Perhatian

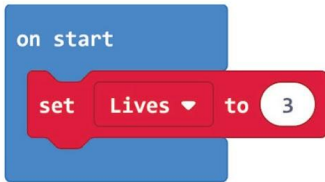
- Bagi memulakan permainan semula, anda perlu menekan butang RESET.
- Jika anda tidak mempunyai rakan yang berkelapangan untuk bermain, anda juga boleh berlawan dengan simulator micro:bit pada *MakeCode Editor*.

TAHUKAH

ANDA??

Di dalam pengaturcaraan kod komputer, kita menggunakan pemboleh ubah untuk menyimpan informasi atau nilai yang berubah semasa program dilaksanakan. Sebagai perbandingan, anda boleh fikirkan bahawa pemboleh ubah adalah sampul surat berlabel yang mengandungi kertas yang mempunyai informasi tertulis padanya. Ketas tersebut boleh dikeluarkan dan diganti dengan kertas yang mempunyai maklumat terkini.

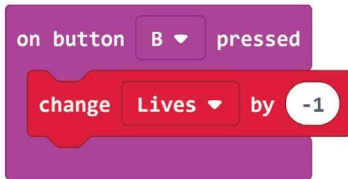
Pada pengaturcaraan program kita tadi, pemboleh ubah *Lives* yang telah dibuat disetkan pada nilai 3 sebagai permulaan.



Nombor 3 ditulis pada kertas adalah informasi permulaan.

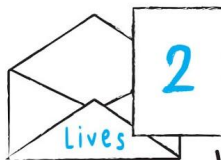
Sampul yang dilabelkan lives adalah pemboleh ubah.

Apabila butang B ditekan, nilai pembolehubah dikurangkan dengan nilai -1.

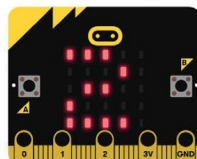


Apabila butang B ditekan, kertas yang ditulis nilai '3' dikeluarkan dan kemudiannya diganti dengan kertas bernilai '2' (contohnya: $3 - 1 = 2$)

Apabila kedua-dua butang A+B ditekan serentak, matriks LED akan memaparkan nilai pada pemboleh ubah *Lives*.



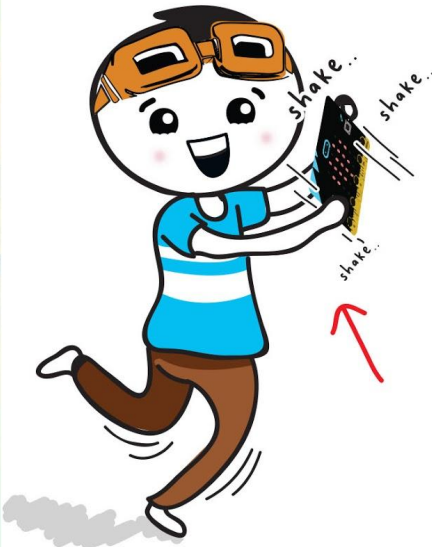
Setelah butang A+B ditekan serentak, kertas di dalam sampul dikeluarkan dan nilai pada kertas tersebut dibaca.





Terokai Blok-Blok Lain

Selain blok [on button _ pressed], anda juga boleh menggunakan blok-blok lain pada kategori [Input] untuk atur cara tetapan peristiwa (*event based*). Tindakan yang dilakukan anda seperti menekan butang, menggoncang micro:bit dinamakan peristiwa. Kod aturcara di bawah akan menghasilkan nyalaan Matriks LED selama 1 saat apabila micro:bit digoncang. Jom kita cuba.



Keputusan

Jika anda klik pada button [shake] pada blok, tettingkap pop-up akan menunjukkan senarai picu tindakan yang boleh digunakan. Cuba programkan EDU:BIT anda untuk menunjukkan ikon yang berlainan untuk setiap picu tindakan. Selamat mencuba!

| | | | |
|-------------|-----------|------------|-----------|
| shake | logo up | logo down | screen up |
| screen down | tilt left | tilt right | free fall |
| 3g | 6g | 8g | |

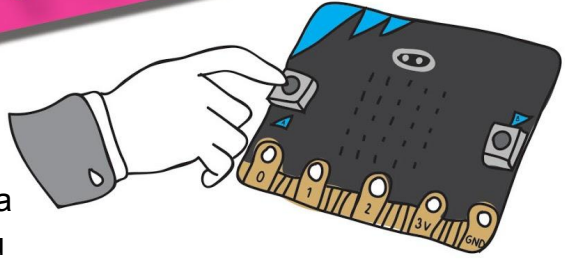
EDU:BIT boleh mengesan goncangan dan mengenal pasti orientasi papan kerana ia mempunyai sensor pengesan pergerakan.



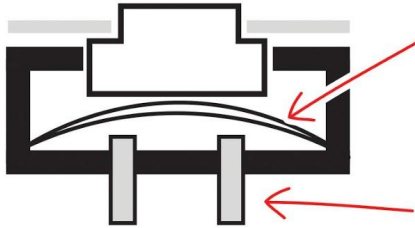
FAKTA MENARIK!



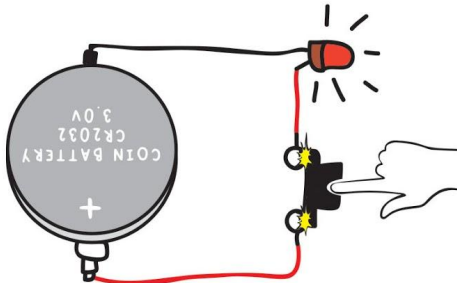
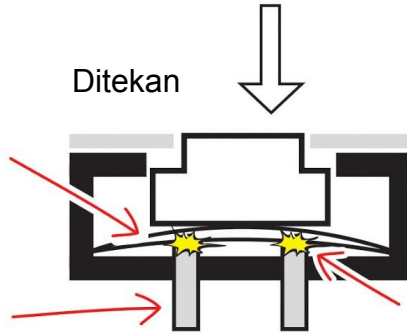
Push button adalah butang input atau suis dengan hanya dua keadaan iaitu tekan atau tidak ditekan



Tidak Ditekan



Ditekan



Apabila *Push Button* ditekan, litar elektrik akan menjadi litar lengkap dan LED akan menyala! Cuba teka apa akan berlaku apabila *Push Button* dilepaskan?

HELP

PANIC
BUTTON



PRESS BUTTON
WHEN YOUR SAFETY
IS THREATENED



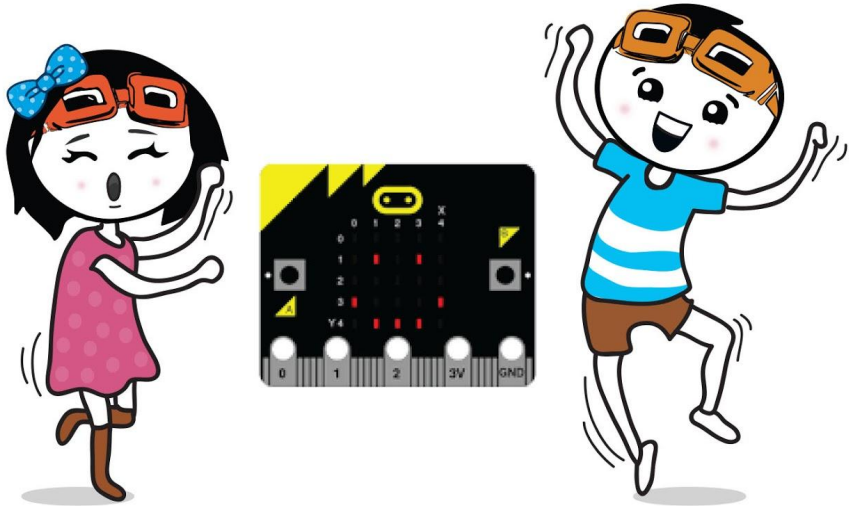
Butang Hitam, Kelabu, Hijau dan Putih kebiasaannya digunakan untuk fungsi ON/OFF dan Butang Merah digunakan untuk Butang Panik atau kecemasan untuk mesin.

Belajar lagi!



youtu.be/t_Qujjd_38o

CABARAN APLIKASI



Bina program EDU:Bit yang berfungsi untuk merekodkan kehadiran pelajar. Pelajar perempuan akan menekan butang A semasa memasuki kelas, manakala pelajar Lelaki akan menekan butan B.

| | |
|---|--|
| <i>On Start</i> | Papar Wajah Senyum. Set pemboleh ubah Perempuan = 0 dan Lelaki = 0 |
| <i>On button A Pressed</i> (Butang Kuning) | Tukar pemboleh ubah Perempuan kepada tambah 1 |
| <i>On button B Pressed</i> (Butang Biru) | Tukar pemboleh ubah Lelaki kepada tambah 1 |
| <i>On button A+B Pressed</i> | Paparkan maklumat pada Paparan LED: Jumlah = (Perempuan + Lelaki); Perempuan = (Perempuan); Lelaki = (Lelaki) |

Jom Bermain Muzik ~

Music Bit (Pembaz Piezo + Jek Audio)

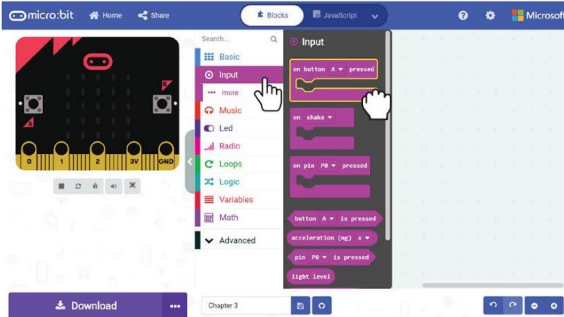


Imbas Saya!

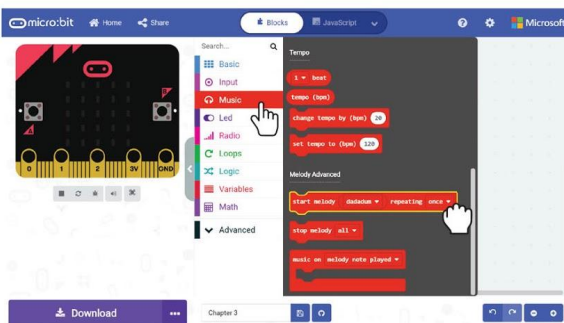


link.cytron.io/edubit-chapter-3

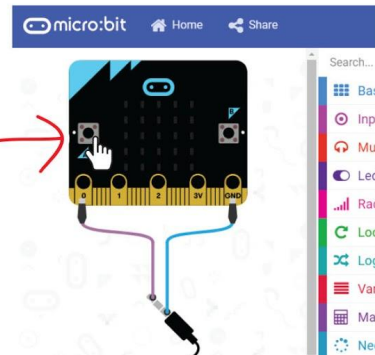
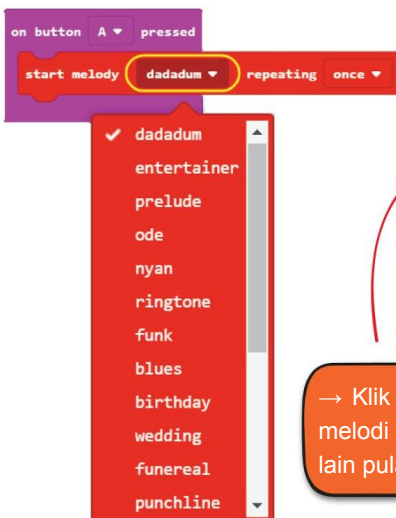
Langkah 1 Layari alamat tapak web <https://makecode.microbit.org> (atau tekan ikon 'Home' jika anda telah berada pada MakeCode Editor) dan mulakan projek baharu. Klik kategori [**Input**] and pilih blok [**on button _ pressed**].



Langkah 2 Klik kategori [**Music**] dan kemudian pilih blok [**start_melody_repeating_**]



Langkah 3 Klik [**dadadum**] dan kemudian pilih melodi 'birthday' daripada pilihan.



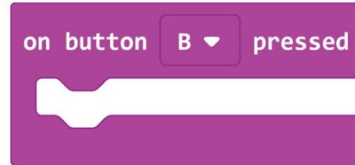
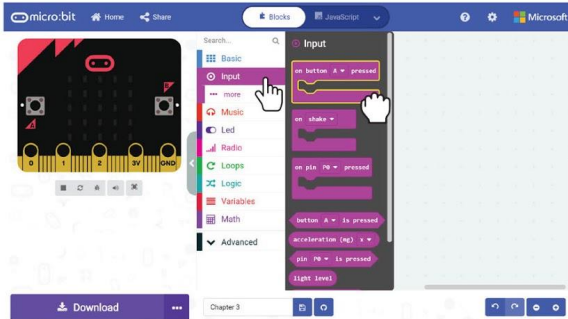
→ Klik pada butang A pada paparan simulasi. Adakah anda dengar melodi yang biasa anda dengar? Mari kita cuba melodi-melodi yang lain pula. * Pastikan pembesar suara komputer anda dihidupkan.



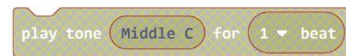
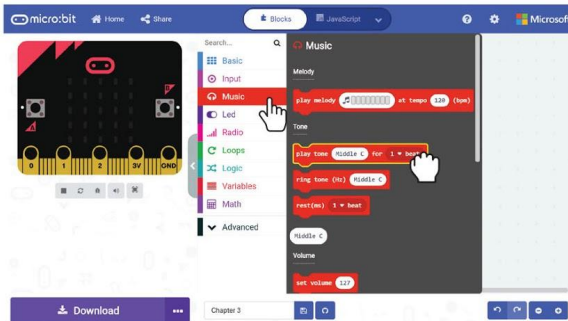
Selain daripada senarai melodi yang telah tersedia, anda boleh mengatur cara EDU:BIT untuk memainkan sebarang lagu yang anda sukai. Jom kita cuba~



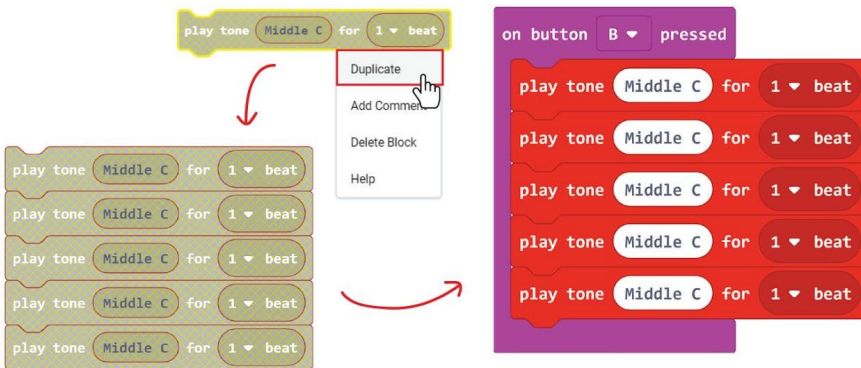
Langkah 4 Klik kategori [**Input**] dan pilih blok [**on_button_pressed**]. Masukkan blok tersebut ke dalam slot [**on button B pressed**].



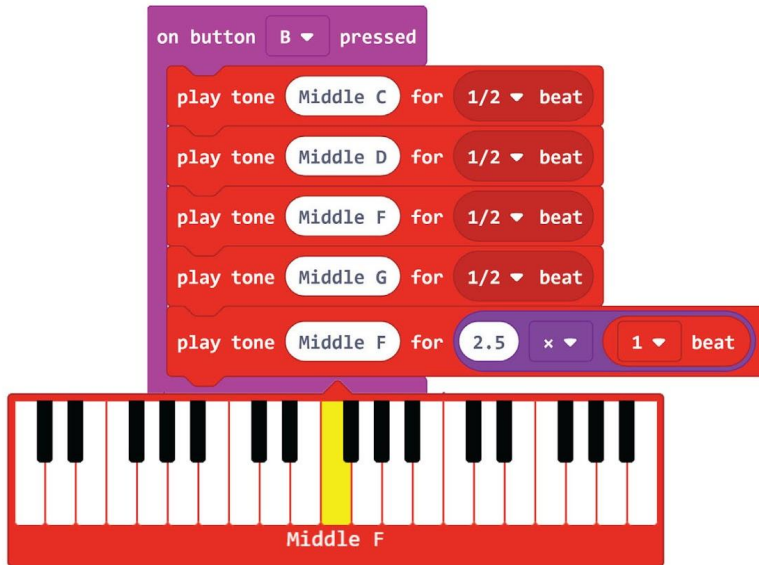
Langkah 5 Klik kategori [**Music**] dan pilih blok [**playtone_for_beat**].



Langkah 6 Pada ruang kerja, klik-kanan pada blok [**playtone_for_beat**] dan kemudian klik "Duplicate". Ulang sehingga anda mempunyai lima blok [**playtone_for_beat**]. Masukkan blok tersebut ke dalam slot [**on_button A pressed**].

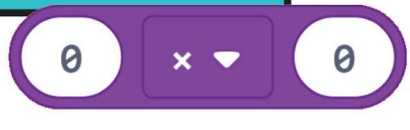


Langkah 7 Pilih 'tone'; dan 'beat' di dalam blok [**play tone_for_beat**] seperti contoh kod di bawah:



Nota!

Blok ungu didapati dari kategori [Math].



Klik pada butang B pada paparan simulasi anda. Cuba teka lagu yang dimainkan?



Apabila anda membuat pengaturcaraan, anda bolehlah menyemak kod selalu untuk memastikan ianya mengikut perancangan yang anda tentukan. Anda bolehlah menggunakan simulasi untuk tujuan tersebut.



Langkah 8 Sambung mengaturcara lagu yang penuh dengan menambah lagi blok [**Play tone_for_beat**] dan ubah 'tone' dan 'beat' yang sepatutnya. Anda boleh rujuk pada 'Tones' dan 'Beat' pada mukasurat seterusnya.



I Will Follow You

on button B ▾ pressed

I will follow you,

Follow you wherever you may go,

There is n't an ocean too deep

play tone Middle C for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle G for 1/2 ▾ beat
 play tone Middle F for 2.5 x ▾ 1 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone Middle C for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle D for 1/2 ▾ beat
 play tone Middle F for 1/2 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High C for 1.5 x ▾ 1 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone Middle A for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat

A mountain so high it can keep, Keep me a-way ...

play tone High C for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High D for 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone Middle B for 1/2 ▾ beat
 play tone High C for 2 ▾ beat
 rest(ms) 1/2 ▾ beat
 play tone High C for 1/2 ▾ beat
 play tone Middle A for 1 ▾ beat
 play tone Middle G for 1 ▾ beat
 play tone Middle A for 1 ▾ beat
 play tone Middle G for 1/2 ▾ beat
 play tone Middle F for 2.5 x ▾ 1 ▾ beat
 rest(ms) 1 ▾ beat



Langkah 9 Muat naik kod yang lengkap kepada EDU:BIT anda

EDU:BIT akan memainkan lagu "I Will Follow You" setiap kali ada menekan butang biru (Butang B) pada EDU:BIT anda.

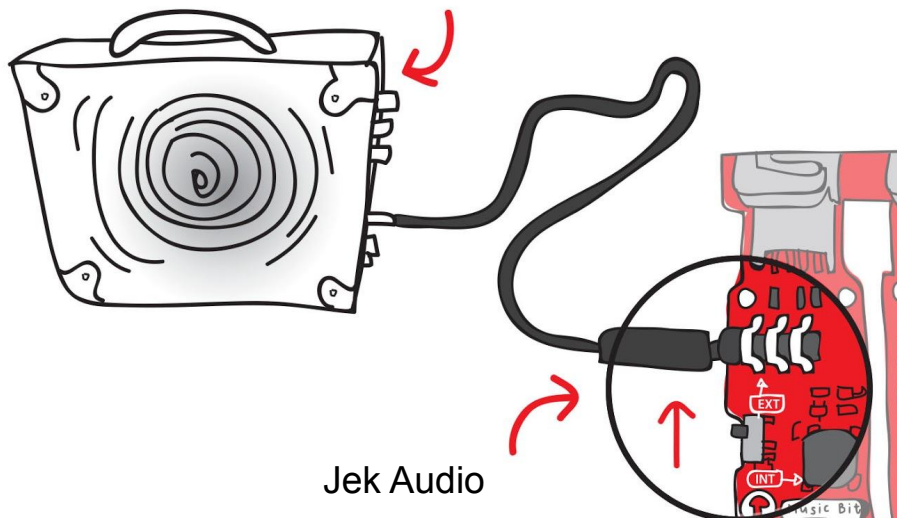


Buzzer Piezo



Selain itu, anda boleh juga menyambungkan pembesar suara luaran ataupun headphones pada jek audio pada EDU:BIT. Anda perlu ubat suis pada EXT (external)

Pembesar Suara

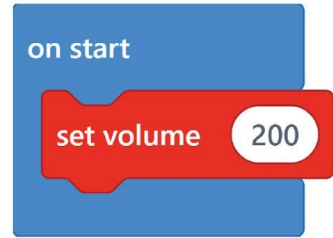
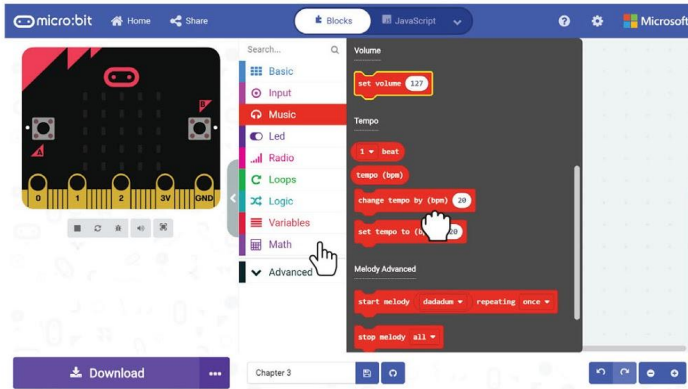


Jek Audio



Adakah muzik tersebut terlalu perlahan? Atau terlalu kuat? Anda boleh tambah blok [set volume_] untuk mengubah volume, dari 0 hingga 255(maksimum)

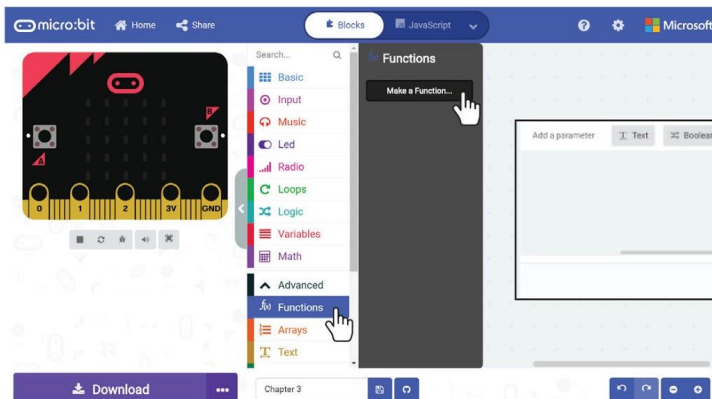
Langkah 10 Klik kategori [**Music**] dan Pliih blok [**set volume_**]. Letakkan blok tersebut ke dalam blok [**on start**] dan ubah nilai volume kepada 200.



Nota!

Anda boleh menyusun blok-blok kod yang menjalankan tugas tertentu ke dalam 'Function', sebagai contoh kod untuk memainkan lagu 'I Will Follow You'. Di dalam pengaturcaraan, 'Function' merujuk kepada rutin atau set prosedur. Setelah 'Function' ditentukan, ia boleh digunakan di mana-mana bahagian pada pengaturcaraan anda tanpa perlu menulis semula blok kod yang sama berulang kali.

Langkah 11 Klik kategori [**Advanced**] dan pilih kategori [**Functions**]. Klik [**Make a Function**] dan namakan semula 'doSomething' kepada '*I Will Follow You*' pada paparan tingkap. Kemudian klik '*Done*'.

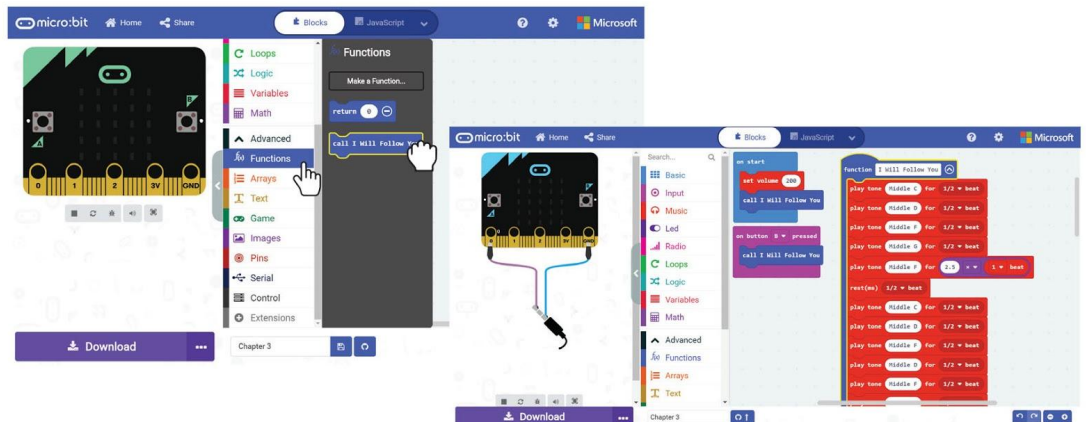


Bab 3 : Jom Bermain Muzik~

Langkah 12 Satu blok [**function I Will Follow You**] akan muncul pada Editor anda. Klik blok yang paling atas dalam blok [**on Button B pressed**] dan tarik semua blok-blok ke dalam slot blok [**function I Will Follow You**].



Langkah 13 Klik kategori [**Function**] dan pilih blok [**call I Will Follow You**]. 'Duplicate' blok. Pasang blok [**call I Will Follow You**] kepada blok [**on start**] dan blok [**on button B pressed**]. Berikut adalah contoh kod:



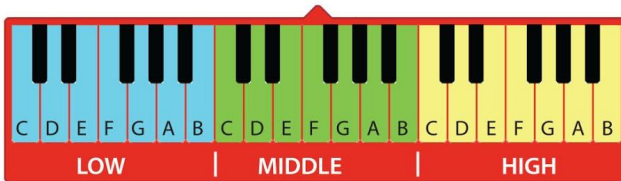
Langkah 14 Pindahkan kod kepada EDU:BIT. Nikmati rentak muzik anda~

TAHUKAH

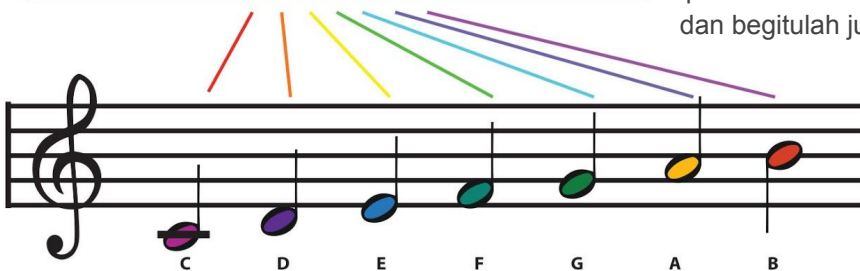
ANDA??



Anda boleh memprogram EDU:BIT untuk memainkan lagu-lagu lain sekiranya anda tahu cara membaca muzik. Ini adalah panduan ringkas untuk membantu anda mengkod susunan muzik.



Kedudukan not muzik pada lima garis lintang menunjukkan ton mana yang hendak dimainkan, Semakin tinggi not itu berada, semakin tinggi 'pitch' atau frekuensi bunyi tersebut dan begitulah juga sebaliknya.



| Sign | Rest | Relative Length | Duration |
|------|------|-----------------|----------|
| | | Whole Note | 4 beats |
| | | Half Note | 2 beats |
| | | Quarter Note | 1 beat |
| | | Eighth Note | 1/2 beat |
| | | Sixteenth Note | 1/4 beat |

Nota muzik yang berbeza digunakan untuk menunjukkan berapa lama sesuatu not akan dimainkan

TAHUKAH

ANDA??

Dengan petunjuk yang diberikan, bolehkah anda menyahkod irama berikut?



Baby Shark

$\text{♩} = 115$

Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark Doo Doo Doo Doo Doo Doo Ba -by Shark.

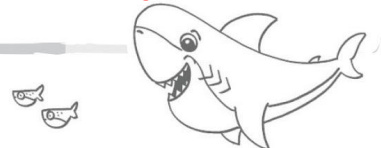
| | | | | | | | | | |
|--------|--------|--------|--------|-----|--------|--------|--------|-----|--------|
| Line 1 | Ba | -by | Shark | doo | doo, | doo | doo | doo | doo |
| Note | High D | High E | High G | | High G | High G | High G | | High G |
| Beat | 1 | | 1/2 | 1/2 | 1/2 | | 1/2 | 1/4 | 1/2 |

| | | | | | | | | | |
|--------|--------|-----|--------|--------|--------|-----|--------|--------|--------|
| Line 2 | Ba | -by | Shark | doo | doo, | doo | doo | doo | doo |
| Note | High D | | High G | High G | High G | | High G | High G | High G |
| Beat | 1/2 | 1/2 | 1/2 | | 1/2 | 1/4 | 1/2 | | 1/2 |

Repeat the same for Line 3

| | | | |
|--------|-----|--------|---------|
| Line 4 | Ba | -by | Shark |
| Note | | High G | High F# |
| Beat | 1/2 | 1/2 | |

Pengaturcaraan EDU:BIT untuk memainkan irama Baby Shark apabila butang kuning (butang A) dan butang biru (butang B) ditekan serentak



Nota!

Guna blok [set volume_] untuk mengubah kekuatan bunyi nada tersebut.



Terokai Blok-Blok Lain

#1 Anda boleh set *tempo* (i.e kelajuan rentak lagu) dengan menggunakan blok [set tempo to (bpm)_]. Semakin tinggi *bpm*(beats per minute), semakin laju rentak lagu dan akan menjadi semakin rancak. Guna blok [change tempo by (bpm)_] untuk mengubah tempo.

#2 Guna blok [stop melody _] menghentikan melodi yang sedang dimainkan.

#3 Anda juga boleh menggunakan [music on_] dan pilihan, seperti *melody started* dan *melody ended*, sebagai *event triggers* di dalam kod anda.

Berikut ialah contoh kod:

```
1 on start
  set tempo to (bpm) 120

2 music on melody started
  show icon [musical note]

3 music on melody ended
  clear screen

4 on button A pressed
  change tempo by (bpm) 50

5 on button B pressed
  start melody [entertainer] repeating forever

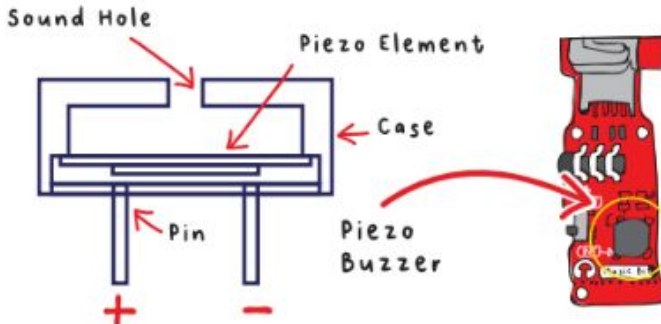
6 on button A+B pressed
  stop melody all
```

- 1 Di dalam atur cara ini, tempo permulaan diset kepada 120bpm.
- 2 Pada *melody started*, matriks LED akan memaparkan ikon not muzik.
- 3 Pada *melody ended*, semua matriks LED akan dipadamkan.
- 4 Setiap kali butang A ditekan, tempo akan bertambah sebanyak 50bpm.
- 5 Melodi 'entertainer' akan dimainkan apabila butang B ditekan.
- 6 Melodi akan berhenti apabila butang A dan B ditekan serentak.

FAKTA MENARIK!

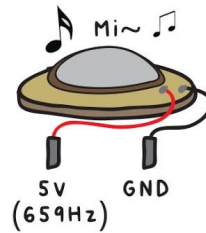
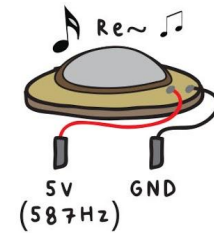
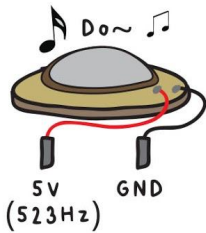


Pembaz Piezo selalunya digunakan untuk mengeluarkan bunyi dengan getaran elemen piezo apabila elektrik mengalir melalui *element* tersebut



Dengan perubahan frekuensi elektrik, kadar getaran akan turut berubah dan menyebabkan pembaz Piezo mengeluarkan bunyi yang berbeza-beza.

Elemen Piezo



Telinga manusia boleh mendengar frekuensi dalam julat 20Hz sehingga 20,000Hz. Bunyi di bawah 20Hz dikenali sebagai *infrasonic* manakala bunyi melebihi 20,000Hz dikenali sebagai *ultrasonic*.

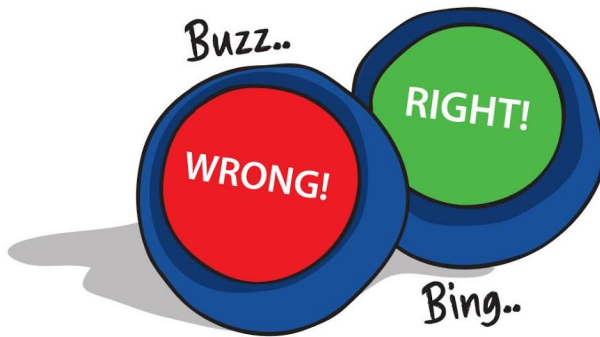


Belajar lagi!



youtu.be/cxfPNc4Wefo

CABARAN APLIKASI



Program EDU:BIT yang berfungsi sebagai *Game Show Buzzer* untuk jawapan betul dan salah.

| | |
|--|---|
| On Start | Paparkan <i>Smiley Face</i> |
| On Button A pressed (Butang Kuning) | Paparkan ikon 'betul' dan mainkan melodi <i>Power Up</i> sebanyak sekali. |
| On Button B pressed (Butang Biru) | Paparkan ikon 'pangkah' dan mainkan melodi 'wawawawaa' sebanyak sekali. |
| On Button A+B pressed | Padamkan paparan |

Win, Lose or Draw ~

Traffic Light Bit



Ayam belanda!



Imbas saya!



link.cytron.io/edubit-chapter-4

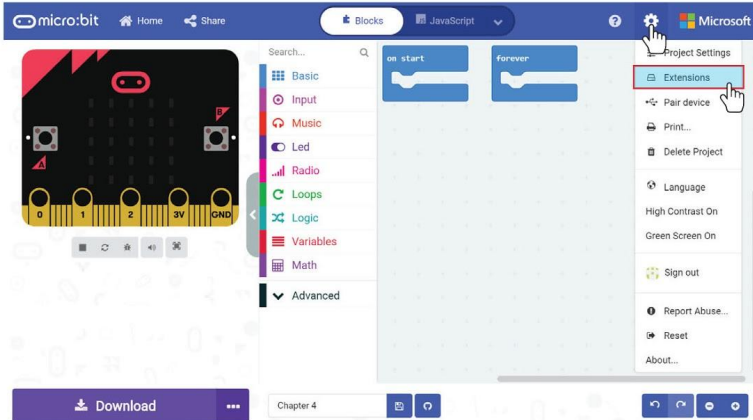


Adakah anda tahu bahawa terdapat satu set LED merah, kuning dan hijau pada EDU:BIT? Itu ialah *Traffic Light Bit*. Untuk mengatur cara, anda perlu tambah EDU:BIT *extention* pada *MakeCode Editor*. *Extensions* ialah satu set blok yang kita tambah ke dalam *editor* bagi membolehkan pengaturcaraan aksesori micro:bit seperti papan EDU:BIT.

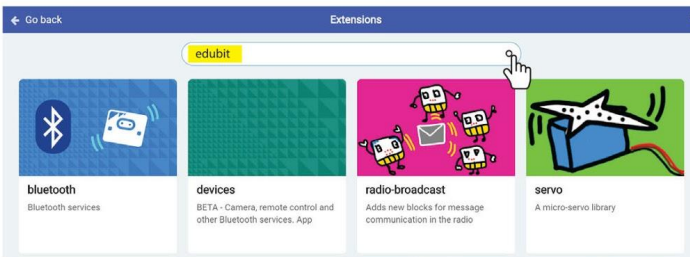


JOM KITA KOD!

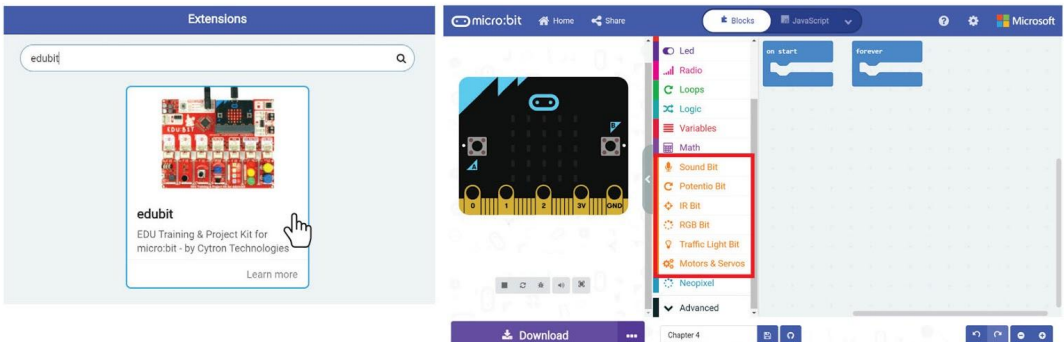
Langkah 1 Dalam *MakeCode Editor* anda, wujudkan projek baharu. Klik pada ikon roda gigi dan kemudian pilih 'Extensions'. Anda memerlukan sambungan internet untuk tambah *extensions*.



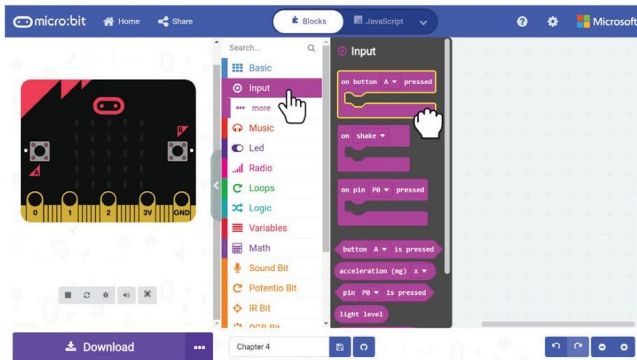
Langkah 2 Taip "edubit" dalam ruangan carian dan klik *Enter*.



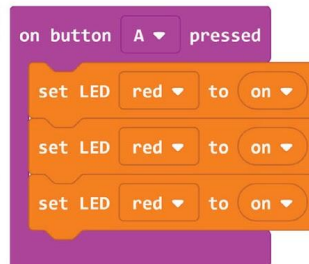
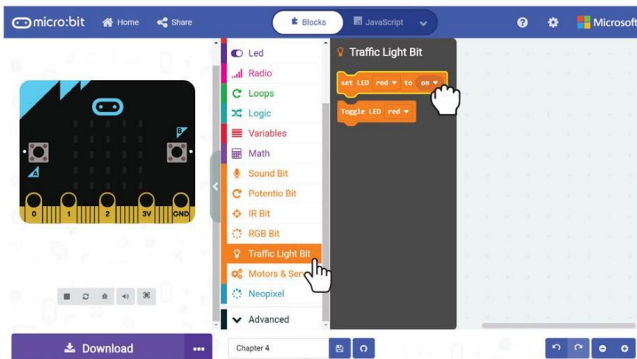
Langkah 3 Klik *extension* 'edubit'. Tunggu sehingga proses muat turun selesai. Anda akan nampak bahagian kategori baharu dalam *MakeCode Editor* anda.



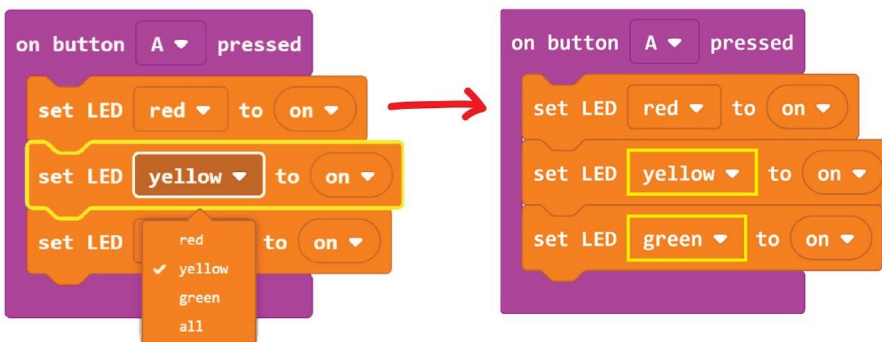
Langkah 4 Klik kategori [Input] dan kemudian pilih blok [on button_pressed].



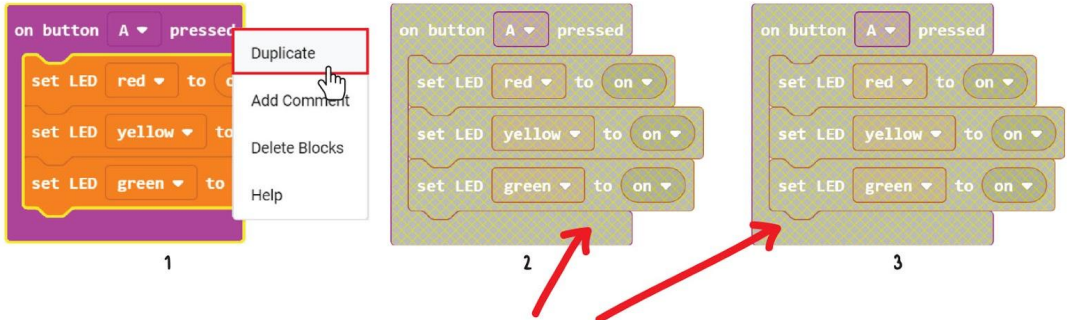
Langkah 5 Klik kategori [Traffic Light Bit] dan pilih blok [set LED_to_]. Pada blok [set LED_to_] di dalam ruang kerja, klik tombol kanan tetikus dan klik 'Duplicate'. Ulangi langkah ini sehingga anda ada tiga blok [set LED_to_]. Masukkan blok-blok tersebut ke dalam slot [on button A pressed].



Langkah 6 Klik pada pilih warna dan tukar blok kedua serta ketiga masing-masing kepada 'yellow' dan 'green'.



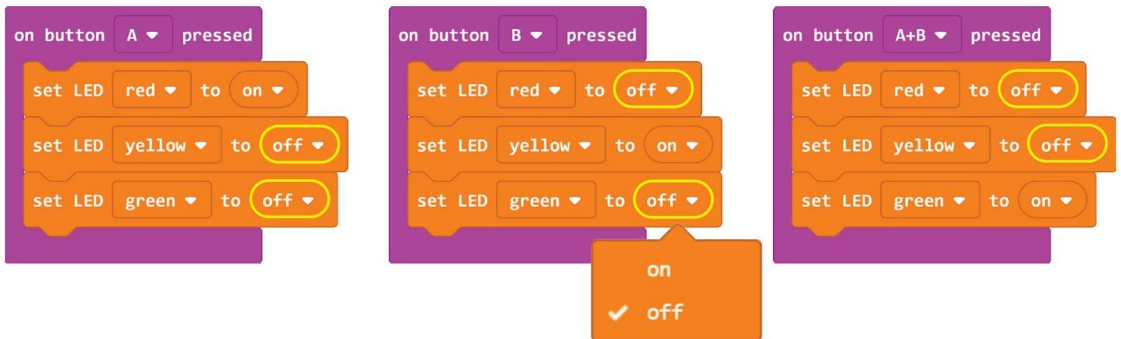
Langkah 7 Pada blok [on button_pressed], klik butang kanan tetikus dan pilih 'Duplicate'. Ulangi langkah ini sehingga dapat tiga set blok yang sama.



Blok-blok ini dinyahaktifkan (*disabled*) dan tidak dapat berfungsi kerana ini adalah blok [on button_pressed] ulangan.

Langkah 8 Tukar "A" pada blok [on button_pressed] yang kedua dan ketiga masing-masing kepada "B" dan "A+B".

Langkah 9 Tukar keadaan LED dari *on* kepada *off* seperti berikut.



Langkah 10 Pindahkan kod ke dalam EDU:BIT anda dan perhatikan apa yang berlaku apabila anda menekan butang A, butang B dan kemudian kedua-dua butang A+B ditekan pada masa yang sama.



Tunggu di luar!

Tunggu!

Masuk!

Woohoo...anda sudah boleh guna EDU:BIT sebagai petunjuk peribadi. Bolehkan anda fikirkan kegunaan yang lain?



MAKLUM BALAS LAMPU ISYARAT



Tolong, saya terkandas!



Masih mencuba!



Sudah dapat!



LED atau *light emitting diode* ialah satu contoh peranti output digital. Ia boleh berada dalam dua keadaan - *ON* atau *OFF*; di mana *ON* biasanya diwakili oleh 1 (satu) and *OFF* diwakili oleh 0 (sifar).



Anda juga boleh mengatur cara EDU:BIT anda supaya dapat berfungsi sebagai indikator penentu masa (*timer*). Ini contoh kodnya.



Penentu masa dicetus apabila anda menggoncang EDU:BIT

Mainkan nada sebagai isyarat bahawa penentu masa sudah mula

LED hijau menyala

LED kuning menyala

LED merah menyala

Mainkan melodi wawawawaa sebagai isyarat bahawa penentu masa sudah tamat

Toggle LED merah sebanyak 10 kali

```
on shake
  play tone Middle C for 1 beat
  set LED red to off
  set LED yellow to off
  set LED green to on
  pause (ms) 2000
  set LED red to off
  set LED yellow to on
  set LED green to off
  pause (ms) 2000
  set LED red to on
  set LED yellow to off
  set LED green to off
  pause (ms) 2000
  start melody wawawawaa repeating once
  repeat 10 times
  do
    Toggle LED red
    pause (ms) 500
```

Dalam contoh kod ini, setiap LED menyala selama 2000ms (2 saat).
Jika anda ingin menyalakan LED selama 1 minit, apakah nilai yang perlu anda letak?
Tip untuk anda:
1 minit = 60 saat

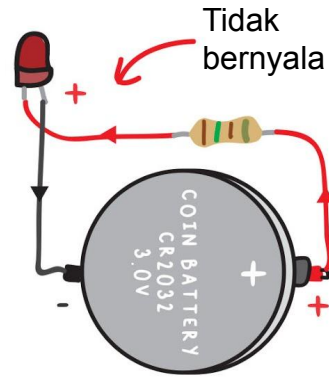
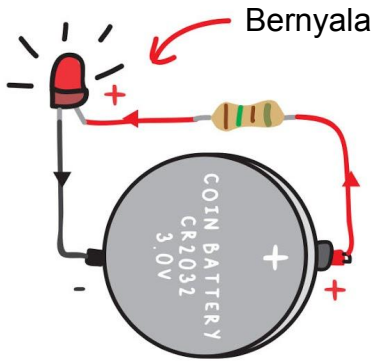


“Toggle” bermaksud bertukar dari satu keadaan kepada keadaan yang lain. Jika keadaan sekarang ialah ON, maka ia akan bertukar kepada OFF dan sebaliknya. Justeru itu, apabila kita toggle LED, maka LED akan kelihatan berkelip-kelip.

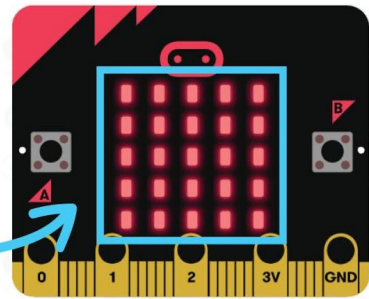
FAKTA MENARIK!



Light-emitting diode (LED) ialah sejenis peranti semikonduktor yang mengeluarkan cahaya daripada tenaga elektrik. Ia ada dua terminal, iaitu terminal positif dan negatif. Apabila LED disambungkan pada kutub yang betul, arus elektrik akan mengalir melaluinya dan menghasilkan cahaya.



LED yang digunakan dalam micro:bit adalah berasaskan *surface-mount technology* (SMT) dan boleh dalam bentuk yang sangat kecil.



Selain LED pada micro:bit, terdapat sebanyak 41 SMT LED pada papan EDU:BIT. Dapatkah anda mencari kesemua LED tersebut?

Belajar lagi!



youtu.be/qqBmvHD5bCw

CABARAN APLIKASI

| | |
|--|--|
| Programkan EDU:BIT supaya dapat berfungsi sebagai pembilang skor (<i>score counter</i>) dan juga penentu masa untuk permainan seperti “ <i>Win, Lose or Draw</i> ” dan “ <i>Charade</i> ”. | |
| <i>On start</i> | Set pemboleh ubah Pasukan A=0 Set pemboleh ubah Pasukan B=0 |
| <i>On Button A pressed</i> (Butang kuning) | Tukar skor Pasukan A sebanyak 1 Papar skor semasa Pasukan A |
| <i>On Button B pressed</i> (Butang biru) | Tukar skor Pasukan B sebanyak 1 Papar skor semasa Pasukan B |
| <i>On Button A+B pressed</i> (Butang kuning+biru) | Tatal (<i>scroll</i>) skor Pasukan A dan Pasukan B |
| <i>On shake</i> | Mulakan penentu masa selama 1 minit dengan nyalakan LED hijau (30 saat), kemudian LED kuning (20 saat) dan akhir sekali LED merah (10 saat). Mainkan melodi “wawawawaa” apabila penentu masa tamat. Toggle LED merah sebanyak 10 kali. |

Tip untuk anda. Anda perlu mewujudkan dua pemboleh ubah dan namakannya sebagai Pasukan A dan Pasukan B.



Jom Kita Main

Win, Lose or Draw~



CARA BERMAIN:

Bahagikan kelas kepada 2 pasukan - Pasukan A dan Pasukan B.

Seorang ahli dari Pasukan A akan mula memilih kad secara rawak. Selepas membaca perkataan pada kad secara senyap, goncangkan EDU:BIT untuk memulakan penentu masa (1 minit)

Dia boleh mula melukis gambar untuk diteka oleh ahli pasukannya. Tidak dibenarkan bercakap atau memberi isyarat.

1 markah akan diberi kepada Pasukan A (tekan *Button A* atau butang kuning) jika ahli pasukan dapat meneka perkataan / frasa dengan betul sebelum masa tamat.

Jika Pasukan A gagal, Pasukan B boleh cuba untuk "mencuri" markah dengan memberi tekaan yang tepat.

Kedua-dua pasukan akan melukis dan meneka secara bergilir-gilir sehingga permainan tamat.

Pasukan yang mendapat paling banyak markah adalah pemenang!

PERHATIAN!

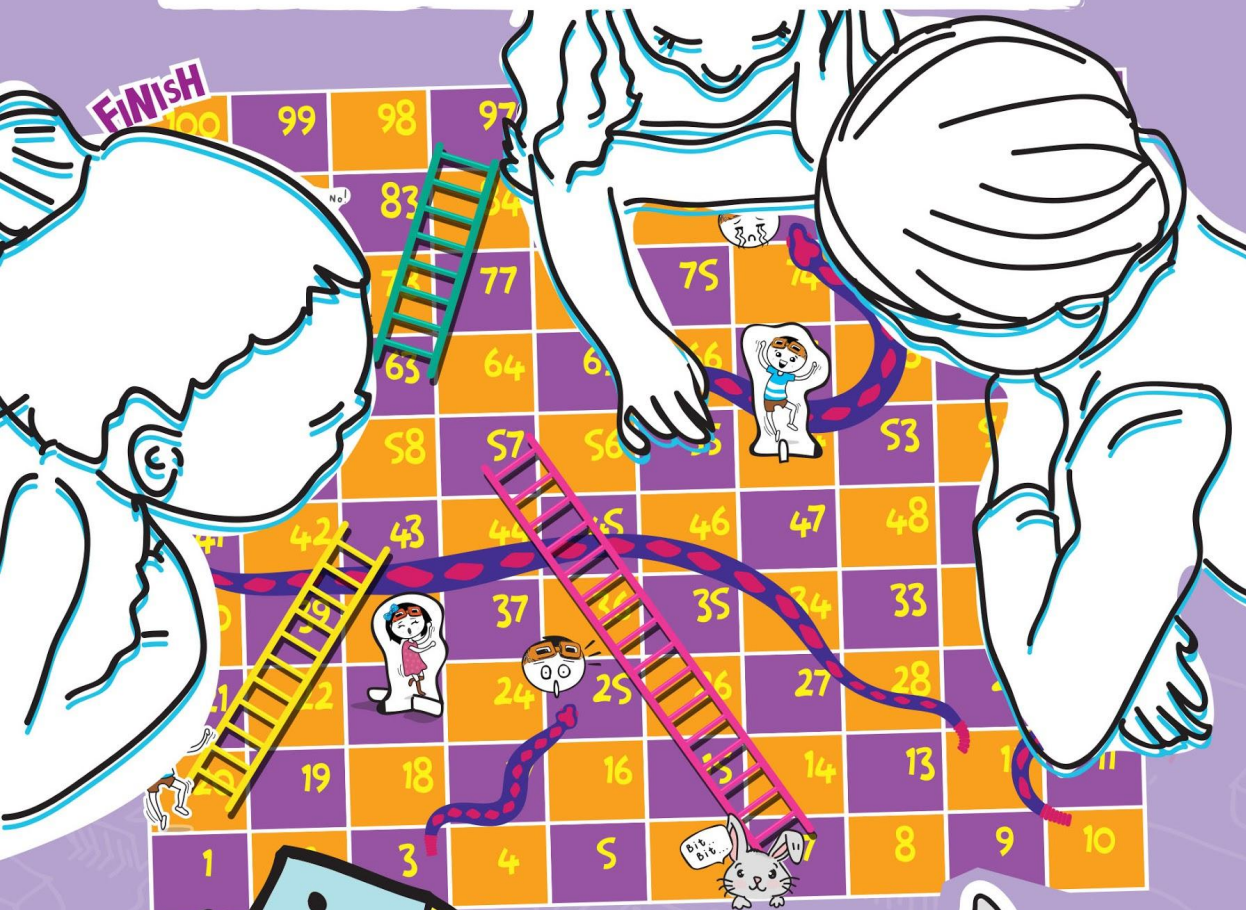
Imbas sini



Imbas sini untuk muat turun kad untuk dicetak bersama dengan cadangan perkataan. Jika anda tidak dapat melukis, anda boleh cuba *Charade*. Peraturan masih sama tetapi bukan melukis, anda menggunakan isyarat pergerakan sebagai petunjuk kepada ahli pasukan untuk diteka.

Jom Bermain Dadu Digital IR~

IR Bit

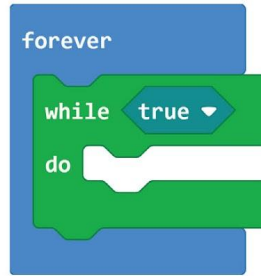


Imbas saya!

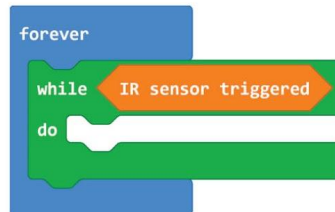
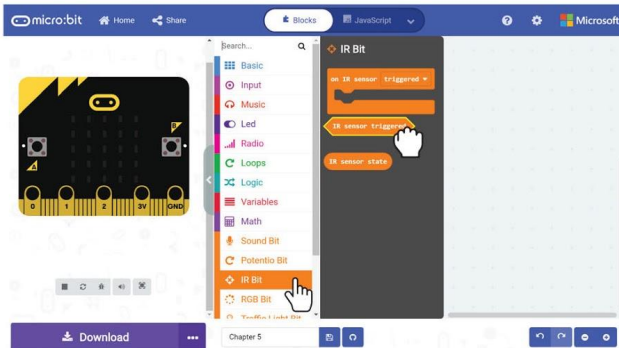


JOM KITA KOD!

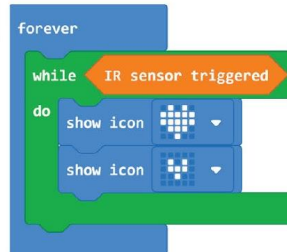
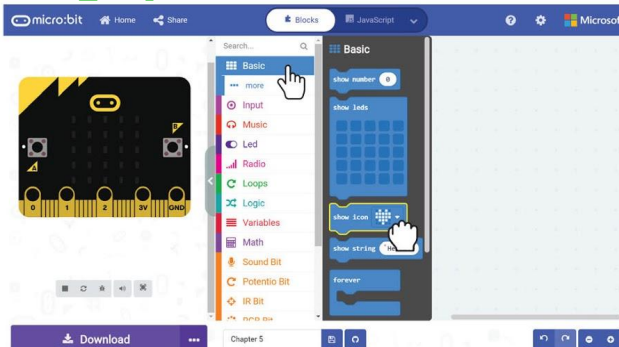
Langkah 1 Wujudkan projek baharu dalam *MakeCode Editor* anda dan tambah **EDU:BIT extension** (rujuk halaman 40). Klik kategori [**Loops**] dan pilih blok [**while_do**]. Masukkan blok ke dalam slot [**forever**].



Langkah 2 Klik kategori [**IR Bit**] dan pilih blok [**IR sensor triggered**]. Masukkan blok ke dalam slot bersyarat iaitu blok [**while_do**].

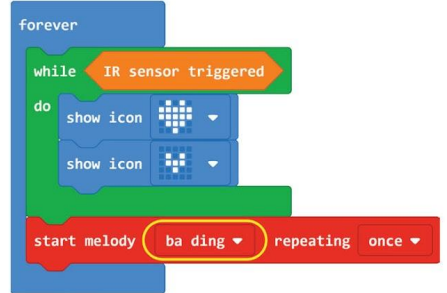


Langkah 3 Klik kategori [**Basic**] dan tambah dua blok [**show icon**]. Tukar salah satu ikon kepada “small heart”. Masukkan kedua-dua blok ke dalam blok [**while_do**].

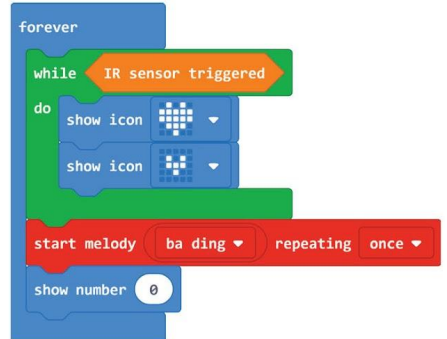
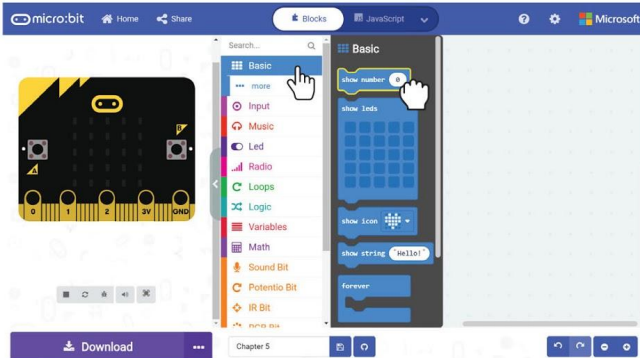




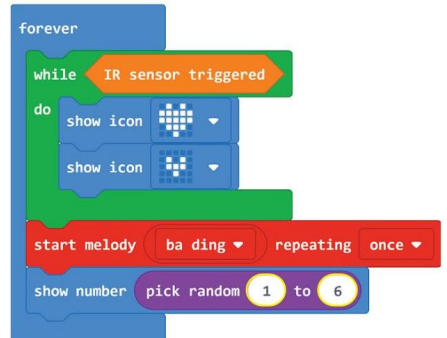
Langkah 4 Klik kategori [**Music**] dan pilih blok [**start melody_repeating_**]. Tukar melodi “dadadum” kepada “ba ding”.



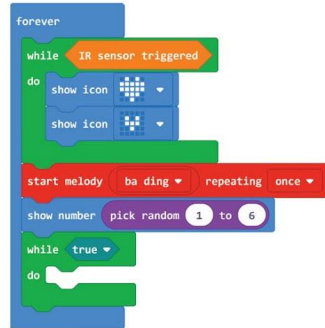
Langkah 5 Klik kategori [**Basic**] dan pilih blok [**show number**].



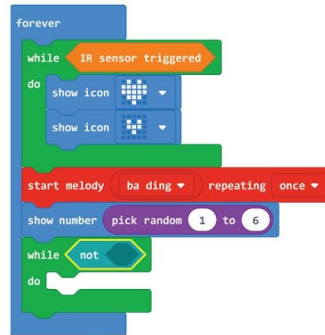
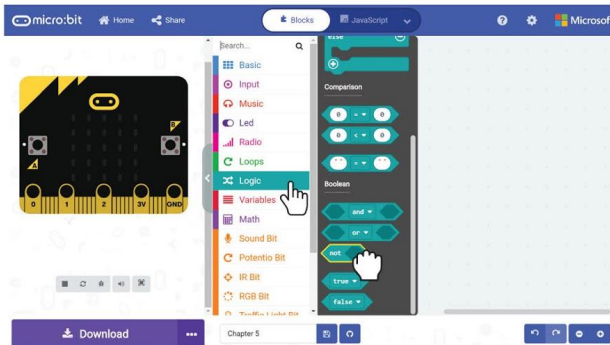
Langkah 6 Klik kategori [**Math**] dan pilih blok [**pick random_to_**]. Set nombor kepada 1 dan 6.



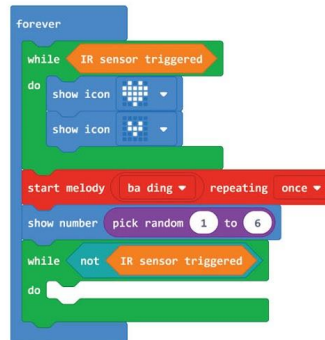
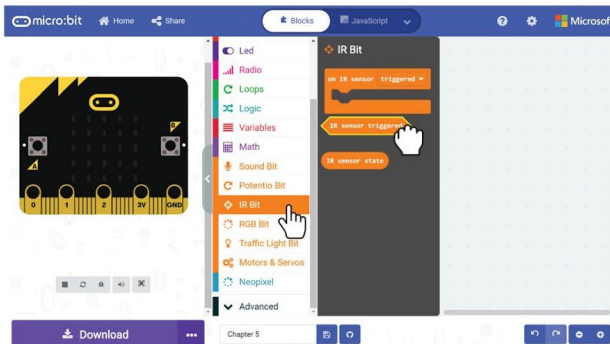
Langkah 7 Klik kategori [**Loops**] dan pilih blok [**while_do**].



Langkah 8 Klik kategori [**Logic**] dan pilih blok *Boolean* [**not**].



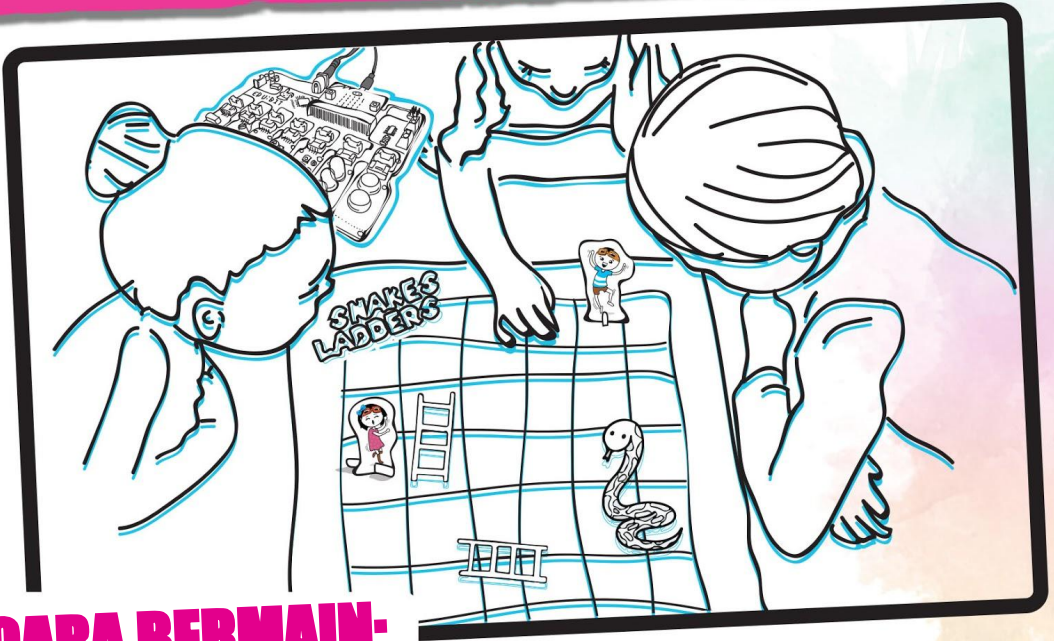
Langkah 9 Klik kategori [**IR Bit**] dan pilih blok [**IR sensor triggered**]. Masukkan blok itu ke dalam slot kosong blok [**not**].



Langkah 10 Pindahkan kod ke dalam EDU:BIT anda.

Jom Kita Main

Permainan 'Snakes and Ladders'



CARA BERMAIN:

Setiap pemain memilih satu watak dan letakkanya di bahagian 'Start Here'.

Pemain-pemain "roll the dice" mengikut giliran - letakkan tapak tangan di atas IR Bit. Apabila anda nampak animasi denyutan nadi, alihkan tapak tangan anda.

Gerakkan watak anda ke depan mengikut nombor yang tertera pada matriks LED (antara 1 hingga 6).

Jika watak anda berhenti di bawah tangga, anda boleh bergerak ke bahagian atas tangga. Jika watak anda berhenti pada kepala ular, anda mesti meluncur turun sehingga ekor ular.

Pemain pertama yang sampai pada nombor 100 ialah pemenang!

seperti ini



PERHATIAN!

Papan permainan Snakes and Ladders dan watak ada disediakan dalam kotak. Keluarkan bahagian watak dan penyokong. Kemudian, cantumkan bersama untuk membentuk watak yang dapat berdiri.

TAHUKAH

ANDA??



Kita telah menggunakan blok [**while_do**] daripada kategori [**Loops**] dalam kod kita sebelum ini. Adakah anda tahu bagaimana *while loop* berfungsi?

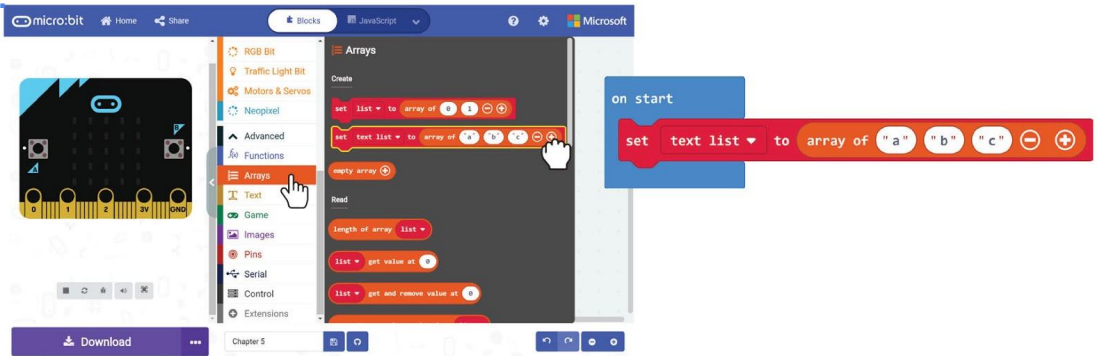
Apabila kita mengatur cara menggunakan blok [**while_do**], ia akan mengenal pasti syarat-syarat. Selagi syarat tersebut dipenuhi (atau *is TRUE*), maka atur cara itu akan melaksanakan kod-kod dalam blok [**while_do**]. Ia akan diulangi sehingga syarat *is NOT* dipenuhi (atau menjadi *FALSE*). Atur cara akan keluar daripada *while loop* dan melaksanakan kod-kod seterusnya.



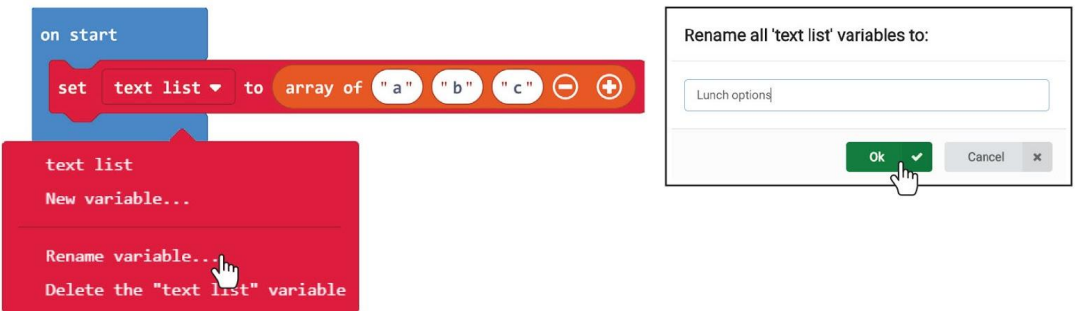


Selain menggunakan **EDU:BIT** sebagai dadu digital, anda juga boleh mengubahsuai pengaturcaraan asal untuk membolehkan **EDU:BIT** anda menentukan pilihan anda disebabkan sukar untuk membuat pilihan yang sedia ada. Sebagai contoh menentukan menu makanan tengah hari yang lazat, Yum yum.

Langkah 11 Klik [**Advanced**] : kategori [**Arrays**] dan pilih blok [**set text list to array of _ _ _**]. Masukkan blok tersebut ke slot [**Basic**] : [**on start**]



Langkah 12 Klik blok [**text list**] dan pilih “ **Rename variable**”. Taip ‘**Lunch options**’ di *pop up window* dan klik OK.

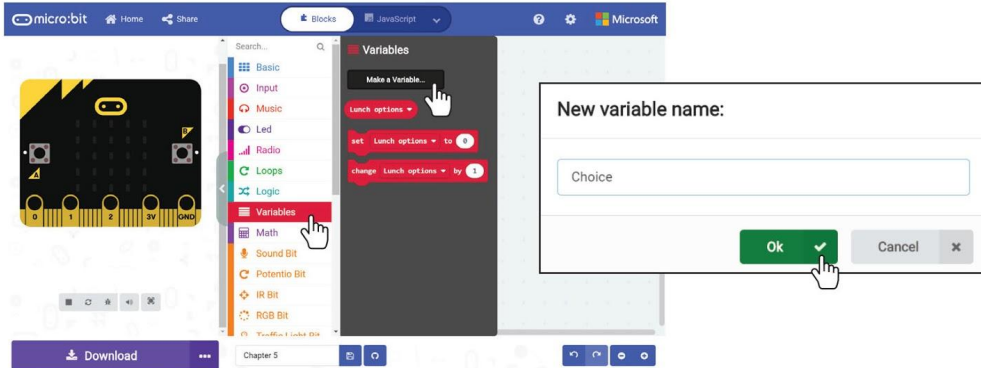


Langkah 13 Klik pada blok *array* satu persatu dan taipkan satu pilihan menu makanan tengah hari di setiap blok.

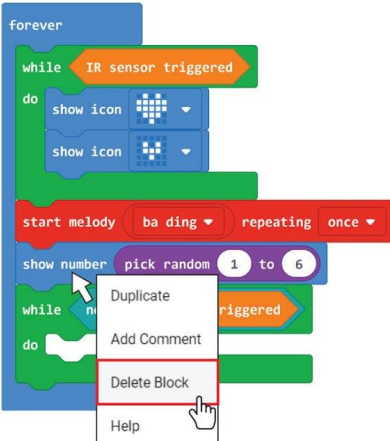


Klik simbol tambah ini untuk menambahkan pilihan.

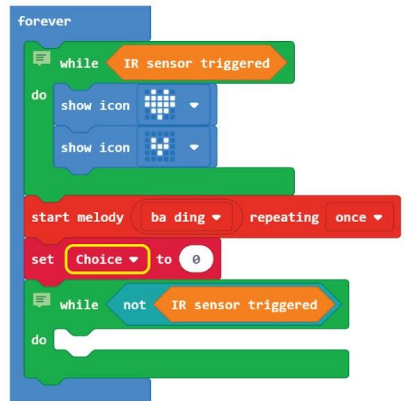
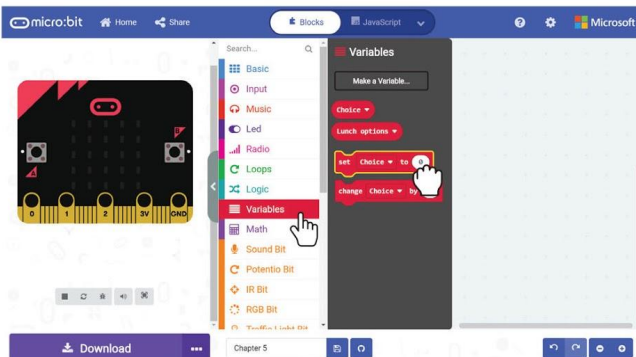
Langkah 14 Klik kategori [**Variable**] dan buat pembolehubah yang baharu bernama 'Choice'.



Langkah 15 Klik kanan pada blok [**show number [pick random _ to _]**] dan pilih 'Delete Blocks'.

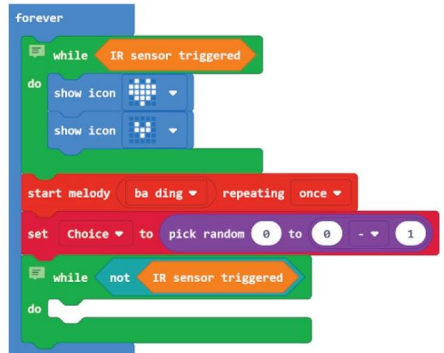
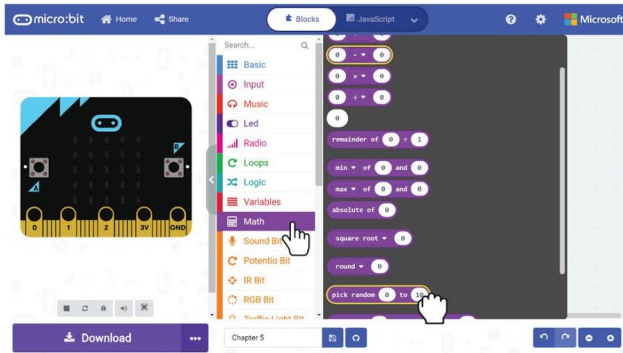


Langkah 16 Klik kategori [**Variable**] dan pilih blok [**set _ to _**]. Letakkan blok tersebut di antara blok [**start melody _ repeating _**] dan blok [**while _ do**]. Tukarkan pembolehubah kepada 'Choice'.

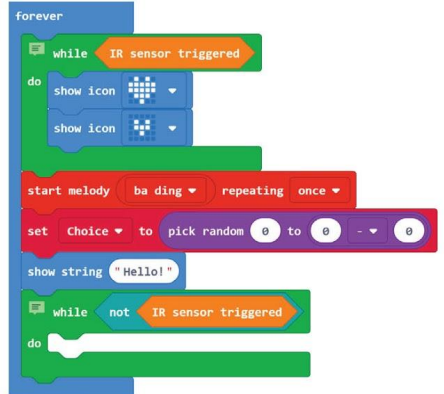
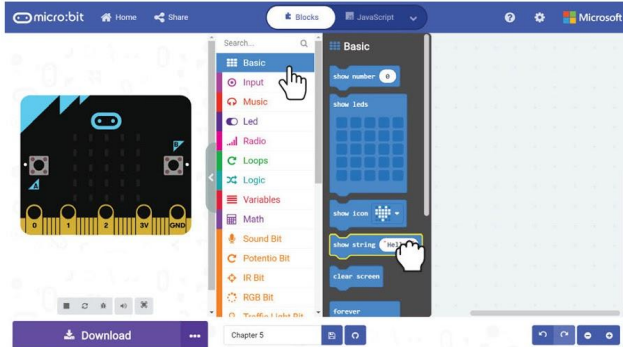




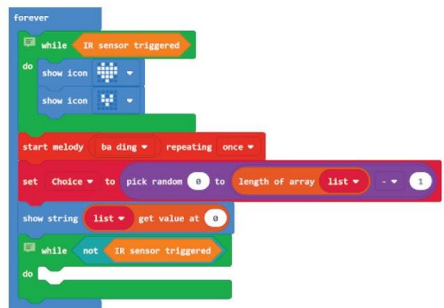
Langkah 17 Klik kategori [**Math**] dan tambahkan blok [**pick random _ to _**] dan [**_ - _**] kepada pengaturcaraan anda. Tukarkan nilai terakhir kepada 1.



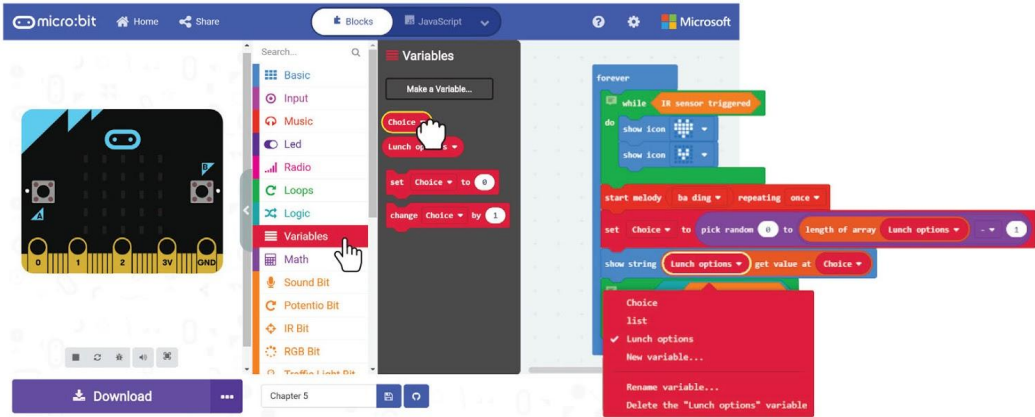
Langkah 18 Klik kategori [**Basic**] dan pilih blok [**show string**].



Langkah 19 Klik kategori [**Advanced**] : [**Arrays**] dan blok [**length of array _**] dan blok [**_ get value at _**] ke dalam kod.



Langkah 20 Klik pada [list] dan pemboleh ubah kepada 'Lunch options' kedua-dua blok. Akhir sekali, klik kategori [Variable] dan pilih blok [Choice]. Masukkan blok tersebut ke slot yang kosong di blok [_ get value at _].



Ini adalah pengaturcaraan lengkap "What's for lunch?":

```
on start
  set Lunch options to array of "Fried rice" "Spaghetti" "Nasi lemak"

forever
  while IR sensor triggered
  do
    show icon
    show icon

  start melody ba ding repeating once
  set Choice to pick random 0 to length of array Lunch options
  show string Lunch options get value at Choice

  while not IR sensor triggered
  do
```

Ini adalah contoh-contoh menu pilihan. Anda boleh menukar dan menambah menu-menu yang anda sukai.



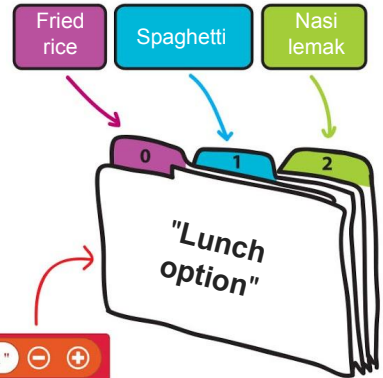
Apabila anda tidak pasti apa yang akan dimakan, anda boleh membiarkan EDU:BIT memutuskan untuk anda dengan mencetuskan IR Bit dan menjauhkan tangan anda dari itu. Anda juga boleh mengubah atur cara ini untuk membantu anda memutuskan permainan mana yang akan dimainkan dengan rakan anda. Adakah anda tahu apa yang perlu anda ubah untuk melakukannya?

TAHUKAH

ANDA??

Array adalah senarai atau koleksi pemboleh ubah yang berkaitan. Anda boleh menganggapnya sebagai fail dengan beberapa bahagian, dan setiap bahagian digunakan untuk menyimpan sebilangan maklumat. Kami menggunakan **array** sehingga kami dapat dengan mudah mengubah pengaturcaraan kami apabila perlu menambahkan atau membuang elemen-elemen dari senarai sedia ada.

Dalam atur cara ini, misalnya, kita membuat **array** dengan tiga elemen dan menamakannya "**Lunch option**". Kita kemudian dapat dengan mudah mengedit item makanan yang ditunjukkan oleh setiap elemen. Kita juga dapat menambah pilihan atau mengurangi jumlah elemen dalam senarai dengan hanya mengklik butang (+) atau butang (-).



```
on start
  set Lunch options to array of "Fried rice" "Spaghetti" "Nasi lemak"
```

```
forever
  while IR sensor triggered
  do
    show icon [grid icon]
    show icon [grid icon]
  start melody [ba ding] repeating once
  set Choice to pick random 0 to length of array Lunch options
  show string Lunch options get value at Choice
  while not IR sensor triggered
  do
```

Jika kondisi ini TIDAK ditemui, EDU:BIT akan melakukan perkara berikut

(i) memainkan melodi 'ba ding' sekali

(ii) memilih item secara rawak dari senarai pilihan makan tengah hari untuk menjadi 'pilihan', dan

(iii) tunjukkan pada paparan LED

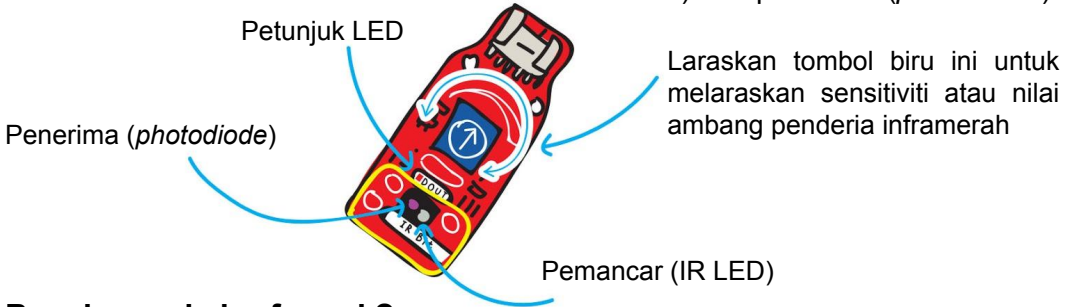


Dalam pengaturcaraan, kita mula mengira dari 0, bukannya 1. Oleh itu, "Nasi goreng" berada di nombor indeks 0 dalam senarai dan elemen terakhir "Nasi lemak" berada di nombor indeks 2, walaupun ia adalah item ketiga.

FAKTA MENARIK!



Penderia inframerah (IR) adalah instrumen elektronik yang biasanya digunakan untuk mengesan halangan. Penderia inframerah terdiri daripada dua bahagian iaitu pemancar (diod pemancar cahaya (LED) IR) dan penerima (*photodiode*).

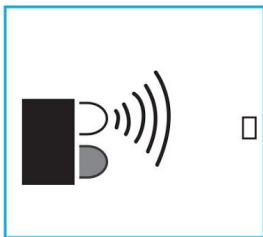


Bagaimana ia berfungsi ?

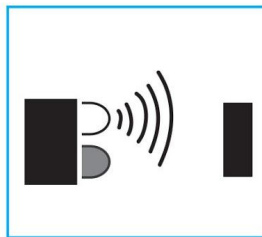
Diod pemancar cahaya inframerah (IR LED) mengeluarkan cahaya inframerah yang akan dipantulkan kepada penerima jika objek diletakkan di depan penderia. IR bit akan terangsang jika jumlah cahaya yang terpantul lebih besar daripada nilai ambang. Apabila terangsang, penunjuk LED ke atas IR Bit akan menyala.

Jika tidak ada objek atau objek itu terlalu jauh, maka sangat sedikit atau tiada cahaya inframerah akan terpantul kepada penerima. Oleh yang demikian IR Bit tidak akan terangsang.

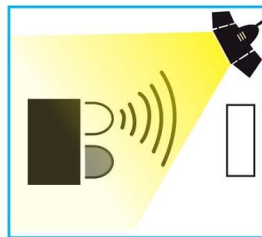
Walau bagaimanapun, penderia IR mungkin tidak berfungsi seperti yang dijangka dalam keadaan berikut.



Objek terlalu kecil.



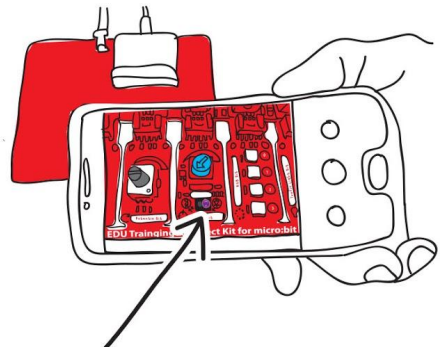
Objek mempunyai permukaan hitam atau gelap.



Gangguan cahaya.

Adakah awak tahu?

Lampu inframerah tidak kelihatan dengan mata kasar, Walau bagaimanapun, anda boleh melihat LED inframerah dengan hanya melihat melalui kamera telefon

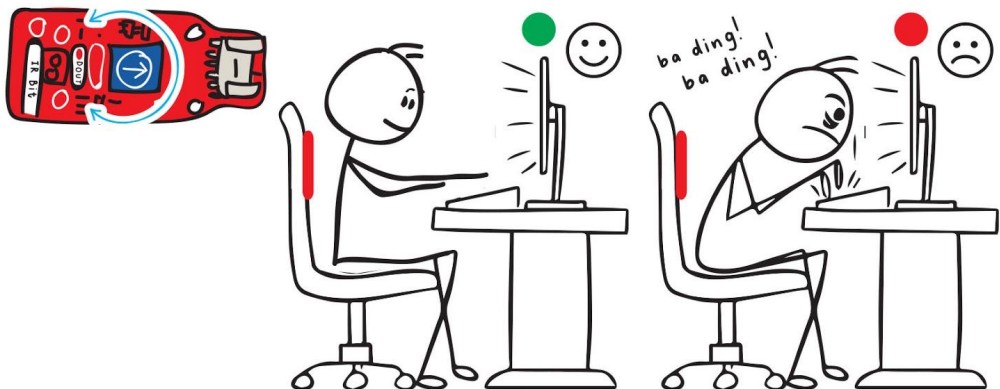


CABARAN APLIKASI

| | |
|---|---|
| Programkan EDU:BIT untuk berfungsi sebagai pengesan anti-kebongkolan | |
| <i>On start</i> | Peringatan Tatal -"Beri perhatian terhadap postur anda. " |
| <i>On IR triggered</i> | Menunjukkan wajah senyum pada paparan matrik LED dan menyalakan LED hijau. |
| <i>On IR NOT triggered</i> | Mulakan melodi 'ba ding' berulang kali. Menunjukkan wajah sedih pada paparan matrik LED dan LED merah berkelip. |

Bagaimana ia berfungsi?

Letakkan EDU:BIT di belakang kerusi anda seperti yang ditunjukkan. Duduk dengan selesa dengan postur yang baik. Laraskan tombol biru pada IR Bit sehingga penunjuk LED menyala (IR bit mengesan belakang anda). Proses ini dipanggil penentuukuran (calibration).



Anda perlu melakukan penentuukuran semula IR Bit jika anda memakai baju berwarna yang berbeza. Adakah anda tahu mengapa?

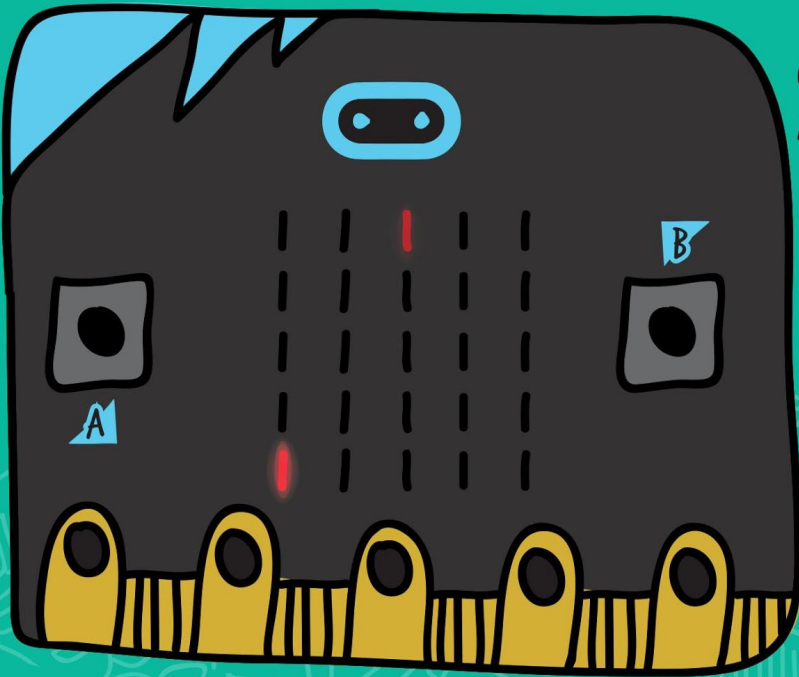


Tag, You're It!

Potentio Bit

Cepat !!!

Ayuh !!



Imbas saya!



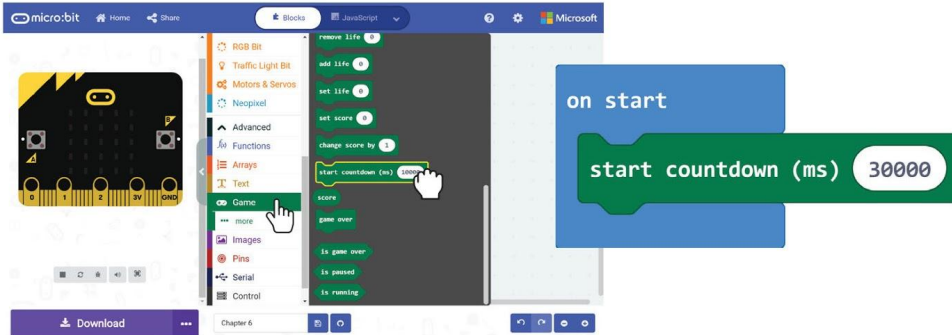
link.cytron.io/edubit-chapter-6



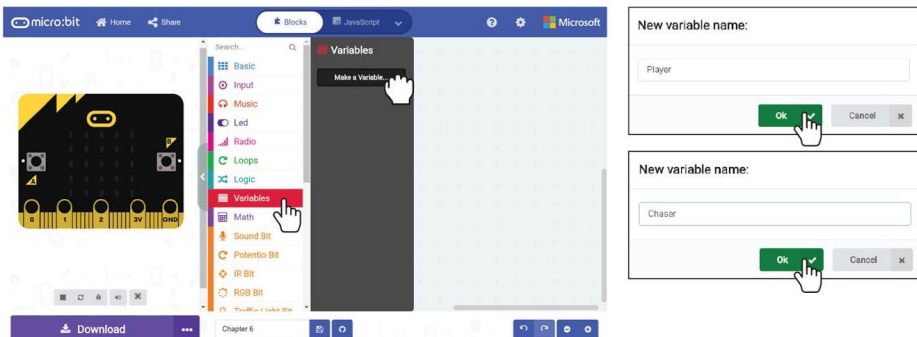
Bab 6 : Tag, You're It!

JOM KITA KOD!

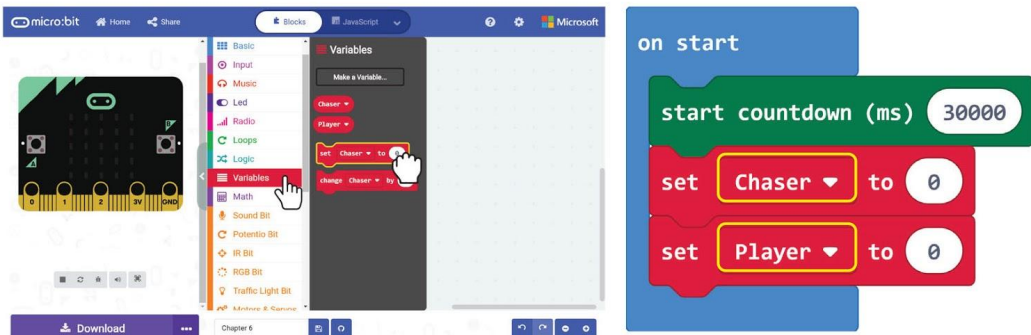
Langkah 1 Dalam Editor MakeCode anda, cipta projek baru dan tambah *EDU:BIT extension* (anda boleh merujuk kepada halaman 40). Klik [**Advanced**] dan kemudian kategori [**Game**]. Pilih blok [**start countdown (ms) _**], masukkan ke blok [**on start**] dan ubah nilai ke 30000.



Langkah 2 Klik kategori [**Variables**] dan klik [**Make a Variable**]. Taip 'Player' dalam *pop up window* dan kemudian klik OK. Cipta pemboleh ubah lain bernama 'Chaser'.

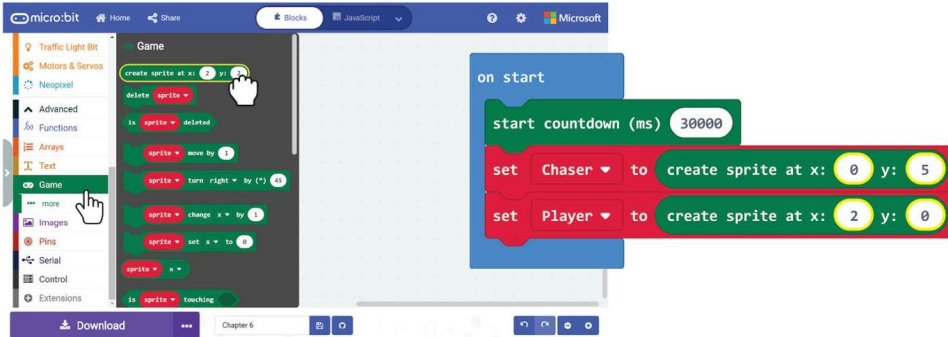


Langkah 3 Klik kategori [**Variables**] dan kemudian pilih blok [**set_to_**]. Salin blok [**set_to_**] dan masukkan kedua-dua blok ke blok [**on start**]. Tetapkan pemboleh ubah sebagai 'Chaser' dan 'Player'.

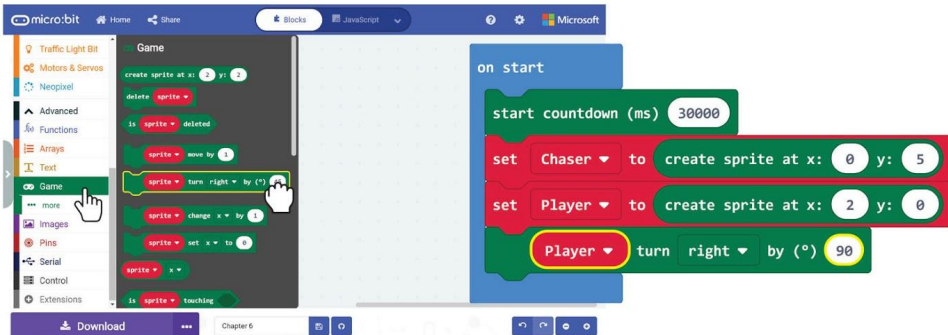


Bab 6 : Tag, You're It!

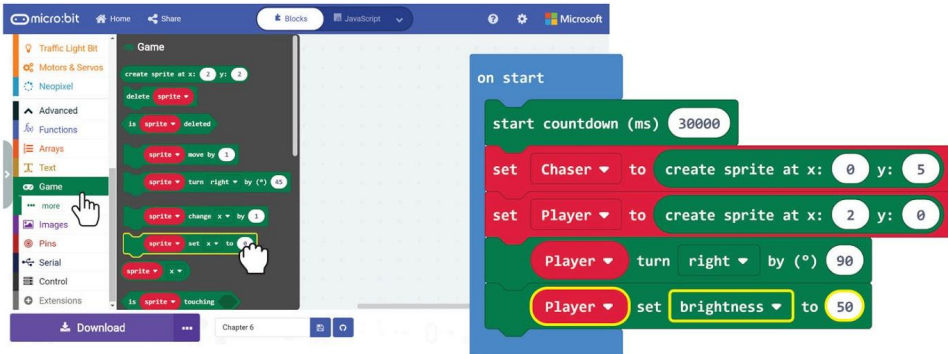
Langkah 4 Klik [**Advanced**] dan kemudian kategori [**Game**]. Pilih blok [**create sprite pada x: _ y: _**]. Salin dan masukkan blok-blok tersebut ke blok [**set_to_**]. Tukar nilai kepada **x:0 y:5** untuk 'Chaser' dan **x:2 y:0** untuk 'Player'.



Langkah 5 Klik kategori [**Advanced**]: [**Game**] dan pilih blok [**_ turn_by (°)**]. Letakkan blok dalam slot [**on start**]. Pilih pemboleh ubah 'Player' dan tetapkan darjah kepada 90.

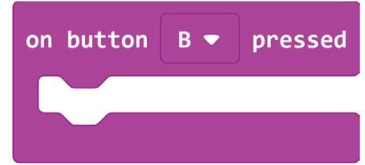
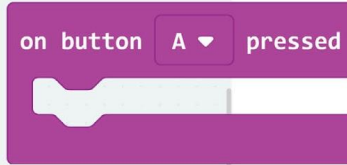
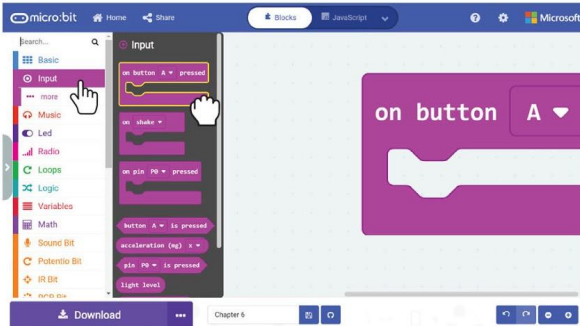


Langkah 6 Klik kategori [**Advanced**]: [**Game**] dan pilih blok [**_ set_to_**]. Pilih pemboleh ubah 'Player', ubah 'x' kepada 'brightness' dan 'brightness' ke 50.

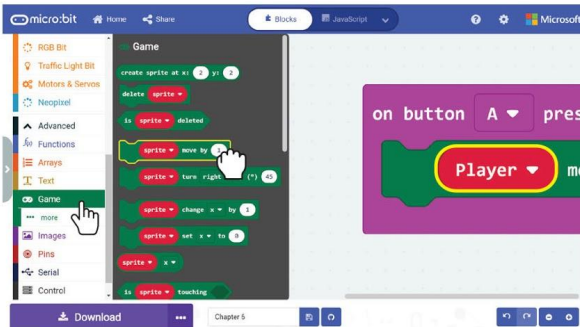




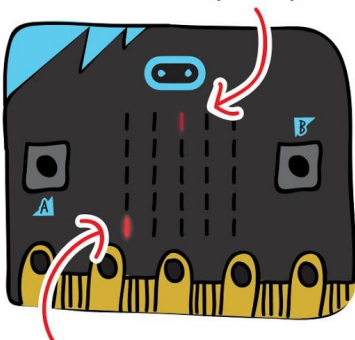
Langkah 7 Klik kategori [**Input**] dan pilih blok [**on button_pressed**]. Salin blok tersebut dan pilih tombol 'B' pada blok kedua [**on button_pressed**].



Langkah 8 Klik kategori [**Game**] dan pilih blok [**_ move by_**]. Salin dan letakkan blok ke dalam slot - slot [**on button_pressed**]. Pilih pemboleh ubah 'Player' untuk kedua-dua blok dan ubah nilai kepada -1 pada (on button A pressed) dan 1 pada (on button B pressed).



Player sprite



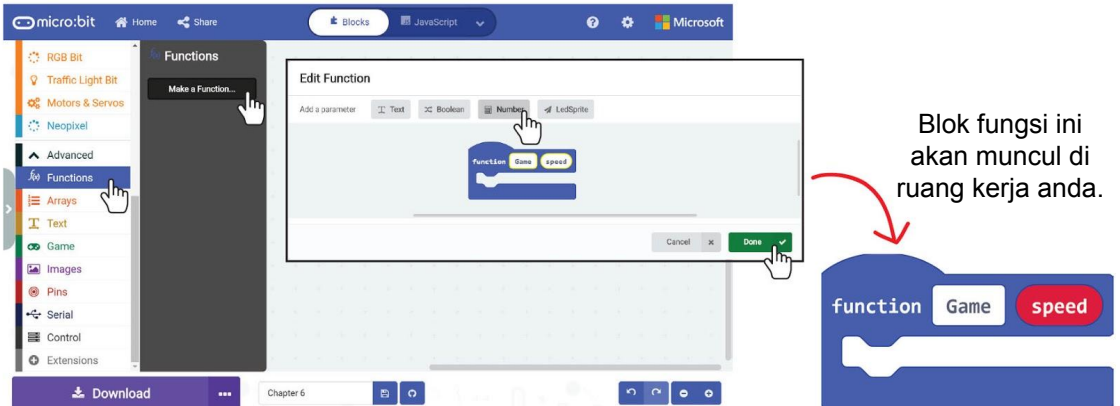
Chaser sprite

Pindahkan atur cara ke EDU:BIT anda. Apabila anda menekan butang biru (Butang B), Adakah anda perasan lampu LED yang malap bergerak ke bawah? Itulah *Player sprite*! **Sprite** ialah seperti "mahluk kecil LED" yang boleh anda kawal. Apa akan berlaku jika anda menekan butang kuning (butang A)?

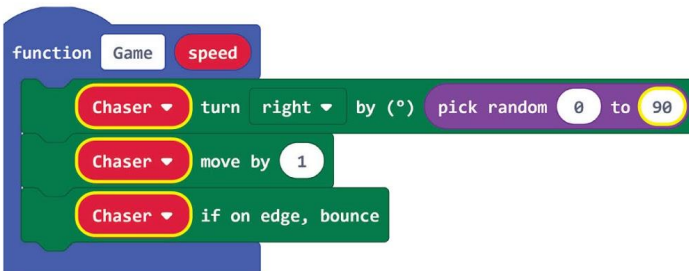
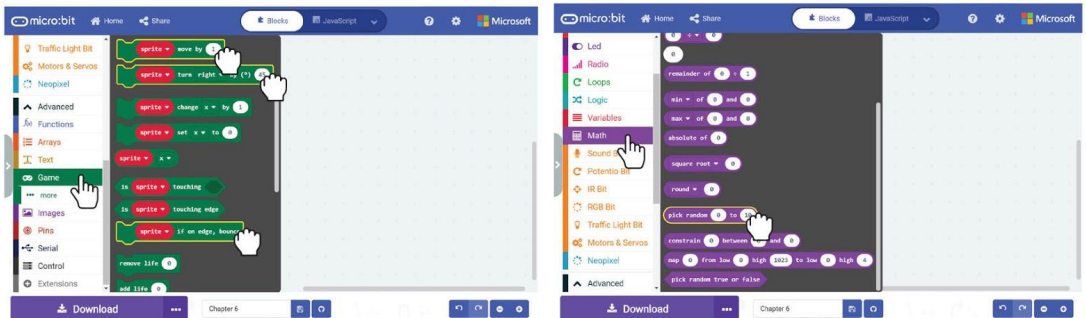


Bab 6 : Tag, You're It!

Langkah 9 Klik kategori **[Advanced]** dan pilih kategori **[Functions]**, kemudian klik **[Make a Function...]**. Dalam *Edit Function window*, namakan semula 'doSomething' untuk **'Game'**. Seterusnya, klik **[Number]** untuk menambah parameter dan menamakan semula 'num' kepada **'speed'** dalam blok fungsi. Kemudian klik **'Done'**.

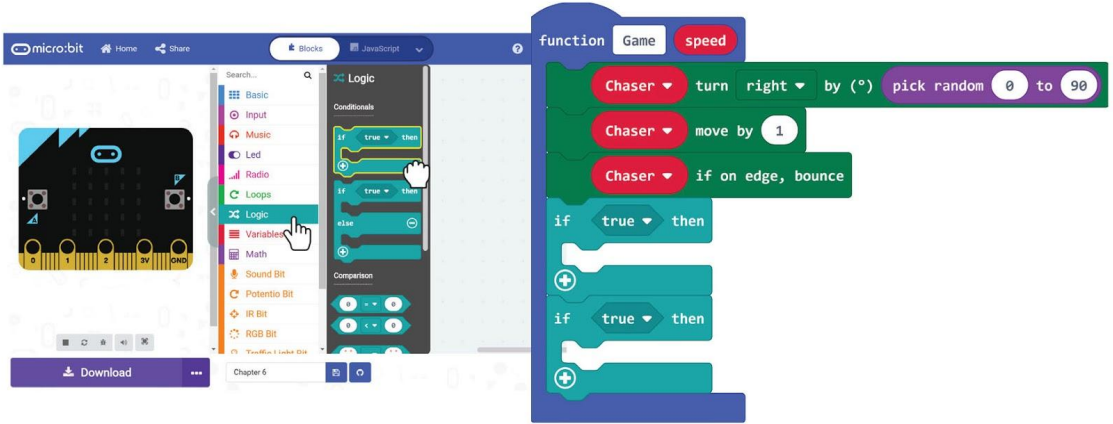


Langkah 10 Teruskan membina atur cara dengan menambah blok - blok dari kategori **[Advanced]:[Game]** dan **[Math]** seperti yang ditunjukkan di bawah. Peringatan untuk menukar pemboleh ubah kepada **'Chaser'** dan nilai kepada 90.

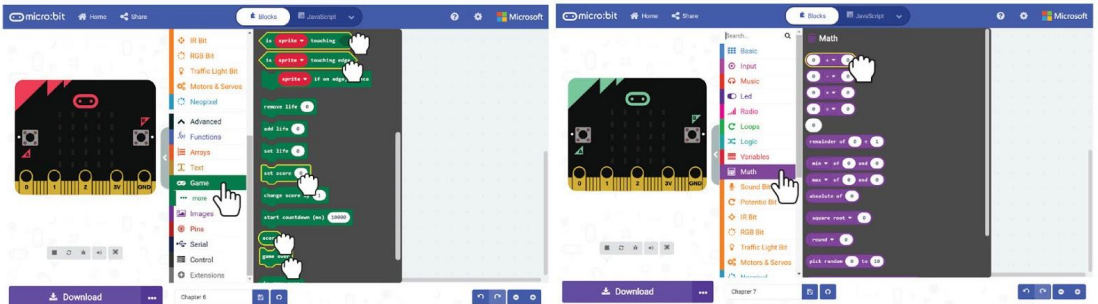




Langkah 11 Tambah dua blok [if-then] dari kategori [Logik] ke pengaturcaraan anda.



Langkah 12 Teruskan membina pengaturcaraan dengan menambah blok-blok dari [Advanced]: [Game] dan kategori [Math] seperti yang ditunjukkan di bawah.



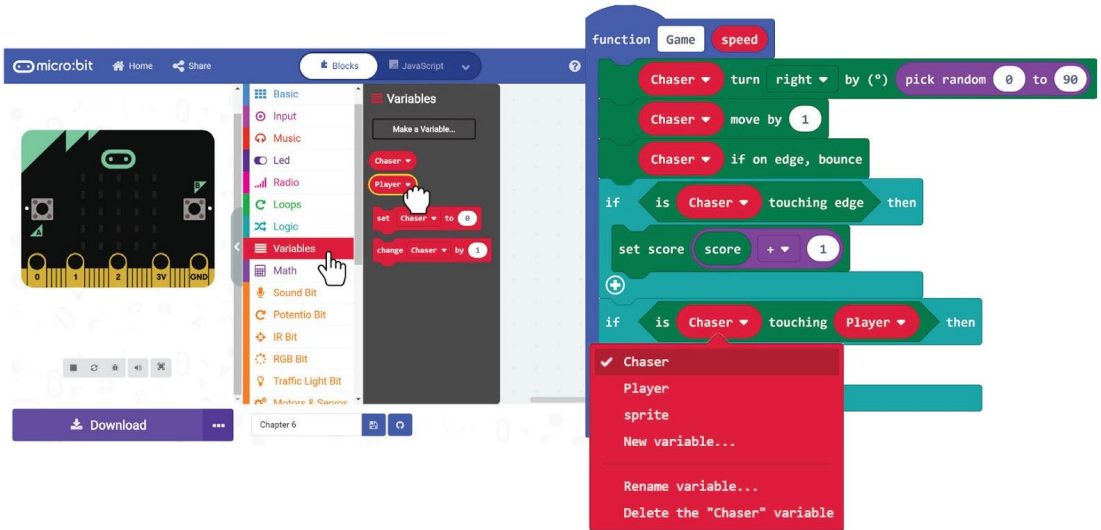
```
function Game speed
  Chaser turn right by (°) pick random 0 to 90
  Chaser move by 1
  Chaser if on edge, bounce
  if is sprite touching edge then
    set score score + 1
  if is sprite touching then
    game over
```

Jom teruskan!

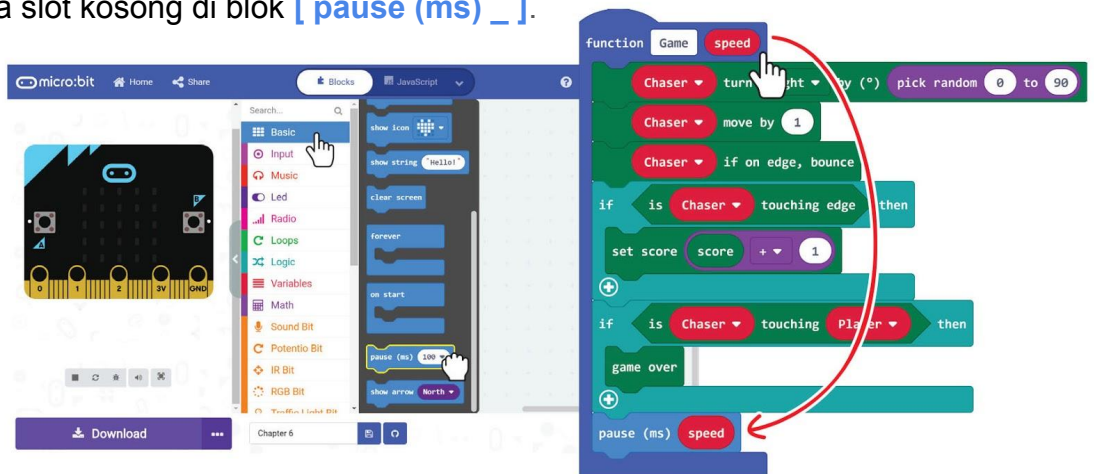


Bab 6 : Tag, You're It!

Langkah 13 Tukar kedua-dua blok [**sprite**] kepada 'Chaser' dengan mengkliknya dan kemudian pilih 'Chaser'. Klik kategori [**Variables**] dan pilih blok [**Player**]. Masukkan ia ke slot kosong blok [**is _ touching _**].



Langkah 14 Klik kategori [**Basic**] dan pilih blok [**pause (ms) _**]. Tambah kepada atur cara anda. Klik pada [**speed**] pada blok fungsi, pegang dan seret ia slot kosong di blok [**pause (ms) _**].



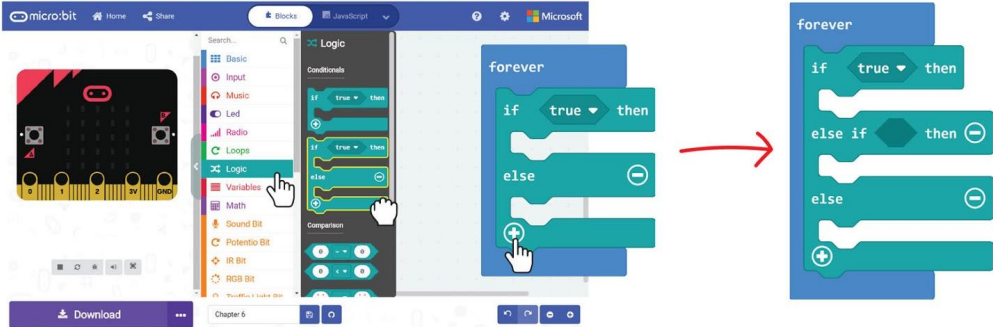
Anda boleh menambah keseronokan permainan dengan memainkan melodi apabila 'Chaser' menyentuh 'Player'. Bolehkah anda memikirkan blok mana yang perlu ditambah dan tempat untuk meletakkan blok?



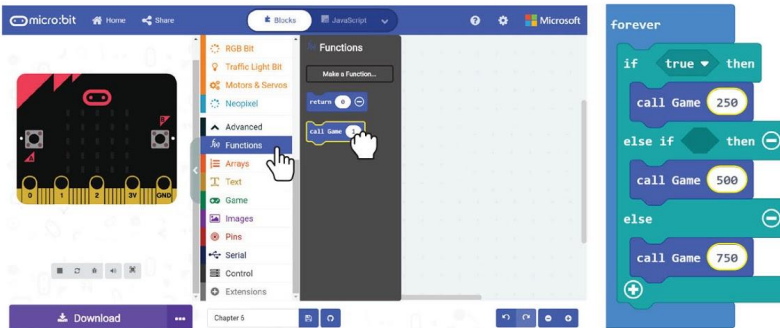


Sekarang, mari kita tambah tahap kesukaran yang berbeza pada permainan!

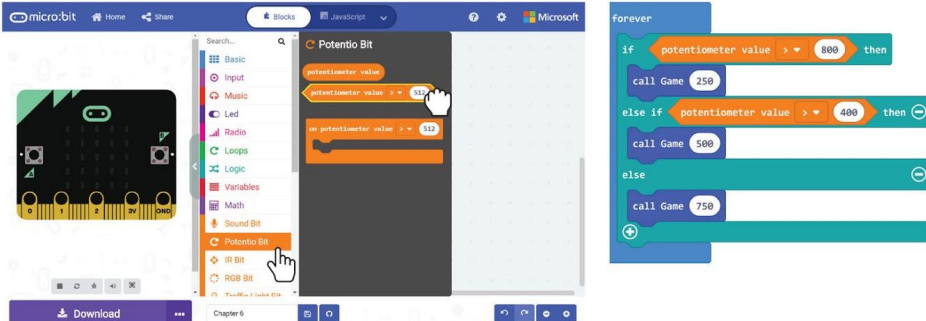
Langkah 15 Klik kategori [**Logik**] dan pilih blok [**if-then-else**]. Letakkan blok tersebut di slot [**forever**]. Klik butang (+) untuk menambah situasi pilihan lain yang difikirkan perlu .



Langkah 16 Klik kategori [**Functions**] dan pilih blok [**call Game_**]. Duplikasikan blok dan masukkan blok-blok tersebut ke setiap slot [**if-then-else**]. Tukar nilai blok [**call Game_**] kepada 250, 500 dan 750 dengan sewajarnya.



Langkah 17 Klik kategori [**Potentiometer bit**] dan pilih blok [**potentiometer value > .**]. Duplikasikan dan letakkan blok-blok tersebut ke dalam slot-slot situasi pilihan blok [**if-then-else**]. Tetapkan nilai kepada 800 untuk blok pertama dan 400 untuk blok kedua.

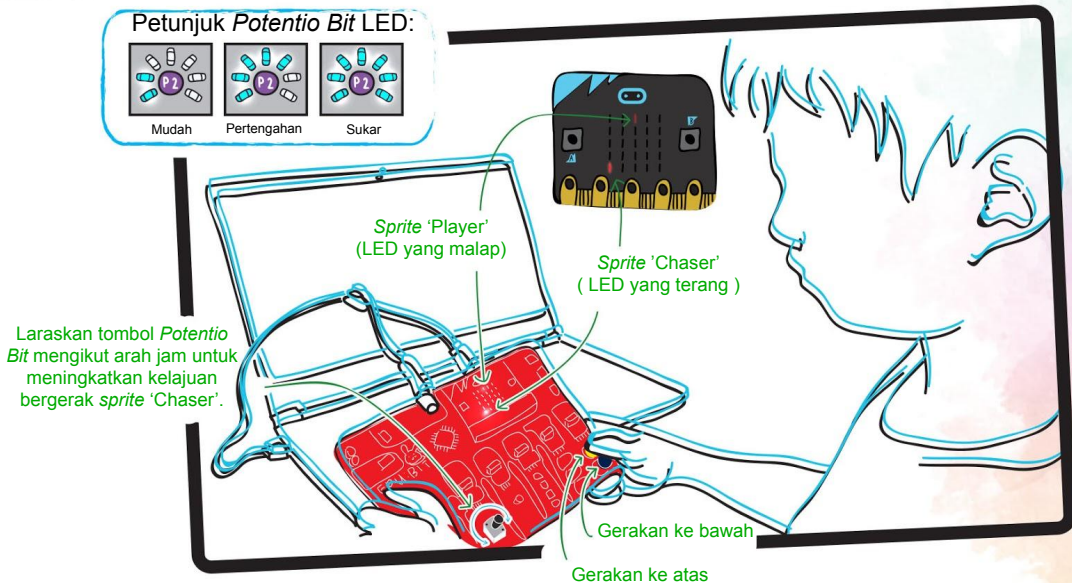


Berikut adalah pengaturcaraan yang lengkap.

| | |
|--|--|
| <p>Papar animasi, dan mulakan 30 saat <i>countdown timer</i>.</p> | <pre>on start start countdown (ms) 30000 set Chaser to create sprite at x: 0 y: 5 set Player to create sprite at x: 2 y: 0 Player turn right by (°) 90 Player set brightness to 50</pre> |
| <p>Wujudkan <i>sprite</i> 'Chaser' dan 'Player' pada koordinat yang ditetapkan.</p> | |
| <p>Pusing <i>sprite</i> 'Player' sebanyak 90 darjah (setkan supaya bergerak secara menegak, iaitu atas dan bawah).</p> | |
| <p>Malapkan <i>sprite</i> 'Player' supaya dapat dibezakan dengan <i>sprite</i> 'Chaser'.</p> | |
| <p>Setkan butang A dan B untuk mengerakkan <i>sprite</i> 'Player' 1 langkah ke atas atau ke bawah apabila butang ditekan.</p> | <pre>on button A pressed Player move by -1 on button B pressed Player move by 1</pre> |
| <p><i>Function Game</i> untuk mengawal <i>sprite</i> 'Chaser'</p> | <pre>function Game speed Chaser turn right by (°) pick random 0 to 90 Chaser move by 1 Chaser if on edge, bounce if is Chaser touching edge then set score score + 1 if is Chaser touching Player then start melody wawawaaa repeating once game over pause (ms) speed</pre> |
| <p>Set <i>sprite</i> 'Chaser' untuk pusing kanan pada sudut rawak antara 0 dan 90 darjah dan bergerak 1 langkah. Jika <i>sprite</i> 'Chaser' tersentuh bahagian tepi, ia akan melantun kembali.</p> | |
| <p>Setiap kali <i>sprite</i> 'Chaser' tersentuh bahagian tepi, tambah 1 markah pada pemboleh ubah 'score'.</p> | |
| <p>Jika <i>sprite</i> 'Chaser' tersentuh <i>sprite</i> 'Player', mainkan melodi 'wawawaaa' dan animasi 'Game Over' serta 'score'.</p> | |
| <p>Hentikan program (ms) berdasarkan nilai pemboleh ubah 'speed' yang ditetapkan.</p> | |
| <p>Selasa periksa bacaan <i>Potentiometer Bit</i>.</p> <p>Jika (IF) nilai <i>potentiometer</i> lebih besar daripada 800, panggil <i>function</i> 'Game' dan setkan pemboleh ubah 'speed' kepada 250 ms.</p> <p>Atau jika (ELSE IF) nilai <i>potentiometer</i> lebih besar daripada 400, panggil <i>function</i> 'Game' dan setkan pemboleh ubah 'speed' kepada 500 ms.</p> <p>Atau (ELSE) panggil <i>function</i> 'Game' dan setkan pemboleh ubah 'speed' kepada 750 ms.</p> | <pre>forever if potentiometer value > 800 then call Game 250 else if potentiometer value > 400 then call Game 500 else call Game 750</pre> |

Langkah 18 Muat naik pengaturcaraan yang lengkap ke EDU:BIT anda dan bermain **Tag, You're It!** yang menyeronokkan bersama-sama rakan anda.

Jom Kita Main



CARA BERMAIN:

Apabila EDU:BIT dihidupkan, *sprite* 'Chaser' akan terus bergerak secara rawak. Gerakkan *sprite* 'Player' ke atas atau ke bawah untuk mengelakkan *sprite* 'Chaser'. Tekan butang kuning (butang A) untuk bergerak ke atas dan butang biru (butang B) untuk bergerak ke bawah.

Permainan tamat jika 'Player' disentuh oleh 'Chaser', atau selepas had masa 30 saat tamat.

Setiap kali *sprite* 'Chaser' "menyentuh" penjurus matrik LED, anda akan skor 1 mata. Pemain dengan skor tertinggi adalah pemenang! Selamat bergembira

TIP!

#1 Untuk mendapatkan skor yang lebih tinggi dalam tempoh 30 saat had, anda boleh meningkatkan kelajuan *sprite* 'Chaser' supaya ia akan menyentuh penjurus lebih kerap.

#2 Selepas permainan tamat, anda boleh menekan butang A + B secara serentak untuk memulakan permainan baru. Ini adalah fungsi terbina dalam blok [Game].

TAHUKAH

ANDA??

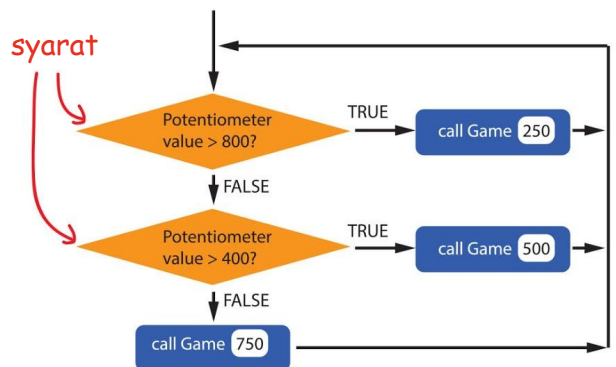
Dalam pengaturcaraan, kita menggunakan pernyataan bersyarat untuk membuat keputusan. Dalam MakeCode, kita menggunakan blok bersyarat [**if-then**] atau [**if-then-else**] dari kategori [**Logik**] untuk membentuk keadaan. Program ini akan menyemak kenyataan bersyarat dan jika ia BENAR, ia akan melaksanakan atur cara dalam blok bersyarat. Jika ia adalah SALAH, ia bergerak ke blok atur cara seterusnya.

```
if [is Chaser touching Player] then
  start melody [wawawawaa] repeating [once]
  game over
```

Jika kondisinya seperti berikut (i.e. *sprite Chaser* menyentuh *sprite Player*),
Maka tindakannya seperti ini (memainkan melodi wawawawaa, kemudian memaparkan animasi 'game over').

Apabila kita mempunyai pelbagai syarat, program ini akan menilai syarat-syarat secara berurutan, dari atas ke bawah, dan melaksanakan atur cara pernyataan bersyarat pertama yang BENAR. Oleh itu, kedudukan urutan kenyataan bersyarat yang pertama mempunyai keutamaan yang lebih tinggi berbanding dengan yang berada di akhir. Sebagai contoh, atur cara di dalam permainan menentukan kelajuan bergerak *sprite* 'Chaser' dengan membandingkan nilai *potentiometer* terhadap ambang yang telah ditetapkan..

```
forever
  if [potentiometer value > 800] then
    call Game [250]
  else if [potentiometer value > 400] then
    call Game [500]
  else
    call Game [750]
```



Jika nilai *potentiometer* > 800, panggil fungsi *Game* (dengan kelajuan pemboleh ubah diset ke 250 ms) akan diberikan keutamaan terlebih dahulu. Seterusnya jika ia tidak menepati pernyataan bersyarat pertama, maka ia akan beralih ke pernyataan bersyarat kedua iaitu dengan nilai *potentiometer* > 400 pula, (dengan kelajuan pemboleh ubah diset ke = 500 ms), dan seterusnya sehinggalah ke pernyataan bersyarat terakhir iaitu *function Game* (dengan kelajuan pemboleh ubah diset ke = 750 ms).

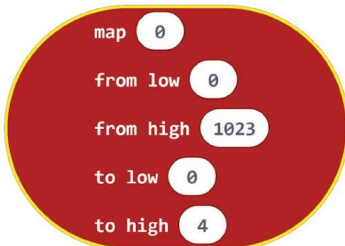


Terokai Blok-Blok Lain

#1 Gunakan blok [**Basic**] : [**show number**] dengan [**Potentiometer Bit**] : [**potentiometer value**] untuk dapatkan bacaan dan paparan nilai semasa *potentiometer*.

show number potentiometer value

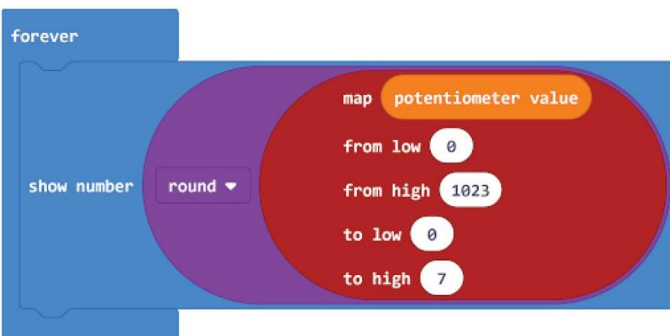
#2 *Potentiometer* akan menghasilkan nilai antara 0 sehingga 1023. Anda boleh menggunakan blok [**map _ from low _ from high _ to low _ to high _**] daripada kategori [**Advanced**] : [**Pins**] untuk memetakan julat bacaan yang diterima bermakna.



#3 Blok pemetaan akan memberikan bacaan nombor perpuluhan (contohnya 1.68, 3.998). Untuk membundarkan nombor perpuluhan tersebut, gunakan blok [**round _**] daripada kategori [**Math**].

round 0

Berikut adalah contoh pengaturcaraan untuk pemetaan bacaan Potentiometer Bit dengan nilai bacaan 0 ke 7. LED Matrik akan memaparkan nilai bacaan yang telah dibundarkan.



Peringatan! Pastikan EDU:BIT telah dihidupkan untuk mendapatkan bacaan yang tepat.



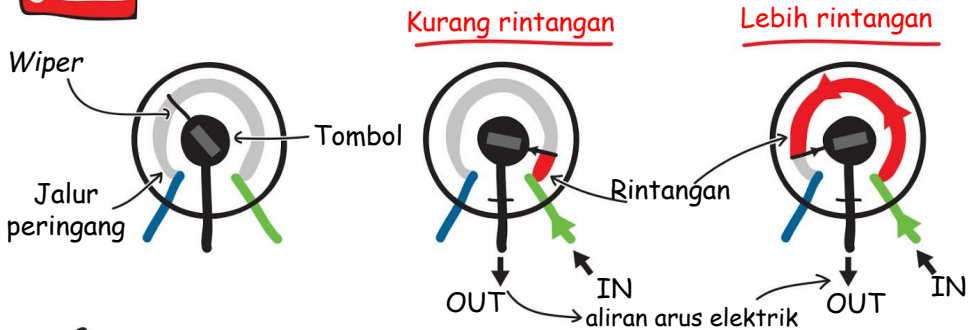
FAKTA MENARIK!



Perintang boleh laras juga dirujuk sebagai *pots*, perintang pemboleh ubah dengan rintangan yang boleh dilaraskan dengan mudah menggunakan tombol atau penggelongsor.



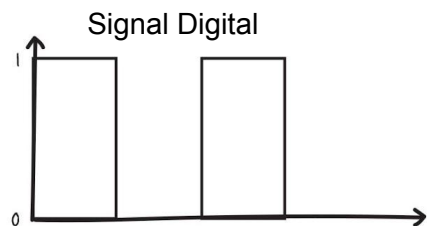
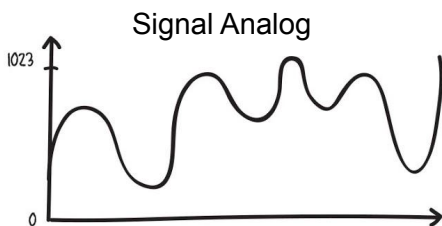
Jika anda mempunyai 10,000 ohm perintang boleh laras, anda boleh dapatkan nilai rintangan antara 0 ohm ke 10,000 ohm dengan menukarkan posisi kedudukan tombol atau penggelongsor.



Berikut adalah beberapa aplikasi harian:

- Kawalan audio pembesar suara.
- Kawalan frekuensi radio.
- Kawalan suhu pemanas air.

Perintang boleh laras pada EDU:BIT adalah sejenis peranti input analog. Ia mengukur potensi elektrik dan menukarkan ukuran voltan (antara 0 V hingga 3.3 V) ke dalam nilai integer antara 0 dan 1023.



CABARAN APLIKASI

Programkan EDU:BIT untuk berfungsi sebagai pencatat masa. Guna Potentio Bit untuk melaras jangka masa (antara 0 kepada 60 saat), Butang A untuk mengaktifkan pencatat masa dan Butang B untuk menetapkan semula pencatat masa.

| | |
|---|--|
| <i>On start</i> | Setkan <i>Mode</i> kepada 0. |
| <i>On Button A pressed</i> (Butang kuning) | Setkan <i>Mode</i> kepada 1. Setkan masa mula untuk memulakan aktiviti. Paparkan ikon muka senyum. |
| <i>On Button B pressed</i> (Butang biru) | Setkan <i>Mode</i> kepada 1. |
| <i>Forever</i> | Sentiasa semak <i>Mode</i> . <ul style="list-style-type: none">• <i>IF Mode = 0</i>, maka set jangka masa kepada nilai bacaan perintang boleh laras yang telah dibundarkan kepada nilai rendah = 0 dan nilai tinggi = 60 serta dipaparkan nilai jangkamasa di matrik LED.• <i>ELSE IF Mode = 1</i>, semak sama ada (<i>running time - Start Time</i>) > (<i>Duration x 1000</i>). Jika BENAR, mainkan melodi wawawawaa dan kemudian set <i>Mode = 2</i>.• <i>ELSE</i>, tunjukkan ikon muka sedih. |

Tip untuk anda:

Tip #1 Anda perlu membina tiga pemboleh ubah ini : *Mode*, *Start Time* dan *Duration*.

Tip #2 *Running time (ms)* blok adalah daripada kategori [Input].

Tip #3 Gunakan penyataan bersyarat untuk menyemak sama ada masa telah tamat.



running time (ms)

- ▾

Start Time ▾

≥ ▾

Duration ▾

x ▾

1000

Bab 7

Jom Kita Dengar Tepukan!

Sound Bit



Imbas saya!



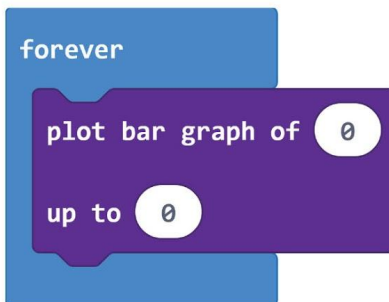
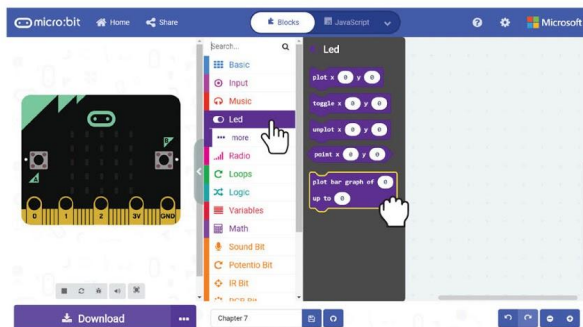
link.cytron.io/edubit-chapter-7



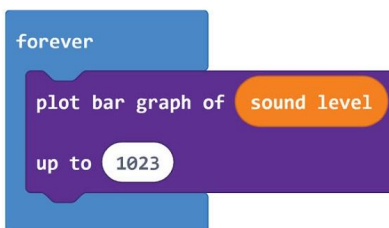
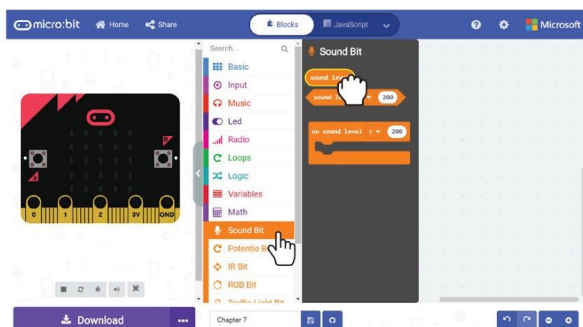


JOM KITA KOD!

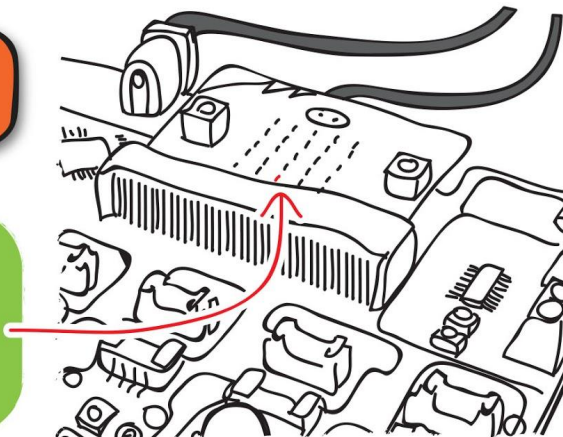
Langkah 1 Wujudkan projek baharu dalam *MakeCode editor* anda dan menambah EDU:BIT *extension* (anda boleh merujuk kepada halaman 40). Klik kategori [**Led**] dan pilih blok [**plot bar graph of up to**]. Masukkan blok tersebut ke slot [**forever**].



Langkah 2 Klik kategori [**Sound Bit**] dan pilih blok [**sound level**]. Masukkan blok tersebut ke blok [**plot bar graph of up to**] dan tukar nilai kedua 0 kepada 1023.



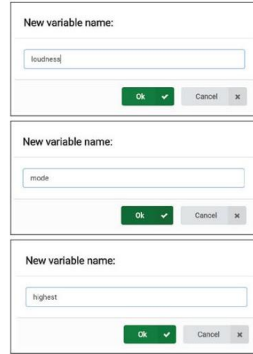
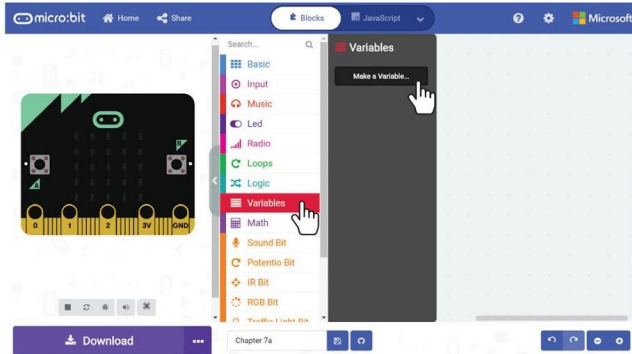
Langkah 3 Pindahkan pengaturcaraan ke EDU:BIT anda. Perhatikan paparan matrik LED apabila anda bertepuk tangan atau bermain dram di atas meja menggunakan jari anda.



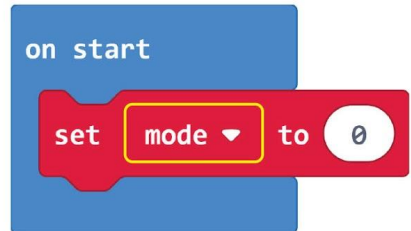
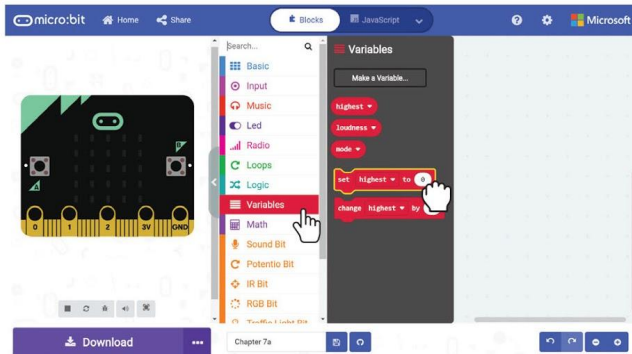


Sekarang mari kita tukarkan EDU:BIT kepada *applause-o-meter*

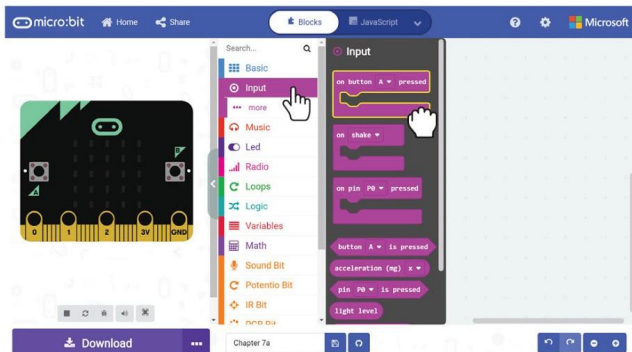
Langkah 4 Mulakan projek baru dan tambah EDU:BIT *extension*. Klik kategori [**Variables**] kemudian klik [**Make a Variable**]. Taip ' **mode** ' di dalam *pop up window* dan kemudian klik OK. Cipta dua pemboleh ubah bernama ' **loudness** ' dan ' **highest** ' .



Langkah 5 Dapatkan blok [**set _ to _**] daripada kategori [**Variable**]. Letakkan blok tersebut ke slot [**on start**]. Tetapkan pemboleh ubah kepada ' **mode** ' .

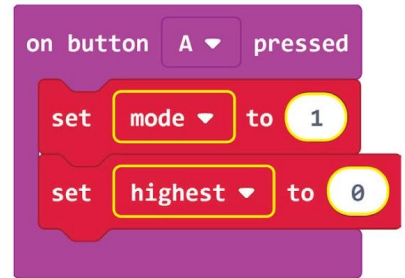
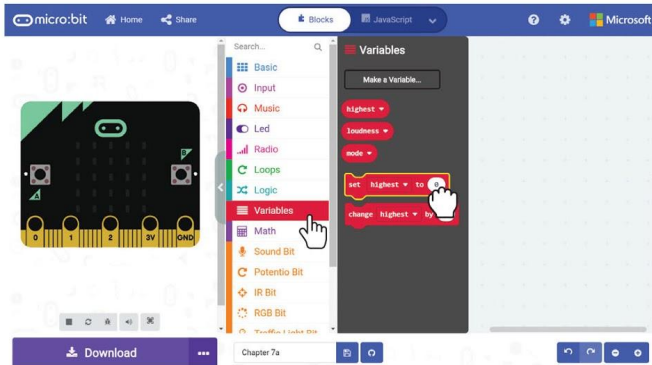


Langkah 6 Klik kategori [**Input**] dan pilih blok [**on button_pressed**] .

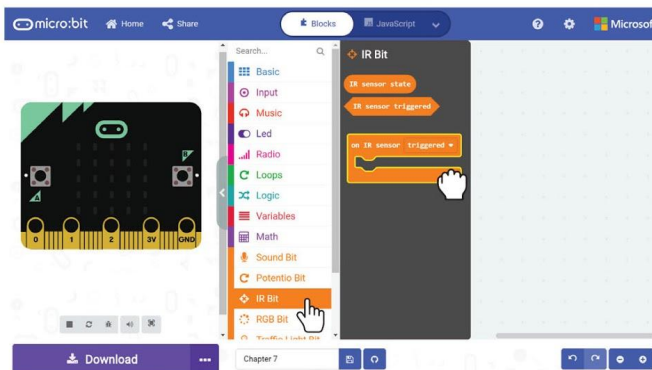




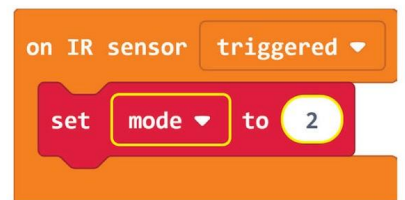
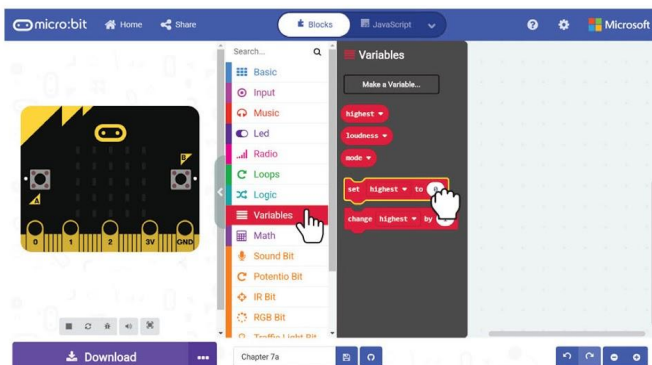
Langkah 7 Dapatkan dua blok **[set_to_]** daripada kategori **[Variables]** dan letakkan blok tersebut pada slot **[on button A pressed]** . Tetapkan pemboleh ubah yang pertama kepada **'mode'** dan bernilai 1, dan pemboleh ubah yang kedua kepada **'highest'** dengan penetapan nilai 0.



Langkah 8 Klik kategori **[IR Bit]** dan pilih blok **[on IR sensor triggered]** .

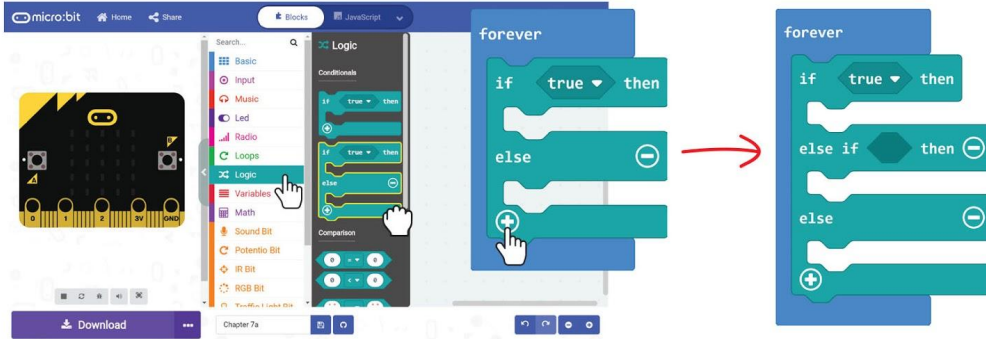


Langkah 9 Klik kategori **[Variables]** dan pilih blok **[set_to_]** . Masukkan blok tersebut ke dalam slot **[on IR sensor triggered]** dan tukar pemboleh ubah kepada **'mode'** dan nilai kepada **2** .

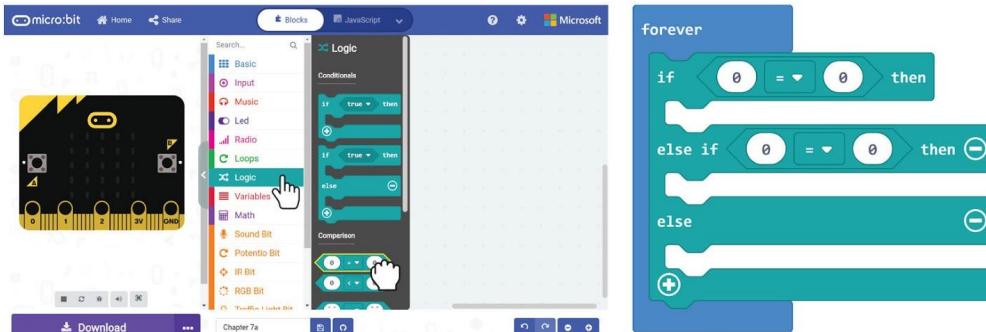


Bab 7 : Jom Kita Dengar Tepukan!

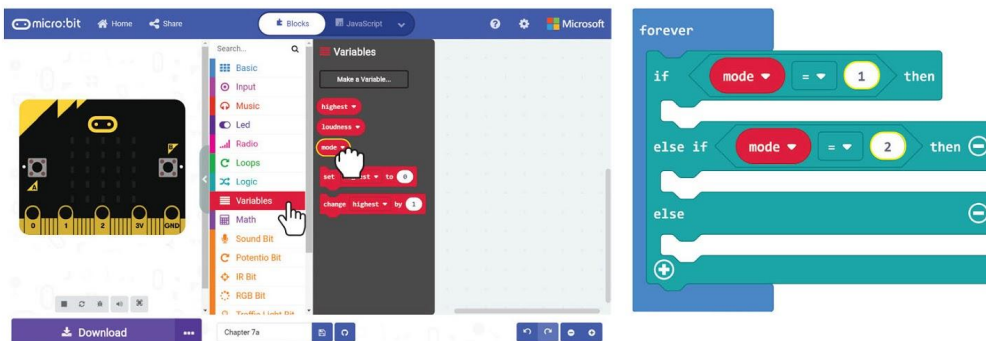
Langkah 10 Klik kategori [**Logic**] dan kemudian pilih blok [**if-then-else**]. Letakkan blok tersebut pada slot [**forever**]. Klik pada ikon tambah (+) untuk menambah blok pernyataan bersyarat *else-if*.



Langkah 11 Klik kategori [**Logic**] dan pilih blok perbandingan [**_ = _**]. Duplikasi blok tersebut dan letakkan blok-blok tersebut ke slot kondisi blok [**if-then-else**].

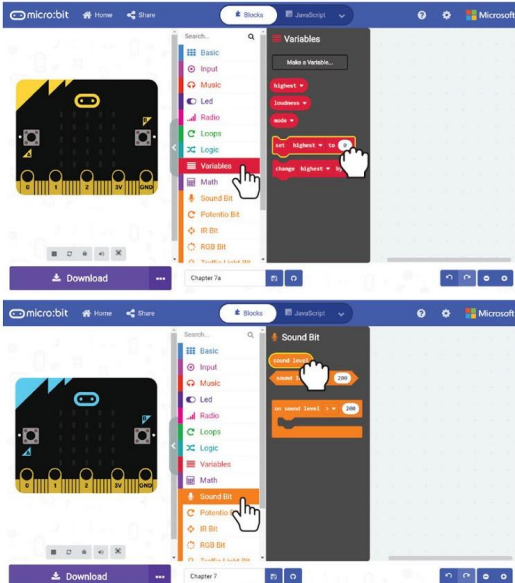


Langkah 12 Dapatkan [**mode**] daripada kategori [**Variable**]. Masukkan blok tersebut ke slot kiri blok perbandingan. Tetapkan slot yang lain-lain ke 1 dan 2 secara berurutan.





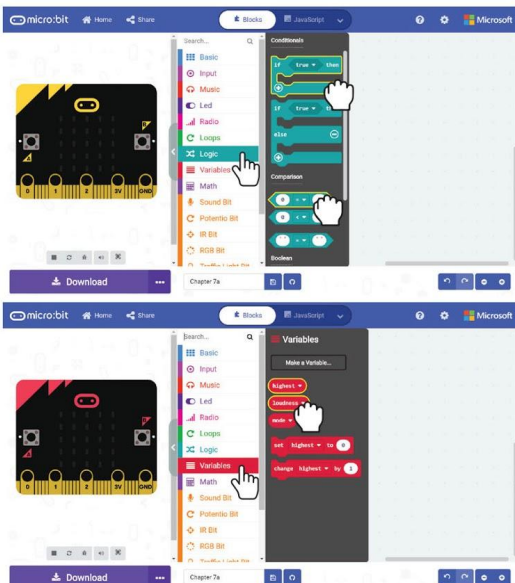
Langkah 13 Dapatkan blok [**set _ to _**] daripada kategori [**Variables**] dan letakkan blok tersebut pada slot [**if-then-else**]. Tetapkan pemboleh ubah kepada 'loudness' dan letakkan blok [**sound level**] daripada kategori [**Sound Bit**] pada slot nilai.



```

forever
  if mode = 1 then
    set loudness to sound level
  else if mode = 2 then
  else
  
```

Langkah 14 Klik kategori [**Logic**] dan pilih blok [**if-then**] dan blok perbandingan [**_ = _**]. Tukar simbol = kepada >. Masukkan [**loudness**] and [**highest**] daripada kategori [**Variables**] ke slot - slot blok perbandingan.

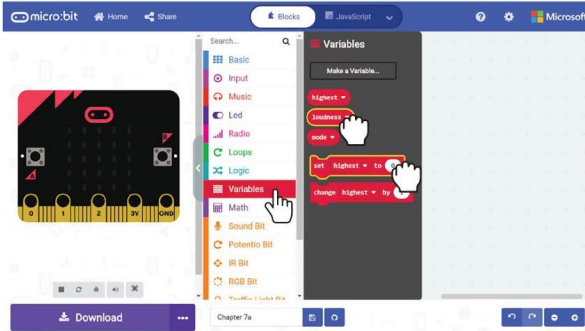


```

forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
  else if mode = 2 then
  else
  
```

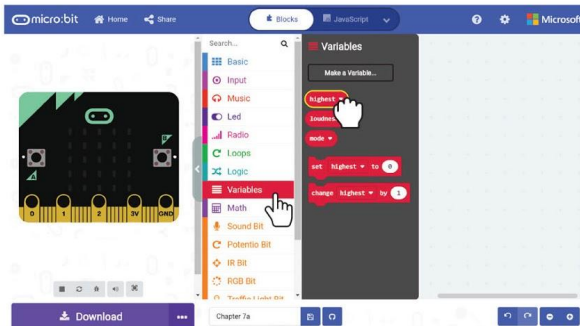
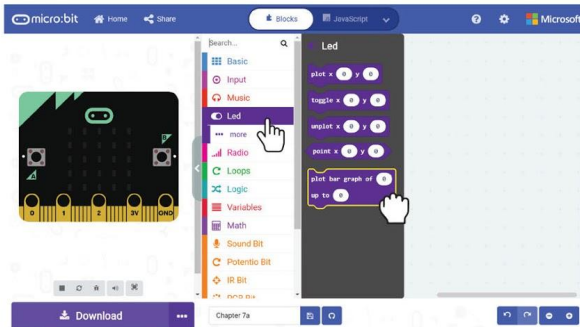
Bab 7 : Jom Kita Dengar Tepukan!

Langkah 15 Klik kategori [**Variables**] dan pilih blok [**set_to**]. Masukkan blok tersebut pada slot [**if-then**] dan [**loudness**] daripada kategori [**Variables**] pada slot nilai.



```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
    else if mode = 2 then
    else
```

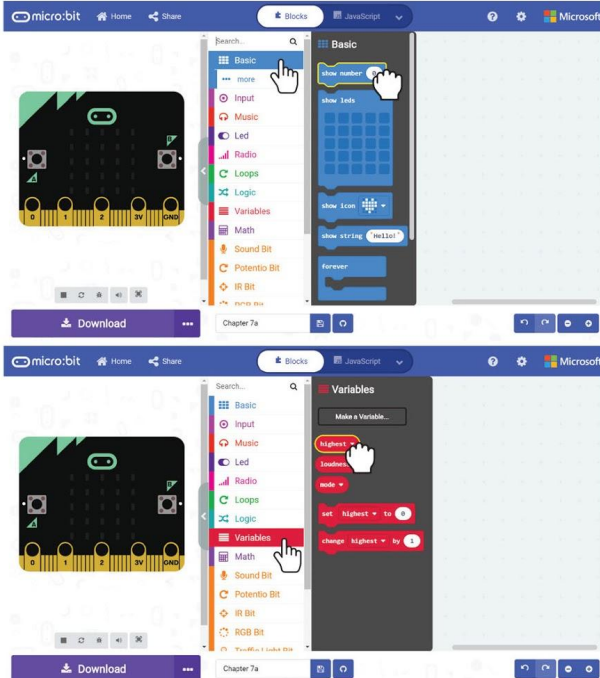
Langkah 16 Klik kategori [**Led**] dan pilih blok [**plot bar graph of _ up to _**]. Klik kategori [**Variables**] dan pilih blok [**loudness**]. Masukkan blok tersebut pada blok [**plot bar graph of _ up to _**] dan tukarkan nilai kepada **1023** .



```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
    else
```

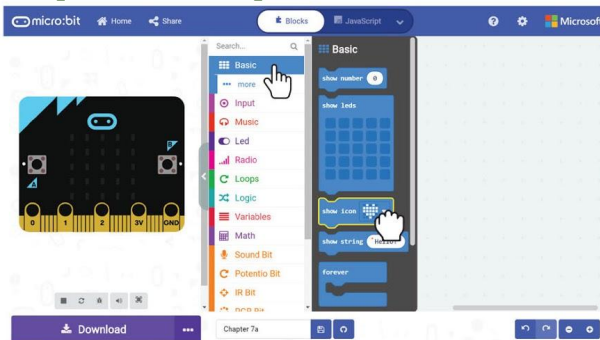


Langkah 17 Klik kategori [**Basic**] dan pilih blok [**show number**]. Masukkan blok tersebut pada slot kedua blok [**if-then-else**]. Dapatkan blok [**highest**] daripada kategori [**Variable**] dan masukkan blok tersebut pada slot nilai blok [**show number**].



```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
```

Langkah 18 Klik kategori [**Basic**] dan pilih blok [**show icon**]. Masukkan blok tersebut pada slot terakhir blok [**if-then-else**].



```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon
```

Bab 7 : Jom Kita Dengar Tepukan!

Ini ialah kod yang lengkap:

Set *mode* kepada 0 pada *On start*

```
on start
  set mode to 0
```

Apabila *button A* ditekan, tukar *mode* kepada 1 dan set pemboleh ubah 'highest' kepada 0

```
on button A pressed
  set mode to 1
  set highest to 0
```

Apabila IR Bit dicetuskan, tukar *mode* kepada 2

```
on IR sensor triggered
  set mode to 2
```

Selalu periksa *mode* semasa

```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon [heart icon]
```

Jika *mode* = 1 (*button A* ditekan), plot graf palang bagi bunyi paling kuat yang dapat dikesan dengan menyalakan LED paparan Matriks.
Semakin kuat bunyi, maka semakin banyak LED yang dinyalakan dan sebaliknya.

Jika *mode* = 2 (*IR sensor* dicetuskan), paparkan nilai semasa pemboleh ubah "highest" (aras bunyi yang dikesan)

Jika *mode* bukan 1 atau 2, maka paparkan ikon hati.

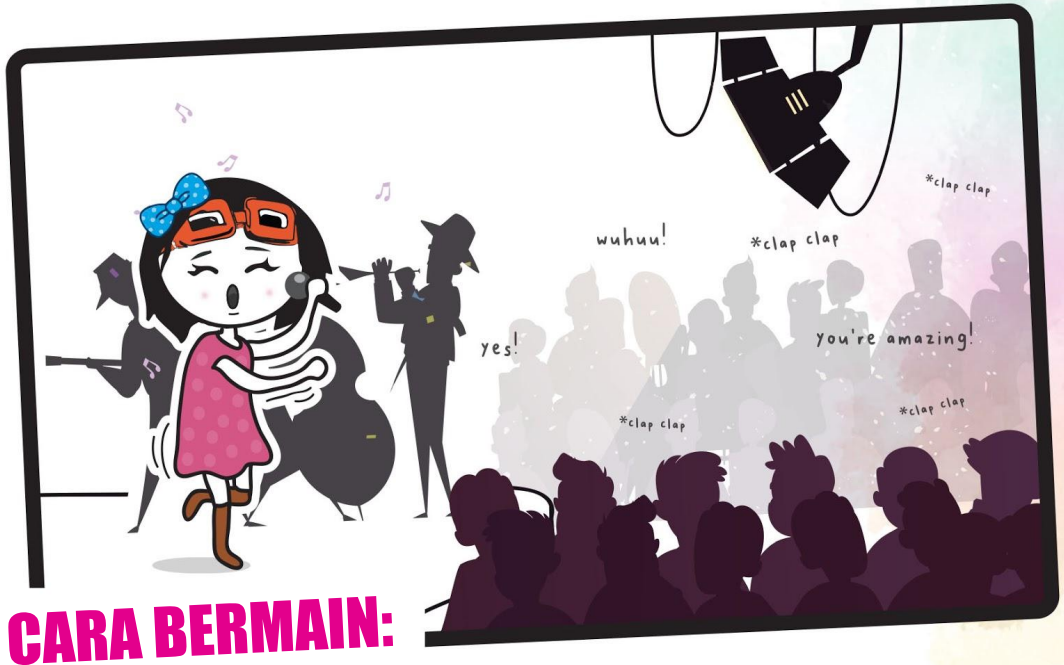
Jom kita cuba!



Langkah 19 Pindahkan kod ke dalam EDU:BIT dan *applause-o-meter* anda sudah boleh digunakan untuk *Talent Time Show*.

Jom Kita Main

Jom dengar tepukan tangan!



CARA BERMAIN:

“Contestants” akan diberi masa untuk membuat persediaan bagi persembahan pendek sama ada secara individu, pasangan atau pasukan. Anda boleh pilih untuk menyanyi, menari atau bercerita jenaka.

Apabila semua orang sudah bersedia, tunjukkanlah bakat anda mengikut giliran. Selepas setiap persembahan, “audience” akan balas dengan tepukan tangan – apabila “audience” semakin terhibur maka semakin kuat tepukan mereka.

Apabila bunyi tepukan berhenti, cetuskan IR Bit untuk menatal skor (rakaman tahap bunyi paling tinggi).

Jangan lupa tekan *Button A* untuk tetapkan semula skor tersebut sebelum persembahan seterusnya.

Individu, pasangan atau pasukan yang mendapat bunyi tepukan paling lantang dikira sebagai pemenang.

Berseronoklah!

TAHUKAH

ANDA??

Apabila kita mempunyai pelbagai *task* dalam satu program, kita boleh menggunakan pencetus peristiwa untuk bertukar dari satu *task* ke satu *task* yang lain. Bagi memastikan program berjalan dengan lancar, kita guna gelung *forever* untuk sentiasa memeriksa *mode* semasa dan kemudian melaksanakan kod bagi blok berkenaan.

Setkan *mode* kepada 0 (iaitu *mode* siap sedia) on start.

```
on start
  set mode to 0

on button A pressed
  set mode to 1
  set highest to 0

on IR sensor triggered
  set mode to 2
```

Blok pencetus peristiwa berubah dari satu *mode* kepada *mode* yang lain semasa *run time*.

Selalu memeriksa *mode* semasa dan kemudian laksanakan *task* (kod dalam blok) yang ditetapkan pada *mode* tersebut.

```
forever
  if mode = 1 then
    set loudness to sound level
    if loudness > highest then
      set highest to loudness
      plot bar graph of highest
      up to 1023
    else if mode = 2 then
      show number highest
    else
      show icon [grid icon]
```



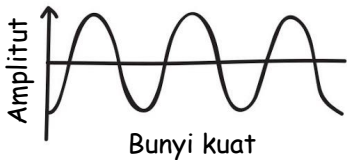
Jika anda mempunyai *mode* / *task* tambahan untuk disertakan, anda boleh sentiasa tambah lagi blok pencetus peristiwa seperti [on shake] dan [on sound level > _] ke dalam kod anda dan klik butang (+) pada blok [if-then-else] untuk tambah lebih banyak situasi.

FAKTA MENARIK!



Bunyi dihasilkan apabila suatu objek bergetar. Contohnya, dram yang dipukul. Getaran akan menyebabkan molekul udara (perantara) di sekitarnya untuk bergetar dan menghasilkan gelombang bunyi.

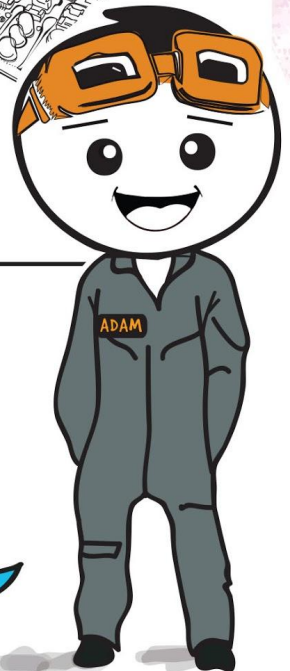
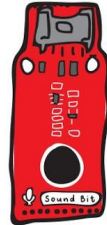
Pengesan bunyi adalah modul untuk mengesan intensiti / kekuatan gelombang bunyi (iaitu kelantangan bunyi) dan menukarkannya kepada isyarat elektrik.



Dalam erti kata lain, pengesan bunyi berfungsi seperti telinga kita yang menukar getaran dalam udara kepada isyarat elektrokimia yang mana akan diterjemahkan oleh otak kepada bunyi yang kita ketahui.



Bunyi dikesan



Di sini terdapat beberapa aplikasi lazim:

- Pengesan bunyi bagi penggera pencuri
- Lampu yang diaktifkan oleh bunyi
- Monitor pendengaran bayi

Pada pendapat anda, adakah pengesan bunyi kita dapat mengesan sebarang bunyi di luar angkasa? Mengapa atau mengapa tidak?



CABARAN APLIKASI

Programkan EDU:BIT supaya dapat berfungsi sebagai monitor bunyi bilik darjah. Nyalakan LED pada *Traffic Light Bit* untuk menunjukkan tahap bunyi.

| Tahap Bunyi | Julat Tahap Bunyi | Nyalakan <i>Traffic Light Bit</i> |
|--|-------------------|-----------------------------------|
| Terlalu bising; sila rendahkan suara anda. | () to 1023 | LED Merah |
| Sederhana bising; sila beri perhatian terhadap suara anda. | () to () | LED Kuning |
| Tahap bising yang boleh diterima. Bagus! | 0 to () | LED Hijau |

Tip untuk anda...

Tip #1: Anda perlu menentukan nilai ambang bagi setiap tahap bunyi terlebih dahulu.

Tip #2: Untuk pengawasan yang lebih stabil, dapatkan nilai purata bacaan tahap bunyi pada selang masa yang sama.



Terlalu senang? Cuba cabaran tahap lebih tinggi ini. Ubah suai kod anda supaya nilai ambang adalah relatif dengan nilai potentiometer.

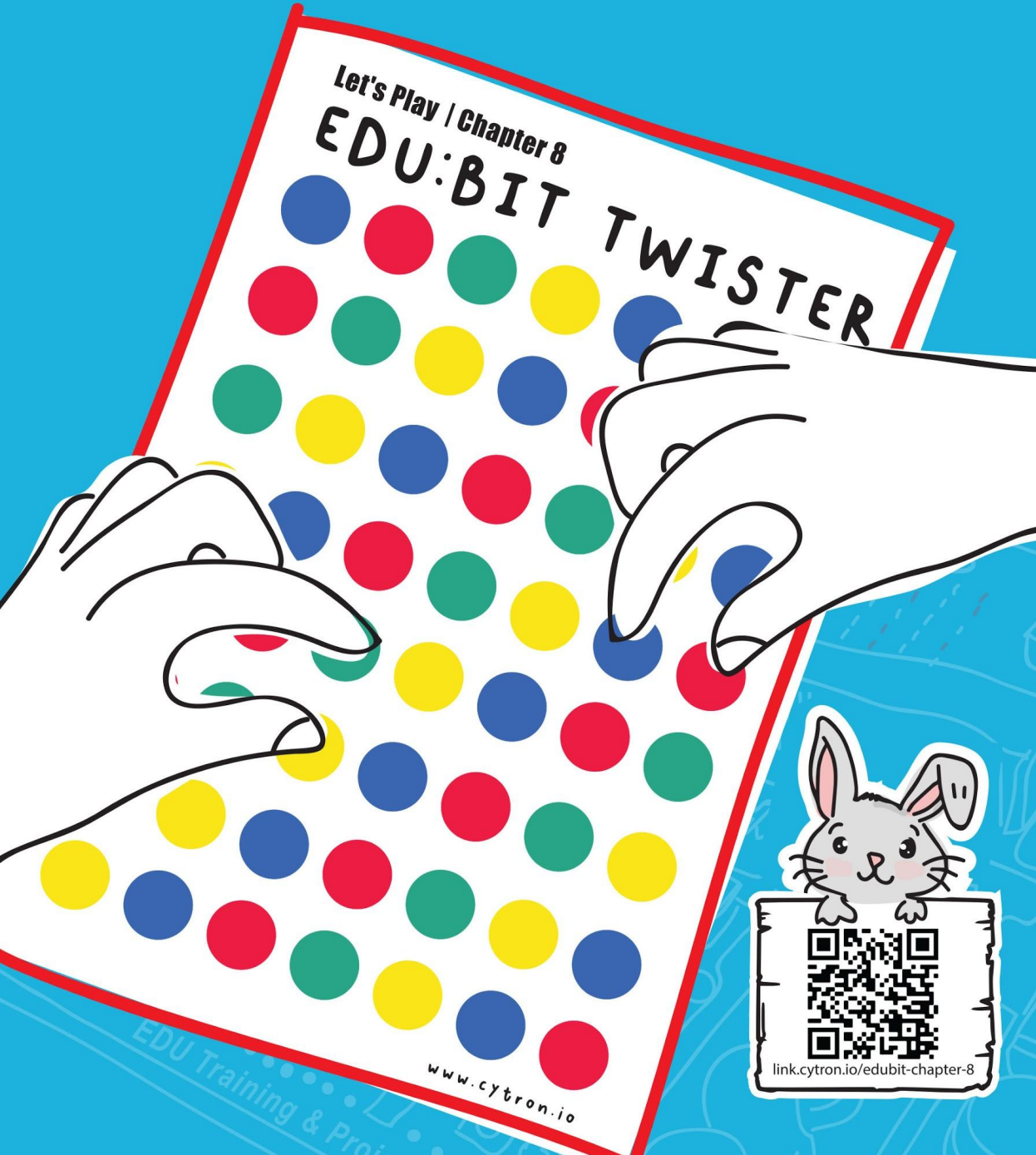


Jom Putar!

Motor DC

Let's Play | Chapter 8

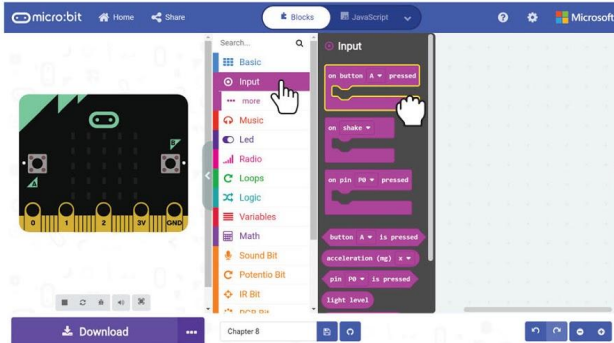
EDU:BIT TWISTER



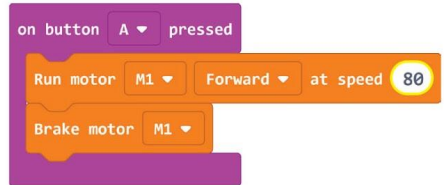
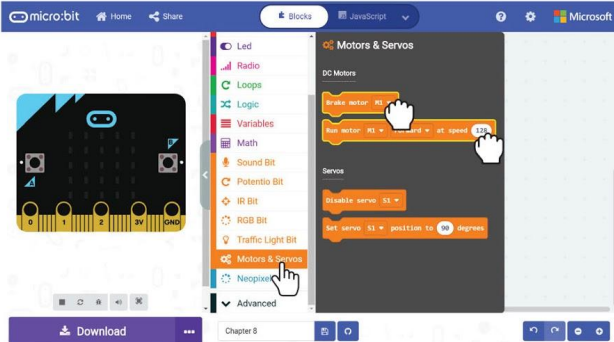
link.cytron.io/edubit-chapter-8

JOM KITA KOD!

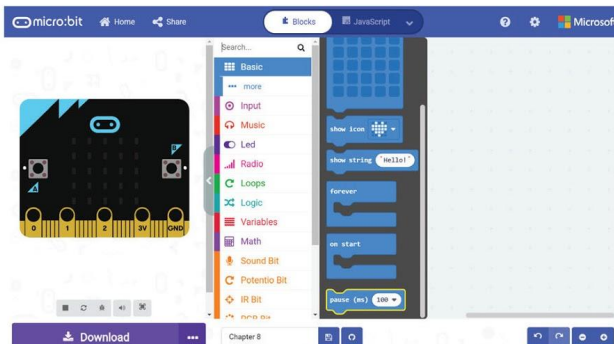
Langkah 1 Dalam *MakeCode Editor* anda, mulakan dengan projek baharu dan tambah *EDU:BIT extention* (anda boleh rujuk halaman 40). Klik kategori [**Input**] dan pilih blok [**on button_pressed**].



Langkah 2 Klik kategori [**Motors & Servos**] dan tambah blok [**Run motor_ at speed_**] serta blok [**Brake motor_**] ke dalam kod anda. Tukar nilai kelajuan kepada 80.

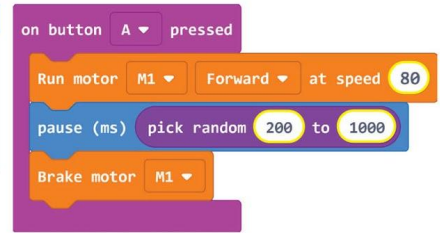


Langkah 3 Klik kategori [**Basic**] dan pilih blok [**pause_**]. Masukkan blok tersebut di antara blok [**Run motor_ at speed_**] dan blok [**Brake motor_**].

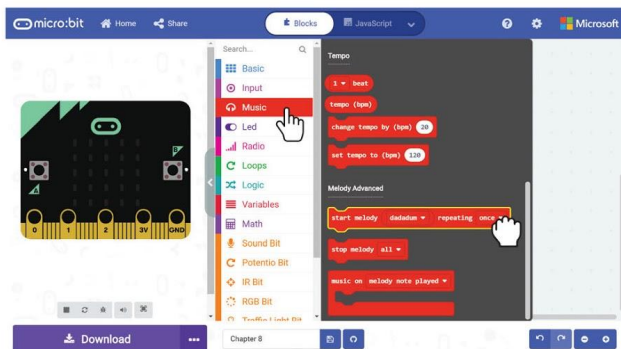




Langkah 4 Klik kategori [**Math**] dan pilih blok [**pick random_to_**]. Masukkan blok itu ke dalam blok [**pause**] dan tukar nilai masing-masing kepada 200 dan 1000.



Langkah 5 Klik kategori [**Music**] dan pilih blok [**start melody_repeating_**]. Tukar melodi kepada “ba ding” (atau pilih mana-mana melodi yang anda inginkan).

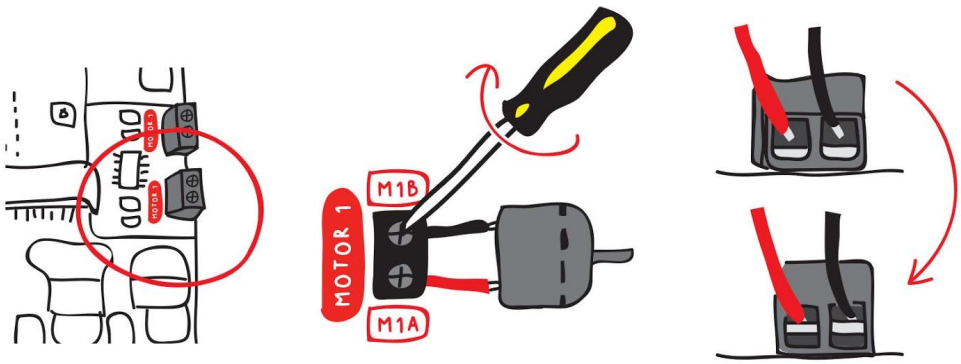


Langkah 6 Pindahkan kod yang telah siap ke dalam EDU:BIT.



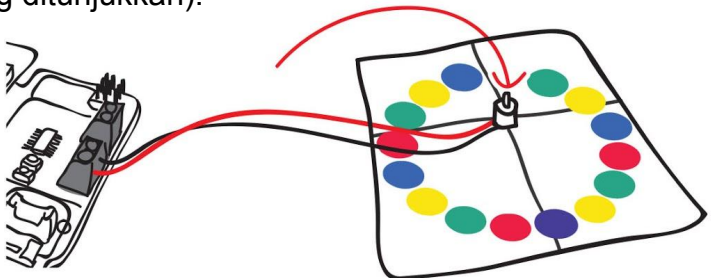
Kita boleh gunakan kod ini untuk mana-mana permainan yang memerlukan pemintal rawak. Biasanya, motor akan mula berputar apabila pemintal dicetus dan akan berhenti selepas suatu tempoh masa yang rawak.

Langkah 7 Sambungkan motor DC ke terminal MOTOR 1 – (i) masukkan wayar yang terdedah, kemudian (ii) ketatkan skru dengan pemutar skru yang disediakan untuk mengukuhkan sambungan.

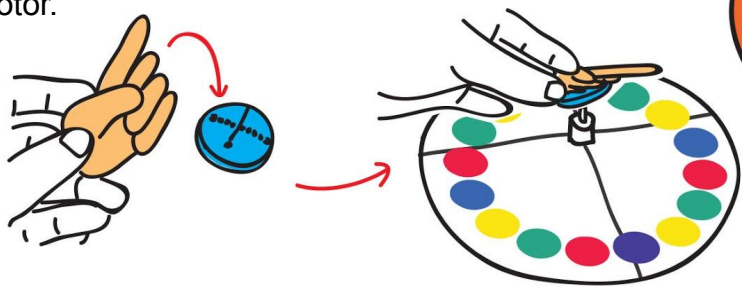


Tekan butang kuning (*Button A*) untuk menguji. Jika motor tidak berputar, periksa serta pastikan sambungan wayar pada terminal adalah kukuh dan EDU:BIT sudah dihidupkan.

Langkah 8 Gunakan pelekat tambahan seperti pita pelekat dua muka atau gam panas untuk lekatkan motor DC di tengah-tengah *Twister Wheel* (seperti yang ditunjukkan).



Langkah 9 Keluarkan jarum penunjuk dan lekatkannya pada disk plastik dengan menggunakan pelekat. Kemudian, pasang disk tersebut pada aci motor.



Twister Wheel, jarum penunjuk dan peta mainan ada disediakan dalam kotak.

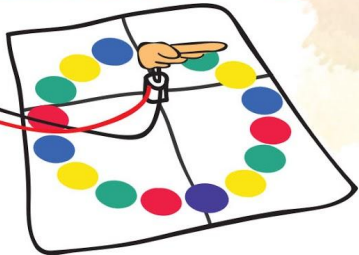
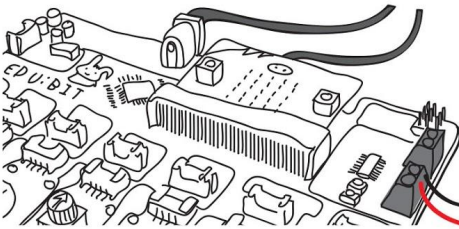


Jom Kita Main

Jom putar!

Persediaan Permainan

- Bentangkan peta mainan di atas meja. Dua orang pemain duduk secara bertentangan. Jika ada pemain tambahan, mereka boleh duduk di sisi yang kosong (maksimum 4 pemain).
- Pengadil akan duduk berdekatan dengan EDU:BIT *Twister Wheel*.



CARA BERMAIN:

Dalam permainan ini, pemain akan meletakkan jari mereka pada bulatan berwarna yang terdapat di atas peta mainan secara bergilir-gilir seperti yang diarahkan oleh pengadil.

Peranan pengadil adalah untuk menekan butang kuning (atau *Button A*) untuk memutarakan penunjuk dan kemudian sebutkan jari serta warna yang ditunjuk oleh jarum penunjuk. Contohnya, "jari telunjuk; merah".

Apabila sampai giliran anda, anda mesti mendengar arahan daripada pengadil, dan meletakkan jari yang disebut pada warna bulatan yang betul. Jika sudah ada jari pada warna bulatan yang disebut, anda mesti cuba untuk tukar ke bulatan lain yang sama warna.

Jika anda gagal berbuat demikian, anda akan dikeluarkan daripada permainan ini.

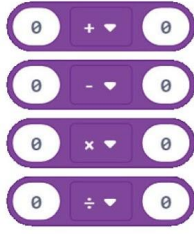
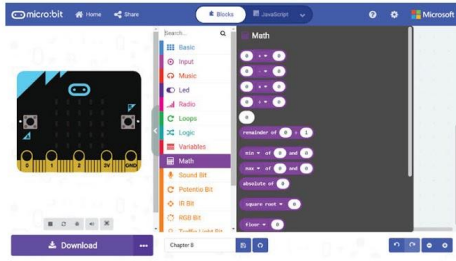
Pemain yang terakhir meninggalkan permainan adalah PEMENANG!



Terokai Blok-Blok Lain

Anda boleh gunakan blok dalam kategori [**Math**] untuk melaksanakan operasi aritmetik dengan pembolehubah anda.

#1 Guna blok yang berikut untuk tambah, tolak, darab dan bahagi.

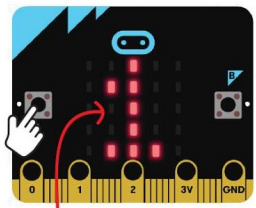


#2 Anda boleh menggunakan blok [**remainder of _ ÷ _**] untuk mencari baki jika operasi bahagi antara dua nombor mempunyai baki.

Contohnya:

```
on button A pressed
  show number remainder of 13 ÷ 2
```

$$\begin{array}{r} 6 \\ 2 \overline{)13} \\ \underline{12} \\ 1 \end{array}$$



#3 Anda juga boleh menggunakan blok [**remainder of _ ÷ _**] untuk mengenal pasti sama ada suatu nombor itu nombor ganjil atau nombor genap. Hanya bahagi dengan 2. Jika baki ialah "1" maka nombor itu ganjil; jika baki ialah "0" maka itu nombor genap. Mari kita cuba!

```
on start
  set Counter to 0

on button A pressed
  change Counter by 1
  show number Counter
  if remainder of Counter ÷ 2 = 0 then
    show string "even"
  else
    show string "odd"
```



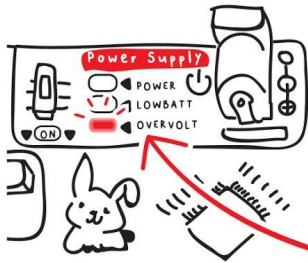
Tahukah anda bahawa EDU:BIT akan memaparkan "odd" (ganjil) apabila anda tekan *Button A* pada kali pertama (1 = nombor ganjil) dan akan papir "even" (genap) semasa anda tekan butang itu sekali lagi (2 = nombor genap)? Apakah yang akan dipaparkan apabila *Button A* ditekan 99 kali?

FAKTA MENARIK!



Motor arus terus atau lebih dikenali sebagai **motor DC** adalah suatu peranti elektrik berputar yang menukar tenaga elektrik kepada tenaga mekanikal.

Kita perlu mengenakan voltan input bagi membolehkan motor DC berputar. Kita dapat mengawal kelajuan putaran dengan mengubah voltan input tersebut. Semakin tinggi voltan input, maka semakin cepat motor itu berputar. Voltan yang disyorkan untuk motor DC pada kit EDU:BIT ialah 3.6V – 6V.

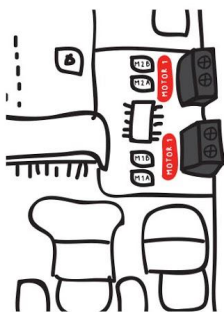
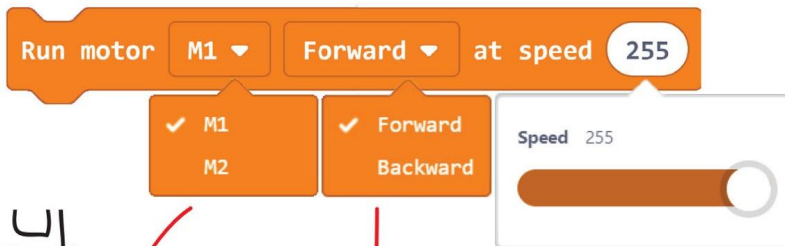


AMARAN!

Menggunakan voltan yang tinggi kepada motor (melebihi voltan yang disyorkan) untuk jangka masa yang panjang akan memendekkan kitaran hayat motor.



Indikator voltan lebihan



Terdapat dua terminal motor DC pada papan EDU:BIT. Pilih terminal yang betul.

arah putaran

Ini ialah nilai relatif dengan julat 0 hingga 255. Semakin tinggi nilai, semakin cepat motor itu berputar.

Tahukah anda bahawa EDU:BIT mempunyai litar ujian motor tersedia? Tekan butang putih (yang dilabel sebagai M1A, M1B, M2A dan M2B) untuk periksa sama ada sambungan anda betul dan motor dapat berfungsi dengan baik.



CABARAN APLIKASI

Programkan EDU:BIT supaya dapat berfungsi sebagai kipas kawalan-suara di mana kelajuannya dikawal oleh *Potentio Bit*.

| | |
|---|--|
| <i>On Start</i> | Paparkan ikon hati (atau mana-mana ikon yang anda pilih) Setkan pemboleh ubah <i>Mode</i> kepada 0 |
| <i>On sound level >_</i> (nilai ambang) | Tukar pembolehubah <i>Mode</i> sebanyak 1 |
| <i>Forever</i> | Setkan pemboleh ubah 'Speed' supaya sepadan dengan nilai <i>potentiometer</i> dari rendah 0 tinggi 1023 kepada rendah 0 tinggi 255. Selalu periksa <i>Mode</i> <ul style="list-style-type: none">• IF <i>Mode</i> ialah nombor genap, maka motor M1 brek• ELSE IF <i>Mode</i> ialah nombor ganjil, maka motor M1 berputar berdasarkan nilai pemboleh ubah "Speed" (seperti yang dipadankan daripada bacaan <i>potentiometer</i>) |



Tip untuk anda...

- Tip #1: Anda perlu menentukan tahap pencetus bunyi (nilai ambang) untuk aktifkan dan hentikan motor.
- Tip #2: Anda perlu mewujudkan dua pembolehubah: *Mode* dan *Speed*.
- Tip #3: Pasangkan sayap kipas pada aci motor dan kemudian jalankan program tersebut. Jika anda tidak dapat merasa tiupan angin semasa motor sedang berputar, maka anda perlu menukar arah putaran dalam kod anda.

Tendangan Penalti...Gol!!!

Motor Servo

Ready..

Get set..

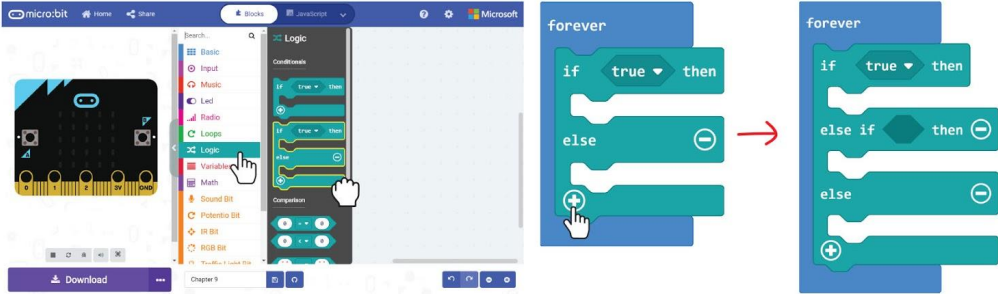
GO!!!!



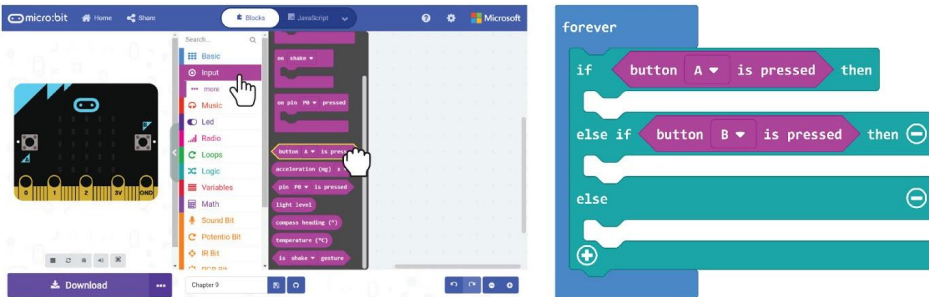
link.cytron.io/edubit-chapter-9

JOM KITA KOD!

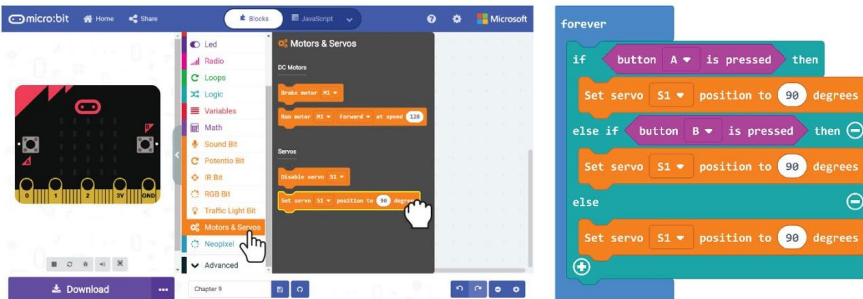
Langkah 1 Dalam *MakeCode Editor* anda, cipta projek baharu dan tambah *EDU:BIT extension* (anda boleh rujuk halaman 40). Klik kategori [**Logic**] dan pilih blok [**if-then-else**]. Masukkan blok itu ke dalam blok [**forever**]. Klik pada ikon tambah untuk tambah satu lagi syarat *else-if* pada blok tersebut.



Langkah 2 Klik kategori [**Input**] dan pilih blok [**button_is_pressed**]. Duplikat blok itu dan masukkan ke dalam slot bersyarat bagi blok [**if-then-else**]. Tukar blok kedua kepada 'button B'.



Langkah 3 Klik kategori [**Motors & Servos**] dan pilih blok [**Set servo_position_to_degrees**]. Duplikat blok itu dan masukkan ke dalam setiap slot bagi blok [**if-then-else**].



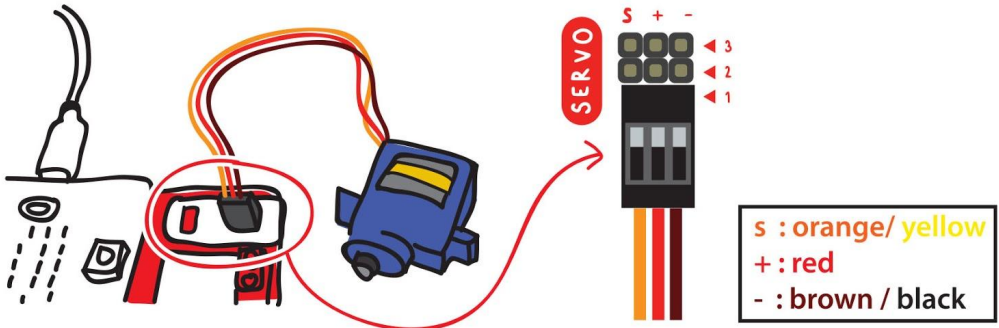


Bab 9 : Tendangan Penalti... Go!!!

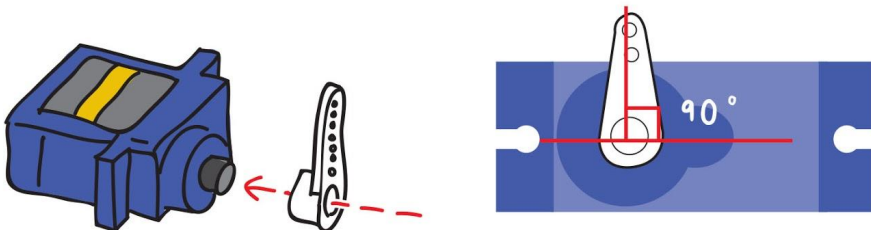
Langkah 4 Tukar nilai kedua-dua blok pertama dan kedua kepada 30 dan 150. Pindahkan kod anda ke dalam EDU:BIT.

```
forever
  if button A is pressed then
    Set servo S1 position to 30 degrees
  else if button B is pressed then
    Set servo S1 position to 150 degrees
  else
    Set servo S1 position to 90 degrees
```

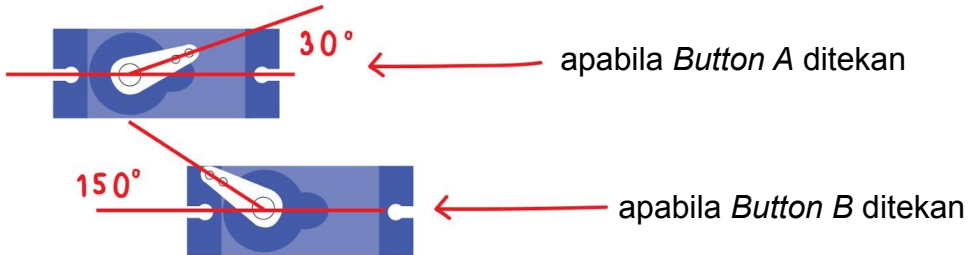
Langkah 5 Sambung wayar motor servo ke *Servo Port 1* pada EDU:BIT seperti yang ditunjukkan di bawah.



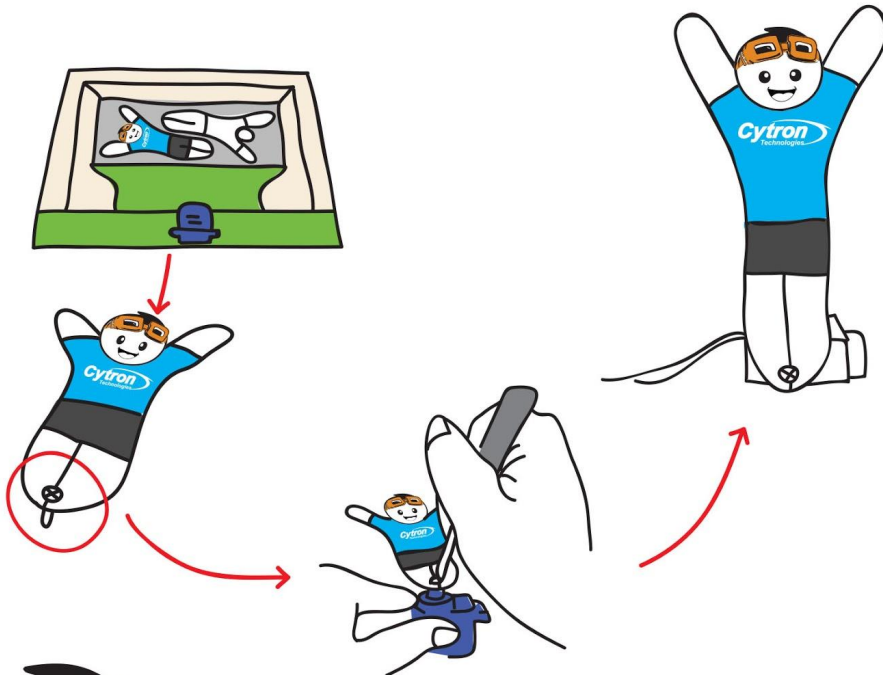
Langkah 6 Hidupkan EDU:BIT dan pasang lengan servo pada aci servo pada sudut 90 darjah seperti yang ditunjukkan di bawah.



Langkah 7 Tekan *Button A* dan kemudian *Button B* untuk menguji.



Langkah 8 Keluarkan penjaga gol daripada kad yang disediakan. Ketatkan penjaga gol pada lengan servo dengan menggunakan pemutar skru dan skru yang disediakan. Kukuhkan penjaga gol pada lengan servo dengan pelekat tambahan seperti pita pelekat dua muka atau gam panas.

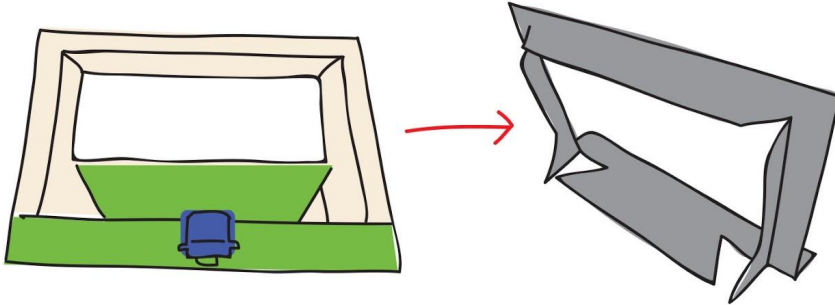


Jika anda berminat, anda boleh menggunakan bahan *pop-out* lain yang disediakan dan mereka cipta jersi untuk penjaga gol anda.

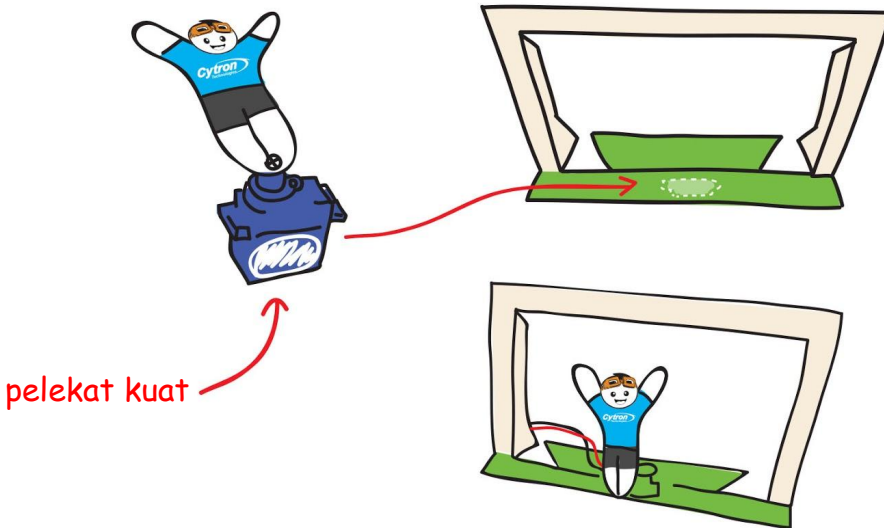


Bab 9 : Tendangan Penalti... Go!!!

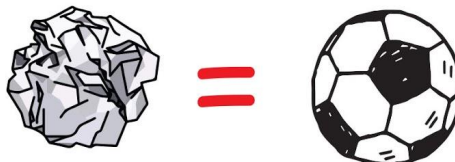
Langkah 9 Keluarkan tiang gol dan bina seperti yang ditunjukkan di bawah.



Langkah 10 Lekatkan servo motor seperti di bawah dengan menggunakan pekat yang kuat seperti pita pekat dua muka atau gam panas.



Langkah 11 Renyukkan sekeping kertas supaya menjadi bentuk sebiji “bola sepak” dan kita sudah bersedia untuk permainan yang menyenangkan iaitu tendangan penalti. Anda sudah bersedia?



Jom Kita Main

Tendangan Penalti... Gol!!!



CARA BERMAIN:

Dirikan tiang gol dan tandakan tempat penalti (kira-kira 1 meter daripada pintu gol, ubahsuai jarak untuk pemain-pemain yang lebih muda).

Pemain-pemain akan menjadi *Kicker* dan *Goalkeeper* secara bergilir-gilir.

Kicker akan memencet bola ke arah gol.

Goalkeeper akan cuba menghalang bola dengan bergerak ke kiri dengan menekan butang kuning (*Button A*) atau ke kanan dengan menekan butang biru (*Button B*).

Dalam satu pusingan, setiap pemain mempunyai 5 kali percubaan. Pemain yang mendapat jaringan gol yang paling banyak dikira sebagai pemenang.



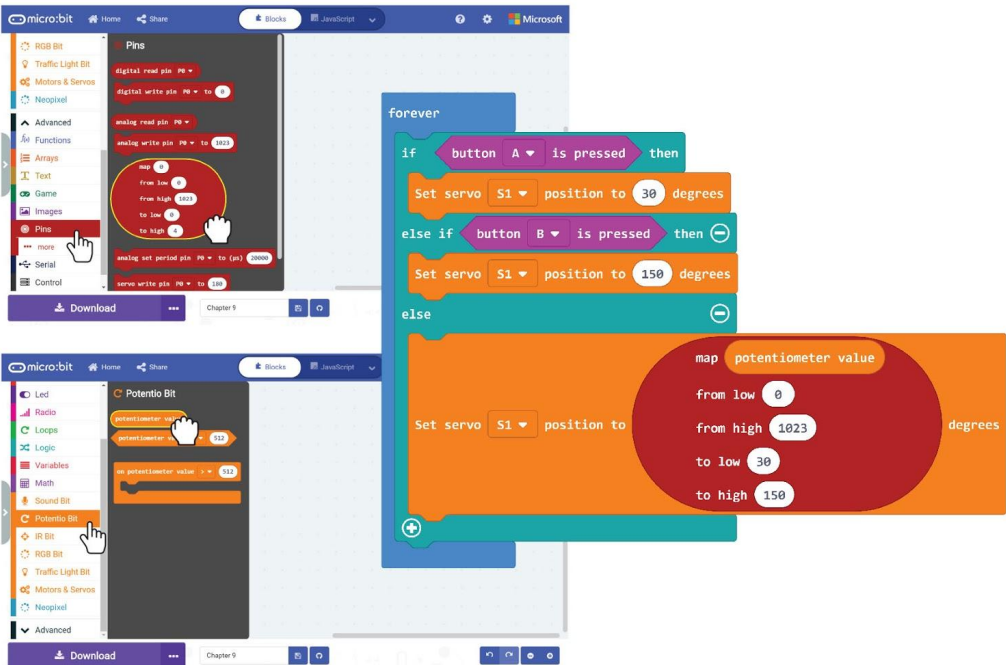
Tahukah anda? Tendangan penalti dilakukan untuk menentukan kemenangan pasukan bola sepak dalam sesuatu pertandingan apabila skor adalah sama di akhir permainan, dan masih lagi seri selepas tambahan masa. Dalam tendangan penalti, setiap pasukan diberi 5 kali sepakan. Pemain akan melakukan sepakan di tanda penalti dan dihalang oleh penjaga gol pasukan lawan sahaja. Kemenangan diberi kepada pasukan yang menjaringkan lebih banyak gol.



Dalam kod sebelum ini, kita telah mengerakkan *Goalkeeper* untuk menghayun ke kiri dan ke kanan ketika butang ditekan. Kita boleh mengubah suai kod tersebut untuk mengawal kedudukan *Goalkeeper* dengan menggunakan *Potentiometer*.

Langkah 12 Klik kategori [**Advanced**] dan pilih kategori [**Pins**]. Tambah blok [**map_from low_from high_to low_to high_**] dalam kod anda.

Langkah 13 Klik kategori [**Potentiometer**] dan pilih blok [**potentiometer value**]. Masukkan blok itu ke dalam blok [**map_from low_from high_to low_to high_**] dan ubah dua nilai yang terakhir masing-masing kepada 30 dan 150.



Langkah 14 Pindahkan kod ke dalam EDU:BIT anda. Sekarang anda boleh mengawal *Goalkeeper* dengan *Potentiometer*. Selamat mencuba!



Jika anda ingin praktis membina tendangan penalti anda sendiri, anda boleh mengubah suai kod kepada "practice mode" dengan menjadikan *Goalkeeper* hayun ke kiri dan ke kanan secara berterusan. Mari kita cuba!

FAKTA MENARIK!

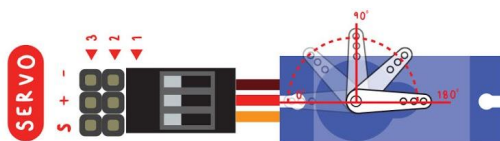
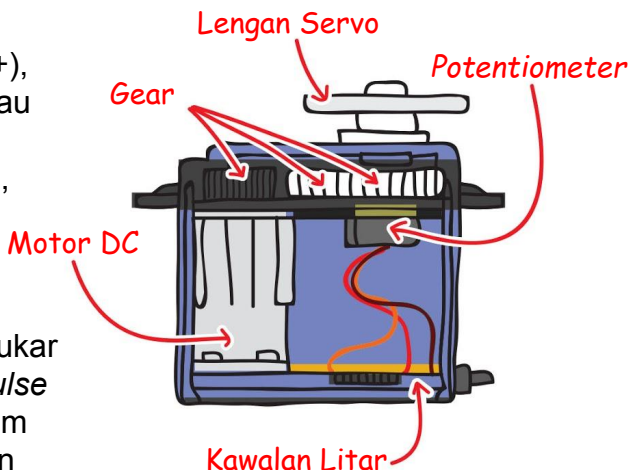


Motor servo dalam kit EDU:BIT juga dikenali sebagai servo RC (kawalan radio). Ia digunakan secara meluas untuk mengawal pergerakan kenderaan mainan RC dan robot kecil.

Motor servo menggunakan sistem tiga-wayar iaitu kuasa (+), bumi (-) dan kawalan atau isyarat (s). Ia mengandungi motor DC, gear, *potentiometer* (sensor kedudukan) dan litar kawalan.

Pengawal tersedia menukar arahan dalam bentuk *pulse* kepada kedudukan dalam darjah. Motor servo akan sentiasa berputar ke hadapan dan berhenti pada kedudukan yang sepadan dengan *pulse* yang diterima.

Berbeza dengan motor DC yang berputar secara berterusan, kita dapat mengawal putaran motor servo ke sudut yang dikehendaki iaitu antara 0 hingga 180 darjah.



Putaran 0-180°

Motor Servo VS Motor DC

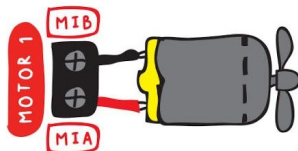
Putaran 360°



Belajar lagi!



youtu.be/okxooamdAP4



CABARAN APLIKASI

Programkan EDU:BIT supaya berfungsi sebagai metronom. Apabila dihidupkan, hayunkan penunjuk (yang dipasang pada aci servo) ke kiri dan ke kanan dengan tempo yang sama. Jadikan tempo supaya dikawal oleh *Potentio Bit*. Apabila butang kuning (*Button A*) ditekan, paparkan tempo semasa (contoh: 120 bpm).

Tip untuk anda ...

Tip #1: Anda perlu mewujudkan dua pemboleh ubah: *Tempo* dan *Delay*.

Tip #2: Julat tipikal metronom adalah dari 40 hingga 200 bpm.

Tip #3: Tempo 60 bpm (atau bit per minit) bermaksud penunjuk menghayun dari hujung ke satu lagi hujung sebanyak 60 kali dalam masa satu minit iaitu sekali dalam masa satu saat.

Tip #4: Semakin cepat tempo, semakin kurang *delay*.



"Metronom ialah sejenis peranti yang menghasilkan rentak boleh dengar atau bunyi lain dalam selang masa yang tetap dan boleh ditentukan oleh pengguna, biasanya dalam bit per minit (bpm). Ahli muzik menggunakan peranti ini untuk berlatih mengikut tempo tertentu. Metronom biasanya mengandungi *synchronized visual motion*."

- wikipedia -

**Mastermind, Bolehkah
Anda Pecahkan Kod?**

RGB Bit

Peleraai kod



Pencipta kod

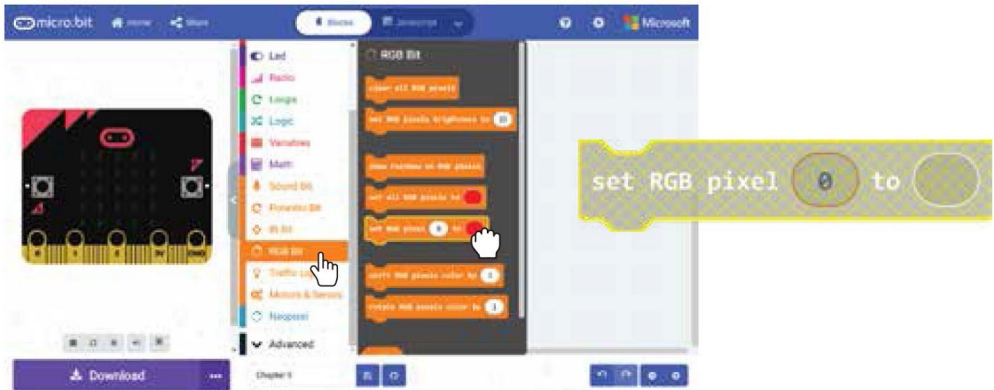


link.cytron.io/edubit-chapter-10

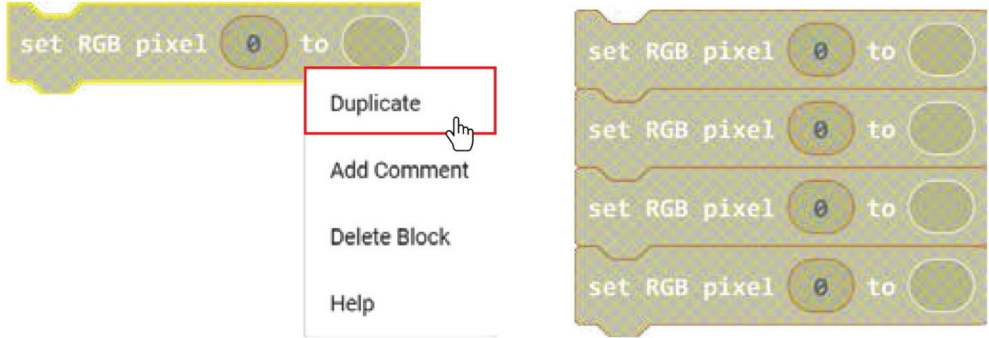


JOM KITA KOD!

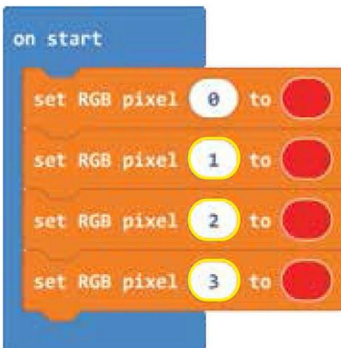
Langkah 1 Dalam *MakeCode Editor* anda, wujudkan satu projek baharu dan tambah EDU: BIT *extension*. Klik kategori [**RGB Bit**] dan kemudian pilih blok [**set RGB pixel_to_**]--



Langkah 2 Dalam editor, *right-click* pada blok [**set RGB pixel_to_**] dan kemudian klik *Duplicate*. Ulangi langkah ini sehingga anda mempunyai empat blok [**set RGB pixel_to_**].



Langkah 3 Letakkan blok-blok tersebut ke dalam blok [**on start**]. Tukar nombor *pixel* RGB dari 0 ke 1, 2 dan 3 bagi blok kedua, ketiga dan keempat.



Terdapat empat LED RGB pada *RGB Bit* and setiap satu diberi nombor tertentu (0-3). Gunakan nombor ini untuk mengatur cara setiap LED secara berasingan.

Langkah 4 Pindahkan kod ke dalam EDU:BIT anda.



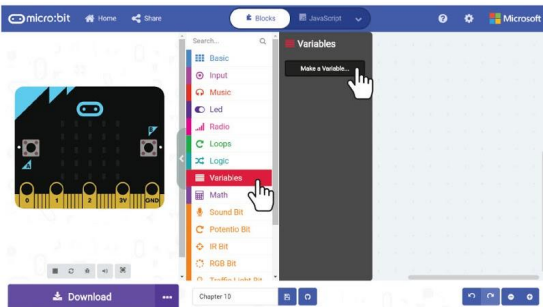
Apabila anda hidupkan papan EDU:BIT, keempat-empat LED pada RGB Bit akan menyala berdasarkan warna-warna yang telah ditetapkan.

Anda boleh menukar warna dengan mudah dengan cara klik pada blok dan pilih warna lain yang ada pada senarai.

Selamat mencuba!



Langkah 5 Tambahkan dua pemboleh ubah baharu dan namakannya sebagai – “Right Color and Position” dan “Right Color but Wrong Position”.



New variable name:

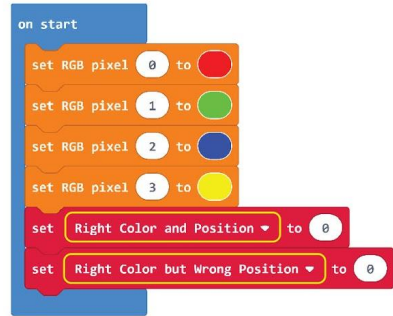
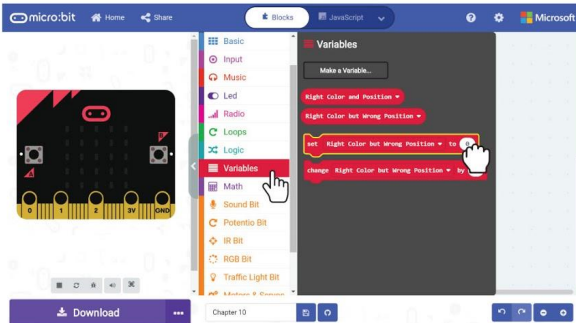
Ok ✓ Cancel ✕

New variable name:

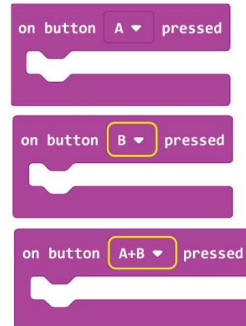
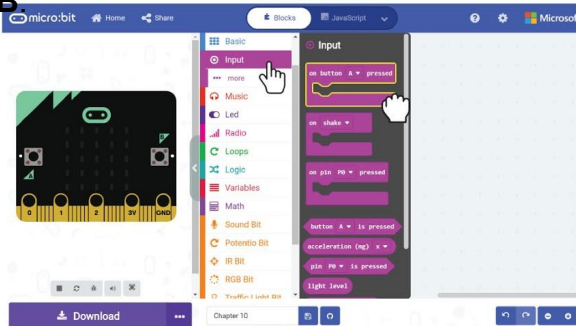
Ok ✓ Cancel ✕



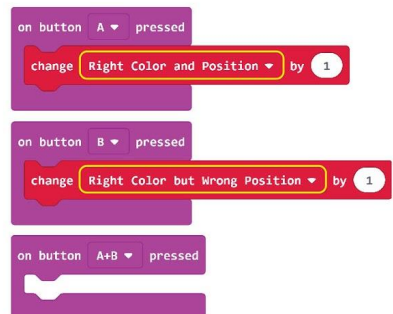
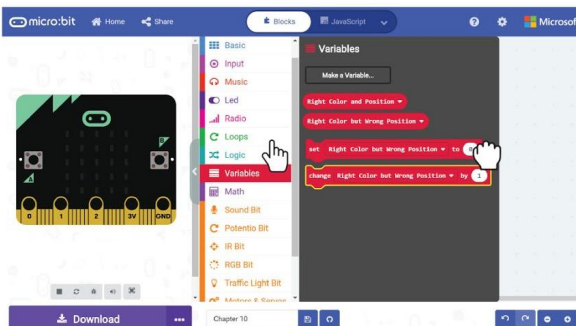
Langkah 6 Klik kategori [**Variables**] dan pilih blok [**set_to_**]. Duplikat blok tersebut dan masukkan ke dalam blok [**on start**]. Tetapkan salah satu pemboleh ubah sebagai “**Right Color and Position**” manakala pemboleh ubah yang satu lagi sebagai “**Right Color but Wrong Position**”.



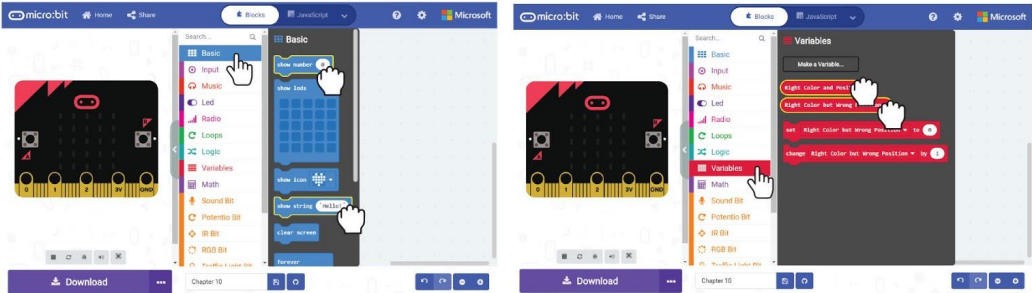
Langkah 7 Klik kategori [**Input**] dan pilih blok [**on button_pressed**]. Duplikat blok tersebut. Tukar blok kedua kepada **B** dan blok ketiga kepada **A+B**.



Langkah 8 Klik kategori [**Variables**] sekali lagi dan pilih blok [**change_by_**]. Duplikat dan masukkan blok tersebut ke dalam blok [**on button A pressed**] dan [**on button B pressed**]. Setkan satu pemboleh ubah sebagai “**Right Color and Position**”, manakala satu lagi sebagai “**Right Color but Wrong Position**”.



Langkah 9 Tambah blok [show number] dan [show string] daripada kategori [Basic], begitu juga dengan blok [Right Color and Position] dan [Right Color but Wrong Position] daripada kategori [Variables].



Langkah 10 Tukar “Hello!” dalam blok [show string] masing-masing kepada “A =” dan “B =”. Ini ialah kod yang telah siap.

```
on start
  set RGB pixel 0 to red
  set RGB pixel 1 to green
  set RGB pixel 2 to yellow
  set RGB pixel 3 to blue
  set Right Color and Position to 0
  set Right Color but Wrong Position to 0
```



```
on button A pressed
  change Right Color and Position by 1
  show number Right Color and Position

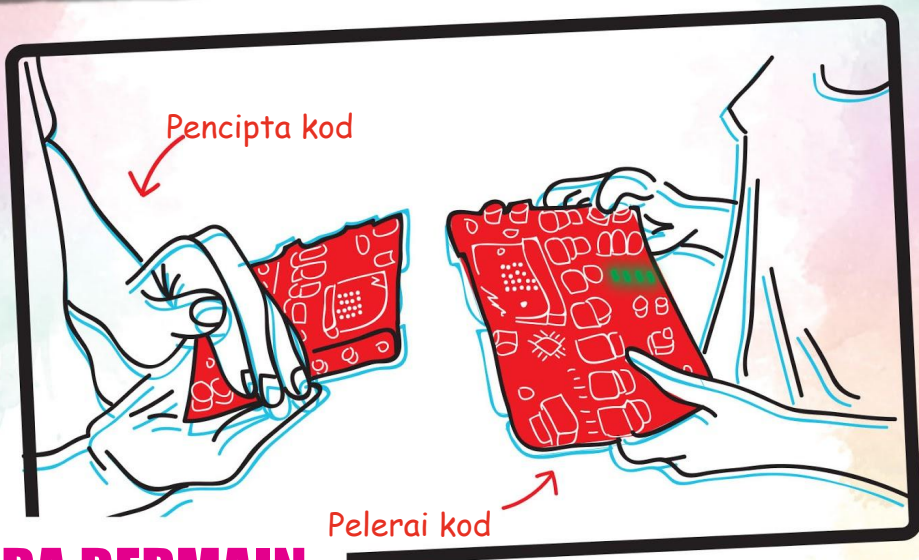
on button B pressed
  change Right Color but Wrong Position by 1
  show number Right Color but Wrong Position

on button A+B pressed
  show string "A ="
  show number Right Color and Position
  show string "B ="
  show number Right Color but Wrong Position
```

Langkah 11 Pindahkan kod ke dalam EDU:BIT anda. Anda sudah boleh bermain permainan *Mastermind* sekarang.

Jom Kita Main

Mastermind, bolehkah anda pecahkan kod?



CARA BERMAIN:

Pemain 1, Pencipta Kod, menetapkan kod rahsia dengan menyalakan empat LED dalam urutan secara rawak. Pilihan warna dihadkan hanya kepada (merah) dan (hijau) untuk permulaan. Pastikan pemain lain tidak dapat melihat corak warna papan anda.

Pemain 2, Peleraai Kod, akan cuba membuat tekaan kod rahsia. Pemain 2 menyalakan LED RGB pada EDU: BIT, dan kemudian menunjukkannya kepada Pencipta Kod.

Pencipta Kod memeriksa dan kemudian menekan butang kuning (Butang A) dan / atau butang biru (Butang B) pada Peleraai Kod EDU: BIT untuk menunjukkan berapa banyak LED yang dinyalakan dengan "warna dan kedudukan sebenar" dan "warna sebenar tetapi kedudukan yang salah".

Peleraai Kod kemudian boleh menekan butang kuning dan biru pada masa yang sama untuk 'membaca' maklum balas yang diberikan. Langkah 2 dan 3 diulang sehingga Peleraai Kod berjaya membuat tekaan urutan warna dengan betul (maksimum 10 percubaan setiap pusingan)

Peranan bertukar untuk permainan pusingan seterusnya.

Cara Untuk Menang:

Anda akan memenangi permainan sekiranya berjaya memecahkan kod (membuat tekaan urutan warna). Jika gagal, kemenangan diberikan kepada Pencipta Kod.



Terokai Blok-Blok Lain

1 Anda boleh menetapkan LED pada RGB Bit untuk memaparkan warna pelangi dengan menggantikan blok [**set RGB pixel to_**] dengan blok [**show rainbow on RGB pixels**].

```
on start
  show rainbow on RGB pixels
```

2 Anda boleh membuat kesan 'running light' dengan meletakkan blok [**rotate RGB pixels color by_**] dalam blok [**forever**]. Pastikan anda memasukkan blok [**pause**] untuk memperlahankan program supaya anda dapat melihat kesannya. Berikut adalah contoh kod:

```
on start
  set RGB pixel 0 to [red]
  set RGB pixel 1 to [green]
  set RGB pixel 2 to [blue]
  set RGB pixel 3 to [yellow]

forever
  rotate RGB pixels color by 1
  pause (ms) 500
```

3 Anda juga dapat mengubah piksel satu persatu dengan menggunakan blok [**shift RGB pixels color by_**] dalam blok [**forever**]. Anda perlu menambah blok [**pause**] untuk memperlahankan program sehingga anda dapat melihat kesannya, di mana piksel akan dimatikan satu persatu. Berikut adalah contoh kod:

```
on start
  set RGB pixel 0 to [red]
  set RGB pixel 1 to [green]
  set RGB pixel 2 to [blue]
  set RGB pixel 3 to [yellow]

forever
  shift RGB pixels color by 1
  pause (ms) 500
```

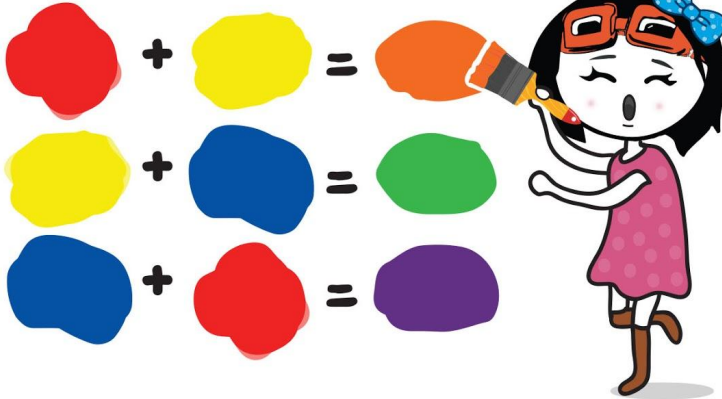


Anda boleh menukar arah kesan tersebut untuk #2 dan #3 di atas dengan mengubah tetapan dari nilai positif kepada nilai negatif.

FAKTA MENARIK!



Di dalam kelas Seni, anda mungkin mengetahui bahawa 3 warna utama adalah Merah, Kuning dan Biru. Apabila anda mencampurnya, anda akan mendapat hasil ini:



Namun, untuk peranti yang menggunakan lampu untuk memaparkan warna, seperti skrin tv dan komputer anda, model warna RGB digunakan.



Sebaliknya, dalam model ini tiga cahaya primer adalah Merah (R), Hijau (G) dan Biru (B); dan apabila digabungkan, cahaya Putih terhasil!

CABARAN APLIKASI

Programkan EDU:BIT menjadi alat latihan permainan memori.

Bagaimana ia berfungsi?

- Untuk bermula, condongkan EDU:BIT ke kiri untuk menyalakan LED pada RGB Bit mengikut corak warna rawak selama beberapa saat.
- Anda perlu memerhati dan menyebut susunan warna selepas LED padam.
- Untuk menyemak jawapan anda, tekan butang biru (Butang B) untuk menyalakan RGB LED mengikut corak warna yang sama sekali lagi.
- Jika jawapan betul, tekan butang kuning (Butang A) untuk kemaskini dan paparkan markah anda. 'Game Over' jika jawapan salah.
- Anda boleh melaraskan tahap kesukaran permainan dengan memutarakan tombol Potentio Bit untuk meningkatkan/mengurangkan jangkamasa LED kekal menyala.
- Pemain dengan markah tertinggi akan memenangi permainan.

Tips untuk anda:

Tip #1 Anda perlu menetapkan dua pembolehubah - Markah dan Corak

Tip #2 Anda perlu pra-tetap susunan warna (warna bagi setiap RGB LED) bagi setiap corak. Gunakan lebih banyak warna atau pra-tetap lebih banyak corak untuk membuatkan permainan lebih mencabar. Sebaliknya, anda mungkin mahu menghadkan warna/corak untuk pemain yang lebih muda.



Bab Bonus

Simon Says dengan LED Komunikasi Radio



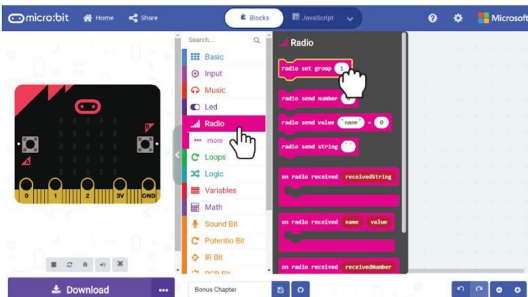
link.cytron.io/edubit-bonus-chapter



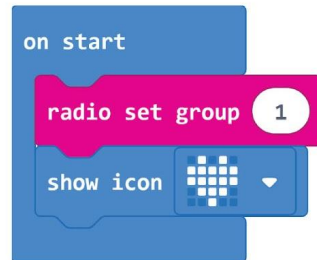
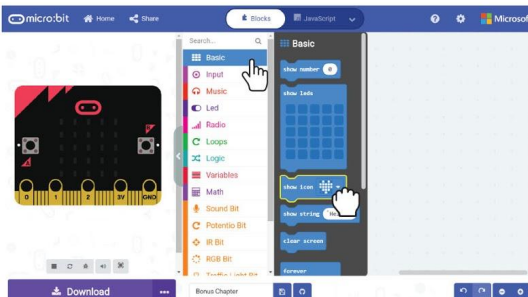
Untuk komunikasi berlaku, kita memerlukan sekurang-kurangnya dua pihak - iaitu penghantar dan penerima. Dalam permainan ini, kita memerlukan dua EDU:BIT untuk berkomunikasi antara satu sama lain dengan menghantar dan menerima isyarat siaran

JOM KITA KOD!

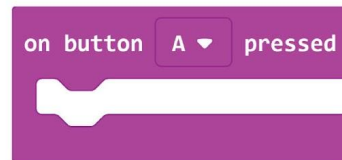
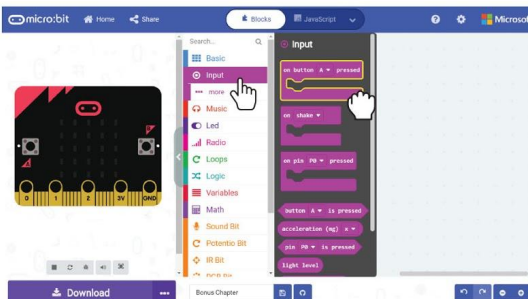
Langkah 1 Dalam Makecode Editor, bina satu projek baru dan tambah EDU:BIT extension. Klik kategori [**Radio**] dan pilih blok [**radio set group**]. Pasangkan blok tersebut ke blok [**on start**].



Langkah 2 Dalam Makecode Editor, klik kategori [**Basic**] dan tambah blok [**show icon**] ke dalam program.



Langkah 3 Klik kategori [**Input**] dan pilih blok [**on button_pressed**].

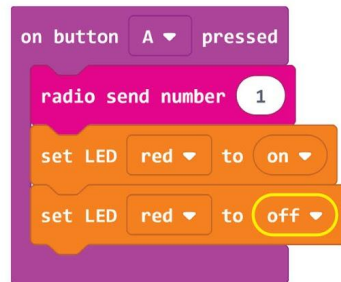
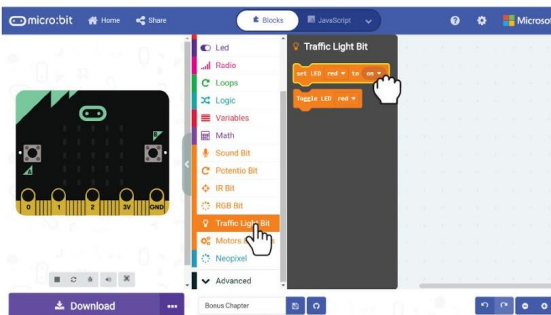




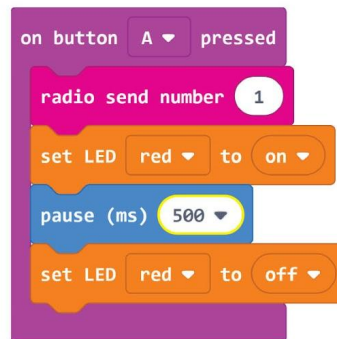
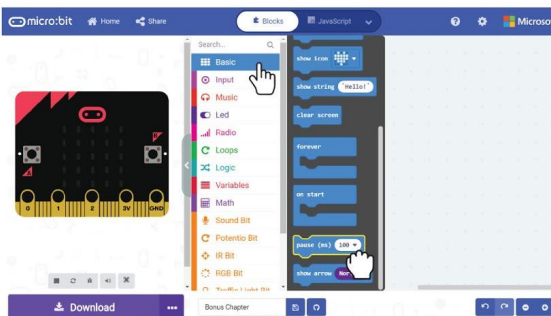
Langkah 4 Klik kategori [Radio] dan pilih blok [radio send number_]. Tukarkan nilai tersebut kepada 1.



Langkah 5 Klik kategori [Traffic Light Bit] dan pilih blok [set LED_to_]. Salin dan pasangkan kedua-dua blok tersebut ke blok [on button A pressed]. Ubah tetapan blok kedua kepada 'off'.

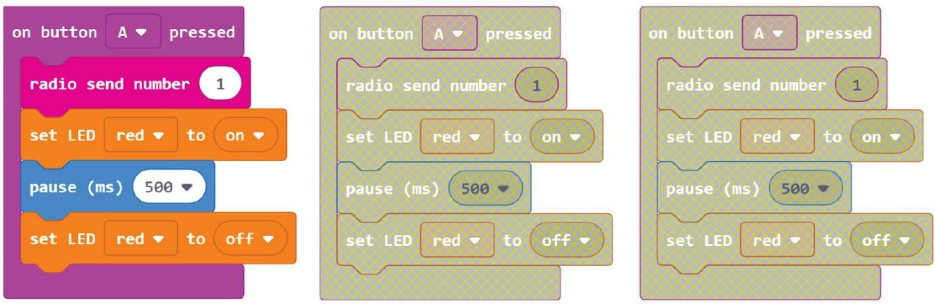


Langkah 6 Klik kategori [Basic] dan pilih [pause (ms)_]. Pasangkan blok tersebut di tengah-tengah [set LED_to_] dan ubah nilai kepada 500.

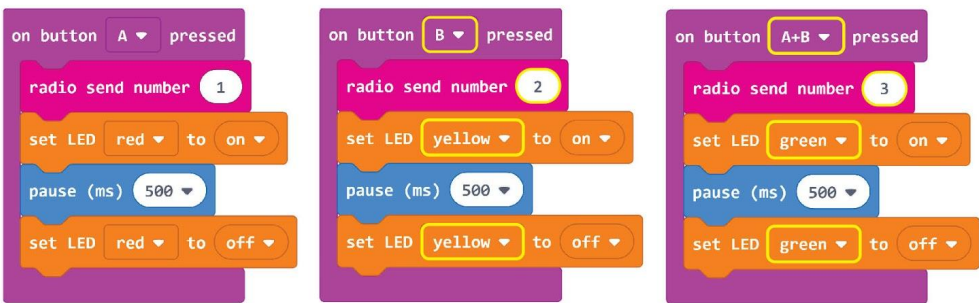


Bab Bonus : Simon Says dengan LED~

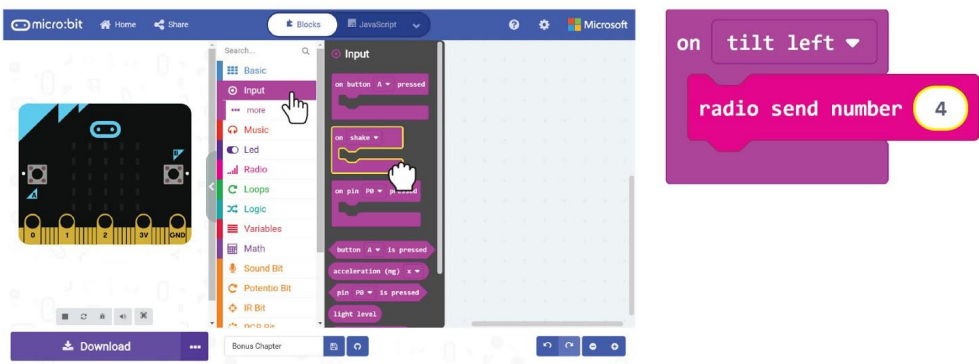
Langkah 7 Klik-kanan blok [on button A pressed] dan kemudian pilih 'Duplicate'. Ulang semula untuk dapatkan tiga set blok yang sama.



Langkah 8 Ubah tetapan untuk butang, nilai nombor dan warna LED bagi kod blok yang disalin seperti berikut:



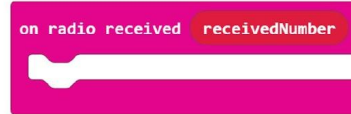
Langkah 9 Klik kategori [Input] dan pilih blok [on_]. Ubah tetapan kepada "tilt left". Klik kategori [Radio] dan pilih blok [radio send number_]. Ubah nilai kepada 4.



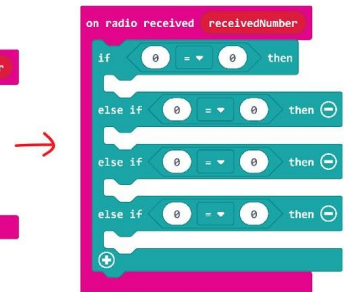
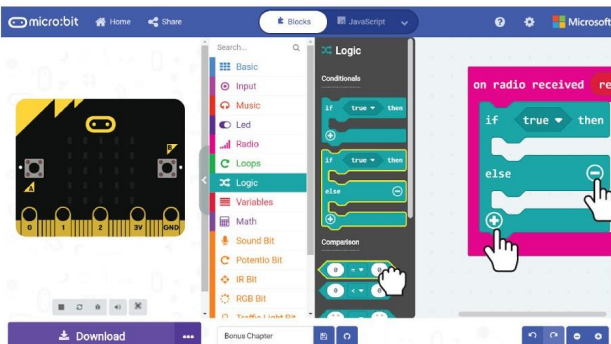


Bab Bonus : Simon Says dengan LED~

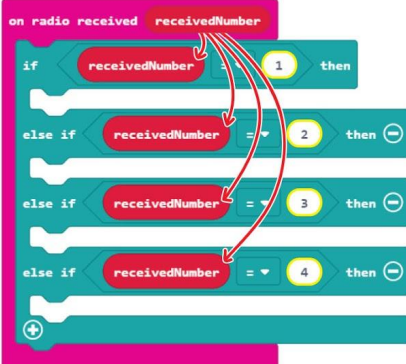
Langkah 10 Klik kategori [Radio] dan pilih blok [on radio received receivedNumber].



Langkah 11 Klik kategori [Logic] dan pilih blok [if-then-else]. Klik tombol (+) untuk tambah tiga syarat [else if] dan klik tombol (-) untuk membuang syarat [else]. Letakkan blok perbandingan [Logic]:[_ = _] untuk setiap slot "if" dan "else if".

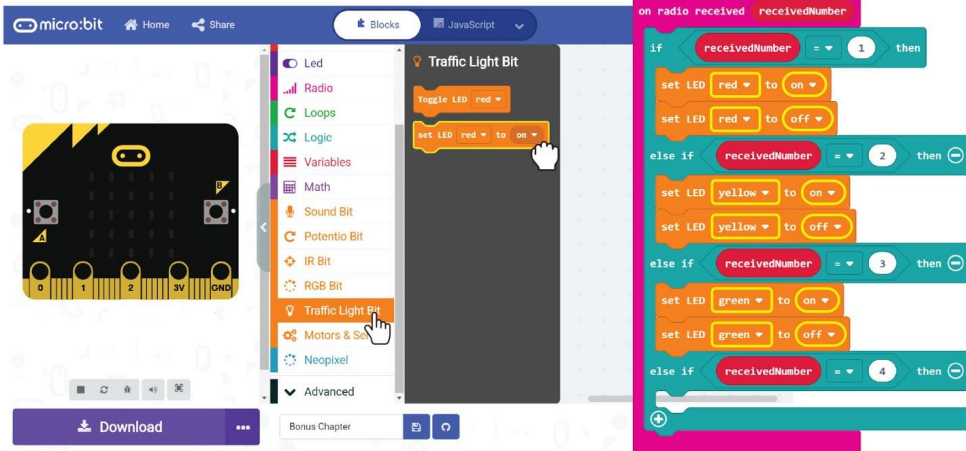


Langkah 12 Klik dan seret pembolehubah 'receivedNumber' ke dalam blok perbandingan seperti ditunjukkan dan ubah nilai masing-masing kepada 1, 2, 3 dan 4.

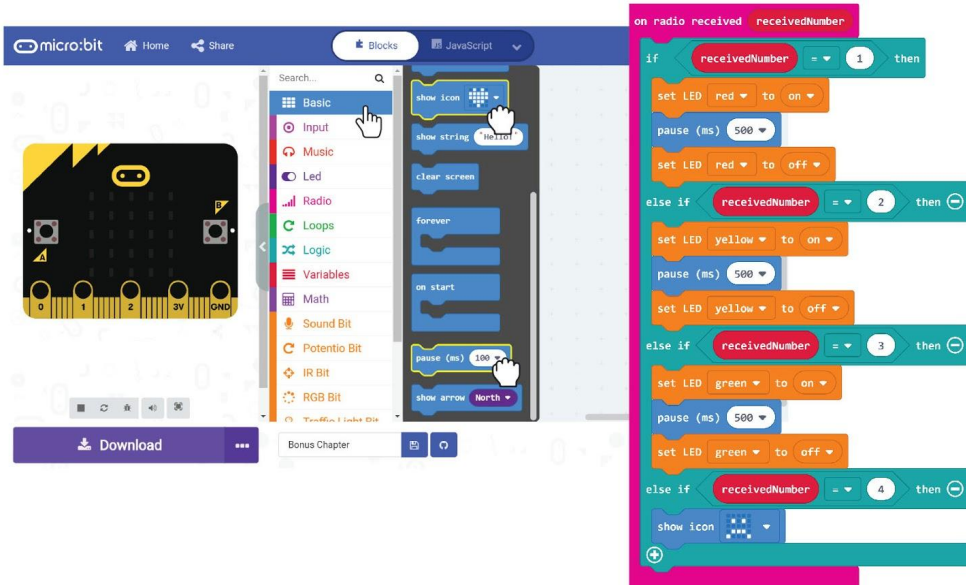


Bab Bonus : Simon Says dengan LED~

Langkah 13 Klik kategori [**Traffic Light Bit**] dan pilih blok [**set LED _ to _**]. Salin blok dan selitkannya ke tiga slot pertama. Ubah warna dan tetapan *on/off* seperti yang ditunjukkan.



Langkah 14 Klik kategori [**Basic**] dan pilih [**pause (ms)_**]. Salin dan selitkan blok tersebut diantara blok [**set LED _ to _**] dan ubah nilai kepada 500. Tambahkan blok [**Basic**] : [**show icon**] kepada slot terakhir "else if" dan ubah ikon tersebut kepada "sad face".

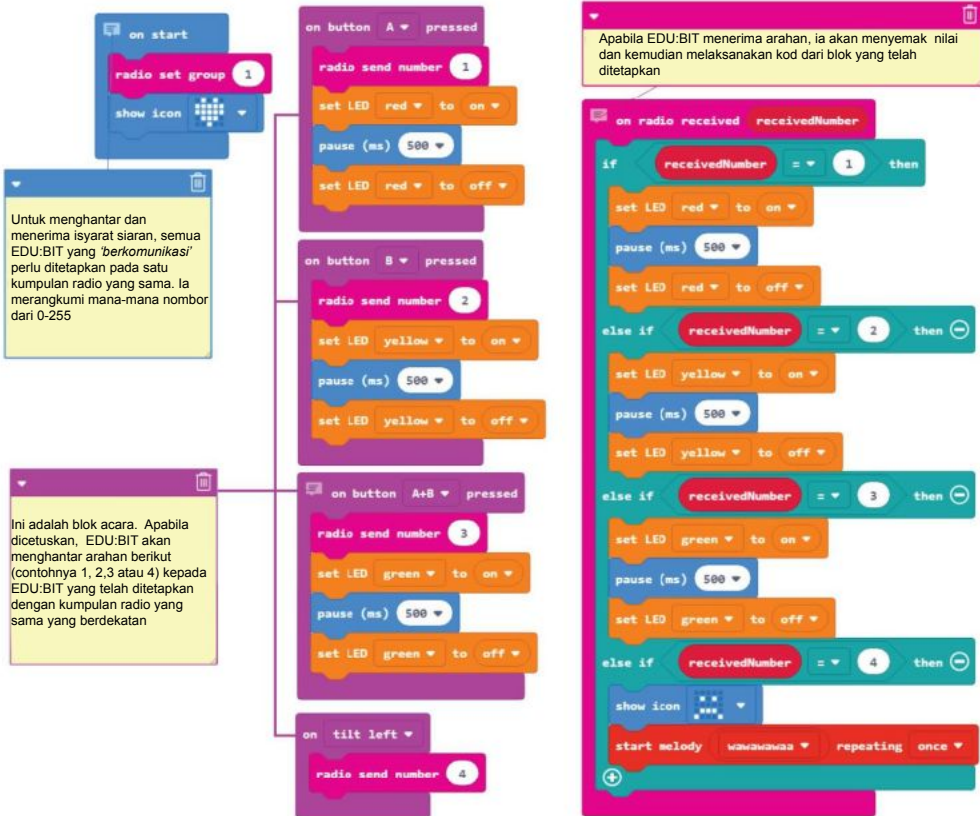




Bab Bonus : Simon Says dengan LED~

Langkah 15 Klik kategori [**Music**] dan tambah [**start melody _repeating_**] untuk melengkapkan kod. Ubah melodi tersebut kepada “wawawaaa” (atau mana-mana melodi pilihan anda).

Kod lengkap di sini.



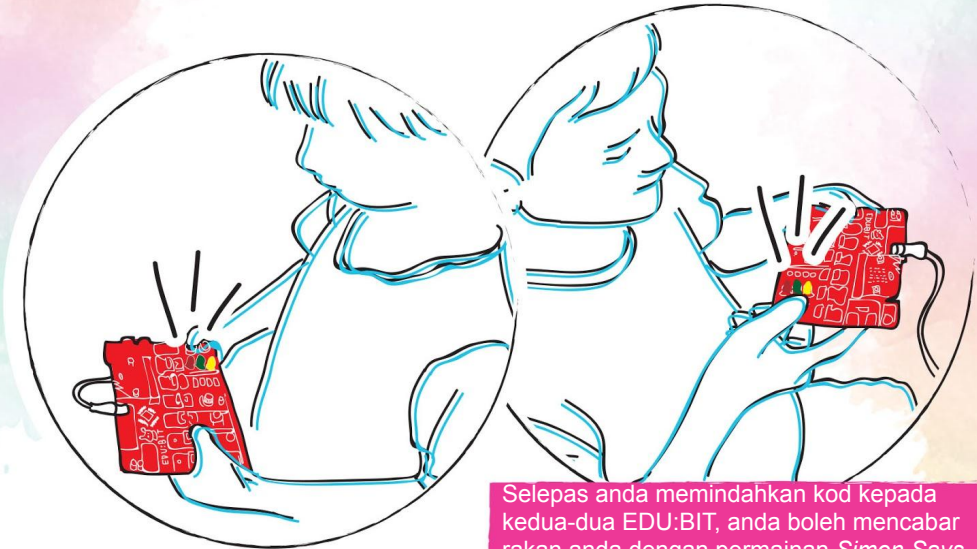
Langkah 16 Pindahkan kod yang telah siap kepada EDU:BIT anda dan rakan anda.



Apabila kedua-dua EDU:BIT dinyalakan, anda boleh menghantar isyarat radio untuk menyalakan LED EDU:BIT rakan anda, dan sebaliknya, rakan anda juga boleh menekan butang pada EDU:BIT kepunyaan untuk menyalakan LED papan anda.

Jom Kita Main

Simon Says dengan LED



Selepas anda memindahkan kod kepada kedua-dua EDU:BIT, anda boleh mencabar rakan anda dengan permainan *Simon Says* versi interaktif.

CARA BERMAIN:

Kedua-dua pemain bergilir-gilir menjadi “*Simon*”. Tekan butang untuk menyalakan LED merah, kuning, dan hijau apabila giliran anda.

Untuk memulakan permainan, Pemain 1 menyalakan SATU LED pada ‘Traffic Light Bit’.

Pemain 2 memerhati dan kemudian menyalakan LED yang sama diikuti dengan LED yang lain.

Permainan diteruskan dengan setiap pemain bergilir-gilir mengulangi susunan terkini, dan menyalakan LED yang lain untuk menambah susunan tersebut.

Jika mana-mana pemain menyalakan LED yang salah (atau susunan yang salah), pihak lawan akan mencondongkan EDU:BIT rakannya untuk menghentikan permainan.

Pemain yang kalah kemudian perlu set semula EDU:BIT rakannya untuk memulakan permainan baru.

Permainan ini bermula dengan mudah tetapi akan berubah kepada lebih panjang dan kompleks setiap giliran. Untuk memenangi permainan, anda perlu memerhati susunan dengan cermat. Permainan ini baik untuk melatih daya ingatan.





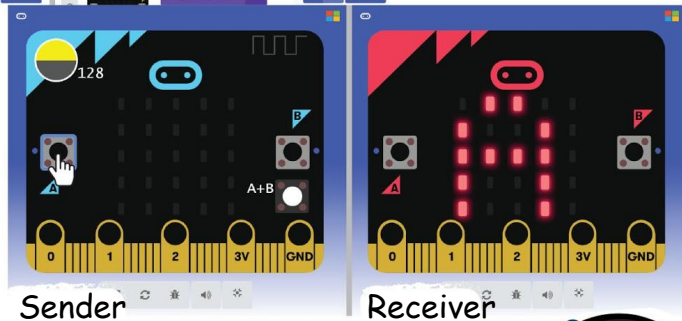
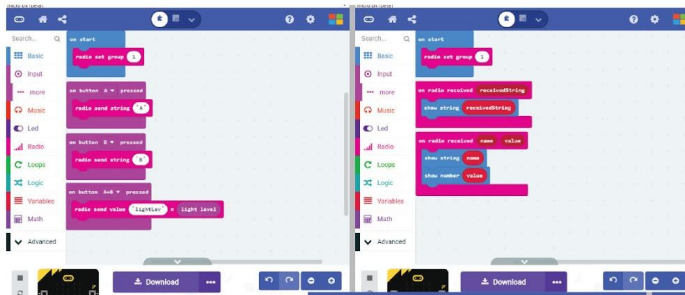
Terokai Blok-Blok Lain

#1 Selain menghantar nombor, anda juga boleh menghantar pesanan teks menggunakan blok [**radio send string** " _ "]. Anda perlu menggunakan blok [**on radio received receivedString**] untuk menerima jalinan isyarat siaran. Jalinan maksimum ialah 19 huruf.

#2 Gunakan blok [**radio send value** " _ " = _] dan blok [**on radio received name value**] jika anda ingin menghantar atau menerima teks dan nombor serentak. Jalinan maksimum ialah 8 huruf.



Jika anda tidak mempunyai akses kepada lebih dari satu EDU:BIT, anda masih boleh menguji fungsi radio komunikasi. Sila lawati makecode.com/multi untuk menulis kod untuk "sender" dan "receiver". Anda boleh melihat hasilnya dalam tettingkap simulasi.



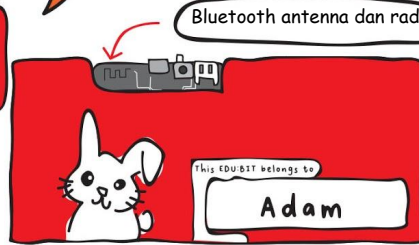
Apabila anda menekan butang A pada "sender" micro:bit, "receiver" micro:bit akan menerima isyarat radio dan memaparkan jalinan yang diterima contohnya, A. Apakah yang berlaku sekiranya anda menekan butang A+B?



FAKTA MENARIK!



Lihat! Saya boleh ingat susunannya. Giliran kamu sekarang, Adam.



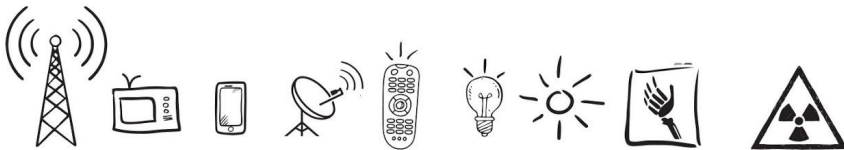
Jika anda terbalikkan EDU:BIT, anda dapat melihat radio pasang siap dan antenna Bluetooth disini



Okay, sekarang giliran kamu semula. Adakah kamu mampu memenanginya.

Antenna memancarkan isyarat dalam bentuk gelombang radio elektromagnetik, yang kebanyakannya digunakan untuk siaran televisyen dan radio serta satelit.

Spektrum elektromagnetik



NOTA!

Bagi membolehkan EDU:BIT anda untuk menghantar dan menerima isyarat siaran radio dari EDU:BIT lain, anda perlu menetapkan semuanya kepada kumpulan radio yang sama



CABARAN APLIKASI

Sediakan satu sistem rangkaian maklum balas untuk kelas anda.

Bagaimana ia berfungsi?

Setkan setiap EDU:BIT dalam kelas dengan kumpulan radio yang sama.

EDU:BIT guru disetkan kepada skrol teks apabila menerima isyarat “string” dan menyalakan LED pada Traffic Light Bit apabila menerima isyarat radio “number”, dimana

| Nombor diterima | LED yang menyala | Maksud |
|-----------------|--|-----------------------|
| 1 | MERAH  | A/ Tidak/ Salah |
| 2 | KUNING  | B/ Mungkin/ Tak pasti |
| 3 | HIJAU  | C/ Ya/ Benar |

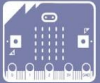
EDU:BIT pelajar disetkan untuk menghantar jalinan dengan nama pelajar dan kemudian menghantar satu nombor (1 atau 2 atau 3) untuk menyalakan LED pada EDU:BIT Guru apabila dicetuskan.

- Tekan butang A untuk menghantar nombor 1.
- Tekan butang B untuk menghantar nombor 2.
- Tekan butang A+B untuk menghantar nombor 3.

Tips untuk anda. Berikan nama samaran yang pendek (dengan dua atau tiga huruf) kepada setiap pelajar untuk mengurangkan masa paparan teks tersebut.



Saya telah menguasai...



- Memprogram EDU:BIT untuk memaparkan teks dan animasi pada matriks LED.
- Muat turun, simpan, terbit dan sunting fail MakeCode .hex.



- Menggunakan blok input untuk pengaturcaraan tetapan peristiwa.
- Menghasilkan dan menggunakan pembolehubah.



- Menggunakan pembaz piezo pada Music Bit untuk memainkan melodi.
- Menghasilkan dan menggunakan fungsi.
- Membaca skor muzik.



- Memprogram EDU:BIT untuk mengawal LED pada Traffic Light Bit.
- Menambah '*extension*' pada MakeCode Editor.



- Memprogram EDU:BIT untuk mengesan objek dengan menggunakan IR Bit.
- Menggunakan '*while loop*'
- Menghasilkan dan menggunakan '*array*'.



- Memprogram EDU:BIT untuk membaca nilai analog dari Potentio Bit.
- Memetakan bacaan input analog.
- Menggunakan blok-blok dari kategori [**Logic**] untuk program bersyarat.



- Memprogram EDU:BIT untuk mengesan bunyi dengan Sound Bit.
- Menggunakan pencetus acara untuk menukar dari satu mod ke mod yang lain.



- Memprogram EDU:BIT untuk mengawal motor DC – arah putaran dan kelajuan.
- Menggunakan blok-blok [Math] untuk melaksanakan operasi aritmetik.



- Memprogram EDU:BIT untuk mengawal motor servo – posisi sudut.



- Memprogram EDU:BIT untuk menyalakan RGB LED pada RGB Bit dengan pelbagai warna/corak.



- Memprogram EDU:BIT untuk menghantar dan menerima isyarat radio.

*Tandakan (✓) dalam kotak jika anda telah menguasai kemahiran tersebut: jika tidak, kembali semula ke bab tersebut untuk ulangkaji.

Pesanan daripada rero EDUteam @ Cytron

TAHNIAH!!!

Anda telah berjaya menyelesaikan semua bab dan belajar untuk mengkod dengan menggunakan Makecode Editor. Kami berharap anda seronok bermain dengan permainan popular kanak-kanak pilihan kami. Sekalung tahniah juga kerana berjaya menghasilkan beberapa aplikasi berguna untuk kelas anda.

Sekarang, anda sepatutnya telah memahami apa yang anda boleh lakukan dengan micro:bit dan semua extra *Bit* dari papan EDU:BIT anda. Psst..tahukah anda bahawa setiap *Bit* boleh diceraikan?

Anda boleh menghasilkan projek baru dengan menggunakan modul-modul yang telah diceraikan. Gunakanlah kabel '*plug-and-play*' yang disediakan untuk menyambung modul-modul tersebut ke papan utama EDU:BIT.

Inilah masanya untuk anda berfikir dan mencipta permainan dan aplikasi baru. Kami tidak sabar untuk melihat apa projek yang bakal anda cipta.

Jangan lupa untuk kongsi ciptaan anda dengan kami. Hantarkan emel atau tinggalkan mesej di laman FB kami.

Yang ikhlas,

Adam & Anna



Kongsikanlah perkembangan projek anda dengan kami.



[link.cytron.io/
edubit-resource-hub](https://link.cytron.io/edubit-resource-hub)

