

Robot Programming Guide

Building the robot and installing drivers.

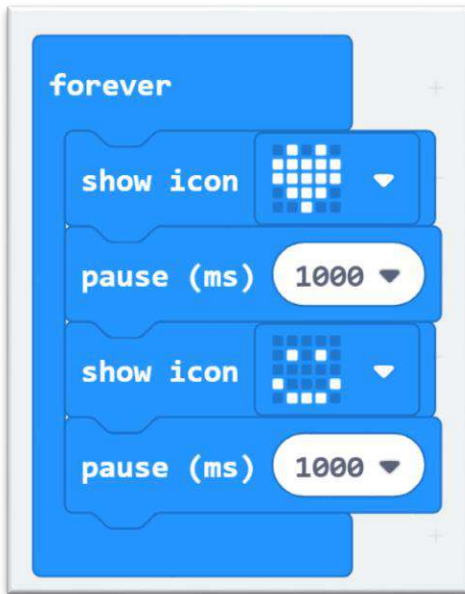
First, build the robot as according to the manual and install the drivers for the robot, both on the download tab of the product page for [KR9262](https://www.jaycar.com.au/product/kr9262).

Then, open up the Makecode website at:

<https://makecode.microbit.org/>

You will use this editor to program your device. You can make an example sketch as below.

First example blink sketch



Make the following code on the right in the “forever” block. Show Icon and Pause are found in the “Basic” category.

Once you have placed the blocks in this structure, you will find the animation on the left will move according to what you have set in the code editor on the right. This is so you can see what you are doing before you put it on to the microbit.

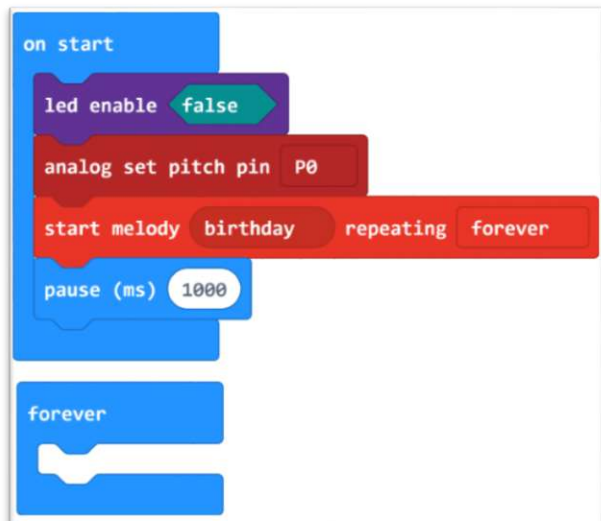
Plug in the microbit, it should show up as a new storage device on your computer. In the web editor you will be able to click “download” in the bottom left hand corner to download a .hex file. Drag this file from your downloads folder onto your microbit storage device. It should automatically disappear and reappear, and the

LEDs on the microbit should be animated in the pattern as programmed.

Passive Buzzer connected on P0

Copy the code to the right, and download onto your microbit. Then insert the microbit into your robot. Connect the 18650 battery and turn both the power and buzzer switches to the ON position.

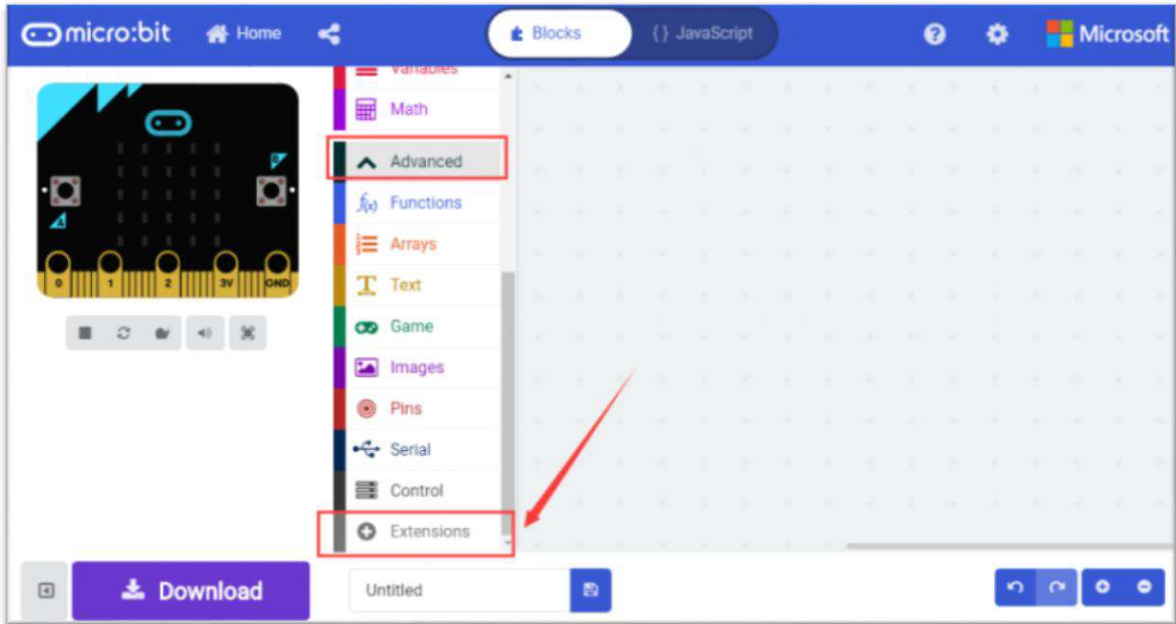
You should hear the microbit robot playing “Happy Birthday”.



Robot Programming Guide

18 RGB LED ring connected on P5

to use the RGB LEDs and some other features of the robot, you must install extensions found in the “advanced” tab of the makecode editor.



Search for and install the “NeoPixel” library.

You will then be able to make the following code and find the RGB strip changing colour.

```
on start
  set strip to NeoPixel at pin P5 with 18 leds as RGB (GRB format)
  strip clear

forever
  strip show color red
  pause (ms) 500
  strip show color orange
  pause (ms) 500
  strip show color yellow
  pause (ms) 500
  strip show color green
  pause (ms) 500
```

Australia
www.jaycar.com.au
techstore@jaycar.com.au
1800 022 888

New Zealand
www.jaycar.co.nz
techstore@jaycar.co.nz
0800 452 922



Robot Programming Guide

2 RGB Forward Facing LEDs

Install the “pca9685” extension and use the following code:

```
on start
  reset 67

forever
  set led duty cycle LED5 100 67
  set led duty cycle LED6 100 67
  set led duty cycle LED7 0 67
  pause (ms) 1000
  set led duty cycle LED5 100 67
  set led duty cycle LED6 0 67
  set led duty cycle LED7 100 67
  pause (ms) 1000
  set led duty cycle LED5 0 67
  set led duty cycle LED6 100 67
  set led duty cycle LED7 100 67
  pause (ms) 1000
```

You should be able to see that the RGB lights will change between Red Green and Blue.

Robot Programming Guide

Light Sensor

Install the extension as shown above in the RGB LED light ring, then use the below code.

```
on start
  set val to 0
  set strip to NeoPixel at pin P5 with 18 leds as RGB (GRB format)
  strip clear

forever
  set val to map analog read pin P1 from low 0 high 1023 to low 0 high 255
  strip show color red 255 - val green 255 - val blue 255 - val
  pause (ms) 100
```

As the room gets darker, the LEDs on the robot should get brighter.

Motors

The motors are managed through the pca9685 chip much like the LED forward facing lights. You must install the pca9685 extension to drive the motors.

Through the extension, LED1 – LED4 manage the direction and speed of motor:

Pin	Controls
LED1	Left Motor Direction (0: forward, 100: reverse)
LED2	Left Motor Speed (0: stop, 100: full speed)
LED3	Right Motor Direction (0: forward, 100: reverse)
LED4	Right Motor Speed (0: stop, 100: full speed)

Robot Programming Guide

Use the following code to make the robot drive around in a square on the ground:

These are the basic modules included on the robot. In the next few pages we will show you how to build robots such as a line following robot, obstacle avoidance, and follow me mode robot.

```

on start
  init 67 0

forever
  set led duty cycle LED1 0 67
  set led duty cycle LED2 100 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 100 67
  pause (ms) 1000
  set led duty cycle LED1 0 67
  set led duty cycle LED2 100 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 50 67
  pause (ms) 1000
  
```


Robot Programming Guide

Line Following robot.

The KR9262 robot comes with 2 infrared sensors near the base of the castor wheel for line following. These sensors have two potentiometers on the top of the robot chassis base to adjust the sensitivity of the sensors.

The left sensor is controlled by P12 and the right tracking sensor is controlled by P13.

To calibrate the sensors, place the robot on a sheet of white paper and adjust the potentiometers so that the D2 and D6 indicators are off, then slowly adjust until they turn on. Once they turn on you should have a good sweet spot of light detection and you should find that the sensors will not trigger on black tape or thick lines drawn in black marker. The source code is as below.

Testing source code:

When running the code as shown on the right, you should find that the robot will show either a line on the left or right side of the LED matrix, or it will show the large / small heart image, to indicate when and where it detects a black line on the ground.

You can use some black masking or electrical tape to build a track, and use the next code over page to see the robot travel around the page in action.

```
on start
  set LL to 0
  set RR to 0

forever
  set LL to digital read pin P12
  set RR to digital read pin P13
  if LL = 1 and RR = 0 then
    show leds
  else if LL = 0 and RR = 1 then
    show leds
  else if LL = 1 and RR = 1 then
    show icon [heart icon]
  else
    show icon [large heart icon]
```

Robot Programming Guide

```
on start
  init 67 0
  led enable false
  set strip to NeoPixel at pin P5 with 18 leds as RGB (GRB format)
  strip clear
  set led duty cycle LED5 100 67
  set led duty cycle LED6 100 67
  set led duty cycle LED7 100 67

forever
  strip show rainbow from 1 to 360
  set s1 to digital read pin P13
  set s2 to digital read pin P12
  if s1 = 1 and s2 = 0 then
    call function turn right
  else
    if s1 = 0 and s2 = 1 then
      call function turn left
    else if s1 = 0 and s2 = 0 then
      call function stop
    else
      call function foward

function foward
  set led duty cycle LED1 0 67
  set led duty cycle LED2 40 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 40 67

function back
  set led duty cycle LED1 100 67
  set led duty cycle LED2 35 67
  set led duty cycle LED3 100 67
  set led duty cycle LED4 35 67

function stop
  set led duty cycle LED1 0 67
  set led duty cycle LED2 0 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 0 67

function turn left
  set led duty cycle LED1 0 67
  set led duty cycle LED2 55 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 20 67

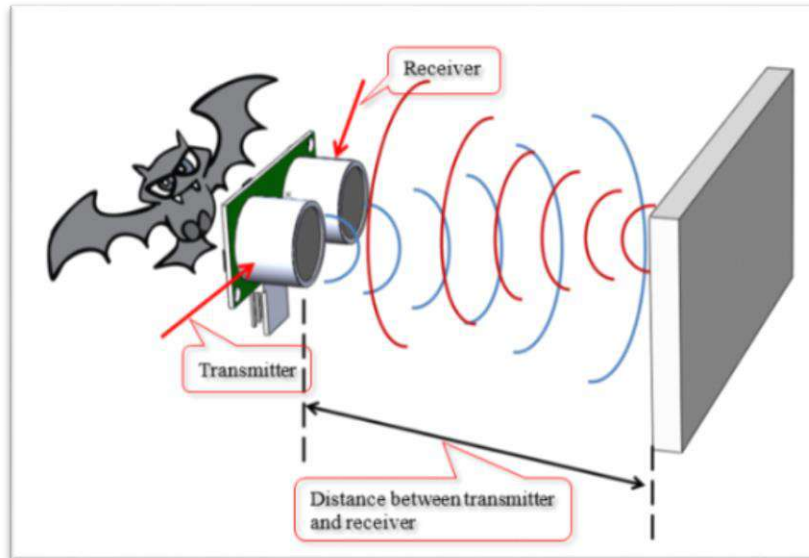
function turn right
  set led duty cycle LED1 0 67
  set led duty cycle LED2 20 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 55 67
```

This code first makes 5 functions, called “turn left”, “turn right”, “stop”, “back”, and “foward”. These make it easier to call in the main function later. Set up the LED strip and LEDs in the start block, and in the forever block, read the sensors, then determine if you are going forward, stop, left or right using if statements; call the functions so that you can reuse them when needed.

Robot Programming Guide

Ultrasonic Following Robot

We can use the ultrasonic sensor on the robot to detect distance, by emitting a sound (ultrasonic so humans can't hear it) and timing how long it takes to come back. This is similar to how a bat finds its way inside a cave.



We can use the code as shown on the right to test the ultrasonic sensor, when an obstacle is closer than 10 centimetres to the robot, the robot will beep.

See over page for the follow me robot source code.

```
on start
  led enable false
  analog set pitch pin P0
  set distance to 0
  set i to 0

forever
  set i to 0
  ping trig P14
  set distance to echo P15 unit cm
  serial write number distance
  serial write line
  if distance < 10 then
    while i < 1
      do
        play tone Middle D for 1/4 beat
        pause (ms) 200
        play tone Middle D for 1/4 beat
        pause (ms) 200
        change i by 1
```


Robot Programming Guide

```
on start
  reset 67
  set Distance to 0
  set strip to NeoPixel at pin P5 with 18 leds as RGB (GRB format)

forever
  ping trig P14
  set Distance to echo P15 unit cm

  if Distance ≥ 10 and Distance ≤ 30 then
    call function front
    strip show color blue
  +
  if Distance ≤ 6 then
    call function back
    strip show color purple
  +
  if Distance > 6 and Distance < 10 or Distance > 30 then
    call function stop
    strip show color red

function front
  set led duty cycle LED1 0 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 50 67

function left
  set led duty cycle LED1 0 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 100 67

function back
  set led duty cycle LED1 100 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 100 67
  set led duty cycle LED4 50 67

function right
  set led duty cycle LED1 0 67
  set led duty cycle LED2 100 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 50 67

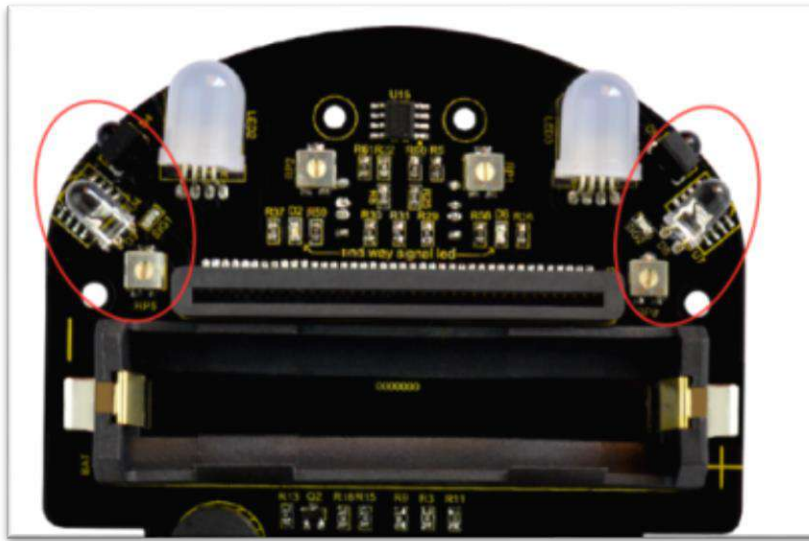
function stop
  set led duty cycle LED2 0 67
  set led duty cycle LED1 0 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 0 67
```

This code sets the same 5 functions as before, but now tests if distance is too close or too far, and drives accordingly.

Robot Programming Guide

Obstacle Avoidance with Follow me mode

The robot also comes with two infrared obstacle avoidance sensors which are facing out from the side of the robot.



The detection distance is at best, around 10 cm, and can be adjusted for sensitivity with the small potentiometers close by. The left IR sensor is connected to P2 and the right is connected to P11 and will read LOW when an obstacle is detected.

```
on start
  set pull pin P2 to up
  set pull pin P11 to up

forever
  set LL to digital read pin P2
  set RR to digital read pin P11
  if LL = 1 and RR = 1 then
    show leds
  else if LL = 0 and RR = 1 then
    show leds
  else if LL = 1 and RR = 0 then
    show leds
  else
    show icon
```

Testing source code:

Set pull pin P2 to UP

Set pull pin P11 to UP.

Then read similar to line following robot.

Read overpage for the following robot with obstacle avoidance.

Robot Programming Guide

```

on start
  set pull pin P2 to up
  set pull pin P11 to up
  set Distance to 0
  init 67 0
  set strip to NeoPixel at pin P5 with 18 leds as RGB (GRB format)

forever
  ping trig P14
  set Distance to echo P15 unit cm
  set LL to digital read pin P2
  set RR to digital read pin P11

  if Distance > 10 then
    if LL = 1 and RR = 1 then
      call function front
      strip show color blue
    else if LL = 1 and RR = 0 then
      call function left
      strip show color purple
    else if LL = 0 and RR = 1 then
      call function right
      strip show color yellow
    else
      call function back
      strip show color red
      pause (ms) 100
      call function left

  else
    if LL = 1 and RR = 1 then
      call function left
      strip show color purple
    else if LL = 1 and RR = 0 then
      call function left
      strip show color purple
    else if LL = 0 and RR = 1 then
      call function right
      strip show color yellow
    else
      call function back
      pause (ms) 100
      strip show color red
      call function left

function front
  set led duty cycle LED1 0 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 50 67

function back
  set led duty cycle LED1 100 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 100 67
  set led duty cycle LED4 50 67

function stop
  set led duty cycle LED2 0 67
  set led duty cycle LED1 0 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 0 67

function left
  set led duty cycle LED1 100 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 0 67
  set led duty cycle LED4 50 67

function right
  set led duty cycle LED1 0 67
  set led duty cycle LED2 50 67
  set led duty cycle LED3 100 67
  set led duty cycle LED4 50 67
  
```

Australia
www.jaycar.com.au
techstore@jaycar.com.au
 1800 022 888

New Zealand
www.jaycar.co.nz
techstore@jaycar.co.nz
 0800 452 922



Robot Programming Guide

Bluetooth Robot

You need to install 2 extensions to use Bluetooth functions:

- Bluetooth
- BlocklyTalkyBLE

You must then use the “on string received” event to determine what button is pressed on the Bluetooth app.

The Bluetooth app to use for this example is located here at the bottom of this page (android only)

IOS users or for other android apps should be able to search for Bluetooth microbit apps and follow the related tutorials for it to work with those apps.

```
on bluetooth disconnected
  show icon

on bluetooth connected
  show icon

on start
  led enable true
  init 67 0

on string received key show stringValue
  if stringValue = F then
    call function front
  +
  if stringValue = B then
    call function back
  +
  if stringValue = L then
    call function left
  +
  if stringValue = R then
    call function right
  +
  if stringValue = S then
    call function stop
  +
```

https://drive.google.com/open?id=1u_wXlpuRjCChpCcVPclha4yWYWhefhW2

Australia
www.jaycar.com.au
techstore@jaycar.com.au
1800 022 888

New Zealand
www.jaycar.co.nz
techstore@jaycar.co.nz
0800 452 922

