

GENIBOT

BBC micro:bit-GENIBOT Radio Programming Guide



GB1-RPG0805P4 August 5, 2024

GENIBOT

BBC micro:bit-GENIBOT Radio Programming Guide

Technical Support and Product Information

Address: G749, 815, Pangyo 2nd Techno Valley Smart Industrial Complex,
Daewangpangyo-ro, Sujeong-gu, Seongnam-si, Gyeonggi-do, 13449, Korea
www.genirobot.com

GENIBOT is product and trade name of Dabida (GENIROBOT) Co., Ltd. Other trademarks or trade names of products and company names mentioned herein are: Microsoft Windows and Make Code, Google Android, Apple iOS, BBC micro:bit Educational Foundation, Scratch Foundation and Python Software Foundation.



© 2024 Dabida (GENIROBOT) Co., Ltd. All rights reserved.

IMPORTANT

Warnings Regarding Use

Statements on symbols, terms and conventions you may see in this guide give the information that you should consider installing and operating the robot GENIBOT.

The robot GENIBOT is designed only for a level of reliability suitable for educational IoT smart devices. It may not be used on or in connection with any equipment, material or system that could reasonably be expected to cause serious personal injury.

In any application this hardware can be impaired by adverse factors, electrical power supply, computer hardware and software malfunctions, unexpected uses or errors on users. If the robot GENIBOT is used and operated with non-certified devices or in any manner not specified by the manufacturer, the protective features of the robot GENIBOT may be damaged and it is not certified for use in hazardous locations.

The materials of the robot GENIBOT are provided “as is,” and is subject to being changed, without prior notice to users.

Dabida (GENIROBOT) Co. Ltd. shall not be liable for events, errors, incidental or consequential damages in connection with improper use, performance of this document or of any information described herein.



Precautions for use

Pay special attention to the requirements and operate the robot GENIBOT only as described in the “Warning” and “Caution” conditions.

Using and operating the robot GENIBOT under conditions other than those specified above may result in danger and injury, as well as damage to the robot GENIBOT and external devices connected to the robot.

AT A GLANCE

Technical Features

Specifications

Processor (CPU):	ARM Cortex M4 32bit processor, 64MHz, RAM 64KiB, Flash 512KiB
Power Source	DC Power limit (Rated 5V DC \leq 300mA) DC Power adaptor (5V DC \leq 1A)
Power Terminal	VDD 5V USB C-type connector
Battery	Li-Polymer 3.7V 1000mAh, overcurrent and overvoltage protection
Connectivity	Bluetooth LE 5 (5.0/5.1) PHY 2Mbps multiprotocol and ISM band 2.4GHz radio protocol Distance < 30m, Signal strength RSSI > -90dBm
Light Sensor and Green LED	Four ambient light sensors and four green LEDs for line following and Grid detection
Acceleration Sensor	Acceleration (State of the-Art 3-axis, range -2g to +2g) Not applicable to GENIBOT Plus version (Model D)
RGB Color LED	RGB color LED (4) for making and changing Color in HSV or RGB color space
Red LED	Red LED on external terminal port for charging battery
OID Module	Optical image detection sensor and decoder for recognizing unplugged coding card or reading position in Cartesian coordinate with 16bit dot code pattern detection.
Speaker	Impedance $8\Omega \pm 15\%$ 1.0V 1kHz Maximum power 1.0W Effective frequency band 200 to 4kHz and flat response center frequency 2kHz

Audio Amplifier	PCM Class D audio amplifier
Memory	SLC NAND flash memory 128MiB (134MB)
Motor	Stepper (50:1 Geared, 360DPS/1000SPS)
External AI and DIO Port	<p>Standard drive output voltage will be from minimum (VDD - 0.3V) to maximum (VDD + 0.3V) where VDD is +3.3V.</p> <p>When using VDD from +3.0V to 3.3V, the absolute maximum input voltage must be less than (VDD + 0.3V).</p> <p>If VDD is greater than 3.3V (VDD + 0.3V), ESD protection will operate, but nothing is guaranteed about its function and operation.</p> <p>Maximum sink current is 6mA with standard drive at (VDD - 0.4V), and 15mA with high drive at (VDD - 0.4V) where VDD is +3.3V.</p>
External AI and DIO Port Power Source	<p>Standard drive output voltage will be from minimum (VDD - 0.3V) to maximum (VDD + 0.3V) where VDD is +3.3V.</p> <p>Maximum output current is 300mA, which can provide sliding current to drive Geek Servo 270 (Rated DC current = 200mA at +3.0V) or Arduino BBC micro:servo 180 motor (Rated DC current is 75mA at +3.0V)</p>



Battery Consumptions

Battery life is approximately 1.5 hours when the motor is running continuously. If the robot is used intermittently during class activities, it can be used for more than a few days. If you rarely use the robot, the battery may last more than a month after a full charge.

When charging the robot GENIBOT, you must use a certified power adapter such as KC, CE, or FCC that has been verified for safety.

Button and Color LED

Power Button

To turn on the robot, press the power button. When the robot starts, it says “Hello, I’m GENIBOT”.

To turn off the robot, press and hold the power button for more than two seconds. When the robot shuts down, it says “Bye, bye see you later”. However, in certain activity situations, the robot may restart without a power-off greeting.

To start or stop the unplugged coding when coding data set is stored in memory, press the power button.

To advertise the robot’s Bluetooth discovery name while the Star robot or smart device is scanning GENIBOT, press the power button.

To connect a star robot or peer robot, just press the power button once.

To set up a BBC micro:bit radio group, press the power button and then tap two number cards.

Color LED

Color will be changed when GENIBOT is connected to a star robot or smart device but also color will be indicated to matching card color when it is detected by GENIBOT OID image detection sensor

Color can be changed in HSV color coordinate as: Red, Green, Blue, Cyan, Magenta, Yellow, Violet, Orange, Spring Green, Light Pink and White by using GENIBOT Android and iOS application, Scratch or Python.

Moreover, with Scratch and Python programming, you can change the left, right, front, and back colors independently.

Hue in HSV (Hue, Saturation and Value) color space can be changed from 0 to 360 in degrees.

GETTING STARTED

Before Start

The GENIBOT application can be used to perform a variety of coding activities and offers expanded functionality for those learning and teaching STEAM (Science, Technology, Engineering, Arts, and Mathematics).

The GENIBOT app for Google Android or Apple iOS is easy to use and offers a variety of coding activities including Action Bar, Controller, Card Coding, Line Tracing, Button Coding, Music Coding, Drawing, Math Coding, and Voice Coding using AI voice recognition. and operational functions.

To start coding with the GENIBOT application for Google Android or Apple iOS, first press a button on the robot to connect it to your smart device, then select an activity such as Remote Control, Card Coding, Music Coding or Voice Coding.

For Unplugged coding without connecting the robot to a smart device, Physical Computing or EPL (Educational Programming Language) activities, refer to this document or others provided separately such as GENIBOT Arduino Reference, BBC micro:bit-GENIBOT Radio Programming Guide and etc.



In the classroom

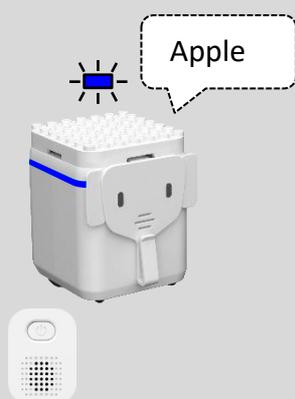
Turn off the robot when not in use. Fully charge the robot to carry out a variety of coding activities over long periods of time in the classroom.

STARTING GENIBOT RADIO

Using Unplugged Card

1. Power on the robot. Press the button once, the robot will blink blue every 0.5s for starting Bluetooth.
2. Tap two number cards in the digit order for matching BBC micro:bit radio group number from 1 to 80. Band number is fixed to 0 in Unplugged.
3. If you want to set radio group number lower than 10, add zero at the beginning like two-digit number, for examples as "01", "04", ... etc.
4. Press the button again to stop Bluetooth, blue light will be blinking three times fast and then the robot will start radio instead Bluetooth.

Example: Setting GENIBOT radio group number 24.



Press the button of the star robot.



While the blue light flashes every 0.5 seconds, tap two number cards designated by the radio group number from 01 to 80.



Press the button again to start listening to the BBC micro:bit.



Setting Band and Group: To listen to BBC micro:bit radio and performs actions according to the radio messages, you can set GENIBOT radio band and group number between 0 to 80.

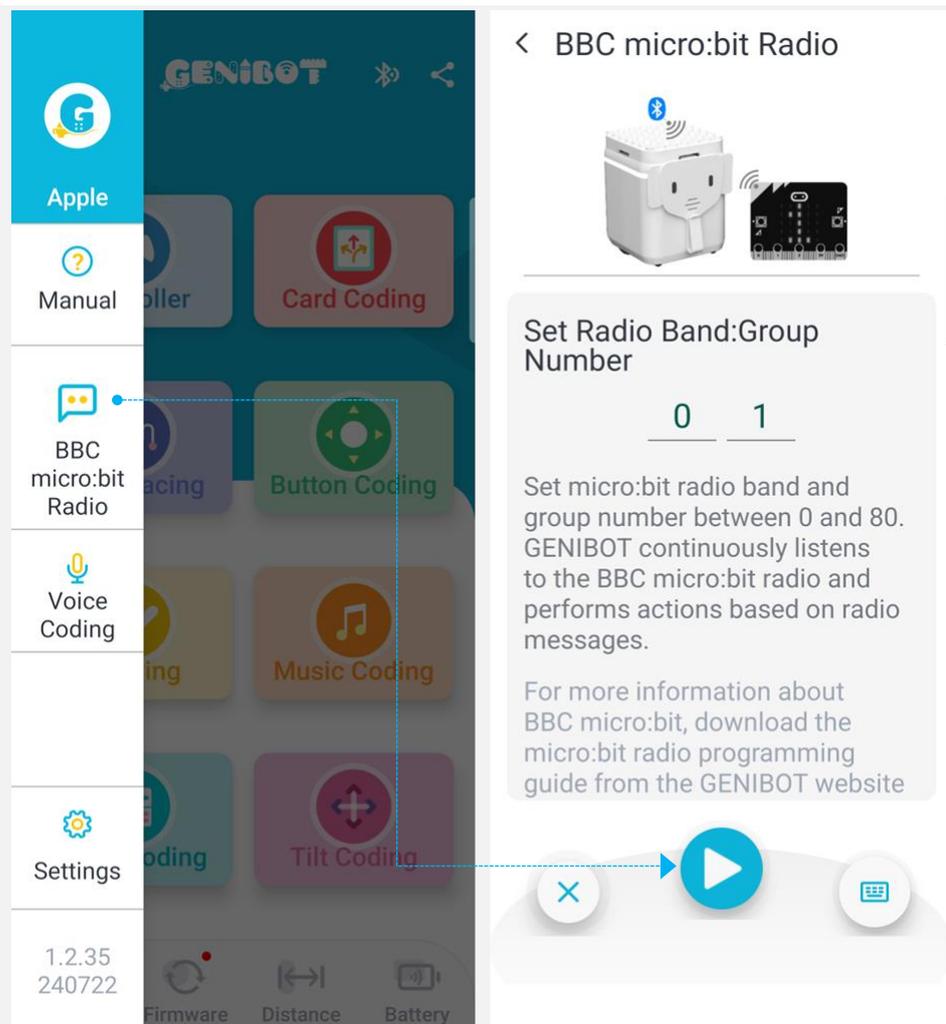
If you configure GENIBOT radio using the Unplugged cards, Bluetooth will be disabled and you will not be able to connect your robot to a computer or smart device until the robot reboots, and the radio band number is fixed at 7, but this band value may change in a later firmware update.

On the other hand, if you set GENIBOT radio band and group with Scratch 3 GENIBOT app for Android or iOS, you can use GENIBOT radio and Bluetooth at the same time. Although GENIBOT can use Bluetooth and radio simultaneously, BBC micro:bit cannot use Bluetooth while using the radio.

To program BBC micro:bit, visit <https://microbit.org/> and set BBC micro: bit's radio band and group number to be the same as GENIBOT's settings.

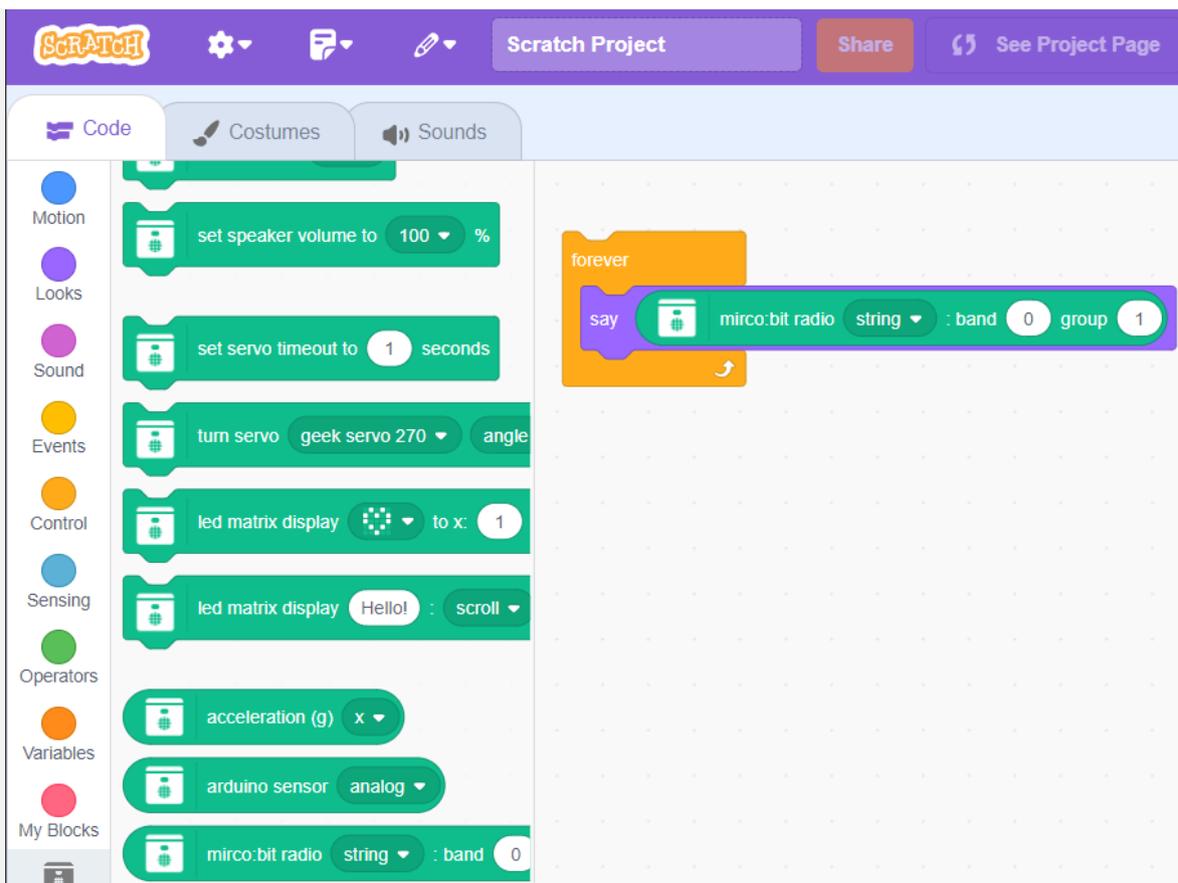
Using GENIBOT App for Android or iOS

1. Power on the robot. Press the button once, the robot will blink blue every 0.5s for starting Bluetooth.
2. Connect the robot via Bluetooth with GENIBOT app for Android or iOS.
3. Select BBC micro:bit from the sliding expansion menu of GENIBOT app, then set the Radio Band and Group Number.
4. You can use GENIBOT app for robot control, card coding, music coding, and other programming while listening to BBC micro:bit.
5. Because GENIBOT can use Bluetooth and radio simultaneously, while the robot listens for radio messages from BBC micro:bit, you can simultaneously send control commands to the robot via Bluetooth.



Using Scratch 3

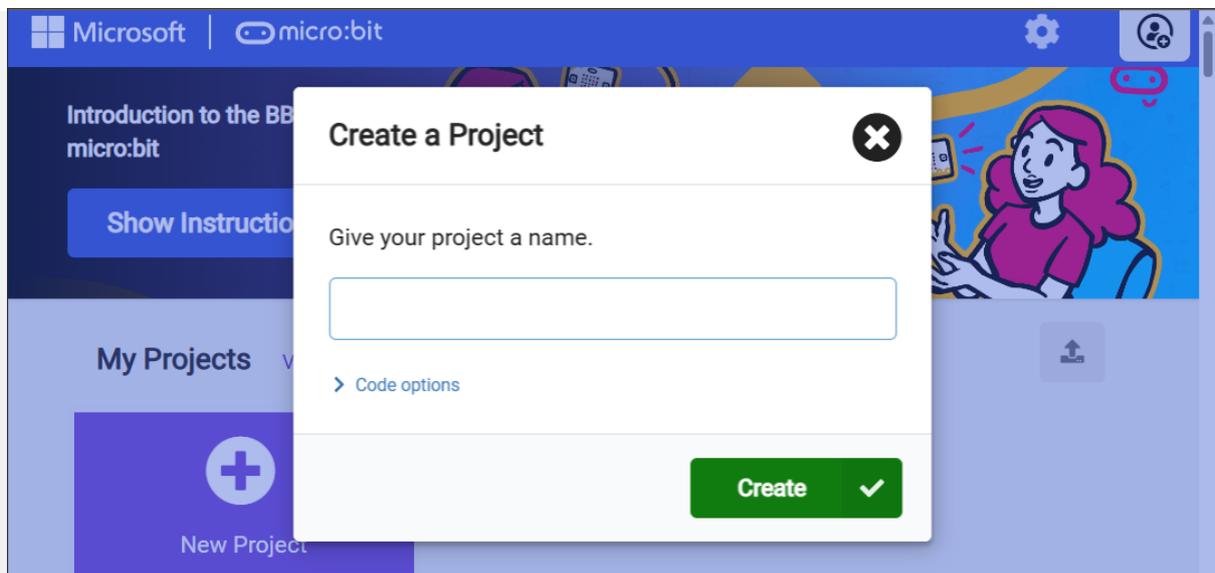
1. Power on the robot. Press the button once, the robot will blink blue every 0.5s for starting Bluetooth.
2. Connect the robot via Bluetooth with Scratch 3 Desktop application.
3. Select BBC micro:bit radio block and set the band and group. When you click on BBC micro:bit radio block once, the robot starts radio and listens to radio messages from BBC micro:bit.
4. Because GENIBOT can use Bluetooth and radio simultaneously, while the robot listens for radio messages from BBC micro:bit, you can simultaneously send control commands to the robot via Bluetooth.



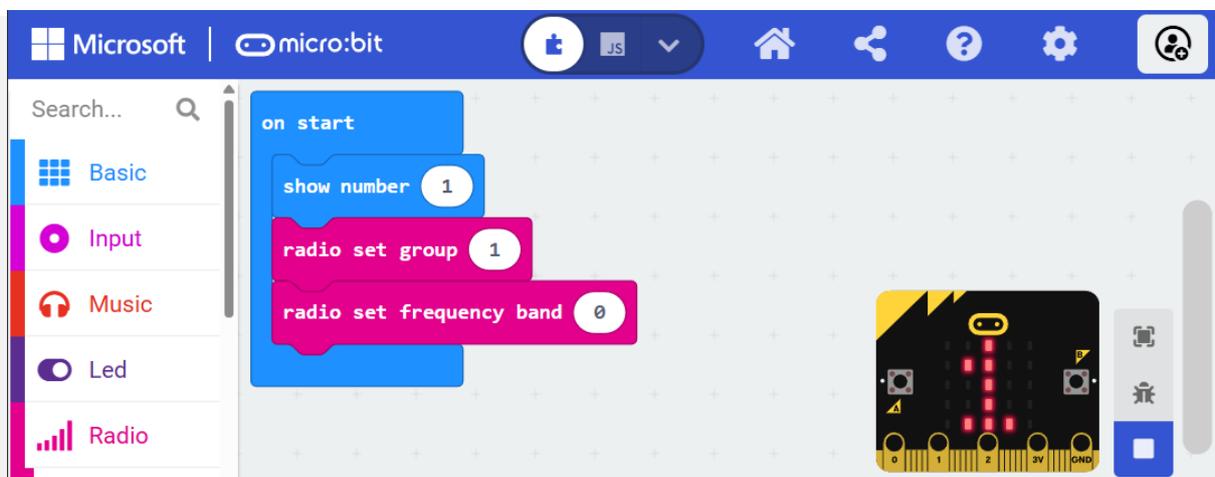
PROGRAMMING BBC micro:bit

Creating a new project

1. To program code using BBC micro:bit, visit Microsoft MakeCode editor site at <https://makecode.microbit.org/>, click **New Project** to create a project.



2. Set or change radio **group** id from 1 to 80 using **radio set group** block. Set radio band 0 to 80 using **set frequency band** block.



3. Place these two blocks in **on start** block to start radio when the program starts.
4. Make a syntax statement with command string and decimal number of the value using **radio send string** block.

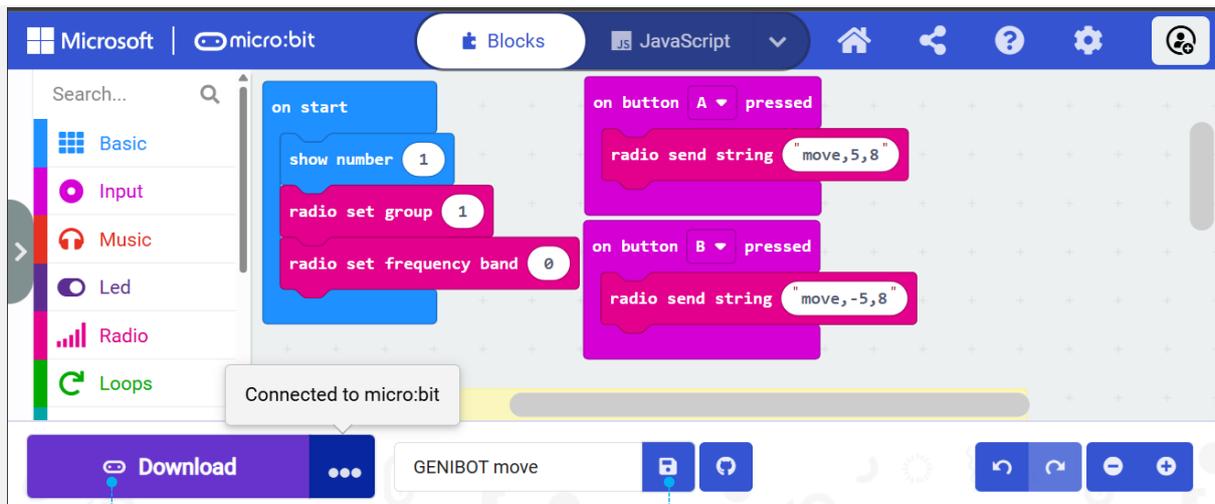


Band and Syntax: If you are configuring GENIBOT radio using the Unplugged cards, edit the band number to 0, but this band value may change in a later firmware update.

Maximum number of characters using **radio send string** block is limited to 19. You **DO NOT use space character** which is not necessary in the statement, if the number of characters increased, it makes an error to transmit an invalid statement with the lack of data set to the robot.

Flashing code to BBC micro:bit

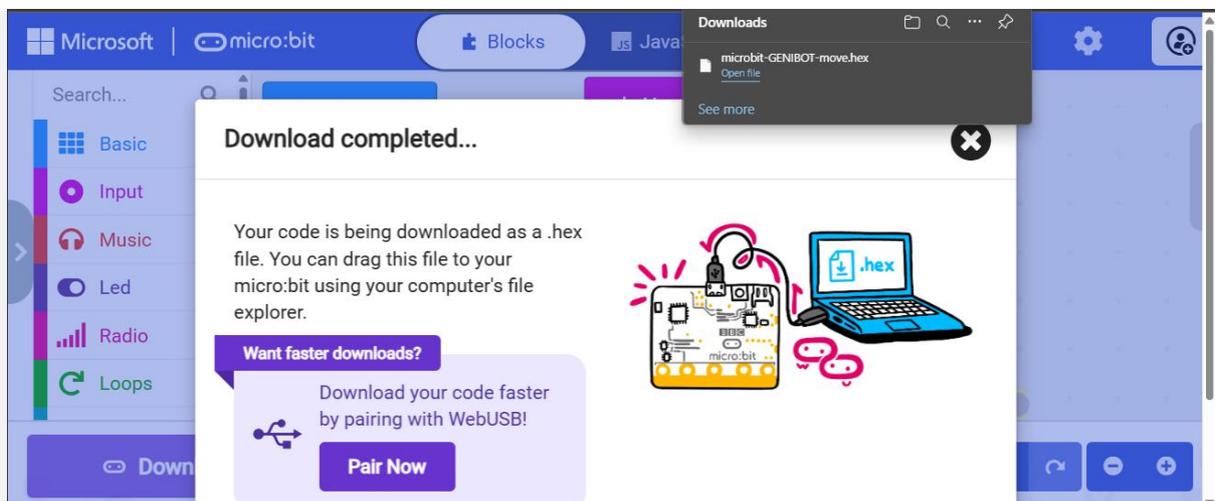
1. In the MakeCode Editor your code is saved automatically in the browser. To flash your code to micro:bit when your micro:bit is paired and connected to your computer, press **Download** button in the bottom left of the screen.



Click disk icon: Save program file .hex on computer

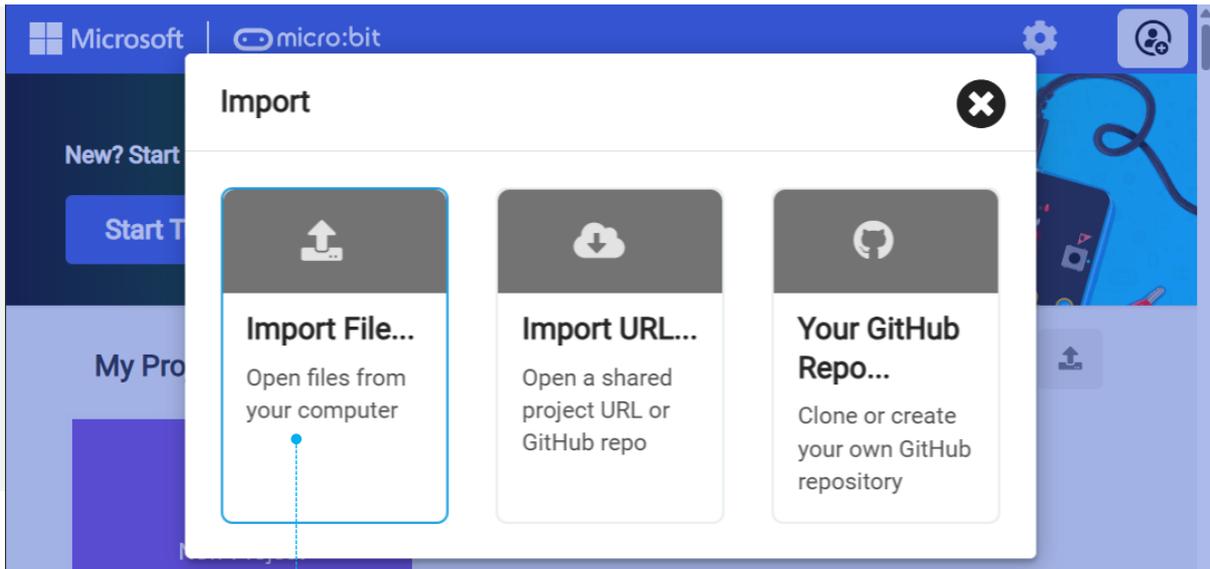
Micro:bit logo appears: Download code to BBC micro:bit

2. To save your code and program on your computer that you prefer, in the MakeCode Editor download your project .hex file on your computer using **Download** button when your BBC micro:bit is disconnected.



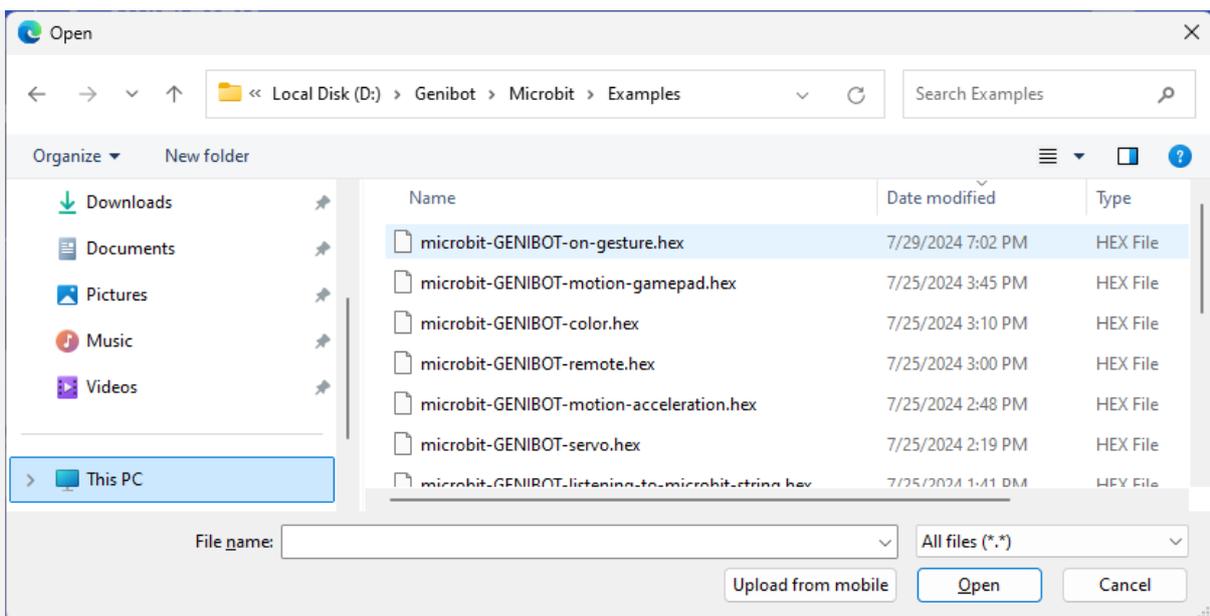
Importing code from computer

1. To import your code, click **Import** button and then click **Choose File** button to browse and open your project **.hex** file from your computer.



Import code **.hex** file from computer

2. Choose your project **.hex** file to import code and program.



Statement Syntax

Commands	Descriptions
motion , <i>velocity</i> <i>left,velocity right</i>	To make linear or rotational motion, set velocity left and right of the robot in SPS (steps per seconds). <i>Velocity left = -5 to 5</i> <i>Velocity right = -5 to 5</i>
move , <i>velocity,distance</i>	To step forward or backward, set velocity of the robot in SPS and distance (cm). <i>Velocity = -5 to 5</i> <i>Distance (cm) = 1 to 15</i>
turn , <i>velocity,angle</i>	To make rotational motion, set velocity of the robot in SPS and rotational angle in degrees. <i>Velocity = -5 to 5</i> <i>Angle in degrees = 1 to 180</i>
color , <i>LED number,color number,brightness</i>	To display colors, set color LED number, color number and brightness level. <i>Color LED number = 1 to 5</i> <i>(left 4, right 2, front 1, back 3, all 0)</i> <i>Color number = 0 to 11</i> <i>(red 0, green 1, blue 2, cyan 3, magenta 4, yellow 5, violet 6, orange 7, spring green 8, light pink 9, white 10, black 11)</i> <i>Brightness level = 1 to 8</i>

Commands	Descriptions
sound , <i>music instrument number,note,volume</i>	<p>To play music note, set music instrument number, note and volume.</p> <p><i>Music instrument number = 0 to 2 (piano 0, flute 1, string 2)</i></p> <p><i>Note = 37 to 57 (A3 La to F5 Fa)</i></p> <p><i>Volume = 1 to 9</i></p>
tone , <i>frequency,duration, amplitude</i>	<p>To generate pure tone with sinusoidal waveform, set frequency (Hz), duration and amplitude.</p> <p><i>Frequency (Hz) = 220 to 1100</i></p> <p><i>Duration in 0.1seconds = 2 to 20 (0.2s to 2s)</i></p> <p><i>Amplitude = 1 to 10</i></p>
remote , <i>acceleration ax,acceleration ay</i>	<p>To make any linear or rotational motion, transmit continuous acceleration measurement data set of ax and ay to the robot.</p> <p><i>Acceleration ax = -1024 to 1024,</i></p> <p><i>Acceleration ay = -1024 to 1024</i></p>
servo , <i>geek or micro:servo,angle,wait time</i>	<p>To make servo motion, set geek or micro:servo, angle, and wait time in seconds.</p> <p>Servo type: geek servo 0, micro:servo 1</p> <p>Angle in degrees: geek -45 to 225, micro:servo 0 to 180</p> <p>Wait time in seconds: 1s to 10s</p> <p>When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task</p>

Commands	Descriptions
	<p>priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.</p>
<p>on,gesture</p>	<p>When you do a gesture which means the way you hold or move the micro:bit, send “on gesture” parameter to the robot. gesture parameters: shake, screen up, screen down, tilt left, tilt right.</p>



Statement Syntax: Some radio block commands may be added or changed in these syntax statements in a later firmware update.

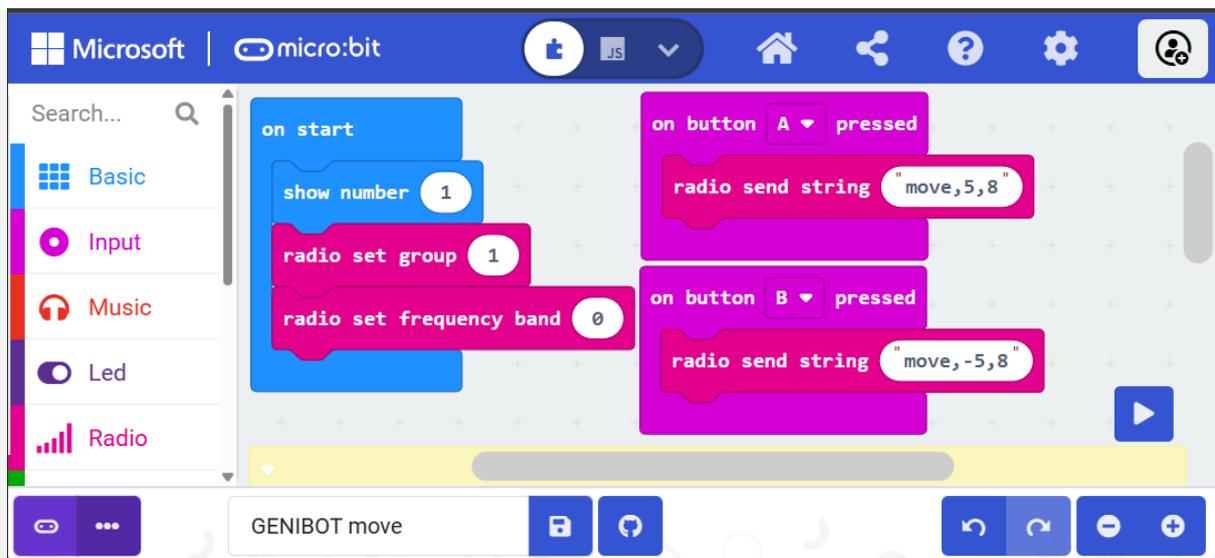
This statement syntax is a function defined to control the robot with **radio send string** block of BBC micro:bit. If the syntax is not defined, it can be used to control the robot by broadcasting a numeric value with **radio send number** block.

Examples: Statement Syntax

move

(*move,velocity,distance*)

Move forward 8cm at velocity 5 when button A is pressed. Move backward 8cm at velocity -5 when button B is pressed.



Example: (move,velocity,distance)

velocity: -5 to 5

distance in cm: 1 to 15

In this example, the move command causes the robot to stop after moving a certain distance, so instead of using this command in a forever loop, the command is sent only when there is a button event.

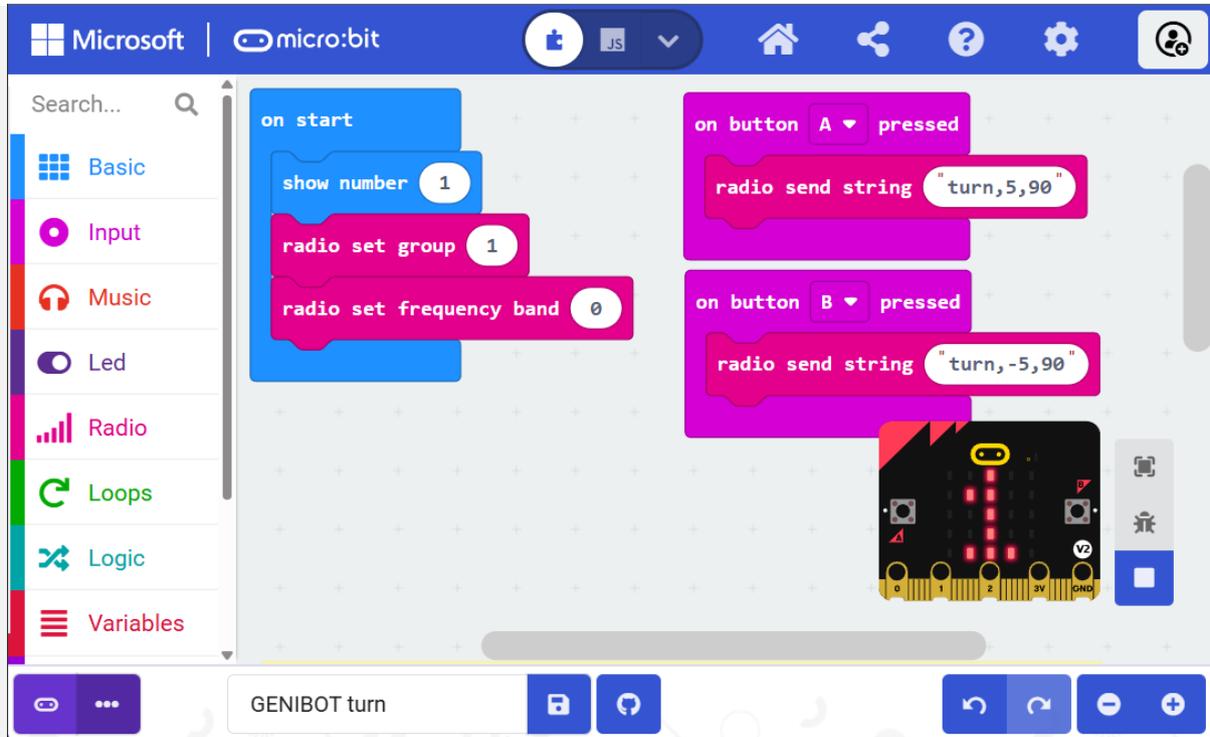
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

turn

(turn,velocity,angle)

Make 90 degree left turns at velocity 5 when button A is pressed. Make 90 degree right turns at velocity -5 when button B is pressed.



Example: (turn,velocity,angle)

velocity: -5 to 5

angle in degrees: 0 to 180

In this example, when a button event occurs, the robot rotates at a constant speed by a set angle.

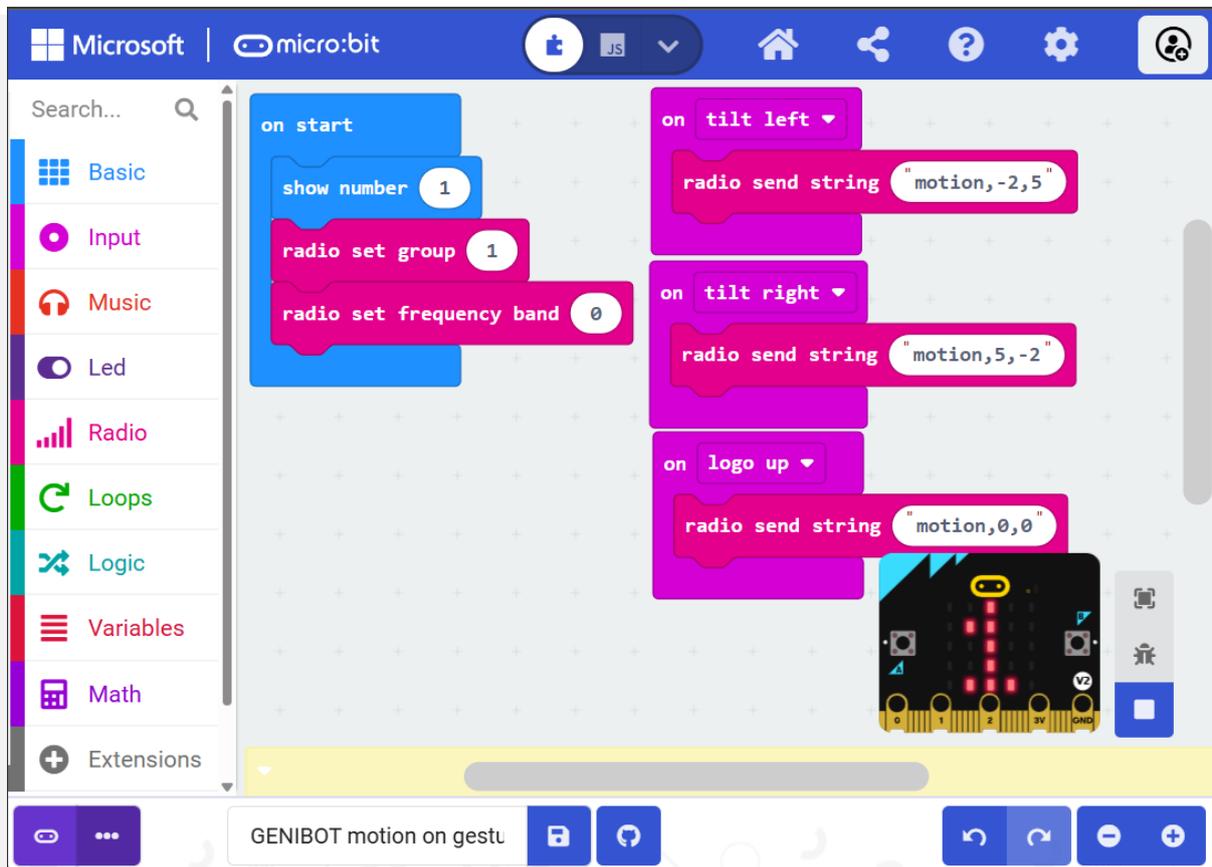
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

motion

(*motion,velocity left,velocity right*)

Make constant forward motion at left velocity and right velocity.



Example: (motion,velocity left,velocity right)

velocity: -5 to 5

In this example, the robot continues to rotate or stop at a constant speed when a tilt event occurs.

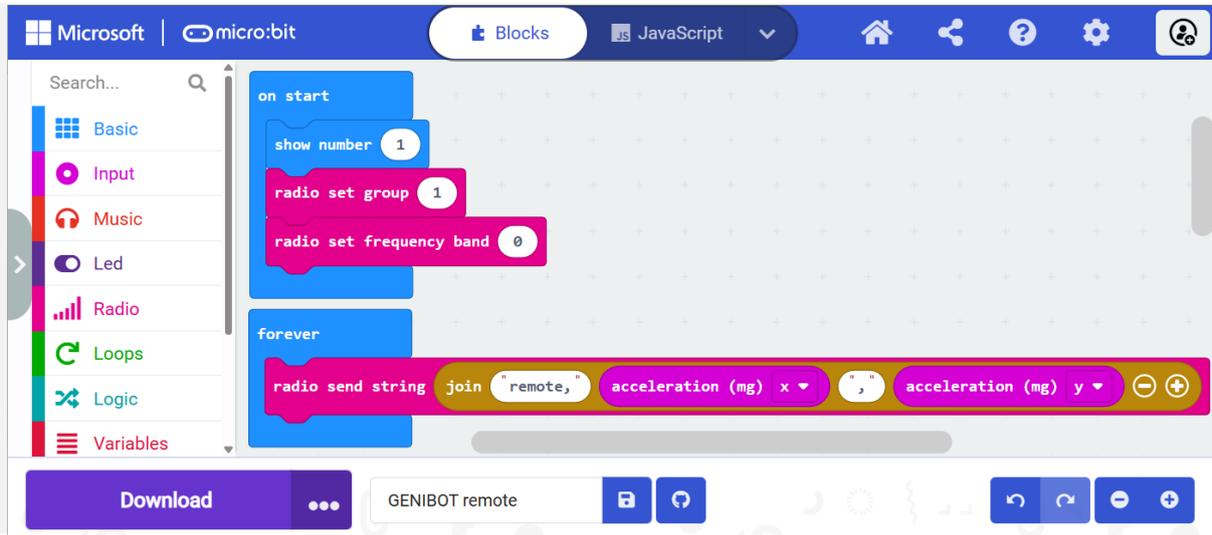
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

remote

(remote, acceleration ax, acceleration ay)

Make any motion by continuous measurement data set of acceleration x and y.



Example: (remote, acceleration ax, acceleration ay)

acceleration ax: -1024 to 1024

acceleration ay: -1024 to 1024

To make any linear or rotational motion, send continuous acceleration measurement data set of ax and ay to the robot.

This example shows moving or rotating a robot with changing acceleration values. When you shake the micro:bit, the value of the acceleration sensor changes quickly, so the robot's movement changes according to the changing value.

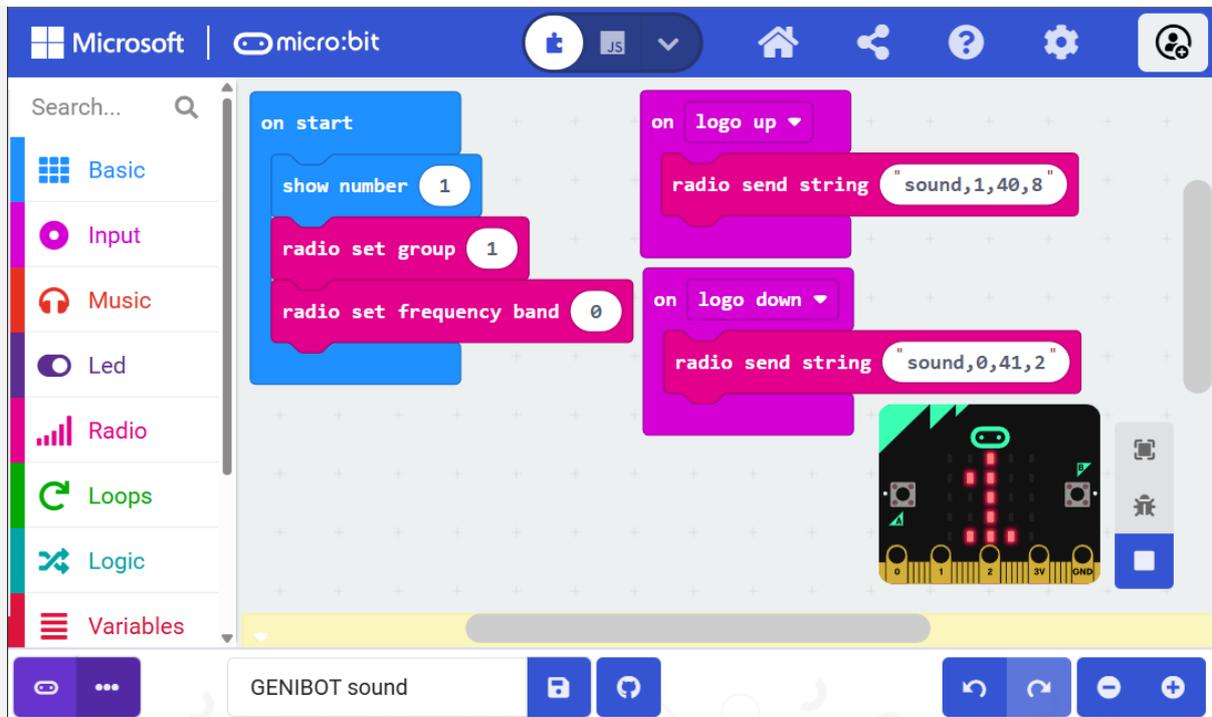
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

sound

(*sound,instrument,note,volume*)

Play flute note 40 at volume 8 on logo up. Play piano note 41 at volume 2 on logo down.



Example: (sound,music instrument number,note,volume)

music instrument number: piano 0, flute 1, string 2

note: 37 to 57 (A3 La to F5 Fa)

volume: 1 to 9

In this example, the robot plays a piano sound when a tilt event occurs.

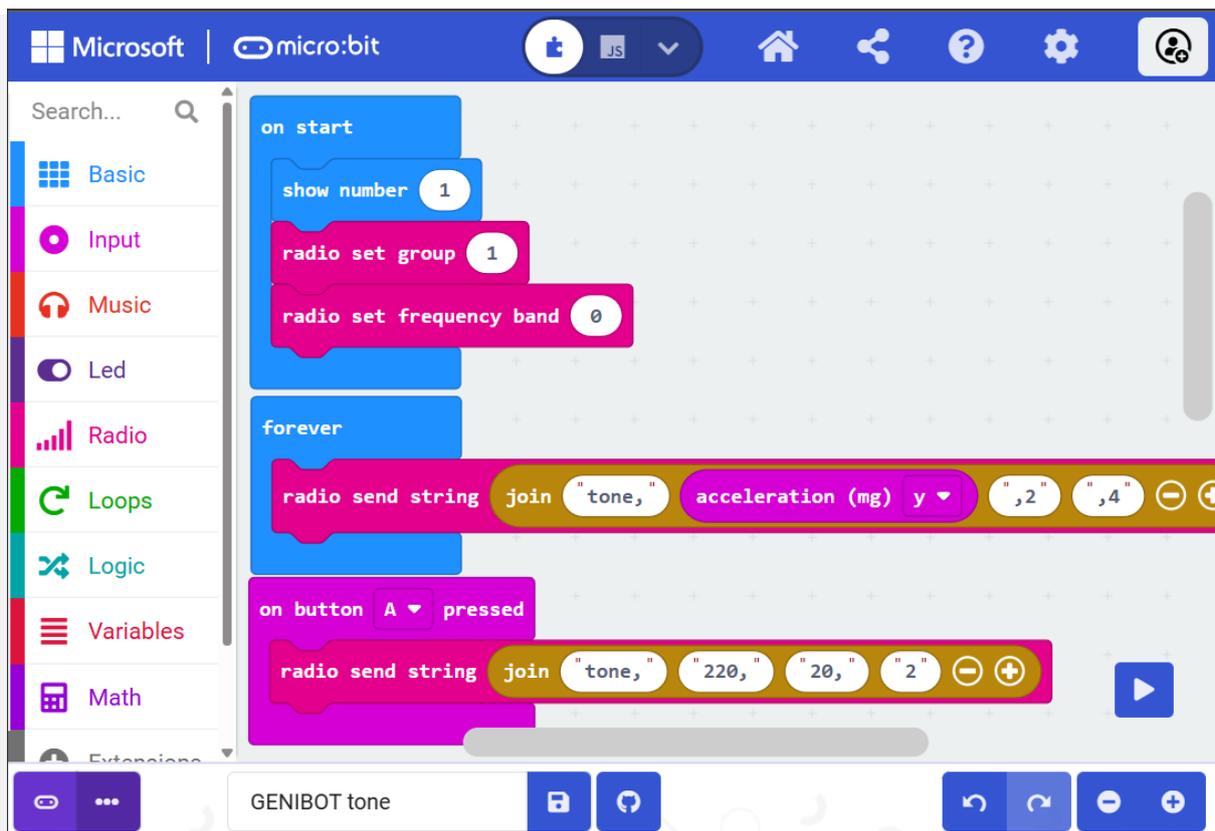
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

tone

(tone,frequency,duration,amplitude)

Generate 220Hz pure tone in 2seconds at volume 2 when button A is pressed.
Generate any tone which is scaled by acceleration ay (acceleration of y-axis) in 0.2seconds at volume 4.



Example: (tone,frequency,duration,amplitude)

Frequency in Hz: 220 to 1100

Duration in 0.1seconds: 2 to 20 (0.2s to 2s)

Amplitude: 1 to 10

To generate pure tone with sinusoidal waveform, set frequency (Hz), duration and amplitude.

In this example, the pitch (frequency) and loudness (amplitude) of the sound are changed using the acceleration sensor value.

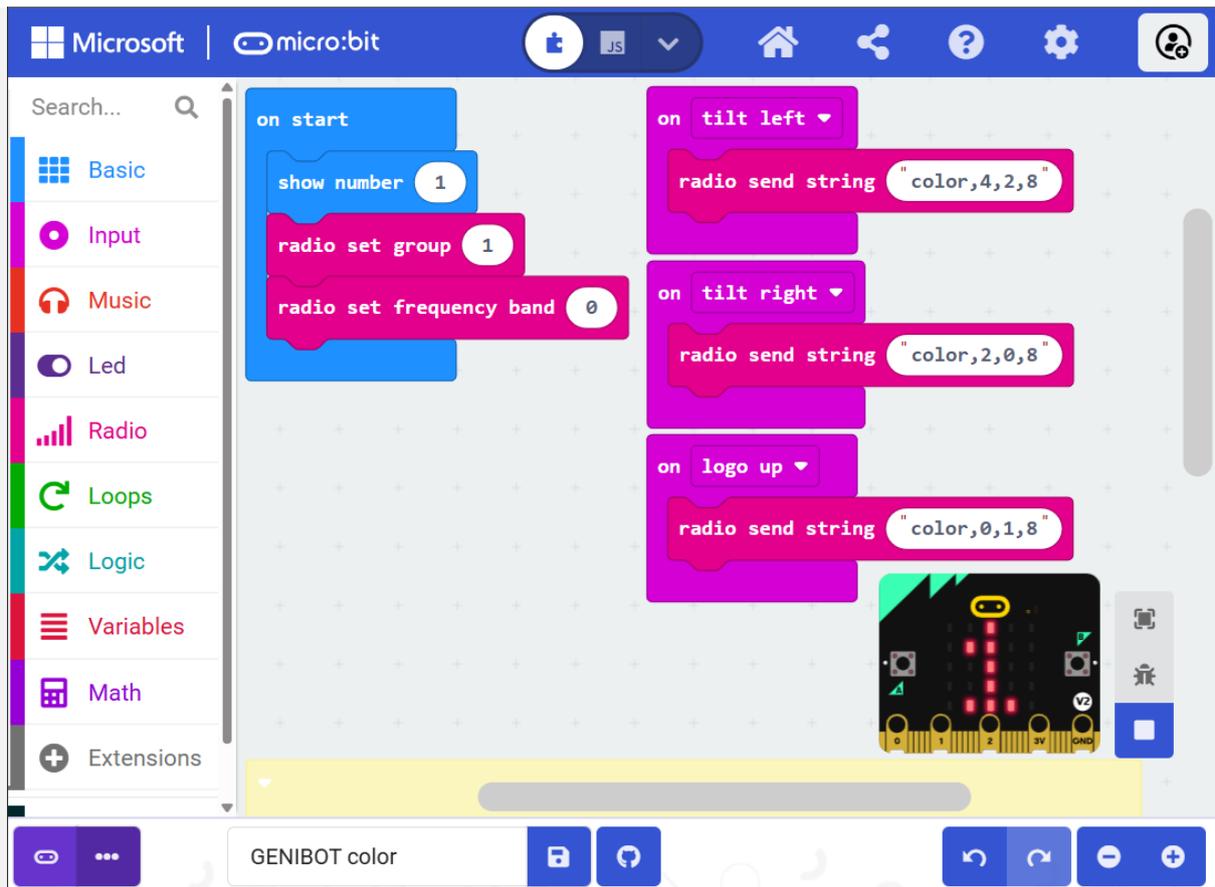
Advanced course:

Let's assign the radio send string to be sent to the robot as a variable and program it to transmit the value to the robot only when this value changes.

color

(color,LED number,color number,brightness)

Show yellow on the left or right side at brightness level 8 when BBC micro:bit status is tilt left or right. Show white on screen up.



Example: (color,LED number,color number,brightness)

LED number: left 4, right 2, front 1, back 3, all 0

color number: red 0, blue 2, cyan 3, magenta 4, yellow 5, purple 6, orange 7, spring green 8. light pink 9, white 10, black 11

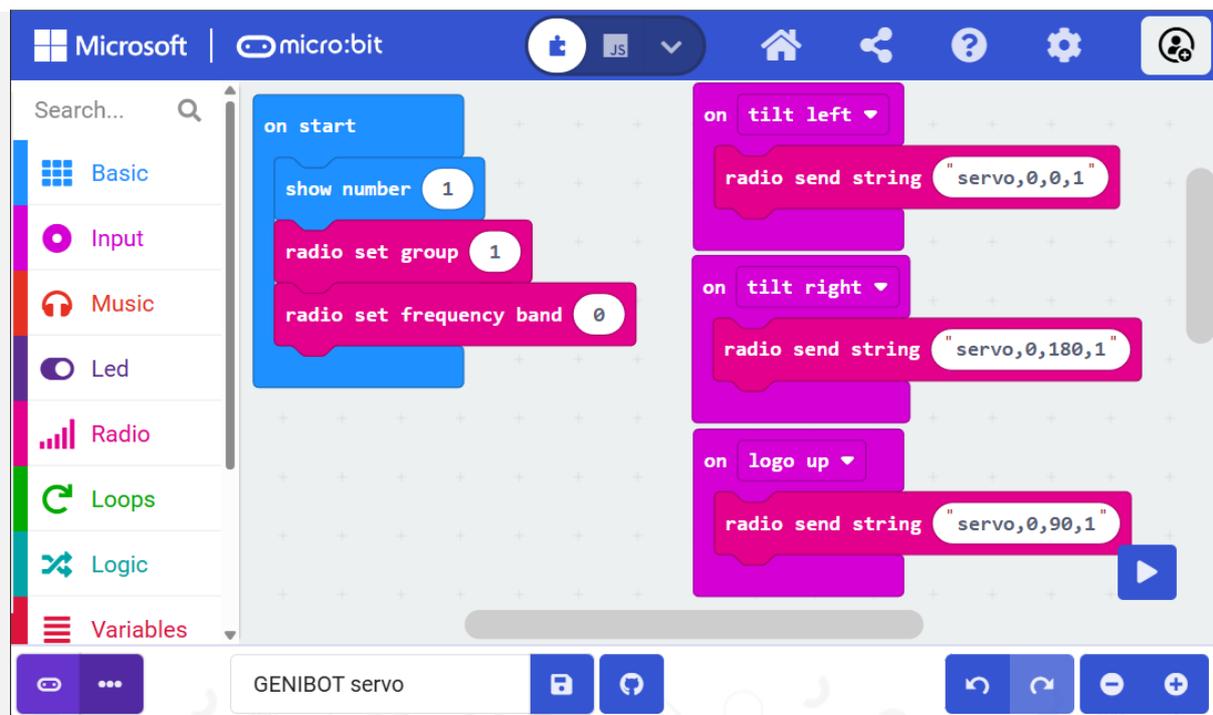
brightness: 1 to 8

In this example, the color of the robot's colored LED is changed by a tilt gesture.

servo

(servo,geek or micro:servo,angle,wait time)

Send servo control commands with radio send string block to the robot when BBC micro:bit status is tilt left, tile right or logo up.



Example: (servo,geek or micro:servo,angle,wait until)

servo type: geek servo 0, micro:servo 1

angle in degrees: geek -45 to 225, micro:servo 0 to 180

wait until in seconds: 1s to 10s

In this example, the rotation angle of a servo motor connected to the robot is controlled using a tilt gesture.

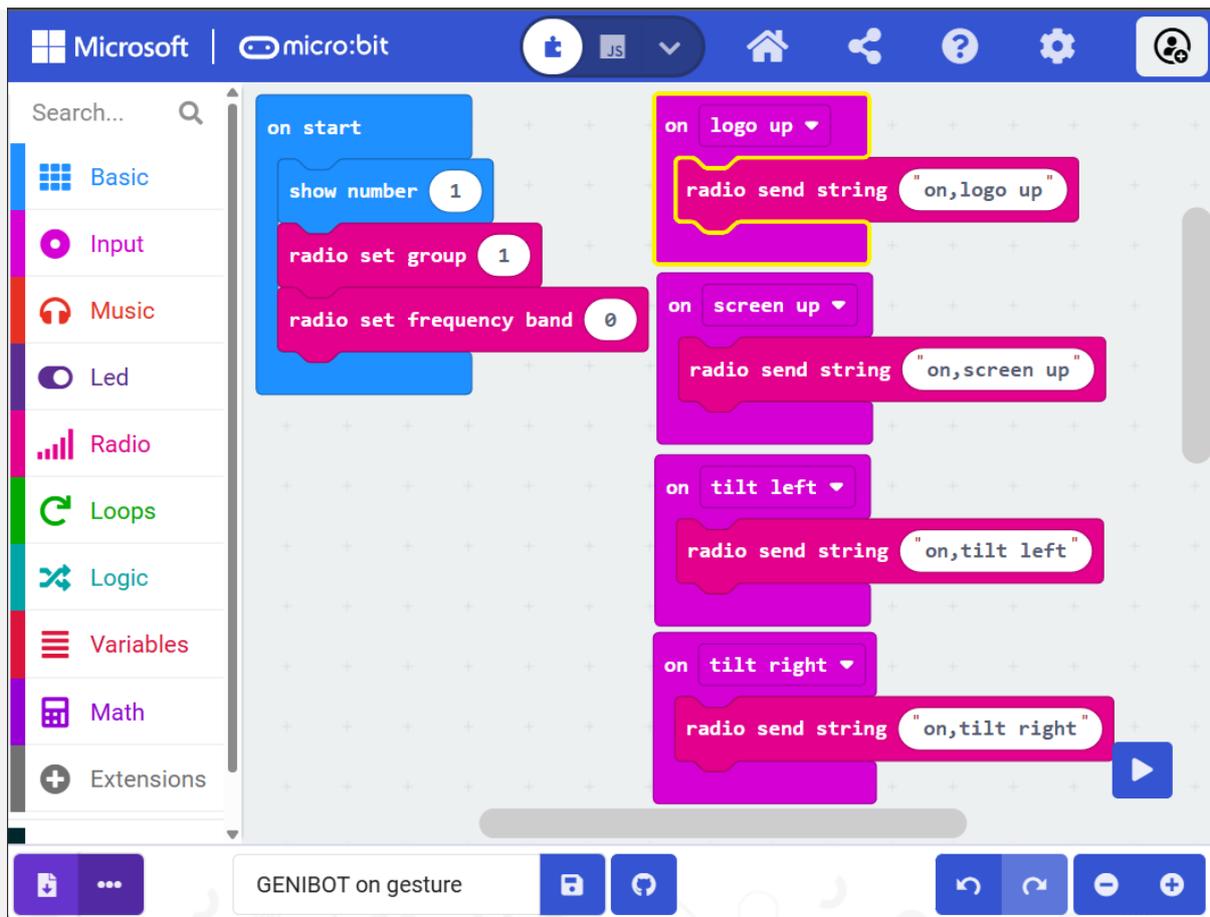
Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

on

(on,gesture)

Send gesture parameters with radio send string block to the robot when BBC micro:bit gesture event such as shake, logo up, logo down, screen up, screen down, tilt left or tilt right occurs.



Example: (on,gesture)

gesture: shake, logo up, logo down, screen up, screen down, tilt left, tilt right

In this example, the micro:bit broadcasts on gesture values to the robot as a radio send string. The on gestures received from the micro:bit can be used for Unplugged tilt coding as well as Python programming.

How to use micro:bit for unplugged tilt coding:

You can use micro:bit instead of the robot, which does not have an accelerometer sensor.

When you press the button on the robot, the robot will start Bluetooth and flash blue. To assign the micro:bit radio group, tap on two number cards while it is flashing blue, for example the numbers 0 and 1. And then, when you press the button again, the robot will stop flashing blue and start listening to radio messages from the micro:bit.

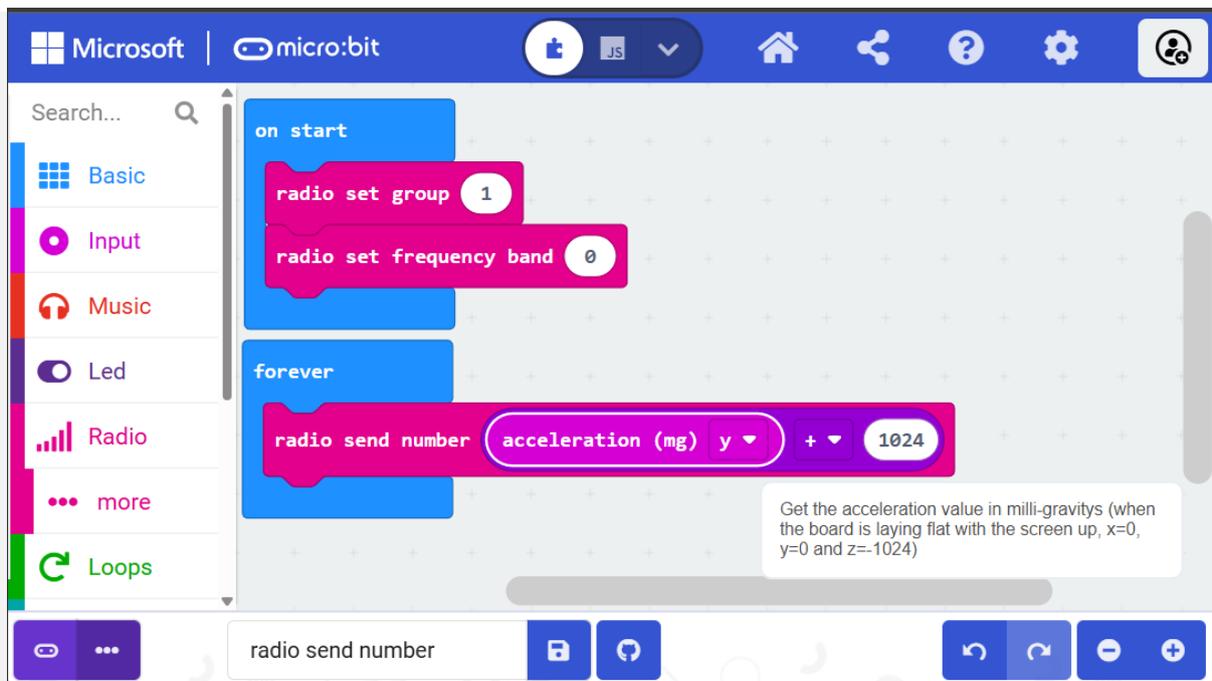
When you create a tilt gesture event with a micro:bit programmed like this example, the robot will add the unplugged card to its coding list depending on the tilt state.

Advanced course:

When using the robot's Bluetooth and micro:bit's radio together at the same time, a value sent by "radio send string" in the "on gesture" event block may be ignored in the robot's Bluetooth event task priority, so it is recommended that this MakeCode program should continue to send event values to the robot using a variable block which receives event values from the "on gesture" block.

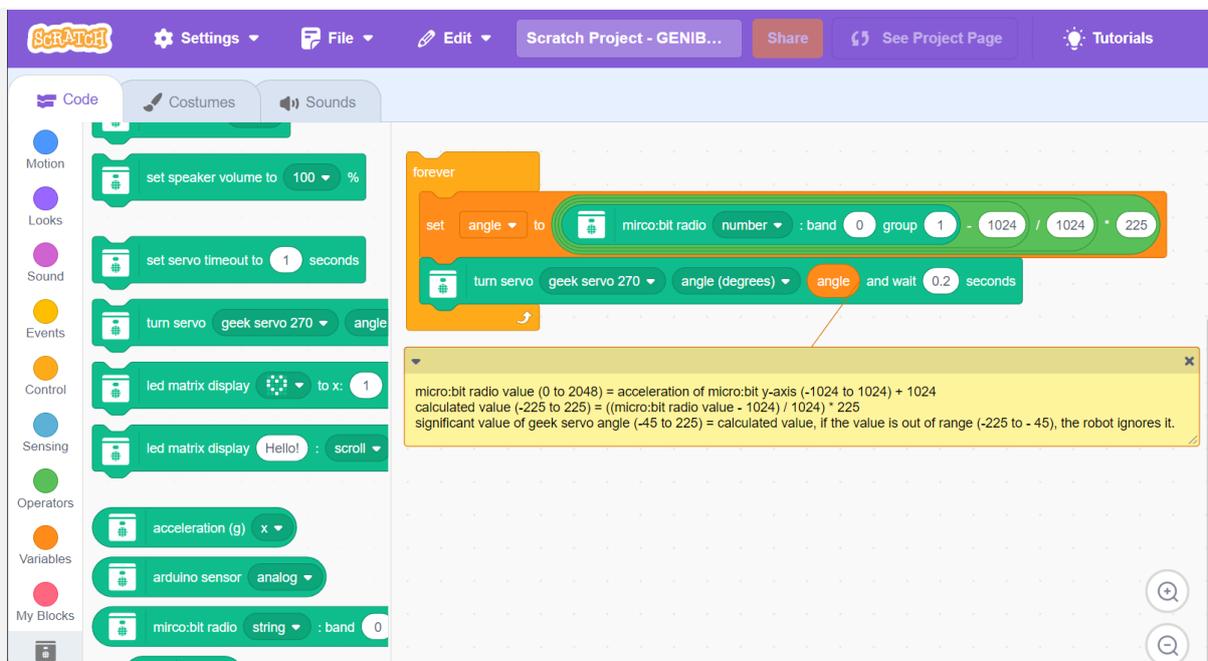
Example: Remote Control with Scratch

1. Connect BBC micro:bit to your computer. To program BBC micro:bit, go to MakeCode website <https://makecode.microbit.org/#editor>.
2. Set BBC micro:bit radio group and band, and create a program that broadcasts the y-axis acceleration value (-1024 to +1024) as a converted positive (unsigned) numeric decimal value (0 to 2048) using **radio send number** block (not use the statement syntax described previously) as follows:



Unsigned Value: In the current version of Scratch, **micro:bit radio** block only represents an unsigned numeric value, and if the value is negative received from BBC micro:bit, an additional calculation using **Operators** block to convert a signed value is required. In a future update of Scratch, this **micro:bit radio** block will be updated to also represent negative value.

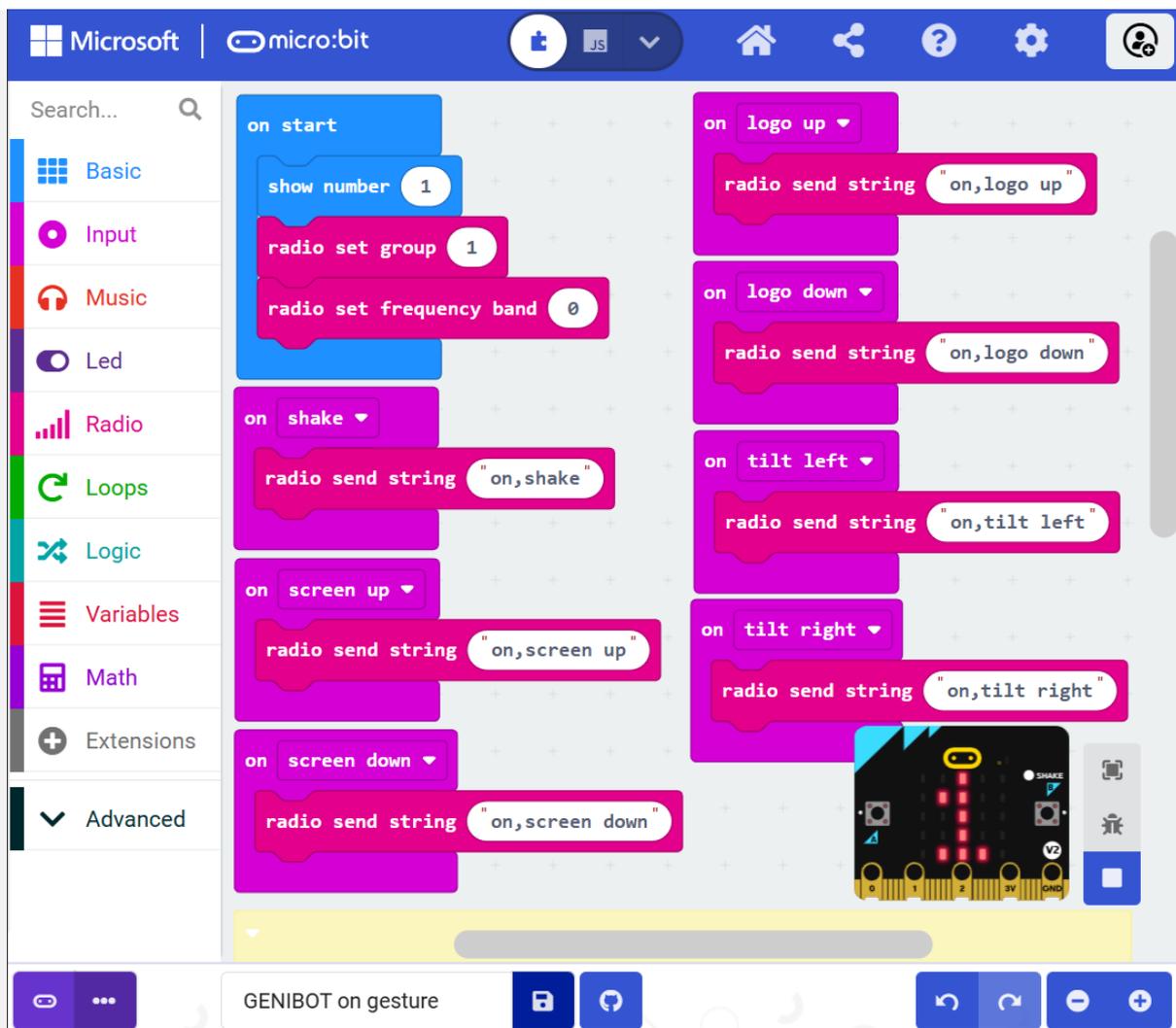
3. After completing the program as above, if you download the program to BBC micro:bit, it restarts and continuously broadcasts the unsigned numeric decimal value (0 to 2048) of the y-axis acceleration (-1024 to 1024).
4. Connect Geek Servo 270 to the robot. For connection instructions, refer to the GENIBOT Arduino Reference and Programming Guide.
5. Start Scratch Desktop app and connect the robot to the computer through Bluetooth connection using Scratch Link.
6. To control Geek Servo 270 with the y-axis acceleration value through listening to BBC micro:bit, create Scratch block program (not use the statement syntax described previously) as follows.

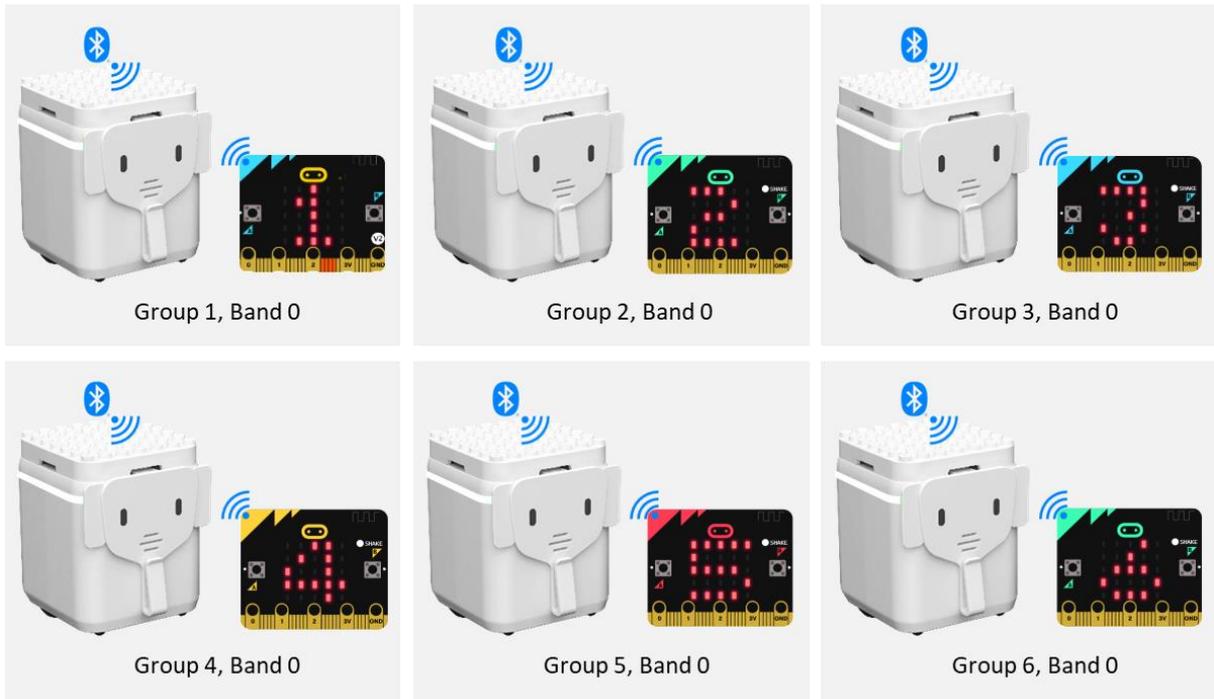


Geek Servo Angle: Converting radio value to Geek Servo angle value as:
BBC micro:bit radio value (0 to 2048) = Acceleration of micro:bit y-axis (-1024 to 1024) + 1024.
Calculated value (-225 to 225) = ((micro:bit radio value – 1024) / 1024) * 225.
Significant value of Geek Servo angle (-45 to 225) = Calculated value, if the value is out of range (-225 to -45), the robot ignores it.

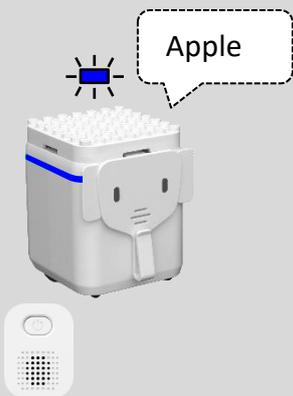
Example: Tilt Coding with “On Gesture”

1. Connect the BBC micro:bit to your computer. To program BBC micro:bit, go to MakeCode website <https://makecode.microbit.org/#editor>.
2. When there are multiple BBC micro:bits in the classroom, set the radio group number independently for each micro:bit. Only GENIBOT and BBC micro:bit with the same group number can broadcast and listen to each other, so you can avoid confusion in the classroom when multiple BBC micro:bits are broadcasting and multiple GENIBOTs are listening at the same time.
3. Create a program using “On Gesture” blocks as follows, download it to the BBC micro:bit and then start the BBC micro:bit.





- Once you have finished programming the BBC micro:bit, set the radio group number of the GENIBOT as shown in the following example. The example below sets the group number to 1. Since the group number is a two-digit number, add the number 0 in front if the number is less than 10. When setting a group number with a card, the band number defaults to the number 0.



Press the button of the robot.



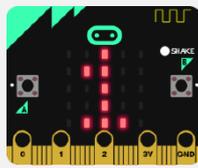
While the blue light flashes every 0.5 seconds, tap two number cards designated by the radio group number from 01 to 80.



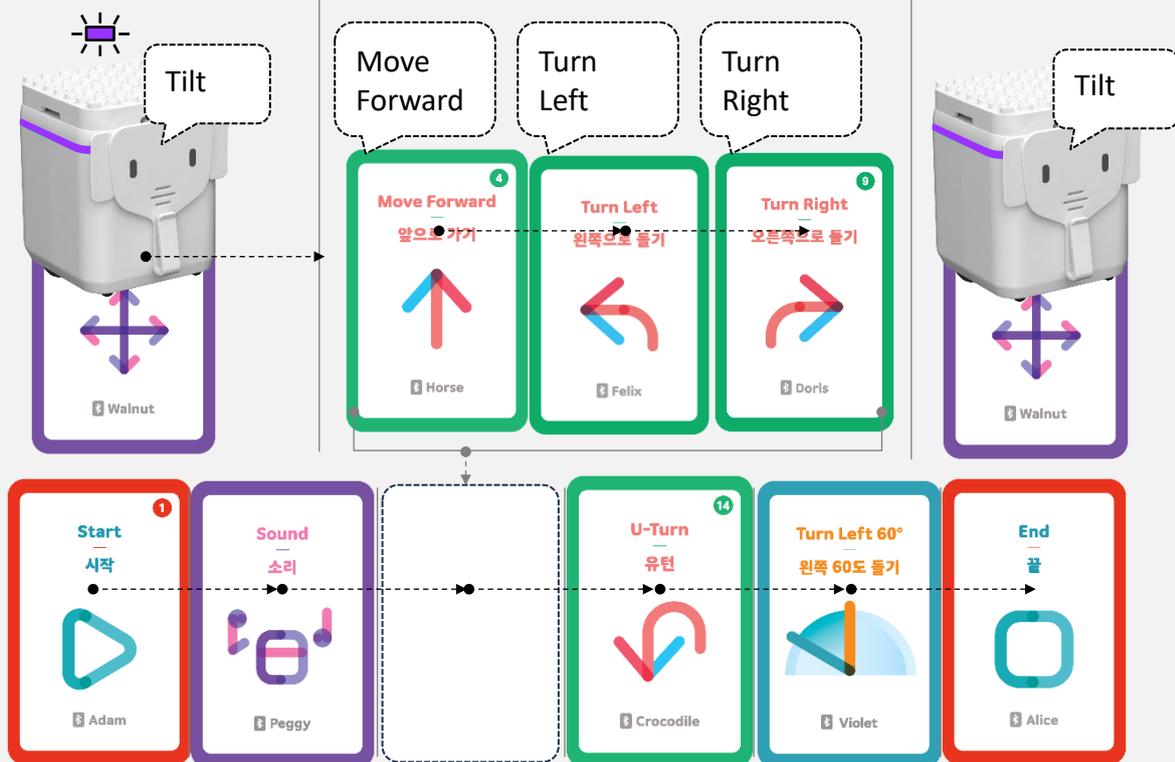
Press the button again to start listening to the BBC micro:bit.

- Once you have completed the radio settings for the BBC micro:bit and GENIBOT as above, you can do tilt coding using the micro:bit's gestures. The tilt cards that correspond to micro:bit gestures are as follows: Move Forward = “logo up”, Move Backward = “logo down”, Turn Left = “tilt left”, Turn Right = “tilt right”.

When you tap the Tilt card, it will blink purple, then you can start tilt coding.



Tap the Tilt card again to complete the tilt coding.





Bluetooth SIG: Declaration ID D052026

KC (Korea Certification): Registration No. R-R-Grb-GB1

CE (EU Integrated Standards Compulsory Certification): Declaration No. GB1-EUDOC0424L1

FCC (US Electrical and Communication Equipment Certification): FCC Identifier 2AWFA-GB1

CPC (US Consumer Product Safety): Certificate No. GB1-CPC1006L1

TELEC (Japan Radio Type Approval): Certificate No. 208-200062

ANATEL (Certification for Telecom Products in Brazil): Certificate No. 18043-2316100

GENIBOT is product and trade name of Dabida (GENIROBOT) Co., Ltd.

Other trademarks or trade names of products and company names mentioned herein are: Microsoft Windows and Make Code, Google Android, Apple iOS, BBC micro:bit Educational Foundation, Scratch Foundation and Python Software Foundation.

© 2024 Dabida Co., Ltd. All rights reserved.

GENIBOT Technical Guide

Document Edition

Part Number: GB1-TRG0318P1 First Edition, August 5, 2024

Further Documentation

Visit GENIROBOT website www.genirobot.com for additional and further technical information or up-to-date documentation.