



Robot Arm for micro:bit User Manual

CONTENT

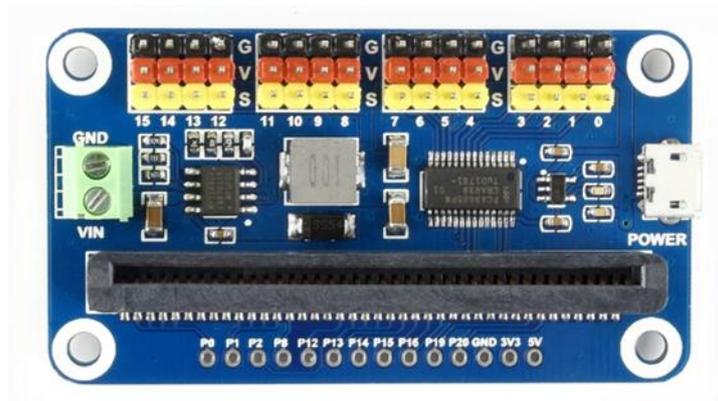
Overview	2
Hardware	2
How to use	3
Demo codes	5
microbit-Servo	5
Microbit-Servo	5
Microbit-Servo-Bluetooth	6
microbit-Servo-Radio	8

OVERVIEW

Equip your micro:bit with this 4-DOF metal robot arm, and get started to explore the world of artificial intelligent robotic.

This starter robotic arm kit includes: servo driver board, metal mechanical arm structure, and 4x metal gear servos. When used with the micro:bit, it is easy to control the arm by smart phone through Bluetooth connection.

HARDWARE



Servo Driver for micro:bit and 5V 3A power adapter are included in this kit. You need to connect the power adapter to the POWER interface to feed Servo Driver board and servos. With regular onboard, it can also offer 3.3V power for micro:bit without external power input.

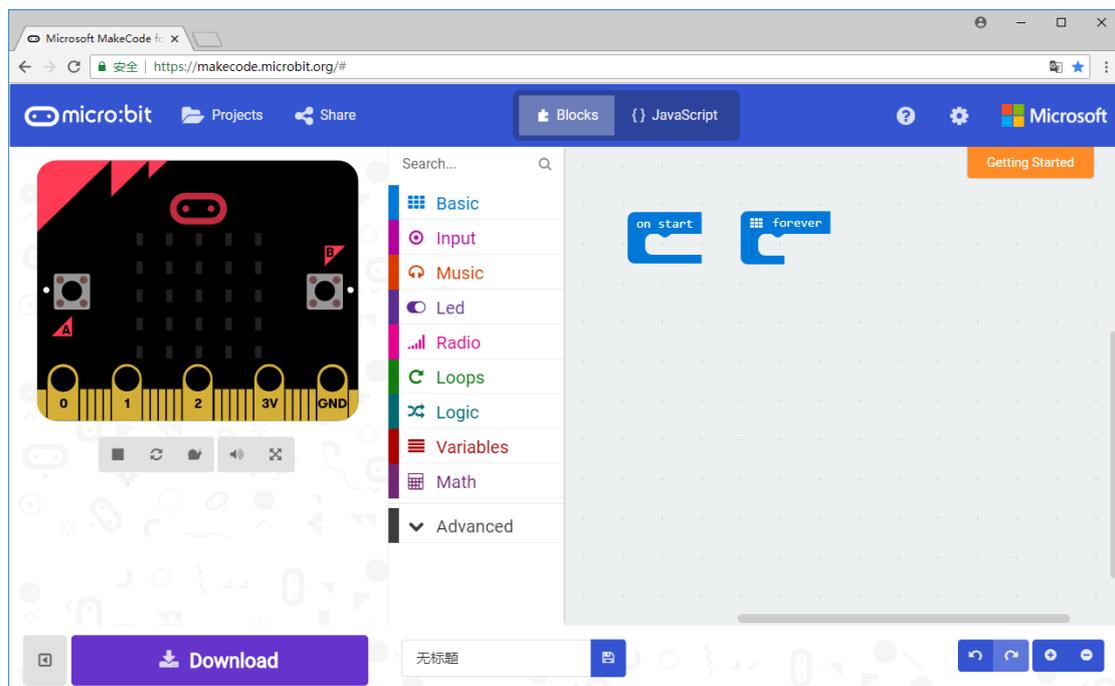
The pins on the top are servo interfaces. Black pins for GND (generally, ground pin of servo are brown), red pins for 5V, yellow pins are PWM pins. This board have channels 0~15, supports up to 16 servos work at the same time. DO NOT wrong connect servos.

【Note】

1. You cannot power the whole Robot Arm kit by micro:bit.
2. You need assemble the Robot Arm kit, before you assemble, please test if the servo work normally
3. **Please set all servos to position that rotate 90 degree before you assemble it.**

HOW TO USE

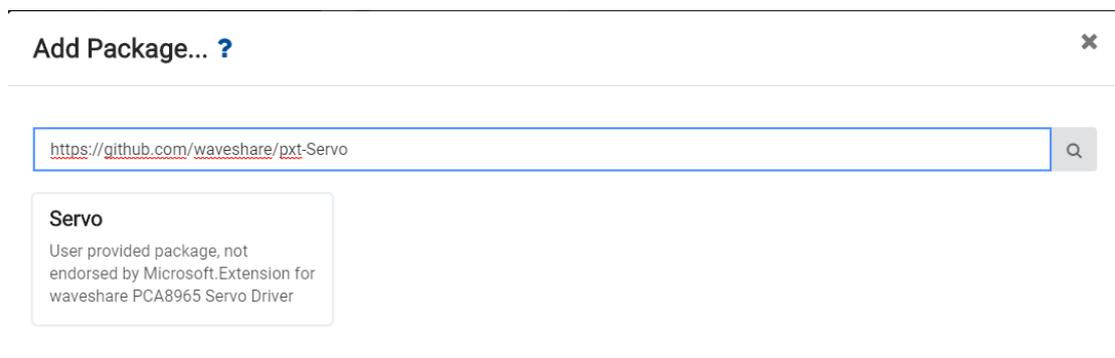
Visit micro:bit programming website in Browser: <https://makecode.microbit.org/>



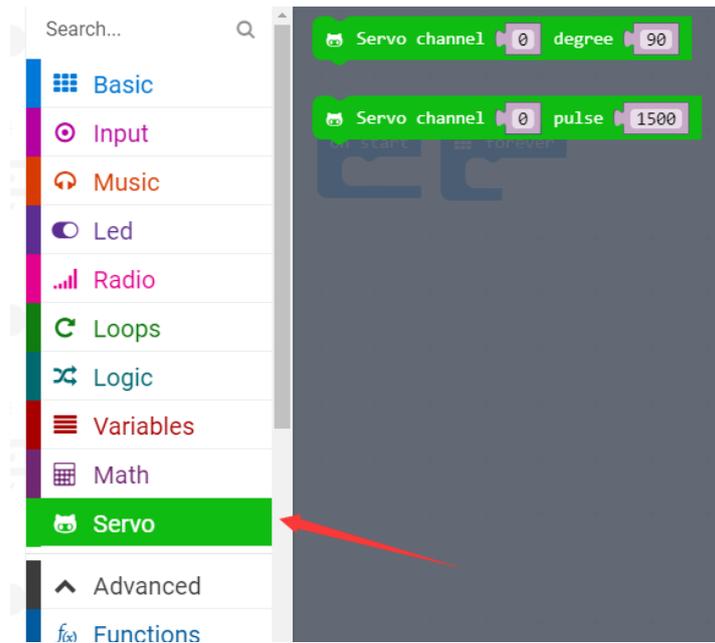
Click Projects-> New Project to create a new project

Click Advanced->Add Package and input the address: <https://github.com/waveshare/pxt-Servo>

Add the package



After adding, you can see that Servo block class appears to block area. In the class, two blocks are included.



: This block is used to control the angle of servo in range 0°~180°. Channel can be change from 0 to 15.



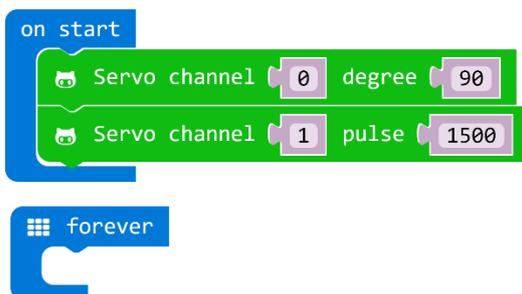
: Use this block, you can set the PWM

pulse in range: 500~2500.

The relationship between start pulse and degree is as below:

500	-----	0°
1000	-----	45°
1500	-----	90°
2000	-----	135°
2500	-----	180°

You can control servo just by setting channel, degree and pulse, it is simple.



DEMO CODES

We provide four demo codes (HEX file) for this module. You can copy them to micro:bit for testing.

Drag HEX file to the web page directly could get details information of demo code:

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

MICROBIT-SERVO

This demo code is used to rotate 16 servos in range 0~180 all the time.

```

on start
  set item to 0
  set flag to 5

forever
  if (flag + item) > 180
  then set flag to -2
  if (flag + item) < 0
  then set flag to 2
  change item by flag
  for i from 0 to 15
  do Servo channel i degree item
  pause (ms) 20
  
```

item: current angle

Flag: step length for every change. Positive value stands for co-rotating, and negative for reverse, interval is 20s.

After running the code, you can see that 16 servos turn from 0° to 180°, then turn 0° again all the time, you can adjust its speed by change the pause on code.

MICROBIT-SERVO

This demo uses A and B keys to control rotation of servo 0. It moves forward when A button is pressed and moves backward when B button is pressed.

```

on start
  set item to 0
  set flag to 5

forever
  if (flag + item) > 180
  then set flag to -2
  if (flag + item) < 0
  then set flag to 2
  change item by flag
  for i from 0 to 15
  do Servo channel i degree item
  pause (ms) 20
  
```

pos: current angle

step: step length for every change. Positive value stands for co-rotating, and negative for reverse, interval is 20s. You can change the pause time to change the rotate speed of servo.

Download the demo code to micro:bit, connect servo to channel 0, then try to control it by buttons A and B.

MICROBIT-SERVO-BLUETOOTH

microbit-Servo-Bluetooth, the function of this code is to rotate four servos 0~4 via Bluetooth.

Assemble servo to robot arm, then you can use them to control robot ram via Bluetooth.

```

on start
  set pos0 to 90
  set pos1 to 90
  set pos2 to 90
  set pos3 to 90
  Servo channel 0 degree pos0
  Servo channel 1 degree pos1
  Servo channel 2 degree pos2
  Servo channel 3 degree pos3

on bluetooth disconnected
  show string "D"

on bluetooth connected
  show string "C"

forever
  if step0 != 0
  then
    change pos0 by step0
    set pos0 to max of pos0 and 0
    set pos0 to min of pos0 and 180
    Servo channel 0 degree pos0

  if step1 != 0
  then
    change pos1 by step1
    set pos1 to max of pos1 and 0
    set pos1 to min of pos1 and 180
    Servo channel 1 degree pos1

  if step2 != 0
  then
    change pos2 by step2
    set pos2 to max of pos2 and 0
    set pos2 to min of pos2 and 180
    Servo channel 2 degree pos2

  if step3 != 0
  then
    change pos3 by step3
    set pos3 to max of pos3 and 0
    set pos3 to min of pos3 and 180
    Servo channel 3 degree pos3

  pause (ms) 30

on event
  from MES_DPAD_CONTROLLER_ID
  with value MICROBIT_EVT_ANY
  if event value = MES_DPAD_BUTTON_A_DOWN
  then
    set step0 to 5
  else if event value = MES_DPAD_BUTTON_A_UP
  then
    set step0 to 0
  else if event value = MES_DPAD_BUTTON_B_DOWN
  then
    set step0 to -5
  else if event value = MES_DPAD_BUTTON_B_UP
  then
    set step0 to 0
  else if event value = MES_DPAD_BUTTON_C_DOWN
  then
    set step1 to 5
  else if event value = MES_DPAD_BUTTON_C_UP
  then
    set step1 to 0
  else if event value = MES_DPAD_BUTTON_D_DOWN
  then
    set step1 to -5
  else if event value = MES_DPAD_BUTTON_D_UP
  then
    set step1 to 0
  else if event value = MES_DPAD_BUTTON_3_DOWN
  then
    set step2 to 5
  else if event value = MES_DPAD_BUTTON_3_UP
  then
    set step2 to 0
  else if event value = MES_DPAD_BUTTON_4_DOWN
  then
    set step2 to -5
  else if event value = MES_DPAD_BUTTON_4_UP
  then
    set step2 to 0
  else if event value = MES_DPAD_BUTTON_1_DOWN
  then
    set step3 to 5
  else if event value = MES_DPAD_BUTTON_1_UP
  then
    set step3 to 0
  else if event value = MES_DPAD_BUTTON_2_DOWN
  then
    set step3 to -5
  else if event value = MES_DPAD_BUTTON_2_UP
  then
    set step3 to 0
    
```

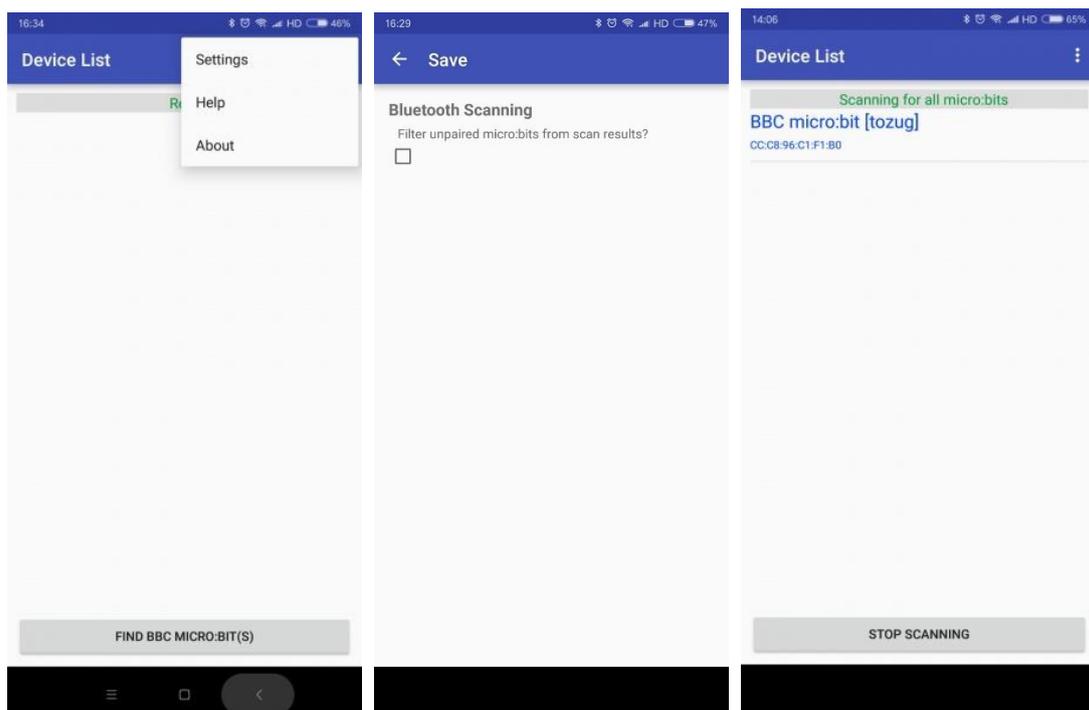
This is the Bluetooth remote control code. pos0~pos3 are current degrees of channel 0~3.
step0~step3 are step length.

When command are received by micro:bit from app, micro:bit will change the step as received to control servo. If step is not equal to 0, change the pos and let servo move. (Note that the value of pos should in: 0~180)

To use this demo code, you should first install APP for Bluetooth communication. (Only support Android)

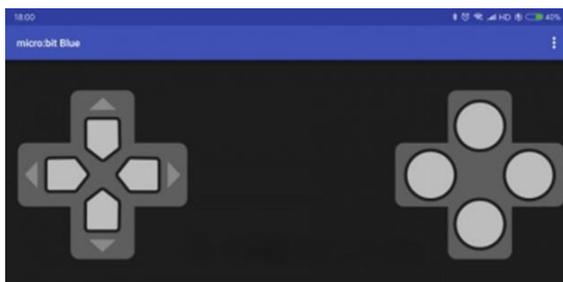
- [Bluetooth App \(for Android\)](#)

Open APP, click Settings on the right top, uncheck Filter unpaired micro:bit from scan results?

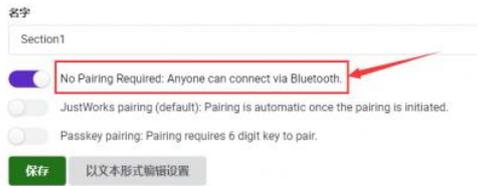


Scanning BBCC micro:bit device and connect it. After connecting, you can enter the control page by click the Joystick icon.

There are 8 buttons, could be used to control servos from channel0 to channel3.



Note: If you want to create new Bluetooth project, don't forget to set project that, choose No Pairing Required: Anyone can connect via Bluetooth.



MICROBIT-SERVO-RADIO

This code we use 2.4Ghz RF function of micro:bit. To test this code, you need two micro:bit, one is connected to Joystick for micro:bit as sender, and another connected to Servo Driver for micro:bit as receiver which will control robot.



Code if sender:

```

on start
  show icon [Joystick]
  radio set group [1]
  Joystickinit

forever
  if DIR Dir U
  then
    radio send string ["U"]
    while DIR Dir U
    do
      radio send string ["N"]
  else if DIR Dir D
  then
    radio send string ["D"]
    while DIR Dir D
    do
      radio send string ["N"]
  else if DIR Dir L
  then
    radio send string ["L"]
    while DIR Dir L
    do
      radio send string ["N"]
  else if DIR Dir R
  then
    radio send string ["R"]
    while DIR Dir R
    do
      radio send string ["N"]
  else if Key C Press
  then
    radio send string ["1"]
    while Key C Press
    do
      radio send string ["N"]
  else if Key D Press
  then
    radio send string ["2"]
    while Key D Press
    do
      radio send string ["N"]
  else if Key E Press
  then
    radio send string ["3"]
    while Key E Press
    do
      radio send string ["N"]
  else if Key F Press
  then
    radio send string ["4"]
    while Key F Press
    do
      radio send string ["N"]
  
```

Code of Receiver:

```

on start
  radio set group 1
  set pos0 to 0
  set pos1 to 0
  set pos2 to 0
  set pos3 to 0
  Servo channel 0 degree pos0
  Servo channel 1 degree pos1
  Servo channel 2 degree pos2
  Servo channel 3 degree pos3

on radio received receivedString
  if receivedString = "U"
  then set step0 to 5
  else if receivedString = "D"
  then set step0 to -5
  else if receivedString = "L"
  then set step1 to 5
  else if receivedString = "R"
  then set step1 to -5
  else if receivedString = "1"
  then set step2 to 5
  else if receivedString = "3"
  then set step2 to -5
  else if receivedString = "2"
  then set step3 to 5
  else if receivedString = "4"
  then set step3 to -5
  else if receivedString = "N"
  then set step0 to 0
  set step1 to 0
  set step2 to 0
  set step3 to 0

forever
  if step0 != 0
  then change pos0 by step0
  set pos0 to max of pos0 and 0
  set pos0 to min of pos0 and 180
  Servo channel 0 degree pos0

  if step1 != 0
  then change pos1 by step1
  set pos1 to max of pos1 and 0
  set pos1 to min of pos1 and 180
  Servo channel 1 degree pos1

  if step2 != 0
  then change pos2 by step2
  set pos2 to max of pos2 and 0
  set pos2 to min of pos2 and 180
  Servo channel 2 degree pos2

  if step3 != 0
  then change pos3 by step3
  set pos3 to max of pos3 and 0
  set pos3 to min of pos3 and 180
  Servo channel 3 degree pos3

  pause (ms) 30
  
```

Like Bluetooth code, pos0~pos3 are current degree of servos from channel 0 to channel 3, step0~step3 are step length.

You should flash sender code to micro:bit which connect Joystick module and flash receiver one to servo module. Then you can control servo by module Joystick for micro:bit.

For more details about Bluetooth and 2.4G, you can refer to the user guide of KitiBot for micro:bit:

https://www.waveshare.com/wiki/Chapter_8_of_KitiBot-Microbit