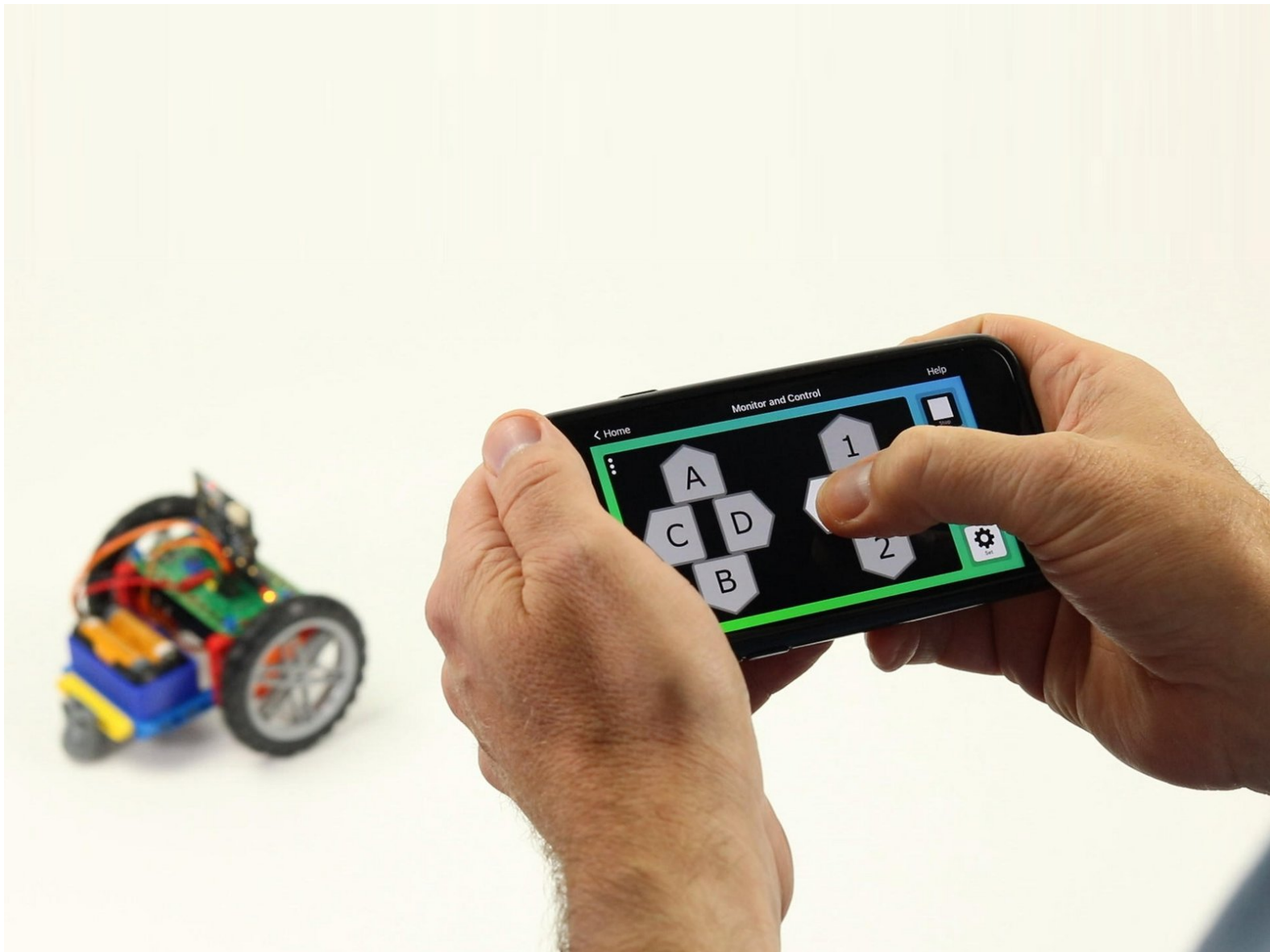




Rover Bluetooth Control

Control your Rover using Bluetooth from a phone or tablet.

Written By: Pete Prodoehl



INTRODUCTION

Control your Rover using Bluetooth from a phone or tablet.

Using a GamePad (or "DPAD") controller you can make the Rover forward & backward, and spin in either direction. You can also open and close the Gripper and adjust the speed of the Rover.

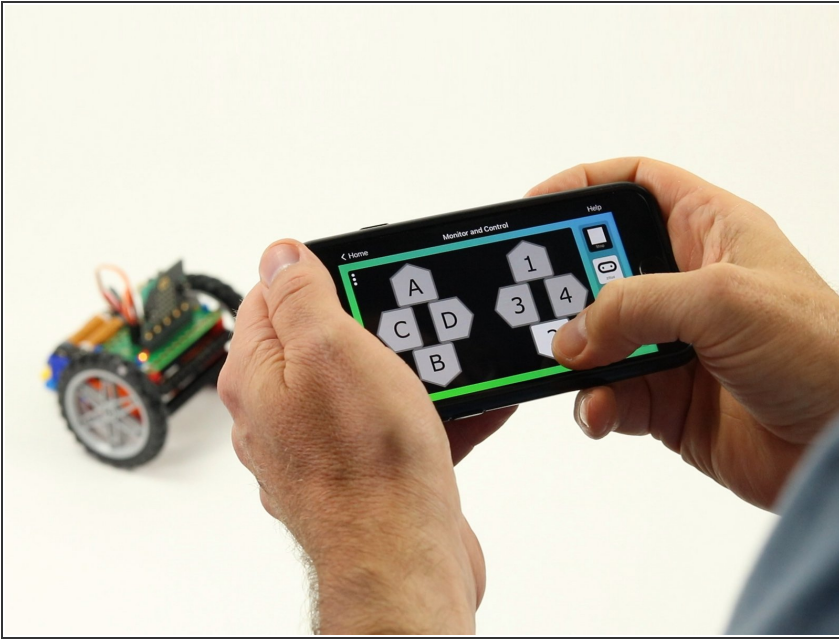
 TOOLS:

- [Computer](#) (1)

 PARTS:

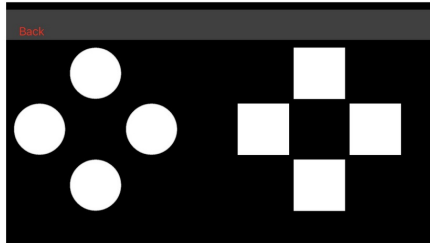
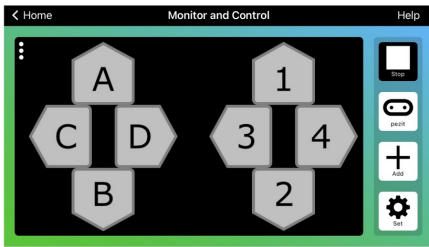
- [Bit Board Rover Kit](#) (1)
- [micro:bit](#) (1)
- [Phone or Tablet](#) (1)

Step 1 — Bluetooth Control



- You can control your Rover using Bluetooth from a phone or tablet along with an app that supports the micro:bit's Bluetooth Messaging Event System.
- If you are using an iPhone or iPad you can use the GamePad (or "DPAD") controller found in the "Monitor & Control" section of the micro:bit iOS app.
- Find the iOS app here:
[https://apps.apple.com/gb/app/micro-bit/...](https://apps.apple.com/gb/app/micro-bit/)
- If you are using an Android phone or tablet you can use "Bitty Controller" which has its own "DPAD" controller option.
- Find Bitty Controller in the Google Play store here:
<https://play.google.com/store/apps/details?id=com.bittysoftware.bittycontroller>
- Find more information about Bitty Controller here:
<https://bittysoftware.blogspot.com/p/app...>

Step 2 — The DPAD



- ⓘ The GamePad may look familiar if you've ever used a video game controller.
- The GamePad will control the Rover in the following manner:
 - 1 will move forward, 2 will move backwards.
 - 3 will spin to the left, 4 will spin to the right.
 - C will open the gripper, D will close the gripper.
 - A will increase speed, B will decrease speed.
- ★ Keep in mind you can change what any of these controls do by editing the code.

```
sequenceDiagram
    participant BT as Bluetooth module
    participant MC as Motor Controller
    BT->>MC: on start
    Note over BT,MC: Bluetooth to pin servo
    Note over BT,MC: Bluetooth button service
    BT->>MC: set speed = to 10
    BT->>MC: set angleClosed = to 185
    BT->>MC: servo write pin P15 (write only) = to angleOpen
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_1_DOWN =
    MC->>MC: spin other way pin P14 (write only) = with speed speed =
    MC->>MC: spin one way pin P13 (write only) = with speed speed =
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_1_UP =
    MC->>MC: turn off motor at pin P14 =
    MC->>MC: turn off motor at pin P13 =
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_2_DOWN =
    MC->>MC: spin one way pin P14 (write only) = with speed speed =
    MC->>MC: spin one way pin P13 (write only) = with speed speed =
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_2_UP =
    MC->>MC: turn off motor at pin P14 =
    MC->>MC: turn off motor at pin P13 =
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_3_DOWN =
    MC->>MC: change speed = to 20
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_3_UP =
    MC->>MC: change speed = to 10
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_4_DOWN =
    MC->>MC: servo write pin P15 (write only) = to angleOpen
    MC->>MC: on event from MES_SPAD_CONTROLLER_ID = with value MES_SPAD_BUTTON_4_UP =
    MC->>MC: servo write pin P15 (write only) = to angleClosed
```

The diagram illustrates the control logic for a 4-wheel robot using a Bluetooth module and a motor controller. It is structured as a sequence of events and actions:

- on start:** The Bluetooth module sends commands to the motor controller: "Bluetooth to pin servo", "Bluetooth button service", "set speed = to 10", "set angleClosed = to 185", and "servo write pin P15 (write only) = to angleOpen".
- Event 1 (MES_SPAD_BUTTON_1_DOWN):** Triggered by the motor controller, it performs "spin other way pin P14 (write only) = with speed speed =" and "spin one way pin P13 (write only) = with speed speed =".
- Event 2 (MES_SPAD_BUTTON_1_UP):** Triggered by the motor controller, it performs "turn off motor at pin P14 =" and "turn off motor at pin P13 =".
- Event 3 (MES_SPAD_BUTTON_2_DOWN):** Triggered by the motor controller, it performs "spin one way pin P14 (write only) = with speed speed =" and "spin one way pin P13 (write only) = with speed speed =".
- Event 4 (MES_SPAD_BUTTON_2_UP):** Triggered by the motor controller, it performs "turn off motor at pin P14 =" and "turn off motor at pin P13 =".
- Event 5 (MES_SPAD_BUTTON_3_DOWN):** Triggered by the motor controller, it performs "change speed = to 20".
- Event 6 (MES_SPAD_BUTTON_3_UP):** Triggered by the motor controller, it performs "change speed = to 10".
- Event 7 (MES_SPAD_BUTTON_4_DOWN):** Triggered by the motor controller, it performs "servo write pin P15 (write only) = to angleOpen".
- Event 8 (MES_SPAD_BUTTON_4_UP):** Triggered by the motor controller, it performs "servo write pin P15 (write only) = to angleClosed".

ⓘ Note: This code will work with either of the apps mentioned in Step 1.

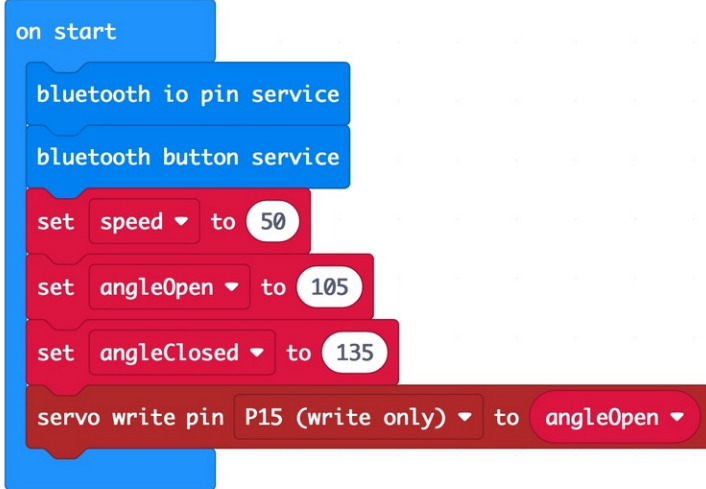
The image consists of two side-by-side screenshots of the BBC micro:bit Bitty Controller app interface.

The left screenshot shows the 'Choose micro:bit' screen. At the top, there's a status bar with 'Search', signal strength, time '9:31 AM', and battery level '88%'. Below the status bar is a navigation bar with 'Home', 'Choose micro:bit', and 'Help'. The main content area has a green-to-blue gradient. It features a section titled 'Currently selected micro:bit' which contains a visual representation of a micro:bit labeled 'pezit'. Below this, there are two paragraphs of text: 'If you want to use a new micro:bit, tap the button below.' and 'If you want to remove a pairing from a micro:bit, go to the Bluetooth section in your device Settings.' At the bottom left, it says 'Having problems? Try the Help page.' and at the bottom right, there's a large white micro:bit icon. A black button at the very bottom says 'Pair a new micro:bit' next to a small micro:bit icon.

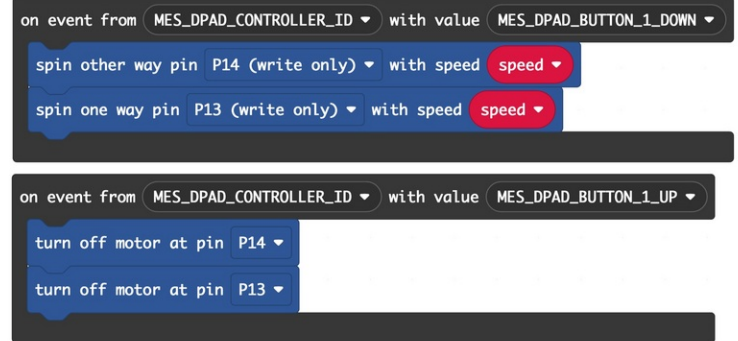
The right screenshot shows the 'Bitty Controller' screen. It has a black background. At the top, the title 'Bitty Controller' is in red. Below the title are three red buttons: 'Scan', 'Options', and 'Help'. A yellow star icon is positioned to the left of a red horizontal bar that contains the text 'BBC micro:bit [pezit]' and the ID '0D9894C0-E880-9D00-2AFB-0AC67AF75261' below it. The rest of the screen is black.

- You'll need to pair the phone or tablet with the micro:bit in your Rover.
- Please see the extensive guide to pairing provided on the [makecode.microbit.org](https://makecode.microbit.org/v0/refs/Makecode%20Microbit%20Rover%20Pairing%20Guide) web site: <https://makecode.microbit.org/v0/refs/Makecode%20Microbit%20Rover%20Pairing%20Guide>
- Once you pair your device with the micro:bit on the Rover you can use the GamePad to control it!

Step 5 — Take it Further - Exploring the Code



```
on start
  bluetooth io pin service
  bluetooth button service
  set speed to 50
  set angleOpen to 105
  set angleClosed to 135
  servo write pin P15 (write only) to angleOpen
```



```
on event from MES_DPAD_CONTROLLER_ID with value MES_DPAD_BUTTON_1_DOWN
  spin other way pin P14 (write only) with speed speed
  spin one way pin P13 (write only) with speed speed

on event from MES_DPAD_CONTROLLER_ID with value MES_DPAD_BUTTON_1_UP
  turn off motor at pin P14
  turn off motor at pin P13
```

- The **on start** block has two commands to enable the Bluetooth services we need.
- The rest of the code in the **on start** block is just like other Rover code.
- The other blocks are all on event blocks, which are looking for data from the **MES_DPAD_CONTROLLER_ID** with a specific value.
- Besides the on start block, each dark gray block looks for a button to be down or up, and reacts appropriately.
- While these **Control** blocks look a bit more complex than other blocks, they are pretty easy to explore and use in your code.