

# RTBobble Circuit Playground Bluefruit

Last modified 4/21/2022

## Connecting RTBobble Circuit Playground Bluefruit

A Bluetooth Low Energy (BLE) API compatible browser needs to be used:

- Chrome on Windows Laptop
- Chrome on Apple laptop
- BlueFy on Apple iPhone or iPad device
- Chrome on Android device

Make sure that Bluetooth is enabled on your connecting device.

Apply power to the RTBobble Circuit Playground Bluefruit with either an optional battery pack or powered USB cable. The RTBobble should have a blinking tiny red LED while waiting for connection.

Go to page:

<https://www.reachandteach.com/test52.html>

Click Connect and select the device RTBobbleCPX

To test, Click on R, O, Y, G, B, V, Rainbow, Blinking LEDs which should cause the neopixel LEDs to light. Click on LEDs OFF to turn off the LEDs

## Recipes

Below are some sample command line recipes that can be copied and pasted into the Execute Command field of the test52 web page.

Touch sensor, turn neopixels to red when touched

```
. 3 X 300 400 ? 0 9 700 N ) 4000 n ( !
```

Light sensor

```
. 2 X 500 600 ? 0 9 700 N ( 1040 n ) . !
```

Light sensor w/ digital out

```
. 2 X 500 600 ? + ) - ( !
```

Temperature sensor, turn neopixels to red when the temperature goes above 86 degrees F

```
. 1 X 855 860 ? 0 9 0 N ( 0 9 700 N ) . !
```

Red, White, and Blue (with pattern stored in neopixel storage 1)

```
0 9 007 N 0 3 700 N 4 6 555 N 1 P
```

Jumping Jack  
. 0 s ( 85 s ) 3500 \*

Touch sensor servo out and led display  
. 3 X 0 90 S ( 0 90 S ) 100 \*

Light sensor servo out and led display  
. 2 X 0 90 S ( 0 90 S ) 100 \*

Turn an attached device ON and OFF in 5 second intervals  
. + ( - ) 5000 \*

## RTBobble Bluefruit Commands

The RTBobble responds to one letter commands which are defined in this section. This makes it easy to control the RTBobble with any Bluetooth BLE UART interface. NOTE: There should be at least one space between each command. Some of these commands take numerical parameters which should come before the one letter command.

+

Sets pad A4 HIGH and sets neopixels with 0 M

-

Sets pad A4 LOW and sets neopixels with 1 M

z (Zap)

Set pad A4 HIGH then LOW with a 700ms delay

F (Forward)

Sets pad A5 HIGH and pad A6 LOW and sets neopixels with 3 M

R (Reverse)

Sets pad A5 LOW and pad A6 HIGH and sets neopixels with 2 M

*angle s*

Sets servo on pad A1 to the angle passed on the stack

*min max S*

Sets servo on pad A1 based on sensor input scaled between min and max angles passed on the stack

## *patnum n*

Sets and plays a neopixel pattern passed on the stack

1 to 9 : red

10 to 19: red, green

20 to 29: red, green, blue

30 to 34: red, green, blue, blue-green

35 to 999: red, green, blue, blue-green, yellow

1000 to 1999: above color pattern along with random blinking white lights

2000 to 2050: rainbow light chase pattern

3000 to 3050: rainbow light sticky pattern

4000 to 4999 (in form 4rgb): chase light pattern where color is combination of r (red), g (green), b (blue). Example: 4700 is red; 4070 is green; 4555 is white

## *start stop color N*

Set a color of a group of neopixel lights passed on the stack as <start> <stop> <color> where color is a 3 digit number representing red, green, and blue. Start and stop are neopixel numbers which can range from 0 to 49 but for the neopixels for the RTBobble Bluefruit are usefully 0 to 9.

Some sample color numbers:

- 700 Red
- 710 Orange
- 730 Yellow
- 070 Green
- 007 Blue
- 405 Violet
- 555 White

## *num P*

Memory store current neopixel pattern to be recalled with M command. Up to 3 patterns can be saved as 0, 1, 2, or 3.

## *num M*

Memory recall neopixel pattern previously saved with P command. Num is 0, 1, 2, or 3.

## *sensnum X*

Set the sensor input based on sensnum passed on the stack (0: analog port A2, 1: temperature, 2: light, 3: cap sense on A3), 4: detect push button (left and right), 5: detect tilt. All sensor inputs are automatically scaled between 0 and 1023. For the temperature sensor, the value is the temperature in Fahrenheit \* 10 so for example, a value of 656 is 65.6 degrees. WARNING: The voltage on analog port A2 should be 3.3 V or below.

## *X*

Reads an attached sensor voltage based on sensor input selected using the X command.

>

Sets high trigger value to the last sensor reading taken with 'x' command.

<

Sets low trigger value to the last sensor reading taken with 'x' command.

*lowvalue highvalue ?*

Set low and high trigger values as passed on the stack.

^

Sets low and high trigger for digital threshold

*cmd )*

Sets command triggered by high trigger value to last command issued

*cmd (*

Sets command triggered by low trigger value to last command. Example: 0 s ( 90 s ) will set the low trigger command to set servo to 0 degrees and the high trigger command to set servo to 90 degrees.

*timems \**

Auto trigger low and high trigger commands based on time in msec passed on the stack.

Example: 0 s ( 90 s ) 3500 \* will automatically cause the servo to move back and forth between 0 and 90 degrees with a 3500 msec (3.5 seconds) delay.

! (exclamation mark)

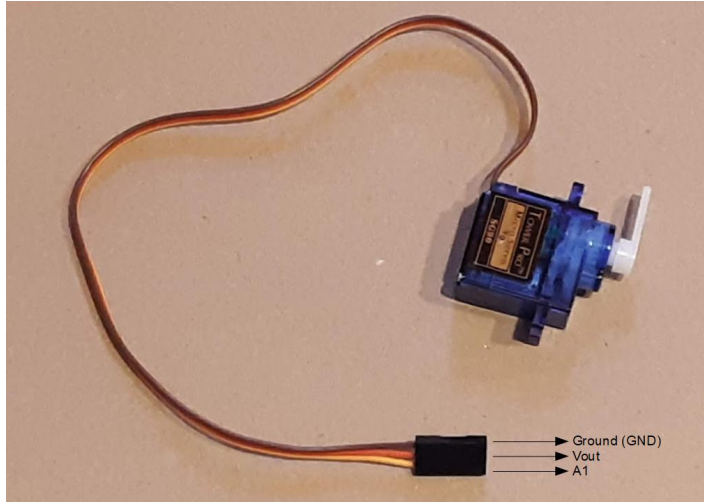
Enable sensor triggering. Example: 2 X 300 400 ? 0 s ( 90 s ) ! will select the light sensor input ('2 X') and configure a low trigger value of 300 and a high trigger value of 400. The low trigger command is configured for setting the servo at 0 degrees. The high trigger command is configured for setting the servo to 90 degrees. '!' activates sensor triggering. When the light input value goes above 400, the servo is set to 90 degrees. When the light input value falls below 300, the servo is set to 0 degrees.

. (period)

Disable sensor triggering and auto triggering

# RTBobble Bluefruit Test Setups

## Attaching a Servo



A servo has three wires which are defined as follows:

- Yellow or white: servo control
- Red: power for the servo
- Brown or black: servo ground

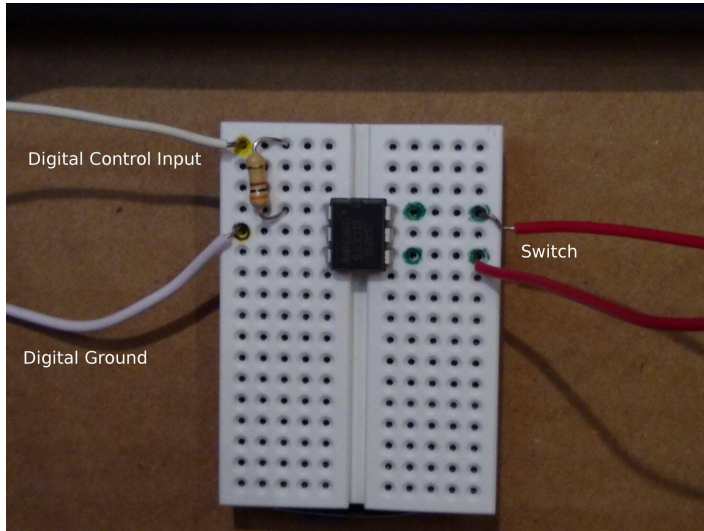
The servo is attached to the RTBobble Bluefruit as follows:

- Yellow or white servo control is connected to pad A1
- Red servo power is connected to pad Vout
- Brown or black servo ground is connected to any Gnd Pad

Test command: **0 s ( 90 s ) 2000 \***

This will cause the servo to cycle between 0 and 90 degrees with a 2 second interval.

## Attaching a Solid State Relay board

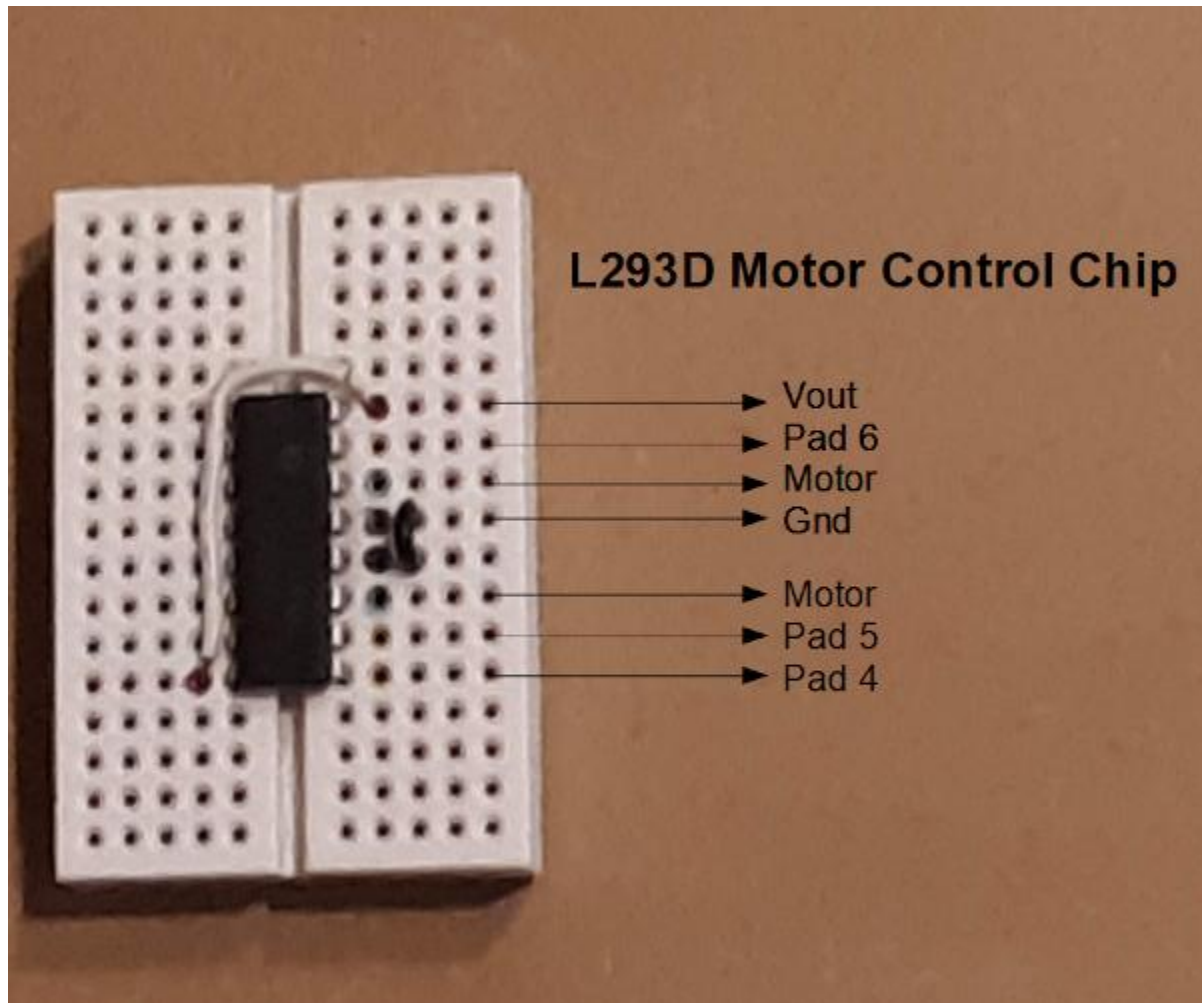


A Clare LCA110 solid state relay chip allows you to switch any low power (battery powered) circuit ON and OFF using the RTBobbie Bluefruit. Connect the solid state relay as shown in the diagram. The Digital Control Input is attached to pad A4. The Digital Ground Input is attached to any of the Gnd pads. The switch output connections will replace the switch in your circuit.

Test command: - ( + ) **3000** \*

This will cause the circuit to switch on and off with a 3 second interval.

## Connecting a Motor using a L293D chip



Connect the Circuit Playground Bluefruit's Vout, Ground (Gnd), Pad 4, Pad 5, and Pad 6 to the L293D as shown. Connect the motor to the two pins labeled Motor in the photo. The "+" command will turn the motor ON. The "-" command will turn the motor OFF. "F" will move the motor in the forward direction. "R" will move the motor in the reverse direction. "Z" (Zap) will turn the motor ON for 750 msecs and then turn the motor OFF.

Test command: **+ F ( R ) 2000 \***

This turns the motor ON and then runs the motor in the forward and reverse directions in a 2 second interval.

