

Light Play 2 Blocks for Scratch

Version 0.2 of the Light Play hardware can control 3 rainbow lights and 1 motor. The board has a one input for a resistive sensor, which can either be a pair of alligator clips or a light sensor.

Light Blocks

turn on [all lights, light 1, light 2, light 3]

turn off [all lights, light 1, light 2, light 3]

toggle [all lights, light 1, light 2, light 3]

set [all lights, light 1, light 2, light 3] **color to** [red, orange, yellow, green, blue, purple, white, surprise]

fade [all lights, light 1, light 2, light 3] **to** [red, orange, yellow, green, blue, purple, white, surprise]

fade in [all lights, light 1, light 2, light 3]

fade out [all lights, light 1, light 2, light 3]

set fade speed [slow, faster, fastest]

Motor Blocks

turn on motor

turn off motor

reverse motor

toggle motor

set motor speed [slow, faster, fastest]

Sensor Blocks

when [clips are connected, clips are disconnected, shadow falls on sensor, light shines on sensor]

Byte Codes

All light and motor commands are encoded in a single byte sent from Scratch to Arduino, according to the following scheme:

Commands

[u u u x x y y]

upper bits set command type:

000 = one argument light commands

001 = motor commands

010 = fade light color to

011 = other/expansion

100 = set light color to

One argument light commands

[0 0 0 x x y y]

the x bits select which light:

0 0 = all lights

0 1 = light 1

1 0 = light 2

1 1 = light 3

the y bits select which other light command:

0 0 0 = **turn on light**

0 0 1 = **turn off light**

0 1 0 = **fade in light**

0 1 1 = **fade out light**

1 0 0 = **toggle light**

1 0 1 = **set brightness low**

1 1 0 = **set brightness medium**

1 1 1 = **set brightness high**

Two argument light commands

set color to

[1 0 0 x x y y]

the x bits select which light:

0 0 = all lights

0 1 = light 1

1 0 = light 2

1 1 = light 3

the y bits select which color:

0 0 0 = red

0 0 1 = orange

0 1 0 = yellow

0 1 1 = green

1 0 0 = blue
1 0 1 = purple
1 1 0 = white
1 1 1 = surprise

fade to

[0 1 0 x x y y y]

the x bits select which light:

0 0 = all lights
0 1 = light 1
1 0 = light 2
1 1 = light 3

the y bits select which color:

0 0 0 = red
0 0 1 = orange
0 1 0 = yellow
0 1 1 = green
1 0 0 = blue
1 0 1 = purple
1 1 0 = white
1 1 1 = surprise

Other kinds of light commands / expansion

[0 1 1 x x y y y]

the y bits select which other command

set fade speed

[0 1 1 x x 0 0 0]

x bits select speed:

0 0 = slow
0 1 = faster
1 0 = fastest

setcolor to black (iPad only)

[0 1 1 x x 0 0 1]

the x bits select which light:

0 0 = all lights
0 1 = light 1
1 0 = light 2
1 1 = light 3

fadecolor to black (iPad only)

[0 1 1 x x 0 1 0]

the x bits select which light:

0 0 = all lights

0 1 = light 1

1 0 = light 2

1 1 = light 3

reset state variables (iPad only)

[0 1 1 0 0 0 1 1]

Motor commands

[0 0 1 x x y y y]

the x bits are set to zero (it's important to avoid sending byte 0x2B, which is ASCII '+', since this is used by the Adafruit BTLE UART to switch to command mode)

the y bits select which motor command:

0 0 0 = **turn on motor**

0 0 1 = **turn off motor**

0 1 0 = **reverse motor direction**

0 1 1 = **toggle motor**

1 0 0 = **set motor speed slow**

1 0 1 = **set motor speed faster**

1 1 0 = **set motor speed fastest**

Sensor codes

No sensor related commands sent from Scratch are necessary. The Arduino will do edge detection in its main event loop and transmit a byte to Scratch only when it detects an edge. The Scratch when blocks trigger on receipt of a byte.

Currently the Arduino firmware only checks for "falling edges" – a transition from a high resistance to a low resistance state - that occurs when the metal clips are connected . When it detects a falling edge it sends a zero. In Scratch there is a "when" block that is triggered upon receipt of a zero.