

Light Play Arduino firmware commands for use with iPad

Version 0.2 of the Light Play hardware can control 3 rainbow lights and 1 motor. The board has an input for a resistive sensor.

Light Commands

set [all lights, light 1, light 2, light 3] **color to** [12-bit RGBW values]
off [all lights, light 1, light 2, light 3]
fade [all lights, light 1, light 2, light 3] **to** [12-bit RGBW values]
fade out [all lights, light 1, light 2, light 3]
set brightness [all lights, light 1, light 2, light 3] [8-bit value]
set fade speed [8-bit value]

Motor Commands

on thisway
on thatway
off
set motor speed [8-bit value]

Other Commands

resetstate
stopfades

Byte Codes

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

Command format

[u u u x x y y y]

upper bits set command type:

001 = motor commands

010 = light commands

011 = other

light commands

[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 light command:

0 0 0 = **set lightcolor to** (RGBW values follow in next 8 bytes, high byte/low byte)

0 0 1 = **turn off light**

0 1 0 = **fade lightcolor to** (RGBW values follow in next 8 bytes, high byte/low byte)

0 1 1 = **fade out light**

1 0 0 = **set brightness (divisor value follows in next byte)**
1 0 1 = **set fade speed (value in seconds follows in next byte)**

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 thisway**

0 0 1 = **turn on motor thatway**

0 1 0 = **motor off**

0 1 1 = **set motor speed (value follows in next byte, ranges from 1 to 10)**

Other commands

[0 1 1 x x y y y]

reset state variables

[0 1 1 0 0 0 0 0] (motorspeed = 10, tfade = 1000, all light powers = 1) (really don't need this since it can all be done with individual commands)

stopfades

[0 1 1 0 0 0 0 1] (stop all fades, to stopall do this followed by motoroff and all lightsoff commands)

Reporting

Bytes sent by the Arduino follow the following protocol:

Bytes with MSB clear contain 7-bit sensor values. These are streamed, unpolled, at ~ 10 Hz.

Bytes with MSB set are fade done messages, sent on completion of a fade. (128=light1 fade done, 129= light2 fade done, 130 =light3 fade done.)