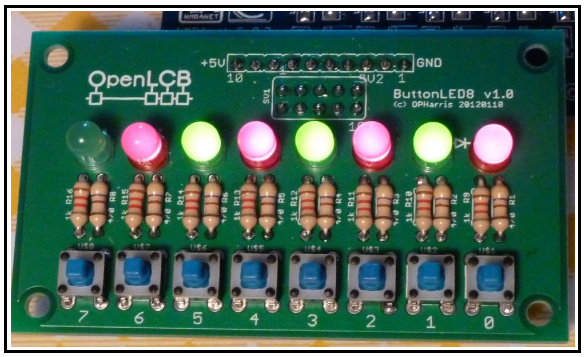
ButtonLED8 User Manual

The ButtonLED8 board is a small PCB board with eight buttons and eight LEDs on it. It is meant to be used as a mini-control panel, particularly with the OpenLCB / NMRAnet Development Kit (DevKit), but it can be useful in its own right.



It has a button and a LED connected to each of eight digital lines. This allows the use of each microprocessor pin to be used as an input and output at the same time, with multiplexing software.

It can be used as either a simple button panel used with the Io INPUTS. Alternately, it can be used as a auxiliary LED panel for the Io OUTPUTS.

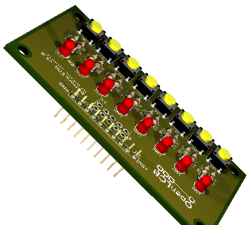
Connectors

The ButtonLED8 can have a number of connectors installed. The assembled ButtonLED8 has a 1x10 female connector installed on the bottom side. By inserting a 1x10 male header into this, the ButtonLED8 can be inserted into the Io INPUTS or OUTPUTS holes for testing purposes.

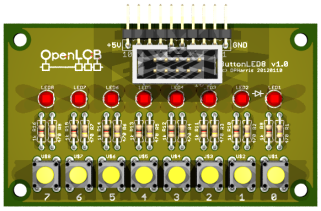
If a 1x10 male header is installed on the Input or Output Io connectors, then a ButtonLED8 with a 1x10 female header can be directly connected to those connectors. Alternately, if screw-terminals are installed on the Io, then a right-angle male connector installed on the ButtonLED8 will let it be attached to the Io. Other possibilities for connector combinations exist.

Another useful option is to install a 2x5 male header onto the top or bottom side. This lets you use a IDC ribbon cable to attach the ButtonLED8 to one of the Io nodes 2x5 connectors. This allows the ButtonLED8 to be used as a mini-panel at some distance from the Io.

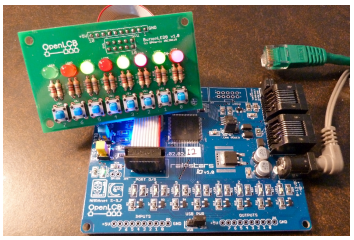
1x10 Male



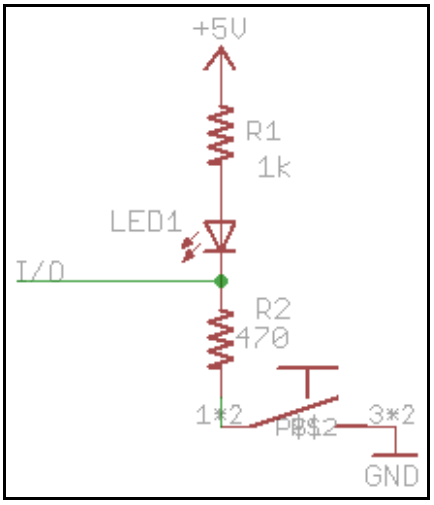
2x5 Male on Top



Using a Ribbon Cable



The Hardware, How It Works



The circuit diagram for one digital line is shown to the right. The microprocessor pin can be in one of three states:

1. Input, or high impedance. This results in the LED reflecting the state of the button. When the button is depressed, the LED will light.
2. Output Low. This sinks current through the LED and lights it.
3. Output High. This turns the LED off.

By multiplexing the pin rapidly, a routine can read the button and control the LED independently.

The ButtonLED8 Library

This library was written to use the above circuit. Each microprocessor pin is represented by an ButtonLED8 object. Its API allows the LED to be set on, off, or flashing according to a pattern. The button state is available as an object variable, as are the duration of the current and last states. To make it operate, the library includes a service routine that has to called repetitively, as least every 32 ms, and this can be easily done by calling it the main program loop.

Each time it is called more often than 32 ms it returns. Other wise it updates the LED state, performing any flashing pattern. The output pin is kept as an output for most of the time, and the button-state is ignored, but every 64 ms, the pin is briefly set an output, and the button-state is sampled, debounced and its state stored.

ButtonLED8 Library API

```
public:
 uint16_t sense;        // active sense of button/LED
 long pattern;          // current output drive pattern (rotates)
 long duration;         // how long the button has been in its current state in msec
 long lastDuration;     // how long it was in the previous state, in msec
 uint8_t pin;           // connected pin number
 uint8_t once;          // mask for single blink
 bool ledState;         // current output drive state
 bool state;            // current button state

 ButtonLed(uint8_t pin, uint8_t sense);  // associate pin and sense for active state
 ButtonLed(uint8_t pin);                 // associate pin, default sense=HIGH
 ButtonLed();
 void setPinSense(uint8_t p, uint8_t s);  // initialize pin and sense, useful for arrays
 void on(long pattern);                  // set the output repeating pattern
 void blink(uint8_t pattern);            // set the output blink pattern one time only
 void process();                         // processes LED and buttons, call periodically
 bool ButtonLed::unique()                // returns if there has been a change in button state
                                         // unique() calls process() internally
```

Example use:

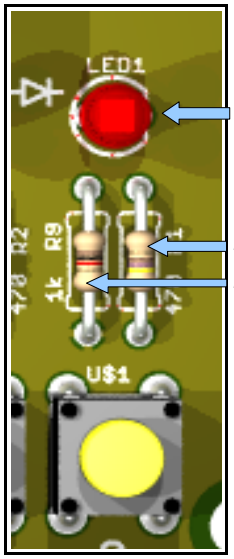
```
#include "ButtonLED.h"    // Include the library
ButtonLED b(15);          // pin 15 is declared to be a BL
void setup() {
  b.on(0L);               // turn LED off
}
void loop() {
  if(b.unique()) {        // Has the button-state changed?, this calls process() internally
    if(b.state)
      b.on(0x0000FFFFL);  // slow flashing
    else
      b.on(0x0F0F0F0FL);  // rapid flashing
  }
}

// defining an array of buttons
ButtonLed bb[64];
void setup() {
 DDRG |= 0x03;
 for(int i=0; i<50; i++) {
   bb[i].setPinSense(i, LOW);
   bb[i].pattern = 0x0F0F0F0FL;
 }
}
int p;
void loop() {
 if(p>=50) p=0;
 bb[p++].process();
}
```

Assembling ButtonLED8

1. Parts list:
   - 1 x PCB board
   - 8 x LEDs
   - 8 x buttons
   - 8 x ~1000 ohm resistors (1200 ohm supplied)
   - 8 x ~470 ohm resistors
   - 1 x 1x10 male female header

   Optionally: 1 x 2x5 male header (you could also use two 1x5 male strips)



2. Insert and solder in the 8 buttons, and solder.
3. Insert eight 1000 ohm resistors, solder and clip.
4. Insert the eight 470 ohm resistors, solder and clip.
5. Insert the eight LEDs, the flat side on the rim should be oriented as shown, solder and clip.
6. Insert the 1x10 female header on the BOTTOM of the board, solder.

   Optionally:
7. Insert the 2x5 male header on the TOP of the board.

   To make use of this, you will also need to insert 2x5 headers onto an Io board, and construct a cable using two 2x5 female displacement connectors and ribbon cable.

Notes:

1. The ButtonLED8 can make a mini-panel when used with a longer ribbon cable.
2. The 1x10 and 2x5 positions were placed slightly close together, and using one or the other may be required, depending on placement on the top or bottom of the board.
3. The 1x10 and 2x5 connectors can be any combination of: male, female, straight, 90-degree, hooded, bare; and the particular choice may depend on the use of the board. Eg, a min-panel may want to use a connector mounted on the back.
4. Button-caps can be mounted onto the button, if desired.