



TinyTiM, The intelligent Matrix

Test Plan
Dec 18, 2014
Justin Shaw

Overview:

TinyTiM is an 8x8 array of smart RGB LED pixels that can be configured into 8 parallel strands of 8 pixels, 1 strand of 64 pixels or many other combinations of rows.

TinyTiM is half size version of TiM.

TinyTiM will be tested in parallel mode using a 16-lead ribbon cable.

In order to avoid the LED lost issue, please consider to add reflow into the production procedure and improve the packaging.

Functionality to be tested

Each LED should be capable of displaying RED/GREEN/BLUE and WHITE.

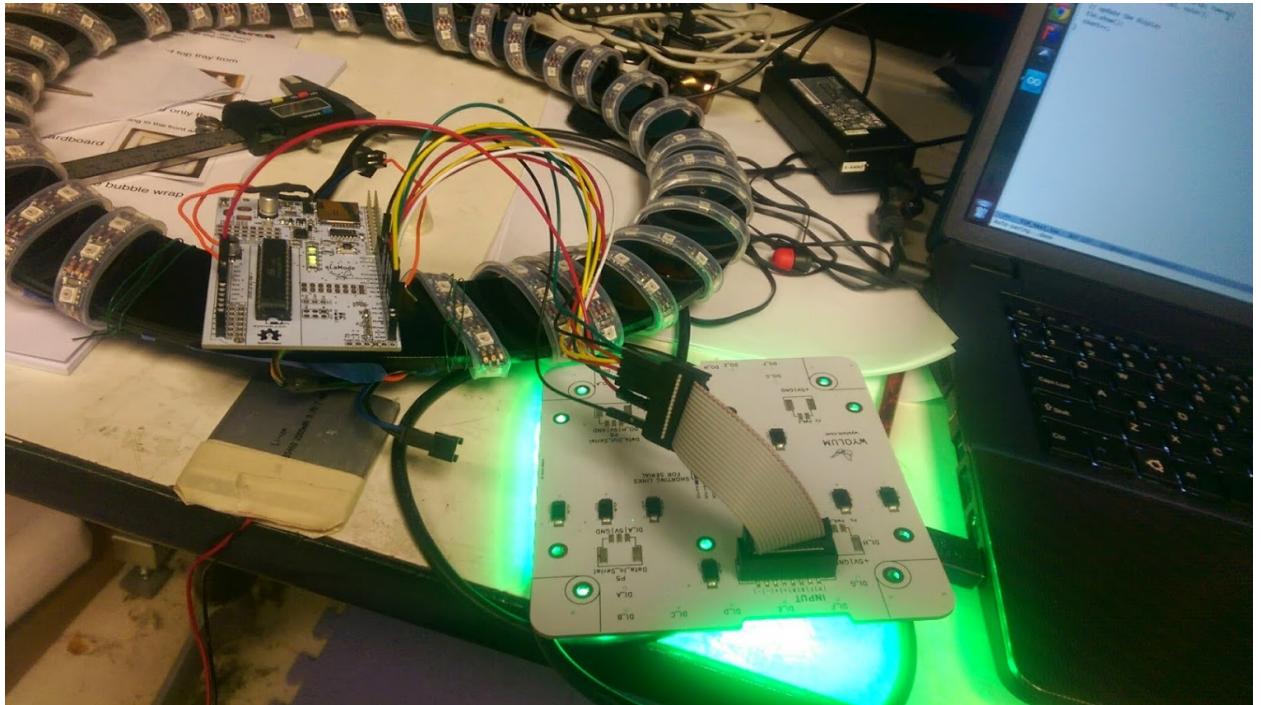
Test Hardware Required

1. Arduino UNO or similar
2. 16 (2x8) lead ribbon cable
3. 3A power supply

Test Setup

1. [Program the Arduino compatible with test code](#). (And posted below)
2. Connect the Arduino compatible to TinyTiM under test.
 - a. Arduino D2 --> TinyTiM P3 Data_In A
 - b. Arduino D3 --> TinyTiM P3 Data_In B

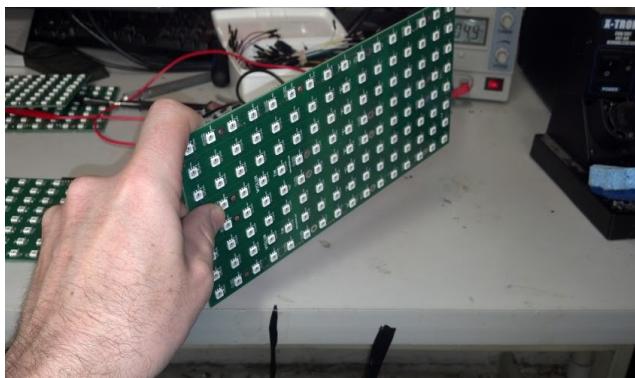
- c. Arduino D4 --> TinyTiM P3 Data_In C
- d. Arduino D5 --> TinyTiM P3 Data_In D
- e. Arduino D6 --> TinyTiM P3 Data_In E
- f. Arduino D7 --> TinyTiM P3 Data_In F
- g. Arduino D8 --> TinyTiM P3 Data_In G
- h. Arduino D9 --> TinyTiM P3 Data_In H
- i. Arduino +5V --> TinyTIM +5V
- j. Arduino GND --> TinyTIM GND



3. Connect power supply
 - a. Power +5V --> Arduino +5V
 - b. Power GND --> Arduino GND

Test Sequence

1. Visual inspection for any defects
2. Vibration testing: bang the board, edge on, against the table. Make sure no LEDs fall off. Run hand over the LED array to see if any LEDs come loose



3. .
4. Power the unit
5. See that ALL LEDs display ALL colors.

Test code

```
#include "TiM.h"
#include "Adafruit_NeoPixel.h"
TiM tim;

// Use Arduino Pins 2-9 for controling the rows of TiM
uint8_t pins[8] = {2, 3, 4, 5, 6, 7, 8, 9};
// uint8_t pins[8] = {0, 1, 2, 3, 4, 5, 6, 7};

void setup(){
    // each Pin has 32 leds
    //      #rows #led pin_ids
```

```
tim.setup(8, 64, pins);
tim.setall(Color(255, 255, 255));
tim.show();
delay(1000);
tim.setall(Color(255, 0, 0));
tim.show();
delay(1000);
tim.setall(Color(0, 255, 0));
tim.show();
delay(1000);
tim.setall(Color(0, 0, 255));
tim.show();
delay(1000);
}

uint32_t count = 0;
int row, col;

void loop() {
    // grab a color from the color wheel
    //                                HUE 0-255  brightness 0-255
    uint32_t color = Wheel(count % 256, 255);

    // turn off column
    for(row = 0; row < tim.n_strip; row++) {
        // tim.setPixel(row, col, 0);
    }

    // turn on new column
    col++;
    col %= tim.strips[0].numLEDs;
    for(row = 0; row < tim.n_strip; row++) {
        tim.setPixel(row, col, color);
    }

    // update the display
    tim.show();
    count++;
}
```