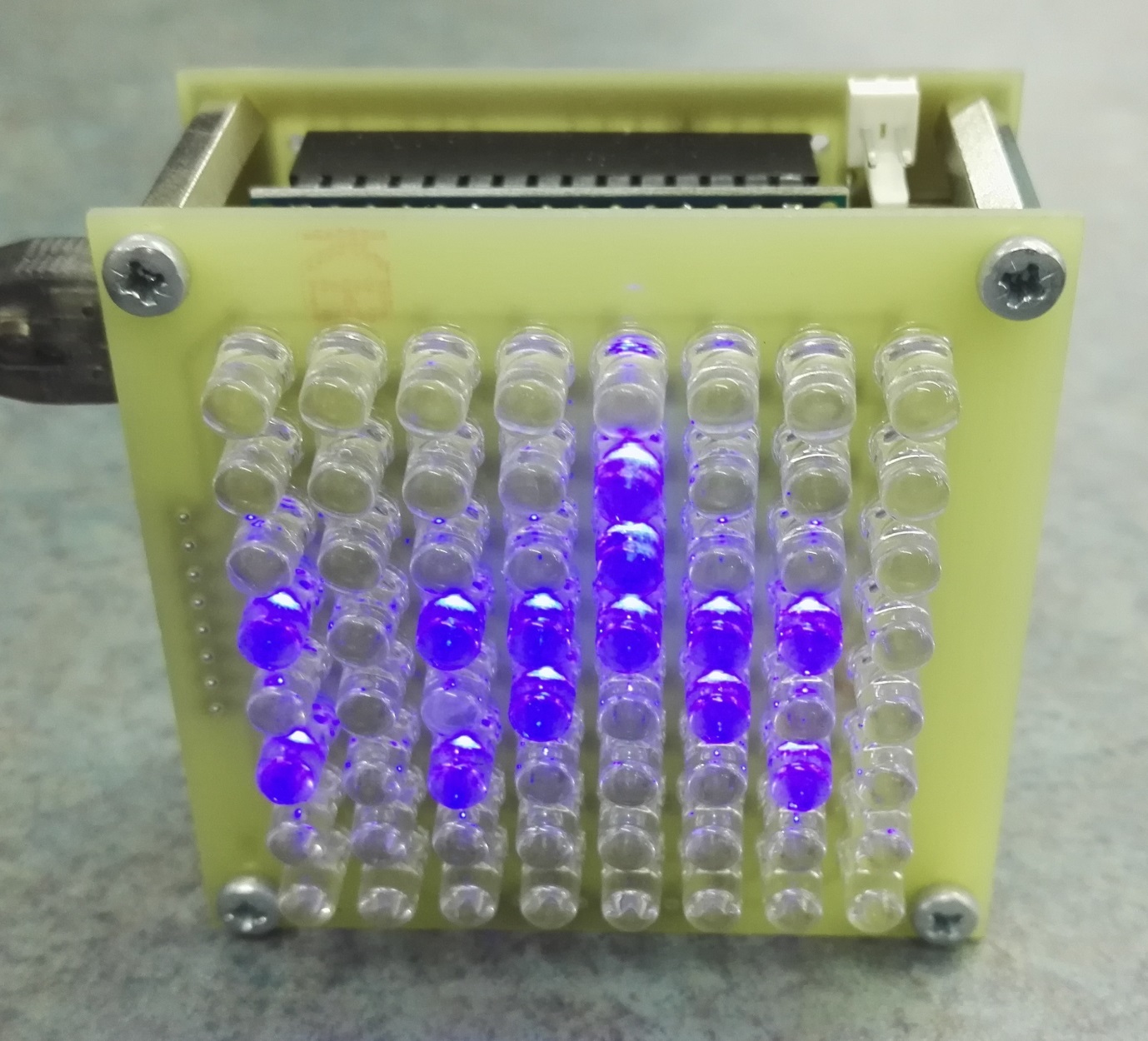
**8x8 LED Matrix – Work Experience Project.**



**Objective:**

To build an LED matrix board that will display a number of custom messages and (roughly) display the current room temperature.

**Skills required:**

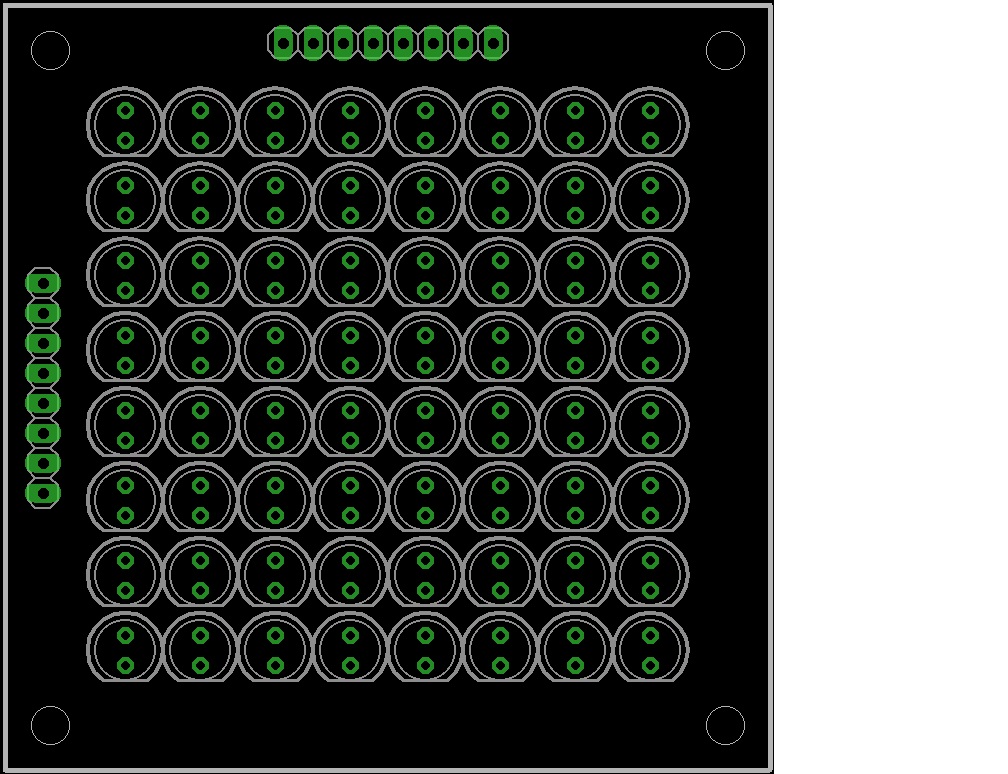
Component identification, soldering, assembly and programming.

**Overview:**

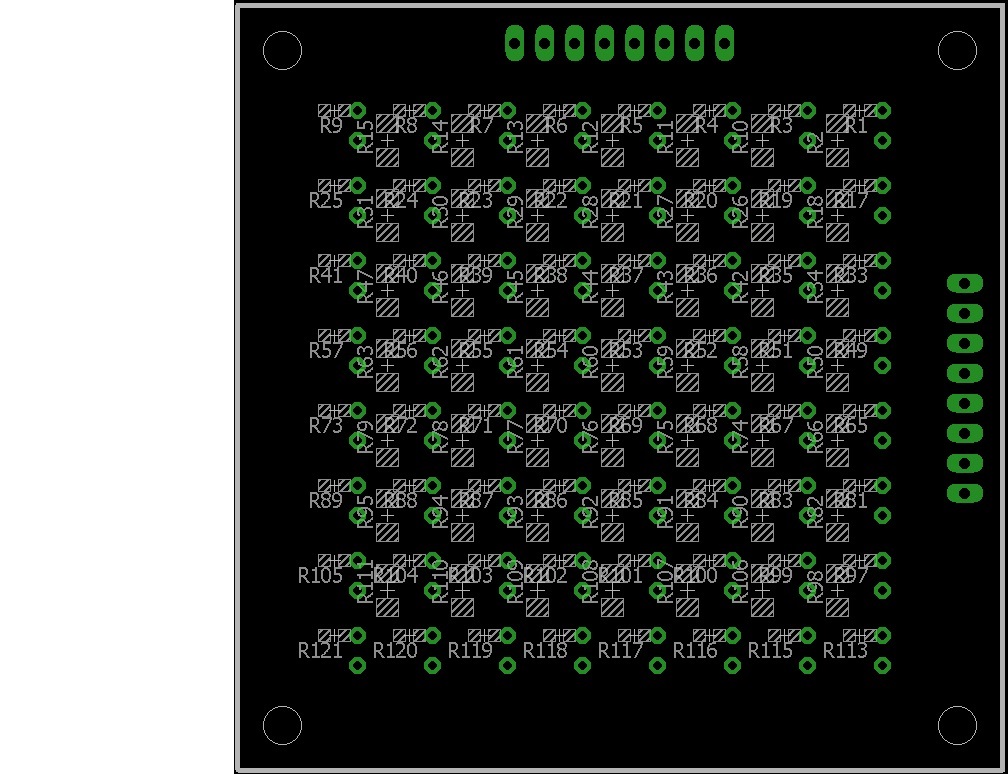
Two circuit boards will need to be routed on the PCB router, one for the LED board and the other for the control board. Once populated, both boards will be connected together to create one functional unit.

The first stage is to populate the LED board with 64 5mm LEDs and accompanying current limiting resistors followed by two sets of 2.54mm pitch pin headers.

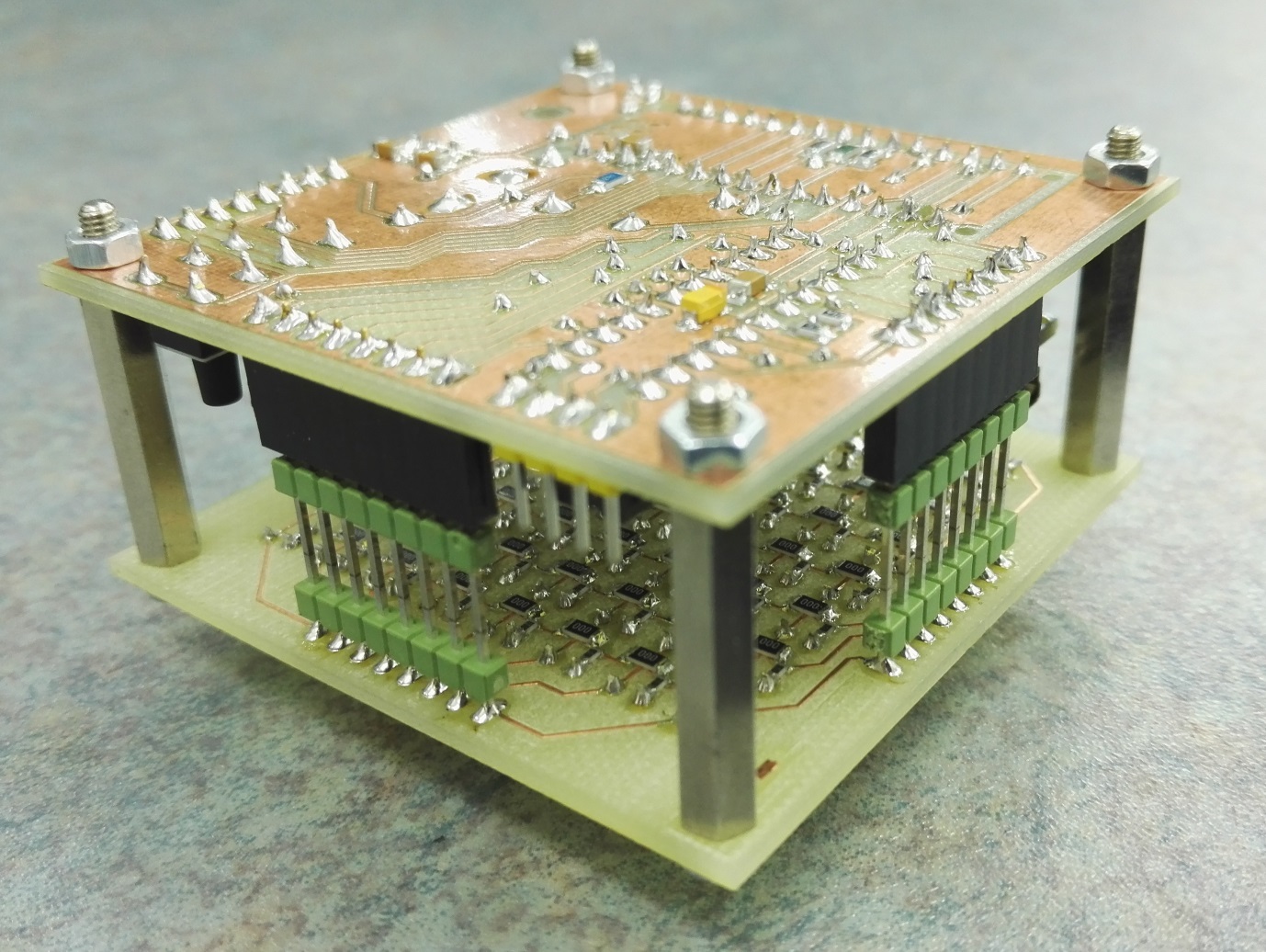
The LEDs are polarised and must be inserted into the board in the correct orientation before soldering.

*(This image is a top view of the LED board.)*

The resistors should now be soldered on to the underside of the board, the smaller 0603 sized resistors are for current limiting and the larger 1206 sized resisters are 0 ohm jumpers.



Next, the headers on the LED board should be soldered into position.



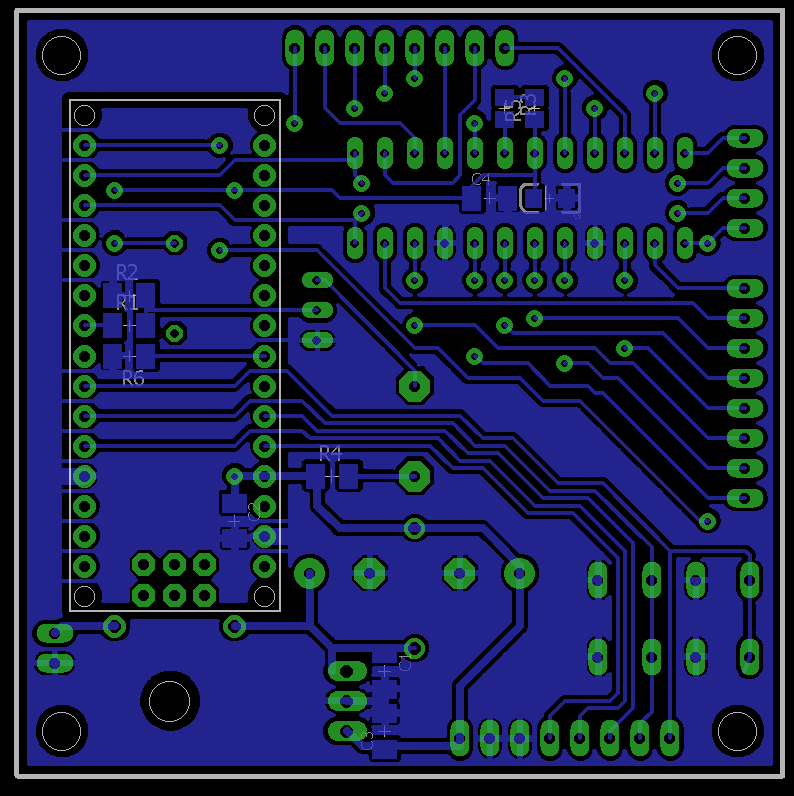
A small strip of masking tape placed over the holes will help align the pins; they should be flush to the top of the board on the LED side and not standing proud.

Now that the LED board is complete we will move on to the control board.

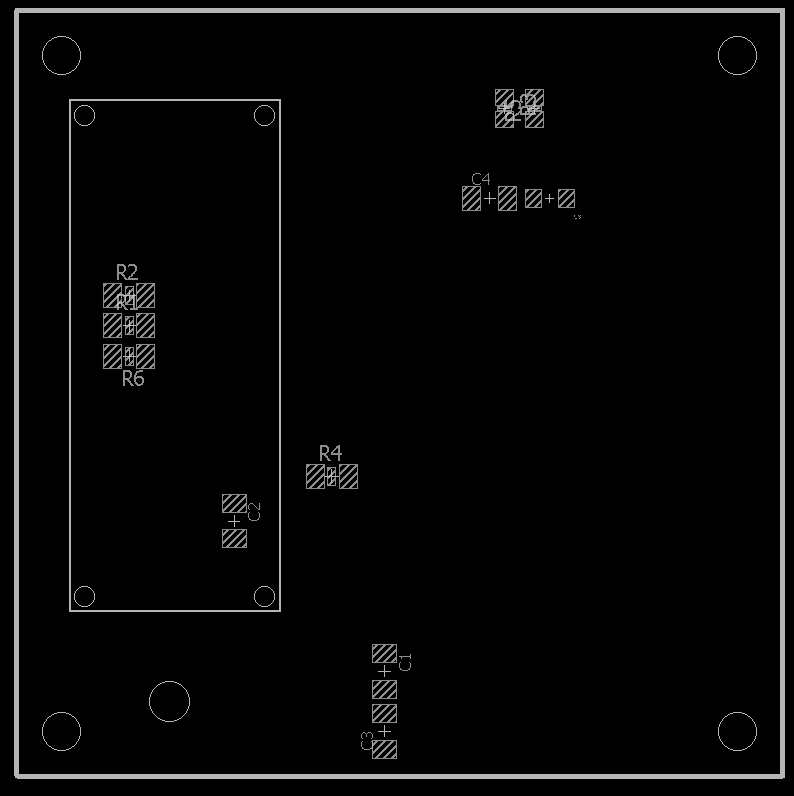
The control board incorporates a separate micro controller board (Arduino Nano), MAX7219 LED controller chip, power supply, buzzer, thermistor (temperature sensor), push buttons and two extra connectors for future expansion.

All components should be gathered and identified ready for soldering.

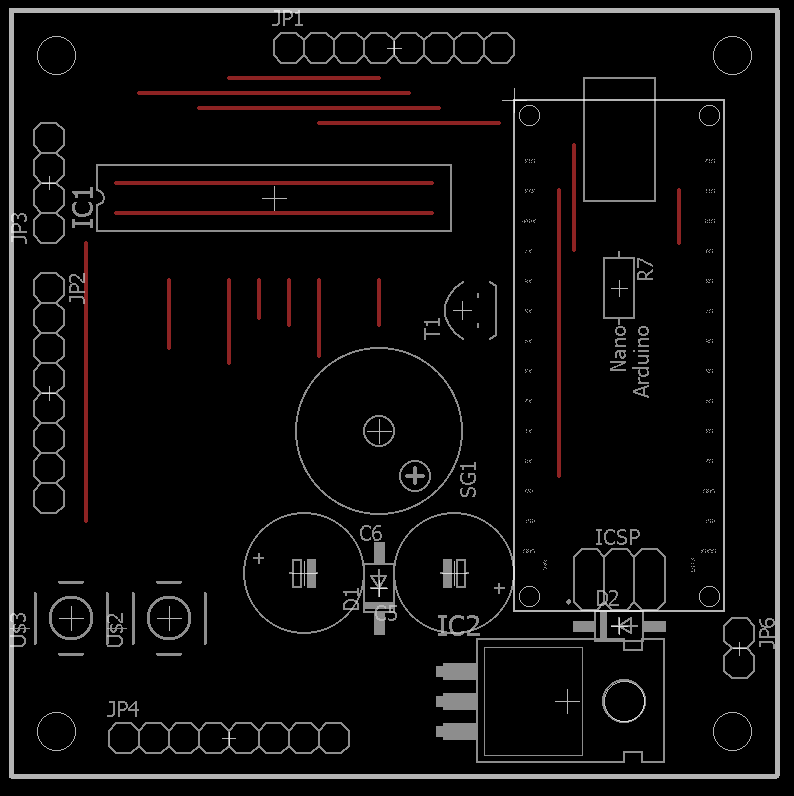
Surface mount devices should be soldered first, followed by link wires then through leaded parts; these should be soldered in size order starting from smallest to largest.



*(This is a view of the underside of the control board.)*



*(This image details the bottom side SMD placement on the control board.)*



*(The red lines in the above image are link wires that must be placed on top of the board and soldered underneath.)*

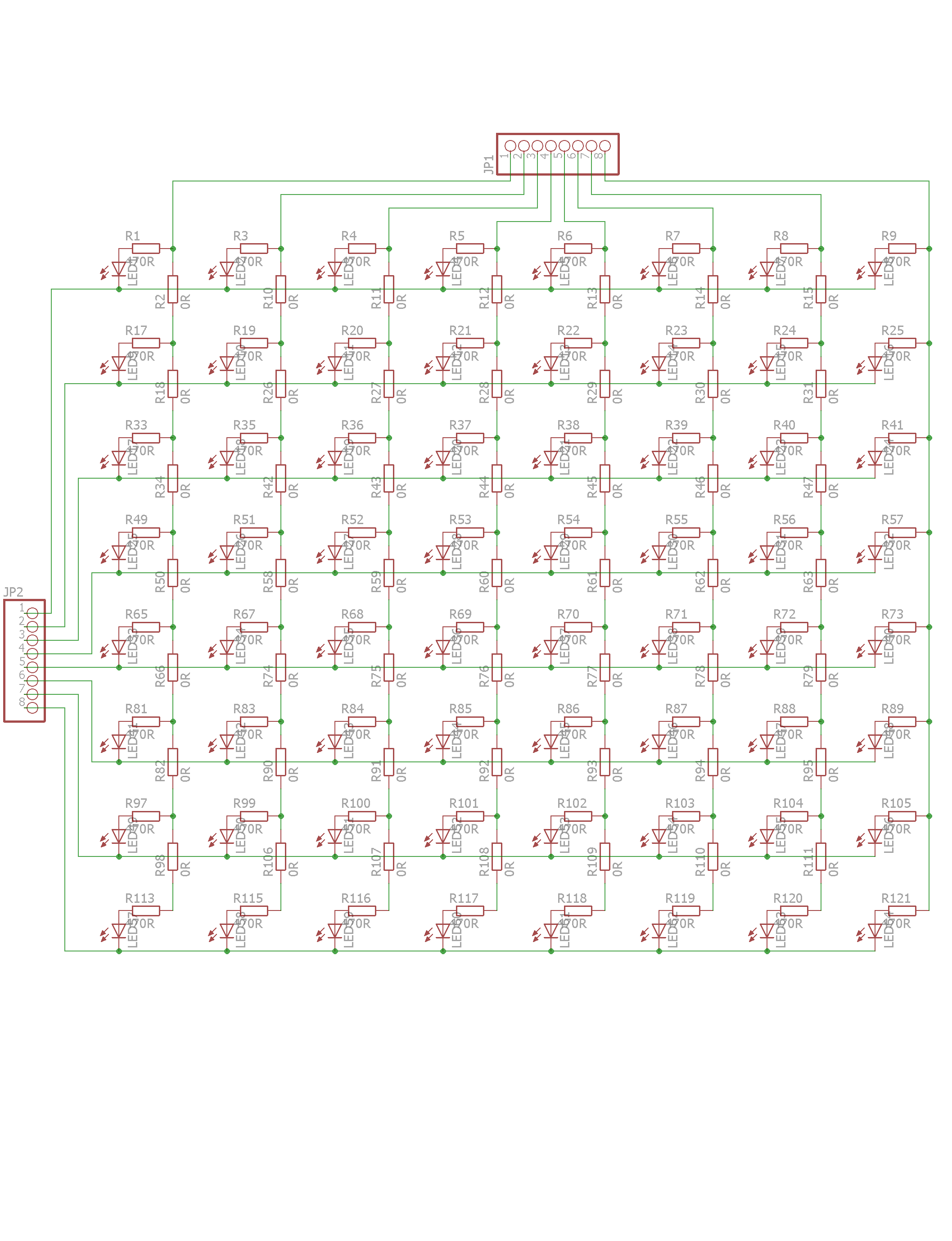
Once this stage has been completed and checked for errors we will then assemble and program the Arduino board. An Arduino “sketch” and libraries will be provided; at this point you will be able to add up to 9 custom messages to display on the matrix. The sketch will then be compiled and uploaded to the Arduino board using the Arduino IDE software.

Connect the Arduino to the control board; mount the two boards together with 25mm hex pillars and M3 screws. Plug in the USB cable to provide power to the unit.

Messages can be sent to the Arduino over a serial connection at 9600 baud. The serial message will be displayed as string0 in the sketch.

**Project info:**

Here are the project schematics and parts lists:



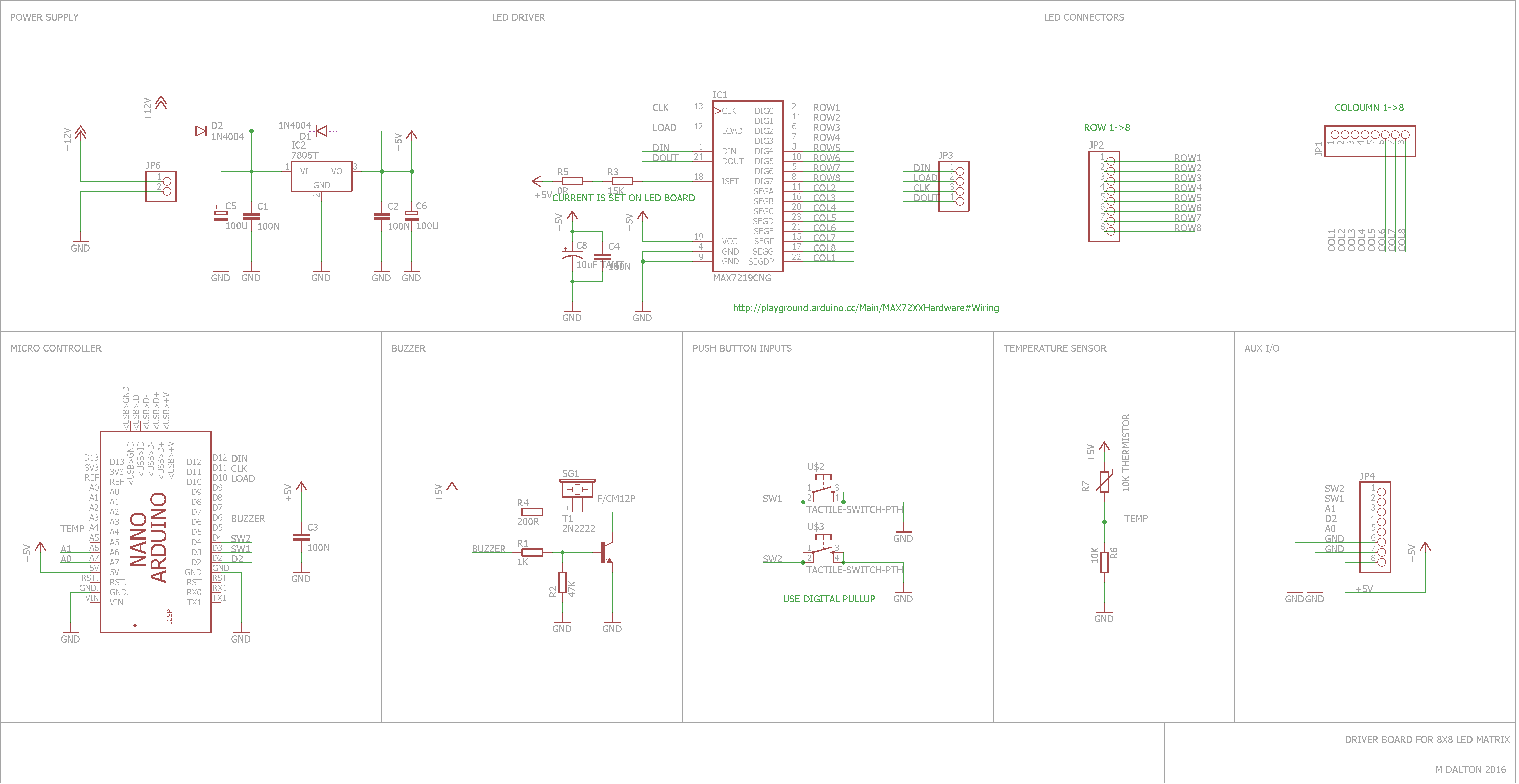
For the LED board, only four different types of component are used, only the LEDs have a polarity and because the two types of resistors are also different sizes, it would be very hard to build this board incorrectly.

LED5MM led

470R R0603

0R R1206

1X08 pinhead



This is the parts list for the driver board.

Partlist

Part Value Device Package

C1 100N C-EUC1206K C1206K

C2 100N C-EUC1206K C1206K

C3 100N C-EUC1206K C1206K

C4 100N C-EUC1206K C1206K

C5 100U CPOL-EUE5-10.5 E5-10,5

C6 100U CPOL-EUE5-10.5 E5-10,5

C8 10uF TANT 10UF-TANT EIA3216

D1 1N4004 1N4004 DO41-10

D2 1N4004 1N4004 DO41-10

IC1 MAX7219CNG MAX7219CNG DIL24-3

IC2 7805T 7805T TO220H

JP1 PINHD-1X8 1X08

JP2 PINHD-1X8 1X08

JP3 PINHD-1X4 1X04

JP4 PINHD-1X8 1X08

JP6 PINHD-1X2 1X02

PCB1 ARDUINO ARDUINO\_NANO ARDUINO\_NANO

R1 1K R-EU\_M1206 M1206

R2 47K R-EU\_M1206 M1206

R3 15K R-EU\_M0805 M0805

R4 200R R-EU\_M1206 M1206

R5 0R R-EU\_M0805 M0805

R6 10K R-EU\_M1206 M1206

R7 10K THERMISTOR-7,5 R-7,5

SG1 F/CM12P BUZZER F/CM12P

T1 2N2222 2N2222 TO92-EBC-OVAL

U$2 SWITCH TACTILE-SWITCH-PTH TAC-SWITCH-PTH

U$3 SWITCH TACTILE-SWITCH-PTH TAC-SWITCH-PTH

**Extra features:**

The control board has some unused hardware for future expansion.

Two momentary push buttons and a buzzer are connected to the Arduino as well as an expansion header for both the Arduino and the Maxim LED driver chip. Extra features may be added into the code for greater functionality.

Multiple LED matrixes may be daisy chained and controlled by the Arduino as long as each matrix has its own controller IC.

A two pin power connector is present to allow the unit to be powered by a 9V battery; this connector is connected to a linear 5 volt regulator which will power both the LED and control boards.

**The Code:**

The following is the exact code used in this project. It is dependent on the MaxMatrix library made by Oscar Kin-Chung Au. (<https://code.google.com/archive/p/arudino-maxmatrix-library/downloads>)

Arduino IDE Download links: <https://www.arduino.cc/en/Main/Software> & <http://www.arduino.org/downloads>

/\*

################################################################################

# 8x8 LED matrix with temperature measurement and serial input. #

# Code taken from Marcelo Moraes, July 9th, 2013 & Arduino playground. #

# #

# visit this web page for further information about MaxMatrix library #

# https://code.google.com/p/arudino-maxmatrix-library/ #

# #

# This code is a public example. #

# #

# Rehashed by Mick Dalton, 2016. #

# #

################################################################################

\*/

#include <MaxMatrix.h>

#include <avr/pgmspace.h>

//thermistor start

#include <math.h>

double ThermistorC(int RawADC) {

double Temp;

Temp = log(10000.0 \* ((1024.0 / RawADC - 1)));

Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 \* Temp \* Temp )) \* Temp );

Temp = Temp - 273.15;

return Temp;

}//thermistor end

PROGMEM const char CH[] = {

3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space

1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !

3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "

5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #

4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $

5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %

5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &

1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '

3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (

3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )

5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // \*

5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +

2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,

4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -

2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .

4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /

4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0

3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1

4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2

4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3

4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4

4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5

4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6

4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7

4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8

4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9

2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :

2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;

3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <

3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =

3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >

4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?

5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @

4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A

4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B

4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C

4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D

4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E

4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F

4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G

4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H

3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I

4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J

4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K

4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L

5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M

5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N

4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O

4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P

4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q

4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R

4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S

5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T

4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U

5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V

5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W

5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X

5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y

4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z

2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [

4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backslash

2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, // ]

3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat

4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // \_

2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `

4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a

4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b

4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c

4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d

4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e

3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f

4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g

4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h

3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i

4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j

4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k

3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l

5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m

4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n

4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o

4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p

4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q

4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r

4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s

3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t

4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u

5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v

5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w

5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x

4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y

3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z

3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {

1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |

3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }

4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~

};

int data = 12; // DIN pin of MAX7219 module

int load = 10; // CS pin of MAX7219 module

int clock = 11; // CLK pin of MAX7219 module

int maxInUse = 1; //change this variable to set how many MAX7219's you'll use

MaxMatrix m(data, load, clock, maxInUse); // define module

byte buffer[10];

char string1[] = " line1 ";

char string2[] = " line2 ";

char string3[] = " line3 ";

char string4[] = " line4 ";

char string5[] = " line5 ";

// DO NOT ALTER THE LINE BELOW.

char string6[] = " \* Temp XX.XXc \* ";

char string7[] = "LOW BRIGHTNESS!!!! ";

char string8[] = "MEDIUM BRIGHTNESS!!!! ";

char string9[] = "NORMAL BRIGHTNESS!!!! ";

char string10[] = "MAXIMUM BRIGHTNESS!!!! ";

const int buzzer = 6;

const int button1 = 3;

const int button2 = 4;

const int io2 = 2;

// add 2 more io's

void setup() {

pinMode(buzzer, OUTPUT);

pinMode(button1, INPUT\_PULLUP);

pinMode(button2, INPUT\_PULLUP);

pinMode(io2, INPUT);

m.init(); // module initialize

m.setIntensity(15); // dot matix intensity 0-15

Serial.begin(9600); // serial communication initialize

}

void loop() {

//thermistor start

int valC;

double tempC;

valC = analogRead(A4);

tempC = ThermistorC(valC);

Serial.print("Temperature = ");

Serial.print(tempC);

Serial.println(" C");

int TempDigit1 = floor((int)(tempC) % 10);

int TempDigit2 = (int)tempC / 10;

int TempDigit3 = (int)(tempC \* 10) - ((int)(tempC) \* 10);

int TempDigit4 = (int)(tempC \* 100) - ((int)(tempC) \* 100) - (TempDigit3 \* 10);

string6[8] = TempDigit2 + 48; //5

string6[9] = TempDigit1 + 48; //6

string6[11] = TempDigit3 + 48; //8

string6[12] = TempDigit4 + 48; //9

//thermistor end

// only use 1 serial port at any given time!

// this is the code if you want to enter a message via USB serial console

while (Serial.available() > 0) {

byte c = Serial.read();

Serial.println(c, DEC);

printCharWithShift(c, 100);

}

delay(100);

m.shiftLeft(false, true);

m.setIntensity(5);

// print the active sentences, "//" out any line to ignore it.

printStringWithShift(string1, 70);

printStringWithShift(string2, 70);

printStringWithShift(string3, 70);

printStringWithShift(string4, 70);

printStringWithShift(string5, 70);

printStringWithShift(string6, 70);

m.setIntensity(1); // dot matix intensity 0-15

printStringWithShift(string7, 80);

m.setIntensity(5); // dot matix intensity 0-15

printStringWithShift(string8, 80);

m.setIntensity(10); // dot matix intensity 0-15

printStringWithShift(string9, 80);

m.setIntensity(15); // dot matix intensity 0-15

printStringWithShift(string10, 80);

m.setIntensity(10); // dot matix intensity 0-15

}

void printCharWithShift(char c, int shift\_speed) {

if (c < 32) return;

c -= 32;

memcpy\_P(buffer, CH + 7 \* c, 7);

m.writeSprite(maxInUse \* 8, 0, buffer);

m.setColumn(maxInUse \* 8 + buffer[0], 0);

for (int i = 0; i < buffer[0] + 1; i++)

{

delay(shift\_speed);

m.shiftLeft(false, false);

}

}

void printStringWithShift(char\* s, int shift\_speed) {

while (\*s != 0) {

printCharWithShift(\*s, shift\_speed);

s++;

}

}

//end of sketch