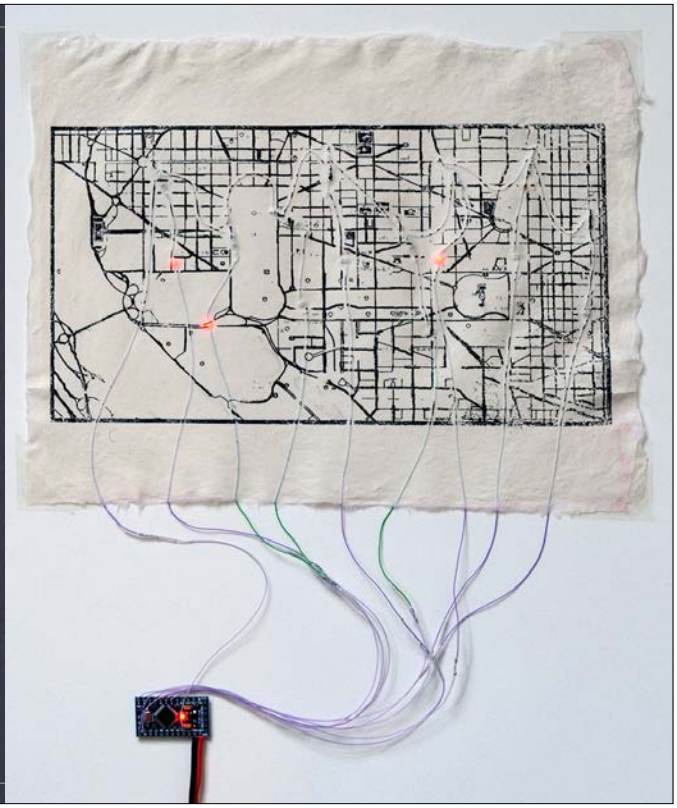
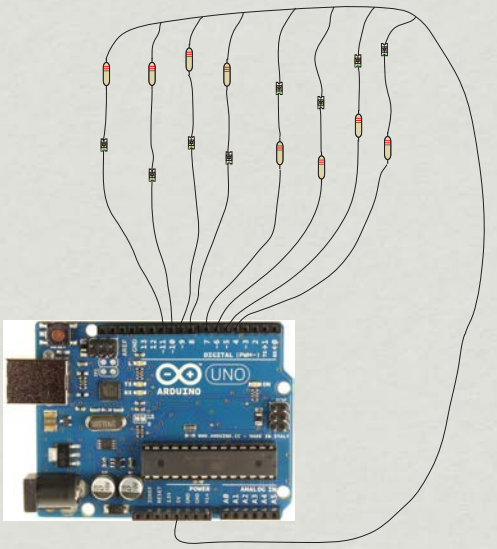
SURFACE MOUNT LED HAND-SOLDERING

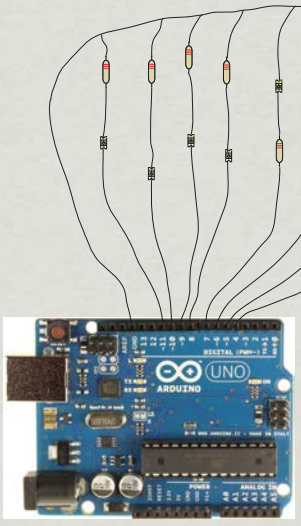
ERIK BRUNVAND



Overview of Arduino connection

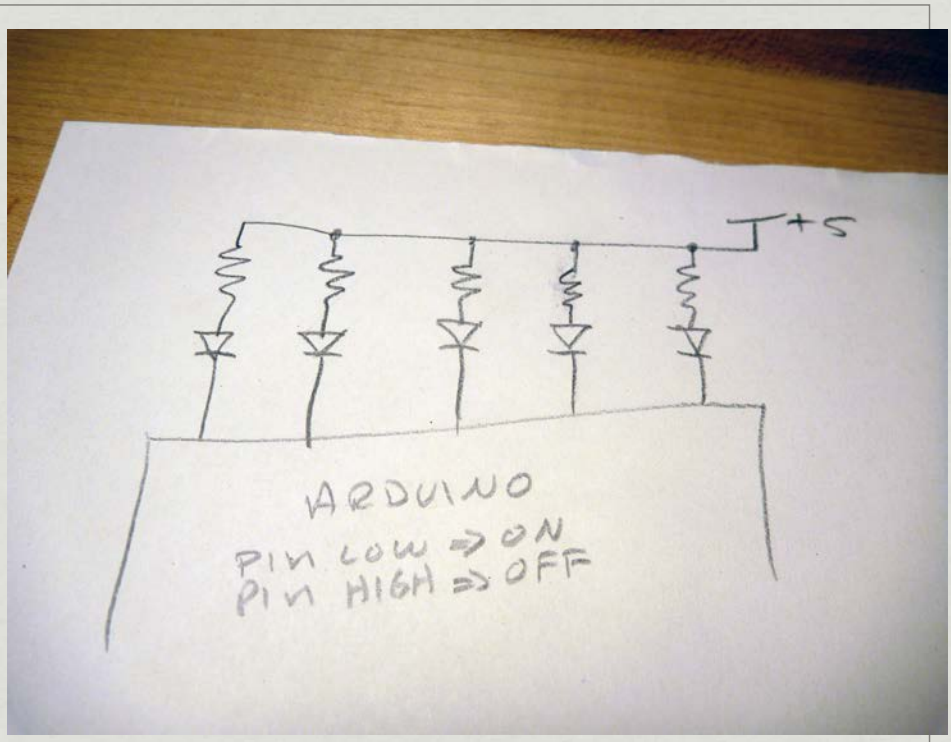


Anodes tied to 5v, cathodes tied to Arduino digital pins. Use digitalWrite(<pin>, LOW); to make the LED light up.

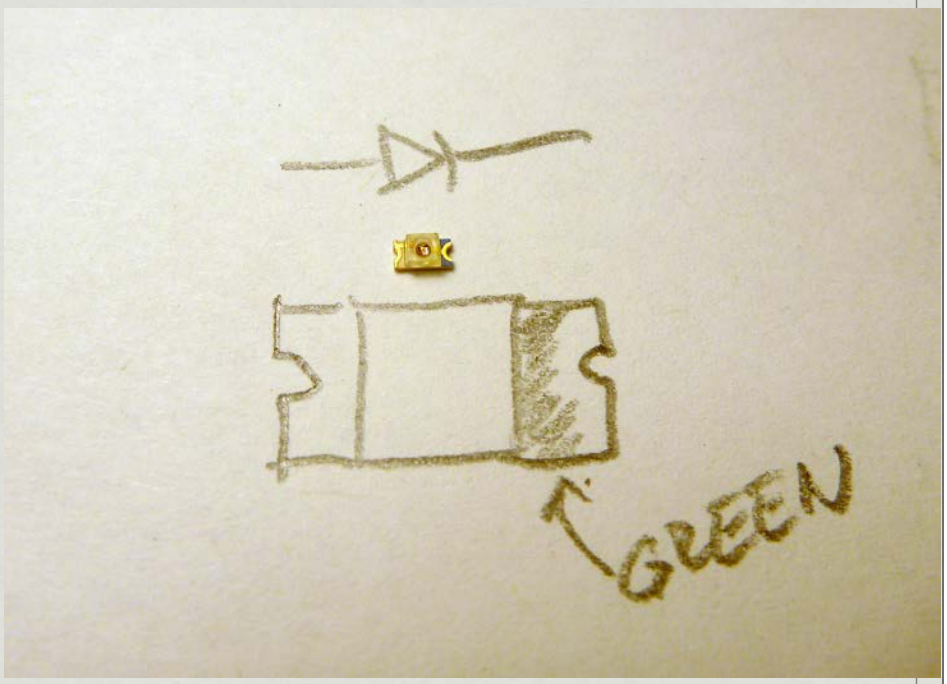


Anodes tied to Arduino pins, cathodes tied to GND. Use digitalWrite(<pin>, HIGH); to make the LED light up.

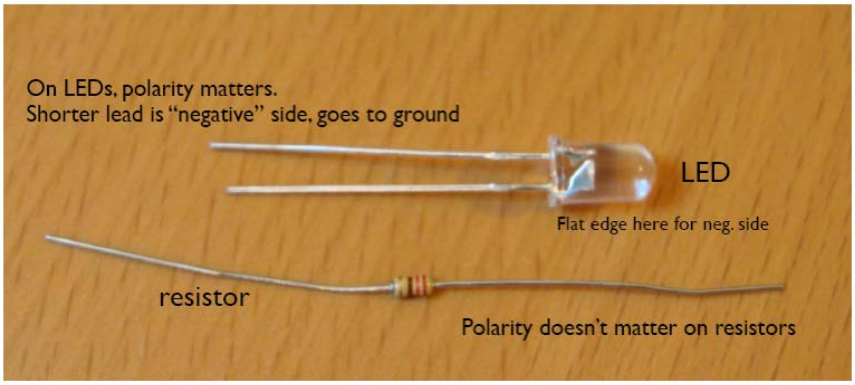
Another view

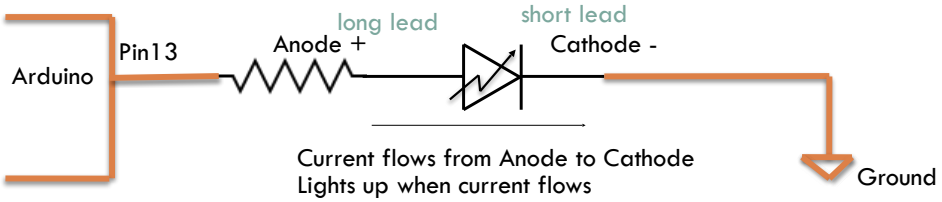


SMD LED Orientation



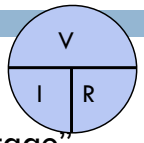
LEDs and Resistors





Current Limiting Resistor

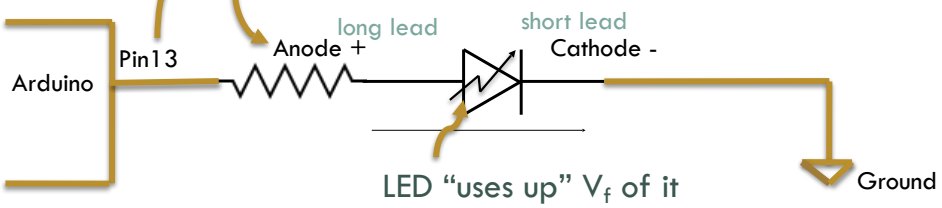
- Ohm's Law
  - V = IR I = V/R R = V/I



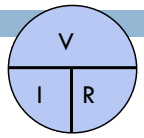
- Every LED has a V_f "Forward Voltage"
  - How much voltage is dropped (used up) passing through the LED

"HIGH" forces output pin to 5v (called V)

Resistor "uses up" the rest $(V - V_f)$

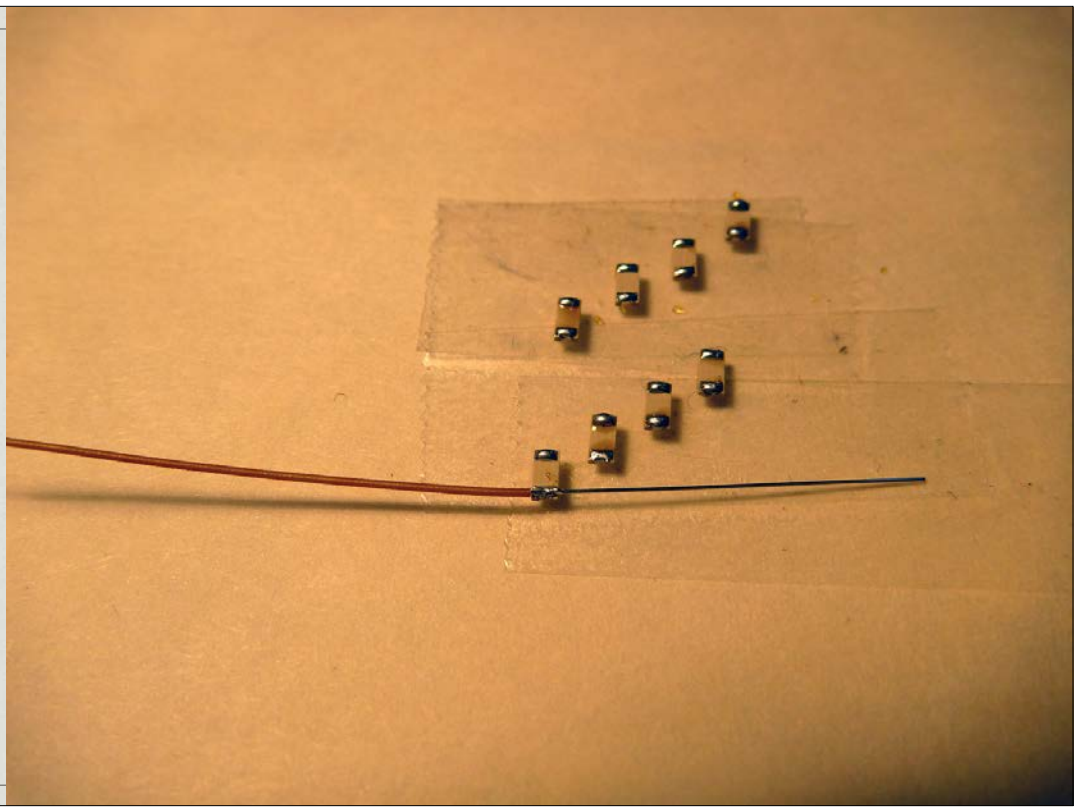


Current Limiting Resistor

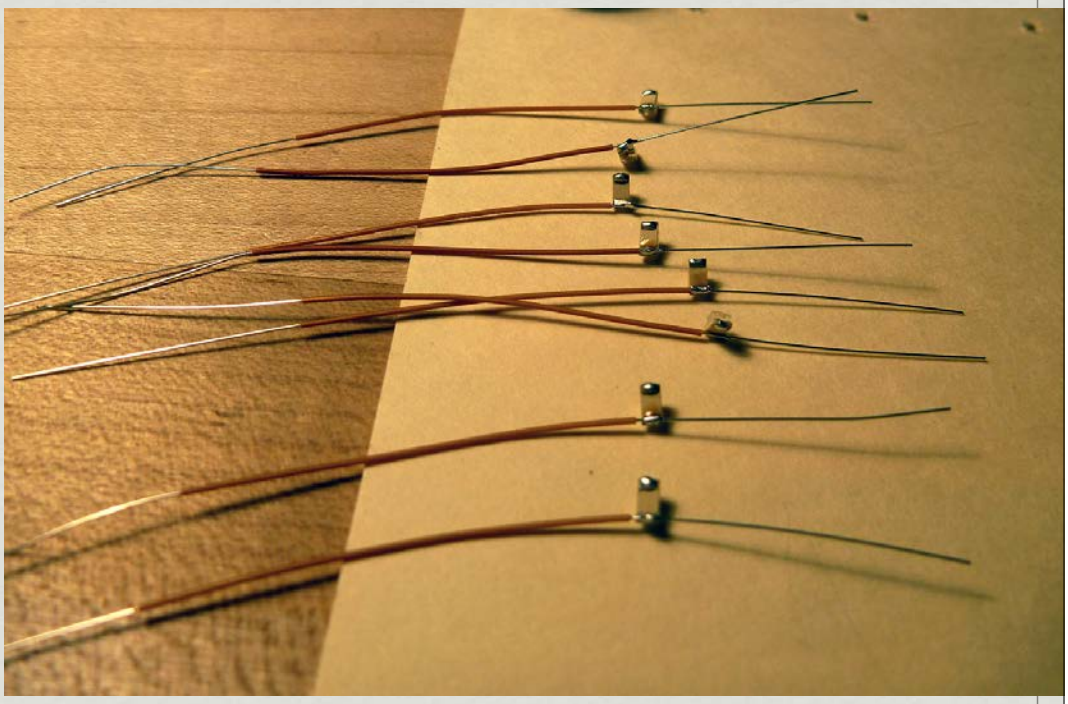


- Ohm's Law
  - V = IR I = V/R R = V/I
- Every LED has a V_f "Forward Voltage"
  - How much voltage is dropped (used up) passing through the LED
- $R = (V - V_f) / I$
  - Example – If V_f is 1.9v (red LED), and V = 5v, and you want 15mA of current (0.015A)
  - R = (5 – 1.9)/0.015 = 3.1/0.015 = 206Ω
  - Exact isn't critical – use next size up, i.e. 220Ω
  - Or be safe and use 330Ω or 470Ω
  - This would result in 9.4mA or 6.6mA which is fine

Get them Started

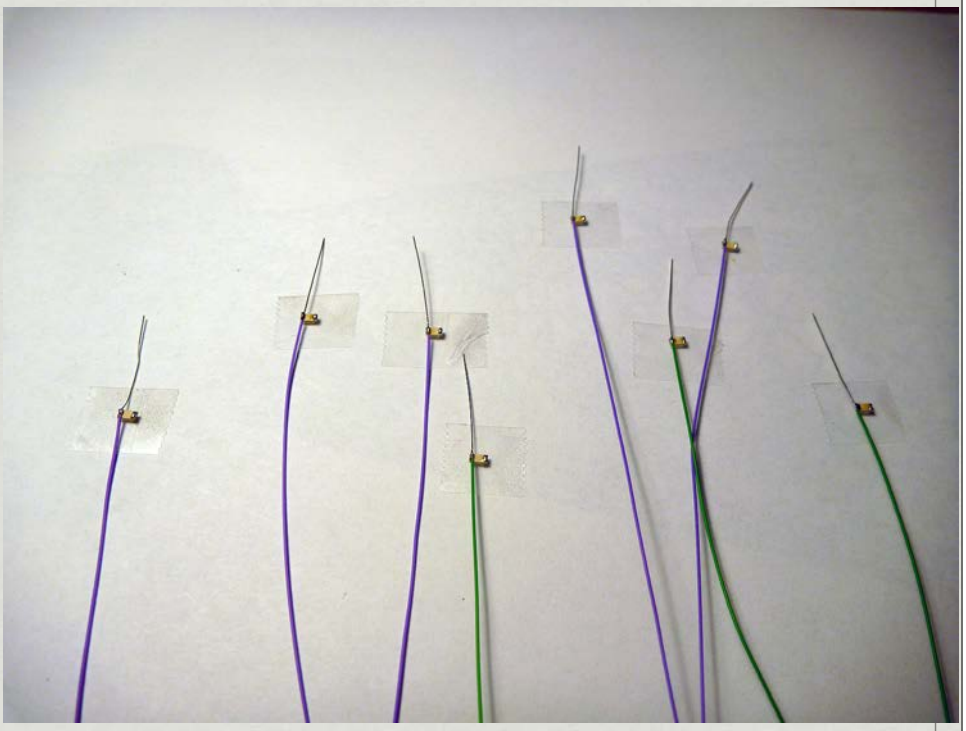


One on, one to go…



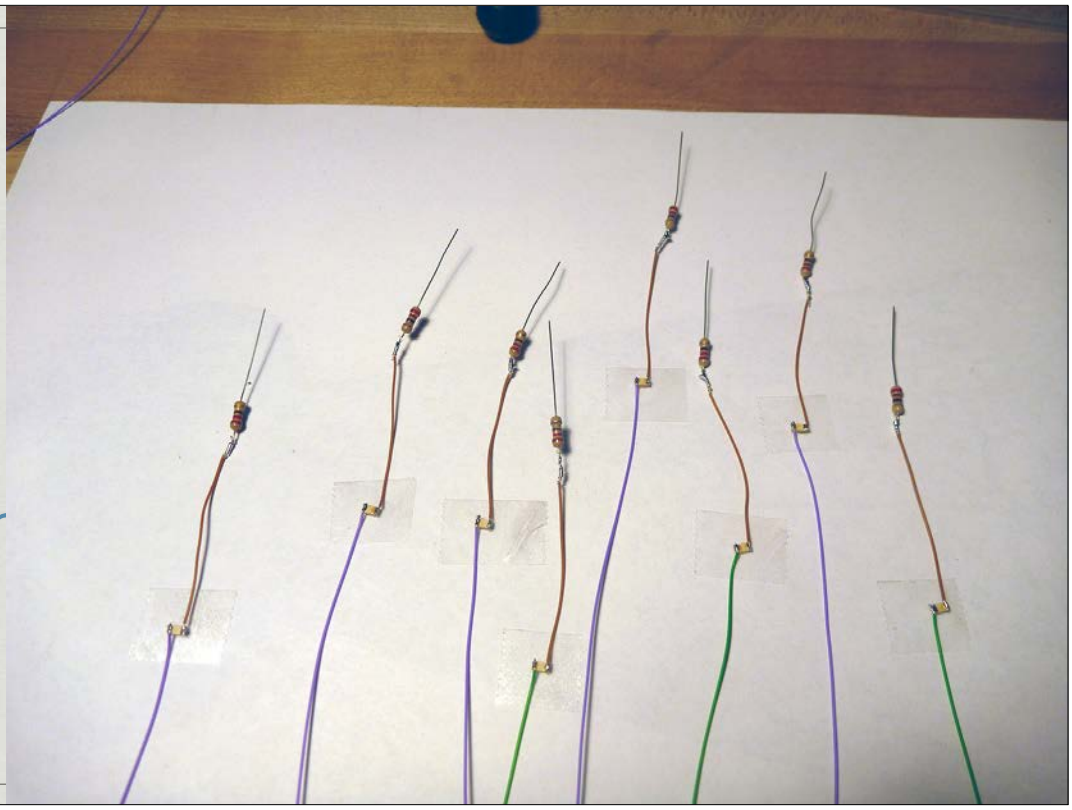
You can also clip the stripped ends of the wire short before soldering. That's the way I usually do it now… These pictures are older…

Another view (longer wires…)

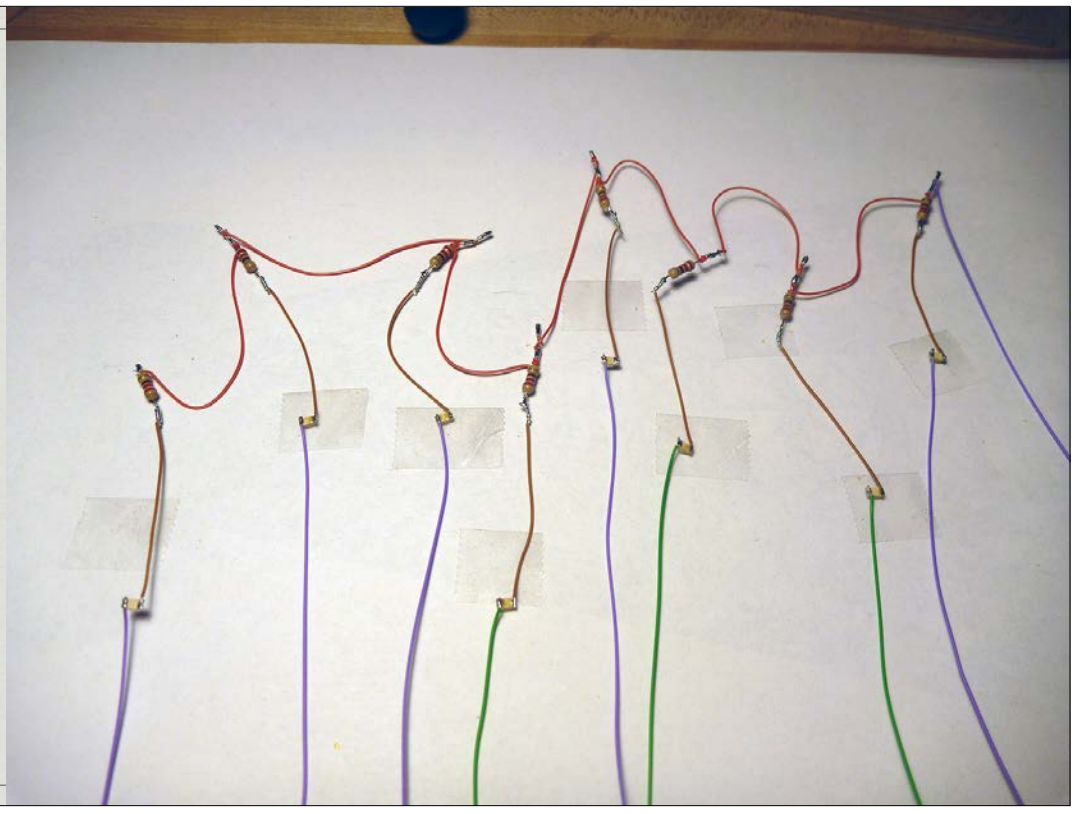


Add resistors (full sized)

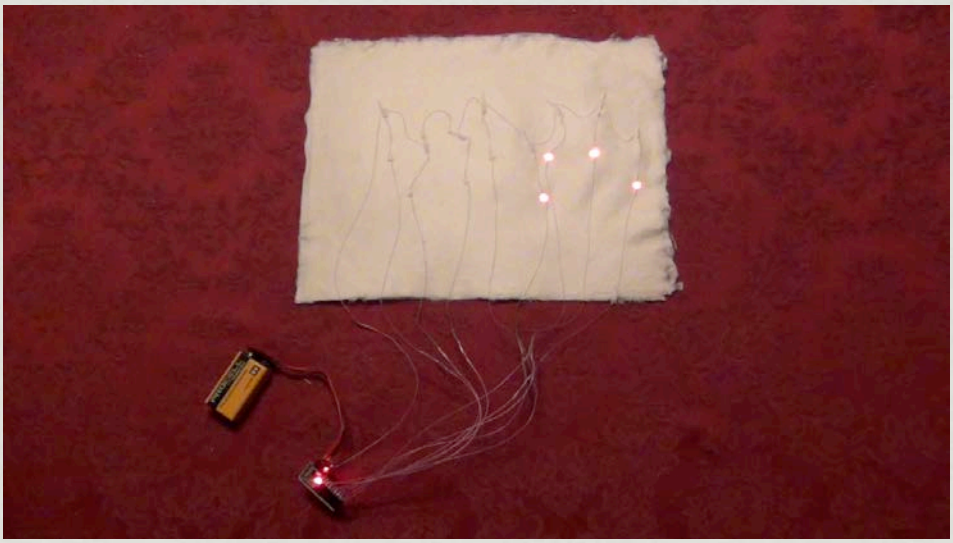
Wire-wrap, then solder



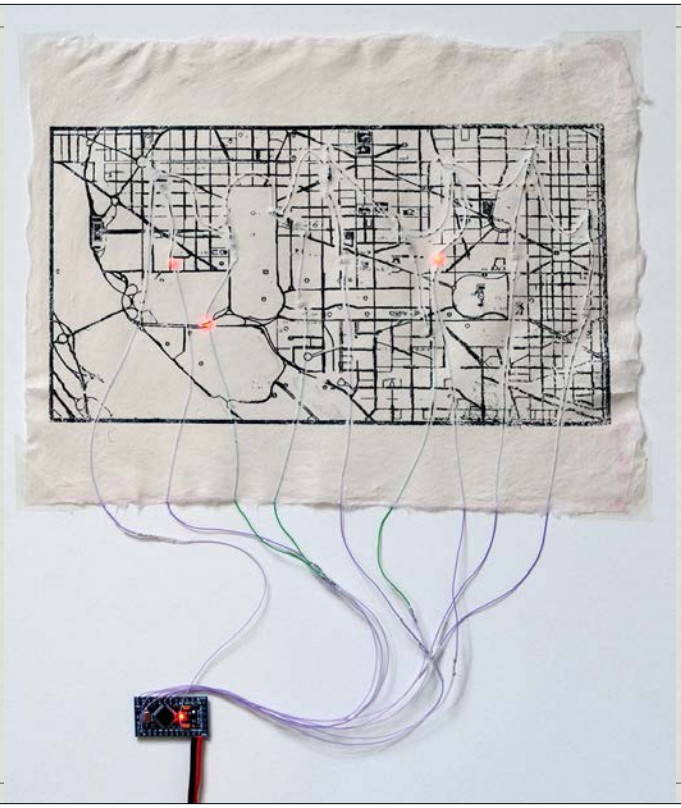
Common anode (or cathode …)



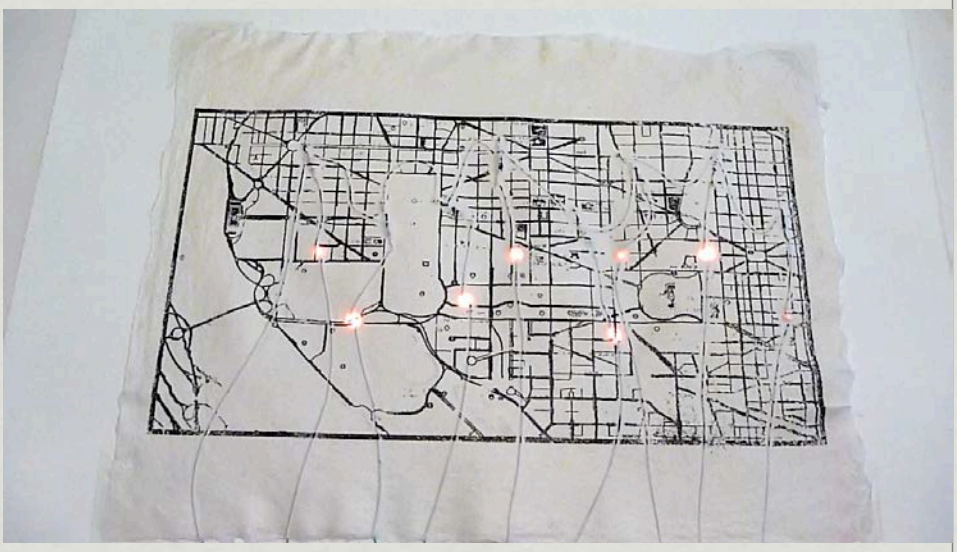
Result - with Arduino Mini



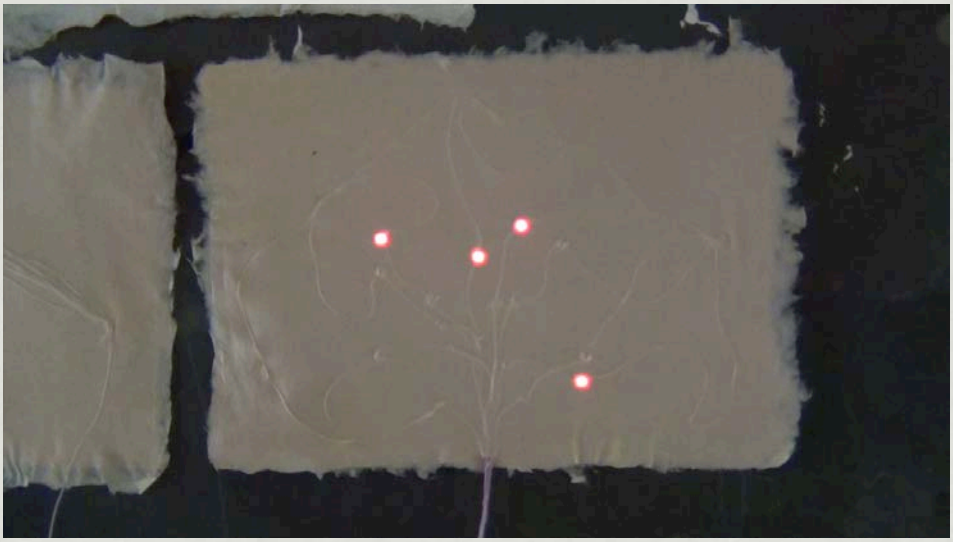
Example: LED paper with Xerox transfer and Arduino mini



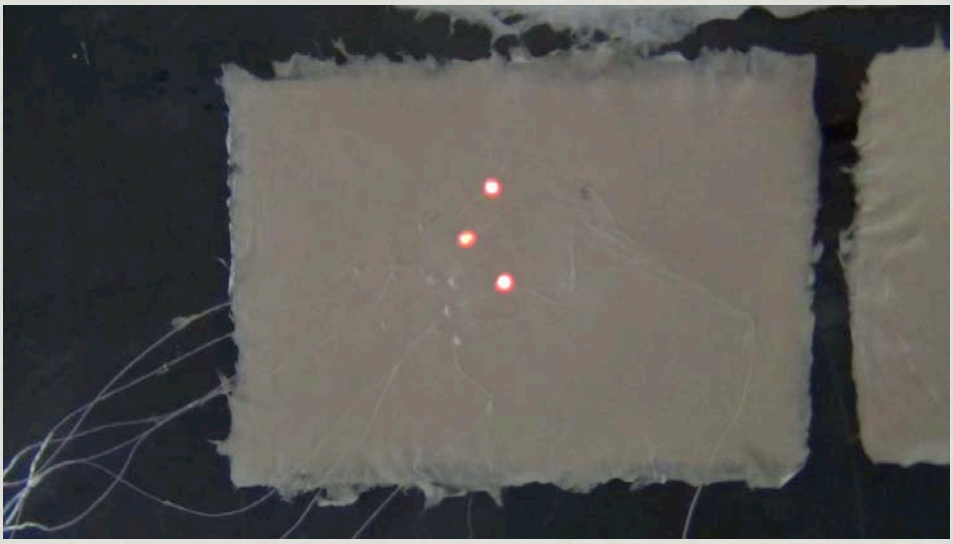
Example: LED paper with Xerox transfer and Arduino mini



Another example

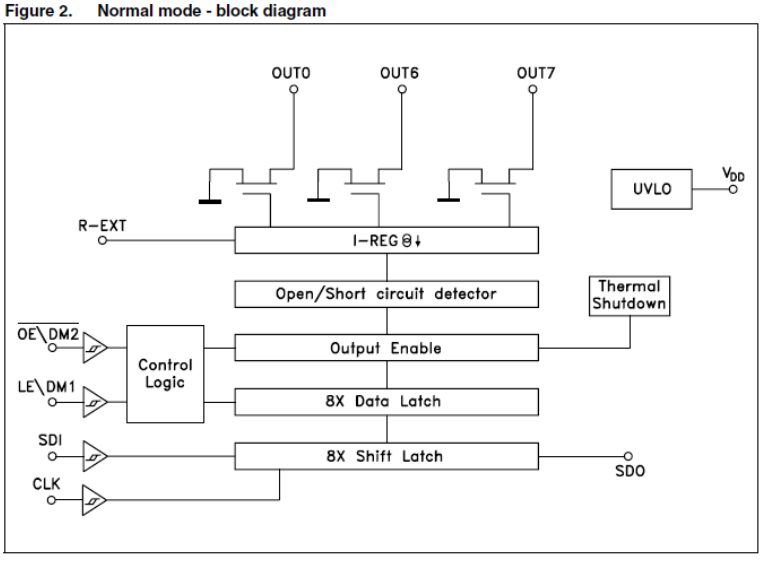


Another example



Example: STP08DP05

Figure 2. Normal mode - block diagram



SDI/CLK shifts data into the 8-bit shift-register

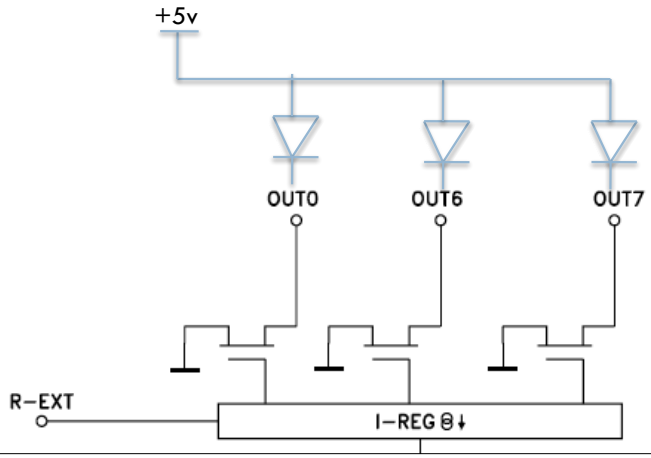
LE moves data to the "data latch" so that it can be seen on the output

OE controls whether the data is enabled to drive the outputs

R-EXT sets the current limit for all outputs

Constant Current Source

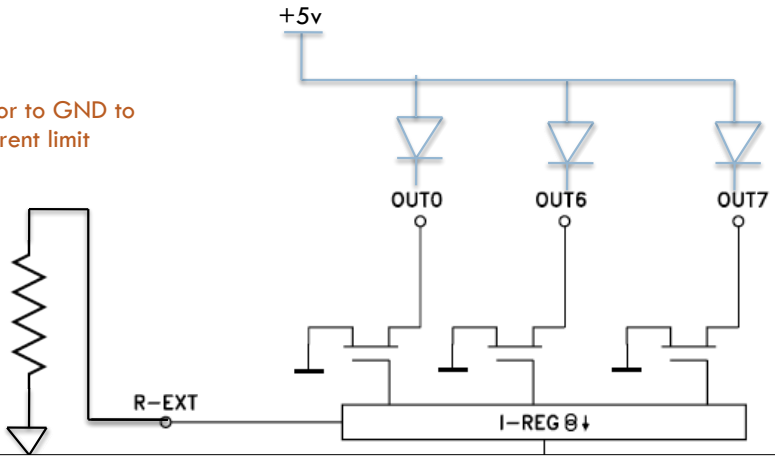
- Note that the constant current source only pulls to ground
  - So – LEDs connect to vdd…



Constant Current Source

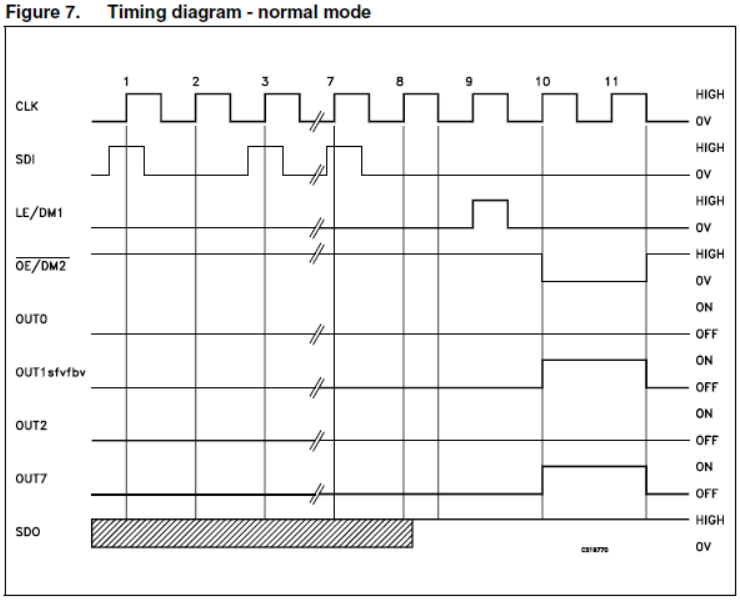
- Note that the constant current source only pulls to ground
  - So – LEDs connect to vdd…

Resistor to GND to set current limit



Example: STP08DP05

Figure 7. Timing diagram - normal mode



Timing diagram shows shifting data in, one bit per clock

Data is transferred to output register on a high LE

Data shows up only when OE is low

This means you can dim all 8 LEDs using PWM on the OE signal

Arduino Code

- Arduino has a built-in function to shift data out for devices like this (most micros do…)

Syntax

shiftOut(dataPin, clockPin, bitOrder, value)

Parameters

dataPin: the pin on which to output each bit (*int*)

clockPin: the pin to toggle once the **dataPin** has been set to the correct value (*int*)

bitOrder: which order to shift out the bits; either **MSBFIRST** or **LSBFIRST**.
(Most Significant Bit First, or, Least Significant Bit First)

value: the data to shift out. (*byte*)

Returns

None

Internal Arduino Code for shiftOut()

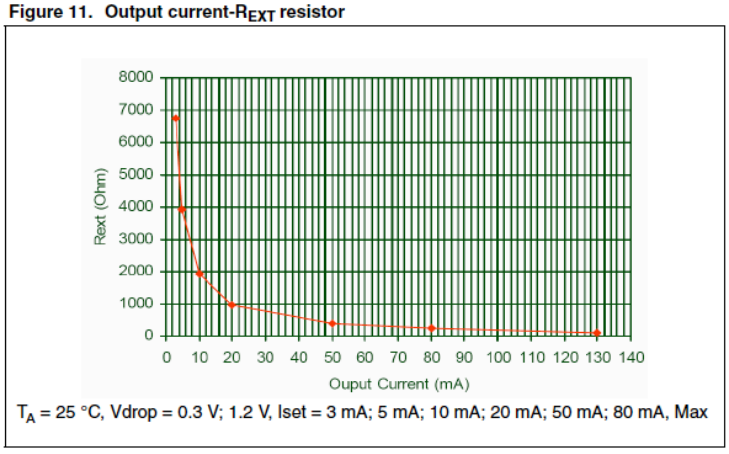
```
void shiftOut(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, byte val)
  {int i;
    for (i = 0; i < 8; i++) {
        if (bitOrder == LSBFIRST)
            digitalWrite(dataPin, !!(val & (1 << i)));
        else
            digitalWrite(dataPin, !!(val & (1 << (7 - i))));

        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
        }
  }
```

Choosing a Resistor

- I chose a 2k ohm resistor for around 10ma

Figure 11. Output current-R_{EXT} resistor



T_A = 25 °C, Vdrop = 0.3 V; 1.2 V, Iset = 3 mA; 5 mA; 10 mA; 20 mA; 50 mA; 80 mA, Max

Table 10. Output current-R_{EXT} resistor

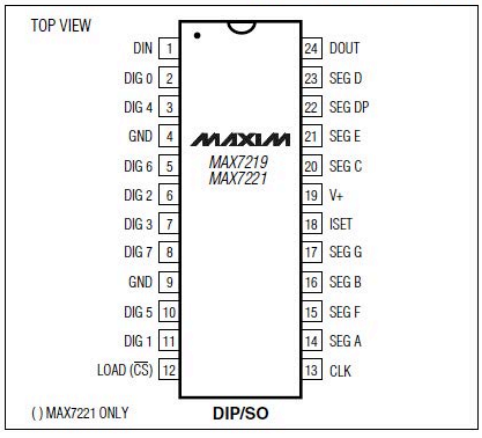
Output current (mA)	3	5	10	20	50	80	130
Rext (Ω)	6740	3930	1913	963	386	241	124

Maximum output current capabilities setting was 130 mA applyingan Rext = 124 Ω

Example: MAX 7219

- Display driver for 8-digits of 7-segment numbers
  - Can also be used for 8x8 array of LEDS
    - (i.e. 64 individual LEDs)
- Drives common-cathode LED digits or LED matrix
  - Cycles between each of 8 digits (or matrix rows) fast enough so they all look ON
- SPI interface
  - Slightly complicated command/data interface
  - Send address of internal register followed by data
  - Each SPI communication is 16 bits
  - Luckily, there's an Arduino library for the chip

Pin Configuration



Typical Application Circuit

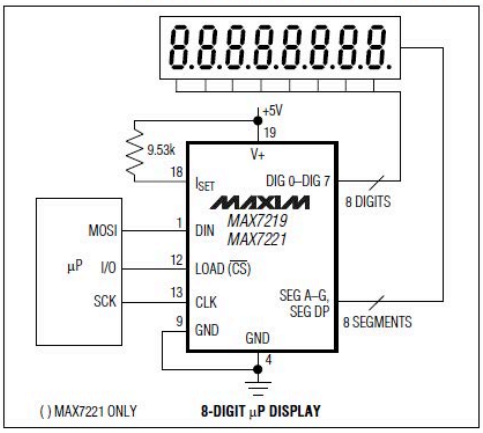
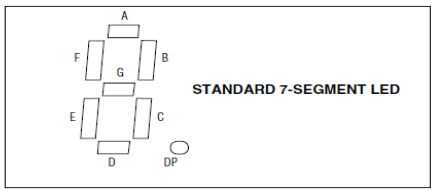


Table 6. No-Decode Mode Data Bits and Corresponding Segment Lines



	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Corresponding Segment Line	DP	A	B	C	D	E	F	G

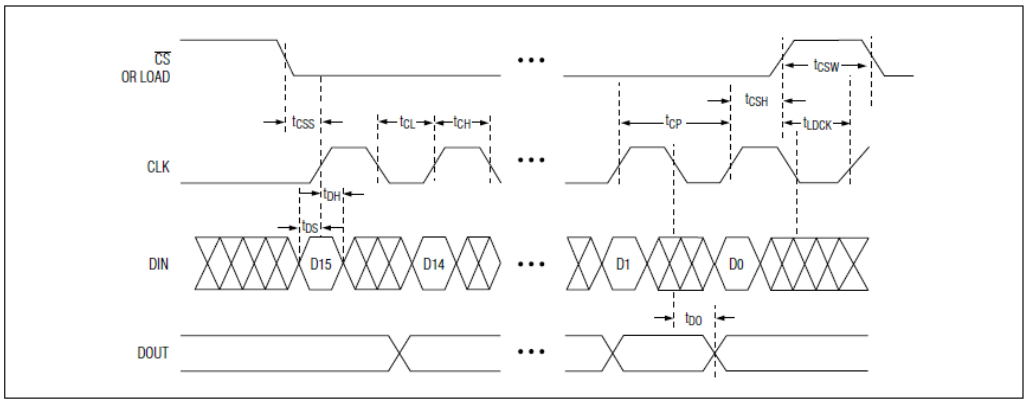


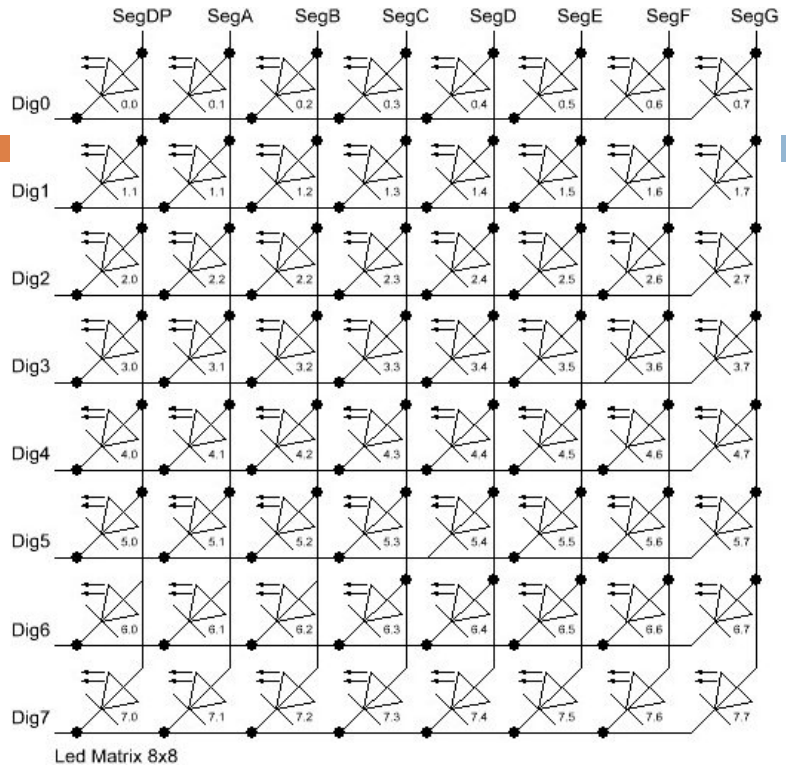
Figure 1. Timing Diagram

Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

Table 2. Register Address Map

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF



Common-Cathode LED array

MAX 7219

- On the one hand – just like STP08DP05
- On the other hand, more complex internal structure
  - Each SPI transfer needs to be 16 bits – address/data
- (at least) Two Arduino libraries available
  - Matrix – built-in to Arduino environment
  - LedControl – download from Playground – more complex control

LedControl Library

```
/* We start by including the library */
#include "LedControl.h"

/*  Make a new instance of an LedControl object
 * Params :
 * int dataPin    The pin on the Arduino where data gets shifted out (Din on MAX)
 * int clockPin   The pin for the clock (CLK on MAX)
 * int csPin      The pin for enabling the device (LD/CS on MAX)
 * int numDevices The maximum number of devices that can be controlled
 */
LedControl lc1 = LedControl(12,11,10,1);
```

LedControl Library

```
void clearDisplay(int addr);
void setLed(int addr, int row, int col, boolean state);
void setRow(int addr, int row, byte value);
void setColumn(int addr, int col, byte value);
void setDigit(int addr, int digit, byte value, boolean dp);
void setChar(int addr, int digit, char value, boolean dp);

/*
 * Display a character on a 7-Segment display.
 * There are only a few characters that make sense here :
 * '0','1','2','3','4','5','6','7','8','9','0',
 * 'A','b','c','d','E','F','H','L','P',
 * ' ','-','_',
 */
```

LedControl Library

```
//include this file so we can write down a byte in binary encoding
#include <binary.h>

//now setting the leds in the sixth column on the first device is easy
lc.setColumn(0,5,B00001111);

//now setting the leds from the third row on the first device is easy
lc.setRow(0,2,B10110000);

//switch on the led in the 3'rd row 8'th column
//and remember that indices start at 0!
lc.setLed(0,2,7,true);
//Led at row 0 second from left too
lc.setLed(0,0,1,false);
```

LedControl Library

```
void writeArduinoOnMatrix() {
  /* here is the data for the characters */
  byte a[5]={B01111110,B10001000,B10001000,B10001000,B01111110};
  byte r[5]={B00111110,B00010000,B00100000,B00100000,B00010000};
  byte d[5]={B00011100,B00100010,B00100010,B00010010,B11111110};
  byte u[5]={B00111100,B00000010,B00000010,B00000100,B00111110};
  byte i[5]={B00000000,B00100010,B10111110,B00000010,B00000000};
  byte n[5]={B00111110,B00010000,B00100000,B00100000,B00011110};
  byte o[5]={B00011100,B00100010,B00100010,B00100010,B00011100};

  /* now display them one by one with a small delay */
  lc.setRow(0,0,a[0]);
  lc.setRow(0,1,a[1]);
  lc.setRow(0,2,a[2]);
  lc.setRow(0,3,a[3]);
  lc.setRow(0,4,a[4]);
  delay(delaytime);
```

```
lc.setRow(0,0,r[0]);
 lc.setRow(0,1,r[1]);
 lc.setRow(0,2,r[2]);
 lc.setRow(0,3,r[3]);
 lc.setRow(0,4,r[4]);
 delay(delaytime);
```

MAX 7219 – Current Setting Resistor

- This resistor goes to Vdd, NOT GND!
  - Sets current for each segment (LED)

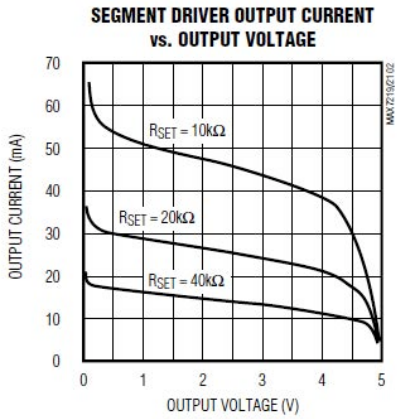
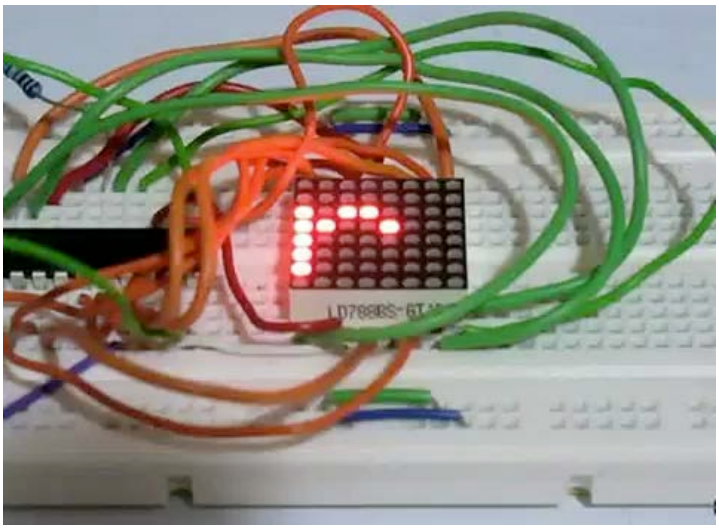


Table 11. R_{SET} vs. Segment Current and LED Forward Voltage

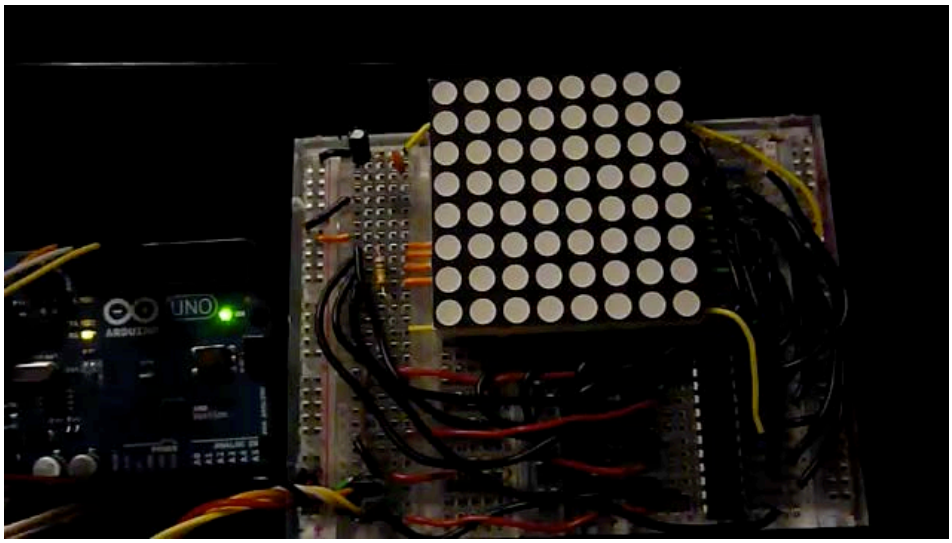
I_{SEG} (mA)	V_{LED} (V)				
	1.5	2.0	2.5	3.0	3.5
40	12.2	11.8	11.0	10.6	9.69
30	17.8	17.1	15.8	15.0	14.0
20	29.8	28.0	25.9	24.5	22.6
10	66.7	63.7	59.3	55.4	51.2

These values are in kOhms!!!

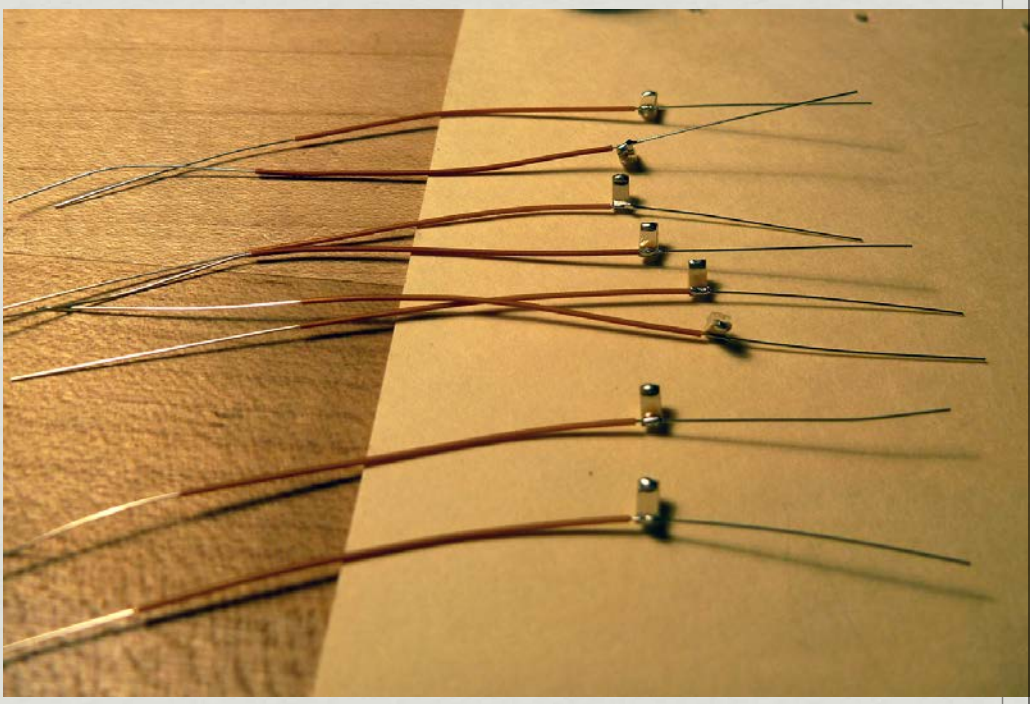
MAX7219 demo



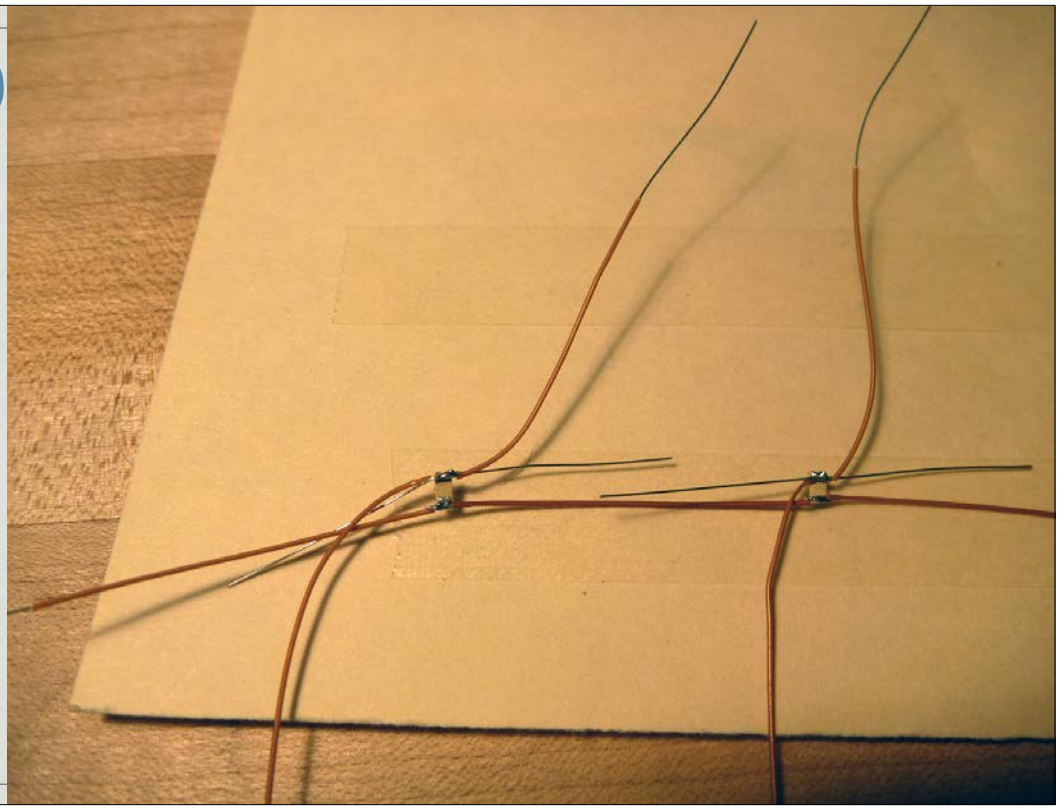
MAX7219 demo



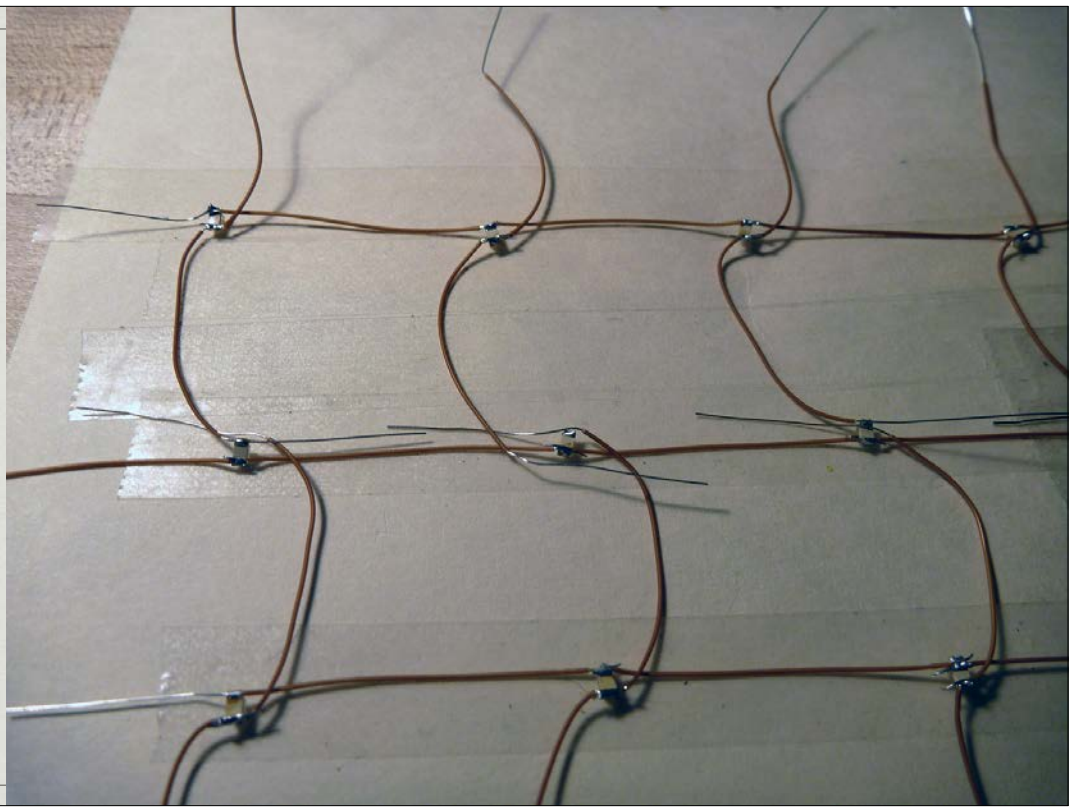
MAX 7219 wants a mesh…



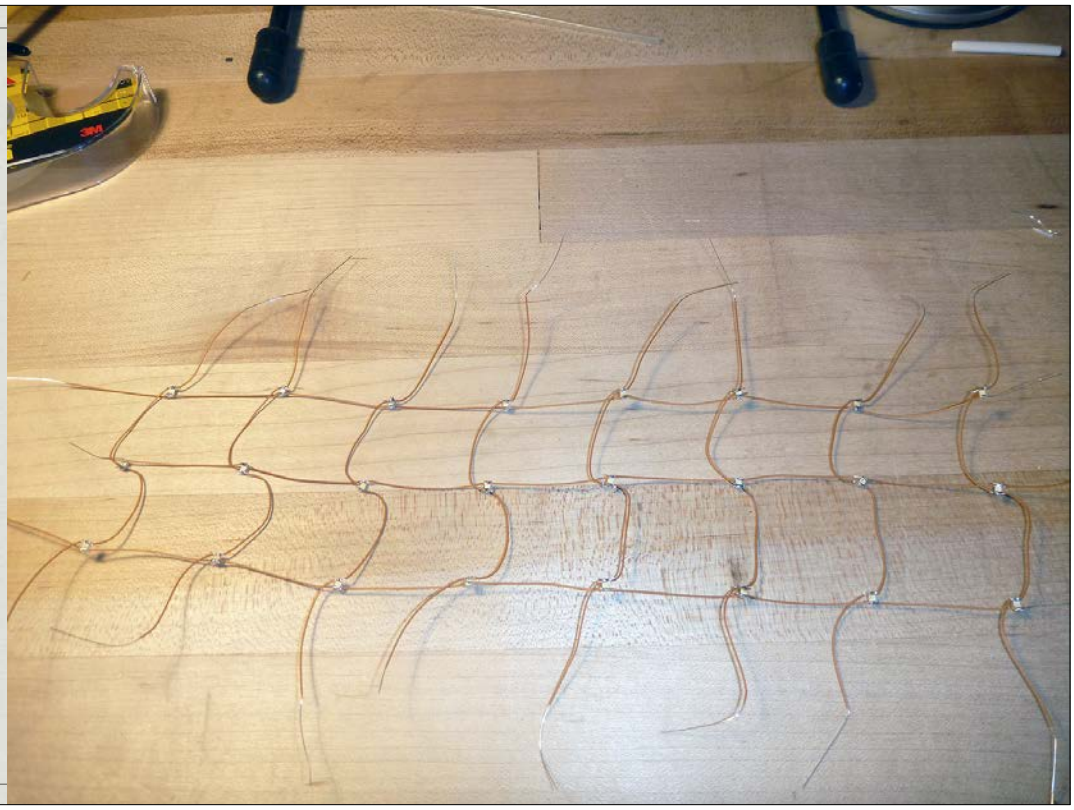
MAX 7219 wants a mesh…



Each (small) node needs two wires…

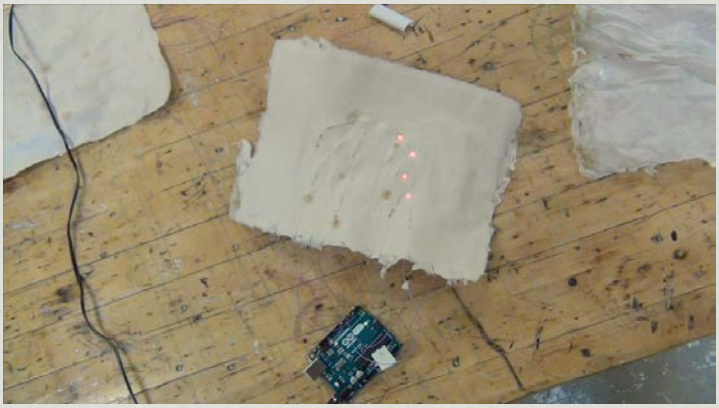


Mesh in progress

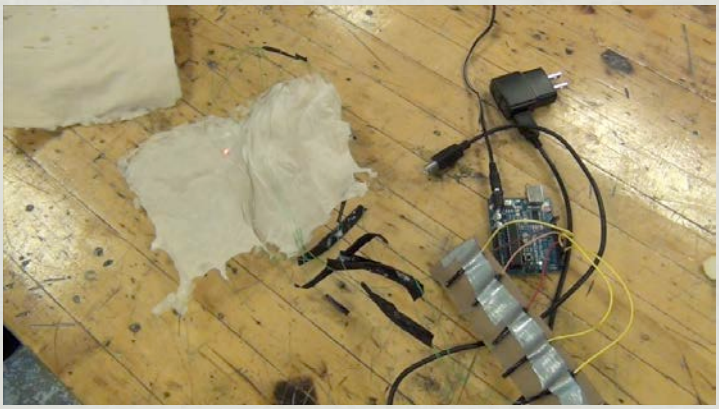


Result - with embedded MAX 7219

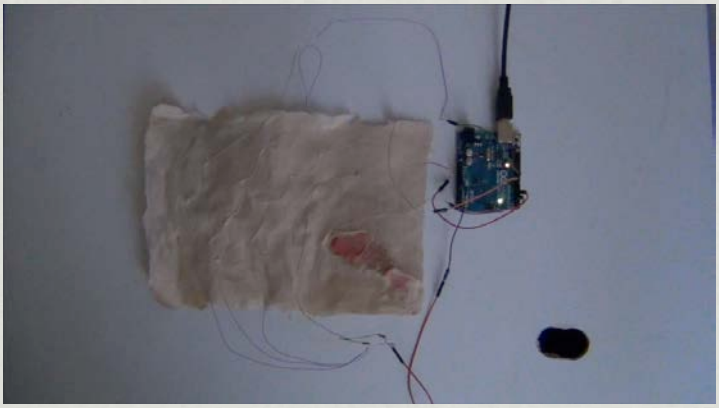
Student Examples



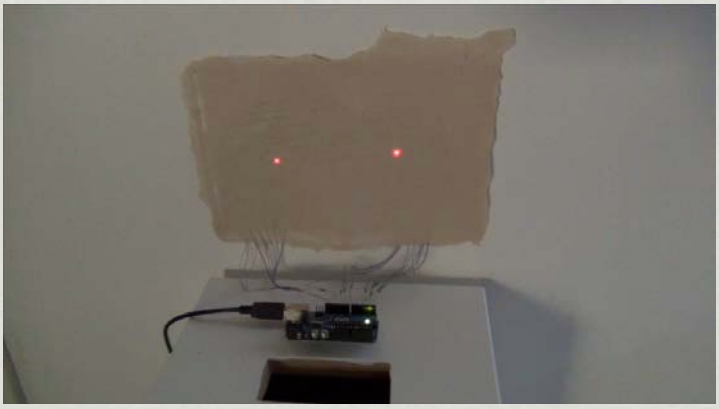
Student Examples



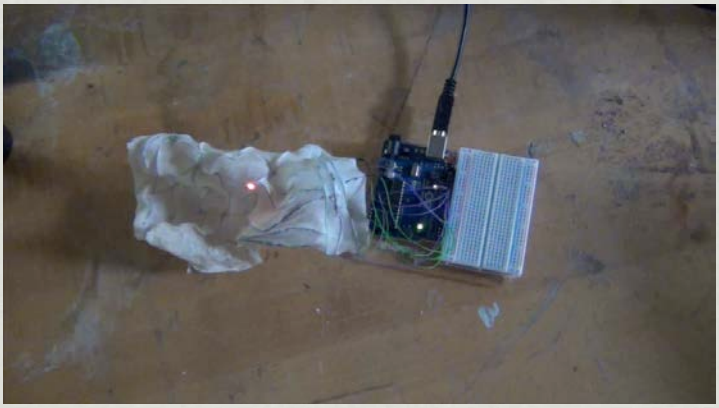
Student Examples



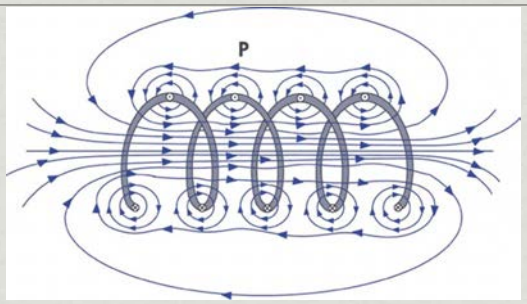
Student Examples

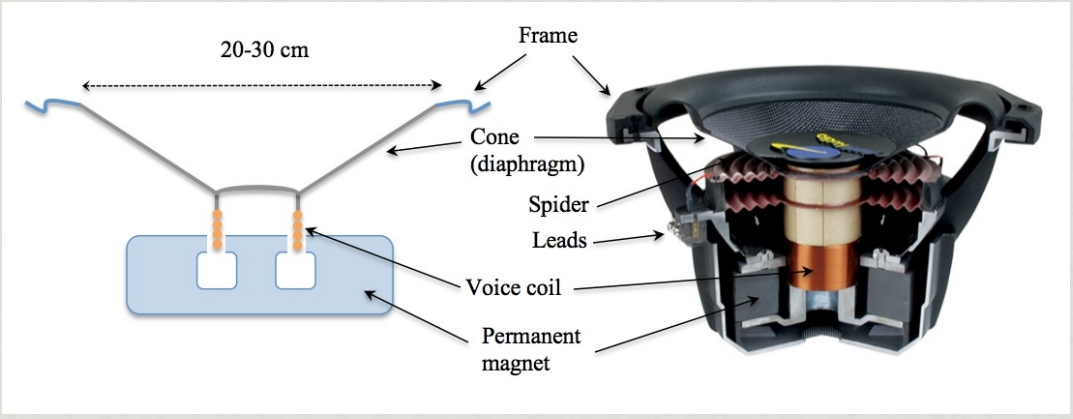


Student Examples

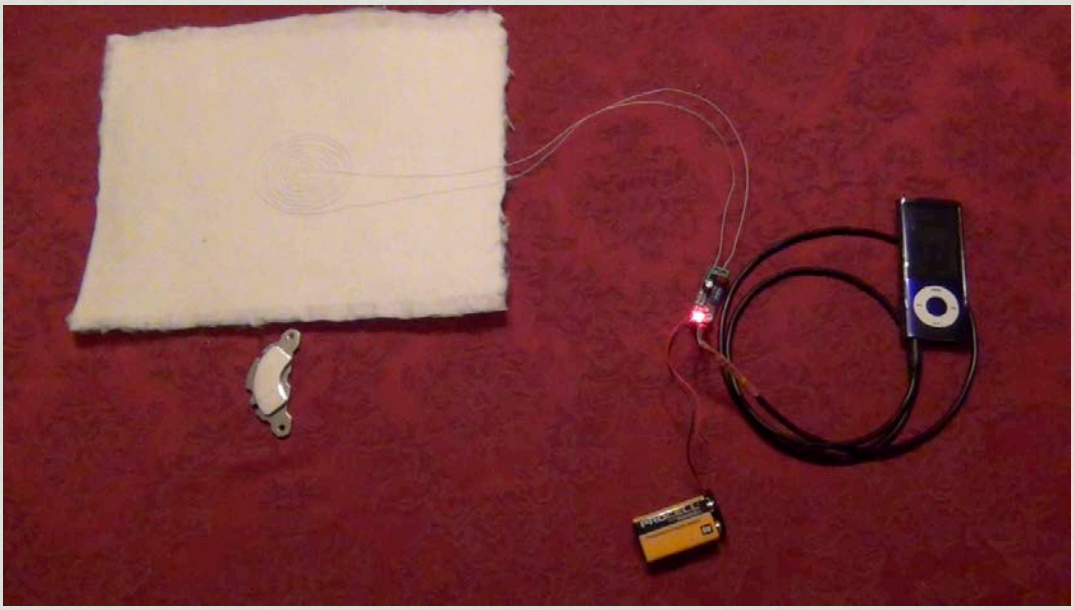


Another Silly Paper Trick

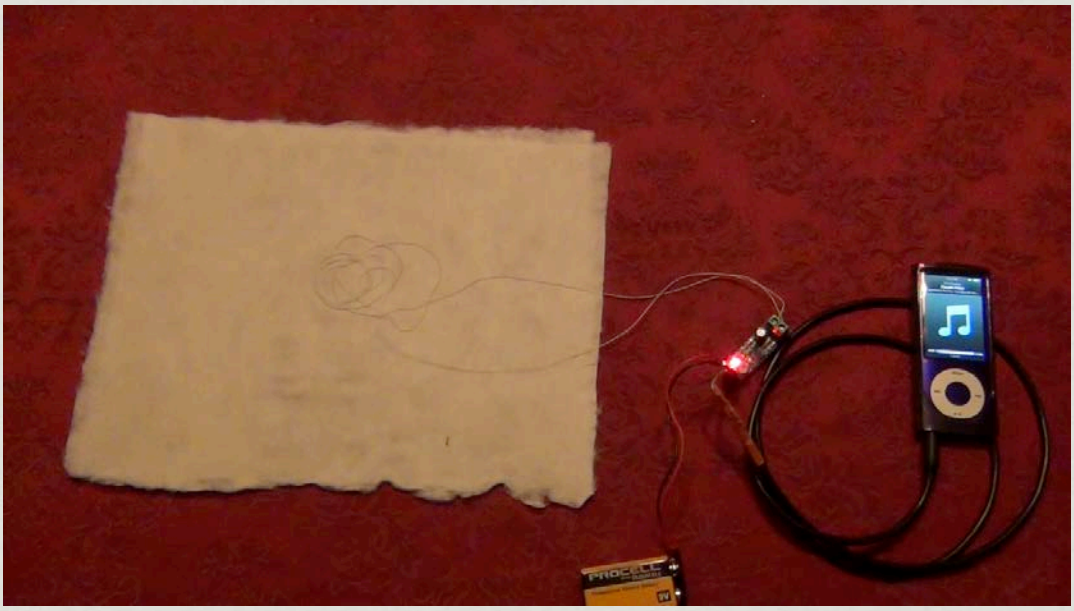




Paper-speaker? Speaker-paper?

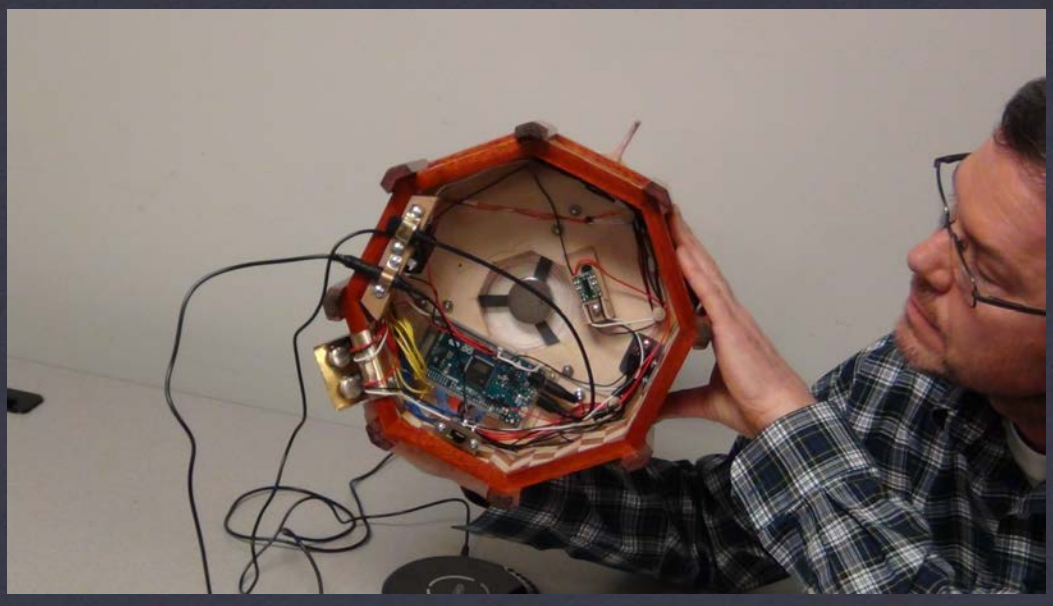


Paper-speaker? Speaker-paper?

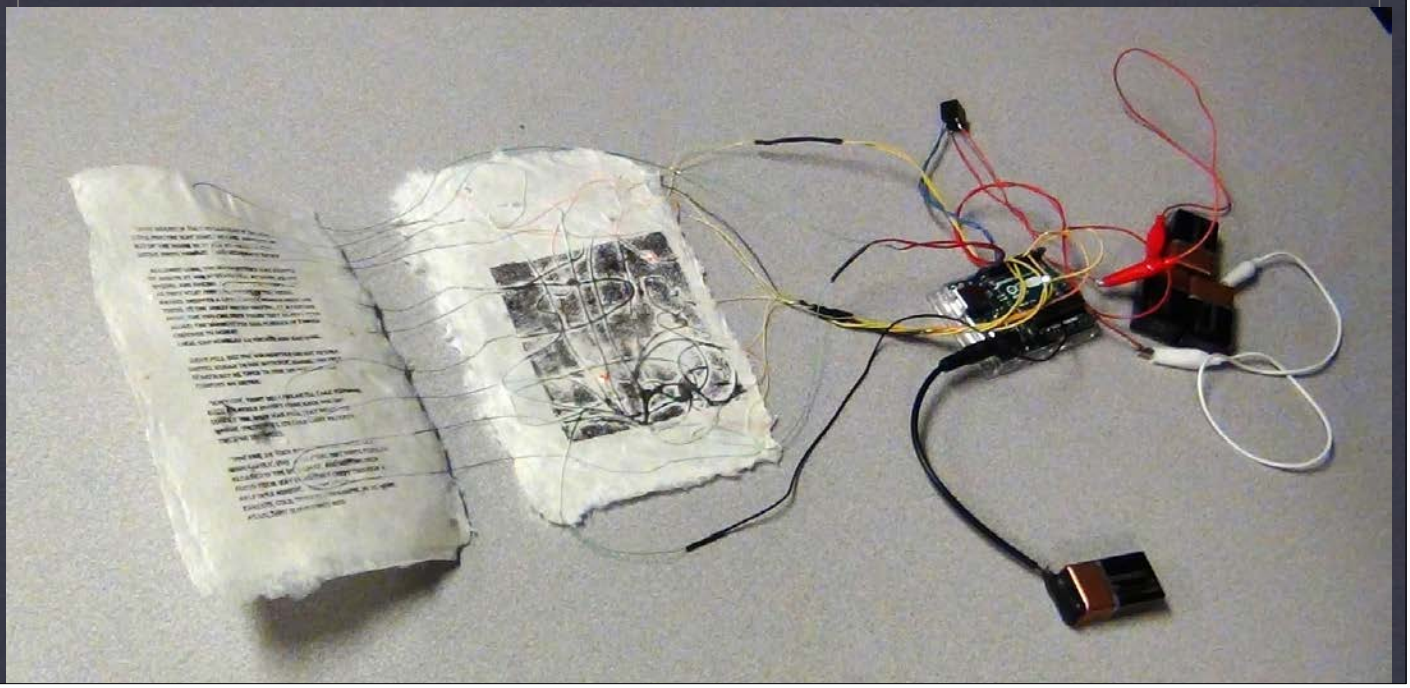


INSPIRATION

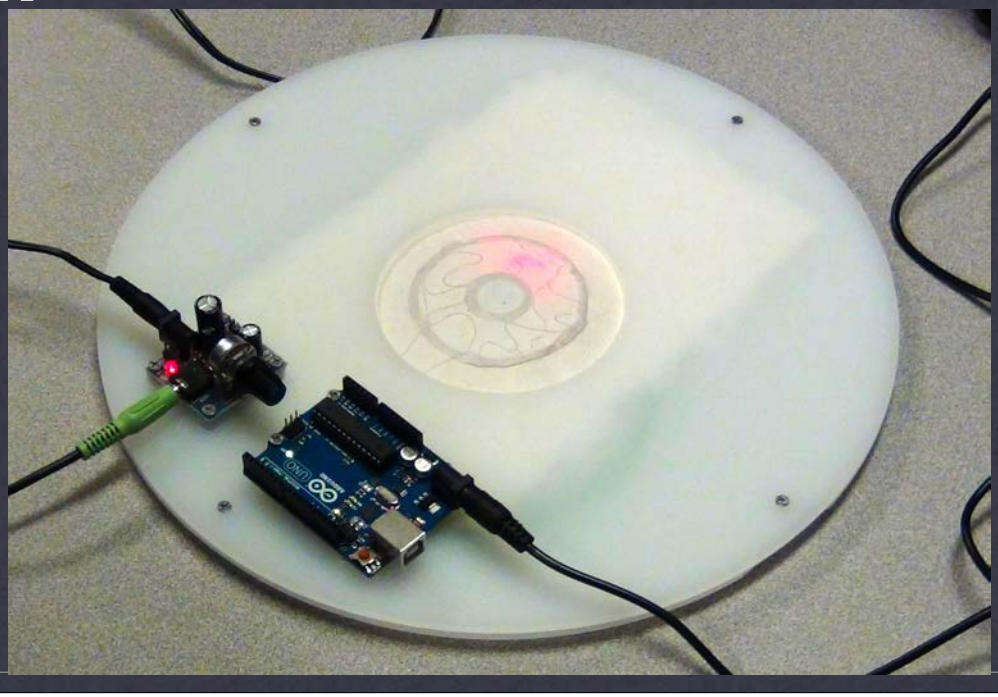
INSPIRATION



INSPIRATION



INSPIRATION

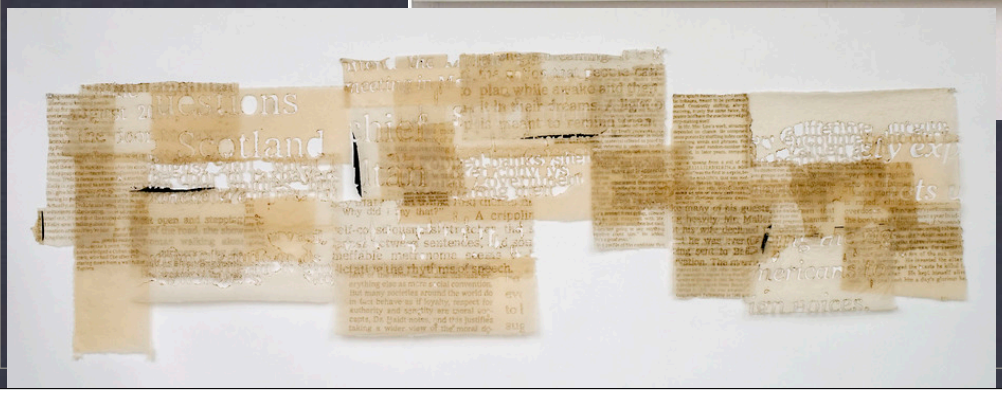


INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION



INSPIRATION

INSPIRATION

INSPIRATION

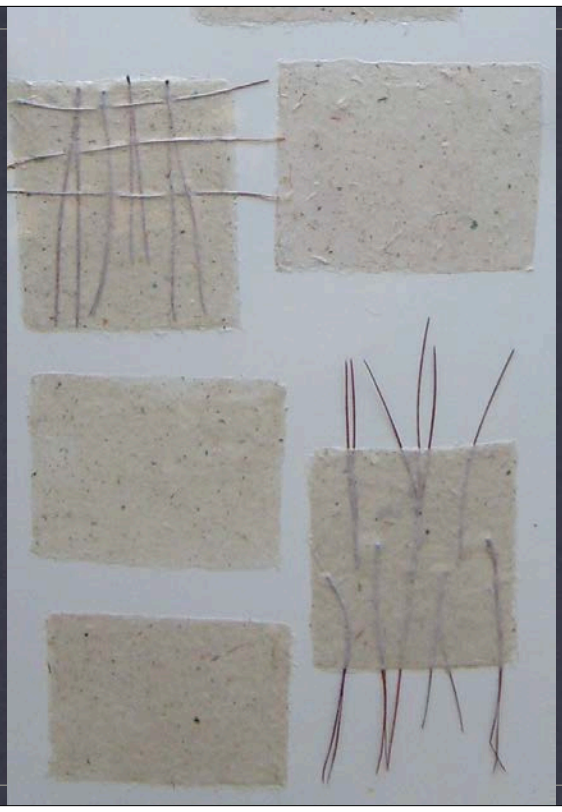
INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION



INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION

INSPIRATION

Paper making

Adjusting the beater

Pre-soaking the “half stuff”

Adding water to the beater

Getting beaten…

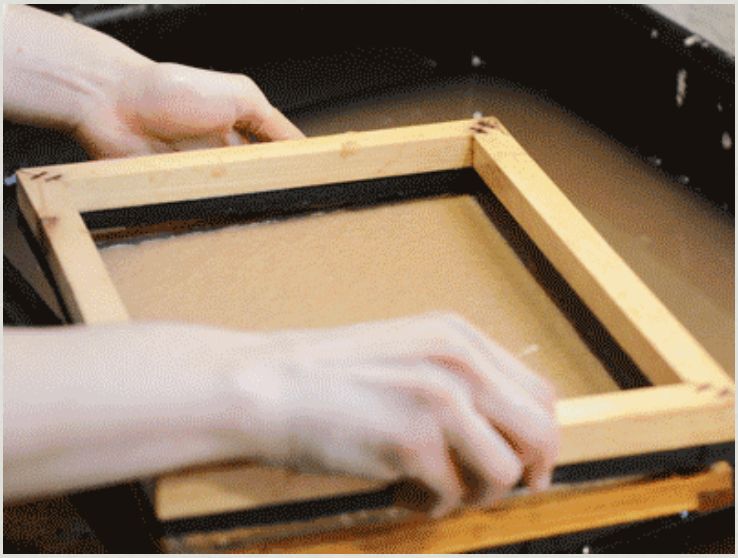
Saving pulp in a bucket for later use

Pull one sheet onto the mold

Pull one sheet onto the mold

Couch the sheet onto the felts

Couch the sheet onto the felts



Place your LEDS on the sheet, and “double couch” another on top to seal the LEDs inside the paper