Bits & Bots

Anna Gerber

Bits & Bots Sessions

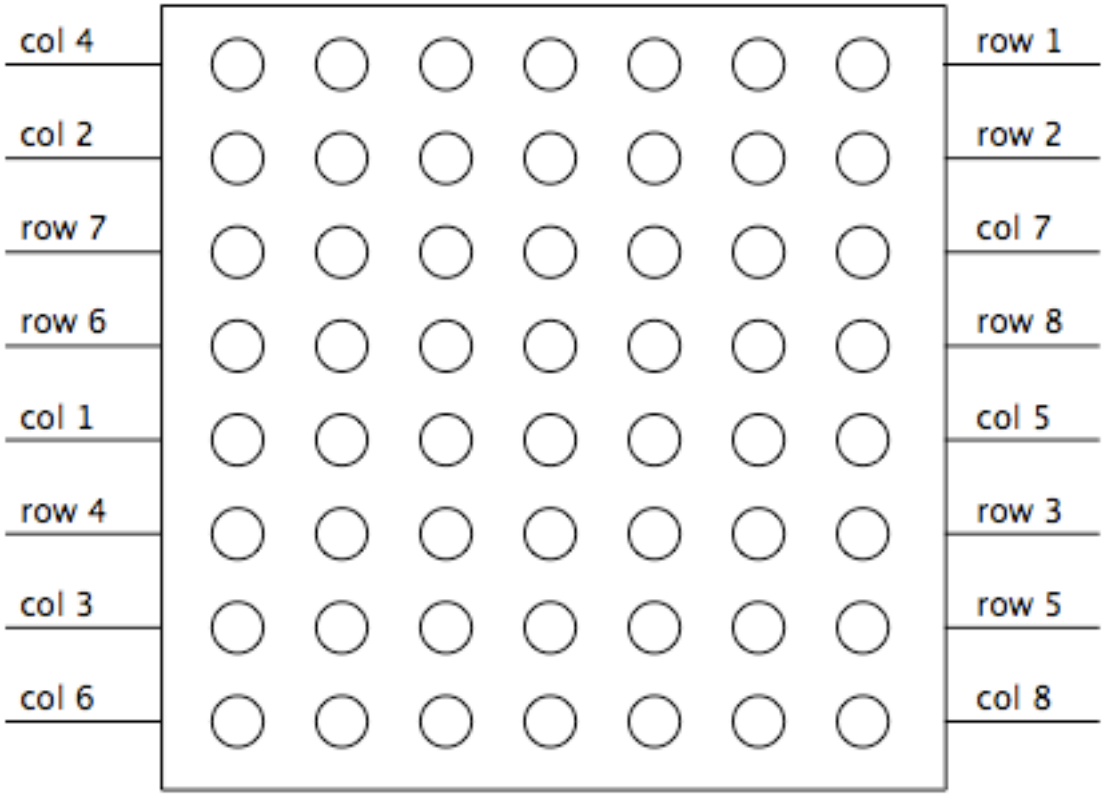
Session	Topic
Tuesday 20th May, 6 – 8pm	**Intro to 3D Design**: Design custom robot parts to print on the 3D printers
Tuesday 27th May, 6 – 8pm	**Intro to Electronics**: Learn how the electronic parts in the kit work, design our robot circuits
Tuesday 3rd Jun,e 6 – 8 pm	**Intro to Arduino**: Write NodeJS programs to read from sensors and control actuators
7th June, 1 – 5 pm	**Intermediate 3D Design**: Design more complex robot parts: gears, claws etc
14th June, 1 – 5 pm	**Intermediate Arduino**: Develop our robots' locomotion, sensing and responding behaviours
21st June, 1 – 6 pm	**Advanced Bits & Bots**: Finalise robot design and assembly, develop advanced robot control programs

Bits & Bots Slides etc

Slides and other materials for the course will be published after each session here:

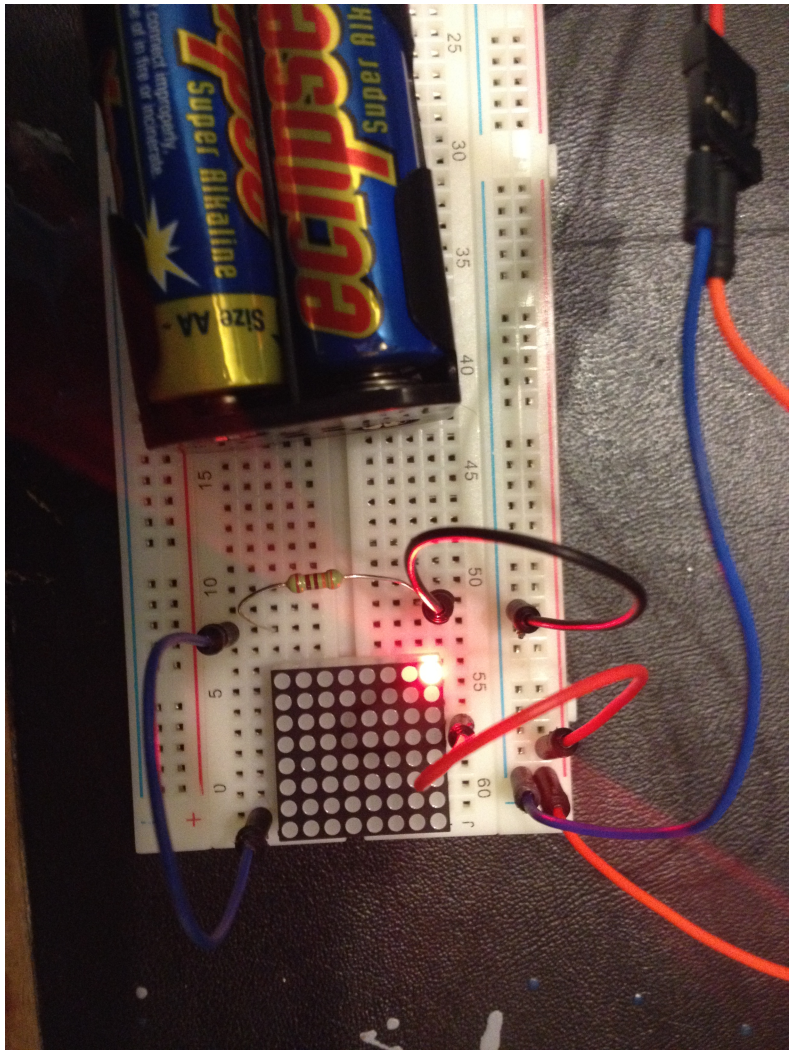
https://github.com/AnnaGerber/bits-n-bots

LED Matrix Pin Out



LEDs light up when corresponding column has 5V and row is connected to ground.
Include current limiting resistor for each row

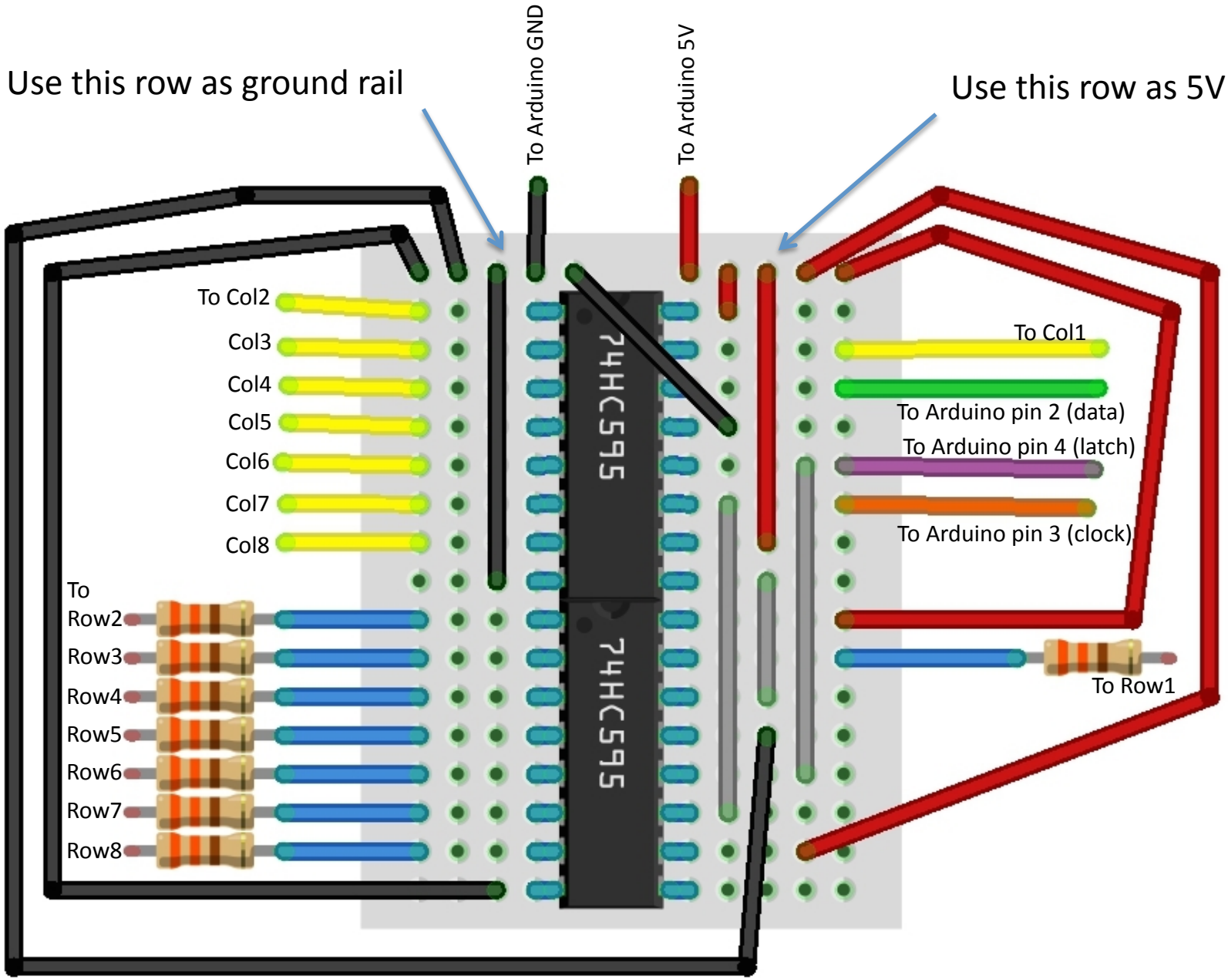
Manually lighting up LEDs



- Connect battery 5V and GND to breadboard rails
- Connect ground rail to resistor via a jumper wire and then to a row via another jumper wire
- Connect 5V rail to a column using a jumper wire
- Move the jumper wires to different rows/columns to light a different LED

Shift Registers

- We could connect every row and column pin to pins on an Arduino to control the pattern displayed but this would require a lot of pins
- We will use shift registers to reduce the number of pins required to control the LED Matrix
- How Shift Registers Work: https://www.youtube.com/watch?v=Z5iFTBJWz68


Use this row as ground rail
To Arduino GND
To Arduino 5V
Use this row as 5V rail
To Col2
Col3
Col4
Col5
Col6
Col7
Col8
74HC595
74HC595
To Col1
To Arduino pin 2 (data)
To Arduino pin 4 (latch)
To Arduino pin 3 (clock)
To Row1
To Row2
Row3
Row4
Row5
Row6
Row7
Row8

Create a ShiftRegister object

```
var five = require("johnny-five"),
    board, shiftRegister;

board = new five.Board();

board.on("ready", function() {
  shiftRegister = new five.ShiftRegister({
    pins: {
      data: 2,
      clock: 3,
      latch: 4
    }
  });

  this.repl.inject({
    sr: shiftRegister
  });
```

Matrix Patterns

- Patterns are represented in binary (each digit represents an LED) and then converted to Hexadecimal values

Pattern	Binary Value	Hex value	Decimal Value
Row or column 8 only	00000001	0x01	1
7 only	00000010	0x02	2
6 only	00000100	0x04	4
5 only	00001000	0x08	8
4 only	00010000	0x10	16
3 only	00100000	0x20	32
2 only	01000000	0x40	64
1 only	10000000	0x80	128

LED on = 1
LED off = 0

Sending patterns

We are shifting an 8 bit pattern in to the first register then a second pattern which will push the first value into the second register.

```
function send2(value, value2) {
  board.digitalWrite(4, board.io.LOW);
  board.shiftOut(2, 3, true, (0xff - value));
  board.shiftOut(2, 3, true, value2);
  board.digitalWrite(4, board.io.HIGH);
};
shiftRegister.send2 = send2;
```

In REPL:

- `sr.send2(0,0) // clear all rows and columns`
- `sr.send2(0xff,0xff) // set all LEDs on`

Creating your own patterns

- Create a pattern for the rows and a pattern for the columns
- Use a calculator to convert the binary number to Hexadecimal or an app e.g.: http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

Creating asymmetrical patterns

- We can only send symmetrical patternsdirectly via the shift register (we could connect to the pins directly or use an alternative controller chip to talk to individual pins)
- To get around this, flash different parts of the pattern in a loop e.g:

```
function flashyX(){
  board.loop(40, function(){
    shiftRegister.send2(0x81,0x81);
    board.wait(10,function(){shiftRegister.send2(0x42,0x42);})
    board.wait(20,function(){shiftRegister.send2(0x24,0x24);})
    board.wait(30,function(){shiftRegister.send2(0x18,0x18);})
  })
}
```