Charlieplexing LEDs with Arduino

Dale Wheat

14 September 2019

Dallas Makerspace

Introduction

- This class will introduce you to some methods of controlling LEDs using an Arduino
- We'll start with simple LED blinking projects and work our way up to more complex arrangements
- No previous Arduino experience is required
- No previous electronics experience is required

Course Outline

- Introduction
- Software setup
- Exercise 1: Blink a single LED
- Exercise 2: Blink an external LED
- Exercise 3: Active low LEDs
- Exercise 4: Blink using a different pin
- Review
- Exercise 6: Multiplexing techniques
- Exercise 7: Cheating
- Exercise 8: Charlieplexing LEDs

Software Setup

- Install Arduino IDE software on participant's laptops
- Install device driver software (if necessary)
  - CH341SER.EXE
- Install LEDstick library
- Distribute Arduino UNOs + USB cables

Set up your Arduino

- Plug the Arduino Uno into your laptop using the USB cable provided
- Open the Arduino IDE software
- Select menu item “Tools -> Board -> Arduino/Genuino Uno”
- Select menu item “Tools -> Port -> COM4”
  - Your COM port number may be (probably is) different
  - Just pick the last (or highest numbered) port available

Install the LEDstick library

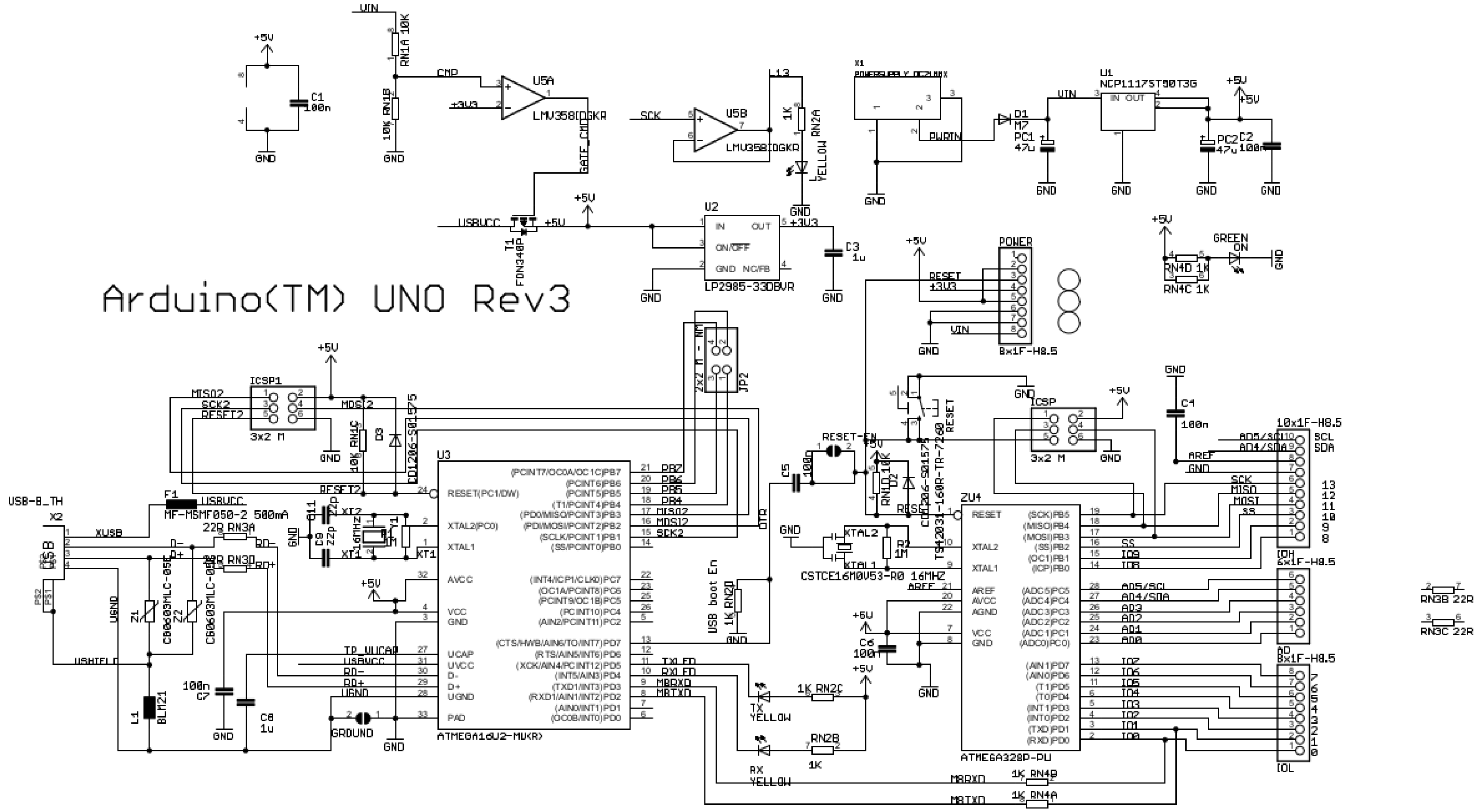
- TODO <instructions>

Exercise 1: Verify “Blink” sketch works

- Select menu item “File -> Examples -> 01.Basics -> Blink”
- Select menu item “Sketch -> Verify/Compile”
  - Look for “Done compiling.” status message
- Select menu item “Sketch -> Upload”
  - Look for “Done uploading.” status message
- Does it blink?

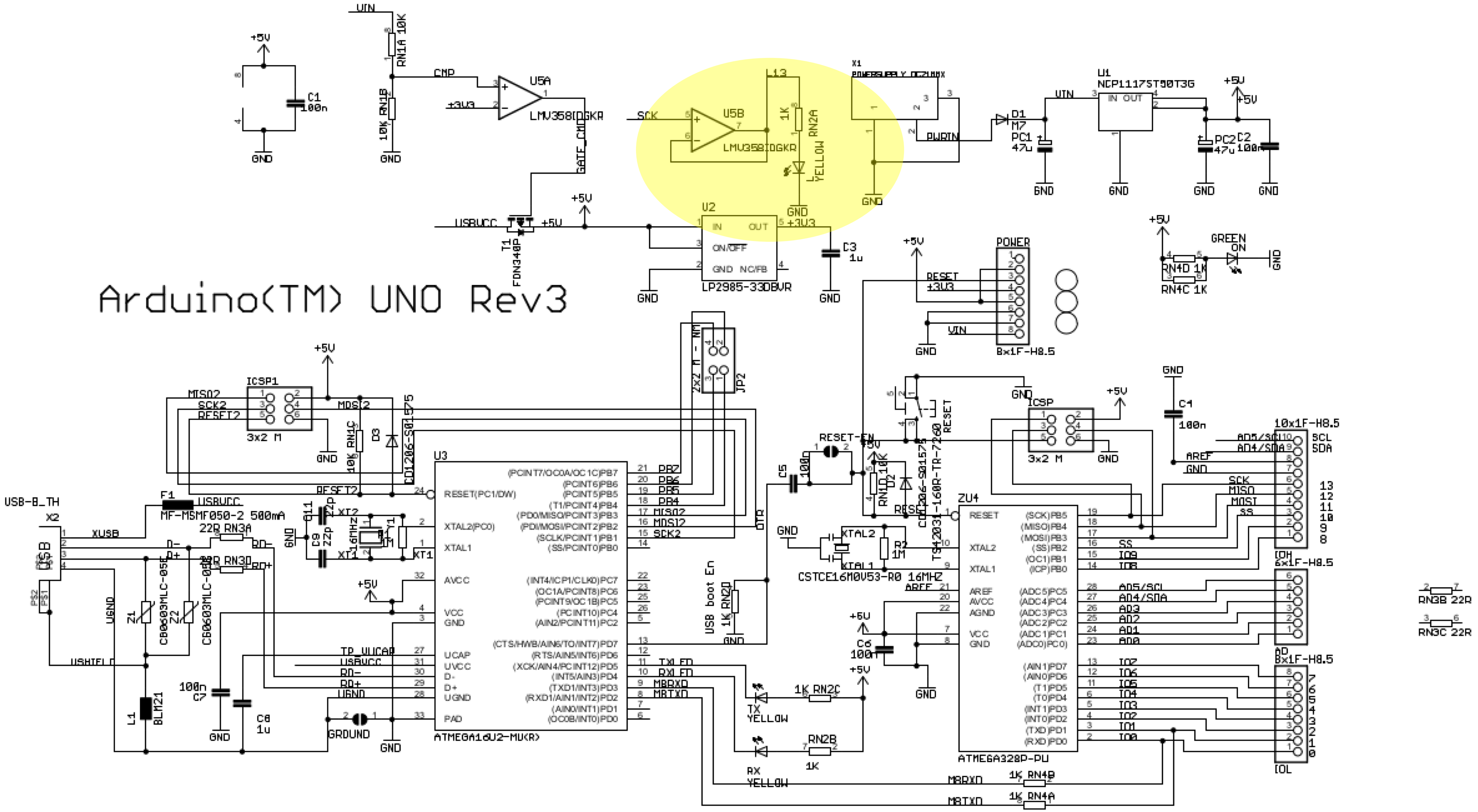
Schematic Diagram

- https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

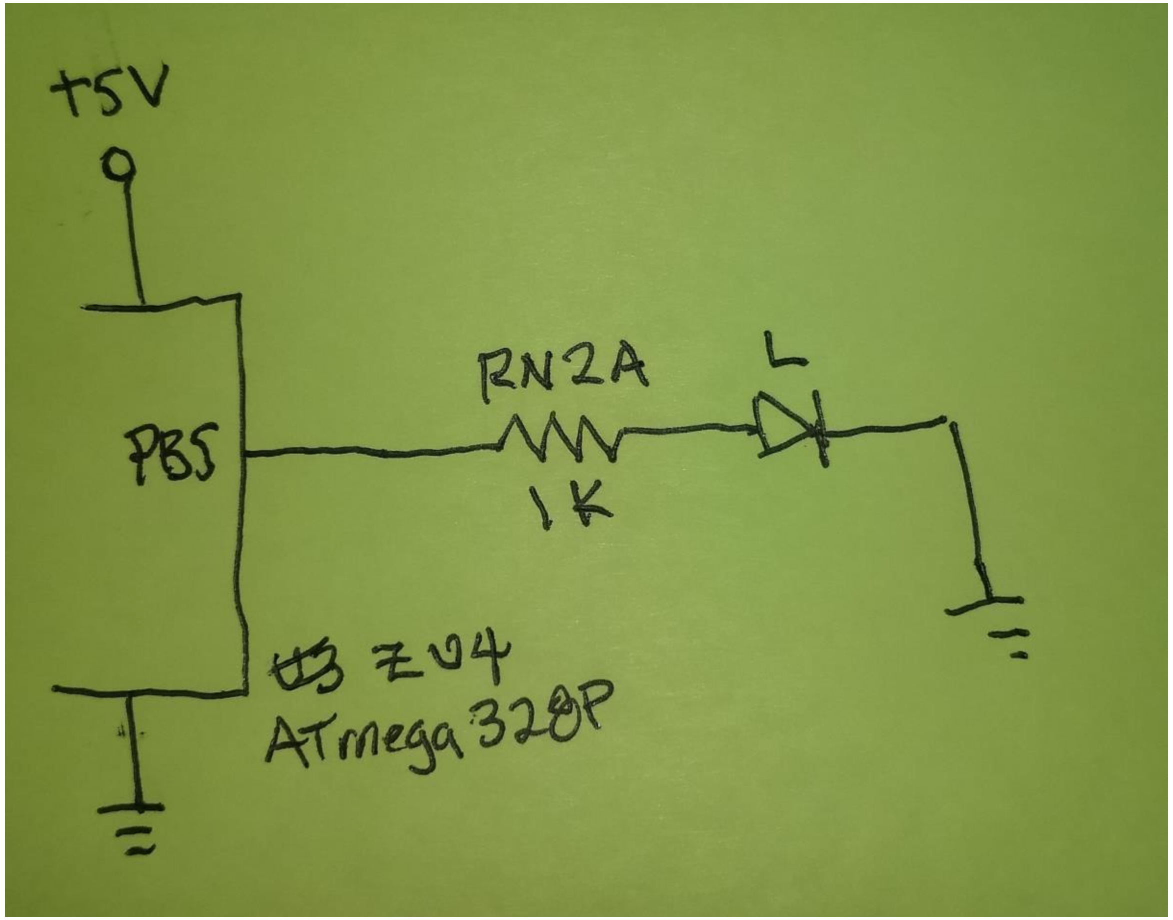


Schematic Diagram

- https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf



Schematic, partial



Analysis of “Blink” sketch

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // short delay
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000); // short delay
}
```

Analysis of “Blink” sketch

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // short delay
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000); // short delay
}
```

Analysis of “Blink” sketch

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // short delay
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000); // short delay
}
```

Analysis of “Blink” sketch

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // short delay
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000); // short delay
}
```

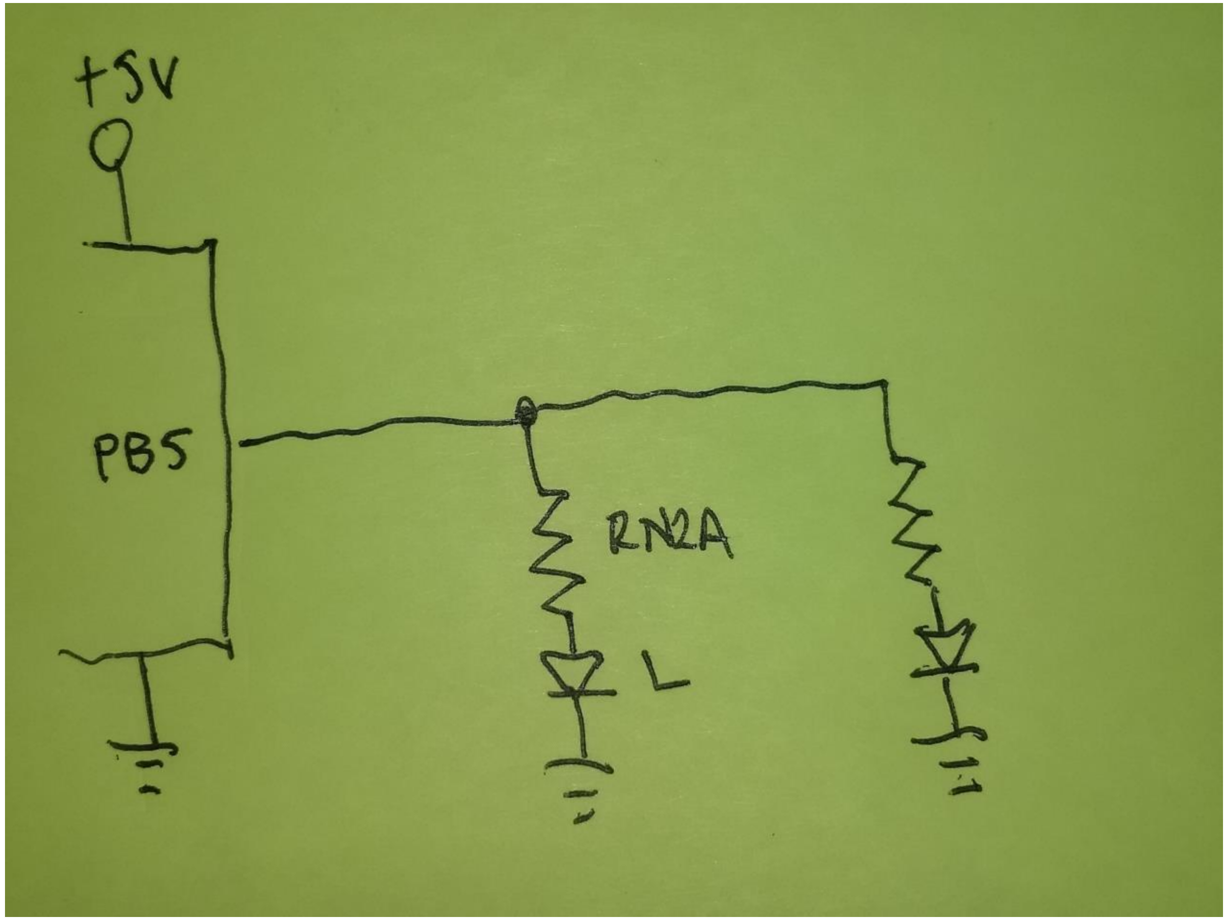
Analysis of “Blink” sketch

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // LED on
  delay(1000); // short delay
  digitalWrite(LED_BUILTIN, LOW); // LED off
  delay(1000); // short delay
}
```

Exercise 2: Blink an external LED

- Use the same “Blink” sketch
- Disconnect the Arduino Uno from the laptop
- Connect external LED between D13 and GND
  - Connect red wire (anode) to D13
  - Connect black wire (cathode) to GND
- Reconnect the Arduino Uno to the laptop
- Observe that the external LED blinks in sync with the on-board LED
- These LEDs are connected in parallel

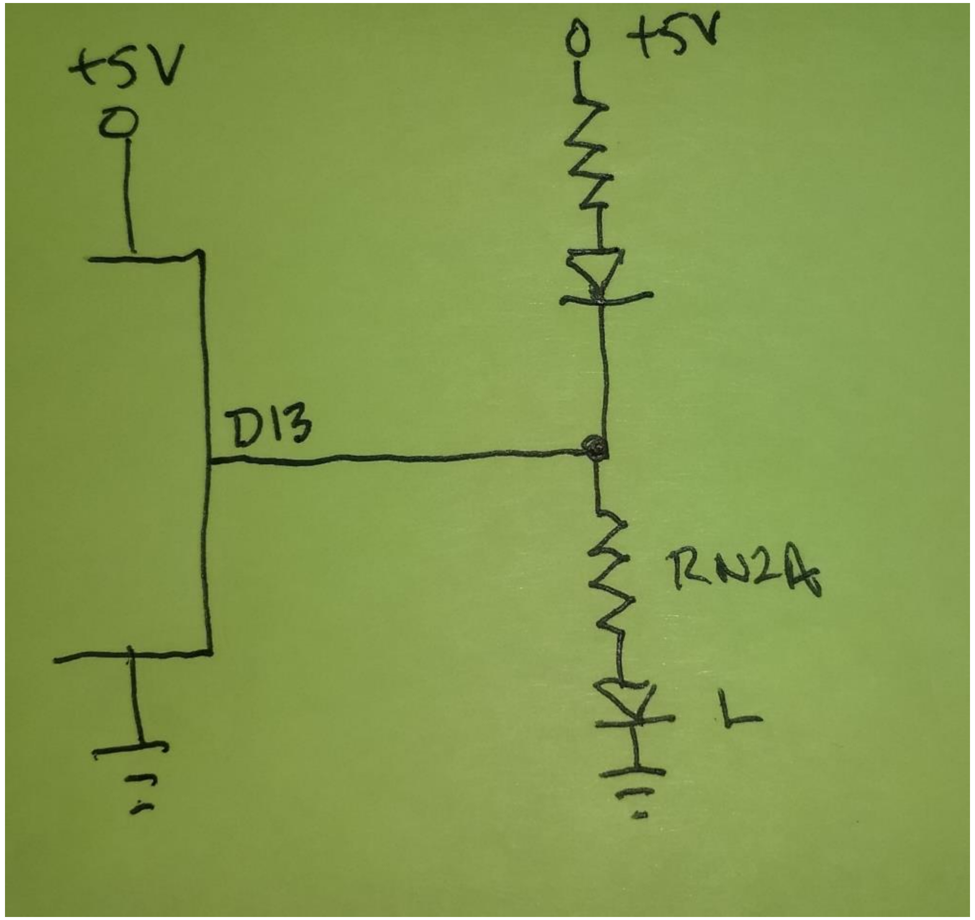
Schematic



Exercise 3: Active low LEDs

- The LED on the Arduino board is connected as “active high”
- This means that the LED is active (on) when the driving signal is high
- Another method is “active low”
- The LED is “reversed” in the circuit and active when the signal is low
- Disconnect the Arduino Uno from the laptop
- Connections:
  - Connect the red wire to 5V
  - Connect the black wire to D13
- Reconnect the Arduino Uno to the laptop
- Observe a different blinking pattern

Schematic



Connections

- Disconnect the Arduino Uno from the laptop
- Connect the 2nd external LED (if available) between D2 and GND
    - Connect the red wire to D2
    - Connect the black wire to GND
- Reconnect the Arduino Uno to the laptop
- Compile and upload the modified sketch
    - Select the menu item “Sketch -> Verify/Compile”
    - Select the menu item “Sketch -> Upload”
- Verify correct blinking pattern

Exercise 4: Blink using a different pin

- Open and run the “blink_2” example sketch:
  - Select menu item “File -> Examples -> LEDstick -> blink_2”
  - Select menu item “Sketch -> Verify/Compile”
  - Look for “Done compiling.” status message
  - Select menu item “Sketch -> Upload”
  - Look for “Done uploading.” status message
  - Does it blink?

Review

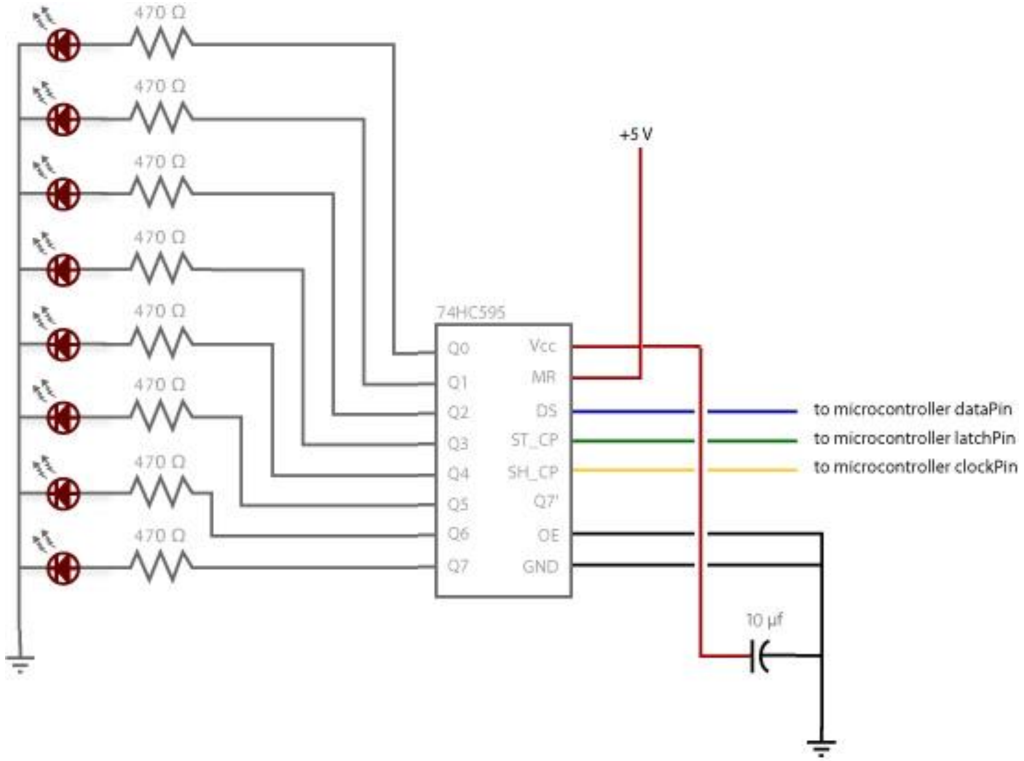
- Blinking single LEDs with Arduino turns out to be simple
- The method we used in the previous exercise is called “direct drive”
- One output pin is used to drive one or more LEDs
- How many LEDs can you blink independently with an Arduino Uno?
  - Using digital pins D2-D13: 12 LEDs
  - Using analog pins A0-A5: 6 LEDs
  - Total: 18 LEDs
- The Arduino Mega 2560 has more I/O pins and can blink more LEDs
  - The device is limited to a maximum of 200 mA total, or about 10 LEDs

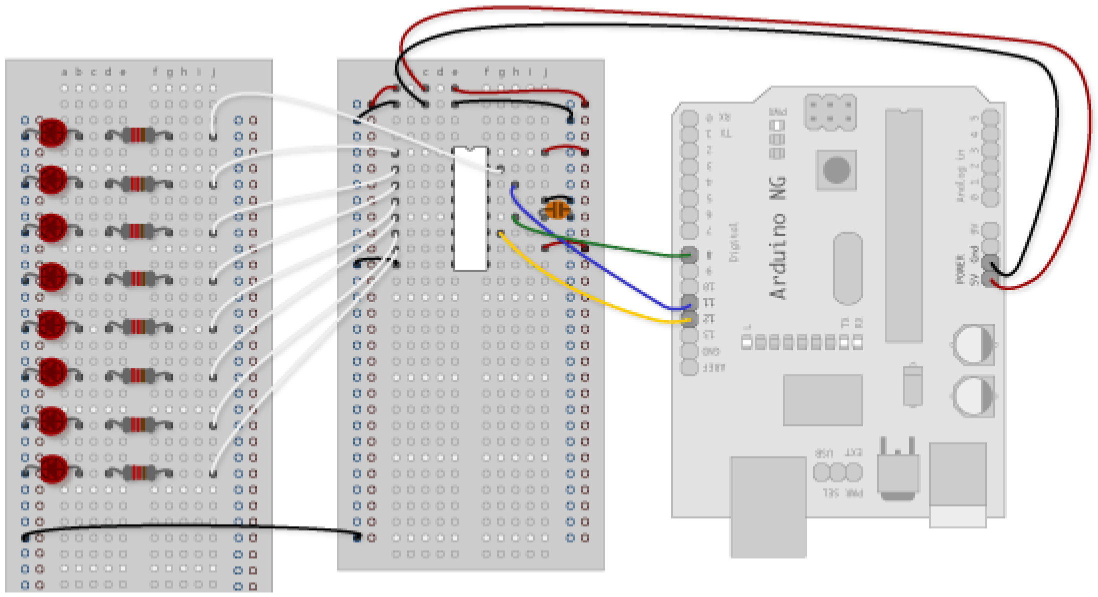
Exercise 6: Multiplexing techniques

- I/O expanders & shift registers
- Digit/segment LED displays
- Row/column LED displays

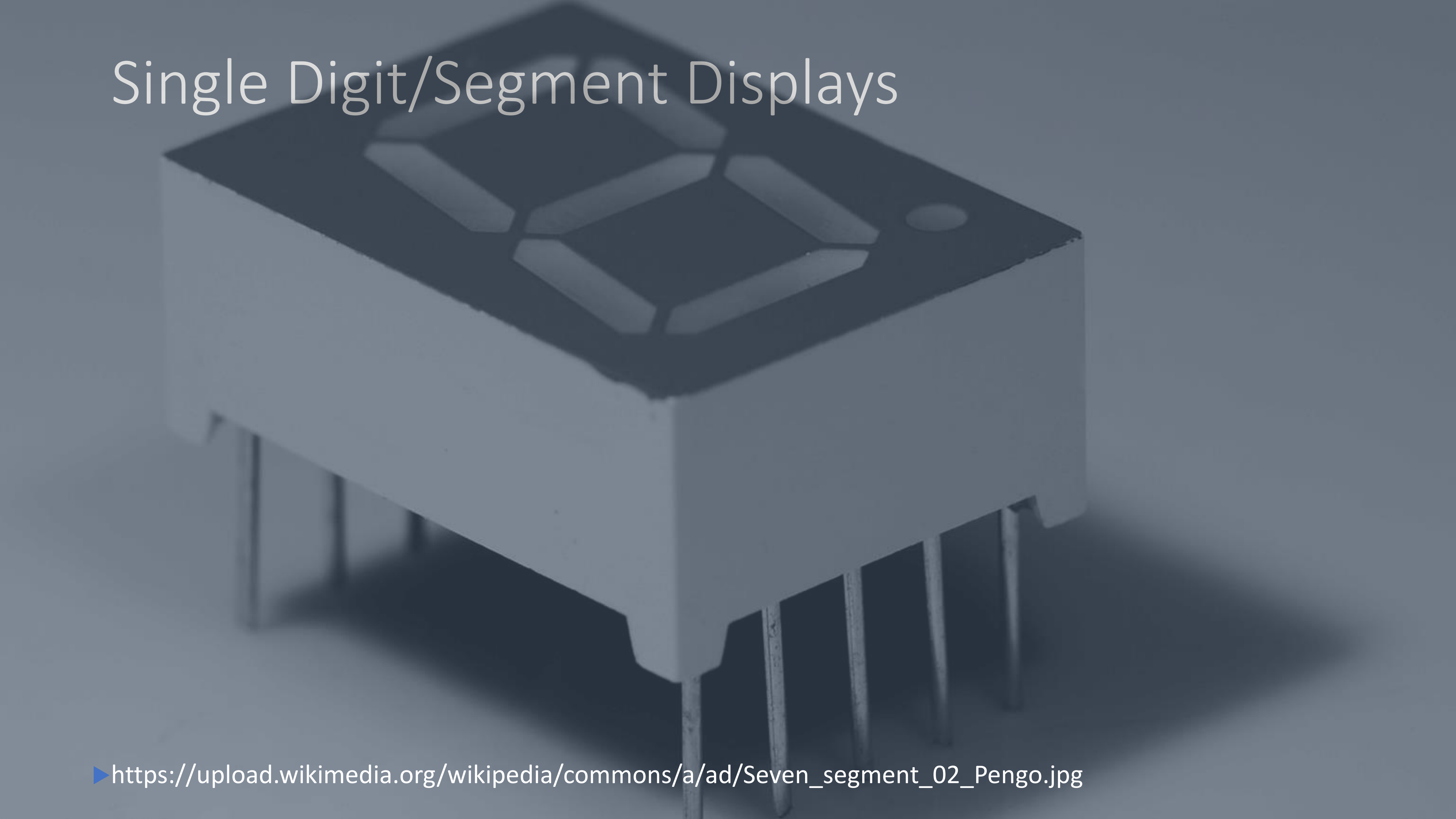
I/O Expanders & Shift Registers

- Using just a small number of control lines, more outputs can be used
- From https://www.arduino.cc/en/Tutorial/ShiftOut





Single Digit/Segment Displays

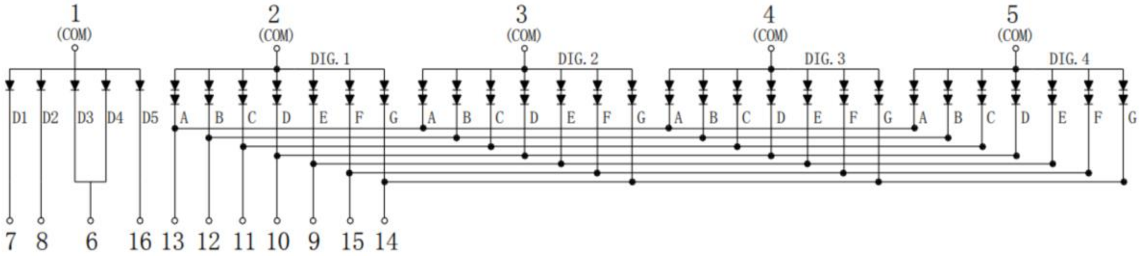


- https://upload.wikimedia.org/wikipedia/commons/a/ad/Seven_segment_02_Pengo.jpg

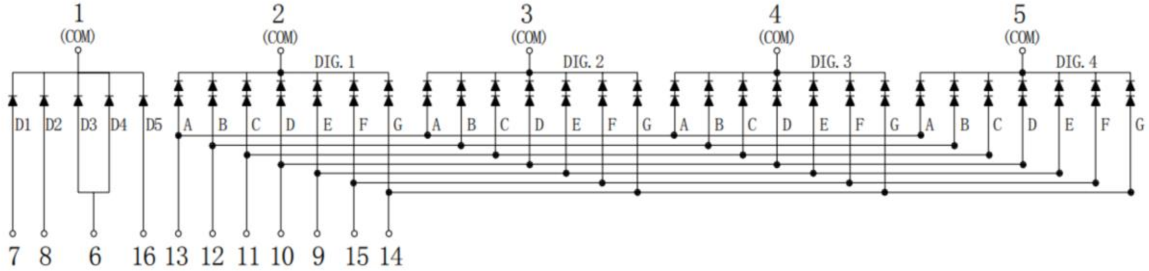
Multi-Digit/Segment LED Displays

https://www.adafruit.com/product/1264

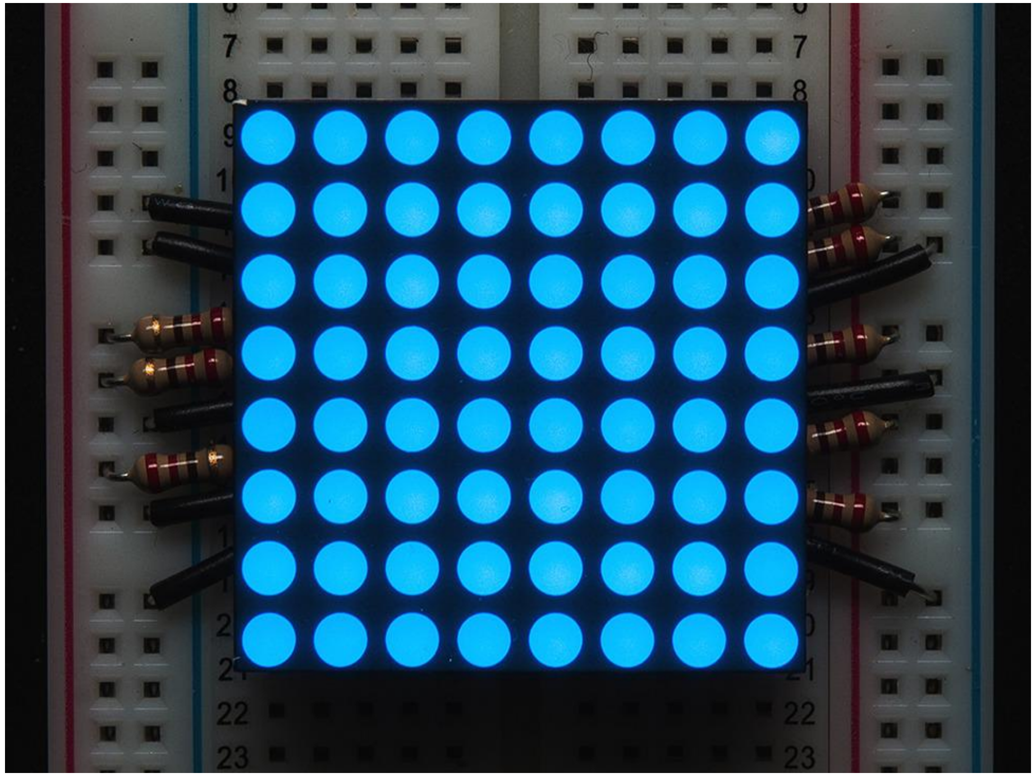
KW4-12041ALA

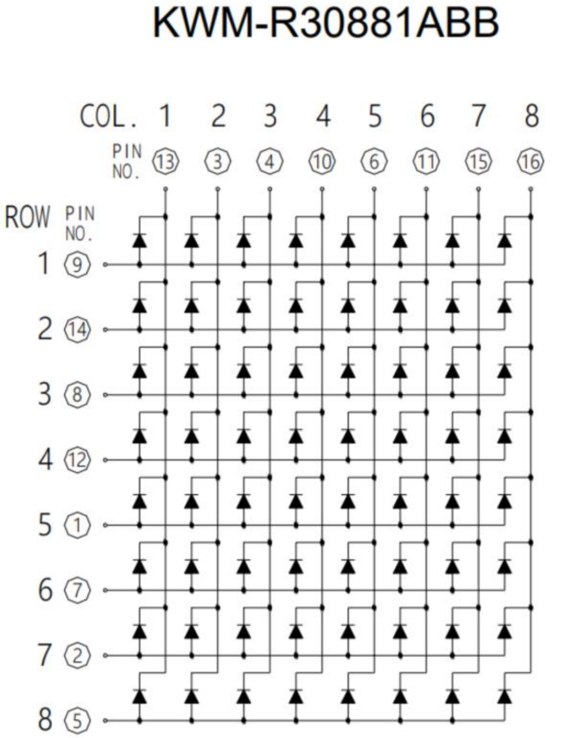


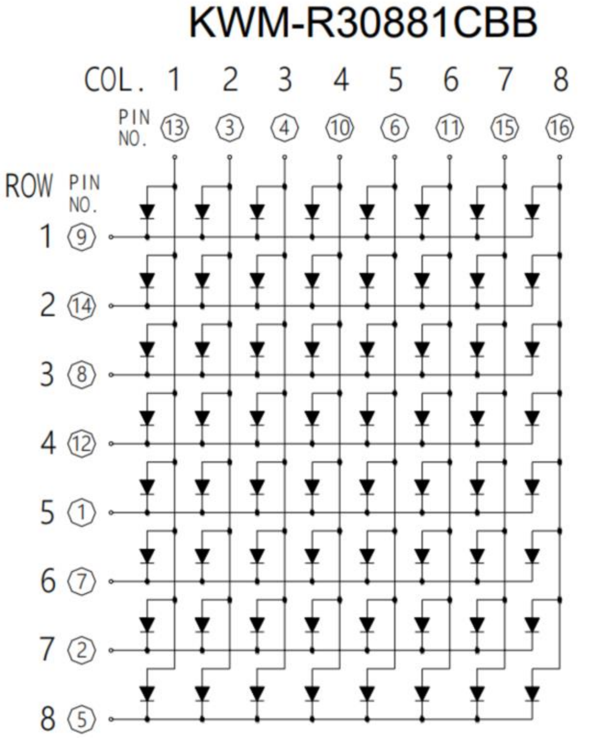
KW4-12041CLA



Row/Column LED Displays







https://www.adafruit.com/product/1047

Exercise 7: Cheating

- Larson scanner
  - Connect tinyCylon to 5V and GND – no sketch required
- NeoPixels
  - Connect 5V, GND & D6 – run “strandtest” sketch
- MAX7219 multiplexer chip with 8x8 LED array
  - Connect 5V, GND, D4 -> DIN, D5 -> CS, D6 -> CLK – run “_8x8_node” sketch

Exercise 8: Charlieplexing LEDs

- Background
- Schematics
- Exercises with the 12LEDstick
- Advantages & disadvantages
- Examples
    - Smart Battery Meter
    - 12LEDcircle
    - Geophone 12
    - 12LEDstick
    - LolShield

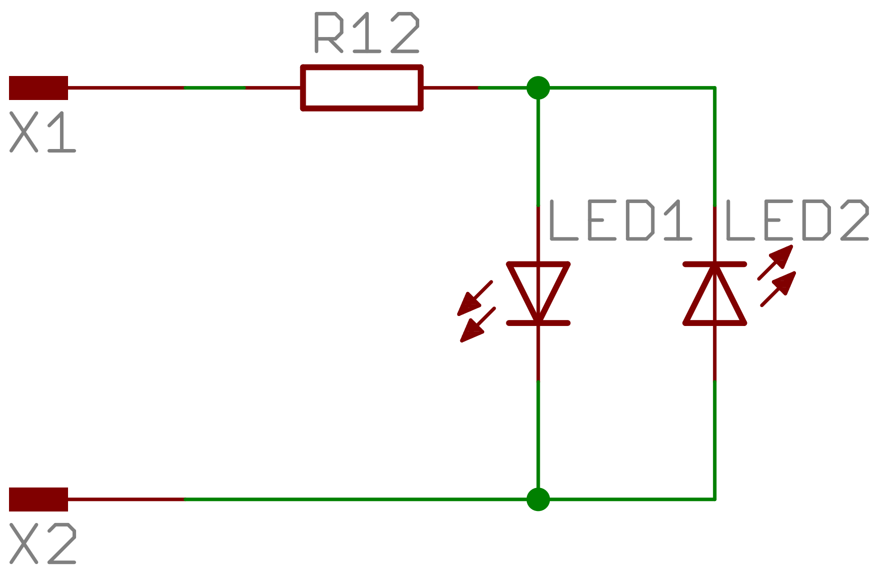
Background

- “Charlieplexing is a technique for driving a multiplexed display in which relatively few I/O pins on a microcontroller are used e.g. to drive an array of LEDs.”
- “Charlieplexing was proposed in early 1995 by Charlie Allen at Maxim Integrated”

  --https://en.wikipedia.org/wiki/Charlieplexing
- Charlieplexing can address n × (n-1) LEDs, where n = number of pins.
- Each pin must be able to output high & low levels
- Each pin must also be able to be tri-stated, or turned into an input.

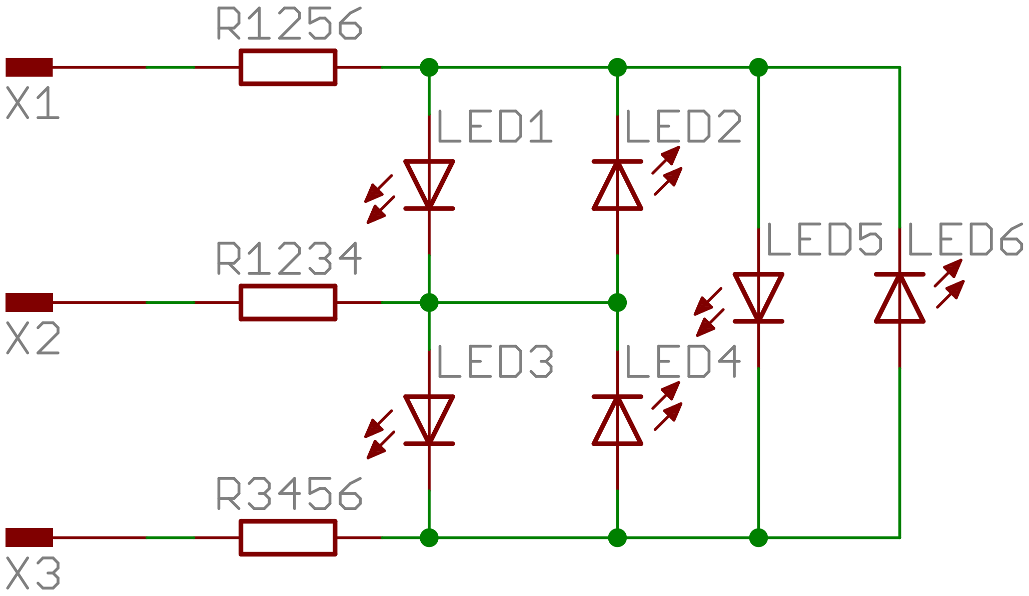
Schematics

Simplest configuration (two nodes)



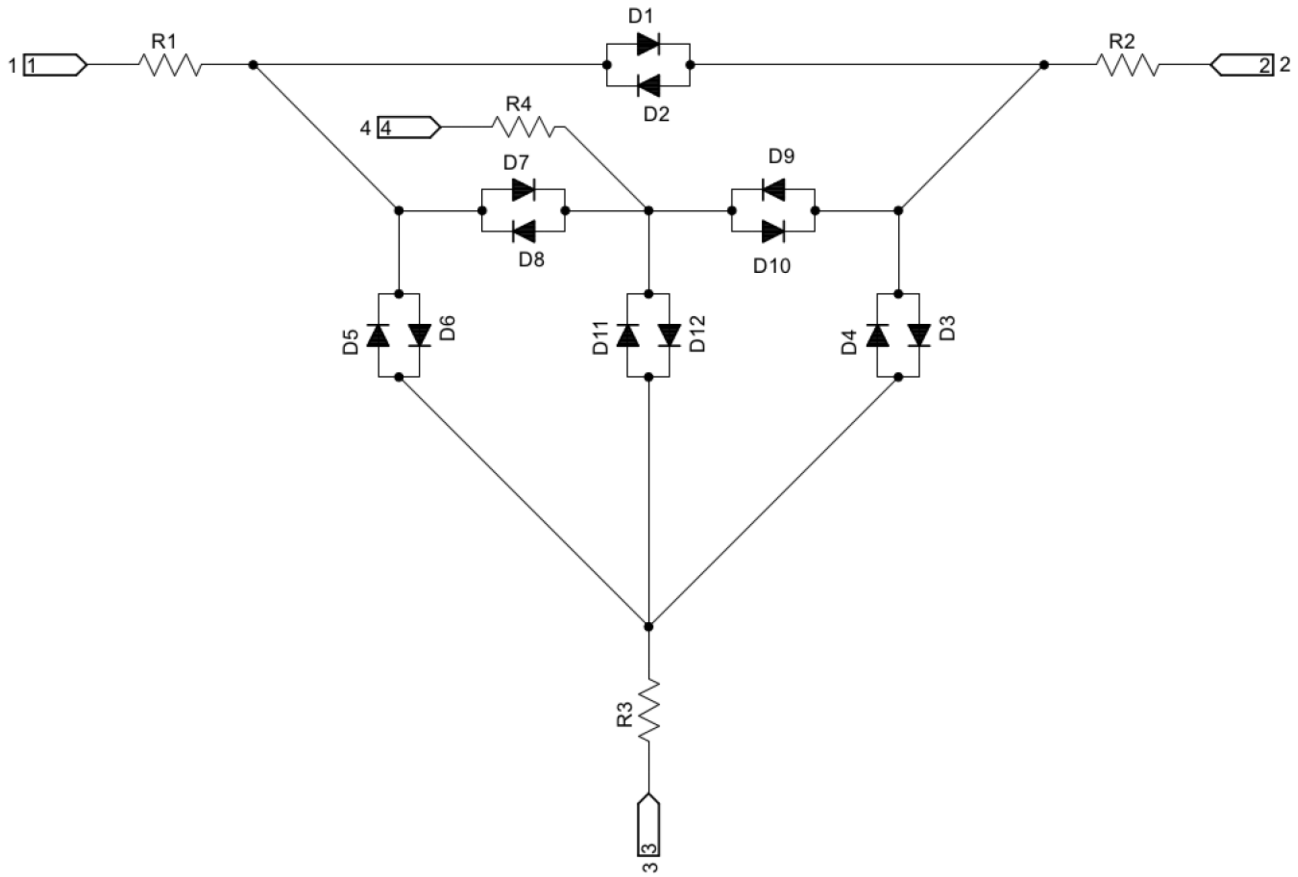
Schematics, continued

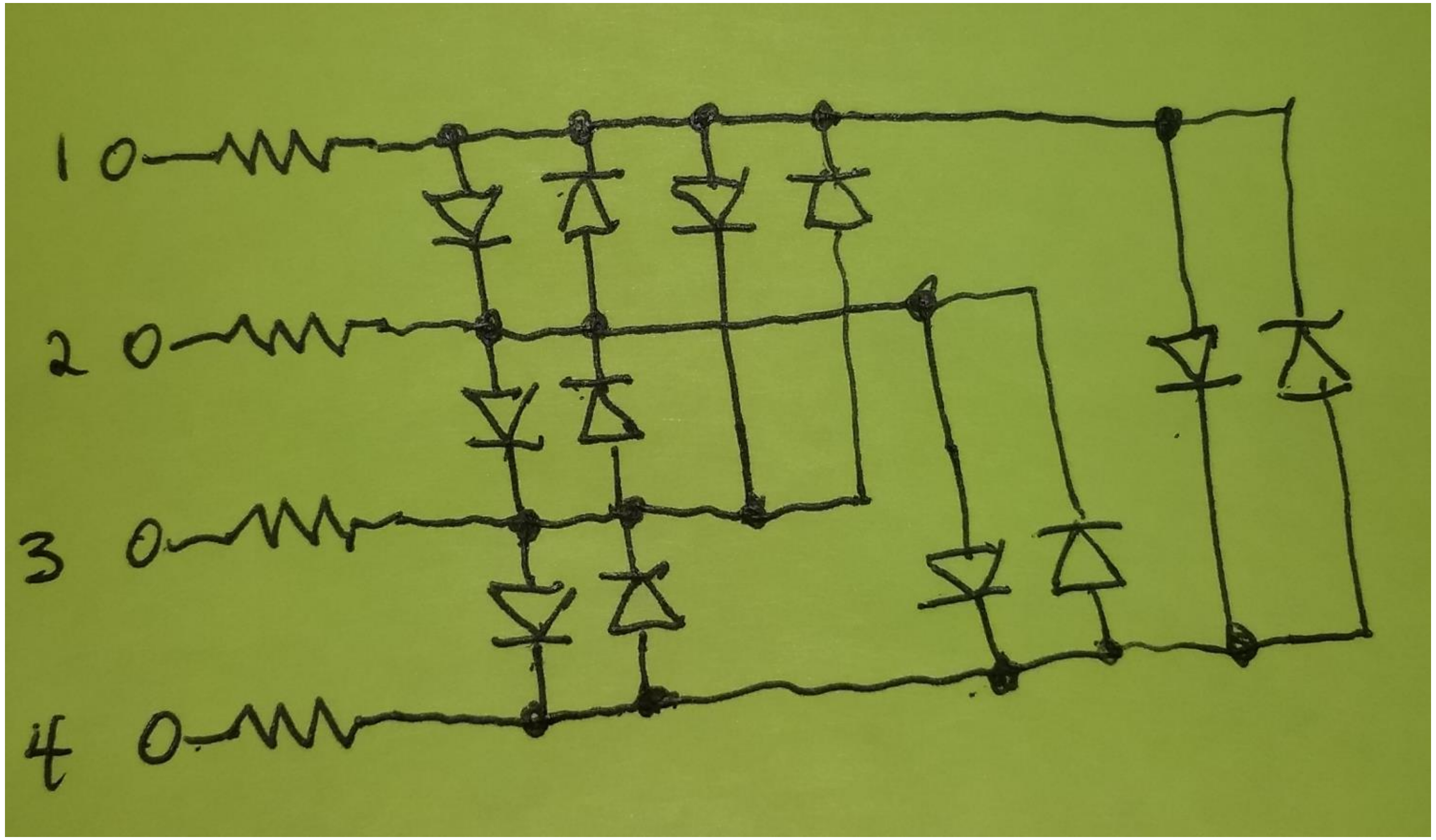
Three pin configuration



Schematics, correct but not helpful

Four wire configuration (12LEDstick)



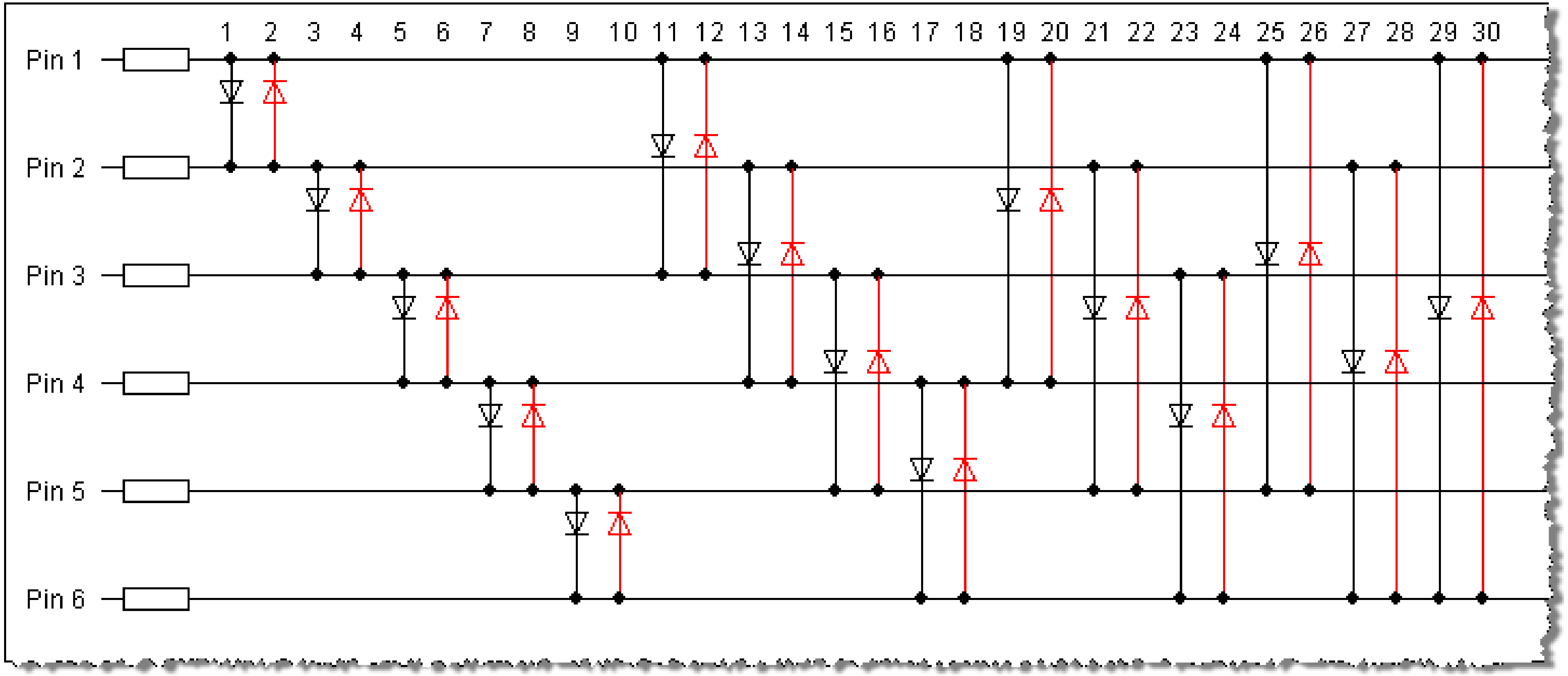


Schematic, hopefully a bit clearer

Four wire configuration

Schematic, six pin configuration

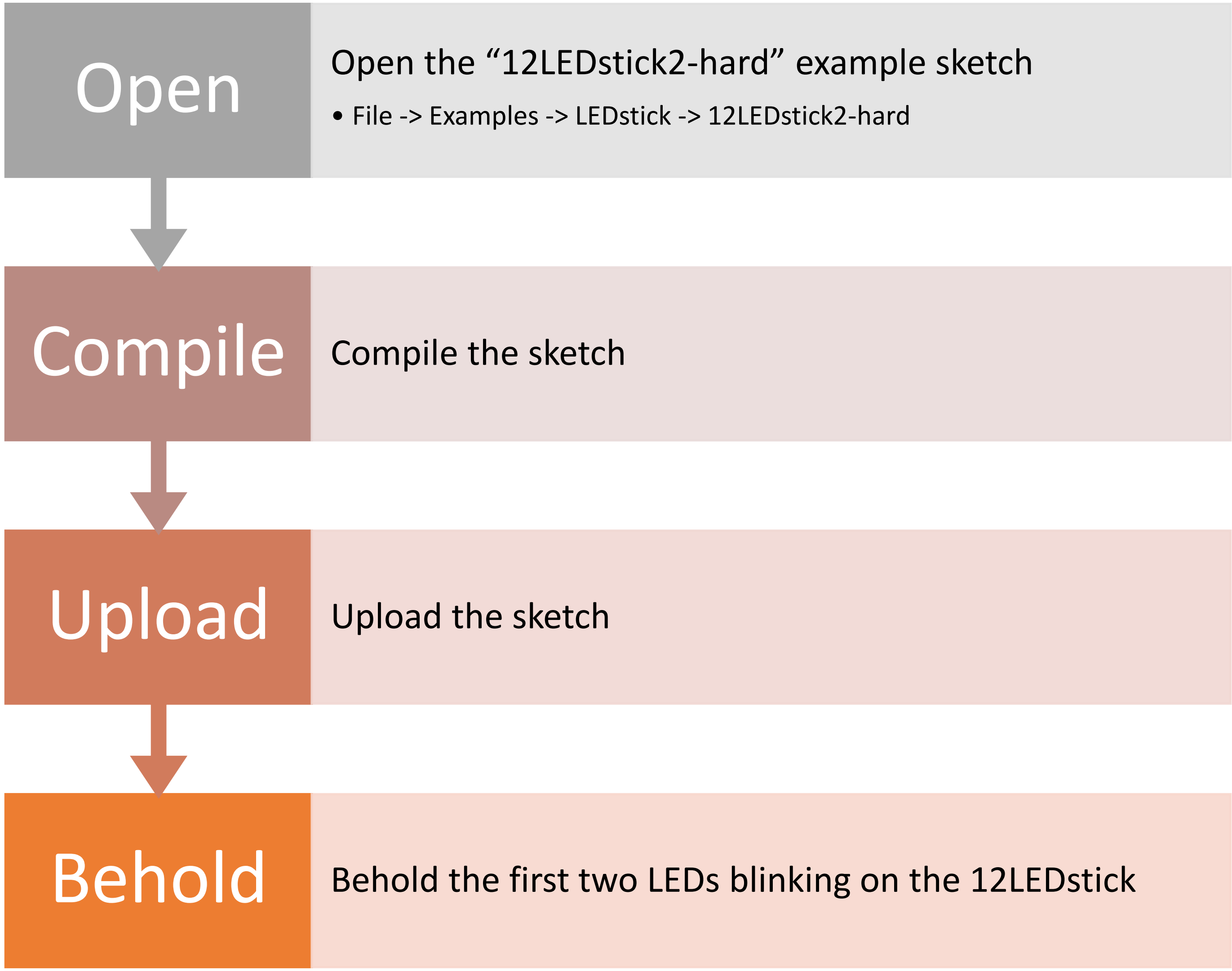
- From https://forum.arduino.cc/index.php?topic=53329.0



Install the 12LEDstick

- Unplug the Arduino Uno from the laptop
- Install the 12LEDstick in pins D2, D3, D4 & D5
- Reconnect the Arduino Uno to the laptop

Charlieplexing the hard way: 2 LEDs



Open

Open the “12LEDstick2-hard” example sketch

- File -> Examples -> LEDstick -> 12LEDstick2-hard

Compile

Compile the sketch

Upload

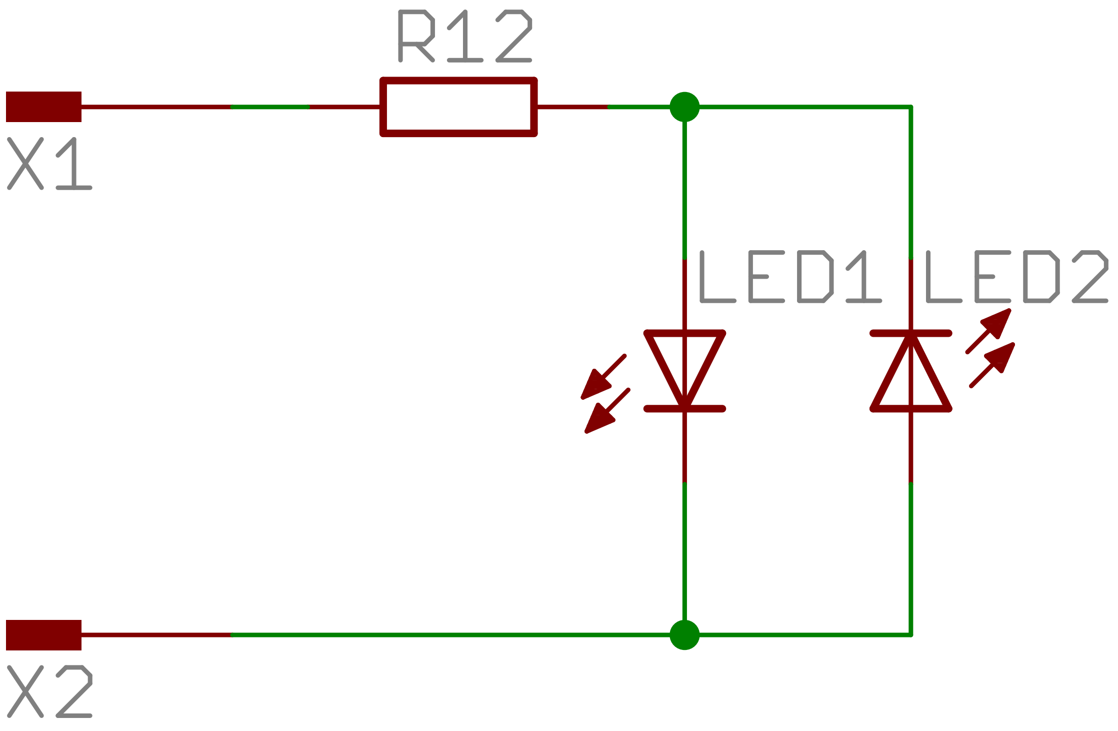
Upload the sketch

Behold

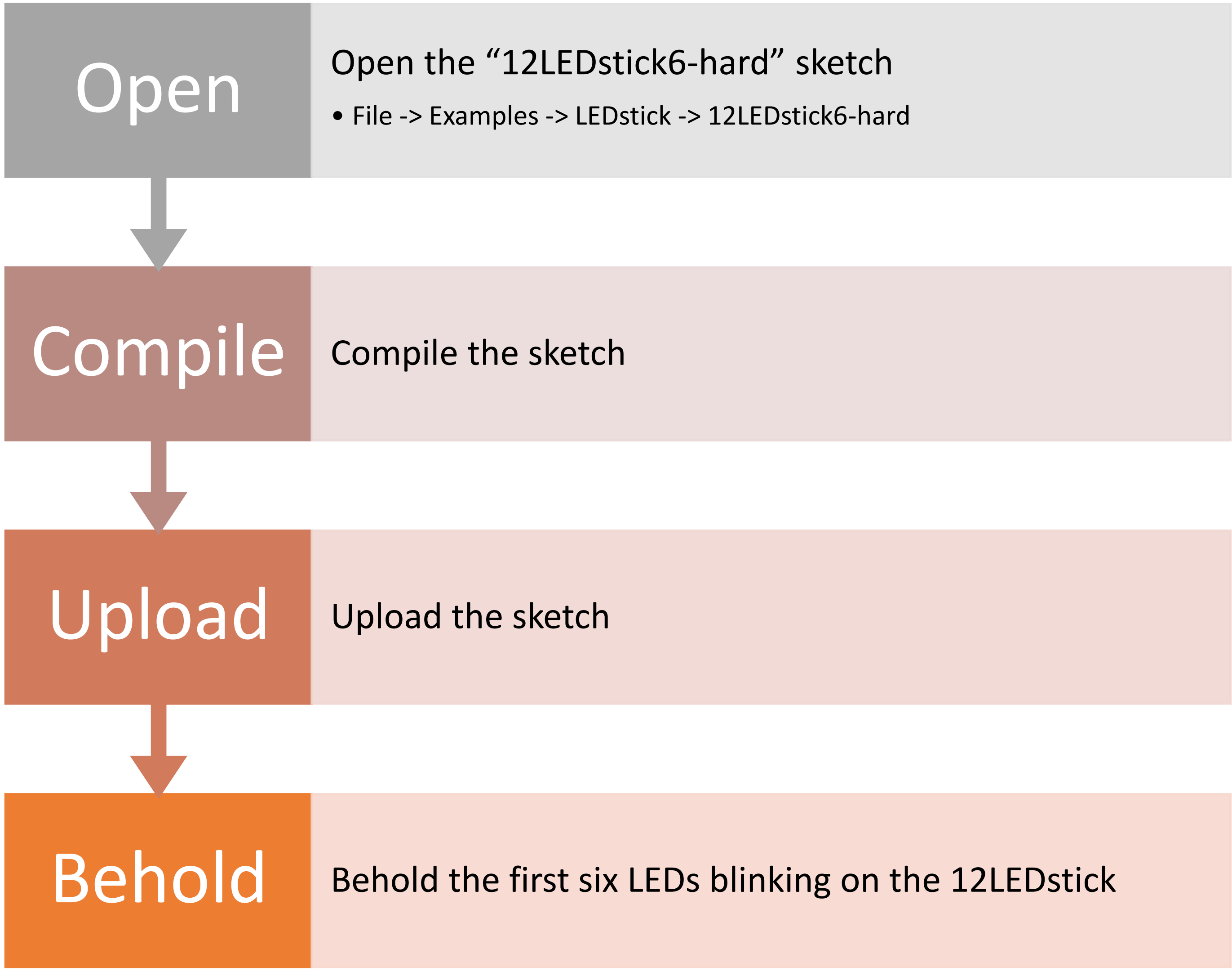
Behold the first two LEDs blinking on the 12LEDstick

12LEDstick2-hard Sketch

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  delay(250); // short delay
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  delay(250); // short delay
}
```



Charlieplexing the hard way: 6 LEDs



Open

Open the “12LEDstick6-hard” sketch

- File -> Examples -> LEDstick -> 12LEDstick6-hard

Compile

Compile the sketch

Upload

Upload the sketch

Behold

Behold the first six LEDs blinking on the 12LEDstick

12LEDstick6-hard Sketch

```
void setup() {
  // none needed
}

void loop() {

  // LED 1 on (1->2)

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, INPUT);
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  delay(250);

  // LED 2 on (2->1)

  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, INPUT);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  delay(250);

  // LED 3 on (2->3)

  pinMode(2, INPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  delay(250);

  // LED 4 on (3->2)

  pinMode(2, INPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(250);

  // LED 5 on (3->1)

  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(250);

  // LED 6 on (1->3)

  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  delay(250);

  // all LEDs off

  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  delay(250);
}
```

Using the LEDstick Arduino Library

- Load the “12LEDstick.ino” sketch
  - File -> Examples -> LEDstick -> 12LEDstick
- Compile+Upload the sketch to the Arduino Uno (the usual way)
- Behold all twelve LEDs blinking in sequence

12LEDstick Sketch

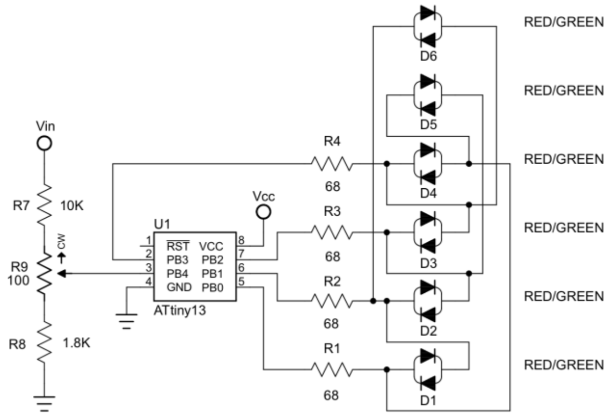
```
#include <LEDstick.h>
LEDstick stick(2, 3, 4, 5); // single instance, defining pins used

void setup() {
  // no special setup needed when using the LEDstick library
}

void loop() {
  // Light up each of the 12 LEDs one at a time
  // Asking for LED 0 means to turn them all off
  for(int i = 0; i < 13; i++) {
    stick.led(i);
    delay(250); // short delay
  }
}
```

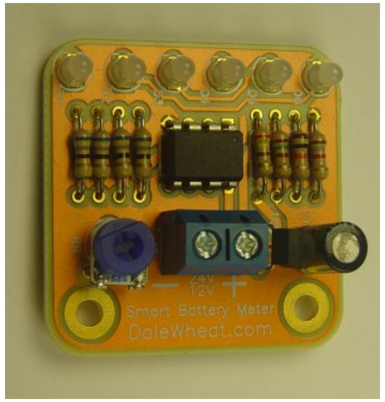
Advantages & Disadvantages

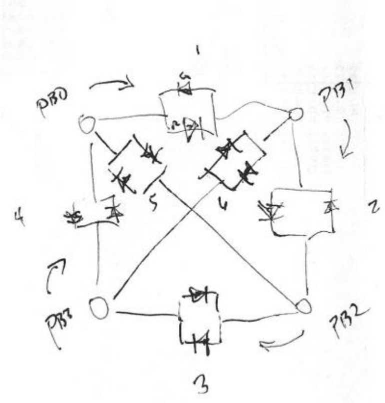
- Advantages:
  - Uses fewer pins
  - Requires fewer current-limiting resistors
- Disadvantages:
  - Not all LEDs can be lit at the same time
  - Requires quick refresh rate to simulate all LEDs on
  - Sometimes hard to isolate defective LEDs
  - Hardware layout can be more complex
  - Software can become more complex

Examples of Charlieplexed LED Displays

- Products developed by Dale Wheat
  - Smart Battery Meter
  - 12LEDcircle
  - Geophone 12
  - 12LEDstick
- LolShield by Jimmie P. Rogers



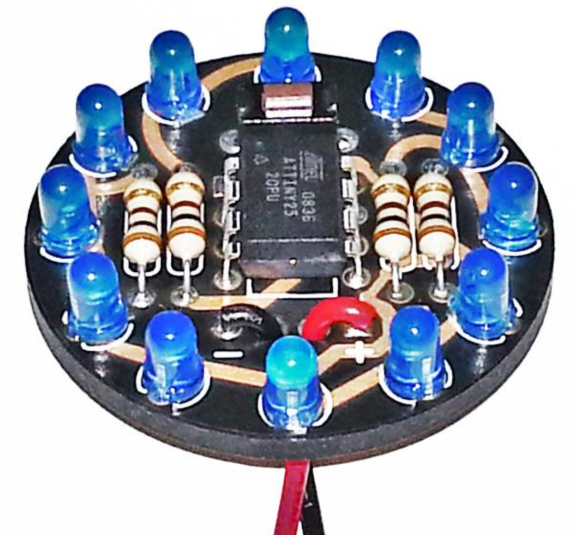


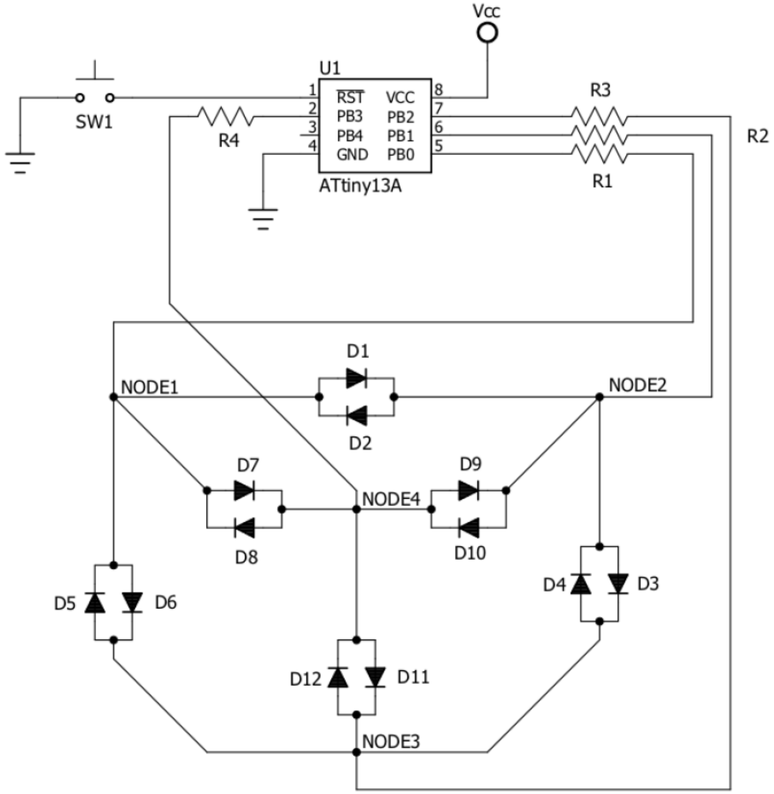


Smart Battery Meter

- Featured in Circuit Cellar Magazine Issue 226 May 2009

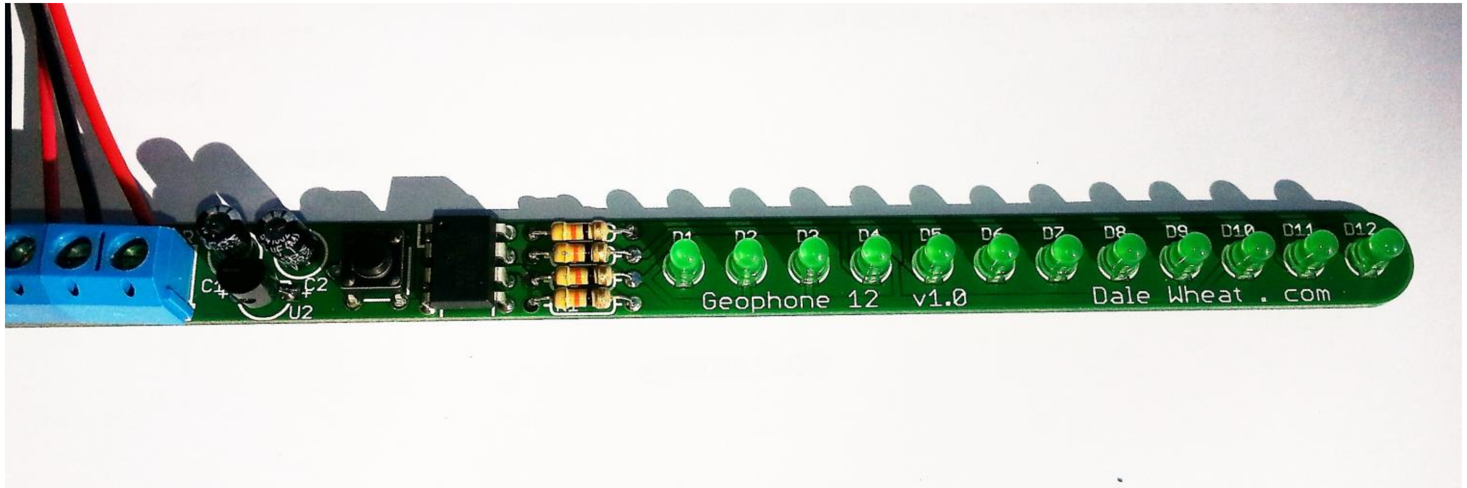
12LEDcircle



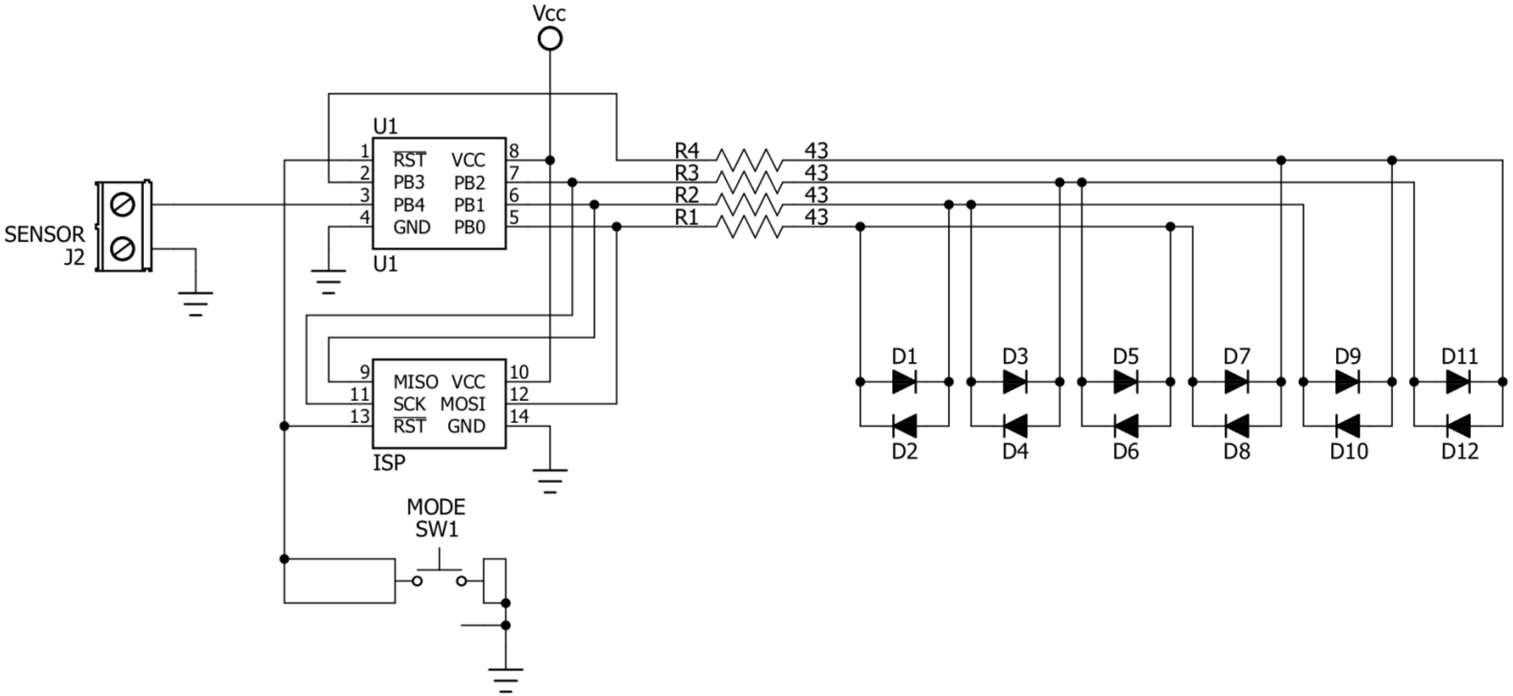


https://youtu.be/8I2yQX_0R4g

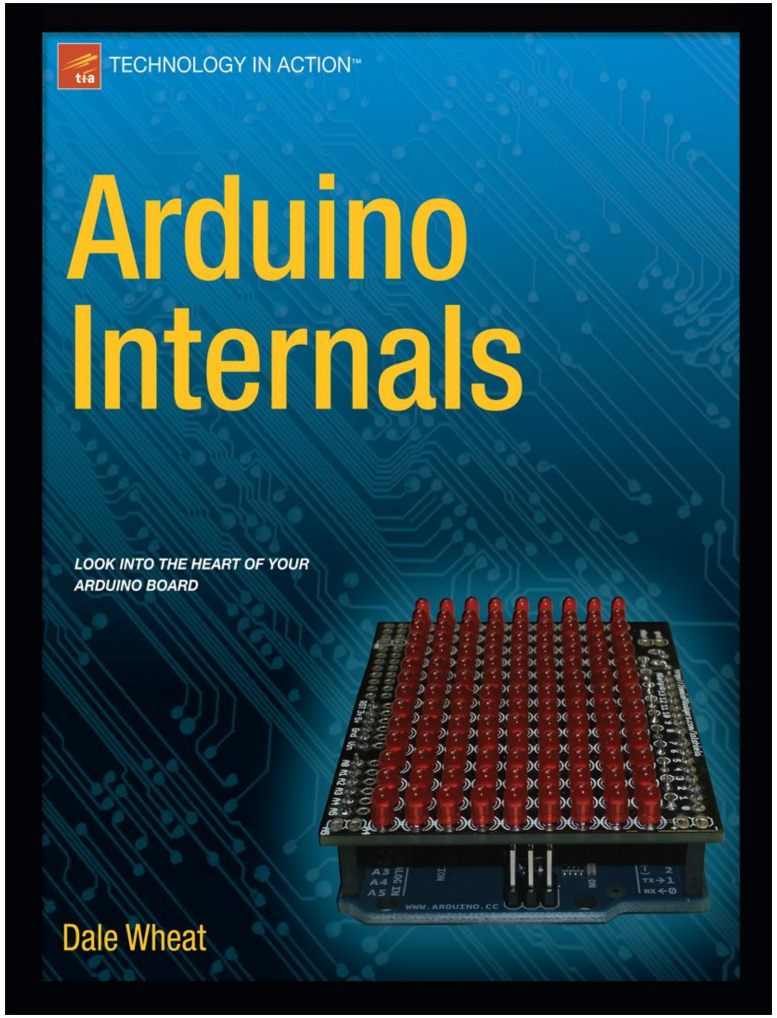
Geophone 12



C1
C2
U2
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
Geophone 12 v1.0
Dale Wheat . com



Vcc
U1
1 RST
2 PB3
3 PB4
4 GND
8 VCC
7 PB2
6 PB1
5 PB0
U1
SENSOR
J2
R4 43
R3 43
R2 43
R1 43
9 MISO
11 SCK
13 RST
10 VCC
12 MOSI
14 GND
ISP
MODE
SW1
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12



LOLShield

- LOL = Lots of LEDs (126 to be exact)
- Designed by Jimmie P Rodgers
- Made the cover of my book

Conclusion

- There are lots of ways to blink LEDs with Arduinos
- Running out of pins does not mean running out of LEDs
- Charlieplexing requires a bit more thought up front

Thank you!

- I really appreciate you attending and participating in this class.
- Please let me know what other classes you would like to take.

Revision History

- Original version v1.0 – September 2019
- Revised to include library support – October 2019