

# Digital I/O

CSE 132

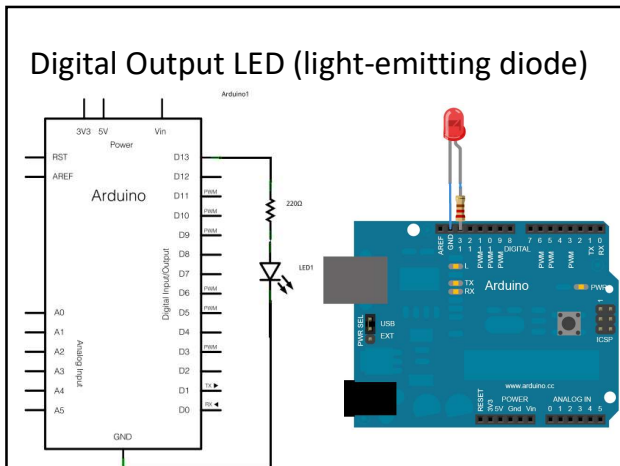
1

## Arduino Input/Output

- 20 pins on physical chip can be configured to do digital input, digital output, analog input, analog output (not all pins can do each function)
- We first configure pins at startup, then use them

```
const int myPin = 13;
void setup() {
  pinMode(myPin, OUTPUT);
}
void loop() {
  //generates square wave
  digitalWrite(myPin, LOW);
  digitalWrite(myPin, HIGH);
}
```

2

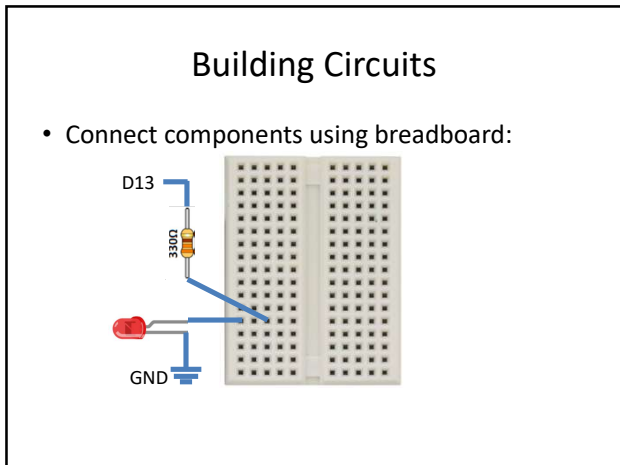


3

## Building Circuits

- 5 horizontal holes are connected:

4



5

## LEDs

- Anode is “+” side, cathode is “-” side
- Anode has longer lead (assuming not clipped)
- Cathode is the flat side on LED body

6

### Resistor Color Codes

- 1<sup>st</sup> two digits are values
- 3<sup>rd</sup> digit is multiplier
- 4<sup>th</sup> digit is tolerance
- 200 to 500 Ω gives good light out of LED
- We will use 330 Ω, or **orange-orange-brown**

**33** x 10<sup>1</sup> Ω

7

### What Inputs Can You Think Of?

- Keyboard
- Mouse
- Webcam
- Temperature
- Microphone
- USB
- Internet
- External memory
- Power
- Emergency stop button
- Proximity detector
- Motion detector

8

### Digital Inputs from Physical World

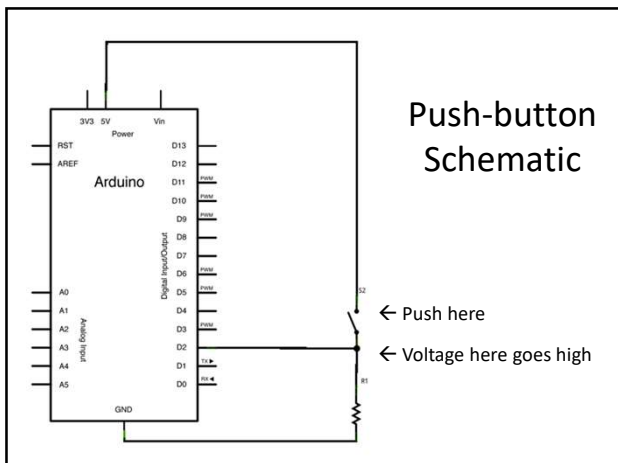
- Example use case
  - Proximity detector
  - Garage door safety beam
- Sensing technologies
  - Capacitive
  - Inductive
  - Optical
  - Radar
  - Hall-effect (magnetic)

9

### Simplest Digital Input

- Push-button switch
  - Pressed = 1, HIGH, TRUE
  - Not-pressed = 0, LOW, FALSE
- Electro-mechanical device
  - When button is pressed, electrical contacts conduct
  - When make/break contact, the contacts can bounce
  - This bouncing can happen over milliseconds
  - But software operates at microsecond scales
  - Even a simple push-button isn't so simple!

10



11

### Software

- Switch between input pin and +5V
  - Input goes HIGH when switch is pressed
  - Input goes LOW when switch is not pressed

```

setup() {
  pinMode(pin, INPUT);
}

loop() {
  inputVal = digitalRead(pin);
}
    
```

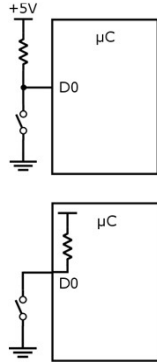
12

### Watch out for signaling convention!

- Switch between input pin and GND  
– Input LOW when switch is pressed
- Why would one do this?
- Because the resistor is available, built into the processor

```

setup() {
  pinMode(pin, INPUT_PULLUP);
}
    
```



13

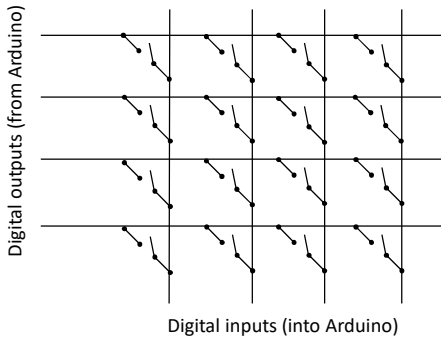
### Switch “Debouncing”

```

Read switch state
Wait enough time for switch to quit bouncing
Read switch state again
if two switch states agree
  Done
else
  Start over
    
```

14

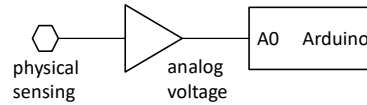
### What about many switches?



15

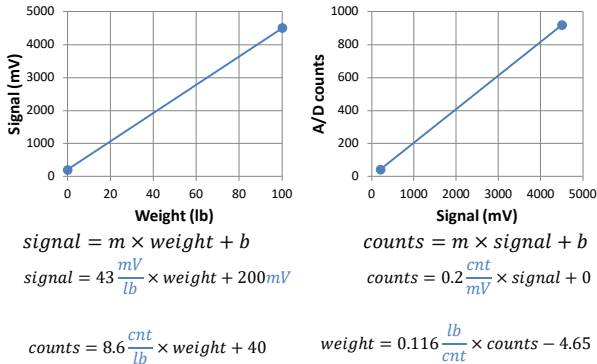
### Analog to Digital Conversion

- Convert physical property to voltage signal
- A/D converter on Arduino converts voltage signal to digital representation  
– 10-bit A/D converter has range 0 to  $2^{10} - 1$  (0 to 1023) for voltage range 0 to  $V_{REF}$



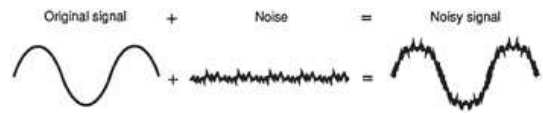
16

### Understanding Ranges



17

### Noisy Analog Signals

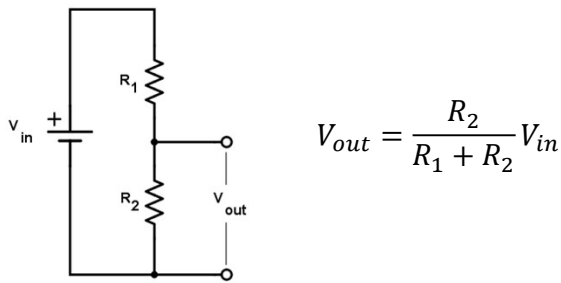


- Noise is ever present in analog signals
- For stable signal, quick fix is to average several readings

$$avg = \frac{1}{N} \sum_{i=1}^N A/D\ input_i$$

18

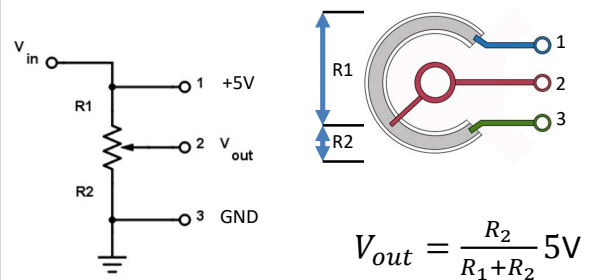
## Simple Voltage Divider



<https://www.allaboutcircuits.com/tools/voltage-divider-calculator>

19

## Simple Voltage Divider



<https://www.allaboutcircuits.com/tools/voltage-divider-calculator>  
<https://ez.analog.com/ez-blogs/b/engineering-mind/posts/designing-a-kevin-varley-potentiometer-part-1-of-3-resistors-in-voltage-divider>

20

## Timing in Java

- Use `Thread.sleep()`
    - Argument is integer number of milliseconds before the method returns
- ```
for (int i=0; i < endTime; i++) {
    Thread.sleep(1000);
    System.out.println(i + " seconds have elapsed");
}
```

21

## Exceptions

- Deviations from the normal flow of control
- “Old style” error checking:
 

```
if (i < 0 || i >= A.length) {
    // handle out of range index
}
else {
    // access array element A[i]
}
```
- Exceptions allow us to be a bit more general

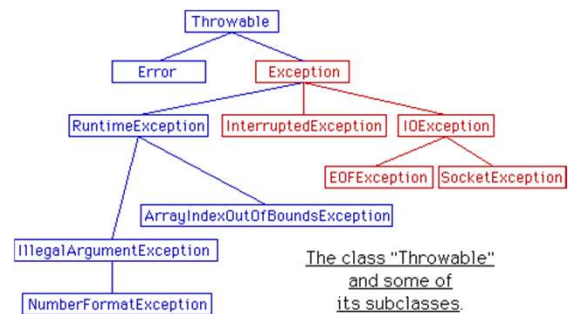
22

## Try/Catch Block

```
try {
    // arbitrary code that might throw an
    // exception when something goes wrong
}
catch (Exception e) {
    // handle the thrown exception
}
```

23

## Unchecked / Checked



24

## Back to Java Timing

- The `Thread.sleep()` method can throw the “InterruptedException,” so we enclose it in a try/catch block

```
for (int i=0; i < endTime; i++) {
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        // default action
        e.printStackTrace();
    }
    System.out.println(i + " seconds have elapsed");
}
```

25

## Logistics

- Assignment 1 is due Jan 27
  - Demos *only* available in first 15 minutes of lab
  - Don't assume lab time available to complete it!!!!
  - Quiz 1B is also due Jan 27 (in the evening is OK)
  - Yes, the logic puzzles are puzzles! Minor points off for not getting them all right
- Module 2 is up now
  - Studio 2 is Monday Jan 27
    - Prep material is posted on webpage
    - Get signed out by a TA in studio
  - Assignment 2 is due Monday Feb 3

46