Taught By: Alexander Rivera

Persistence of Vision Display

Project Info

Project Description: Spinning LED display using arduino to show/animate text

Target Audience: Those with interest in...

- Hardware/Software Design
- High Level Microcontroller programming
- Through hole/Surface Mount Soldering

Class Overview

Classes will be no longer than an hour. Structure will generally be 30 mins of instruction followed by 30 min of open discussion/questions.

Class Structure:

- Class 1: Introduction
- Class 2: Hardware Overview/ Working principle
- Class 3: PCB Design Tutorial 1/2
- Class 4: PCB Design Tutorial 2/2
- Class 5: Soldering Tutorial/ Assembling Hardware
- Class 6: Coding
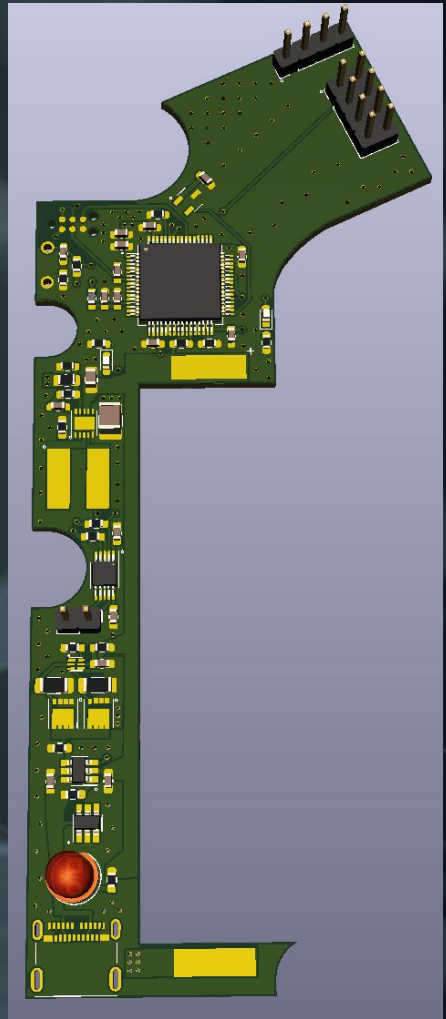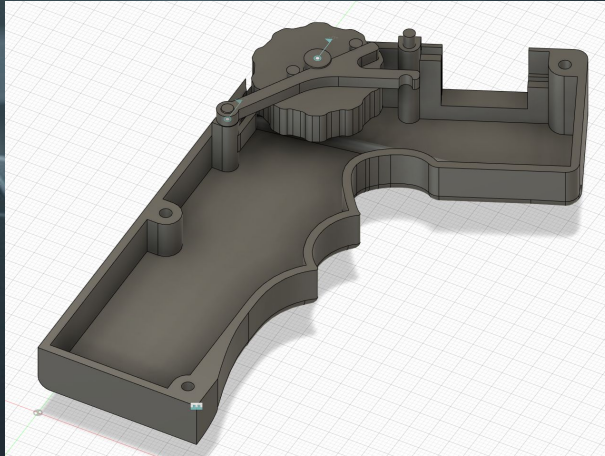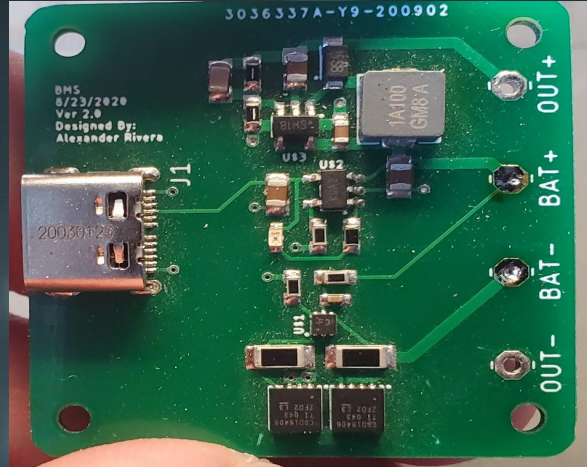- Class 7: Finishing Touches/Debugging

About Me:

- Senior Electrical Engineering
- Originally from Visalia, Ca
- Two Truths/ One Lie:

1) I've driven 60 hrs cross country for a dog

2) I've seen the other side

3) I've never seen a Star Wars Movie

Why should you listen to me?

Some stuff I'm working on:

- Protect/Charge/Boost Li-Ion Project
- Ionic Propulsion Plane *Capstone
- Electric Skateboard from Scratch

What is a microcontroller?

A microcontroller in its simplest form is piece of hardware that is embedded in a ton of products to bridge the gap of digital and physical.
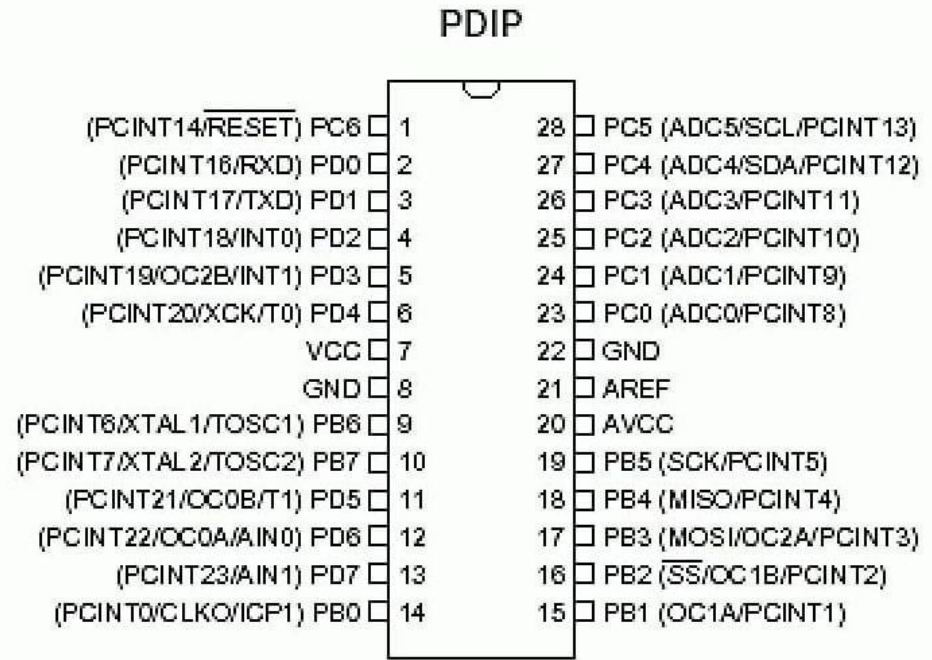
How do you code a microcontroller?

Arduino is a software that takes up memory on a microcontroller typically the ATMEL series microcontroller, and abstracts many of the complexities typically found in coding a microcontroller.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```
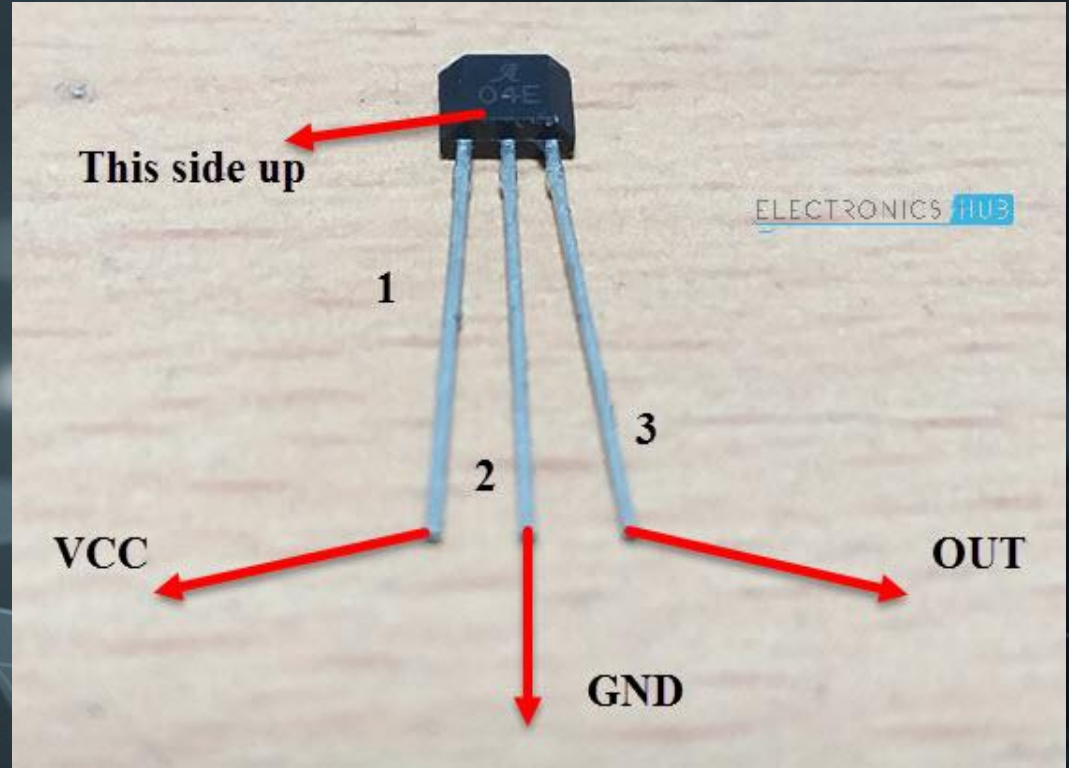
Interrupts, Registers and Timers

- Literally interrupt code
- Registers are a placeholder for data
- Atmega 328p has several timers available in our case we are going to use 2 timers

PDIP

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Hall Effect Sensor

- Senses changes in magnetic field
- This particular model is a switch hall effect sensor
- Let's use this to trigger an interrupt in arduino

Displaying Letters

Background Info/ Stuff to think about

- Hall Effect Sensors and how they could be implemented
- Registers, Timers, Interrupts
- How would you go about designing this project given these tools?

Open Discussion

Questions, Comments, Concerns…