Hardware Project: An Attempt at a A.I Robot

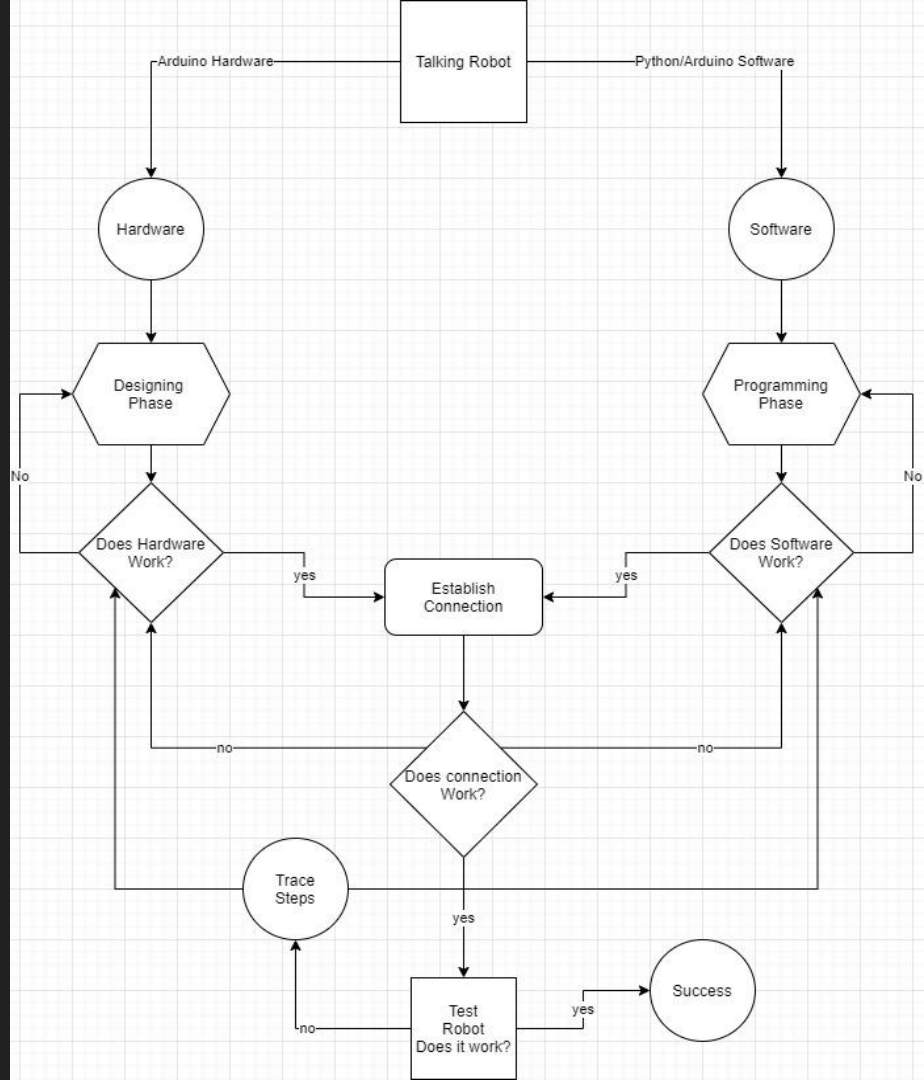
Zakaria Gorgis A16177454
12/6/2020 Fall Quarter 2020
ECE 196

Introduction and Motivation. Why this project?

- Cog Sci w/ spec ML background (minor in CS)
- Huge interest in artificial intelligence
  - Doomsday
- Wanted to focus on software and its manipulation through hardware
- Enjoyment of learning new techniques and libraries through programming
- Gateway to understanding how hardware works
  - Not too overwhelming of a project
- Love for this kind of stuff!

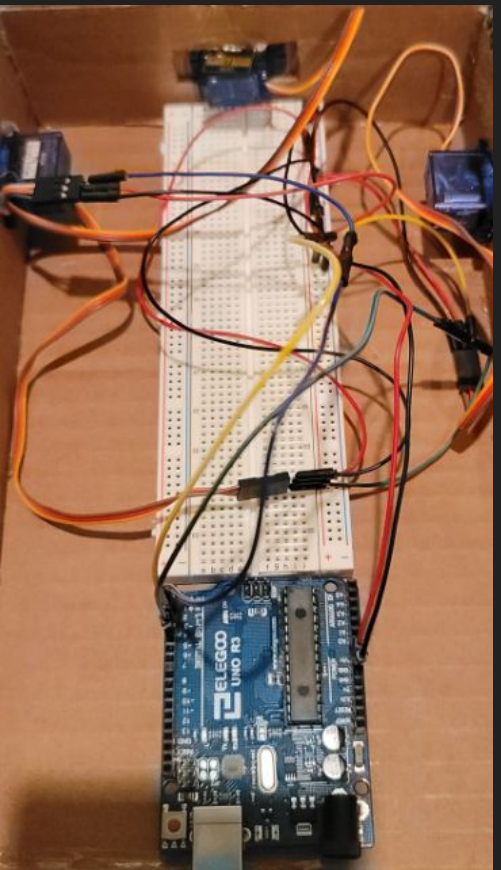
How the problem solving approach was laid out:

- Created a logic flow diagram
- Split up problem into two main groups
  - Software and Hardware
- Worked on each part individually
  - Then established a connection
- Throughout all steps, testing was very much an important process
  - That being said, not all bugs could be ironed out (comes with the territory)

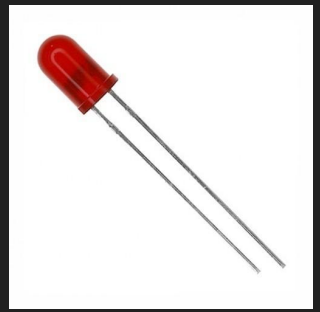


Now to Look at the Solution in depth

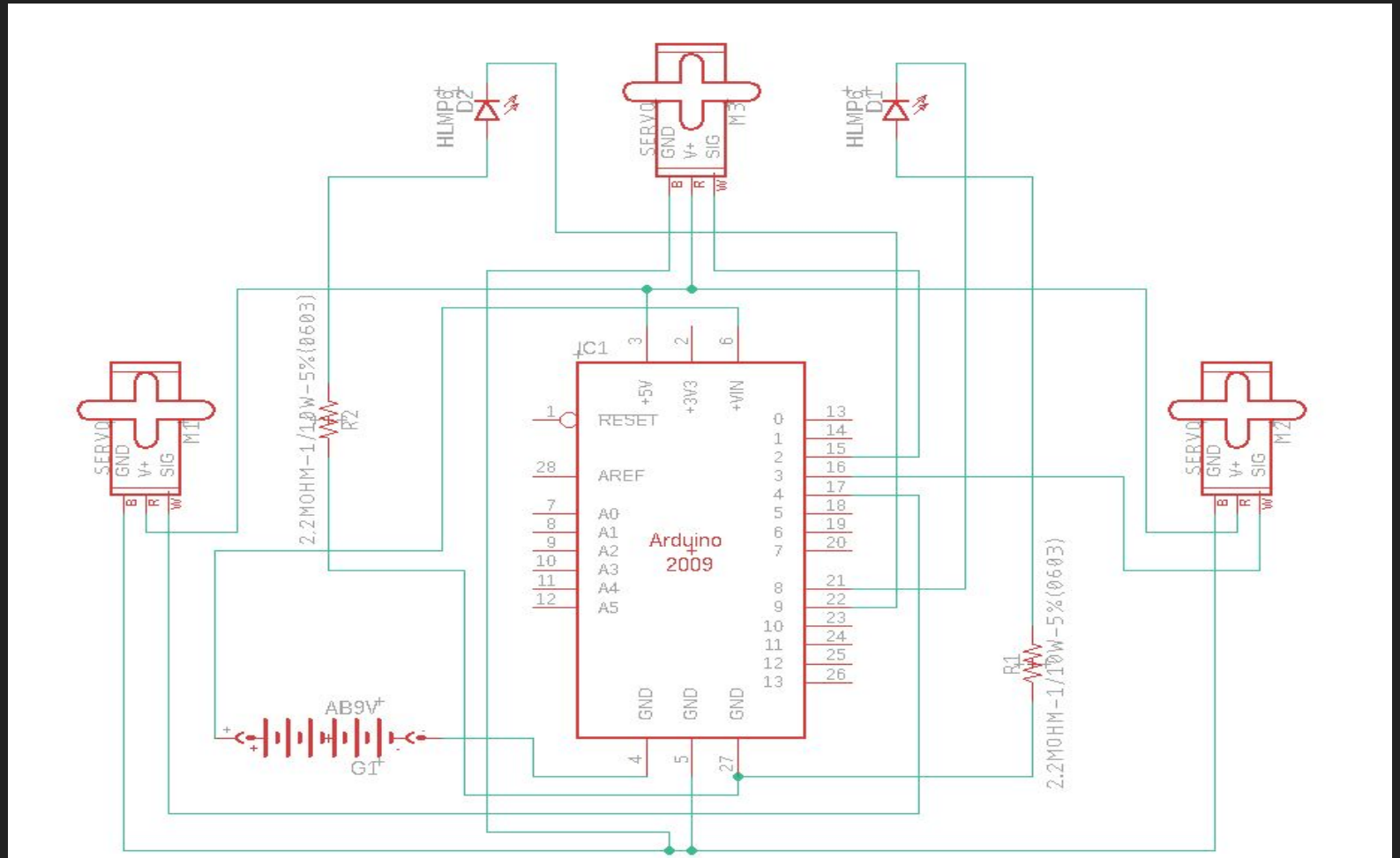
Taking a look at the hardware [High level]



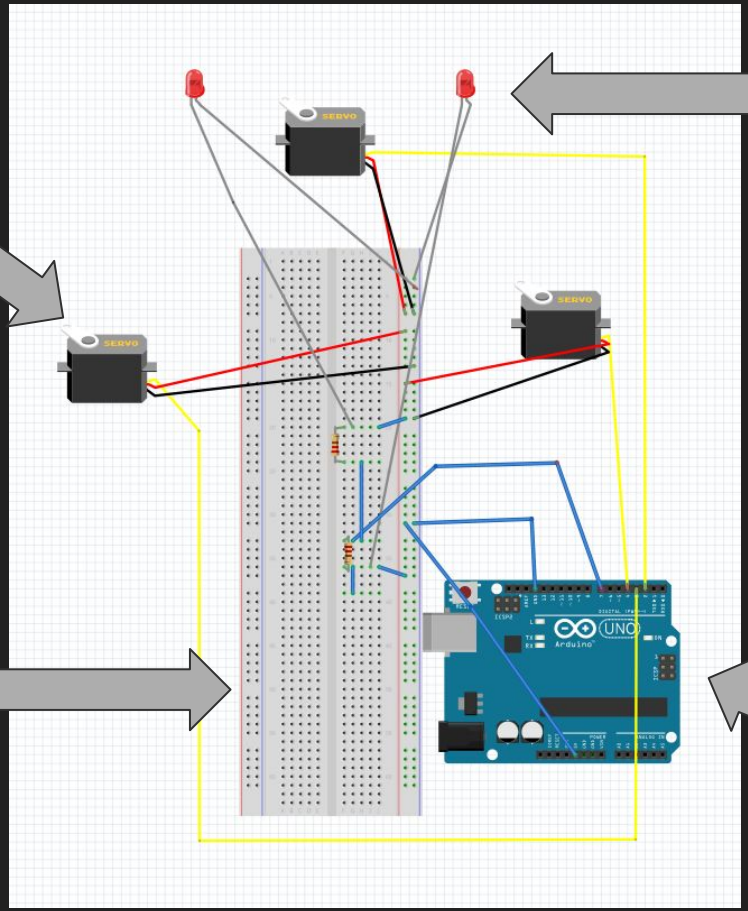
- Like previously stated, the hardware itself was pretty basic:
  - Breadboard, arduino, 3 SERVO motors, 2 LEDs, wiring, and the Robot Housing
    - Robot housing made by yours truly
- Understanding the wiring was tough at first
  - When I say no experience, I literally mean none
- But soon became enjoyable



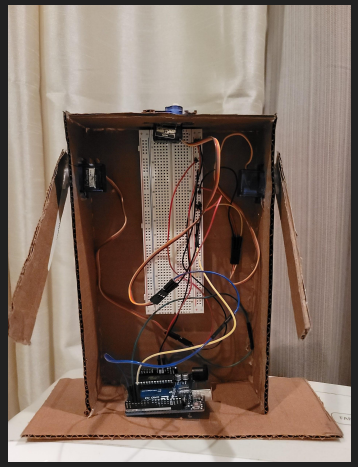
EAGLE Schematic of Hardware



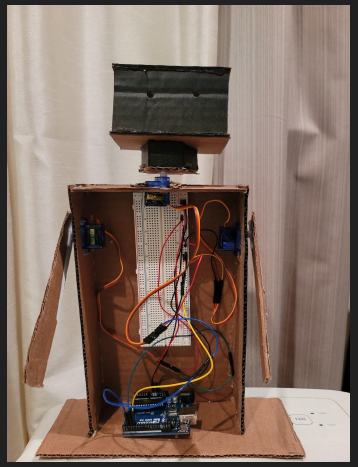
FRITZING Schematic of Hardware



Physical Progression of the Robot Prototype; _____ Cardboard



Evolution 1

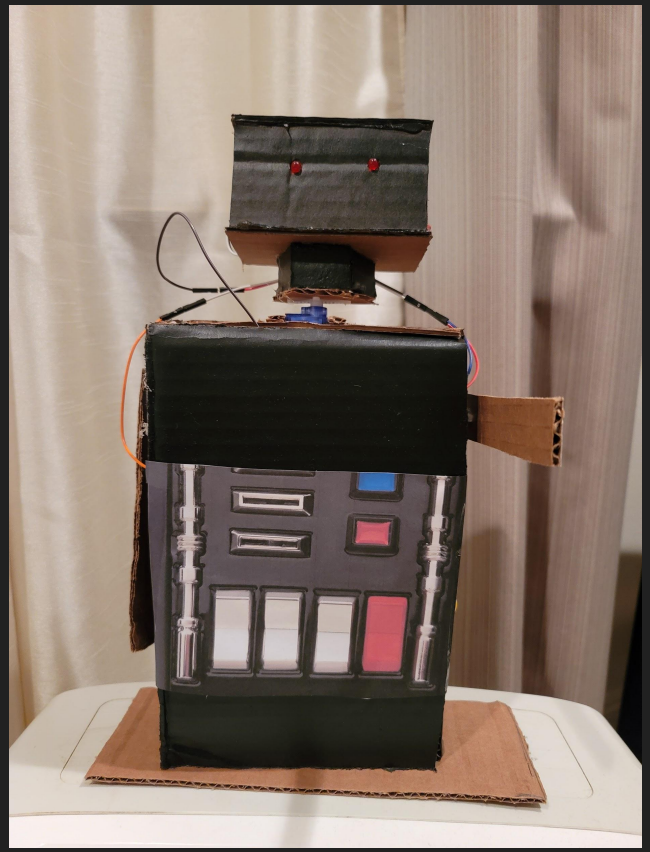


Evolution 2

Evolution 3

Evolution 4

Physical Progression of the Robot Prototype; Darth Cardboard



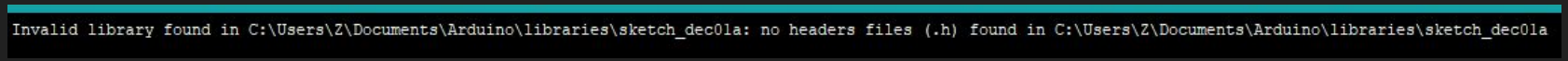
Evolution 5
= ?

Taking a look at the software (python) [High level]

- Started with glorified text editor, VScode
- Needed to install different libraries that I never used before
  - Pip install speechrecognition
    - Useful for reading user speech
  - Pip install pyaudio
    - input/output stream library
  - Pip install pyttsx3
    - Text to speech
  - Pip install pyserial
    - Port communication
- Testing the software was a bit easier than the hardware part

Taking a look at the software (arduino) [High level]

- Only used one library, servo.h
  - However, reoccurring issue kept showing up, I believe because of this, the servo motors would jitter
- Required the use of a lot of loops and change in positional coordinates
  - Very tedious
    - Including timings, such as delays and when to write
- Servo motors initialized at 2 for head, 3, 4 for respective arms
  - Digital ports on the arduino
- Had more trouble with programming on the arduino IDE than on python



Establishing a connection between software and hardware

- Surprisingly, this was not a hard step to establish
  - Arduino is plugged into the COM3 USB PORT
- On arduino IDE, the port is communicated with using
  - Serial.begin(9600)
    - This is used to exchange messages with the serial monitor at 9600 bps
      - Arduino has an established way to communicate with the computer, very easy
- On pythons end, it's a few more steps
  - Import serial, for communication
    - Every time a command is given, the serial library writes a character
      - Example, port.write(b 'q')
        - The arduino IDE would read the q, and run the subsequent function
  - For instance, if I ask Darth to dance, in the python code, I would port.write(b 'd')
    - In the arduino IDE, inside of the void loop() function, it would be called in the form;
- Not much else to it

Python

```
if word_list[0] == 'dance':
    port.write(b'd')
```

Arduino IDE

```
if(portValue == 'd'){
  dance();
  delay(1500);
}
```

Example in depth communication between python and arduino:

Step 1

- For example, I said “Darth, play Professional by The Weeknd”
  - The robot will wake up, as Darth is the wake up word, and then recognize play as a function call, starting the process
  - The words come in as a parsed string, easier to read

```
if word_list[0] == 'play':
    talk("fine, but dont ask again, the emperor is not as forgiving as I am")
    extension = ' '.join(word_list[1:])
    port.write(b'y')
    pywhatkit.playonyt(extension)
    port.write(b'p')
    return
```

Step 2

Step 3

```
if(portvalue == 'y'){
  hands_up();
  delay(3000);
}
if(portvalue == 'p'){
  standby();
  look_left();
  delay(1500);
}
```

And that's really all to it!

For those who might attempt something like this in the future, a few tips, and my opinions on the project

- Some overall thoughts
  - Start early
  - Ask questions
  - Research and prototype
  - Youtube videos are your friend! (No seriously)
- The actual hardware part is the most difficult
  - The wiring and fiddling with the Arduino IDE to be specific
  - Easiest part might of been the python
    - So many useful libraries make what you want come to life with ease
- If taken one step at time, it becomes doable - don't overwhelm
  - And most importantly, as stated by the instructional staff, choose something you will enjoy so the process itself becomes enjoyable!

Links

- Link to Presentation
  - https://www.youtube.com/watch?v=aAxVPqlhXic&feature=youtu.be
- Link to Live demonstration
  - https://www.youtube.com/watch?v=IXLGFT_Ao48&feature=youtu.be