Multiple Retrofit Blinds Controllers

Design Document

Team 11

Client/Advisor

Leeland Harker

Team Members

Daniel Andrews

Andrew Deick

Jacob Nett

Caleb Townsend

Hieu Nguyen

Logan Shada

Sddec22-11@iastate.edu

https://sddec22-11.sd.ece.iastate.edu

Executive Summary

Development Standards & Practices Used

- "IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems," in IEEE Std 2050-2018 , vol., no., pp.1-333, 24 Aug. 2018, doi: 10.1109/IEEESTD.2018.8445674.
- "Preprint Proposals for the 2022 Edition of the National Electrical Safety Code (NESC(R))," in Preprint Proposals for the 2022 Edition of the National Electrical Safety Code (NESC(R)) , vol., no., pp.1-867, 1 July 2019, doi: 10.1109/IEEESTD.2019.8753791.
- "IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks--Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Enhanced Throughput for Operation in License-exempt Bands above 45 GHz," in *IEEE Std 802.11ay-2021 (Amendment to IEEE Std 802.11-2020 as amendment by IEEE Std 802.11ax-2021)* , vol., no., pp.1-768, 28 July 2021, doi: 10.1109/IEEESTD.2021.9502046.

Summary of Requirements

- The device shall be able documented in a way that prioritize device reproducibility
- When the device experience power failure it state shall be maintained
- The device shall be retrofit onto existing blinds forming a new system that does not impede on the old system

Applicable Courses from Iowa State University Curriculum

- CS309 - Software Development Practices
- CS319 - Construction of User Interfaces
- CS363 - Introduction to Database Management Systems
- CPRE288 - Embedded Systems
- EE230 - Electronic Circuits and Systems

New Skills/Knowledge acquired that was not taught in courses

- Part ordering
- Connectivity of premade parts
- Geolocation service
- Server maintenance and modification
- Using specialized software tools like Next-Auth

Table of Contents

1. Project Description **4**

1.1 Problem Statement 4

1.2 Solution Description **4**

2. Design

2.1 DESIGN CONTEXT

2.1.1 Broader Context

2.1.2 User Needs

2.1.3 Prior Work/Solutions

2.1.4 Technical Complexity

2.2 DESIGN EXPLORATION

2.2.1 Design Decisions

2.2.2 Ideation

2.2.3 Decision-Making and Trade-Off

2.3.1 Design Visual and Description

2.3.2 Functionality

2.3.3 Areas of Concern and Development

2.4 Technology Considerations

2.5 Design Analysis

2.6 Design Plan

2.7 Changes From First Semester **4**

3. Functional Requirements **4**

3.1 Hardware Requirements. 4

3.2 Software Requirements 5

4. Nonfunctional Requirements **5**

4.1 Hardware 5

4.2 Software 5

5. Relevant Standards **5**

6. Constraints **6**

7. Security Concerns & Countermeasures **6**

7.1 Physical Security 6

7.2 Cybersecurity 7

8. Implementation Details 7

8.1 Hardware Implementation 7
8.2 Software Implementation 7

9. Testing Process 8

10. Project in Context 8

10.1 Related products 8
10.2 Related literature 8

Appendices 9

1. Operation Manual 9

2. Alternative Versions 4

3. Other Considerations 4

1. Project Description

1.1 Problem Statement

The objective of our project is to build a system to control the blinds in the TLA (Transformative Learning Area) in Coover on Iowa State campus for anyone who uses the space to improve convenience and time-usage for the user of the blinds. This system will be accessible anytime of the day (morning, afternoon, or evening), user-friendly, and able to be maintained. We will use a motor and control system connected to a designed application to solve this problem.

1.2 Solution Description

Our solution relies on a hardware system to physically turn the blinds, and a website to control the hardware. The hardware shall be contained within a box affixed to the wall, and have a connection to the blinds. The website shall be available to all students and faculty using the TLA.

2. Design

2.1 Design Context

2.1.1 Broader Context

The problem we are solving is for the users of the transformative learning area (TLA.) This area is primarily used by engineering students and they are primarily using computers. The issue is there is not a good way to adjust the window blinds at the moment. Our project is to design a better way to control this so the light does not bother the students while trying to use the computers.

List relevant considerations related to your project in each of the following areas:

Area	Description	Examples
Public health, safety, and welfare	Can help users by saving time and by not having to look into the sun.	Less time trying to adjust blinds. Lets users accomplish work quickly by not having light obscure screens. Users will not have to look into the sun if blinds are easily put down.
Global, cultural, and social	This project affects the workplace of the TLA.	Students and faculty will have an easier time studying and working in the TLA.
Environmental	The project is based around retrofitting the old blinds instead of getting new ones, therefore less waste.	If new blinds were purchased, old would be thrown out, adding plastic to a landfill somewhere.
Economic	Retrofit saves money by not having to purchase all new blinds.	Less new items to be purchased.

2.1.2 User Needs

Engineering students need a way to use the TLA lab without the sun, because the sun's rays create glare on screen.

Engineering faculty need a way to work in the TLA lab without the sun, because the sun's rays create glare on screen.

2.1.3 Prior Work/Solutions

Similar products currently available in the market would be remote controlled blinds. Example would be Select Blinds. Remote control controlled blinds for ~$150 (depending on model and special features.)

We did not think this was the best solution for the problem because it would be quite expensive, and add questions such as who should get control of the remotes.

“Select Blinds” https://www.selectblinds.com/roman-shades/classic-roman-shades.html

2.1.4 Technical Complexity

The design is complex from a subsystems perspective. The design includes a motor, microcontroller, and a web application that includes both a front end and back end server. Each of these components will take time to get to work appropriately and many tests to ensure this is the case. The components will also all need to be seamlessly integrated and thoroughly tested to make sure that there are no issues in those connections.

2.2 Design Exploration

2.2.1 Design Decisions

Our design incorporates a device making use of an Arduino Uno Rev2 Wifi and a Stepper motor allowing for wireless communication and precision operation over the blinds to enable easily storable and retrievable state in case of system failure. Alongside this we have developed a JavaScript website that can act as a controller for several devices and one and a spring boot application server allowing for device addressing and communication.

2.2.2 Ideation

In the beginning when choosing the motor, we found and tested various motor options looking for the best speed and control in congruence with the system as a whole. We then looked into batteries and boards that would work well with the motor unit.

Our plans for the motor required a hard shift as we entered our second semester as we realized problems with motor drift that would create difficulties in our plans of software state estimation; which in effect, required us to think of alternative solutions. A shaft encoder and stepper motor were considered as replacements for the motor. The final product was determined to be the stepper motor.

2.2.3 Decision-Making and Trade-Off

For the type of application we went through a few options. We considered a (1) java application downloaded onto the computers in the lab, but decided that having to download could be an issue, as well as they wouldn't be accessible from a mobile phone.. We also thought about doing either a (2) Native mobile application or (3) hybrid mobile application so that the blinds could be controlled mobily, but decided against these options because they would require users to download before they could be useful. They would also create the opposite problem as before where they couldn't be used on a computer. In order to solve the problem of having to download software, we thought of a web application. We weren't sure if we wanted it to be (4) a web application designed for desktops, or (5) a web application designed for mobile devices. So we arrived at a final solution where we would make the design being a web application that was made to fit both computers and mobile devices.

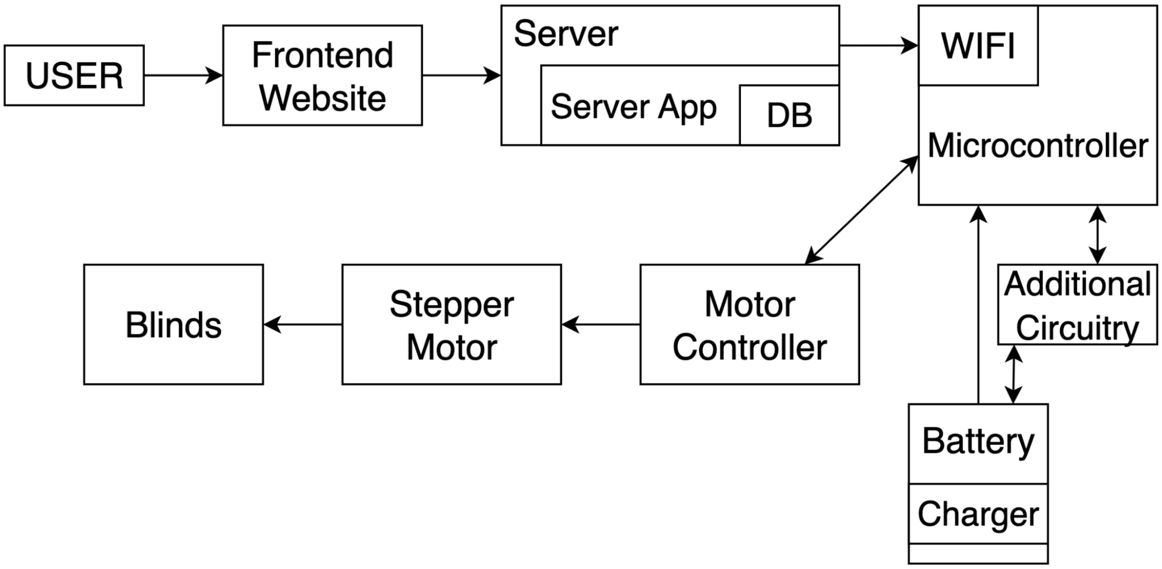
2.3 Proposed Design

Things tried- various ways to connect motor to blinds, different app layout styles

Things implemented- motor, testing speed controller, test blinds

Things tested- different motors, different speed controllers, different battery capacities

2.3.1 Design Visual and Description



2.3.2 Functionality

The design will function in the TLA lab by having the unit attached to the blinds, and connected to the online application. Anyone will be able to access the application and from there they can interact with the UI in order to send signals to the microcontroller. The microcontroller will then send the corresponding signals to the motor which will turn the blinds.

The design meets the functional requirements as it allows wholly to control the blinds, while having ranges in which they can do so, preventing them from breaking the blinds. Nonfunctional requirements will be met by making sure the website is easy to use and easy to get to, so that users can have a pleasant experience with it.

2.3.3 Areas of Concern and Development

- Security of the device and application, potential to circumvent the application account hierarchy to mess the device due to the nature of wireless.
- Potential for the device to interact in destructive ways with the blinds causing them to become unusable and defeat the purpose of the initial retrofit.

- Potential for use outside of the intended application causing inconvenience to the users such as blinds being continuously activated from a remote location.
- The use of strong network protocol should help mitigate some of the potential to access the device from outside the Official method of change.
- Use of a form of inexpensive lynch pin should help incase of brackage allow for safety of the device and blinds and should be easily fixed.
- A possible solution could be location based access to blinds from the app.

2.4 Technology Considerations

We are limited to the prices we can spend on our components. The project is something that should be reproducible at a relatively low cost. This does mean, though, that we are only able to use particular motors, microcontrollers, and speed controllers in order to reduce the overall cost associated with the full unit.

2.5 Design Analysis

Our initial design proposal while similar ultimately Required changes most of the device while we maintain our microcontroller throughout the motor and driver end up as completely different units. In our initial analysis of the design we failed to realize its core flaws ultimately lead to greatly changing our device.

2.6 Changes From the First Semester

Between the first and second semester of this senior design project we had to critically analyze the system we were creating and the goal we were trying to achieve. Thus we had to go beyond our plan and be flexible with our design.

2.6.1 Hardware Changes

There were several design changes to our hardware device between the first and second semester of this course. We realized that our motor control would not be precise enough for the original software design. We therefore had changed the hardware to have more precision, thus swapping our motor and motor controller with stepper motor and microstep driver.

2.6.2 Software Changes

Going forward into the second semester we decided go forward with the Implementation of location services as means of deterrent against would be pranksters. Our device code we need a way to maintains state through power loss thus we made use of a battery monitoring and persistent storage through the controllers eeprom to create our solution.

3. Functional Requirements

3.1 Hardware Requirements.

- The device shall be able to receive UDP (User Datagram Protocol) requests from the server.
- When a UDP packet is received by the device it shall modify the current state of the blinds to the provided position
- The device shall have precision control such that the blinds attached to it will not break.
- When the Power on the device reaches a low state the device shall save its current state and disable wireless communication.

3.2 Software Requirements

- When a user logs in to the application, it shall use NextAuth to authenticate them over Google.
- The application shall be accessible over both standard computers and mobile devices.
- When a user logs on and is authenticated by the admins list they shall be able moderate the blinds system.
- The web application shall be able to communicate with the server over HTTP.
- The application shall display the current states of the blinds given server information.

4. Nonfunctional Requirements

4.1 Hardware

- Reduce unnecessary size
- Reduction in wiring
- Maintainable by Ease of Access to Components in Box
- Scalable and Reproducible
- Unobstructive Look in TLA

4.2 Software

- Easily Navigable and Intuitive
- Maintainable by Admins
- Accessibility - No Training to Use

5. Relevant Standards

- *"IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems," in IEEE Std 2050-2018 , vol., no., pp.1-333, 24 Aug. 2018, doi: 10.1109/IEEESTD.2018.8445674.*

  These standards apply to our project due to the incorporation of a small scale embedded system. We will be operating this in real time in order to incorporate direct procedural interaction between the interface and the action of a physical motor that will, in turn, open or close the blinds.

- *"Preprint Proposals for the 2022 Edition of the National Electrical Safety Code (NESC(R))," in Preprint Proposals for the 2022 Edition of the National Electrical Safety Code (NESC(R)) , vol., no., pp.1-867, 1 July 2019, doi: 10.1109/IEEESTD.2019.8753791.*

The standard related to this has to do with how safety is handled in matters using electrical supplies. We need to follow this standard in our project in order to make sure we follow proper safety while building our circuit that will be connected to the dc motor.

- *"IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks--Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Enhanced Throughput for Operation in License-exempt Bands above 45 GHz," in IEEE Std 802.11ay-2021 (Amendment to IEEE Std 802.11-2020 as amendment by IEEE Std 802.11ax-2021) , vol., no., pp.1-768, 28 July 2021, doi: 10.1109/IEEESTD.2021.9502046.*

  This standard relates to our project as it is the standard for wireless network and information transfer using them as our blinds will ultimately be controlled across wifi. This standard will be used heavily.

6. Constraints

- Cost/Resources - $500 for Project, and minimal for reproduction
  - The estimate cost of this prototype
    - 1 prototype should cost below 130, with selling price at 150.
      - Hard to get parts that are cheap.
- Ordering Parts
  - Ordering parts take up almost two weeks minimum for it to arrive.
- Time - Two Semesters
  - Different Schedule from each team member due to work/School/class load.

7. Security Concerns & Countermeasures

7.1 Physical Security

Our devices are housed in the TLA so thus they are kept safe physically by any building security the Coover has access to such as locking doors and security cameras.

7.2 Cybersecurity

Our device connection from website to server is made over https to enable encryption of data between them.

The server and devices exist on Iowa State network making them inaccessible to attacks from off campus

8. Implementation Details

8.1 Hardware Implementation

- Power
  - 18650 battery (quantity: 4) with battery series connection holder
  - Power Buck Module 6V32V12V24V to Q3.0 fast charging single USB
- Assembly
  - Stepper Motor Driver
  - Stepper Motor
  - Arduino UNO wifi
  - Industrial Box
  - Plexi-plate
  - Bolts with spacers
- Wiring Connection and Circuitry
  - Voltage divider between battery and arduino
  - EEPROM (electrically erasable programmable read-only memory) button reset
  - Stepper motor controller to arduino
  - Stepper motor controller to stepper motor
  - Battery to motor controller

8.2 Software Implementation

- Backend
  - Linux Ubuntu server VM
    - Installed the software mysql, java 17, and maven
    - Created bash script and service to simplify the application startup process
  - Spring boot application uses:
    - Setup CORS for usage with javascript frontend
    - Configured for https
- Frontend
  - React JavaScript Frontend
    - Chakra-UI as a component library

    - Landing Page and Subsequent Pages
        - User log on before loading up other pages
    - Room Pages
        - Pages for editing the blinds specific to a room, such as the TLA
    - Admin Page
        - Authenticated users can access admin page where they can add/remove blind systems

9. Testing Process

9.1 Hardware

- Voltage input and output
    - Ardurino (between 1 to 5 volt) =>4.83Volt
    - Stepper motor (between 9- 42 volt) => 14.50 Volt
    - lithium ion battery testing Series => 14.50 Volt
- Testing on part assemble
    - Non engineering students can assemble the hardware component from start to finish within 30 minutes.

9.2 Software

- Automated Testing done with Cypress
    - Cypress is a service intended for testing react javascript programs
    - Tests frontend buttons and operations
    - Tests are written and run to completion for all pages
- Made use of a postman to test Rest APIs
- Arduino serial monitor for packet reception

10. Project in Context

10.1 Related products

- Custom-Fit Light Filtering Roman Shades | SelectBlinds.com
- Select Blinds are a good comparison
    - Select Blinds have you first pick blinds, and then can optionally add motorization, and remotes.
    - These Blinds would have to be purchased, installed, and the current blinds would have to be removed for this solution.
    - These blinds also do not provide a web page to control, but remotes instead.

Appendices

1. Operation Manual

Blinds Arduino Setup Guide

Prerequisite Software:

Arduino IDE (downloadable at https://www.arduino.cc/en/software)

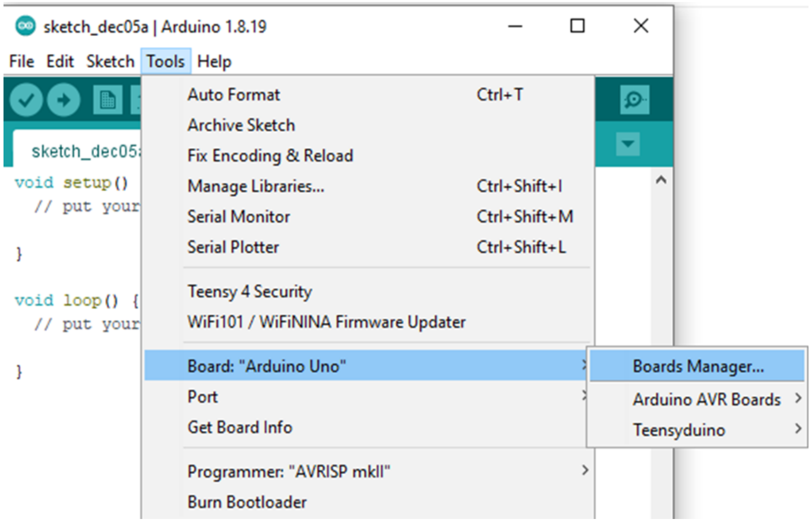
Git Bash(downloadable at https://git-scm.com/downloads)

Setup Up Arduino IDE

In order to connect the device to your computer there are a few step that must be completed first

Click Board Manager from

Boards under the tools tab of the ide



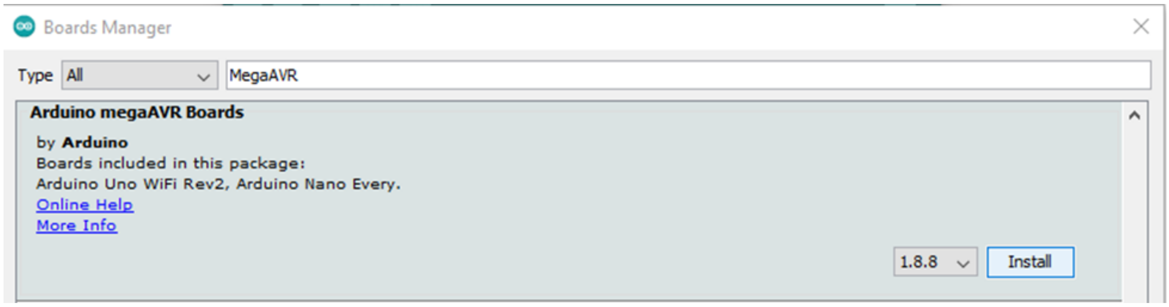
```
void setup()
  // put your

}

void loop() {
  // put your

}
```

Type “MegaAVR” into the search bar and click install

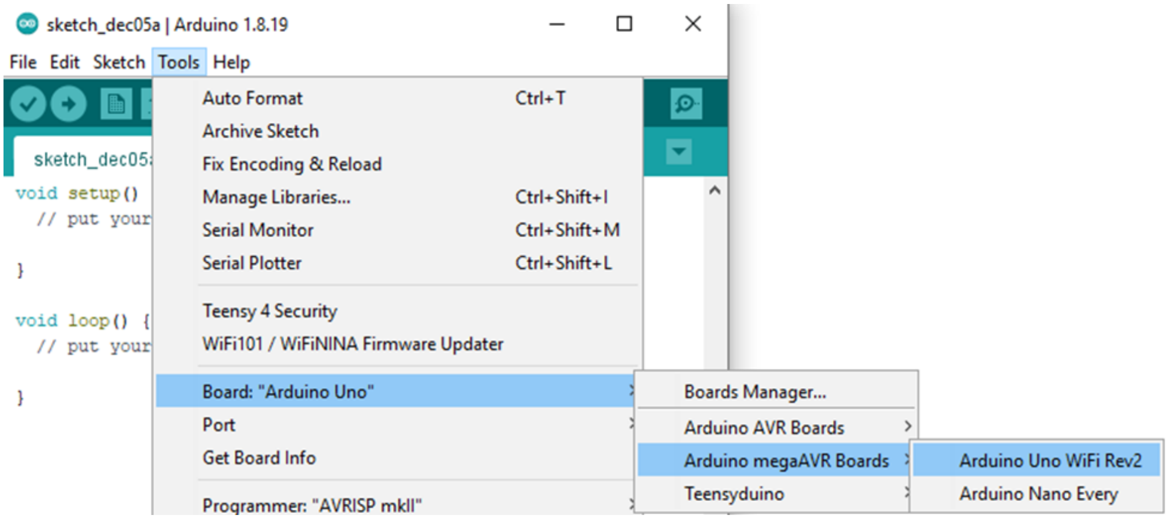


Type: All — MegaAVR

Arduino megaAVR Boards

by **Arduino**
Boards included in this package:
Arduino Uno WiFi Rev2, Arduino Nano Every.
Online Help
More Info

Moving back the where we found board manager find the “megaAVR” tab and click “Arduino uno WiFi rev2”



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

In the port drop down in the tools tab you will change it to whatever your Arduino device has been connected to. You may have to consult device manager to find the connected port.

Obtaining the Arduino MAC

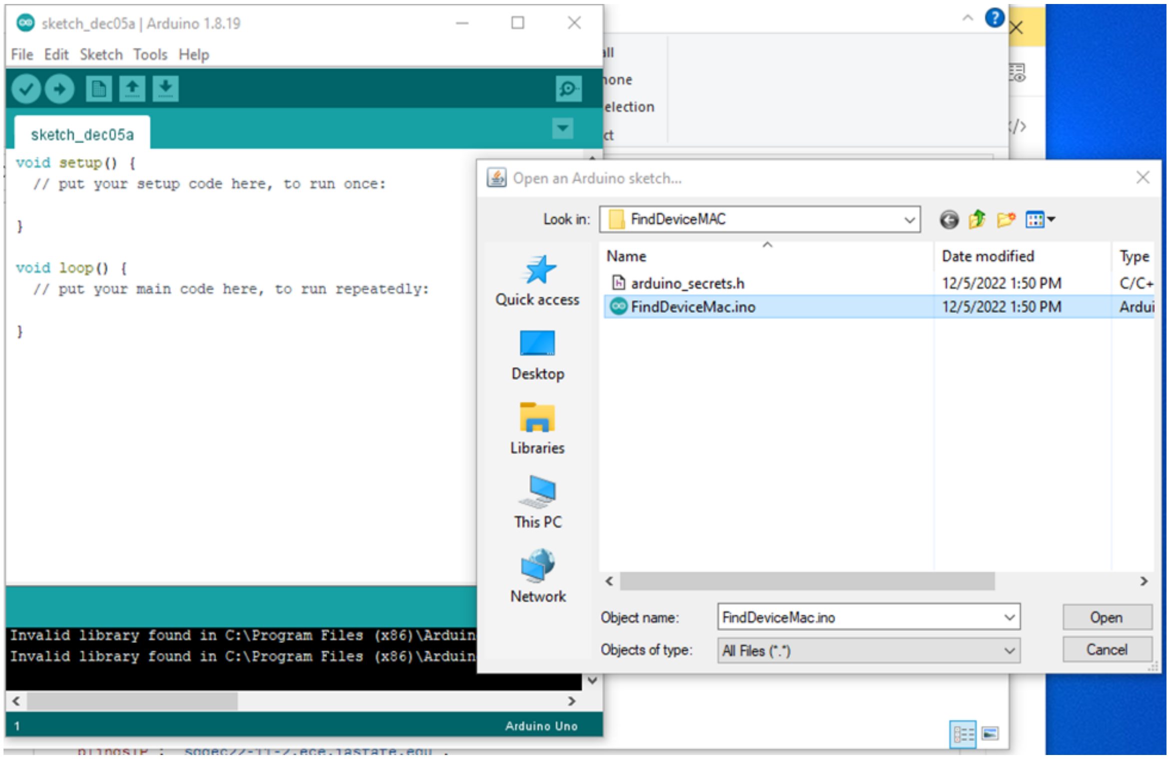
First in a make new folder to house the git repository. Open git bash and cd into this new folder. Once navigated there type the following:

git init

git clone https://git.ece.iastate.edu/sd/sddec22-11.git

This should add a folder called sddec22-11 into your directory.

Going back to the Arduino ide under file tab click open and navigate to the hardware subdirectory of git directory we just cloned. From here you will opn the FindDeviceMac.ino from the FindDeviceMAC subdirectory of hardware.


sketch_dec05a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_dec05a
Open an Arduino sketch...
Quick access
Desktop
Libraries
This PC
Network
Open
Cancel
Arduino Uno

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
Invalid library found in C:\Program Files (x86)\Arduin
Invalid library found in C:\Program Files (x86)\Arduin
```

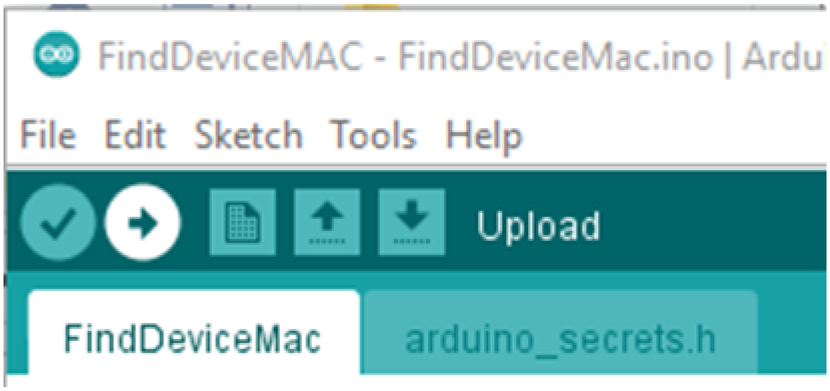
Look in: FindDeviceMAC

Name	Date modified	Type
arduino_secrets.h	12/5/2022 1:50 PM	C/C+
FindDeviceMac.ino	12/5/2022 1:50 PM	Ardui

Object name: FindDeviceMac.ino

Objects of type: All Files (*.*)

Now click the upload button

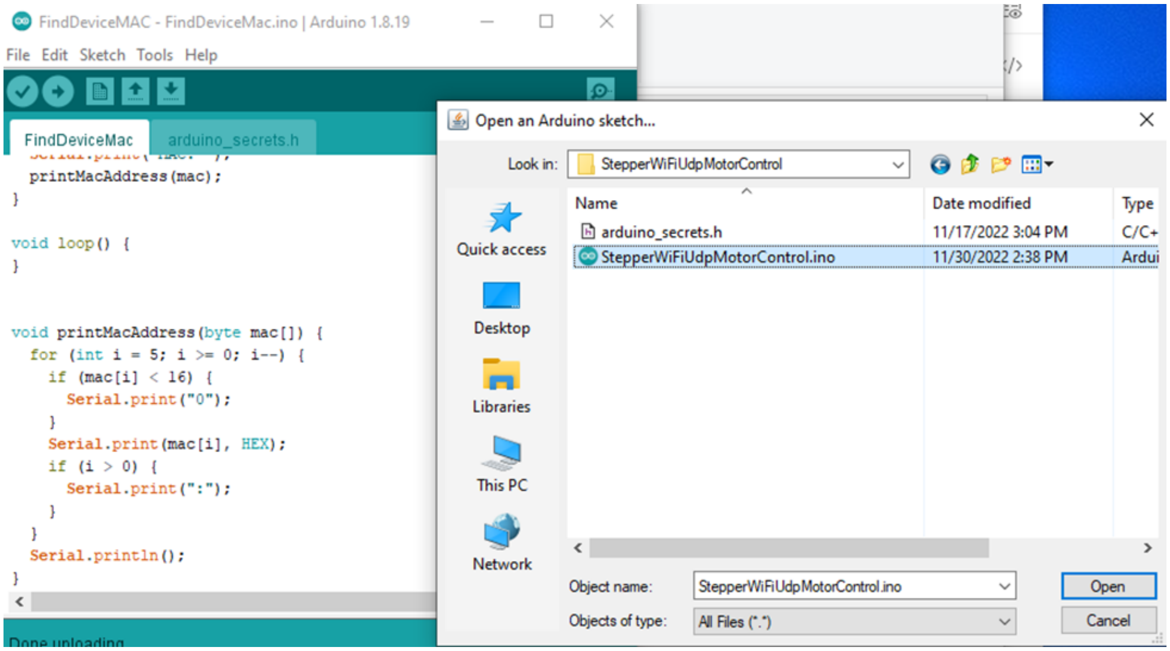

FindDeviceMAC - FindDeviceMac.ino | Ardu
File Edit Sketch Tools Help
Upload
FindDeviceMac
arduino_secrets.h

The on the keyboard hit Ctrl + Shift + M. The mac address will now be displayed on the screen use this to register iastate network over ASW.

Note: The way device are found in the web application is based on the ip naming format sddec22-11-{#}.ece.iastate.edu. Should you want to change this modification to the frontend would be required

Installing the Control Code

Using the same process we used to obtain our Mac address will now open in the Arduino ide the StepperWiFiUdpMotorControl.ino.


FindDeviceMAC - FindDeviceMac.ino | Arduino 1.8.19
File Edit Sketch Tools Help
FindDeviceMac
arduino_secrets.h
Open an Arduino sketch...
Quick access
Desktop
Libraries
This PC
Network
Open
Cancel

```
  printMacAddress(mac);
}

void loop() {
}


void printMacAddress(byte mac[]) {
  for (int i = 5; i >= 0; i--) {
    if (mac[i] < 16) {
      Serial.print("0");
    }
    Serial.print(mac[i], HEX);
    if (i > 0) {
      Serial.print(":");
    }
  }
  Serial.println();
}
```

Done uploading.

Look in: StepperWiFiUdpMotorControl

Name	Date modified	Type
arduino_secrets.h	11/17/2022 3:04 PM	C/C+
StepperWiFiUdpMotorControl.ino	11/30/2022 2:38 PM	Ardui

Object name: StepperWiFiUdpMotorControl.ino

Objects of type: All Files (*.*)

Again click the upload button and now your Arduino is ready to be assembly with the blinds device.

Blinds Server Setup Guide

Prerequisite:

On your server have the following programs installed:

If you want to make use of different version some change to software maybe required

Java version 17 (easily installed using apt-get)

Maven version 3.8.6 (you may need to install this from binaries and apt-get was not pulling it. This version is the minimum requirement to run java 17)

Mysql version 8.0.31 (again this should be easily setup with)

Setup:

Clone the Git:

Type the following-

```
mkdir gitRepo

cd gitRepo

git init

git clone https://git.ece.iastate.edu/sd/sddec22-11.git
```

Sql setup and adding application.properties:

```
sudo mysql

CREATE USER 'ETG'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON * . * TO 'new_user'@'localhost';

FLUSH PRIVILEGES;


CREATE DATABASE BlindsDatabase;

GRANT permission ON database.BlindsDatabase TO 'ETG'@'localhost';

FLUSH PRIVILEGES;
```

```
cd /getRepo/sddec22-11/Backend/BlindsProject/src/main/resources

nano application.properties
```

Type the following

```
server.port=443

server.ssl.key-store: classpath:keystore.p12

server.ssl.keystore-password: your choice of password

server.ssl.keyStoreType: PKCS12

server.ssl.keyAlias: blinds.app


# Configuration for MySQL Database

spring.jpa.hibernate.ddl-auto=create

spring.datasource.url=jdbc:mysql://localhost:3306/BlindsDatabase

spring.datasource.username= This will be a username of your choice to the sql server

spring.datasource.password= This will be a password of your choice to the sql server
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.show-sql:true

spring.jpa.database=MYSQL
```

save this

In the same folder then run

```
Keytool -genkey -alias blinds.app -storetype PKCS12 -keyalg RSA -keysize 2048 -keystore keystore.p12 -validity 3650
```

It will ask you for a password type the one you chose in the application.property

It will also ask several questions just hit enter until it asks you for the confirmation of correct information and type yes.

Installing server services

Type the following-

```
cd sddec22-11

sudo ./server.sh install
```

Building server app from source code

```
cd sddec22-11

./server.sh build
```

Starting and Stopping Spring application

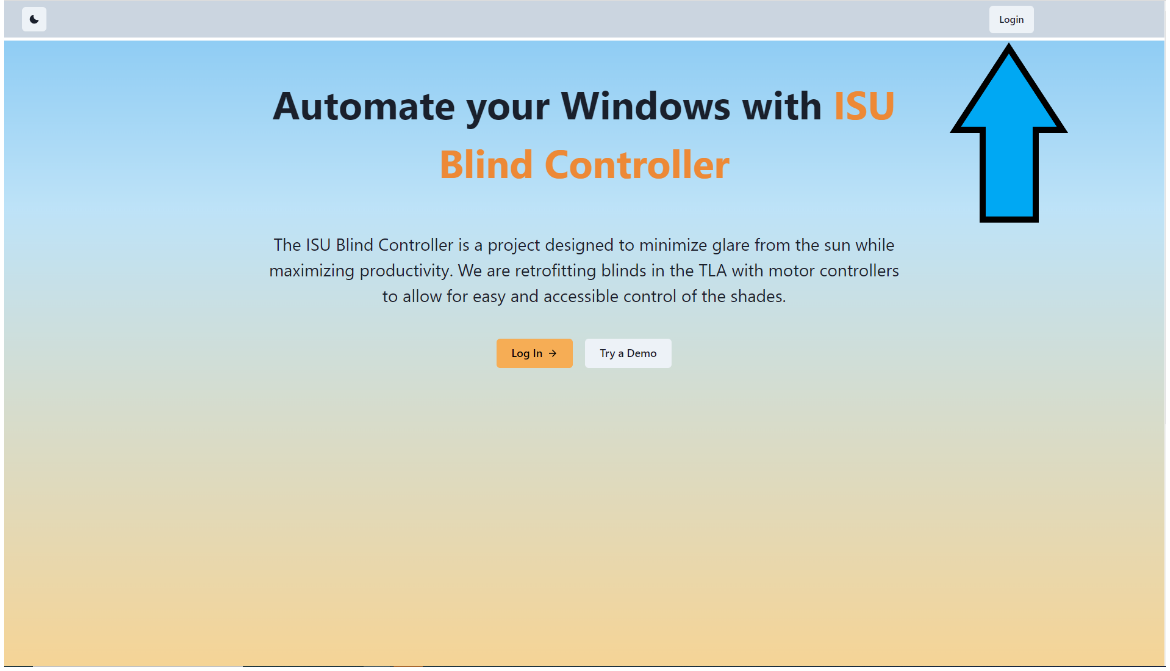
```
sudo systemctl start system-web-demo.service
```

```
sudo systemctl stop system-web-demo.service
```

When rebuilding the application you should first stop the application.

Website Use Guide

1. Enable https for the website
    - a. Go to https://sddec 22-11.ece.iastate.edu/blinds
    - b. Approve use for our website
2. Blinds Website Access - Go to blinds.vercel.app
    - a. Log In



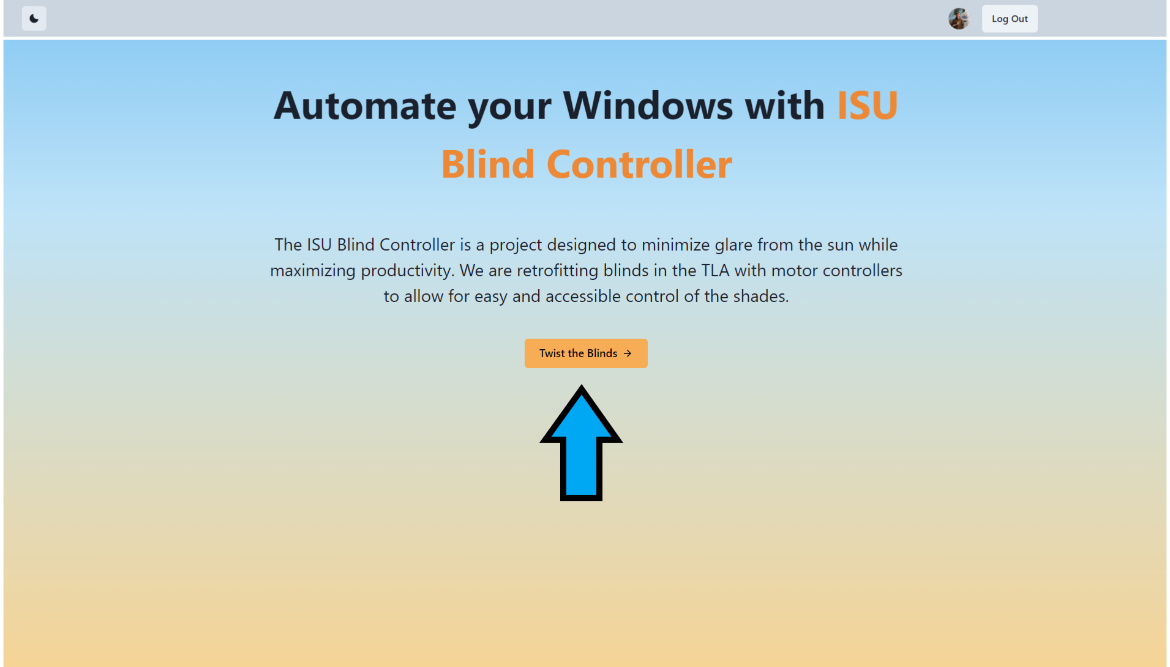
Automate your Windows with ISU Blind Controller

The ISU Blind Controller is a project designed to minimize glare from the sun while maximizing productivity. We are retrofitting blinds in the TLA with motor controllers to allow for easy and accessible control of the shades.

i. Users will then be prompted with google log on.

ii. Users must log in with an @iastate.edu email to access the blinds.

b. Click the “Twist the Blinds” button and it will take you to the next page



Automate your Windows with ISU Blind Controller

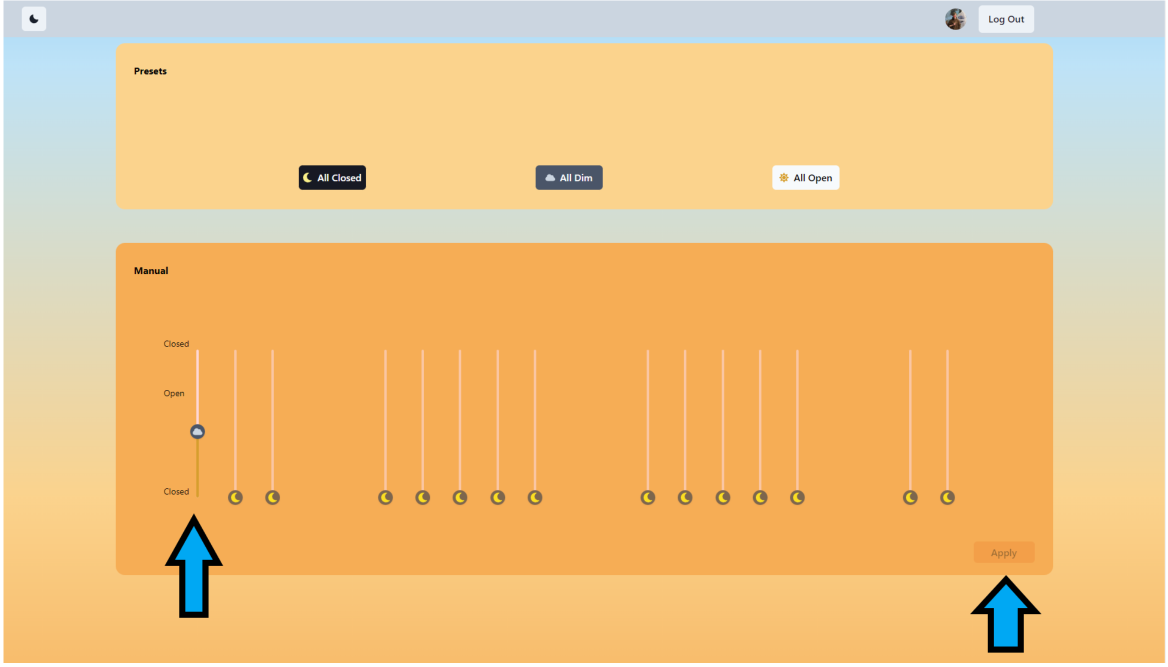
The ISU Blind Controller is a project designed to minimize glare from the sun while maximizing productivity. We are retrofitting blinds in the TLA with motor controllers to allow for easy and accessible control of the shades.

c. Logged on users, who are in the TLA can adjust sliders

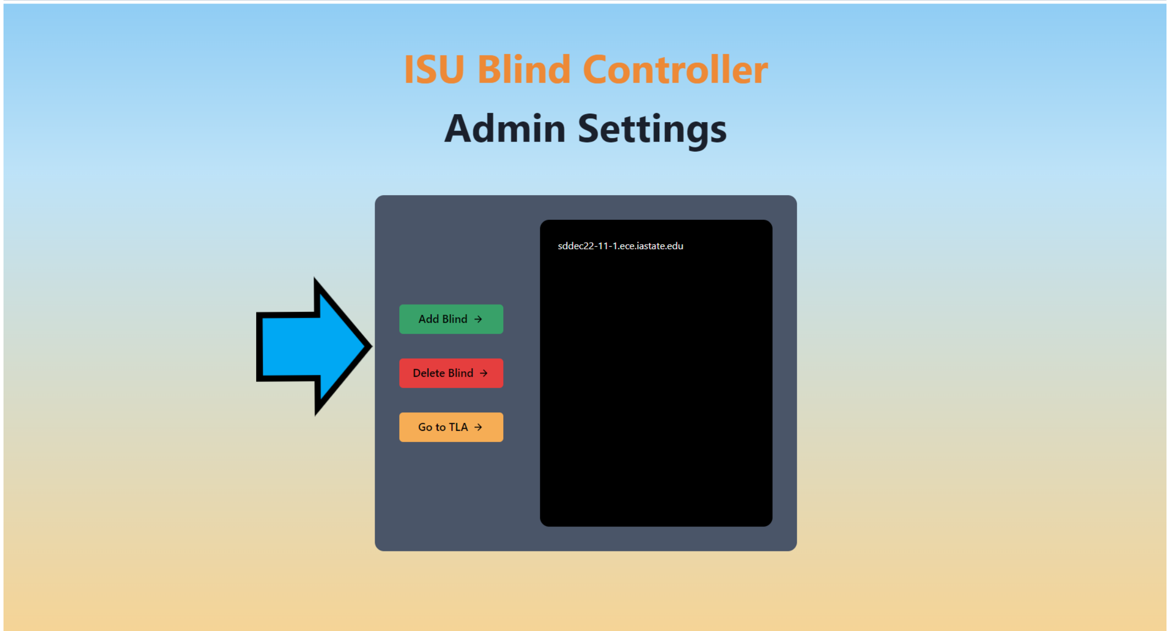
i. Sliders represent how twisted the blinds are

ii. Sliders can be move to rotate the blinds when “Apply” is pressed

iii. When the blinds are at the wanted positions “Apply” sends a signal to the machine that will change the blinds position.



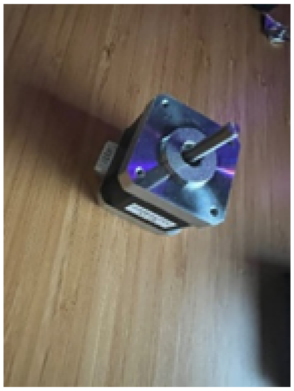
3. Admin Page
    a. Go to the Page “https://blinds.vercel.app/admin”
    b. Log in again (or stay logged in from before)
    c. Admin users will have the “Add Blind” and “Delete Blind” Buttons



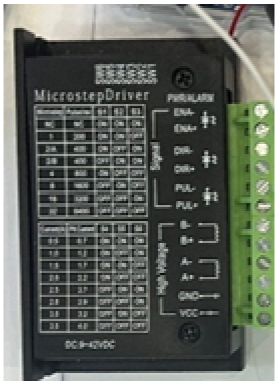
    d. These buttons have the ability to add and remove blind systems from the website’s control. This allows users to use net registered devices (See “Blinds Server Setup Guide”) .

Device build Guide

- **There are a total of 6 major items.**
    - **Stepper Motor x1**

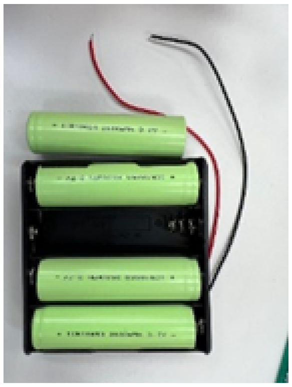


- **Stepper Motor controller x1**



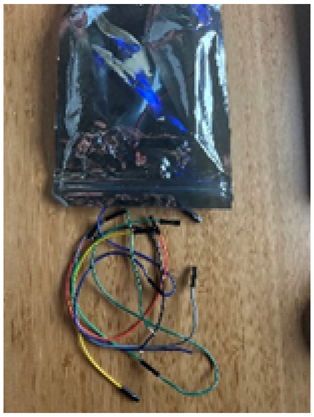
- **Ardruino Uno x1**

- **Battery 18650 x4**
- **Battery 18650 Holder x1**



o Clear plastic tube x1

O Wire

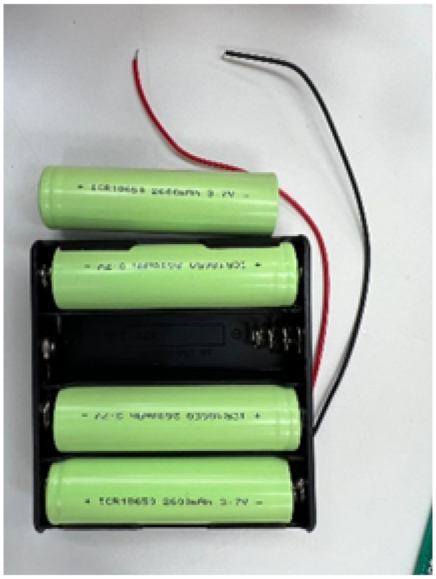


§

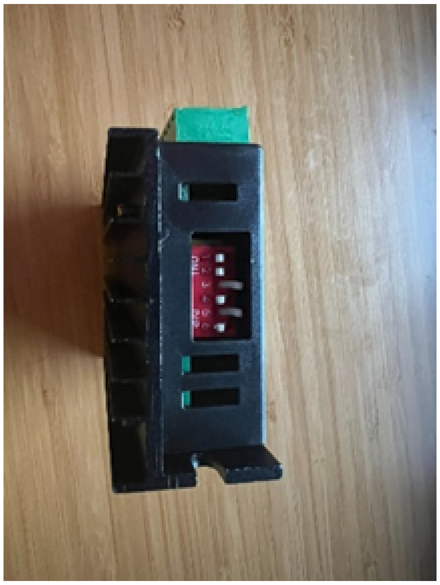
Schematic:

- **Part Preparation:**

o Battery: When putting battery in the holder, please make sure the Positive and negative is in the correct location to prevent any injury. Please keep the positive (red) and negative (black) wire away from each other when the battery is in place to prevent any connection between them.

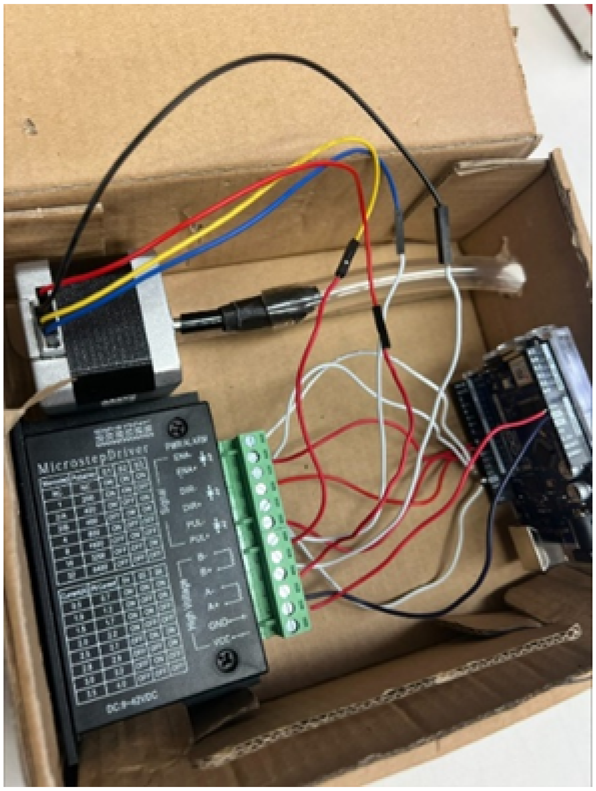


· **Stepper motor controller: Please follow as shown in the figure below. For better instruction please adjust 1,2,4,6 to the pointing down position while 3,5 are pointing upward.**



· **Step 1: On the first step, make sure the battery is working correctly, by connecting it to a multimeter. if the battery is place correctly the multimeter should show around 14.5 to 15.0.**

· **Step 2: Connection**



- **First we going to do the connection between the stepper motor to the stepper**
  - **There are 5 pins on the motor, we will be using 1,3,4,6**
    - **1st wire: we connect the red Wire to the far-left side of the stepper motor to the A+ (High Voltage) as shown on the micro step Driver**
    - **2nd wire: we connect the black wire to the third pin, started counting from left side. Connecting it to A-**
    - **3rd wire: we connect the yellow wire to the fourth pin. Connecting it to B+**

        - **4th wire: we connect the blue wire to the sixth pin. Connecting it to the B-**

- **We will be doing the connection between stepper motor driver to the Arduino Uno**
    - ○ **There are a total of 8 wire connections**
        - **1st wire: we connect ENA- (white wire) to the GND pin**
        - **2nd wire: we connect ENA+ (Red wire) to the 10 pins**
        - **3rd wire: we connect DIR- (White wire) to the GND pin**
- ` **4th wire: we connect DIR+ (red wire) to the 9 pins**
        - **5th wire: we connect PUL- (white wire) to the GND pin**
        - **6th wire: we connect PUL+(red wire) to the 8 pin**
        - **7th wire: we connect the Vin on the Arduino Uno to Stepper motor driver VCC**
        - **8th wire: we connect the GND on the Arduino Uno to stepper motor Driver GND**