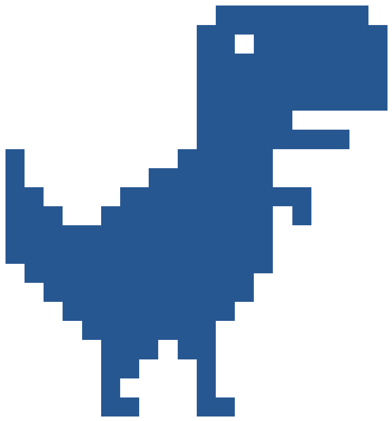
AE 3531

T-RECS

User Manual



Brian Kaplan & Morgan Cook
GEORGIA INSTITUTE OF TECHNOLOGY

Contents

1. Introduction

The purpose of this user manual is to provide the students of AE 3531 with a full understanding of how the Transportable Rotorcraft Electronic Control System (T-RECS) operates. This document will feature an overview of the components that make up the T-RECS and their respective function along with an assembly guide for the entire system. The necessary software installations for testing and operating the T-RECS will also be covered. Finally, an overview of the software that runs on the T-RECS will be provided to give students an intuitive understanding of how the system's PID controller functions.

It is assumed that students possess a basic understanding of controls theory, programming techniques (preferably in C++ or a similar language), and designing/modeling in SolidWorks. However, students are not expected to have experience with more advanced programming techniques (digital filtering, timer interrupts, serial data logging, etc.) or electronic design (PWM signals, analog to digital converters, PCB design, etc.).

2. Parts List & Overview

The T-RECS is composed of several components that students will be provided in the form of a kit along with a set of mechanical parts that students will be expected to manufacture on their own via 3D printing or laser cutting. The kit will contain the following components:

1. 1x Propeller
2. 2x Ball Bearings
3. 2x M2 Screws
4. 2x M2 Nuts
5. 4x M3 Screws
6. 4x M3 Standoffs
7. 4x M3 Nuts
8. 1x M4 Screw
9. 1x M4 Nut
10. 1x Printed Circuit Board (PCB)
11. 1x Arduino Nano
12. 1x Power Transistor
13. 1x Rotary Position Sensor
14. 1x 40-Pin Female PCB Header
15. 2x 2-Pin Male PCB Connector
16. 1x 4-Pin Male PCB Connector
17. 1x Wiring Harness
18. 1x Electronic Speed Control (ESC)
19. 1x Brushless DC Motor
20. 1x Mini USB Programming Cable
21. 1x Power Supply

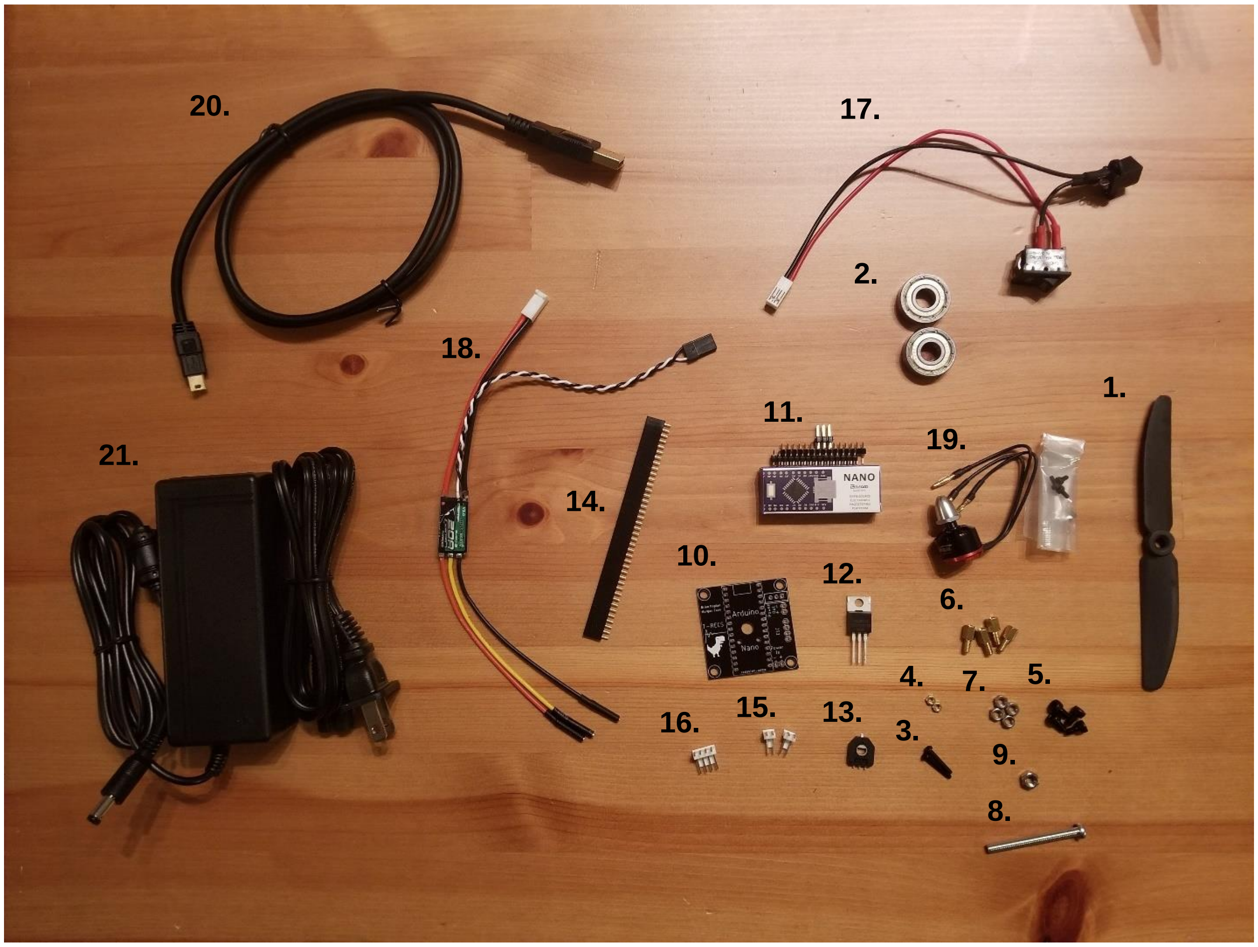


Figure 2.1. Overview of kit components.

2.1 Electrical Components

The electrical system of the T-RECS is summarized in the schematic below:

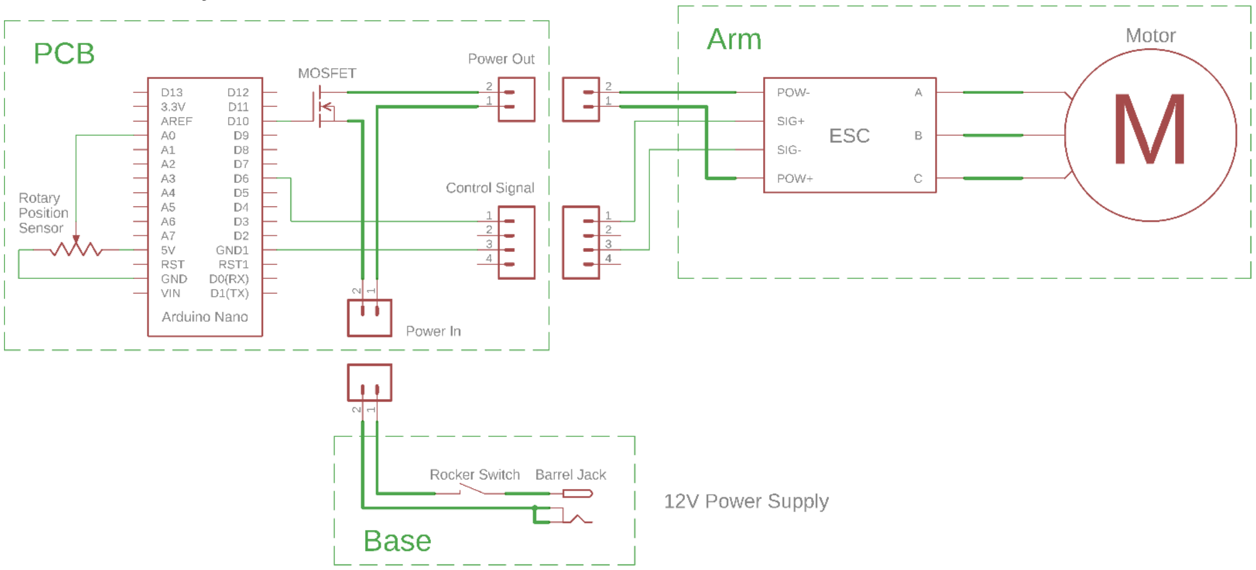


Figure 2.2. Electrical system of the T-RECS. The thick electrical connections denote the high current path that powers the motor on the T-RECS. The thinner electrical connections represent low-current signal lines that are used for controlling the T-RECS.

2.1.1 Brushless Motor

The T-RECS is propelled by a brushless DC motor (BLDC). These are very common within the RC hobbyist community and can be found on quadcopters, RC helicopters, RC airplanes, etc. The operation and internal structure of brushless motors is more complex than conventional brushed DC motors. While brushed DC motors possess 2 simple input terminals (positive and negative), brushless motors possess 3 input terminals whose function varies as the motor is driven. As a result, complex electronic control circuitry is required to properly drive then. The details of this, however, are beyond the scope of this project.

2.1.2 Electronic Speed Control (ESC)

To avoid the complexity of BLDCs, the T-RECS utilizes an electronic speed controller (ESC) to handle the complexities of properly driving the motor. An ESC converts a DC power input (i.e. the power supply or a battery) combined with a simple control signal into the appropriate drive signal to operate the motor. Like BLDCs, ESCs are also very common amongst RC hobbyists and they provide a linear means of controlling the output thrust of the T-RECS.

2.1.3 Printed Circuit Board (PCB)

Most of the system's electronics are housed on a printed circuit board (PCB) to keep the wiring and circuitry simple and unobtrusive. The circuit board features an Arduino Nano as the system's microcontroller, a transistor for controlling power to the rest of the system, a rotary position sensor for measuring the system's pitch, and several receptacles for connecting the necessary wires to the board.

2.1.4 Power Electronics

The other power electronics can be found in or attached to the base of the T-RECS. The rocker switch and barrel Jack are connected to form the system's wiring harness into which the power adapter will connect.

2.2 Mechanical Components

The mechanical structure of the T-RECS is composed of four unique components that will need to be manufactured via 3D printing or laser cutting. These parts are a single base, two towers, two shafts, and one arm.

2.2.1 Base

The base serves as the anchor point for the system and houses the rocker switch and barrel jack for providing the system with power. The base should always be clamped down or securely fastened to a solid flat surface such as a table or workbench whenever the system is running. The base should be oriented such that the arm hangs off the edge of the table, free of any obstructions that could interfere with the propeller.

2.2.2 Towers

The two towers are designed to be identical for ease of manufacturing and simplicity. The towers feature a protective overtravel stop to prevent the arm from going past about 80° above horizontal in the event of a malfunction or improperly tuned controller. Both towers contain bearings for supporting the necessary low-friction rotation of the arm. The left tower is intended for mounting the PCB based on the PCB's design and wiring of the system.

2.2.3 Shafts

The shafts connect the arm to the bearings in the towers and mate with the rotary position sensor the T-RECS uses for measuring its pitch. A secure fit is required between the shaft's square peg and the arm *and* between the shaft's semicircular dowel and the sensor to ensure reliable measurements. However, the fit of the shaft in the bearing is far less important.

2.2.4 Arm

The arm is the dynamic component of the T-RECS. It houses the ESC at its pivoting end and the motor on its free end. The ESC slides firmly into the designated slot while the motor is mounted securely via four M2 screws on the underside.

3. Manufacturing & Assembly

3.1 3D Printing

3.1.1 From STLs to G-code

3D printing (often referred to as additive manufacturing) uses G-code similarly to many other CNC machining tools to precisely deposit thin layers (0.4mm to 16µm) of the manufacturing material one by one to form a complete three-dimensional part.

To start, the provided STL (an abbreviation of “stereolithography”) files that contain models of the mechanical parts must be converted to the necessary G-code that 3D printers use. This can be done at any of makerspaces on campus. The most frequently used makerspaces for 3D printing are the Aero Makerspace and the Invention Studio. The STL-files should be brought on a USB drive and the Prototyping Instructors or Mentors at the makerspace can assist in converting the STL files to appropriate G-code for the specific 3D printers in that makerspace. This conversion is called “slicing” and the software that performs this is known as “slicing” or “slicer” software. Common slicing programs include Ultimaker Cura, Simplify3D, and Slic3r.

3.1.2 Print Settings and Configuration

The manufacturing material used by 3D printers is usually found in the form of 1.75mm or 3.00mm thick strands of the material stored as large spools and is commonly referred to as “filament.” Most of the printers on campus utilize either PLA (an abbreviation of “polylactic acid”) or ABS (an abbreviation of “acrylonitrile butadiene styrene”) as their filament. However most hard plastic-based filaments (such as PETG or Nylon) will be sufficient for this system.

A layer height of 0.2mm or greater is recommended to minimize print time. It is also necessary to print the parts with supports for the portions of the parts that have overhangs. When arranging the components on the printer’s build plate, be sure to print hem such that the least amount of support material is used. This both reduces print time and improves print quality because there is less material to print and unsupported surfaces tend to print more cleanly.

The recommended orientations are as follows: The base should be printed upside down; the towers should be printed with the protrusions facing upwards. The arm can be printed in two orientations, either with the holes at the end facing up or with the square hole in the side facing up so the roof for the ESC makes a U shape. Finally, the shafts should be printed with the square peg facing down. It is also recommended that at least two extra shafts (beyond the two required) be printed. This is because the shafts are fragile and can also vary slightly in their precision which can result in improper fits with the arm, bearing, or sensor. Four shafts are recommended for adequate redundancy.

3.1.3 Troubleshooting

Sometimes a print may fail for any of several reasons. A common cause of failed prints is poor adhesion of the parts to the printer's build plate. To combat this, a glue stick can be applied to the build plate immediately before the print begins to improve adhesion. The parts may also be printed with a brim or raft to better adhere; however, this approach will lead to longer print times. Additionally, prints can fail if parts are situated too close to the edges of the build plate due to uneven heating. Dividing a print into two smaller batches is the simplest way to avoid this problem if large batch prints are failing.

3.1.4 Completing Prints

To improve the chances of successfully completing prints on time, Morgan will hold weekly workshops in the Invention Studio after hours to provide an in-depth overview of 3D printing. Additionally, any printers that are not in use after hours will be available to the students that choose to attend.

For students that do not print their parts during these workshops, it will probably be necessary to show up about 15 minutes before a makerspace opens (10:00am for the Invention Studio and 9:00am for the Aero Maker Space) to secure a printer. Otherwise, students will likely have to wait in a queue for several hours until a printer frees up.

If printing resources become too limited during open hours, Morgan will open additional hours for printing during the week and over the weekends as needed. It is still highly recommended that students show up early in the week and early in the morning to guarantee that their prints complete successfully. Bring failed prints to check-offs so Brian and Morgan can provide replacement parts.

3.2 Soldering

The fabrication of the PCBs has been completed by a professional board manufacturer to minimize the complexity of the circuitry required for the T-RECS. However, the assembly of the PCB is the responsibility of the students. This means that the students will be required to solder all the necessary components onto the PCB for this project.

Soldering is a process by which two or more items (usually metal) are joined together by melting and putting a filler metal ("solder") into the joint, with the filler having a lower melting point than the adjoining metals.

There are seven components that need to be soldered to each PCB: Two 2.54mm female headers for the Arduino Nano, three connector receptacles for connecting wires to the PCB, one transistor, and one rotary position sensor.

3.2.1 Through-Hole Components

Most of the components that need to be soldered for this project are all through-hole components, meaning that they consist of metal pins that are insert **through holes** on the PCB that are plated with a corrosion resistant metal such as tin, nickel, or gold. The soldering then connects the component to the PCB by joining the pin to the plated hole.

3.2.2 Surface Mount Components

The rotary position sensor is a surface mount component, meaning that it simply rests on metal-plated pads on the PCB without being secured through mounting holes. This requires a slightly more advanced technique to properly solder such components to their boards. Great care must be taken when soldering the rotary position sensor to the PCB due to the importance of the sensor being oriented in the proper position on the board so that it may properly connect with the shaft.

3.2.3 Techniques and Tips

Soldering can be tricky for those who have never tried it; however, a few simple tricks can significantly improve the quality and reduce the required effort for a solder job.

3.2.3.1 Heat

Molten solder will readily adhere to the nearest metal surface that is sufficiently hot. This is because a hot surface keeps the solder in its liquid state and metal surfaces help break solder's surface tension in a process known as "wetting," allowing solder to flow easily. Therefore, to ensure that solder properly flows into the joint being soldered, both parts of the joint (the pin and the plated hole) must be heated by the iron.

3.2.3.2 Flux

In addition to being hot as described above, a surface must also be metal for solder to easily adhere to it. Often, these metal surfaces develop a thin oxide layer from exposure to the air that prevents the underlying metal from contacting the solder. This is what flux is designed to combat. Flux is a chemical cleaning agent that should be applied to a surface before soldering to remove any oxidized metal from the surface and seal out air to prevent further oxidization.

3.2.3.3 Positioning

It is common for soldering novices to improperly secure the PCB they are soldering to their workbench (or not secure it at all). This often leads to unsteady and imprecise soldering as the board or components shift around while whomever solders tries to steady the PCB, the component, the solder, and the soldering iron all with only two hands. To prevent this, it is often wise to secure the PCB and component to the workbench with the help of a clamp, helping hands, or simply something heavy to keep components from easily shifting around.

3.2.3.4 Other Resources

There are several readily available resources for students to learn or improve on their soldering skills. A quick internet search yields countless tutorials and instructional videos on how to properly solder starting at a beginner's level. Additionally, most of the makerspaces on campus also possesses soldering equipment for students to use freely and most of the Prototyping Instructors/Mentors in those makerspaces are well-versed in soldering and are happy to teach students.

3.2.4 Completing Soldering Work

As mentioned above, most of the maker spaces on campus possess soldering equipment that students may use to solder their PCBs for this project. Additionally, Brian and Morgan will host workshops after hours in the Invention Studio similarly to the 3D printing workshops. This will provide students with an in-depth overview of the soldering required specifically for this project.

3.3 Assembly

The assembly procedures for 3D printed and laser cut systems is very similar for the most part, with the laser cut systems requiring only a few additional steps and considerations. If opting for a laser cut system, be sure to have a complete set of pieces before starting any of the assembly procedure. **(Figure 3.1)** The key difference in assembly between laser cut and 3D printed systems is in the towers' mounting hardware. Laser cut systems require that the hardware on the left tower piece be mounted **before** gluing the pieces together. **(Figure 3.2)** This step is crucial because it is nearly impossible to adjust the mounting hardware after the pieces are glued in place.

3.3.1 Tools and Preparation

Before beginning the assembly of the T-RECS, a few tools are required:

- Hex Key/Allen Wrench/Torx Bit for M2 screws: 1.5mm, 1/16in, or T6 (Required)
- Hex Key/Allen Wrench for M3 screws: 2.5mm (Required)
- #2 Phillips Head Screwdriver (Required)
- Needle-Nose Pliers (Recommended)
- X-ACTO Knife or Similar Utility Cutting Tool (Recommended for laser cut systems)
- Sand Paper (Recommended for laser cut systems)

These specialty drivers can be ordered online or found at the Invention Studio. It is also recommended that students verify the proper fit of the components with the mechanical parts before beginning assembly. Check that the shafts possess a very secure fit within the arm and sensor and a satisfactory fit within the ball bearings. Additionally, check that the bearings also fit comfortably within both towers. Finally, verify that the barrel jack fits appropriately into its mounting hole in the base. If any of these components do not fit due to the mounting holes being too small/tight, a file or sandpaper can be used to expand the hole and improve the fit.

3.3.2 Laser Cut Assembly

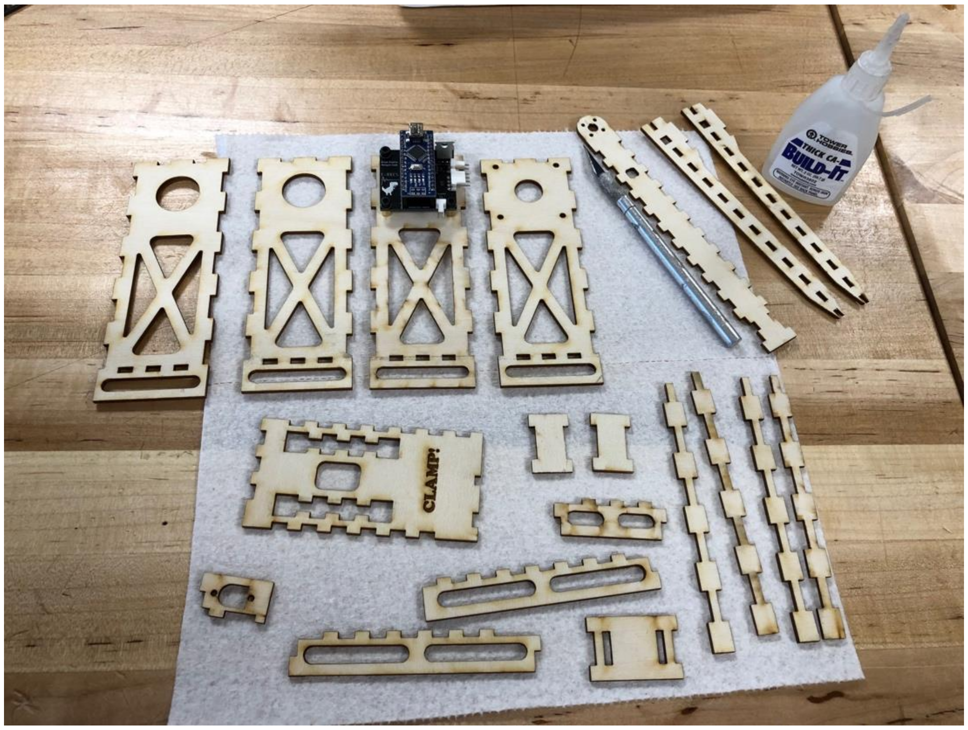


Figure 3.1. Complete set of laser cut parts.

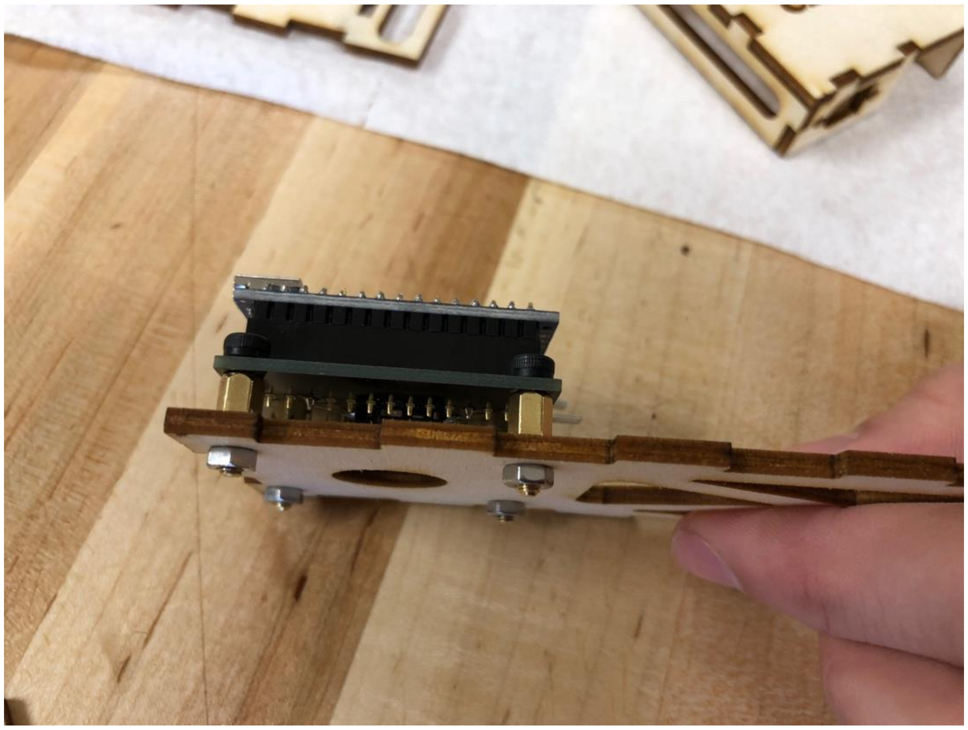


Figure 3.2. Mounting hardware attached to a tower piece with a smaller central hole. Note that the board does not need to be mounted yet, even though it is pictured here.

1. Assemble the base, paying attention to which side has the barrel jack and room for the rocker switch. When mounting the rocker switch later, it may be prudent to use glue to hold it in place. A clamp and a pressed fit may be sufficient, but if the rocker switch becomes too difficult to mount, then remove material as necessary with an X-ACTO knife to make room for it. **(Figures 3.3-5)**

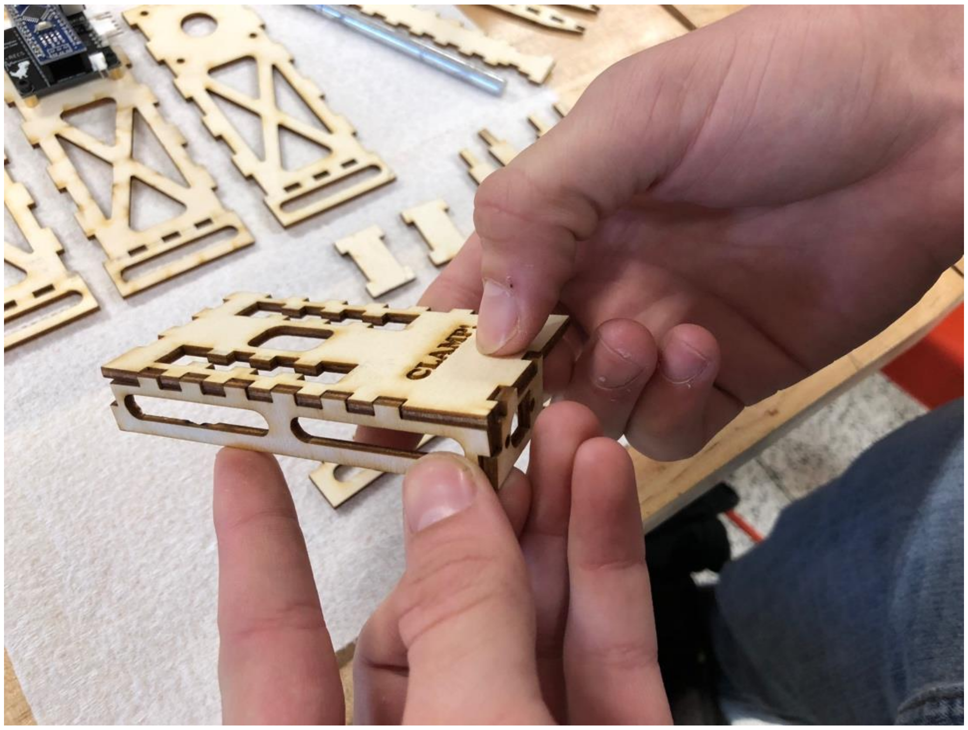


Figure 3.3. Start of base assembly.

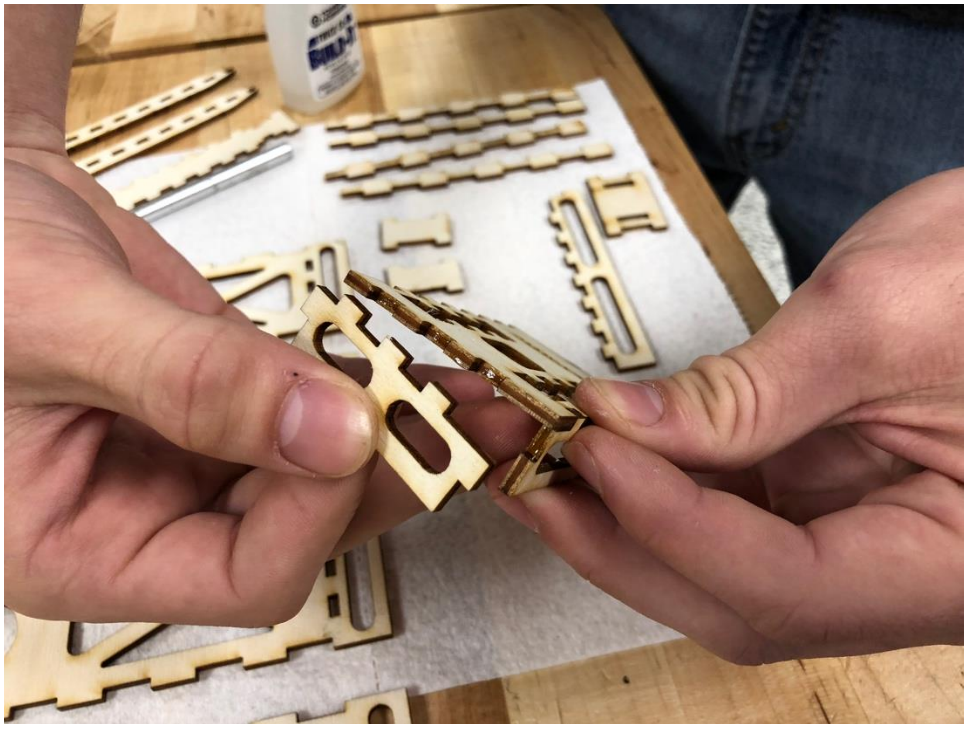


Figure 3.4. Front piece of base.

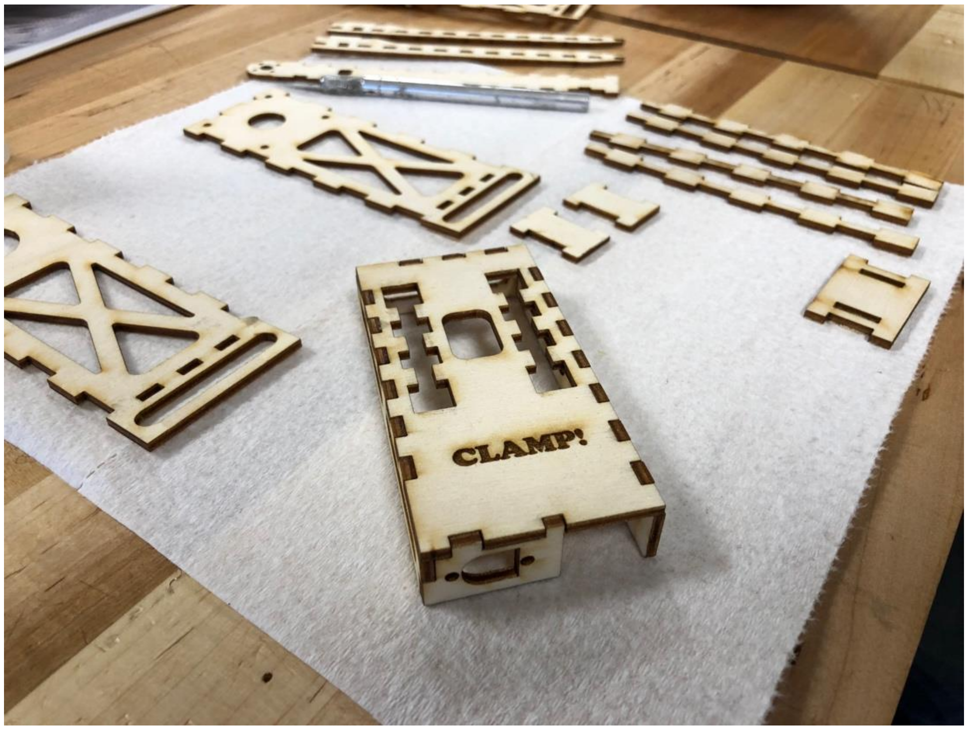


Figure 3.5. Completed base.

2. Assemble the towers by inserting their pieces into the corresponding holes in the base and pressing them against a side, making sure that the tower pieces with larger holes are located medially and the pieces with smaller holes are located laterally. These holes

hold the ball bearings in place. Once again, make sure the tower with the mounting hardware is on the left side with the standoffs facing outward. Use the longer thin pieces to hold them in place. **(Figures 3.6 & 3.7)**

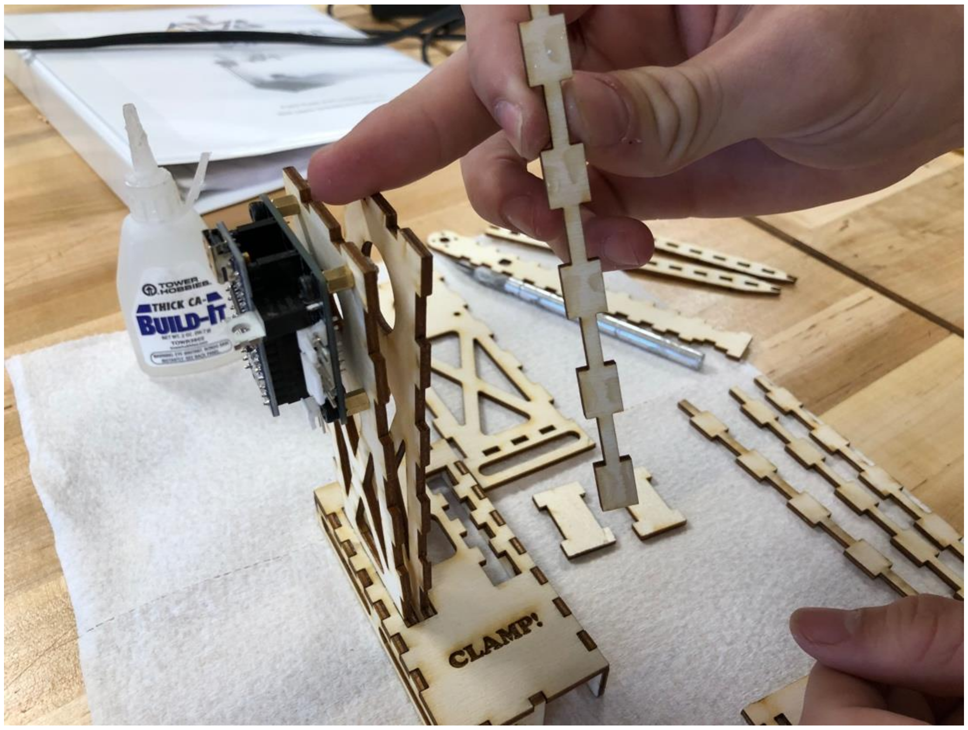


Figure 3.6. Beginning of tower assembly. Once the thin piece is glued, it will be impossible to adjust the mounting hardware, so make sure it is completely secure at this point.

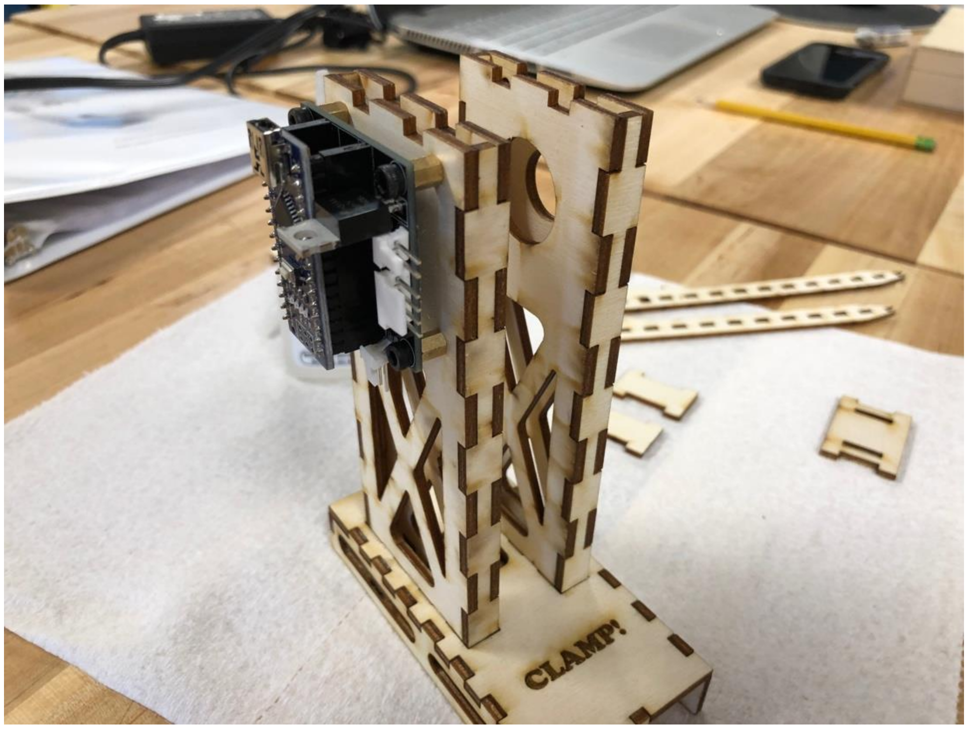


Figure 3.7. Completed towers.

3. The arm is made up of four pieces: a long piece for mounting the motor, two side pieces to stabilize, and a cover for the ESC. Be sure to stabilize the towers with the final piece on top after mounting the arm. **(Figures 3.8-10)**

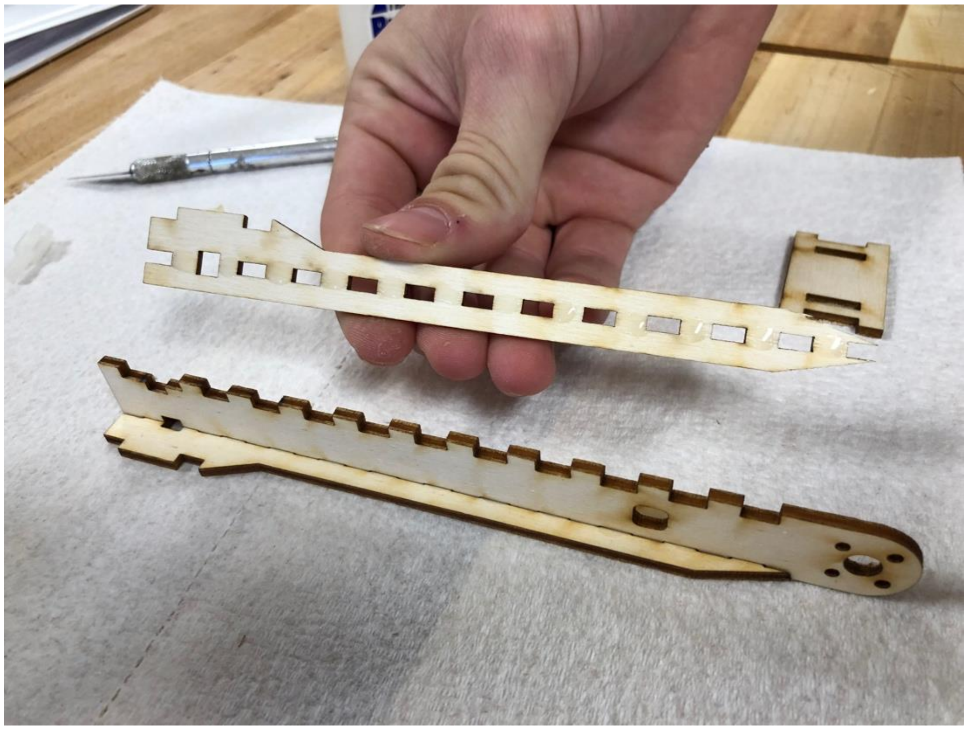


Figure 3.8. Start of arm assembly.

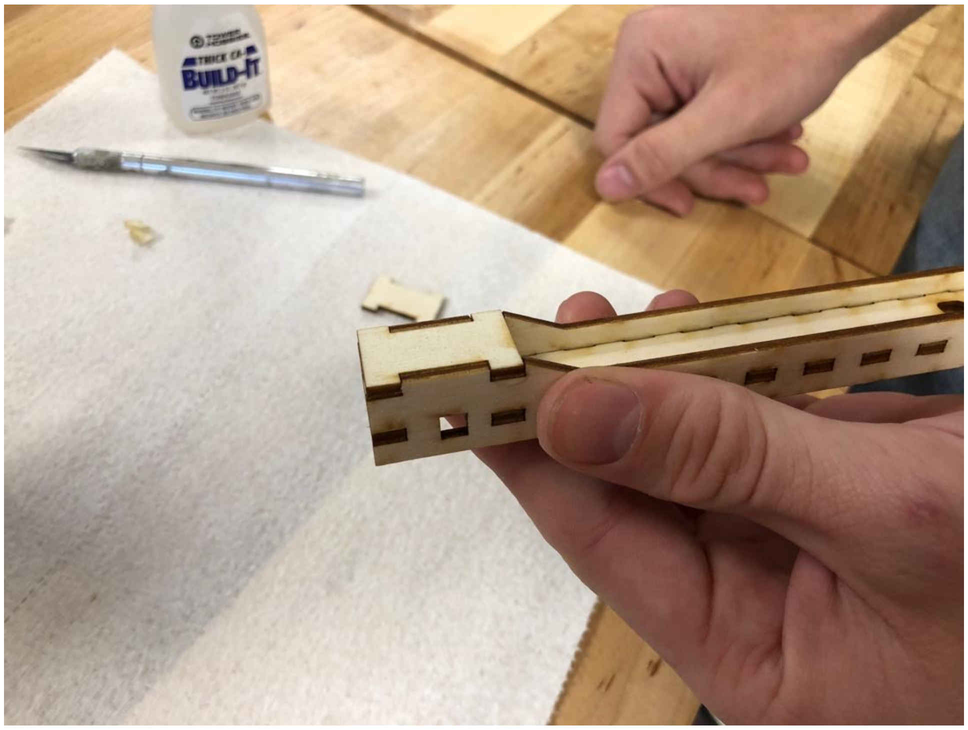


Figure 3.9. ESC cover.

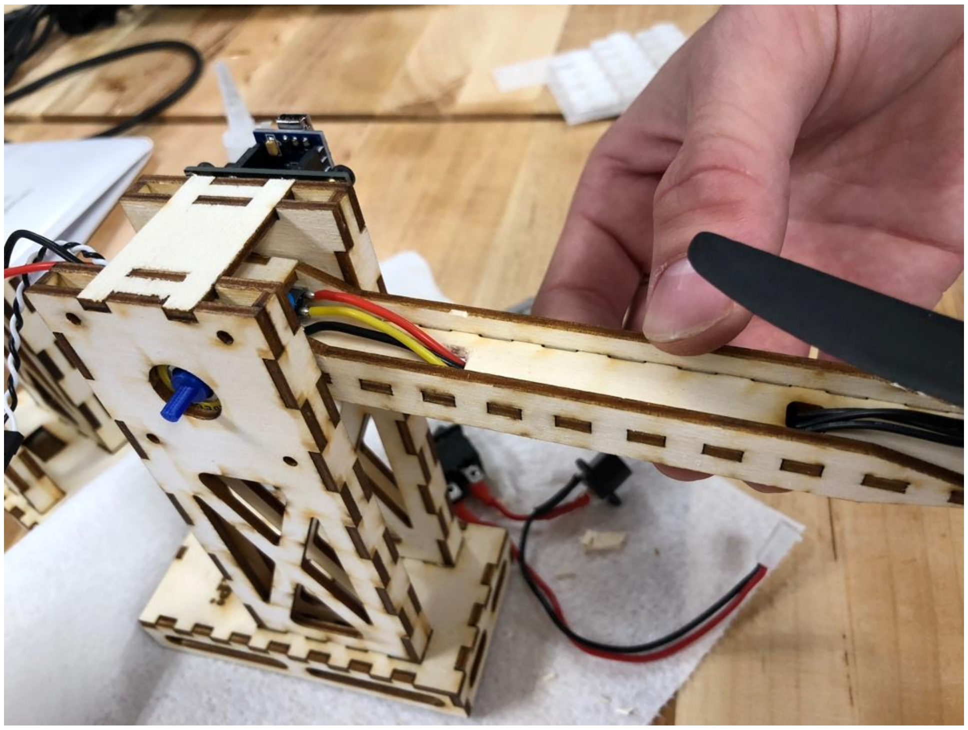


Figure 3.10. Completed laser cut system with tower stabilizer. When mounting the ESC and motor, be sure to pass the wires through the slots to avoid interfering with the propeller.

3.3.3 Base and Wiring Harness

1. Insert the entire wiring harness into the base through the rectangular hole for the rocker switch so that the connector and barrel jack are inside the base and the rocker switch is still sticking out. **(Figure 3.11)**
2. Press the barrel jack into its slot and secure it with the two provided long M2 screws and nuts. Since the barrel jack only needs to be mounted with a finger-tight hold (no screwdriver required), it is easier to access the nuts if the screws are oriented such that the nuts are on the outside. **(Figure 3.12)**
3. Press the switch into its mounting hole with the “OFF” side facing outward until it clicks securely into place.
4. Pull the PCB connector through the central hole in the base so that it sticks out the top.

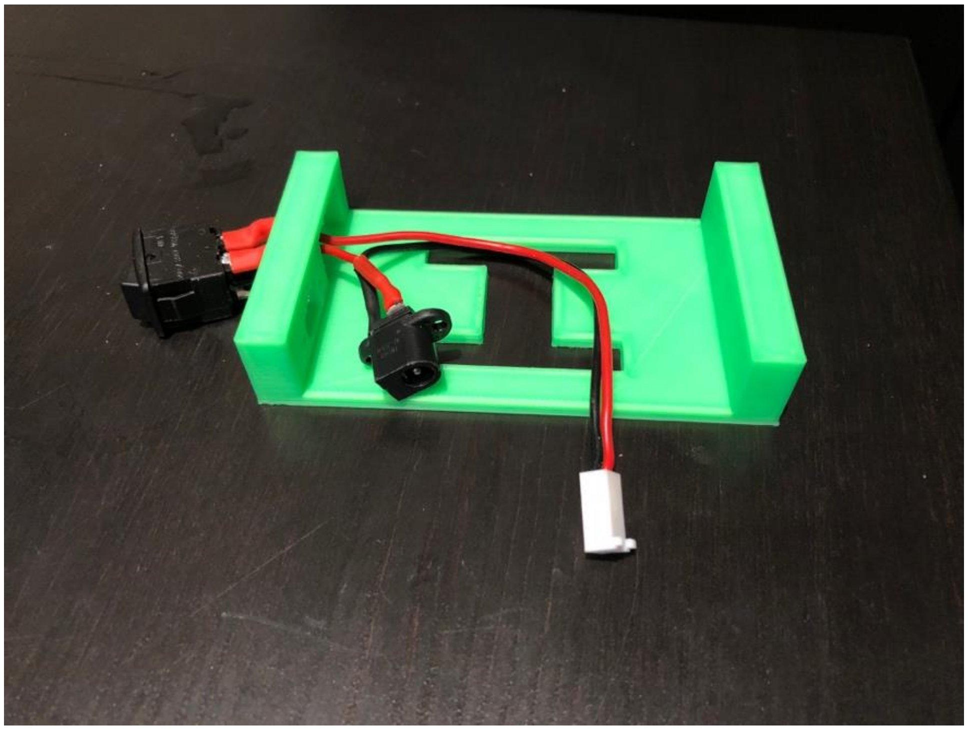


Figure 3.11. Inserting wiring harness into base (Step 1).

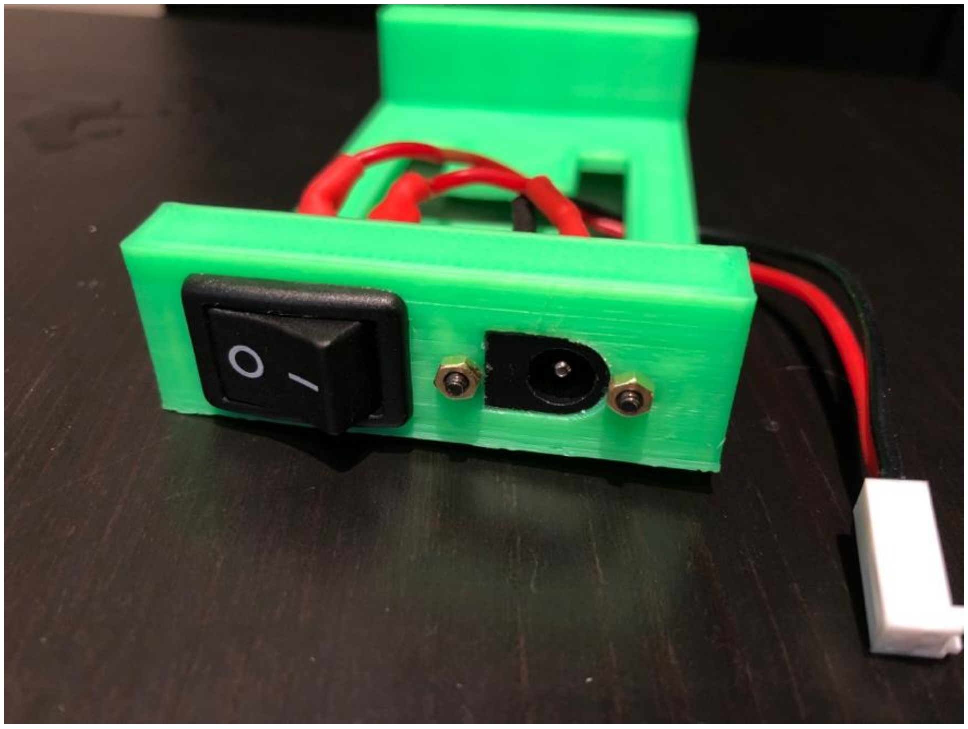


Figure 3.12. Mounted rocker switch and barrel jack (Steps 2-4).

3.3.4 Towers and Standoffs

NOTE: This manual uses the red and orange tower as the left tower and the green tower as the right tower. In this orientation, the arm extends out the front with the power switch and barrel jack on the back.

5. Insert the left tower into its designated slot from beneath the base, being careful not to pinch the wires of the wiring harness in the process.
6. Mount the four M3 standoffs to the left tower using the provided nuts and hand tighten them. **(Figure 3.13)**
7. Insert the right tower into its designated slot beneath the base, again watching for the wiring harness so that no wires are pinched.
8. Press fit a ball bearing into the appropriate slot of each tower. The fit does not need to be particularly tight, so long as the bearing isn't so loose that it can rattle around.
9. Pass long M4 bolt through designated holes in both overtravel stops on the towers and secure with the corresponding nut on the other side. This connection should be relatively tight since it greatly reduces nonlinearities and oscillations in the response of the system. **(Figure 3.14)**

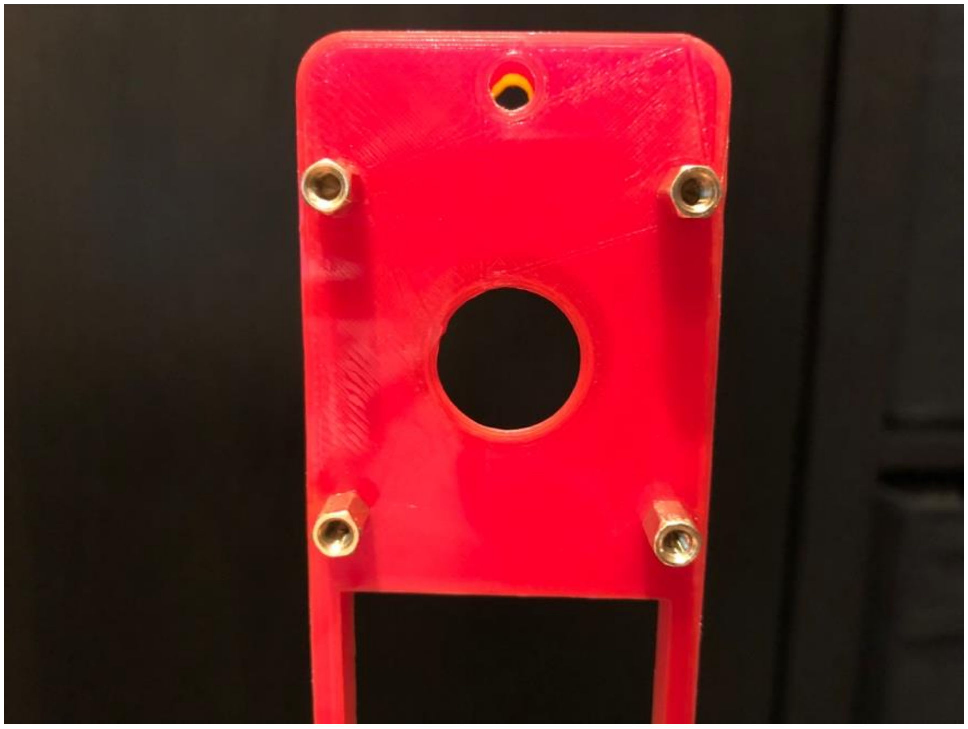


Figure 3.13. Mounting standoffs for PCB on left tower (Steps 5 & 6).

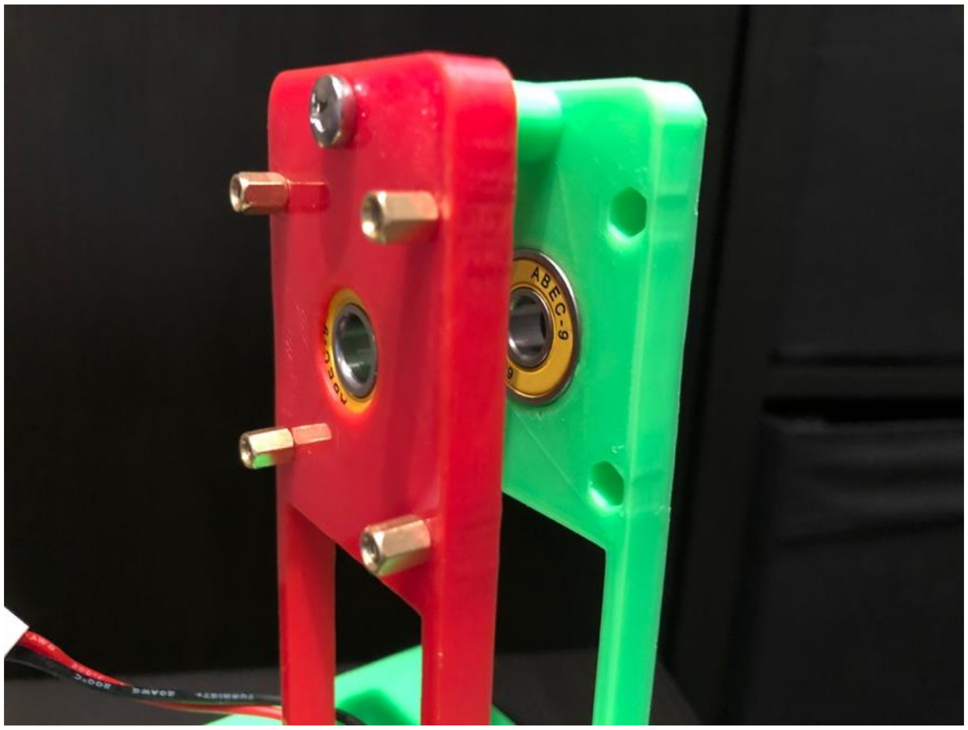


Figure 3.14. Completed tower assembly with secured overtravel stops, PCB standoffs, and ball bearings (Steps 6-9).

3.3.5 Arm, ESC, and Motor

NOTE: This manual uses a counterclockwise motor (silver nose) with a 5030 propeller.

10. Connect the three wires of the motor to the corresponding three wires of the ESC. Do not worry about which motor wire corresponds to which ESC wire; this will be determined later.
11. Using the four short M2 screws, mount the motor to the arm with its three wires facing to the right. Be sure the screws are tightly fastened because vibrations from the motor can gradually loosen screws that have not be properly fastened over time. However, do not tighten the screws to the point that the heads start digging into the plastic.
12. Wrap the extra wire counterclockwise around the arm so that no loose wires hang from the arm.
13. Slide the ESC into the designated slot in the arm. The fit should be tight enough that the ESC is not at risk of coming loose while the arm swings up and down. If it is, wrap a layer or two of scotch or electrical tape around the ESC to thicken it sufficiently for a better fit. **(Figure 3.15)**
14. Mount propeller on motor by removing propeller cap, pressing propeller onto motor shaft, and very tightly re-securing propeller cap. Like the motor mounting screws, motor vibrations can gradually loosen improperly fastened propeller caps leading to a safety hazard.

SAFTEY NOTE: Clockwise spinning motors (black noses) have reverse-threaded caps (lefty-tighty, righty-loosey) because the torque from the clockwise rotation would loosen normally threaded caps. Counterclockwise spinning motors have silver noses and tighten normally. Make sure the propellers match the motor. The easiest way to verify this is to check if the number on the propeller blade is followed by an ‘R’ (e.g. “5030” vs. “5030R”). The ‘R’ stands for reverse threaded and corresponds to clockwise (black nose) motors. Incorrectly paired motors and propellers can lead to the nose and propellers detaching mid-flight, risking personal injury and damage to the motor. **(Figure 3.16)**

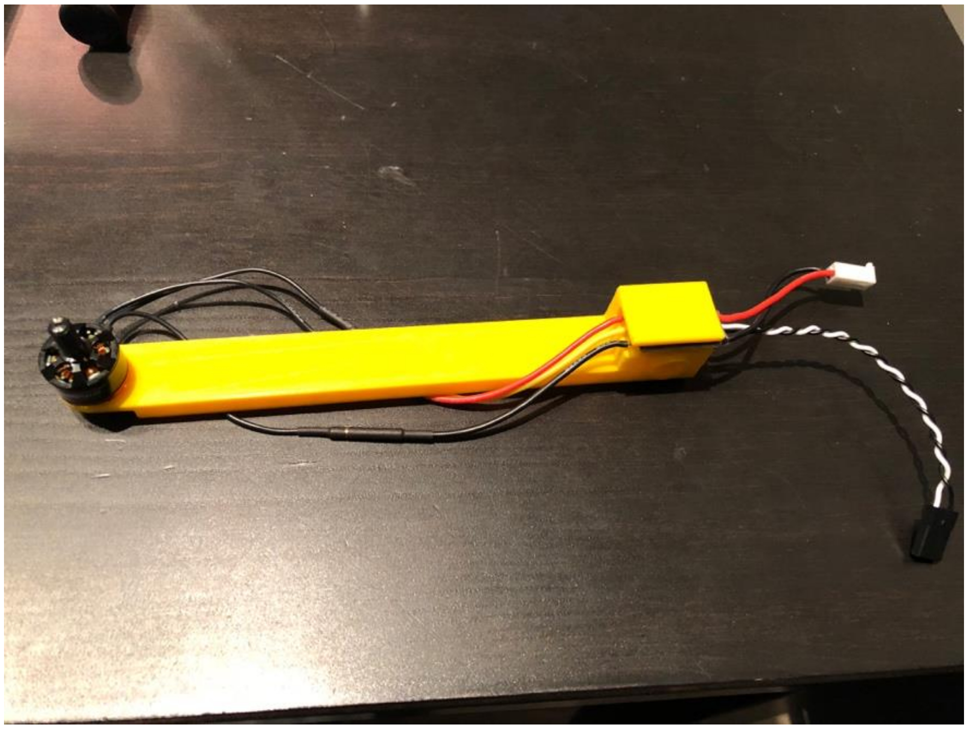


Figure 3.15. Motor and ESC mounted to system arm (Steps 10-13).

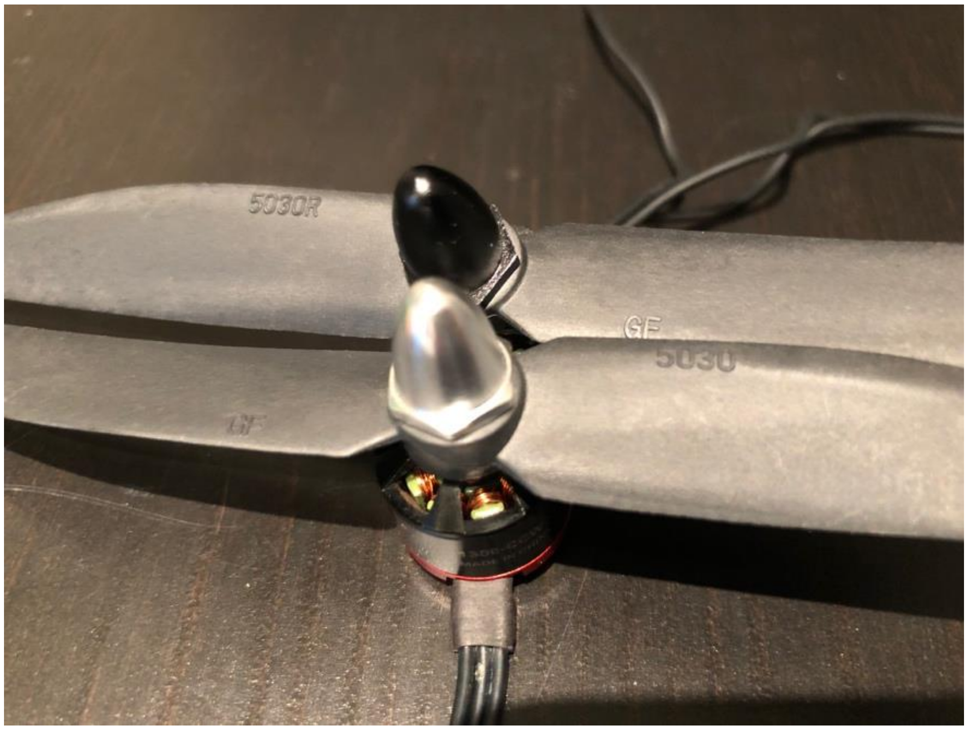


Figure 3.16. Example of motor orientations with corresponding propellers (Step 14).

3.3.6 Shaft and PCB

15. Place the completed arm assembly between the two towers with the arm's square hole aligned with the bearings' holes on the towers. It may be necessary to loosen the overtravel bolt at the top to fit the arm in comfortably but be sure to re-tighten the bolt once the assembly has been secured. The tighter the bolt, the more natural damping the system will have. The bolt also removes nonlinearity from lateral oscillations.
16. Insert a shaft through both bearings and into the corresponding square holes in the arm. The shaft should be oriented such that the flat part of the semicircular dowel is facing up when the arm is held horizontal. **(Figure 3.17)**
17. Mount the fully soldered PCB (without the Arduino connected) onto the mounting standoffs, using the four provided M3 screws. Be sure to properly align and insert the dowel into the rotary position sensor before fastening the board with the screws.
18. Add the Arduino to the PCB with the USB port facing upward.
19. Finally, connect all the appropriate wires to the PCB's connectors. Be sure that the ESC's white wire is aligned with the top pin of the ESC connector on the board. **(Figure 3.18)**

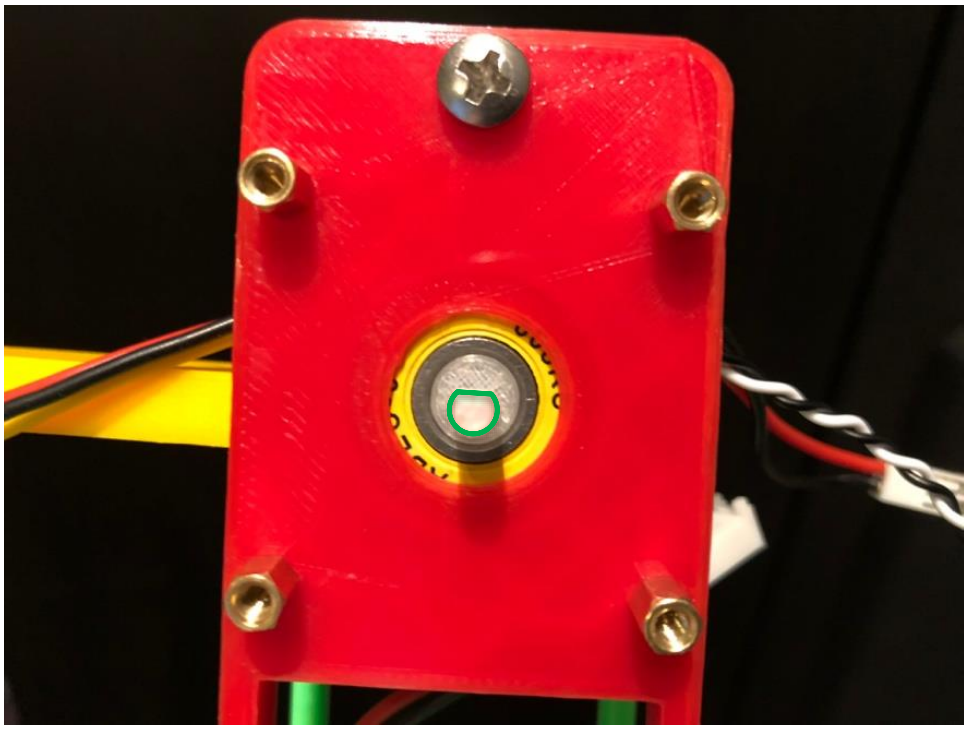


Figure 3.17. Shaft inserted through arm and ball bearing as viewed from the left tower with flat portion of shaft facing up (Steps 15 & 16).

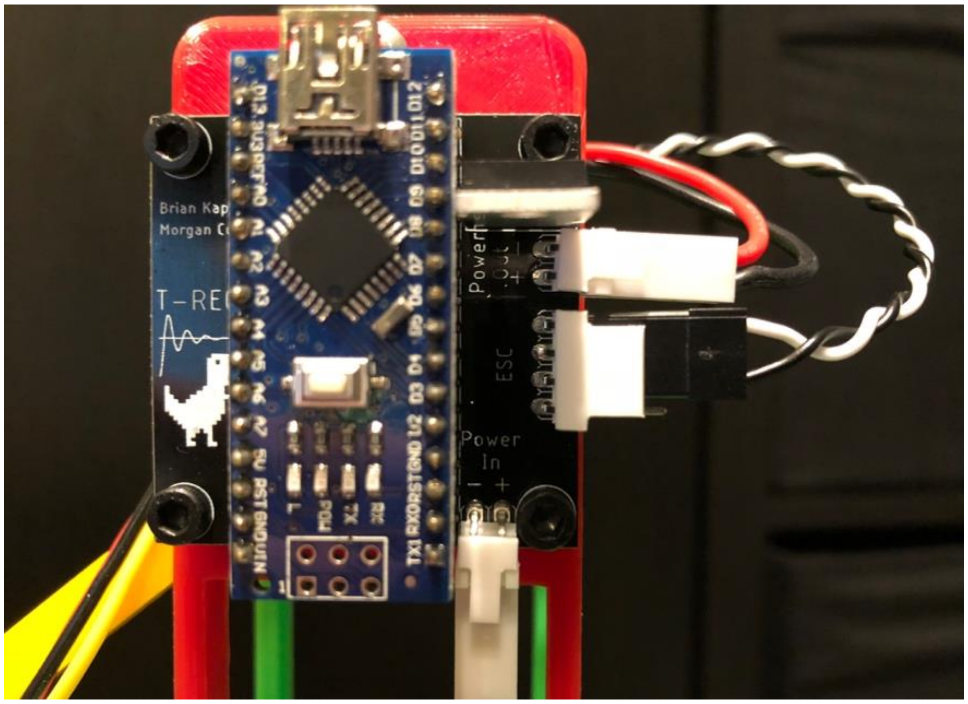


Figure 3.18. Fully wired PCB mounted on left tower (Steps 17-19).

4. Testing & Operation

4.1 Arduino IDE

The Arduino IDE is the development environment used for writing Arduino programs, which are often referred to as “sketches,” and uploading them to Arduino boards. The Arduino IDE is available as a desktop application for Windows, Mac OS, and Linux operating systems.

4.1.1 Installation & Setup

The latest version of the Arduino IDE can be downloaded and installed at the following URL: https://www.arduino.cc/en/Main/Software. After downloading the appropriate installer, follow the onscreen instructions to complete the installation of the Arduino IDE. This process should be quick and straightforward. Launch the Arduino IDE once installed and plug in the Arduino to one of the computer’s USB ports using the provided cable to begin configuring the proper settings for the T-RECS.

From the “Tools” dropdown menu, navigate to “Board” and select “Arduino Nano.” From that same dropdown menu, select “ATmega328P (Old Bootloader)” as the option for “Processor” and “AVRISP mkII” as the option for “Programmer.” For the “Port” option, the selection will vary depending on the computer and operating system the Arduino IDE is running on. For Windows systems, this will usually be the word “COM” followed by a number (i.e. “COM3” or “COM5”).

For Mac OS, this option is usually more complicated and varied, but it should look something like “/dev/tty.usbserial” or “/dev/tty.usbmodem.”

If the correct serial port doesn’t show up, then a driver needs to be installed to allow the computer to communicate with the serial chip onboard the Arduino. The driver can be found at the following URLs:

- http://www.wch.cn/download/CH341SER_EXE.html (Windows)
- http://www.wch.cn/download/CH341SER_MAC_ZIP.html (Mac OS)
- http://www.wch.cn/download/CH341SER_LINUX_ZIP.html (Linux)

Note that the website will be in Chinese (so Google Chrome’s translate feature may come in handy). This is because the Arduino’s CH340 serial chip is an exclusively Chinese product (which is what allows these Arduinos to sell for less than 4 dollars); however, the drivers on this website are perfectly safe, so there is no need for concern.

Lastly, navigate to “Sketch” > “Include Library” > “Manage Libraries…” which should open the Arduino Library Manager. In the search bar at the top, enter “MsTimer2.” From the list of search results, select and install the “MsTimer2” library by Javier Valencia. This library allows the Arduino to perform precisely timed updates that the T-RECS requires for its controller.

4.1.2 Using the Arduino IDE

There are a few key features of the Arduino IDE to be made aware of. The first is opening and editing sketches. Navigate to “File” > “New” to start a new sketch. Notice that this new blank sketch (and all other Arduino sketches, for that matter) is divided into two functions: `setup()` and `loop()`.

As the name implies, `setup()` is used for setting up the rest of the software to run properly, and therefor this function is called one time when the Arduino first boots up. This function should include preparing external hardware (such as the ESC) for operation, initializing serial communication with the computer, configuring any of the physical pins of the Arduino for their intended function, and so on.

As its name also implies, `loop()` is intended to run continuously in an infinite loop after `setup()` has first been called. This is where the main body of an Arduino program should lie because this is the code the Arduino will be executing for most of its operation.

Once a sketch is ready to be transferred to an Arduino for testing, simply navigate to “Sketch” > “Upload.” This causes the Arduino IDE to compile (or “verify”) the given sketch and (assuming successful compilation) uploads the compiled sketch to the Arduino. The Arduino will then immediately begin executing the sketch once it has been successfully uploaded.

The final important feature of the Arduino IDE that will be used for the T-RECS is the Serial Monitor. This feature is the Arduino IDE’s main method for allowing a user to interface directly with an Arduino board. To open the Serial Monitor, be sure an Arduino is connected to the computer and then navigate to "Tools" > "Serial Monitor." This prompts the Arduino IDE to open a new window with a text box along the top, a few settings along the bottom, and a large blank pane taking up most of the window. The Arduino can print information to this pane for users to view and users may send data, commands, etc. to the Arduino via the text box at the top, allowing for two-way communication.

4.2 SerialPlot

SerialPlot is an open-source program developed for plotting and logging data sent over a serial connection (like the one the Arduino uses). This software allows students to see a live plot of the response and operation of the T-RECS as well as record this information in log files for later analysis.

4.2.1 Installation & Setup

The latest version of SerialPlot for Windows can be found at the following URL: https://bitbucket.org/hyOzd/serialplot/downloads/. Download and run the latest install file which (at the time of writing) is serialplot-0.10.0-win64.exe. Select all the default installation options in the installer and proceed with the installation.

For Linux users, simply enter the following commands into the Terminal to install SerialPlot:

```
sudo add-apt-repository ppa:hyozd/serialplot
sudo apt update
sudo apt install serialplot
```

Launch SerialPlot once it has been successfully installed and navigate to "File" > "Load Settings" and select the provided "T-RECS Settings.ini" file. This settings file configures the SerialPlot software for communication with the T-RECS.

4.2.2 Using SerialPlot

To establish a serial connection to the Arduino in SerialPlot, select the proper serial port at the top of the application window or in the "Port" tab under the plot. This port name should be the same as the one used in the Arduino IDE. Then select "Open" to start the connection.

Once connected, navigate to the "Commands" tab under the plot. This should display a series of commands that can be sent to the T-RECS to control it during operation. These commands can be altered as needed and new commands may be added if students choose to modify or expand upon the T-RECS software.

Next navigate to the "Record" tab. This tab allows the data output by the T-RECS to be logged in a CSV file which can later be view and manipulated in Excel or MATLAB for later analysis. Simply select "Browse" to designate a filename and location for the CSV log file. Do not forget to manually add the ".csv" extension to the end of the filename as SerialPlot does not do this automatically.

Feel free to explore and experiment with the other tabs beneath the plot to customize SerialPlot as desired. Reloading “T-RECS Settings.ini” as described in the previous section is a quick way to restore the software to a known working configuration in case something goes wrong.

Finally, at any point during the use of SerialPlot, the “Take Snapshot” button at the top of the window may be pressed. This captures the data that was plotted when the button was pressed for later use. These snapshots may be viewed by navigating to “Snapshots” > “Snapshot [XX:XX:XX]” where the ‘X’s will be replaced by the timestamp of the desired snapshot. This will open a plot of the data from when the snapshot was captured in a new window. There are also a few options to explore for modifying the plot’s appearing under the “View” dropdown menu. The snapshot can also be saved as a CSV file by navigating to “Snapshot” > “Save as CSV.”

4.3 Testing Functionality

Before jumping into the PID controller code that normally operates the T-RECS, a few diagnostic tests and calibrations must be performed. These tests are implemented across a few simple Arduino sketches that are provided alongside the PID controller code.

4.3.1 Sensor Test

From the Arduino IDE, open the Sensor_Test.ino sketch and upload it to the T-RECS. As its name implies, this sketch verifies that the sensor is connected and operating correctly. After uploading, open the Arduino serial monitor and make sure the Baud rate is set to 115200. The Arduino should begin outputting a stream of raw sensor readings paired with the corresponding angle measurements based on the position of the arm. Move the arm up and down to verify that the sensor readings accurately correspond to the angular position of the arm. If the Arduino is not making accurate angle measurements, please see Morgan or Brian for assistance.

NOTE: An angle reading of zero should correspond to the arm being horizontal and positive readings correspond to the arm being above horizontal and vice versa.

4.4.2 Data Logging Test

Open the Logging_Test.ino sketch in the Arduino IDE and upload it to the T-RECS. Now open SerialPlot and connect to the Arduino as described in Section 4.2.2. SerialPlot should begin plotting a set of four overlapping sinusoids. Test the command functionality of SerialPlot by sending any of the available commands to the T-RECS, which should pause the plotting. Sending any other command should resume the plotting. Next, test the snapshot functionality of SerialPlot by taking a snapshot of the sinusoids and then try to view it afterward. Finally, test the recording functionality of SerialPlot by capturing the plotted output as a CSV file. Then open the saved file in Excel or a similar program to ensure the data was properly logged.

4.3.3 ESC Test and Calibration

WARNING: From this point forward, the software that will run on the T-RECS can spin the propeller. The motor is capable of theoretical speeds greater than 37,000RPM and far lower RPMs are more than capable of injuring hands or fingers. **Always** securely fasten the T-RECS to a flat surface with a clamp or other mounting hardware and clear the area around the propeller of any obstructions (including hands and fingers) before powering on the system. <u>In the event of an emergency, the system can be immediately powered off by performing any of the following: unplugging the power adapter, toggling the rocker switch, or disconnecting the USB programming cable.</u>

The next test sketch is intended to verify that the Arduino can properly control the ESC (and therefore the motor) on the T-RECS. From the Arduino IDE, open the ESC_Test.ino sketch and upload it to the T-RECS. Then open the serial monitor with a Baud rate of 115200 as before. If this is the very first time the system is being used to control the ESC, a calibration signal must first be sent to properly set the ESC's throttle thresholds for reliable operation. To send a calibration signal, simply send the character 'c' to the Arduino over the serial monitor. This should trigger a series of beep tones that the ESC and motor will emit over a period of about 10-15 seconds. Once calibration is complete, the system should be ready to control the motor.

To control the ESC, simply begin entering numerical throttle commands into the serial monitor. The throttle values are on a scale of 0-1000. However, the system should never approach anywhere near the max throttle value of 1000, which is enough to fly the T-RECS off the workbench if not properly secured. Start with a throttle value of 50 and increase throttle in increments of 10-20 until it is clear the system is functioning properly. Students should be hesitant and cautious of driving the ESC past a throttle value of 300.

NOTE: If the propeller is spinning in the wrong direction, first disconnect the T-RECS from power. Then simply swap the connections of any two of the three wires between the ESC and the motor and try again.

4.3.4 PID Controller Test

If all the previous tests have succeeded, then the T-RECS is ready to run the PID controller software. Open the PID_Controller.ino sketch in the Arduino IDE and upload it to the T-RECS as before. The PID controller sketch can be operated from both the Arduino serial monitor and from SerialPlot. The controller responds to the following commands:

- 'p' – Pause the controller
- 'c' – Calibrate the ESC
- 't' – Tune the P, I, and D gains of the controller
- 'f' – Change the refresh frequency of the controller (careful changing this setting, as it can lead to jittery, unpredictable behavior)
- A numerical value can be sent to change the requested pitch (in degrees) of the T-RECS

As mentioned before, be very careful when operating the T-RECS. Keep clear of the propeller and be sure the system is securely fastened to the workbench **before** powering it on.