Autonomous Motorized Mount for the PATHS Sensor

Team Members:
Brandon Bogue
Marvin Hernandez
Quoc Pham

Teaching Assistant:
Yamuna Phal

Leading Professor:
Lara Waldrop

ECE 445
Group #23

Fall 2017

Table of Contents:

1 Introduction

1.1 Objectives

Existing commercially available motorized mounts for astronomical instruments are designed to be directly used by a human, either by going to a point keyed in by the human, or tracking an object specified by a human. In most use cases, this is sufficient, as the primary market is amateur astronomers who want to personally look at specific objects in space. This is insufficient for some types of tasks, such as observing in a set pattern for multiple nights, moving in a custom path, or functioning in concert with other instruments.

In addition, most astronomers would use an equatorial mount to track celestial objects. The issue with an equatorial mount is that, outside of celestial objects, their movements can be difficult to calibrate between arbitrary spherical coordinates since it was built to track objects moving along a diurnal motion. Most motorized equatorial mounts would also move at a set pace compensating for the earth's movement and offer no actual control over the speed for the user. For this project, the mount will be a manual altazimuth mount which we will attach stepper motors, sensors, and a microcontroller to create the control scheme behind the autonomous, custom paths required to track the object of interest, which will be located in the upper atmosphere.

Our objective is to create an automatable motorized mount for an existing sensor, to be used by the PATHS sensor project. This mount will be a smaller component of an external tracking system. It will be able to position the sensor based on input angles and move between two locations in the sky over a given period. The angles fed by the external computer relies on a camera attached to the PATHS sensor. Through a series of image processing techniques, this external system will calibrate the mount to an absolute reference using the North American Nebula, and feed the corresponding attitude and the azimuthal angle to calibrate to our mount.

1.2 Background

One of the current challenges in atmospheric research has been the ability to study atmospheric density, spatial distribution, and temporal variability. Therefore, the PATHS project has been proposed to implement remote sensing and controls theory to capture the airglow emission of Hydrogen for computing its density. This project will allow to overcome single-line-of-sight viewing geometry by allowing multi-angle viewing through a common volume to enable tomographic formulation for solving an inverse problem which will yield accurate H density.

The PATHS instrument used in this project is a novel ground-based photometer that will capture the brightness of H airglow along multiple lines-of-sight in an array configuration. The sensors used in the array are ~ 20 x 40 cm binocular optical assemblies, one of which must move over the course of the night.

The PATHS sensor needs to be pointed very precisely, within a fraction of a degree. It also needs to act semi-autonomously given a set of spherical coordinates, as it is grossly impractical to have a human attempt to directly control it.

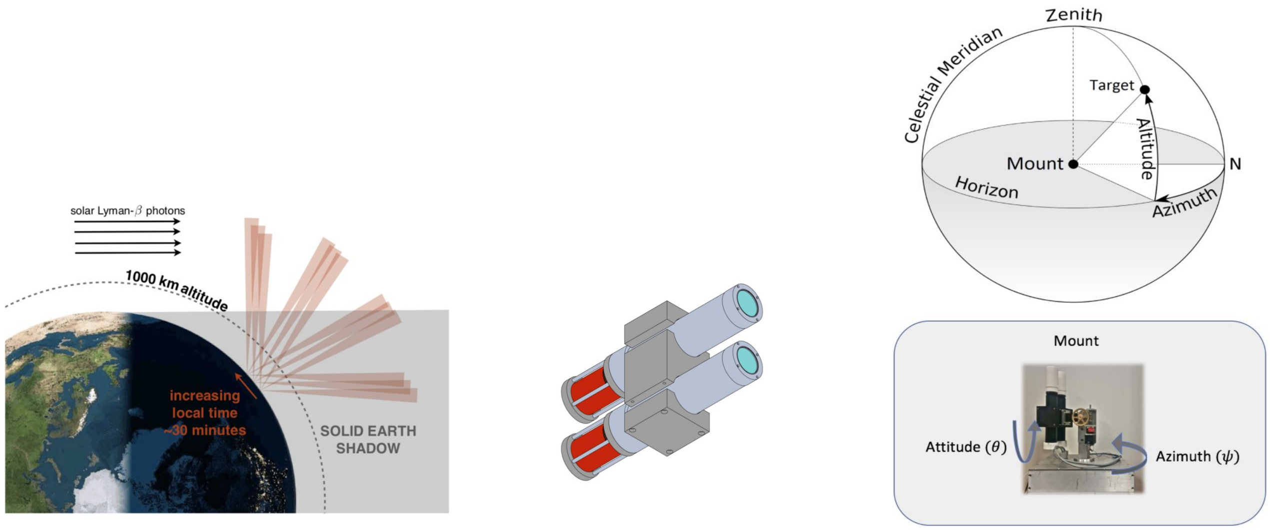


Figure 1: (Left) PATHS viewing geometry. (Center) Schematic of PATHS sensor. (Right) Mount and described axes of rotation

1.3 Performance Requirements

In order to provide a reliable and robust mount for the research, we had to meet the following objectives.

1) The mount needed to hold and move the sensor along the 2 axes of rotation.
2) It must offer a smooth tracking for best results in the measurements.
3) It must be accurate to within 0.01 degrees
4) Communicate with a external system to receive coordinates
5) Implement a closed-loop feedback control.

1.4 High Level Modular Design

Our block diagram is divided into three main modules: Mount, Autonomous Assembly, and the Power Supply. The mount module represents the physical assembly where the PATH sensor will be mount. Also, this will provide a housing for our autonomous assembly as well as the power supply. The autonomous assembly module describes the high level electronic circuit performing the control. Internally, this module communicates with the other submodules components for feedback, reference tracking, and control. The power supply is the third main module. This is important because it will distribute, control, and regulate the power distribution among the different components in the autonomous assembly. The three modules are separate but interface among each other to fully operate and satisfy our design.

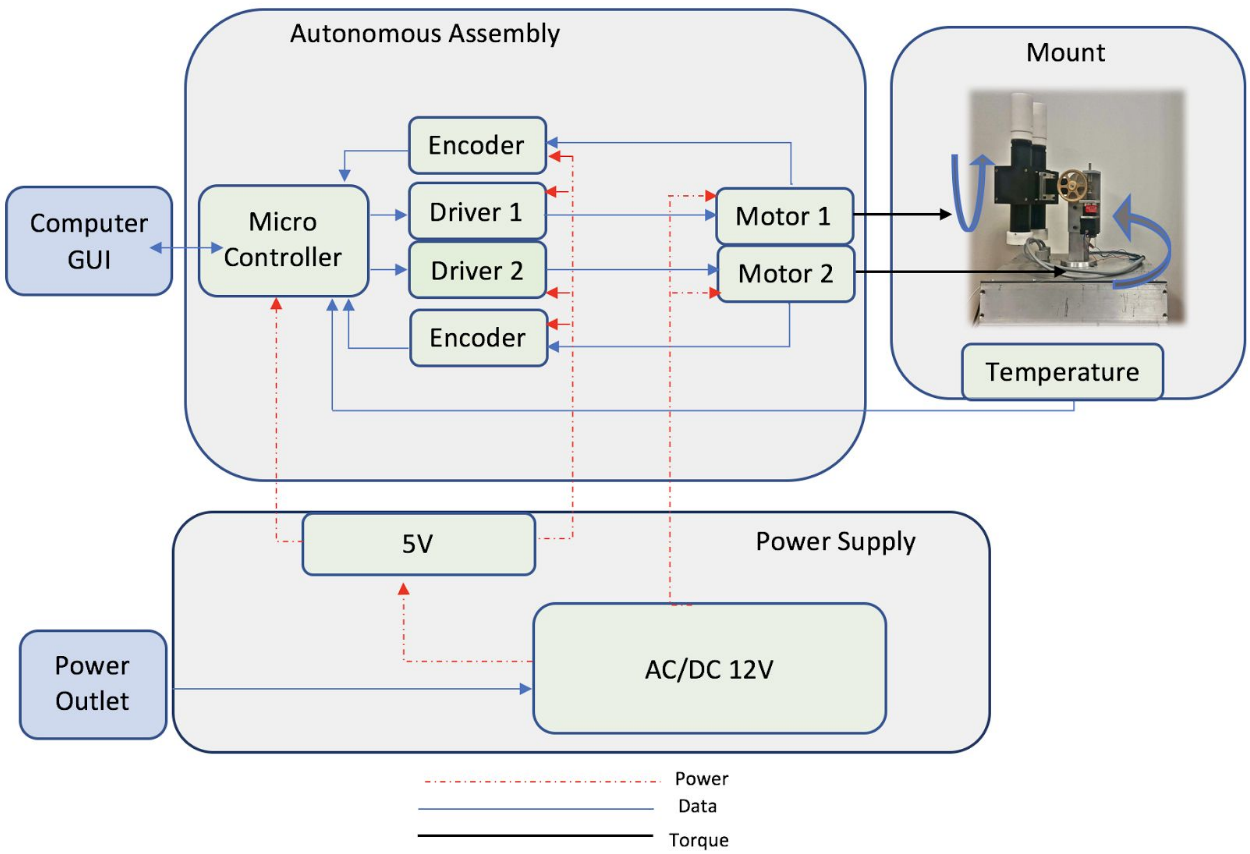


Figure 1: High Level Block Diagram

1.5 Block Level Changes

The power supply initially was composed of an AC/DC converter and 3 set of voltage regulators; 2X 12V and 1.7 Amp which will provide power to the motors, and 1 for 5V 500mA supplying the microcontroller and drivers. The power supply must output 12V ±0.1V DC with maximum current of 6A to drive the entire circuit. The calculations for this are described in detailed in section 2.5. The calculations show that due to the high current needed to drive the high torque stepper motors, the power supply introduced a bulkiness and complexity that would take us apart from the project main aspects and possibly be a safety concern. Therefore, we decided to utilize an off the shelf power supply meeting the requirements.

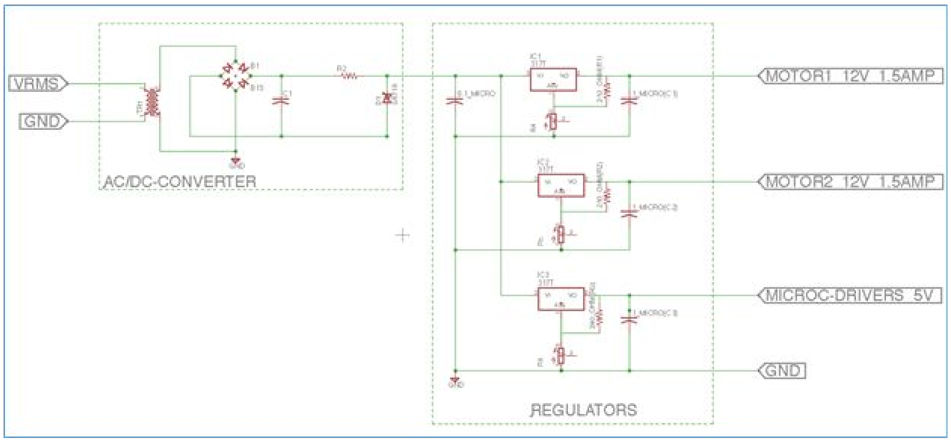


Figure 5. Initial Power Supply Schematic Design.

2 Design

2.1 Design Procedure

The specifics of our design components began with meeting the two requirements, an accuracy of 0.01 degrees and being able to hold and move a 4kg mounted sensor. We decided on a stepper motor for its precision and holding capability. The selection of our final stepper motor meant that the driver should be able to deliver a max of 3A at 12V, the encoder should have a minimum resolution of 2048 pulse per revolution, and the power supply should be able to deliver a maximum of 6A at 12V for both motors. The design must also communicate with an external tracking system via USB so our microcontrollers needed to have the right number of interrupt pins, analog input pins, and digital logic pins.

2.2 Overall System Schematic

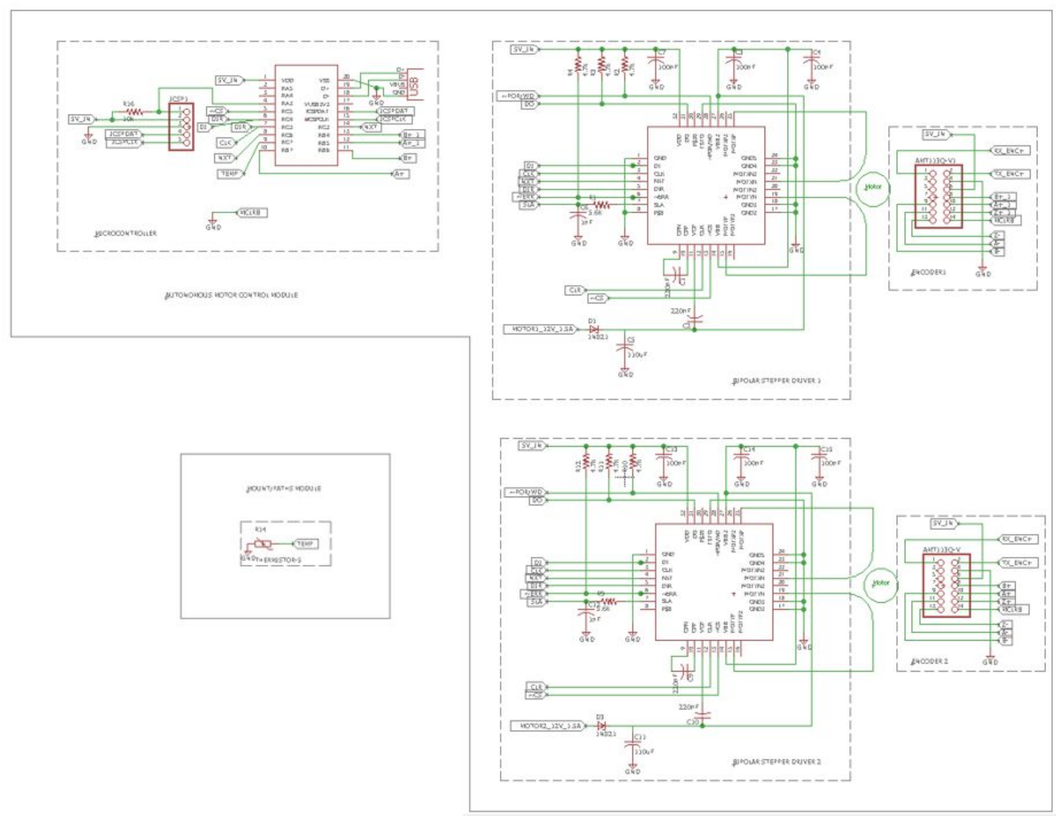


Figure 2. Schematic diagram of Motor Control Module, Power Supply Module, Microcontroller Module, and the Mount Module from the block diagram in Figure 1.

2.3 Description of Blocks

2.3.1 Microcontroller Module

The microcontroller is responsible for configuring the motor drivers, providing appropriate control values for the motor drivers, tracking motor movement via the encoders, and interfacing with an external device via USB to receive commands. We plan to use the PIC16F1459 microcontroller for this task due to its ability to directly communicate via USB, as well as having sufficient interrupt pins to accommodate our design. The software for this microcontroller is detailed in section 2.4.

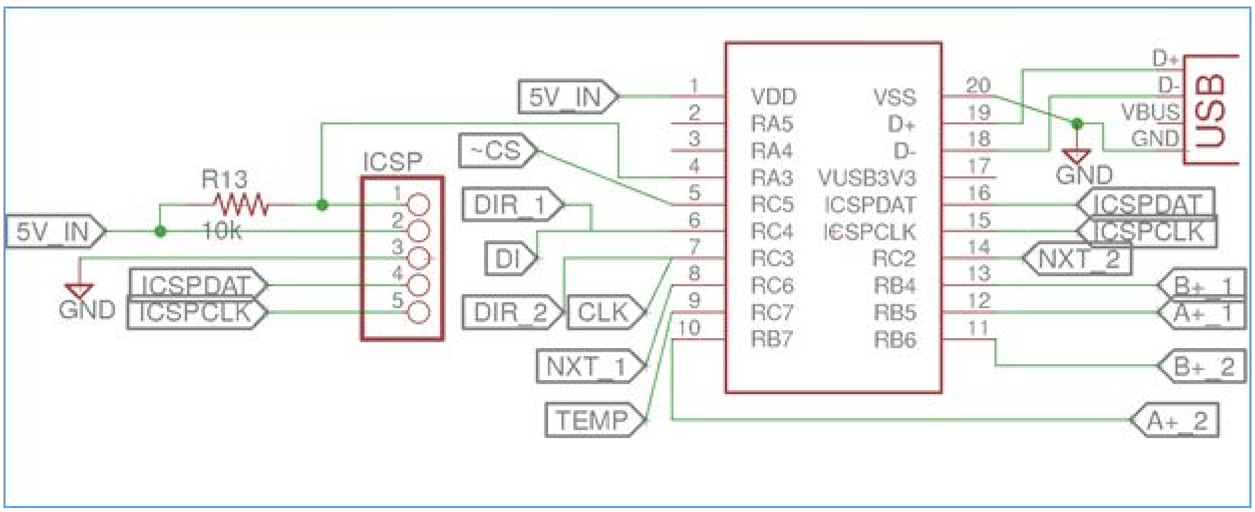


Figure 3: Microcontroller Schematic

In figure 3, one should note that since there are two separate identical motor circuits, many signals are replicated. Where that is the case, the signal is numbered by the motor circuit it corresponds to. It is also worth noting that some pins are shared between multiple signals. This is intended, and is due to technical limitations, but should not produce adverse behaviour, as long as the software does not attempt to reconfigure the motor driver while moving the driver. Since this behaviour would likely cause undefined behaviour anyways, this is not seen as a significant issue.

2.3.2 Motor Control Module

The motor control module consists of a pair of motors each with its own driver and encoder. One set of motors will control the altitude angle while the other set controls the azimuth angle. The encoders will allow us accurate and precise control through a closed loop feedback method. Both the drivers and encoders will run on a 5V power supply while the motor will run on a separate, higher voltage line specified by the motor specifications.

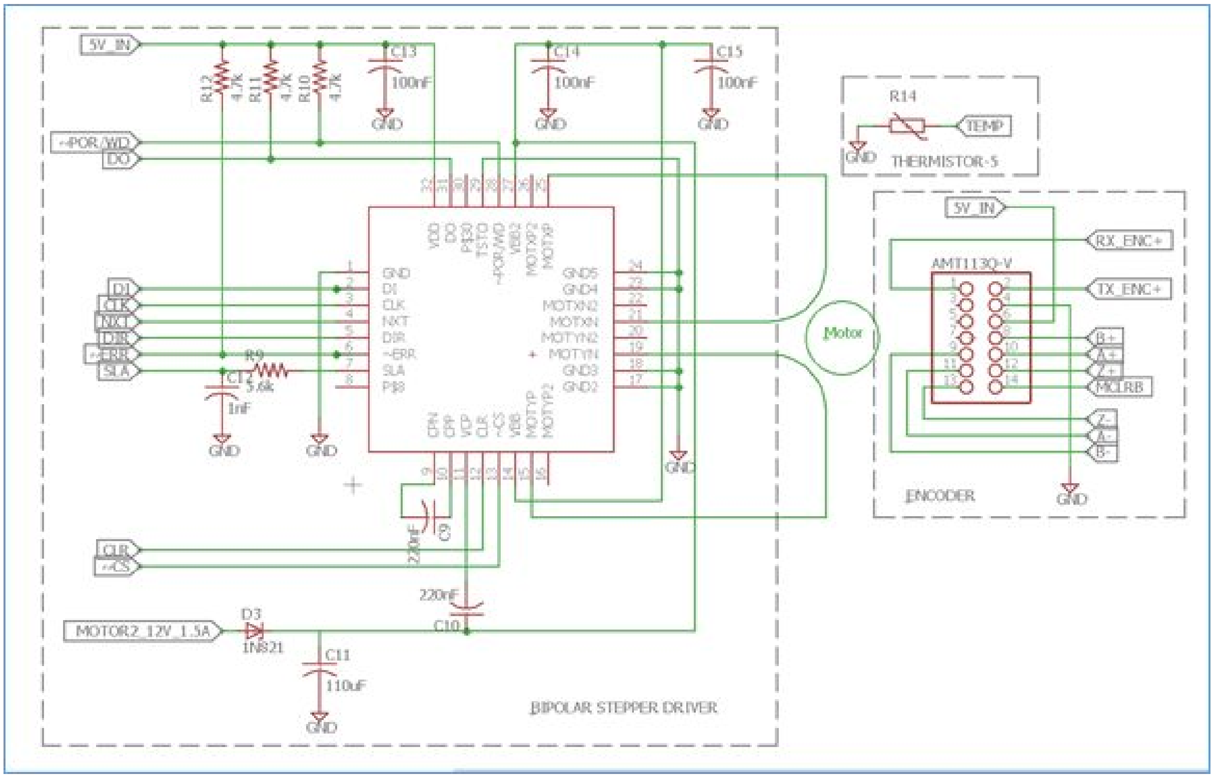


Figure 4. Motor Control Schematic for one of two sets of a motor, driver, and encoder. The motor's connections are MOTXP, MOTXN, MOTYP, and MOTYN.

2.3.2.1 Stepper Motors

The motor has to have enough torque to both move and hold the combined weight of the PATHS sensor and mount which comes out to be a safe maximum of 3 N-m holding torque. Therefore, we decided that the type of motor will be a bipolar stepper motor. Stepper because we need to be able to maintain specific angle positions and have precise control of rotations as well and bipolar because it allows a greater torque to be generated. In addition, we had to meet the angle tolerance of 0.01 degrees. As shown below, the output of our final design achieved an incredibly finer degree of precision due to the gearbox and worm gear ratios.

The NEMA17 bipolar stepper motor step angles are as follows.

- Motor Step: 1.8 degrees
- Motor Planetary Gearbox Ratio: 27:1
- Motor Output Step 0.0667 degrees

On the mount side the step angles are as follows:

- Mount Worm Gear Ratio: 80:1
- Mount Output Step: 0.00083 degrees

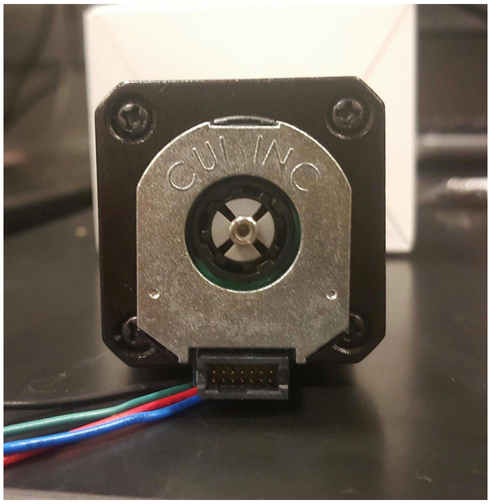
These operate at 12V±1V at a continuous current draw of 1.7A and a maximum current draw of 3A. These current and voltage specifications will directly affect our driver choices, and the resolution of the step angle will affect our encoder choices. This stepper motor can produce a maximum torque of 77 kg-cm. However, the gearbox is only rated for 30 kg-cm of continuous torque, and 80 kg-cm for brief overloads.

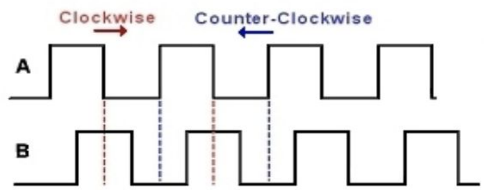
2.3.2.2 Drivers for Bipolar Stepper Motor

Since we will be using bipolar stepper as opposed to an unipolar stepper, we require a driver with an H-bridge connection. In addition, we need a driver that is compatible to our specific motor's electrical requirements: a continuous current draw of 1.7A, a voltage output within 12-24V, and a maximum current tolerance of 3A. Since our steppers have a very low step angle of 0.067 degrees, we do not need the microstepping options that stepper drivers often offer. The driver chosen for these design parameters is the AMIS-30543 Stepper Motor Driver. This chip also has useful safety features such as thermal warning and shutdown protection and shorted load protection. The driver's SPI interface places some further requirements onto our microcontroller, namely the number of pins, but it also slightly complicates our coding in interfacing with the driver. Despite some cons, this driver was our best available choice.

2.3.2.3 Encoders

We used modular incremental encoders SERIES AMT11. These offer incremental resolutions, from 48 to 4096. We used 2048 resolution in this project.





With a resolution of 2048 PPR, we have a period T of

Resolution: R = 2048 PPR
Period: T = 360/R = 0.17578 deg

With period T, the encoders have the following number of highs (rising edges) per period:

# of highs = (Step Angle)/Period
10 highs = (1.8 degrees) / (0.17578 secs)

We have 10 highs for every 0.0667 deg of motor, and 10 highs for every 0.00083 degrees of the mount. That is, with 10 rising edges for every 0.00083 degrees, we have a reliable method of measuring within our 0.01 degree tolerance.

2.3.3 Mount/PATHS Sensor Module

The PATHS sensor used in the research will be sensitive to temperature variations. As a design request from Professor Waldrop, the mount will keep track of the ambient temperatures surrounding the sensor.

2.3.3.1 Temperature Sensor

Since the PATHS sensor's optical performance varies with temperature at approximately ± 2 degrees C, we have a TMP36 temperature sensor accurate up to ± 1 degrees C.

2.4 Software Design

For responsiveness and efficiency, the software will be primarily interrupt driven. This is directly required in order to accurately interact with the quadrature encoders, which require the listener to detect all signal changes in order to accurately track change in position. This is depicted in Figure 7.

The main thread end in a loop that reports temperature and position back to the external computer. Since it is configured to report every 50ms and that the data being reported totals 32B, in a form specified in Figure 7, the total rate of communication is only 5.12kbps, which is well within the constraints of Low Speed USB ,let alone higher bitrates. Additional communication is done using packets of the same format, with 'type' being a single character that represents the type of the packet. However, these packets do not contribute significantly to the overall bitrate.

There are two types of movement of the mount. In goto movement, we only care about whether we're at the final position, so we can ignore the reports generated for intermediate positions. In panning movement, we can establish a maximum reasonable angular speed. Since we are tracking the center of the Earth's umbra, we can assume a pan speed of 360° per 24 hours, or 0.0042°/s. Since our reporting frequency is 1Hz with negligible delay, we can assume that this creates an error of at most ±0.0042°, which is negligible compared to the allowed error of ±0.5°. Thus, this is a reasonable reporting frequency without further correction.

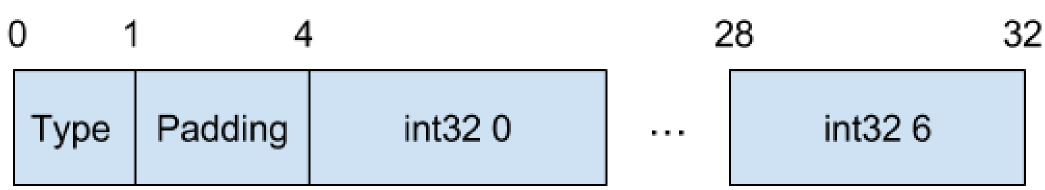


Figure 6. Communication Packet Format.

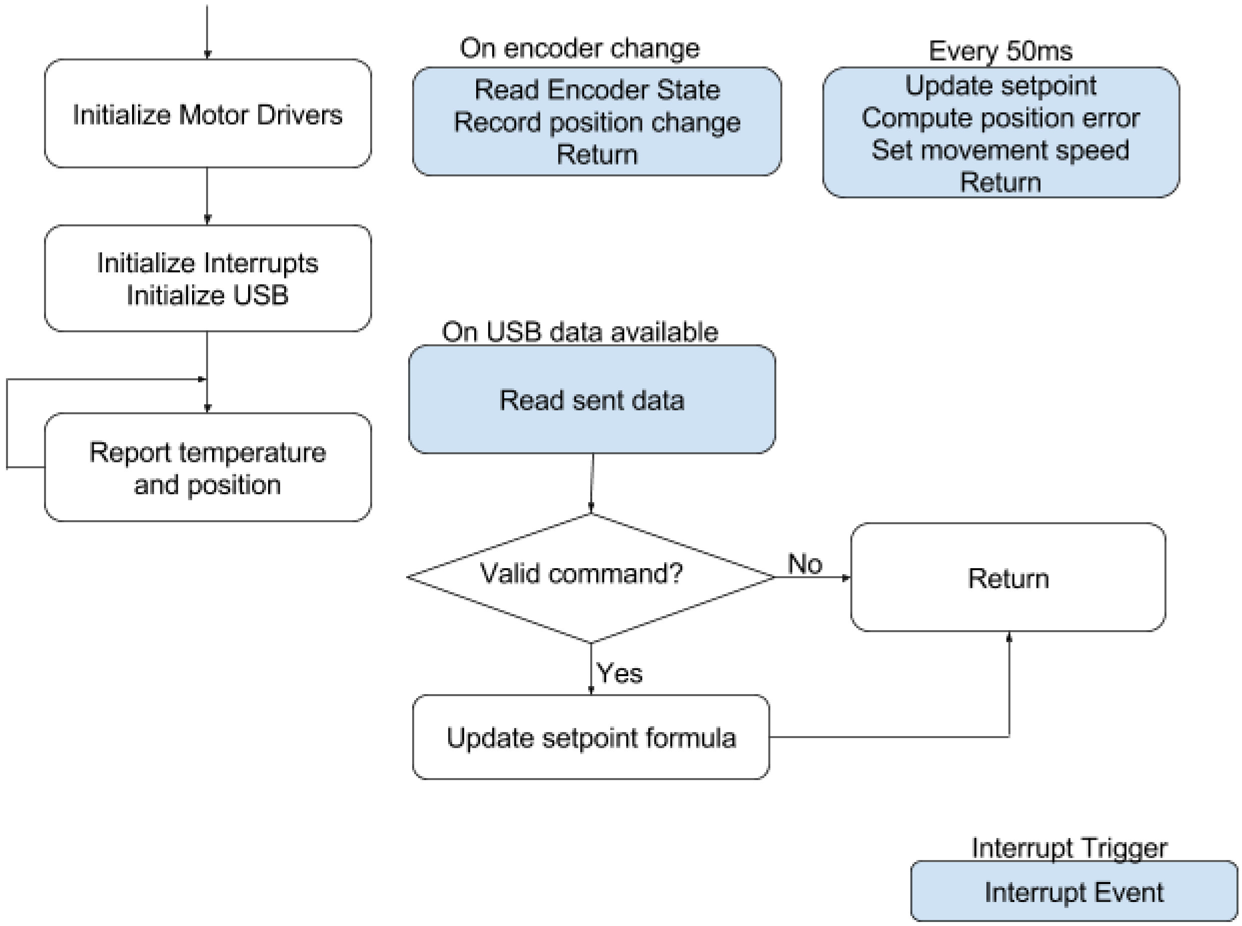


Figure 7. Software Design.

3 Tolerance Analysis

One of the most important components in our design is the bipolar stepper motors. These stepper motors have been chosen specifically to hold what we believed would be a generous amount of torque required for the mount's operation. Thus in order for this design to achieve our high-level requirements, to accurately track and to do so stably, it must at the very least be able to hold the load.

The power supply module of our design is also a crucial component. Since we will be working with constant power draw for the entirety of the operation, if the power supply should fail at any point then the mount will not work. In our design, the power will be coming from a conventional 120 V wall outlet rated for 15A. The maximum amount of current our motors can pull is 3A, and in the worst case scenario, both motor will pull a total of 6A at 12V or 72W. We want our power supply to be able to handle twice this amount of power at 140W. The outlet's 1800W rating should be more than enough but we have to handle the heat dissipation and power regulation cautiously.

3.1 Torque Approximation Calculations

From our sources, the PATHS sensor has an approximate weight of 2-4kg. Unfortunately, we were not able to directly measure the mass model we will be working out before ordering the parts so we chose to go with the conservative weight of 4kg. This weight should also cover the weight of the mount's arm itself.

The mount's arm will be approximately 5-6 inches, or 0.15m, in length, which means the radial distance will be no longer than 3 inches from the axis of rotation, or 0.075m.

First we find the force of the weight:

$$F = m * a$$
$$F = (4kg) * (10m/s)$$
$$F = 40N$$

Plugging this into the torque equation for the angle with the highest torque, $\theta = 90\ (degrees)$:

$$\tau = r * F * sin\theta$$
$$\tau = 0.075m * 40N * sin(90)$$
$$\tau = 3Nm$$

From this holding torque we based our stepper motor selection, and in turn our stepper motor driver, encoder, microcontroller, and power supply.

4 Requirements and Verification

As we received each component, we verified and tested its functionality according to our guidelines (please refer to the Appendix for more details). Most of the logical components were binary tests, e.g. whether the microcontroller's pins worked, if the encoder was reading properly, etc., and they were verified without trouble. For our motors and power supply, we measured the power draw for both stepper motors in their idle state and in their active state while the motors were attached to the physical altazimuth mount. The power draw were well within our original design and far below the 72 watt capacity of our power supply.

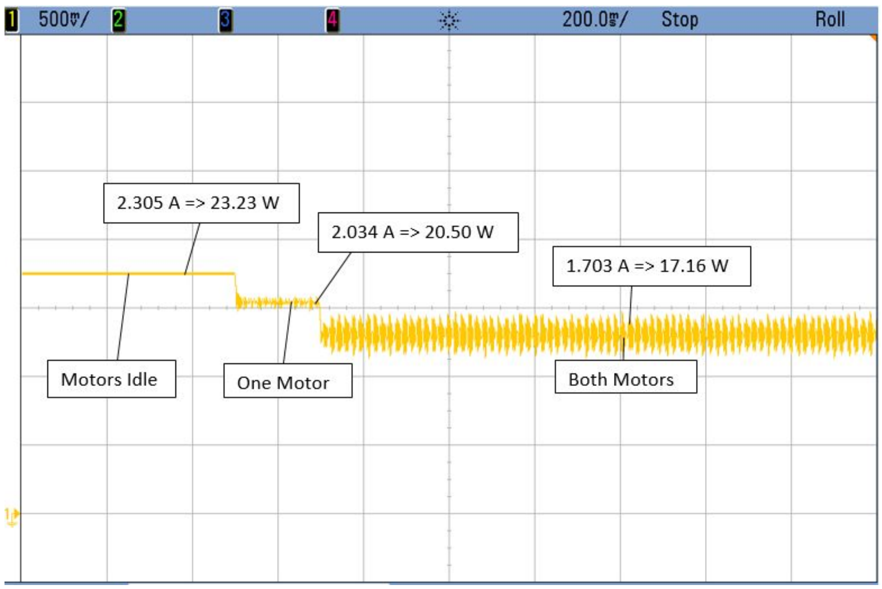


Fig. 8: Motors Power Verification; The voltage due to the current draw of the 12V power supply when motors are idle

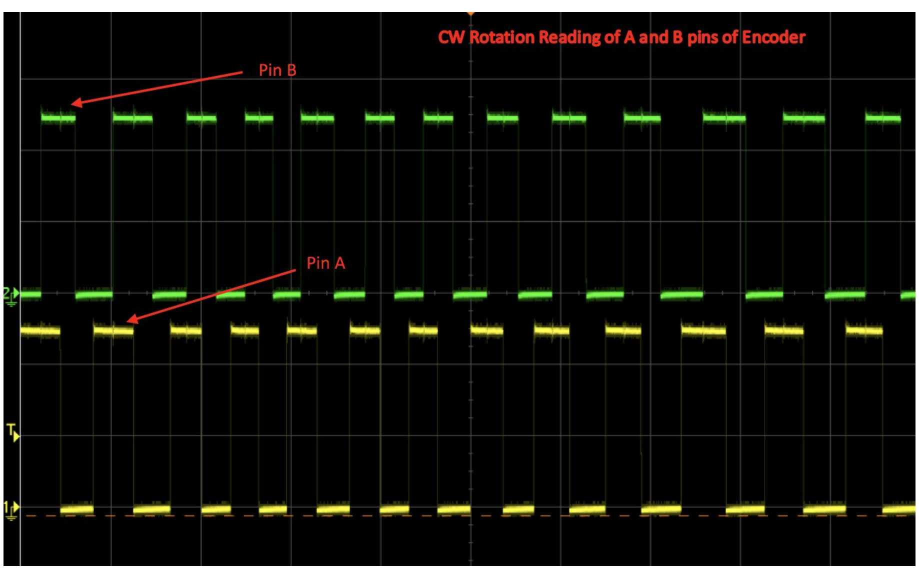


Fig. 9: Encoders Verification; Clockwise rotation reading of A and B pins of encoder.

5 Cost

We estimate a development cost by assuming a pay rate of $40 per hour, 10 hours worked per week, 14 weeks of development, and three developers. We then multiply this estimate by 2.5 to achieve our final design labor estimate.

$$3 * \frac{\$40}{hr} * \frac{10\ hr}{wk} * 14\ weeks * 2.5 = \$42,000$$

We ordered our parts as shown in Table 2, including spare parts for crucial parts of the system.

Name	**Model #**	**Units**	**Price Per Unit**
Stepper Motor	RB-Phi-210	2	$52.99
Motor Driver	AMIS-30543	4	$19.95
Encoder	AMT113Q	4	$35.85
Microcontroller	PIC16F1459	2	$1.70
Power Supply	NEWSTYLE-PT3003	1	$18.98
Temperature Sensor	TMP36	1	$1.50
Regulators	LM317	3	$0.44
Minor Components		--	$5.00
		Total :	$359.30

Table 2. Cost Analysis

$$\text{Total Cost: } \$42,000 + \$359.30 = \$42,359.30$$

6 Conclusion

In short, the design has been found to satisfy the key requirements of the project. The mount is able to move a relative angle as specified by an external computer. Elements of the original design that were incompletely implemented have very few technical barriers to being completed. For instance, the addition of the temperature sensor requires only a small amount of additional firmware and an additional hardware connection. We have satisfactorily recorded demonstrations of the mount:

1) Mount Rotation: https://www.youtube.com/watch?v=k6Dfyu8zYkU

2) Mount Tracking:

https://drive.google.com/file/d/1h4Z9Nr4JOBhQS8DZJGAvcIcVlogBNd6T/view?usp=sharing

7 Ethics and Safety

This design, if implemented correctly, does not bring up many major ethical concerns. There are, of course, potential safety issues that arise from the use of electromechanical parts, but these are minimal, as the stepper motors used are designed for slow, precise motion, even if the motion is done with significant torque. To comply with item 1 of the IEEE code of ethics, we should cover gears with a shroud if possible, and do our best to make the fact that the device is operating visible to a casual observer, as to prevent accidental damage to the observer or mount [4].

This device does bring up a major ethical concern when one considers flaws in implementation. This device will potentially be used as part of academic research. This means that improper behaviour of this device could result in invalid data, which could result in the researcher publishing erroneous results. This would injure the reputation of the researcher, which could be seen as a violation of item 9 of the IEEE code of ethics [4]. Although our actions would not be directly false or malicious, we would arguably be culpable, due to a lack of due diligence in guaranteeing the correct behaviour of our design. We should thus be particularly open and clear about any flaw or limitation of our implementation, and strive to eliminate these flaws when they could potentially cause improper behaviour.

Users should take caution when the Autonomous Motorized Mount is in operation and quit the operation if mounts are stalling. Stalling the stepper motors drains excess power exceeding the power supply limits potentially damaging the power supply. Also, it can cause the stepper motors as well as the controller integrated circuits to heat up to dangerous temperatures that can damage the mount. Although our program will raise a flag when it detects stalling via encoders, the user needs to properly respond to that event in order for operation to continue.

Therefore, the power supply will have a fuse included from the power supply output to the input of the regulators. This will provide protection against excessive power consumption and protect the power supply circuit. Each driver driving the stepper motors posses an overcurrent protection that switches off the functionality of the stepper motor protecting the life of these.

In addition, since the mount will be operating outdoors, sometimes working at high temperatures, a heat dissipator will be included for the power supply to handle high temperatures.

9 Citations

[1] "AMIS-30543 Micro-Stepping Motor Driver Data Sheet", Web, Pololu, accessed September 2017. [Online]. Available: https://www.pololu.com/file/download/AMIS-30543-D.pdf?file_id=0J869

[2] "Modular Incremental Encoder Data Sheet", Web, CUI INC., accessed September 2017. [Online]. Available: http://www.mouser.com/pdfdocs/amt11-v.pdf

[3] "Low Voltage Temperature Sensor Data Sheet", Web, Sparkfun, accessed September 2017. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Temp/TMP35_36_37.pdf

[4] "IEEE Code of Ethics" Web, IEEE, accessed September 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

[5] "1N5333B Diode data sheet", Web, Mouser, accessed September 2017. [Online]. Available: http://www.mouser.com/ds/2/308/1N5333B-D-102336.pdf

[6] "LM317 3-Terminal Adjustable Regulator", Web, Texas Instruments, accessed September 2017. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm317.pdf

[7] "LM317 Data Sheet", Web, ON Semiconductor, accessed September 2017. [Online]. Available: https://www.onsemi.com/pub/Collateral/LM317-D.PDF

[8] "Field Rotation with an Alt-Az Telescope Mount", Web, RASC Calgary Centre, accessed September 2017. [Online]. Available: http://calgary.rasc.ca/field_rotation.htm

[9] "Microstepping: Myths and Realities", Web, Micromo, accessed September 2017. [Online]. Available: https://www.micromo.com/technical-library/stepper-motor-tutorials/microstepping-myths-and-realities

10 Appendix

Requirement	Verification
Power System 1. AC/DC Power supply must output 12V ±0.1V DC with maximum current of 6A. 2. 5V DC regulator must supply a voltage of 5V±0.2V with maximum current of 500mA. 3.The Autonomous Motorized Mount system shall be able to have power at all times for continuous operation.	Power System 1. Verification Process for Item 1: (a) Attach 2Ω resistor as load. Attach oscilloscope across load. Measure voltage and verify that output is steady between 12.1-11.9V. 2. Verification Process for Item 2: (a) Attach 10Ω resistor as load. Attach oscilloscope across load. Measure voltage and verify that output is steady between 4.9-5.1V. 3. Verification Process for Item 4: (a) Set the power supply with regulators as in verification process 1, 2, and 3, operational mode. (b) Let it run for at least 12 hrs. Verify the power supply is still operating properly and there are not damaged parts. (d) Repeat 2 more times.
Stepper Motor Control 1. Drivers must be able to control the direction of the motors. 2. Steppers must be able to supply enough torque to hold and move the load smoothly.	Stepper Motor Control Verification Process for Item 1: (a) Microcontroller will output a LOW (0V) to the DIR pin and a HIGH (1V) to the NXT pin on both drivers. Verify that both motors are moving in the clockwise direction. (b) Microcontroller will output a HIGH (1V) to the DIR pin and a HIGH (1V) to the NXT pin on both drivers. Verify that both motors are moving in the counterclockwise direction. Verification Process for Item 2: (a) Attach a load to the stepper motor equivalent to the weight of the PATH sensor at the radial length of the mount arm. Supply power to the motor/driver with LOW (0V) on NXT pin. Verify that the stepper can hold the torque. (b) Output a HIGH (1V) on NXT pin. Verify that the stepper is moving smoothly

3. Drivers must be able to supply at least 12V ± 1V at 1.7A.	Verification Process for Item 3: (a) Attach a 10kΩ resistor as a load. (b) Hook up an oscilloscope across the resistor. Verify that the output for this load has a voltage of at least 11-13V.
Encoder Requirements 1. The encoder must be able to detect the smallest resolution step, 0.134 degrees, on the stepper motor.	Encoder Requirements (a) Output a HIGH (1V) to the NXT pin for only one pulse (b) Read the inputs A+ and B+ from the encoder. Verify that there is exactly one change (rising or falling edge) from either A or B.
Temperature Sensor 1. Sensor must be able to read within ~1 degree C of accuracy	Temperature Sensor Verification Process for Item 1: (a) Place a digital temperature sensor alongside the thermo-resistor. Attach a multimeter across the thermistor. Read the resistance value and check temperature at the resistance against the datasheet. Verify that the temperature is within 1 degrees of the digital temperature sensor.
Firmware Requirements 1. Firmware must correctly initialize and control the drivers. 2. Firmware must be able to correctly detect a step of the encoder. 3. Firmware must correctly move motors to a location specified via USB. 4. Firmware must report position and temperature via USB.	Verification Process for item 1: (a) Program microcontroller to move motors along each axis when powered up. Power system and verify movement. Verification Process for item 2: (a) Program microcontroller to move motors 10° along each axis when powered up. (b) Power up system and measure whether angles moved is within 0.5° of 10°. Verification Process for item 3: (a) Issue a goto command via usb. Verify correct angles moved. Verification Process for item 4: (a) Read serial data from microcontroller via usb. (b) Verify correct relative position reported within position tolerance. Verify correct temperature reported within temperature tolerance.