AN ABSTRACT OF THE THESIS OF

Colin Stark for the degree of Honors Baccalaureate of Science in Electrical and Computer Engineering presented on May 22, 2008. Title: Design and Implementation of an Automated Star Tracking System.

Abstract approved: ______________________________________

Dr. Pavan Hanumolu

An Automated Star Tracking System was designed and built for an undergraduate senior design project. The system was designed as a portable and easy to use means of tracking celestial objects, and is usable for a variety of both personal and professional purposes. The project was built around exploring the capabilities of a demo camera provided by Aptina Imaging, and has resulted in a fully functional tracking system. This thesis describes in detail the design process, implementation and future possibilities of that project. A detailed technical description of both the project as a whole and the individual components is included, as well as a reflection on the project development.

Key Words: engineering, electrical, automated, star, tracking

Corresponding e-mail address: starkco@onid.orst.edu

Design and Implementation of an Automated Star Tracking System

by

Colin Stark

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Presented May 22, 2009
Commencement June 2009

Honors Baccalaureate of Science in Electrical and Computer Engineering project of Colin Stark presented on May 22, 2009.

APPROVED:

Mentor, representing Electrical and Computer Engineering

Committee Member, representing Electrical and Computer Engineering

Committee Member, representing Computer Science

Head, School of Electrical Engineering and Computer Science

Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Colin Stark, Author

ACKNOWLEDGEMENT

I would first like to thank my mentor, Pavan Hanumolu, and other committee members, Ben Lee and Chris Wallace, for agreeing to be a part of this process with me. Their willingness to be on my committee has lifted a burden off of my shoulders and allowed me to focus my efforts on the project. I would also like to thank my project team members Nathan Blackwell and Marilyn McGettigan, as well as the corresponding computer science members Aaron Veden and Chris Dimbat, for their efforts and contributions. Finally, I would like to thank our project mentor Bill Gazeley of Aptina Imaging for providing this project with a demo camera, and offering continued support throughout the entire project.

CONTRIBUTION OF CO-AUTHORS

As this paper reflects on the work of a senior design group project, some of the materiel contained within has been worked on by all members of the group. This includes work done by both Nathan Blackwell and Marilyn McGettigan for the final specification document.

TABLE OF CONTENTS

INTRODUCTION 1

Project Background 1

Class Expectations and Influences 3

Project Team 4

Company Expectations and Influences 4

DESIGN 5

Project Research 5

Component Research 8

Microcontroller 8

Motors 11

Motor Drivers 12

Mount 13

System Requirements 13

Block Diagram and Interface Definitions 14

IMPLEMENTATION 17

Prototype 17

Microcontroller 18

Motor Drivers 20

Motors 20

Mount 21

RESULTS 21

Project Results 21

Testing Requirements 22

CONCLUSION 22

BIBLIOGRAPHY 24

APPENDICES 26

Appendix A – Microcontroller Pseudo-code 26

Appendix B – System Tests 32

LIST OF TABLES

Table 1: System Comparison 6
Table 2: Microcontroller Comparison 9
Table 3: Motor Comparison 11
Table 4: System Interface Definitions 16

INTRODUCTION

Project Background

This project used a Micron camera system mounted on a tracking device in order to automatically keep celestial objects such as planets and constellations in view over an extended period of time. The Micron camera used a 5 Megapixel (MP) Aptina® CMOS M9PO31 that had the ability to capture either continuous video or single frame images. This system enables viewing without the need for constant readjustments to compensate for the earth's rotation. This allows for the ability to capture an image that may require the camera's shutter to be open for long periods of time, without causing blurring.

Leaving the shutter open for long periods introduces more noise to the picture as extra light may enter the camera from other sources, or from imperfections in the camera itself. The software developed for this system used correction algorithms to compensate for this additional noise. This includes obtaining reference images known as dark, light, and bias frames. These frames are stored, so that when the system captures an image, they may be subtracted out to enhance the quality for making correct tracking calculations.

Many current tracking systems rely on either calculating current time and position measurements or using GPS guidance to continually track objects. These methods are typically expensive and sometimes inaccurate if being used for extended durations. Some systems also have user-controlled motors, which also miss-guide over time. This project was designed so that by taking a series of photos, comparing the target object in each consecutive photo, and calculate the necessary motor movements needed to keep the image in view, the target could be tracked reliably. This reliable automation is one of the features that distinguish it from the other systems available.

The target customer for the automated star searching telescope is someone interested in observing celestial objects, including those sent to space from earth, without needing to manually re-center the object in the lens as the earth spins. It is important to the customer that the system is able to track objects moving at various speeds across the sky and in different lighting conditions. Because of this target customer, the final system will remain at a very low cost. Also, the final design is lightweight (under 50 pounds) to improve portability and storage costs.

Though it is a quite ideal product for many amateurs, this project is not restricted to personal use. This can be used by astronomers looking to automatically track objects that do not require a significantly more powerful telescope. The true value of the system is that it will track an object for hours with no user input needed, and will therefore be a great product for anyone looking for this kind of automation. This system was designed to track celestial objects at night, but may also have potential to serve use to anyone looking to visually track a moving object. The recognition will be based on seeing the brightness of the object against the black night sky, and could be modified to track other objects with some clear distinction from their background.

Class Expectations and Influences

Because this project was completed as a senior design project, there were additional expectations beyond completing a working project. The senior design series is three terms long, and each term has a certain aspect of the project at its core. The purpose of the first term is to do all of the design work, without beginning any of the implementation. During the second term, the project must be put together and tested. The final term is meant for completing any project additions and preparing for and attending the engineering expo.

This process created some difficulties while working on this project. Primarily, it was difficult to complete a thorough design for the project without the ability to acquire any parts and work with them. Hands-on experience allows a greater understanding of how the components work and their limitations; without being allowed to do this, there was much additional time spent researching certain components and trusting in data sheets.

Building the complete system in one term presented another challenge. Parts were ordered at the beginning of the term, which caused some delay in getting started. Having a completed pre-planned design allowed the parts to be put together as soon as they arrived. Required Gannt charts helped with project planning and task division, which also helped the speed of the building process. There were mid-term check-offs for the individual sub blocks, which meant that each of these needed to be working prior to putting the whole system together.

Project Team

Another expectation of the senior design class was that this project would be completed by a team. This added some difficulties as well as some welcomed support. Working in a group allowed each member to focus on their skills and therefore become more valuable to the completion of the project than if they were to approach it on their own. This project had a five member team: three electrical computer engineering majors and two computer science majors. This meant that the project was divided into two separate portions, and because each senior design class is different added some major difficulties.

Communication was key for this project to work, as none of the deadlines for each class lined up and there were often different expectations for the two sets. Creating a well defined line between the electrical portion and software portion of the project allowed each group to focus on a specific portion of the project and redefine requirements for that portion to correspond with the class expectations. This also meant that the interface between the two parts needed to be well defined and understood by both groups.

Company Expectations and Influences

The basis for this project was discovering something that could be done with a demo camera provided by Aptina Imaging. The company providing the camera also provided a project mentor, Bill Gazeley, to offer insight and guide the course of the project. There were weekly group meetings with the mentor and additional Aptina

Imaging staff interested in the project that assisted in keeping the project focused and on schedule.

DESIGN

Project Research

There are several ways to create a mount for various telescopes, with or without motors. Since the system will be automatic, it will be using a motor-controlled system. One of the most popular types of mounts is the alt-azimuth mount, which uses two perpendicular axes for rotation. One axis moves vertically across what is known as the altitude and the other movement, called azimuth, moves horizontally. The other popular type of mount is the equatorial mount. This mount uses one motor to track objects that have diurnal motion by setting the motor to move at a constant speed. In order to track the object effectively, the rotational axis must be placed parallel to the Earth's axis of rotation [1].

Significant research was done prior to developing a solution for this project, which began with researching similar projects. Table 1 details information about nine commercial star tracking systems. The knowledge gained here gives a better understanding of what characteristics are important to a working solution.

Table 1: System Comparison

	Price ($)	Maker	Weight	Aperture	Focal Ratio	Focal Length	Software	Mount	Resolution	Zoom
Konusmotor -500 [2]	149	Konus	20 lbs.	4.5"	f/4.4	500mm	Planetary / Star charting	Equatorial	1.02 arc seconds	200x
SkyQuest XT10 [3]	480	Orion	55 lbs.	10"	f/4.7	1200mm	None	Altazimuth	0.46 arc seconds	400x
ACF-8AT LXD75 [4]	1499	Meade	65 lbs.	8"	f/10	2000mm	AutoStar Software Suite	Equatorial	0.57 arc seconds	400x
ETX-90PE [5]	599	Meade	21 lbs.	3.5"	f/13.8	1250mm	AutoStar Suite AE	Altazimuth	1.3 arc seconds	186x
DS-2080AT-LNT [6]	349	Meade	12.5 lbs.	3.1"	f/10	800mm	AutoStar Suite	Altazimuth	1.45 arc seconds	160x
Northstar 127 [7]	520	Bushnell	22 lbs.	5"	f/12.2	1550mm	None	Altazimuth	0.91 arc seconds	258x
LS60T/CaK [8]	795	Lunt Solar Systems	No Info.	2.4"	f/8.33	500mm	None	None	1.93 arc seconds	No Info.
CGE 1400 (XLT) [9]	6634	Celestron	179 lbs.	14"	f/11	3910mm	NexRemote	Equatorial	0.33 arc seconds	550x
AstroMaster 90AZ [10]	229	Celestron	20 lbs.	3.5"	f/11.1	1000mm	TheSky	Altazimuth	1.29 arc seconds	167x

The Konusmotor-500 from Konus is the cheapest telescope looked at. It is also one of the lightest, but is still more powerful than some of the mid range telescopes. The large aperture size compared to the telescope cost is a strong point, as well as the mid range resolution. With a low zoom and an equatorial mount, this telescope would be great for the low budget consumer interested in tracking objects [2].

The SkyQuest XT10 from Orion is a larger, more expensive telescope, but falls into the mid range area. Having the second largest aperture size and second best resolution will allow this telescope to produce some very nice images for a very reasonable price. The lack of software may be a downside for some, but there is a lot of great software out available to be purchased if one wants [3].

The ACF-8AT LXD75 from Meade is the most expensive Meade telescope researched. It is significantly more expensive than the SkyQuest XT10, yet is very similar in specifications. This is because Meade is known as one of the best manufacturers of telescopes. This is a very good telescope that should yield similar results to the XT10, so the cost may not be worth it, but its equatorial mount adds value over other models [4].

The ETX-90PE from Meade is a much more reasonably priced telescope, but again, the price seems quite high for the specifications. This model is significantly more expensive than the Konusmotor-500, and yet appears less capable. This would be a very good starting telescope if one had the money to invest in a name brand [5].

The DS-2080AT-LNT was the third Meade telescope researched, and it is overpriced like the other Meade telescopes. This telescope has the second smallest aperture size, the second largest resolution, and the smallest zoom capabilities, and is still more expensive than three other models shown [6].

The Northstar 127 from Bushnell is one of the most well rounded telescopes. The price seems to match accordingly, and there is no real weak part of the telescope that brings its quality down. It is very mid range, and would be a good telescope to have, lacking only software. Again, there is plenty of software available to any consumer willing to spend additional money [7].

The LS60T/CaK from Lunt Solar Systems is a high priced, low quality telescope. This price is not backed up by a company name such as Meade, and comes with no software and no mount. This leaves options open for people who may already have software and a mount, but is not reflected in the cost. This telescope would be difficult to use as is, and after paying additional money for software and a mount, has the smallest aperture size and the worst resolution [8].

The CGE 1400 (XLT) from Celestron is both the most expensive telescope researched and by far the best. It is much larger than any of the other telescopes, and may be difficult for some people to move. The large aperture, high zoom, and excellent resolution will allow this telescope to produce great images, and the equatorial mount will make tracking those images much easier [9].

The AstroMaster 90AZ from Celestron is on the opposite end of the scale from the CGE 1400 (XLT). This telescope will produce lower quality imaging, but also has a lower price. Celestron is also a well known company for telescopes, but seems to have much more moderate pricing on this model than Meade had on a comparable model [10].

Component Research

The project research provided a good concept for the overall design of what the system would need to resemble, however it does not provide much help understanding how to build up to that point. With a better understanding of what the end goal is, the project was broken down into sub blocks. Further research was conducted to determine the best solution to those sub blocks.

Microcontroller

Table 2: Microcontroller Comparison

Microcontroller	**Speed**	**Price**	**Voltage**	**I/O**	**Memory**
ATtiny26 [11]	0 - 16 MHz	$7.81	4.5-5.5 V	Analog comparator: yes ADC(10-bit): 11ch Analog gain stage: 7ch 2-wire serial interface USART	Flash: 2 kB EEPROM: 128 B SRAM data: 128 B Registers: 32
ATmega128 [12]	0 - 16 MHz	$10.94	4.5-5.5 V	Analog comparator: yes ADC (10-bit): 8ch Analog gain stage: 2ch Full duplex SPI 2-wire serial interface 2 full duplex USART	Flash: 128 kB EEPROM data: 4096 B SRAM data: 4096 B Registers: 32 Has external memory interface
Atmega25 60V [13]	0 - 8 MHz	$17.48	1.8-5.5 V	Analog comparator: yes ADC (10-bit): 16ch Analog gain stage: 2ch Full duplex SPI 2-wire serial interface 4 full duplex USART	Flash: 256kB EEPROM: 4096 B SRAM data: 8192 B Registers: 32 Has external memory interface

The microcontroller will be the connection between the computer and the mount's motors. This makes it a vital piece, as it allows the computer software to output coordinates rather than motor functions. This means that the software can focus on the sky map rather than the altitude and azimuth of the telescope and the relationship between them. Because this is done with a microcontroller, there will need to be a sufficient number of input and output lines to communicate with the computer as well as with the motors. There will also need to be enough memory to hold data that can relate the coordinates to the altitude and azimuth of the telescope.

The ATtiny26 is a good microprocessor for this task, and is one that the team has used before and is still available. This is good, because reusing a part will save money and time. Using a part that the team has worked with before would save time that would need to be spent learning about the microprocessor. This allows time to be focused on writing good code and using it in the most efficient way. The major downside to using the ATtiny26 is the limited memory available, which may cause problems when keeping track of the relationship between the telescope's altitude and azimuth and the coordinates in the sky.

The ATmega128 is similar to the ATtiny26, with more input and output, and more memory. The Atmega128 is also available from previous classes, which provides the same benefits the ATtiny26 had offered. The ATmega128 that is available is already on a TekBot board which supplies pin connections for inputs and outputs, making it easier to work with.

The ATmega2560V has the most input, output, and memory, but a slower clock speed. The clock speed should not pose a problem, because 8 MHz will still be fast enough to handle the calculations that need to be made. The set back of this microprocessor is the unfamiliarity the team would have with it. It would take additional money to purchase this, as well as additional time to learn how to use it.

The ATmega128 was chosen as the best microcontroller to use for this project, because it does not have to be purchased, and it is very familiar. The ATmega128 has all the capability required, and should have enough memory to allow precise enough detail to matching coordinates of the sky to the altitude and azimuth of the telescope.

Motors

Table 3: Motor Comparison

Motors	**Type**	**Price**	**Voltage**	**Accuracy**
NEMA 23[14]	Stepper	$19.50 each.	12V	Step angle of 1.8 degrees +/- 5%
Unipolar Stepper Motor (#27964) [15]	Stepper	$12.00 each	12 V	3.75 degrees / step
High Power Gear Box H.E. [16]	DC	$12.99 each	N/A	Gear Ratios: 41.7:1 64.8:1
BSM R-Series [17]	AC Brushless Servo	$3.00 each	115 VAC and 230 VAC drives	Acceleration torques from 4.2 lb-in (0.48 N-m) to 33.6 lb-in (3.8 N-m)

The motors will need to be able to easily and smoothly move the telescope on both axes of rotation. It is difficult to compare the differences between specific AC, DC, and stepper motors, as there are many inherent differences. The motors above are possible motors that could be chosen depending on which type of motor is the best for the design. The AC motors seem like a very powerful and effective choice at a low cost, however the main drawback of an AC motor is that it needs an AC supply. As the rest of the system is based on DC, this adds some additional unnecessary components to allow communication of the directions from the microcontroller, which is in a DC form, to the motors. Also, the entire system is kept portable with only DC, by using batteries, and adding AC motors would add the need for a generator if this system was to be used far from any AC source.

The DC motor also offers a low cost; however, using a DC motor adds more difficulty due to accuracy. A DC motor will accelerate up to a moving speed and then need to slow down before it can stop at the location it needs to turn to. This means that there is a good chance for over or under shooting on how far the motor shaft will turn, which would lead to tracking errors.

The stepper motor provides a good solution to the accuracy problem a DC motor has, because it moves in increments rather than a fluid motion, so there is no over or under shooting the target location. There are some accuracy concerns depending on the size of each step which can be reduced by choosing a large enough gear ratio to reduce the movement of the mount per step. The stepper motor also runs at 12V DC, and will require a motor driver circuit to allow the microprocessor to run it. For the above reasons, stepper motors were chosen to control the mount.

Motor Drivers

Because the stepper motors were chosen as the motors for the system, motor drivers were needed to step up the voltage and current supplied from the microcontroller. The ATmega128 output pins are capable of supplying 5V up to 20mA, which is not enough to cause the motors to step. In researching the stepper motors, all of the ones looked at had motor drivers listed that were capable of working with the motors, so no additional research was necessary to determine which motor drivers to purchase.

Mount

Due to the custom nature of this project, the mount was built by the team from scratch. The design for the mount was based off of the initial system research, which gave a good concept to start from. The decision was to make an alt-azimuth mount, because it allows for error corrections to be made while tracking. The design is also easier to implement while maintaining accuracy, which was a large benefit due to the lack of mechanical experience on the team.

System Requirements

These system requirements were established as a team, and were approved by both the project mentor and course instructor. The final project was held to these requirements as a minimum during final testing.

- The system must be able to track a celestial object for at least one hour.
- The system must be able to track a celestial object occupying at most 30% of the frame.
- The system must be able to track a celestial object with at least 50% difference from its background.
- The system must be able to track a celestial object moving at most 1% of the frame per second.

- The system must be able to track celestial objects within the full viewing region of the sky.
- The system must be light enough for easy relocation.
- The system must use a computer for image processing and tracking calculations.
- The system must be able to acquire two constellations.
- The system must be able to identify two constellations.

Block Diagram and Interface Definitions

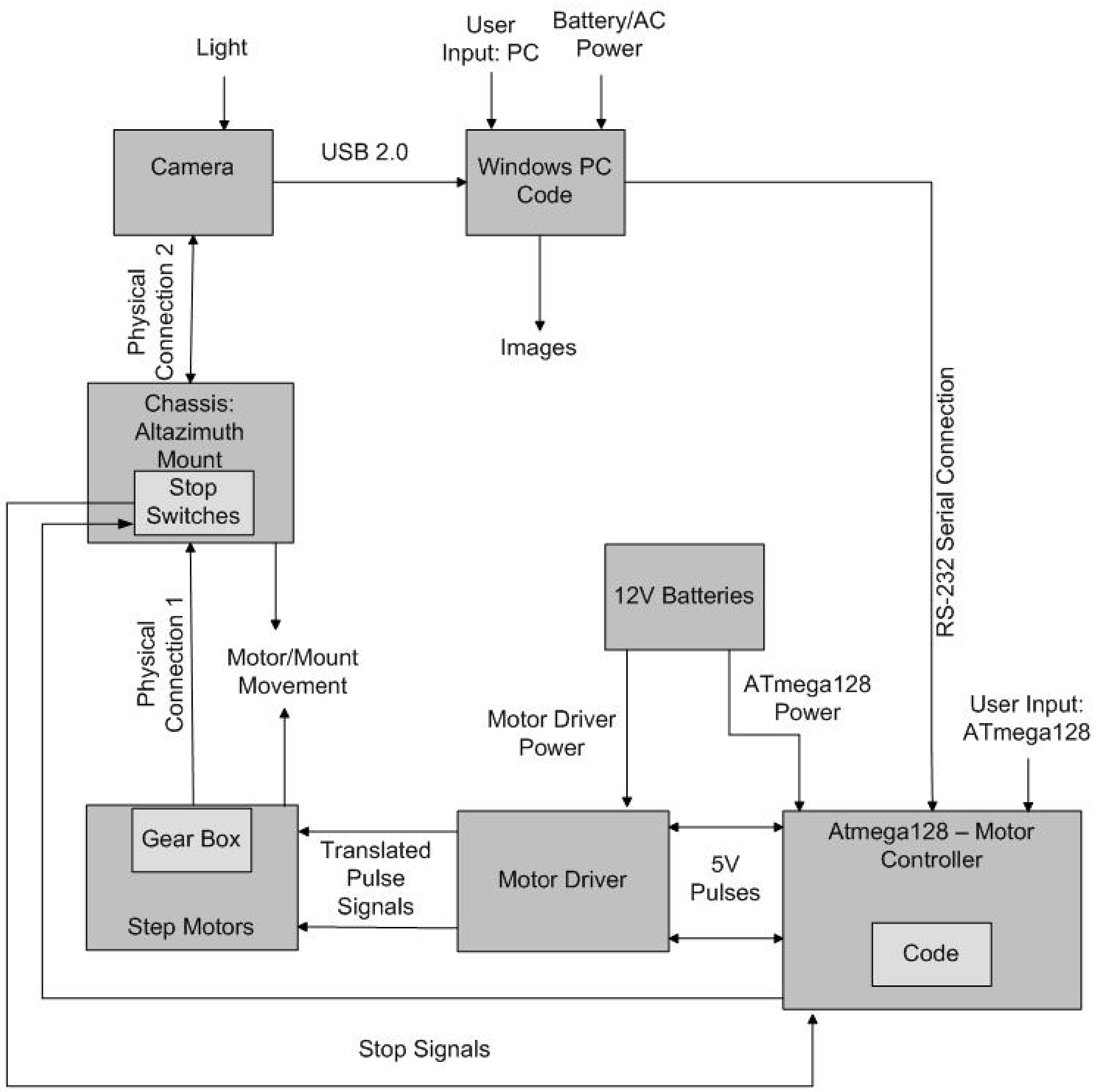

Figure 1: System Block Diagram

Table 4: System Interface Definitions

Name	**Type**	**Description**
Light	Input	This is the light taken in by the camera.
USB 2.0	Output	Voltage: 5V Current: 500mA max Power: 2.5W max
User Input: PC	Input	Declination and right ascension inputs given to keyboard using arrow keys
Battery/AC Power	Input	Voltage: 11.1V Current: 1.5A average Capacity: 4400mAh
Images	Output	Images captured from the camera displayed to the screen.
RS-232 Serial Connection	Input	Voltage: -3V to +3V Current: 20mA max Clock rate: 250kbps Protocol: Data sent as 6 bytes Byte 1: Start byte Byte 2: Command byte Byte 3: Positional data 1 Byte 4: Positional data 1 Byte 5: Positional data 2 Byte 6: Positional data 2
User Input: ATmega128	Input	Voltage: 5V Current: 20mA max per pin Parallel switches Sample every 5ms
ATmega128 Power	Input	Voltage: 12V Current: 500mA max Power: 6W max
5V Pulses	Output	Voltage: 5V Current: 20mA max Pulse sent to indicate motor movement

Name	Type	Description
Stop Signals	Input/Output	Voltage: 5V Current: 20mA max 2 output lines held at 5V to one end of each switch on the mount, the other an input switch connected to the other end of each switch.
Motor Driver Power	Input	Voltage: 12V Current: 750mA/phase max Power: 54W/phase max
Translated Pulse Signals	Output	Voltage: 12V Current: 750mA/phase max Power: 54W/phase max
Physical Connection 1	Connection	Gear ring.
Motor/Mount Movement	Output	Two separate motors can rotate, which will result in rotation of either axis of the mount.
Physical Connection 2	Connection	Camera's standard tripod screw connected to mount.

IMPLEMENTATION

Prototype

During implementation, it became quickly apparent that the most difficult and time consuming portion of the project would be building the alt-azimuth mount. Due to this, a prototype was built first as an easy way to test all of the other components as early as possible. For the prototype, a small and inexpensive prebuilt mount was purchased. The expectation was that this prototype would be fully functional and pass all of the requirements. Although this means that the project could be complete with this prototype, the final goal was to complete the full size mount capable of supporting a telescope.

Microcontroller

The ATmega128 serves as the brain of the motors. Pseudo-code for the microcontroller is seen in appendix A. The control signals for the motor are inputs to the microcontroller via relative or absolute data from the serial connection from the PC or from manual user input from the board directly. The serial connection operates at ± 3V at 20mA max and at a clock rate of 250kbps. This connection uses the DB9 connection on the board which uses pins 2 and 3 on port D. The first byte in the transmission is a high byte to help ensure both sides of the transmission know this is a new transmission. The second byte is a command byte to help the microcontroller determine how to interpret the data bytes. There are three main commands: relative coordinates, absolute coordinates, and store.

If the command is relative coordinates the first data bytes includes a signed integer representing how many pixels the object moved horizontally, while the second set of data bytes includes a signed integer representing how many pixels the object moved vertically. The data integers are positive for right and up and negative for left and down. If the command is absolute coordinates the first byte of the first set of data bytes is the whole number part of the declination coordinate with the second byte being the fractional part of the declination coordinate. The second set of data bytes follows the same pattern for the right ascension coordinate. For the store command, the data is in the same form used by the absolute coordinates command, but the microcontroller also stores the received data and links that data to the current motor positions. This is done three times. One time each for two stars and once for the terrestrial coordinates of the mount on earth.

The user input from the buttons on port D are de-bounced and sampled every 5ms via a timer interrupt. The user is able to move each motor in any direction and also stop the motors. These manual commands override any other commands. The declination motor uses buttons/pins 6 and 7, the right ascention motor uses buttons/pins 4 and 5, and the stop is on buttons/pins 0 and 1. Buttons/pins 2 and 3 are not used for user input because they are reserved for the serial connection.

Two pairs of stop signal wires are also connected to the microcontroller via two parallel switches, one for each motor. The microcontroller outputs high on one wire on the open switch loop with the partner wire for that switch being an input. When the mount hits the switch, the output pin then drives the input pin high, allowing the microcontroller to determine the mount has turned its maximum amount in that direction and prevent further movement in that direction. The two output wires are connected to port B pins 0 and 2, while the input wires are on pins 1 and 3.

Once the motor directions have been calculated, the port pins will be set accordingly and sent out to the motor driver. Each motor requires 2 pins, one for direction and one for controlling the steps. The pins on the motor driver corresponding to the declination motor use port B pins 4 and 5, while the right ascension motor driver inputs use port B pins 6 and 7.

Motor Drivers

The motor driver serves the purpose of increasing the power to the motors from signals received from the microcontroller, while protecting the microcontroller I/O pins from pushing too much current since each pin outputs 5 volts and is only able to drive 20mA max. Both motors require a separate motor driver. Power for the motor driver can range from 6V-30V DC, which includes the 12V battery pack. Each motor will pull 750mA/phase max but only when the motors are actually stepping. If needed, only one motor can be stepped at a time to reduce the needed current. Coil A of each motor is connected to pins 1 and 2 of the motor drivers while the B coils are connected to pins 3 and 4.

Motors

The motors are powered by the 12V rechargeable battery pack through the motor driver. Each motor will pull 750mA/phase max, at 4 phases when stepping. One step corresponds to a 3.4 degree turn. One motor is used for changing along the declination axis of the mount, with the other responsible for changing the angle along the right ascension axis. With every step of the motors the corresponding mount axis rotates, causing the camera to move to track the target object. Each motor is connected to a separate motor driver. Coil A of each motor is connected to pins 1 and 2 while coil B is connected to pins 3 and 4. The right ascension motor connects to the base of the mount, turning it via the 48 pitch rack gear that is connected to the base. The declination motor rotates its axis via the regular 48 pitch gear attached to the fork arm, perpendicular to the base.

Mount

The mount is an alt-azimuth style mount made from plywood in order to keep it lightweight and inexpensive. It has two perpendicular axes, each with an attached stepper motor that controls the position of each of these two main parts. At the end of rotation in each direction, there exists a spring-like switch that is monitored by the microcontroller and activated when one of the motors has reached its stepping limit in that direction. The camera is connected to the mount via the camera's build in tripod screw and connects to a fixed-position ball head for easy attachment.

RESULTS

Project Results

This project had some great success, as well as some parts that are still in progress. With the prototype, our portion of the project was fully successful, passing all of the testing requirements set forth for the class. The project mentor from Aptina Imaging was satisfied that the functionality provided was more than sufficient for what he was looking for and expected out of the project.

The full scale mount gave some early difficulty, as no one on the team had any experience with mechanics. Originally, the setup used motors that could not produce enough torque to rotate the heavy mount with a telescope. With sufficient research and trial and error, more powerful motors and new gearing were purchased, and a working solution was implemented.

At this time, the computer science members of the team have not completed their image analysis and tracking software, and this keeps the system from being a completely automated system. Without the software, the system is still a functional tracking system as long as the user is around to watch the images and make corrections manually. This is still in progress however, and should be implemented in the near future.

Testing Requirements

Because the computer science portion of the project was not completed, the final testing was modified such that it could be demonstrated that the rest of the system was working such that all of the original tests could be passed with the addition of working software. Those original tests can be found in Appendix B. All of the modified tests were met, demonstrating that the electrical computer engineering portion of the system was fully complete.

CONCLUSION

This project has resulted in a working star tracking system that is portable and easy to use. This system can be used by anyone interested in star tracking, no matter what previous experience they may have in star tracking or astronomy. The design and implementation of this system was an educational and enjoyable experience that has led to a greater understanding of and appreciation for astronomy. The final system is usable in its current state, however during the build process, some new ideas came about that may be exciting to add on in the future. For example, the system has some movement limitations due to connections to the computer; this could be eliminated by adding wireless capability.

BIBLIOGRAPHY

[1] “Different Types of Telescope Mounts,” astronomywa.net.au, Jan. 10, 2008. [Online]. Available: http://www.astronomywa.net.au/index.php?option=com_content&view=article&id=187:different-types-of-telescope-mounts&catid=44:telescopes&Itemid=8. [Accessed: Oct. 17, 2008].

[2] Konus, "Telescopes," Konus, [Online]. Available: http://www.konususa.com/. [Accessed: October 19, 2008].

[3] Orion, "SkyQuest XT10 IntelliScope with Object Locator," Orion, 2008. [Online]. Available: http://www.telescope.com/control/product/~category_id=dobsonians/~pcategory=telescopes/~product_id=27184[Accessed: October 19, 2008].

[4] Meade, "LXD75™ Series," Meade, 2008. [Online]. Available: http://www.meade.com/lxd75/index.html. [Accessed: October 19, 2008].

[5] Meade, "ETX Premier Edition," Meade, 2006. [Online]. Available: http://www.meade.com/etx_premier/index.html. [Accessed: October 19, 2008].

[6] Meade, "60mm Meade DS-2080AT-LNT," Meade, 2007. [Online]. Available: http://www.meade.com/starterscopes/ds_series.html. [Accessed: October 19, 2008].

[7] Bushnell, "Northstar® Telescopes," Bushnell, 2007. [Online]. Available: http://www.bushnell.com/general/telescopes_northstar_78-8850.cfm. [Accessed: October 19, 2008].

[8] Lunt Solar Systems, "60mm Calcium Solar-Telescope with B600 blocking filter and 2" Crayford Focuser," Lunt Solar Systems, [Online]. Available: http://www.luntsolarsystems.com/. [Accessed: October 19, 2008].

[9] Celestron, "Celestron CGE 1400 (XLT) Computerized Telescope," Celestron, 2007. [Online]. Available: http://www.celestron.com/c2/product.php?CatID=15&ProdID=90. [Accessed: October 19, 2008].

[10] Celestron, "AstroMaster 90AZ Telescope," Celestron, 2008. [Online]. Available: http://www.celestron.com/c2/product.php?CatID=62&ProdID=425. [Accessed: October 19, 2008].

[11] AVRFreaks, " ATmega128 ," AVRFreaks, [Online]. Available: http://www.avrfreaks.net/index.php?module=Freaks%20Devices&func=displayDev&objectid=54. [Accessed: November 2, 2008].

[12] AVRFreaks, " ATtiny26," AVRFreaks, [Online]. Available: http://www.avrfreaks.net/index.php?module=Freaks%20Devices&func=displayDev&objectid=61. [Accessed: November 2, 2008].

[13] AVRFreaks, " ATmega2560V," AVRFreaks, [Online]. Available: http://www.avrfreaks.net/index.php?module=Freaks%20Devices&func=displayDev&objectid=113. [Accessed: November 2, 2008].

[14] Circuit Specialists Inc, "Stepper Motor NEMA 23 (57BYGH207)," Circuit Specialists Inc, [Online]. Available: http://webtronics.stores.yahoo.net/stmo5.html. [Accessed: November 2, 2008].

[15] HobbyEngineering, "Stepper Motor (Unipolar)from Parallax," HobbyEngineering, November, 2008. [Online]. Available: http://www.hobbyengineering.com/H1431.html. [Accessed: November 2, 2008].

[16] HobbyEngineering, " High Power Gear Box H.E.from Tamiya," HobbyEngineering, November, 2008. [Online]. Available: http://www.hobbyengineering.com/H2013.html. [Accessed: November 2, 2008].

[17] Baldor, " AC Brushless Servo Motors," Baldor, 2008. [Online]. Available: http://www.baldor.com/products/servomotors/r_series/bsm_rseries.asp. [Accessed: November 2, 2008].

APPENDICES

Appendix A – Microcontroller Pseudo-code

```
Timer Interrupt:
    copy motor output pins
    if buffer not full
        receive data
    debounce buttons
    restore motor pins
end interrupt
main
    move motors to middle position
    while 1
        receive data
        if data received
            if check data
                command = received command
            else
```

```
        command = 0
    end if
else command = 0
end if
if command > 0
    switch command
        case CMD_ABS_COORDS
            if not initialized, break
            convert abs to relative coords
            convert relative to steps
            move motors
            empty data buffer entry
        case CMD_REL_COORDS
            convert relative to steps
            move motors
            empty data buffer entry
        case CMD_INIT
```

```
                    get both target stars from PC

                    get current terrestrial loc. from PC

                    initialize = 1

                end case

        end if

        check buttons

end while

receive data

    get data from PC & put in receive array

    data received = 1

    send acknowledge

end

move motors(x, y)

    if switches activated

        if x not equal to 0

            for all values of x

                step motor x by 1
```

```
            if either switch activated
                turn motors off
                break
            end if
        end for
    end if
    if y not equal to 0
        for all values of y
            step motor y by 1
            if either switch activated
                turn motors off
                break
            end if
        end for
    end if
    if abs. coordinates initialized update abs. coords
end if
```

```
end

abs2relative(x, y)

    convert x, y input to relative data based off parameters and current abs. coords

end

realtive2steps(x, y)

    convert relative input to equal number of steps

end

move2middle

    //move both motors to one side

    while both switches not activated

        if switch 1 and switch 2 not activated //move both at once if possible

            move both motors by 1

        else //move one at a time if we can't move both

            if switch 1 not activated

                move motor 1 by 1 step

            if switch 2 not activated

                move motor 2 by 1 step
```

```
    end while

    move motors to middle by counting steps

end

motors_off

    turn motor outputs off

end
```

Appendix B – System Tests

Test 1:

The system must be able to keep the center of a celestial object selected for tracking within 20% of the center of the frame.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.
3. Select a celestial object that will be viewable for at least one hour with the software.
4. Save the image seen by the camera once every minute for at least one hour.
5. Check the images and observe the location of the object being tracked.

Pass: The center of the object being tracked does not leave the center 20% of the frame for at least 60 consecutive images.

Fail: The center of the object being tracked leaves the center 20% of the frame more often than once every 60 images.

Test 2:

The system must be able to keep the center of a celestial object occupying at least 30% of the frame within 20% of the center of the frame.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.

3. Select a celestial object that occupies at least 30% of the frame with the software.
4. Save the image seen by the camera once every minute for 20 minutes.
5. Check the images and observe the location of the object being tracked.

Pass: The center of the object being tracked does not leave the center 20% of the frame in any of the images.

Fail: The center of the object being tracked leaves the center 20% of the frame in at least one image.

Test 3:

The system must be able to keep the center of a celestial object with at least 50% difference from its background within 20% of the center of the frame.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.
3. Select a celestial object with at most 50% difference from its background with the software.
4. Save the image seen by the camera once every minute for 20 minutes.
5. Check the images and observe the location of the object being tracked.

Pass: The center of the object being tracked does not leave the center 20% of the frame in any of the images.

Fail: The center of the object being tracked leaves the center 20% of the frame in at least one image.

Test 4:

The system must be able to keep the center of a celestial object moving at most 1% of the frame per second within 20% of the center of the frame.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.
3. Select a celestial object moving at least 1% of the frame per second with the software.
4. Save the image seen by the camera once every minute for 20 minutes.
5. Check the images and observe the location of the object being tracked.

Pass: The center of the object being tracked does not leave the center 20% of the frame in any of the images.

Fail: The center of the object being tracked leaves the center 20% of the frame in at least one image.

Test 5:

The system must be able to keep the center of a celestial object within 20% of the center of the frame as long as the object is located between 30 degrees above the horizon and 80 degrees above the horizon.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.
3. Select a celestial object just below 30 degrees above the horizon with the software.
4. Save the image seen by the camera once every minute for 20 minutes.
5. Check the images and observe the location of the object being tracked.
6. Select a celestial object just above 80 degrees above the horizon with the software.
7. Save the image seen by the camera once every minute for 20 minutes.
8. Check the images and observe the location of the object being tracked.

Pass: The center of the object being tracked does not leave the center 20% of the frame in any of the images.

Fail: The center of the object being tracked leaves the center 20% of the frame in at least one image.

Test 6:

The system weight must be less than or equal to 50 pounds.

1. Weigh the system.

Pass: The system weight is less than or equal to 50 pounds.

Fail: The system weight is more than 50 pounds.

Test 7:

The system must contain a computer that accepts images as input, and outputs movement coordinates to track a celestial object. The movement coordinates are determined from the images with the tracking software.

1. Observe the system.

Pass: The system contains a computer.

Fail: The system does not contain a computer.

2. Setup the system on a level surface with a clear view of the night sky.
3. Locate two known celestial objects and identify them with the software.
4. Select a celestial object to track with the software.

Pass: The computer displays images from the camera and outputs data to control the motors.

Fail: The computer does not display images from the camera or output data to control the motors.

5. Disconnect the computer from the rest of the system.

Pass: The system is no longer able to track.

Fail: The system is still able to track.

Test 8:

The system must be able to locate two different constellations selected from an available list by the user and display them on the screen.

1. Setup the system on a level surface with a clear view of the night sky.
2. Locate two known celestial objects and identify them with the software.
3. Select an available constellation with the software.
4. Save the image seen by the camera once the mount stops moving.
5. Select a different available constellation with the software.
6. Save the image seen by the camera once the mount stops moving.
7. Check the images and compare them to pictures of the two selected constellations.

Pass: The saved images are pictures of the selected constellations.

Fail: The saved images are not pictures of the selected constellations.

Test 9:

The system must be able to identify two different constellations that can be seen in the frame.

1. Setup the system on a level surface with a clear view of the night sky.

2. Locate two known celestial objects and identify them with the software.
3. Locate one of the known constellations such that it appears in the frame.
4. Check for software identification.
5. Locate a different one of the known constellations such that it appears in the frame.
6. Check for software identification.

Pass: The two constellations were correctly identified.

Fail: The two constellations were not correctly identified.