SIDEREAL TECHNOLOGY

Setup Manual

Copyright (c) 2005, 2009 by Dan Gray

Version 1.1a

Revised October 19, 2009

THE BIG PICTURE *(READ THIS FIRST!)* 7

1 FEATURES 8
1.1 BASIC BUILT-IN STANDALONE CONTROLLER FEATURES 9
1.2 ADDITIONAL FEATURES AVAILABLE USING THE SITECHEXE PROGRAM 10
1.3 COMPARISON BETWEEN STEPPER SYSTEM AND SERVO SYSTEM 11
1.4 SAFETY 12
1.5 SOFTWARE VERSIONS 13

2 HARDWARE SETUP 14
2.1 SERVO CONTROL SYSTEM INTRODUCTION 14
2.1.1 What are the "minimum" hardware requirements for a servo control system? 15
2.1.2 What are "optional" servo control system components to consider? 15
2.2 DESIGN CONSIDERATIONS 16
2.2.1 Portable vs. Permanent or Semi-Permanent 17
2.2.2 Visual or Imaging 18
2.2.3 Local Imaging Operations vs. Remote Control Operations 19
2.3 MECHANICAL CONSIDERATIONS 19
2.3.1 Types of Mounts 19
2.3.2 Types of scopes 24
2.3.3 Considerations if initially building vs. converting existing 25
2.3.4 Gear Ratio Considerations 25
2.3.5 Drive Type considerations 27
2.3.6 Wheel and worm Gears 28
2.3.7 Gear reductions 32
2.3.8 Clutches 33
2.3.9 Mount Encoders 34
2.4 ELECTRICAL AND ELECTRONICS CONSIDERATIONS 35
2.4.1 Electrical Power Source 35
2.4.2 To use a PC interface or not? 35
2.4.3 Computer and Operating System Requirements 35

3 SERVOCONFIG.EXE 37
3.1 DO I REALLY NEED TO USE SERVOCONFIG? 37
3.1.1 Should I use ServoConfig 37
3.1.2 Should I use ServoConfig if I use only use ScopeII? 37
3.2 PURPOSE AND BRIEF DESCRIPTION 38
3.3 INSTALLING SERVOCONFIG.EXE 38

4 SERVOCONTROLLER 39
4.1 MOUNTING THE CONTROLLER 39
4.2 POWER 39
4.3 LED OPERATION 39
4.4 SERIAL PORT CONNECTION 40

5 SETTING UP THE MOTORS 41
5.1 MOTOR WIRING 41

5.2 MOTOR CONFIGURATION **42**

6 HAND PAD **42**

6.1 RADIO HAND PAD **43**

7 TELESCOPE ENCODER WIRING AND SETUP **44**

8 USING SERVOCONFIG TO CONFIGURE THE CONTROLLER **44**

8.1 MAIN SCREEN OVERVIEW **47**

8.1.1 Text Box Area 47

8.1.2 Virtual Handpad Area 48

8.1.3 X and Y bits status area 48

8.1.4 Main Menu Items 49

8.2 SETUP COMM PORT **49**

8.3 EDIT PARAMETERS **50**

8.3.1 Motors/Encoders Tab 51

8.3.3 Equatorial and GEM Hemisphere Invert Button Setup 55

8.3.4 Auto Tracking tab Dialog 57

8.3.5 Drag And Track Mode 58

8.3.6 Equatorial Mode 63

8.3.7 Misc. and Action Screen 65

8.4 TEST AND TUNE **70**

8.4.1 General PID Tuning Principles 71

8.4.2 Accel 72

8.4.3 Read Flash and Write Flash 72

8.4.4 Refresh 72

8.5 TERMINAL WINDOW **73**

8.6 TICKS DETERMINATION **74**

8.6.1 General Principal 75

8.6.2 Picking Starting and Ending Positions 75

8.6.3 General Procedure 76

8.6.4 Specific Procedure 76

8.6.5 More Info 77

8.6.6 Ticks Determination using mount encoders 78

8.7 BACKLASH HELP **79**

8.7.1 General Backlash Procedure 79

8.7.2 More Notes on Backlash 80

8.8 UPGRADE FLASH ROM **80**

8.9 UPDATE SCOPE II **82**

8.10 EXIT SERVOCONFIG **82**

8.11 HELP **82**

APPENDIX A – CONTROLLER SPECIFICATIONS / WARRANTY / TECH SUPPORT **83**

CONTROLLER SPECIFICATIONS **83**

DISCLAIMER **83**

WARRANTY **83**

TECHNICAL SUPPORT **84**

APPENDIX B - AGENCY APPROVALS 85

APPENDIX C: UNDERSTANDING AND USING THE X_BITS AND Y_BITS 86

APPENDIX D: COMMUNICATION PROTOCOL 88
ASCII COMMANDS 89

APPENDIX E: CABLE DIAGRAMS 96
DIAGRAM OF COMMUNICATION CABLE FOR THE ARGO NAVIS™ 96
DIAGRAM OF COMMUNICATION CABLE FOR THE SKY COMMANDER™ 97
DIAGRAM OF COMMUNICATION CABLE FOR CONNECTING TO A PERSONAL COMPUTER 98

APPENDIX F: BLOCK DIAGRAMS OF SOME POSSIBLE CONFIGURATIONS 99
COMPLETE SYSTEM: 99
SYSTEM USING ARGO NAVIS™ FOR FULL GOTO AND TRACKING 100
DRAGNTRACK SYSTEM 101
SLEWNTRACK SYSTEM 102

APPENDIX G: WIRING DIAGRAM FOR EARLIER CONTROLLER 103
DIAGRAM FOR PULL-UP RESISTORS FOR EARLY CONTROLLER ENCODER CABLES 104

APPENDIX H: DIAGRAM FOR LATER CONTROLLER 105

APPENDIX I: DIAGRAM FOR RADIO HAND PAD RECEIVER 106

APPENDIX J: AUTOPECSYNCHSENSOR SCHEMATICS 107
SIMPLEST RA PEC DETECTOR 107
OPTICAL PEC DETECTOR USING A TRANSISTOR 108
MICROSWITCH PEC DETECTOR 109
HALL EFFECT DETECTOR 110

APPENDIX K: WIRING DIAGRAM FOR GURLEY HIGH RESOLUTION ENCODERS 111

APPENDIX L: SUPERVISORY POTENTIOMETERS DIAGRAM 112

APPENDIX M: THERMALLY COMPENSATED TEMP SENSOR AND ZERO POSITION INIT SENSOR 113

APPENDIX N: LIMIT SWITCHES 114

APPENDIX O: HOMING SWITCH CIRCUIT AND SETUP SCREEN 115

APPENDIX P: SAMPLE ENCODER INSTALLATIONS 117
GURLEY ENCODER INSTALLATION ON G-11 SITECH CONVERSION 117
GURLEY ENCODER INSTALLATION ON AP1200 SITECH CONVERSION 119
GURLEY ENCODERS PLUS FLEX COUPLINGS 120
"GEARED UP" ENCODER MOUNTING 120

APPENDIX Q: TROUBLESHOOTING **123**
MOTOR OPERATION **123**
HANDPAD OPERATION TROUBLESHOOTING **123**
MOUNT ENCODERS TROUBLESHOOTING **124**

Revision History for the SETUP Manual

Version 1.1	October 7, 2009	Baseline edition
Version 1.1a	October 19, 2009	Geared up encoder info added, Sky Commander pinouts added

THE BIG PICTURE *(READ THIS FIRST!)*

There are two parts to the documentation for the Sidereal Technology Servo Control System's documentation. The first part is this SETUP manual, and the second part is the OPERATIONS manual. Scan them both to get a feel for all the topics and capabilities, but to keep from getting lost in all the capabilities this system has to offer, we really recommend when you start reading for content, you start with this SETUP manual, and read through it first. It's written in a "building block" type of approach.

- The idea is to cover considerations about initially building your drive system, or perhaps converting an existing drive system to a servo control system, and what components are required, and what are optional.

  The next part assumes you have mechanically installed your servo motors. It covers how to get all the "required" components hooked up to your controller: Motors, handpad, power supply, and a serial port to a PC (and if your PC does not have a serial port, how to use a USB-to-serial converter (dongle) to provide a serial port).

- The next step, and the majority of this manual, then covers how to update the controller with the latest firmware (to tell it that it is a controller), and to update it with the basic "required" information about your telescope mount. It starts with how to install a very powerful program called SERVOCONFIG, which is what you will use to send information to the controller. It also has tools for helping you measure and set up critical parameters for your system.

  When you get to this point, you should be able to actually USE your mount. None of the really cool bells and whistles will have been covered, and in fact basic operations of the mount do not even need a PC to be hooked up to the controller! (Remember, we want to learn how to walk before we run!) This section includes information on how to use your mount in a standalone mode without a PC connected (i.e. in Drag & Track mode, Slew & Track Mode, or Equatorial Mode). You should do this to get familiar with this capability, even though in the future you plan to normally always have a PC connected to be ready for those times when your laptop dies!

- After this point, you will want to transition to the OPERATIONS manual to learn the basics of using the powerful SiTechExe program on your PC to add important capabilities like GOTO, and how to use external programs like planetarium programs to control your mount (i.e. click on a target in the planetarium program to aim the telescope at that target, etc.) . The Operations manual will start with how to install SiTechExe, and some related software that it needs to have installed. And, just like with the controller, SiTechExe will need to be updated with some basic information about your mount, and your location, etc. So, that will be one of the first things covered.

  Now you will have a mount that has a really sophisticated control capability, but a very simple user interface. We recommend you spend some time at this point simply USING the system out under the stars to get down using the basics before you try to also come to grips with the more powerful capabilities the system has to offer.

- When you are comfortable with operating the system using your PC and SiTechExe, then it's time to start into the "advanced" operations. Use the OPERATIONS manual for this. There are some

advanced capabilities that make more sense than others to incorporate first, such as using the SiTechExe tools to build a model of your mount to dramatically improve both tracking and GOTO accuracy, and using the Periodic Error Correction capability to eliminate periodic error in the RA drive for an equatorial mount. If you need to go back and change any configuration information in the controller as part of using any of these advanced capabilities, the OPERATIONS manual will reference you to where to go in the SETUP manual.

- There are several appendices in each of the manuals that are related to either setup activities, or operations activities. For example, graphics of how the components are cabled up, and the pin outs of the various cables are in the SETUP manual. The circuit diagrams for advanced capabilities like an optional autosync detector for the periodic error correction routine, and a sensor for the thermal compensated focusing capability are also in the SETUP manual appendices. The OPERATIONS manual appendices contain sections on how to configure several of the more popular planetarium programs to interface with SiTechExe, and on doing some really cool things like satellite tracking, and using your CCD camera to "write" with a star's images. There are also appendices that contain "Cue Cards". A Cue Card is a shorthand checklist to use out under the stars to remind you of how to do things rather than having to try to thumb through the Ops Manual to try to remind yourself of things.

- And finally, both manuals have troubleshooting sections for when things don't seem to go right. Hardware does occasionally have issues (and gasp, even software!), and the SiTech system has some powerful capabilities to help with the troubleshooting to allow working your way through correcting things.

1 FEATURES

There are Six ways to use the SiTech Controller on your telescope:

1. Stand alone. In this mode, the controller will track your Alt/Az telescope using the DragNTrack or SlewNTrack modes. It will also track your Equatorial telescope or Equatorial Platform using the Equatorial mode. Many users prefer the stand alone mode, for truly no hassle, telescope tracking. This mode is also the most economical. It is also the mode that this SETUP Manual will initially have you configure your system to use to allow basic operations, and to confirm your system is wired up right and mechanically functional.
2. Use the Argo Navis™ from Wildcard Innovations, for complete and accurate GoTo and Tracking of your telescope. This is a really cool way to go if you don't want the hassle of using a PC at the telescope. Using the SiTech unit along with the Argo Navis™ has several advantages over other systems which use the Argo Navis™. If optional telescope encoders are used, they are connected to the SiTech unit instead of the Argo Navis™ (the Argo Navis™ encoder port is un-used), and the serial port of the SiTech unit is connected to the Argo Navis™ serial port. Some of the advantages are that the encoders draw power from the more powerful Servo Controller Batteries, so the Argo Navis batteries last much longer, and also, the fact that the telescope simply doesn't need external encoders to work with the Argo Navis™! The user has the option of using external encoders, or simply use the motor location instead (if you have a system that doesn't slip), the best of both worlds!
3. Use the provided FREE ASCOM driver from Sidereal Technology. This requires a Laptop or Desktop computer, running Windows 98 through Windows XP. The advantage of using the ASCOM driver, is almost any planetarium program that supports ASCOM will accurately control your Alt/Az

or Equatorial telescope. No special Virtual Serial Port drivers or NULL modem cables to otherwise unused serial ports. This ASCOM driver is very powerful, and supports almost every ASCOM Version 2.1 function. The Argo Navis is not required!

4. Use Mel Bartels SCOPE II Windows/Mac/UNIX software (Java Based), observatory quality, incredibly accurate GOTO and Tracking. Mel has spent many hours providing almost every possible feature in SCOPE II. If you want better than 1 arc minute pointing, sky to sky, or want to do serious photography, Mel's SCOPE II is the way to go.
5. Use Mel Bartels Windows software, ScopeIII. We are excited about this, and wait with anticipation what Mel will have put together in this new software.
6. Use a Palm Pilot, a Blue Tooth Serial module and Astromist Software by Cyrille Thieullet. We are also excited about this option, as Cyrille has created a brilliant piece of software with many features. Please check out http://www.astromist.com for more details. In addition to the PDA running Astromist, you will need a very basic computer with a bluetooth receiver that can run an ASCOM to LX200 interpreter called LXtoSitech.exe to allow the controller to understand LX200 commands from the PDA via a Bluetooth link, and then the PC connects to the controller.Sounds complicated but it is actually very straightforward!
7. Also, soon to come is control using JavaME, which can be found on many Cell phones!

Finally, a servo telescope control that is affordable! You can purchase a bare bones servo controller with motors and a handpad for much less than you would imagine!

1.1 Basic Built-in Standalone Controller Features

Directly controls two servo motors with the motor position control loops running at nearly 2000 times a second, with high speed shaft encoders for position feedback on each motor.

In addition to the two encoders mounted on the motors, the SiTech unit can connect to two additional high speed encoders for the telescopes altitude and azimuth (or right ascension and declination). No need to add an additional "box" to connect shaft encoders. Mount shaft encoders can correct for slippage of friction drives and high resolution encoders can eliminate periodic error of worm drives.

Open Collector RS232 Serial port for connecting 2 or more controllers on the same RS232 bus. Other controller axis's can be used as field de-rotators and/or focusers (Available using PC control only).

Connects to a wired handpad or optional radio handpad for slewing/panning/guiding.

Signals from the wired or radio handpad work immediately, no delay as with some other systems, because the same micro-controller that is controlling the servo motors, "listens" to the handpad signals. No communication protocol delay. The Radio handpad protocol, is a lean and mean protocol, taking only the time to transfer 16 bits at a quick bit rate. There is absolutely no perceived delay between when you press a button on the handpad (radio or wired) and when the servo motors move.

Can automatically track your telescope in "computer-less" modes like the Equatorial mode, the DragNTrack mode, or the SlewNTrack mode without having a personal computer connected.

When running in the computer-less modes mentioned in the above paragraph, you may connect a computer running just about any planetarium program, and the servo controller will emulate the Tangent Digital Setting Circle protocol. This allows just about any planetarium program to provide the "crosshair" or other indicator, of where the telescope is pointed.

Can control a dual axis tracking platform, with auto stop and rewind, and dual axis correction without having a personal computer connected.

Servo Motors have a FASTEST mode of operation, where if the motors can't quite keep up with where it is supposed to be, but it is moving in the proper direction, it will not trip on position error which is the industry standard, instead, position error is held at the point of the position error limit. This feature makes it possible to run your motors near their limit of speed, and you can rest assured, it will not trip on position error if your battery voltage falls a bit, or the wind pushes against the scope.

Local search: Upon a quick combination keypad request, your telescope will search for an object, in a circular pattern. This is the most efficient search pattern, as there is no overlap between successive search patterns. This is a true circle, no matter where in the sky you are looking. No inefficient ellipses at different altitudes. You have two choices of radius, reverse, forward and pause control, and speed of the search can be changed while looking through the eyepiece!

Internal Backlash: Backlash is internal, and instantaneously applied. Each axis has a backlash variable associated with it, and an adjustable backlash area speed.

Equal Area Pan/Guide Speeds: The Azimuth/RA Pan and Guide speed is automatically adjusted for different Altitude/Declinations. As you near the Zenith (or Pole), the Pan and Guide speeds are automatically increased so the same sky area is covered in Azimuth/RA as Altitude/Dec. This is a really cool feature, and if you experience it, you won't want to go back!!!

1.2 Additional Features Available Using the SiTechExe Program

PointXP for corrections of pointing errors is incorporated into the SiTech.exe software. Using a PC, you can align on stars across the sky once, and then have extremely high pointing accuracy with corrections for most mount, polar alignment, and telescope errors on subsequent nights. Pointing accuracies of several arc seconds have been achieved.

PlateSolveXP is a stand-alone program that determines the center of field of view of a camera and passes those coordinates to PointXP for exceptional pointing capability with the mount It graphically shows the extracted stars, the catalog stars and the matched stars. It also displays the J2000 coordinates of the cursor, a useful feature for astronomical research and object identification.

CalPointsXP is a utility that works with SiTechExe to automate the mapping of the sky using PlateSolveXP and PointXP to generate a sky database of mount corrections for extremely accurate mount pointing.

Built-in periodic error correction and PemPro™ operation is available in SiTech.exe with a PC connected. You can record periodic error directly in SiTech, or use PemPro™ to determine the

correction pattern. The controller will then apply the corrections automatically. This feature requires either manually synchronizing the worm gear rotation or using an AutoPECSynch sensor on the worm gear. Circuits for these sensors are shown in the appendix.

Use high resolution Gurley encoders on the mount with automatic Interpolation Error Correction for sub arc-second tracking. SiTechExe can use either the 320K or 500K encoders. When used on an Equatorial mount, periodic error in tracking is eliminated. When used with Alt-Az mounts, slippage and periodic error in the drive trains are eliminated.

Park and UnPark can be used to retain sky alignment from session to session. A repeatable park position alignment, either by optically sighting on a target or using a precision level on a GEM balance shaft and scope and accurate time on a PC can allow immediate slewing to targets after startup.

Scripting can be used to plan and run a night's observing session, or even run the mount remotely. Using a PC, SiTech.exe has a script building application you can use to make and run the script.

Polar Alignment of an equatorial mount can be quickly accomplished with the PC application built-in to SiTech.exe.

Built-in Simulator capability of SiTech.exe automatically senses if there are no servos connected and operates with the note: "Faking Servos". This powerful feature allows you to learn and test the controller system along with Ascom connections to other software such as TheSky™, Maxim, PemPro™, Elbrus and many other programs.

Built-in list of alignment stars and Deep Sky Objects (Messier, NGC, IC) for synch or GoTo slewing.

Customization of the display of SiTech.exe controls on a PC allows you to set the colors of buttons and displays to suit your taste.

Small size, 2.5" by 4.25" by 1".

Power Efficient. Will last several nights on one 7 amp hour Gel Cell on most telescopes.

Built in flash memory for saving parameters.

Firmware in the controller is field upgradeable to later versions of the firmware, no need to send it back to the factory for additional features.

Windows Configuration program is supplied free with the unit or is available online for free downloading.

1.3 Comparison between Stepper System and Servo System

A servo system has many advantages over a stepper system.

1. Wider dynamic range. This means a servo system will track your telescope nearly perfectly and still obtain faster slew speeds than a comparably geared stepper system.

2. Angular accuracy of the motor while tracking is much better. There are many errors in a stepper system when being micro stepped. Generally speaking, the micro steps are not nearly as accurate as the resolution of the micro step. There are many errors such as mechanical winding differences, magnetic hysteresis, torque error, and several others that can be partially compensated for in software with a stepper system, but never will be as good as a servo system, which will almost all the time be within 1 encoder tick of it's desired location at all times (while tracking).
3. More torque. Small servo motors will control telescopes as large as 41 inches!
4. Much lower current during tracking. Batteries will last much longer.
5. No resonant frequency's to battle.
6. Motors will never "miss" steps. Many times stepper motors will simply stop instead of turning during slewing. The computer controlling the stepper motors will loose its position information when this happens.
7. A real time operating system is in the Servo Controller itself, thus making a real time operating system unnecessary for the controlling computer (some stepper systems have this too). This means the same computer can operate your telescope with a planetarium software interface while still tracking, perform imaging, auto guiding the telescope, controlling a dome, etc, all at the same time using the same computer. This can be a significant cost savings of a servo system over a stepper system, evening the cost savings of a stepper over a servo system.
8. Much smaller size (in the case of the SiTech controller anyway).
9. If only tracking is desired, no external computer is necessary (SiTech feature).
10. If no external computer is desired, but GoTo's are, the SiTech Controller can be controlled by an Argo Navis™ or a Palm computer.

The stepper system has only one advantage over the servo system that we can think of: Cost, although a servo system can be less than four or five hundred dollars, including controller, surplus motors, and software if you do most of the work yourself. Although cost is an advantage of the Mel Bartels DOS based Stepper system, Mel's system is the only stepper system we know of that is less than the SiTech servo system, and then if you want to do imaging, planetarium software control (while still tracking) or other simultaneous computer operations, you'll still have to use a 2nd computer.

Some books or articles may mention Servo Lag as an advantage of a stepper system over a servo system. This isn't an advantage when compared with the SiTech Servo system. When tracking, the servo motors will almost all the time be within **ONE** motor encoder tick and occasionally, two motor encoder ticks of its desired position. This is usually between 1/10th to 1/4 of an arc second (depending on your gear ratio).

Despite the many advantages of a servo system over a stepper system, many successful stepper systems have been created, which work extremely well.

1.4 Safety

The servo motors are powerful motors that can do extreme damage to equipment, and/or plant and animal life. Please be careful that your telescope doesn't move to a mechanical limit (such as the ground). You may damage the motors or gears, and you may injure yourself or others. Use this product at your own risk.

1.5 Software Versions

The manuals are written and illustrated with screen shots using current versions of SiTech configuration software (ServoConfig v1.1), control software (SiTech.EXE v054M), and Firmware Version 35C. Please update to these or later versions in order to use all the features available. You can find the latest versions at http://www.sitechservo.info/downloads

So, enough of the BIG picture!

Scan through both of the manuals to get a view of things from 40, 000 ft, and then come back to here and dive in starting with the next section, "Hardware setup"!!!!

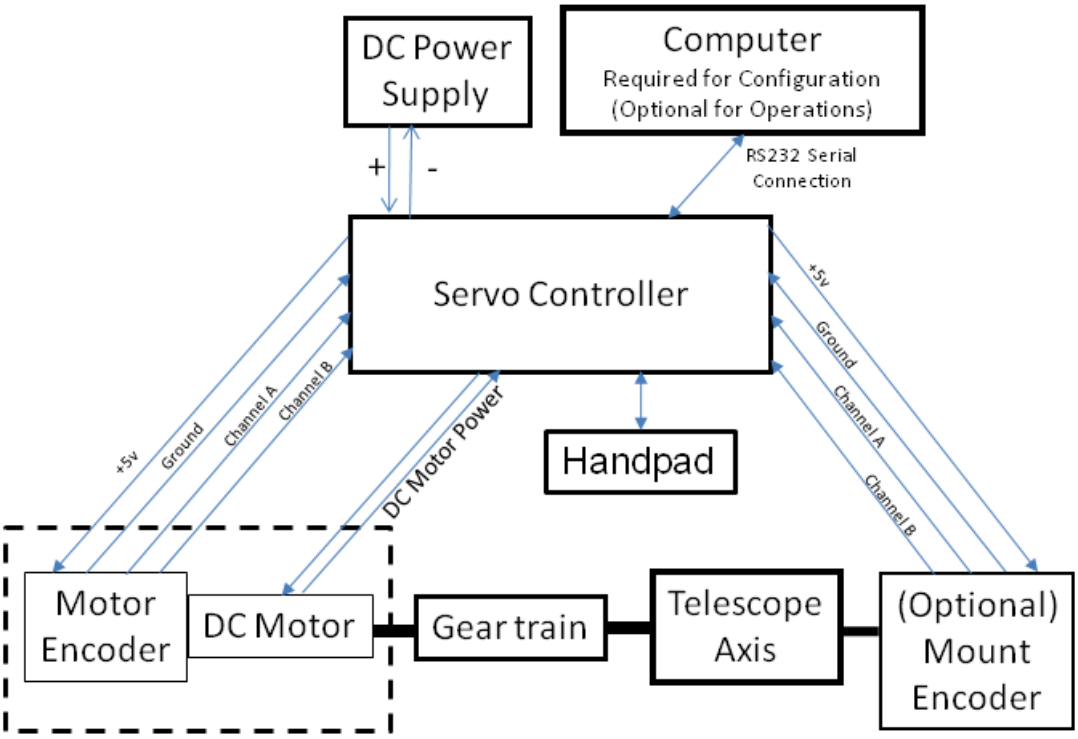
2 Hardware Setup

The SiTech controller system is one of the most versatile designs available to today's amateur astronomers. It can be installed in any number of mounts and scope types for visual or imaging, fixed or portable and even remote operations. Since it is an "after market", no two installations are alike. This section is then devoted to describing the factors and considerations that you need in order to understand and make the best decisions about what you need, how to install it, and how to use it. We try to cover all the factors, but we know there will be questions left unanswered. If you haven't already, you should become a member of the Yahoo SiTechservo Group at http://tech.groups.yahoo.com/group/SiTechservo/ where you can get expert information and talk with the designer of the system.

2.1 Servo Control System Introduction

Servo systems are actually very straightforward. They are a DC motor (which has two power leads to it), and an "encoder" (which has 4 leads to it) which provides information about the position of the motor shaft., and a controller that uses software (firmware) to send power to the motor to move it to a desired position and uses the motor encoder for feedback on how it's doing. The feedback is the key to the system, and the magic is how the controller uses the feedback to precisely position the motor (or more correctly control the rate the motor is turning, since the position is simply rate x time).

The following sketch shows the essential elements of a servo control system (only shows a single axis to keep the sketch simple).



2.1.1 What are the "minimum" hardware requirements for a servo control system?

A servo system requires a DC motor and an encoder (normally attached to the motor). You can use most any DC motor that the controller can support (controller provides 12-24 v input, and can provide up to 4 amps DC total, with 2.4 amps max for either axis). The motor does not HAVE to have the encoder as an integral part of the motor, but an encoder should be attached to the motor's shaft to provide the highest resolution possible for good performance. The controller will have the DC power and the motor encoder cables that go to each of the motor/encoders. The controller then needs a handpad attached (can be a wired handpad or wireless), and a DC power supply (12-24v dc). A 7amp hr battery will run the system for several evenings without re-charging (assuming you do not spend the WHOLE night simply slewing around!!!!). For 110v power supplies, its best to use a regulated power supply, but a switching power supply will also work (like PC power supplies if rated for enough amperage). A computer is required to run the configuration software (ServoConfig.exe) and a serial connection from the computer to the controller. If your computer does not have a native serial port, you can use a USB port and an inexpensive serial to USB dongle to create a com port. You only need one comm port, even if you end up using two controllers (like one for the RA/Dec axes, and one for two focuser motors, etc.). The software is Windows software (XP runs the best). But Mac's in emulation mode work fine. Windows 98 machines will be fine for the configuration software (ServoConfig.exe) but may be marginal to inadequate for the more powerful SiTechExe Ascom driver software.

Gear trains are discussed in Section 2.3.4. A general rule of thumb is to shoot for about 10 motor encoder ticks per arc second for general use and very good accuracy. It depends on the slewing speeds you want, and the mass and inertia of your scope as to whether you want more or less gearing.

2.1.2 What are "optional" servo control system components to consider?

The computer is really optional for operations, since all the basic telescope mount control and operations are available in a "standalone" capability, once the controller is configured. However, all of the more sophisticated capabilities are only available using a computer than can run windows software (a PC or a Mac in emulation mode). There is a middle ground, which allows using an ArgoNavis, or a Sky Commander. You can also use a PDA (running Astromist) but will need a very basic computer with a bluetooth receiver that can run an interpreter called LXtoSitech.exe to allow the controller to understand LX200 commands from the PDA via a Bluetooth link, and then the PC connects to the controller.

Clutches between the gear train and the Telescope axis are optional. They are handy when you want to manually grab the scope and move it around without using the motors, but are not required.

A second set of encoders (in addition to the motor encoders) are also optional. These are referred to as either Scope or Mount encoders, and provide a complete closed loop feedback capability for how the mount is actually moving. They will detect errors introduced by the gear train (worm gear periodic error, roller drive slippage, clutch slippage, declutching and moving the scope by hand, etc.). They are certainly not required however, to achieve excellent performance from your mount. There is considerable information in this manual and appendices on considerations for installing mount encoders, since inaccurate installations can actually also add errors that are not present in the gear train errors, with the result the software corrects for them and you can actually decrease accuracy of tracking. So before

adding the expense of mount encoders, have a good plan for how you will mount them so they do not make things worse!

2.2 Design Considerations

How do you plan on using the system?

Most readers already have a scope and mount and are planning to convert their mount to the SiTech GoTo system as an upgrade. But there are also those interested in creating new mount designs with their own specialized features. The SiTech GoTo system can control nearly any size scope/mount in the amateur community, from small equatorials to "yard scope" 36" Alt-Az size installations. The controller can operate servo motors with up to 2.4 amps /servo and 4.0 amps total current draw. The popular Pittman servo motors in various sizes and gear ratios can provide enough torque to operate these scopes and mounts as long as they are reasonably balanced and have low friction bearings.

The generalized following rules apply to all telescope installations:

1. Rigidity and stiffness of the structure is GOOD! Whether visual or imaging you want the scope to remain pointed steady at the target when focusing, looking in an eyepiece or adjusting a camera, or even in this day and age: manual guiding. You cannot be Too Stiff!
2. You need a stable and preferably level mounting on the ground (or floor). A fixed installation probably has all of this, but a portable system that will go to random locations may need special accommodation for soils or terrain and the ability to be "leveled".
3. Ruggedness and robustness of design means minimizing things that can cause problems. The installation of servo gear motors often have the motors "sticking out" where they might be caught with a cable or accidently hit during setup or adjustment. Gears and worms or belts that are open to the night sky can also get dirt or contamination that will affect (adversely) tracking and slewing. Bugs and debris can affect open bearings or wheels in tracks. Cables on GEMs used for imaging are notorious for catching on "things" especially during meridian flips.
4. Balance definitely affects both tracking and slewing accuracy. It also affects the power required to move the scope.
5. Orthogonality or the "squareness" of the Alt-Az or RA-Dec is important to accurate slewing and good tracking. This applies to new mount designs which requires care in making the mount. Installing the SiTech system on an existing mount that is not square is not going to work as well as a mount that is "square". However, the SiTech system has a built in mount modeling tool called PointXP (donated by Dave Rowe), that allows modeling of most mount fabrication errors and will dramatically help in most cases.
6. The servo drive and encoder installations should not have any slippage. Random slippage is almost impossible to accommodate. The SiTech system has the ability to use encoders on the mount axes, and can use them to operate the system in a closed loop approach. Using very high resolution encoders for this can virtually eliminate the effects of periodic error and slippage, but adds somewhat to the cost, and the mountings of these encoders must be reasonably precise or using them can actually ADD to the errors being seen.

The bottom line is the better and more accurately the mount is built, the less the software has to compensate for fabrication errors, and the better the overall performance will be.

2.2.1 Portable vs. Permanent or Semi-Permanent

There really isn't much difference of concerns between portable vs. permanent or semi permanent installations for the SiTech installation. For accuracy, you need a good rigid servo installation and no slippage in the drive system. For portable systems you have to be careful with any servo motor or encoder system to prevent damage during transit or storage. But these items cannot tolerate much abuse in normal use in permanent installations either. Cables and wiring, and the controller box itself should be neat and well labeled to prevent incorrect hookups and unplanned disconnects.

In the past, systems that are disassembled for transport or storage had to be built to be able to be re-assembled VERY accurately to preserve any mount modeling corrections. The ability to use an automated mount modeling routine within SiTechExe that uses a camera and an automated plate solving routine has dramatically reduced the effort and increased the accuracy of mount modeling. The mount still needs to be dimensionally stable when assembled, and well balanced, but after re-assembly and re-collimation, building a new and current model of the mount to be used for error correction is now only a matter of 20 minutes or less.

Piers for Permanent Installs.

The pier should have a large reinforced concrete base to sit on. The larger and deeper, the better. Users have had great results with large diameter, over 20", cardboard tubes filled with rebar and concrete. It's best to plan ahead. A link to an example: http://www.rocketroberts.com/astro/observatory.htm

For temporary use of a permanent pier, you can get away with a few shallow postholes being dug and filled in with concrete. A thick steel plate bolted to the top of the postholes or extra reinforced concrete joining the three or more postholes will work. Again, this type of install is not meant for CCD imaging, as there is some "sway" visible on the chip. The ground may shift over time, particularly if you have over 500 lbs. on the pier. This will disturb the polar alignment.

Piers themselves can be reinforced concrete or a steel pipe with adjustments for azimuth and level on top. Below is a Sky Shed design that has proven to be good for up to about a 16" SCT. The top most plate can be adapted to fit nearly any mount.

Some specs on the pictured pier top:
1/4" thick steel circles, 1/2-13 threaded rods, welded on nuts, thick steel angle to attach it to the pipe. The center of the bottom circle has a long threaded rod attaching it to the bottom for added stability. The pipe is a thick wall standard steel pipe available at nearly any local metal supplier.

If you plan on using a tall pipe, you may want to have a local welding shop put gussets and a plate on the bottom for added stability. The bottom plate can be round or square. ½" threaded rod is typically

used to secure it in the concrete. Keep the gussets as close as possible to the mounting bolt holes for maximum stiffness.

2.2.2 Visual or Imaging

In general, imaging (especially CCD imaging) will require MUCH better performance from your mount than does visual imaging. A CCD detector is so sensitive that every error in tracking is recorded, while the human eye is actually a GREAT integrating detector, and can still provide superb details even if the target in the eyepiece is not perfectly motionless.

With that in mind, you should strive for the best performance possible, but for visual systems your needs may steer you to gear ratios that allows faster slews (i.e. more targets to get to), and reasonably accurate GOTOs. If you are using an eyepiece with a reasonable FOV, the GOTO accuracy can be less than when trying to put a target onto the usually small FOV of a CCD chip.

With the incorporation of mount encoders, the type of drive components really is more a matter of choice than necessity. Roller drives that are notorious for slipping can have that slippage corrected for with medium to high resolution encoders. This can save a good deal of money over using high quality worm gears for example, and allow precision GOTO performance, and very good tracking.

2.2.3 Local Imaging Operations vs. Remote Control Operations

SiTechExe is fully compatible with Ascom Standards so it can operate with nearly all amateur astronomy software and imaging tools. SiTech is actually an Ascom Hub so you don't need to use POTH to connect to other software. This feature allows you to use a PC to control the mount and cameras for imaging and guiding when locally set up (You and the PC near the mount). Since all functions of SiTech operation can be controlled by the software, remote operation is also possible. Remote operation requires some additional equipment, depending on the "degree of remoteness".

For instance remote operation from a nearby location such as another building, room, tent, motor home or car can be done with another PC or laptop connected with an Ethernet cable or wireless network. This allows you to remain in comfort while the mount is working (in the cold, with insects flying around, or whatever). Yet you are close enough to verify getting on target, to monitor cables while slewing, close up the scope and turn power off when done, or even protect it from changes in the weather.

A SiTech mount can be used in a totally remote location, with internet control of the mount and cameras from another city, state, or country. This is a much more elaborate installation, requiring a fixed observatory, additional software and equipment, and probably on site personnel to maintain or reset the equipment. This scenario is very possible with SiTech; however it also requires more equipment and software that is beyond the scope of this manual, so we won't go into detail here.

2.3 Mechanical Considerations

2.3.1 Types of Mounts

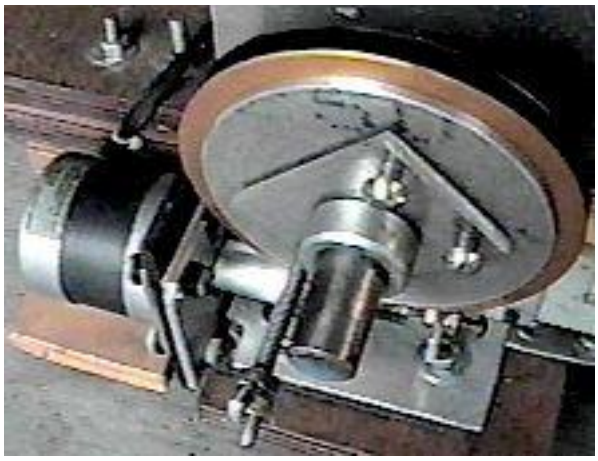
There are only two types of mounts: Alt-Az and Equatorial. However there are subsets of both types that we need to consider in setting up a SiTech controller for your mount. We will list the mount options and discuss the suitability of SiTech for each condition.

Basic Alt-Az Mounts are the simplest mounts mechanically, and they are almost always the lowest cost solutions. Since they have an azimuth axis that is essentially a level flat bearing surface, gravity is evenly distributed on the bearing. With virtually no side load, this bearing can be large, simple, and non precision – often a set of Teflon pads on an ebony plastic or metal surface. The Altitude bearing is also simple, with essentially a constant radial load and almost no side load, simple Teflon pads can support a cylindrical bearing surface at very low cost. These low cost bearings are typical on Dobsonian Mounts where the mount is manually moved and needs low but constant friction. Adapting such a design to use a SiTech system requires the servo motors to overcome the friction on both axes. If you use large ball bearings or rollers instead of sliding pad bearings, the size of the scope that can be used is very large. SiTech systems are used on very large scopes up around Yard scopes (36" scopes).

The three pictures above show different approaches for altitude drives. The top left shows a large, inexpensive 720 tooth HDPE gear with a worm. This provides direct, no slip operations. The top right shows the use of a large disk with a belt drive. This is inexpensive and effective, but if the belt has any stretch can have hysteresis, and potentially slippage. The bottom picture is a roller drive against conventional altitude trunions (ok, these are cool looking trunions that have been designed to be lightweight, but they are still trunions!). The roller that is driven is itself driven by a smaller worm gear. This provides a compact package, and any periodic error in the small worm gear is reduced in amplitude by the roller drive ratio. This type of altitude drive needs to be well balanced and for light weight OTAs, may need to have something to improve mesh pressure for the rollers to avoid slippage (or use a mount encoder on the axis to correct for slippage).

The servo drive systems have to be able to turn the azimuth completely around (more than 360°) but the altitude drive only has to go slightly more than 90°. Drives for GoTo operation need to be repeatable, with enough precision to find the targets. However these scopes are not generally used for long exposure astrophotography, so periodic error is not a great concern. Drives using belts, chains, cables, plastic gears with brass worms and friction drives are common and all work to varying degrees. Success

is often based on the quality of workmanship. The SiTech controller can also use high resolution mount encoders to aid in precision pointing and detect and compensate for any slippage or imprecision in the drives for both Alt and Az axes.



The pictures above show three different azimuth drive approaches. The one on the left uses a belt around a disk (the disk is on the ground board), and silver box is the reduction gearing. Note the clean mounting of this system that is retrofitted to an existing Dob. The picture on the right uses a large HDPE gear to build a “turntable” the rocker box sits on in the same manner as a conventional dob. This eliminates the need for a clutch since the mount can be manually moved on the rotating turntable the same as a normal dob (a mount encoder on the azimuth axis corrects for any movement of the rockerbox on the turntable.

The bottom pictures show a tapered roller under a turntable which is actually the bottom of the rockerbox). The shaft the roller is on has a worm gear with a simple clutch on it (the long thumbscrew tightens it). A tapered roller is much better for a drive roller since it does not “skid” on the drive surface like a non-tapered roller does, and so is more accurate. Alternatively, the drive roller can be very narrow to decrease the effects of the skidding, or use a mount encoder on the azimuth axis.

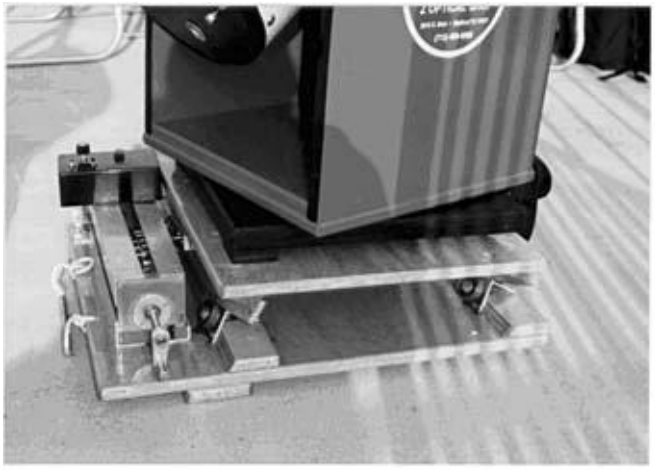
Alt-Az with Rotator Camera can be used for astrophotography. SiTech controllers can control either the scope or a focuser and rotator using servomotors. Two SiTech controllers can be coordinated (connected to the same com port to a PC) to control all these functions. This system allows imaging with very large "light bucket" scopes. The length of exposures is limited by the precision of the scope drives and the rotator. This is probably the best way to perform longer exposure astrophotography with an Alt-Az mount, but current CCD image processing programs can rotate individual images when a series of shorter images are taken and averaged together to correct for the effects of field rotation as long as the individual images do not have rotation smears in the them. The downside is the corners" of the final image end up getting trimmed off.

The above left picture shows an example of an image plane field rotator integrated into the secondary cage of an Alt/Az scope. The focuser is mounted on a small HDPE worm gear, and the worm and servo motor are mounted to a plate that allows the worm to be swung into and out of mesh if desired. It also allows remotely manually rotating the camera to change the composure of a target to better fit the field of view.

The above right picture is of the same telescope, and shows the auto guider scope mounted in a guide scope rotator to eliminate the translation effects of a separate guide scope being used with an alt/az system. When cameras like SBIG cameras with an internal guider, or the Starlight Xpress cameras using their Star2000 units, or when using a 2nd auto guider camera thru an off axis guider, then everything rotates together using the image plane rotator, which greatly simplifies auto guiding an alt/az mount.

Alt-Az with Equatorial Platform is another way to perform astrophotography with an Alt-Az mount. Basically the platform bearings are polar aligned to give equatorial tracking for a period of time, while the Alt-Az mount sits on the platform. The SiTech system can operate an equatorial platform, with a rewind control to be able to reset for another tracking run. However, SiTech doesn't have any way to coordinate Alt-Az GoTo's with the platform equatorial motion. In order to use SiTech control for

manual GoTo operation plus an equatorial platform, two SiTech controllers are required. Achieving polar alignment and smooth, precise motion, not to mention Dec guiding, with a platform can be a real challenge.



Equatorial (Fork, Porter Style (Split ring, Horseshoe), English, etc.) Mounts. Fork mounts are common with commercial SCT scopes and are a viable option for home made mounts. Generally the commercial fork mounts are attached to a "wedge" structure that tilts the fork axis to point to the pole. The fork structure limits access to the eyepiece or camera size for targets near the pole, but otherwise works well at mid latitudes such as in the continental US. SiTech servo control works very well with these mounts, and allow all the features to be used for accurate GoTo's and imaging.

Porter style mounts (Horseshoe, split ring, etc.) are very popular with larger heavier telescopes. Andrew (Drew) Aurigema's 28" f/4.3 (f/1.07 primary) cassegrain system is a good example of a large telescope on a Porter Style mount:

German Equatorial Mounts are very popular for viewing and imaging with moderate size scopes. SiTech servo control provides these mounts with precision operation well suited to long exposure imaging. There are many examples of home made GEMs using SiTech control as well as many commercial mounts converted to use the SiTech system. The features available with the SiTech controller with a PC are much better than any mid range commercial mount, and better than many high end mounts. German Equatorial Mounts have the scope mounted off center and therefore they have the

complexity of control when looking East vs. looking West and the transition called a Meridian Flip. The SiTech controller handles these conditions for you. For imaging, precision tracking and ease of guiding is required. The SiTech controlled GEMs have built-in features to provide periodic error control or to use high resolution mount encoders that can virtually eliminate periodic error in RA tracking. The pictures below are an older Starliner GEM converted to use the Sitech System:

The picture below is of Dan Gay's GEM named Pole Dancer. It was machined from solid blocks of Aluminum by John Pike. Dan's faithful companion "Little Man" is keeping close guard on the mount!

2.3.2 Types of scopes

There are basically three types of telescopes available for amateur astronomy. They are Reflectors (all mirror), Refractors (all lenses), and Catadioptric (combination of mirrors and lenses). Of course there are nearly infinite variations of designs, structures, focal lengths and diameters and these variations give different performance for various observing conditions.

All Mirror Telescopes – Reflectors, like Newtonians come in all sizes for amateur astronomy use. Newtonian scopes are commercially available from about 4-1/2 inch diameter on up. Common sizes for

serious observers are larger. It is not uncommon to see Newtonian scopes at 20 inches diameter up to around 36 inches diameter. These "light buckets" have the ability to "see" very dim objects. These scopes usually have relatively fast focal ratios around 4 to 6, so they have fairly bright images for direct viewing. Primary mirrors vary greatly in quality, since they can be produced by first-time mirror makers or professionals.

The large sizes available mean that the mounts have to be able to handle the size and weight for the scopes. As described earlier, Alt-Az mounts are common for these scopes and work well with SiTech control.

Refractor – All lens telescopes are popular, but almost all are commercially made. Refractors are commonly F5 to F9 focal ratios, although some years (decades) ago f15 was popular. For amateur use, scopes with 4 to 6 inch apertures are readily available, with 7 and 8 inch refractors in use. Refractors vary in quality since each color of light is refracted by glass at different angles. Expensive refractors called apochromatic scopes use multi-element lens made with two or three types of glass to bring all the colors into focus at the same point. Since refractors have the eyepiece or camera at the back end of the scope, most refractors are used on German Equatorial mounts rather than Alt-Az or fork equatorial mounts.

Catadioptric Telescope such as Schmidt-Cassegrain, Maksutov and Ritchey-Chretien use combinations of mirrors and lenses. They are commonly commercially made scopes and range in quality from good to outstanding. Most of these scopes have longer focal lengths, with focal ratios equal 10 or greater. Focal reducers are often used to obtain faster F ratios for imaging. Common diameters for commercial SCT scopes range from about 5 inches to 16 inches. Since the light path is "folded", the structural size is basically a "barrel" or relatively short length. Their compact size makes them good for portable systems. Maksutov scopes such as Questars have outstanding sharp views, but are generally a F14 or so, so visually they are somewhat dim. Ritchey-Chretien scopes are now becoming available commercially in larger sizes, larger than 20 inch diameter, but they are expensive. The RC scopes generally have outstanding optics and very large fields of view with sharp focus.

2.3.3 Considerations if initially building vs. converting existing

Design freedom and design quality are two factors that may influence your decision whether to convert an existing mount or build a new one. If you have a design idea or access to special materials or tools, you may elect to deign your own mount. However if you have a mount that lends itself to conversion to GoTo, then conversion is the way to go. Large Newtonian scopes that will use an Alt-Az mount may need the mount to be newly made to fit the scope, since large off-the-shelf mounts are simply not available. Keep in mind the list of factors listed in Section 2.2 are important for any mount design.

2.3.4 Gear Ratio Considerations

The selection of gear ratio for the servo motor and the overall mount drives determine how the mount will operate when viewing. Said the other way, how you want the mount to operate determines what gear ratio you need. Here's the story.

In general, visual observing puts a premium on accurate GOTOs, and fast slews to many targets each night. High speed slewing of the scope to new targets wants a numerically low gear ratio for very fast slewing. However, for accurate guiding while imaging, high resolution encoder signals and precise servo motor control suggests using numerically high gear ratios. Other factors come into play as well. Large, long scopes and heavy mounts or mounts with significant bearing friction may overload a gear ratio optimized for speed. If you are not going to image, but use your scope visually, you don't need sub arc-second tracking precision. You can have accurate slews, and ok tracking accuracy, or vice versa. (a roller drive that slips under slew loads can track very accurately for example. Or a geared system that does perfect slews may have terrible tracking due to large and abrupt periodic error in the gear teeth). GOTOs for both visual and the often tiny FOVs in CCD imaging both benefit from accurate slewing. This suggests a system that does not slip, or that compensates for any slippage that occurs (more on this later).

The actual gear ratio of each drive (Alt-Az or RA-DEC) is aimed at having the right number of encoder ticks per revolution of the mount axis so that the servo has smooth operation and provides the position accuracy desired. So the encoder ticks are part of the formula for deciding on gear ratio for the system.

The servo motor is controlled by the SiTech controller based on the encoder ticks. The common servo motors in use are the Pittman brand DC motors that have an encoder built onto the motor housing and driven by the motor armature shaft. Typically these encoders are 500 ticks per rev and are read using quadrature as 2000 ticks per rev. But be sure of the encoder type, because there are encoders with 512 and other ticks per rev possible. Since the encoder is connected to the motor shaft, the motor gear box multiplies the servo motor output ticks by the exact gear ratio. For Pittman motors the advertised gear ratio is an approximation. The tables below show the exact gear ratio for many available Pittman motors.

The overall gear ratio and ticks per rev for the mount also have to account for any gearing or drive reduction between the servo motor and the Axis shaft. For instance worm-gear drives are commonly available with 180, 225, 270, 359 and 360 teeth. Friction drives can have non-integer ratios. If you use a belt drive or chain, you need to account for their drive ratio contributions.

GM8700 Ratios Stamped value:		3/16" Dia Output Shaft Numerator	Denominator	Real Ratio	
6.3	:1	693	110	6.300000000	:1
10	:1	17577	1760	9.986931818	:1
19.5	:1	21483	1100	19.530000000	:1
31	:1	544887	17600	30.959488636	:1
60.5	:1	665973	11000	60.543000000	:1
96	:1	16891497	176000	95.974414773	:1
187	:1	20645163	110000	187.683300000	:1
297.6	:1	523636407	1760000	297.520685795	:1
582	:1	640000053	1100000	581.818230000	:1
922.5	:1	16232728617	17600000	922.314125966	:1
1804	:1	19840001643	11000000	1803.636513000	:1

GM9400 Ratios Stamped value:		1/4" Dia Output Shaft Numerator	Denominator	Real Ratio	
5.9	:1	920	156	5.897435897	:1
11.5	:1	1518	132	11.500000000	:1
19.7	:1	36800	1872	19.658119658	:1
38.3	:1	60720	1584	38.333333333	:1
65.5	:1	1472000	22464	65.527065527	:1
127.7	:1	2428800	19008	127.777777778	:1
218.4	:1	58880000	269568	218.423551757	:1
426	:1	97152000	228096	425.925925926	:1
728	:1	2355200000	3234816	728.078505856	:1
1419	:1	3886080000	2737152	1419.753086420	:1
2426	:1	94208000000	38817792	2426.928352854	:1
4730	:1	1.55443E+11	32845824	4732.510288066	:1

You calculate the overall gear ratio by multiplying each individual gear ratio. For example:, your **Final Gear Ratio is the result of multiplying:**

servo motor gearbox ratio * belt/chain ratio * worm-gear ratio * drive roller ratio

Note: use a 1.0 for items you don't have, for instance with no belt, then belt = 1.0.

Multiply the gear ratio by the ticks per rev for the servo motor to obtain tick per rev for the mount axis. This is the value you will need to enter in the ServoConfig.exe software to set up your SiTech system. This number will also determine how fast you can slew and how precise your tracking can be. **Typical recommended total ticks per rev are about 10 million**. This gives reasonable guide/tracking accuracy and fairly fast slewing. The SiTech controller can keep the servo motor within 1 tick most of the time, with an occasional 2 tick error. To evaluate the tracking accuracy, divide your ticks/rev by 1,296,000 (arc-sec per 360°). So 10 million / 1,296,000 = 7.72 ticks per arc-sec.

Slew speeds are limited to about 6,500,000 setting in raw speed in ServoConfig.exe. With your ticks per rev set in that software, typing in 6,500,000 in the Slew Speed box will result in the software calculating your maximum slew speed in degrees per sec. Values of slew speed about 2.5 degrees/sec are common, although you can go faster, if you dare.

2.3.5 Drive Type considerations

Worm/gear, chain, belt, and friction drives are all commonly used for SiTech controlled mounts. They all work, but none are perfect. Gears and sprockets (for chains and belts) all have some errors in the precise form of individual teeth – no two teeth are exactly alike. Most worms turn exactly once for each tooth advance, so any error in the worm is repeated with every tooth. This is the typical periodic error we try to overcome. However each tooth has an error, so the total error at any time is the sum of the individual errors. Cog belts and chains have these errors too. Generally the error in the final drive – the last stage of the drive – is the error that most affects us. Errors in the teeth of gears in the servo motor gearbox are averaged out and are so small that we hardly can see them.

Friction drives are simple to make, but always have the possibility of slippage. When they work without slippage, they work very well. But SiTech using just servo motor encoders relies on no slippage to keep track of current position. Using SiTech with both motor and mount encoders can allow SiTech to detect and compensate for limited slippage, but significant slippage can make the mount unusable. SiTech allows the user to set the allowable error (threshold) between motor and mount encoder readings, but when the threshold limit is reached, SiTech assumes the mount encoder is "truth" and resets its reported position to the position reported by the mount encoders. This puts a premium on accurately installing the mount encoders!

All the drive types need to have clean surfaces to ensure smooth operation. Metal worm/gears need lubrication, so any contamination will mix with the lubricant and cause problems. Chains, too need lubrication which can attract or catch debris and affect performance. Friction drives are perhaps the most affected by contamination, either causing slippage or changing the effective gear ratio as a roller rolls over something. All of these conditions suggest that the drive design should include provisions to keep the components clean and away from contamination sources. It should be noted that ALL of the high end commercial mounts for astrophotography use enclosed worm/gear drives for precision and longevity.

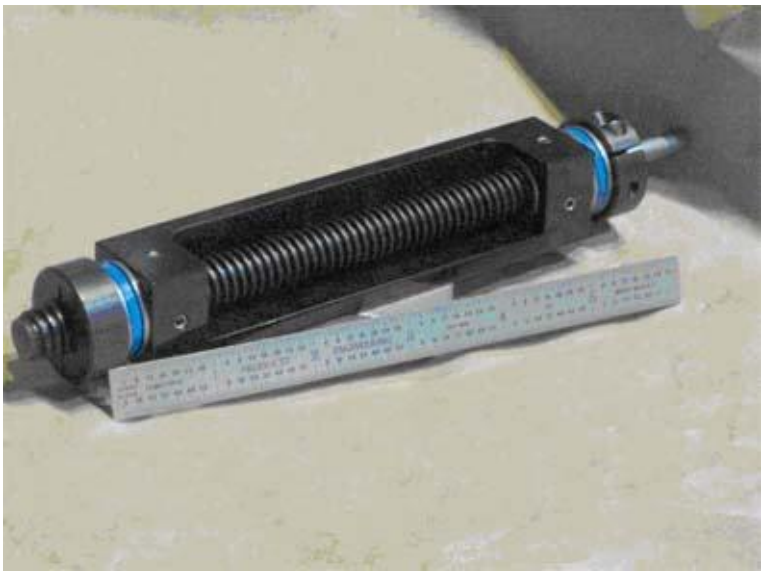
2.3.6 Wheel and worm Gears

If you plan on primarily using your mount for just visual use and not for long exposure imaging, you can use a high density polyethylene (HDPE) type gear on either axis. This type of gear is much cheaper to buy and will provide satisfactory results for or visual use or short exposure, auto guided, imaging use. The gears can be used on a DOB, fork, or equatorial mount. The periodic error (PE) of these gears is usually only a concern at very high magnifications or on a CCD chip. Even then, the PE is not that bad. Costs for the gears can be as little as $165.00.

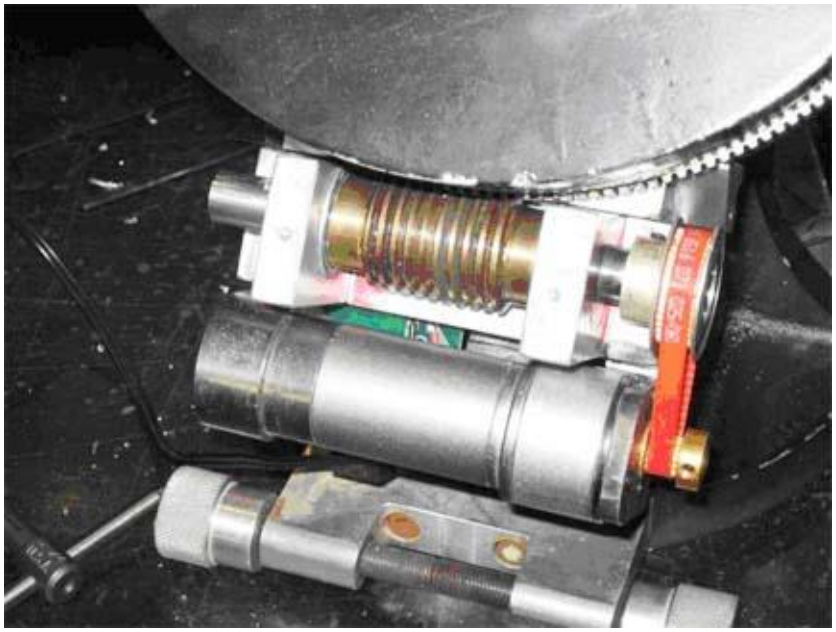
Aluminum gears can be had at a few places online. PE varies; finish, ratio and cost all vary wildly.
www.aeroquest-machining.com
www.gototelescopes.com

Worm mounts
These come in a variety of designs. Some are aluminum, some are solid steel. Some are multiple pieces some are solid one piece designs. A few examples are below.

You can see that the worm shaft is a single solid piece. The mount is a single solid piece of aluminum. The worm shaft is also coated with black oxide. The ends use thrust bearings and ground washers. The bearings are a standard needle roller type. The shaft collars are a clamp on type, not a disc with a set screw. This allows a more accurate placement of the collar to the thrust bearing, and in general for all couplings is a much better approach than using set screws.

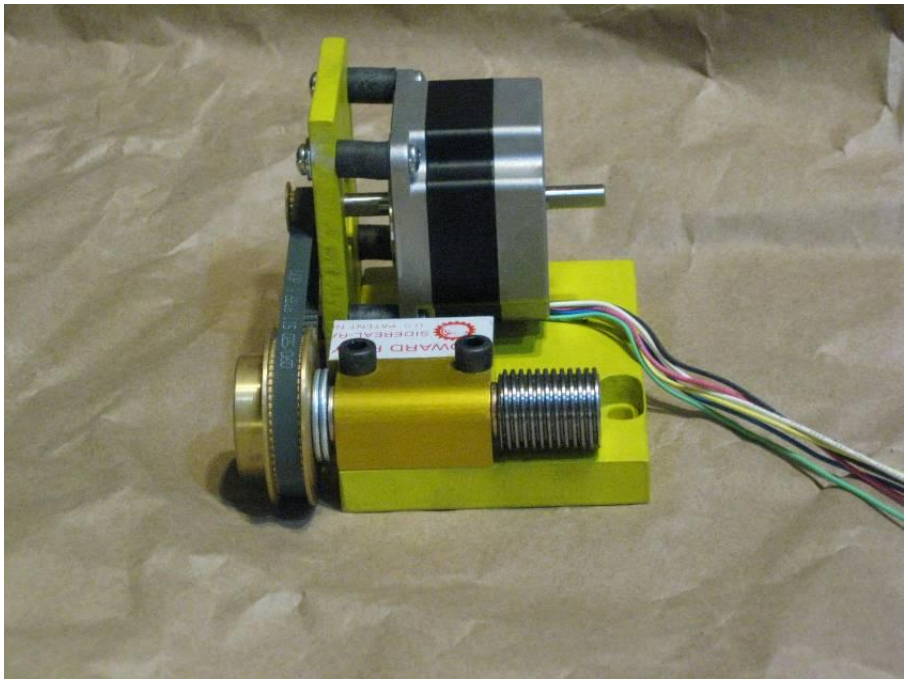


This is a one piece steel worm mount, ½" worm, thrust bearings and clamp on shaft collars. Again, needle roller bearings are used. There is also a small shaft on the end for placing a timing belt.

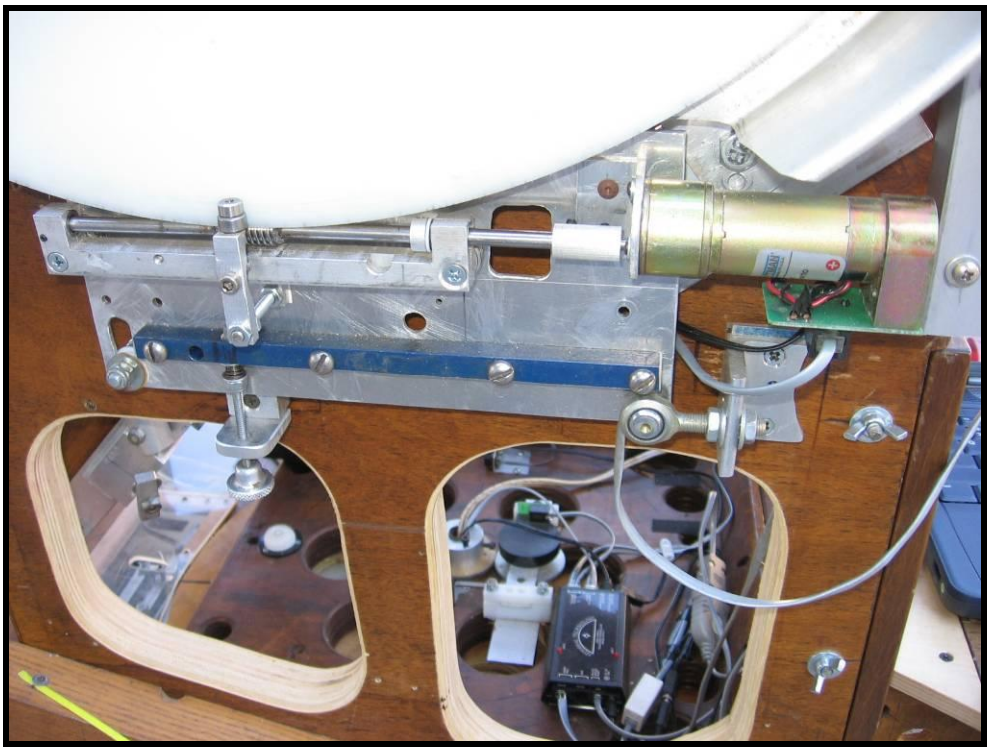


This is an older design of a worm mount. Notice the lack of shaft collars or thrust bearings. The worm mount is a 3 piece Al. design too. This worm is less free to rotate, but it can work. It is much cheaper to make a worm mount like this one.

This is a modified worm mount. Originally, there were plastic bushings inside the gold part instead of bearings. Notice the lack of support on one side of the worm as well. The only available option was to use bronze bushings impregnated with oil (oilite) in place of the plastic bushings.



This is the same mounting with the addition of a larger mounting plate that was made to mount a stepper motor coupled to the worm via a cog belt..



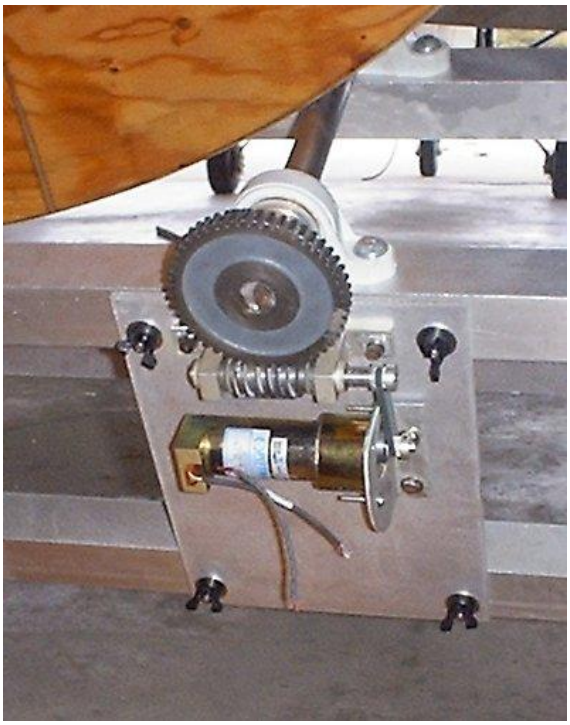
This worm is mounted on a large HDPE gear on the altitude axis of an alt/az mount. Notice it is on a shaft supported in bearings at both ends. Since the HDPE gear is not necessarily flat and it also changes size slightly due to thermal changes, the worm mounting must accommodate those changes. The worm has follower bearings to keep it centered on the gear (inside one is hidden by the gear). The motor is

directly coupled to the worm shaft. Everything is mounted on a flat plate that is a kinematic configuration (a 3 point mounting that uses a ball socket, the gear follower bearings, and a stud with Teflon washers). The stud also goes through a slot in the plate, and there is an adjustable compression spring to keep the worm in mesh with the gear in case the worm is not perfectly round. This mounting, nicknamed the "Blade" mounting, also allows the plate to rotate down out of mesh for easy disassembly for travel.

2.3.7 Gear reductions

One of the most vexing parts of building or converting a mount is finding suitable gear reduction components. Obviously, if you have a large ratio worm gear or drive roller system, you may not have to have any additional gear reduction. However, if you do need more gear reduction, you can add spur gears, or even multiple cog belt reduction setups. Both work well. The better quality of the spur gears, the better the final performance will be, but if they are driving an already steep drive roller ratio, error in the spur gears are diminished out by the roller drive reduction. You can also use an all roller drive reduction, or a compound roller drive reduction. The smaller driver roller however must be as round and run as concentrically as possible to avoid introducing periodic error.

The pictures below are two configurations that have been used on the Porter style mount built by Drew Aurigema. Note the large horse-shoe sector that is driven by the roller shaft to provide the roller drive reduction, and then the two options for driving the roller shaft (compound belt drives on the left and a spur gear drive on the right (Drew reported he favors the compound belt drive for its smoothness)



2.3.8 Clutches

These allow the user to move their scope around regardless of the motor's state. They are not a necessity and can introduce errors in pointing and P.E. Aside from that, a clutch on the gear can cost a lot more than a hub for mounting it to a shaft. If you do decide to have clutches on each axis, do look for Teflon, thin Teflon, at www.mcmaster.com for the clutch material. Thicker Teflon can allow the gear to wobble when rotated. Alternatively, thin, hard, fiber clutches with high quality grease impregnated into them also work quite well (masonite, or fiber gasket material, etc.)

The pictures above show two approaches for simple clutches. Both have the gear "sandwiched" between a hub that is on the backside of the gear that is attached to the axis, and a disk that has an adjustment for how tightly it is clamped. The one on the left is a simple central bolt that is also the axle. The disk is plywood, and the clutch material is cork. The clutch material is not a disk, but is a ring. That allows the plywood disk to deform and still apply even pressure on the gear. The clutch on the right uses a ring that is set-screwed to the axle, and has three thumbscrews that apply pressure to the disk.

This picture shows a Losmandy G-11 declination shaft removed, and the plastic clutch disk being replaced by a fiber disk that has a high quality grease (Slick 50) rubbed into the fiber. This provides a solid clutch function capable of "locking" the axis with a minimum of stiction and clamping pressure (unlike the "standard" plastic clutch disk)

2.3.9 Mount Encoders

SiTech is unique for its ability to use both servo motor encoders along with mount encoders in 4 different modes for control and accuracy of telescope mounts. The first mode is "ignore" and the mount encoders are simply ignored (use this if you do not have mount encoders). Precision mode applies the error limit threshold at all times (slews and tracking). Polite mode only applies the threshold correction during slews, but not during tracking, since that could mess up an image for instance). Tick Management (TM) mode is unique to SiTech. Basically this feature combines both motor and mount encoder ticks while tracking, and adjusts the servo motor speed to keep the mount ticks at a constant intervals. Using very high resolution encoders such as 320K or 500K tick Gurley encoders, SiTech using TM can virtually eliminate period error effects from other sources such as the drive gears and worms. The error limit Thresholds are not used when TM is enabled, since the errors are never allowed to get that large. There is more information in the Operations Manual on TM Mode, and in the appendices of the operations manual.

Gurley encoders obtain their very high resolution by interpolating between optical ticks in the encoder. However the interpolation process is not perfectly linear. An interpolation error pattern has been measured, and seems to be very repeatable, so the SiTech software can sense it and make accurate corrections for the interpolation error. It takes SiTech a couple minutes to measure this error and fully correct for it. The use of Gurley encoders and this correction works for both Alt-Az and Equatorial mounts. The Gurley encoders not only help with accurate tracking, but also help with very accurate pointing and slewing. To obtain the benefits of these medium to high resolution encoders, they must be mounted very accurately, and the mounting cannot introduce side or axial loads on the encoder's shaft. There are examples in the appendices of this Setup manual for Gurley mountings.

2.4 Electrical and Electronics Considerations

How you plan to use your scope dictates what you will have available to power and control your scope and mount. Basic questions are portable vs. permanent installations, and is AC power available? Also for consideration are questions about use of the scope, such as visual only, where power is needed only to slew and track, or imaging with cooled CCD cameras and PC or lap top running with a large power draw. We will take these one at a time.

2.4.1 Electrical Power Source

The SiTech servo system runs on 12 to 24 volt DC. The controller can handle two servo motors with a peak current draw of 2.4 amps each, but limited to 4.0 amps total at any time. Most of the time the mount is tracking with a low power draw, typically much less than ½ amp per axis. When you slew the scope, the motors will draw much more current, but stay below 2 amps per motor. The controller itself takes very little power to run its electronics and the encoders. So basic SiTech operation power can be supplied by a motorcycle battery for a night or more.

The SiTech controller has so much more capability if it is used with some form of computer. The simplest and most low power computers would be an Argo Navis or Sky Commander. Both of these can connect to the SiTech controller and have a built-in database of many objects that allows the scope/mount to locate and slew to desired targets.

2.4.2 To use a PC interface or not?

You have to make the decision about whether to use a PC or not during operations (you MUST use a PC to input configuration settings to the controller however). The SiTech system is versatile enough to work in almost any situation, PC, Argo Navis or Sky Commander or without any connection other than a hand pad. The functions available for controlling the mount vary with each system. Also, the impact on the 'neighborhood" is a concern if you are using a PC or Laptop with a bright screen. If you are going to image, you most likely will need a PC to control the camera, guiding, and review images. Here is the summary:

1. Basic SiTech with No Interface allows manual control of Alt-Az or Equatorial mount with hand pad slewing. Tracking speeds can be adjusted manually with the hand pad for visual tracking with Alt Az mounts. This is the most portable setup and requires the least power (battery size)
2. SiTech with either Argo Navis or Sky Commander interface allows synchronizing SiTech with the sky and using the database available to GoTo slew and track most visible targets available. This works with Alt-Az and Equatorial mounts. With this system on Alt-Az mounts, the tracking is automatically adjusted across the sky.
3. The Full-up SiTech with a PC allows all the tools and features to be used for outstanding observing experiences with: Precision GoTo's and tracking, Periodic Error Correction or elimination of PE with precision mount encoders, Scripted operations, Ascom connection for mount and camera controls, and more.

2.4.3 Computer and Operating System Requirements

The SiTech software system is a Windows Application and runs best on WinXP. Win98 may work, however most current software for imaging and planetarium programs are set up for WinXP. Any

applications that use USB will work better on WinXP (Win98 was the first to incorporate USB and the USB drivers were much improved with WinXP). What about Vista? It seems to work, at least there are few problem reports from users with it.

Mac Users have two choices, either run SiTech in a Windows environment or use Mel Bartle's Scope II software. Scope II is an older program (as of 2009) and is very usable, but doesn't include most of the latest features in SiTechExe. Mel is working on Scope III version, due out late 2009 that will have many more features.

SiTech on Windows requires the Microsoft Dot .Net 1.1 installed. It is free to download at:
http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&displaylang=en
If you want to control the SiTech Telescope Driver with a planetarium program, then you should install the ASCOM version 4.1 from here:
http://download.ascom-standards.org/ascom41.exe

Once these two applications are installed, then you are ready to install the SiTech Telescope Driver.

A note about the ASCOM system. ASCOM 4.1 has many telescope and other software drivers as part of the basic ASCOM installation. After installing version 4.1, download and install ASCOM version 5. ASCOM 5 does not come with any individual drivers – you have to find and load each one you want to use. By loading version 4 first, you get the drivers and they are available when you load version 5. The SiTech drivers are the same for both Ascom 4 and 5.

Detail instructions to load ServoConfig.Exe are in section 3.3. ServoConfig.Exe allows setup of the parameters for the SiTech controller to operate the servo motors, read the encoders, and relate the scope position to the sky (with SiTechExe).

3 SERVOCONFIG.EXE

3.1 Do I really need to use ServoConfig?

3.1.1 Should I use ServoConfig

Should I use ServoConfig if I want to use the controller standalone (no PC connected) or with a PC using the SiTech ASCOM driver (SiTechExe.exe)?

Yes, you must! The following parameters that are **stored** in the controller are **used** by the controller standalone, but are also **read and used** (but not configured) by the SiTech ASCOM driver (SiTechExe):

1. Motor Encoder Ticks Per Rev (Alt/Az or RA/DEC)

2. Telescope Encoder Ticks Per Rev (Alt/Az or RA/DEC)

3. Slew Speeds, Pan Speeds, and Guide Speeds.

4. Motor Directions

5. Telescope Encoder Directions

3.1.2 Should I use ServoConfig if I use only use ScopeII?

The answer is really up to you, but our philosophy would be to configure everything possible in the controller, and make it work in the stand alone mode perfectly, then start using ScopeII. There is a feature in ServoConfig that will transfer all common parameters into your ScopeII configuration, so although not totally necessary to use ServoConfig, it is highly recommended, and certainly isn't a waste of time.

One important note: If you only use ScopeII, you may need to know that many of the parameters of the SiTech Servo Controller are also configured in ScopeII, which overwrites the controller parameters, so you may set a slew speed using ServoConfig, then wonder why the slew speed changes when running ScopeII. In order to make the process simpler, we've included an “Update ScopeII Configuration”, feature, so the time you spend configuring the controller using ServoConfig, won't have to be duplicated when using ScopeII.

Mel Bartels has provided a configuration screen for the servo controller, but I believe it hasn't been updated with the addition of new features after version 1.5 of the controller firmware. Mel is working on a new version of SCOPE III as this manual is published.

If you're not using ScopeII, then you will surely want to use ServoConfig.

3.2 Purpose and Brief Description

The purpose of ServoConfig is to configure the Sidereal Technology Dual Servo Controller with required information about your mount. The user interface is designed to be very user friendly, and is in a separate program than the SiTechExe ASCOM driver to keep both programs focused on their intended reasons for existing (Setup vs Operations) which allows both of them to be made simpler.

This software is supplied for your convenience. You may use it as required. It is free software.

It is supplied as is with no guarantee of any kind. Please use at your own risk.

3.3 Installing ServoConfig.exe

On the CDRom you will have a copy of ServoConfig configuration software, or if not, it is available for downloading at http://www.siderealtechnology.com, or at http://sitechinfo/downloads

We recommend that you use version 1.1 (the latest current version).

The program will be installed into a directory at: C:\ServoConfig. Find the file ServoConfig.exe and make a shortcut to it on your desktop for convenience.

NOTE: The Icons for ServoConfig and for SiTechExe are identical except for the titles, so place them at different spots on your desktop to avoid confusion if you later install SiTechExe.

You will need to provide a serial port connection from your PC to the RS232 port on the controller. If your PC only has USB ports and does not have a serial port, you will need to purchase and install a serial-to-usb adaptor (sometimes referred to as a "dongle"). Follow the instructions that come with your dongle to install it, and then go to the manufacturer's website to download and install the latest drivers. Many times the drivers that ship with these devices on the install CDROM are NOT the latest drivers available. Using the latest drivers will avoid a lot of heartaches trying to get what is ultimately a very simple device to work.

NOTE: These dongles "create" a Comm. Port. You will need to know what the com port number is to configure ServoConfig to use it. To find out, click on Start/Control Panel and double click on System. Then click on the Hardware tab, then on the Device Manager button. Then click on the "+" sign by PORTS, and click on comm Port, and it will show you what number the comm port is that was created.

NOTE: If you plug the dongle into a different USB port, many of the brands of dongles will create another comm. Port (different number than the one created for the other USB port). To avoid having to then also change the comm port setting in ServoConfig and SiTechExe, try to always plug in the dongle you will use to communicate with the controller into the same USB port every time.

At this point, you need to mount and cable up your controller, and provide it with at least 12v power

4 SERVOCONTROLLER

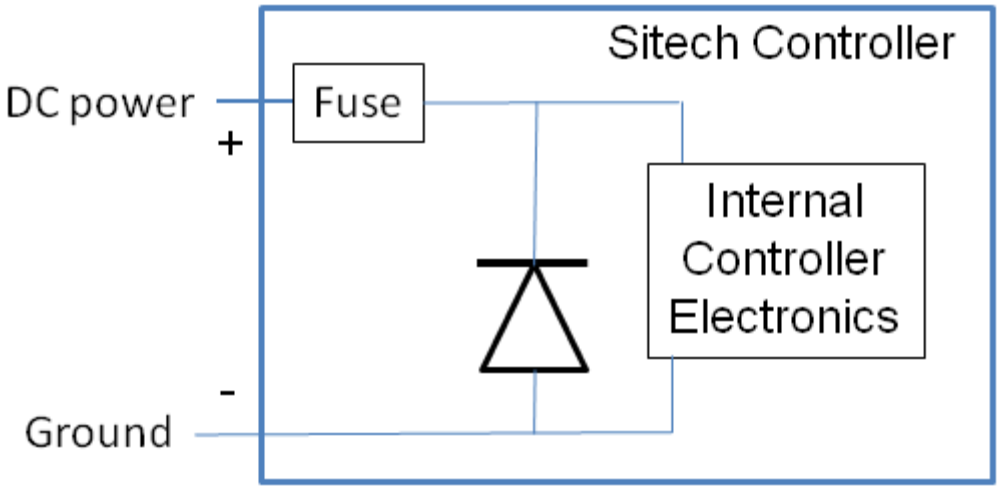
4.1 Mounting the Controller

A piece of Velcro with sticky back is a good way. Mount so both ends of the controller are available for the connections. If exposed, do not mount vertically, otherwise dew or rain may enter the controller.

4.2 Power

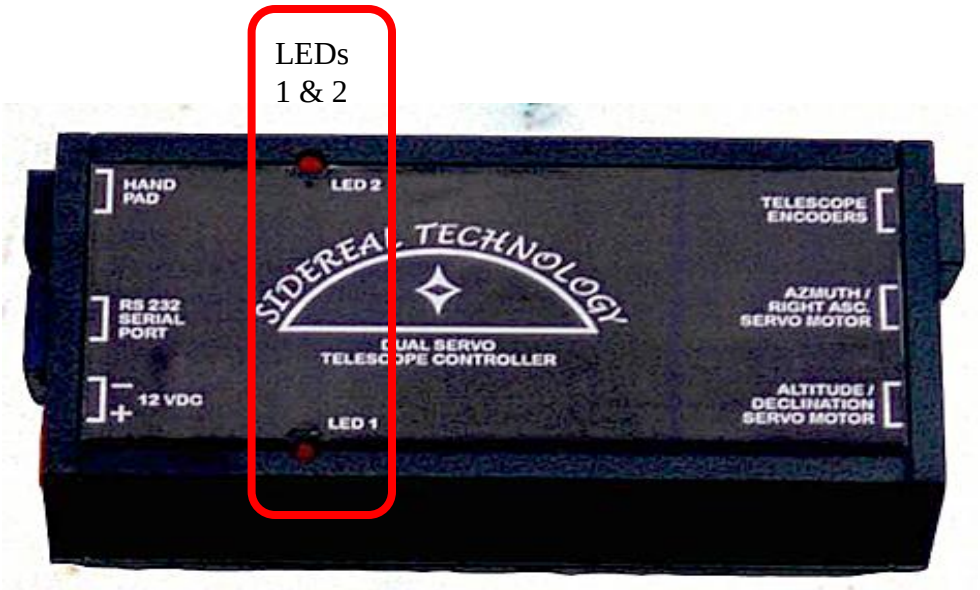
The controller has a power connector/terminal block. Please provide a regulated supply or battery, 4 amp minimum (if full torque is needed) 12 to 24 volt D.C. Supply. A 7 amp hour Gel Cell battery is the best portable way to power the unit. This should last 2 or 3 observing nights before recharging is necessary, if you have a low friction, balanced telescope.

Please observe polarity. The internal protection diode is not wired in series with the power lead as is typical, because the extra .6 volts is crucial to operation of the servo when running at 12 volts because if the input voltage drops even a small amount, there is an internal "H-bridge" that would drop out and the system would not work. It is wired in parallel, after the fuse. If you hook the power up backwards, you will surely blow the fuse, and possibly damage other components. Please don't be the second person to test this!!!!



4.3 LED Operation

There are two LED's on the controller, labeled LED 1 and LED 2. These LED's can keep you informed about what is happening with the controller.



Normally, LED 1 indicates the status of the Altitude/Declination motor, and LED 2 indicates the status of the Azimuth/Right Ascension motor. The motors have two modes, Automatic, and Manual. If the LED has a steady 1 second period blink, then that motor has been forced to manual by a command from the serial port, or by sensing motor runaway. Motor runaway is caused by 3 possible scenarios, Faulty motor encoder or motor encoder wiring, phasing backwards between motor polarity and motor encoder pulses, or motor can't run as fast as it is being told to run. To return to automatic mode, the controller must be powered down, then powered up again, or the command to return to Auto must be given via a command on the serial port. If you are using version 1.6 (or later) of the SiTech Servo firmware, you can also press the top left and the top right buttons on the handpad simultaneously, and both motors will be returned to automatic.

If both LED's are on very faintly, but steadily, there is a serious problem with the CPU in the controller and the unit will probably need to be returned to our factory for repairs. In firmware versions 1.3 or earlier, there was a bug in the software that allowed the flash ROM to be erased under very rare conditions. When this happened, the controller had to be returned to the factory for re-programming. If you have firmware 1.3 or earlier, please upgrade to the latest firmware version. You can download this version from our website.

If the battery voltage falls below 11 volts, both LED's will flash fast.

4.4 Serial Port Connection

The RS232 serial port is an 8 pin modular jack on the earlier model controllers and a 4 pin modular jack on the later model controllers. The pinouts are in the schematics in the appendices of this SETUP manual. Please do not confuse this connector with the motor connectors. It is adjacent to the power connector/terminal block, and on the opposite end of the controller from the motor connectors. Even if you don't intend to use a computer to control your telescope, you will need a serial cable to configure the unit.

Pin 3 of a laptop or PC is the transmit signal and pin 3 of the controller is the receive signal, so the cable can be wired, pin 2 to pin 2, pin 3 to pin 3, and pin 5 to pin 5. It is a good idea to jumper pins 1,4 and 6 together, and jumper pins 7 and 8 together (at the 9 pin female connector) in case special software uses the handshaking on the PC serial port.

If you would like to connect two or more Dual Servo Controllers on the same RS232 communication bus (to also control two focuser motors, or an image plane rotator and a focuser, etc.), simply make up a cable as described above, but parallel another modular jack in parallel with the existing jack. This will require some soldering. This cable can also be ordered from Sidereal Technology (or it's representatives). Also you will have to "assign" the second controller another address. This is described later in the manual.

Note: Mel Bartels Scope II software or Sidereal Technology's ASCOM driver and ServoConfig software do not use hardware handshaking, but if you create special software for the controller, you may have to install the jumpers in the 9 pin female connector listed above.

5 SETTING UP THE MOTORS

5.1 Motor Wiring

Normally the motors will come pre-wired, ready for installation. If wiring is required, see the drawings in the appendix at the end of this document for wiring connections.

Each Servo Motor will have two wires to provide 12v power and ground, and will have 4 more wires that go to its encoder (+5v, ground, channel A, and Chanel B). The later model controllers provide the 12v motor power to each motor via separate plugs and dual wire cables. The controller is marked for where these plugs plug in. The encoders for each motor use wires that have modular connectors on the ends. Each motor that Mel and Dan sell have connectors on a small printed circuit board that the 4 wire modular connector for the encoders plugs into. Then these wires from each of the motors come back to the controller, and both 4 wire cables are attached to a single 8 wire modular connector, and that 8 wire connector plugs into the controller where it is marked Motor Encoders. The pin outs for these cables are shown appendices G and H in this manual.

The earlier model controllers (the earlier controllers used an orange power input plug, and did not have a connector labeled AUX port, and do not have the separate power plugs for the motors) provide motor power and the wires to the motor encoders all in a single 8 wire cable to each motor. The pin outs for this configuration are shown in appendix G. For best noise immunity, the motors should be connected using flat cable, and terminated at the controller end with 8 position modular connector. CAT5 cable will work, but the encoders can pick up electro-magnetic interference from the motor leads, especially if the leads are too long or run in parallel with the encoder leads. The motor encoder cables will be less susceptible to electronic noise (for flat and Cat 5 cables) if "pull-up" resistors are added to the cables for the early controllers (the later version controllers have the pull-up resistors added to the controller). The schematic for adding pull-ups to your motor encoder cables is also in Appendix G. **If you wire your own motors, be very careful with the wiring, as if you mix up the motor leads with the encoder leads, you can destroy the Servo Controller. This is not covered by warranty.**

If you have 24 volt or 36 volt motors, and only have 12 volts available, don't worry, it will still work fine, but the top speed of the motor may not be as fast as you like.

5.2 Motor Configuration

Once the motors have been wired, double check all wiring, including the power polarity. Double check that both motors are free to spin at full RPM in either direction, because it may be that the motor wiring moves the motor in the opposite direction that the encoder is expecting. If this happens, the motor will "run away" at full speed, until the controller determines that the motor is running the wrong way, and shuts the motor off. There will be no ramping, it will instantaneously be at it's fastest RPM for the supply voltage.

Now plug in just the altitude motor, and then power up the unit. This is where you will find if you are a lucky guy/gal or not. If the motor takes off in either direction at full speed and then stops, don't buy any lottery tickets today. Your 50% chance worked against you. You have 3 options. 1. Use the supplied ServoConfig software, and change the motor direction, or the encoder direction (easiest approach) 2. At the motor, swap the motor leads. 3. At the motor encoder, swap the A and B encoder phases. (Note: if you purchased your motors from Mel Bartels or Sidereal Technology, they will be pre-wired and tested).

In 3.0I of the firmware, I made the scope encoder routines more efficient. While doing this, I inadvertently reversed one of the scope encoders. Can't remember which one. This creates a problem when someone upgrades their firmware from 3.0A or earlier to 3.1 or later. Now they have to go to servoConfig and reverse the encoder that I accidentally reversed.

If you have a backwards encoder, the controller always shuts itself off. The problem comes when you break an encoder wire. The motor starts speeding up slowly (it's the integral doing it), until it races away. In firmware version 3.2C, I added code so it will start racing away, but eventually will stop. I did this because the operators of the 32 inch at Pine Mountain Observatory really worried about a broken encoder running the scope into the concrete!

If the motor didn't take off, you should see that it is stopped, and that it's difficult (or impossible) to turn by hand. If this is so, the motor is now being controlled by the servo controller. The motor may be vibrating noisily, or moving back and forth quickly. If this is so, the PID parameters need to be "tuned". See section on "Tuning Your Motors". **Note:** This is not very likely, the default settings work on most motors.

Repeat the above process for the azimuth motor.

6 Hand Pad

The early style controller is compatible with the old style of BBAstroDesign's hand pad. If you have a handpad of this type, be sure to put the 3 position jumper near the hand pad connector toward the micro processor (towards center of the board). If you are using a new handpad type (there is no switch on a new type, there are only push buttons), you need to be sure the jumper is toward the edge of the board.

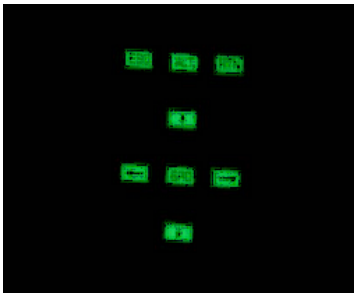
The location of the jumper is shown in Appendix G. One other item if you use the old style handpad, you will have to put negative numbers in the pan and slew speeds for proper operation. If you are using the later model controller with the separate round motor power connectors, then you can't use an old handpad.

The new handpad design allows the movement of both axis's at once. The RF handpad has the same layout and functionality of the new style handpad. The Slew/Pan switch on the old handpad is replaced with a push button. The controller turns the push-button into a "switch" internally, which is used to change between slewing or panning. There are certain functions in the Drag and Track and in the Platform mode that are not available if you don't have a new handpad type.

6.1 Radio Hand Pad

There is a radio handpad available as well. This is a really cool option, it eliminates 1 more cable to get tangled**.** There is a lot more information on installing and configuring and using the RF handpad in the OPERATIONS Manual, but here are some of the features of the radio handpad:

1. **Connects to the Sidereal Technology Dual Servo Controller** for wireless remote control of your telescope. Eliminates the wired Handpad for tangle free operation.
2. **Can be used on Mel Bartels Stepper System:** The receiver has a mode which emulates the original Mel Bartels handpad, so can be used with the Mel Bartels Stepper system with only the application of a jumper and a ground wire on the receiver.
3. **Addressable:** (easily changeable with a keyboard sequence). This allows many users with many telescopes on the same telescope field.
4. **Low Duty Cycle Transmit Time:** Special micro controller routine only transmits about 1/20th of the time, even when a button is continually pressed. There are no transmissions when no buttons are pressed. This allows multiple use from several identical hand pads and telescopes without interruption of any one user.
5. **Instant, unperceivable delay** between a button press and action of the telescope. This is a real plus when centering an object at pan speeds, there will be no overshoot, as the scope starts and stops movement at exactly the same time as you press and release the key (perception anyway).
6. **Built in Micro-Processor Based astronomers LED flashlight.** There are 4 super bright LEDs which work in tandem. The brightness is adjustable from very dim to very bright in 16 linear perceived steps. The brightness is “remembered” across a power down.
7. **Keypad lock feature:** When in the locked mode, only the 4 direction keys work. This is a great (nearly mandatory) feature for public star parties where the handpad is given to the general public for panning the moon at high power (it also helps us more experienced observers at 3:00 AM!!!!).
8. **Glow in the dark keypad:** A lot of research went into finding the perfect glow in the dark base and the percentage used. This glow in the dark is not too bright for even the most hardcore dark sky fanatics! Approval was given by both Chuck Dethloff and Howard Banich (experienced dark sky observers from the Portland Oregon area). In addition to this, the glow lasts all night long, making it easy to find on the observing table in the darkest night. Although the labeling on the buttons may not be discernible, the shape (and thus the location of specific keys) are easily discerned.



2 Second time exposure in a dark room

9. **Keypad:** Reliable, custom manufactured Silicone One Piece Push Button pad, which press against gold plated contacts.
10. **Industry standard, auto guider port** is built in to the receiver. This port can be used to auto guide a telescope, with or without the ASCOM driver or Mel Bartels scope control software.
11. **Auto Off** for both the flashlight (15 minutes) and the transmitter (30 minutes).
12. **Long Battery Life (**many months, if buttons are not inadvertently left pressed). It uses easily replaceable standard 9 volt alkaline batteries.
13. **Wide keypad spacing** for gloved operation.
14. **Transmitter can be used with a direct cable connection** to the telescope controller (in case of interference or if the battery runs down (SiTech Controller Only).
15. **Handy necklace** is included.

7 TELESCOPE ENCODER WIRING AND SETUP

Study the connection drawings in Appendices G and H at the end of this document, and you will see that you need to make a Y cable for the telescope encoders (if cable not purchased and supplied with the system). For the earlier style controllers, a 6 pin modular jack is used on one end of a flat cable. The cable must be fabricated so it branches off to both telescope encoders. The 5 volts and ground must appear at both encoders. For the new style controllers, the modular connector is an 8 pin connector, and the Y-cable is easier to build, and each of the 4 wire flat cables to each of the encoders simply goes into 4 of the 8 pins on the modular connector. Once you have your encoders wired properly, connect your computer to the controller with the serial cable and run the ServoConfig software. On the main screen of the software, you should see the encoder values change as the telescope is moved. The altitude numbers should increase on raising the altitude (or declination with the equatorial scope on the west side "looking east") and the azimuth should increase as the telescope is moved clockwise (bird's eye view, or from Celestial Pole if equatorial mount). If the numbers move backwards, you can change the direction by swapping the A and B encoder signals, or use the ServoConfig software and invert the telescope encoder(s).

The resolution of the telescope encoders can be as much as 500,000 before missing encoder ticks is a problem, even for very **very** fast slews.

8 USING SERVOCONFIG TO CONFIGURE THE CONTROLLER

Once the mechanical configuration of the motors is completed, and the servo controller, power supply, handpad, motors (and motor encoders) are cabled up, and the telescope encoders (if used) are mounted and also cabled up, and ServoConfig.exe is installed and a serial port connection has been provided, its time to use ServoConfig.exe to update the controller with important information about your mount.

You will need the following information written down to be ready to input into the controller:

1. Motor Encoder Ticks per Rev (Alt/Az or RA/DEC)
2. Telescope Encoder Ticks per Rev (Alt/Az or RA/DEC)
3. Slew Speeds, Pan Speeds, and Guide Speeds.

Just to be clear, each servo motor has a small encoder attached to it. This provides closed loop control of the motor, and is referred to as the "Motor Encoder". **These Motor Encoders are not optional, they are mandatory!**

However, the "**Telescope Encoders**" are optional. The system works just fine without them, but if available, the system can enhance its accuracy significantly, and also provide some other capabilities (like Drag and Track, which will be explained in a later section). They are encoders that are attached to the mount axes, and provide information on the actual position of the mount axes back to the controller. Unlike some other systems, there is no separate encoder interface box. These encoders plug directly into the SiTech Controller. They are referred to as Telescope encoders, but sometimes just referred to as "scope" encoders, or "mount" encoders…

The ticks that are entered are for one 360 degree rotation of the axis the encoder is measuring. That means if it is a motor encoder attached to the RA axis of a mount, you will input the total number of ticks generated to rotate the RA axis of the mount a full 360 degrees. Not how much it rotates in a sidereal day, but a full 360 degrees of rotation. Same thing for the telescope encoders, how many ticks to rotate that axis a full 360 degrees.

An encoder "tick" is just that, one "count" from the encoder. Both the motor and telescope encoders are read in quadrature, which means each line on the encoder disk actually generates 4 counts (or ticks). Most (but NOT all) encoders are labeled with the number of ticks they generate in one turn of the encoder disk. So an encoder labeled as "8192" usually has an encoder disk that has 2048 lines on it, and read in quadrature results in 8192 ticks per rev of the encoder's input shaft. A high resolution Gurley Encoder has 5000 lines on its disk, and read in quadrature it would have 20,000 ticks per rev. However, these high end encoders also then interpolate these 20,000 ticks to further divide them up. The 320,000 tick Gurley encoder interpolates these ticks into 16 more divisions, to get 320,000 ticks per encoder revolution. The 500,000 tick Gurley interpolates them into 25 more divisions to get to 500,000 ticks per rev. ServoConfig needs to have entered the Quadrature count for the ticks (or the quadrature/interpolated count if interpolation is also used)

In addition, if the mount encoders are not directly coupled to the telescope axes (like with a cog belt or spur gears), and have a gear ratio other than 1:1 (i.e. up or down) that needs to be taken into consideration.

A number of the servo motors are labeled with 500 for their encoder ticks and some use 512. These are not the quadrature readings, so they have to be multiplied by 4. For motors, if they have a gearbox attached to them, you have to also multiply the ticks by that gear ratio. And if that gearbox is turning a worm that is a 360:1 reduction (a 360 tooth worm wheel), you have to multiply it by that also!

A couple of examples will help:

I have a Pittman servo that is labeled 10:1 gearbox, and 500 count encoder, and I attached it to the 360 tooth worm gear on my mount via a 1:2 cog belt drive. So what do I enter for the RA motor ticks/rev?

500 (encoder ticks) x 4 (quadrature) x 10 (gearbox ratio) x 2 (cog belt reduction) x 360 (worm gear drive) = 500 x 4 x 10 x 2 x 360 = 14,400,000 ticks per rev That's a lot, but it's actually about what you want to shoot for. You want to shoot for at least 10 ticks/arcsecond. This example has about 11.1 ticks/arcsecond. More ticks/arcsec is good, but you trade off slew speeds with a steeper gear ratio. However, if you have a really heavy mount, you may not want it to slew really fast, and want more power to overcome its inertia… Life is a trade, eh?

Important Note: Just to make life interesting, the gear head Pittman Servo Motors labeled 10:1 for the gearbox reduction are actually only 9.98693181818182 : 1, and have a 500 tick encoder. The older and larger gear head Pittmans that are labeled as 5.9:1, are actually 5.897435897436 : 1, and have a 512 tick encoder. A Table of actual gear ratios is in Section 2.3.4.

If you are using a roller drive (or a cable drum drive) for your drive system, it may be really difficult to get an exact number of ticks/rev. Not to worry, there is a really cool "Ticks Determination" tool in ServoConfig you can use to nail the exact value (more on using it later). However, to start out with you need to enter a value that is your best approximation to use the tool. So, measure the diameters of the rollers/drums and use that ratio to start with.

You also want to think about the Slew speed, Pan speed, and Guide speeds you want to use.

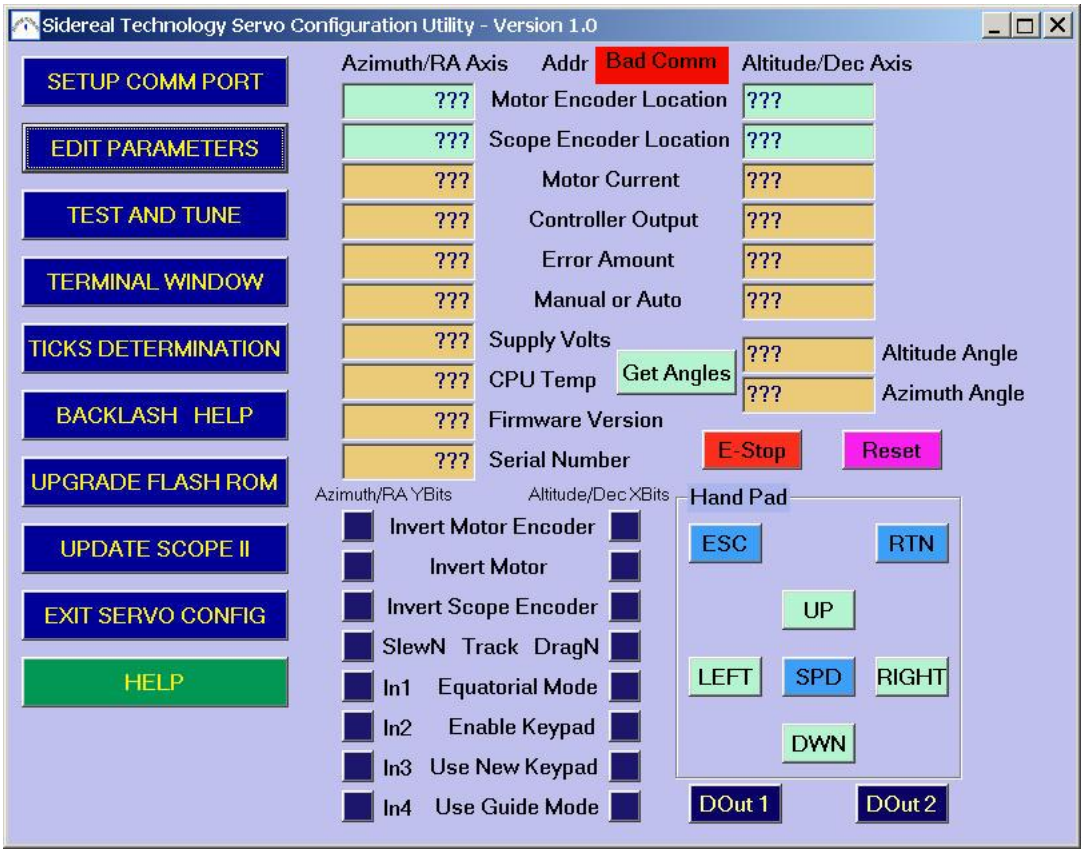
The Slew Speed is the speed the mount will rotate when slewing to another target using a GOTO command or if you are in the "Slew" mode with the handpad and push a button. I would start out conservatively on this, and use something like 1 or 2 degrees per second (dps) and adjust up or down from there. My mounts are set to use 3.5 dps (yes, I am impatient!!)

Pan Speed is what you use when you have slewed to a target, and are tweaking it to the center of the eyepiece or CCD chip. Again, this is personal preference. Too slow and it takes too long, too fast and it's easy to overshoot. I use my G-11 for CCD imaging, so I use a speed of 3 minutes per sec (those are the units ServoConfig will be looking for). That's a bit slow for my taste for visual work, so for my Alt/Az mount that I use for visual observing I set Pan at about 15 mps.

Guide speed is the speed that you would use to manually guide (or have an auto guider use) for imaging, or for visual observing to tweak the position in the eyepiece. If you are auto guiding for imaging, I recommend using a value of about 7 seconds per second (sps, which is the units for ServoConfig). Remember, sidereal rate is only 15 sps, so you do not want to actually back drive the RA gear (and end up with dealing with RA gear backlash if you are autoguiding). For my visual scope, I use a much faster rate of 60 sps. Guide mode actually gets applied on top of the sidereal tracking rate already going on……

So, armed with this initial data you want to enter into your controller, let's start ServoConfig and explore its capabilities!!!!

8.1 Main Screen Overview



This is the main screen of the software. There are 4 main "areas" on the main screen.

8.1.1 Text Box Area

This area is in the top center and right area of the screen, and has two columns of text boxes. The left column is, in general, information on the Azimuth/RA axis, and the right column is, in general, for the Azimuth/Dec axis. Various self explanatory information is updated automatically if there is communication between this program and the SiTech Servo Controller.

If there is no communication, there is a red message that appears at the top of this area to let you know there is "Bad Comm" as shown above. If there is communication, the "module address" (**not the Comm Port number**) is displayed at the top center of the screen

IMPORTANT: While SiTechExe can talk to multiple controllers on the same com port, ServoConfig can only talk to ONE controller at a time. So, if you have two controllers (one for your RA/Dec motors, and one for your focusers for your main and guide scopes for example), you will need to unplug the RS232 plug on the controller you are NOT re-configuring, leaving the one plugged in that you want to configure. Do this BEFORE you start ServoConfig to avoid confusion by yourself or the software..

Hold the mouse over any text box to see additional information about it.

The light green boxes are boxes where you can change the values. Simply type in a new value and press the enter key. This will update the controller with the new value. You can do this with the following:

Altitude/Dec MOTOR Location
Azimuth/RA MOTOR Location
Altitude/Dec SCOPE Location
Azimuth/RA SCOPE Location

The tan background data boxes are just that, data as described by their titles. You cannot enter data into these tan boxes. Data for the Azimuth or RA axis is in the left column, and for the Declination or Altitude axis on the right. Also in the left column are boxes to show the firmware version loaded in the controller, and the serial number of the controller.

If you have set up the Motor and Telescope Encoder Ticks per rev properly, if you click the light green Get Angles Button (near the bottom of the right hand column), the actual telescope angle is returned. This works even if you have no telescope encoders, if you set up the telescope encoder ticks per rev to be the same as the Motor Ticks per rev, and you select the SlewNTrack mode, or the Ignore Scope Encoders option (more on these later).

As you move the telescope clockwise (to the west if equatorial in Northern Hemisphere), the Azimuth/RA angle should increase. In the Southern Hemisphere the motion should be counter-clockwise to the west for increasing Azimuth/RA angle.

As you move the telescope up using the up button (to the north if equatorial in the Northern Hemisphere and the mount is looking to the west), the Altitude/Dec angle should increase. In the Southern Hemisphere, pressing the upper button should move the scope to the South.

8.1.2 Virtual Handpad Area

Also, the same light green color of buttons are on the 4 directional inputs on the handpad indicator. You can click on these four buttons, and the motors will move correspondingly.

To Stop any motor movement, click on the red E-STOP button.

The ESC, RTN, and SPD virtual buttons are indicators only, and have no effect when clicking on them. Also, if you have a handpad receiver, there are two digital outputs available for the system to use. You can test them by clicking on the two buttons labeled DOut1 and DOut2.

8.1.3 X and Y bits status area

In the lower center area labeled “Azimuth/RA Ybits”, and “Altitude/Dec Xbits”, there are “indicators” that turn from dark blue to green to indicate that state described is “on”. For example, if you are using a

new style keypad (most of us are), and to set the controller to know that (which we will do later), the 2nd from the bottom indicator in the right column will be green, not blue. You do not set these states/configurations here, these are only indicators, like the tan boxes they are only data boxes.

If you have an RF handpad receiver, there are 4 digital inputs associated with it. You can view the status of these inputs on the Azimuth/RA YBits section, they are labeled In1 through In4. These may toggle on their own if there is nothing connected to them.

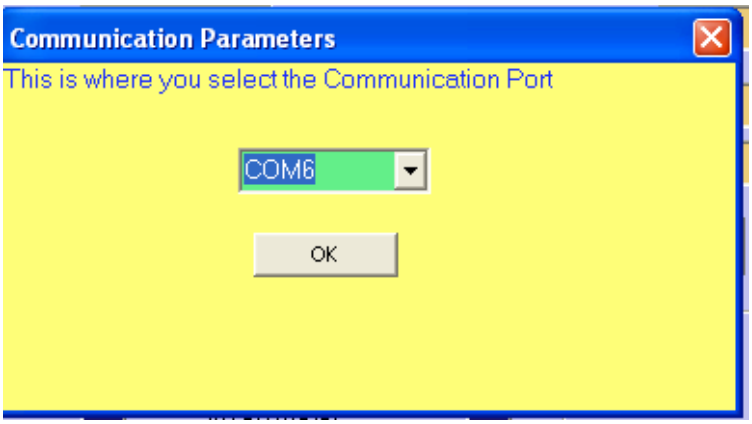
8.1.4 Main Menu Items

The 4th area contains 10 rectangle buttons along the left side of the Main Screen. Each of these (except Exit and Help) will call up another screen that controls the various configuration tools available in ServoConfig. We will introduce each of them in the sections that follow, starting at the top of the column of buttons

8.2 Setup Comm Port

You can select which PC comm port to use with this button. Generally, if the Main Screen is displaying the red "Bad Comm" status at the top, you need to either plug in your RS232 cable, power up the controller, close SiTechExe, or select the proper comm port to use. You can use most USB to serial converters if your computer does not have a 9 pin Serial Port. If you are using a serial-USB dongle and do not know what the com port number is that the dongle created, click on Start/Control Panel and double click on System. Then click on the Hardware tab, then on the Device Manager button. Then click on the "+" sign by PORTS, and click on comm port, and it will show you what number the comm port is that was created.

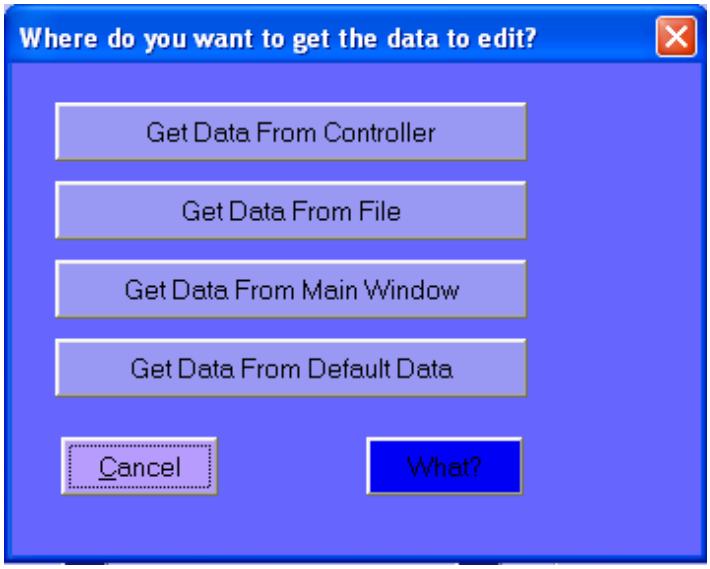
NOTE: If you plug the dongle into a different USB port, many of the brands of dongles will create another comm port (different number than the one created for the other USB port). To avoid having to then also change the comm port setting in ServoConfig and SiTechExe, try to always plug in the dongle you will use to communicate with the controller into the same USB port every time.



Use the drop down menu to choose the correct comm port to use, then click OK

8.3 Edit Parameters

When you select Edit Parameters, you must choose where the data you want to edit will come from. You have 4 choices. The Controller; a file you've previously saved (the file extension will be .car); the main Window; or default data. A screen pops up for you to make the selection:



The data that populates the windows in the Edit Parameters Tool will come from whichever of these sources you pick. **Most of the time you will use the "Get Data From Controller" option.** Select this option as we go through initially setting up your controller.

<u>Something VERY important to remember about the Controller:</u>

There are functionally three areas of memory in the controller from a user's perspective:

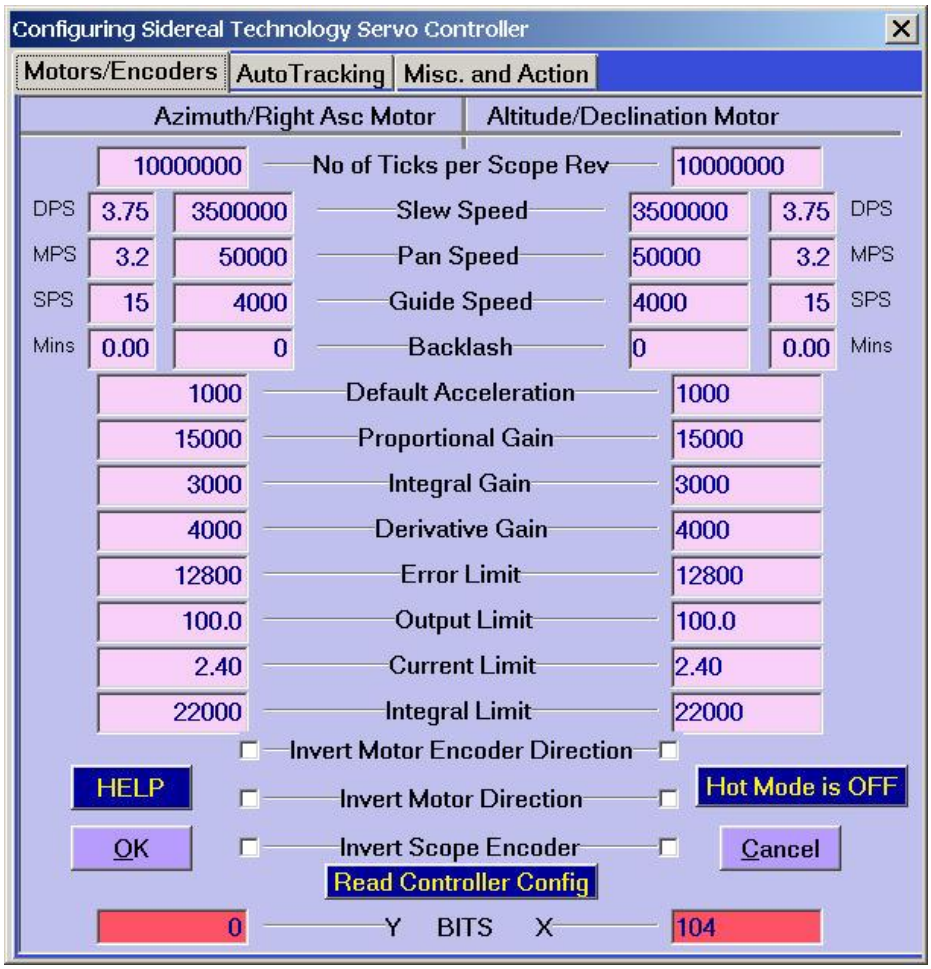
FIRMWARE This is in the Flash ROM. It contains the instructions for the controller to act like a controller, so as new SiTechExe capabilities are added, occasionally this firmware must be updated to work with them. Use the "Update Flash ROM" tool discussed later to update the Firmware.

Controller FLASH ROM Part of the Flash ROM (Read Only Memory) is used to store the mount unique information needed to actually operate the mount. Things like Ticks/rev, and slew/Pan/Guide speeds, PEC tables, motor tuning information, etc. When the Controller is powered up, it reads the Firmware and the mount unique configuration data from ROM, and places it into RAM (Random Access Memory). Flash ROM does not go away when the controller is powered off.

Controller RAM This is the "working" are of memory, when you uplink changes to the controller, they go initially to RAM, and **<u>you must remember to SAVE them to the Flash ROM</u>!** If you do not, they will go away when you power off the controller. .

8.3.1 Motors/Encoders Tab

Note that this screen has three "tabs" along the top. There is a lot of capability to be able to configure using the Edit Parameters Tool and its divided into three main areas as shown on the tabs. Lets start with the Motor/Encoders Tab (which is the default that comes up). The overall layout has two columns, the left one for the Azimuth/RA axis, and the right column for the Altitude/Dec axis.



IMPORTANT NOTE ABOUT SERVOCONFIG OPERATION

If you are using version 0.9 (a pre-release version) **you MUST click on "Read Controller Config" (at the bottom center of the Motors/Encoders Tab) AFTER clicking on "Edit Parameters" on the Main ServoConfig Screen.** If you don't, you will be editing default values, not what is currently in the servo controller. (It's OK to still click on the Read Config if you use 1.0 or later)

Once you've made the changes you want, **you MUST click on the "Misc and Action" tab and "Send Configuration to Controller"**. Many operation descriptions in this document assume that you will be doing this, and won't remind you again. Don't forget this if you are using version 0.9. If you are using version 1.0 or later, you will get prompts to remind you to do this. At any rate, be sure values you are editing, and the changes you make are from the controller, and go back to the controller.

OK, let's take a minute to introduce some "Programmer-Lingo", since it will crop up occasionally.

A structure is simply an area of memory where the parameters are stored, in the controller, or in this program. Think of them as sets of data. As far as configuration goes, there are 4 different parameter structures (aka set of data) relating to this software (Main, Configuration, RAM, and ROM). So if you get asked whether you want to write over this "data structure" it's a programmer's way of asking if you want the "data" being displayed to be replaced with data from some other set of data, like from a file for instance, etc.

It's also important to note that any changes made here without using the hot mode, and also using it properly are only made in the configuration structure (i.e. on the display), not in the controller. You will have to "Send Configuration to Controller" from the "Misc. and Action" page to send and save the parameters to the controller.

While not mandatory to use, one of the most useful features is a button called "Hot Mode" in the lower right hand part of the window. Click on the button to turn the Hot Mode on or off. This button turns Red when in the "Hot Mode". When in the hot mode, every time you press the enter key on your keyboard after editing a number in a magenta window, the number is sent to the controller's working memory (RAM.).

<u>Two things to note while in the HOT MODE:</u>

1. The data goes to the controller RAM, not the controller FLASH ROM. RAM is volatile, which means it goes away if the controller loses power. FLASH ROM is not volatile, so is there even with the controller is powered off. When the controller is powered back up, the information in the FLASH ROM is read into the RAM to be used. To save to the FLASH ROM, you must select the "Misc. and Action" tab, and select the item "Save controller configuration to flash ROM".

2. The number will not be sent to the controller unless you press the Enter key. Pressing Tab or clicking the mouse in another text box will not send the data to the controller.

Clicking the OK button will return you to the main menu, and the data you've edited will be sent back to the main structure.

Clicking the Cancel button will return you to the main menu, but the data you've edited will be lost (unless you saved it to the controller or to a file using the "Misc. and Action" page).

All of the Magenta colored text boxes can be edited. The red boxes at the bottom are the XBits and the YBits and are automatically updated when you select the options using the radio buttons. They are the language the controller talks in with Dan for troubleshooting or working on things, or just when Dan sits and talks with a Controller (kinda scary to think about that one, eh?)

So now, its time to actually start entering the mount unique data into ServoConfig's Edit Parameters Tool, and then uplink it to the Controller RAM, and then save it to the Controller ROM:

No of Ticks per Scope Rev

The Top numbers are the number of MOTOR encoder ticks for one complete revolution of the scope axis. You can get this accurate the first time if you have an all geared system. But for roller drives or wire drum drives, enter it based on the best measurements you can make, and plan on using the Ticks Determination Tool to refine it later.

Important Note!!!!!

Editing the Motor encoders and the scope encoders is probably the most important item in the configuration activity!!. If these aren't set properly, none of the other items, such as slew speed degrees per second, the Local Search Radius and Speed, and many other items will be displayed properly!

Speed Text Boxes

The numbers with DPS, MPS, or SPS beside them are speeds in Degrees Per Second, arc Minutes Per Second or arc Seconds Per Second. For these to be accurate, you must first enter the number of MOTOR encoder ticks per full revolution of the scope in the top location.

As far as these speeds are concerned, you can enter them in DPS, MPS, or SPS as indicated, or you can enter the raw numbers the controller uses. Whenever you enter one or the other of these numbers, the other number is updated automatically. As an example if you have 5,000,000 motor encoder ticks per revolution, and you specify 5 DPS, the raw number will change to 2,330,318. Inversely, if you enter 4,000,000 in the "raw" box, the DPS will change to 8.58 degrees per second (better wear a hardhat while slewing your scope!!!!).

Note: It is important to stay below about 6,500,000 raw Slew Speed, regardless of your gear ratio. If your desired Degrees per Second raises the raw speed to greater than this, manually put the raw slew speed values to 6,500,000. With typical ratio's, you will have plenty of slew speed at this number; it will often be more than 7 deg's per second.

A handy way to see what the maximum slew speed is for your gearing is to enter 6,500,000 in the raw speed for slews, and the rate is calculated for you. Cool, eh?

The slew and Pan speeds are the speeds the motors move when slewing or panning with the handpad.

The guide speed is a bit different. Instead of just making the motors move at a given speed, this speed is added (or subtracted) from the current speed. If a motor is tracking, and the guide mode is used, a motor may simply slow down and speed up, rather than actually change directions. This makes it easier for an auto guider to obtain accurate control. To avoid back driving the RA gear and introducing backlash into your RA guiding, we recommend staying below the sidereal tracking rate of 15 arcseconds per Second. A good guide speed is usually about 7 arcseconds per second.

BTW, the guiding and handpad inputs have undetectable delays. They are nearly instantaneous! The only delay is caused by the switch de-bounce routines, and is on the order of milliseconds.

One more item, there is a feature that changes the Pan and Guide rates for the azimuth/RA as you raise in altitude/declination. This feature is called Equal Area Panning and Guiding. What happens when you

near the zenith (or pole on an equatorial mount), the azimuth has to move a much farther distance for an equal area of sky while you are pointed at the horizon (or celestial equator). In version 1.7, the pan speed and the guide speed are automatically increased as you move toward the zenith (or celestial pole). This is a truly cool feature, and when experienced, you won't want to go back to the old way!

For this to work properly, the controller needs to know the altitude (or declination). If you are using the DragNTrack mode, the controller already knows the altitude. If you have an Alt/Az telescope, but are using ScopeII or the Argo Navis®, you might consider configuring the scope for the DragNTrack (or SlewNTrack) mode, and point the scope at the celestial pole before turning the power on the controller. The DragNTrack (or SlewNTrack) mode will be run just long enough for the telescope to initialize the altitude, before detecting ScopeII or the Argo Navis® and dropping out of the DragNTrack (or SlewNTrack) mode. If you are using the ASCOM driver, you don't need to worry about this, as the ASCOM driver will update the altitude. Mel Bartels has promised to add this to ScopeII in the near future. If you are using the equatorial mode, you will need to set the declination to 0 before powering up the controller for this option to work properly.

Backlash

The backlash settings for both AZ/Ra and AL/Dec can be set here. You can also set the speed for the backlash on the “Misc. and Action” tab. Mel’s ScopeII and the ASCOM driver also have this feature, however be sure to use **ONLY ONE** backlash control at a time (i.e. not in both your auto guider program and also in the SiTech Controller). Use the Backlash Help Tool to refine your estimate of your backlash later. For now you may want to leave it zero.

Acceleration

The Acceleration is the rate at which the motor ramps up to speed. It's best to figure what works best using the "TEST and TUNE" main menu item. Lower numbers create a very slow ramp. Higher numbers make a faster ramp. I like my telescope to be about 1000 to 1500. Heavy telescopes, or ones that are really long have a higher moment of inertia, and your gear train will appreciate having a slower acceleration/deceleration.

Proportional, Integral, and Derivative (PID)

The Proportional, Integral and Derivative are motor tuning parameters. Again, these are best adjusted using the "TEST and TUNE" main menu item. There is a more complete description of these parameters on the TEST and TUNE dialog box. The default values of 15,000-4000 work for most all motors.

Error Limit

The error limit is the point where the controller stops the motor if the motor is not moving properly. The ERROR is the difference (in motor encoder ticks) between where the motor is, and where it should be. If this difference exceeds the ERROR LIMIT value, one of two things happens:

1. If the motor is going in the proper direction at a fairly fast rate, the controller "decides" that things are moving along fine, but the motor just can't keep up. Instead of stopping the motors (like most servo chips), the SiTech controller "Holds" this error, and keeps your telescope moving. The motor position setpoint is held to the motor position plus (or minus) the error limit value.

2. If the motor is not moving fast enough, or is moving the wrong direction, as soon as the error reaches the value of the error limit, it stops the motor and puts that particular motor in MANUAL mode.

The advantage of this scheme is when your battery voltage is going a bit low, or there is a lot of friction or inertia, your motors won't stop, and need to be re-initialized. This allows a more rigorous setting, i.e., you can run your motors at top speed, without fear of stalling. Initially leave it at its default value, but raise it if you have problems.

Output Limit

If you are using a 12 volt motor, but supplying the controller with 24 volts, you will want to put an output limit on. The default is 100%, which is full output. With a 12 volt motor supplied with 24 volts, you would want this value to be 50%.

Current Limit

The controllers are rated at max 2.4 amps, each motor, and a maximum of 4 amps, both motors combined at the same time (i.e. if both are set to 2.4, the controller can only provide 4.0 instead of 4.8). You can make the current limit be less than the 2.4 amps default, by lowering this value. The value 2.4 represents limiting the current to 2.4 amps. You will have less torque when lowering this value.

Integral Limit

The Integral portion of the PID control adds more and more to the output over time, until the motor reaches the desired position or the set point. Sometimes this can go too far, and it takes too long to "un-correct" when slowing back down. To solve this problem, you can limit the integral windup using this parameter. The default is 22,000.

8.3.3 Equatorial and GEM Hemisphere Invert Button Setup

Use the “Invert” buttons for encoder and motor direction as needed to make your mount move in the proper directions for the hemisphere you are in.

Northern Hemisphere Invert Button Setup

1. While using ServoConfig, manually point the telescope toward the Western horizon. Make sure pressing the up arrow moves the scope North (and the encoder counts increase). If this is not correct, change the direction using the appropriate “Invert” button(s). To change the direction, be sure and change both the motor and the motor encoder configuration to the opposite of what they are.

2. Make sure that pressing the right button moves the telescope even more west (goes towards below the horizon if still looking at the western horizon) and the encoder counts for AZ/RA increase. If this is not correct, change the direction using the appropriate “Invert” button(s). To change the direction, be sure and change both the motor and the motor encoder configuration to the opposite of what they are.

Southern Hemisphere Invert Button Setup

1. While using ServoConfig, Manually point the telescope toward the Western horizon. Also be sure to set the proper Latitude in the ServoConfig /AutoTracking tab. Make sure pressing the up arrow moves the scope North (and the encoder counts increase). If this is not correct, change the direction using the

appropriate “Invert” button(s). To change the direction, be sure and change both the motor and the motor encoder configuration to the opposite of what they are.

2. Make sure that pressing the right button moves the telescope even more west (goes below horizon if still looking at the western horizon). If this is not correct, change the direction using the appropriate “Invert” button(s). To change the direction, be sure and change both the motor and the motor encoder configuration to the opposite of what they are.

Invert Motor Encoder Direction Radio Button

This makes the numbers decrement instead of increment on encoder pulses (or vice versa) for a given motor direction. If your motor runs away quickly then stops, you must either re-wire it, or select this radio button without selecting the **"INVERT MOTOR DIRECTION"** button.

Invert Motor Direction Radio Button

This puts out opposite polarity to the motor for a given error. If your motor runs away quickly then stops, you must either re-wire it, or select this radio button without selecting the **"INVERT MOTOR ENCODER DIRECTION"** radio button.

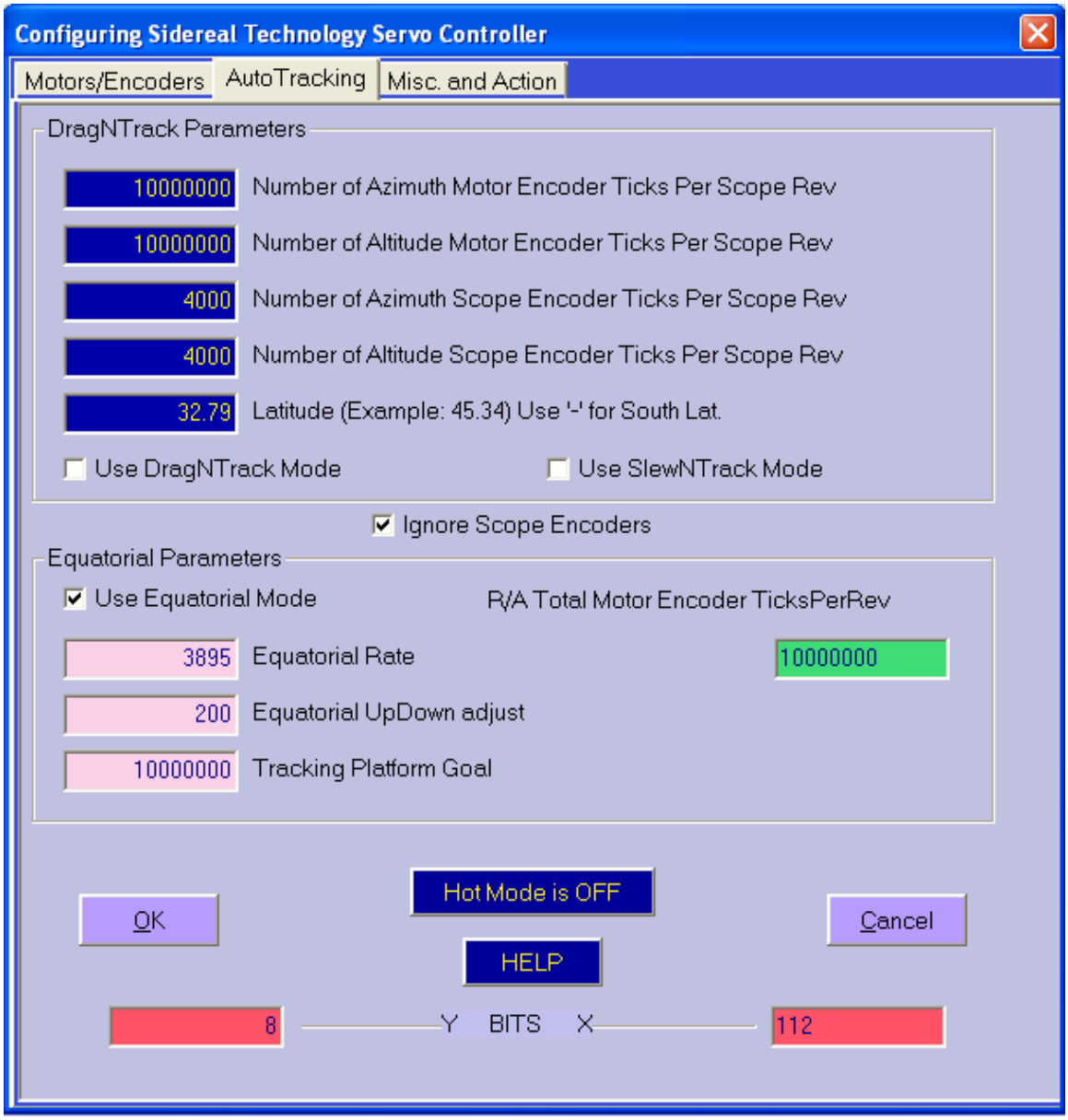
If your motors work properly but move the wrong way, then change the current state (select or de-select) of **BOTH** of the above items.

While selecting or de-selecting the above items, especially in the "HOT Mode", it's possible that the motors may run away, and then stop in the manual state. This will be indicated by flashing lights on the servo controller. To go back to automatic mode, save your parameters to flash ROM using the "Misc and Action" page, and reboot the controller by selecting "Reset the Sidereal Technology Controller". Alternatively, you can put the controllers back to the AUTO mode by going back to the main menu, selecting "TEST AND TUNE", and selecting the "AUTO" radio buttons.

Invert Scope Encoders

Selecting or de-selecting these radio buttons will make the SCOPE Encoders count the other way.

8.3.4 Auto Tracking tab Dialog



This screen also has the Hot Mode button to make configuring quicker. Two things to note while in the HOT MODE:

1. The data goes to the controller RAM, not the controller FLASH ROM. To save to the FLASH ROM, you must select the "Misc. and Action" page, and select the item "Save controller configuration to FLASH ROM".

2. The number will not be sent to the controller unless you press the Enter key. Pressing Tab or clicking the mouse in another text box will not send the data to the controller.

Clicking the OK button will return you to the main menu, and the data you've edited will be sent back to the main structure.

Clicking the Cancel button will return you to the main menu, but the data you've edited will be lost (unless you saved it to the controller or to a file using the "Misc. and Action" page).

All of the Magenta colored text boxes can be edited. The red boxes at the bottom are the XBits and the YBits and are automatically updated when you select the options using the radio buttons.

8.3.5 Drag And Track Mode

The DragNTrack mode is used primarily with Alt-Az mounts if you would like to push your telescope by hand, or slew with the handpad, but don't need super-accurate tracking or GoTo capabilities, but would like computer-less accurate tracking. The DragNTrack or SlewNTrack mode may be a nice feature to use. If you have a telescope of aperture 16 inches or more, it may be more cost effective, and there are many other advantages to use the dual servo controller and the Drag and Track mode instead of a tracking platform.

Note: If you have an equatorial mount, you will use the Equatorial mode, not the DragNTrack or SlewNTrack mode.

To use the drag and track mode, your Alt/Az telescope must have the following hardware (in addition to the servo controller):
1. A pair of servo motors to drive the telescope in altitude and azimuth.
2. Clutches on both altitude and azimuth connected between the motors and the telescope, so you can "drag" the telescope anywhere in the sky, without damaging the servo motors. (Note: you can use the DragNTrack mode, even if you have no clutches, but do have telescope encoders. You will have the disadvantage of only slewing with the handpad, instead of pushing the telescope by hand).
3. A pair of encoders mounted to your telescopes' altitude and azimuth axis's.
4. A handpad.
5. A 12 -24 volt power supply.

A computer is not required for operation, only to configure your servo controller.

Drag N Track Parameters

For Drag and Track to work properly, the SiTech controller must "know" the following:

1. How many motor encoder ticks per complete revolution of the scope for both Altitude and Azimuth.

2. How many Scope encoder ticks per complete revolution of the scope for both Altitude and Azimuth.

3. Where the scope is pointed in Altitude and Azimuth.

4. Your observing site's latitude (only required to make initialization easier).

When you calculate the encoder ticks, it's important to understand the resolution of your encoders. Generally speaking, the effective number of ticks is 4 times the resolution of the encoder. This document uses the effective encoder ticks unless otherwise stated. If your encoders have 2048 as part of the part number, the encoder wheel has 2,048 lines on it. The effective encoder ticks will be 4 times this, or 8,192.

Using the ServoConfig software, and the "Edit Parameters" button, be sure all of the motor and encoder ticks per revolution are set properly (in effective encoder ticks). Also, be sure that the Right button on the handpad moves the scope clockwise, and the Up button on the handpad moves the telescope up. If not, change them using the features of ServoConfig. Now select the "Auto Tracking" tab. Now click on the "DragNTrack" radio button.

For the DragNTrack to work properly, it's important to note that the UP handpad key needs to move the telescope up in altitude and the RIGHT key needs move the telescope azimuth in a clockwise direction. It would be easy to put a negative number into the slew and pan rates to obtain these results, and the controller would respond with the opposite direction, but the trigonometry would still be wrong.

It's also important to note that when the telescope is moving clockwise, the azimuth telescope encoder should be increasing, and when the telescope is moving up, the altitude telescope encoder should be increasing. If these move backwards, then the direction needs to be changed using the ServoConfig software, or change the wiring of the encoders.

Note: If using an old BBAstroDesign handpad (the kind with a switch in the middle instead of a push button) and Servo firmware version 1.5 or earlier, the slew and pan speeds **MUST** be a negative number, but the direction of motion must still be the same as listed above (unless using servo version 1.6).

Operation of Drag and Track Mode

If you want the scope to start tracking without initialization, point the telescope at the celestial pole, and then turn the power on the controller. At this point, if you've already set up and saved your latitude, the system will be initialized. It will immediately start tracking. No further initialization is required (while pointing at the celestial pole, the scope won't appear to track because the celestial pole doesn't move!).

If your latitude isn't set up and saved to flash ROM, you will have to initialize at the scope zenith and the celestial pole. To initialize the Altitude, move the scope (by hand or with the handpad) up to the scope zenith. It could be convenient to have a mechanical stop mounted on your scope, so it is easy to find this location. Press and hold the **top right** (RTN) key for more than 4 seconds. At this point, the controller adjusts your altitude so it reads 90 deg's. (Important Note: there was a bug in the controller flash ROM in versions 1.5 and earlier, and this number was calculated wrong. If you have firmware version 1.5 or earlier, please upgrade to version 1.6 (or later) or enter the latitude using ServoConfig).

To initialize the azimuth, move the scope to the celestial pole. Now press and hold the **top left** key (ESC) for more than 4 seconds. At this point, the controller initializes the azimuth, and if you haven't initialized the altitude, it initializes the altitude to your saved latitude, otherwise it initializes the latitude.

If you have initialized both the altitude and azimuth, you may save the new latitude value by pressing and holding both top keys for more than 10 seconds. This saves the latitude to the flash ROM, so next

time you can simply initialize the scope on the celestial pole, and press the left hand key for more than 4 seconds.

To perform photography, or for very fine centering, it may be desirable to enter the guide mode. This is done by holding down the top right key, then pressing the right key. Now the speed changes by adding or subtracting the guide rate from the current tracking rates. To revert back to the normal mode, hold down the top right key, and press the right key again. This feature doesn't work with the old BBAstroDesign handpads with a center switch. If you are using servo version 1.6 and you have a radio handpad receiver, you will toggle through slew/pan/guide modes using the SPD button.

You may slew the telescope with the handpad, or move the telescope by hand by loosening the clutches and moving and retightening them, and when done slewing or moving, the telescope will immediately start tracking again, at the new drive rates based on the telescopes altitude, azimuth and the latitude. To stop tracking, press the top right key. To start tracking again, press it again. When the tracking is stopped, the LED's will blink off briefly every few seconds (version 1.6 of firmware and later).

Review of Initialization options

It may be confusing, because there are several ways to initialize the telescope. The user has three choices.

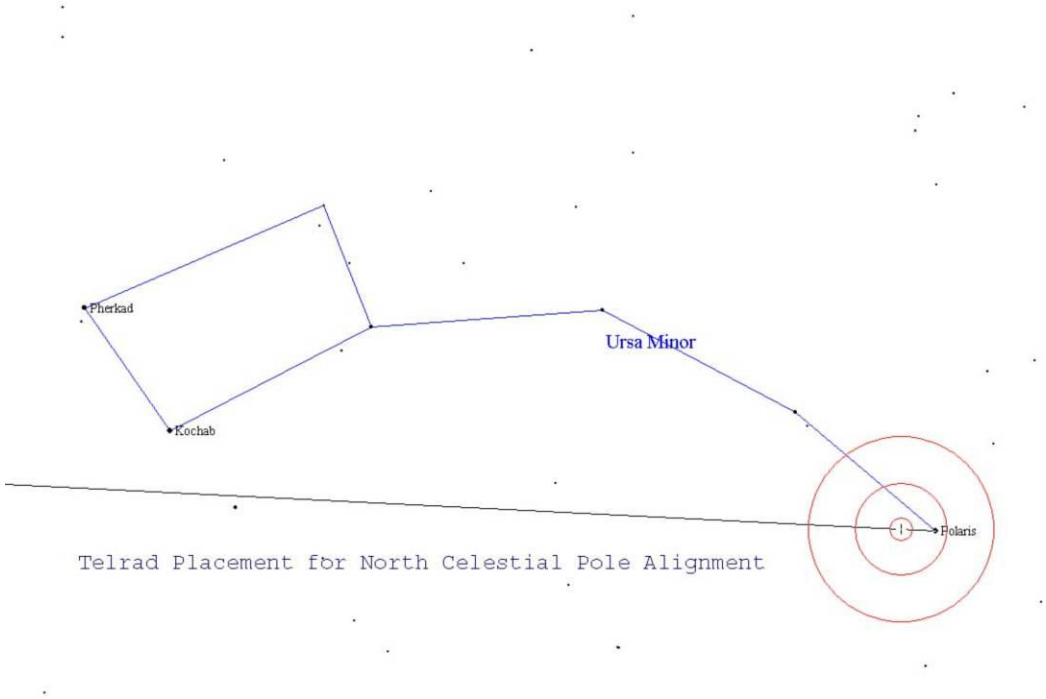
Choice 1:
Step 1. Before turning on power to the controller, position the telescope to the celestial pole (NOT at Polaris! The Celestial pole!). Turn on power.
The controller initializes the azimuth, and initializes the altitude to the same value as the last saved latitude. Using Choice 1 means you don't even have to use the hand paddle, although you must have previously saved your latitude.

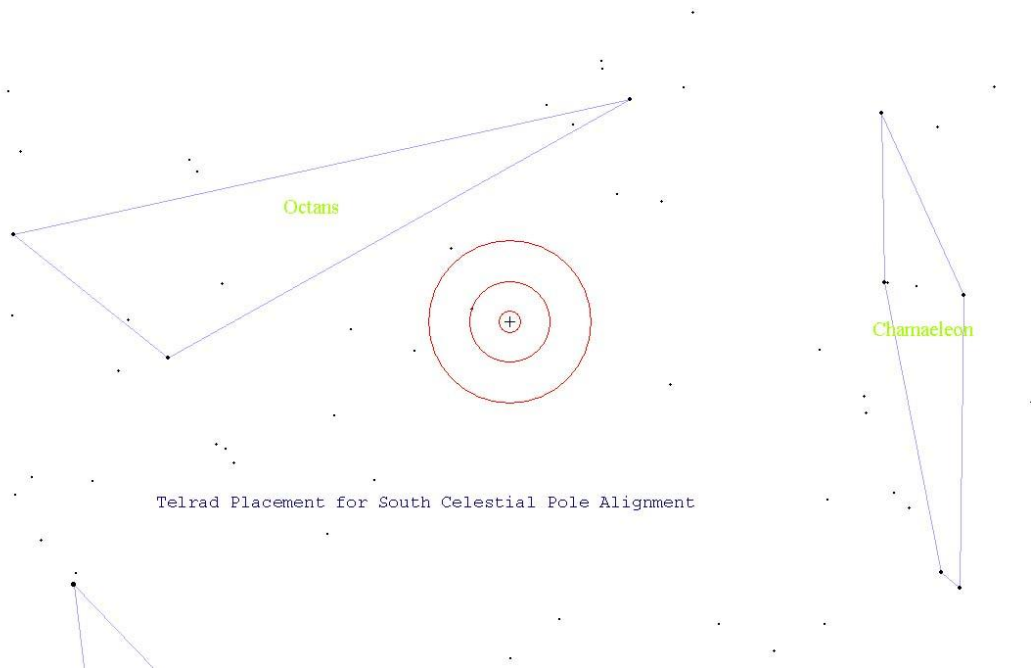
Choice 2:
Step 1. Move the scope to the telescope zenith and hold down the right hand bottom button for 4 seconds
The controller initializes the altitude position at 90 deg's.
Step 2. Move scope to celestial pole, and hold down the left hand bottom button for 4 seconds
The controller initializes the azimuth, and initializes the latitude. (Note, must have version 1.6 of firmware for this to work properly).

Choice 3:
Step 1. Move scope to celestial pole, and hold down the left hand bottom button for 4 seconds
The controller initializes the azimuth, and initializes the altitude at the same value as the last saved latitude.

Before a user can use choice 1 or choice 3, the latitude needs to be set up and saved to flash ROM.
If a user used choice 2, they may want to save the latitude, so next time they don't have to initialize at the zenith. This is done by holding down both the left and the right hand bottom buttons for more than 10 seconds. This action will save the latitude into the flash ROM, thereby making it possible that next time the telescope is set up (assuming the base is reasonably level, and you are near the same latitude), the user will not have to use choice 2, but only choice 1 or choice 3.

Here are some screen shots from Earth Centered Universe which show placement for celestial pole alignment. The QuickFinder has the two middle circles but not the outer circle. The two middle circles are the same size in both the QuickFinder and the Telrad. Limiting magnitude is 6.0.





While the controller is in the DragNTrack mode, the serial port can be connected to a laptop running almost any planetarium software. Set your software to connect to a "Tangent" encoder interface box, and you will be able to surf the sky using the planetarium software! When setting up your planetarium software, set up both encoder resolutions to 18000. Also, do not use the option that your telescope is on a tracking platform. Think of it like a telescope without a platform where you have this little guy pushing it all the time, exactly keeping up with the stars. That's what your planetarium program will "think"!

Also, set the baud rate of your planetarium software to 19200, no parity, 8 data bits, and 1 stop bit. If your software doesn't support 19200 baud rate, contact us, there is a way to set the controller to 9600 baud if you have firmware version 1.5 or later.

It's a good idea to set up the DragNTrack mode even if controlling with a computer or the Argo Navis™. When connected to a computer or Argo Navis™, the controller automatically leaves the DragNTrack, SlewNTrack or Equatorial modes, but it doesn't save the mode to the flash ROM. If you do this, you will have tracking (if initialized on the celestial pole) even before you connect the Argo Navis™ or the computer.

If tracking before connecting a computer or Argo Navis™ is not desired, there is no need to initialize the controller on the celestial pole.

Slew and Track Mode

This mode is similar to the drag and track mode, but is used if you don't have shaft encoders on the altitude and azimuth of the telescope.
To use the Slew and Track mode, your Alt/Az telescope must have the following features:

1. A pair of servo motors to drive the telescope in altitude and azimuth.
2. A handpad.
3. A 12 -24 volt power supply.

This mode uses the servo motor encoders as the telescope position. You must use the handpad to slew the telescope. You must select the SlewNTrack mode from the “Auto Tracking” tab of the “Edit Parameters” of the ServoConfig software.

You must also set up the number of telescope encoder ticks to be the same number as the number of motor encoder ticks using the ServoConfig software. This is done automatically for you if you use the ServoConfig software, and you click on SlewNTrack mode.

Important Note: If your telescope has encoders for altitude and azimuth, but doesn't have clutches, your best bet would be to use the DragNTrack mode, but slew with the handpad instead of “dragging” the scope.

If you don't have external encoders, and you plan to use the SlewNTrack mode, you must set your controller up for SlewNTrack so the controller will ignore the telescope encoders if using the Argo Navis™. If tracking before connecting a computer or Argo Navis™ is not desired, there is no need to initialize the controller on the celestial pole.

8.3.6 Equatorial Mode

The Equatorial mode is provided for equatorial platforms, or equatorial mounts, both Fork and German Equatorial Mounts. The configuration options allow the Right Ascension motor to start tracking when powered up, at the rate specified. This allows operation of the mount without a PC connected. However there are many features that require a PC in order to achieve the maximum utility of the system. With a PC, the SiTech can Park and UnPark, use PointXP to provide extremely accurate pointing for GoTo operation, PEC error correction and (with a high resolution mount encoder on RA) using Tick Management (TM) the mount can be free of PEC and achieve sub arc-second tracking.

Setting Up the Equatorial Mode

Using the ServoConfig software, click on "Edit Parameters", then "Auto Tracking". Click on the Equatorial mode radio button. Set up the Equatorial rate. If you know the number of effective encoder ticks for one revolution of the Right Ascension axis, enter that number in the green text box to the right. After pressing the Enter key, the proper Equatorial rate will be calculated and will appear in the "equatorial rate" text box. (The number you entered in the green box will also change to accommodate the math the controller uses internally. Not to worry, if you entered the correct ticks/rev, the slight change the controller needs will not mess things up for you!) If you do not know the exact number to enter and want to experiment, a larger number will make the system track faster.

You can adjust the tracking rate up and down by pressing a certain sequence on the handpad (described in the next section). The amount it is adjusted per handpad press is in the text box labeled "Equatorial Up/Down Adjust".

If you have a platform, set the "Tracking Platform Goal" to the number of motor encoder ticks where you want the platform to stop. The SiTech controller will stop at this point, and then you can then perform a re-wind.

If you have an equatorial telescope, set this platform goal to at least 10,000,000. This will make it impossible to stop at the platform goal, it will continually track forever (if you don't stop it).

If the motor tracks the wrong way, you can put a negative number in the "Tracking Platform Goal".

Operation of the Equatorial or Tracking Platform Mode

If using an equatorial platform, before powering up the controller, be sure the platform is fully rewound. This will be its home position.

Upon power up, or after a rewind, the Right Ascension motor will turn at the configured speed, until it reaches the configured goal.

To rewind, press both top buttons on the handpad. This is possible at anytime. The servo motor will return to the same location it was when power was turned on; at its programmed slew speed rate. To Cancel rewind, press any direction key.

If using an equatorial telescope, the above paragraph will not be applicable, as the platform goal will be set for 10,000,000 or greater.

To stop tracking, press the top right key. To start tracking again, press it again. When the tracking is stopped, the LED's will blink off briefly every few seconds (version 1.6 of firmware and later). Only the new handpad will allow fine tuning the tracking speed.

To raise the tracking speed, press the top left key, and while holding it down (think of it as an ALT key) press the up direction.

To lower the tracking speed, press the top left key, and while holding it down (think of it as an ALT key) press the down direction.

The up/down speed adjustment will not work if you have set up the local search feature (version 1.7 or later of the firmware). To enable the UP/Down adjustments, you will have to enter a value of zero in the local search distance and speed parameters.

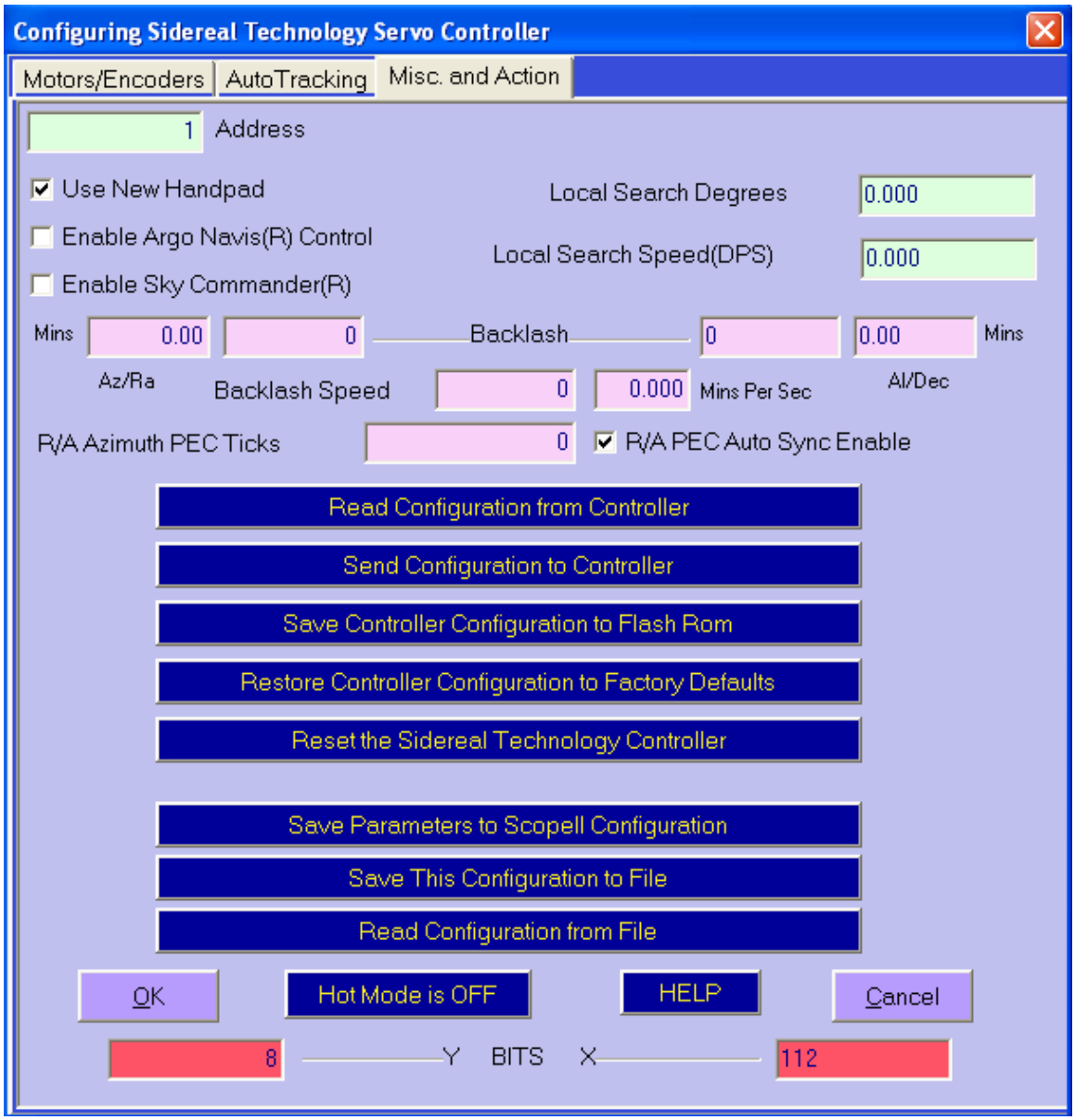
To perform photography, it may be necessary to enter the guide mode. This is done by holding down the top right key, then pressing the right arrow key. Now the speed changes by adding or subtracting the guide rate from the current rates. To revert back to the normal mode, hold down the top right key, and press the right key again. This feature only works with the new handpad. This will be un-necessary if you are using a radio handpad receiver, as in this case, the SPD switch on the transmitter will toggle between Slew/Pan/Guide modes. If you are using an auto guider, as soon as the guider commands are sent to the controller, the SiTech controller is automatically set for guide speeds.

Challenging Idea: If you have or have access to, a very large aperture telescope, 28 inches or larger, you may be able to see the crab pulsar blink. Here's how. Obtain a servo motor with encoder that is able to run at least 1,800 RPM, you could connect this motor to a servo controller, and put an interrupter on the shaft of the motor (a circle with a pie shape cut out of it), balance the interrupter, and hold this assembly between the secondary mirror and the eyepiece, while running at somewhat less than 1800 RPM. You may be able to see the pulsar wink in and out! To test this out, you may use one of the LED's. It winks at the same frequency of the crab pulsar while in the equatorial mode! You can configure the equatorial mode to automatically run this motor at the proper speed.

With the controller in the standalone Equatorial mode, LED 1 flashes at the same rate as the crab pulsar. This is too fast to see it as a flash, your persistence of vision makes it look like it's on steady, a little fainter than normal. If you shake your head as you're looking at it, you will then see the flash! You may wonder why we did this, it's purpose is for testing an interrupter you may fabricate to see the crab pulsar central star fade in and out. See section on the Equatorial mode for more details.

If you do this, please let us know, we would really like to hear about it!

8.3.7 Misc. and Action Screen



Clicking the OK button will return you to the main menu, and the data you've edited will be sent back to the main structure.

Clicking the Cancel button will return you to the main menu, but the data you've edited will be lost (unless you saved it to the controller or to a file using the "Misc. and Action" page).

Address

The first option is the Module address of the controller, **this is NOT the PC comm port number being used.** The controller has an open collector RS232 port for communication. This allows more than one controller to be on the same communication port. If this is desired, all the connected controllers have to have a different address. There are 3 addresses available, 1 (default), 3, and 5.

If you change the address, the module will respond differently. Now all ASCII commands that started with an X or a Y, will have to start with a T and a U for address 3, and V and W for address 5.

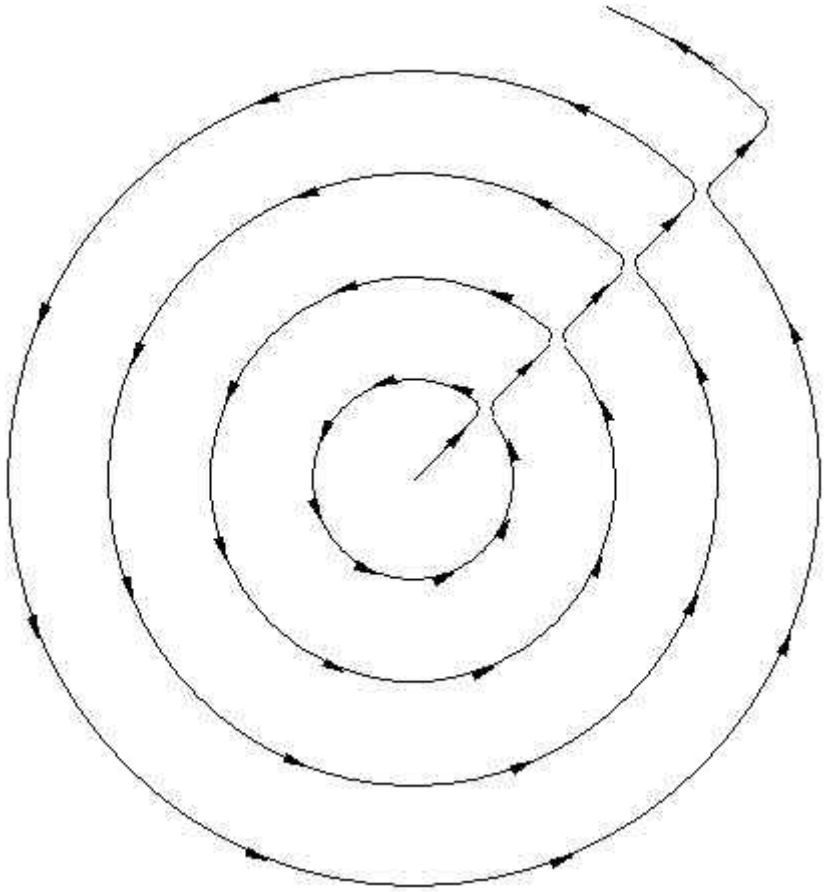
After changing the address in the textbox, it would be a good idea to "Send Configuration to Controller". Otherwise this program may try to communicate with the wrong address.

Local Search Setup

This is how many degrees the local search radius will be. You need Servo Firmware Version 1.7 or later for this feature. As pointed out by Ed Harvey, a Spiral search isn't the most efficient search algorithm, because if you don't want to miss any sky, you will have a lot of overlapping sky. A more efficient algorithm is to move the scope 1 radius out, then do a circle, move another radius out, and do another circle, etc. This is how the local search routine works on the SiTech servo controller. You must have proper data in the controllers Motor Encoder Ticks per rev for the calibrated speed and distances to work properly.

A good starting point is to figure the field of view of your most frequently used high power eyepiece, and enter this number. When you use this eyepiece, initiate with the TOP LEFT / DOWN key. When using a lower power eyepiece, initiate the search with the TOP LEFT / UP key.

Note: It's not really a true circle, but a 16 sided polygon, which looks like a true circle at the eyepiece.



Here is a brief description of how to use the Local Search feature:

1. Set up the Local Search Distance (radius) and speed (Degrees per second). Make sure the motor encoder ticks per revolution are set up properly too.

2. While aimed at the area of sky you believe the object to be is, hold down the top left key of the keypad, and then press the UP key. You can also initiate it for a higher power eyepiece which halves the circle radius, by holding down the Top Left key, and pressing the DOWN key.

3. To increase the search speed, hold down the Top Left key and press the UP key.

4. To decrease the search speed, hold down the Top Left key and press the DOWN key.

5. To reverse directions and start working inwards, press the DOWN key. You can revert to the normal direction by pressing the UP key. This can be done at any time during the search.

6. To PAUSE the search, press the speed key. To resume, press the UP, DOWN, or the speed key again.

7. To exit the Local Search mode, press the LEFT or the RIGHT direction key.

Use New Handpad

If you have a handpad for Mel's old stepper system, it will work with an earlier style controller, but you will not be able to slew both axes at once. If you have an old handpad and want to use it with an earlier style controller, de-select this menu item. For the new handpad, select it.

Enable ARGO NAVIS®

Check this item if you would like to use the Argo Navis® to control the Servo Controller. If you select this, the controller will not respond to any other commands until it decides that the Argo Navis® is not "listening". This takes about 10 seconds. After this initial delay, if the Argo Navis® is not detected, the Servo Controller will communicate normally with all other protocols. You may see the communication freeze up in this program while the Servo Controller is deciding there is no ArgoNavis® connected. To re-connect to the ArgoNavis® press the Top Two handpad buttons simultaneously (same as a GOTO command for the Argo Navis® control.

Enable Sky Commander®

Check this item if you would like to use the Sky Commander® with the Servo Controller. Set up of Sky Commander® is similar to the set up with Argo Navis®. The two units are connected with a flat cable to the comm. ports. Use the Hand Pad top two buttons to communicate and command GoTo's to SiTech.

The SiTech controller RS-232 port is used for both connecting to a PC for setup with ServoConfig as well as connecting to the Sky Commander™. You need to set both the Sky Commander and SiTech Controller to the same 9600 baud rate. The easiest way to set the baud rate for SiTech is to use the "Terminal Window" in ServoConfig. You can issue commands like this without doing anything else:

type SB1 where: SB# = Set Baud Rate (1 = 9600, 2 = 19200). Save the setting change by typing XW.

Backlash Settings

The servo controller has a provision for backlash. Mel's ScopeII also has this feature. **It is extremely important that you don't have backlash enabled in both ScopeII and the servo controller.** ScopeII backlash is fairly slow to respond, so we think it would be better to use the controller backlash, as the response is immediate. Also, if the telescope is being controlled by the ASCOM driver, the Argo Navis®, or any other device backlash MUST be enabled and adjusted in the controller in order to compensate for your mechanical backlash.

And finally, many auto guider programs have internal settings to control backlash. The auto guider routine and the SiTech controller routine may conflict. If they do, enable one and deselect the other.

To Disable backlash in the SiTech Controller, set the speed to zero.

There are two windows for each Backlash axis, 1 is raw motor encoder ticks, the other is in Arc Minutes. You can enter either, and both are adjusted when you press enter or Tab. The initial settings can be set using the "Backlash Help" from the MAIN MENU. You can manually change these setting here. Be careful using Backlash as it will affect guiding if you use an auto guider for astrophotography. If the Backlash setting is too strong, good guiding may become impossible.

Backlash Speed

When the motor is in the backlash area, it means the telescope isn't being moved when the motor moves. We call this the No-Mans land. The Motor doesn't "like" to be in No- Mans land, and will move out of it in the proper direction as necessary. The speed in which the motor moves out of the No-Mans land is labeled Backlash Speed. This speed is added to the commanded speed, so it will always move out of this area. For a slow backlash speed, consider about 30,000 raw counts. For a quick backlash speed, consider a value of about 500,000. You need to find the right speed that quickly takes up the backlash, but doesn't jar the telescope when the backlash is taken up. The speed in Minutes per second is an average of the two axis's, and may be different for each axis, depending on your motor encoder ticks per revolution. We also recommend that you set the backlash to a value very slightly below where you see a "jar" in a high powered eyepiece… Better to not "quite" remove it all than to move too far in removing it.

If you are using a GEM, it is recommended you evaluate and set the backlash while "Looking West" to avoid setting a negative setting that goes the wrong way.

It is also possible to change these settings directly from the SiTech.exe driver by:
Click on Features, then Controller Stuff.
Click in the terminal window, and then type the commands:
"XXPnnnn<E>" for RA/Azimuth backlash and
"XXOnnnn<E>" for declination/altitude backlash.

R/A PEC Ticks and Auto Synch Enable

If you want to use periodic error correction for astrophotography you need to have a way to synchronize the position of the worm or friction wheel and specify the number of encoder ticks for one worm rotation. Enter the number of ticks for one rotation of the worm in the box. The number is the total ticks per scope revolution divided by the number of teeth in the RA gear. There are several circuits for AutoPECSynch available in Appendix J to this manual that synchronizes the Periodic Error (PE) every worm revolution. Otherwise you can manually start and then synchronize periodic error correction if you have a way to observe the worm position (i.e. a pointer on the worm shaft and an index mark). The SiTech.exe provides a way to record PE or you can use PemPro version 2.5 or later. Once the PE is recorded to the controller, it will automatically correct for PE, even without a PC connected.

Read Configuration from Controller

Selecting this button reads the configuration from the controller to the Configuration Structure so you can see it, edit it, or save it.

Send Configuration to Controller

Selecting this button sends the current Configuration Structure to the controller. What you see on these 3 Configuration Tabs is sent to the controller, and saved in the Flash ROM.

Save Controller Configuration to Flash ROM

If you've been using the Hot Mode, the changes will be in effect in the controller, but if the controller was to re-boot, you would lose these changes. To make the controller "remember" them across a power failure, select this button. <u>It is not necessary to save to flash if you use the Send Configuration to Controller button since it automatically is saved to the flash in that case.</u>

Restore Controller to Factory Defaults

All of the parameters in the controller are set to factory defaults if this item is selected. You are prompted with an “Are You Sure” or equal. There is no "undo" feature!

Reset the Sidereal Technology Controller

The command resets the controller. Any changes you made using the hot mode are lost unless you already saved them to flash. If the controller is forced to manual by a motor fault, you can put them back in auto by clicking on this item.

Save This Configuration to File

If you want to save your configuration to a file, select this item. The default extension for the files is .car. The configuration sent to the EDIT PARAMETERS dialog is called default.car, and has the same values in it that are the factory defaults. These files are normally saved in the same directory as ServoConfig (C:\ServoConfig).

Read Configuration from File

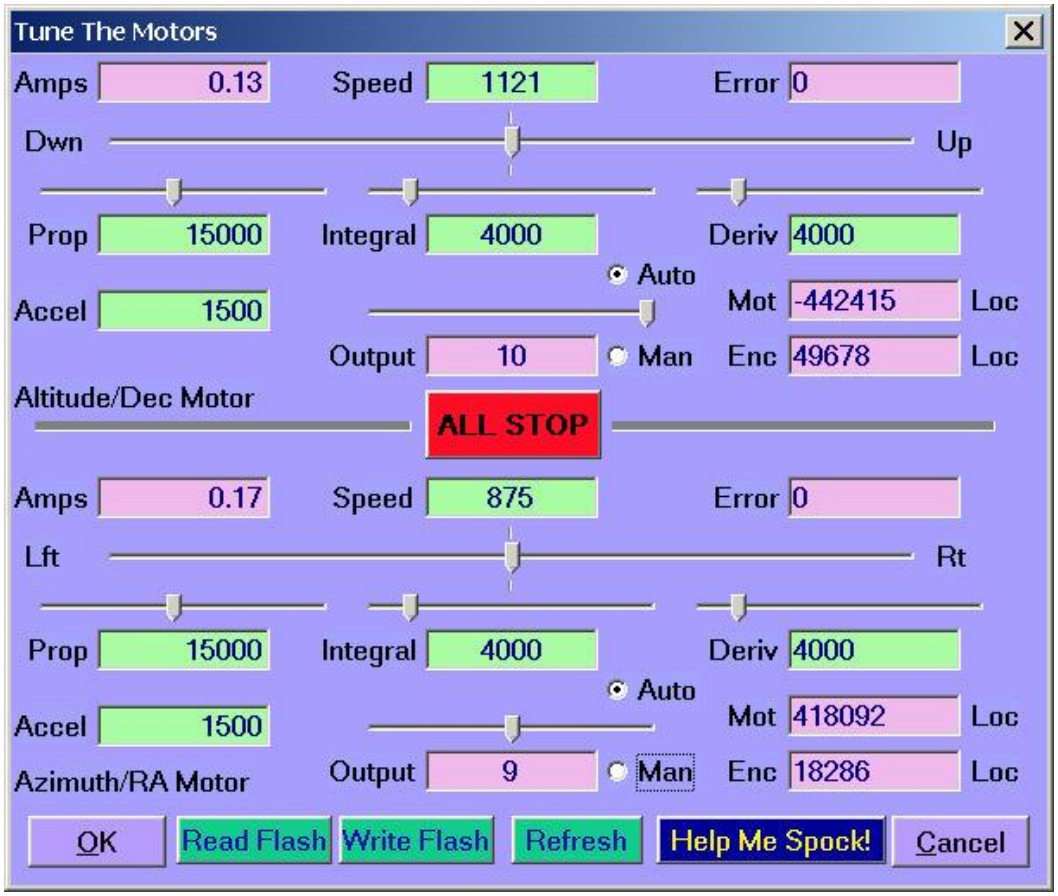
Clicking here, you can load the Parameter structure with a file of your choice (probably one you've previously saved). This doesn't automatically send it to the controller. After loading, you can then Send Configuration to Controller if you want to. Most times you will simply click on “Get data from Controller”

8.4 Test and Tune

This brings up another window named “Tune the Motors” where you can operate both motors easily, and at the same time, change the PID (proportional, Integral, Derivative) parameters, along with the acceleration, to get the motors operating smoothly and accurately.

The layout of the window had the top half dedicated to the Altitude/Dec axis, and the lower half primarily dedicated to the Azimuth/RA axis, with a big red ALL STOP panic button in the middle.

The good news is 99% of the time, the default values in the system work great, but if you end up with a good deal on some really different motors, this makes quick work of getting them running smoothly!



This dialog screen is a "live" screen, in other words, it continually communicates with the controller, and updates all of the Magenta colored text boxes, and every time you change a green text box or a slider, the new number is sent to the controller.

The green text boxes are not continually updated from the controller. On entry to this dialog box, these values are read from the controller. Every time you change one of the green textbox numbers, it is displayed, and it is also sent to the controller. If there was a communication problem, these values may not be correct. Clicking the "Refresh" button will re-read these values from the controller.

Clicking the OK button will return you to the main menu, and the tuning data you've changed will be sent back to the main structure.

Clicking the Cancel button will return you to the main menu, but the data you've changed will be lost (unless you saved it to the controller).

If there is no communication, there will be a slight delay before this screen appears. All of the green text boxes can be changed by typing in a new number, or by moving the slider associated with it. The Acceleration doesn't have a slider.

To move a motor, be sure it is in the Auto mode, grab the speed slider with the mouse and move it. The motor should start to move. As an alternative, you can also enter a speed number in the text box.

8.4.1 General PID Tuning Principles

A complete discussion of Proportional Integral Derivative control is not appropriate here, but there are a lot of places on the Internet if you do a search for it. As it applies to the Sidereal Technology Servo Controller, it is fairly simple to adjust if you follow these directions:

Make sure both motors are stopped, and if they are in manual, click the AUTO radio button. If the controller is address 3 or 5, the indicated auto and manual may be wrong. To be sure, click on the auto radio buttons. You can also look at the lights on the controller. If they are on steady, the controller is in auto. If a light is flashing at a 1 second rate (or so) the axis is in manual.

Step 1. Turn the integral all the way to zero.

Step 2. Turn the proportional down to zero.

Step 3. Adjust the Derivative. The motor should turn sluggishly but freely now, since only the derivative is working. Adjust the derivative higher and higher until the motor starts humming or vibrating. Adjust it back down until it doesn’t vibrate, but "resists" you when you move the motor by hand. Typical values are about 4000 to 6000.

Step 4, Adjust the Proportional Band. Bring the Proportional band up, a few thousand at a time. As soon as it is over a few hundred, the motor will "want" to return to it’s desired position. Keep adjusting this up until the motor becomes unstable. Now back it down until the motor is stable. Typical values are about 15000.

Step 5, Adjust the Integral. To adjust the integral, start increasing the number from about 1000, until the motor becomes unstable. Then back it off until the motor is completely stable. Typical values are about 4000.

Step 6, Run the motor at several speeds, and be sure it operates smoothly at all speeds. If it doesn’t, start by working backwards through the steps, first lower the integral, then lower the proportional. Play around until you’re happy with the response of the motor.

Pay attention to the Error textbox while operating the motors. When you are moving the motors very slowly, the error should be -1, 0 or 1, with an occasional 2 or -2. If higher than this, increase the Proportional, and if unstable, increase the derivative.

If the two motors are the same type, the same numbers should work in both motors.

During the above testing, the controller may put the motor in manual if the error reaches the error limit. If this happens, simply click on the Auto button, and then you can move the speed slider to start it moving again.

If you are communicating with an address of 3 or 5, the auto and manual radio buttons may not be updated correctly. To be sure they're in the mode you want, just click on the radio button. This will put the controller in the mode you want.

8.4.2 Accel

The Acceleration is the rate at which the motor ramps up to speed. Lower numbers create a very slow ramp. Higher numbers make a faster ramp. I like my telescope to be about 1000 to 1500.

8.4.3 Read Flash and Write Flash

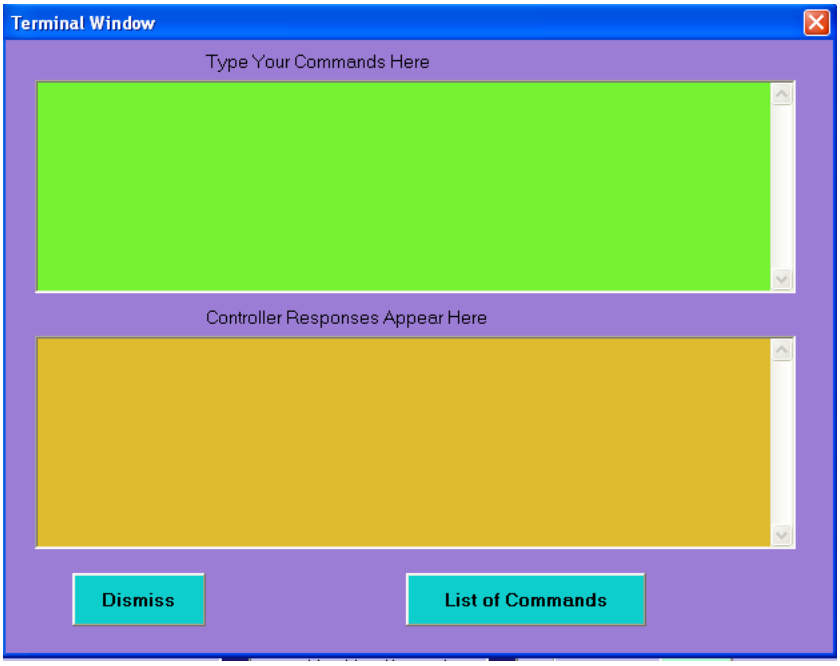
If you like your settings, and want to save them to flash, there is a button near the bottom of the window labeled "Write Flash". If you think you've messed up the control and want to revert back to the controller’s original settings, you can select the "Read Flash" button.

8.4.4 Refresh

The green text boxes are not continually updated from the controller. On entry to this dialog box, these values are read from the controller. Every time you change one of the green textbox numbers, it is displayed, and it is also sent to the controller. If there was a communication problem, these values may not be correct. Clicking the "Refresh" button will re-read these values from the controller.

8.5 Terminal Window

The Sidereal Technology Dual Servo Controller has many ASCII commands. Using the terminal option allows you to type commands, and see the returning data from the controller. There is a button on this screen that displays the ASCII commands that the controller will accept. Since the controller only "listens" to upper case characters, the characters you type are changed to uppercase if necessary. The line you type is sent after you press the enter key. This allows normal editing changes to the string before the enter key is pressed.

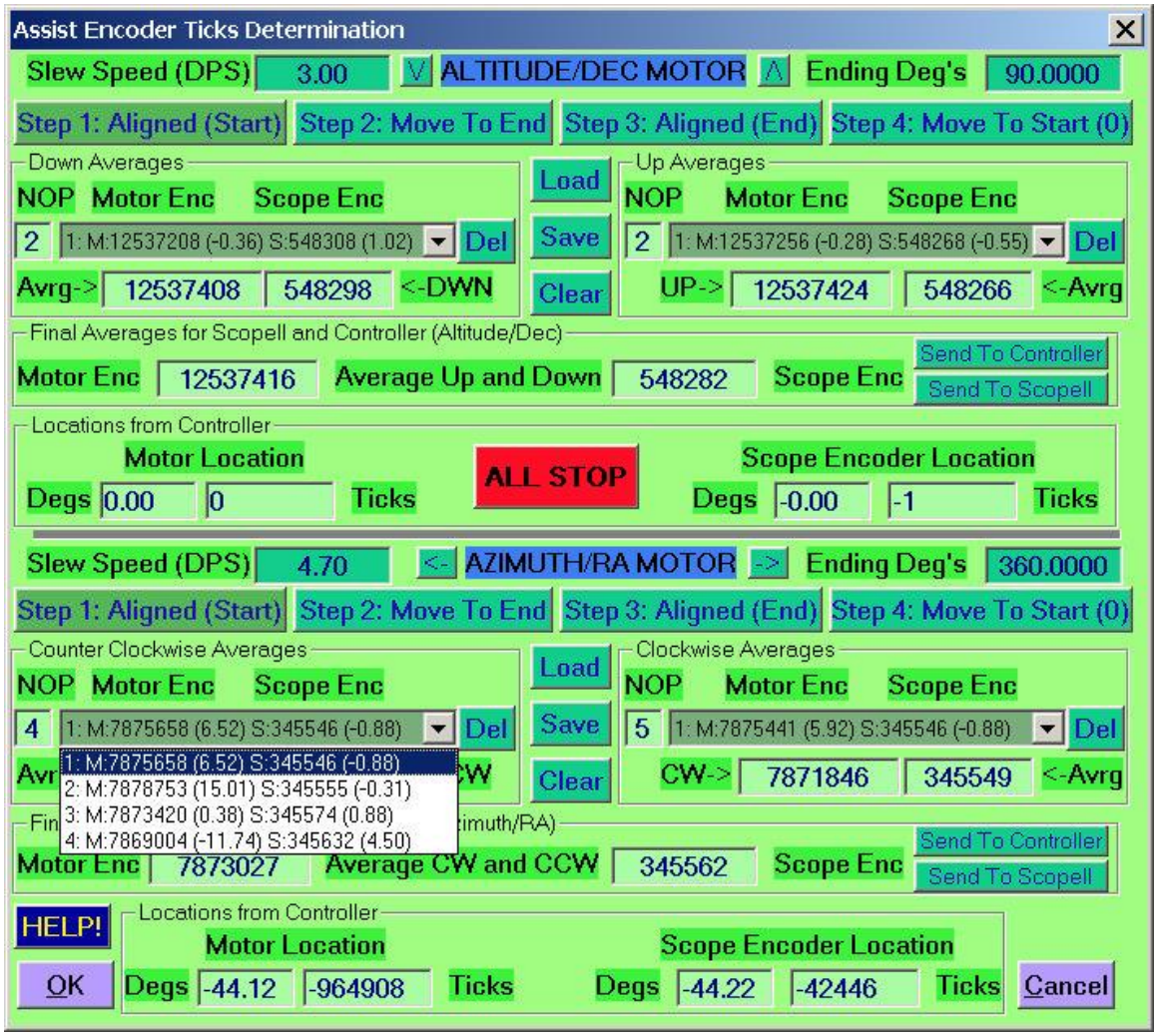


The Really really good news is as the tools have evolved, the use of these ASCII commands by most users has disappeared except for troubleshooting when you can Dan. There are still some cool things this capability provides for the true Geeks amongst us however!!! (you KNOW who I am talking to, right????)

8.6 Ticks Determination

Clicking on this button brings up the "Assist Encoder Ticks Determination" screen. I've spent a lot of time double checking my encoder counts per scope revolution. I've always done it manually, by using the handpad and also by typing commands using a terminal program. This feature semi-automates this procedure to make the task easier to accomplish.

When you first see this window, it is really scary looking!!!! Then see that it is really the same thing on the top for the Alt/Dec motor as it is on the bottom for the Az/RA motor. It's actually quite simple to use!!



If your telescope axis's are all geared, with no clutches or rollers, you might be better off calculating your encoder ticks per scope revolution, and entering these numbers in ScopeII's configuration file if you use Mel's program, and also the Configuration portion of this program. This dialog window is primarily for users of telescopes with roller or friction drives. (However, at least one user used this tool to discover his RA gear had a different number of teeth than he thought when the calculated ticks/rev was not working right and he then used this tool and compared the differences).

Problem: You want to know how many encoder ticks per revolution your scope has. You've done the math, and have a close estimate, but you're using a roller drive, and reality is not quite the same as the math. ScopeII needs these numbers, as well as the controller itself if you are planning to use the DragNTrack or the SlewNTrack modes.

Solution: Test it out using this software tool.

I've done this many times by hand with my 14" and also my 28" scopes. I've done it by typing ASCII commands to the servo (or the stepper) controllers, and commanding the scope to move all the way around in azimuth, or horizon/zenith in the altitude, adjusting the final position using the handpad, reading the scope position, commanding the scope to return to the starting position, adjusting the final position using the handpad, then reading the scopes position again. I did this back and forth many times, and averaged the results. The final numbers I plugged into the configuration file.

I had the idea of semi automating this procedure, and here is the result.

I'm sorry the screen is so busy looking, look again, and you will see that only the "Forest Green" items are changeable, so that's not too bad eh? Also, have a look at the steps. It's as easy as 1,2,3,4 for each axis!

8.6.1 General Principal

Move the scope between two known angular distances as many times as necessary, count the encoder ticks necessary to accomplish this, send the average to ScopeII and the Sidereal Technology controller.

I've discovered using both of my telescopes that there is some difference when moving one direction than the other. This is especially true of the altitude, because any unbalance will tend to make the scope move more in one direction than the other. This is why there are separate average arrays for each direction.

8.6.2 Picking Starting and Ending Positions

If you are doing the altitude axis, I've found the best solution to be using a precision square bolted to the altitude axis, and a precision level that is placed on the square at zenith and horizon. In this case, you would select 90 degrees to be your ending position.

If you are doing the Azimuth axis, you can use Polaris for starting and ending position, or a far away tree or radio tower also works well. Another solution would be to clamp a laser to your scope, and mark the position of the red dot a few meters away on a wall, car, or bicycle. In this case, you would select 360 degrees to be your ending position.

One last way I've thought of to pick your starting and ending positions. If you have 8192 ticks per revolution encoders mounted directly on your telescopes axis, you can simply watch the scope encoder ticks in the window to align your starting and ending positions. This would not be as accurate as sighting on a radio tower, but for many purposes, this would be good enough.

8.6.3 General Procedure

Align scope on starting position. Zero the scope axis. Move and align scope to the ending position. Read the scope axis and store number in the Up or Clockwise average array Move and align scope back to starting position.

Read the scope axis and store number in the Down or Counter Clockwise array Repeat procedure until satisfied that there are enough data points for a good average. Average the results, and send to ScopeII and the Controller.

8.6.4 Specific Procedure

1: Make sure the Ending Degrees is proper for your setup. **It's got to be more than 10 degrees**. For a typical Alt/Az telescope, normally the Altitude should be 90 degrees and the Azimuth should be 360 degrees. These are the defaults. If you're using a German equatorial, you may be able to have the declination set to 360 degrees as well, especially if you can loosen the gear clutch and prevent the scope from rotating, but can see the gear itself to make an index mark on it to judge when it has rotated 360 degrees.

Also make sure before you run this procedure that the Motor Encoder Ticks Per Revolution have been set up to a reasonable proximity of reality. That's not absolutely necessary as you could use the handpad to find the first ending alignment point.

2: Align your scope axis manually on your starting position.

3. Now click Step One (Align Start). If this is the first time step 1 is pressed, the software doesn't take a reading, it simply "zeros" the axis. You should see the position of the scope change to 0 in the Motor Location and Scope Location boxes. On subsequent Step One's, a reading is taken and stored in the average array. Important note: If you want the same number of samples taken in the clockwise (or Up) directions as the opposite direction, Step One should also be the LAST step performed!!!!!

4. Now click Step Two (Move to End). The scope will move to the ending position. If this is the first time, it uses whatever Motor Encoder Ticks Per Revolution you've set up using the configuration. When the scope stops (you can stop it with the handpad if necessary) align the telescope up carefully (using the handpad) on the ending position. If this is the azimuth, it's probably going to be the starting position plus 360 degrees. If this is the altitude, you are probably looking at the zenith.

5. Now click Step Three (Aligned End). Your first (or new) reading should appear on the right half of the screen. You will see the number in the drop down choice box, and the clockwise or Up averages will be updated.

6. Now click Step Four (Move to Start (0)). The scope will move to the starting position.

7. Align the scope up on the starting position carefully and then press Step One again. Now the readings on the left half of the screen will be updated. You will see the new numbers on the drop down choice box, as well as the new updated averages.

Keep repeating the 1,2,3,4 buttons, until you're satisfied with the average numbers.

Always end by pressing the step one button.

After you are done, you can click on "Send to Controller" and/or Send to ScopeII" buttons to save these numbers in respective locations.

One more thing you may want to do. If your drive system has backlash in it, when aligning to your starting or ending positions, always end up pressing the same direction button before clicking the alignment step button. This will assure that the mechanical hysteresis will be taken up for all the readings. Mel's ScopeII software has a parameter for dealing with backlash, but for our purposes here, we just want to know how many ticks a complete revolution takes, all in one direction.

8.6.5 More Info

1. The number below the "NOP" label is the Number of Points you have taken

2. The text in the drop down choice box has the following format: N: M: nnnnn (nn.nn) S: nnnn (nn.nn)

The first N is the particular Data Point the M: is the label, letting you know the first number is the data point from the Motor Encoder.

The number in parenthesis is the deviation from average (in arc minutes) for the Motor Encoder

The S: is the label, letting you know the next number is the data point for the SCOPE encoder.

The ending number in parenthesis is the deviation from average (in arc minutes) for the Scope Encoder If the scope encoder are not giving reasonable numbers, it is flagged as "bad"

3. The Del button to the right of the drop down choice box allows you to delete a reading, if for some reason it is flawed.

4. The left number below the drop down choice box is the current average of the direction readings for the Motor Encoder

5. The right number below the drop down choice box is the current average of the direction readings for the Scope encoder

6. The two numbers in the frame labeled "Final Averages for ScopeII and Controller" are the (left) Motor and (right) Scope Encoder values. There are two buttons on the very right for updating the controller and ScopeII.

7. The lowest frame (for the particular axis) has the current motor and scope locations, in degrees on left, and encoder ticks on right.

Clicking the "SAVE" button will allow you to save your work. Clicking the "LOAD" button will load the previously saved button. There are no filename choices; it always uses the same two files, one for azimuth and one for altitude. If you want to save them to a permanent file, you will have to copy the data file to another file. The filenames are "AlArray.dat" and "AzArray.dat".

Clicking the "CLEAR" button will erase all of the data points on your screen, and you can now start from scratch. You will be prompted before the erasure.

Clicking the OK button will return you to the main menu, and the data you've edited will be sent back to the main structure if option chosen. You will have a chance to abort.

Clicking the Cancel button will return you to the main menu, but the data you've edited will be lost (unless you saved it to the controller or to a file using the "Misc. and Action" page). You will have a chance to abort.

8.6.6 Ticks Determination using mount encoders

When using servoConfig and ticks assistant, instead of using a laser and finding marks, doing trig, etc, it may be easier to just use the altitude encoder as a reference. With an 8192 encoder, you have 2 arc minutes (and change) of resolution. There are a couple of tricks to this. When you are on the low side, click on the step one button, then slowly raise the scope (guide mode) until the next tick comes in, then stop right away. Now click on the step one button again. This resets the altitude encoder to zero (as well as the motor encoder).

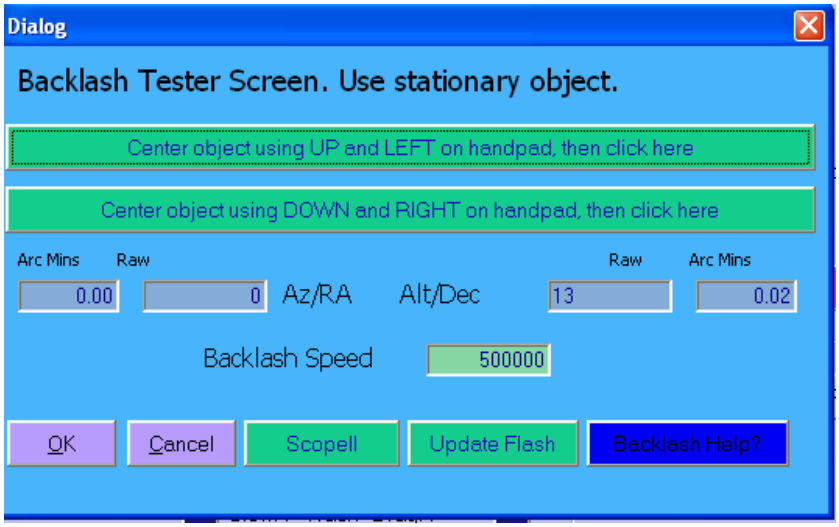
So you're using a tick threshold. Now do step two, but stop the scope short of the goal. Now make sure the tick you use is a multiple of 4, because it's a quadrature encoder. If you use the same "tick sensor", then the accuracy will be better than any old tick. As an example, say you can only go to from 10 deg's altitude to 80 deg's altitude. that's a total of 70 deg's.. That would be 1593 ticks. But don't use that as your 70 deg reference, because it's not divisible by four. Use 1592 ticks. This is divisible by 4. Now that you know this angular distance, 69.9609375, put this number in at the top of the screen, where the default is 90 degs.

So don't do a step 3, instead do a step 4, because now we've got to start over, but we know the exact angle to put in the servoconfig box, and we start over.

Once you're near zero, then go a bit below, then slowly creep up until the scope encoder just changes to 0. Now proceed as step1 step2, step3, step4, step1, and repeat as many times as you want, but keep in mind to always creep in on the 1592 tick, and always creep in on the 0 tick.

8.7 Backlash Help

Backlash control is included in the basic capabilities in the Servo Controller. Clicking on the Backlash Help button brings up a window for measuring and setting the Backlash parameters. These values can be modified on both the “Motors/Encoders” tab and the “Misc. and Action” tab at any time.



You need to use a crosshair eyepiece, or, alternatively, you could mount a laser on your scope, and look at the dot for movement. The important thing is it needs to be a stationary object, such as Polaris (OK, so it’s only “mostly” stationary!), a distant radio tower, or something. You shouldn't do this on any other star other than Polaris, or the southern hemisphere equivalent or the star motion will mess up the calculation. If you select a distant tree, be sure the wind is not blowing, or this could mess you up too (I speak from experience here!!!).

8.7.1 General Backlash Procedure

First, line up the object on the crosshairs of a crosshair eyepiece. Be sure that you only use the UP and LEFT buttons on the handpad. If you move too far, then press the DOWN (and/or RIGHT) buttons, until the scope moves, then press the UP (and/or LEFT) again, when you let up on the UP and also the LEFT buttons to be sure all mechanical backlash is taken up.

Now click the top button labeled "Center object using UP and LEFT on handpad, then click here".

Now line up the same object again, this time using the DOWN and RIGHT buttons. Be sure the DOWN and RIGHT buttons are the last buttons you press, and be sure the scope is moving when you let up the DOWN and RIGHT buttons to be sure all backlash is taken up.

Now click the top button labeled "Center object using DOWN and RIGHT on handpad, then click here".

The new backlash values, both in motor encoder ticks and in arc minutes is displayed for each axis.

If you press the OK button, these values are returned to the main window.

If you press the Cancel button, you return to the main menu without changing the values.

You can choose to update ScopeII and/or the controller flash ROM.

If you are using version 1.5 or earlier of the controller firmware, there is no place to save the backlash in the controller configuration.

If you are using version 1.6 of the controller firmware, you can save it to the flash ROM, but it won't do anything.

If you are using version 1.7 or later of the controller firmware, the backlash will be acted upon if it is non-zero in the controller configuration.

Don't forget to set the Backlash Speed so the motor will move out of the backlash area. This is done on the “Edit Parameters/Misc and Action” window, or can also be done at this window. The speed should be set so the motor quickly moves out of “no-mans” land, but doesn't jar the telescope when it leaves the “no-mans” land area.

If you are using version 1.7 or later of the controller firmware, you wouldn't want to use the backlash in ScopeII, as the controller will do it on a much lower level, which is good. It is also much faster to respond, nearly instantaneous. If you do use the backlash in ScopeII, then you should set the controller backlash values to zero. Also, if you are using backlash in the servo controller, you should turn off backlash in Scope.cfg.

If you choose to update ScopeII, it will simply zero out the backlash in the Scope.cfg file.

8.7.2 More Notes on Backlash

If one of your telescopes axes is out of balance, the mechanical system may already take up the backlash. If this is the case, you need to set the backlash to zero.

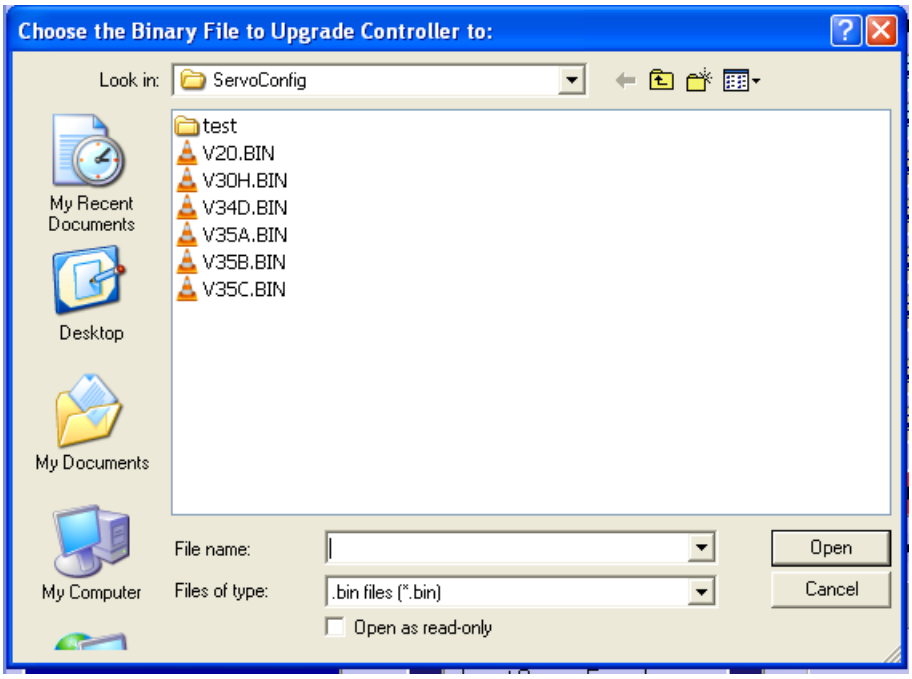
Another way to test the backlash is to use the "Edit Parameters" menu item, Select the "Misc and Action" tab, turn the Hot Mode to on, then looking through the eyepiece and operating the handpad, manually type in new values until it works properly. You look for almost instantaneous response in movement, without jarring when the backlash is taken up. You should probably use the guide mode or set the pan mode speed to very slow to do this. Watch the eyepiece for movement, and be sure it doesn't jerk when reversing directions.

8.8 Upgrade Flash ROM

The current latest version of firmware in the SiTech controller is 35C. If you have a version of firmware prior to this, you should upgrade. You can download version 35C at:
http://www.siderealtechnology.com/V35C.BIN

Download the file and place it in your working directory for this program (C:/ServoConfig). Then select this option. , browse to this file, and you will be upgraded.

You will get the screen above to confirm you want to upgrade, if you do, click YES



The screen above will appear to allow you to browse to the right file, highlight it and click on open. The update process will start and you can monitor it as it uploads the new firmware. When complete, you will get a window pop up that tells you it is complete. **Till that window comes up, watch the digital neurosurgery show going on and do not mess with anything!**

Every great once in a while, the FLASH ROM can get scrambled somehow. If that happens, there is a way to still get the firmware loaded. Power OFF the controller. Plug in a wired handpad (if RF handpad, use the handpad with a cable wired directly to the controller). Then, while holding down the top two buttons on the handpad, power the controller up. Once powered up, release the top two buttons. Then you can do the firmware update to the current firmware…..

8.9 Update SCOPE II

If you use Mel Bartels very accurate and cool software to control the SiTech controller, you may want to update the PID parameters in Scope II that you figure work best using this software. This button will do this for you automatically if you choose.

8.10 Exit ServoConfig

Stops the ServoConfig program and closes the Main Screen. Use this button instead of the X in the top right of the window to make sure things are shut down in an orderly fashion

8.11 Help

Displays a help file with most information from above.

APPENDIX A – Controller Specifications / Warranty / Tech Support

Controller Specifications

Input Voltage: 11.5 -28 volts D.C.
Input Current: 4 amps max
Maximum Motor Current: 2.4 amps each, total of 4 amps max.

RS232 Protocol:
1. The RS232 port has a diode and pulldown resistor built in, for open collector communication. This allows one RS232 port to control up to 3 SiTech Dual Servo Controllers.
2. Extended Pic Servo Emulation (see http://www.jrkerr.com/psdata.pdf for protocol documentation).
3. Very simple, extended ASCII Protocol.
4. Custom Binary protocol for very fast and efficient (low communication overhead) control of your telescope.

Motor Encoder Resolution 2 Each: 32 bit (+/- 2 billion encoder ticks)
Scope Encoder Resolution 2 Each: 32 bit (+/- 2 billion encoder ticks)

Temperature Rating:
Storage: 0 deg's F to 160 deg's F
Operation 0 deg's F to 120 deg's F

Motor Encoder maximum read frequency: Up to 248 KHz. (May need external encoder pull up resistors for operation near maximum frequency, depending on the motor encoders installed).

Telescope Encoder maximum read frequency: Up to 25 KHz.

Disclaimer

Sidereal Technology provides 1 year warrantee period for replacement or repairing cost of the controller only. If the telescope knocks you off the ladder, the eyepiece gives you a black eye, or the telescope dumps the mirror, or any other disaster, Sidereal Technology or Dan Gray will not be responsible!

Warranty

The Sidereal Technology unit is warranted for a period of 1 year from the time of delivery. If the controller has been damaged by lightning, faulty wiring, moisture, or other misuse, the warranty is void. If the controller develops problems beyond the above mention reasons, Sidereal Technology will repair or replace the controller at Sidereal Technology's discretion, if within 1 year of delivery.

Technical Support

If you are using ScopeII and having issues, you need to contact Mel Bartels for Technical Support. Mel is extremely patient and will work with you until you are up and working, if necessary, he will provide telephone support.

If you are having problems with the controller, radio handpad, or the ASCOM driver, or are having problems with the Argo Navis™ controlling your telescope, Sidereal Technology will help you. We are available by email anytime. You will receive a response, generally within 8-10 hours, sometimes within minutes. You can also call Dan Gray anytime before 10:00 PM Pacific Time. His phone number is 503-887-3701.

grayarea@siderealtechnology.com

APPENDIX B - AGENCY APPROVALS

FCC:
This equipment has been tested and found to comply with the limits for a class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
* Reorient or relocate the receiving antenna.
* Increase the separation between the equipment and receiver.
* Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
* Consult the dealer or an experienced radio/TV technician for help.

The user is cautioned that changes and modifications made to the equipment without the approval of manufacturer could void the user's authority to operate this equipment.

C-Tick
Manufacturers Name: Sidereal Technology
Manufacturers Address: 6040 N. Cutter Circle #302
Portland, OR 97217 (USA)

Australia Representative Name:
Wildcard Innovations
ACA Suppliers Code: N11511

Declares that the products
Dual Servo Telescope Controller, Model DSTC 1
Handpad Transmitter, Model HT1
and Handpad Receiver, Model HR 1
conforms to the following standards:
EMC: AS/NZS 3548 Class B Complies
AS/NZS 61000-4-3 Complies, Criterion A
These units are not for mains connections.

Industry Canada
United Sates Company number: 5942A
Canadian Representative: Nova Astronomics
Canadian Representative Company number: 5930A

APPENDIX C: UNDERSTANDING AND USING THE X_BITS AND Y_BITS

This section is written for protocol developers only. You will not have to understand this section if you use the ServoConfig software.

The controller has two bytes that are used to change various options, or read the digital inputs from the handpad receiver. One byte is dedicated to the X axis (altitude), the other is dedicated to the Y axis (azimuth). Some of the bits in the X bits are common to both motors.

A byte is a group of 8 bits. The first bit, bit 0, has a "weight" of 1. The second bit, bit 1, has a "weight" of two. The 3rd bit, bit 2, has a weight of 4, and so on until the eighth bit, bit 7, has a "weight" of 128.

Bit 0: Weight of 1: Altitude Motor Encoder Direction
Bit 1: Weight of 2: Altitude Motor polarity
Bit 2: Weight of 4: Altitude Scope Encoder Direction
Bit 3: Weight of 8: A '1' here, puts the controller in the Drag and Track or Slew and Track Mode (**X Only**)
Bit 3: Weight of 8: A '1' here, puts the controller in the Slew and Track Mode **(Y Only)**
Bit 4: Weight of 16: A '1' here, puts the controller in the Platform mode (X Only). Digital Input from the Radio Handpad Receiver if you are using Servo Version 1.6 (Y Only).
Bit 5: Weight of 32: A '1' here, enables the handpad to control the motors. You must also set up the Slew and Pan rates (X Only). Digital Input from the Radio Handpad Receiver if you are using Servo Version 1.6 (Y Only).
Bit 6: Weight of 64 A '1' here, tells the controller you're using a "new" handpad, one with only push buttons, no switch (X Only). Digital Input from the Radio Handpad Receiver if you are using Servo Version 1.6 (Y Only).
Bit 7: Weight of 128: A '1' here, tells the controller it's in the "guide" mode (X Only). In the guide mode, when a direction button is pushed, the guide value is added or subtracted from the current speed. You must also set up the Guide rates. Digital Input from the Radio Handpad Receiver if you are using Servo Version 1.6 (Y Only).

To figure out what number to program into the X or Y bits, figure out the modes you want, then add up all the "weights" of the bits that are a '1', and that is the number you use.

As an example, lets say you want to use the drag and track mode, you're using a new handpad, you're altitude motor needs to be reversed and the altitude telescope encoder needs to be reversed.

Bit 1 = 2, reverse the motor polarity
Bit 2 = 4, reverse the scope encoder
Bit 3 = 8, Drag and Track mode
Bit 5 = 32, Enable the handpad
Bit 6 = 64, Using a new handpad.

This adds up to 110. Issue the command "XB110<E>". Now read back the value, "XB<E>", the controller responds with "B110". Now save this to flash by issuing the command "XW<E>".

Substitute 'Y' for 'X' for the azimuth motor.

Here is a description of the bits:
Bit 0: If this bit is a '0', the motor encoder is incremented normally. If this bit is a '1', then it is reversed.
Bit 1: If this bit is a '0', the motor moves one direction for a given number. If it is a 1, it moves the opposite direction.
This being said about bit 0 and bit 1, it is important to understand the implications of changing these bits. Lets say that the controller has just been powered up. In this state, before you've pressed a handpad button, or given the motor a command to move, the controller is trying very hard to keep the motor at position zero. If the position moves to 1, 2, or 3, the controller responds by putting an output to the motor to move it the opposite direction, and it will move back, 3, 2, 1, 0. If the motor starts moving in the other direction, -1, -2, -3, the controller responds by putting an output to the motor to move it back to zero, the opposite polarity as in the first example. Now try to think what would happen if the position moved to 1, 2, or 3, but the controller responded by moving the motor to even higher numbers. This would create positive feedback, and the motor will "run away" out of control. More Error would create more wrong voltage, which would create more error, etc, until maximum output voltage is reached.

Bit 0 and bit 1 of X_Bits and Y_Bits give you control over which direction the motor turns, and which way the encoder counts.

Bit 0 and bit 1 rules:

1. If the motor runs away, you can invert either one of the bits, bit zero or bit 1. The motor will now run properly.
2. If the motor runs properly, but in the wrong direction, you must invert both bits.

Bit 2: If this bit is a '0', the telescope encoder is incremented normally. If this bit is a '1', then it is reversed.

APPENDIX D: COMMUNICATION PROTOCOL

You will not need to read or understand this section unless you intend to write your own telescope driver.

List of ASCII commands (all get terminated with an <E> (CR)). Be sure to use upper case.
For the Azimuth or Right Ascension servo, use 'Y' instead of 'X' The # symbol means there is a number required.
The X servo is the Altitude or Declination. The Y servo is the Azimuth or Right Ascension.
If you change the Address of the controller ("AD1", "AD3", or "AD5"), the controller will respond differently. If the address is 3, then all beginning X's should be T's, and beginning Y's should be U's. If you've changed your address to Address 5, then use V and W for X and Y.

To change the address of a controller from the default (1) to (2), send the byte sequence: "AD2<E>" (Make sure any other controllers are not connected to the serial port first). If you want to make this permanent, you will have to write to the flash ROM. This would normally be the "XW" command, but since the controller address is 2 now, you would issue the command "TW".

If the controller is Address 1, and if you send an <E> with nothing before it, the controller responds with a lot of useful information as follows:
X# Y# XZ# YZ# XC# YC# V# T# X(A or M) Y(A or M) K#

Where:
X# is the location of the Altitude/Dec Axis in MOTOR encoder ticks
Y# is the location of the Azimuth/RA Axis in MOTOR encoder ticks
XZ# is the location of the Altitude/Dec Axis in SCOPE encoder ticks
YZ# is the location of the Azimuth/RA Axis in SCOPE encoder ticks
XC# is the Altitude/Dec motor current * 100.0
YC# is the Azimuth/RA motor current * 100.0
V# is the controller power supply voltage * 10.0
T# is the controller CPU temperature (Deg's F)
XA (or XM) is the Altitude/Dec Motor status, if XA, it's in Auto (normal) if XM, it's in manual.
YA (or YM) is the Azimuth/RA Motor status, if YA, it's in Auto (normal) if YM, it's in manual.
K# is the handpad Status. (Version 1.5 and later).

The handpad number is an 8 bit number where each bit represents a push button.
The handpad bits are as follows:
Bit 0 Left
Bit 1 Right
Bit 2 Up
Bit 3 Down
Bit 4 Speed (this is a toggle, a 1 is Pan mode and a 0 is slew mode).
Bit 5 Top Right key
Bit 6 Top Left key
Bit 7 Changed key.

The Changed key (bit 7) is used to tell if someone has pressed a handpad button since the last speed command is received. This feature is useful so a new tracking coordinate can be obtained, and the new tracking setpoint will track from the new position. This bit is automatically cleared if a new speed command is received.

About the Velocity:
Here are some "C++" functions that may help you calculate velocities:

```
double DegsPerSec2MotorSpeed(double dps, double TicksPerRev)
{
	return Round(TicksPerRev * dps * 0.09321272116971);
}
double MotorSpeed2DegsPerSec(double Speed, double TicksPerRev)
{
	if (TicksPerRev == 0.0) TicksPerRev = 1.0; // so we don't ever divide by zero
	return Speed / TicksPerRev * 10.7281494140625;
}
public static double TicksPerRev2SiderealRate(double TicksPerRev)
{
	return Round(TicksPerRev * 0.000389468194);
}
```

To really understand the speed relationship between the speed value and RPM, please have a look at the PicServo document, available at: http://www.jrkerr.com/psdata.pdf

ASCII Commands

Notes:
1. All commands MUST BE IN ALL CAPS! (Both ServoConfig terminal and also the Terminal in The SiTech Telescope driver "ToUpper" the commands for you).
2. The # character represents a decimal number that you must type
3. After each command, press the ENTER key.
4. All Altitude commands begin with an "X".
   All Azimuth/RA commands begin with a "Y"

----------COMMANDS----------

SB# Set Baud Rate (1 = 9600, 2 = 19200) Changes all controllers on serial bus. (1.5 or later)
X# Move Servo (-2147483648 to +2147483647) You can tag a speed command at the end.
	Example: **X**-2345**S**1000000<E>
X Returns the X position of the servo
XF# Forces the X position to be equal to the number (-2147483648 to +2147483647) (This stops the controller if moving)
XS# Velocity of X Servo (0-2147483647)
XS Returns the X velocity of the servo (what it would be if it was currently moving at fully accelerated speed, this may not be what it acually is at the moment).
XR# Ramping speed or Acceleration of X servo (0-3900)
XR Returns the Ramp speed.

XP# Proportional Band of X servo (0-32767)
XP Returns the Proportional band
XI# X Integral (0-32767)
XI Returns the Integral

XD# X Derivative (0-32767)
XD Returns the Integral
XE# Maximum position error limit before servo turns off (0-32767)
XEL Returns position error limit.
XE Returns the position error of the servo
XO# X Output limit (0-255)
XO Returns the PWM output of the servo (0-255)
XC# X Current Limit (0-240 = 0-2.40 amps)
XC Returns the X motor current * 100 (240 MAX)
XM# X to manual mode, the number is the PWM value, -255 to +255.
XA X to Auto mode.
XN X Normal Stop (ramps down, then stops. Automatically clears when new position is given)
XNT Just like XN, except when it slows down enough, it starts tracking in the DragNTrack or SlewNTrack mode.
XG X Emergency stop (stops immediately)
XL# Set Integral Limit (0-24000)
XL Returns Integral Limit
XB# Number sets the servo bits like direction, etc. (0-255)
XB Returns current X Bits Value.
XZ# Forces the scope encoder position to be equal to the number (-2147483648 to +2147483647)
XZ Returns the scope encoder position

The following commands don't have a 'Y' command, only X. They affect both servos
XK Returns the handpad info in Decimal.
XH Returns the temperature of the CPU chip (in deg's F)
XV Returns the firmware version * 10.
XJ Returns the motor power supply voltage * 10 (please divide returned number by 10)
XQ Resets the servo system (both)
XU Programs factory defaults into the flash ROM.
XW Writes the configuration of both the x and y parameters from working RAM to the flash ROM.
XT Reads the configuration from the Flash Rom into the working RAM.
FC Flash Configure, Send 128 more bytes, then 2 bytes of checksum (Simple Addition of all data bytes sent), and it will write the new data to configuration. It also reads the new data into the working RAM.

SC Send Configuration, The controller Sends 128 bytes, followed by two bytes of checksum. This data is what is in the flash, not necessarily what is in the working RAM.

All of the following XX extended commands have a corresponding read command
as an example, if you type "XXL<E>" it responds with the latitude.

XXL# Store the latitude to the controller (4500 = 45 deg's north, -4500 = 45 deg's south)
XXZ# Store the azimuth encoder ticks per full circle.
XXT# Store the altitude encoder ticks per full circle.
XXU# Store the number of encoder ticks for the Altitude Motor Encoder to the controller
XXV# Store the number of encoder ticks for the Azimuth Motor Encoder to the controller
XXA# Stores the Altitude Slew Rate to the controller
XXB# Stores the Azimuth Slew Rate to the controller
XXC# Stores the Altitude Pan Rate to the controller
XXD# Stores the Azimuth Pan Rate to the controller
XXE# Stores the Platform tracking rate to the controller
XXF# Stores the Platform up/down adjuster to the controller
XXG# Stores the Platform Goal to the controller
XXH# Stores the Altitude Guide Rate to the controller
XXI# Stores the Azimuth Guide Rate to the controller
XXJ# Stores the PicServo Timeout Value to the controller (Seconds).
XXQ# Turns off or on the digital outputs of the Radio Handpad Receiver
XXN# A zero turns off the Argo Navis™ mode. A one turns it on.
XXK# Local Search Radius in arc Minutes
XXM# Local Search Speed in arc seconds per second.
XXO# Altitude Backlash
XXP# Azimuth Backlash
XXW# Backlash Motor Speed (Speed while in the backlash area) Should be 200,000 to 500,000.
XXX# Altitude in deg's times 100. The altitude should be sent to the controller by the host computer (ScopeII, the Ascom driver, etc) after initialization. Then the Local searches will be round, and the Pan and Guide speeds will be linear in relation to the sky (version 1.7 or later).
XXY# Sets the Azimuth/RA in the controller. Not useful until version 1.8.

When the controller receives a 'Q' as the first character in a line, it responds emulating the Tangent "Q" protocol. The encoder resolution is always 18000 (the controller uses the configured values for encoder ticks per revolution for the Scope Encoders, and converts it to 18000).

0AAh Puts the servo into the PicServo Emulation mode (no CR required). See http://www.jrkerr.com/psdata.pdf for a description. There are some items different, all motion control is the same. The differences are described below.

The following two ASCII commands with Binary data are for reading and controlling the servo controller in a very efficient manner. They have been added at version 1.5 of the controller firmware.
XXS<E>
The controller responds with 21 bytes (32, starting with firmware version 2.0) of information. Four long ints of position, then other bytes for keypad status, X and Y Bits, and other bits as follows:
Byte 0-3 Current Altitude/Dec Motor Position (least significant byte first)
Byte 4-7 Current Azimuth/RA Motor Position.
Byte 8-11 Current Altitude/Dec Scope Encoder Position
Byte 12-15 Current Azimuth/RA Scope Encoder Position.

Byte 16 is Keypad Status (see the top of this section for a description of bits)
Byte 17 is XBits
Byte 18 is YBits
Byte 19 is Extra bits

Bit 0 - Set if Altitude/Dec motor is stopped
Bit 1 – Set if Altitude/Dec motor is in manual
Bit 2 – Digital Input 0 starting version 2.0
Bit 3 – Digital Input 1 starting version 2.0
Bit 4 - Set if Azimuth/RA motor is stopped
Bit 5 – Set if Azimuth/RA motor is in manual
Bit 6 - future
Bit 7 – future

Byte 20 is a 1 byte checksum of the previous 20 bytes (simple addition) in versions prior to version 2.0. For versions starting with firmware version 2.0, there are 11 additional bytes as follows:
Byte 20-21 Analog input 0
Byte 22-23 Analog input 1
Byte 24-27 32 bit millisecond clock.
Byte 28 Temperature (deg's F)
Byte 29 Input voltage (volts * 10)
Byte 31, low byte of checksum, non inverted
Byte 32, High byte of checksum, inverted.

XXR<E>
The controller now waits for an additional 20 bytes (21 bytes starting version 2.0) from a host as follows:
Byte 0-3 New Altitude/Dec Motor Position Goal (least significant byte first)
Byte 4-7 New Azimuth/RA Motor Position Goal.
Byte 8-11 New Altitude/Dec Motor Speed
Byte 12-15 New Azimuth/RA Motor Speed
Byte 16 Various bits

Bit 0 – if 0, the following new XBit and YBit values are ignored. If '1', the new XBit and YBit values are used.

Bit 1 -7 future

Byte 17 New XBits value (must have previous bit 0 set)
Byte 18 New YBits value (must have previous bit 0 set)
Byte 19 Checksum of all above data bits (simple 8 bit addition) (LSByte of 16 bit checksum, starting version 2.0)
Byte 20 Checksum high byte (starting with version 2.0), this byte must be the high byte of a 16 bit addition of all prior bytes, but this high byte is inverted.
The checksum doesn't include the “XXR<cr>”

After sending the XXR command, the controller will respond with a binary info response, same as the response to the XXS command.

If you use the above binary command to control the servo controller, be sure to have a restart communication if the command gets out of sync by a communication fault. If you receive a bad checksum, you should resync communication by sending a few of XXS commands.

The flash ram values are loaded from the flash Rom to Ram on reset.
Both the X and Y parameters are stored to the flash ROM.

Use X and Y for module address 1.
If the module address is 3, use 'T' and 'U', for address 5, it's 'V' and 'W'.

Description of the bits for the XBits and YBits:
0 if 1, the motor encoder is incremented the other direction
1 if 1, the motor polarity is reversed
2 if 1, the azimuth (or altitude) encoder is reversed
3 if 1, (x only) we're in the computerless drag and track mode
3 if 1, (y only) we're in the computerless slew and track mode (no clutches, must use handpad to slew) (must be in drag and track too)
4 if 1, (x only) we're in the tracking platform mode
5 if 1, (x only) we enable the hand paddle
6 if 1, (x only) hand paddle is compatible with New Hand Paddle (allows slewing in two directions, and guiding)
7 if 1, (x only) we're in the guide mode. The pan rate is added or subtracted from the current tracking rate.
Note about the Guide Mode: If PC software puts the controller in the guide mode, and you are controlling an Equatorial mount, you must give the declination motor a destination, even if you want it to be stopped. If you want the declination axis to be normally stopped, give the declination a far far away goal, and set the speed to zero. This is because the controller adds or subtracts from the current speed and direction, and the controller gets confused if the axis is at its destination. This is changed inf servo version 1.6, and this is automatically done for you in the controller.
The Y_Bits bits 4-7 are the digital input status for the 4 digital inputs from the radio handpad receiver. This is only available if you're using servo version 1.6 and also if you have a handpad receiver connected.

PicServo Mode Differences:
If you intend to write a telescope driver for the SiTech controller, we recommend you use the new binary protocol mentioned above, as it is simpler and faster. If you have existing software for the PicServo chips, you can use the PicServo protocol, but there are a few differences as described here:

The controller address must be set beforehand, and saved to flash ROM. The PicServo address commands are ignored with the SiTech controller. When the controller is set for address 1, the altitude motor will be addressed with address 1, and the azimuth with address 2. You can set address using the ASCII command AD1, AD3, or AD5. You then save the address to flash ROM using the following ASCII commands. “XW” for address 1, “TW” for address2, and “VW” for address 3. This must be done with only 1 controller connected at a time. You can also set the address of the controller using the ServoConfig software.

When the Define Status specifies the Home position for return, the “HOME” position returns the location of the telescope encoder location, instead of the HOME position.

When the Define Status specifies the A/D to be returned, it returns the motor current times 10 instead. Bit 6 of the Define Status must be clear.

When you set bit 6 of the Define Status, the motor voltage will be returned in the A/D location if it is an odd address for the controller.

When you set bit 6 of the Define Status, the CPU temperature will be returned in the A/D location if it is an even address for the controller.

The ReadVersion bit of the Define Status returns the handpad status instead of the version. The handpad bits are as follows:
The handpad number is an 8 bit number where each bit represents a push button.
Bit 0 Left
Bit 1 Right
Bit 2 Up
Bit 3 Down
Bit 4 Speed (this is a toggle, a 1 is Pan mode and a 0 is slew mode).
Bit 5 Top Right key
Bit 6 Top Left key
Bit 7 Changed key.
The Changed key (bit 7) is used to tell if someone has pressed a keypad button since the last speed command is received. This feature is useful so a new tracking coordinate can be obtained, and the new tracking setpoint will track from the new position. This bit is automatically cleared if a new speed command is received.

Several more functions have been added to the PicServo protocol. They are accessed using the PicServo NOP function with various lengths specified.

Length of 0 (send a 0Eh. This is the same as the PicServo NOP function)

Length of 1 (send a 1Eh for the PicServo function, then another control byte).
If Control byte = 4, the controller reads the configuration from flash ROM
If Control byte = 8, the controller writes the current configuration to flash ROM
If Control byte = 16 the controller writes the factory defaults to the flash ROM
If Control byte = 32 the controller reverts to the ASCII mode.

Length of 2 (send a 2Eh for the PicServo function, the control byte, then another data byte.
If Control byte = 2 the controller will update the Xbits (or Ybits if even address) with the data byte.

Length of 5 (send a 5Eh for the PicServo function, a control byte, then four data bytes.
If Control byte = 1 the controller will update the Altitude (Azimuth if even address) location with the next 4 bytes.

If Control byte = 64 the controller will update the Altitude (Azimuth if even address) guide rate with the next 4 bytes.

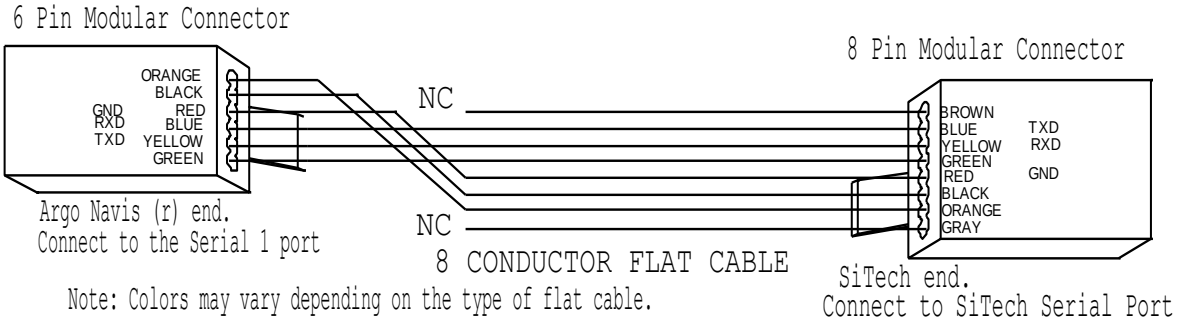
Length of 9 (send a 9Eh for the PicServo function, a control byte, then eight data bytes.
If Control byte = 32 the controller will load the next two long ints (8 data bytes) into the Slew rate and the Pan rate. For odd address it is the altitude, for even addresses it is the azimuth.

If the controller hasn't received a PicServo command for more than the PicServo timeout period, the controller will stop any motion, and revert to the ASCII mode. The PicServo timout is set with the XXJ command, and the units are Seconds. Set to zero to disable PicServo timout.

APPENDIX E: CABLE DIAGRAMS

Diagram of communication cable for the Argo Navis™

This is for the early controller version:



Later Controller Version (4 x 4 connector at SiTech end) For Argo Navis

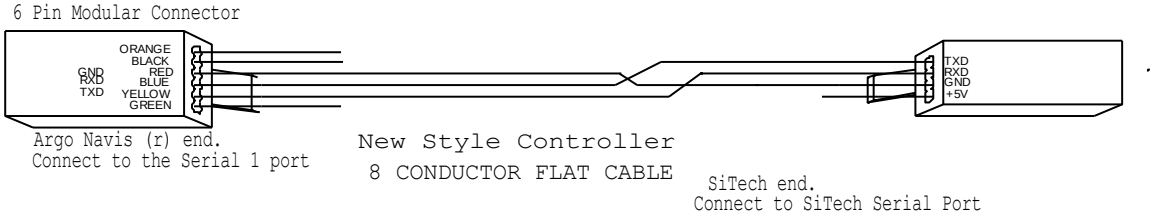
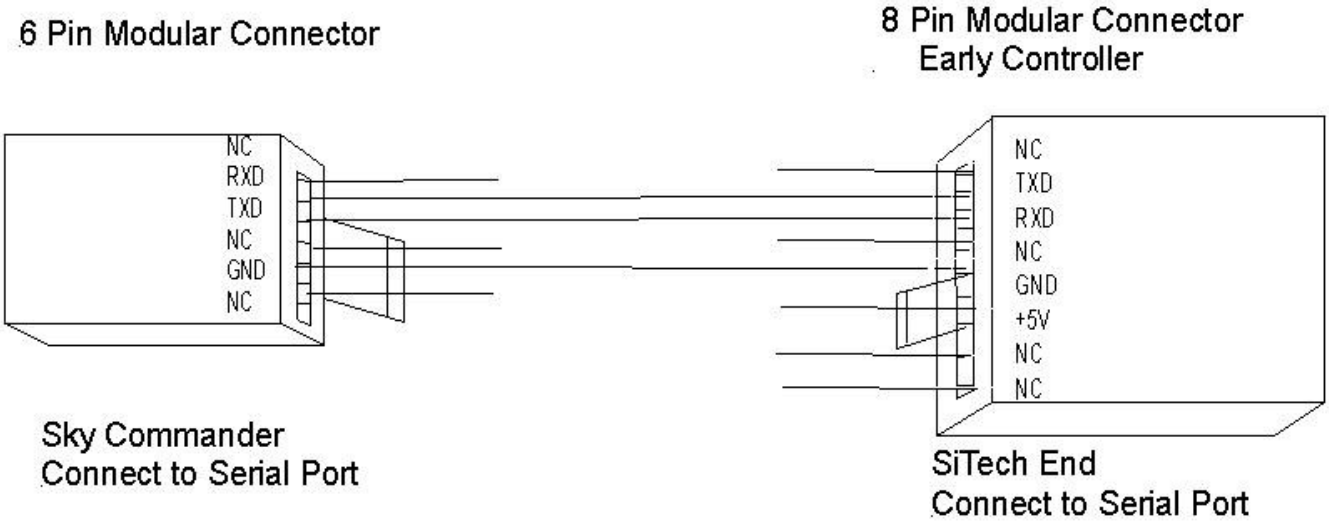


Diagram of communication cable for the Sky Commander™

Early Controller Version (8 x 8 connector at SiTech end) For Sky Commander



Later Controller Version (4 x 4 connector at SiTech end) For Sky Commander

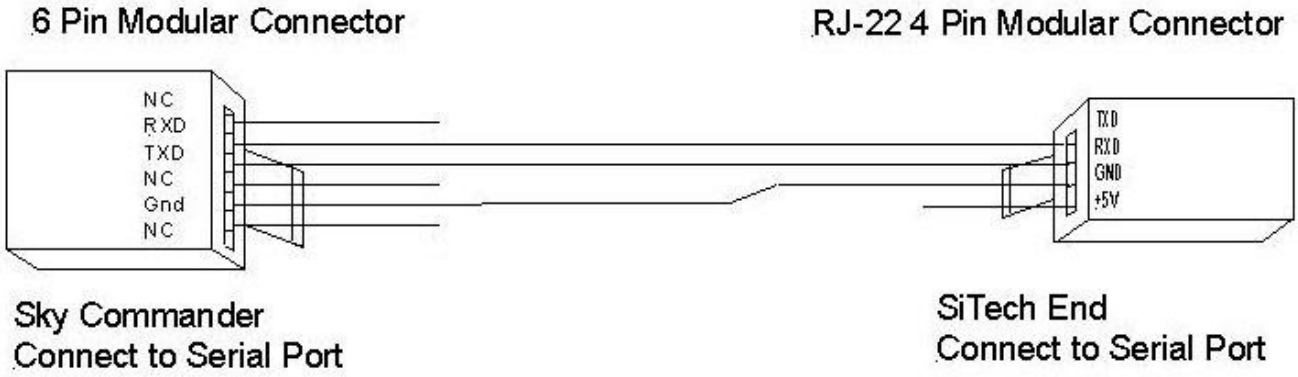
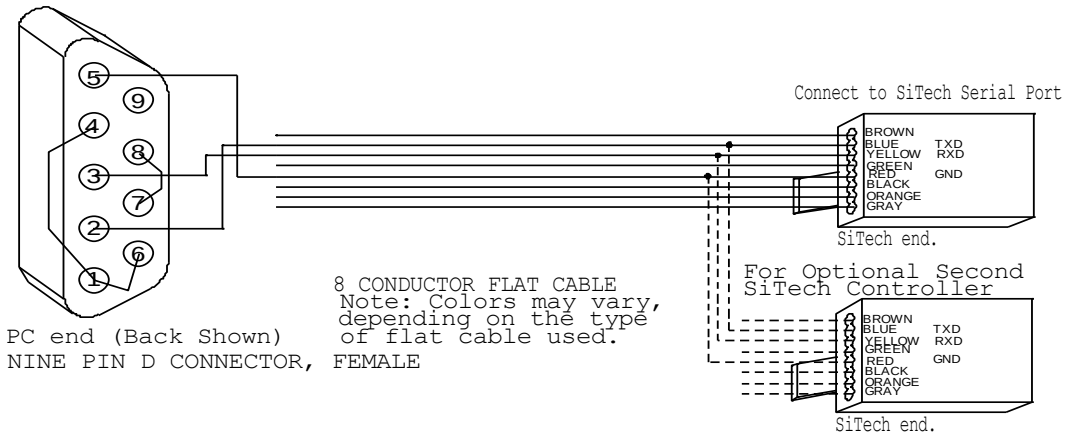
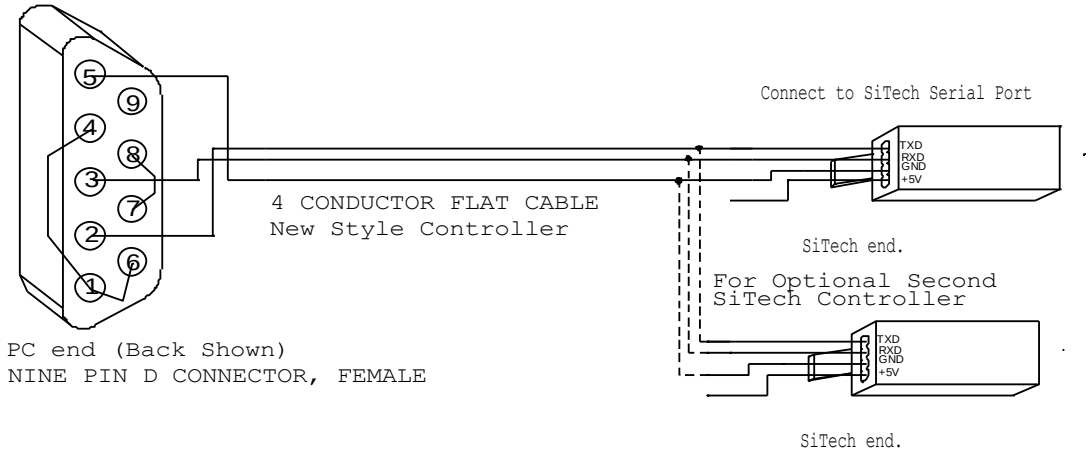


Diagram of communication cable for connecting to a Personal Computer

Early Controller Version (8 x 8 connector at SiTech end)

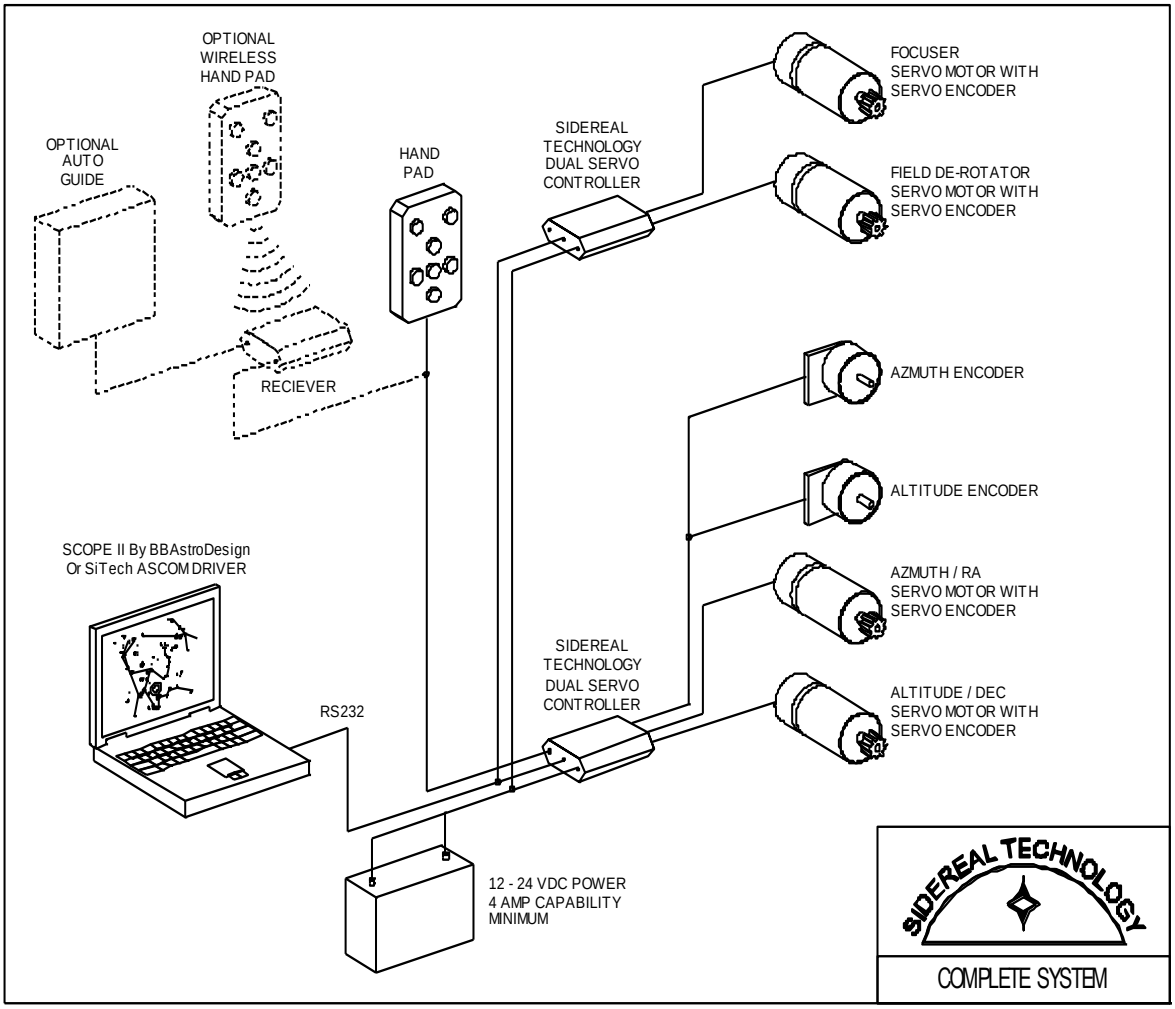


Later Controller Version (4 x 4 connector at SiTech end)

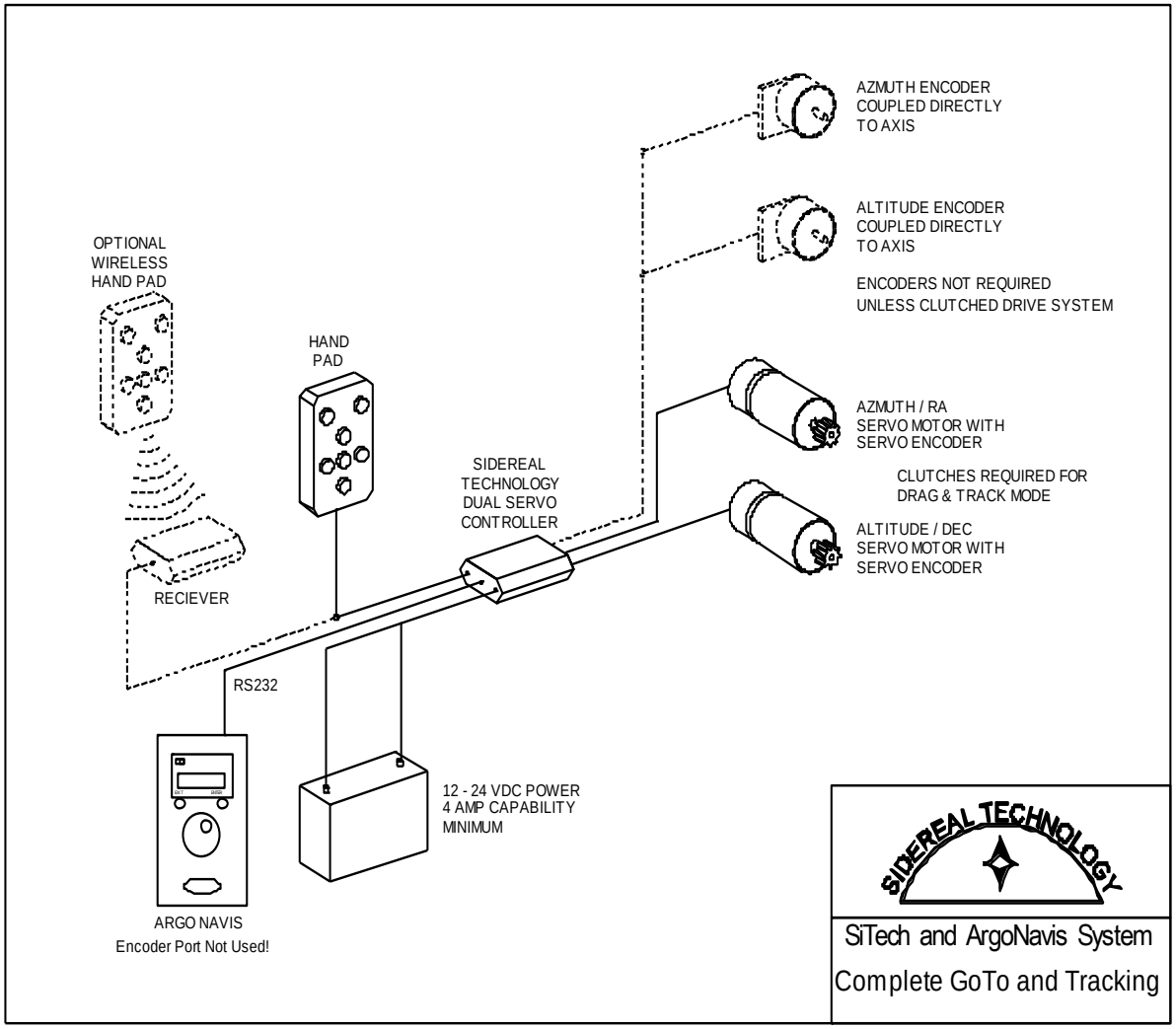


APPENDIX F: BLOCK DIAGRAMS OF SOME POSSIBLE CONFIGURATIONS

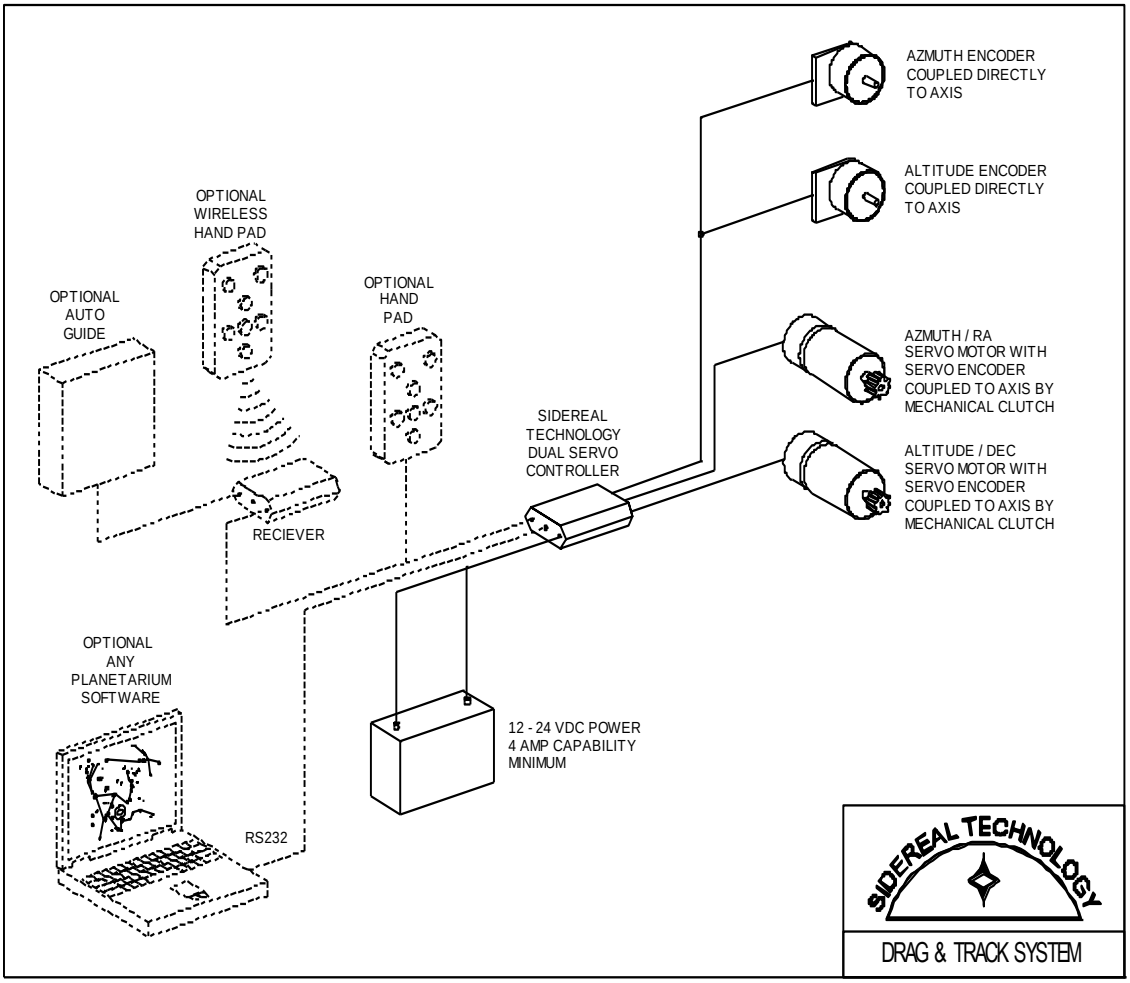
Complete System:



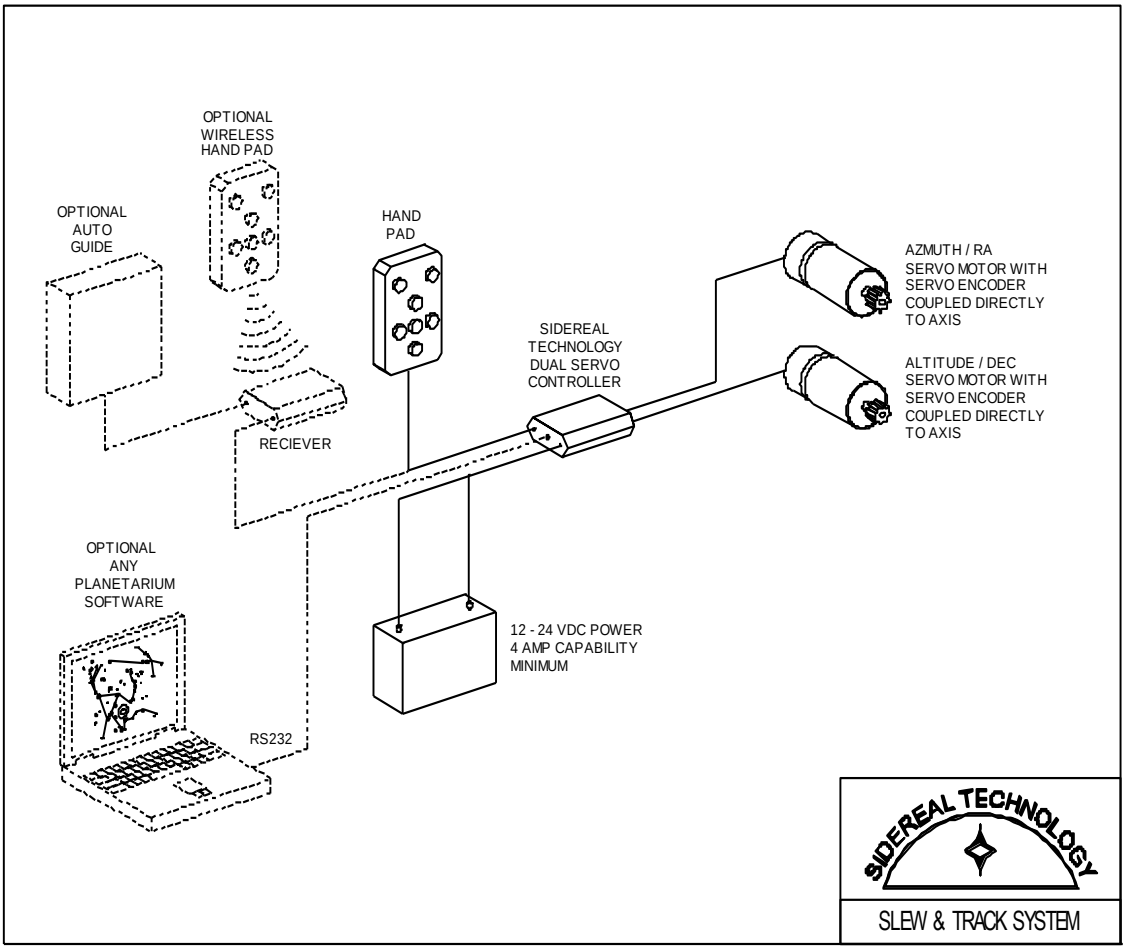
System using Argo Navis™ for full GoTo and Tracking



DragNTrack System



SlewNTrack System



APPENDIX G: WIRING DIAGRAM FOR EARLIER CONTROLLER

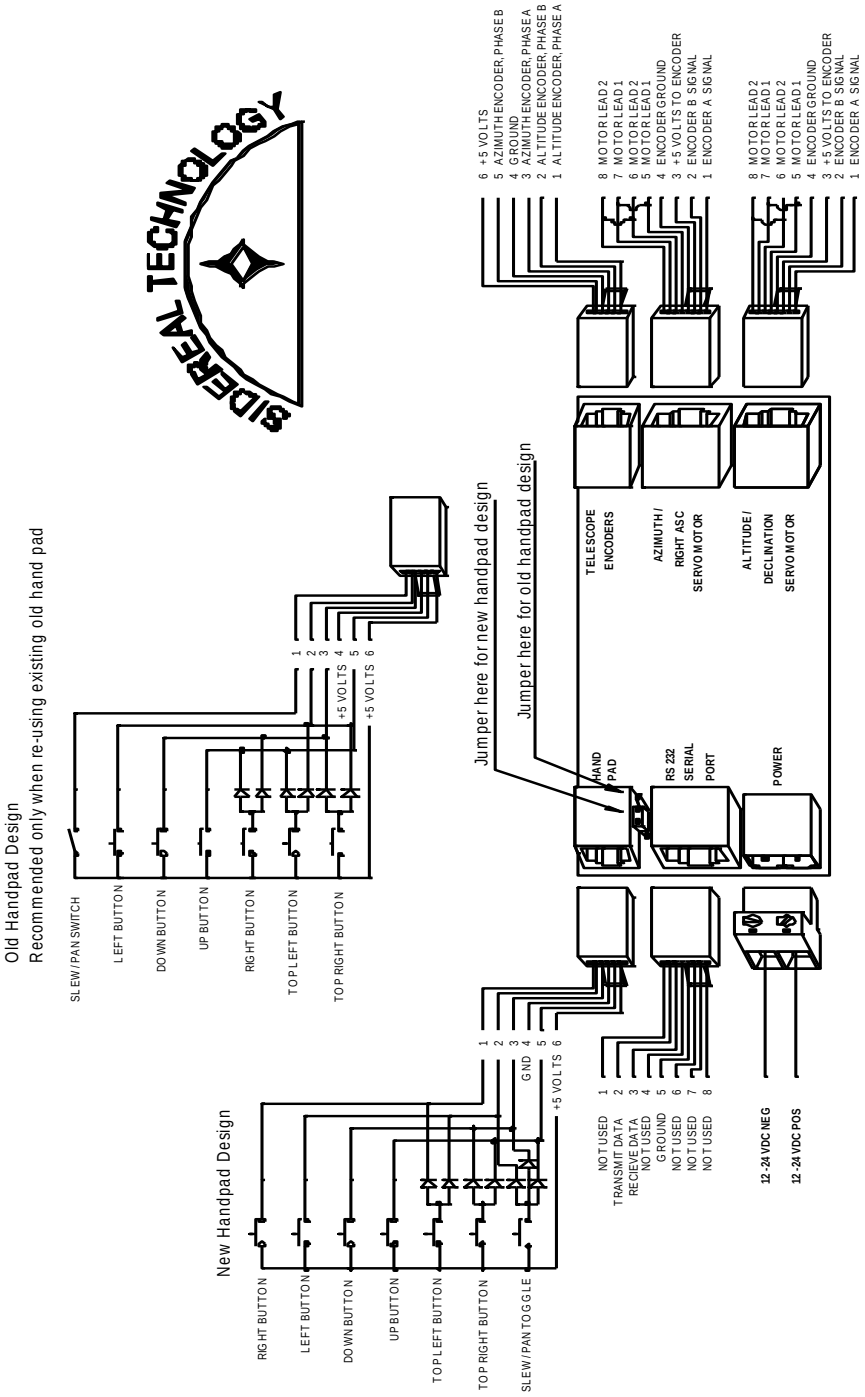
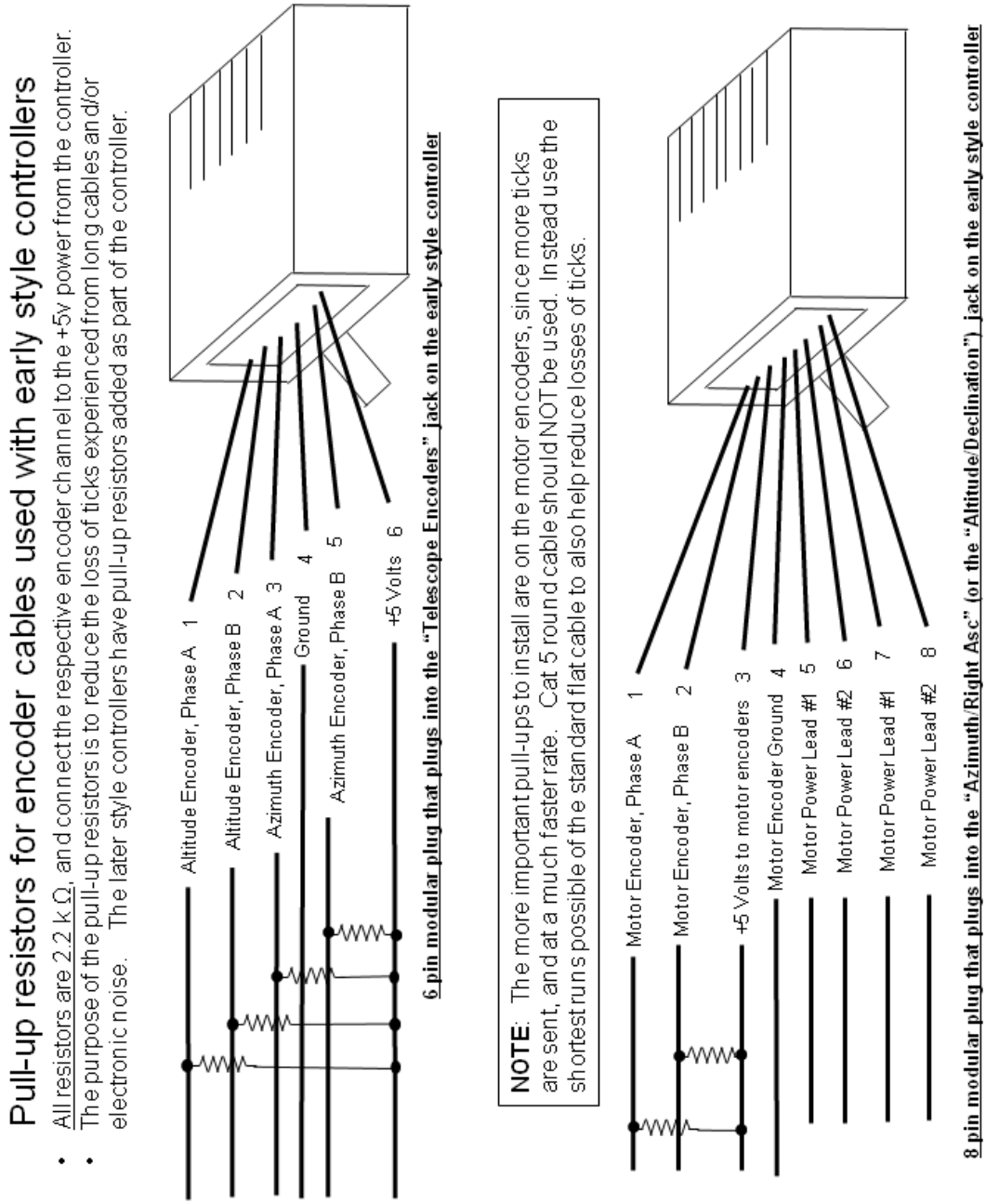
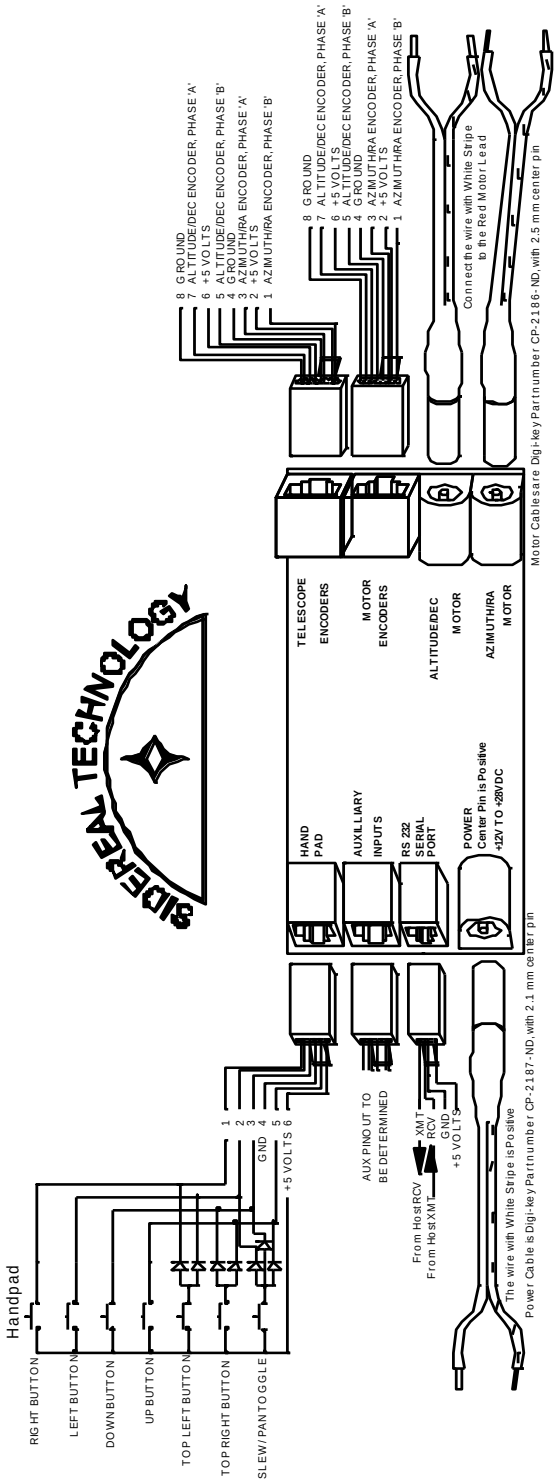


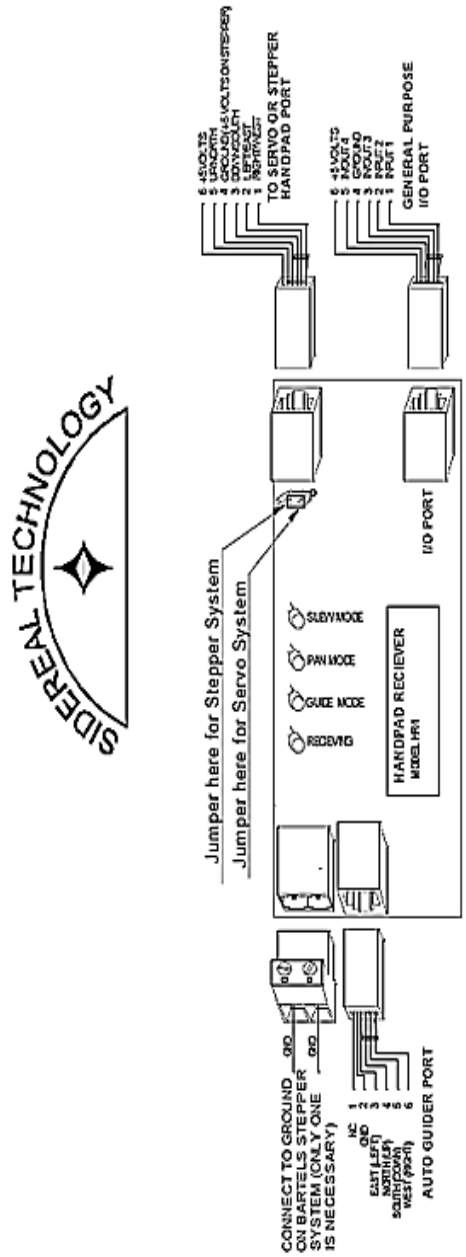
Diagram for Pull-Up Resistors for Early Controller Encoder Cables



APPENDIX H: DIAGRAM FOR LATER CONTROLLER



APPENDIX I: DIAGRAM for Radio Hand Pad Receiver



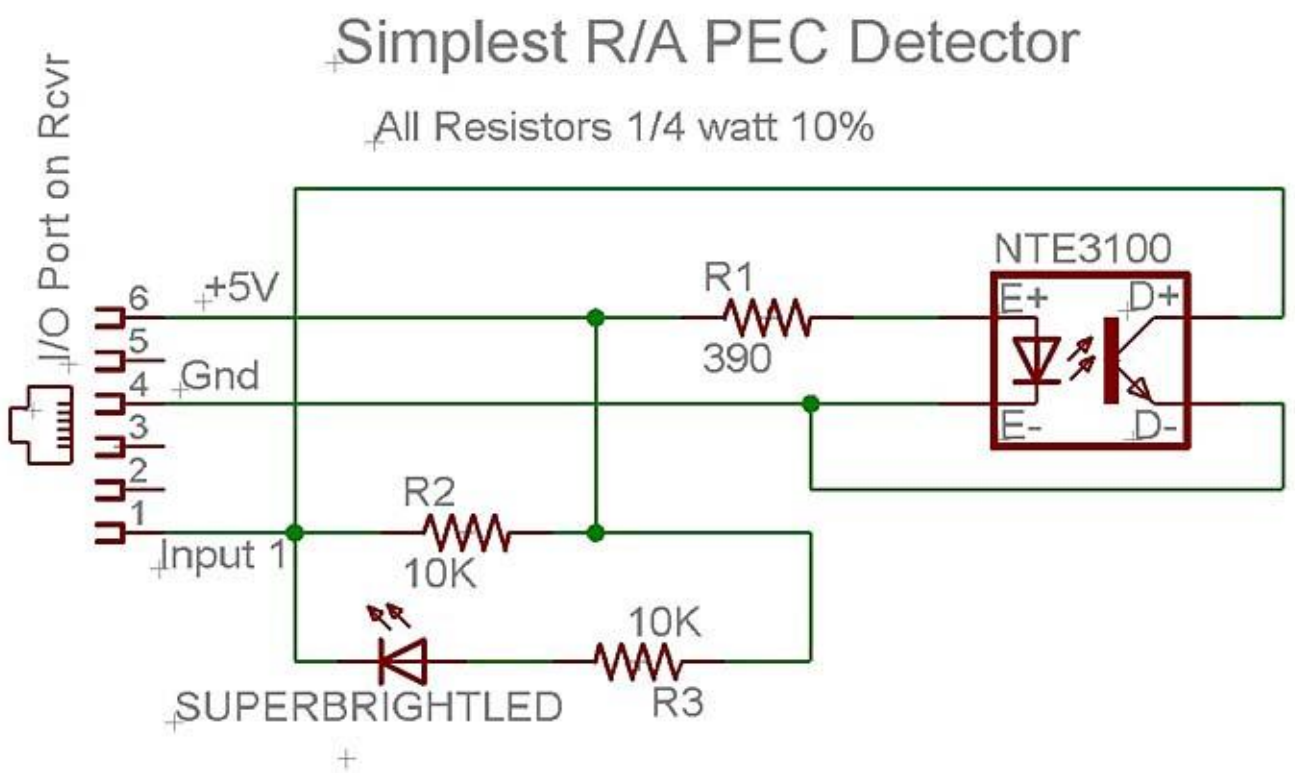
Note, the auto guider port is looking for the direction pins to be pulled down to ground, in the same manner as an SBIG auto guider interface uses. The plug allows simply plugging in an SBIG auto guider or relay box.

Pin 1:not used; Pin 2:Ground; Pin 3:East/Left; Pin 4:North/Up; Pin 5:South/Down; Pin 6:West/Right

APPENDIX J: AutoPECSynchSensor Schematics

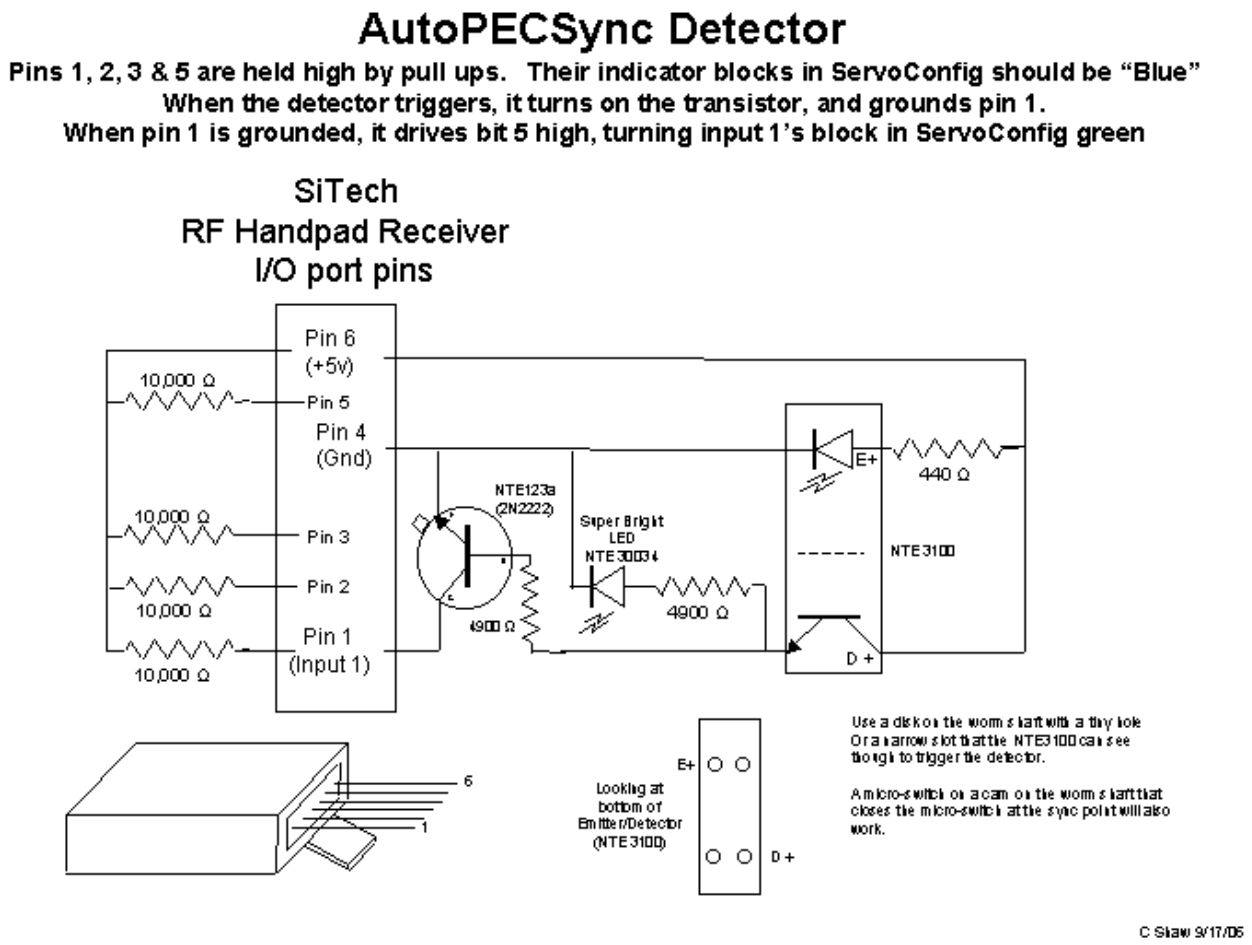
Simplest RA PEC Detector

Uses NTE3100 with a disk that has a slot or hole that rotates past sensor to trigger.



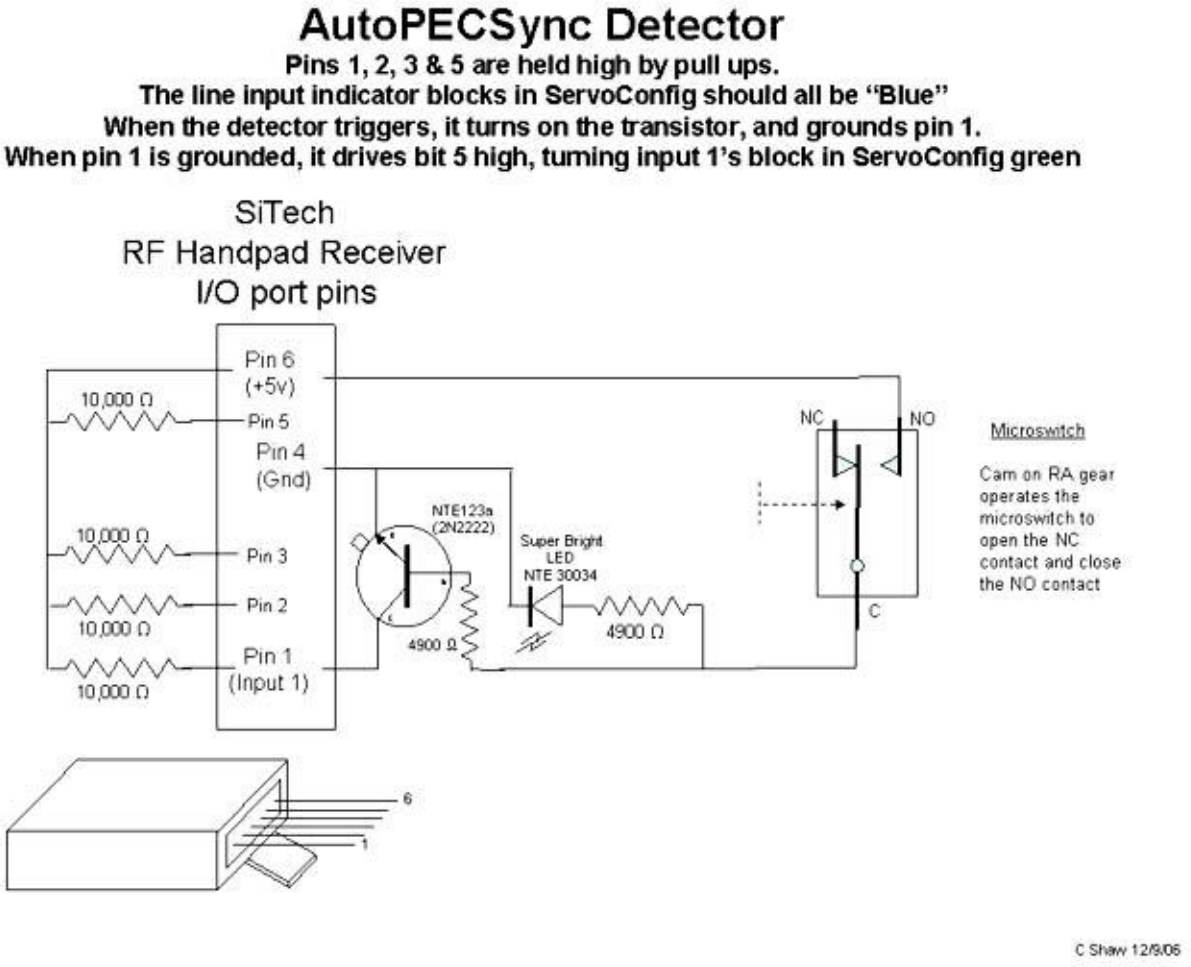
Optical PEC Detector using a transistor

Uses NTE3100 with a disk that has a slot or hole that rotates past sensor to trigger. This circuit uses a transistor to signal output.



Microswitch PEC Detector

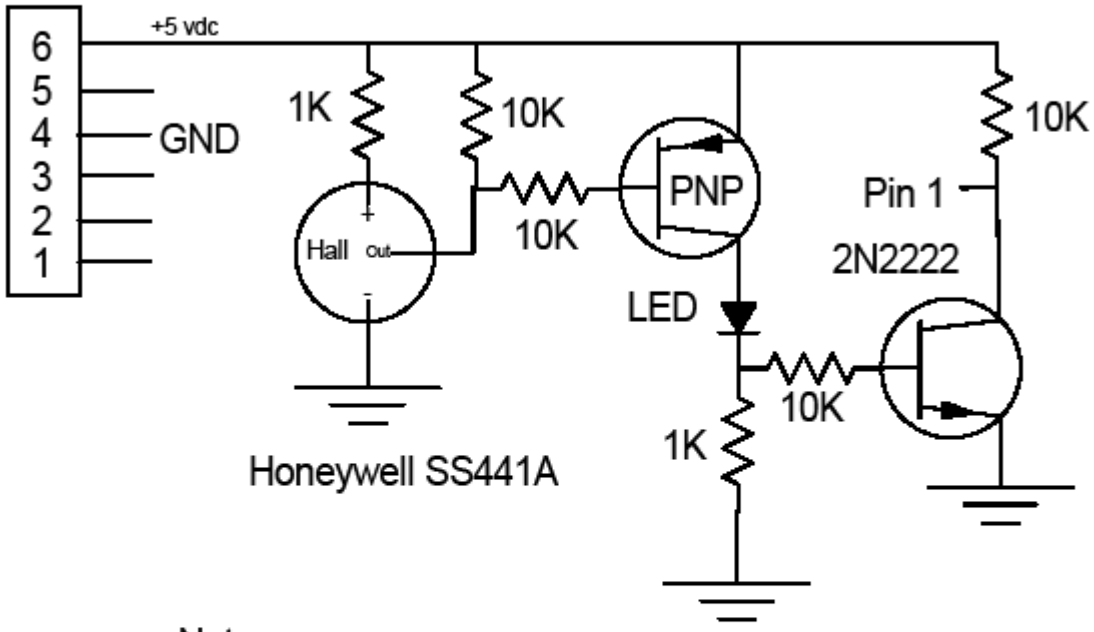
Uses cam action on worm to trigger switch as a sensor.



Hall Effect Detector

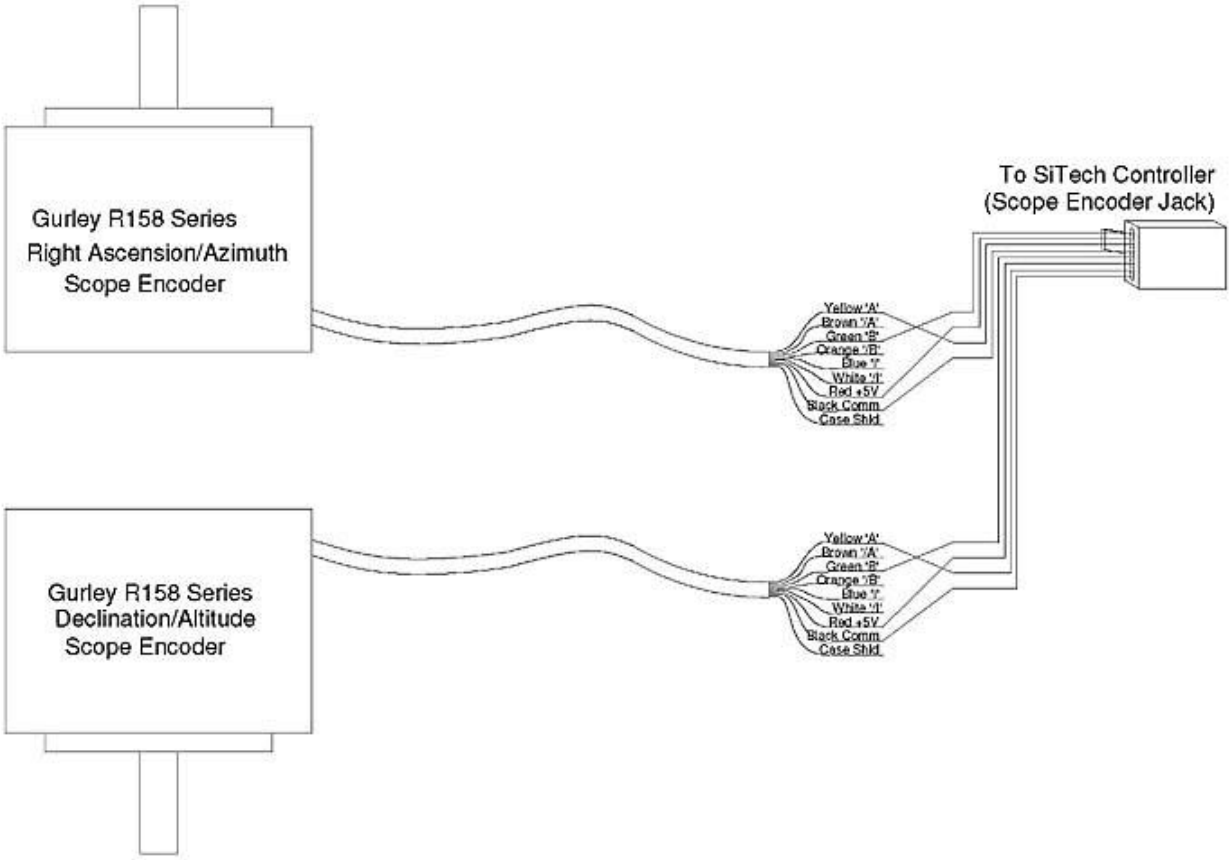
Uses passing magnet to trigger PEC Synch. This circuit does not require touching the worm or shaft with a switch/cam or mounting set of optical sensors on each side of a disk.

Hall Effect AutoPECSynch Detector

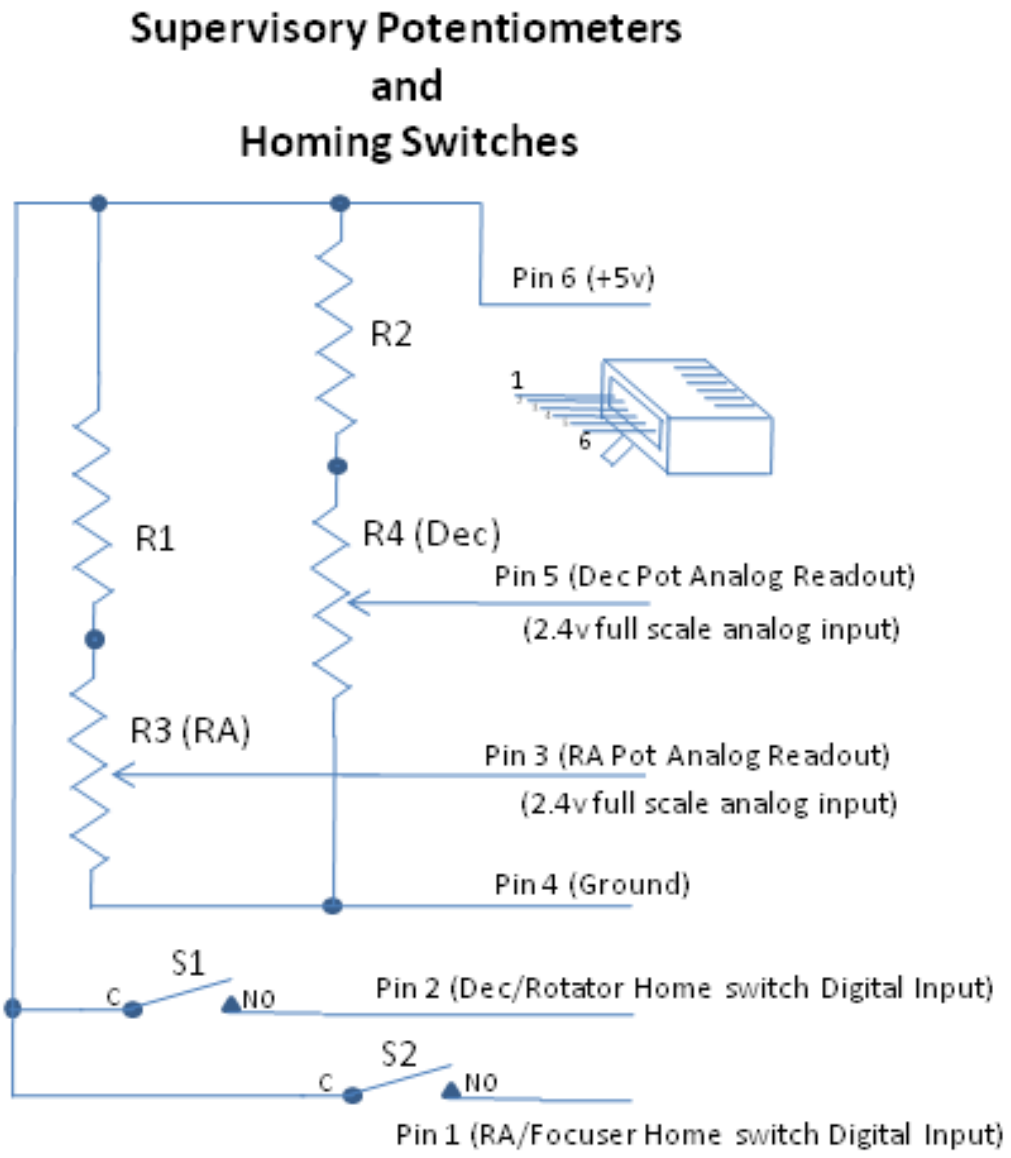


Notes:
Hall sensor "Solid State Switch SS441A" available from Newark Electronics
Mount Hall sensor within 1/16" of magnet. I used GOOP to glue sensor in place on aluminum structure.
Orientation of Hall sensor and magnet pole is critical, you have a 1 in 4 chance of getting it to work. Reversing either the magnet or the Hall senso and it will not work.
LED will light when Hall is triggered by magnet.
PNP transister available at Radio Shack

Appendix K: Wiring Diagram for Gurley High Resolution Encoders



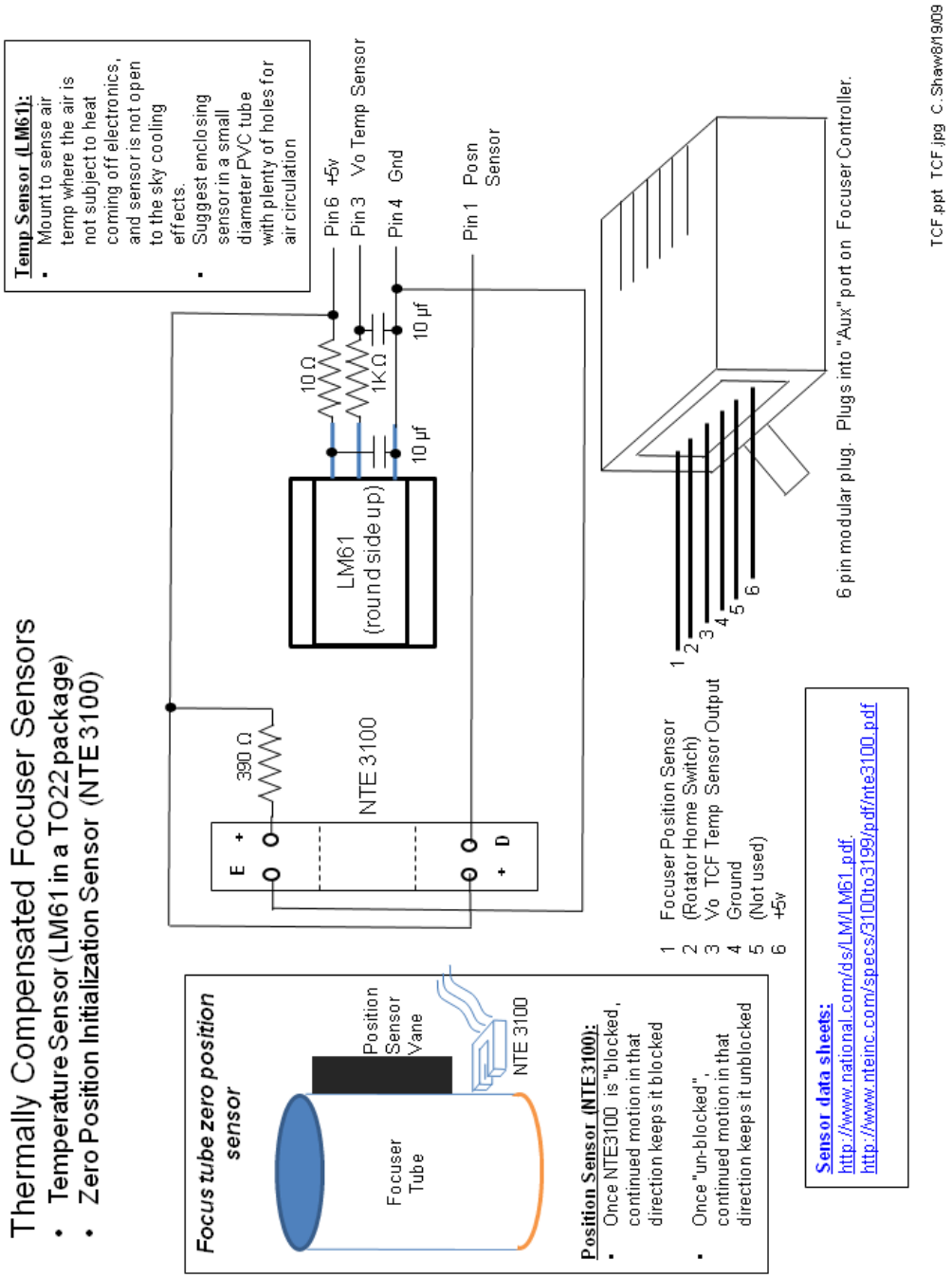
APPENDIX L: SUPERVISORY POTENTIOMETERS DIAGRAM



R1, R2	2.1k ohm	Digikey Part # MFR-25FBF-2K10
R3, R4	2.0k ohm pot, single turn, no mechanical stop	
	NTE 502-0081	
	or Spectrol 357-0-0-1S22-202	
S1, S2	Microswitches for detecting home positions	

Appendix M: Thermally Compensated Temp Sensor and Zero Position Init Sensor

The critical thing in the temp sensor is the mV/DegC. An LM62 has a sensitivity of 15mV/DegC, but only comes in an SOT23 package rather than the more convenient TO-92. The accuracy of the LM61 is not great, but the specs say the long term stability (assuming that relates to re-producability) is +/- 0.2 degC.



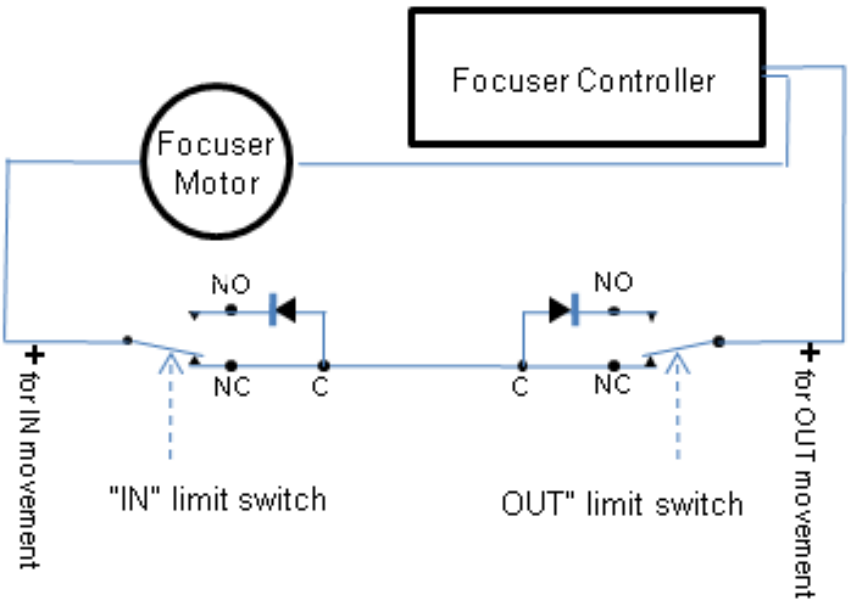
APPENDIX N: LIMIT SWITCHES

If your mount or focuser has the ability to hit a hard stop or obstruction, it may be wise to use limit switches. These switches have nothing to do with the software. Using a focuser as an example (could be the RA axis also, etc.) the idea is a microswitch will sense when the hardware has moved as far as you want it to move before a collision. At that point, power to the motor will simply be interrupted by "opening" the microswitch.

The controller will sense this has happened because it may still be commanding the motor to move, but will know it has been stopped due to the info it receives from the motor encoder. Seeing this, it will put the motor into manual mode (Blinky mode). The LED on the controller will start blinking, and the bottom MSG button in the SiTech window will change from automatic to blinky mode.

To move the motor in the opposite direction, away from the impending collision after the microswitch interrupted power to the motor, the diodes in the circuit below will allow current to flow only in the direction to back away from the collision. There are two limit switches in the circuit, one for each end of travel. The circuit diagram is labeled assuming it is being used on a focuser. NO stands for Normally Open, and NC stands for Normally Closed. C stands for Common.

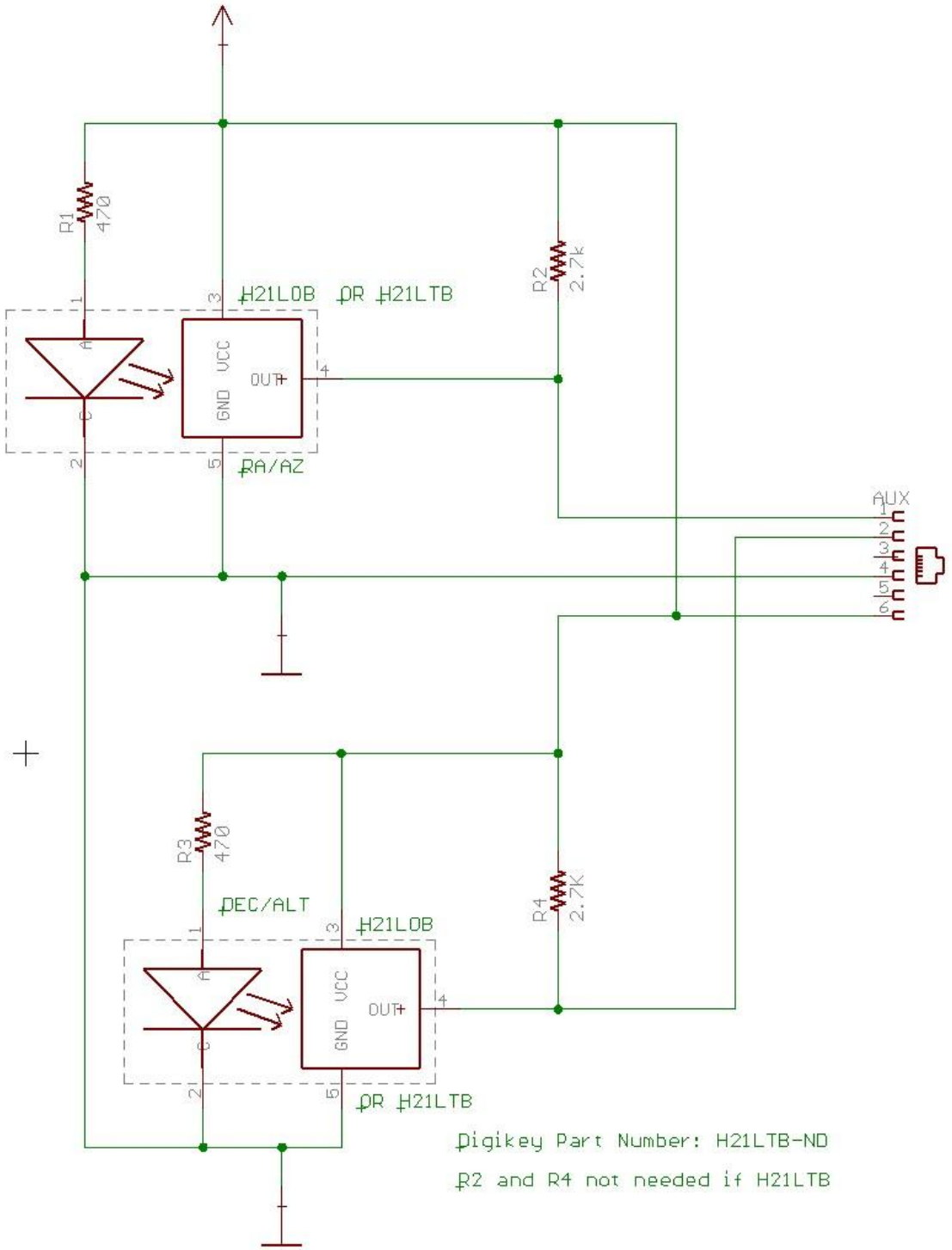
Rig your microswitches so that they are "adjustable" for fine tuning when they interrupt motor power. To test them, use a handpad and drive the motor in the direction of the collision, and manually trip (and keep tripped) the microswitch. The motor should stop, and shortly thereafter the controller will go to blinky mode. Release the handpad button, and put the motor back into automatic by clicking on the bottom button in the SiTech window, or pressing the top right and left buttons on the handpad simo. Then use the handpad to drive the motor away from the collision. If it does not move, you have the diode wired backwards…..



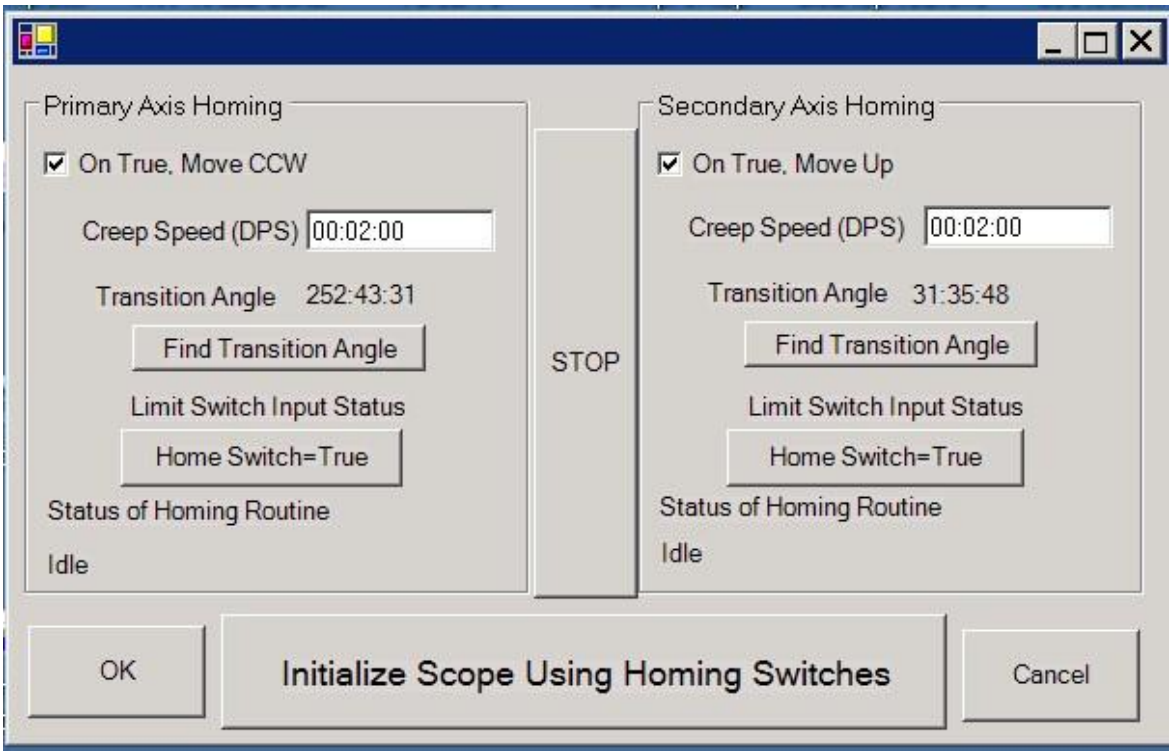
Appendix O: Homing Switch Circuit and Setup Screen

The homing switches are the same as the switches on the supervisory pot schematic.

There's a new, really nifty sensor shown on the schematic (the Digikey part number is also on the schematic), that has an output amp built in, so no need for resistors, except for the LED side of it



Here's a screen shot of the setup screen that is available as of SitechEXE.exe version 55E:



You just click on the BIG button at the bottom, and the scope will seek the switches, then slow down, back up, then do a final creep at the creep speed you select.

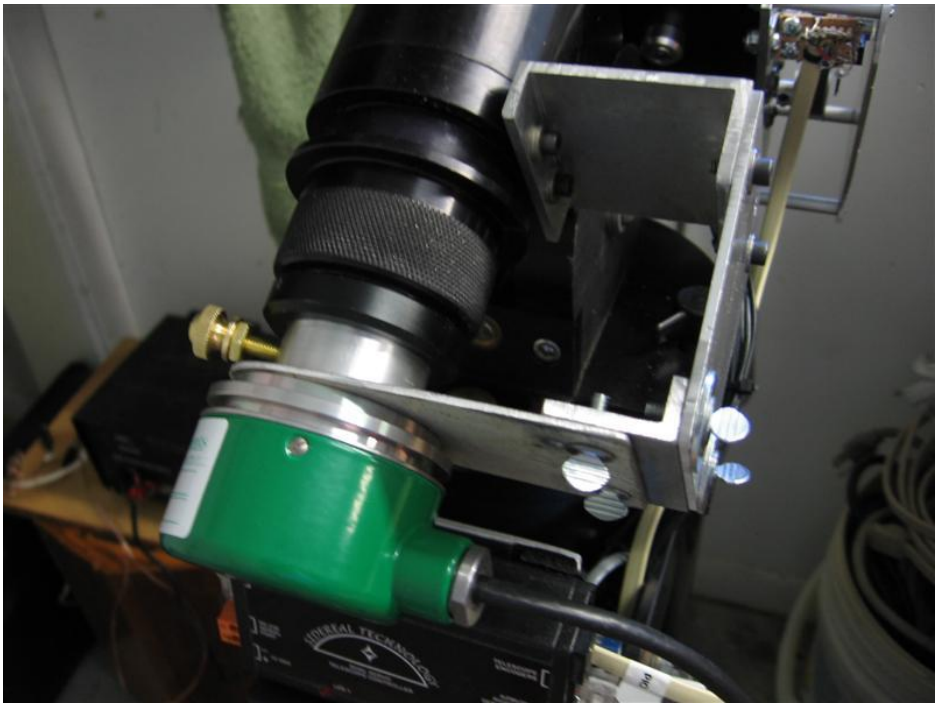
Before you use it, you've got to "find transition angle" for both axis's. This should be done after you do a model, and everything is synched up, etc. It saves (and loads) the current PXP file, similar to the park. It's filename is HomePointXP.pxp.

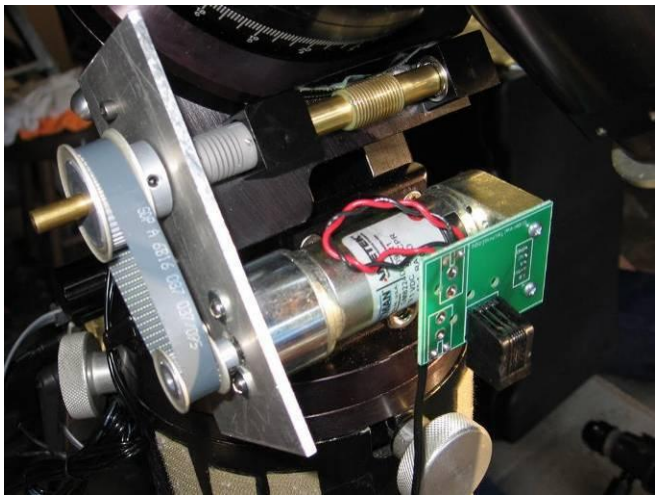
There is an ascom FindHome function. In order for the ASCOM function to work, you have to turn on the FindHome in the config/ASCOMAndTroubleshooting.

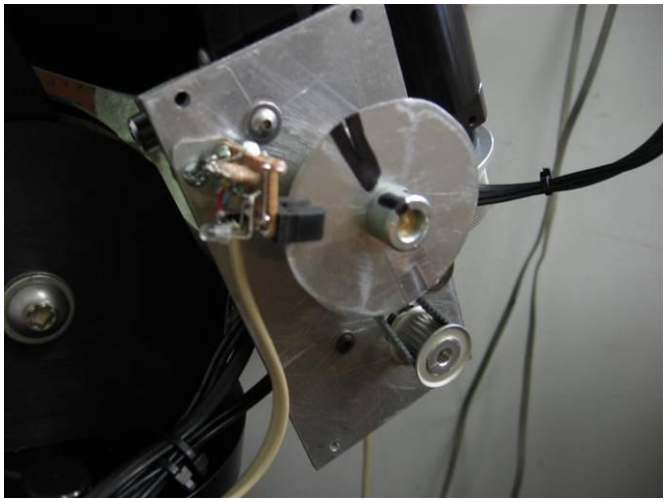
APPENDIX P: SAMPLE ENCODER INSTALLATIONS

Gurley Encoder Installation on G-11 SiTech Conversion

Here is Chuck Shaw's installation of a Gurley Encoder on his G-111 Mount. The Gurley needs a firm, accurate, mounting to be able to measure all of the RA rotation, yet the Gurley is sensitive to any axial or side load on the encoder shaft. He used aluminum angles with thumb screws to allow lateral and axial alignment to connect the encoder shaft to the mount RA axis to prevent any side-loads on the encoder shaft. The brackets keep the encoder body from rotating so the encoder measures all of the RA shaft rotation. The brass knurled thumbscrew clamps the Gurley's shaft in an aluminum plug. The aluminum plug has the same dimensions as the G-ll polar alignment scope and is captured in the hollow RA axis with the same threaded collar used for the polar align scope.



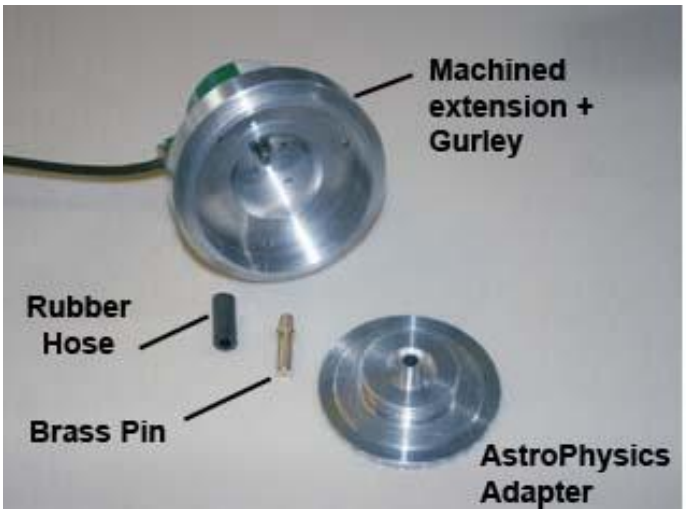




The above left picture shows the Dec conversion. The metal plate attaches to the same place the old motor attached, and also provides a mounting for the motor. The picture on the right shows the RA conversion. A slotted disk with an Autopecsync sensor is attached to the pulley that is coupled to the worm. Cover plates (shown in the picture below) protect the belts.

Gurley Encoder Installation on AP1200 SiTech Conversion

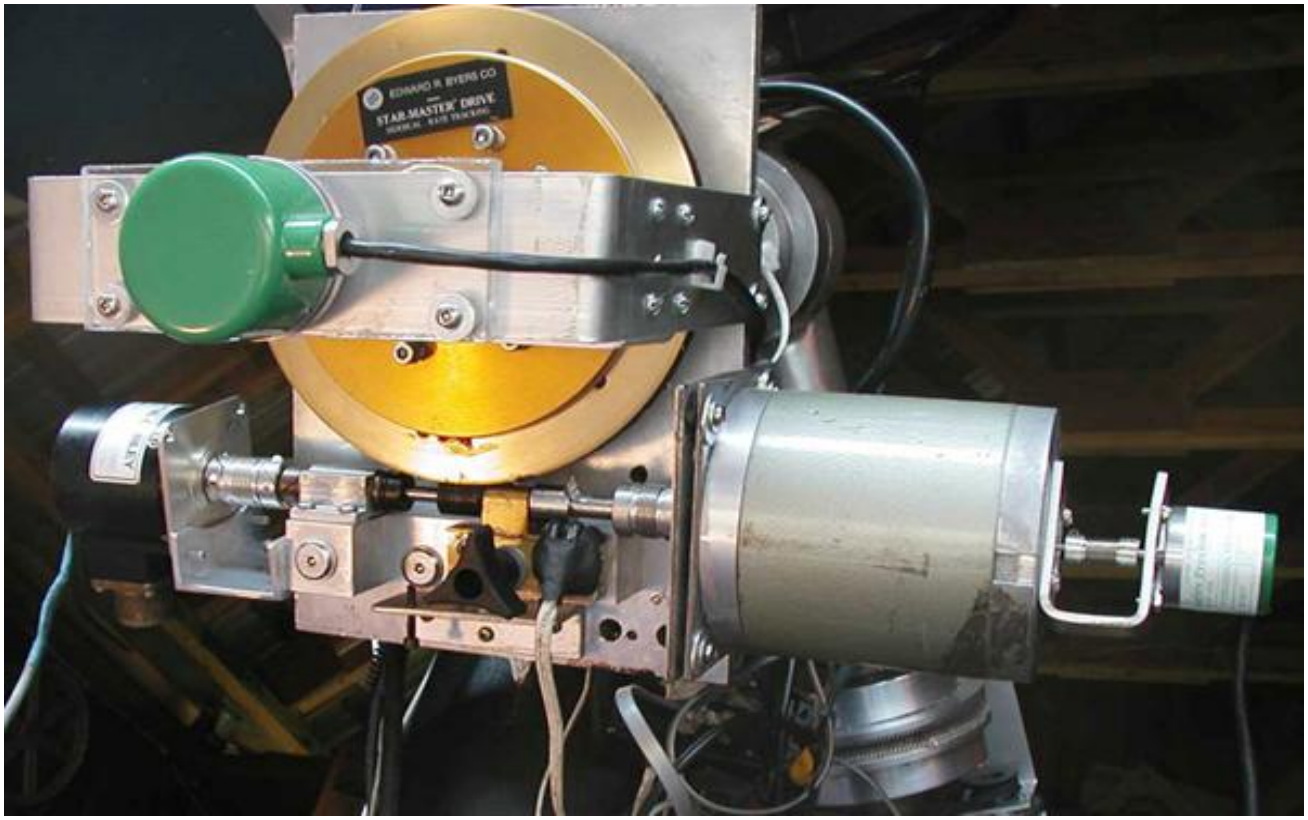
Here is Don Westergren's installation on an AstroPhysics 1200 mount. It uses a machined extension that screws into the mount housing, holding the Gurley about an inch from the mount. A ¼" diameter brass pin fits snug into the AP adapter (aluminum disc with thin wall ¼" ID section. A 3/16" ID rubber hose is the coupling.



The SiTech servos replaced the original stepper motors with direct drive of the worm gears. A1/4" aluminum plate was machined to mount each servo motor so the gearbox shaft mated to the AP shaft coupling. These servo motors are Pittman 9000 series motors with 65:1 gearbox. Slews are ~ 300X.

Gurley Encoders Plus Flex Couplings

Here is a view of the installation by Juan Herrero. He uses a large diameter direct drive motor with encoders on the opposite end of the motor and the worm shaft. Actually the two encoders on the worm shaft are because Juan is testing the output of a second encoder (the black one to the left). Only one encoder is needed on the shaft for normal operations. Both the motor encoders and the motor use high quality shaft couplings to transmit torque without alignment problems.



For the mount encoder, Juan has a high resolution Gurley on the RA shaft mounted with another flex coupling (out of sight under the encoder bracket).

"Geared up" Encoder Mounting

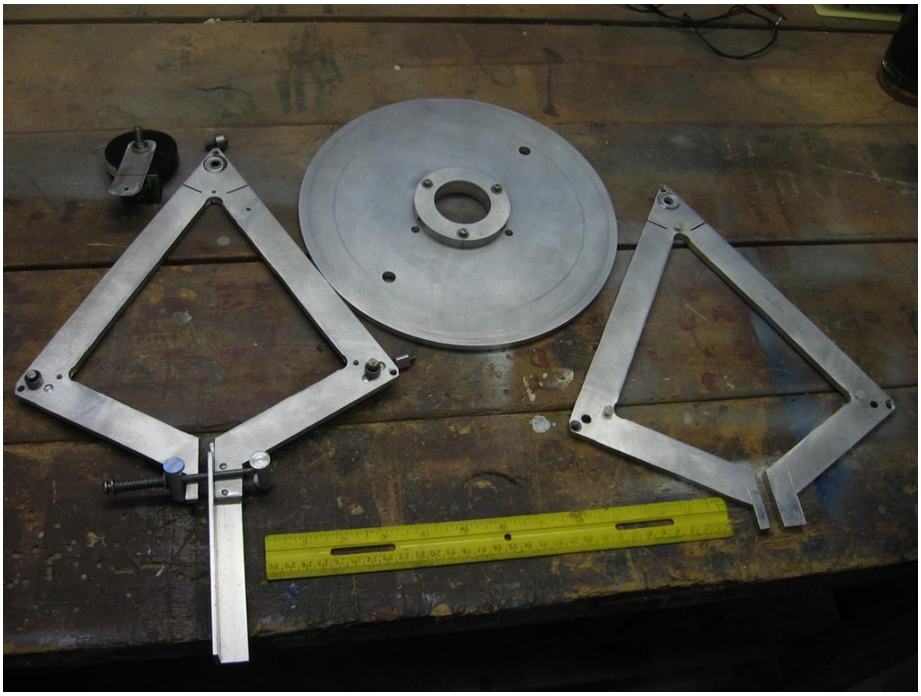
Instead of using a high resolution encoder like a 320,000 or 500,000 tick Gurley, it is also possible to use a lower resolution encoder like the more common (and inexpensive) 8192 tick encoders that are available, and gear them "up" to spin much faster than the telescope axis they are measuring. The problem is the increased resolution that results, also is subject to mechanical imperfections in the mounting and gearing up hardware that to the encoder will look like errors in the telescope axis's motion (since there is no way to tell if the motion errors come from the telescope axis or the gears). Spur gears

are an easy straightforward way to gear up a mount encoder, and the typical errors in a "good" set of spur gears are usually far less than the encoder resolution when the spur gears are used 1:1 (an 8192 tick encoder, geared 1:1 provides 158.2 arcseconds, or 2.64 arcminutes of resolution). However, to get into the 320,000 tick range, an 8192 tick encoder needs to be geared up by approximately 40:1 This will require a multiple stage spur gear set, or a very large spur gear driving a very small one. In either case, the expense is increased, and so are the mechanical errors.

Alternatively, a disk drive can be used. The picture below shows a prototype disk drive unit fabricated by Andy Saulietis that is fitted to a Starliner GEM that was converted to Sitech. Lenord Stage has also mounted encoders on telescopes using a round surface on a large diameter RA worm gear. The Starliner's HDPE gear (the white disk in front of the aluminum disk) did not have such an available surface, so a separate disk was used.

The 10 inch diameter (1/4 inch thick) disk is attached to the RA axis. The trapezoidal frame (one on each side of the disk) has 3 rollers (two at the bottom, and the encoder shaft is the 3rd one at the top). The frame is split at the bottom so it can be clamped together to have all 3 roller's contact pressure increased to the point there is no slippage. The three rollers all run in small needle roller bearings to minimize friction. Since the disk is 10 inches in diameter, and the encoder shaft is ¼ inch in diameter, the result is the 8192 encoder is geared "up" by 40:1, giving 327,680 ticks per rev (8192*40). The "tang" at the bottom prevents the frame from rotating

The components of this prototype encoder installation are shown below, with a 12 inch ruler in the picture for scale. The performance of this prototype was measured against the performance of a 320,000 tick gurley that was attached to the RA axis directly using a test routine written by Dan Gray called "ChucksLogger.exe" which logs encoder ticks from each encoder.



The test results confirmed the approach can work very well, but also confirmed the approach is VERY sensitive to dirt and grit on the disk surface (implying the assembly needs to be enclosed or at a minimum have a “sweeper” brush on either side of the encoder roller), and any old setscrew marks on the encoder shaft surface cause errors (many used encoders have these marks). In addition, and any non-squareness of the encoder shaft roller to the surface of the disk will introduce errors (so the two frames were match drilled on a drill press). A slight chamfer on the edges of the disk does markedly help with this also (avoiding ill effects from tiny nicks or dents in the edges of the disk). Performance is also very sensitive to how much clamping pressure is applied. Too little pressure and the encoder shaft skids easily, and too much pressure increases the rolling friction (even with the bearings) and also causes skidding. Very fine grit lapping of the encoder shaft to the disk should help with all of these problems also.

In addition, if the encoder shaft is not running PERFECTLY concentrically (i.e. it is not PERFECTLY round), then periodic error will be introduced. Runout errors (measured as Max minus Min) as small as 0.0005 (one half of one thousandths of an inch) will introduce periodic errors as large as 20 arcseconds peak to peak, at a period equal to low long it takes the encoder roller to make one rotation (for the prototype it was 138 seconds, or 2 minutes and 18 seconds). This runout error also applies to encoder mounted axially on the end of the RA axis, but the period is spread out over the entire rotation of the telescope axis, however, it can make itself felt when you do long slews, and this same long period (24 hour) PE can be introduced if the disk is not mounted perfectly centered on the RA shaft (this is in addition to the eccentricity PE error from the encoder shaft.

The lesson from this is to take the time to do some careful machining, and it will pay off in accuracy, however if you have to pay a lot for the machine shop work (which needs to be reasonably high quality), it may be more cost (and hassle) effective to simply use an axially mounted high resolution encoder.

Appendix Q: TROUBLESHOOTING

Motor Operation

Problem: When power is applied to the SiTech controller and a handpad direction button is pressed, the motor slowly increases, faster and faster until it's full speed.
Solution: The motor encoder is not supplying feedback. Check cabling, connectors, and encoder you can swap the altitude and azimuth motors for testing. See if the problem follows the motor, or if the problem remains. If the problem remains, there may be an issue with the controller. If it follows the motor, it is definitely a wiring or encoder problem.

Problem: Motor runs away as soon as power is applied or commanded to move. After running fast for a short time, it stops, and the light flashes.
Solution: The motor encoder relationship to the motor polarity is wrong. See section on "Setting Up the Motors".

Problem: When commanded to move, the motor starts ramping up, then stops, and the LED flashes.
Solution: It could be that you are using higher voltage motors with only 12 volts applied to the controller. Either slow down the speed, or use a higher voltage on the power to the controller.

Problem: Nothing works at all, and the two LED's are on very faintly.
Solution: There is a major problem with the CPU. If you were using version 1.3 or earlier, there was a bug that allowed a remote chance of the flash memory to be erased. You will have to send the controller back to the factory for a re-program.

Problem: Motor gets abnormally hot:
Solution: Check the motor current. You can do this using the ServoConfig software. The motor current appears on the main screen. If more than 0.5 amp while the scope is moving slowly, you may have more friction on the gear train, or maybe your scope is out of balance. Sometimes worm gears add a lot of friction to a drive system. Be sure the thrust clearance on the worm is not too tight, and be sure the worm pressure against the worm gear is not too much. Be sure lubricant is applied properly.

Problem: Upon applying power, one of the motors starts slewing, all by itself. Nothing will stop it.
Solution: Unplug the handpad and see if it still happens. If it stops moving, you've a faulty handpad. If it keeps on moving, it may be there is a problem with the digital input for the handpad. Please contact us.

Handpad Operation Troubleshooting

Problem: I'm using an old style handpad (the one with a center switch) and the scope only moves in two directions instead of 4.
Solution: Check that the internal jumper is set properly for the old handpad.

Problem: Some of the switches don't work properly.
Solution: Use the ServoConfig software. At the main screen, look at the virtual handpad. Pressing each handpad button should make the appropriate virtual button highlight. If not, check the handpad for problems. If the virtual handpad works properly, check the slew and pan rates.

Mount Encoders Troubleshooting

Problem: My mount encoders don't indicate any motion.
Solution: Several things can cause the mount encoders to not work.

1. Check the cables to the encoders to make sure they are plugged into the Controller and the encoders. Look inside the connector to see if the wire contacts are bent out of position. Make sure the cable is in good condition – not damaged or cut. You can try another cable if you have one available.
2. Make sure you have set up the ticks per rev for the encoders - that they are not set at zero.
3. If using SiTechExe, make sure the configuration is set not set to "Ignore Encoders"

Problem: My Gurley mount encoders don't follow the motor encoders and cause the controller to stop with an encoder error.
Solution: Most likely there is slippage either in the motor drive or in the encoder connection to the mount. If you have any friction drive elements between the servo and mount or between the mount and encoders, try to eliminate slippage there. It could also be that a coupling has come loose – especially if the coupling uses a set screw to lock onto a shaft.

The Gurley encoders do not like any significant side load on the shaft. When a side load is too much, the output signal can "jump" some degrees in reading. When mounting the Gurley, make sure it has a flex coupling to allow shaft misalignment OR a flex mounting of the encoder body to allow for shaft movement. The rotation of the Gurley shaft is very free, so even an elastomeric (rubber) coupling should work.