CAUTION: This device contains voltages that could damage, shock or electrocute personnel. Please disconnect ALL power before working on device.

CAUTION: This device can create up to one horsepower of work, and could damage camera's, other hardware, or even personnel. Please take necessary precautions to safeguard the area.

**Basic Hardware Description:**

The CPU board is the board on the top of the stack, with the 4 encoder terminal blocks. The two power boards, are the boards below the CPU board, with the larger, 10 position terminal blocks.
In the following drawing of the CPU board, many of the items you can connect are optional. What are not optional are the motor encoders (they could be BiSS Absolute Renishaw Encoders), the USB port and the power.

All other inputs are optional inputs, such as the handpad inputs, scope encoder inputs, Spare Digital Input, Homing and Limit switches, along with the autoguider port, and the 2nd serial port.

Connect up your CPU board like this drawing:
Connect up both of the power boards like this:
Power Connections:

CPU Board:
The power for the CPU board is fairly low current, less than 1 Amp. This power must be 12 volts, +/- 2 volts, because the fan in the enclosure is 12 volts.

Motor Power Board:

Gate Power:
Terminal 1, and terminal 3 are both connected to the ground connection of the GATE power. Terminal 2 is the Gate power, which can be anywhere from 9 volts to 48 volts D.C. The Gate Power is fully electrically isolated from both the CPU board, AND the power to the motors.
On the power board, the gate power is converted to 5 volts with a small switching regulator, then there are 4 isolated 15 volt supplies that are powered by this 5 volts. One of the 15 volt supplies provides gate power to the 3 FETS that are grounded (low side FET's), and the other three 15 volt supplies provide gate power to the three FET's that are connected to the positive motor voltage. This supply also provides the power to the Brake output.

Brake:
Next is the Brake connection if you're using a Brake. Connect the negative of the brake to terminal 4, and the positive of the brake will be connected to terminal 5. When the brake is energized, terminal 4 is shorted to ground by an FET transistor.

The brake should be released when this terminal is energized. There is a free wheeling diode on the circuit board, but it may not hurt to add a free wheeling diode right at the brake coil. Connect the Anode to the brake negative terminal, and the cathode to the positive terminal of the brake coil.

Terminal 5 is rectified and filtered Gate Voltage Output. This is the power that is supplied to the brake coil.

Motor Power:
Motor power is supplied to the power board using terminals 6 (Negative) and terminal 10 (Positive). This power is not common to anything, it is fully isolated from both the CPU power AND the gate power.

The motor power can be anywhere from 10 volts to a maximum of 150 volts D.C.

It's also ok to power the gate voltage from the CPU power supply. If you do this, add another 500mA to your CPU power supply, so a 1.5 amp 12 volt supply would be needed.

You should have a separate power supply for your motor(s). This can be simply a transformer and a full wave bridge rectifier, with a large filter capacitor, or it could be a regulated supply. It's important that this supply be isolated from ground.

Motor Terminals:
Brushless D.C. Servo motors are generally 3 phase permanent magnet field motors, so there is no field connection. The Stator (or armature in this case) are connected to terminals labeled Motor A (terminal
7), Motor B (terminal 8), and Motor C (terminal 9). The Brushless Controller is capable of operating a conventional BRUSHed type motor too. In this case, set the Motor Ticks Per Electrical Pole to ZERO, and then connect the brushed motor to the A and the C terminals (leave the B terminal empty).

**Configuring the Controller Using BrushlessServoConfig:**

**DO NOT POWER UP YOUR POWER BOARDS WITH MOTOR POWER YET!**

Power up the CPU board only. Turn the power switch on until some LED's light.

Connect the USB to your P.C. Make sure you have a new Comm port. If not, install this driver: [http://www.ftdichip.com/Drivers/CDM/CDM20828_Setup.exe](http://www.ftdichip.com/Drivers/CDM/CDM20828_Setup.exe)

Now run BrushlessServoConfig. If you don't have the latest version, email Dan Gray [grayarea@siderealtechnology.com](mailto:grayarea@siderealtechnology.com)

The current version as of this writing (May of 2016) is version 1.5b.

[http://siderealtechnology.com/BrushlessServoConfigProject1.5b.zip](http://siderealtechnology.com/BrushlessServoConfigProject1.5b.zip)

After unzipping somewhere, and running BrushlessServoConfig.exe (make sure it says V1.5b or later), you will see the main screen or “information” screen. It looks like this:
First thing to do when running BrushlessServoConfig is “Choose Comm Port” on upper left of the Information display tab. Only valid ports appear here. Select the new comm port that was created when you plugged in the USB cable to the controller. You will then be connected to your Brushless Controller.

Pay special attention to the checkboxes on the right. These are NOT options, they're indicators. In the above screen shot, you can see that there is no gate voltage on the Azimuth/RA, and there is no motor voltage on the Altitude/Dec.

Otherwise, this screen is mostly information. If you know the ascii command set, you can enter the commands in the text box labeled “enter Your serial command below”.

You can see the status of all of the limit switches, the HOME switches, and the PEC Sync switches. You can also see the guider inputs, and the handpad inputs. These turn green when the appropriate input is activated. There are several other indicators, such as encoder position, etc.
Near the left side of the Information screen there are several items displayed.

**Motor Location:** This is the motor location in encoder ticks.

**Scope Encoder:** This is the scope encoder location in encoder ticks.

**Motor Electrical Angle:** This is the motor electrical angle. If you have a 4 pole conventional brushless motor, with a 10,000 tick encoder, then there will be 2 complete cycles of motor electrical angle as you turn the motor 1 revolution.

**PWM to motor:** This is the PWM in percentage to each winding.

**Motor Current:** This is the motor current in amperage.

**Power Board Volts:** This is the motor power voltage applied to the power board.

**Power Board Firmware Version:** The current version of the power boards is 3.5. If less than this, please upgrade your firmware to the latest.

**Analog Input:** This is the analog input value. This is currently not used by any features of SiTechExe or BrushlessServoConfig, but has been added for future possibilities.

Near the lower right of the screen is the CPU board supply volts and the CPU board version number. The current version of the CPU board is 14.9. If your version is less than 14.9, please upgrade your firmware to the latest.

There are three buttons for each axis near the bottom right of the screen. These buttons allow you to override the brake output. Normally, the brake output is fully automatic, it will energize the brake at all times when the servo motor is in automatic, and it will energize the brake when it the manual or winding control mode when the power output reaches the brake release threshold. You can force the brake on or off. Please note that if you force the brake ON, then it will de-energize the brake, which will activate the brake.

There are two lines at the bottom of the screen (which is duplicated on all other tabs), which has information about the two motors. The information is as follows:

**MotLoc** is the motor encoder location in encoder ticks.

**ScpLoc** is the scope encoder location in encoder ticks.

**PosErr** is the position error of the servo loop. If the motor is not in automatic, this will be zero. When the motors are in automatic, this is the difference in true motor location and the desired motor location. It is displayed as the number of encoder ticks. This number can be useful for tuning the motors while tracking.

**MotAng** is the motor electrical angle. If you have a 4 pole conventional brushless motor, with a 10,000 tick encoder, then there will be 2 complete cycles of motor electrical angle as you turn the motor 1 revolution.

**MotCur** is the motor current in amps.

**A=nn.n** is the PWM output of the PID if in automatic or manual. When in winding control mode, it is the PWM to the A phase.

**B=nn.0 C=nn.0** is the PWM output to 3 phases if in winding control mode.

**Configuration Tab:**
Now you need to configure your controller for the motors that you have.

Click on the Configuration tab. The VERY first thing you need to do is click on “Read From Controller”. If you don't you may not be editing the actual data from the controller. Note: When you
start up BrushlessServoConfig.exe, and the serial port is configured already from a previous session, the software will automatically read in the parameters. Whenever you change serial ports you MUST “Read From Controller” again.

Here's what the Configuration screen looks like:

![Configuration Screen](image)

The important parameters are all highlighted in orange. These parameters MUST be entered correctly.

**Parameters:**
First of all, and I repeat, click on “Read From Controller”, so you know the adjustments are the ones that came from the controller. When you make a parameter change, they will only take effect when you click on “Write to Controller”.

You can do most of the initial configuration, then click on “Write to Controller”. You can also “save to file”, and “read from file”. When you “read from file”, you have to “Write To Controller” for the
settings to be saved in the controller.

**Motor Ticks per Revolution:**
You must do the math on your own, but if you're using a direct drive encoder, with a ring from Renishaw, use the ticks per revolution of the ring. If you're using a Renishaw tape and Renishaw read head, then you'll measure the diameter, figure out using the encoder read head you're using, and interpolate that for a full number of ticks per revolution. If you're using a conventional brushless motor, then you'll have to know the motor encoder ticks (in quadrature) times the gear reduction of your mount.

This parameter does not need to be 100% accurate at the beginning, you can fine tune this parameter later.

Also, you can use an absolute 26 bit Renishaw encoder with their BiSS interface. If that's what you're doing, you must check the checkbox adjacent to the number, and enter $2^{26}$, or 67108864 for your ticks per revolution.

**Scope Ticks per Revolution:**
This parameter only needs to be completed if you're going to be using scope encoders. This will be how many encoder ticks (after quadrature) for a full revolution of the scope.

Also, you can use an absolute 26 bit Renishaw encoder with their BiSS interface for the scope encoder(s). If that's what you're doing, you must check the checkbox adjacent to the number. In this case, you'll have conventional incremental encoders on your motor, and absolute high resolution BiSS encoders on your telescope axis. Again, enter $2^{26}$ or 67108864 for your ticks per revolution.

When using high resolution encoders, absolute, or incremental, in SiTechExe, you'll select the “Cascade” mode for the encoders.

**Motor Ticks per Electrical Pole (VERY IMPORTANT):**
This is probably the most important parameter. If this isn't right, your brushless motors simply won't work. They'll just go into current limit at some point, with winding current holding the motor near a certain point.

If you have a direct drive motor, typically, it's the (Motor Ticks Per Revolution / Number of poles) * 2.

If you're using a conventional brushless motor, then it's the number of (motor encoder ticks / motor poles) * 2.

If you have a 4 pole motor conventional brushless motor with a 10,000 tick encoder, then you would enter 5,000.

If you have a 6 pole motor, then your motor ticks per electrical pole will probably be a fraction. Every 3 electrical cycles, the number in this box is added to the motor electrical pole value. So, for instance, if you have a 6 pole motor with a 4000 tick encoder, you would enter the number “1” in this box, and the number 1333 in the Motor Ticks per Electrical Pole (1333 * 3 + 1 = 4000). Another example is a 6 pole motor with an 8000 tick encoder. You would enter a “2” in this box, and the number 2666 in the
Motor Ticks Per Electrical Pole (2666 * 3 + 2 = 8000).

You can count the poles of your motor if this is unknown. Simply put a battery between phase A and B. Be sure your battery does not exceed the rated current of the motor by measuring the resistance of the windings. If necessary, use a lower voltage, or put a high wattage low ohm's resistor in series with the battery. You can also do this with BrushlessServoConfig using the “Control” tab (documented later in this document). After applying voltage, move the motor by hand one full revolution, counting the magnetic “Humps”. The number of poles is the number of magnetic humps * 2. For instant if you have a conventional brushless motor, and you have two magnetic humps, it's a 4 pole motor.

The Brushless Controller is capable of operating a conventional BRUSHed type motor too. In this case, set the Motor Ticks Per Electrical Pole to ZERO, and then connect the brushed motor to the A and the C terminals of the power board.

**Slew Pan and Guide Speeds:**
These WILL change if you change the Servo PID Sample Rate!!! The raw values will remain the same, but the DPS, etc, will change.

**Slew Speed:**
You can enter this in DPS if you've already filled in the motor ticks per revolution. There is no limit, only the limit that the motors are able to do, as well as the encoder read speed, which would be about 120 deg's per second if you have 18,000,000 ticks per revolution (this probably will never be a problem for you!)

**Pan Speed.**
This is an intermediate speed that the motors will move when in the “Pan” mode. It's in Arc Minutes per second, and raw speed.

**Tweak Speed:**
This is the slowest mode that the handpad will move the motors. It's in Arc Minutes per second, and raw speed.

**Guide Speed:**
This is the speed the motors will add (or subtract) to (from) the tracking rate when receiving a guide command. It's rated in Arc Seconds Per Second.

**Acceleration:**
This is how fast you want the motors to accelerate and de-accelerate. It's rated in DPS2

**PID Parameters:**
The next few entries are having to do with the P.I.D. (proportional integral derivative) parameters. These entries are extremely important and probably the hardest parameters to adjust. The P, I, and D parameters have two parameters for each parameter, one for tracking speed, and one for slewing speed. This complicates the procedure, however, has a tremendous advantage of having two parameters, one for tracking and one for slewing. You can make the tracking parameters more aggressive for finer tracking control, and when moving faster, you will make less aggressive adjustments, so there will be no oscillations while slewing, sort of the best of both worlds.
For a conventional brushless motor, the parameters are easier to adjust. Start with about 3000 for both slewing and tracking proportional, a tracking integral of 100 and a slewing integral of about 50, a tracking and slewing derivative of about 6000.

For a direct drive motor with a 26 bit absolute encoder, try these:
- Tracking proportional: 800
- Slewing proportional: 400
- Tracking Integral: 10
- Slewing Integral: 50
- Tracking Derivative: 4000
- Slewing Derivative: 8000

Note: If you change the Servo PID Sample rate, the Integral and Derivative will change in effect. The Proportional effect will remain the same.

**Integral Limit:**
Set your Integral limit to be about 22000 (the default should be fine)

**Error Limit:**
The error limit is the deviation of the actual motor position from the desired motor position. For instant if you have a conventional brushless motor with a 10,000 tick encoder on it, if you set the error limit to be 10,000, then when the difference between motor position and motor desired position is greater than 10,000 encoder ticks, the motor will turn to manual mode with 0 volts applied to the motor, in essence, it will stop.

On a direct drive motor, this should be about ½ of an electrical cycle. So if your motor ticks per electrical pole is 1,000,000, try 500,000.

**Output Limit.**
In this version of BrushlessServoConfig (version 1.3A), this is 0-32000. In a later version, it will be 0-100%. If you don't want 100% of the motor supply voltage applied to the motors, then set this to a lower value. Just don't forget that at high RPM's, there is more electro motive force, so the motor will need more voltage than at lower RPM's.

**Zero Crossing Filter:**
In rare cases, this can help stabilize the PID when there is zero torque on the motor. Use sparingly, if at all. Normally, this will be 0.

**Current Limit:**
Set this to the current rating of your motor, but no more than 8 amps. Your motor may “squeal” a bit when it's in current limit. This current is folded back over time to your setting / 4 * 3. For instance if you set your current limit to 6 amps, it will start current limit at 6 amps, but over time, will fold back to 4.5 amps. This means you can put a higher current limit than your motor continuous current rating. Make sure it will fold back to the continuous current rating.

**Startup PWM High:**
When power is applied to the motor power, and you have gate power, then the system will automatically initialize. It will start applying voltage at the PWM Low setting, to the B winding positive, A and C negative, then ramp up to the PWM High setting. Once the PWM is at the high setting, the firmware will change the applied angle clockwise then counterclockwise, and then clockwise again, then stops the angle, and now it remains a time for the motor to “find” its magnetic center, then it will go to the “auto” mode.

To set these values, do a bit of math. Find the upper current limit for your motor, and the motor winding resistance, and the applied motor voltage, then put this number to make the motor current be

As an example, say you have a 10 ohm resistance measured from coil A to coil B, and you have 48 volts on the motor supply. Your maximum current is 5 amps. You have a “Y” wound 3 phase motor. Your individual winding value will be 5 ohms. Since the two bottom windings will effectively be in parallel, their value will be 2.5 ohms. The total resistance will be 7.5 ohms.

In this case, set your Startup PWM High value to 80%, which will be 5 amps.

Be very careful that this number is not higher than the controller will put out. This can damage your controller or motor. When the current reaches your current limit, it will be folded back for safety, so in effect, your current limit setting will limit this.

It's also important that full motor current be reached, so the motor will “find” its magnetic position accurately.

One more thing, if you have absolute encoders for your motor, then it will only do the above procedure one time. This angle is saved in the flash rom, and next time you power up, it will not need to do the initialization procedure. If you want to renew the motor electrical angle, you can click on the button labeled “Init Motor Angle” on the Information screen.

**Startup PWM Low:**
If you have a direct drive system, it's possible you also have a brake. During initialization, you would not want your mount to fall. Thus, there is a low PWM setting. Set this to a value which will keep your mount from “falling” while doing an initialization. This can be nearly zero for a conventional brushless motor, or a motor that doesn't have a brake.

**Brake Release Threshold:**
The brake output will be de energized (brake activated) during manual or the initialization process, until the output reaches this threshold, then the brake output will be energized (released).

**Motor Position at 270:**
If you have absolute motor encoders, these are the saved offsets for the motor electrical angle initialization. These are for information only.

**Acceleration Feed Forward:**
If you want to improve the acceleration and de-acceleration, it's possible to improve the response by adding a bit of feed Forward here. 3200 would be 10% output change with a certain change in acceleration, before the motor had position error. Thus the controller can anticipate the position error created by acceleration or de-acceleration. If you want to use this, try small amounts first, 200-400, and increase incrementally as it gets better, then when it starts to get worse, back it off 10 or 20%.
Encoder Directions (VERY IMPORTANT):
In the process of setting up your motor, you may have to reverse the motor phase, and/or the motor encoder directions for proper operation.

First of all, make sure the motor encoder directions are correct. You can do this before powering up the power boards.

When the telescope is moved to the west (equatorial) or clockwise (birds eye view if alt/az), the right ascension (or azimuth) motor encoder should INCREASE.

If you are using a GEM, When the telescope is “looking west”, and the telescope is moving to the north, the Declination motor encoder should Increase. If you're using an equatorial, if your telescope is moving to the north, the motor encoder should increase. If you're using an alt/az, when your telescope is moving UP, the motor encoder should increase.

If you're using scope encoders, make sure they are working the same direction as the motor encoders.

You can check this by moving the motors by hand, and watching the motor positions on BrushlessServoConfig.

Reverse Motor Phase (VERY IMPORTANT):
Once you have your encoder directions working properly, you must set the motor direction properly. Figuring out if you need to reverse the motor phase is a little bit more complicated. The simplest way is to just power the controller and see if it works. If the motors go to a certain position, the current reaches the current limit, and there is no position control, then try checking the “reverse motor phase” button, then click “Write To Controller”, then reset the controller.

The more reliable way, and a bit more complicated is power up the controller, after it goes through the initialization procedure, then click on the CONTROL tab, click on To Winding Control button.

Apply a positive output (to the right of center) on the Winding control output level (the lower slider). You will see the motor move slightly. Once there is enough torque, click the “Phase A Highest” followed by “Phase B Highest” followed by “Phase C Highest”, then Phase A Highest again, in a repeating pattern. The motor encoder should “STEP” in the negative direction, IE the motor encoder should go to a more negative number.

If not, then click the “Reverse Motor Phase” button, then click “Write TO Controller”, then reset the controller, and try again.

Servo Loop Sample Rate:
The Servo Loop frequency is the frequency that the motor encoder is read, the PID is calculated, and the output is applied to the windings. This frequency can be adjusted from 1953 hertz to 4000 hertz. This allows you to avoid a resonant frequency, or calculate the motor electrical angle faster, etc.

You can enter the frequency in the textbox named “PID Sample Rate” to try various frequencies. Higher frequencies will allow faster RPM on conventional brushless motors, as the motor electrical
angle is calculated faster. Also, tuning direct drive motors can be easier at higher sample rates. The text box will allow a faster rate than is possible, so please limit it to about 4000 hertz. The minimum value is 1953 Hertz.

Configuration Axis Swap, etc:
There's a button that allows you to swap the alt/dec and az/ra configurations. This might be nice for setting up two identical motors.

You can swap the two axis's, move from Dec/Alt to RA/Az, or vice versa.

Testing and Tuning:
Once you have all of your parameters set, you're ready to power up the power boards. Apply power, and you should see on the “information” screen the motor supply voltage, the fact that you have gate voltage, and your controller will put the motors through the initialization process. After a few seconds, the ToAuto buttons should turn green, and your motors should be working. If you've already set up the motor encoder directions, as described earlier, then they should be fine. If the motors “find” a magnetic hump, and the motor current goes to the current limit, then go to the configuration screen, “read from controller”, then check the “reverse motor phase” checkbox, then “Write To Controller.” Now reset the controller (there's a button on the information screen to reset the controller).

If the motors oscillate, this means they're working ok, but they need to be tuned. click on the Test and Tune panel. BE SURE THAT YOU'RE TUNING THE PROPER MOTOR!!!! There's two radio buttons at the top of the screen. If you're tuning the Right Ascension/Azimuth, BE SURE YOU'VE SELECTED THE “Test RA/Azimuth” radio button.
On the right hand side of this screen are the SLEWING PID parameters, and the left side of the screen are the TRACKING PID parameters.

If oscillating, the first thing to do is take both the tracking and slewing integral to zero. Now increase derivative and decrease proportional until it stops oscillating. Now increase the integral slightly until there is nearly zero error. You can move the fine and course sliders at the bottom of the screen to move the motors.

If you have about 13,000,000 ticks per rev (about 10 ticks per arc second), the raw sidereal tracking rate is about 5000. Use the bottom slider (the FINE Speed Adjustment), and move it to about 5000. This will move the axis at about the sidereal rate. You can now pay special attention to the PosError, and adjust the tracking PID parameters to get this as low as possible. If tuned properly, the peak PosError should be less than ±3, with an RMS of about ±1.

If the motors go to manual for any reason, you can click on the “ToAuto” button to try again. Once you have the settings as you like them, you can click on Save Settings to Flash” to save them to
flash rom. If you think you made it worse, you can click on “Read In Flash Settings” and the values will return to the last time you saved them to flash.

Normally the slewing integral will be significantly lower than the tracking integral (about 1/4), and the slewing proportional will be about ½ the tracking proportional, and the slewing derivative will by about 2 times the tracking derivative.

If you want to change the frequency of the Servo PID Sample rate. You can use the slider labeled “PID Sample Rate” to try various frequencies. Normally, you will keep as high as possible without exceeding 4000.

Higher frequencies will allow faster RPM on conventional brushless motors, as the motor electrical angle is calculated faster. Also, tuning direct drive motors can be easier at higher sample rates.

Also, if using a direct drive motor, there may be a mechanical resonant frequency in your mount, and changing this frequency can be a help.

The slider will allow a faster rate than is possible, so please limit it to about 4000 hertz.

**Upgrading Flash ROM Firmware:**
IMPORTANT!! You MUST use BrushlessServoConfig version 1.5b or later to upgrade your flash rom. Failure to do so can ruin the flash rom, and you would have to send the boards back to the factory for a reflash!

IMPORTANT: You MUST Upgrade the CPU first, and do NOT “downgrade” to an earlier CPU version.

IMPORTANT: Please remove motor power before upgrading the flash in the CPU board OR the power boards.

There is firmware in both of the motor power boards, and the CPU board, and the BiSS absolute encoder interface CPU. These are upgradable by selecting the “Upgrade Flash Firmware” tab. There are simply 4 buttons on this tab. If available, always upgrade the CPU first.

**Upgrading the CPU:**
The latest CPU firmware file is here:


Select the Upgrade Flash Firmware tab.

After selecting the button “Upgrade Flash in CPU”, browse to the latest binary file and select it. You will see LED's flash as the CPU is upgraded. You will see the block number advance in BrushlessServoConfig. If there is a communication error, it will retry 3 times for each block. Once the flash is upgraded, the controller will reboot. During the reboot process, select the “Information” tab, and you should see the new CPU version number near the lower right hand side of the window.
Upgrading the Power Boards:
The firmware for the two power boards is identical.
The latest Power board firmware file is here:

http://siderealtechnology.com/BrushlessPWR3.5.BIN

After selecting the button “Upgrade Flash in RtAsc/Azimuth Power board”, browse to the latest binary file and select it. You will see the LED flash adjacent to the ribbon cable of the power board you are upgrading. You will see LED's flash on the power board as it is upgraded. You will see the block number advance in BrushlessServoConfig. If there is a communication error, it will retry 3 times for each block. Once the flash is upgraded, the controller will reboot. During the reboot process, select the “Information” tab, and you should see the new Power board firmware version number near the middle left hand side of the window.

Upgrading the Dec/Altitude power board is the same as upgrading the RtAsc/Azimuth Power board.

Upgrading the Flash in the BiSS CPU is similar to the power boards or CPU.
Here is the latest binary file for the BiSS CPU.

http://siderealtechnology.com/BiSS1.1.BIN

Once everything is upgraded, you can power up the motors again.

The meaning of the LED's:

CPU Board LED's:
There are two LED's on the CPU board, adjacent to each ribbon cable. These are the status LED's for the particular motor. If the LED is flashing, the motor is NOT in automatic. It is either in manual torque mode, or it's in the winding control mode.

If the LED is flickering, then your motor is coasting. This is not a regenerative servo motor controller, so it counts on the fact that the motors will slow down on their own. In this case, the LED will flicker, to let you know that it is happening. If this is happening, try slowing down your acceleration, so the motor will always have positive torque, even when slowing down.

There are 3 LED's adjacent to the handpad input jack. Red is the Slew LED. If this is on, the controller is in the slew mode. Next is a yellow LED, if it's on, it's in the PAN mode. Next is a green LED. If it's on, it's in the “tweak” mode.

There is a red LED adjacent to the USB connector. It will flash when there is communication.

Power Board LED's:
There are 4 LED's on the power board:
The first one is the +15 volt LED. It is powered by the Gate Voltage, after conversion to +5 volts, then converted to +15 volts. If this LED is on, you have Gate Voltage.

Next is the BRAKE LED. If this LED is on, the brake output has been activated (brake is released).
Next is the Overcurrent LED. If this led comes on, there has been a hardware overcurrent, and the FET's have been disabled. It times out after about 1 second. But this puts the motor into the “Manual” mode to protect the system. You can put back to auto by clicking on the “ToAuto” button.

Next is the FAULT LED. Normally this LED is on steady. If there is no GATE voltage, it flashes fast. If there is no motor power, it flashes slow. If this LED is off, the power board has no power, or is not working.

Running SiTechExe:

The latest version of SiTechExe is version 0.90Q. It's available here: [http://siderealtechnology.com/SiTechSetup090Q.exe](http://siderealtechnology.com/SiTechSetup090Q.exe)

IMPORTANT: You must use SiTechExe 0.90Q or later, OR you must stick to the 1953 hertz on the Servo PID Sample Rate. Version 0.90Q also has a brushless controller information screen.

You can use the other instruction manuals for operation of SiTechExe, it operates the same as the Servol, or the ServoII. However there is one IMPORTANT difference: Under the Config tab, click on “Change Parameters”. Then click on the “ASCOM and Logging” tab. Now select the “Brushless Controller” Option. If you don't do this, you will have NO communication from SiTechExe.

Also important, you can't connect SiTechExe and BrushlessServoConfig to the controller at the same time. You need to close one application before running the other application.

Hopefully this manual will get you started. If you have trouble, please feel free to call me or email me anytime.

[grayarea@siderealtechnology.com](mailto:grayarea@siderealtechnology.com)

1-503-887-3701