myFocuserPro2 v291

V2.91 (10 February 2018)

A DIY project - stepper motor focuser solution based on Arduino

This document describes

- the range of myFocuserPro2 products
- building the myFocuserPro2 controller units
- attaching the myFocuserPro2 stepper motor to your telescope
- determining the correct myFocuserPro2 values for your telescope
- what to do if you lose your settings
- sample schematics and strip-board layouts
- the operation of Windows applications and ASCOM drivers available
- how to initially setup the myFocuserPro2 controller for your telescope
- testing procedures and programs
- manual and automated focusing methods

Once built - You must setup the focuser as described in Initial Setup otherwise you can damage your telescope.

Note:

myFocuserPro refers to v1xx of the myFocuserPro products. This was the first design and also works with the Moonlite drivers, so can be supported on the Mac (tested on MacBook with TheSky and Moonlite drivers) and Linux systems (tested with the INDI Moonlite driver under Ubuntu and Ekos)

myFocuserPro2 refers to v2xx of the myFocuserPro products. This design is v2 and is NOT compatible with Mac and Linux systems. Version 2 controllers do have a number of additional features that are not available on the v1 design and use a different protocol. You cannot use v2xx ASCOM or Windows software with a v1xx programmed controller or vice-versa.

Both systems are supported by either a V1 or V2 Windows application and ASCOM drivers.

If you have built a v1xx controller, and want to upgrade to v2xx ASCOM drivers and Windows application, this is easy. Simply load the equivalent v2xx firmware into the controller and use the supplied v2 ASCOM drivers and v2 Windows application.

Table of Contents

DISCLAIMER

This project is released into the public domain as is where is, with no obligation or responsibility accepted on the part of the author, for any mishaps or failures caused by this product or use of this product. Users intending to use this project or code do so at their own risk and usage of product and code is deemed to be acceptance of those risks.

In other words, the author accepts no responsibility to damage caused to any equipment or goods or self by using the ideas, schematics and code associated with this project, or loss of income or all other losses that may be incurred. No warranty if offered or implied.

COPYRIGHT RESTRICTIONS

The schematic, code and ideas are released into the public domain. Users are free to implement these for their personal use, but may NOT sell projects based on (or derived from) this project for commercial gain without express written permission granted from the author.

OVERVIEW

This is a DIY Stepper Motor Focuser Solution based on an Arduino Nano, bread boarded (optional PCB available) and enclosed in a project hobby box.

myFOCUSERPRO2 SPECIFICATIONS

- ASCOM driver, tested with FocusMax, Nebulosity, APT
- Windows Application (and Mini Application) for manual/remote control of focuser
- Display options (LCD1602/OLED/TFT) for positional information (Current and Target positions etc)
- Push button option (x2) for manual control of IN and OUT, manual zero position (hold down both push buttons for 2 seconds or reset in software)
- Keypad option
- LED indications for IN, OUT and External PWR
- Absolute focusing
- Temperature probe and Temperature Compensation
- Stepsize support
- Key focuser parameters saved in EEPROM
- Highly configurable
- External power supply required for driving stepper motor 7.5-12VDC @ 2A
- Reverse voltage protection on external 12VDC input
- Multiple driver board versions are available. Two versions detailed in this document – for other driver boards please go to the driver boards folder on Sourceforge
- PCB's (double sided, plated through holes, silk masked) available for all options

See this location for all the current available ready-made PCB.
https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/FRITZING%20PCB%20FOR%20ALL%20VERSIONS/

See this file on how to order
https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/FRITZING%20PCB%20FOR%20ALL%20VERSIONS/To%20Order%20PCB.txt/download

VIDEOS

Part1a Working Out Your Hardware Requirements
https://youtu.be/J6vh_iS2JsE

Part1b Assembling your focuser
https://youtu.be/aB0vM2evxEM

Part1c Testing your focuser
https://youtu.be/Kye_BaU67Aw

Part1d Adjusting the stepper motor current of the myFocuserPro2 controller
https://youtu.be/u2Z0dFaiiyM

Part2a Get the Software
https://youtu.be/KwAxsUjqi_o

Part2b Load ASCOM and Arduino IDE and Libraries
https://youtu.be/m1BJGCBR-nU

Part2c Programming the Firmware
https://youtu.be/2f4X6omc2NI

Part3a Initial setup of Focuser settings
https://youtu.be/mZmWG_vQ1uM

Part3b Overview of Focuser settings
https://youtu.be/yuXUDxkdAgU

Part3c Overview of Windows App Menu Options
https://youtu.be/jsq7kXDv3_Y

Part3d Overview of ASCOM Driver Settings
https://youtu.be/Gh4dpqDFouQ

Part3e Upgrading the Controller Firmware
https://youtu.be/cSvOVw8Djsw

Part3f Controlling more than one myFocuserPro2 controller
https://www.youtube.com/watch?v=sEvvWYNMCFs

Part4 myFocuserPro2 Home Position Switch
https://www.youtube.com/watch?v=ADi2W0nsypI

Part5a myFocuserPro2 Temperature Compensation
https://www.youtube.com/watch?v=YXRqP-V1fcM

Part5b Temperature Compensation in myFocuserPro2
Part2: Understanding temperature compensation
https://youtu.be/uIEgBXL9Cks

Part6 Using a 4x4 keypad with myFocuserPro2
https://youtu.be/Xws4XfaG-3s

myFocuserPro Arduino Focuser Part 1
https://www.youtube.com/watch?v=Ap4k-0QlNwQ

myFocuserPro Arduino Focuser Part 2
https://www.youtube.com/watch?v=MaQzxvlpFWE

myFocuserPro Arduino Focuser Part 3
https://www.youtube.com/watch?v=j5l-s9YrlL0

myFocuserPro RotaryEncoder
https://www.youtube.com/watch?v=9CnBHNKSRgs

myFocuserPro Breadboarding Nano and EasyDriver
https://www.youtube.com/watch?v=O3rwse9nAsI

SOME FOCUSER BASICS TO START WITH

This is important. Please ensure you read this section first as the size of your telescopes CFZ (critical focus zone) will determine which option is best for you.

To build the correct focuser that matches your system we need to know

- The critical focus zone
- How much your focuser moves in one full rotation (in mm)
- These two measurements will determine what stepper motor resolution is needed (steps per degree or stepsize in microns)

CRITICAL FOCUS ZONE

The critical focus zone (CFZ) is related to the focal ratio of a telescope, and defines the distance over which the image is in focus (measured in microns). It is ideal to get at least a few steps within the CFZ (5 minimum), and this is done by altering the gearing ratio or using half stepping to increase the number of steps per revolution of the stepper motor. As the focal ratio gets smaller (i.e. a faster telescope optic) the CFZ reduces and thus a higher resolution stepper motor (more steps per revolution) is required.

The basic formula I have used is

CFZ in microns = focal_ratio * focal_ratio * 2.2;

For accurate focusing, it is necessary to get at least 10 steps within the CFZ. This is done by altering the gearing ratio (such as using gears or a pulley belt system) or using micro stepping to increase the number of steps per revolution of the stepper motor.

MEASURING FOCUSER DISTANCE PER ONE REVOLUTION OF FOCUS KNOB

We need to know how much your focuser tube moves per one full revolution of the focuser knob (measured in millimetres). For a refractor and Newtonian telescope this should be relatively easy to determine.

For determining this value for Schmidt-Cassegrain (SCT) telescopes, look here.

MEASURING STEP SIZE

Step size is the distance (in microns) the focuser tube moves for one stepper motor step. First **measure** how much your focuser tubes moves for one full revolution of the focuser knob (distance in milli-meters). Then divide this number by the number of steps per 1 revolution of your stepper motor. The answer is the distance that the focuser moves per step (called **Step Size**).

Step Size in microns = (distance one full focuser rotation in mm * 1000) / stepper motor steps per revolution

This value is used to determine how many stepper motor steps there will be within the CFZ of the telescope (ideal is about 10). Simply divide the CFZ by the step size to get the step-size value.

Step Size is supported by myFocuserPro2 controllers. Some applications may require a valid setting for Step-Size in order to work correctly. If enabled, then the Step-Size value stored by the controller will be returned. If Step-Size is not enabled, the ASCOM driver will throw a “Not implemented” exception which the client application should handle. For more information in this PDF, click here. You can specify the Step-Size and enable/disable this setting when connecting to the controller. Setting values are remembered.

Calculating CFZ, Step Size and Stepper Motor Resolutions

Telescope	Stepper Motor
Orion ED80T-CF	**28BYJ-48**
With a focal ratio of f6 the CFZ is about 79microns. One full revolution of the focus knob moves the focuser 18.5mm	28BYJ-48 has 2048 steps per revolution The stepper motor is connected direct to the focuser shaft so one revolution of the stepper is one revolution of the focuser knob Hence, 2048 stepper motor steps move the focuser 18.5mm, so 1 step moves the focuser (18.5*1000)/2048 = 0.009mm which is the stepsize The focuser moves about 9 microns per full step, this there are about 9 full steps of the stepper motor within the CFZ (79/9=8.77)
	NEMA17-PG5
	NEMA17-PG5 has 1036 steps per revolution (at full steps) The stepper motor is connected direct to the focuser shaft so one revolution of the stepper is one revolution of the focuser knob The focuser moves about 17.85 microns per full step of the stepper motor (18.5*1000/1036) There are about 4 full steps of the stepper motor within the CFZ (79 / 17.85=4.4) Full stepping does not give enough steps within the CFZ, so try half-stepping Using half-steps the focuser moves about 8.93 (17.85/2) microns per half step There are about 8-9 half-steps within the CFZ (79 / 8.93 = 8.85) The NEMA17PG5 should be operated in HALF_STEP mode! We need a driver board that supports half stepping for a NEMA17-PG5
	NEMA17-PG27
	NEMA17-PG27 has 5370 steps per revolution (at full steps) The stepper motor is connected direct to the focuser shaft so one revolution of the stepper is one revolution of the focuser knob The focuser moves about 3.44 microns per full step of the stepper motor (18.5*1000/5370) There are about 23 full steps of the stepper motor within the CFZ (79 / 3.44=22.9) Full stepping gives enough steps within the CFZ
	NEMA14-0.9°
	NEMA17-PG5 has 400 full steps per (1600 steps at quarter stepping) The stepper motor is connected direct to the focuser shaft so one revolution of the stepper is one revolution of the focuser knob The focuser moves about 11.56 microns per 1/4 step of the stepper motor (18.5*1000/1600) There are about 4 full steps of the stepper motor within the CFZ (79 / 11.56=6.4) Full and stepping will not give enough steps within the CFZ, so try 1/4-stepping We need a driver board that supports 1/4 stepping for a NEMA17-PG5

Spreadsheet Calculations Showing Various Options

Example1: StellarVue Refractor f7.1 and NEMA17-PG5 half stepping

SV102T with NEMA17PG5 STEPPER MOTOR

usbFocus		SVFocuser
	mm per	mm per
Degrees	myFocuser	360
Per Step	Step	degrees
0.3475	0.0164	**17**
1036	steps per 360	
16.41	microns full step	
8.20	**microns 1/2 step**	

Per step the myFocuser motor will move the focuser 16.41 microns Full Steps

CFZ for StellarVue SV102T in microns			Number of steps for Motor	
Focal Ratio		cfz	Full-Steps	Half-Steps
7.1	**Prime**	**111**	**7**	**14**
14.2	2x Barlow	444	27	54
21.3	3x Barlow	998	61	122
36	5x Barlow	2773	169	338
5.68	**0.8 Reducer**	**71**	**4**	**9**

SV102T Focus Position		NEMA17 Steps (Half)	
0cm	Min Travel		
1cm		0	START
2cm		1218	
3cm		2436	
3.2cm	OSSAO	2680	
3.5cm		3045	
4cm		3654	
4.4cm	**Focus**	**4141**	
4.5cm		4263	
5cm		4872	
5.65cm		5664	STOP
6cm		6090	
6.5cm	Max Travel		

609.4117647 Full steps per cm
1218.823529 Half Steps per cm

Note: Motor is set to Reverse Direction and Half Steps for SV102T

As you can see, this will give 14 half-steps within the CFZ at f7.1 and 9 steps at f5.68, which is adequate for repeatable accurate focusing using a direct connection with a flexible coupler.

Example2: Orion ED80T-CF f6 Refractor with 28BYJ-48 Stepper Motor and ULN2003 Driver

ED80T-CF with myFocuser 28BYJ-48 Stepper

myFocuser			OrionFocuser
		mm per	mm per
Degrees		myFocuser	360
Per Step		Step	degrees
0.1758		0.0090	**18.5**
2048	steps per 360		
9.03	um per step		
4.52	um per half-step		

Per step the myFocuser 28BYJ-48 motor will move the focuser 9.03 microns

CFZ for Orion ED80T-CF in microns			Number of steps for Motor	
Focal Ratio		CFZ	Full-Steps	Half-Steps
6	Prime	79	9	**18**
12	2x Barlow	317	35	70
18	3x Barlow	713	79	158
30	5x Barlow	1980	219	438
4.8	0.8 Reducer	51	6	**11**

Note: Motor is set to Normal Direction and Full Steps for ED80T-CF
Note: ULN2003 driver does not support half-steps - if half-steps are needed, then use LD293D shield

As you can see, this will give 9 full-steps within the CFZ at f6 and 6 steps at f4.8. Using half-steps, this would just be adequate for repeatable accurate focusing using a direct connection with a flexible coupler.

Examples
My example focuser has four (4) full turns of the focuser knob from the minimum full IN position to the maximum full OUT position.

Example 28BYJ-48 Stepper
For the 28BYJ-48 stepper motor at 2048 steps per revolution, with the stepper attached to the single knob of the focuser, then this gives 2048*4 or 8192 maximum possible steps. We unclamp the flexible coupler and manually position the focuser to be ½ turn out from the minimum IN position, then re-clamp the flexible coupler.

We turn on the focuser and run the myFocuserPro2 Windows application, select the correct COM port and connect to the controller. The focuser is currently at position 5000 (the controller has defaulted to position 5000). We enter 0000 as the focuser position and click the SET POSITION button, which tells the controller that the current focuser position is reset to position 0. This ensures that the stepper cannot drive the focuser fully home (it will stop one half turn away).

To determine **maxStep**, we also assume that we will drive the focuser OUT but stop one half turn before the maximum stop of the focuser. For our example, this is three full turns of the focuser knob. In stepper motor steps this is 2048*3 = 6144, so we need to set maxStep to 6144 in the setup dialog box.

In the myFocuserPro2 Windows Application we enter 6144 as the Maximum Position and click the SET button to send this value to controller.

For the initial focuser position, we determine the half-way point (0-6144) and so the initial focuser position will be 3072.

Next, we move the focuser from position 0 to the mid-point by entering 3072 in the focuser position text box and then click the GOTO POSITION button to move the focuser. Once the focuser has stopped moving, we can then close the application and power off the focuser. *If you notice that the focuser does not move when the GOTO POSITION command is sent to the controller, it is likely that the direction is incorrect. Try enabling Reverse Direction and then clicking the GOTO POSITION button again.*

To check that everything is set, we turn on the focuser and restart the myFocuserPro2 Windows application. You will see that the focuser position will be set to the midway point (in our example 3072) and that the maximum position is set to 6144.

As long as the focuser is not manually moved, or the coupler disconnected, the focuser is now setup with the correct values. Each time we connect to the focuser, the correct settings will be sent to the controller and will be saved so they can be recalled next time we run the software or access the ASCOM driver.

Example NEMA17-PG5 Stepper
For the NEMA17-PG5 stepper motor using half-steps, there are 2072 steps per revolution, with the stepper attached to the single knob, then this gives 2072*4 or 8288 maximum possible steps. We unclamp the flexible coupler and manually position the focuser to be ½ turn out from the minimum IN position, then re-clamp the flexible coupler.

We turn on the focuser and run the myFocuserPro2 Windows application, select the correct COM port and connect to the controller. The focuser is currently at position 5000 (the controller has defaulted to position 5000). We enter 0000 as the focuser position and click the SET POSITION button, which tells the controller that the current focuser position is reset to position 0. This ensures that the stepper cannot drive the focuser fully home (it will stop one half turn away).

To determine **maxStep**, we also assume that we will drive the focuser OUT but stop one half turn before the maximum stop of the focuser. This is three full turns of the focuser knob. In stepper motor steps this is 2072*3 = 6216, so we set maxStep to 6216 in the setup dialog box.

In the myFocuserPro2 Windows Application we enter 6216 as the Maximum Position and click the SET button to send this value to controller.

For the initial focuser position, we determine the half-way point (0-6216) and so the initial focuser position will be 3108.

Next, we move the focuser from position 0 to the mid-point by entering 3108 as the focuser position and then click the GOTO POSITION button to move the focuser. Once the focuser has stopped moving, we can then close the application and power off the focuser. *If you notice that the focuser does not move when the GOTO POSITION command is sent to the controller, it is likely that the direction is incorrect. Try enabling Reverse Direction and then clicking the GOTO POSITION button again.*

To check that everything is set, we turn on the focuser and restart the myFocuserPro2 Windows application. You will see that the focuser position will be set to the midway point (in our example 3108) and that the maximum position is set to 6216.

As long as the focuser is not manually moved, or the coupler disconnected, the focuser is now setup with the correct values. Each time we connect to the focuser, the correct settings will be sent to the controller and will be saved so they can be recalled next time we run the software or access the ASCOM driver.

In order for the focuser to work correctly. Incorrect values for Maximum Position or setting the zero position incorrectly may cause damage to either the focuser or stepper motor.

Please note that the values will be different for your focuser and these will need to be determined by you in order for the focuser to work correctly. Incorrect values for Maximum Position or setting the zero position incorrectly may cause damage to either the focuser or stepper motor.

It is important that the stepper motor stops and does not try to drive past the minimum and maximum points of your focusers travel.

STEP SIZE AND CRITICAL FOCUS ZONE

This section will examine the relationship of Step-Size (SS) to the CFZ.

Q: I Have an SCT telescope, focal ratio is f10. How do I calculate the step size?

A: This involves a number of inter-related maths. We know that CFZ in microns = focal_ratio * focal_ratio * 2.2, thus for an f10 telescope this is

CFZ = 10 * 10 * 2.2
CFZ = 220 microns

To get 10 steps within the CFZ the required Step-Size will be

SS = 220 / 10
SS = **22**

thus a Step-Size of around 22microns is required. Now for the sake of simplicity, we will make some assumptions.

1. The stepper motor is connected direct to the focuser shaft
2. We have selected a stepper motor whose current is around 400mA at 12V

What we do NOT know is how far your focuser moves in one full turn of the focuser knob. This is important and we cannot go much further without this information. That distance is something you will need to measure (in milli-meters) before continuing.

Q: How do we measure how far the SCT focuser (primary mirror) moves in one focus knob revolution?

A: Using a Bahtinov mask to determine best focus, attach a diagonal and eyepiece (around 40mm is okay) which is inserted fully into the diagonal) and achieve good focus.

Next rotate the focus knob one full revolution. Then, without changing focus, slowly move the eyepiece outwards of the diagonal till focus is achieved. If focus gets worse as the eyepiece is slowly retracted, then you will need to start again, and after achieving best focus with the eyepiece fully inserted, this time rotate the focuser knob in the other direction.

Next measure the distance that the eyepiece has moved away from the top lip of the diagonal. Now we have the distance for one revolution of the focuser knob. Let us assume that you did measure it and your focuser moves 20mm in one full revolution.

Now we will proceed on that basic and in the following you can substitute the real value instead of the 20mm I am using.

Let us chose a NEMA stepper motor that runs at 12V, is rated at 400mA and has 200 steps per revolution.

CASE 1: NEMA at 200 steps per revolution at FULL STEPS

1 full stepper motor revolution is 200 steps and 1 full focuser knob revolution moves 20mm, so per step = 20/200 = 0.1mm or 100microns. This is not good enough because we need a step size of 22microns.

Even using HALF STEPS there would be 400 steps per revolution giving a step size of 50 microns, still too large. This means a standard NEMA is inadequate and we need a geared stepper motor for higher resolution (more steps per revolution)

CASE 2: NEMAPG5: 1028 Steps per Revolution at FULL STEPS

1 full stepper motor revolution for the NEMAPG5 is 1028 steps and 1 full focuser knob revolution moves 20mm, so per step = 20/1028 = 0.019mm or 19microns. This is OK as we need a per step size of 22microns and what we have is 19.

If your SCT focuser moved 20mm per focuser knob revolution then a PG5 NEMA will be good to go.

So, what is the important need to know information - how far your focuser moves in one revolution.

For a refractor or Newtonian telescope, this is much easier and has already been covered.

Q: What can I do about backlash with the SCT focuser?

A: Backlash is a major problem with an SCT focuser. The only method is always try to focus in one direction without reversing. This means using an autofocusing program such as FocusMax.

An alternative is to affix a Crayford type focuser to the rear cell, and focus using the Crayford rather than the SCT focus knob. This eliminates the worst of the backlash/ Special focusers are available that can be used with a field flattener/reducer (they house the flattener/reducer within the focuser). But using a Crayford focuser attached to the SCT rear cell can be expensive.

STEPPER MOTORS

If using direct drive (stepper motor is connected to the focuser shaft using a flexible coupler) then the ideal requirements are

- Bipolar stepper motor
- 12VDC at less than 500mA
- Holding High torque (> 75oz.in)
- Small step angle (0.9°) or geared planetary reduction drive, micro-stepping
- Low weight (<300g)

CHOOSING A STEPPER MOTOR DEPENDS UPON A NUMBER OF FACTORS

- Focal ratio of telescope (the smaller the number the more steps per revolution you will need)
- Weight of optical train (heavier requires a stepper motor with higher torque or a geared drive)
- 12VDC
- Maximum current around 400mA

Choosing the right stepper motor is also combination of voltage, current, steps per revolution, inertia torque, holding torque, size and weight). The voltage and current requirements are controlled by the driver board being used (or you could build your own driver circuit to supply higher voltages and currents).

TYPICAL STEPPER MOTOR CHARACTERISTICS

The number of steps per revolution required is directly related to the focal ratio of the telescope and hence the CFZ. For focal ratios of f7 or lower, a geared stepper motor (or belt drive system) such as the PG27 is preferred.

Stepper	Current	Voltage	Steps Per Revolution
28YBJ-48	320mA	5-7.5V	2038
NEMA17	400mA	12V	200
NEMA17-PG5	400mA	12V	1028
NEMA17-PG27	400mA	12V	5370
NEMA14	400mA	12V	400

The Nema17 stepper motor provides much greater torque than the 28BYJ-48 stepper motor. The Nema17-PG27 provides the highest torque and the most number of steps per revolution and is suitable for heavy imaging trains.

When using NEMA17 stepper motors with the L293D Motor Shield and L298N driver boards (and the ULN2003 with the 28BYJ-48), the number of steps per revolution must be specified in the Arduino firmware file. For all other driver types, this is not necessary.

const int stepsPerRevolution = 1028;// NEMA17-PG5 motor
// you need to change the above line to reflect your stepper motor, examples below

AVAILABLE/RECOMMENDED STEPPER MOTORS

The following stepper motor is recommended for heavy imaging trains
Gear Ratio 27:1 Planetary Gearbox with Nema 17 Stepper Motor 17HS13-0404S-PG27
http://www.omc-stepperonline.com/gear-ratio-271-planetary-gearbox-with-nema-17-stepper-motor-17hs130404spg27-p-249.html
$28.29USD
5370 steps per revolution full-step, 10740 steps per revolution half-step
*ample torque and suited to fast telescopes f2 – f8

Other stepper motors
http://www.omc-stepperonline.com/gear-ratio-51-planetary-gearbox-with-nema-17-stepper-motor-17hs130404spg5-p-140.html
Gear Ratio 5:1 (5.18:1) Planetary Gearbox with Nema17 Stepper Motor 17HS13-0404S-PG5, $27USD
1036 steps per revolution full-step, 2072 steps per revolution half-step
*ample torque for heavy imaging trains and enough steps for f7 telescopes

http://www.omc-stepperonline.com/nema-17-stepper-motor-34mm-12v-04a-26ncm37ozin-17hs130404s-p-166.html
8.28USD, 200 steps per revolution full-step, 400 steps per revolution half-step
*steps per revolution is not enough for accurate focusing < f7 (would need to use gears or belt drive)

http://www.ebay.com/itm/261110217491?_trksid=p2060778.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT
1.06USD, FULL STEP = 2038 steps per rev, HALF STEP = 4076 steps per rev (only with L293D Shield)
* May not have enough torque for heavier imaging trains
* Operate on 7.5VDC else motor will overheat if using Coil Power ON

Keep in mind that the stepper motor will be the single most expensive item, and it is best to get a stepper motor that will give great results. The stepper motors recommended here are excellent choices.

CONCERNED ABOUT STEPPER MOTOR SIZE AND WEIGHT

An alternative is the NMEA14 which is much lighter, but uses 0.9 degree step movement with 400 steps per revolution. Using microstepping, this gives 800 steps at half-stepping. This stepper is best used with the DRV8825 driver board.

This stepper motor is ideal for the majority of focusing solutions.

Nema 14 Bipolar Stepper Motor .9deg 0.4A 11Ncm 14HM11-0404S

START HERE

Decide on the stepper motor and driver board combination you will use. If you have not, we recommend the below

We recommend you use the NEMA17PG27 stepper motor [17HS13-0404S-PG27].
We recommend you use the Nano+DRV8825-HW203 driver board or REV4 PCB

28YBJ-28	ULN2003 + Nano
	L293D Shield + UNO
NEMA17	L293D Shield + UNO
	DRV8825 + Nano
	EasyDriver + Nano
	RAPS128 + Nano
NEMA17PG5	L293D Shield + UNO
	DRV8825 + Nano
	EasyDriver + Nano
	L298N + Nano
	RAPS128 + Nano
NEMA17-PG27	L293D Shield + UNO
	L298N + Nano
	DRV8825 + Nano
	EasyDriver + Nano
	RAPS128 + Nano

Decide if you want the optional LCD, push buttons and temperature probe

YES	Implement the FULL option (and use code with extension _F)
NO	Implement the Minimal option (and use code with extension _M)

Download the required files

- Documentation file contains good information on build instructions, initial setup and usage
- Schematic
- Layout
- Wiring of connectors, RS232 etc.
- Test programs if available
- Arduino code
- Windows application
- ASCOM driver

Build the controller

Test the controller (remember to use precautions in connection and a 12V external power supply)

Connect the stepper motor to your focuser and ensure that the initial setup is completed

WHAT IS THE EASIEST TO BUILD

All builds require some amount of soldering and each build has some part that has a degree of difficulty. The Arduino UNO + L293D Motor Shield mounted in an Arduino case requires soldering of the push buttons, RS232 connector, temperature probe socket,12V power socket, and some header pins on the L293D shield in order to be able to connect +5, GND, D2 and other pins to components.

We recommend using the DRV8825_HW203_F PCB (PCB available online).

RECOMMENDED BUILD OPTION

The recommended builds are

- A Fritzing PCB (supports Full, Minimal plus Temperature probe and Minimal options)
- DRV8825 driver chip with a NEMA17PG5 (or NEMA17PG27, or NEMA14)

DIY BOARD OPTIONS

- Stripboard Full but used as Minimal (you can add the LCD and temperature probe later if desired)
- PCB Full but used as Minimal (you can add the LCD and temperature probe later if desired)

DECIDING ON HARDWARE AND FIRMWARE TO USE

Board	Driver	Full?	Stepper	Stepping	.ino file
Nano	ULN2003	F	28BYJ-48	F	Focuservxxx_ULN2003
Nano	ULN2003	M	28BYj-48	F	Focuservxxx_ULN2003_M
Uno	L293D	F	Nema17PG5/28BYJ-48	F/H	Focuservxxx_L293D_F
Uno	L293D	F	Nema17PG27	F/H	Focuservxxx_L293D_F
Nano	L293DMini	F	Nema17PG27	F/H	Focuservxx_L293DMini_F
Nano	L298N	M	Nema17PG27	F/H	Focuservxxx_L298N_M
Nano	L9110S	F	Nema17PG27	F/H	Focuservxxx_L9110S_F
Nano	**DRV8825/HW203**	**All**	**Nema17PG27**	**F/H/4/8/16/32**	**Focuservxxx_DRV8825_HW203_xx**
Nano	ST6128	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuservxxx_ST6128_xx
Nano	RAPS128/HW203	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuservxxx_RAPS128_HW203_xx

F=Full - code support for LCD, Push Buttons, Temperature Probe
M=Minimal, NO code support for LCD, No Push Buttons, No Temperature Probe

ASCOM DRIVER	myFocuserASCOMSetupxxx
WINDOWS APPLICATION	myFocuserWin_xxx

NOTE: If you just want to use the ASCOM driver, you can test the ASCOM driver installation and operation using the following application: myFocuserAscomAppVxxx

NOTE: The recommended method to reset the maxStep value is to use the Windows Application. You would do this during the initial setup of the focuser. All versions are supported by a Windows Application and an ASCOM driver.

FRITZING PRINTED CIRCUIT BOARD OPTIONS

There are a number of PCB designs (double sided, plated through holes, silk screened, professionally made) that can be ordered (with or without a part pack – not all parts are in the pack and some may need to be purchased from an alternative supplier such as eBay) online.

Professional PCB: MYFOCUSERPRO2 PCB CAN BE ORDERED ONLINE

DRV8825HW203-M-MT-F-BT	https://aisler.net/p/DWEURMAC	
DRV8825HW203_FIRBT	https://aisler.net/p/UGHLNFJC	for infra red controller
DRV8825HW203_FRE	https://aisler.net/p/KMHRZAON	for rotary encoder
A4998-M-MT-F-BT	https://aisler.net/p/RLQDYNPU	
DRV8825-M-MT-F-BT	https://aisler.net/p/QVXMBSWW	
EASYDRIVER-HW203-M-MT-F-BT	https://aisler.net/p/NCMLHIHM	
EASYDRIVER-HW203_FRE	https://aisler.net/p/ISHXZBYS	
L293DMINI-M-MT-F-BT	https://aisler.net/p/WXCWWRJH	(not TFT)
L298N-M-MT-F-BT	https://aisler.net/p/QWESFIOS	
L9110S-M-MT-F-BT	https://aisler.net/p/ZSXWHVTG	
ULN2003-M-MT-F	https://aisler.net/p/DEYGMRQC	

The F (Full) boards also support MT (Minimal+Temp) and M (Minimal) options if the associated components are not mounted on the PCB. In other words, you can use the same Fritzing PCB build to support different build options

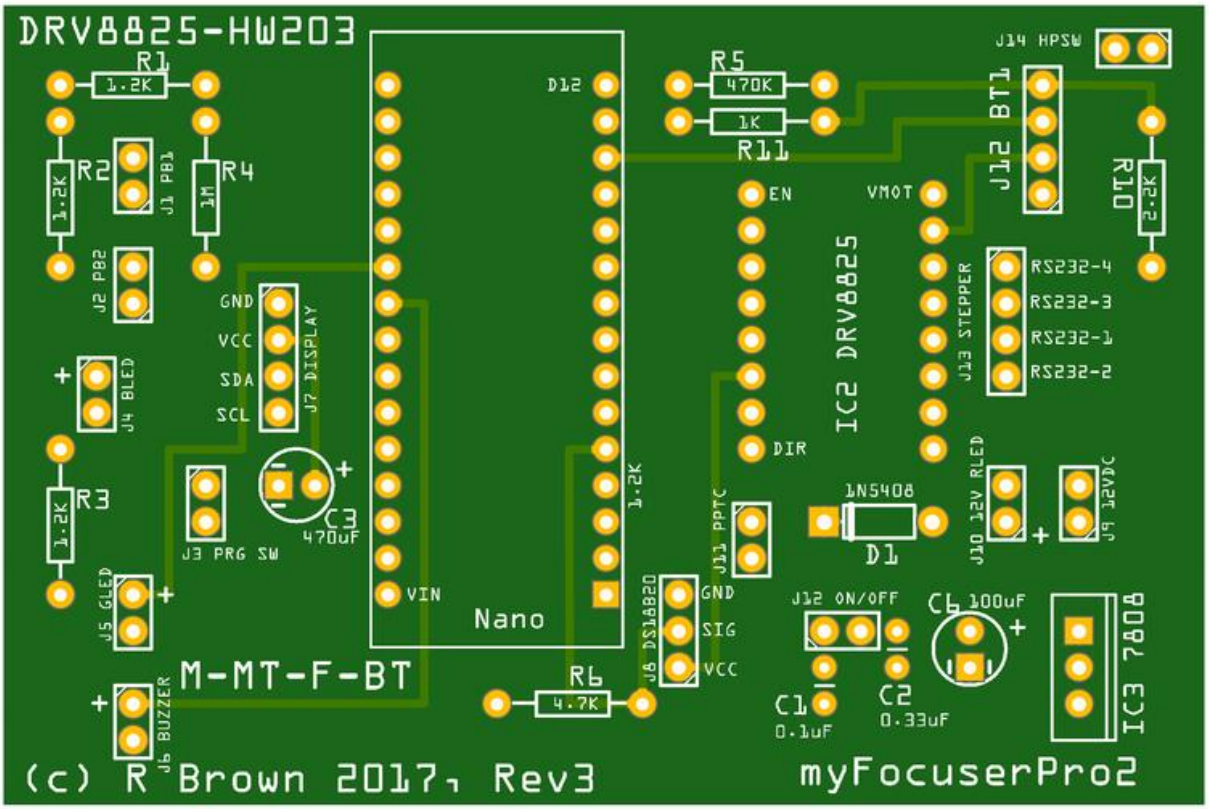
- Minimal
- Minimal plus temperature probe
- Full including display, push buttons

simply by adding or omitting certain components and changing the firmware version file.

The ASCOM driver and Windows software supports ALL options. The correct Arduino code version must be used with the correct driver board build option.

There is NO Fritzing PCB for the UNO+L293D Shield.

FRITZING DRV8825HW203-M-MT-F-BT PCB



This PCB supports

- Buzzer, IN/OUT LEDS and PWR LED
- IN/OUT Push buttons
- Temperature Probe
- Display (OLED and LCD)
- Microstepping to 1/32
- Power-On reset circuitry
- Bluetooth
- Home Position switch
- Voltage regulator for operating focuser from 12V

Not all options need to fitted to the board. You can leave options out and then configure the software to match the options you have added to the board.

OTHER BUILD OPTIONS DIY PCB AND STRIPBOARD

There are a number of other build options for Stripboards and DIY PCB. The ASCOM driver and Windows software supports ALL options. The correct Arduino code version must be used with the correct driver board build option.

The latest DIY PCB/Stripboard is DRV8825_HW203_F_WIFI_REV4 ***which has a common layout that supports most options.***

In other words, you can use the same DIY Rev4 build to support different build options

- Minimal
- Minimal plus temperature probe
- Full including display, push buttons
- Full including Blue tooth control

simply by adding or omitting certain components and changing the firmware version file.

Current Limits for Build Options

Each build option has specific current limits imposed by the stepper motor driver selected. The current limit of the driver board affects the choice of stepper motor.

Driver Board	Suggested Maximum Current	Suggested Stepper Motor
ULN2003	300mA	28BYJ-48
L293D Motor Shield/Uno	500mA	Nema17PG5/PG27 or 28BYJ-48
DRV8825	**1.5A**	**Nema17PG5/PG27 or NEMA1414HM11-0404S**
EasyDriver v44	600mA	Nema17PG5/PG27
L298N	2A	Nema17PG27
A4988	1.5A	Nema17PG5/PG27
ST6128	2A	Nema17PG5/PG27
RAPS128	2A	Nema17PG5/PG27
L293D Mini Driver Board	600mA	Nema17PG5/PG27
L9110S Driver Board	800mA	Nema17PG5/PG27

If you already have a stepper motor, find its operating voltage and current levels and select a driver board from above that will be able to support your stepper motor (and still have some room to cope – you cannot run a 2A stepper motor on a 2A driver board as the driver board cannot run at maximum all the time).

If your stepper motor is rated at 2A or above, it cannot be supported due to the high current and you will need to buy a suitable stepper motor – the recommend stepper motor is the NEMA17-PG27

If your stepper motor is a unipolar stepper motor, it can only be supported by the ULN2003 and L293D motor shield options. Bipolar motors are preferred as they provide higher torque than a unipolar stepper motor.

myFOCUSERPRO2 COMPARISON OF DRIVER BOARDS

Driver Board	Typical I	Peak I	Typical V	Max V	Steps	Stepper
ULN2003	350mA	500mA	12V	12[1]	F[1]	28BJY-48
L239D Shield[2]	450mA	600mA	12V	12[1]	F $\frac{1}{2}$	NEMA17-PG5/PG27
DRV8825[3]	**1.5A**	**2.2A**	**12V**	**45**	**F** $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$	**Nema17PG5/PG27 or NEMA1414HM11-0404S**
EasyDriver	500mA	750mA	12V	30	F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
L298N	2A	3A	12V	35	F $\frac{1}{2}$	NEMA17-PG27
A4988	1.5A	2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
ST6128	< 2A	2.2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
RAPS128	< 2A	2.2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
L293D Mini Driver Board	< 600mA	1.2A	12V	25V	F $\frac{1}{2}$	NEMA17-PG5/PG27
L9110S	< 600mA	800mA	12V		F $\frac{1}{2}$	NEMA17-PG5/PG27

Note: The L293D, L9110S, L298N and L293D Shield options only provide for half stepping. This means the best stepper motor choice will be the PG27 or PG5 (which depends on the step size and how many steps you can into the CFZ)

Note: The ULN2003 provides FULL stepmode only.

The firmware to use for each driver board is shown in the table below

Driver Board	Arduino Code
ULN2003	Focuservxxx_ULN2003
L239D Shield[2]	Focuservxxx_L293D
DRV8825_HW203[3]	**Focuservxxx_DRV8825_HW203**
EasyDriver	Focuservxxx_DRV8825_HW203
L298N	Focuservxxx_L298N
A4988	Focuservxxx_A4988_HW203
ST6128	Focuservxxx_ST6128
RAPS128	Focuservxxx_RAPS128_HW203
L293D Mini Driver Board	Focuservxxx_L293DMINI
L9110S	Focuservxxx_L9110S

[1] Limitation of board
[2] Can only be used with Arduino Uno or Mega
[3] This is the recommended stepper motor to use

DRIVER BOARD FILES

The following table has links to each of the PDF's for each driver board build option.

A4988
DRV8825-HW203
DRV8825-TFT22
PCB DRV8825-HW203-OLED-ROTARY-ENCODER
PCB DRV8825-HW203 REV4
EASYDRIVER-HW203
L293D Mini
L298N
L9110S
RAPS128
ST6128
ULN2003
UNO + L293D MOTOR SHIELD
LET'S MAKE A DRV8825 HW203 FULL myFocuserPro2
LET'S MAKE A L298N MINIMAL myFocuserPro2
LET'S MAKE A ULN2003 MINIMAL myFocuserPro2

To find out more about the available driver board options, the Driver Boards folder contains is a list of all the Driver board options, Schematics, Layout, PCB, Parts Lists and other details.

LIBRARY FILES

Many build options will require additional 3rd party libraries (for example, temperature probe, LCD display). After extracting the firmware zip file, the Library files will be in a folder called **myFocuserPro2libraries**.

All of these folders MUST be copied into the Documents\Arduino\libraries folder on the Windows computer.

FIRMWARE FILES

The CODE ARDUINO FIRMWARE folder contains a .ZIP archive of all the current focuser firmware options for each driver board. Each folder contains all the files needed for that option. Please do not try to edit or change these files. It is best to ensure that the file compiles cleanly before attempting to make any changes. The filename looks like

Focuserv262_DRV8825_HW203

In this example, the version number is 262, the driver board is DRV8825 with the HW203 layout. The file contains all the code for each variation, such as a FULL version that supports a display, temperature probe and push buttons, and BT supports control via Blue Tooth. Key notations are

M	Minimal
MT	Minimal plus temperature probe
F	Display, Push Buttons, Temperature probe
RE	Rotary Encoder instead of push buttons
IR	Infra-red remote controller
BT	Bluetooth support

Within each firmware file there is an option to enable or disable each option. This lets you build a controller with specific options and then use a single file to generate the required firmware.

STEPPER MOTORS

MICROSTEPPING

It is important to get a sufficient number of steps per revolution as this will determine the accuracy of focusing. The number of steps required depends on the focal ratio of the telescope. It is possible to determine what is best for a particular telescope (see the section on Critical Focus Zone).

One method of increasing the number of steps per revolution is to use a gear system with a flexible belt drive. Another way to increase the steps per revolution is to use micro stepping (stepping the stepper motor in-between full steps) or a geared stepper motor (PG5, PG27).

Please note that the ULN2003 option does not support half-stepping. The L293D shield supports half stepping on both the 28BYJ-48 and the NEMA17-PG5 stepper motors.

If you have an f6 or f7 refractor telescope, then half-stepping should be used with the NEMA17-PG27 unless the stepper motor has been attached using a belt drive and reduction gears.

The downside associated with half-stepping is the decrease (30%) in available torque.

A NOTE ABOUT TORQUE

Torque is a measure of how much force the stepper motor can exert on an object. The higher the torque value the greater the force that can be exerted. Torque is dependent upon voltage, current, number of coils, the efficiency of the motor, strength of the magnets used and other factors.

- Bipolar stepper motors provide 40% more torque than an equivalent unipolar stepper motor
- Choosing to operate the stepper motor in HALF-STEP mode will result in a 30% reduction of torque
- Torque can be increased significantly by using a gear reduction or pulley/belt drive system, but there are trade-offs
- It requires more torque to drive a focuser when the telescope is in the vertical position. The system should always be tested with the telescope pointing at Zenith to see if the stepper can drive the focuser inwards without any issues (such as missed steps or failure to move)
- For a small telescope with a light focuser and light camera, the 28BYJ-48 stepper should have enough torque to drive the focuser
- Increasing the voltage or current is not really an option. The L293D Motor Shield cannot readily support higher currents than 400mA continuous
- For the L293D Motor Shield, the AF_MOTOR library supports SINGLE and DOUBLE parameters when stepping the motor, with DOUBLE resulting in more torque. The controller code uses DOUBLE
- With any system, the user must ensure that the stepper motor does not attempt to drive the focuser either below or beyond the focuser mechanical limits

There are really two different types of torque that you need to consider. The first is the inertia force, the amount of force that the stepper can apply when attempting to drive a stationary motor so that the stepper motor can overcome the inertia of the system, weight of focuser, imaging train and friction and begin to move.

The second is the holding torque, which is the force the stepper motor exerts to prevent the motor from turning when the motor is stationary. This is done by applying power to the coils (referred to as coil power in the software). If the holding torque is low then the focuser imaging train could have enough weight to move or slip either during a move command or once the move command is finished. A geared motor with coil power ON is preferred for heavier imaging trains or imaging near the Zenith.

STEPPER MOTOR WIRING (FRITZING PCB)

DRIVER Jx-STEPPER HEADER 4P	RS232 Pin	NEMA17PG27/NEMA17PG5	NEMA17	14HM11-0404S
DRV8825/A4998/DRV8825-HW203				
Jx-Pin1	4	RED (Coil2)	GREEN	RED
Jx-Pin2	3	BLUE (Coil2)	BLUE	BLUE
Jx-Pin3	1	GREEN (Coil1)	BLACK	BLACK
Jx-Pin4	2	BLACK (Coil1)	RED	GREEN
L9110S				
B+	4	RED (Coil2)	BLUE	BLUE
B-	3	BLUE (Coil2)	GREEN	RED
A-	1	BLACK (Coil1)	RED	GREEN
A+	2	GREEN (Coil1)	BLACK	BLACK
L293D-MINI				
A+	4	RED (Coil2)	BLACK	BLACK
A-	3	BLUE (Coil2)	RED	GREEN
B+	1	BLACK (Coil1)	GREEN	RED
B-	2	GREEN (Coil1)	BLUE	BLUE
L298N				
B1	4	RED (Coil2)	BLACK	BLACK
B2	3	BLUE (Coil2)	RED	GREEN
A2	1	BLACK (Coil1)	GREEN	RED
A1	2	GREEN (Coil1)	BLUE	BLUE

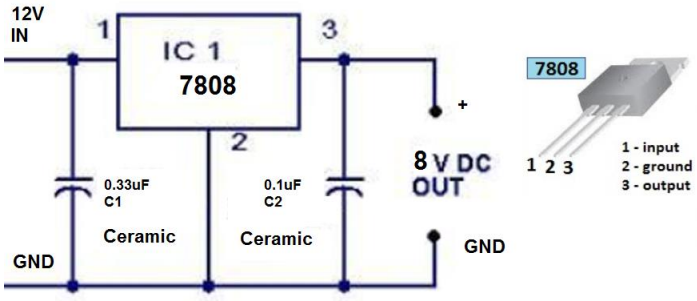
myFOCUSERPRO2 PURCHASE LIST

Please refer to the** spreadsheet **for the required parts. Fritzing PCB's can be ordered** online**.

POWER AND PROTECTION FOR VIN

You cannot power everything from a USB cable. A USB connection has a limited power capability, which can be exceeded if you try to power everything including the stepper motor from a USB connection.

The recommended NEMA stepper motors require a 12V external power supply. This voltage can also be used to power the controller via a voltage regulator circuit. The Arduino chip looks at the voltage on the VIN pin, and if higher than the 5V USB supply will use that power connection as a preference. Some clone Nano chips purchased off eBay tend to suffer damage if run off 12V, so a step-down voltage regulator is used to lower the voltage from 12V to 8V for VIN.



This means the controller can be powered in the field from 12V or a car battery. If using a USB connection, the external power supply is still required to power the stepper motor.

The LM7808 voltage regulator circuit provides for over-voltage protection of the VIN input for the Nano controller.

The capacitors provide noise suppression on the input and output of the voltage regulator and are required.

The L293D Motor Shield does not require power-on circuitry or protection for VIN. On all other builds, this circuit is required to avoid potential burnout of the Nano controller if the supply voltage is too high.

REVERSE VOLTAGE PROTECTION

WHY USE A 10A10 DIODE?

A 10A10 diode rated at 10A provides reverse voltage protection on the external power supply rail. If this diode is omitted, then accidental reversal of the input voltage to the controller will have catastrophic results.

The maximum current draw to the stepper motor is around 350mA (for 28BYJ-48) or 400mA for the Nema17. The forward voltage drop across the 10A10 diode is around 1.1V. A 2A quick blow fuse provides excess current protection. The voltage drop does not cause any issue with the circuits and is not an issue.

The maximum recommended input voltage is 12VDC. Ensure this protection circuit for VIN is implemented if attempting to power the controller from a car battery.

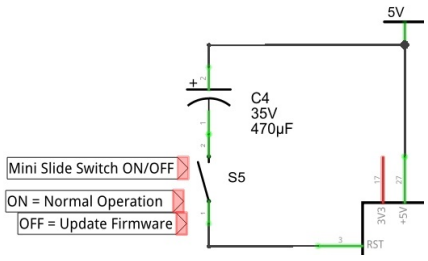
Note: Use 7.5-9VDC with the 28BYJ-48 stepper motor if using Coil Power ON otherwise the stepper motor may get very hot.

Use an external power supply of 9V for the ULN2003 driver board
Use an external power supply of 12V for all other driver boards

POWER ON RESET CIRCUITRY

The following optional circuit provides a means of disabling the Arduino reboot that occurs when connecting to the controller via USB.

myFocuserPro2 comes with its own Windows Applications and ASCOM drivers. When connecting, the drivers attempt to restart the controller so it starts in a known state. If you wish to prevent this reset when the software connects to a controller, an optional power-on reset circuit is required.



The downside is that a switch must be used to disable the circuit. In normal operation, the switch is in the ON position. However, to reprogram the Nano with a firmware update, the switch must be moved to the OFF position before uploading the new firmware.

The advantage of NOT fitting the power on reset circuit is that the controller will always initialize to a known state when connection is made from the Windows app or ASCOM driver.

In general, the power-on reset that occurs when connecting to a myFocuserPro2 controller is not an issue when using the supplied Windows application or ASCOM driver. The reset takes about 2-3s before the controller is in a state to respond to any request.

However, some client applications expect a quick response and may generate a timeout error when attempting to connect to a myFocuserPro2 controller. If this is the case, then disabling or turning off the power-on reset circuit is the preferred option.

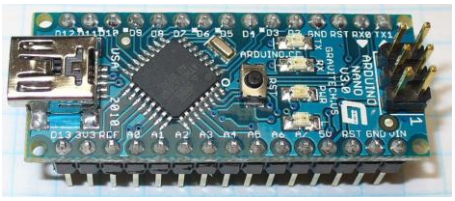
PRECAUTIONS

WARNING - NEVER CONNECT/UNPLUG STEPPER MOTOR CABLE WHEN EXTERNAL POWER IS ON

WARNING - NEVER CONNECT/UNPLUG TEMPERATURE PROBE CABLE WHEN POWER IS ON

NANO CH340G

The controller uses the "Mini USB Nano V3.0 ATmega328 16M 5V Micro-controller CH340G board for Arduino". This board does NOT use an FTDI chip so there will be no issue powering the board from 12VDC via an LM7808 voltage regulator wired to VIN.



The Arduino Nano can derive power from the mini-USB, VIN (pin 30) or 5V (pin 27). The Nano will select the highest voltage source (which will be VIN when the 12VDC is connected).

FTDI Nano Chips – Not recommended

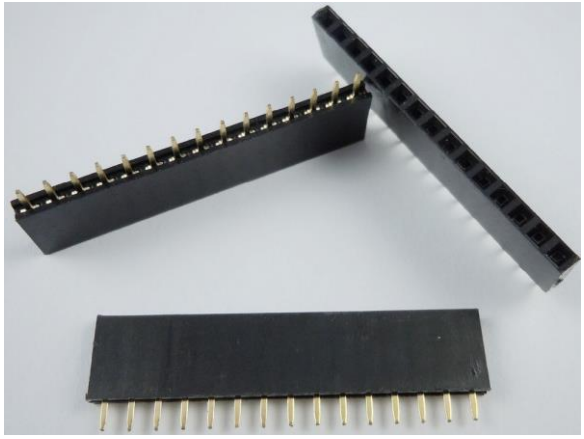
Nano chips such as the FTDI FT232RL chip is only powered if the board is being powered over USB. As a result, these chips when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

This means if we use a Nano FTDI chip we cannot connect VIN to 12VDC as we would lose the RS232 connection, so it is best not to use a Nano chip which has an FTDI driver.

The CH340G Nano ATMEGA328 board as recommended for this project.

Be sure to purchase the CH340G Nano option. Other Nano boards may not work.

Mounting the Nano controller on a PCB or strip-board



It is highly recommended to mount the Nano using Arduino 15 pin headers. This means the board can be tested to ensure that the correct voltage is present on the VIN pin.

Once this has been tested, power off the board, then insert the Nano into the 15P headers and testing can then continue.

If anything happens to the Nano controller, the faulty part can easily be removed and a new controller inserted, without altering anything on the board.

PRECAUTIONS

Never disconnect or connect the stepper motor when the myFocuserPro2 Controller or external power is ON. This will result in damage to the driver board

Never disconnect or connect the temperature probe when the myFocuserPro2 Controller or external power is ON. This may result in damage to the Nano board

myFOCUSERPRO2 MODES OF OPERATION

The focus controller can operate in a number of different modes, depending on the user requirements

- **Local Manual**
  The controller operates on External Power (7.5-12VDC) and the user presses the IN and OUT buttons to control the focuser position
- **Using the ASCOM driver OR the Windows application to control the focuser**
  The controller operates on External power and is connected to a computer via USB cable. The client application sends commands to control and move the focuser

UPLOADING ARDUINO FIRMWARE CODE TO THE UNO/NANO

The Arduino Sketch IDE software v1.6.8 has been used with this project and it is recommended you use this version.

To upload any firmware to the controller, select the correct board from the Tools > Board menu. Next select the correct serial port from the Tools > Serial Port menu.

Press the upload button  in the Arduino environment. The board will automatically reset and the sketch will be uploaded.

The term "Arduino Firmware Code" refers to the Arduino program (ends in .ino) that contains the focuser code and when executed by the Arduino, makes the chip act like a myFocuserPro2 controller.

PUSH BUTTON MOMENTARY SWITCHES (Optional)

Two momentary switches (SPST ON-OFF) connected via a voltage divider network provide an option for manually moving the stepper motor IN or OUT.

12mm Waterproof Lockless Momentary Push button Mini Round Switch

The switches are implemented using a voltage divider network and connected to a single analogue pin (A0).

Holding down both switches for 2 seconds will reset the current focuser position to 0. An audible beep is sounded once the position has been set to 0, at which point the switches can be released.

Instead of push buttons, you could use a Rotary Encoder (Keyes 040 supported) or an IR Remote Control. Please read the Driver Board Option PDF's for more information on these options.

When temperature compensation is enabled, the push buttons are disabled as well as any move commands.

Note: If the momentary switches are NOT implemented, then it is important to remove the push button switch code from the Arduino source otherwise the controller will not function correctly if the FULL version of the firmware is loaded on the controller (identified with the _F in the filename). Alternatively, you can use the MINIMAL Arduino code version (identified using _M or _MT) which does NOT support the LCD, push-buttons or the temperature probe.

WHICH WAY IS IN AND OUT?

Normal convention is that IN moves the imaging train (or eyepiece) IN - closer to the telescope, and OUT moves the imaging train (or eyepiece) away from the telescope.

Depending on the wiring of the stepper motor coils to the driver board or the way in which the stepper motor is physically connected to the focuser, pressing the IN button or sending an IN command (using the Windows or ASCOM driver) could move the focuser in the wrong direction. If this is the case, check the Reverse Direction checkbox to ensure that when an IN command is sent or the IN button pressed, that the focuser moves INWARDS.

LCD1602/I2C DISPLAY (Optional)

The focuser project provides for an optional LCD1602 display that shows the current and target positions of the focuser.

This would be useful in manual control where the push buttons are used to control the focuser.

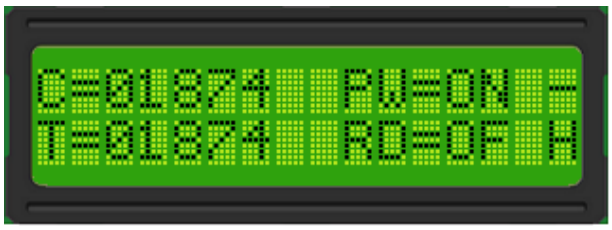
LCD1602 IIC I2C TWI 1602 Serial Port LCD Display Module

Initial Startup Screen	**Explanation**
MyFocuser_XXXXX	Driver Board version
2.x.x	Program version
Program running	
C=NNNNN PW=OF +	Current focuser position, Coil Power, Temperature Compensation
T=NNNNN RD=ON F	Target focuser position, Reverse, Stepping Mode

The LCD display is split across two pages, displayed one after the other after a short delay. The length of time an LCD screen is displayed for is user configurable.

LCD Screen Page 1

The first screen displays the focuser current position, the target power, the step mode and the status of coil power, reverse direction and temperature compensation settings.



PW indicates if power is kept to the stepper coils once a move is completed. For further information, please refer to the Readme.htm file that is available when the ASCOM driver is installed.

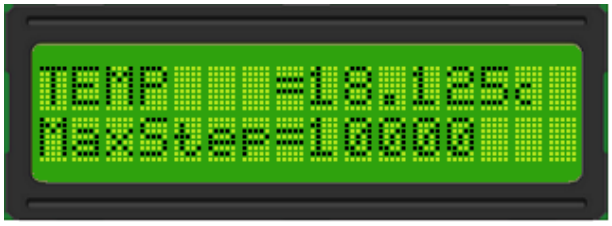
RD indicates reverse direction is either ON or OF(F)

F/H indicates the stepping mode (F=Full, H=$\frac{1}{2}$, 4=$\frac{1}{4}$, 8=$\frac{1}{8}$, 16=$\frac{1}{16}$, 32=$\frac{1}{32}$, 64=$\frac{1}{64}$, 1+=$\frac{1}{128}$ step mode)

- (or +) indicates if the temperature compensation is disabled/enabled

LCD Screen Page 2

The second screen displays the temperature (c= Celsius and f = Fahrenheit) and the current value for maxSteps.

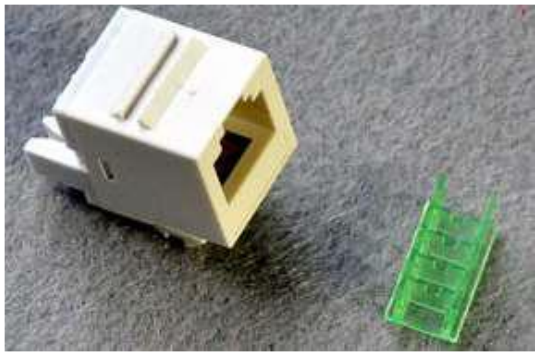


TEMPERATURE PROBE (Optional)

One temperature probe (DS18B20) is supported. The default resolution is set by the controller to 10-bit giving 0.25 degree of resolution (the resolution can be changed). The accuracy of the measured temperature is within 0.5 degrees Celsius.

Temperature Probe Connection

The prototype controller using 1 3pin stereo audio connection for the temperature probe (Tip=Yellow, Ring=Red, Sleeve=Black). Some users reported damaging their controller when removing or connecting the temperature probe with power ON.



The probe connects to an RJ11 6P4C panel mount female connector. *Disconnection or reconnection of the probe must be done when power is OFF*. VCC and GND are wired separately.

The controller automatically detects the DS18B20 probe on start-up or reset. If no temperature probe is connected the temperature is set to 20 (Celsius).

The temperature probe is optional.

myFocuserPro2 firmware files that support a temperature probe have a suffix of _F or _MT

Temperature Probe Placement

The sensor end of the probe is fitted so that it is on the metal tube of the telescope about ½ way between the optics and the focuser (perhaps secured using tape or plastic tie). Cable length of the purchased probe from ebay is around 1 meter (though you can extend this by wiring an extension cable with suitable connectors).

Temperature Probe Calibration

The ASCOM driver and Windows application can add an offset value to the value returned by the temperature probe. This provides a means of calibration for the probe, and the offset value is remembered by the Windows application and the ASCOM driver.

Warning - Never Plug-In or Unplug the Thermometer Cable When Power is ON

myFOCUSERPRO2 Software

1. Arduino Firmware code that runs on the Arduino and can be operated in manual mode by pressing the IN/OUT buttons to move the focuser
   - The focus controller on power up defaults to the last known position 5000. This can be overridden with a specific value when connecting to a controller
   - The 0 position can be set by holding down both IN/OUT buttons for 2 seconds (there will be a beep to confirm – then release both buttons) – or by using the Windows software or ASCOM driver
2. A myFocuserPro2 Windows application that can remotely control the focuser
3. A myFocuserPro2 ASCOM driver that can be used with the controller
   - To run two focus controllers, you need to install the secondary ASCOM driver (see Appendix D)

myFOCUSERPRO2 Recommended Client Applications

As well as free Windows application and ASCOM drivers (which is fully ASCOM compliant), myFocuserPro controllers have been tested and work with the following software applications

FocusMax v4.x
Nebulosity 4.x
APT v3.x

These are reported by users as working. Other applications not listed here either cause issues, do not adhere to correct ASCOM client implementation, have yet to be tested, or have yet to be reported.

MAXINCREMENT

This value is used by an ASCOM client application. The value for maxIncrement specifies the maximum number of steps permitted in any one move. The value for maxIncrement is implemented by the controller manufacturer.

When an ASCOM client connects to a focus controller, it should request the value of maxIncrement from the controller. The client then knows the maximum number of steps it can use when sending a move command to the controller, and whether the target position can be reached via a single move or if multiple moves are required.

For example, assume the current focuser position is 5000 and maxIncrement has a value of 1000. If a move to position 6500 is requested, the client application is meant to accept that this is not possible in a single move, so should first issue a MOVE +1000, and when completed, issue the final MOVE +500 and the controller should then be positioned at position 6500.

The myFocuserPro2 controller will set the value of maxIncrement to 1024 on connection. If a set maxIncrement command is sent to the controller, the controller will update the value for maxIncrement.

The setting for maxIncrement is NOT remembered by the controller.

BOUNDARY RULES FOR maxStep AND maxIncrement

The following table defines the boundary rules for maxStep and maxIncrement as implemented in the Arduino code.

Variable	Minimum Value	Maximum Value
maxStep	1000	v246 and lower = 65000 v247 and above = 2000000000
maxIncrement	1024	maxStep

maxStep is the maximum position of the focuser. **maxIncrement** is the maximum number of steps permitted in a single **move** command. *You will need to determine maxStep for your system*. It is safe to set maxIncrement to the same value as maxStep you determined for your system.

If a **move** or **Set Target Position** command is sent to the focuser which would result in the focuser being less than 0 or greater than maxStep, then the focuser will stop at either 0 or maxStep respectively.

CONNECTING THE STEPPER MOTOR TO THE FOCUSER

There are two ways to connect the stepper motor to the focuser

1. Direct to the focuser shaft (use the single speed knob only) using a flexible coupler
2. Using a pulley-belt or gear type system giving a gear reduction and possible increase in torque

A direct connection has the least noticeable backlash of any method used to connect the stepper motor to the focuser unit and gives the best repeatable results (see Appendix B).

Pulley/gear systems may slip intermittently if the weight is too much and this will result in lost steps and inaccurate movement. These systems also suffer from increased backlash. *They are ideal if you want to control the focus of a DLSR camera lens where direct connection of the stepper motor is not possible* (see Appendix C).

A direction connection requires the use of a mounting bracket (normally L or U shaped) that permits connection of the stepper motor shaft to the focuser shaft via a flexible coupler. The bracket aligns the center of the focuser shaft with the center of the stepper motor shaft and provides space to connect the shafts via a flexible coupler. Slots in the bracket provide the necessary adjustments in order to line the stepper motor correctly as well as attaching it to the focuser body.

Once connected with a flexible coupler, you cannot move the focuser manually. You must remove the Focus Lock screw or keep the screw loose.

The photo shows an L-bracket (bolted to bottom of focuser) that allows the direct connection of the stepper motor to the focuser shaft using a flexible coupler.

The L-bracket which attaches to the base of the focuser holds the stepper motor. The slots provide ample room for aligning the stepper motor with the focuser shaft.

For more information on bracket design and connection methods, please see Appendix B.

Should I Connect the Stepper Motor to the Fine Focus Knob?

NO. There is a misconception that driving via the 10-speed reduction is safe. This is not good idea as the mechanics and manufacture of the mechanisms employed mean that they are not robust enough to have that level of force (from stepper motors) consistently applied to them. There are small gears (normally plastic or cheap metal) involved that do not have the strength to handle the force a stepper motor can apply. Over time this results in increased wear and tear in the small gears, leading to increased backlash, or if the focuser limits are exceeded, the complete breakdown of the fine focus mechanism by catastrophic failure (breakage) of the gears - in other words, not a good idea.

So why do some users try connecting to the fine focus knob? The answer is that it already provides a 10:1 reduction so this increases the resolution without any cost. The downside is the cost in replacing the focuser mechanism when the 10:1 reduction is damaged by the stepper.

If you need to drive the fine focus mechanism (FFM) then it would be better using a belt drive to prevent damage in the event of exceeding the focuser limits. It is not recommended.

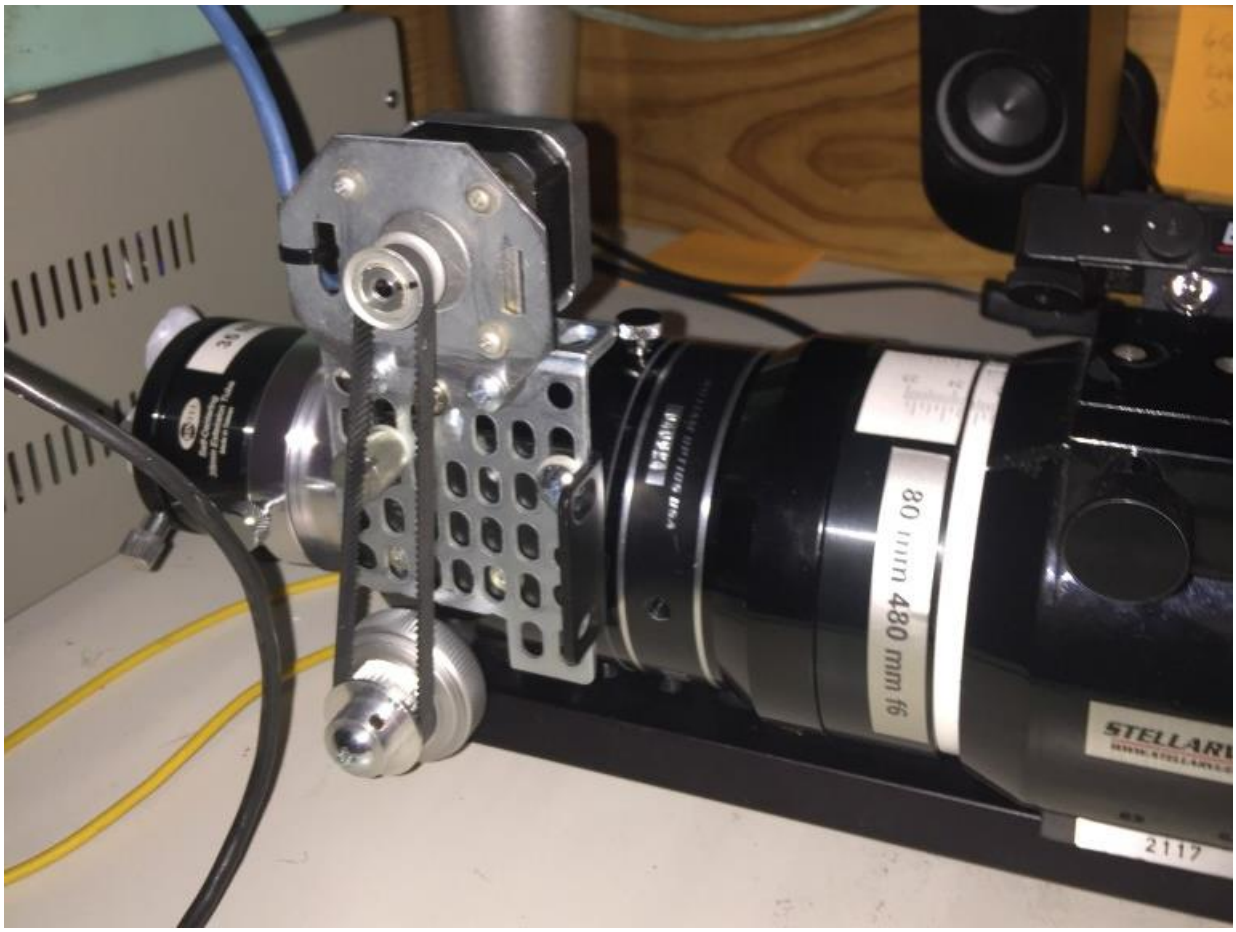
Remember NOT to use the Focus Lock Screw on your focuser; leave the screw loose or remove it. The Stepper motor will hold the focuser in place. If you leave the screw in and accidently tighten the focus lock screw, then serious damage can occur to the stepper motor or focuser.

HOW DO I CONNECT THE STEPPER MOTOR TO THE FOCUSER?

You will need to find a way to connect the stepper motor to the focuser. You will probably need to make a mounting bracket. How you affix the stepper motor will also vary depending upon telescope type (SCT-MAK, Refractor, Newtonian).

The following ZIP file has a lot of user submitted photos showing how they attached their stepper motor to their focuser. You will find many different ways in which this can be done.

https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/Documentation/Mounting%20Examples.zip/download



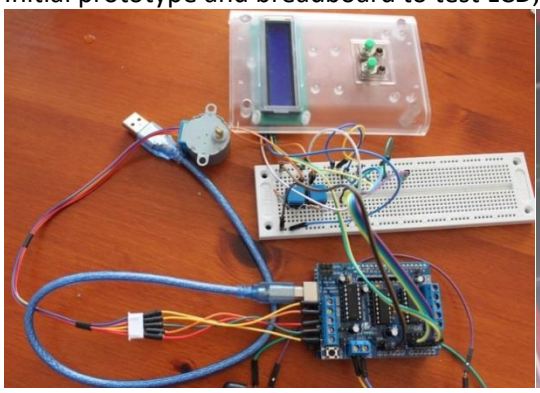
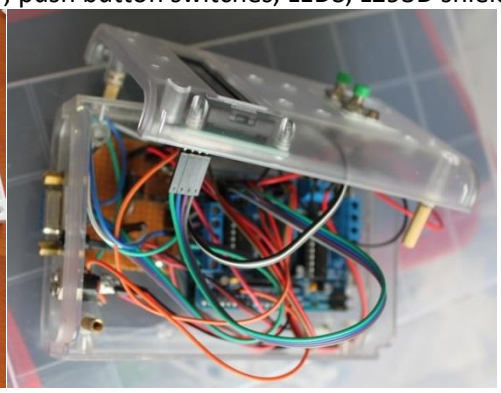
I AM USING GEARS. HOW DO I CALCULATE THE LENGTH OF BELT REQUIRED?

The following spreadsheet is a belt length calculator. If you are using gears, use this calculator to determine the length of the belt you require

https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/Documentation/Belt%20Length%20Calculator.xlsx/download

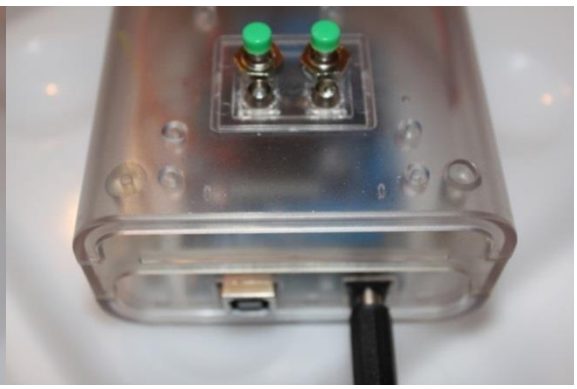
myFOCUSERPRO2 PROTOTYPE BUILD PICTURES

Initial prototype and breadboard to test LCD, push button switches, LEDS, L293D shield and stepper motor

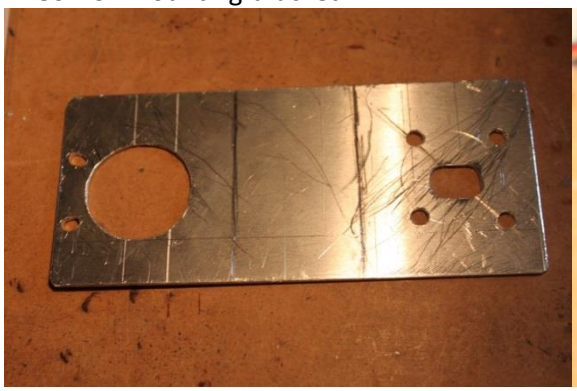
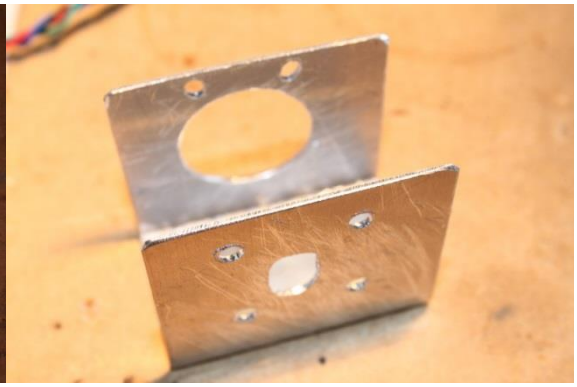



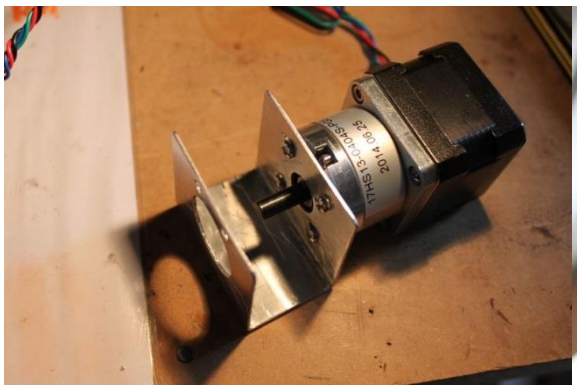
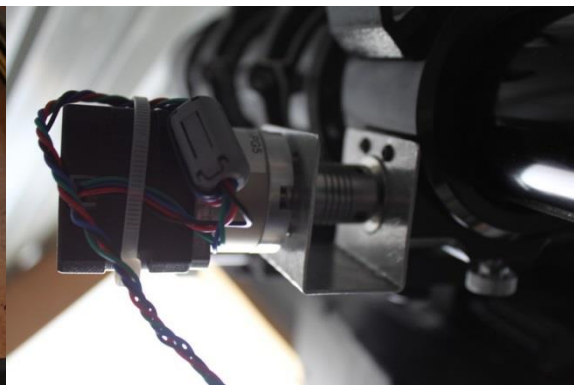
Arduino, L293D shield, LCD, switches enclosed in Link-Sprite case


Top plate with RS232 connector (for stepper motor), Power LED, 12V DC Input jack, Temperature probe input jack, on right photo LEDS for push buttons fitted

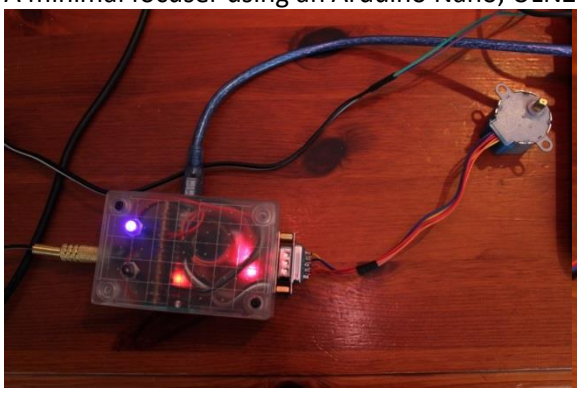
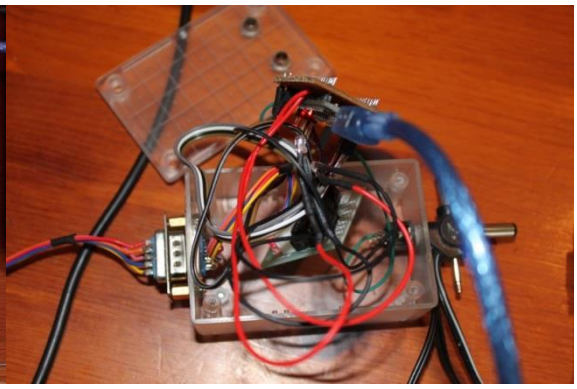



ED80T-CF Mounting bracket

A minimal focuser using an Arduino Nano, ULN2003 driver mounted in a small case

DO NOT MANUALLY MOVE THE FOCUSER ONCE SETUP

Manually moving the focuser position between sessions invalidates the saved focus position in the controller. For example, a user uses the Windows application to set the focus position as 4605. That position is saved by the controller as the last known position when the application is closed.

A few days later, the user turns the focus knob manually by hand half a turn. Then the user starts the application software, which defaults to the last known position of 4605, which is now invalid (not the same) because the physical position was altered.

DO NOT CHANGE STEP MODE SETTING ONCE SETUP

The positions of 0 and maxStep, once set, are related to the step mode in use at the time. If you have done all your calculations for step size and CFZ then you will know the step mode to use during initial setup of the focuser controller.

Once you have determined your step mode setting – do not change it.

Consider the case where the focuser has been set up as 0 to 6000 maxStep and the focuser is currently at position 4000. The step mode is Half steps. Using some maths, this means there are 2000 half-steps available before the maximum position is reached (or in terms of full-steps, 1000).

The user decides to change the step mode to Full steps and then issues a Goto to focuser position 5500. This equates to 1500 full-steps from the current position of 4000. So the focuser will attempt to drive to position 5500 using full steps (doing some maths that is 3000 half-steps or a final real position of 7000). This could damage the focuser by driving beyond the maximum safe position.

INITIAL SETUP FOR ALL DRIVER BOARDS

By now you should have calculated

- **StepSize** in microns
- **Critical Focus Zone**
- **Stepping mode** required to get about 10 stepper motor steps within the critical focus zone
- **maxStep** being the maximum focuser position

In operating the focuser, you will need to determine the correct settings for **maxStep** that matches your focuser and type of stepper motor being used.

Once you have determined the right step mode setting to use, do NOT change it. If you decide to use full steps then perform the initial setup using full steps. If you decide later to change to half steps, you will need to perform the initial setup again. It is **NOT** recommended to change step mode during an observing or imaging session.

CHANGING STEP MODE INVALIDATES THE FOCUSER POSITION. YOU WILL NEED TO DO THE INITIAL SETUP AGAIN IF YOU CHANGE THE STEP MODE.

Remember that if using half steps or a gear drive pulley belt system, the number of steps can be quite large.

However, using a stepper motor at full steps which has 100 steps per revolution and is direct connected means that that you cannot use a maxStep size of 32000 as this is 320 full turns of the focuser knob, and would result in damage to either the stepper motor or the focuser.

What you will do as part of the initial setup is connect the focuser to the computer via a USB cable, run the Windows application, and ensure that the stepmode, step-size and maxStep values are entered and sent to the controller.

Within the Windows application, the parameters for Step-Size and maxStep (maximum permissible) are set on the Extra Settings menu and must be set before connecting to the controller.

1: ENSURE THAT THE FOCUSER HARDWARE IS WORKING CORRECTLY AND THE STEPPER MOTOR IS MOVING

1-1 FIRMWARE CHANGE: LCD ISSUES

If you have LCD issues, look here. You might also need to make changes if you are using an I2C LCD1602. Please see the PDF document at this location for more information.

2: SERIAL PORT BAUD RATE

If you want to change the baud rate to a higher value, then it must first be changed in the firmware file and then the controller reprogrammed. Find the following lines in the firmware file applicable to your controller and then set the correct value for **SerialPortSpeed** (default value is 9600). Once changed, upload the modified firmware to your controller.

// define serial port speed - valid values are **9600** 19200 38400 **57600** 115200 230400
#define SerialPortSpeed 9600

For example, if a speed of 19200 was desired, then the change would look like

// define serial port speed - valid values are 9600 19200 38400 57600 115200 230400
#define SerialPortSpeed 19200

Remember that if you change this value, you then must use the same baud rate setting in the ASCOM driver and the Windows application to communicate with the controller.

3: USING A FULL BOARD WITH NO PUSH BUTTONS?

3-1 Disable Pushbutton code

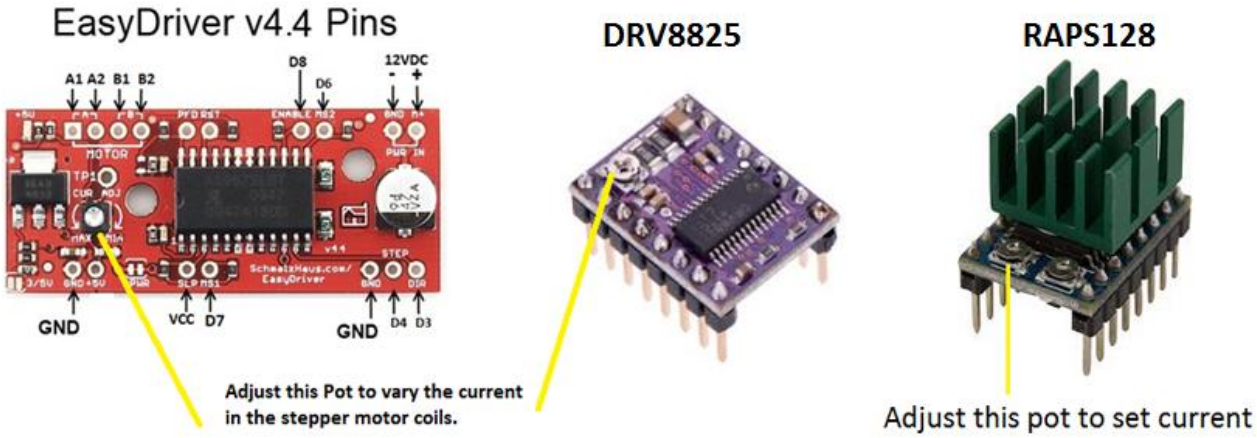
If you are using a FULL version but did **NOT** wire the Push buttons, the code for the push buttons needs to be disabled. Look for the following lines

// To enable the Push Buttons for manual focusing, uncomment the next line
//#define PUSHBUTTONS 1

and make sure that the #define is commented out (preceded with //)

4: ADJUSTING THE POT ON THE DRIVER BOARD FOR CURRENT MAXIMUM

This only applies to DRV8825, EasyDriver and RAPS128 driver boards. You will need to adjust the POT on the DRV8825/EASYDRIVER/RAPS128 board to get optimal stepping of the stepper motor. This pot adjusts the current that flows in the coils of the stepper motor.



Adjusting the Stepper Motor manually

It is best to use a ceramic or plastic screwdriver when adjusting the pot. I would suggest a plastic knitting needle which has the end filed down to look like a screwdriver.

1. With the controller connected via USB, and 12V power to the driver board, set the focuser position to 0 and the Motor Speed to SLOW
2. Set the step mode to what you have calculated as needed for your system
3. Enter a focuser position of 5000 and click the Goto button
4. Wind the pot all the way anticlockwise until the motor stops moving
5. Now very slowly turn the pot clockwise until you see the motor start to turn. If the maxSteps is reached, just reset the focuser position to 0 and then type in 5000 for the position and click Goto again
6. Slowly turning the pot, when you see the stepper start to move ok without jerking, then slowly turn no more than 1/8th clockwise from that point

7. It should now be close enough
8. If you go too far then there will be too much current and the motor will run hot. You should use no more than 12V external power

On some driver-boards clockwise might be anticlockwise. Once current is set, the stepper motor should run smoothly without missing steps (a missed step will be a sudden jerk which you will be able to feel or hear). If there is any of this, you might need to ever so slightly turn it a little more. Be careful as a little turn can make significant changes in current.

5: FIRMWARE CHANGE FOR L293D, L293DMini and L298N and DRIVER BOARDS

When using NEMA17 stepper motors with the L293D Motor Shield, L293D-Mini and L298N driver boards the number of steps per revolution must be specified in the Arduino firmware file.

Find and change the line in GREEN below

```
#define stepsPerRevolution 200      // NEMA17 motor Full steps
// you need to change the above line to reflect your stepper motor, examples below
// #define stepsPerRevolution 2048      // 24BBYJ-48 motor, if half stepping multiply by 2
// #define stepsPerRevolution 1036      // NEMA17-PG5 motor, if half stepping multiply by 2
// #define stepsPerRevolution 200       // NEMA17 motor, if half stepping multiply by 2
// #define stepsPerRevolution 5370      // NEMA17-PG25 motor, if half stepping multiply by 2
```

to reflect the number of steps for your stepper motor. For instance, if you are using the PG25 Nema17 stepper motor at half steps, the changed line (shown in RED) would be

```
#define stepsPerRevolution 10740      // NEMA17-PGG25 motor half steps
// you need to change the above line to reflect your stepper motor, examples below
// #define stepsPerRevolution 2048      // 24BBYJ-48 motor, if half stepping multiply by 2
// #define stepsPerRevolution 1036      // NEMA17-PG5 motor, if half stepping multiply by 2
// #define stepsPerRevolution 200       // NEMA17 motor, if half stepping multiply by 2
// #define stepsPerRevolution 5370      // NEMA17-PG25 motor, if half stepping multiply by 2
```

Once changed, upload the modified firmware to your controller.

6: FIRMWARE CHANGE: L293D Motor Shield Driver Board

The L293D Motor Shield driver board supports 4 motor connectors labelled M1 to M4. If you want to use a different motor port on the L293D shield, you will need to change the firmware file.

Find and change the line below IN GREEN to reflect the motor port you are using

```
// Stepper Motor stuff - YOU NEED TO USE THE CORRECT ONES FOR YOUR STEPPER MOTOR
// Motor port on the L293D shield to use
#define Motor_Port 2     // use M3 and M4 as its easier to connect
// you need to change the above line to reflect which port you are using on the L293D shield
// it is either 1 (M2/M1) or 2 (M3/M4)
```

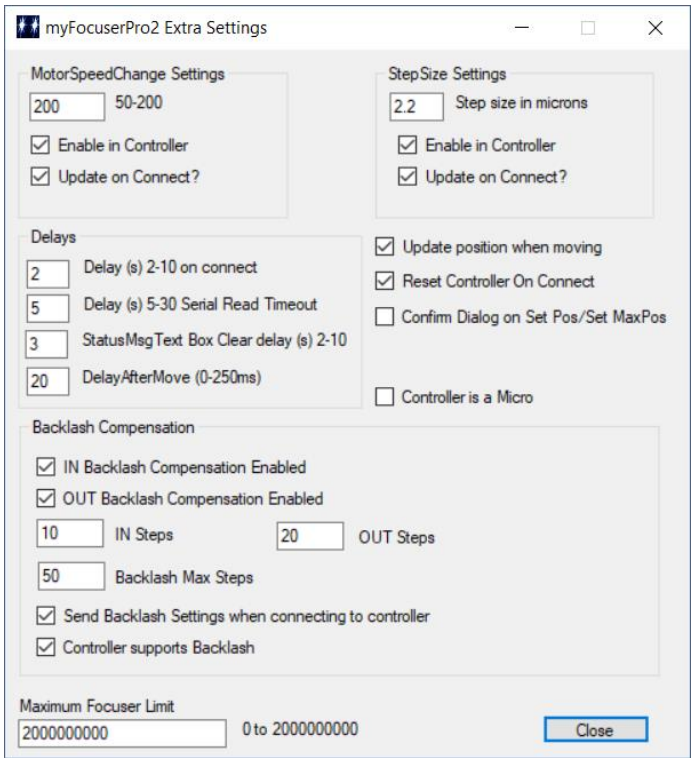
For example, if you decided to use Motor Port 1 because that made wiring easier, you would change the lines to as shown below (in RED)

```
// Stepper Motor stuff - YOU NEED TO USE THE CORRECT ONES FOR YOUR STEPPER MOTOR
// Motor port on the L293D shield to use
#define Motor_Port 1     // use M1 or M2
// you need to change the above line to reflect which port you are using on the L293D shield
// it is either 1 (M2/M1) or 2 (M3/M4)
```

7: Set StepSize, StepMode and Maximum Permissible value for maxStep

a) With the stepper motor disconnected, move the focuser to the in-most position as position 0 (suggest you position the focuser at ½ turn outwards and make that 0)
b) Next clamp the stepper motor in place and engage the stepper with the focuser shaft (tighten screws on clamp etc)
c) Start the windows application BUT do NOT connect to the focuser yet

Before connecting to the controller, access the menu bar of the Windows application and select “Extra Settings”. This displays the following form (note that your values can be different).



7-1: Initial Setup of StepSize

Enter your calculated stepsize in the Step size box, and ensure that Enable in Controller is set to YES and Update on Connect is set to YES.

7-2: Initial Setup of Maximum Permissible steps

Enter your calculated maxSteps value into the Maximum Focuser Limit box. Click Close to save this setting in the Windows application. This feature is only available under Windows Application version 2330 and higher and Arduino firmware v247 or higher. In all cases, this should be the same as the maxStep value for your focuser.

7-3: Initial Setup of StepMode

Make sure you have done steps 1 and 2. Stepmode is on the main form. It is activated when you connect to your controller. First, select the correct COM Port and Port Speed then click the Connect button. After a few seconds the values returned from the controller will be displayed on the main form. To set the stepmode, simply click on the desired step mode setting, then Click the Send StepMode button to send the setting to the controller.

The maximum value of step mode is 128. However, only the RAPS128 and ST6128 driver boards support this level. Each driver board type has its own limit for micro-stepping. If you try to send an invalid step-mode to any driver-board it is ignored by the controller firmware. For instance, sending a step-mode of 64

to a DRV8825 controller will cause the controller to ignore the request and the controller will use the default of half-steps.

THE FOLLOWING ONLY APPLIES TO ALL EASYDRIVER/DRV8825/RAPS128/A4988 FIRMWARE CODE
If you are using a stepmode which is $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$ or greater then you will need to have the diodes in place else the stepper motor may have trouble moving smoothly at the selected step mode.

7-4: Initial Setup of maxSteps
Make sure you have done the previous steps 1-3. To set the maxStep setting, enter the maxStep value and then click the set button to the right of the maxStep text box. The value will be sent to the controller and remembered for later use.

8: CHECK THE SPEED SETTINGS
The next step is to check the speed settings of Slow, Medium and Fast.

a) Enter 0 and the focuser position and click Set Position
b) Enter 5000 as maxSteps and click Set
c) Set the stepmode to what you have calculated as best for your focuser
d) Select the SLOW motor speed from the menu
e) Click the +500 button and check the motor moves smoothly
f) Repeat for the MEDIUM and FAST motor speeds
g) If the motor does not run smoothly then adjustment of the speed settings is required

8-1 Motor Speed Settings
THE FOLLOWING APPLIES TO ALL FIRMWARE CODE
In v252 and above the speed of the motor is now controlled in a different manner. The following lines of code should NOT BE CHANGED. If you are having a speed issue please contact the author.

```
// do not change these values
#define   motorSpeedSlowDelay 5000      // DO NOT CHANGE
#define   motorSpeedMedDelay  3000      // DO NOT CHANGE
#define   motorSpeedFastDelay 1500      // DO NOT CHANGE
```

9: Testing Direction
Now that the focuser has the correct values, you can test the direction setting to ensure the focuser is moving in the correct direction.

This assumes that you have set the stepper current correctly if using a DRV8825/EasyDriver/RAPS128 or A4998 driver board.

a) In the focuser position text box, enter 0 as the current focuser position and click the SET POSITION button to send the position to the controller
b) Ensure that external power is ON to the stepper motor. The focuser is currently at position 0
c) Click the +100 button
d) If the focuser does not move at all, then click the Reverse Direction button to enable that setting, then click the +100 button again
e) The focuser should move 100 steps outwards
f) Clicking any + button should move the focuser outwards and any – button the focuser should move inwards towards 0

10: Set Coil Power

If you are using microstepping then Coil Power should be left ON. This is because with micro-stepping the stepper motor can only hold its position if current is flowing in the stepper motor coils. If coil power is OFF, then the stepper will move to the closest full step, and over time this results in the real focuser position not being accurate.

The controller will remember stepsize, maxSteps, stepmode, coil power, reverse direction and focuser position.

You can also set other defaults such as Motor Speed, LCD Display Time and other settings at this time.

FROM THIS POINT ON, DO NOT CHANGE THE STEP MODE OR ENTER A NEW FOCUSER POSITION AND CLICK SET POSITION AS THIS WILL ALTER THE CONFIGURATION OF THE STEPPER AND RESULT IN LOSS OF ACCURACY OF POSITION AND ALSO POSSIBLE DAMAGE TO THE FOCUSER OR STEPPER MOTOR.

CONGRATULATIONS: Your focuser is now ready to use!

myFOCUSERPRO2 FIRMWARE SETTINGS

These settings are stored in the controller EEPROM (may vary depending upon minimal, minimal plus temperature probe and full versions), remembered from session to session, and can be configured using the Windows Application as part of the Initial Setup of the myFocuserPro2 controller.

Initial Focuser Position	*// last focuser position*
Maximum Focuser Position	*// max steps*
Step mode	*// indicates stepmode, full,* $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$
Reverse Direction	
Coil Power	
Tempmode	*// temperature display mode, Celsius=1, Fahrenheit=0*
updatedisplayintervalNotMoving	*// refresh rate of display - time each page is displayed for*
stepsizeenabled	
StepSize	*// in microns*
lcdupdateonmove	*// display the position on the LCD during a Move*
ds18b20resolution	*// resolution of DS18B20 temperature probe*
tempcompenabled	*// indicates if temperature compensation is enabled*
tempcoefficient	*// steps per degree temperature coefficient value*
delayaftermove	
backlash_in_enabled	
backlash_out_enabled	
backlash_steps_in	
backlash_steps_out	
ControllerIsMicro	
EEPROMWrites	

These values are retrieved from the controller when connecting using the Windows Application or ASCOM driver. Some of these settings can be updated when connecting to the myFocuserPro2 controller.

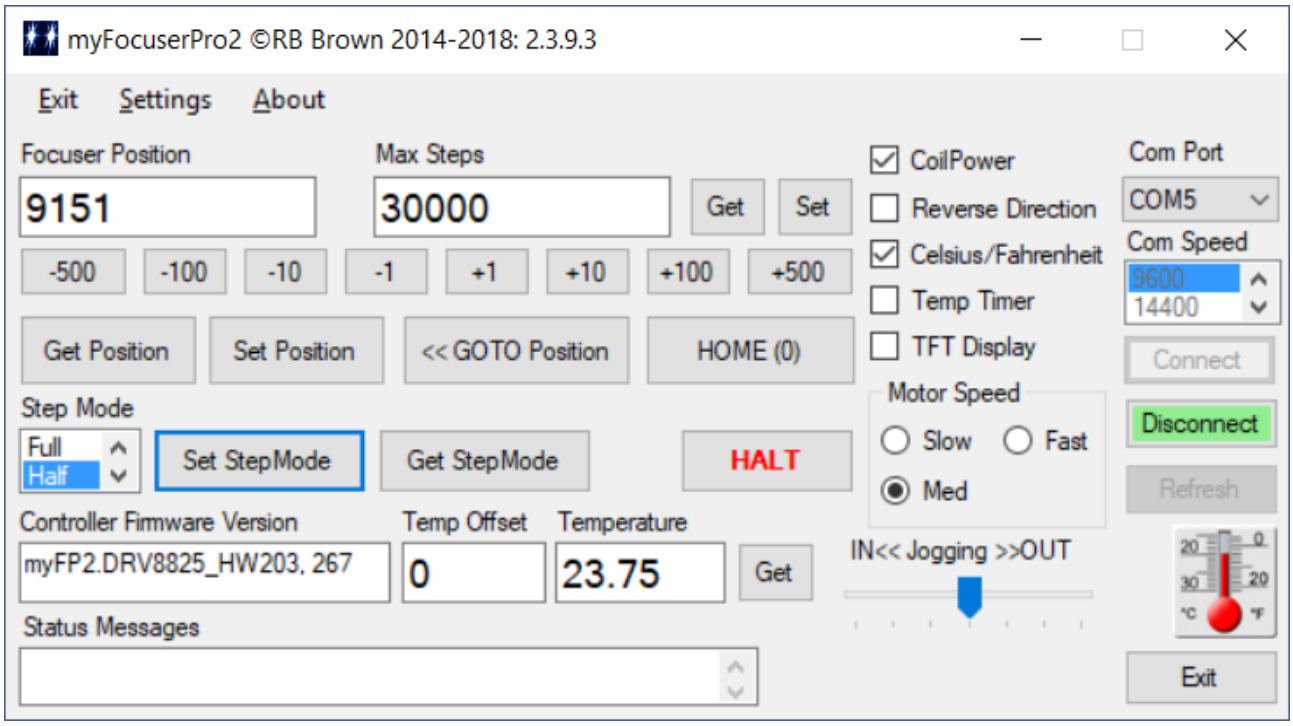
Note: Stepper Coil power means that at the end of the move, when the stepper motor is stationary, power is either OFF or ON to the stepper motor coils. If OFF, this saves power, but it might mean that a heavy focuser might start to slip if pointed towards zenith. To prevent this set the Stepper Coil Power to ON. The ON setting consumes power when the stepper motor is not moving. Some stepper motors might get hot in operation if this setting is ON. You will need to check your stepper motor if this is happening. The recommended PG27 stepper motor is fine with Coil Power ON. If using micro-stepping (any stepping mode other than FULL steps) then Coil Power must be set to ON.

Note: Remember NOT to use the Focus Lock Screw on your focuser; leave the screw loose or remove it. The Stepper motor will hold the focuser in place. If you leave the screw in and accidently tighten the focus lock screw, then serious damage can occur to the stepper motor or focuser.

Note: Use 7.5-9VDC with the 28BYJ-48 stepper motor if using Coil Power ON otherwise the stepper motor may get very hot

myFOCUSERPRO2 WINDOWS APPLICATION

The myFocuserPro2 application gives full control of the focuser (supports ALL build options). Note that the majority of settings are remembered/saved by the application or within the controller.



The Menu provides access to Exit (quit the program), Settings and About (Copyright message).

myFocuserPro2 Main Window Buttons

The main screen is reasonably self-explanatory.

- ☑ **Coil Power** When ON coil power is ON and the stepper coils remain powered after the move is completed
- ☑ **Reverse Direction** When ON the motor moves in the opposite direction (IN means OUT and OUT means IN)
- ☑ **TFT Display** When ON allows a 10 second delay on Connect to allow the controller to initialise
- ☑ **Celsius/Fahrenheit** When C displays temperature value in Celsius
- ☑ **Temp Timer** When ON the application will periodically poll the myFocuserPro2 controller and request a temperature and position update. The polling interval is set under the settings menu

Goto Position - type in the desired focus position in the Focuser Position text box (digits only, < maxStep, 0 or > 0, then click the << Goto Position button. The focuser will move to the specified position.

Get Position - returns the current myFocuserPro2 controller position in the Focuser Position text box.

Set Position - type in the desired focus position in the Focuser Position text box (digits only, less than Maximum Position, 0 or greater than 0, then click the Set Position Button. The focuser will NOT move but the position is updated.

HOME - The focuser will move to position 0 and stop.

Get Maximum Position - returns the current myFocuserPro2 maxStep value in the Maximum Position text box.

Set Maximum Position - type in the desired maxStep value in the Maximum Position text box (digits only, greater than FocuserPosition, then click the Set Button.

HALT - Will halt the focuser if currently moving.

CLEAR - Clears the TX and RX Text boxes.

TX - Text box used to indicate status messages and Transmit commands.
RX - Text box used to indicate status messages and responses from myFocuserPro2 controllers.

COM Port - Use the dropdown list to select the correct comport that the myFocuserPro2 is connected to
Com Port Speed - Select the baud-rate that matches the firmware in the controller (default 9600)

Connect - After selecting the correct comport, click Connect to connect to the myFocuserPro2 controller.

Disconnect - disconnects the myFocuserPro2 controller.

Refresh - refreshes the list of available comports.

Get Firmware Version - Displays the current firmware version from the myFocuserPro2 controller.

EXIT - Exits the application.

Temp Offset - The Temp offset entry-box provides a mechanism for adjusting the returned temperature value for calibration. Adjustment values range from –3 to +3. For example, typing -1.5 into the entry-box will subtract 1.5 degrees C from the returned temperature value. A comma or decimal point can be used to signify the decimal point. To set the temperature offset, click inside the entry-box and type the desired value (for example -1,32) and then press Enter. Once the Enter key is pressed, the entered value is validated (rounded to 2 decimal places and bound checked at -3 and +3) and shown corrected in the entry-box (using a decimal separator of a decimal point).

Get Temperature - requests the temperature from the myFocuserPro2 controller (if a temperature probe is supported and attached) and display the values in the RX text box, and the adjusted value in the Temperature Text Box.

Set StepMode - To set the StepMode, select the desired stepping mode from the dropdown list, then click the Set StepMode button. This should be set only ONCE during the initial setup and after than remain unchanged. Changing stepmode during a session invalidates the focuser position.

Get StepMode - Click the Get StepMode to retrieve the current stepping mode from the myFocuserPro2 controller. This will update the value shown in the dropdown list.

-500 to 500 - These buttons provide a means to move the myFocuserPro2 controller by a specified number of steps. For example, clicking +10 will move the focuser +10 steps.

Jogging - This control lets the user move the focuser under jog control. Dragging the control to the left moves the focuser IN whilst dragging the control to the right moves the focuser OUT. There are three position stops to the left and right of 0 (midpoint). Position 1 drives the focuser at Slow speed, Position 2 at Medium speed, and Position 3 at Fast speed. Jogging continues as long as the mouse is held down or the control is NOT at the midpoint of the control. When the mouse is released, the control is set to 0 and de-activated, jogging stops and the current focuser position is retrieved from the controller. Jogging also stops if the HALT button is clicked. A single click on the control stops the focuser moving by sending a HALT command to the controller.

MotorSpeed - This controls the relative speed of the stepper motor. The speed can only be changed when the focuser is NOT moving. Whatever the speed setting, this is sent to the controller when connecting.

Notes:

1. Microstepping (StepMode) is ignored when using the ULN2003 or L298N build options.
2. Focuser Position, Maximum Position (maxStep), Coil Power, Reverse Direction and StepMode are updated from the controller when the application software connects to the controller.

Temperature Display

To display a running graph in real time of the temperature reading from the focuser, click on the Temperature Display icon.

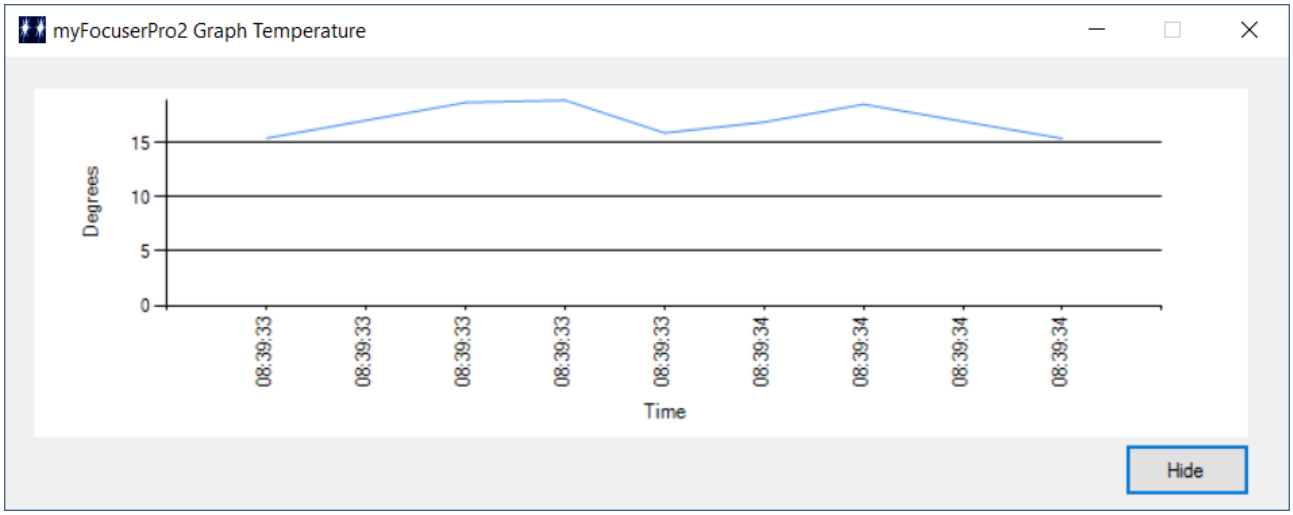
A Graph window will appear. Each time a temperature request is send to the controller, the returned temperature value and current time will be plotted on the graph. After nine plotted points the next plotted point will cause the graph to scroll to the left, with the 9 most recent values being displayed.

The Hide button will hide the graph window. The Window position of the graph is remembered.

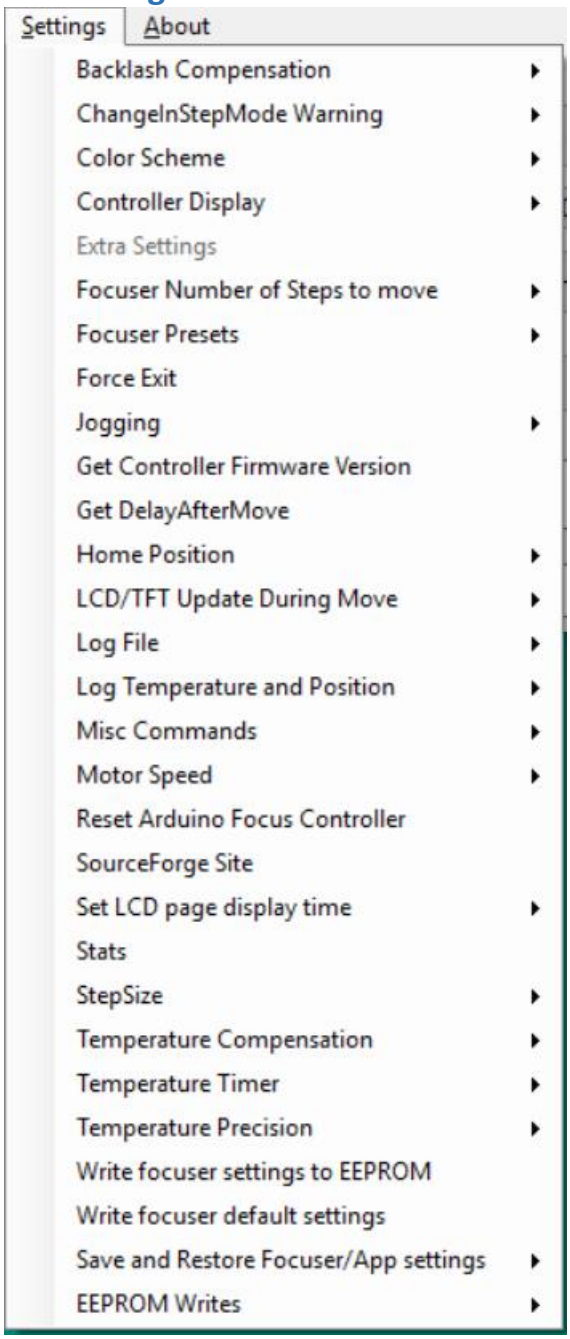
Temperature values will be plotted whenever a temperature request is made. This can be done in two ways

1. Whenever the GET Temp button is clicked
2. Whenever Temperature logging is enabled

The recommended way is to enable Temperature logging from the menu, and select the desired refresh rate. The graph will then reflect every request and display the values in real-time whilst temperature logging is enabled.



The Settings Menu



Backlash Compensation - A set of options to query the backlash settings or send settings to the controller

ChangeInStepModeWarning – Enabled or disabled. When enabled, it warns the user about changing the StepMode when the focuser is connected.

Colour Scheme - The colour scheme option allows the user to specify two defined colour schemes (default and one defined), or customize their own colour scheme by selecting the custom option. The colour setting is saved and reloaded when the application restarts.

Controller Display - The LCD display on the myFocuserPro2 controller can be disabled or enabled via this setting.

Extra Settings - Displays the extrasettings form.

Focuser Number of Steps to move - When "Settings>Double Step Size" is selected from the menu bar, the step button values are doubled (-500 becomes -1000). The double step size setting is NOT remembered by the application program.

Focuser Presets - This allows you to specify up to FOUR preset focuser positions and move the focuser to any of the four preset positions. These presets are saved by the application.

Force Exit - in the event of problems, this provides a clean method of exiting the program

Jogging - Get the jogging status and the last jogging direction

Get Controller Firmware Version - Get the firmware version of the connected myFocuserPro2 controller

Get DelayAFterMove - The default value is specified on the Extra Settings form page. This value specifies the delay in milliseconds (0-250) that is applied after move to prevent any blurring of star caused by any mechanical residual movement or vibration of the stepper or focuser during focusing.

Home Position - Get status of home position switch

LCD/TFT Update During Move - This option is used to enable/disable the update of the focuser position on the LCD during a move, and can also return the status (enabled/disabled)

Log File

- Enabled, turns error logging ON
- Disabled, turns error logging OFF
- Reset Error Log File path displays a dialog box to specify the drive and directory where the error log file and other log files are to be stored.

Log Temperature and Position - Save the temperature and focuser position to a log file in automate mode.

Motor Speed - Covered next.

Reset Arduino Focus Controller - This will restart the Arduino focus controller. Please wait 3-5s before issuing any command.

SourceForge Site - This will open a web browser and display the myFocuserPro2 website where you can download the latest drivers, software and documentation.

Set LCD page display time - The time in seconds that an LCD screen is displayed for can be adjusted using this menu option, from 2s to 4s delay.

Stats - Displays a message box of all the stats for the current session.

Step Size - This menu has two options – "Check if StepSize is enabled in the controller" queries the controller and displays if StepSize is enabled, and "Get controller StepSize" displays the current controller setting for StepSize.

Temperature Compensation - Please read the section on temperature compensation.

Temperature Timer - This setting controls the polling interval at which the application will request a temperature update from the myFocuserPro2 controller. If another command is already in progress when a temperature request occurs, then that temperature request will be ignored.

Temperature Precision - This setting allows the user to control the resolution of the DS18B20 temperature probe from 0.5 degrees down to 0.0625 degrees. This setting is remembered by the controller.

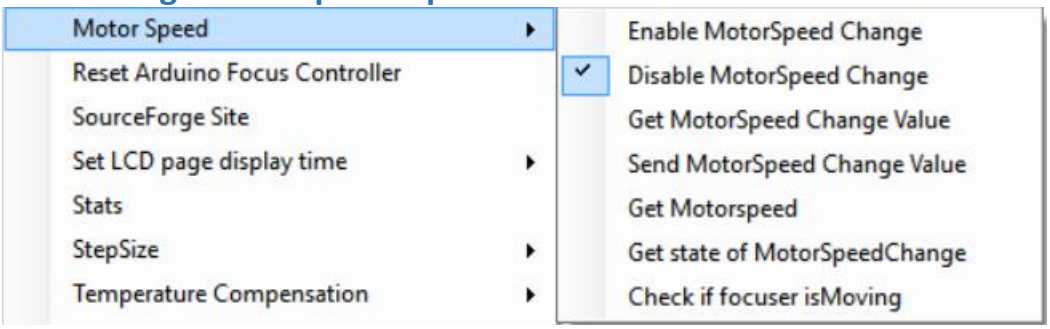
Write Focuser Settings to EEPROM - save the controller settings in EEPROM.

Write focuser default settings - used to write a standard configuration setting to the myFocuserPro2 controller. The default configuration is specified within the Arduino firmware file. This may be useful after upgrading the firmware file to a newer version. For more information, click here.

Save and Restore Focuser/App settings - this option provides for the saving and restoring of focuser and application settings. This is designed to be used before and after upgrading the Windows application and myFocuserPro2 controller firmware.

EEPROM Writes - Get the number of EEPROM writes, or reset value to 0

The Settings Motorspeed Options



Enable MotorSpeedChange - Enables the slowing down of the focuser speed when nearing the target position. The value at which the threshold changes between the current speed and slow speed is adjustable (50-200) on the extrasettings form and must be set before connecting to the focuser (default is 50 steps).

Disable MotorSpeedChange - Disables the change in motorspeed when nearing the target position.

Get MotorSpeed Change Value - Gets the current threshold setting (in steps) for slowing down when approaching the target position.

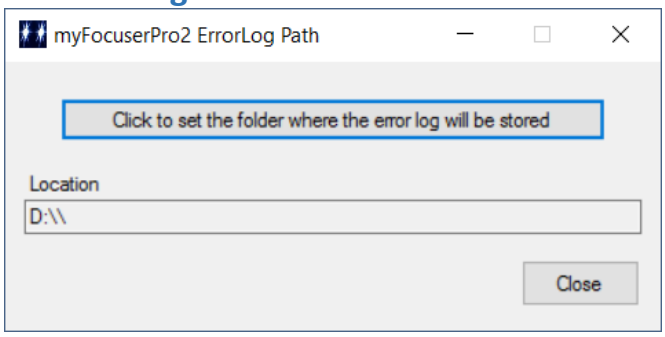
Send MotorSpeed Change Value - Sends the new value for the motorspeed change threshold setting (found on the extra settings form).

Get Motorspeed - Gets the current focuser speed setting.

Get state of MotorSpeedChange - Queries the controller for whether motorspeedchange is enabled or disabled.

Check if focuser isMoving - Gets the moving status of the controller

The ErrorLogPathName Form

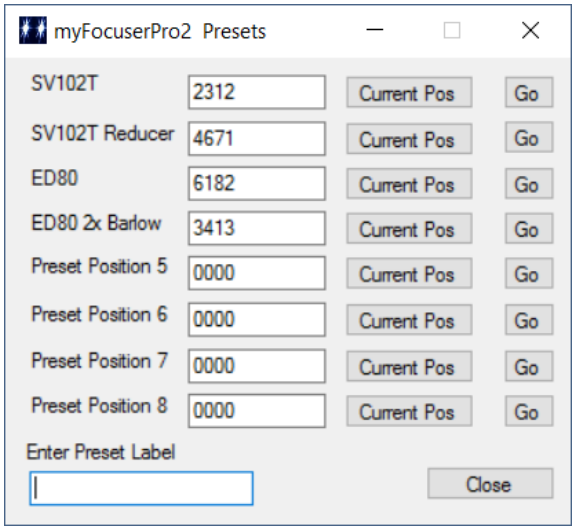


You can specify the directory/folder where the error log file and other settings files will be stored. The path is remembered by the application. This dialog box is accessed from the menu bar Settings>Log File>Reset Error Log File Path

When the application is first installed, this path is set to NULL, so that when the application is run for the first time, this dialog box appears at start-up for the first time.

The Focuser Presets Settings Form

Selecting the “Enter Presets” option from the Focuser Presets menu under the Settings displays the following form



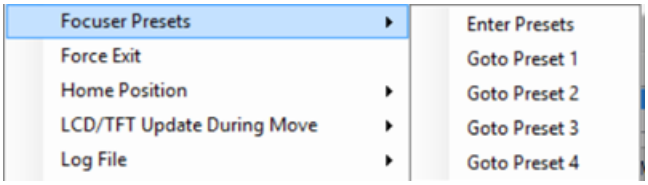
You can enter up to 4 preset focuser positions. The values are only checked when a focuser move is initiated (if the value is wrong then a message will be displayed in the RX textbox and the move cancelled).

Typical values must be greater than 0 and less than maxStep. To copy the current focuser position to a preset, click the associated **Current Pos** button.

Once you have entered the preset values, click the **Close** button to close the form.

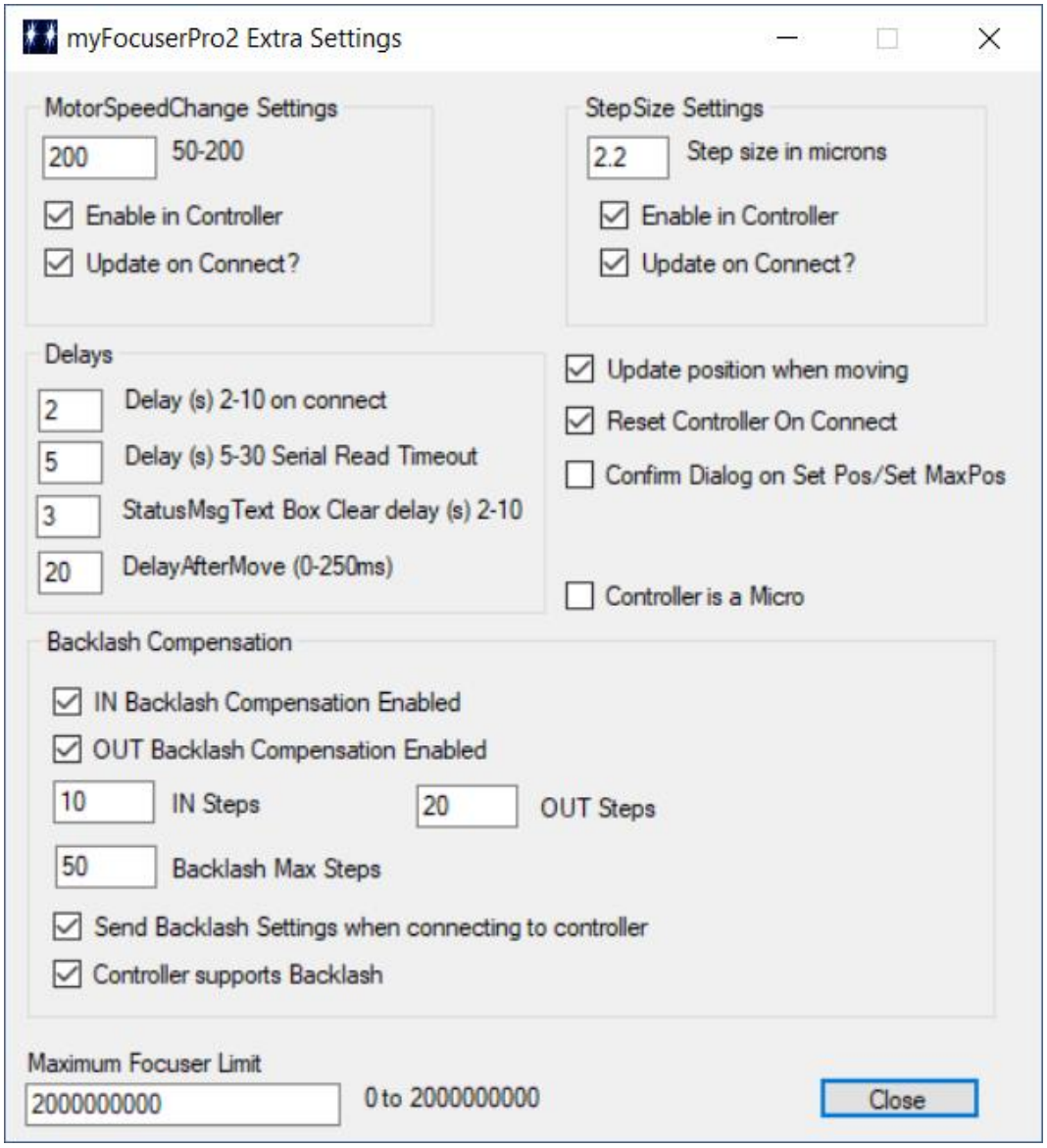
To change a label, first type the new text into the Enter Preset Label textbox (limit of 17 characters) then click on an existing label to replace that labels text. Labels are remembered by the application.

To command the focuser to move to a preset value (assuming one has been set), use the Settings>Focuser Presets menu, as shown, and clicking on one of the presets



The Extra Settings Form

Selecting the “ExtraSettings” option (only enabled when the application is NOT connected to a myFocuserPro2 controller) under the Settings menu displays the following form



Delay Settings Options

Delay (s) 2-10 on Connect - specify the delay in seconds after connecting that the driver will wait before sending a command to the myFocuserPro2 controller. Valid values are 2-10 seconds.

Delay (s) 5-30 timeout on Serial Read - specify the delay in seconds that the application will wait when attempting to read from the serial port after sending a command to the myFocuserPro2 controller (default = 5). For Bluetooth or slower devices, a value of 8 or 10 may suffice.

DelayAfterMove - specifies the delay in milliseconds (0-250) that is applied after move to prevent any blurring of star caused by any mechanical residual movement or vibration of the stepper or focuser during focusing.

Clear Rx/Tx Text Box after delay - after writing text to the Rx/Tx status boxes this delay specifies the number of seconds that will elapse before the text box is automatically cleared.

StepSize Settings Options

Step Size in microns - this setting defines the StepSize in microns. You can enter your measured / calculated value for your focuser and this will be sent to the myFocuserPro2 controller when connecting. This allows client applications using the ASCOM driver to retrieve the StepSize setting from the myFocuserPro2 controller. Please see the section in this PDF on determining your focuser step size value.

Enable in controller? - this setting turns ON or OFF the reporting of the StepSize value from the myFocuserPro2 controller. If you do not know your step size value, DO NOT enable this feature.

Update on Connect - enable the checkbox to send these values to the myFocuserPro2 controller on Connect.

MotorSpeedChange Settings

MotorSpeedChange Threshold Value (50-200) - this setting defines the number of steps that the focuser will switch to slowspeed when approaching a target position. Valid ranges are 50-200 steps. The original motor speed setting is remembered and restored once the target position is reached. This setting can be disabled on the Settings>Motor Speed menu.

Enable in controller? - this setting disables or enabled the MotorSpeedChange feature in the myFocuserPro2 controller.

Update on Connect - enable the checkbox to send these values to the myFocuserPro2 controller on Connect.

Maximum Focuser Limit

This setting specifies the maximum number of steps for the focus controller and will vary depending upon each user's configuration. You should set this to the maximum number of steps that you have calculated for your focuser [see determining maxStep]. This setting must be specified before connecting to the controller, and will be remembered by the application. See the section on the initial setup of the focuser for further information.

Update position when moving

This setting, when checked, will update the focuser position on the main form when a long move is sent to the controller. The focuser position will be updated once per second. On some builds, this might cause the focuser to slightly pause every second, so the default value is unchecked.

Reset Controller on Connect

This setting, when checked, will reset the Arduino controller when connecting. The recommended setting in checked so that the Controller starts with a known state.

Confirm Dialog on Set Pos/Set MaxPos

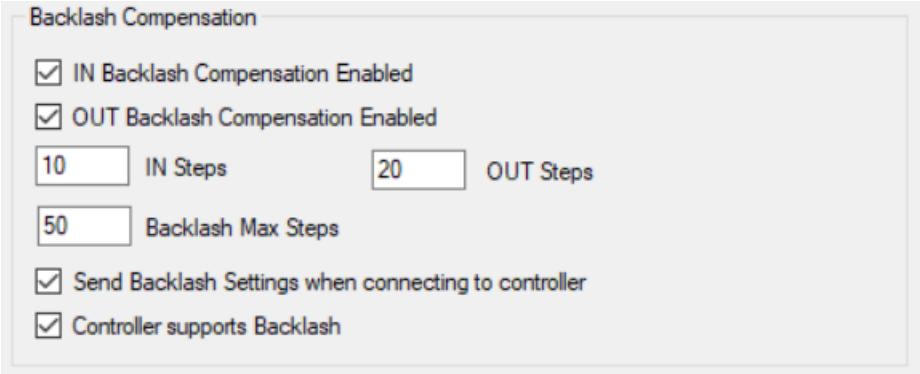
This setting, when checked, will display a confirm (Yes/No/Cancel) dialog box when the user clicks on the Set Focuser Position and Set Max Position buttons. Click Yes to confirm and update the value, or click No/Cancel to abort the change and restore the previous setting.

Controller is a Micro

Check this box if the Arduino controller is a Micro. For all other controllers, leave unchecked. *This also requires special changes to the Arduino firmware file.*

The above options MUST BE SET before connecting to the focuser. The settings will be remembered by the application and next time the application is run, those settings will be restored.

BACKLASH



IN Backlash Compensation Enabled - Enable or disable backlash compensation of IN moves

OUT Backlash Compensation Enabled - Enable or disable backlash compensation of OUT moves

IN Steps - The number of steps of backlash to apply after the IN move has completed

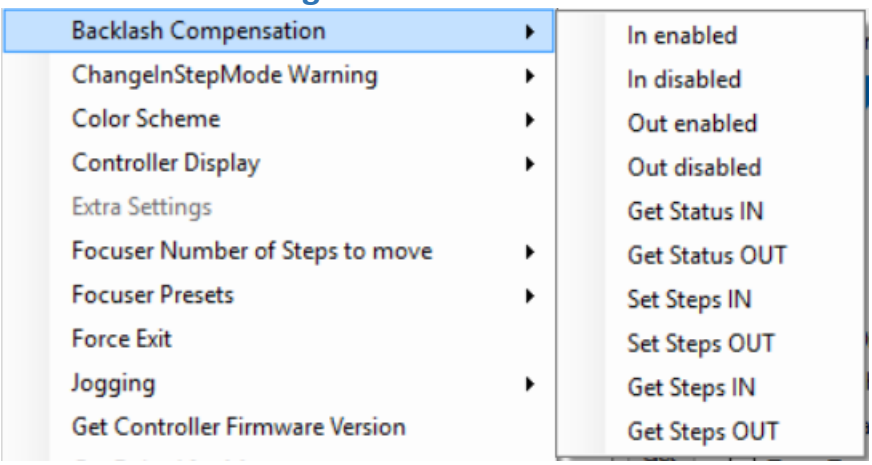
OUT Steps - The number of steps of backlash to apply after the OUT move has completed

Send Backlash Settings when connecting to controller – Check this box to send the backlash settings to the controller on connect. If unchecked, the values will be updated from the controller when connection is made.

Backlash Max Steps - The maximum number of possible backlash steps. This value should match the defined variable MAXBACKLASHSTEPS in the firmware file. The default value is 50.

Controller supports Backlash - If the controller firmware does NOT support backlash, then ensure this option is unchecked.

The Backlash Settings Menu



Notes about backlash

Backlash is only applied if

1. Backlash is enabled in both the settings and firmware, and
2. The new focuser direction is not the same as the previous focuser direction

If a backlash call is made to a controller that does not support backlash (compiled without the BACKLASH enabled), then a serial timeout error will occur. In this case, uncheck the **Controller supports Backlash** checkbox on the Backlash Settings form before connecting to the controller.

The backlash compensation in myFocuserPro2 works slightly differently than some other types implemented in other focusers.

1. When a move is requested, the direction of the move is determined to see if it is the opposite direction to the previous move.
2. If the answer is yes, then (if backlash compensation is enabled for the direction being requested) the focuser firmware performs the backlash steps without adjusting the target position (but ensures that it does not exceed the boundaries of 0 and maxSteps).
3. The focuser then performs the move to the desired target position

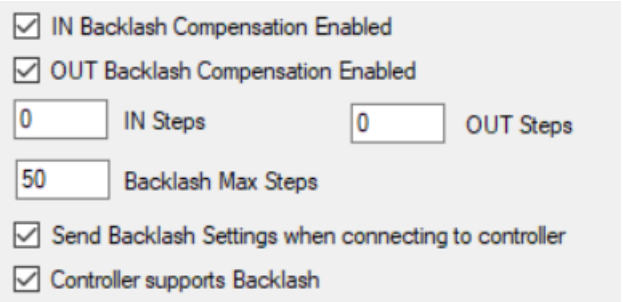
Because the backlash is implemented in this way, the final position will end up being the same position as that requested by the move.

How to determine and set Backlash

To determine backlash, you need the controller fitted to the telescope, with all the controller settings as per normal usage and the controller firmware supporting backlash. This can be done during the day.

You will need to use the Windows application to perform the tests below. Remember that backlash is only applied if enabled in the firmware and the direction of move is opposite to the previous move.

It is best to use a dial indicator (or digital caliper) to determine if the focuser has moved. These can be setup on the end of the focuser lip, eyepiece, imaging camera etc to give an indication of movement. If you do not have one, you could mark a small dot on the larger focuser knob or gear.



Once you have the dial indicator positioned correctly, start the Windows Application (ensure 12V is connected and turned on else the stepper will not move).

Open the extra settings form and set the backlash as indicated on the left.

Close the extra settings form and connect to the controller.

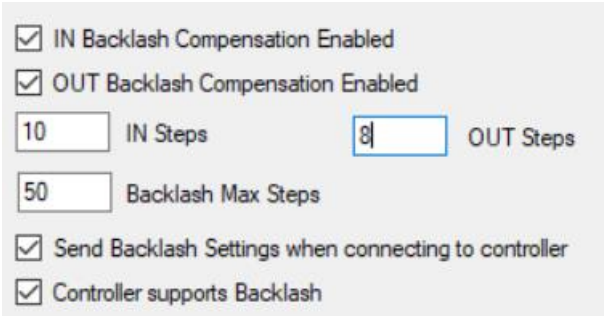
Determine the amount of IN backlash

1. Move 100 steps OUT (100+)
2. Ensure that the dial indicator is positioned correctly
3. Note the focuser position
4. Move 1 step IN
5. After each single step, check the reading of the dial indicator to see if there has been movement
6. If there has been no movement, go back to step 4 and move IN again by one step
7. When you have detected movement, this means the backlash has been overcome. Note the focuser position and the difference between the position at step3 and now is the IN backlash steps
8. Remember this value (write it down, my value was 10)

Determine the amount of OUT backlash

1. Move 100 steps IN (100-)
2. Ensure that the dial indicator is positioned correctly
3. Note the focuser position
4. Move 1 step OUT
5. After each single step, check the reading of the dial indicator to see if there has been movement
6. If there has been no movement, go back to step 4 and move OUT again by one step
7. When you have detected movement, this means the backlash has been overcome. Note the focuser position and the difference between the position at step3 and now is the OUT backlash steps
8. Remember this value (write it down, my value was 8)

Disconnect from the focuser. Open the extra settings screen. Enter your values then click Close.



Connect to the focuser to send the values to the controller.

Backlash and Client Applications

If the client application supports backlash compensation, it is recommended to enable the backlash compensation in the firmware and DISABLE the backlash compensation in client application.

DO NOT have backlash compensation in both the client application and the firmware enabled at the same time.

Backlash and ASCOM drivers

As ASCOM provides no means for handling backlash, the settings will only be able to be configured on the extra settings form from the setup-dialog form for ASCOM (in the same way that MotorSpeed and other values are handled). Any backlash setting can then be sent to the controller when connecting and would then be set and remembered.

Once you have setup the correct backlash values in the controller, you can UNCHECK the *Send Backlash Settings when connecting to controller* checkbox.

Do Not Be Fooled

Do not be fooled by trying to determine backlash visually. Inherently you will get different and inconsistent values. You must make sure that the focuser is outside the critical focus.

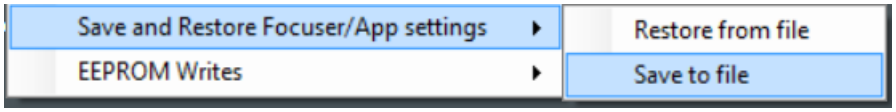
The original goal was to get around 10 steps within the critical focus zone. If you try to determine backlash when the focuser is within the critical focus zone, you may need up to 10 steps or more before any visual indication of change occurs. The exact amount of backlash steps cannot be determined.

Determining backlash using the visual approach is fraught with inherent issues and should be avoided.

The Settings Save/Restore Options

This option has been added as an aid in upgrading the firmware or windows application. Users should not use this option for saving settings between sessions as this is already automated.

These options are designed to work with upgrades to the Arduino firmware or Windows application.



Restore from file - Restores the previously saved myFocuserPro2 controller and Windows application settings from a text file in the specified folder. After restoring the settings, the application will save the myFocuserPro2 controller settings to EEPROM, reboot the myFocuserPro2 controller, and exit the application. Next time the application is started the restored settings will be in effect.

Save to file - Saves the current myFocuserPro2 controller and Windows application settings to a text file in the specified folder (the name of the file is automated)

WRITE FOCUSER DEFAULT SETTINGS

The Windows application has a menu option “Write Focuser Default Settings”. This is useful if you want to set the focuser to a default configuration.

You can define your own default configuration for your focuser. When this option is selected from the Windows Application menu, a command is sent to the controller to write default settings for all the parameters stored in EEPROM. What these default settings are have been defined within the Arduino code for the controller and can be altered by the user.

You can change what these default settings are, but it must be done in the firmware file for the controller.

To change the default settings to be written

Double click on the firmware file for your controller, and scroll down till you find the below code. Please do NOT add or remove any setting as this will cause issues with focuser operation.

```
void setfocuserdefaults()
{
  myfocuser.validdata = 99;
  myfocuser.maxstep = 10000L;
  myfocuser.fposition = 5000L;
  myfocuser.coilPwr = true;
  myfocuser.ReverseDirection = false;
  myfocuser.stepmode = 1;    // full stepping
  myfocuser.updatedisplayintervalNotMoving = 2500L;
  myfocuser.stepsizeenabled = false;            // default state is step size OFF
  myfocuser.stepsize = DefaultStepSize;
  myfocuser.tempmode = true;                    // default is celsius
  myfocuser.ds18b20resolution = TEMP_PRECISION;
  myfocuser.tempcompenabled = false;
  myfocuser.tempcoefficient = 0;
  myfocuser.lcdupdateonmove = false;
  myfocuser.DelayAfterMove = 0;
  myfocuser.backlashsteps_in = 0;
  myfocuser.backlashsteps_out = 0;
  myfocuser.focuserdirection = 0;
  myfocuser.backlash_in_enabled = false;
  myfocuser.backlash_out_enabled = false;
  writeEEPROMNow();    // update values in EEPROM
}
```

Any parameter value in green can be changed. Once you have changed the parameters, you will need to program the controller again with the new file.

Remember that the controller is only updated if connected and when the menu setting option “Write Focuser Default Settings” is selected from the menu bar.

WHAT HAPPENS WHEN THE WINDOWS APPLICATION CONNECTS TO THE myFOCUSERPRO2 CONTROLLER?

If the Serial Port is selected
    If the Serial Port is opened
        Set the baud-rate
        Set Bits=8, No Parity, Stops=1
        Set read and write timeout values
        IfControllerIsAMicro
            Set additional Serial Port Parameters
        Connect
        If TFT display then wait 10s else wait DelayOnConnect value
        Send Connect code
        If response is valid then continue else abort
        Get maxStep
        Get maxIncrement
        Get position
        Get coil power
        Get reverse direction
        Get step mode
        Send Displaystate Celsius or Fahrenheit
        Send LCDEnable/Disable state
        Get firmware version
        Get updatedisplayintervalNotMoving (LCD page display time)
        Get the number of stepsperdegree temperature compensation
        Get temperature precision
        Get temperature
        Get LCDUpdateWhenMoving status
        Send motor speed
        If Update Stepsize Settings On Connect
            Send stepsizeenabled state
            Send stepsize value
        Else
            Get stepsizeenabled state
            Get stepsize
        If MotorSpeedChange Enable is True
            Send MotorSpeedChange enable state True
        Else
            Send MotorSpeedChange enable state False
        If MotorSpeedChange Update On Connect
            Send MotorSpeedChange threshold value
            Send MotorSpeeedChange enable setting
        Else
            Get motorspeedchange threshold value
            Get motorspeedchange enabled state
        Send DelayAfterMove
        If ControllerSupportsBacklashInFirmware is True
            If BacklashSendOnConnect is True
                Send BackLashInEnable state
                Send BacklashOutEnable state
                Send BacklashInSteps
                Send BacklashOutSteps
            Else
                Get BacklashInEnable state
                Get BacklashOutEnable state
                Get BacklashInSteps
                Get BacklashOutSteps

myFOCUSERPRO2 MINI APPLICATION

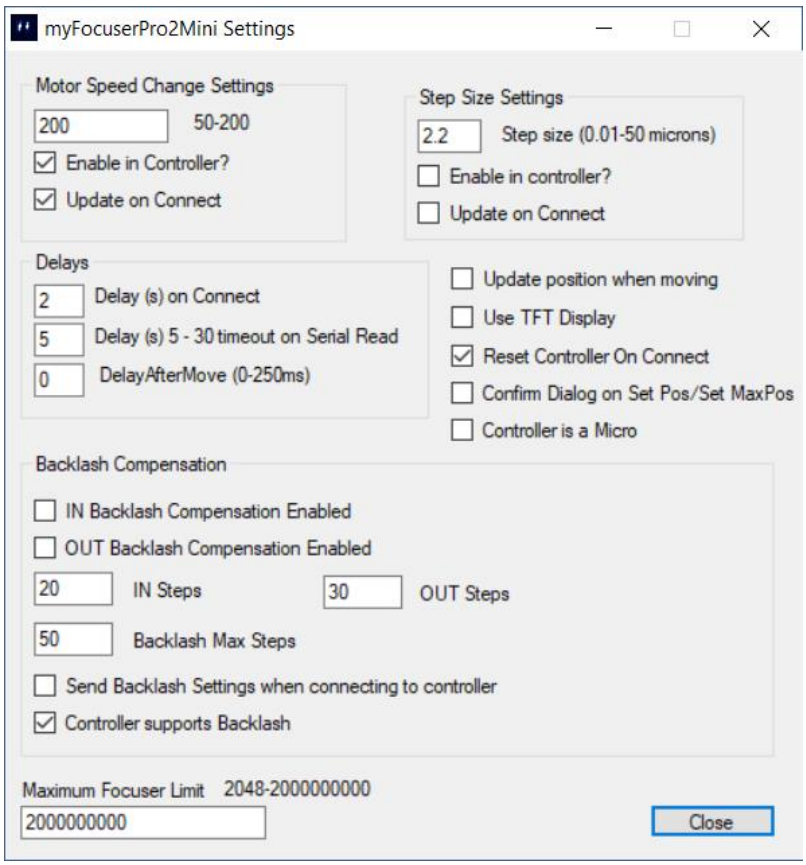
The myFocuserPro2Mini is a slimmed version of the myFocuserPro2 Windows Application program, but retains most of the same features. It is designed to be compact and save on screen real-estate.

Tool tips will appear over the focuser move buttons to indicate how many steps the focuser will move when that particular button is clicked.

The majority of the menu options are the same as the main Windows application.

The main differences are as follows

- Some settings are moved to the Settings form and must be set before connecting to the myFocuserPro2 controller
- After connection is established, the Disconnect button will flash and the computer will beep (if sound enabled) two times to indicate that the application has successfully connected
- The maxStep values cannot be specified (read only)



myFOCUSERPRO2 WINDOWS APPLICATION SETTINGS

The following list are the settings that are remembered by the Windows Application and their default values. The current settings are saved when the application closes and reloaded when the application starts.

ActivateFormLocation	0,0
BacklashEnabledIn	False
BacklashEnabledOut	False
BacklashFormLocation	0,0
BacklashMaxSteps	50
BacklashSendOnConnect	False
BacklashStepsIn	0
BacklashStepsOut	0
CheckChangeInSteppingMode	True
ClearEEPROM	False
ClearEEPROMPassword	
CoilPower	False
ComPortSpeed	9600
ComPortString	
ConfirmDialogSetPosSetMaxPos	False
ControllerFormLocation	0,0
ControllerIsAMicro	False
ControllerSupportsBacklashInFirmware	True
CustomBackColor	SILVER
CustomColors	0
CustomForeColor	RED
CustomTxtBoxForeColor	RED
CustomTxtBoxBackColor	SILVER
DelayAfterMove	0
DelayOnConnect	2
DelayTempCalls	5000
DisplayInCelcius	True
EEPROMFormLocation	0,0
EnableStepSize	False
ErrorFormLocation	0,0
errorlogPath	
ExtraFormLocation	0,0
FocuserPosition	
FormLocation	0,0
LCDEnabled	True
loggingerrors	False
MaxFLimit	2000000000
MaxFocuserPosition	2000000000
MaxLowerLimit	1024
MotorChangeThreshold	200
MotorSpeed	0 (slow)
MotorSpeedChangeEnable	True
MotorSpeedChangeUpdateOnConnect	True
Preset1Label	Preset Position 1
Preset2Label	Preset Position 2
Preset3Label	Preset Position 3
Preset4Label	Preset Position 4
Preset5Label	Preset Position 5
Preset6Label	Preset Position 6
Preset7Label	Preset Position 7
Preset8Label	Preset Position 8
PresetPosition1	0

PresetPosition2	0
PresetPosition3	0
PresetPosition4	0
PresetPosition5	0
PresetPosition6	0
PresetPosition7	0
PresetPosition8	0
ReadTimeOutWait	5
ResetSerialPort	True
ReverseDirSetting	False
SettingsFormLocation	0,0
SteppingMode	2 (Half steps)
StepSize	2.2
StepSizeLowerLimit	0.001
StepSizeUpdateOnConnect	False
StepSizeUpperLimit	50.0
TempCEnabled	False
TempCompRecording	False
TempCStepsPerDegree	0
TempFormLocation	0,0
templogging	False
TempOffset	0.0
TempReFocusNow	False
TempTimerRunning	False
TimerRefreshRate	10000
TimerTickBox	False
UpdatePosWhenMoving	False
UseTFTDisplay	False

Some values are sent to the controller when connecting and some values are retrieved from the controller when connecting.

Please see the following video's
https://www.youtube.com/watch?v=mZmWG_vQ1uM

https://www.youtube.com/watch?v=yuXUDxkdAgU

https://www.youtube.com/watch?v=jsq7kXDv3_Y

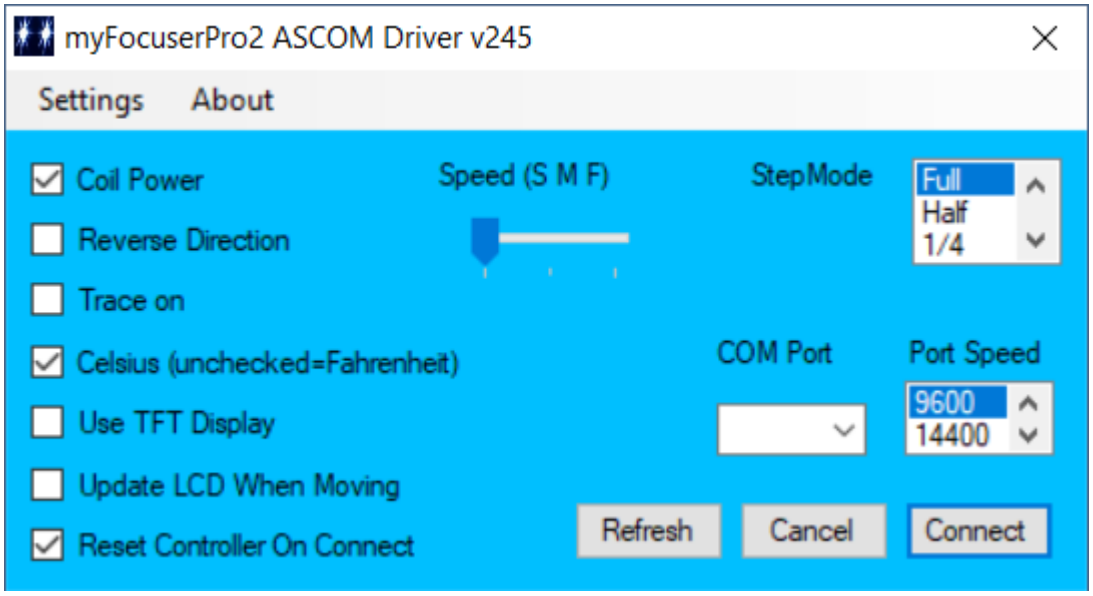
myFOCUSERPRO2 ASCOM DRIVER

The myFocuserPro2 ASCOM driver provided comes with an installer program. The ASCOM driver works with ALL build options. Half steps are ignored when using the ULN2003 build option.

Additional settings are accessed via the Settings menu bar option and must be set before connecting to the controller.

To run two focus controllers, you need to install the secondary ASCOM driver (see here)

IT IS IMPORTANT THAT IF USING BOTH THE ASCOM DRIVER AND WINDOWS APPLICATION TO CONTROL THE FOCUSER THAT YOU USE THE SAME SETTINGS FOR REVERSE DIRECTION, HALF STEPS (or stepmode) AND COIL POWER IN BOTH THE ASCOM DRIVER AND WINDOWS APPLICATION.



These controls must be SET to their correct state BEFORE connecting to the controller.

Coil Power - When enabled (ticked), indicates that coil power is ON and the stepper coils are powered after the move is completed

Reverse Direction - When enabled (ticked), indicates that the motor moves in the opposite direction (IN means OUT and OUT means IN)

Trace On - Debug messages are written to a trace file

Celsius/Fahrenheit - When checked, temperature is displayed in Celsius (default). When unchecked, temperature is displayed in Fahrenheit. This temperature display setting can be changed at any time.

Use TFT Display - Enable this if the controller is fitted with a TFT display

Update LCD when Moving - The LCD display will show an update of the focuser position when moving

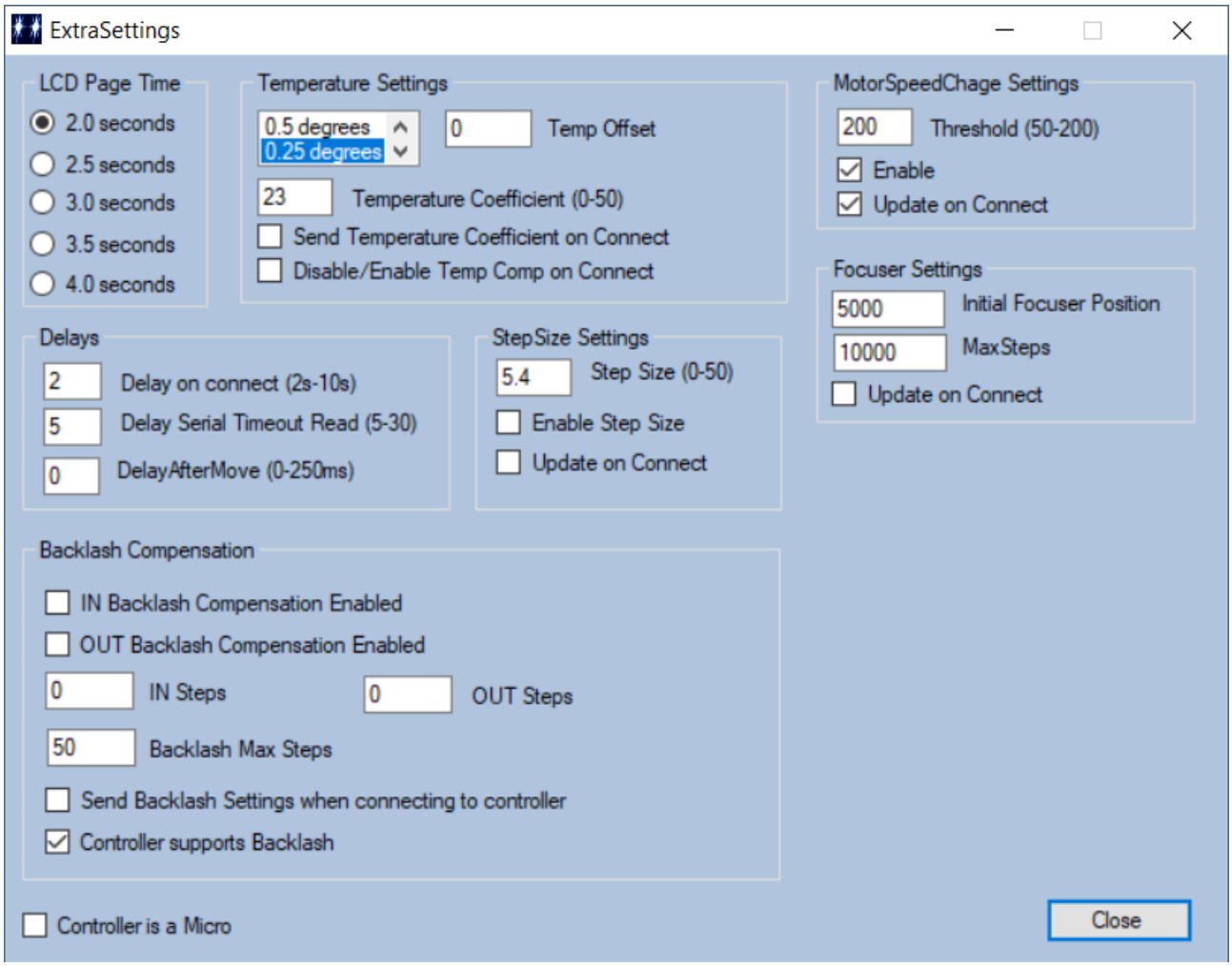
Reset Controller on Connect - When checked, will reset the Arduino controller when connecting. The recommended setting in checked so that the Controller starts with a known state.

Speed (SMF) - The speed of the stepper (delay between steps) can be adjusted in the range Slow-Medium-Fast. The speed settings are NOT remembered by the ASCOM driver

Step Mode - The desired micro stepping mode

Com Port and Com Port Speed - You need to determine the correct Com Port that the focuser is using (you can get this easily by using the Windows Application). Once you have the Com Port value, you need to select it from the Combo List for the Com Port on the ASCOM setup dialog box before clicking the Connect button. The application saves the selected Com Port value. The default speed is 9600.

EXTRA ASCOM DRIVER SETTINGS



These controls must be SET to their correct state BEFORE connecting to the controller.

LCD Page Time - The time in seconds that an LCD screen is displayed for can be adjusted using this menu option, from 2s to 4s delay

Temperature Resolution - Allows the user to control the resolution of the DS18B20 temperature probe from 0.5 degrees down to 0.0625 degrees (9=0.5, 10=0.25, 11=0.125, and 11=0.0625). This setting is remembered by the myFocuserPro2 controller.

Temp Offset - The value to subtract from the temperature reading returned from the myFocuserPro2 controller before returning it to the requesting application. Adjustment values range from –3 to +3. For example, typing -1.5 into the entry-box will subtract 1.5 degrees C from the returned temperature value

Temperature Coefficient - The value in steps per degree for temperature compensation, and is specific to each focuser and must be calculated by the user. The myFocuserPro2 Windows application has special support for calculating this value.

Send Temperature Coefficient on Connect - Use this setting to send the temperature coefficient value to the myFocuserPro2 when connecting. Please note that a myFocuserPro2 controller boots with temperature compensation disabled. Once connected, temperature compensation can be enabled or disabled from the client application.

Disable/Enable Temp Comp on Connect - Use this setting to enable or disable temperature compensation in the myFocuserPro2 controller when connecting. Default is for this setting to be OFF (unchecked). Once connected, temperature compensation can be enabled or disabled from the client application. Only enable this setting with caution. Once temperature compensation is enabled, the focuser will not respond to any move requests until temperature compensation is turned off.

MotorSpeedChange Threshold Value (50-200) - this setting defines the number of steps that the focuser will switch to slowspeed when approaching a target position. Valid ranges are 50-200 steps. The original motor speed setting is remembered and restored once the target position is reached. This setting can be disabled on the Settings>Motor Speed menu.

MotorSpeedChange Enable in controller? - this setting disables or enabled the MotorSpeedChange feature in the myFocuserPro2 controller.

MotorSpeedChange Update on Connect - enable the checkbox to send these values to the myFocuserPro2 controller on Connect.

Delay (s) 2-10 on Connect - specify the delay in seconds after connecting that the driver will wait before sending a command to the myFocuserPro2 controller. Valid values are 2-10 seconds.

Delay (s) 5-30 timeout on Serial Read - specify the delay in seconds that the application will wait when attempting to read from the serial port after sending a command to the myFocuserPro2 controller (default = 5). For Bluetooth or slower devices, a value of 8 or 10 may suffice.

DelayAfterMove - specifies the delay in milliseconds (0-250) that is applied after move to prevent any blurring of star caused by any mechanical residual movement or vibration of the stepper or focuser during focusing.

StepSize in microns - this setting defines the StepSize in microns. You can enter your measured / calculated value for your focuser and this will be sent to the myFocuserPro2 controller when connecting. This allows client applications using the ASCOM driver to retrieve the StepSize setting from the myFocuserPro2 controller. Please see the section in this PDF on determining your focuser step size value.

StepSize Enable in controller? - this setting turns ON or OFF the reporting of the StepSize value from the myFocuserPro2 controller. If you do not know your step size value, DO NOT enable this feature.

StepSize Update on Connect - enable the checkbox to send these values to the myFocuserPro2 controller on Connect.

FocuserSettings Initial Position - This setting specifies the initial position of the focuser.

FocuserSettings MaxSteps - This setting specifies the maximum number of steps for the focus controller and will vary depending upon each user's configuration. You should set this to the maximum number of steps that you have calculated for your focuser [see determining maxStep]. This setting must be specified

before connecting to the controller, and will be remembered by the application. See the section on the initial setup of the focuser for further information.

FocuserSettings Update on Connect - If enabled, the Initial focuser position and maxStep values will be sent to the controller when connecting. If unchecked, the focuser will use the previous saved values in the controller.

IN Backlash Compensation Enabled - Enable or disable backlash compensation of IN moves

OUT Backlash Compensation Enabled - Enable or disable backlash compensation of OUT moves

IN Steps - The number of steps of backlash to apply after the IN move has completed

OUT Steps - The number of steps of backlash to apply after the OUT move has completed

Send Backlash Settings when connecting to controller – Check this box to send the backlash settings to the controller on connect. If unchecked, the values will be updated from the controller when connection is made.

Backlash Max Steps - The maximum number of possible backlash steps. This value should match the defined variable MAXBACKLASHSTEPS in the firmware file. The default value is 50.

Controller supports Backlash - If the controller firmware does NOT support backlash, then ensure this option is unchecked

Controller is a Micro - Check this box if the Arduino controller is a Micro. For all other controllers, leave unchecked. *This also requires special changes to the Arduino firmware file.*

Please see the following video
https://www.youtube.com/watch?v=Gh4dpqDFouQ

myFocuserPro2 ASCOM Settings

These following settings are stored in the ASCOM driver (also accessible via the ASCOM Profiler).

BacklashEnabledIn
BacklashEnabledOut
BacklashMaxSteps
BacklashSendOnConnect
BacklashStepsIn
BacklashStepsOut
CoilPower
ComPortSpeed
ControllerIsAMicro
ControllerSupportsBacklashInFirmware
DefaultMaxfocusersteps
DefaultStepSize
DelayAfterMove
DelayOnConnect
FormLocation
InitFocusPos
LCDPageDisplayTime
MaxStep
MotorSpeedChangeEnable
MotorSpeedChangeThresholdValue
MotorSpeedChangeUpdateOnConnect
MotorSpeedSetting
MyComPort
ResetControllerOnConnect
Reversed
SendTCOnConnect
SendTempCompOnConnect
SerialPortReadTimeout
SettingsFormLocation
StepMode
StepSize
StepSizeEnabled
StepSizeUpdateOnConnect
TempCompEnabled
TempCompSetting
TemperaturePrecision
TempMode
TempOffset
TFTDisplay
TraceEnabled
UpdateLCDPos
UpdateOnConnect

ASCOM DRIVER CONFORM VALIDATION

Every release of the myFocuserPro2 ASCOM drivers are tested against the CONFORM utility. This program tests to ensure that the functions/methods/properties provided by the ASCOM driver are compliant to the ASCOM standard.

All myFocuserPro2 drivers are fully ASCOM compliant and the validation reports are available on the myFocuserPro2 site. They are always tested before release and tested against the latest available version of Conform.

Temperature Compensation and the ASCOM driver

An **ASCOM client** has the ability to send a command to an ASCOM driver and enable/disable temperature compensation. The myFocuserPro2 ASCOM driver and firmware supports temperature compensation.

Temperature compensation is enabled by the ASCOM Client application. If your client application does not support this then please contact the developer to have this function added to their application.

For temperature compensation to work, you need to calculate the temperature co-efficient for your focuser/telescope. The Windows application can automatically calculate the temperature coefficient and store the value in the controller. Use the Windows application to calculate the temperature coefficient and store this value in the controller. Once set, the ASCOM driver can use this value.

Here is the part of the ASCOM Conform report for the myFocuserPro2 related to temperature compensation

```
17:30:58.416 TempCompAvailable      OK      True
17:30:58.456 TempComp Read          OK      False
17:30:58.465 TempComp Write         OK      Successfully turned temperature compensation on
17:30:58.473 TempComp Write         OK      Successfully turned temperature compensation off
17:31:06.998 Move - TempComp True           Moving to position: 61748
17:31:07.005 Move - TempComp True   OK      .NET InvalidOperation Exception correctly raised as expected
Conformance test complete
No errors, warnings or issues found: your driver passes ASCOM validation!!
```

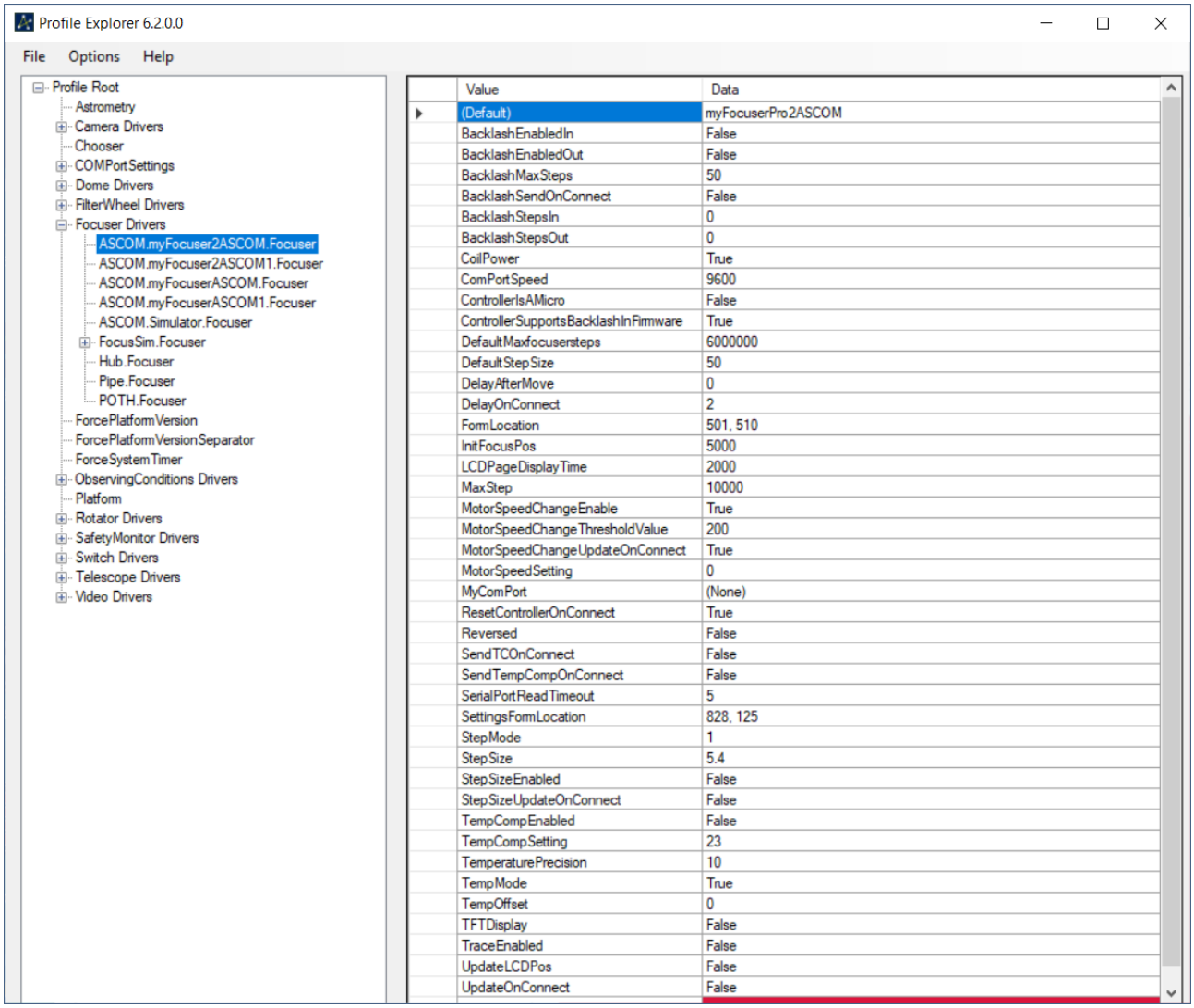
What happens when the ASCOM driver connects to the myFocuserPro2 Controller?

The ASCOM driver performs the following actions when connecting to a myFocuserPro2 controller

```
IfControllerIsAMicro
        Set additional Serial Port Parameters
 If update focuser settings
        Send maxStep
        Get maxIncrement
        Send maxIncrement
        Send Position
 Get maxStep
 Get maxIncrement
 Get Position
 Send Reverse Direction
 Send StepMode
 Send Coil Power
 Send MotorSpeed
 Send LCDPageDisplayTime
 Send Celsius or Fahrenheit
 If update Stepsize
        If StepSize Enabled
                Send Enable Step Size
                Send Step Size
 Send Temperature Precision
 If update MotorSpeedChange
        If MotorSpeedChange Enabled
                Send MotorSpeedChange Enabled
                Send MotorSpeedChange Threshold
 If TempCompValue enabled on Connect
        Send TempComp value
 If Enable Temperature Compensation on Connect
        Enable Temperature Compensation
 else
        Disable Temperature Compensation
 Send DelayAFterMove
 Send backlash settings if enabled
```

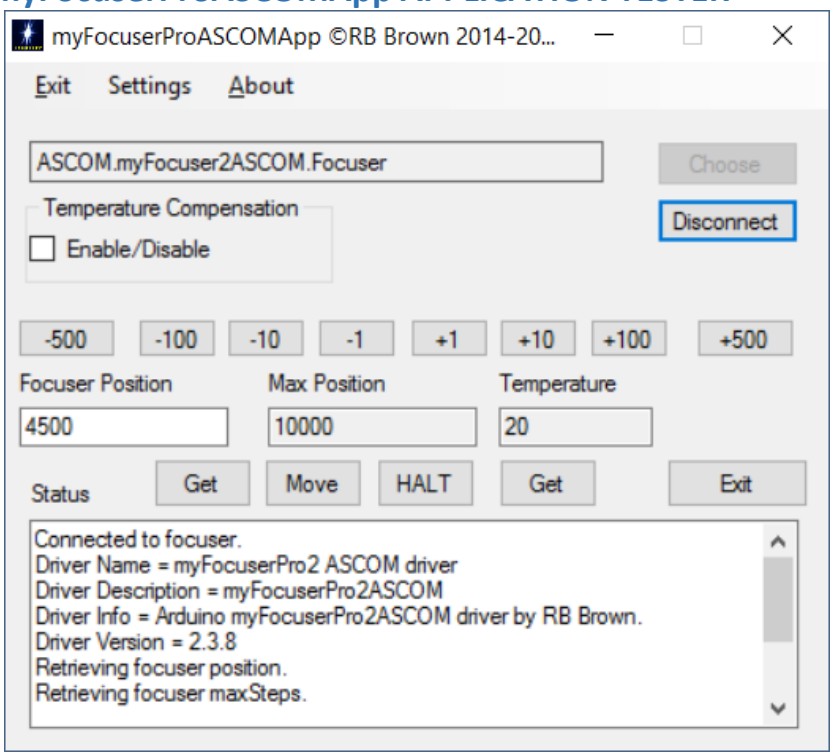
USING THE ASCOM PROFILE EXPLORER TO VIEW THE ASCOM DRIVER SETTINGS

If you have the ASCOM development tools installed, you can view the ASCOM settings saved by the ASCOM driver.



It is not recommended to edit or change these settings using the Profile Explorer as this could lead to the driver not being able to connect to the myFocuserPro2 controller.

myFocuserProASCOMApp APPLICATION TESTER



The myFocuserASCOMAPP is a Windows software application that is actually an ASCOM client and will talk to any ASCOM focuser driver and allow you to control the focuser.

The settings menu has a number of options, similar to those provided by the Windows Application and the Windows Mini Application for controlling the myFocuserPro2 controller.

This is a client program and you can use it to test any of the features of the myFocuserPro2 ASCOM driver, including temperature compensation.

HOW TO RUN TWO FOCUSERS

What happens if you have two telescopes and want to have two myFocuserPro2's, one on each scope?

There are some simple rules.

1. You can only run one instance of the myFocuserPro2 Windows application at any time
2. You can only run once instance of the myFocuserPro2Mini application at any time
3. You can only run one instance of the ASCOM driver at any one time (but you can run both the ASCOM and ASCOM1 drivers at the same time)

Here are some options for running two focusers

Scope	Option 1	Option 2	Option 3
Scope 1: Orion ED80	myFocuserPro2 App	myFocuserPro2 App	ASCOM driver
Scope 2: SV102T	MyFocuserPro2Mini App	ASCOM Driver	ASCOM1 driver

You **cannot** use two instances of the same Windows application to control both scopes. The reason for this is because the Windows application uses an application setting file that stores certain settings and these settings need to be different for each scope (such as com port) and hence the application can get confused as to which controller it is communicating with. As indicated above, Option 3 indicates controlling both scopes using the ASCOM and ASCOM1 available drivers.

Please see the following video
https://www.youtube.com/watch?v=sEvvWYNMCFs

TEMPERATURE COMPENSATION

Temperature compensation is the automatic adjustment of focus position based on changes in temperature. It will be different for each telescope. This feature is turned ON or OFF by an **ASCOM client** by sending the correct command to the ASCOM driver. The Windows application can talk directly to the controller and enable/disable temperature compensation.

Each user must determine their own compensation value for their equipment. This means calculating a **temperature coefficient** value, which is the amount of steps the focuser needs to move to best focus when the temperature changes by 1 degree.

The user waits for the telescope optics to achieve thermal equilibrium and then takes a series of measurements over a period of temperature change. These measurement results are readings of focuser position versus temperature which can be plotted on a graph. The slope of the graph then is the temperature coefficient (ideally the graph should be a linear line but this might not be the case).

Generally, as temperature drops over the course of the evening, the focuser position will move inwards. We are going to record the in movements over changes in temperature.

Generally, temperature compensation is only applied in one direction (inwards or as a drop in temperature), though some systems might have compensation in both directions.

Assuming we only do temperature compensation in one direction, this also avoids backlash issues which might accrue from a reverse change in focuser direction.

So, manually, we might do the following

1. Wait for the temperature to stabilize (thermal equilibrium)
2. Slew to a target star, enable mount tracking and guiding software
3. Use a Bahtinov mask (or FWHM value) to get the best focus
4. Wait for the temperature to drop by a specified amount (3 degrees)
5. Refocus
6. Record the temperature and the focus position
7. Calculate the temperature coefficient
8. Update the controller settings and enable temperature compensation
9. Remove the Bahtinov mask and start imaging

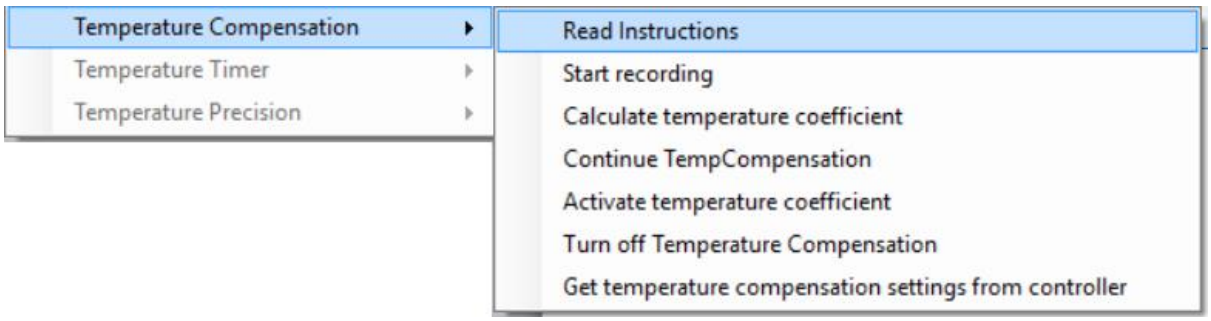
The following pages outline how to do this with the myFocuserPro2 controller and application software.

Step 0 Read the Instructions

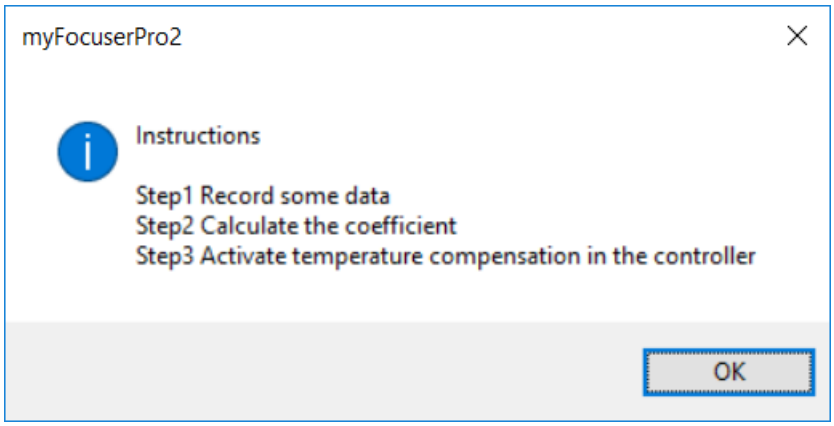
The following assumptions are

1. The temperature probe is located near the optics of the telescope (for a refractor this would be near the front lens cell)
2. The telescope has reached thermal equilibrium. This may take up to 30 minutes or more to occur and varies depending on the conditions and telescope type and size
3. The myFocuserPro2 controller is connected and the Windows Application is running
4. The telescope is at optimal focus and tracking the target star

The first step is to access the **Read Instructions** option from the Temperature Compensation menu of the myFocuserPro2 windows application. Select the Read Instructions option

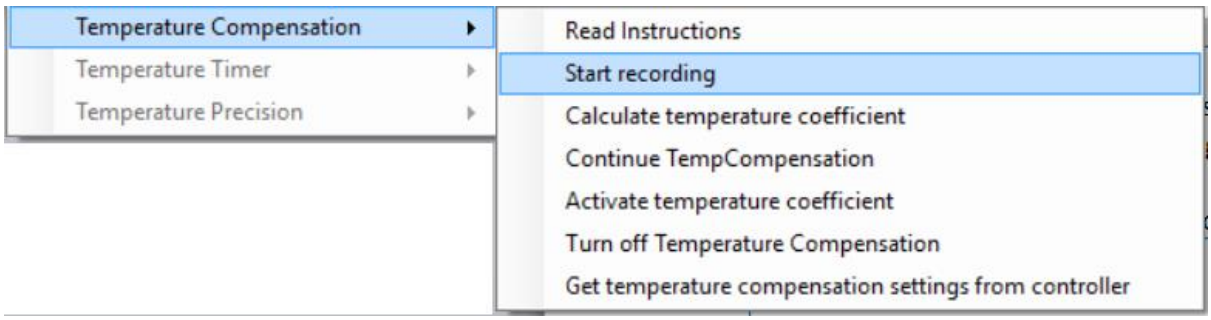


This displays the following MessageBox of the steps you must follow to determine and apply a temperature coefficient value and enable temperature compensation for your myFocuserPro2 controller. The steps are performed one after the other in order. If you make a mistake, simply start again from Step 1.

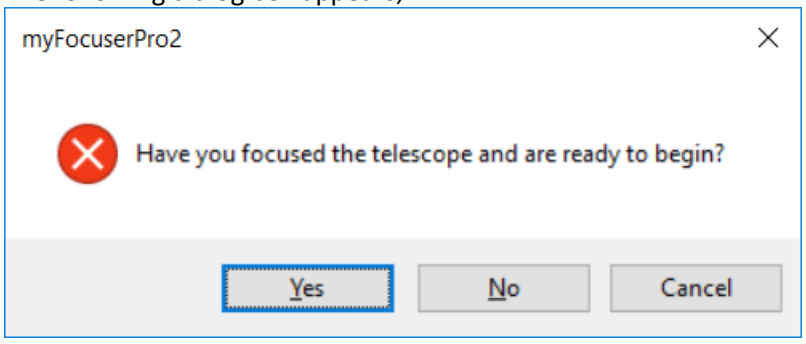


Step 1 Start Recording

The next step is to access the **recording** option from the Temperature Compensation menu of the myFocuserPro2 windows application. Select **Start recording**

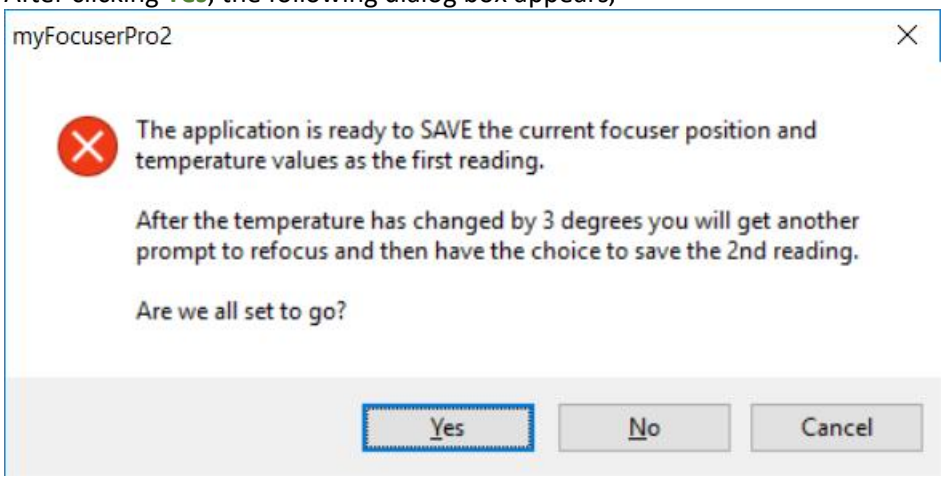


The following dialog box appears,

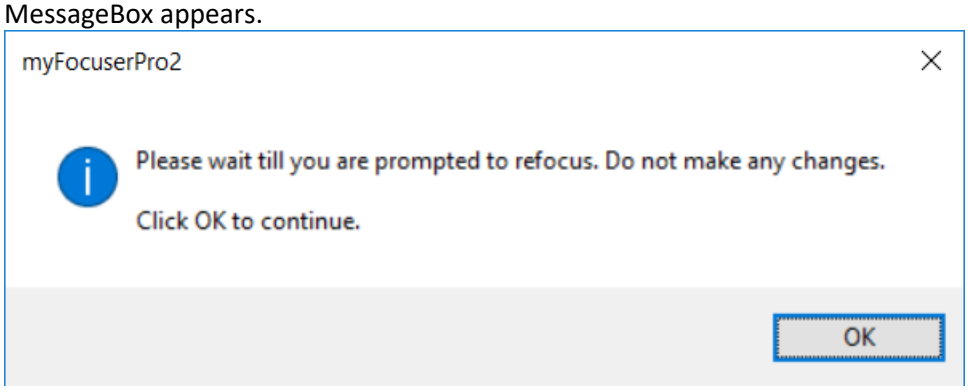


If you have already focused the telescope within the last few minutes, click **Yes**, else click No and refocus the telescope before starting again.

After clicking **Yes**, the following dialog box appears,



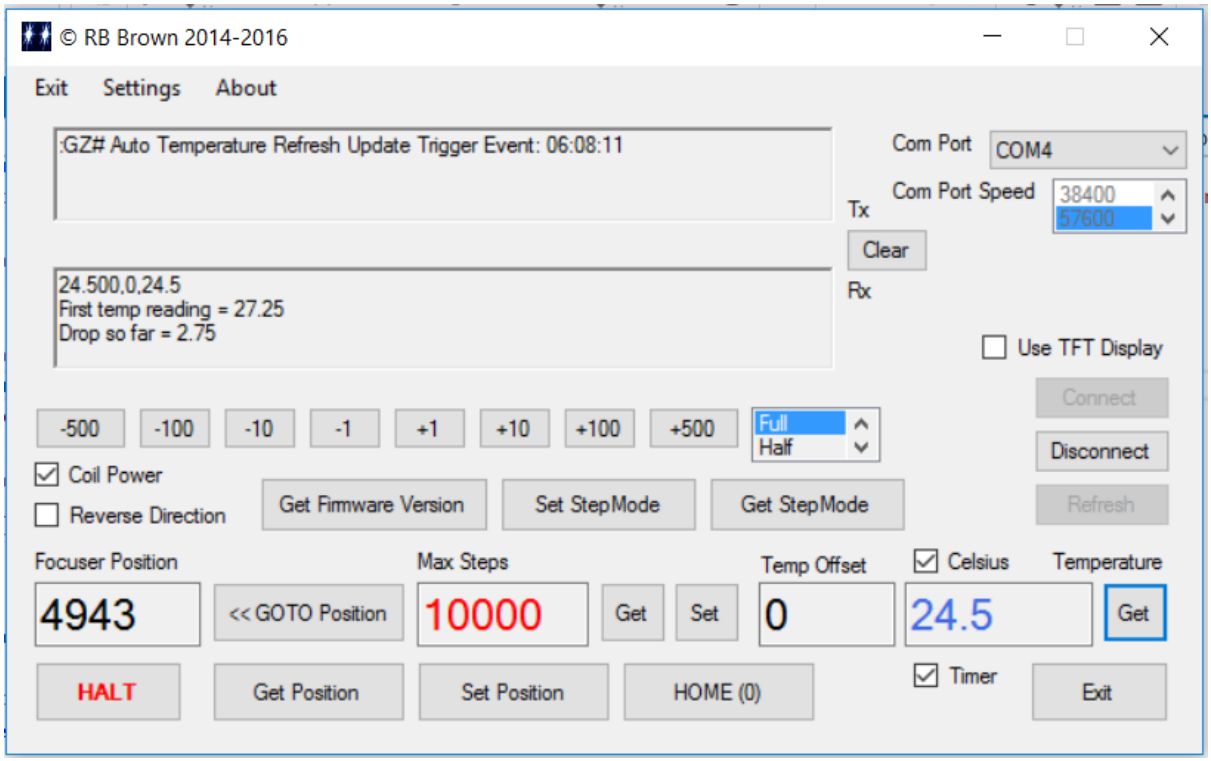
When you click **Yes**, you will be automatically taken to the record dialog option and the following MessageBox appears.



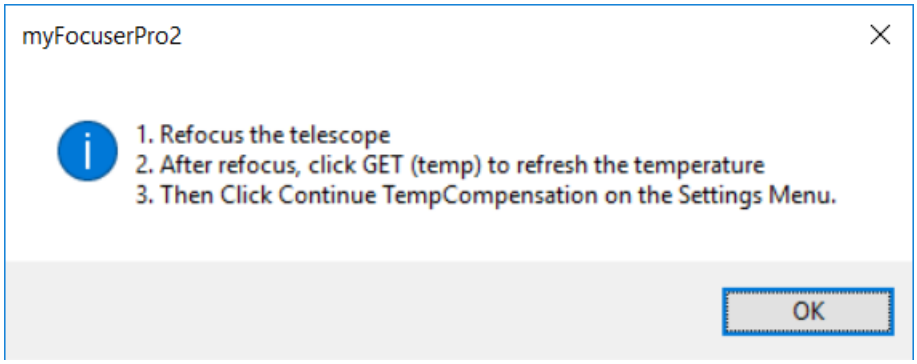
At this point, the program has automatically

- saved the current temperature and focuser position
- started the auto refresh timer for the temperature updates and enabled it to a 10s refresh cycle
- started to monitor the temperature change

Just wait for the application to monitor the temperature and prompt you when ready. During this time interval DO NOT MAKE ANY CHANGES TO ANY SETTINGS. You will be able to see the progress in the Rx textbox, as indicated below

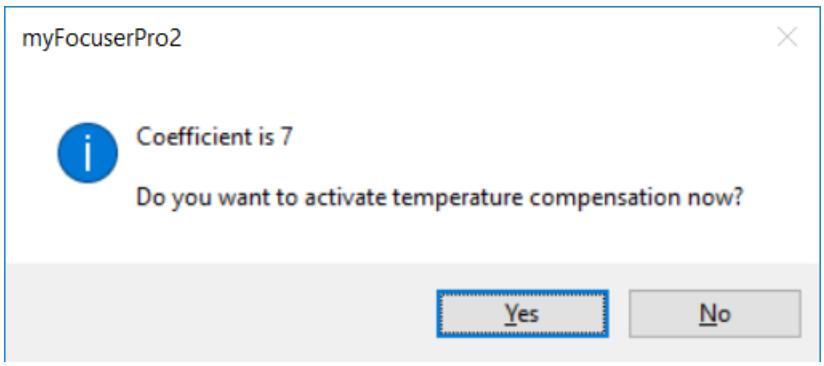


Once the program detects a 3 degree change in temperature, a new dialog MessageBox will automatically appear asking you to refocus the telescope. Click **OK** and then refocus the telescope.

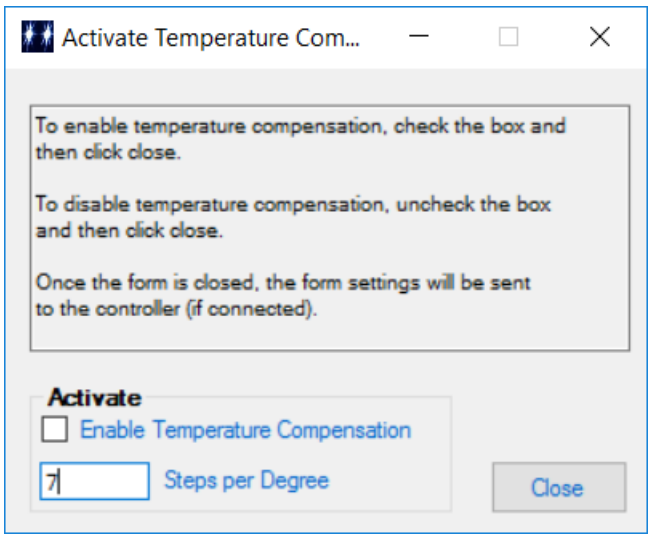


After refocusing, get the latest temperature reading by clicking the GET button for temperature. Once the new temperature value is displayed, click the **Continue TempCompensation** menu option on the settings menu under Temperature Compensation.

The application will automatically calculate the temperature coefficient and display the value in a MessageBox (example below) and include an option for you to now update the controller and enable temperature compensation.



Click **YES** to Goto the next step. The application will now display the Apply Coefficient Form as shown below.



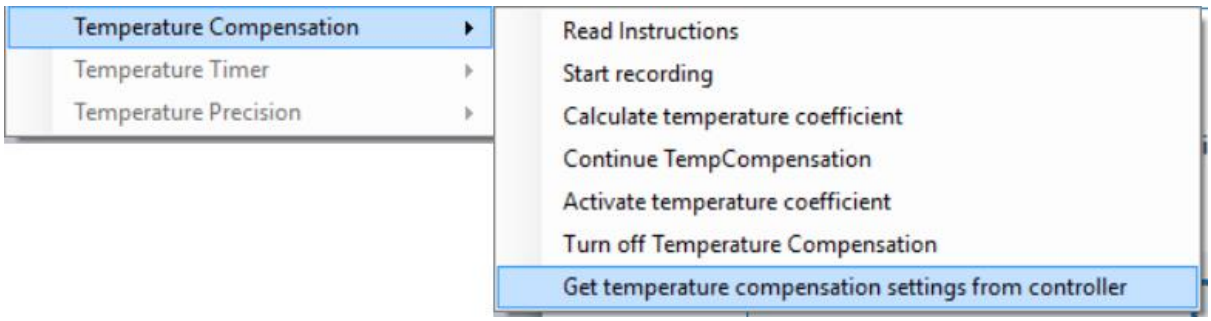
The Steps per Degree is automatically preloaded from the previous calculation. To send this value to the controller, and enable temperature compensation, check the box **Enable Temperature Compensation** and then click the **Close** button.

Note: If you know the temperature coefficient value for your focuser, you can access this menu directly and enter the Steps per Degree value manually.

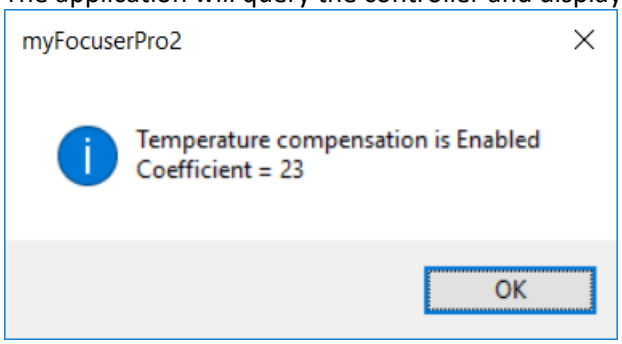
The values are sent to the myFocuserPro2 controller and temperature compensation is enabled. The push buttons are disabled when temperature compensation is enabled in the controller.

Display the Current Myfocuserpro2 Controller Temperature Compensation Settings

To display the current temperature compensation settings, select the **Get temperature compensation settings from controller** option



The application will query the controller and display the current settings.



The Windows Application and ASCOM driver support temperature compensation.

TEMPERATURE COMPENSATION FAQ AND ISSUES

Please read the separate FAQ document.

Please see the following video
https://www.youtube.com/watch?v=YXRqP-V1fcM

EEPROM USAGE

The myFocuserPro2 controller remembers

- Focuser position setting
- maxStep (maximum focuser position setting)
- StepMode
- Reverse Direction
- Coil Power
- Refresh rate of display – how long each LCD page is displayed
- The StepSize value in microns for the focuser (user specified)
- stepsizeenabled - if the StepSize ASCOM request is enabled or should it return a Not Implemented Exception
- temperature probe resolution
- temperature compensation enabled/disabled
- temperature coefficient
- backlash settings
- EEPROMWrites (cumulative total)

These values are stored in the EEPROM of the controller. A smart algorithm is used to minimize the number of EEPROM writes, as there is a limit of around 10,000 writes before the EEPROM location becomes unusable. The location in EEPROM of where this data is dynamically updated so that the entire range of EEPROM locations is utilized.

The myFocuserPro2 controller code, on power up or reset, checks the EEPROM for the last saved position and the setting for maxStep. If found these are updated and sent to the ASCOM driver or application software. If not found, then default values are used (when the program is run the first time they do not exist so must be created by default).

If using an **ATMEGA168** which has a 512 byte EEPROM, you need to change the following lines from

```
// #define EEPROMSIZE 512// ATMEGA168 512 EEPROM
#define EEPROMSIZE 1024 // ATMEGA328P 1024 EEPROM
```

to this

```
#define EEPROMSIZE 512// ATMEGA168 512 EEPROM
// #define EEPROMSIZE 1024 // ATMEGA328P 1024 EEPROM
```

The include file “eepromanything.h” must be in the same folder as the Arduino code (ino file).

The focuser position is written to EEPROM after a MOVE command AND when the focuser is then idle for 10s (configurable). This overcomes continual writes which would happen if the focuser was being controlled by FocusMax at each focuser move. The idle-time before a write is controlled by the line

```
long interval = 10000;    // interval in milliseconds to wait after a move before writing settings
                          // to EEPROM, 10s
```

and could be increased without affecting operation of the focuser. You would consider increasing the value if FocusMax was taking longer than 10s between each focuser move and image capture in determining the FWHM of a star. It is very unlikely that the contents of the EEPROM will wear out even after years of use.

WHAT TO DO IF YOU LOSE YOUR FOCUSER SETTINGS

1. Unclamp the focuser coupler connecting the stepper motor to the focuser (or remove belt if using a pulley system)
2. Manually move focuser to initial 0 position (1/2 turn out as described above)
3. Power focuser and start myFocuserPro2 Windows application software (not ASCOM driver)
4. Set the step mode to what you used in the initial setup of the controller
5. Enter 0 as the focuser position and click the SET POSITION button
6. Clamp the focuser coupler so that the focuser motor can now drive the focuser
7. Enter your determined maxStep value into the Maximum Position text box and click the Set Button
8. The focuser is now setup. Enter the position for reasonable focus into the Focuser Position text box and click the GOTO POSITION button to move the focuser to the focus position

UPGRADING FIRMWARE AND SOFTWARE – METHOD 1

In MOST circumstances, you can apply the new firmware directly using the Arduino IDE and install the new application by running the setup program.

New releases often occur after you have built and set-up your myFocuserPro2 controller.

It is important to realize that new drivers, software and firmware (the Arduino code file) fix issues in previous versions as well as introduce new features. Make sure you save any downloaded firmware and software in a folder in case you want to revert back to a previous version.

Support is freely given concerning any current release. Please contact me and I will do my best to work with you to help resolve any issues you might have. Previous releases are not supported.

I know that updating is a pain, and that sometimes you might be hesitant to change something that works. Having said that, rest assured that newer releases are produced for sound reasons, and they might fix an issue that you might be having.

Sometimes the NOTICE with updates will ask you to uninstall the current software before installing the new release (this occurs with the Windows Application and sometimes the ASCOM driver).

When a new release of firmware or application software is released, before upgrading, save the settings to a file. Then install the update, reload the new application, connect to the controller, and then restore the settings.

Any changes that you make to the either the application or controller settings after a save and before a restore are lost. The save to file option saves BOTH the firmware settings and the software application settings at the time that the save to file option is run.

NOTE: I always move the focuser to position 0 before updating any new firmware. This means that once the new firmware is loaded and the new Windows/ASCOM installed, all I need to do is reset the current focuser position to 0 after reloading all the settings.

STEP 1: SAVE SETTINGS

Ensure that the controller is connected. Run the Windows application.

Select from the Settings Menu the Save and Restore Focuser/App settings, then **Save to file**

STEP 2: UPDATE FIRMWARE OR APPLICATION

Close the application. Proceed to update either the firmware or application to the next release.

STEP3: RESTORE SETTINGS

Ensure that the controller is connected. Run the Windows application.

Select from the Settings Menu the Save and Restore Focuser/App settings, then **Restore from file**

Once the settings are restored from the file, the firmware settings are written to EEPROM, the controller is rebooted, and the application will exit (it needs to exit to update the settings).

When the application is restarted, the new settings will take effect.

UPGRADING FIRMWARE AND SOFTWARE – METHOD 2

Previous manual method still works and is effective. Only retained here as an alternative.

What to do first before upgrading

The first thing you should do is write down your important settings, as you may need to re-enter this information after updating.

The important settings are (some of these are recent so might not be on your system)

- Backlash settings
- Focuser Position
- Max Steps
- Coil Power
- Reverse Direction
- Motor speed
- Step mode
- Temp offset
- Step Size
- Temperature Precision
- Temperature Compensation value (if you have undertaken to calculate this)
- Refresh rate of display – how long each LCD page is displayed

You can get these values by running the existing software (like the Windows application). Be sure to write them down.

NOTE: I always move the focuser to position 0 before updating any new firmware. This means that once the new firmware is loaded and the new Windows/ASCOM installed, all I need to do is reset the current focuser position to 0 after reloading all the settings.

Download the required files

The next step is to download the Arduino firmware file, the Windows application and ASCOM driver. As part of the previous step you recorded the firmware version of your current controller. Use this (or the filename) to determine what Arduino firmware file you need.

For example, let's say that your current firmware file is Focuserv217_DRV8825_HW203_F.ino

So you would look for the latest file Focuserv2??_DRV8825_HW203.ino (as of 6th January 2018 it would be Focuserv263_DRV8825_HW203.ino)

Update the controller firmware

Use the Arduino IDE to reprogram the controller with the new firmware file. First make any required changes to the file (such as serialportspeed etc) before reprogramming the controller.

If you built the myFocuserPro2 controller with the power-on reset circuitry, remember to slide the switch into the program position first before turning on power to the controller.

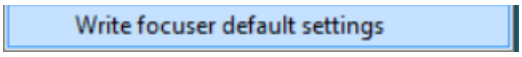
What I do is just remove the controller from the telescope and then reprogram the controller on the work-bench. Once you have finished re-programming the controller, remember to slide the power-on reset switch back to its normal setting (if fitted).

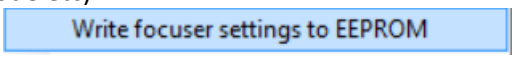
Update the Windows application and ASCOM driver

Once you have updated the Arduino firmware, proceed to installing the new version of the Windows application and ASCOM driver.

The install file should always be run from the same location. Simply create a folder on your hard drive where you can save all the files needed. Then you can download any new updates to the same folder. If you attempt to run the installer from a different directory compared to the previous install, then Windows will complain and you will need to uninstall the application before installing the new version.

Run the Windows Application and set the controller default settings

1. Connect to the controller
2. Select this option 

from the Settings menu
3. Disconnect
4. Wait 10s
5. Reconnect to the controller
6. Now set all the focuser settings as per the settings you wrote down earlier (such as focuser position, maxSteps, stepmode etc)
7. Select this option 

from the Settings menu to write these new settings to the controller

If you have any issues, please feel free to contact me for assistance/advice.

IT IS IMPORTANT THAT TO REALIZE THAT THE ARDUINO FIRMWARE AND WINDOWS APPLICATIONS OFTEN INTRODUCE NEW FEATURES AT THE SAME TIME.

WHAT THIS MEANS IS YOU MUST UPDATE THE FIRMWARE AS WELL AS THE WINDOWS APPLICATION AND ASCOM DRIVER TOGETHER. YOU CANNOT RUN THE LATEST WINDOWS APPLCIATION OR ASCOM DRIVER ON ARDUINO FIRMWARE THAT MAY BE SEVERAL VERSIONS EARLIER.

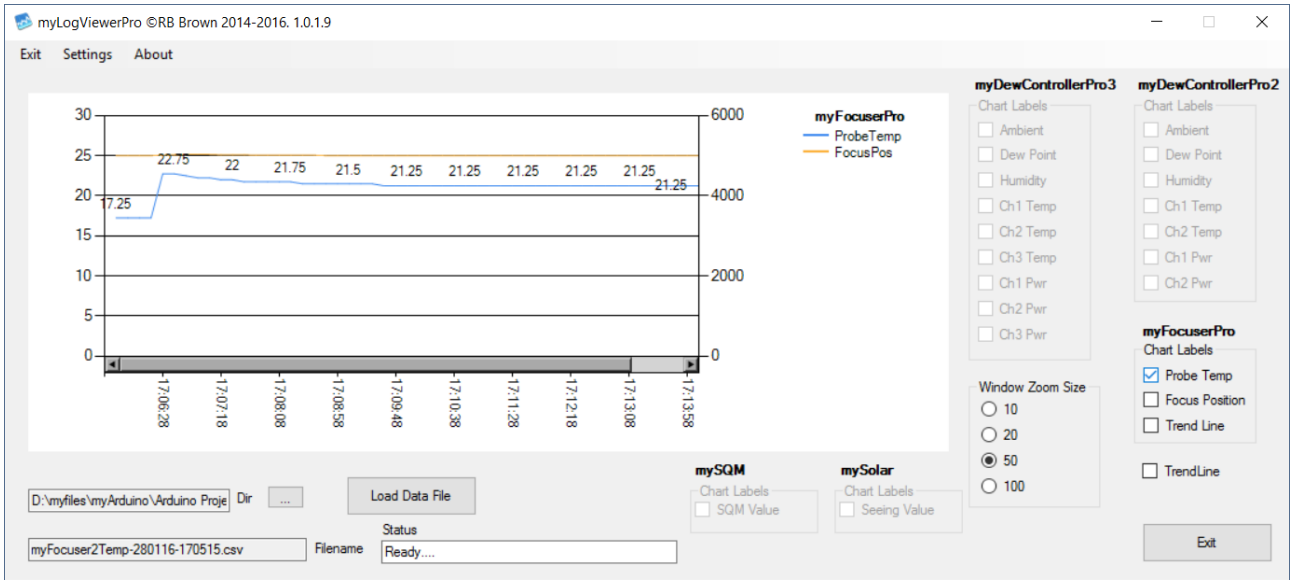
Please see the video
https://www.youtube.com/watch?v=cSvOVw8Djsw

DATA LOG VIEWER

In [myFocuserPro2 windows application] version 2.1.3.8 and greater, a new menu option was added to the settings menu. The new option "**Log Temperature and Position**" allows the user to save the temperature probe reading and the focuser position value to a log file for later analysis (such as determining temperature compensation values for your focuser setup).

This feature is **enabled** only when the **automate** feature is enabled. During the automatic update (when the automate check box is checked), the routine will get the current temperature and focuser position from the myFocuserPro2 controller. If the temperature logging file is enabled, then these values will also be written to the associated log file.

Below is an example of the new application **myLogViewerPro**, which can display both [myDewControllerPro3] and myFocuserPro2 data log-files for analysis.



RECOMMENDED PCB BUILD IS [NANO+DRV8825 HW203 BOARD]

RECOMMENDED STRIPBOARD BUILD IS [NANO+DRV8825 HW203 BOARD]

MANUAL AND AUTOMATED FOCUSING OPTIONS

Manual Focusing

Several programs allow you to see the peak intensity or FWHM (full width half maximum) profile of a star. Examples of these are MaximDL, Nebulosity, PHD2, ScopeFocus, APT and others.

In general, you would

1. slew the telescope to a star
2. enable sidereal tracking so the star does not drift out of view
3. adjust the exposure time of the camera so the star is not overexposed
4. display the star profile and watch the FWHM value
5. adjust the focus till the FWHM value is lowest

In step 5 you would move the focuser by sending commands to the myFocuserPro2 controller to move IN or OUT (I prefer to go OUT first till the star is out of focus, then slowly move IN). This means

1. looking at the star profile and FWHM value
2. letting the values settle for a few exposures to take into account variations in seeing
3. moving in (perhaps by 5 or 10 steps depending on how many steps are within the critical focus zone of your focuser setup)
4. repeat 1-3

Manual Focusing with a Bahtinov Mask

A Bahtinov mask is a valuable focusing tool which is easy to use to find good focus. To achieve focus using a Bahtinov mask

1. slew the telescope to a star (mag 4-5)
2. set the focuser to approximate focus position
3. enable sidereal tracking so the star does not drift out of view
4. adjust the exposure time of the camera so the star is not overexposed
5. place the mask over the objective end of the telescope
6. Adjust focus till the center diffraction is centered (ignore any FWHM or other values as a mask is being used)

Bahtinov Mask Focusing with Nebulosity

1. Start Nebulosity and connect to camera and focuser
    a. Turn on reticle grid – View – Overlay – Grid
    b. Preview 1s exposure, ensure star is centred in FOV
    c. Click Frame and Focus – Use ZOOM if necessary, center star in FOV, adjust focus for best result
    d. Abort
    e. Preview, Click on star
    f. Click Fine Focus
    g. Adjust focus for best results (center Diffraction spikes)
    h. Abort
2. Remove Bahtinov mask

Also see https://www.youtube.com/watch?v=rcGQ7FhlrNQ

Bahtinov Mask Focusing using Bahtinov Grabber And Nebulosity

1. Start Nebulosity and connect to camera
    a. Turn on reticule grid – View – Overlay – Grid
    b. Preview 1s exposure, ensure star is centred in FO
    c. Click Frame and Focus –ZOOM to 100% and center star in FOV by using sliders, adjust focus for best result
2. Run Bahtinov Grabber
    a. Set capture area over the star in Nebulosity
    b. Enter telescope data related to OTA and camera
    c. Eg; Telescope ED80, f=0.480, D=0.080, pixelsize=4.54
    d. Eg; Telescope SV102T-25SV, f=0.714/, D=0.102, pixelsize=4.54
    e. Config – choose ASCOM focuser, 1, AF Speed=3.00 (allows for download)
    f. Autofocus
    g. Quit
3. Nebulosity
    a. Abort
    b. Preview
4. Remove Bahtinov mask

Automated Focusing

Right now, out of the box, you can do automated focusing on stars with any myFocuserPro2 controller fitted to a telescope. You can use the controller with MaximDL, APT and Scopefocus. You could also use FocusMax v3 which was the last free version of Focusmax before it became commercial. We recommend FocusMax v4.

Scopefocus is free at https://scopefocus.blob.core.windows.net/scopefocusbeta/publish.htm

Automated focusing requires that you first configure or train your system to determine the slope for each side of focus and enter details related to your OTA, camera and focuser configurations. These can then normally be saved in a "profile" settings file. After entering the required details, it is then necessary for the focusing program to learn about the focuser by taking a number of exposures at different settings, which are used to create a V curve (an upside-down bell shaped curve that plots how focus of the star is related to focuser position). A number of V curves are generated and averaged to create a V curve for that configuration. This V curve can then be used to automate focusing.

In automated focusing, the focuser will first move to one side of focus and off focus. The star profile is then measured and compared to the V-curve plot. Now the program has a good estimate of where the best focus position will be and will move the focuser close to that position and recapture the star profile. A few further small adjustments may be necessary.

The advantage is quicker focusing, no need for a Bahtinov mask or having access to the telescope (which could be hundreds of miles away in a remote location).

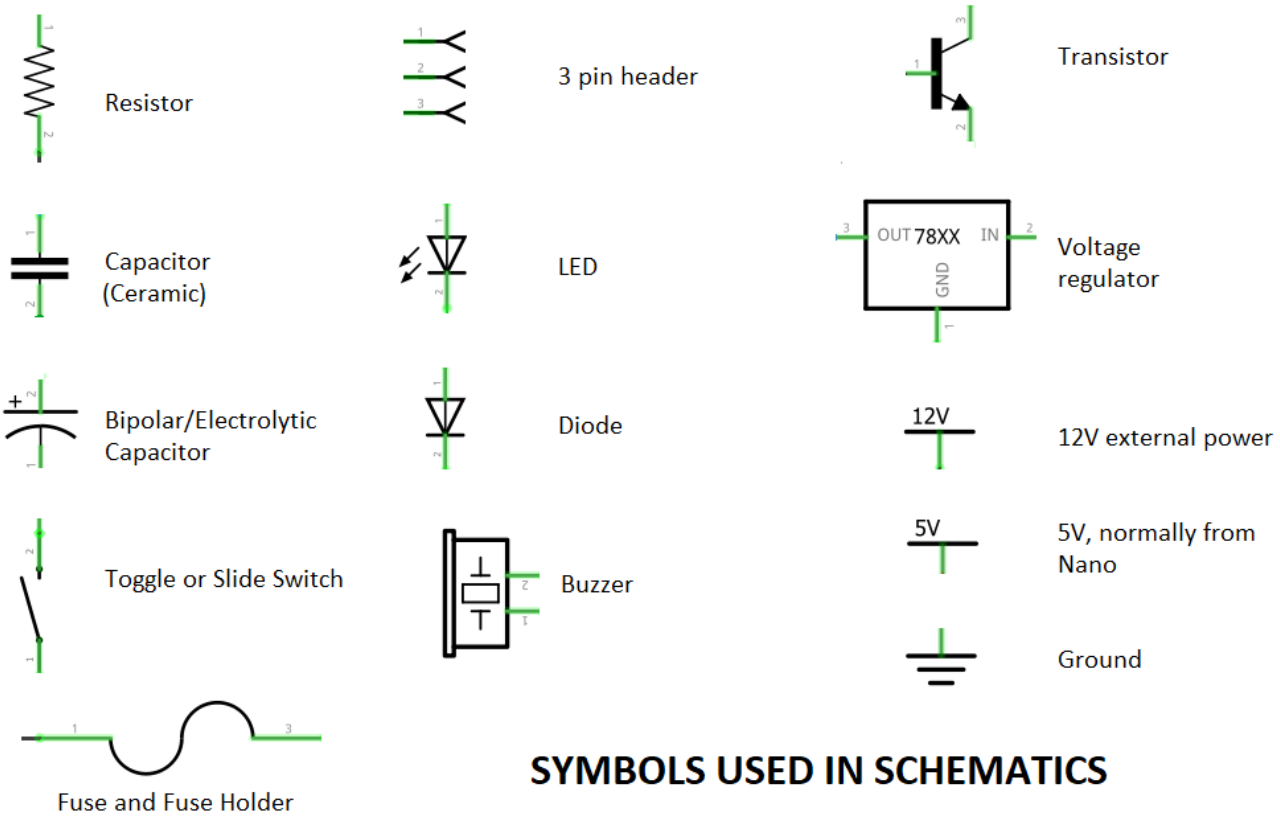
The downside is the time required to train your system, as well as the generated V curve only works for that particular configuration. Change anything like adding a focal reducer or a different telescope or a different camera and you have to start all over and generate new V-curves.

If you want to do automated focusing on DLSR lens that is a different issue. APT is best for this.
Below are photos of a belt drive for a Coronado SolarMax and DLSR which shows you how to attach the stepper motor. You would need to make a suitable bracket. Be aware that focusing a DLSR lens is much more difficult as the steps between focus and out of focus occur over such a small range it is at best, problematic.

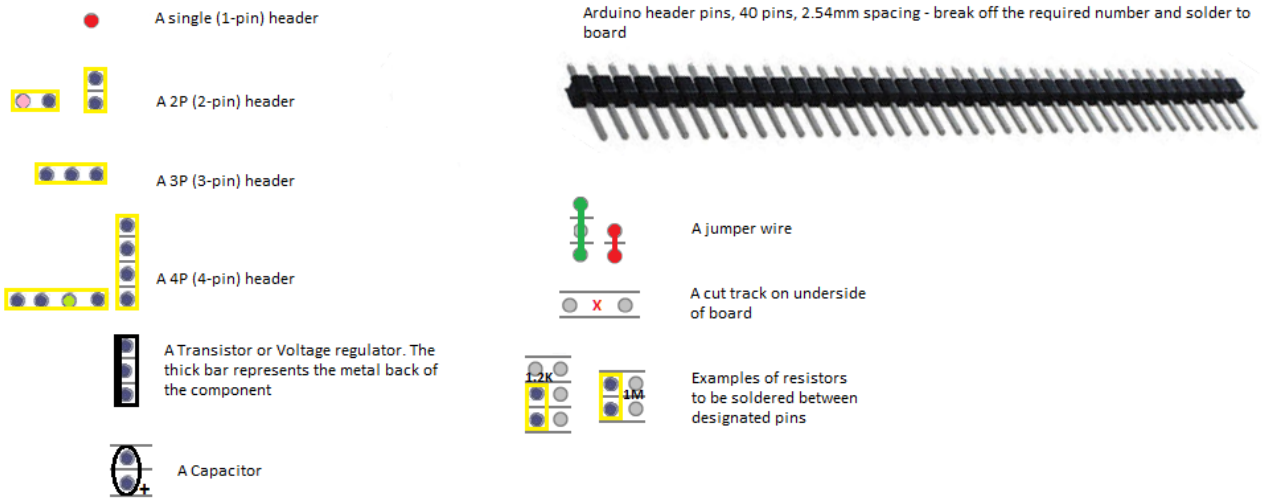
Nema 17 with belt drive as a focuser for a Canon EOS Lens (f2.4)

Nema 17 with belt drive as a focuser for a Coronado SolarMax telescope

SYMBOLS AND MEANINGS USED IN BOARD LAYOUTS

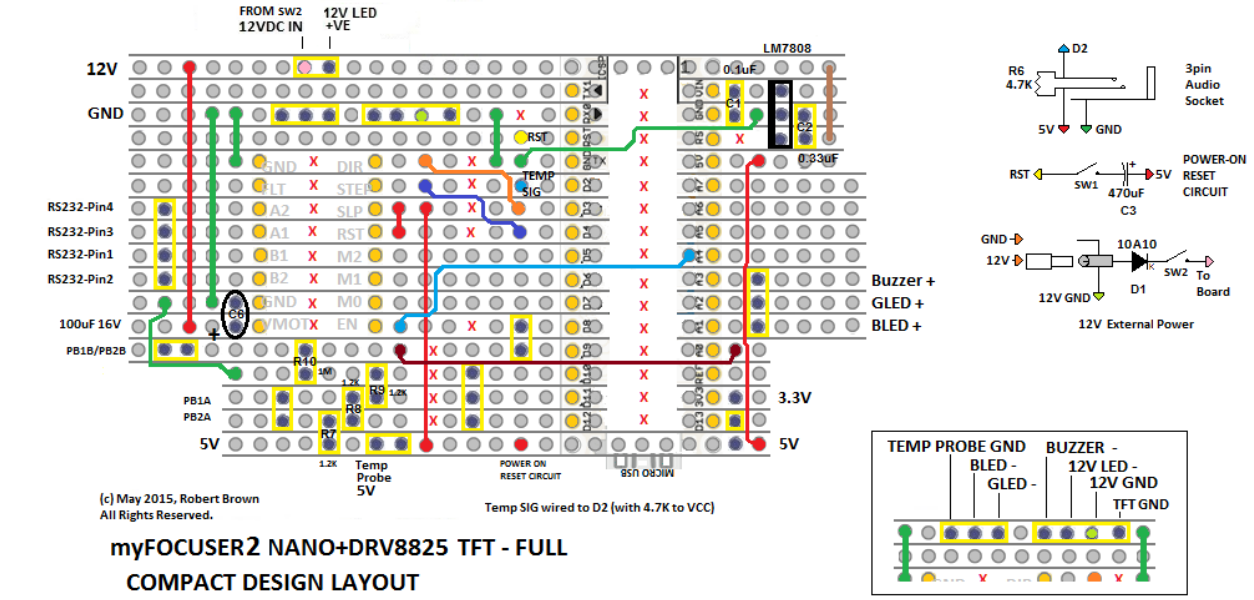


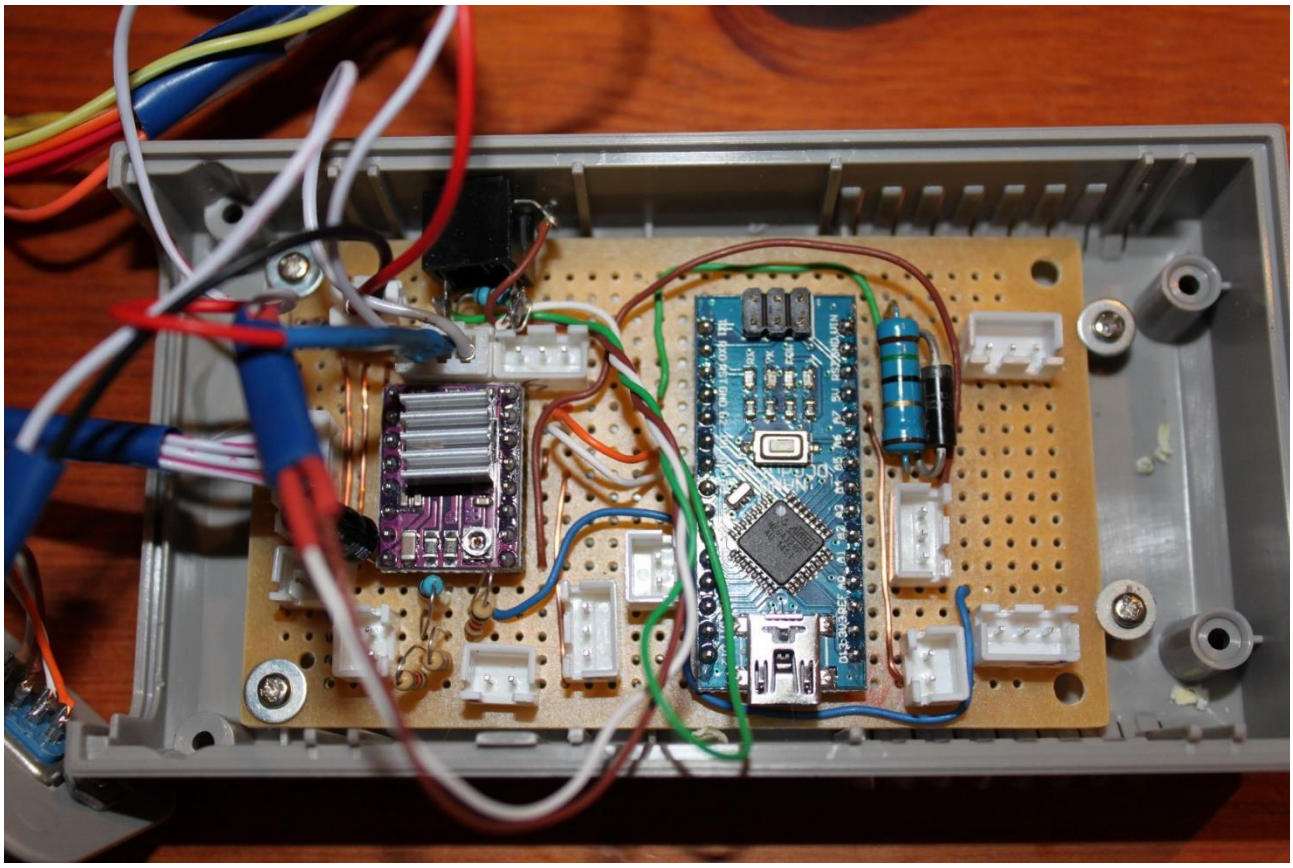
SYMBOLS USED IN SCHEMATICS



STRIP BOARD SYMBOLS USED IN DIAGRAMS AND LAYOUTS

COMPARISON BETWEEN STRIPBOARD AND FINISHED BOARD





Note: This is a previous version stripboard which used a Zener diode regulator which has since been replaced with a LM7808 voltage regulator (top right of board).

APPENDIX A ASCOM SUPPORT

The following lists the ASCOM support provided by myFocuserPro2 ASCOM Driver. The myFocuserPro2 ASCOM driver has passed the CONFORM report.

Property	
Absolute	Implemented
Connected	Implemented
Description	MyFocuserPro2 ASCOM Driver
DriverInfo	Implemented
DriverVersion	Implemented
InterfaceVersion	2
Halt	Implemented
IsMoving	Implemented
Link	Implemented
MaxIncrement	Implemented
maxStep	Implemented
Move	Implemented
Name	myFocuserPro2 ASCOM driver
Position	Implemented
StepSize	Implemented
SupportedActions	returns a NULL list as not implemented
TempComp	Implemented
TempCompAvailable	Implemented
Temperature	Implemented

APPENDIX B TESTING THE myFocuserPro2 CONTROLLER

A number of sample test programs are listed below which assist in verifying correction operation of the controller once built. It is recommended to ensure that the assembled controller passes each one of these tests before the full release program is tried for the first time.

Start at test 1 and run each test in order. Run each program and view the output of the serial port monitor. Compare the operation of the controller against the displayed messages. If any issue is detected (unexpected result), correct the problem first before running any other test. Problems could be shorted tracks on the Vero-board, unsoldered pins, tracks which have not been cut, and wires/components mounted in the wrong place.

All programs use the serial port monitor at 57600bps.

Test2	Test the stepper motor, forward and reverse, 28BYJ-48 and ULN2003 (Build Option 1)
Test4	Test the stepper motor, NEMA17 and L293D Shield (Build Option 2)

BasicDRV8825HW203Test	Test the stepper motor and DRV8825 driver
TestDS18B20Probe	Test the DS18B20 temperature probe
TestLCDI2C	Test the LCDI2C display
TestPB	Test the push buttons

Test programs are available on the Sourceforge site

Testing the Home Position Switch

Note: For the L293D Motor Shield only, all available pins are in use; this build option does NOT support the Home Position Switch

Perform the following tests with the 12V power to the focuser OFF.

You need to be at the telescope to perform this test.

1. Run the windows application and ensure the focuser is somewhere in the middle (at least 500 steps away from the home position).

2. From the settings menu, select Home Position->Check Status, and the focuser will report the status in the Status message text box. The status should say **open**.

3. **If the status message reads closed then the switch has been wired wrong, or the resistor is the wrong value or not connected etc, and you will need to correct this before continuing.**

Assuming that everything is OK and the status is open, then

4. Place you finger on the switch in order to hold it down closed
5. Repeat the Home Position->Check Status test, and this time the focuser should report closed in the Status message text box.

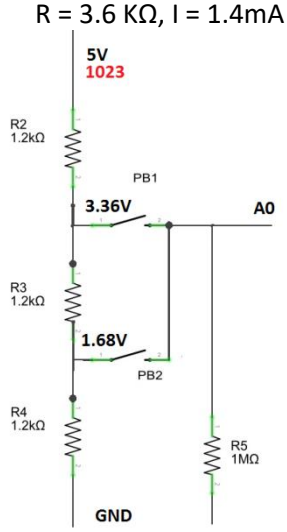
So, the switch should report OPEN when not activated and CLOSED when activated (held down). If this is not the case, you will need to troubleshoot the wiring to ensure that the switch is reported correctly before enabling the code in the firmware.

APPENDIX C DERIVING VALUE RANGES FOR THE TOGGLE SWITCHES ON A0

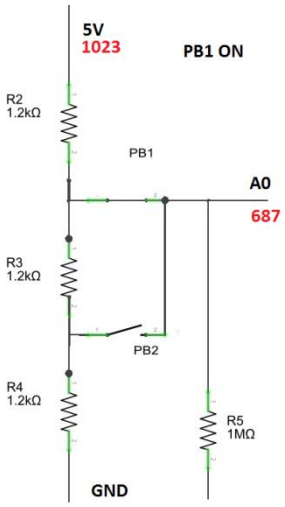
Working out the Toggle Switch Diver Network
Must use 1MΩ resistor from A0 to GND, this would pull it low
The internal 20KΩ pull-up to 5V should return 1023 when no switches on
Try to use 1.2KΩ 1% $\frac{1}{4}$W resistors

Reading of 1023 = 5V

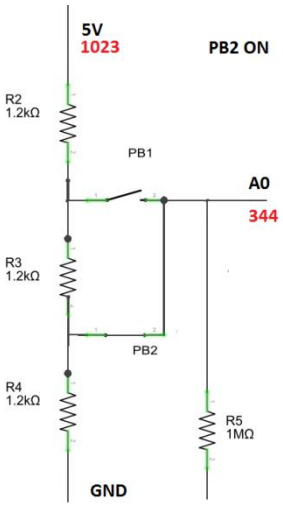
SW1/SW2 both OFF, Total R = 3.6 KΩ, I = 1.4mA



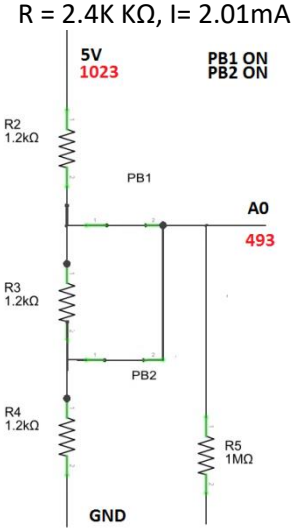
SW1 ON



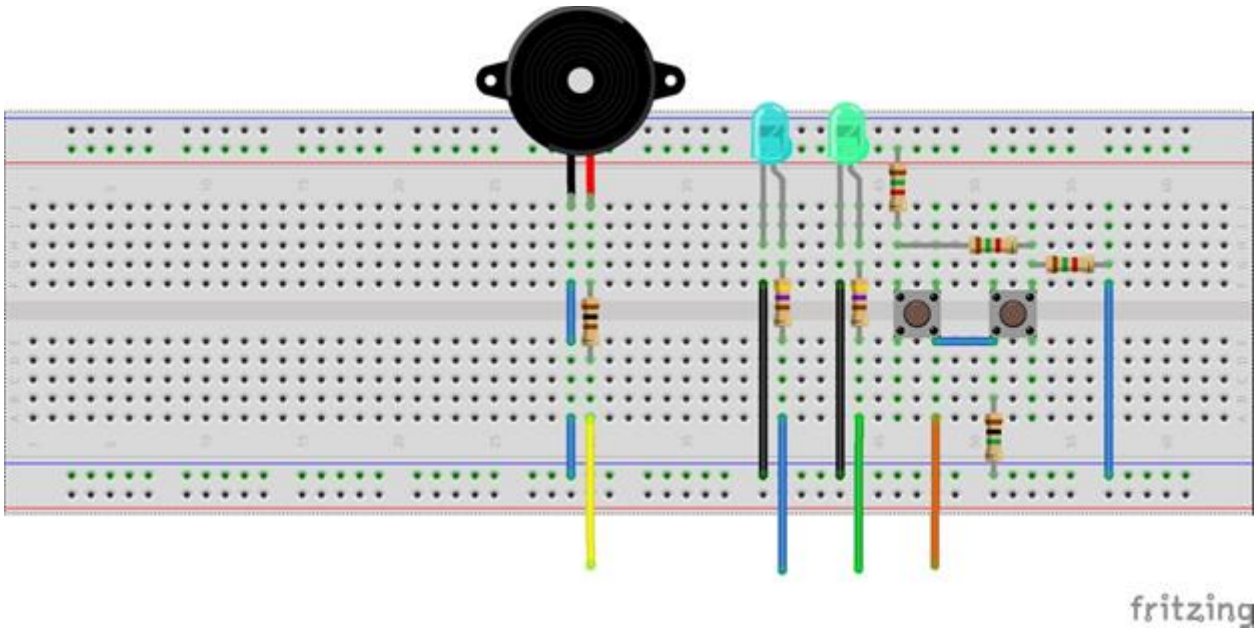
SW2 ON



SW1/SW2 both ON, Total R = 2.4K KΩ, I= 2.01mA



	Predicted	Actual
SW1	687	681
SW2	344	338
SW1+SW2	493	509



Example Breadboard layout for testing the Push Button Switches, LEDs and Buzzer

In general, if you use the specified resistors, you will NOT need to change the code for the push buttons as described below.

The push buttons can be pretty tricky, especially if you have changed the resistor values. It is worthwhile running the test program to ensure that you are getting the correct values.

It has to be wired correctly to work, to A0.

If the push buttons are wired incorrectly then you will get strange results. So best to run a check to see what values are being returned when pb1 is pushed, pb2 is pushed, pb1+pb2 is pushed and when none is pushed.

The Arduino code then uses a boundary check (both sides, like -40 to +40) around each of these values. For example, if pb1 returned 681 then the check for PB1 would look like

```
int readpbswitches(int pinNum) {
        // sw1 (681) 650-720, sw2 (338) 310-380, sw1 and sw2 (509) 460-530
        int val = 0; // variable to store the read value
        digitalWrite((14 + pinNum), HIGH); // enable 20k internal pullup, 14=A0
        val = analogRead(pinNum); // read the input pin
        if ( val >= 650 && val <= 720 ) {
        return 1; // toggle sw1 ON and SW2 OFF

        // other code here

}
```

You need to run the test program to find out what your values are. They should be close to that above if the wiring is correct and you have used the correct value resistors.
// sw1 (681) 650-720, sw2 (338) 310-380, sw1 and sw2 (509) 460-530

If not, then you will need to make changes.

A test program like this will show the values on the Arduino serial port monitor

```
// Test program 1
// requires push button switches, LEDS IN and OUT, Buzzer /
#include <Arduino.h>

// define Push Buttons, use voltage divider network for two push button switches using A0
// use software debouncing
define PBswitchesPin A0 // push button switches wired to A0 via resistor divider network

int PBVal = 0; // holds state of pushbutton switches

// read the push button switches and return state of switches
// 1 = SW1 ON AND SW2 OFF, 2 = SW2 ON AND SW1 OFF, 3 = SW1 ON and SW2 ON, 0 = OFF
int readpbswitches(int pinNum)
{
        // sw1 (681) 650-720, sw2 (338) 310-380, sw1 and sw2 (509) 460-530
        int val = 0;                    // variable to store the read value
        digitalWrite((14 + pinNum), HIGH);      // enable 20k internal pullup, 14=A0
        val = analogRead(pinNum);               // read the input pin
        return val;
```

```
}

// Setup
void setup()
{
	// initialize serial for ASCOM
	Serial.begin(9600);

	Serial.println("Test Program 1A:");
}

// Main Loop
void loop()
{
	// check pushbutton switches
	PBVal = readpbswitches(PBswitchesPin);
	Serial.println(PBVal);
}
```

You then run the program to determine the values when each or both of the push buttons are pressed. These values can then be used to determine the correct boundaries for each push button and the readpbswitches() code modified accordingly.

For example, we run the above test program and get the following values

PB1 = 655, PB2=325, PB1+PB2=492

We add +-40 to each value, giving boundaries for each Push Button of

PB1 = 615-695, PB2 = 285-365, PB1+PB2 = 452-532

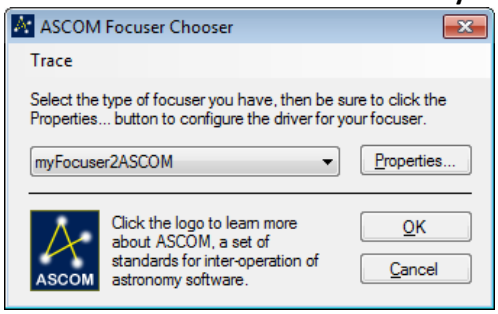
We then use these values to change the readpbswitches() code

```
int readpbswitches(int pinNum)
{
 //	PB1 = 615-695, PB2 = 285-365, PB1+PB2 = 452-532
	int val = 0;                                    // variable to store the read value
	digitalWrite((14 + pinNum), HIGH);               // enable 20k internal pullup, 14=A0
	val = analogRead(pinNum);                       // read the input pin
	if ( val >= 615 && val <= 695 )
	{
		return 1;                               // toggle sw1 ON and SW2 OFF
	}
	else if ( val >= 452 && val <= 532 )
	{
		return 3;                               // toggle sw1 and sw2 ON
	}
	else if ( val >= 285 && val <= 365 )
	{
		return 2;                               // toggle sw2 ON and SW1 OFF
	}
	else return 0;                                  // switches are OFF
}
```

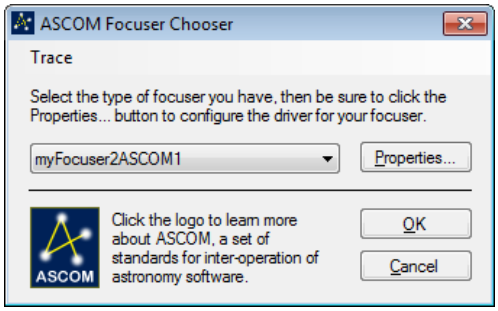
APPENDIX D RUNNING TWO FOCUSERS

The ASCOM driver is not re-entrant so multiple instances cannot be run. This means that to run two myFocuserPro2 controllers on the same computer requires two separate ASCOM drivers.

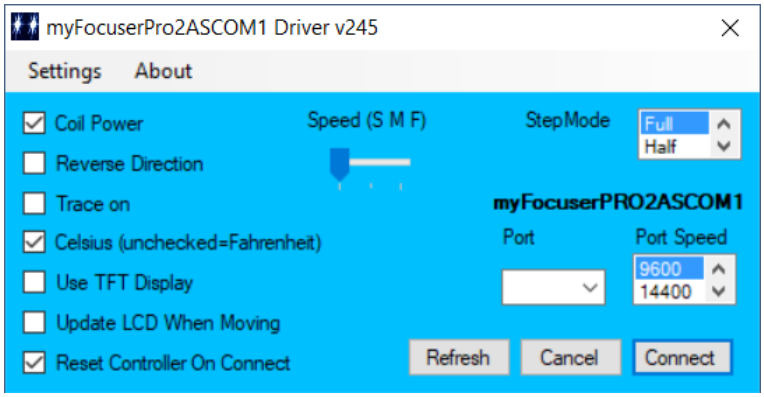
The **first** ASCOM driver is known as **myFocuserPro2ASCOM** in the chooser.



The **second** ASCOM driver (if installed) is known as **myFocuserPro2ASCOM1** in the chooser (and must connect to a different focuser controller than the first driver).



Consider the case where you have a DRV8825 controller on COM PORT3 and a second ULN2003 controller on COM PORT 4. To set this up, you would connect the myFocuserPro2ASCOM to the DRV8825 controller by specifying COM PORT3 under properties, and specify myFocuserPro2ASCOM1 to the ULN2003 controller by specifying COM PORT4 under properties for that driver.



You can then work with both focusers independently.

To install the second ASCOM driver, run the setup program for the second ASCOM driver (myFocuserPro2ASCOM1Setupxxx.exe)

The second ASCOM driver setup dialog box is in a different colour and labelled so you can easily identify it is the second driver as is called **myFocuserPRO2ASCOM1**.

Remember that there are additional settings accessible from the “Settings” menu.

Please see the video
https://www.youtube.com/watch?v=sEvvWYNMCFs

APPENDIX E WHAT ABOUT STEP SIZE?

Step size is the amount in microns that the focuser travels for a single step. The myFocuserPro2 controller supports the implementation of step size (ASCOM driver can return the step size if enabled, else the driver returns a not implemented exception).

Be aware that there can be no common value for this as each implementation is different, depending upon the step mode, stepper motor, gearing and connection to the focuser.

If you have some software that needs step size (like Maxim DL), then you will need to calculate the correct value and then use that value in the software application (like Maxim DL) and specify it for your controller.

You can only calculate the step size once your focuser is fully setup.

Note that if you change the stepping mode then the step size will also change. So, the best thing to do is use one step mode (like half steps) and never change the step mode!

To calculate the step size, position the focuser at say 1000 steps. If your focuser has indicator marks note the position. If the focuser does not have any position marks, try to use an electronic calliper to measure how far out the focuser is and use that position as 0. Now you will send a command to move the focuser outwards 1000 steps from its current position. Once the focuser has moved to the new position, take another measurement, and subtract the first measurement from it.

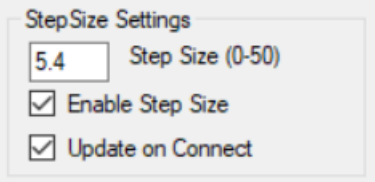
If the first reading was 62mm and the final reading was 87mm, then the distance the focuser actually moved for 1000 stepper motor steps was 25mm. To calculate the step size, divide the distance in microns (to convert a millimetre to a micron multiply by 1000) by the number of steps

25 * 1000 / 1000
= 25000 / 1000
= 25 microns

Note: 1mm = 1000 microns.

The controller implements bounds checking for the value of Step Size, which has been set to Step size > 0 and < 50

The settings for StepSize and whether it is enabled in the controller is specified in the Extra Settings form of the Windows Application or the Settings form for the ASCOM driver.

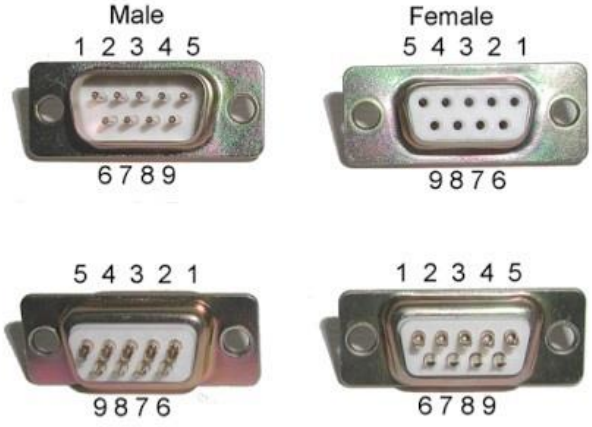


To enable stepsize in the controller, you must specify the step size value, check the Enable Step Size box and check the Update on Connect box before connecting to the controller.

APPENDIX F STEPPER MOTOR TO CONTROLLER CONNECTION

The final version uses an RS232 cable and connectors between the controller and the stepper motor.

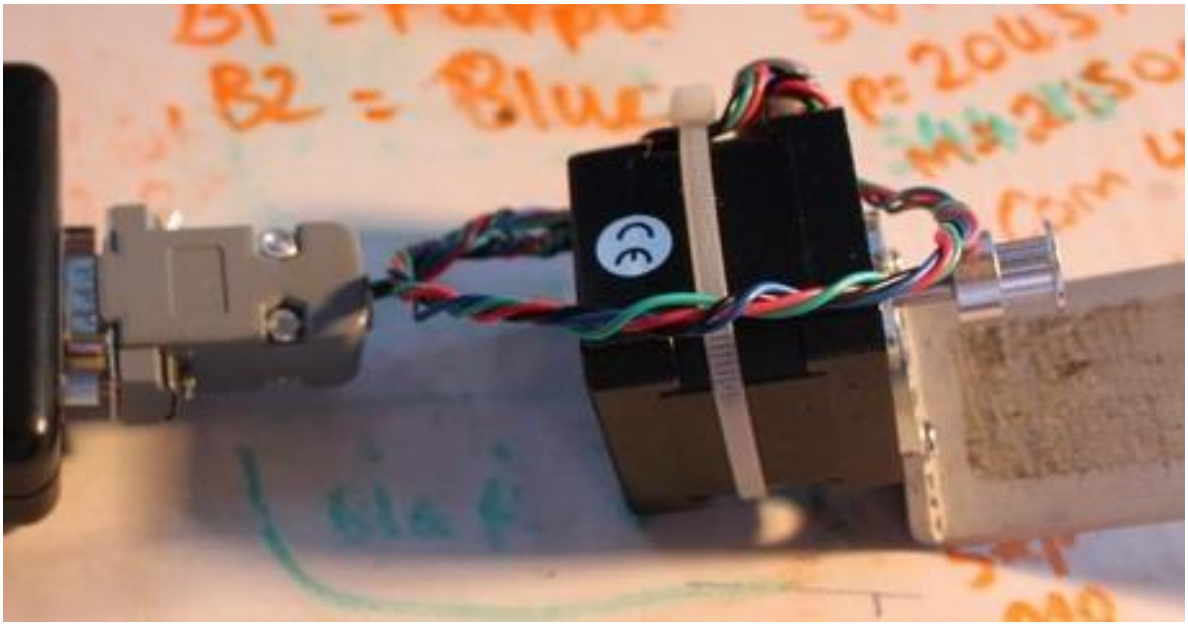
The stepper motor wires are terminated using a RS232 female connector. A TDK ferrite core clip-on cable clamp is clamped over the stepper wires (close to the stepper motor) to minimize back EMF.



On the controller box, the M1/M2 (or M3/M4 in the final version) wires from the L293D shield are wired to a RS232 female socket mounted on the case.

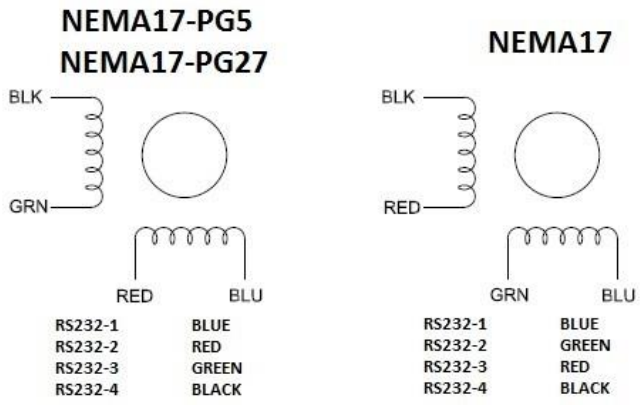
An RS232 cable (male to male) connects the controller to the stepper motor. This means the controller can be mounted off-mount onto a pier.

The maximum length of the cable for reliable operation depends on the characteristics of the cable wire, but should be kept as short as practical for reliable operation.



I have used 6 feet RS232 cables between the myFocuser2Pro controller (mounted on a Pier/tripod) to the stepper motor (mounted on the telescope) without any loss in performance.

WIRING THE NEMA17 COILS TO RS232 CONNECTOR

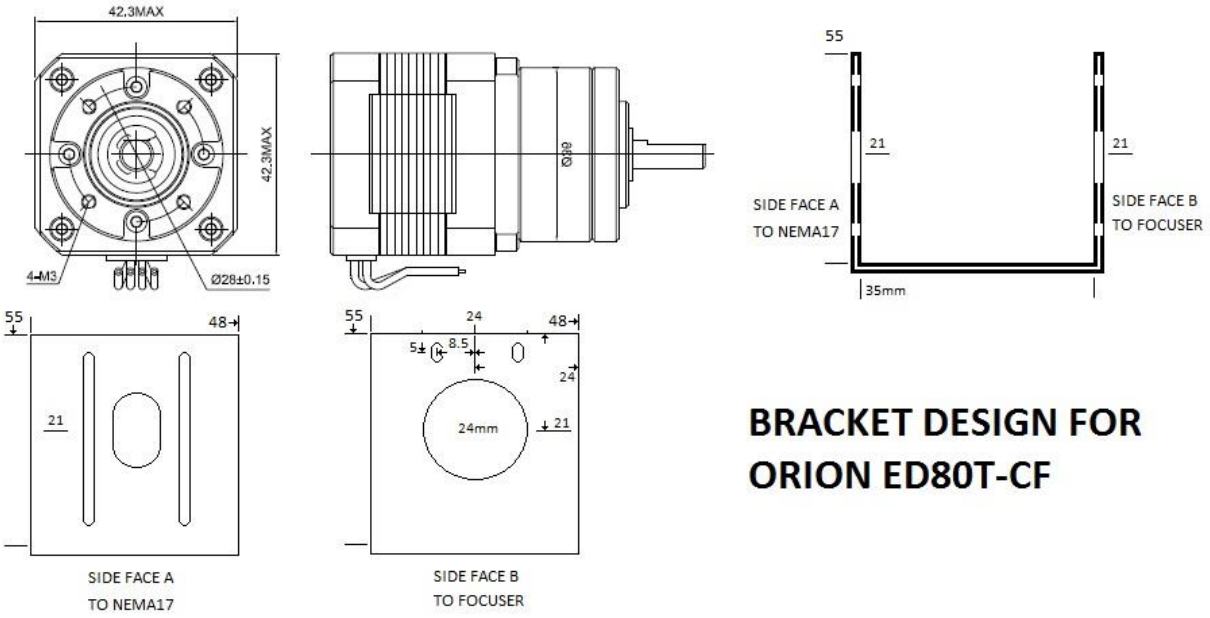


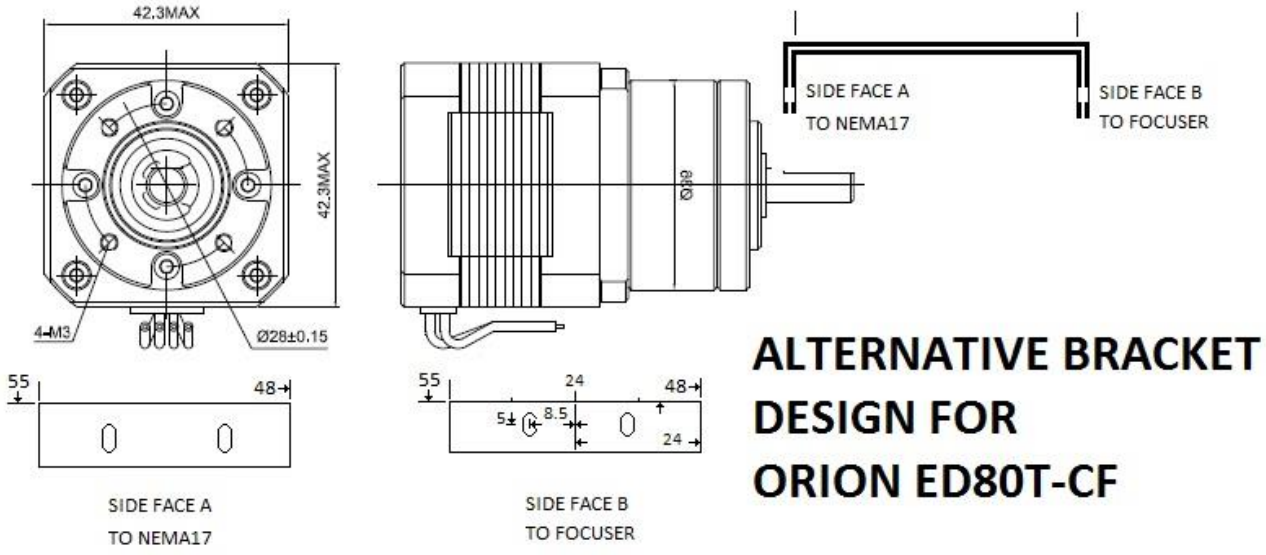
For wiring of the L293D shield to the RS232 connector

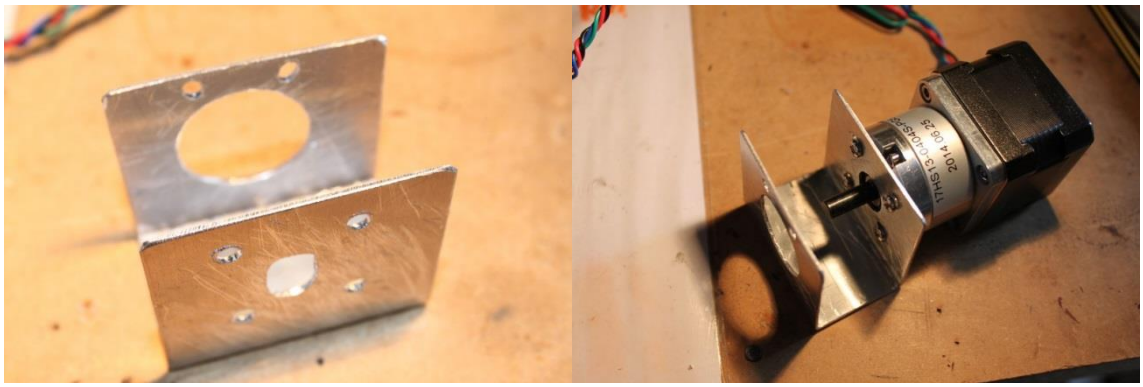
MotorPort-1	**RS232Pin**	**Nema17-PG5**	**28BJY-48**	Stepper Motor -> RS232 DB9pin Female Connector
M2	1	Blue	Yellow	
M2	2	Red	Blue	
GND	5	-	Red	
M1	3	Green	Pink	L293D Motor port -> RS232 DB9pin Female connector
M1	4	Black	Orange	
MotorPort-2	**RS232Pin**	**Nema17-PG5**	**28BJY-48**	To connect the myFocuserPro2 controller to a stepper motor use a RS232-DB9 Male-to-Male cable (straight through extension cable)
M3	4	Black	Orange	
M3	3	Green	Pink	
GND	5	-	Red	
M4	2	Red	Blue	
M4	1	Blue	Yellow	

APPENDIX G NEMA17-PG5 STEPPER MOTOR BRACKET

The NEMA17-PG5 connects via a home-made bracket to the focuser. The following diagram and photos show the U-shaped bracket that is used on the Orion ED80T-CF refractor.







APPENDIX H USING A PULLEY AND BELT DRIVE

A belt reduction drive can be used to connect the stepper motor to focuser or DLSR camera lens. You can also put the belt over the focus knob and drive the knob using the belt.

© Speed_Mart, Pitch 2mm, 6mm wide, GT2 pulley and belt

The above pulley 14T has 14 teeth, the 320-GT2 belt has 160T, giving a ratio of 1:22.857 (eBay supplier speed_smart). This means that using the NEMA17 motor which has 200 full steps per revolution, it will take 4571 steps to rotate the focuser (or lens) ONE complete revolution, which is more than adequate for most setups.

If purchasing the pulley and belt separately, ensure that the pitch of the pulley matches that of the belt.

APPENDIX I WHAT TO DO IF THE LCD PRINTS GARBAGE

In some cases it has been reported that the LCD only prints the first one or two characters of each text or data.

This appears to happen with the Sainsmart HD44780 Controller or equivalent LCD1602 modules. Here is a work-around.

First download the New Liquid Crystal Library by F Malpartida.

Next, install the Library into the Arduino IDE (see Link1 and also Link2). It is recommended to use the Arduino IDE version 1.6.4

Next, load the focuser file into the Arduino IDE by navigating to the folder that contains the firmware file (ends in .ino) and then double click the firmware file

Make the following changes. Find the lines in RED and REPLACE those lines with the lines in GREEN

find this line

```
#include <LiquidCrystal_I2C1602.h>  // needed for LCD1602-I2C
```

and replace with these two lines

```
#include <LCD.h>
#include <LiquidCrystal_I2C.h>      // needed for LCD16020-I2C - Sainsmart Library
```

find this line

```
LiquidCrystal_I2C lcd(0x27, 16, 2);    // connects to A4/A5
```

and replace with these lines

```
#define I2C_ADDR 0x27 // <<----- Add your address here. Find it from I2C Scanner
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
```

find this line

```
LiquidCrystal_I2C lcd(0x27, 16, 2);    // connects to A4/A5
```

and replace with this line

```
LiquidCrystal_I2C lcd(I2C_ADDR, En_pin, Rw_pin, Rs_pin, D4_pin, D5_pin, D6_pin, D7_pin);
```

find these two lines in setup()

```
 lcd.init();      // initialise the lcd display
 lcd.backlight(); // enable the backlight
```

and replace with these lines

```
 lcd.begin (16, 2);
  // Switch on the backlight
 lcd.setBacklightPin(BACKLIGHT_PIN, POSITIVE);
```

```
lcd.setBacklight(HIGH);
```

Compile and download the file to your myFocuser2Pro controller. The LCD should now work correctly. If you still have issues, please post a message on the discussion board providing details of the issues, or email me direct.

APPENDIX J DRV8825 DRIVER BOARD

PRECAUTIONS

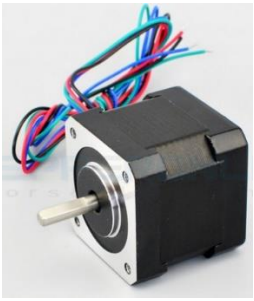
Never disconnect or connect the stepper motor when the Arduino or External Power is ON. This can result in damage to the driver board

BUILD: ARDUINO NANO + NEMA17 BIPOLAR STEPPER 12V 0.4A + DRV8825 DRIVER

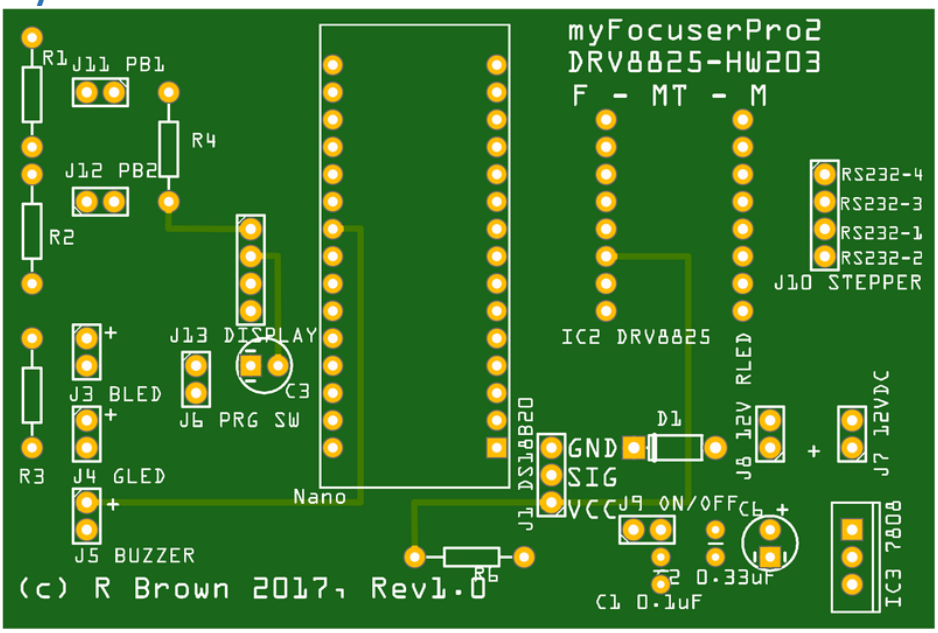
The NEMA17PG27 stepper motor (17HS13-0404S-PG27, 3Nm), purchased from omc-stepperonline, operates on 12VDC at 400mA, and is controlled by the DRV8825 motor driver board. A standard NANO R3 mounted on Vero-board is used. The maximum current draw supported by the DRV8825 is 1.5A continuous (without heat-sink) with the stepper requiring 400mA. The Arduino code version to use with this option is FocuservXXX_DRV8825.ino

The advantage of using the DRV8825 is that higher torque (more current = more torque) stepper motors can be used, as well as using a fairly low resolution stepper motor (200 steps) with micro-stepping (up to 32 times). With maximum micro-stepping of 32, a 200-step motor would give 6400 steps per revolution.

For the recommended NEMA17PG27, most solutions would use FULL or ½ stepping to get the desired resolution (number of stepper motor steps within the Critical Focus Zone).



myFOCUSERPRO2 DRV8825 FRITZING PCB



myFOCUSERPRO2 DRV8825 PURCHASE LIST

Please refer to the separate spreadsheet for a full parts list

myFOCUSERPRO2 DRV8825 CONTROLLER DEFAULTS

Maximum Focuser Position	10000	maxStep
Initial Focuser Position	*5000*	
Maximum steps per move	2048	maxIncrement
Stepper Coil Power	*ON*	
Reverse Direction	*OFF*	
Stepmode	*1 (Full steps)*	

Note: Stepper Coil power means that at the end of the move, when the stepper is stationary, power is either OFF or ON to the coils. If OFF, this saves power, but it might mean that a heavy focuser might start to slip if pointed towards zenith. To prevent this set the Stepper Coil Power to ON. **Because this board uses micro-stepping, Coil Power should be set to always ON**.

Note: Remember NOT to use the Focus Lock Screw on your focuser; leave the screw loose or remove it. The Stepper motor will hold the focuser in place. If you leave the screw in and accidently tighten the focus lock screw, then serious damage can occur to the stepper motor or focuser.

MICROSTEPPING

The DRV8825 board supports micro-stepping which can increase the resolution of the stepper motor (give more steps per revolution).

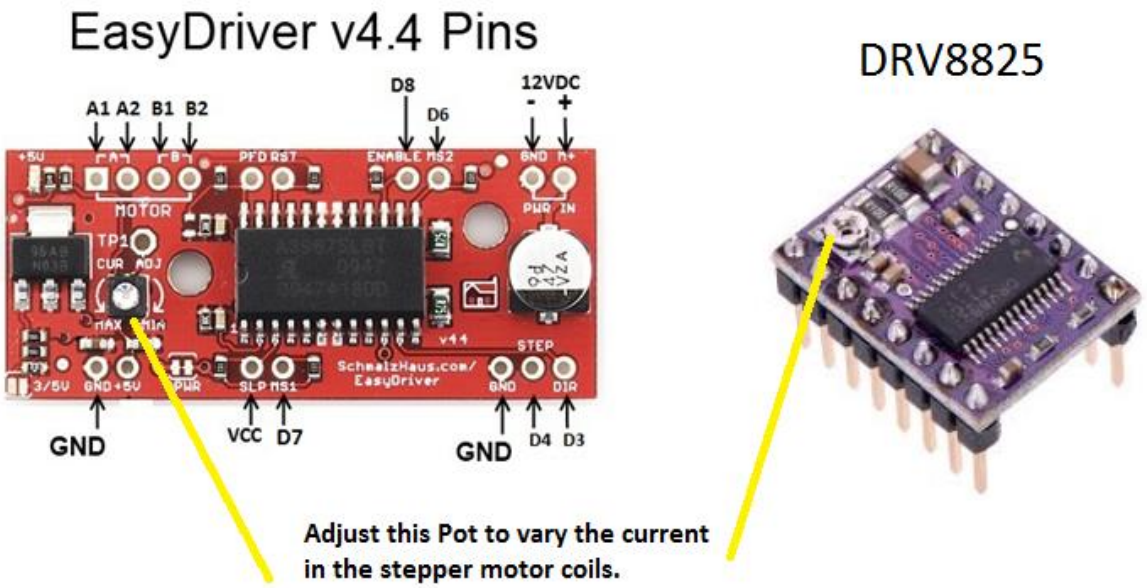
- Micro-stepping capability, Full, half, 1/4, 1/8, 1/16, 1/32, giving 200, 400, 800, 1600, 3200 and 6400 steps per revolution with a NEMA17 200-step bipolar stepper motor

The micro-stepping is controlled by the M0/M1/M2 pins of the DTRV8825 driver board.

HOWEVER – the current limiting potentiometer must be correctly set else the stepper motor may vibrate or miss steps when stepping. For micro-stepping to work correctly, Coil Power should be ON.

ADJUSTING THE POT ON THE DRIVER BOARD FOR CURRENT MAXIMUM

You will need to adjust the POT on the DRV8825 or EASYDRIVER board to get optimal stepping of the stepper motor. This pot adjusts the current that flows in the coils of the stepper motor.



Note1: Adjusting the Stepper Motor by Current

The best way to do this is using a multi-meter.

1. Ensure power is OFF
2. Connect a multi-meter in series with one coil of the stepper motor
3. Set the multi-meter to read current (1A)
4. The NEMA 17 recommended motor uses 400mA. We will use a setting of 350mA, so we need to set the current flowing through one coil to 0.7 * 350mA or 245mA
5. Connect the stepper motor to the controller with the multi-meter in series with one coil
6. Connect the controller to the computer using a USB cable
7. Turn on the external 12V for the stepper
8. Run the windows application for myFocuserPro2, and set the Coil Power to ON
9. With a small screwdriver, slowly turn the pot on the driver board till the current reads 245mA on the multi-meter
10. Disconnect from the controller and exit the application
11. Turn off the 12V external supply for the stepper motor
12. Unplug the USB cable connecting the controller to the computer
13. Remove the multi-meter and reconnect the coil wire as per the original wiring

Note2: Adjusting the Stepper Motor manually

It is best to use a ceramic or plastic screwdriver when adjusting the pot. I would suggest a plastic knitting needle which has the end filed down to look like a screwdriver.

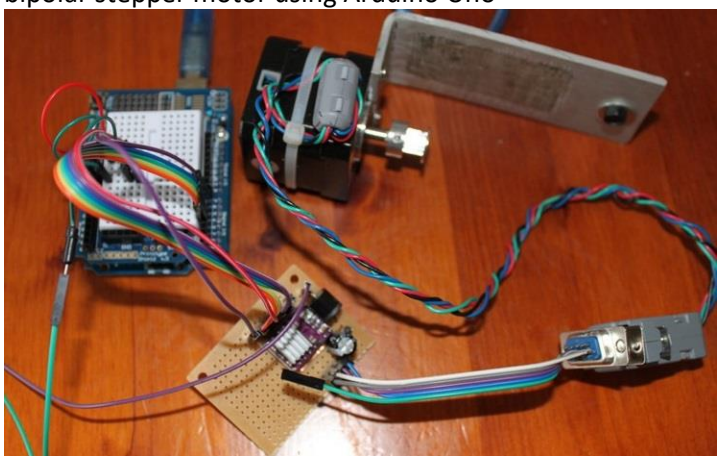
9. With the controller connected via USB, and 12V power to the driver board, set the focuser position to 0 and the Motor Speed to SLOW
10. Enter a focuser position of 5000 and click the Goto button
11. Wind the pot all the way anticlockwise until the motor stops moving

12. Now very slowly turn the pot clockwise until you see the motor start to turn. If the maxSteps is reached, just reset the focuser position to 0 and then type in 5000 for the position and click Goto again
13. Slowly turning the pot, when you see the stepper start to move ok without jerking, then slowly turn no more than 1/8th clockwise from that point
14. It should now be close enough
15. If you go too far then there will be too much current and the motor will run hot. You should use no more than 12V external power

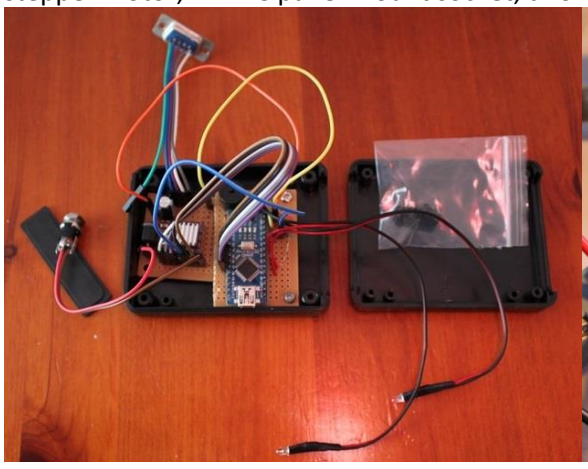
On some driver boards clockwise might be anticlockwise. Once set, then switch to 1/4 stepping and repeat the 0 then 5000 Goto. The motor should run smoothly without missing steps (a missed step will be a sudden jerk which you will be able to feel or hear). If there is any of this, you might need to ever so slightly turn it a little more. Be careful as a little turn can make significant changes in current.

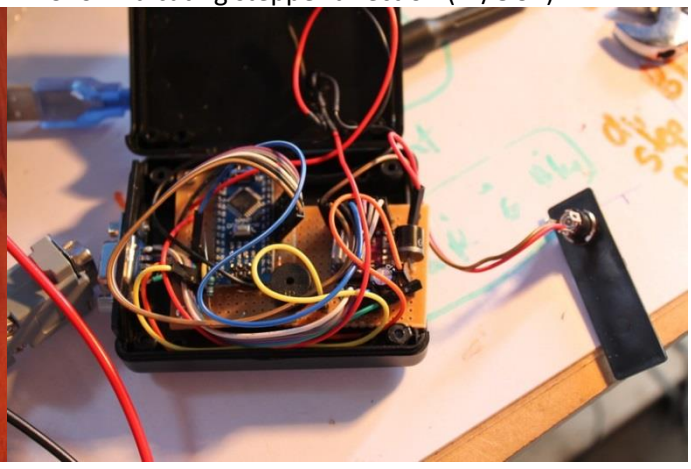
myFOCUSERPRO2 DRV8825 PROTOTYPE BUILD PICTURES

Initial prototype and breadboard (DRV8825 sub-board) to test DRV8825 driver board with NEMA17 hybrid bipolar stepper motor using Arduino Uno



Arduino Nano, DRV8825 sub-board housed in small plastic case (minimal solution) and RS232 connector for stepper motor, 12VDC panel mount socket, two LEDS for indicating stepper direction (IN/OUT)





Finished DRV8825 controller

WIRING THE DRV8825 BOARD TO RS232 CONNECTOR

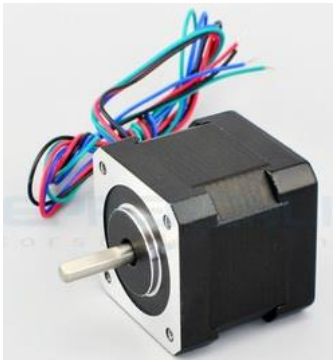
The DRV8825 driver board has four outputs for driving a bipolar stepper motor. These are connected as shown below

NEMA17 Hybrid	RS232 Connector Pin	DRV8825 Board
BLUE	1	A1
GREEN	2	A2
RED	3	B1
BLACK	4	B2

APPENDIX K myFocuserPro2 used with Canon EOS Lens

This article discusses how the myFocuserPro2 can be used to provide a focusing solution for a CANON EOS Telephoto lens (which could be connected to a DLSR or Astro-imaging camera.

A toothed belt system is used to connect the myFocuserPro2 unit to the Canon EOS lens. The focuser is using a minimal build of Arduino Nano + DRV8825 driver board and NEMA17 bipolar stepper motor (200 steps).



The NEMA17 stepper motor is fitted with a 14-tooth gear and drives a 6mm wide with 2mm pitch toothed belt. The following table shows the gearing ratio.

Gear Belt	Length (mm)	Pitch (mm)	Teeth	Ratio
320-2GT	320	2	160	22.857:1 (with 14T pulley on motor shaft)

200 Full steps @ 27:1 ratio = 4571 steps per one revolution of the Canon lens

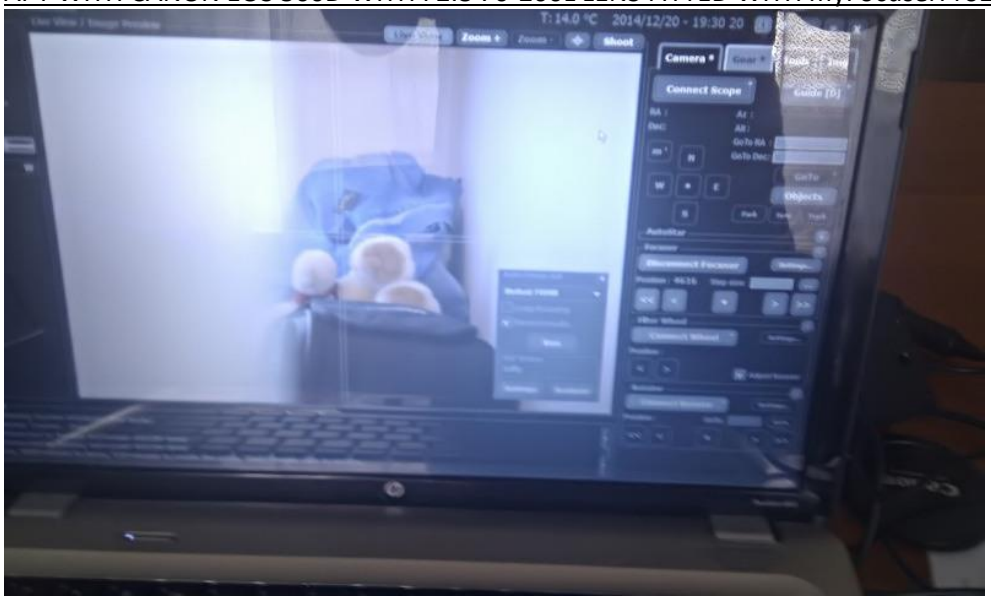
Using half-stepping with the 320mm belt thus gives 9142 steps per revolution of the Canon EOS Lens (each step = 0.039 degrees)

The NEMA17 stepper motor is attached to an L shaped bracket and fitted with the 14T gear. The stepper is driven in this case by the DRV8825 controller (minimal solution). The following photo illustrates how the brackets and stepper motor are positioned with the toothed belt relative to the Canon EOS lens.

In the first instance, the toothed belt is kept loose and the lens is rotated manually till the object is very close to focus. Now the brackets are adjusted so that belt is tightened.

The myFocuserPro2 application is started. As the focuser is already near focus, any fine adjustment of focus will be relatively small, so the initial focus position is set to 5000, maxStep set to 10000 and half-steps enabled. This should provide more than enough steps to achieve good focus. Focus is then achieved by moving the myFocuserPro2 controller IN or OUT as required.

APT WITH CANON EOS 500D WITH F2.8 70-200L LENS FITTED WITH myFocuserPro2 DRV8825 and NEMA17



Focusing done in LIVEVIEW WITH ZOOM and ASCOM focuser jogged till best focus achieved.

APPENDIX L NEMA17 STEPPER MOTORS AND VIBRATION

In general, the NEMA17 stepper motors should cause little or no vibration concerns when used to control the focuser.

There should be no need to be concerned about vibration when using the recommended setups. However, if using different stepper motors or drivers, even though unlikely, this might become an issue.

Vibration will increase with stepper motor speed. Final fine focusing should be done at the lower speed whilst coarse focusing can be done at medium or high speed.

In addition, direct coupled stepper motors will cause more vibration than a belt system as the rubber flexible belt acts as a vibration damper.

In micro-stepping modes, where coil power must be ON to hold the stepper at the required position, this can sometimes result in the stepper motor pulsing or vibrating between two rotor positions.

If vibration is a concern, then the mounting of the motor must be isolated from the telescope through some form of damper so that the motor vibration is not transferred to the telescope through the mounting bracket. Rather, the motor vibration is absorbed by the damper which fits between the motor and the mounting bracket.

It is important to note that using something like cork between the motor and mounting bracket as a damper will NOT work. This is because vibration will be transferred via the mounting screws that connect the motor to the mounting bracket. Even the screws must be isolated.

There are a number of available NEMA17 dampers available at reasonable cost.

These do a very effective of completely isolating the motor from the mounting bracket (including screws), and are essentially two plates separated by rubber.

The motor screws onto one plate whilst the other plate is screwed to the mounting bracket. Ensure that the length of the screws are not excessive or can touch anything else.

These can be purchased online from robotdigg for about $3USD each and the part number is 17DAMPER

I have found NO issue with stepper motor vibration in the various set-ups and configurations I have tried.

APPENDIX M TEMPERATURE DS18B20 PROBE PRECISION

The resolution of the temperature sensor is user-configurable to

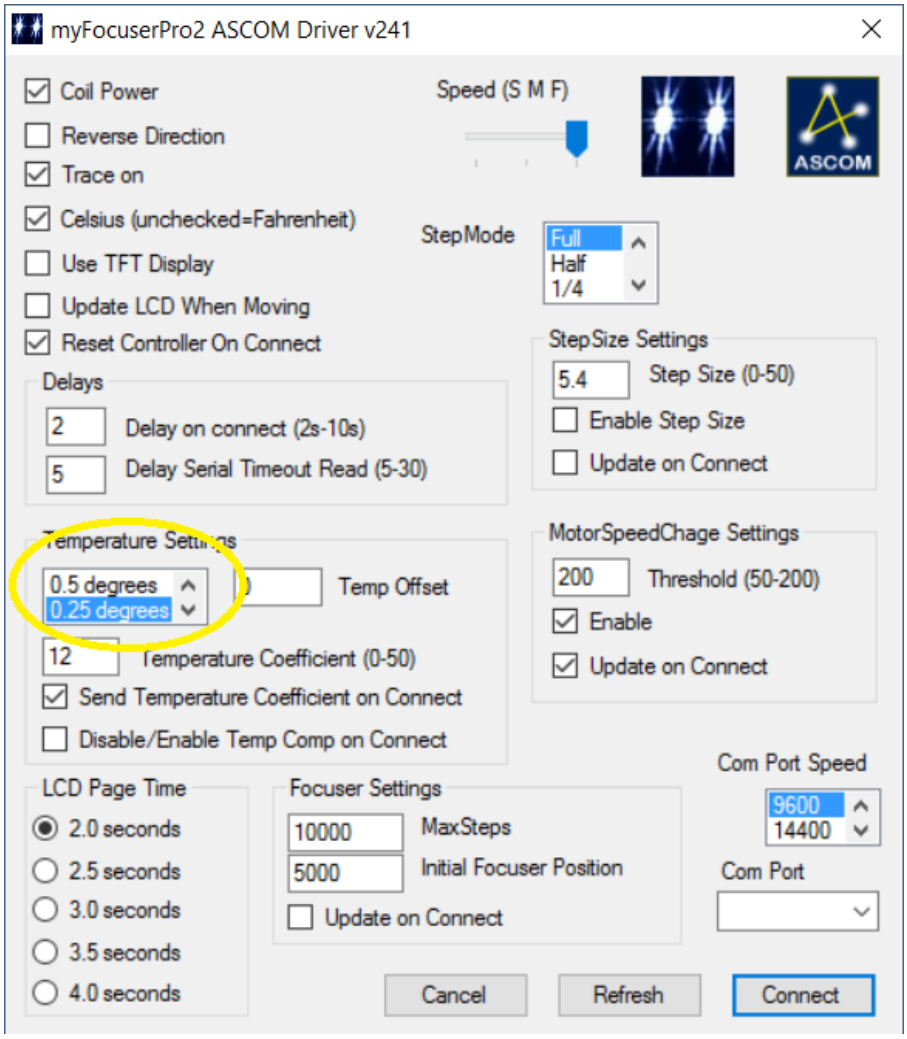
9	0.5°C
10	0.25°C
11	0.125°C
12	0.0625°C

The default resolution at power-up is 12-bit. Each increase in resolution takes longer to convert/read and the Arduino firmware code in the controller uses the following code fragment to wait a specified time between requesting a temperature reading to reading the result.

```
delay(750 / (1 << (12 - TEMP_PRECISION)));          // should enough time to wait
```

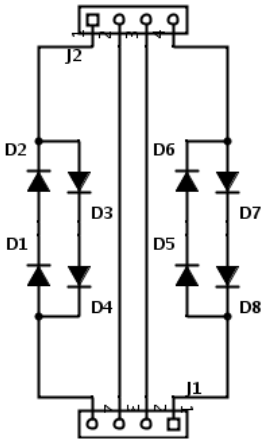
Both the Windows application (via the Settings, Temperature Precision menu) and ASCOM driver (on the main Setup dialog form) lets the user specify the temperature precision when connecting to a myFocuserPro2 controller.

The temperature precision value is saved in EEPROM of the myFocuserPro2 controller, thus is remembered from session to session.



APPENDIX N SMALL STEP MODES AND DIODES

A big thanks to Ken who kindly researched and provided this information.



When using DRV8825 at high stepping modes like 1/8 or smaller, the waveforms to the stepper can cause it to miss steps. The pulses tend to become more like a sinewave than a square-wave. To restore the waveform pulses a diode network can be used.

This rounding of the pulses can be overcome using pairs of diodes which are wired in series with the coil pairs of the stepper motor. The diodes help reshape the waveform going to the motor.

The diodes used are 1N5404.

Original Article

http://cabristor.blogspot.co.nz/2015/02/drv8825-missing-steps.html

An alternative would be to use 1N5822 Schottky diodes which have a similar rating but react faster and have a lower forward voltage drop than the 1N5404 diodes.

Or you can purchase the TL-Smoother boards from eBay which do the same thing.