myFocuserPro

01-September-2018

A DIY project - stepper motor focuser solution based on Arduino

This document describes
- the range of myFocuserPro products
- building the myFocuserPro controller units
- testing procedures and programs
- attaching the myFocuserPro stepper motor to the telescope focuser
- sample schematics, strip-board layouts and PCB (Eagle)
- how to initially setup the myFocuserPro controller
- determining the correct myFocuserPro settings
- what to do if you lose your settings
- the operation of Windows applications and ASCOM drivers available

Restrictions

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

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. V2 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 software with a V1xx programmed controller or vice-versa.

Both systems are supported by a Windows application and ASCOM drivers.

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

CONTRIBUTIONS

If you wish to make a small contribution in thanks for this project, please use PayPal and send the amount to user **rbb1brown@gmail.com (**Robert Brown). All contributions are gratefully accepted.

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.

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.

1. You cannot sell kitsets or assembled units to others
2. You cannot copy the design and code and make your own version to sell to others, even if you make some code changes or change pin designations
3. You cannot use these designs and code ideas to make your own focuser and sell versions of those to others

OVERVIEW

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

myFocuserPro SPECIFICATIONS

- myFocuserPro ASCOM driver, tested with FocusMax, Nebulosity, APT
- myFocuserPro Windows Application (and Mini version)
- Optional LCD1602/I2C display for positional information (Current and Target positions)
- Optional push buttons (x2) for manual control of IN and OUT with manual zero position (hold down both push buttons for 2 seconds or reset in software)
- LED indications for IN, OUT and External PWR
- Absolute focusing
- External power supply required for driving stepper motor 7.5-12VDC @ 2A
- Reverse voltage protection on external 12VDC input
- Multiple versions available. Two versions detailed in this document – for others see addendums
  - 28BYJ-48 Stepper Motor and ULN2003 driver board using Arduino Nano
  - NEMA 17HS15-0404S-PG5 Stepper Motor and L293D Motor Shield using Arduino Uno
- Professional PCB available on-line, double sided, plated through holes, silk masked

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. As the focal ratio gets smaller (a faster telescope optic) the CFZ reduces requiring a higher resolution stepper motor (more steps per revolution).

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 figure can then be 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 this value.

StepSize is supported. Some applications may require a valid setting for StepSize in order to work correctly. If enabled, then the StepSize value stored by the controller will be returned. If StepSize 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 the step size when connecting to the controller, and these settings are saved.

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

Spreadsheet Calculations Showing Various Options

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

SV102T with NEMA17PG5 STEPPER MOTOR

usbFocus			SVFocuser
Degrees Per Step		mm per myFocuser Step	mm per 360 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**

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

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

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.

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 stepsize (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 critical zone we require a stepsize of around

SS = 220 / 10
SS = 22

thus a SS 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.

Proceeding on that basic 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 is 1028 steps, 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 we have 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

- 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).

If you decide not to use the recommended motors suggested here, then make sure that the current rating of your stepper motor (x2) does NOT exceed that of the driver board you have selected. For example, if you selected a 1.4A stepper motor, then the current draw when both coils are energized is 2.8A, which exceeds most of the driver boards listed here.

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

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.

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
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

Use the NEMA17PG27 stepper motor [17HS13-0404S-PG27].
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 SOFTWARE TO USE

Board	Driver	Full?	Stepper	Stepping	.ino file
Nano	ULN2003	F	28BYJ-48	F	Focuserv1xx_ULN2003_F
Nano	ULN2003	M	28BYj-48	F	Focuserv1xx_ULN2003_M
Uno	L293D	F	Nema17PG5/PG27/28BYJ-48	F/H	Focuserv1xx_L293D_F
Nano	DRV8825	F	Nema17PG5/NEMA17PG27	H/H/4/8/16/32	Focuserv1xx_DRV8825_F
Nano	DRV8825	M	Nema17PG5/NEMA17PG27	H/H/4/8/16/32	Focuserv1xx_DRV8825_M
Nano	EasyDriver v44	F	Nema17PG5/NEMA17PG27	F/H/4/8	Focuserv1xx_DRV8825_F
Nano	EasyDriver v44	M	Nema17PG5/NEMA17PG27	F/H/4/8	Focuserv1xx_DRV8825_M
Nano	L298N	M	Nema17PG27	F	Focuserv1xx_L298N_M
Nano	**DRV8825/HW203**	**All**	**Nema17PG27**	**F/H/4/8/16/32**	**Focuserv1xx_DRV8825_HW203_xx**
Nano	ST6128/HW203	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuserv1xx_ST6128_xx
Nano	RAPS128/HW203	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuserv1xx_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	myFocuserProASCOMSetupxxx
WINDOWS APPLICATION	myFocuserProWin_xxx

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

NOTE: The recommended method to reset the MaxSteps 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.

Board Type	Order
hw203_DRV8825_FIRBT	https://go.aisler.net/p/KQROKYSY
DRV8825HW203_FRE	https://go.aisler.net/p/KNIUPUGX
DRV8825HW203_F	https://go.aisler.net/p/MDZYKHMA
EASYDRIVER-HW203_FRE	https://go.aisler.net/p/BRFHRDTI
EASYDRIVER-HW203_F	https://go.aisler.net/p/GIWHTENQ
A4998_MT	https://go.aisler.net/p/OQXFVIMV
L298N_F	https://go.aisler.net/p/IGNDSKNR
ULN2003_F	http://aisler.net/p/UXHJUIIE

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

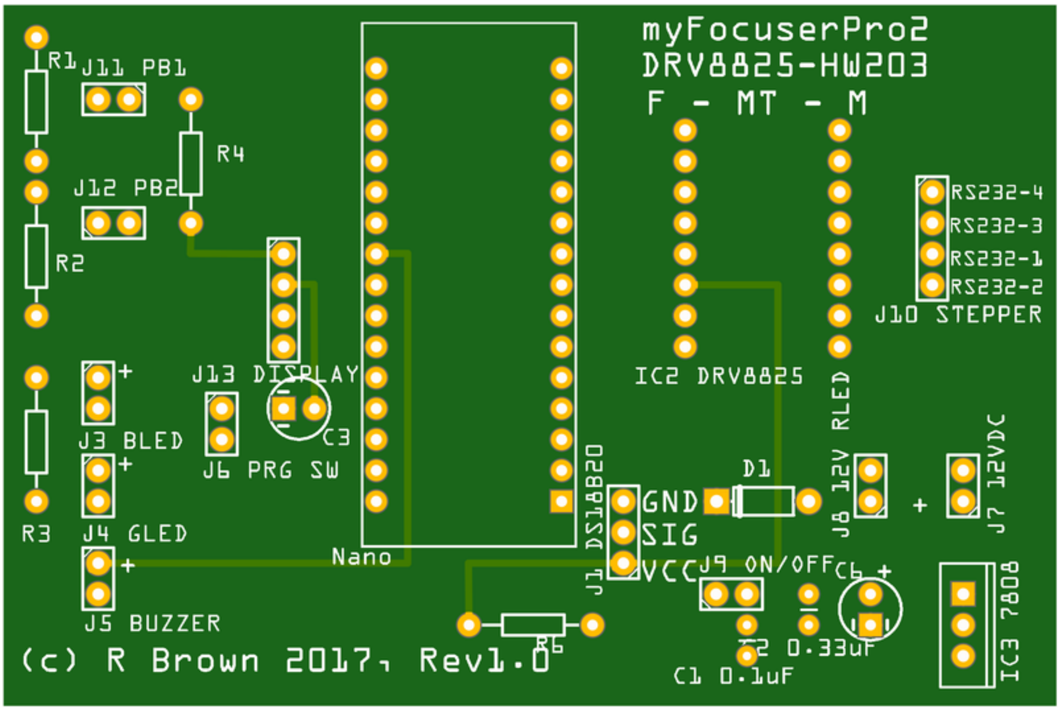
- Minimal
- Minimal plus temperature probe
- Full including display, push buttons

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 DRV8825-HW203-F PCB



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.

myFocuserPro BUILD OPTIONS

The controller comes in a number of build options (the options differ on Arduino board and motor driver board). The ASCOM driver and application software supports ALL options. The correct Arduino code version must be used with the correct option.

The most recent DIY PCB/Stripboard is DRV8825_HW203_F which has a common layout that supports most options.

In other words, you can use the same board 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 affects the choice of stepper motor.

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

myFocuserPro 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** $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$	**NEMA17-PG5/PG27**
EasyDriver	500mA	750mA	12V	30	F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$	NEMA17-PG5/PG27
L298N	2A	3A	12V	35	F $\frac{1}{2}$	NEMA17-PG27
A4988	1.5A	2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$	NEMA17-PG5/PG27
ST6128	< 2A	2.2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	NEMA17-PG5/PG27
RAPS128	< 2A	2.2A	12V		F $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	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_XX
L239D Shield[2]	Focuservxxx_L293D_F
DRV8825_HW203[3]	**Focuservxxx_DRV8825_HW203_XX**
EasyDriver	Focuservxxx_DRV8825_HW203_XX
L298N	Focuservxxx_L298N_XX
A4988	Focuservxxx_A4988_HW203_MT
ST6128	Focuservxxx_ST6128_F
RAPS128	Focuservxxx_RAPS128_HW203_F

[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.

Driver Board Option PDF
A4988
DRV8825-HW203
EASYDRIVER-HW203
L298N
RAPS128
ST6128
ULN2003
UNO + L293D MOTOR SHIELD
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). Please see the Libraries folder for further information.

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

Focuserv168_DRV8825_HW203

The version number is 68, the driver board is DRV8825 with the HW203 layout, and the firmware supports a display, temperature probe and push buttons with BT control via Blue Tooth. To enable an option requires configuring the firmware file to enable or disable options at compile time (when generating and uploading the firmware to the controller)

The available firmware files are.

```
<Focuserv172_A4998_HW203>                                   <DIR> 2018-02-06
<Focuserv172_DRV8825>                                       <DIR> 2018-02-06
<Focuserv172_DRV8825_HW203>                                 <DIR> 2018-02-06
<Focuserv172_DRV8825_HW203_IR>                              <DIR> 2018-02-06
<Focuserv172_DRV8825_HW203_RE>                              <DIR> 2018-02-06
<Focuserv172_L293D>                                         <DIR> 2018-02-06
<Focuserv172_L298N>                                         <DIR> 2018-02-06
<Focuserv172_RAPS128_HW203>                                 <DIR> 2018-02-06
<Focuserv172_ST6128>                                        <DIR> 2018-02-06
<Focuserv172_ULN2003>                                       <DIR> 2018-02-06
```

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. New firmware versions 254 or greater for the L298Mini driver board, the L9110S driver board, and the L298N driver board now support half-stepping.

If you have an f6 or f7 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
- Using half steps reduces the available torque by 30%
- 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 shield cannot readily support higher currents than 400mA continuous
- The AF_MOTOR library supports SINGLE and DOUBLE parameters when stepping the motor, with DOUBLE resulting in more torque. The controller code uses DOUBLE
- 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
- 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 could have enough weight to move or slip either during a move command or once the move command is finished.

myFocuserPro PURCHASE LIST

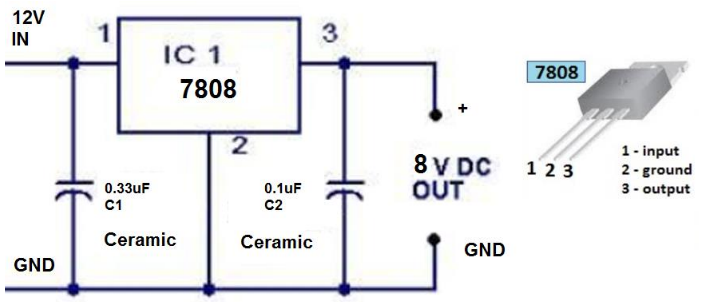
Please refer to the* spreadsheet *for the required parts.

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.

Using a NEMA stepper motor requires a separate voltage supply to the driver board, which can 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.

The NEMA stepper motor requires 12V. It is logical to use the same external 12V to power the Nano controller. This means supplying a voltage to the VIN pin on the Nano. 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.

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.

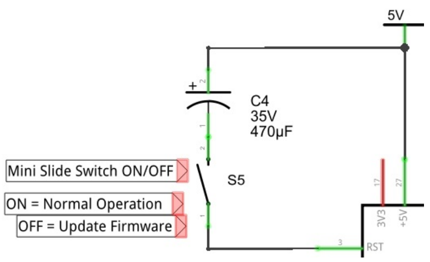
Use an external power supply of 9V for the ULN2003 driver board, and do not fit the LM7808 regulator. Apply the 9V direct to VIN

Use an external power supply of 12V for all other driver boards

POWER ON RESET PREVENTION CIRCUITRY

The following 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, applications like Moonlite drivers, APT, Focusmax and other applications/drivers attempt to restart the controller so it starts in a known state. The controller can take up to 3s to respond after a reset, during which time the application can time out and not connect to the controller. To overcome this timeout, the reset of the Nano on a serial connection request must be prevented. If you wish to prevent this reset when the software connects to a controller, the power-on reset prevention 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 AL". This board does NOT use an FTDI chip so there will be no issue powering the board from 12VDC via VIN and a LM7808 voltage regulator.

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

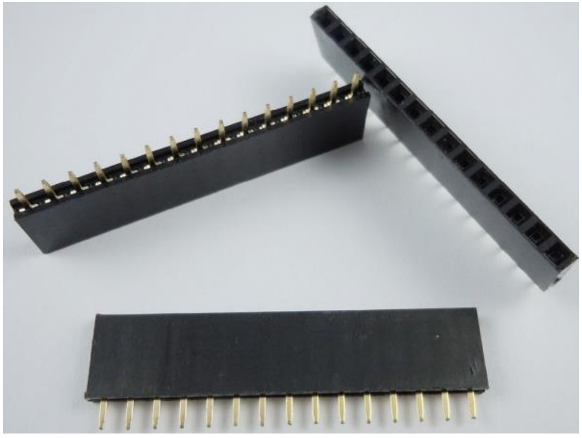
Nano chips such as the FTDI FT232RL chip is only powered if the board is being powered over USB. As a result, these chips when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

This means if we use a Nano FTDI chip we cannot connect VIN to 12VDC as we would lose the FTDI RS232 connection. If so, we would have to use an RS232 connector wired direct to digital pins 0 (RX) and 1 (TX) rather than use the mini USB cable. It would mean having to place a connector on the side of the enclosure, wire it to pins 0 and 1 of the Nano, and install a RS232 cable back to a COM PORT or RS232-USB adapter. So it is best not to use a Nano chip which has an FTDI driver.

The CHSER board as recommended for this project, that does not use the FTDI chip, this limitation is removed. It is strongly recommended to use this board.

Note: 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

EEPROM USAGE

The controller remembers

- Focuser position setting
- MaxSteps (maximum focuser position setting)
- StepMode
- Reverse Direction
- Coil Power
- Refresh rate of display – how long each LCD page is displayed
- Refresh rate of display when focuser is moving-

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 controller code, on power up or reset, checks the EEPROM for the last saved position and the setting for MaxSteps. 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 under the following conditions

- after a MOVE command <u>AND</u> when the focuser is 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 unlikely that you will wear out the contents of the EEPROM. If you image every night of the year, the EPROM should last 7 years of continual use. At once a week, the EEPROM should last about 44 years.

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
- **Local/Remote Computer**
  The controller operates on External Power (7.5-12VDC) and the mini-USB cable connects the controller to a computer. The computer can send serial commands to the controller to alter parameters such as position and command the stepper to move in and out.
- **Using the ASCOM driver OR the Windows application to control 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.

It is recommended to use version 1.6.8 of the Arduino IDE.

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.

If the focuser is moving in the wrong direction when sending a position command (such as -10 or -100 actually moves the focuser OUT instead of IN) from the Windows Application, then enable Reverse Direction to ensure the focuser moves in the correct direction.

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.

The switches are implemented using a voltage divider network and 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.

12mm Waterproof Lockless Momentary Push button Mini Round Switch

Instead of push buttons, you could use a Rotary Encoder (Keyes 040 supported) or an IR Remote Control. Please read the PDFs in the Driver Boards folder for more information on these options.

Note: If the momentary switches are NOT implemented and you are using a FULL version of the firmware (denoted with extension _F), then it is important to remove the push button switch code from the Arduino firmware otherwise the controller will not function correctly. Alternative you can use the Minimal Arduino code version which does NOT support the LCD, push-buttons or the temperature probe.

LCD1602/I2C DISPLAY (Optional)

The focuser project provides for an optional LCD1602 display that shows the current and target positions of the focuser, and other settings such as maxSteps and temperature.

Displaying the position values would be useful in manual setups where the push buttons are used to control the focuser.

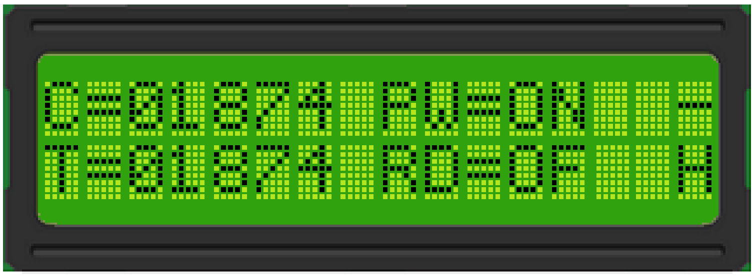
LCD1602 IIC I2C TWI 1602 Serial Port LCD Display Module

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 page after the other with a delay between pages. The length of time an LCD page is displayed can be configured and altered by the user.

LCD Screen Page 1

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



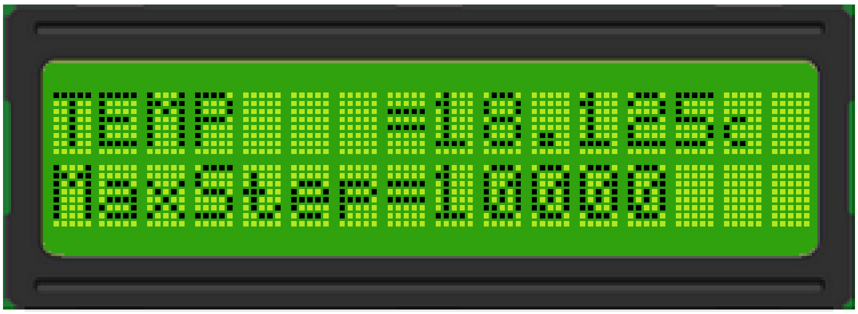
Pwr indicates if power is kept to the stepper coils once a move is completed. For further information, please refer to the Readme.htm file that is available when the ASCOM driver is installed.

RD indicates reverse direction is either ON or OFF

F/H indicates the stepping mode (F=full, H=1/2, 4=1/4, 8= 1/8, 16=1/16, 32=1/32 step mode)

LCD Screen Page 2

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

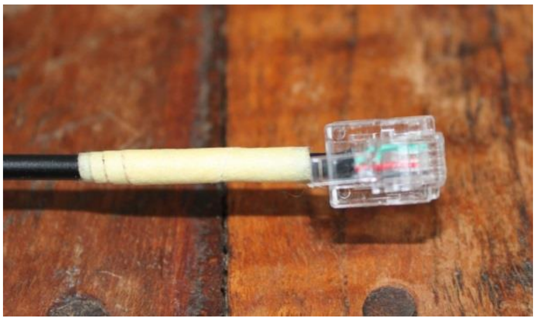


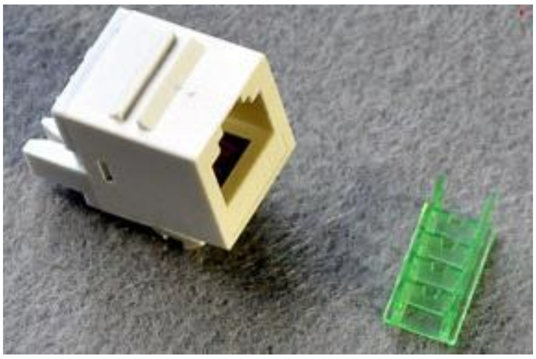
If the focuser is moving in the wrong direction when sending a position command (such as -10 or -100 actually moves the focuser OUT instead of IN) from the Windows Application, then enable Reverse Direction to ensure the focuser moves in the correct direction. (- moves IN, + moves OUT)

TEMPERATURE PROBE (Optional)

One temperature probe is supported. The temperature probe is a DS18B20 and the resolution is set by the controller to 10-bit giving 0.25°C resolution. The accuracy of the measured temperature is within 0.5°C.

The prototype used a 3pin stereo jack and socket to connect the temperature probe to the controller. In later productions, this was replaced with a RJ11/45 connector.





The probe wires are crimped to a 6P4C connector. A 6P4C socket is mounted on one side of the hobby box and connected to the main board. *Disconnection or reconnection of the probe must be done when power is OFF*. VCC and GND are wired separately. This permits a slightly faster read of temperature than if the probe operated in parasitic mode.

You could use an RG45 Ethernet connector or a 3pin stereo audio connector.

The sensor end of the probe is fitted so that it is in contact with the main metallic area of the focuser. Cable length of the purchased unit is around 1 meter (though you extend this using an extension cable).

The lugs on the top and bottom of the 6P4C jack are removed with a hobby knife and then the jack is hot glued in place.

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

The ASCOM driver and Windows application has the capability of adding an offset value to the value returned by the temperature probe. This provides a means of calibration for the probe.

Temperature compensation is NOT provided with myFocuserPro but the improved myFocuserPro2 does support temperature compensation.

WARNING - NEVER PLUG-IN OR UNPLUG THE TEMPERATURE PROBE CABLE WHEN POWER IS ON

myFocuserPro 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 myFocuserPro Windows application that can remotely control the focuser
3. A dedicated myFocuserPro 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)

myFocuserPro 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
ScopeFocus
On Linux and Mac systems, works with applications that support a Moonlite driver

These are recommended as they have passed testing and are used by a number of users. 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 normally used by client applications that access the focuser (via ASCOM). 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 6500 position 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 1024L on connection. If a set maxIncrement command is sent to the controller, the controller will set maxIncrement to the same value as maxSteps.

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 have 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

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*.

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. 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 shaft 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 a good idea, as the mechanics and manufacture of the mechanisms mean that they are not robust enough to have that level of force (from stepper motors) consistently applied to them. There are small gears 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.

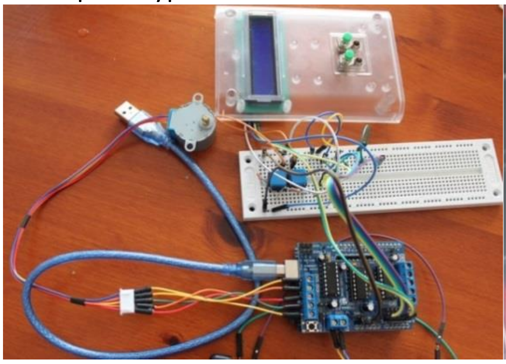
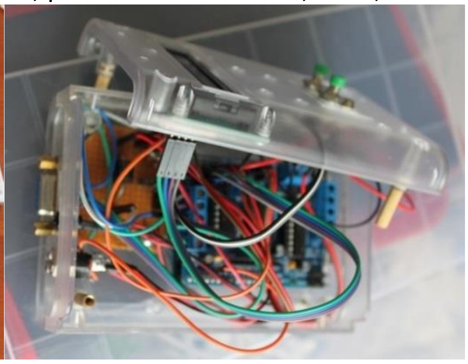
So why do some users try connecting to the fine focus knob? The answer is that it already provides a 10:1 reduction so this increases the resolution without any cost. The downside is the cost in replacing the focuser mechanism when the 10:1 reduction is damaged by the stepper.

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.

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.

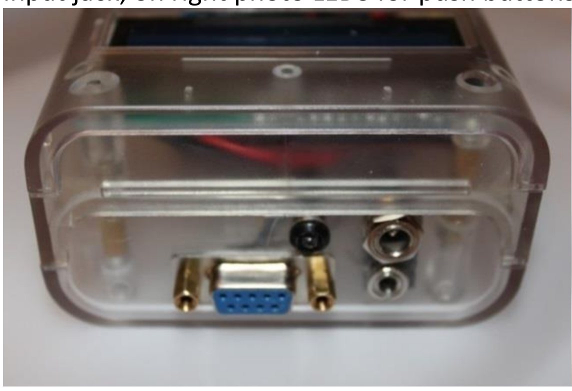
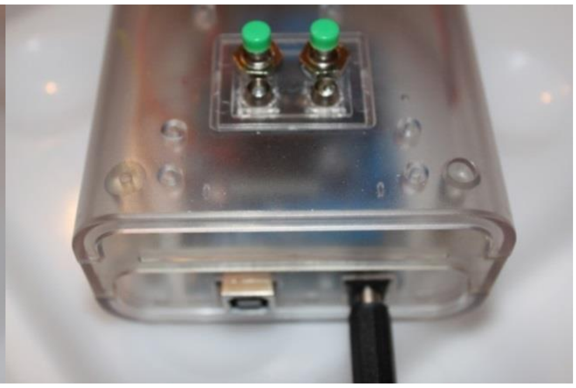
myFocuserPro PROTOTYPE BUILD PICTURES

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

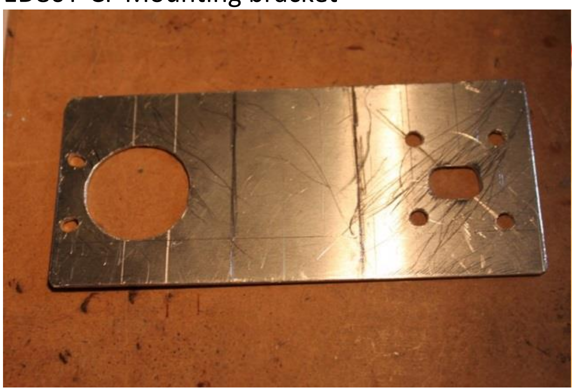
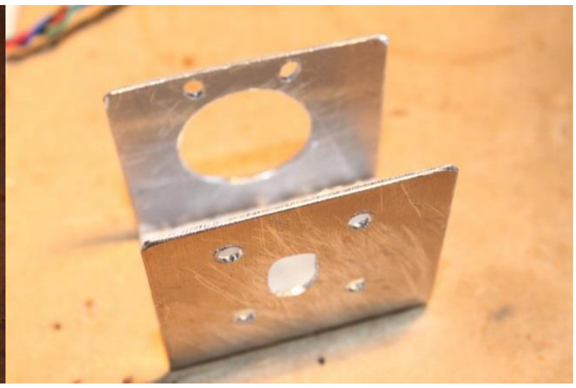


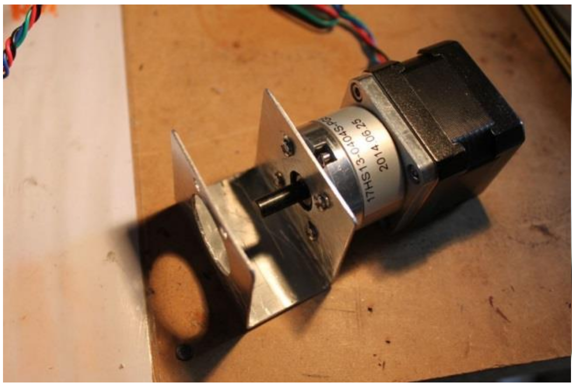
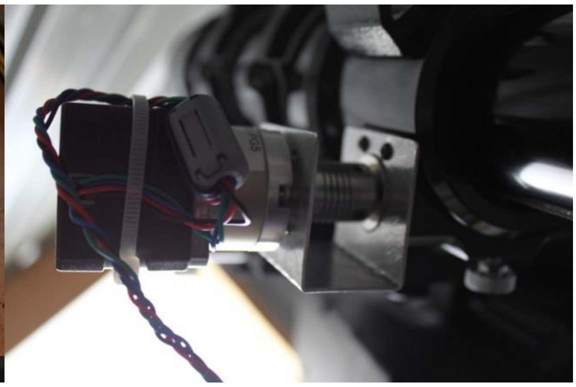
Arduino, L293D shield, LCD, switches enclosed in Link-Sprite case

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

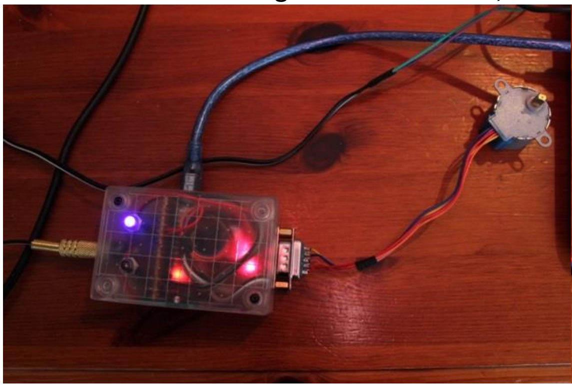
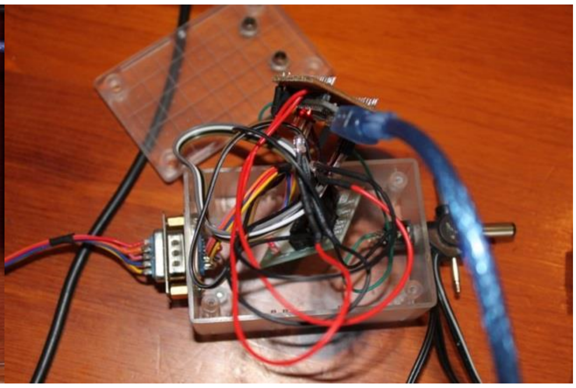


ED80T-CF Mounting bracket



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 will invalidate 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, step mode and type of stepper motor being used.

Once you have determined the right step mode setting (full, 1/2 etc) 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

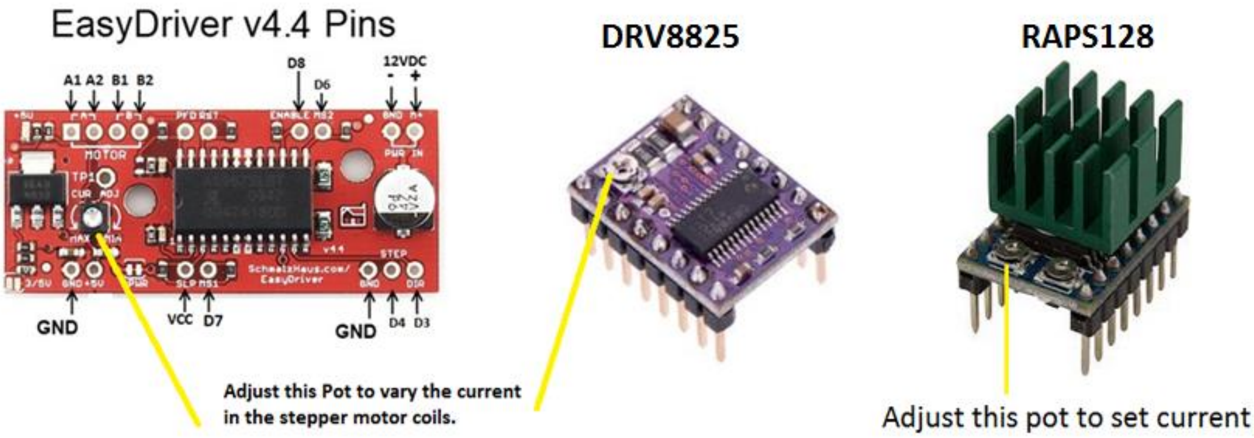
If you are using a FULL version but did NOT wire the Push buttons, the code for the push buttons needs to be disabled. Look for the following lines near the beginning of the firmware file

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

Ensure that the two // appear in front of the #define statement. Upload the modified firmware to your controller.

4: FOR DRV885/EASYDRIVER/RAPS128 BOARDS – SET CURRENT LIMITS

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.



Note1: 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. Enter a focuser position of 5000 and click the Goto button
3. Wind the pot all the way anticlockwise until the motor stops moving
4. 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
5. 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
6. It should now be close enough
7. 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.

5: FIRMWARE CHANGE FOR L293D AND L298N DRIVER BOARDS

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.

Find and change the line in GREEN below

```
const int stepsPerRevolution = 1036;      // NEMA17-PG5 motor
// you need to change the above line to reflect your stepper motor, examples below
// const int stepsPerRevolution = 2048;   // 24BBYJ-48 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 1036;   // NEMA17-PG5 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 200;    // NEMA17 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 5370;   // NEMA17-PG27 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

```
const int stepsPerRevolution = 10740;      // NEMA17-PG27 Half-steps motor
// you need to change the above line to reflect your stepper motor, examples below
// const int stepsPerRevolution = 2048;    // 24BBYJ-48 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 1036;    // NEMA17-PG5 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 200;    // NEMA17 motor, if half stepping multiply by 2
// const int stepsPerRevolution = 5370;    // NEMA17-PG27 motor, if half stepping multiply by 2
```

Once changed, upload the modified firmware to your controller.

6: FIRMWARE CHANGE: L293D Driver Board

The L293D driver board supports 4 motor connectors labelled M1 to M4. By default, the firmware supports the stepper motor connected to Motor Port 2. 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 1      // 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 4 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 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)
```

7: 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

7-1 MOTOR SPEED SETTINGS FOR L293D SHIELD OPTION

For the L293D Shield, the motor speed is controlled by the following lines of code in the firmware file

```
// motor speeds in approx. RPM - you need to adjust these depending on the stepper motor you select
const int    motorSpeedSlowRPM = 1;
const int    motorSpeedMedRPM  = 4;
const int    motorSpeedFastRPM = 8;
```

Change one value at a time (possibly doubling or halving the value), upload the changed firmware and test the speed. Reprogram the controller after each change and check that the motor moves correctly for each speed setting at the correct step mode for your controller. For the L293D shield very small changes can have a huge effect on speed.

7-2 MOTOR SPEED SETTINGS FOR L298N OPTION

For the L298N board, the motor speed is controlled by the following lines of code in the firmware file

```
const int   motorSpeedSlowRPM = 10;
const int   motorSpeedMedRPM  = 30;
const int   motorSpeedFastRPM = 50;
```

Use a similar process as in 7-1 above to test different values. Reprogram the controller after each change and check that the motor moves correctly for each speed setting at the correct step mode for your controller. For the L298N board very small changes can have a huge effect on speed.

7-3 MOTOR SPEED SETTINGS FOR ULN2003 OPTION

For the ULN2003 board, the motor speed is controlled by the following lines of code in the firmware file

```
const int   motorSpeedSlowRPM = 5;
const int   motorSpeedMedRPM  = 10;
const int   motorSpeedFastRPM = 30;
```

Use a similar process as in 7-1 above to test different values. Reprogram the controller after each change and check that the motor moves correctly for each speed setting at the correct step mode for your controller.

Please note that the speed is adjusted with the step mode set to that selected for your focuser. Changing the step mode at a later date will invalid the settings you have made for setting the speed of the stepper motor.

8: Set StepSize, StepMode and maxStep

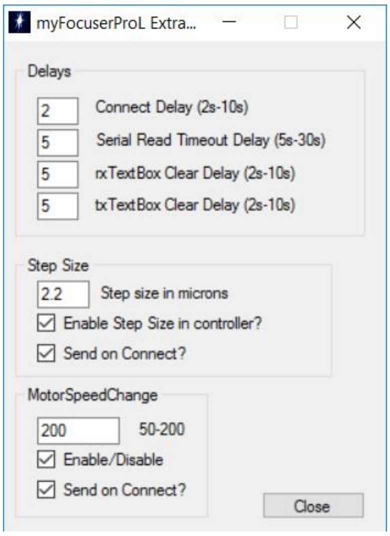
a) With the stepper motor disconnected, move the focuser to the in-most position as position 0 (suggest you position the focuser at ½ turn outwards and make that 0)
b) Next clamp the stepper motor in place and engage the stepper with the focuser shaft (tighten screws on clamp etc)
c) Start the windows 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).

If you are using a stepmode which is 1/8, 1/16, 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.

8-1: 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. Close the form by clicking the Close button



8-2: Initial Setup of StepMode

Make sure you have done step 1 on the Extra Settings form. 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.

8-3: 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.

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 the settings for 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.

https://www.youtube.com/watch?v=BH5izUvj9Ck

CONGRATULATIONS: Your focuser is now ready to use!

myFocuserPro FIRMWARE SETTINGS

The following settings are stored in EEPROM and remembered by the controller (may vary depending upon minimal, minimal with temperature probe and full versions)

fposition	last focuser position
maxstep	maximum focuser position
stepmode	indicates stepmode, full, half, 1/4, 1/8. 1/16. 1/32
ds18b20resolution	resolution of DS18B20 temperature probe
updatedisplayintervalNotMoving	time each LCD page is displayed for
stepsize	the step size in microns, ie 7.2
ReverseDirection	
coilPwr	
tempmode	temperature display mode, Celsius=1, Fahrenheit=0
stepsizeenabled	if true, controller returns step size
lcdupdateonmove	update position on LCD when moving

Note: Stepper Coil power means that at the end of the move, when the stepper stops, power is either OFF or ON to the stepper motor coils. If OFF, this saves power, but could 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 is not moving. Some steppers 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.

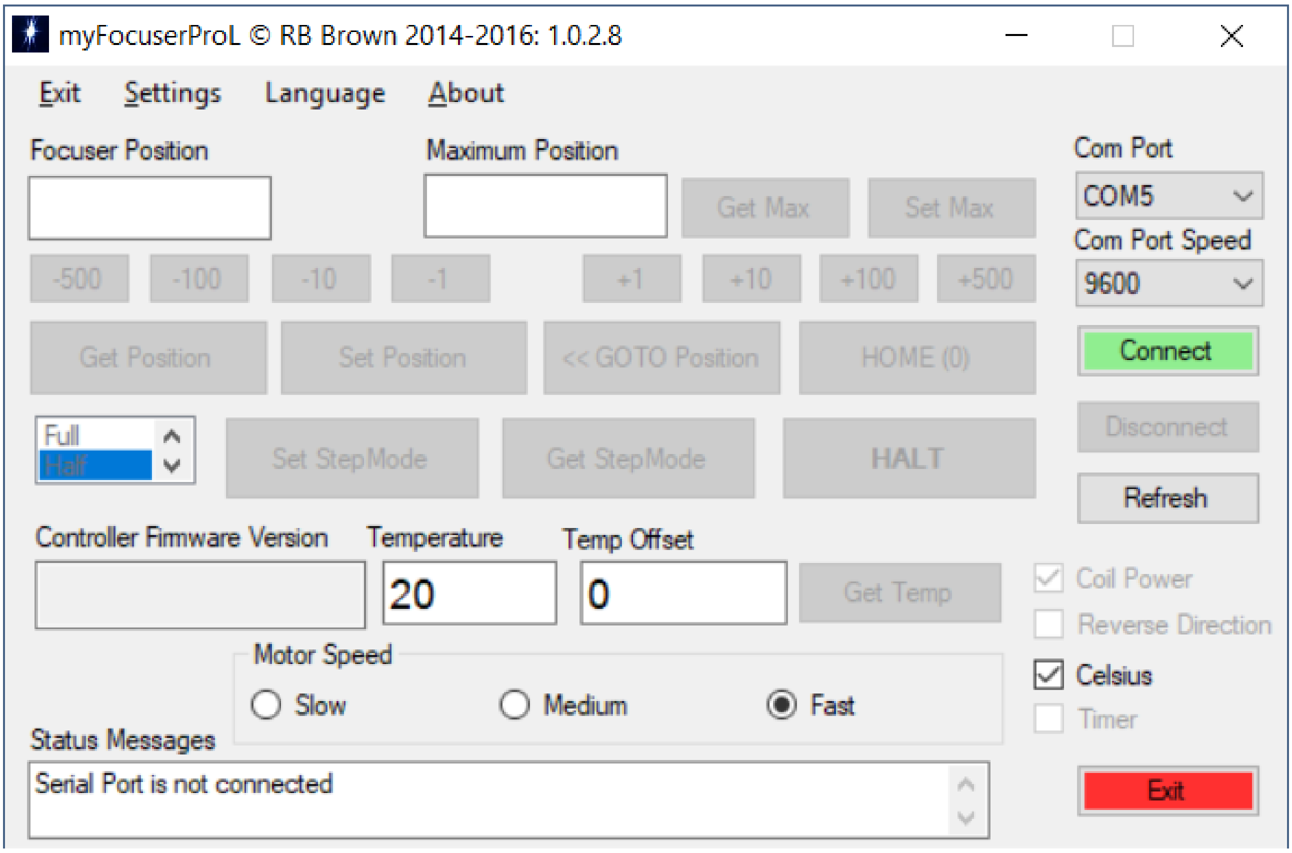
Note: Use 7.5-9VDC with the 28BYJ-48 stepper motor if using Coil Power ON otherwise the stepper motor may get very hot.

These values are retrieved from the controller when connecting. Once connected, these settings can be changed.

myFocuserPro WINDOWS APPLICATION

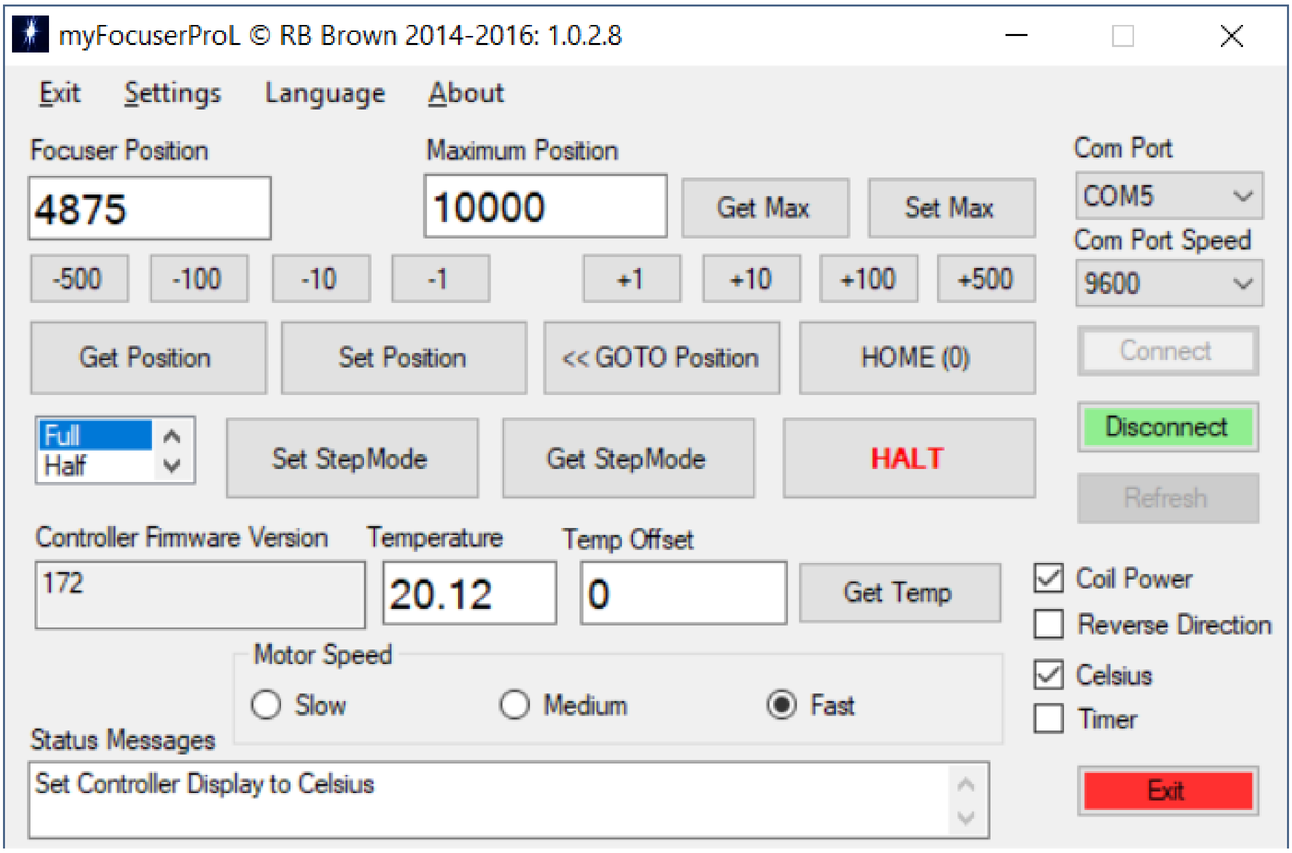
The myFocuserPro application gives full manual or remote control of the focuser (supports ALL build options). Note that the settings (Coil Power, Reverse Direction and Step-Mode Setting) are remembered by the application and controller.

To run the myFocuserPro Windows Application, first connect the myFocuserPro controller, then click on the myFocuserPro application Icon (or use the Start Menu to locate the program entry)



Most options are greyed out (disabled) until the application connects to the myFocuserPro controller. To connect to a myFocuserPro controller, select the correct serial port from the drop-down list and then click Connect.

When connected, the myFocuserPro controller values for Focuser Position, Maximum Position, Step Mode and other settings are retrieved from the controller and displayed.



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

myFocuserPro Main Window Buttons

The main screen is reasonably self-explanatory.

☑ Celsius - When enabled (ticked), the temperature is displayed in Celsius. When disabled (un-checked), the temperature is displayed in Fahrenheit

☑ Timer - When enabled (ticked), indicates that the application will periodically poll the myFocuserPro controller and request a temperature update. The polling interval is set under the settings menu

☑ 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)

Goto a specific focus 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 << Goto Position Button. The focuser will move.

Get Position - returns the current myFocuserPro 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 myFocuserPro MaxSteps value in the Maximum Position text box.

Set Maximum Position - type in the desired MaxSteps 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 myFocuserPro controllers.

COMPort - Use the dropdown list to select the correct serial port that the myFocuserPro is connected to.
Connect - After selecting the correct comport, click Connect to connect to the myFocuserPro controller.
Disconnect - disconnects the myFocuserPro controller.
Refresh - refreshes the list of available comports.

EXIT - Exits the application (if connected to a myFocuserPro controller, you must disconnect from the controller before exiting the application).

Temp Offset - The Temp offset entry box provides a calibration mechanism for adjusting the returned temperature value. 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 point separator).

Get Temperature - requests the temperature from the myFocuserPro 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.

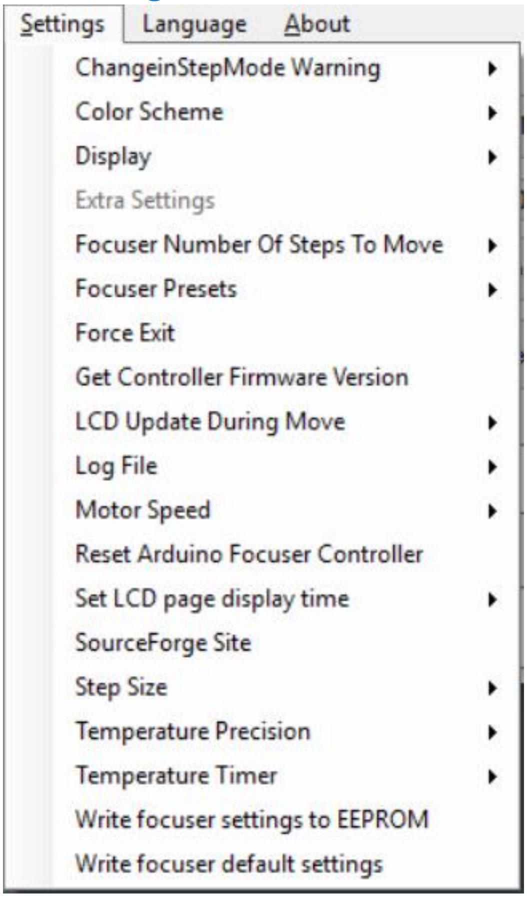
Set StepMode - To set the StepMode, select the desired stepping mode from the dropdown list, then click the Set StepMode button

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

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

Notes:
1. Microstepping (StepMode) is ignored when using the ULN2003 or L298N build options.
2. Focuser Position, Maximum Position (MaxSteps), Coil Power, Reverse Direction and StepMode are updated from the controller when the application software connects to the controller.
3. The default setting for temperature is Celsius
4. If Get Temp is used on a controller that has no temperature probe support (such as the minimal controller, or a full controller that has no temperature probe connected), the value returned will be set to 20.0 Celsius (-4° Fahrenheit).

The Settings Menu



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

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

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

Extra Settings - Displays a Form which specifies a number of additional parameters.

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

Focuser Presets - This is used 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.

Get Controller Firmware Version - Returns the controller firmware version when connected/

LCD Update During Move - This disables/enables the updating of the LCD display on the myFocuserPro controller during moves. Some users prefer to disable this in order to get higher motor speeds.

Log File
-Enabled, turns error logging ON
-Disabled, turns error logging OFF
-Reset Error Log File path displays a dialog box to specify the directory where the error log file will be saved

Motor Speed - The speed of the stepper (delay between steps) can be adjusted in the range Slow-Medium-Fast. The speed settings are remembered by the application program and the motor speed setting is sent to the controller when connecting.

Reset Arduino Controller - Will reset the Arduino controller.

Set LCD Page Display Time - Controls how long each LCD page is displayed for (2000-4000 milliseconds).

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

StepSize - Options to query the controller for StepSize and whether the controller has stepsize enabled. These values can also be sent to the controller on Connect.

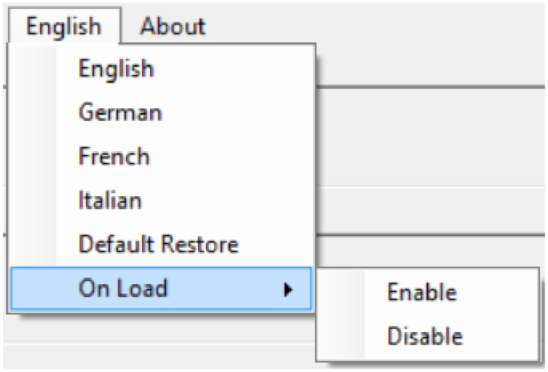
Temperature Precision - Options to set and get temperature precision settings.

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

Write focuser settings to EEPROM - Write the current focuser settings to EEPROM immediately.

Write focuser default settings - Overwrite the current settings with the controller default settings specified in the firmware file. This needs to be followed by a reset of the Arduino controller. To reset the Arduino controller, select Reset Arduino Focuser Controller from the settings menu. The controller will reset and start with the default settings specified in the firmware (**Warning** – all settings changed by the user will be lost).

The Language Menu



The Language settings allow the user to specify the language of existing text, menus, labels and messages. Current support is for English, German, French and Italian. Other languages should be relatively easy to add.

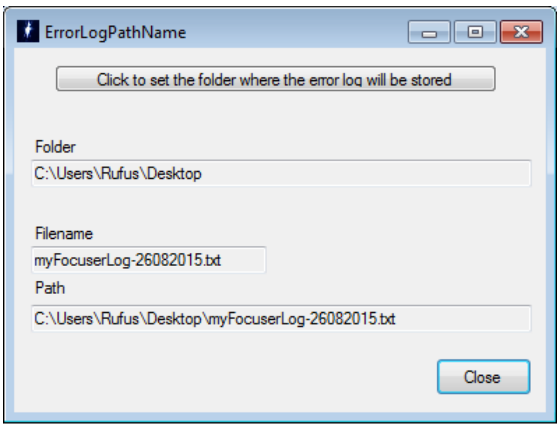
Selecting the desired language will change text to that language. To remember the language settings, On Load Enable should be selected else the focuser application will default to English when run the following time.

Please note that language translation was done via Google Translate and https://www.freetranslation.com thus might not be an accurate translation of the existing text. Most of the important menus, buttons, labels and text messages have been translated (but not all – only what has deemed to be important or most often used). Additional languages will be added as time permits.

The ErrorLogPathName Form

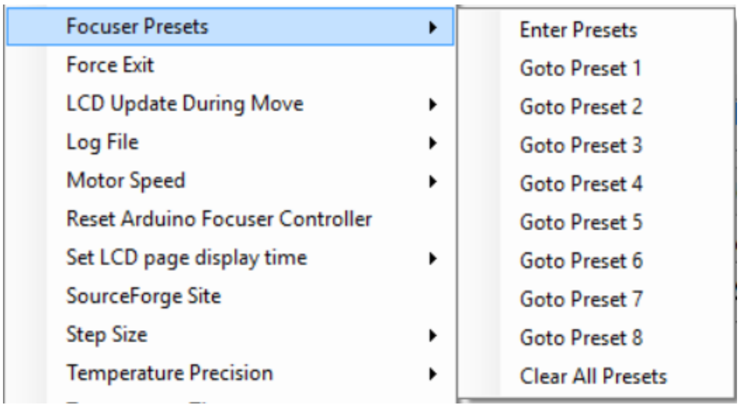
You can specify the directory/folder where the error log file will be stored. The path is remembered by the application. This dialog box is accessed from the menu bar Settings-Log File-Reset Error Log File Path

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.



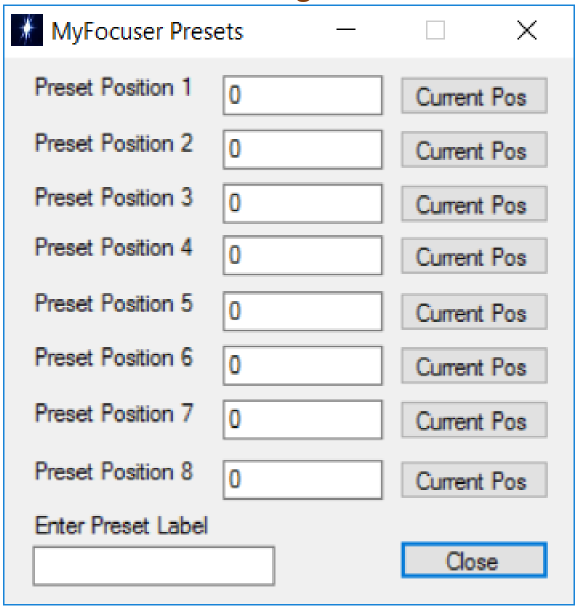
The Focuser Presets Settings Form

Selecting the “Enter Presets” option from the Focuser Presets menu under the



Settings>Focuser Presets>Enter Presets displays the following form

Enter Presets and Change Preset Labels



You can enter up to 8 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).

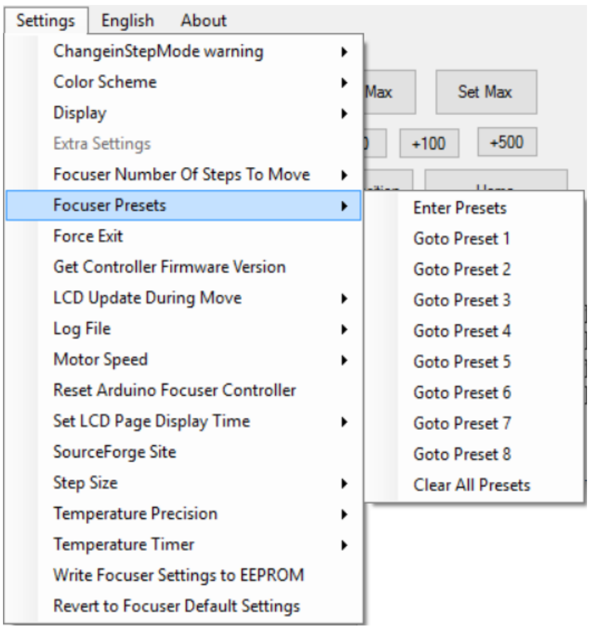
Typical values must be greater than 0 and less than MaxSteps. To copy the current focuser position to a preset, click the associated **Current Pos** button.

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

You can also use this form to change the focuser Preset labels. Enter the desired text for a Preset label in the Enter Preset Label textbox, then single click the Preset label (such as Preset Position 1) to make the change.

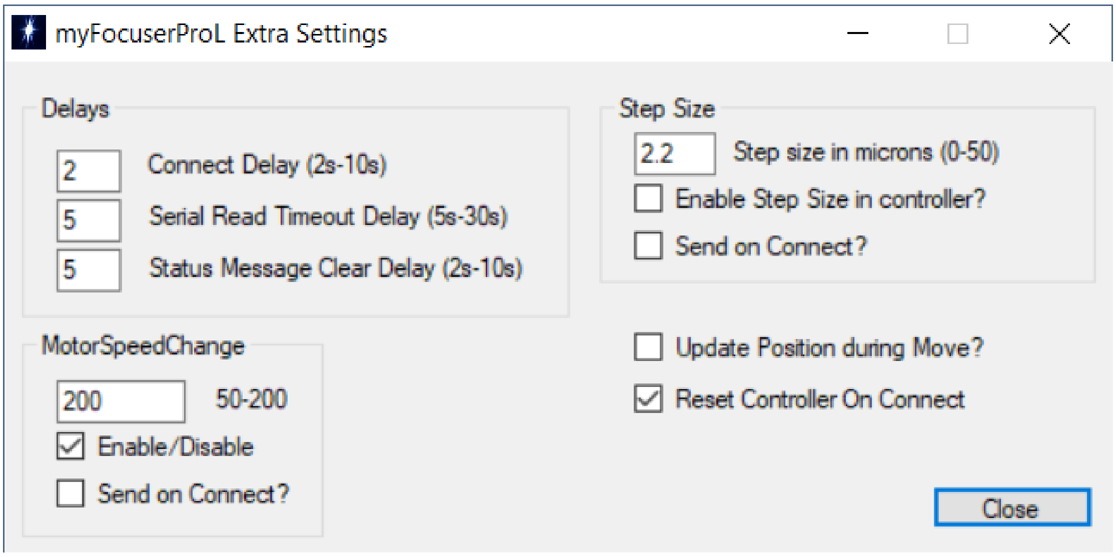
Move to a Preset

To command the focuser to move to a preset value (assuming one has been set), use the Settings/Focuser presets menu, as shown



The ExtraSettings Form

These options MUST BE SET before connecting to the focuser. Selecting the "ExtraSettings" option (only enabled when the application is NOT connected to a myFocuserPro controller) under the Settings menu displays the following form



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

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

Status Message Clear Delay (2s-10s) - the value in seconds after writing to the Status Message Textbox that the message is cleared.

Update Position during Move? Check this box to show updates of focuser position during a move.

Step Size - specifies the user defined step size (the amount in microns that the focuser moves in ONE step)

Enable Step Size in Controller - Enables the stepsize setting in the controller

Send on Connect? - When checked, the stepsize will be sent to the controller and the controller will also enable the stepsize setting

MotorSpeedChange Threshold - The number of steps on which the focuser will slow down when approaching a target position

MotorSpeedChange Enable/Disable - When Checked, the MotorSpeedChange threshold value is sent to the controller on connection as well as the MotorSpeedChange being enabled in the controller. If unchecked, the application will request the current controller settings for these values when a connection is established

MotorSpeedChange Send On Connect - When checked, the MotorSpeedChange value and enable setting will be sent to the when a connection is established

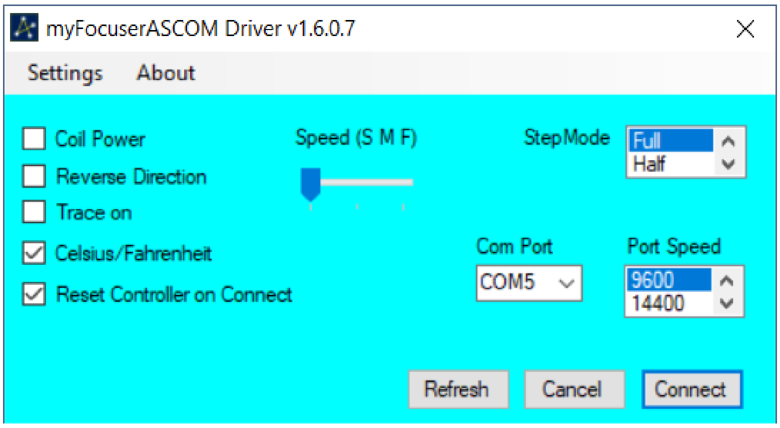
myFocuserPro CONNECTION SEQUENCE

When connecting to a myFocuserPro controller, the command sequence is

```
// GET version number of controller
// GET maxStep from focuser
// GET initial focuser position
// GET initial maxIncrement
// GET coil power from controller
// GET reverse direction from controller
// GET stepmode from controller
// Get LCD Display setting Celsius or Fahrenheit
// Get MotorSpeed
// Get UpdatedisplayintervalNotMoving
// Get MotorSpeed Threshold value
// Get MotorSpeed Threshold Enable/Disable
// check if we need to send stepsize etc from the ExtraSettings form
        // Yes
                // Send StepSize
                // send Stepsize enable
        // No
                // Get Stepsize enable state
                // Get Stepsize
// Get temperature precision
// Get update of position on LCD when moving (00=disable, 01=enable)
// Get temperature
```

myFocuserPro ASCOM DRIVER

An ASCOM myFocuserPro driver is provided, which comes with an install program. The ASCOM driver works with ALL build options. Half steps are ignored when using the ULN2003 and L298N build options.

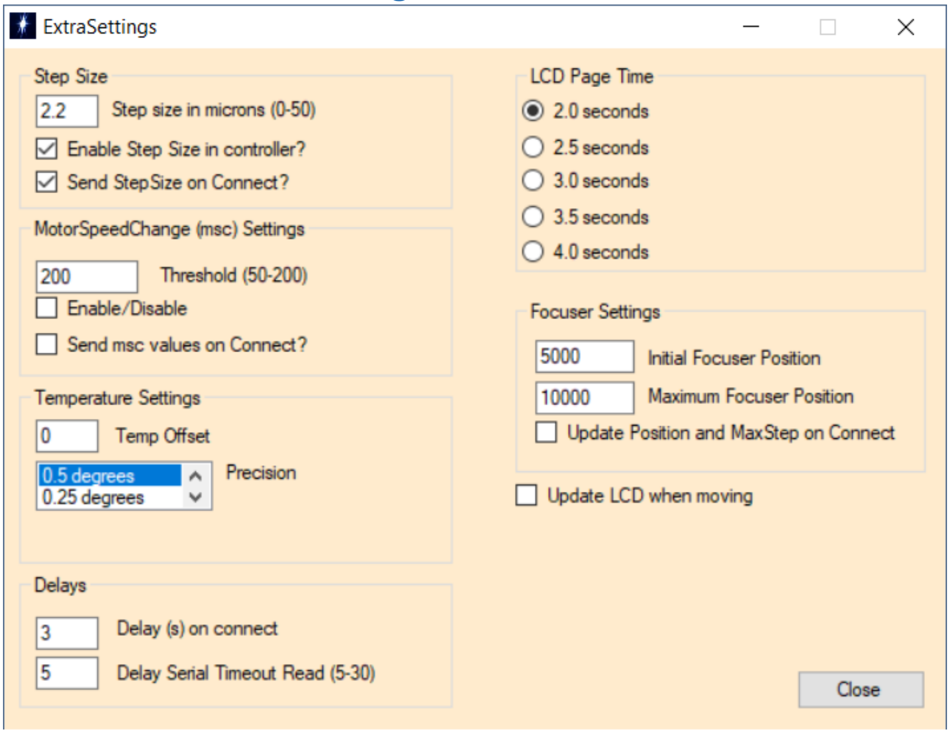


You need to determine the correct Com Port that the focuser is using (you can get this easily by using the Windows Application). Once you have the Com Port value, you need to select it from the Combo List for the Com Port on the ASCOM setup dialog box before clicking the Connect button. The application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

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

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

The ASCOM Driver Settings Menu



ASCOM Dialogbox Settings and Controls

The controls must be SET to their correct state **BEFORE** clicking the Connect button. **Please refer to the Windows Application for more information of what each of these settings do.**

myFocuserPro ASCOM DEFAULT OVER-RIDE SETTINGS

These settings are stored in the ASCOM driver and updated when a connection is made (some are sent to the controller when connecting).

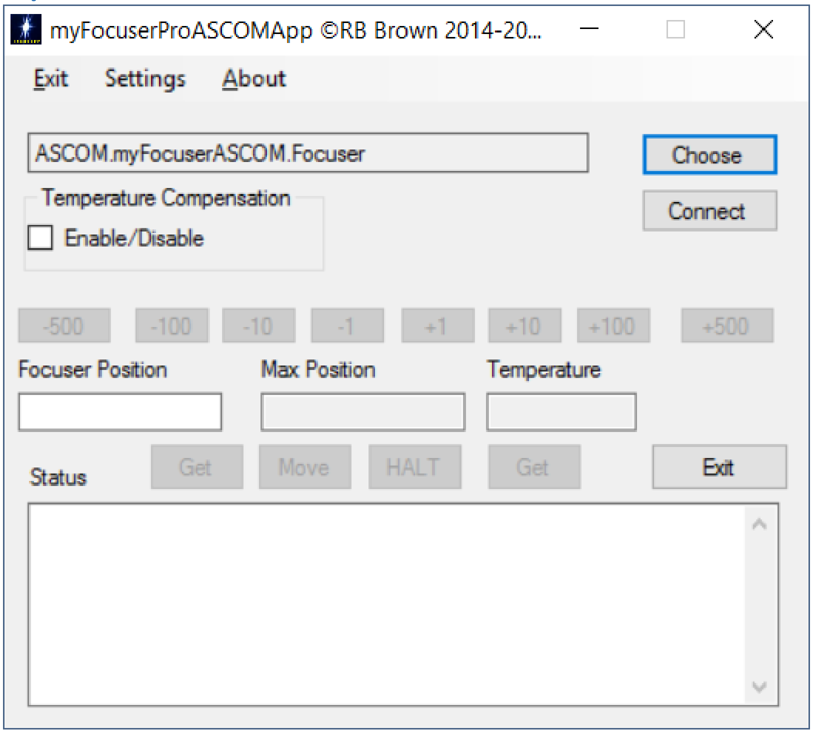
CoilPower	False
ComPortSpeed	9600
DelayOnConnect	3
EnableStepSize	False
InitFocusPos	5000
LCDPageDisplayTime	2000
MaxStep	10000
MotorChangeEnabled	False
MotorChangeThreshold	200
MyComPort	(None)
Reversed	False
SendStepSizeOnConnect	False
Serial Port Read Timeout	3000
SpeedSetting	0
StepMode	1
StepSize	2.2
TemperaturePrecision	9
TempMode	True
TempOffset	0
TraceEnabled	False
UpdateLCDWhenMoving	False
UpdateOnConnect	False

myFocuserPro ASCOM DRIVER CONNECTION SEQUENCE

When connecting to a myFocuserPro controller, the command sequence is

- If UpdateOnConnect checkbox is enabled
    - Send MaxSteps
    - Send Focuser Position
- If UpdateLCD When Moving is enabled
    - Send Enable UpdateLCDWhenMoving
- If Send MotorSpeedChange Settings on Connect
    - Send MotorChangeEnabled state
    - Send MotorSpeedChangeThreshold value
- If Send StepSize Settings on Connect
    - Send StepSize Enabled state
    - Send StepSize value
- Send Celsius or Fahrenheit
- Send MotorSpeed
- Send LCD PageDisplayTime
- Send StepMode
- Send Temperature Precision Setting
- Send Coil Power
- Send Reverse Direction
- Get Maxsteps
- Get MaxIncrement
- Get Focuser Position
- Get Temperature

myASCOM APPLICATION TESTER



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

It can be used as a focus controller (using the ASCOM driver) and its main purpose has been to test the ASCOM driver, as it provides a means of viewing status messages from the controller.

HOW TO RUN TWO FOCUSERS

What happens if you have two telescopes and want to have two myFocuserPro controllers, one on each scope?

There are some simple rules.

1. You can only run one instance of the myFocuserPro Windows application at any time
2. You can only run once instance of the myFocuserProMini 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: eg Orion ED80	myFocuserPro App	myFocuserPro App	ASCOM driver
Scope 2: eg SV102T	MyFocuserProMini 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 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.

WHAT TO DO IF YOU LOOSE 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 myFocuserPro 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 MaxSteps 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

A SPECIAL NOTE ABOUT STEP MODE

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.

Board	Driver	Full?	Stepper	Stepping	.ino file
Nano	ULN2003	F	28BYJ-48	F	Focuserv1xx_ULN2003_F
Nano	ULN2003	M	28BYj-48	F	Focuserv1xx_ULN2003_M
Uno	L293D	F	Nema17PG5/PG27/28BYJ-48	F/H	Focuserv1xx_L293D_F
Nano	DRV8825	F	Nema17PG5/NEMA17PG27	H/H/4/8/16/32	Focuserv1xx_DRV8825_F
Nano	DRV8825	M	Nema17PG5/NEMA17PG27	H/H/4/8/16/32	Focuserv1xx_DRV8825_M
Nano	EasyDriver v44	F	Nema17PG5/NEMA17PG27	F/H/4/8	Focuserv1xx_DRV8825_F
Nano	EasyDriver v44	M	Nema17PG5/NEMA17PG27	F/H/4/8	Focuserv1xx_DRV8825_M
Nano	L298N	M	Nema17PG27	F	Focuserv1xx_L298N_M
Nano	**DRV8825/HW203**	**All**	**Nema17PG27**	**F/H/4/8/16/32**	**Focuserv1xx_DRV8825_HW203_xx**
Nano	ST6128/HW203	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuserv1xx_ST6128_xx
Nano	RAPS128/HW203	All	Nema17PG5	F/H/4/8/16/32 /64/128	Focuserv1xx_RAPS128_HW203_xx

You will want to choose a combination that gives you around 10 steps within the critical zone for your telescope. The higher the micro-stepping value the less torque there is available. In real terms, this means using half steps with a DRV8825 driver board with a NEMA17PG5 or NEMA17PG27 as the best possible solution.

If you try to send an invalid step-mode to a 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 whatever the step-mode was before the command was received will be used instead.

HOW TO UPGRADE YOUR FIRMWARE AND SOFTWARE

New releases often occur after you have built and set-up your myFocuser system.

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.

Support is freely given concerning any current release. If you have any issues, please contact me and I will do my best to work with you to help resolve any issues you might have.

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.

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)

- Focuser Position
- Max Steps
- Coil Power
- Reverse Direction
- Motor speed
- Step mode
- Temp offset
- Step Size
- Temperature Precision
- Firmware version

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

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 Focuserv133_DRV8825_HW203_F.ino

You would look for the latest file Focuserv1??_DRV8825_HW203_F.ino (as of 10th April 2016 it would be Focuserv156_DRV8825_HW203_F.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 myFocuser 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.

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/driver before installing the new version/

Check your focuser settings

The last step is to check that your controller is configured with your settings. It is probably best to use the Windows application for this as you have access to a wider range of functions than the ASCOM driver.

Run the new Windows application and connect to your myFocuserPro2 controller. Verify that all your settings are what they should be. You do this by comparing the settings shown by the Windows application against the settings you recorded before doing the updates. If the settings do not match, make the required changes to ensure that your focuser settings are correct.

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.

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

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

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

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 reticle 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 packages with MaximDL, APT and Scopefocus. You could also use FocusMax v3 which was the last free version of Focusmax before it became commercial.

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).

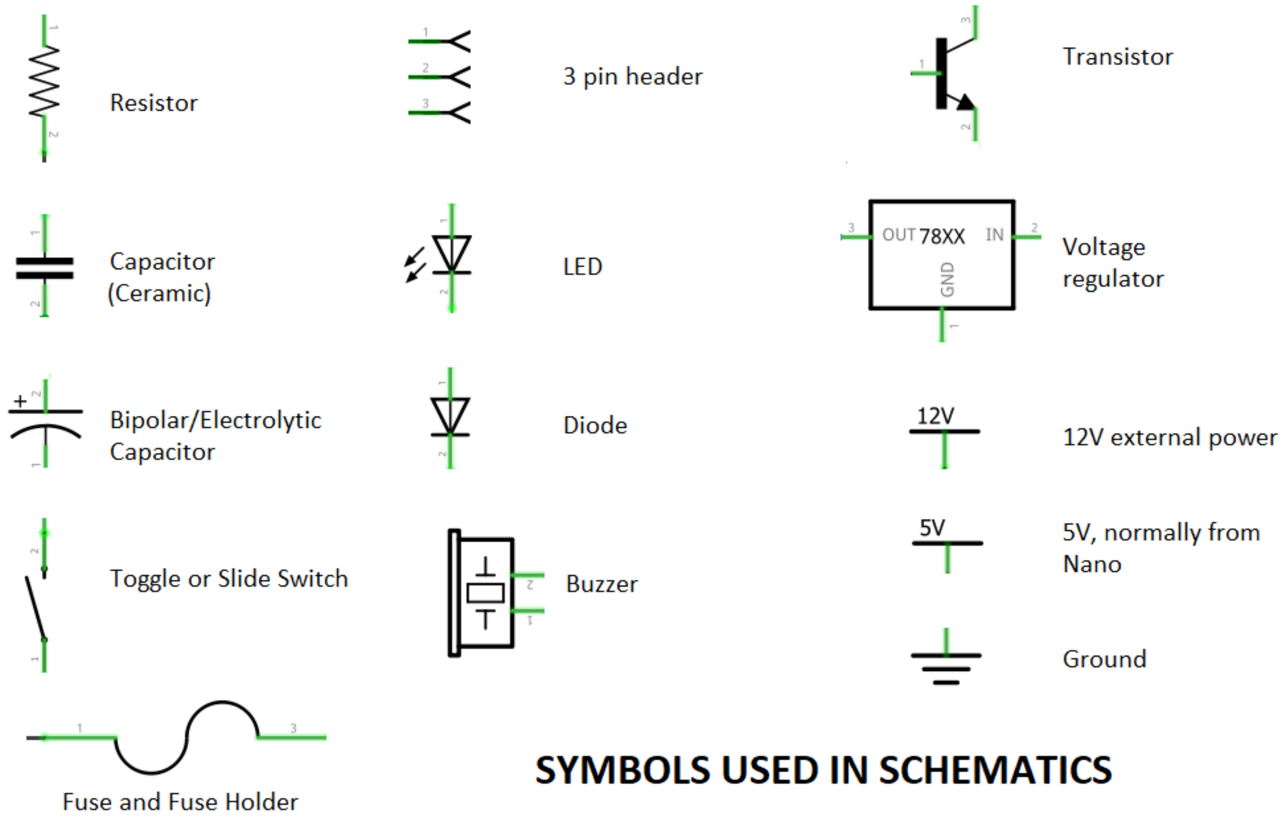
The downside is the time required to train your system, as well as the generated V curve only works for that particular configuration. Change anything like adding a focal reducer or a different telescope or a different camera and you have to start all over and generate new V-curves.

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.

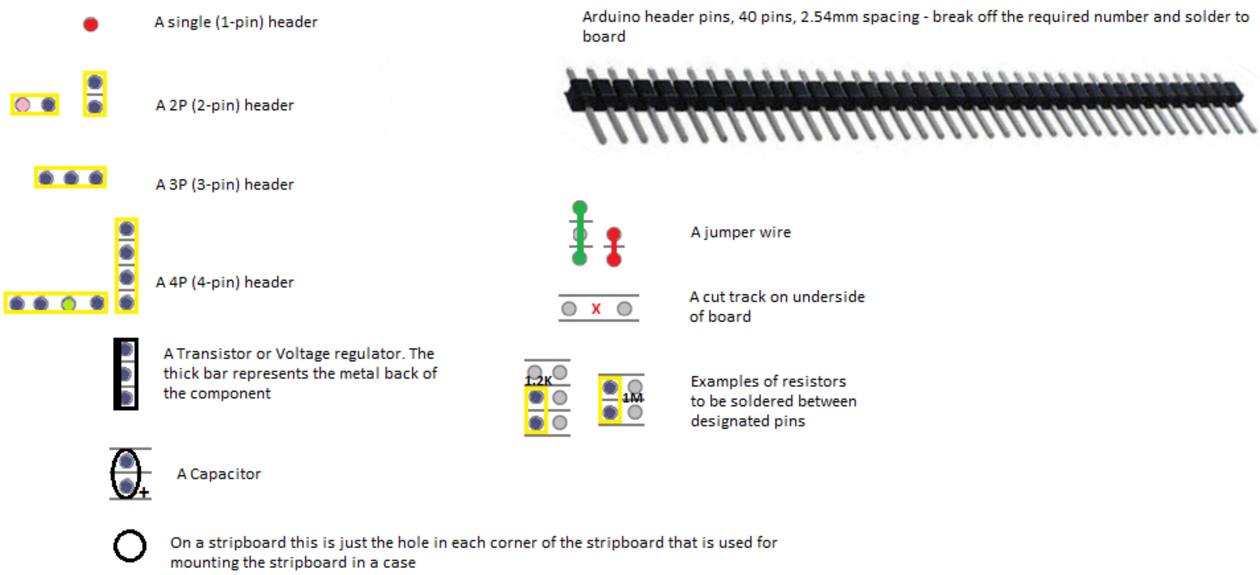

Nema 17 with belt drive as a focuser for a Canon EOS Lens (f2.4)

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

SYMBOLS AND MEANINGS USED IN BOARD LAYOUTS

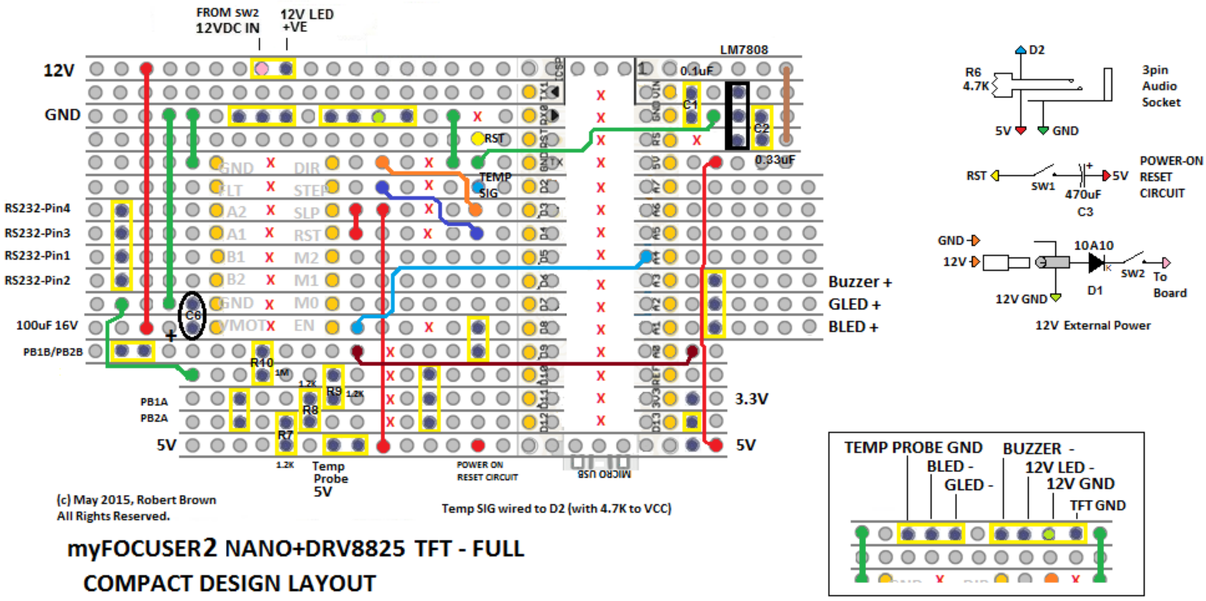


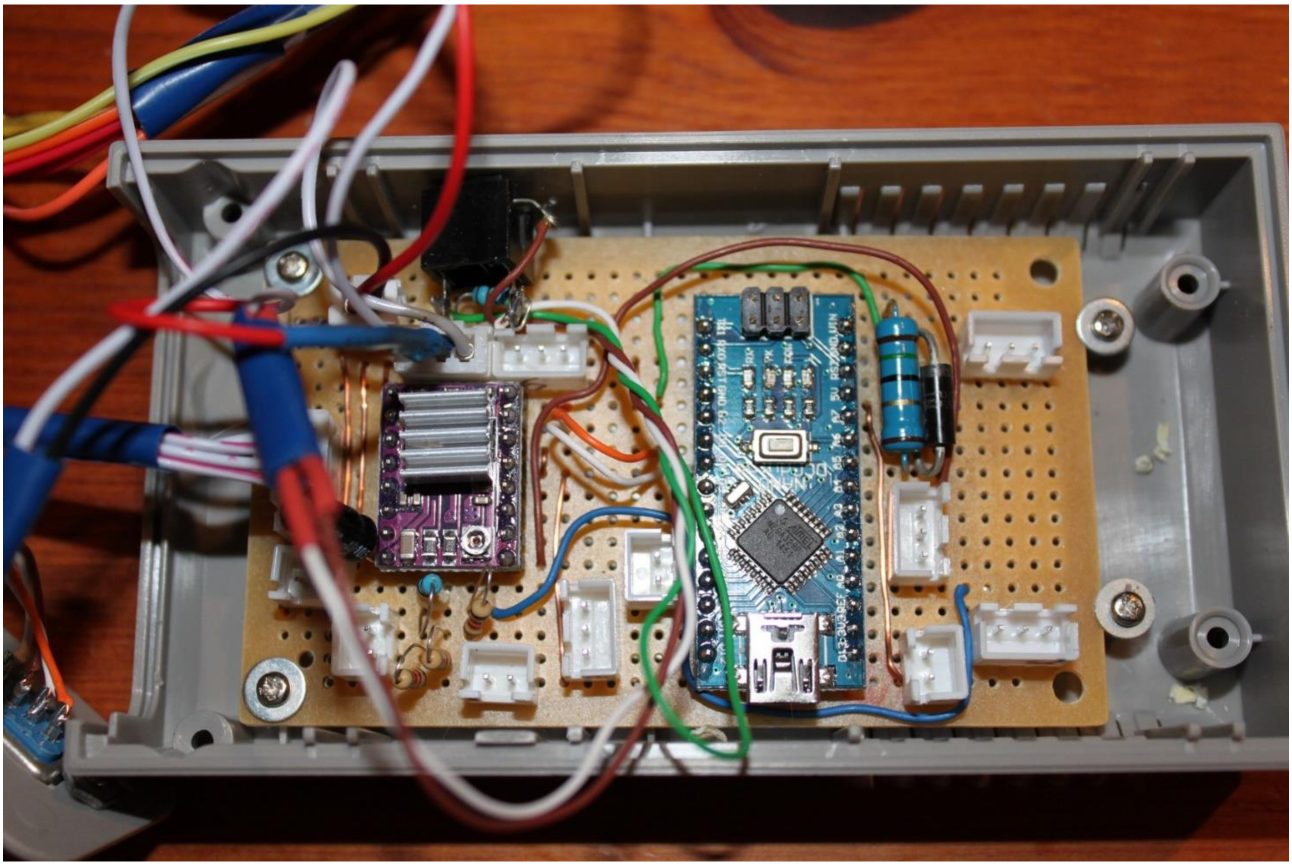
SYMBOLS USED IN SCHEMATICS



STRIP BOARD SYMBOLS USED IN DIAGRAMS AND LAYOUTS

COMPARISON BETWEEN COMPONENT BOARD AND FINISHED PCB





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 myFocuserPro ASCOM Driver. The myFocuserPro ASCOM driver has passed the CONFORM report (available on the SourceForge website).

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

APPENDIX B TESTING THE myFocuserPro 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.

- Test1 Test the momentary switches (optional)
- Test2 Test the stepper motor, forward and reverse, 28BYJ-48 and ULN2003 (Build Option 1)
- Test3 Test the DS1820 sensor probe (optional)
- Test4 Test the stepper motor, NEMA17 and L293D Shield (Build Option 2)
- Test5 Test the LCD1602/I2C (optional)

Test programs are available on the Sourceforge site

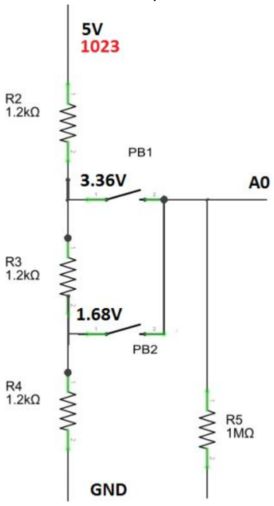
APPENDIX C DERIVING VALUE RANGES FOR THE TOGGLE SWITCHES ON A0

You should not have to do this if you use the recommended resistors.

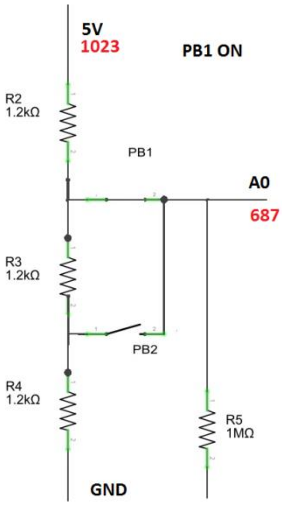
Working out the Toggle Switch Driver Resistor Network.
Must use 1MΩ resistor from A0 to GND, this would pull it low
The internal 20KΩ pull-up to 5V should return 1023 when no switches are on
Try to use 1.2KΩ 1% $\frac{1}{4}$W resistors

Reading of 1023 = 5V

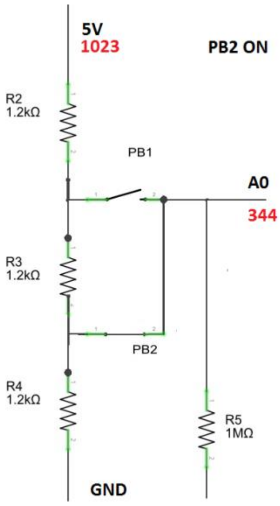
SW1/SW2 both OFF, Total R = 3.6 KΩ, I = 1.4mA



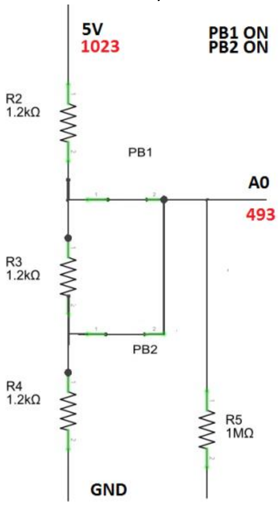
SW1 ON



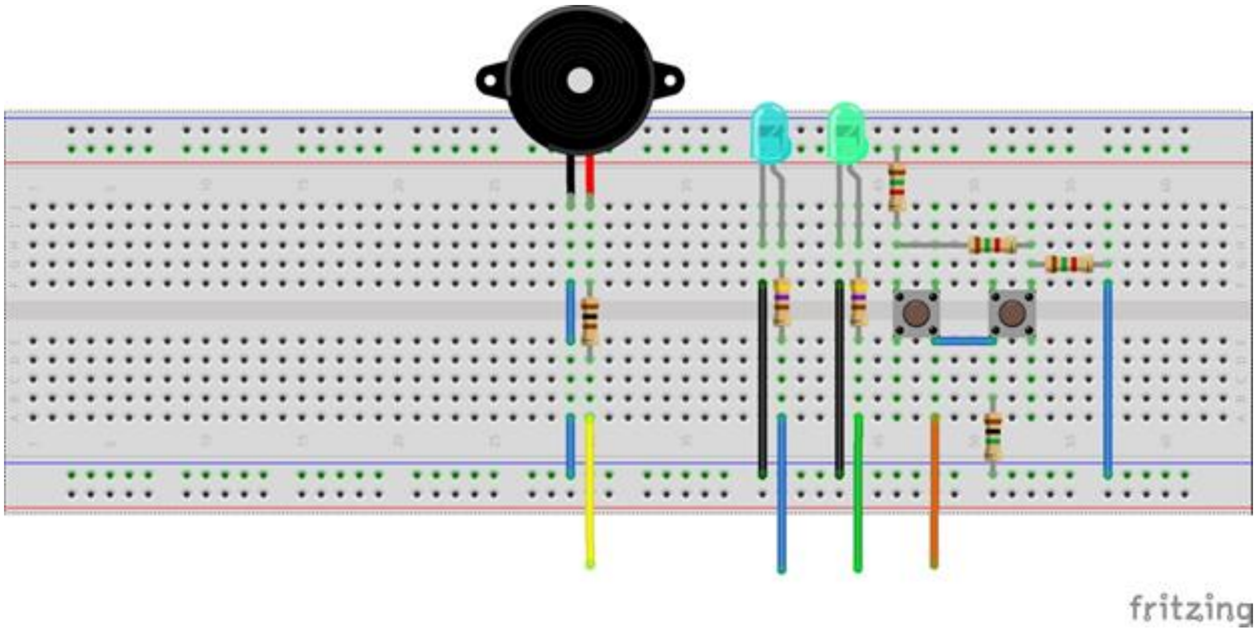
SW2 ON



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

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 wrong, then you will get strange results. So it is 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 neither button is pushed.

Basically the Arduino code 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

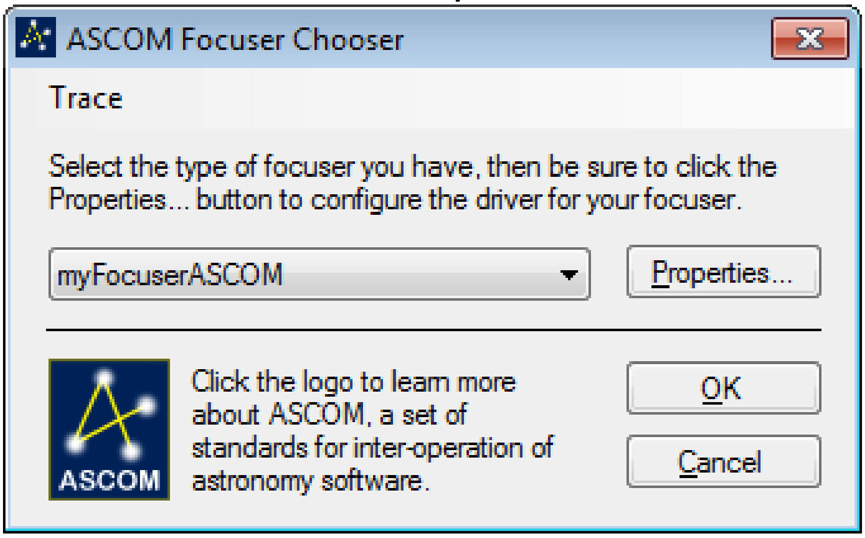
We then use these values to change the readpbswitches() code

```
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
}
```

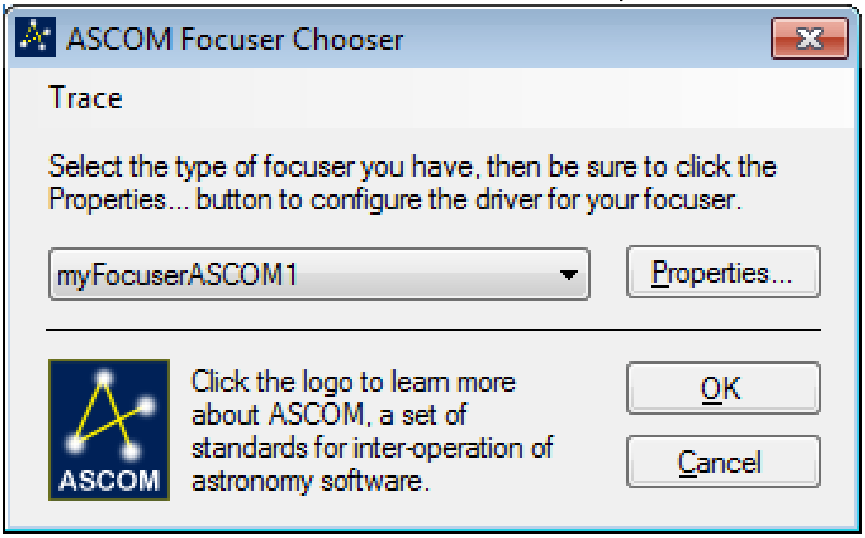
APPENDIX D RUNNING TWO FOCUSERS VIA ASCOM

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

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

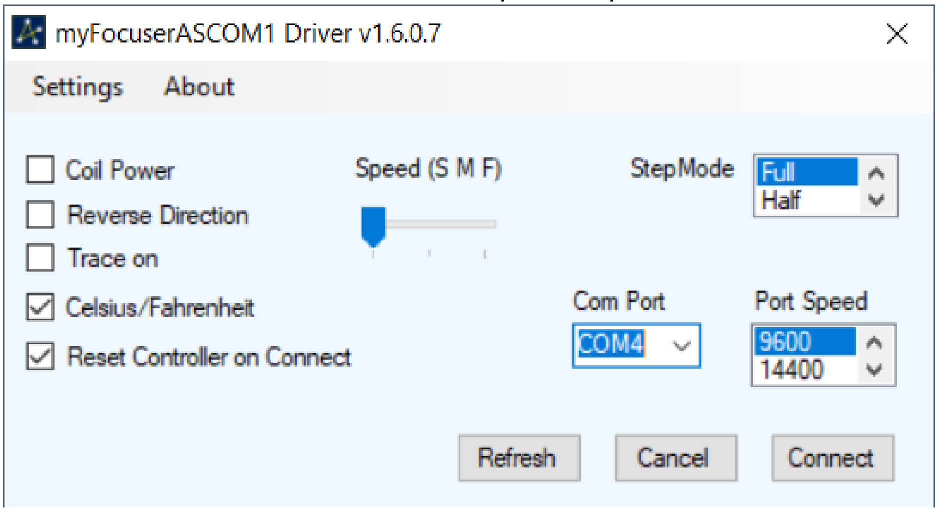


The **second** ASCOM driver (if installed) is known as **myFocuserASCOM1** in the chooser (and must connect to a different focuser controller than the first driver).



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 myFocuserASCOM to the DRV8825 controller by specifying COM PORT3 (or whatever the serial port is) under properties for driver myFocuserProASCOM, and specify myFocuserASCOM1 as the driver for the ULN2003 controller using COM PORT4 (or whatever the comport is) under properties for the second driver.

You can then work with both focusers independently.



To install the second ASCOM driver, you must run the setup program for the second ASCOM driver (myFocuserASCOM1Setupxxx.exe).

The second ASCOM driver setup dialog box is called **myFocuserASCOM1**.

APPENDIX E WHAT ABOUT STEP SIZE?

Step size is the amount in microns that the focuser travels for a single step of the stepper motor. The myFocuserPro implementation does not implement step size at this time (ASCOM driver returns not implemented exception).

Be aware that there can be no common value for this as each implementation is probably different, depending upon 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).

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

So 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

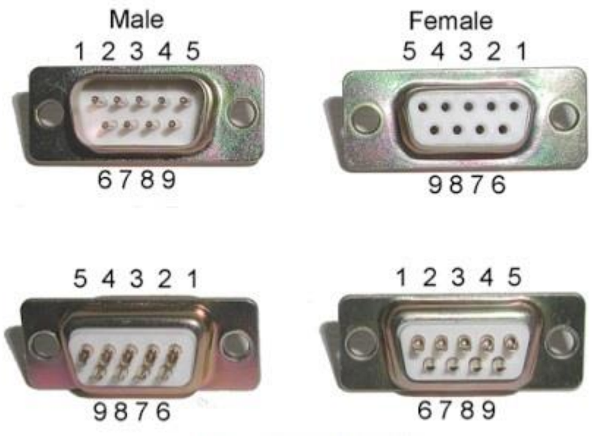
Note: 1mm = 1000 microns.

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

APPENDIX F STEPPER MOTOR TO CONTROLLER CONNECTION

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

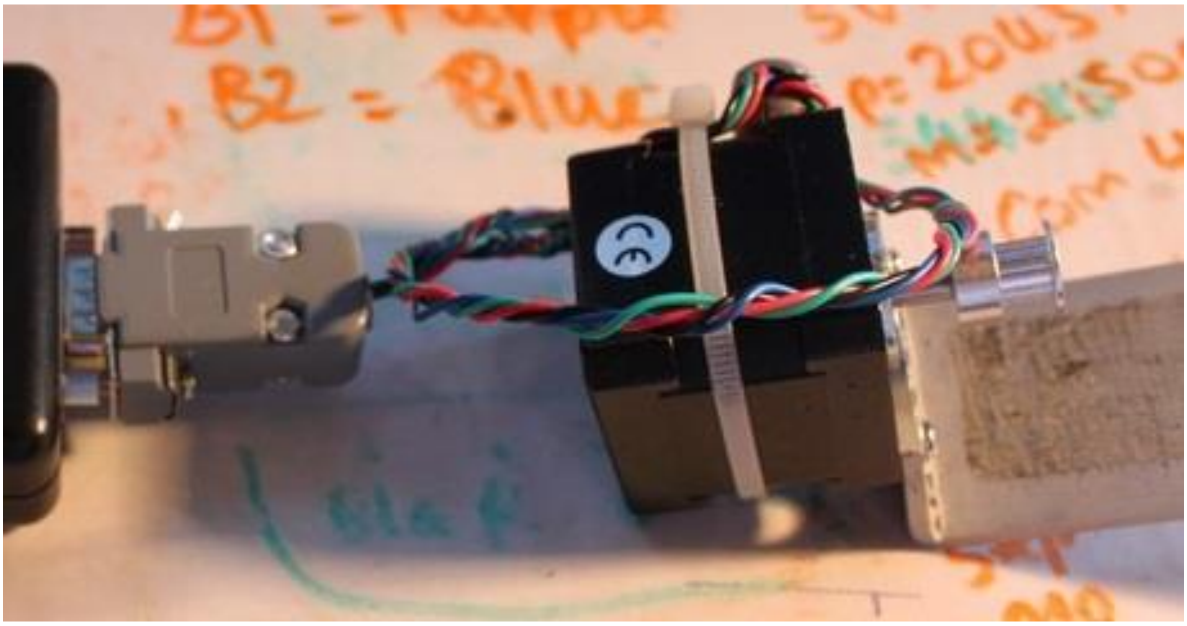
The stepper motor wires are terminated using a RS232 female connector. A TDK ferrite core clip-on cable clamp is clamped over the stepper wires (close to the stepper motor) to minimize back EMF.



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.

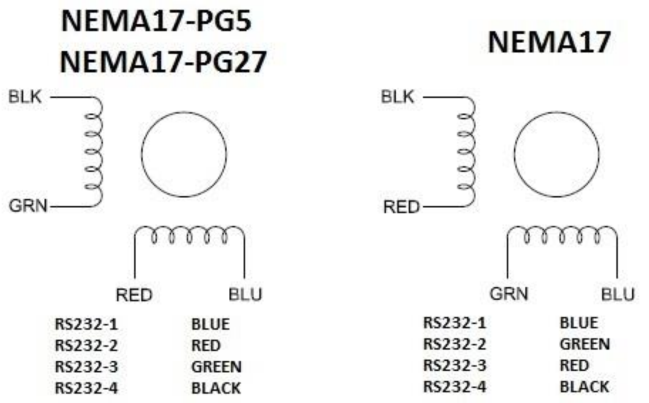
The maximum length of the cable for reliable operation depends on the characteristics of the cable wire, but should be kept as short as practical for reliable operation.



WIRING THE STEPPER MOTOR TO THE RS232 FEMALE CONNECTOR

RS232	28BYJ-48	NEMA17-PG5	NEMA17 HYBRID
Pin 1	YELLOW	BLUE	BLUE
Pin 2	BLUE	RED	GREEN
Pin 3	PINK	GREEN	RED
Pin 4	ORANGE	BLACK	BLACK
Pin 5	RED		

WIRING THE NEMA17 COILS TO RS232 CONNECTOR



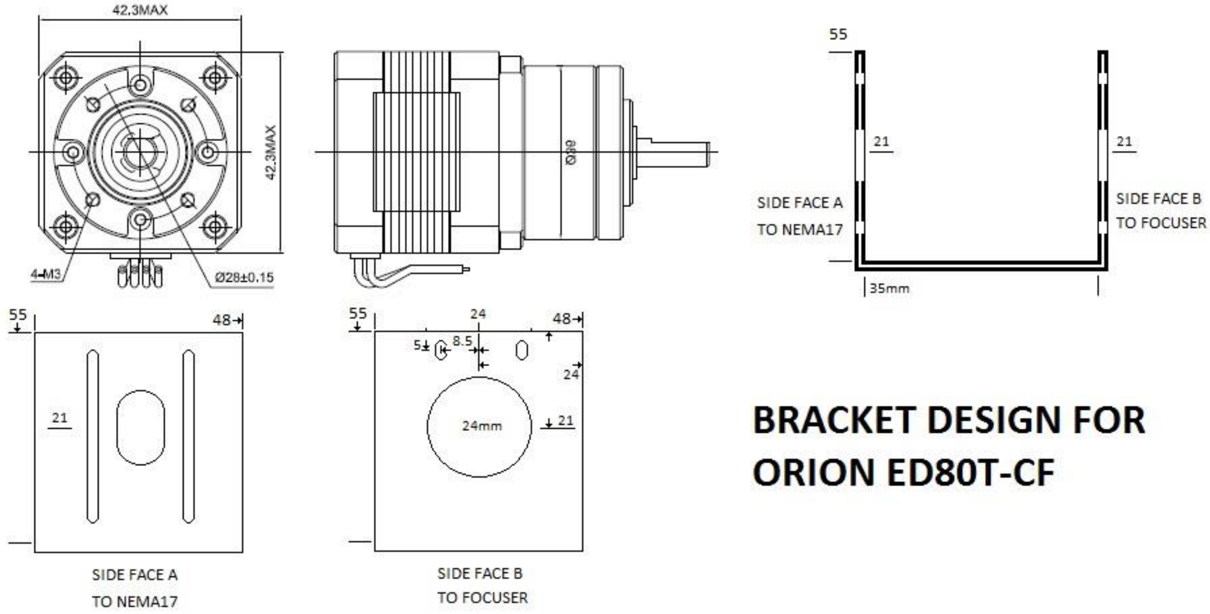
WIRING THE L293D SHIELD TO THE RS232 CONNECTOR

For wiring of the L293D shield to the RS232 connector, please see the sections "USING THE 28BYJ-48 STEPPER WITH THE L293D SHIELD" and "USING THE 28BYJ-48 STEPPER WITH THE L293D SHIELD" or the table below

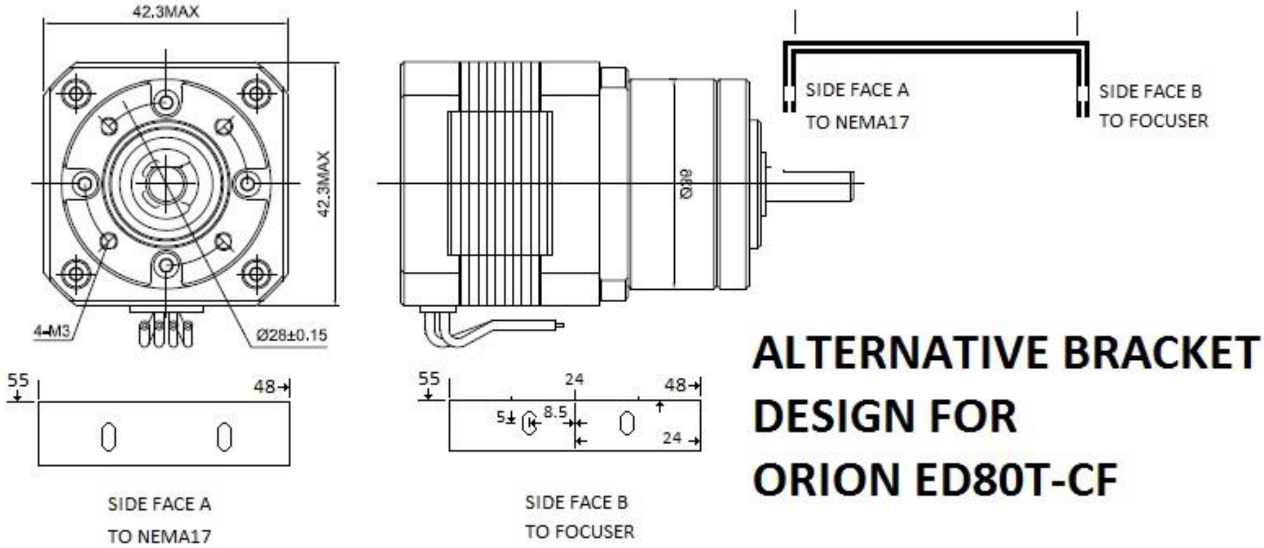
<table>
<tr><th>MotorPort-1</th><th>RS232Pin</th><th>Nema17-PG5</th><th>28BJY-48</th><td rowspan="4">Stepper Motor -> RS232 DB9pin Female Connector</td></tr>
<tr><td>M2</td><td>1</td><td>Blue</td><td>Yellow</td></tr>
<tr><td>M2</td><td>2</td><td>Red</td><td>Blue</td></tr>
<tr><td>GND</td><td>5</td><td>-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

APPENDIX G NEMA17-PG5 STEPPER MOTOR BRACKET

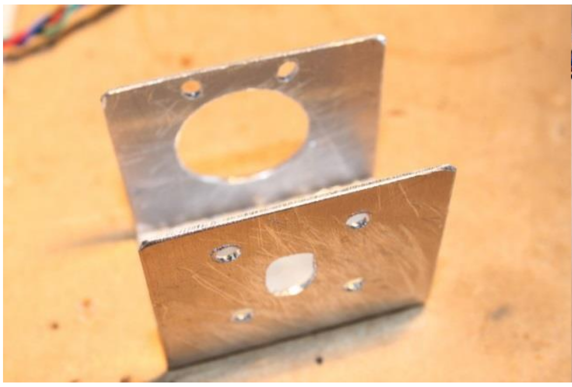
The NEMA17-PG5 connects via a home-made bracket to the focuser. The following diagram and photos show the U shaped bracket that is used on the Orion ED80T-CF refractor.

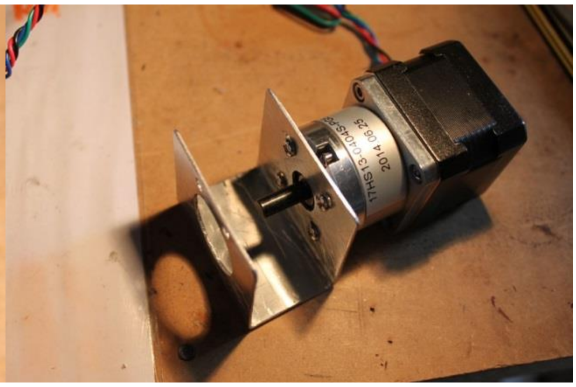


BRACKET DESIGN FOR ORION ED80T-CF



ALTERNATIVE BRACKET DESIGN FOR ORION ED80T-CF





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 belt ratio of 11.4 (eBay supplier speed_smart). This means that using the NEMA17 motor which has 200 full steps per revolution, it will take 2285 steps to rotate the belt ONE complete revolution. The actual rate of rotation of the focuser shaft will depend on the size of the pulley or diameter of the focuser knob. You will need to calculate this to determine if it is adequate for your system (refer section on critical focus)

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

Sometimes, depending upon the LCD purchased, the LCD could only print 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 myFocyser2Pro 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 myFocuserPro USED WITH CANON EOS LENS

This article discusses how the myFocuserPro 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 myFocuserPro 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

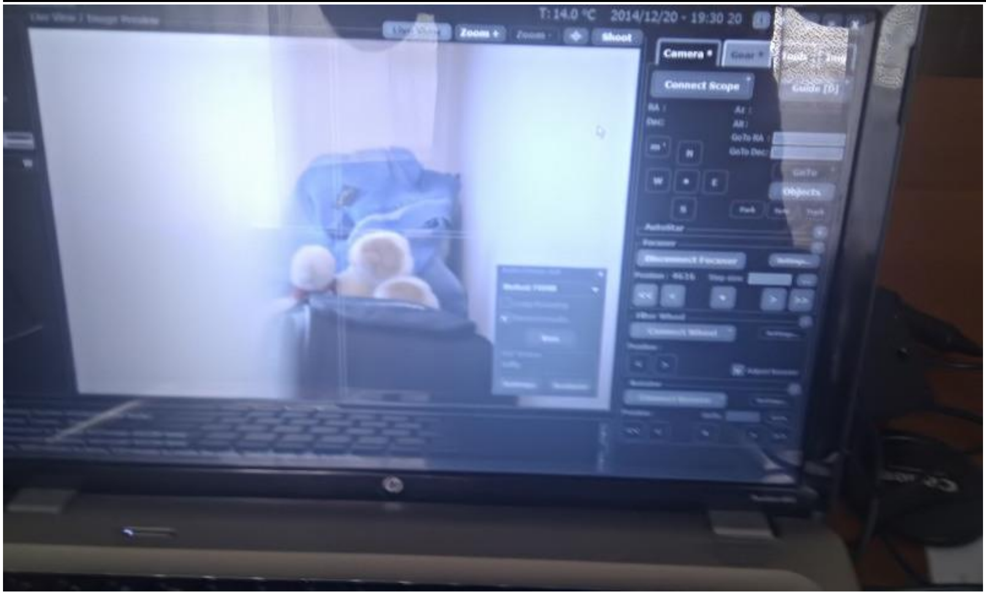
Using half-stepping with the 320mm belt thus gives 9142 steps per revolution of the Canon EOS Lens (each step = 0.039 degrees)

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 myFocuserPro 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, MaxSteps 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 myFocuserPro IN or OUT as required.

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



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

APPENDIX K NEMA17 STEPPER MOTORS AND VIBRATION

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.

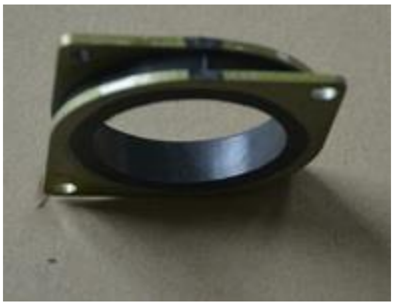
Vibration will increase with stepper motor speed. Thus final fine focusing should be done at the lower speed whilst coarse focusing can be done at medium or high speed.

In addition, direct coupled stepper motors will cause more vibration than a belt system as the flexible rubber 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 the 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 are effective at 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

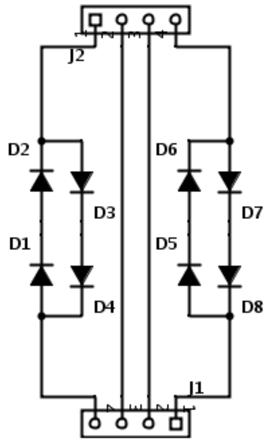
In operation I have not found the vibration of the NEMA17 stepper to be an issue

APPENDIX L 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. This 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