myFOCUSERPRO2

v301b – 16 October 2020
Applies to firmware 301+

A DIY project - stepper motor focuser solution based on Arduino

This document describes
- the range of myFP2 products
- building the myFP2 controller units
- attaching the myFP2 stepper motor to your telescope
- determining the correct myFP2 values for your telescope
- what to do if you lose your settings
- sample schematics and strip-board layouts
- the operation of Windows applications and ASCOM drivers available
- how to initially setup the myFP2 controller for your telescope
- testing procedures and programs
- manual and automated focusing methods

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

Note:

myFocuserPro refers to v1xx of the myFocuserPro products. This was the first design and 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)

myFP2 refers to v2xx of the myFocuserPro2 products. This design is v2 and is NOT compatible with Mac and Linux systems. Version 2 controllers do have a number of additional features that are not available on the v1 design and use a different protocol. You cannot use v2xx ASCOM or Windows software with a v1xx programmed controller or vice-versa.

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

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

CONTRIBUTIONS

Please contribute to the ongoing development of this project via PayPal and send your donation to user rbb1brown@gmail.com (Robert Brown). All contributions are gratefully accepted.

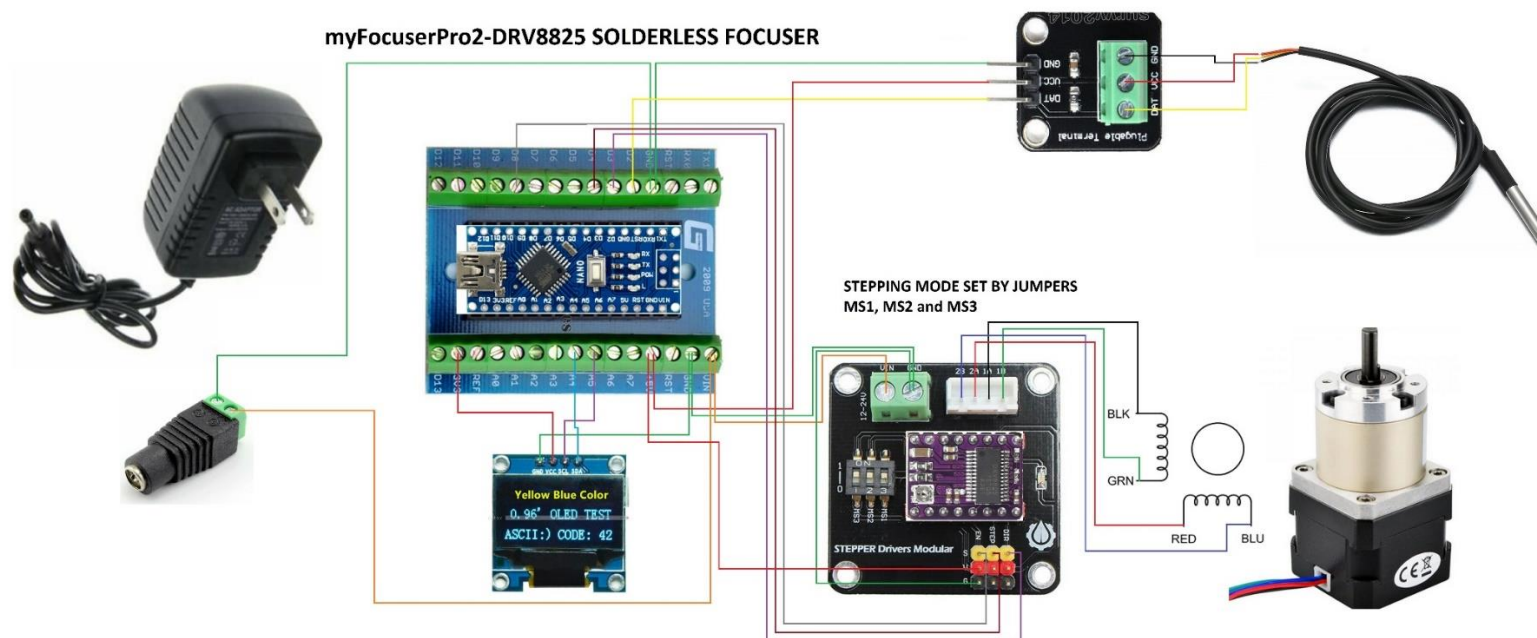# SOLDERLESS DRV8825 myFOCUSERPRO2 ASCOM FOCUSER SOLUTION

NOT COMFORTABLE SOLDERING?
LIMITED SOLDERING SKILLS?
PREFER TO JUST CONNECT WIRES USING SCREW TERMINALS?

NOW YOU CAN.

Please see
https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/STRIPBOARDS/SOLDERLESS%20DRV8825%20NANO/
for building a solderless myfocuserpro2 drv8825 ASCOM focuser solution.

myFocuserPro2-DRV8825 SOLDERLESS FOCUSER

STEPPING MODE SET BY JUMPERS
MS1, MS2 and MS3

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.
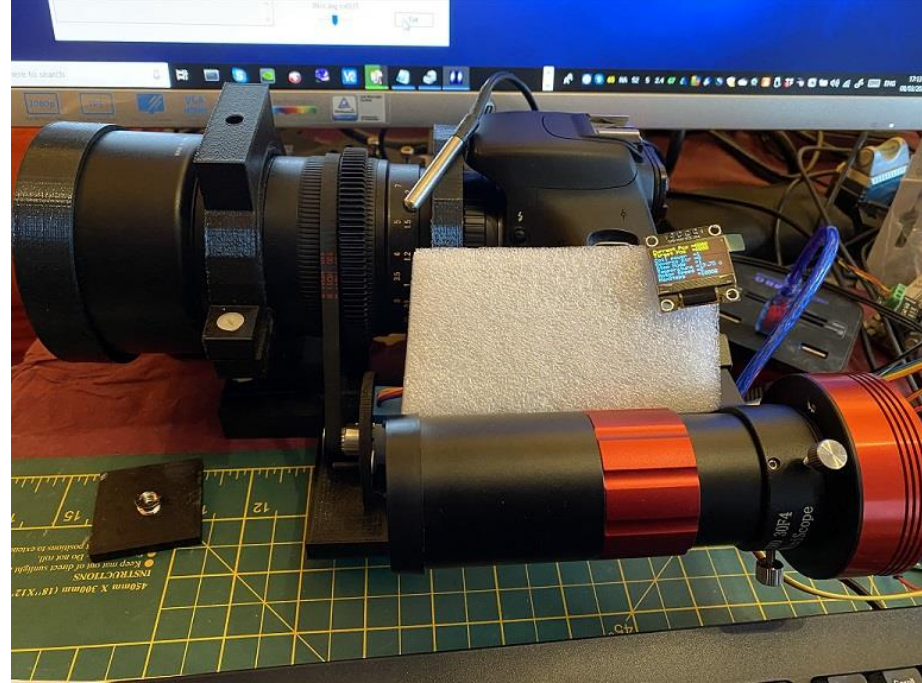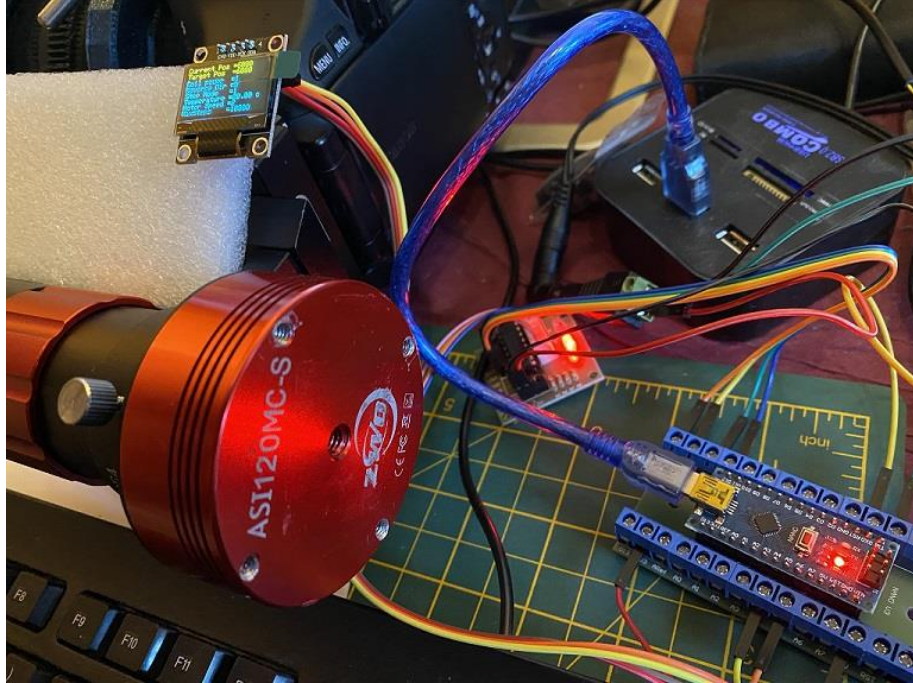
SOLDERLESS MYFP2-ULN2003 BY ACAPULCO ROLF

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. The author(s) accept no responsibility to damage caused to any equipment or goods or self by using the ideas, schematics and code associated with this project, or loss of income or all other losses that may be incurred. No warranty is offered or implied.

COPYRIGHT RESTRICTIONS

The schematic, firmware and other 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(s).

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

FIRMWARE LICENSE RESTRICTIONS

This code is released under license. If you copy or write new code based on the code in these files you must include a link to these files AND you must include references to the author(s) of this code.
Firmware is released subject to the following restrictions

© Copyright Robert Brown 2014-2019. All Rights Reserved.
© Copyright Holger M, 2019, who wrote large portions of code for state machine, SPIFFS and COMMS
© Copyright Pieter P - OTA code based on example from Pieter P
All rights reserved

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), the rights to use, copy, modify, merge, publish, distribute, sublicense copies of the Software (from the authors), **subject** to the following conditions:

1. The above copyright notice shall be included in all copies or substantial portions of this Software.

2. The software may NOT be sold or charged for in any way, either as part of a package, license fee, annual charges or premiums and shall include the copyright notice above.

3. The source-code to any software MUST be made available in full source-code form (including modified source files) and free to download without payment and include the copyright notice above

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

OVERVIEW

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

myFP2 SPECIFICATIONS

- ASCOM driver, tested with FocusMax, Nebulosity, APT, NINA
- Windows Application (and Mini Application) for manual/remote control of focuser
- INDI driver
- UNIX and Android applications available
- Bluetooth and WiFi options
- Display options (LCD1602/OLED/TFT) for positional information (Current and Target positions etc)
- Push button option (x2) for manual control of IN and OUT
- Keypad option
- LED indications for IN, OUT and External PWR
- Absolute focusing
- Temperature probe and Temperature Compensation
- Stepsize support
- Key focuser parameters saved in EEPROM
- Highly configurable
- External power supply required for driving stepper motor 12VDC @ 2A (ULN2003 requires 9V)
- Reverse voltage protection on external 12VDC input
- Multiple driver board versions are available. Two versions detailed in this document – for other driver boards please go to the driver boards folder on Sourceforge
- PCB's (double sided, plated through holes, silk masked) available for all options

myFP2 VARIATIONS

The myFP2 comes in a variety of different models.

myFP2

- The standard myFP2 focus controller based on an Arduino Nano controller with support for a wide variety of driver boards and configurations
- https://sourceforge.net/projects/arduinoascomfocuserpro2diy/

myFP2N

- A basic myFP2 focus controller with a Nextion touch screen display (DRV8825 and ULN2003 drivers)
- https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/myFP2N%20NEXTION%20TFT%20DISPLAY/

myFP2ESP

- The latest line of myFP2 controllers, based on the ESP8266 and ESP32 WiFi chip, supporting DRV8825, ULN2003, L298N, L293DMini driver boards, Push buttons, IN/OUT LEDs, OLED display, WiFi operation via TCP/IP, Webserver and ASCOM REMOTE server)
- https://sourceforge.net/projects/myfocuserpro2-esp32/

myFP2 CONTROLLER

This is the controller as outlined in this PDF. The controller consists of a PCB featuring an Arduino Nano and a motor driver board [for the DRV8825 the PCB has a socket mount for the driver chip; all other boards use a separate driver board].
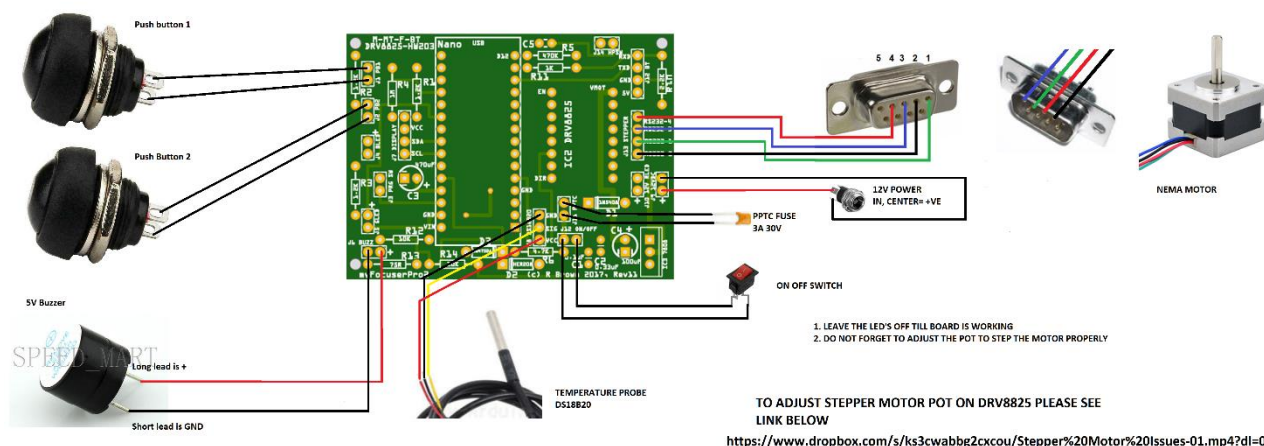
You will first need to determine your focuser requirements as outlined here. This will help you determine what type of controller and stepper motor you will need. Once you have finalised that information, you can then make a judgment call on what PCB and driver chip is required.

The myFP2 controller
- Interfaces with either USB Serial [COM port] or Bluetooth
- Requires 12V to work
- ASCOM drivers, INDI drivers, Windows/LINUX/Android apps available
- There are additional options for
  o Push Buttons
  o IN-OUT Direction LED's
  o Displays [LCD, OLED, TFT, Nokia 5110]
  o Temperature Probe
  o Infra-red remote controller
  o Rotary encoder
  o Home Position Switch
  o Stepper Power Detection

Some options may require a specific PCB. Some options are not compatible with other options [Like either Rotary encoder OR Infra-red remote controller]. If you need help, please ask on the Discussion page or by sending me an email.

myFP2M CONTROLLER

The myFP2M controller is a basic myFP2 Controller, either in DRV8825 [supporting NEMA stepper motors] or ULN2003 [supporting 28BYJ-48 stepper motors.

The controller is ready built, programmed and fully tested. The controller package includes

- 12V cable connector
- Temperature Probe
- Stepper Driver Smoother Diodes [for DRV8825]
- Assembled in 3D printed case
- 3D printed focuser brackets available
- Bluetooth option available
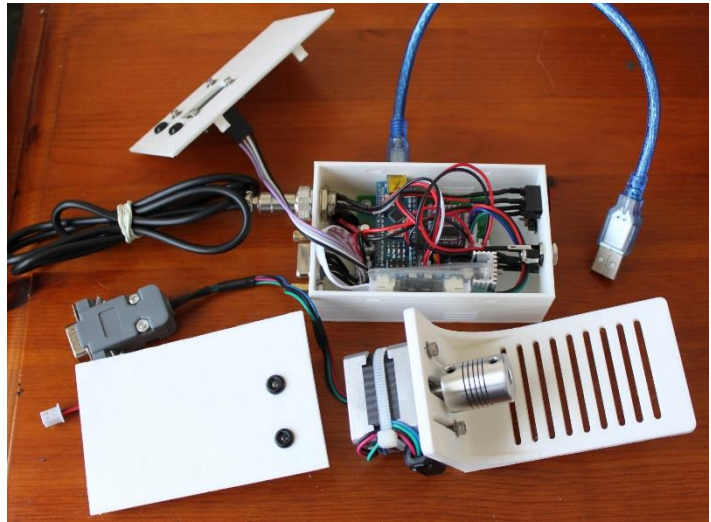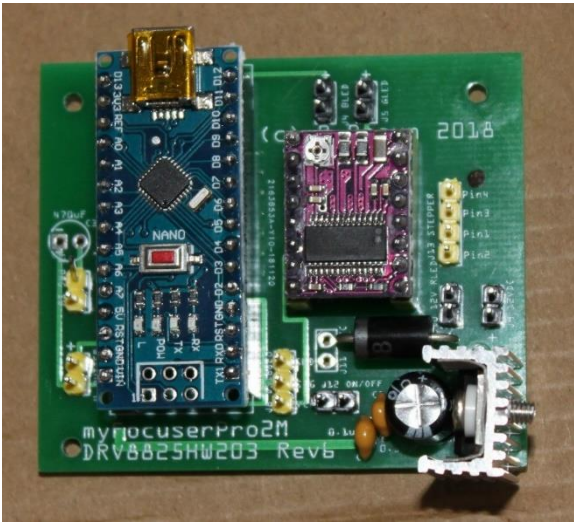
The myFP2M is no longer provided with an OLED display. If you are interested please click here.

myFP2ESP [ESP8266/ESP32] WiFi CONTROLLER

This is a WiFi based controller (not Arduino) that can interface to a focuser motor and provide automated focusing.

Available on the Sourceforge site is

    Documentation
    Parts List
    PCB
    Firmware support for
        ACCESS POINT/STATION MODER via TCP/IP
        Webserver
        ASCOM REMOTE server
        Bluetooth support
        Local Serial support via USB
    Windows controller application
    ASCOM driver
    Android App
    3D printed case

Please see the link below for further information
https://sourceforge.net/projects/myfocuserpro2-esp32/
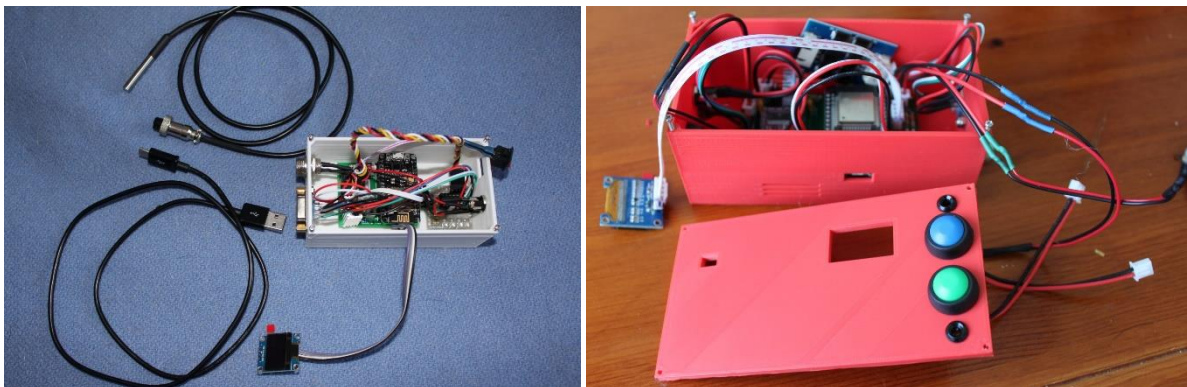
ESP32 Videos

Part 1	ULN2003	https://www.youtube.com/watch?v=o1UVqP6QM2Q
Part 2	DRV8825	https://www.youtube.com/watch?v=7UhIwJS8AEg
Part3	L293D	https://www.youtube.com/watch?v=pJeu5f40Hko

myFP2N-Nextion Touch Screen Display

This update comprises the following
- 3.5" TFT NEXTION ENHANCED DISPLAY NX4832K035_011R 480x320 (Touch screen)
- Nextion firmware
- myFP2 modified firmware for use with Nextion display

The TFT could be mounted on the outside of the focuser or connected using a 4P cable (GND, 5V, RXD, TXD) to an existing myFP2-DRV8825-HW203 controller.

This could be valuable addition to those who operate their telescopes in the field and are not using a laptop or have limited battery power available.

Control is based on a simple GUI comprising a number of screens. Most functions have been implemented.

For further information,
https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/NEXTION%20TFT%20DISPLAY/
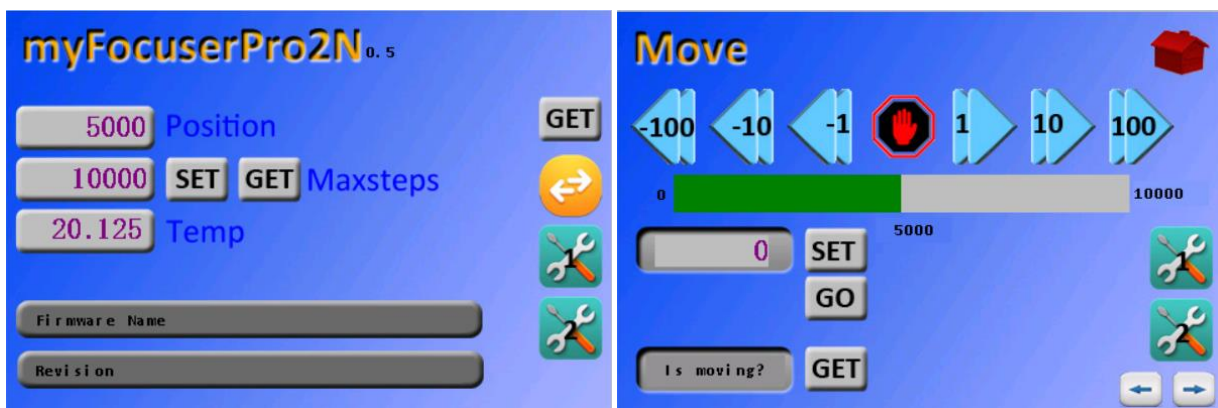
For a video demonstration
https://youtu.be/NYa4qjsf1uc

myFP2 SUPPORT VIDEOS

The following list of videos provide vital information about the myFP2 focus controller.

If you are considering building a myFP2 controller, please watch the Part1a and Part1b videos as a minimum. Part1a will help you decide what you need and Part1b shows how to build a focuser based on a PCB. Be sure to read the section Focuser Basics in this PDF

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

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

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

Part1d Adjusting the stepper motor current of the myFP2 controller
https://youtu.be/u2Z0dFaiiyM

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

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

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

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

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

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

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

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

Part3f Controlling more than one myFP2 controller
https://www.youtube.com/watch?v=sEvvWYNMCFs

Part4 myFP2 Home Position Switch
https://www.youtube.com/watch?v=ADi2W0nsypI

Part5a myFP2 Temperature Compensation
https://www.youtube.com/watch?v=YXRqP-V1fcM

Part5b Understanding temperature compensation
https://youtu.be/uIEgBXL9Cks

Part6 Using a 4x4 keypad with myFP2
https://youtu.be/Xws4XfaG-3s

myFP2 Bluetooth Android App
https://www.youtube.com/watch?v=I7mUmHR_bx8

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

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

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

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

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

myFP2 FAQ

Ensure you read the Frequently Asked Questions [FAQ] document.

https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/Documentation/myFocuserPro2FAQ.pdf/download

This document contains explanations for many issues.

myFP2 - BUILDING YOUR OWN CONTROLLER

The myFP2 focus controller uses an Arduino Nano controller [CH340G] and a driver chip. The driver chip controls the stepper motor.

For each driver chip you require a different printed circuit board. The circuit boards are listed below.

OPTION NOTATION

M = Minimal, LED's to indicate IN/OUT
MT = Minimal + Temperature Probe
F = FULL, includes LCD, temperature probe, Push buttons, LED's etc
BT = Bluetooth option
IR = Infrared controller option
RE = Rotary encoder option

MYFP2 PCB CAN BE ORDERED ONLINE

Gerber files for each PCB are located at
https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/Gerber%20Files/

The **Gerber Files** folder also contains the schematics, board layout and parts lists.

After downloading the free GERBER file for your chosen controller, go to a PCB site like EasyEDA to order the board online. Generally, 5 PCB boards will cost around $2USD excluding shipping charges. These PCB sites have nothing to do with me or the project; they are totally independent.

I would suggest building two controllers [1 as spare] and perhaps giving the other boards to friends or donating a controller to your local astronomy club. Helping others is what this project is really about.

Sample Wiring Detail for DRV8825-HW203-M-MT-F-BT PCB
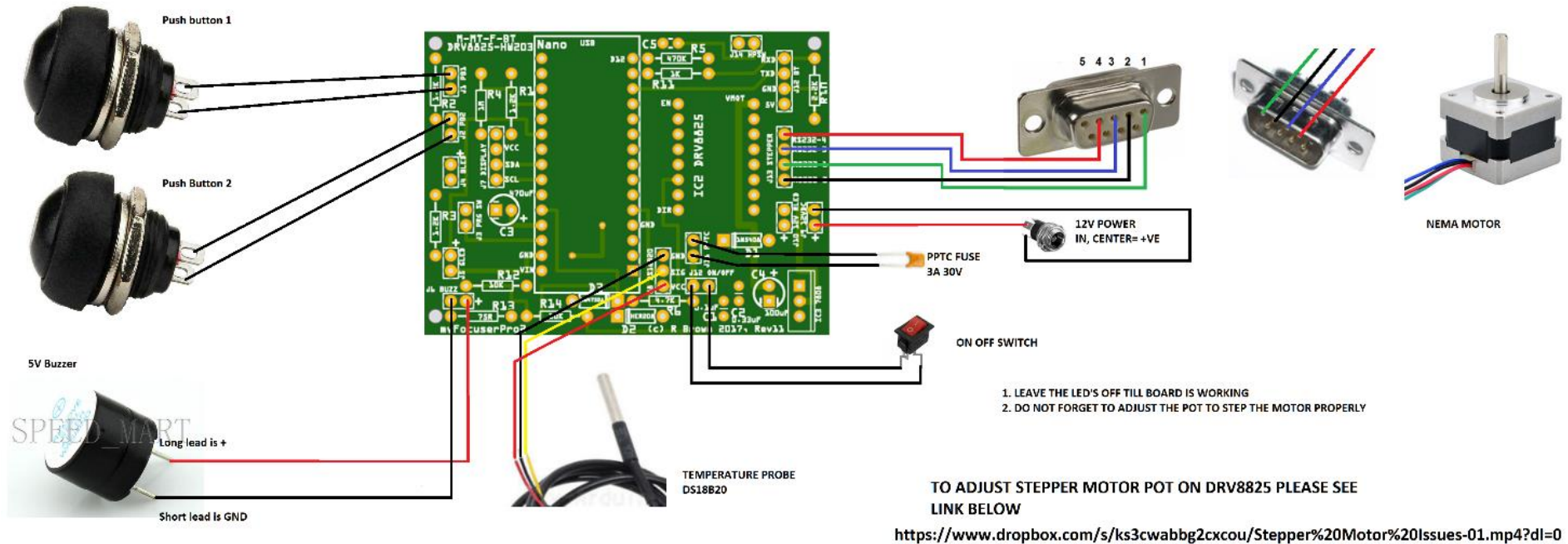
Push button 1

Push Button 2

5V Buzzer

Long lead is +

Short lead Is GND

5 4 3 2 1

NEMA MOTOR

12V POWER IN, CENTER= +VE

PPTC FUSE
3A 30V

ON OFF SWITCH

1. LEAVE THE LED'S OFF TILL BOARD IS WORKING
2. DO NOT FORGET TO ADJUST THE POT TO STEP THE MOTOR PROPERLY

TEMPERATURE PROBE
DS18B20

TO ADJUST STEPPER MOTOR POT ON DRV8825 PLEASE SEE
LINK BELOW
https://www.dropbox.com/s/ks3cwabbg2cxcou/Stepper%20Motor%20Issues-01.mp4?dl=0

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.

The PCB lets you build a minimal or complete (full or all options) focuser. You decide what options to include. Or add them later.
Option M: Leave out push buttons, display, temperature probe
Option MT: leave out push buttons, display
Option F: Include every-thing

FOCUSER BASICS
HARDWARE SELECTION
BUILD INSTRUCTIONS
SETUP AND CONFIG INSTRUCTIONS
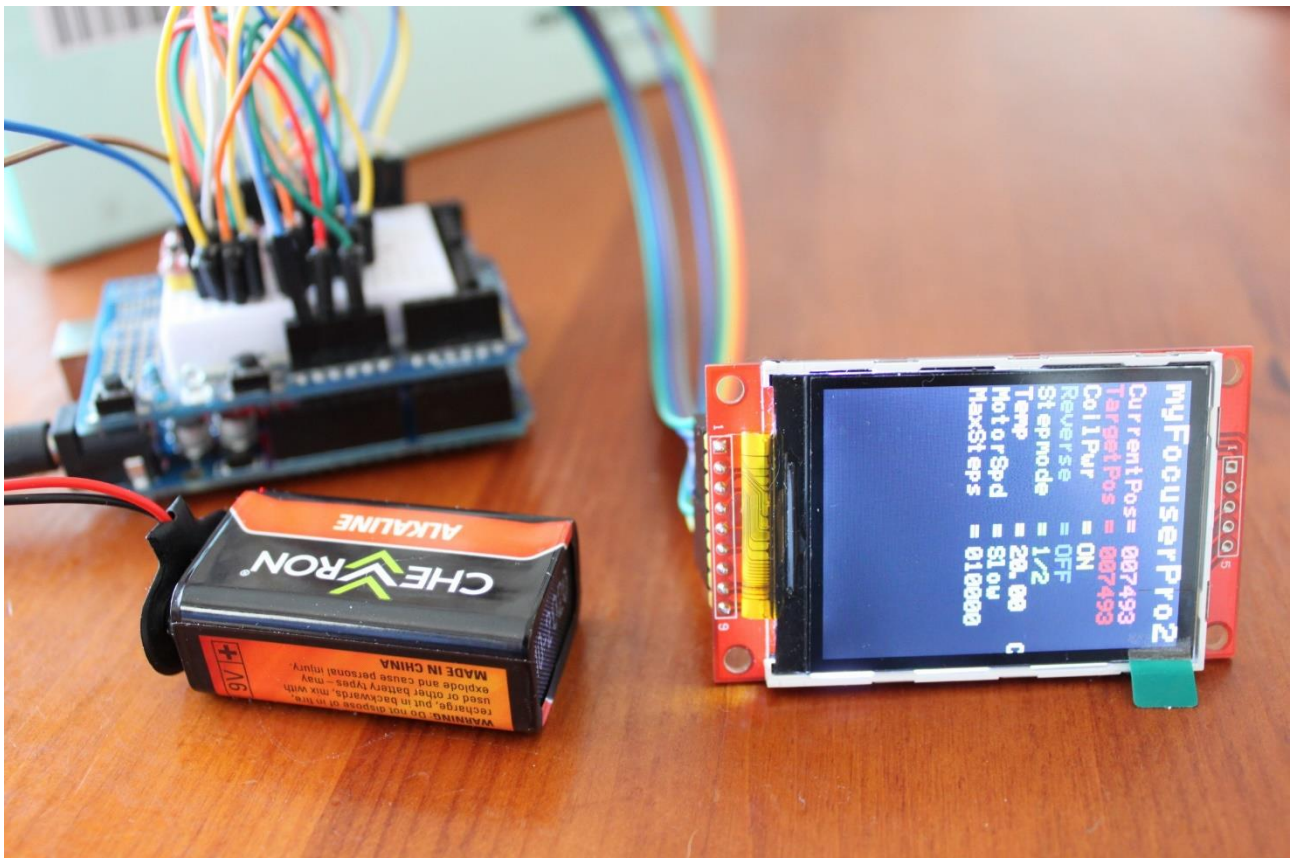
myFocuserPro2 with 1.8" TFT color display

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 step-size 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 motor steps within the CFZ (we recommend 10), 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 about 10 steps within the CFZ. This is done by altering the gearing ratio (such as using gears or a pulley belt system) or using micro stepping to increase the number of steps per revolution of the stepper motor.

MEASURING FOCUSER DISTANCE PER ONE REVOLUTION OF FOCUS KNOB

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

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

MEASURING STEP SIZE

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

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

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

Step Size is supported by myFP2 controllers. Some applications may require a valid setting for Step-Size in order to work correctly. If enabled, then the Step-Size value stored by the controller will be returned. If Step-Size is not enabled, the ASCOM driver will throw a "Not implemented" exception which the client application should handle. For more information in this PDF, click here. You can specify the Step-Size and enable/disable this setting when connecting to the controller. Setting values are remembered.

Calculating CFZ, Step Size and Stepper Motor Resolutions

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

Spreadsheet Calculations Showing Various Options

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

SV102T with NEMA17PG5 STEPPER MOTOR

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

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

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

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

609.4117647 Full steps per cm
1218.823529 Half Steps per cm

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

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

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

ED80T-CF with myFocuser 28BYJ-48 Stepper

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

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

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

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

As you can see, this will give 9 full-steps within the CFZ at f6 and 6 steps at f4.8. Using half-steps, this would 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 myFP2 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 myFP2 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 myFP2 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 myFP2 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 myFP2 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 myFP2 Windows application. You will see that the focuser position will be set to the midway point (in our example 3108) and that the maximum position is set to 6216.

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

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

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

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

STEP SIZE AND CRITICAL FOCUS ZONE

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

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

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

> CFZ = 10 * 10 * 2.2
> CFZ = 220 microns

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

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

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

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

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

Please watch this video to review the information on CFZ and stepsize and how to calculate this.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Q: For an SCT, which direction is IN and OUT?

With a conventional refractor IN means the focuser drawtube moves inwards against gravity if the tube where pointing upwards to the sky. In an SCT IN means when the primary mirror is moving towards the corrector plate (ie against gravity). Maxsteps would of course be from full one way to full the other way with full one way being 0 and full the other way being Maxstep.

With an SCT following the pdf you calculate how much distance is travelled by the primary mirror for one turn of the focus knob (to calculate the step size and thus how many steps you have in the critical zone), so if there were 80 turns then it is 80 * (number of steps of motor to move focuser knob one full turn).

That is ONLY if you wanted to operate the SCT over the full range. You can of course operate it over such a smaller range of say 10 [or even 5] focuser knob revolutions by first positioning the focuser at focus, then step in 5 turns and set that to 0, and out 10 turns and set that to maxstep.

Focusing is normally done over probably less than a couple of turns of the focuser knob.

STEPPER MOTORS

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

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

CHOOSING A STEPPER MOTOR DEPENDS UPON A NUMBER OF FACTORS

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

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

If you decide not to use the recommended motors suggested here, then make sure that the current rating (*2) of your stepper motor 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
NEMA14	400mA	12V	400

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

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

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

AVAILABLE/RECOMMENDED STEPPER MOTORS

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

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

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

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

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

CONCERNED ABOUT STEPPER MOTOR SIZE AND WEIGHT

An alternative is the NMEA14 which is much lighter but uses 0.9° 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.

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

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
NO	Implement the Minimal option

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.

RECOMMENDED BUILD OPTION

The recommended builds are
- PCB (supports Full, Minimal plus Temperature probe and Minimal options)
- DRV8825 driver chip with a NEMA17PG5 (or NEMA17PG27, or NEMA14)

We recommend building a focus controller using a PCB, the DRV8825_HW203 PCB

DIY BOARD OPTIONS

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

DECIDING ON HARDWARE AND FIRMWARE TO USE

Board	Driver	Full?	Stepper	Stepping	.ino file
Nano	A4998	F	Nema17PG5/27	F/H/4/8/16	Focuservxxx_A4998
Nano	**DRV8825/HW203**	**All**	**Nema17PG5/27**	**F/H/4/8/16/32**	**Focuservxxx_DRV8825_HW203**
Uno	L293D	F	28BYJ-48	F/H	Focuservxxx_L293D
Uno	L293D	F	Nema17PG5/27	F/H	Focuservxxx_L293D
Nano	L293DMini	F	Nema17PG27	F/H	Focuservxx_L293DMini
Nano	L298N	M	Nema17PG27	F/H	Focuservxxx_L298N
Nano	L9110S	F	Nema17PG27	F/H	Focuservxxx_L9110S
Nano	RAPS128/HW203	All	Nema17PG5	F/H/4/8/16/32/64/128	Focuservxxx_RAPS128_HW203
Nano	ST6128	All	Nema17PG5	F/H/4/8/16/32/64/128	Focuservxxx_ST6128
Nano	TB6612FNG	F	Nema17PG27	F/H	Focuservxxx_TB6612FNG
Nano	ULN2003	F	28BYJ-48	F/H	Focuservxxx_ULN2003
Nano	ULN2003	M	28BYj-48	F/H	Focuservxxx_ULN2003

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

ASCOM DRIVER	myFocuserASCOMSetupxxx
WINDOWS APPLICATION	myFocuserWin_xxx

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

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

PRINTED CIRCUIT BOARD OPTIONS

There are a number of PCB designs (double sided, plated through holes, silk screened, professionally made) that can be ordered online.

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 (no display, no push buttons, no temperature probe)
- 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 PCB for the UNO+L293D Shield.

Nomenclature
TP = Temperature Probe
PB = Push buttons for manual control
HPSW = Home Position Switch
NEMA = NEMA stepper motor
IR = Infrared controller
RE = Rotary encoder
BT = Bluetooth

PCB	Driver	Display	TP	PB	HPSW	NEMA	IR	RE	BT
A4998-M-MT-F-BT	A4998	LCD	Y		Y	Y			Y
DRV8825HW203-M-MT-F-BT	DRV8825	LCD/OLED	Y	Y	Y	Y			Y
DRV8825HW203_FIRBT	DRV8825	LCD	Y	Y	Y	Y	Y		Y
DRV8825HW203_FRE	DRV8825	LCD	Y		Y	Y		Y	
EASYDRIVER-HW203-M-MT-F-BT	EASYDRIVER	LCD	Y	Y	Y	Y			Y
EASYDRIVER-HW203_FRE	EASYDRIVER	LCD	Y			Y		Y	
L293DMINI-M-MT-F-BT	L293DMINI	LCD/OLED	Y	Y	Y	Y			Y
L293D-M-MT-F-NOKIA	L293D	5110	Y	Y		Y			
L298N-M-MT-F-BT	L298N	LCD	Y	Y	Y	Y			Y
L9110S-M-MT-F-BT	L9110S	LCD/OLED	Y	Y	Y	Y			Y
L9110S-M-MT-NOKIA	L9110S	5110	Y	Y		Y			
TB6612FNG-M-MT-F-BT	TB6612FNG	LCD	Y	Y	Y	Y			Y
ULN2003-M-MT-F	ULN2003	LCD	Y	Y	Y				Y
ULN2003-M-MT-NOKIA	ULN2003	5110	Y	Y					

// NOTE 1: YOU CANNOT USE A STRAIGHT 4P-4P HEADER CABLE FOR THE OLED DISPLAY
// NOTE 2: IF USING LCD, USE A STRAIGHT 4P-4P HEADER CABLE

PCB DRV8825HW203-M-MT-F-BT

This is the recommended PCB and supports

>DRV885 with up to 1/32 micro-stepping with a NEMA stepper motor
>Buzzer, IN/OUT LEDS and PWR LED
>IN/OUT Push buttons (optional)
>Temperature Probe (optional)
>Display (OLED and LCD) (optional)
>Power-On reset circuitry
>Bluetooth (optional)
>Home Position switch (optional)
>Voltage regulator for operating focuser from 12V

With this PCB you can choose to implement a minimal PCB (just a focuser) or a fully implemented focuser with display, push buttons and temperature probe.

DRV8825HW203-M-MT-F-BT SCHEMATIC

myFocuserPro2 DRV8825 HW203-M-MT-F-BT
(c) R Brown 2017-2019
Revision 11, Feb 2019
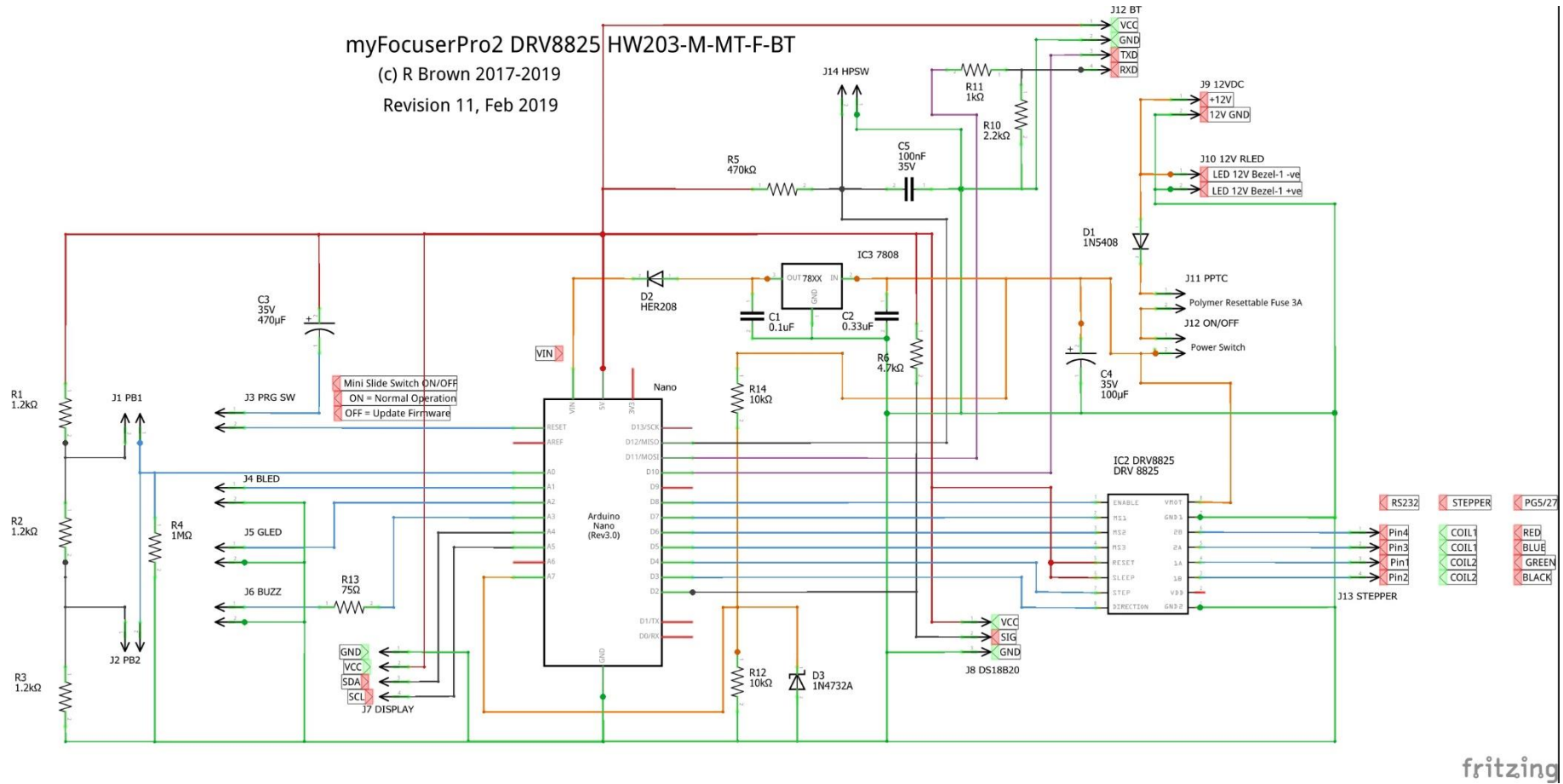
Not all options need to fitted to the board. You can leave options out and then configure the software to match the options you have added to the board.

PCB DRV8825HW203-M-MT-F-BT
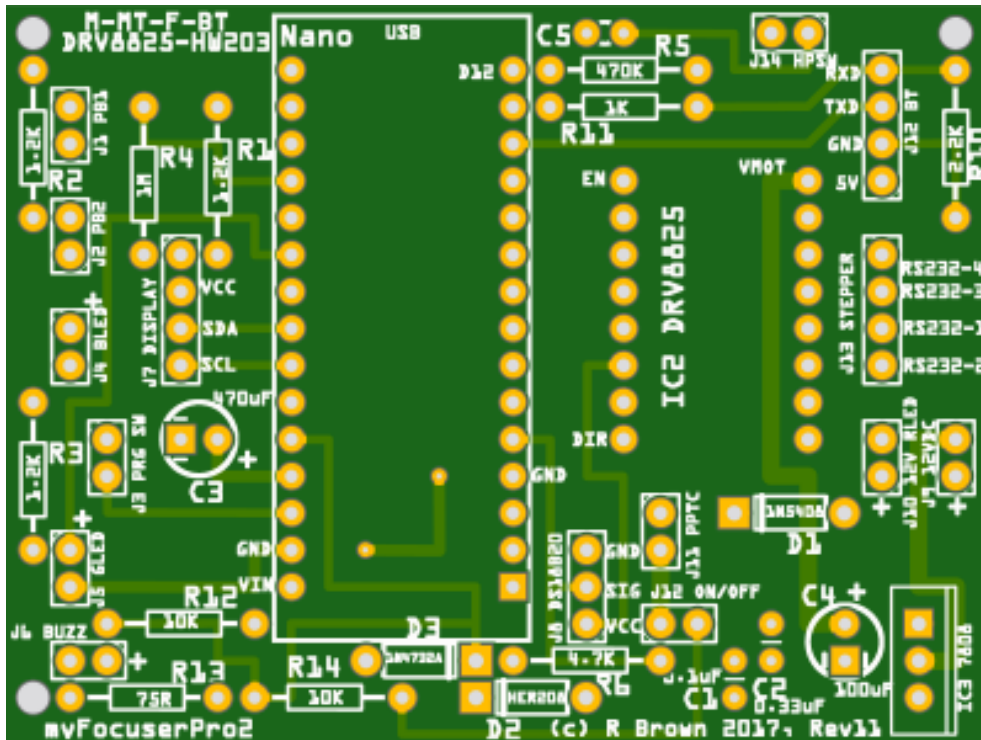
If you want to leave off some options, the table below identifies the hardware option and associated components

Push Buttons	Push button 1 and 2, R1, R2, R3, J1, J1. Note: You must fit R4
Display	Display, J7, 4P to 4P header cable
Temperature Probe	DS18B20 probe, R6, J8
Stepper Power Detect	R12, D3
IN and OUT LED's	Blue and Green 5V Prewired LED, J4, J5
Buzzer	Continuous tone buzzer, R13, J6
Home Position Switch	Small momentary switch, R5, C5, J14
Bluetooth HC05	HC05, R10, R11, J12

WIRING DRV8825HW203-M-MT-F-BT
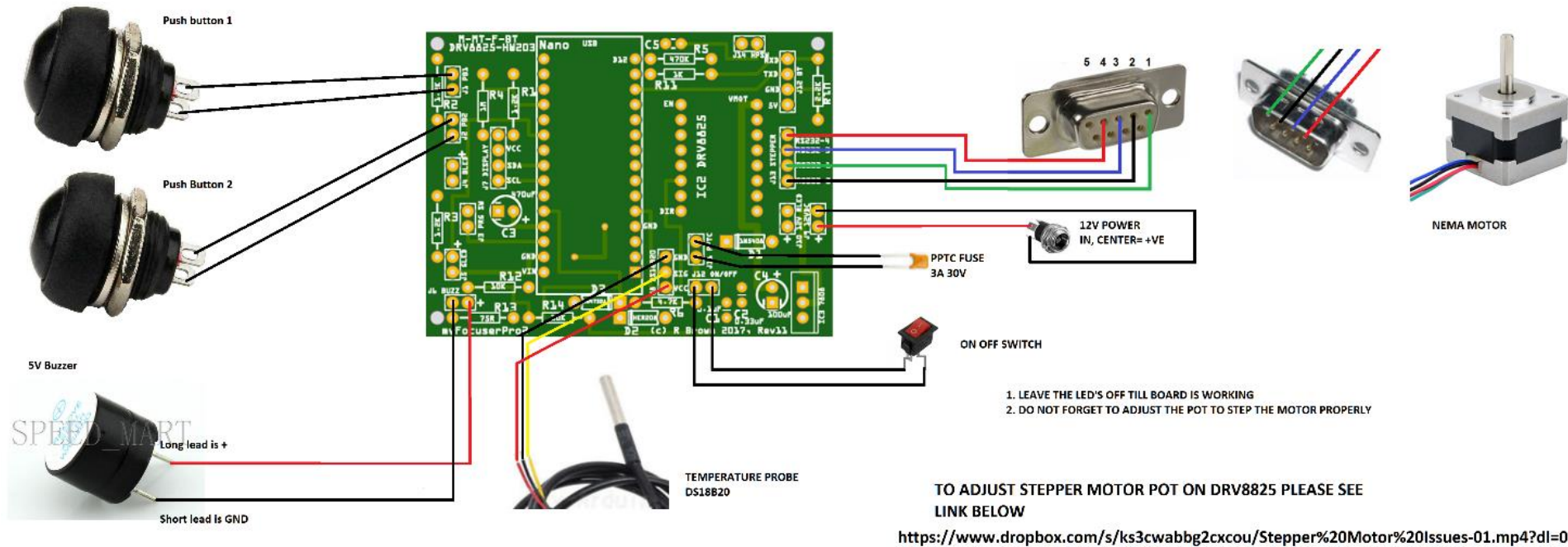
Push button 1

Push Button 2

5V Buzzer

Long lead is +

Short lead Is GND

5 4 3 2 1

12V POWER
IN, CENTER= +VE

NEMA MOTOR

PPTC FUSE
3A 30V

ON OFF SWITCH

TEMPERATURE PROBE
DS18B20

1. LEAVE THE LED'S OFF TILL BOARD IS WORKING
2. DO NOT FORGET TO ADJUST THE POT TO STEP THE MOTOR PROPERLY

TO ADJUST STEPPER MOTOR POT ON DRV8825 PLEASE SEE
LINK BELOW

https://www.dropbox.com/s/ks3cwabbg2cxcou/Stepper%20Motor%20Issues-01.mp4?dl=0

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.

DIY STRIPBOARD

If you do not wish to purchase a PCB you can build a stripboard version. There are a number of other build options for Stripboards. The ASCOM driver and Windows software supports ALL options. The correct Arduino code version must be used with the correct driver board build option.

Stripboards are based on the original prototypes. This means that they have not been updated with new features that have been added to the PCB versions.

However, stripboards are
- relatively easy to make
- require skills in soldering
- take longer to assemble than PCB versions
- easier to make mistakes

The latest recommended DIY Stripboard is [DRV8825-HW203-F](#) which has a common layout that supports most options.

In other words, you can use the same strip-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.

Some options may not be available as stripboards.

CURRENT LIMITS FOR BUILD OPTIONS

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

Driver Board	Suggested Maximum Current	Suggested Stepper Motor
ULN2003	300mA	28BYJ-48
L293D Motor Shield/Uno	500mA	Nema17PG5/PG27 or 28BYJ-48
DRV8825	**1.5A**	**Nema17PG5/PG27** **or** **NEMA1414HM11-0404S**
EasyDriver v44	600mA	Nema17PG5/PG27
L298N	2A	Nema17PG27
A4988	1.5A	Nema17PG5/PG27
ST6128	2A	Nema17PG5/PG27
RAPS128	2A	Nema17PG5/PG27
L293D Mini Driver Board	600mA	Nema17PG5/PG27
TB6612FNG Driver Board	1.2A	Nema17PG5/PG27

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

If your stepper motor is rated at 1A or above, it cannot be supported due to the high current and you will need to buy a suitable stepper motor – the recommend stepper motor is the NEMA17-PG27

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

NOTE
The L9110S driver board is no longer recommended.

myFP2 COMPARISON OF DRIVER BOARDS

Driver Board	Typical I	Peak I	Typical V	Max V	Steps	Stepper
A4988	1.5A	2A	12V		$F \; \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
DRV8825[1]	**1.5A**	**2.2A**	**12V**	**45**	$F \; \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$	**Nema17PG5/PG27 or NEMA1414HM11-0404S**
EasyDriver	500mA	750mA	12V	30	$F \; \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
L293D Mini Driver Board	< 600mA	1.2A	12V	25V	$F \; \frac{1}{2}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
L298N	2A	3A	12V	35	$F \; \frac{1}{2}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
L239D Shield[2]	450mA	600mA	12V	12[1]	$F \; \frac{1}{2}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
RAPS128	< 2A	2.2A	12V		$F \; \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
ST6128	< 2A	2.2A	12V		$F \; \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
TB6612FNG	< 1A	2A	12V	15V	$F \; \frac{1}{2}$	Nema17PG5/PG27 or NEMA1414HM11-0404S
ULN2003	350mA	500mA	12V	12[1]	$F \; \frac{1}{2}$	28BJY-48

[1] This is the recommended stepper motor to use

Note: The ULN2003, L293D, L298N, TB6612FNG 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)

EXAMPLE

I purchased a NEMA stepper motor from eBay which has a rated coil current of 1.6A. Which option should I chose?

Sorry – None. The maximum current will be 3.2A (there are two coils in a NEMA stepper motor) and no driver board above will supply that much current.

It is best to use the recommended NEMA stepper motors for this project which all have a current rating of 400mA.

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

Driver Board	Arduino Code
A4988	Focuservxxx_A4988_HW203
DRV8825_HW203[2]	**Focuservxxx_DRV8825_HW203**
EasyDriver	
L293D Mini Driver Board	Focuservxxx_L293DMINI
L293D Mini Driver Board - NOKIA	Focuservxxx_L293DMINI
L239D Shield[1]	Focuservxxx_L293D
L298N	Focuservxxx_L298N
RAPS128	Focuservxxx_RAPS128_HW203
ST6128	Focuservxxx_ST6128
TB6612FNG	Focuservxxx_TB6612FNG
ULN2003	Focuservxxx_ULN2003
ULN2003-NOKIA	Focuservxxx_ULN2003

[1] Can only be used with Arduino Uno or Mega

[2] This is the recommended stepper motor to use

STRIPBOARD DRIVER BOARD FILES

The following table has links to each of the driver board build options.

A4988
DRV8825-HW203
DRV8825-TFT22
PCB DRV8825-HW203-OLED-ROTARY-ENCODER
PCB DRV8825-HW203 REV4
EASYDRIVER-HW203
L293D Mini
L293D Mini-NOKIA [not available as stripboard]
L298N
RAPS128
ST6128
TB6612FNG
ULN2003
ULN2003-NOKIA [not available as stripboard]
UNO + L293D MOTOR SHIELD
LET'S MAKE A DRV8825 HW203 FULL myFP2
LET'S MAKE A L298N MINIMAL myFP2
LET'S MAKE A ULN2003 MINIMAL myFP2

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.

STEPPER MOTORS

MICRO-STEPPING

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).

The L293D shield supports half stepping on both the 28BYJ-48 and the NEMA17-PG5 stepper motors.

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

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

A NOTE ABOUT MICRO-STEPPING

Micro-stepping requires coil power in both stepper coils. This means that in order to hold the stepper motor in position, coil power must be maintained. The myFP2 firmware and applications has a feature called Coil Power, which when ON maintains coil power to the stepper motor to maintain stepper position.

When micro-stepping, if coil power is not maintained, then the position of the stepper motor becomes invalidated over time. Consider a focuser move with micro-stepping. Coil power must be ON to move the stepper. The stepper motor completes the move, and then the user disabled coil power off. At this point, the stepper motor will fall to the FULL step. Where that is depends on a lot of things (strength of magnets, position of rotor, spacing between rotor and magnets etc) and may NOT be always the same. Sometimes the motor shaft might fall backwards, other times forward. The net effect though is that the focuser has indeed lost some "steps" for every move, even though the focuser software has not adjusted for this loss.

Another issue with micro-stepping is the loss of torque. The full power of the stepper is no longer available, and this can make the difference between moving the focuser tube and not moving the focuser tube.

Using a GEARED stepper motor significant increases torque AND reduces the requirement to micro-step the motor. Hence, using a geared stepper motor at full steps is the recommended solution.

It is recommended to use coil power ON if using micro-stepping.

This also explains why using a modified "position" of converting microsteps to full steps like "Position:microsteps" is a really bad idea and is no more accurate than any other method. Other systems which automatically adjust the position based on changes in microstepping also suffer from this issue.

A NOTE ABOUT TORQUE

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

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

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

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

Whatever you do, DO NOT purchase a stepper motor that requires more than 0.6A [600mA] coil current to operate. Doing so can damage the driver boards or severely shorten the working life of the driver board.

STEPPER MOTOR WIRING (PCB)

DRIVER Jx-STEPPER HEADER 4P	RS232 Pin	NEMA17PG27/NEMA17PG5	NEMA17	14HM11-0404S
DRV8825/A4998/DRV8825-HW203				
Jx-Pin1	4	RED (Coil2)	GREEN	RED
Jx-Pin2	3	BLUE (Coil2)	BLUE	BLUE
Jx-Pin3	1	GREEN (Coil1)	BLACK	BLACK
Jx-Pin4	2	BLACK (Coil1)	RED	GREEN
L9110S				
B+	4	RED (Coil2)	BLUE	BLUE
B-	3	BLUE (Coil2)	GREEN	RED
A-	1	BLACK (Coil1)	RED	GREEN
A+	2	GREEN (Coil1)	BLACK	BLACK
L293D-MINI				
A+	4	RED (Coil2)	BLACK	BLACK
A-	3	BLUE (Coil2)	RED	GREEN
B+	1	BLACK (Coil1)	GREEN	RED
B-	2	GREEN (Coil1)	BLUE	BLUE
L298N and TB6612FNG				
B1	4	RED (Coil2)	BLACK	BLACK
B2	3	BLUE (Coil2)	RED	GREEN
A2	1	BLACK (Coil1)	GREEN	RED
A1	2	GREEN (Coil1)	BLUE	BLUE

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.

myFP2 PURCHASE LIST

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

PRECAUTIONS

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

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

NANO CH340G

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

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

## FTDI Nano Chips – Not recommended

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

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

The CH340G Nano ATMEGA328 board as recommended for this project.

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

Mounting the Nano controller on a PCB or strip-board
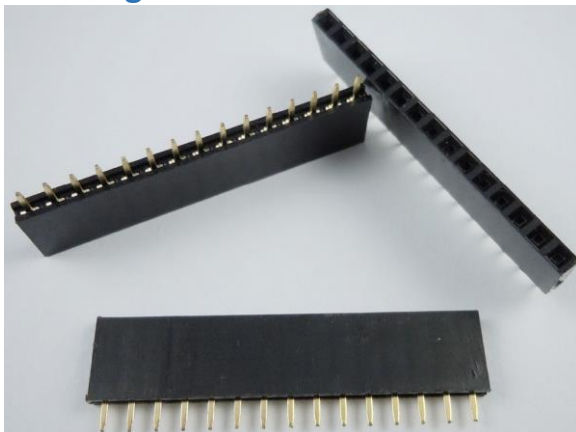
It is highly recommended to mount the Nano using Arduino 15 pin headers. This means the board can be tested to ensure that the correct voltage is present on the VIN pin.

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

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

PRECAUTIONS

Never disconnect or connect the stepper motor when the myFP2 Controller or external power is ON. This will result in damage to the driver board

Never disconnect or connect the temperature probe when the myFP2 Controller or external power is ON. This may result in damage to the Nano board

POWER AND PROTECTION FOR VIN

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

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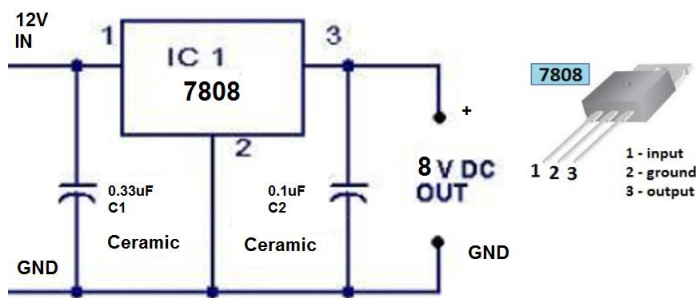
This means the controller can be powered in the field from 12V or a car battery. If using a USB connection, the external power supply is still required to power the stepper motor.

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

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

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

REVERSE VOLTAGE PROTECTION
WHY USE A 10A10/1N5408 DIODE?

A diode 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 diode is around 1.1V. A 3A Resettable Polymer fuse provides excess current protection. The voltage drop of the diode does not cause any issue with the circuits and is not an issue.

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

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

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

CONNECTION RESET PREVENTION CIRCUIT

The following circuit provides a means of **disabling** the Arduino reboot that occurs when connecting to the controller via USB.
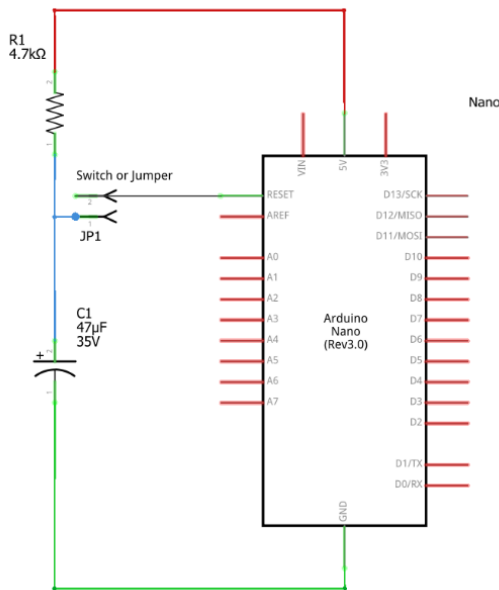
myFP2 comes with its own Windows Applications and ASCOM drivers.

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.

Connection reset prevention circuit explanation

Purpose: Prevent Arduino resetting when a connection is made on the serial port to the controller

When an application or driver in Windows or ASCOM connects to an Arduino, a reset is applied which forces the Arduino to reset. During the reset, the Arduino cannot respond to requests on the Serial port till the reset is finished.

For an Arduino controller such as a focuser controller this can take 2-3 seconds. If an application connects to the controller a reset happens, the controller cannot respond until 2-3 seconds has elapsed.

if the application connects then immediately sends a request without waiting, the application can time out with a serial response error - the moonlit driver for example will only wait 1s.

To prevent the Arduino resetting when a connection is made, which stops the reset cycle and thus making the Arduino respond immediately, we need to stop the reset.

This is done via a capacitor between the RST line and either VCC or GND. There are many ways to do this. Including using resistors.

We need to prevent the RST line from going low, so it has to be held high.

One way is to put a 4.7K resistor from RST to 5V, and then use a 47uF electrolytic (bipolar) capacitor between RST and GND (+ve lead to RST, -ve lead to GND). This keeps the RST line high and prevents the reset from occurring when a connection is made.

The downside is that you will not be able to reprogram the Arduino chip unless the reset prevention circuit is removed. There needs to be some way to remove this circuit (either using jumpers or a switch so that the Arduino chip can be reprogrammed with new firmware updates.

myFP2 MODES OF OPERATION

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

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

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.

Another way to change direction is to swap the stepper coils over (swap coil A for coil B, in other words for a NEMA motor, swap pins 1 and 2 of the RS232 connector to pins 3 and 4, and swap pins 3 and 4 to pins 1 and 2).

LCD1602/I2C DISPLAY [Optional]

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

This would be useful in manual control where the push buttons are used to control the focuser.
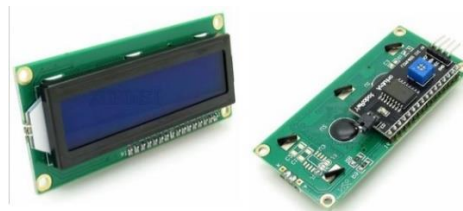
LCD1602 IIC I2C TWI 1602 Serial Port LCD Display Module

Initial Startup Screen	Explanation
MyFocuser_XXXXX	Driver Board version
2.x.x	Program version

| Program running	
C=NNNNN PW=OF +	Current focuser position, Coil Power, Temperature Compensation
T=NNNNN RD=ON F	Target focuser position, Reverse, Stepping Mode

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

LCD Screen Page 1

The first screen displays the focuser current position, the target power, the step mode and the status of coil power, reverse direction and temperature compensation settings.
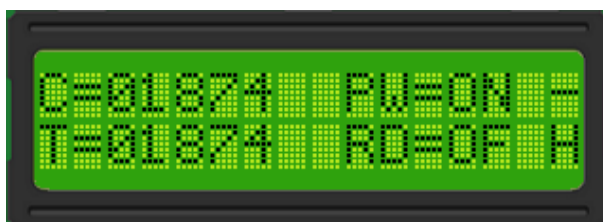
PW indicates if power is kept to the stepper coils once a move is completed. For further information, please refer to the Readme.htm file that is available when the ASCOM driver is installed.

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

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

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

LCD Screen Page 2

The second screen displays the temperature (c= Celsius and f = Fahrenheit) and the current value for maxSteps.
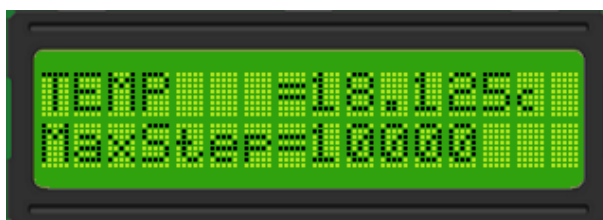
// NOTE 1: YOU CANNOT USE A STRAIGHT 4P-4P HEADER CABLE FOR THE OLED DISPLAY
// NOTE 2: IF USING LCD, USE A STRAIGHT 4P-4P HEADER CABLE

ENABLING THE LCD DISPLAY IN THE FIRMWARE

If an LCD display is fitted to the controller, it needs to be enabled in Section 1 of the file *focuserconfig.h*

The example below shows the LCD display enabled in the firmware

```
// ----------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// ----------------------------------------------------------------------

// To enable the LCD DISPLAY uncomment the next line (LCD1602, I2C)
#define LCDDISPLAY 1

// To enable the OLED DISPLAY uncomment the next line
//#define OLEDDISPLAY 1
```

WHAT ABOUT SUPPORT FOR DIFFERENT LCD TYPES?

There is a section in the firmware to specify the LCD type. Supported types are 1602, 1604 and 2004. Set the correct value in Section 2 of the file *focuserconfig.h* to match your LCD display (if fitted). The example below sets the display type to 2004 (20 characters wide by 4 lines).

```
// ----------------------------------------------------------------------
// SECTION TWO: SPECIFY LCD DISPLAY TYPE
// ----------------------------------------------------------------------
// only uncomment one of the following LCDxxxx lines depending upon your lcd type
// only uncomment one of the following LCDxxxx lines depending upon your lcd type
//#define LCD1602          1          // 16 character, 2 lines
//#define LCD1604          2          // 16 character, 4 lines
#define LCD2004           3           // 20 character, 4 lines
```

LCD CABLE WIRING

The LCD display uses an I2C backpack and the 4 wires of the I2C backpack connect to the specified display connector on the PCB. You must ensure that each wire goes to the correct pin.
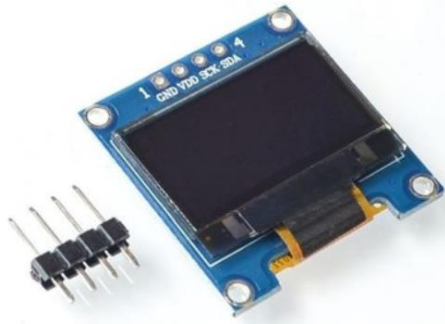
The example below shows the PCB DRV8825HW203-M-MT-F-BT and an LCD I2C display.
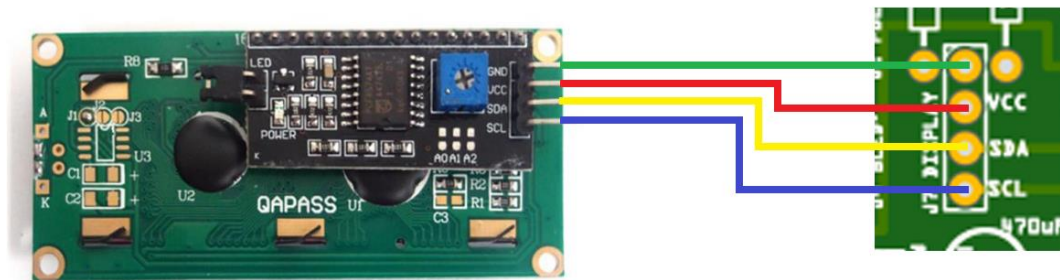
LCD DISPLAY ISSUES

If you have issues with the display not working, look here.

OLED I2C 0.96" DISPLAY [Optional]

An OLED 0.96" display I2C display (128x64, SSD1306) is supported on all controller versions with firmware 228+.

We recommend that you purchase from the link provided.

In addition, you need to replace the 4pin connector with a right-angle connector, so it needs to come un-soldered.

https://www.ebay.com/itm/0-96-I2C-IIC-Serial-128X64-128-64-Blue-OLED-LCD-LED-Display-Module-for-Arduino/273364806892

**If you purchase a different OLED chip the firmware is not guaranteed to run.**

ENABLING THE OLED DISPLAY IN THE FIRMWARE

If an OLED display is fitted to the controller, it needs to be enabled in Section 1 of the file *focuserconfig.h*
The example below shows the OLED display enabled in the firmware

```
// -------------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// -------------------------------------------------------------------------

// To enable the LCD DISPLAY uncomment the next line (LCD1602, I2C)
//#define LCDDISPLAY 1

// To enable the OLED DISPLAY uncomment the next line
#define OLEDDISPLAY 1
```

OLED WIRING

4P-4P XH2.54 Straight through cable 15cm
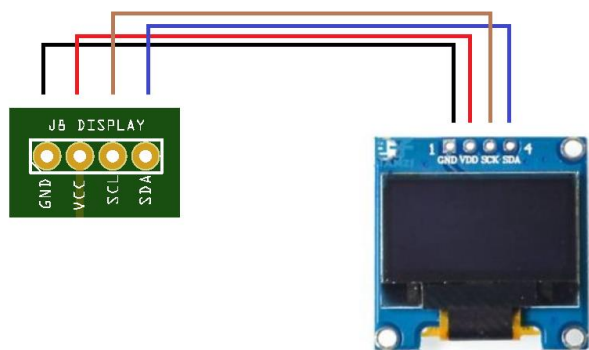
The firmware supports the SSD1306 OLED I2C 128x64 display. Other displays are not supported.

The default I2C address for this display is 0x3C [defined as OLED_ADDR]. Some Adafruit displays use 0x3D. The I2C address of the OLED can be found using the I2CScanner.ino program.

If the OLED display you purchased is at a different address than 0x3C then you will need to modify OLED_ADDR to match the I2C address of your display.

OLED DISPLAY TYPES SSD1306 AND SSH1106

The 0.96" OLED I2C display uses an SSD1306 driver chip.

The 1.3" OLED I2C display uses an SSH1106 driver chip. When using this display with myFocuserPro2 firmware, users often report an issue with the text on the display, where the first character or two characters are missing.

The default OLED supported is the SSD1306. If you intend to use an OLED 1.3" display that uses the SSH1106 driver chip, you need to make a change in the firmware to support this.

This is Section 3 of the file *focuserconfig.h* file where you need to make the change for using an SSH1106 OLED display.

```
// ------------------------------------------------------------------------------
// SECTION THREE SPECIFY OLED DISPLAY TYPE
// ------------------------------------------------------------------------------
// only uncomment one of the following USE_SSxxxx lines depending upon your lcd type
// For OLED 128x64 0.96" display using the SSD1306 driver, uncomment the following line
#define USE_SSD1306    1

// For OLED 128x64 1.3" display using the SSH1106 driver, uncomment the following line
//#define USE_SSH1106    2
```

To use the SSH1106 display, comment out the USE_SSD1306 line and uncomment the USE_SSH1106 line as follows.

```
// ------------------------------------------------------------------------------
// SECTION THREE SPECIFY OLED DISPLAY TYPE
// ------------------------------------------------------------------------------
// only uncomment one of the following USE_SSxxxx lines depending upon your lcd type
// For OLED 128x64 0.96" display using the SSD1306 driver, uncomment the following line
//#define USE_SSD1306    1

// For OLED 128x64 1.3" display using the SSH1106 driver, uncomment the following line
#define USE_SSH1106    2
```

After making the change, reprogram the controller.

DISPLAY PAGES

If you are using a display, the following lists the various pages that are displayed for each of the display types [some display types are not available on some PCB boards].

LCD 1602 [16characterx2lines]
```
Lcd page 1
POS=
TAR=

Lcd Page 2
CoilPwr=0 | 1
RevDir =0 | 1

Lcd Page 3
Temp    =
MaxStep=

Lcd Page 4
Speed    =0 | 1 | 2
StepMode=

Lcd Page 5
TempComp=0 | 1
TempMode=Celsius | Fahren
```

LCD 1604 [16characterx4lines]
```
Lcd page 1
POS=
TAR=
CPwr= RDir=
MSpd= StepM=

Lcd Page 2
Temp     =
MaxStep =
TempComp=0 | 1
TempMode=Celsius | Fahren
```

LCD 2004 [20characterx2lines]
```
Lcd page 1
POS=
TAR=
CPwr= RDir=
Temp=

Lcd page 2
MSpd =0 | 1 | 2
MStep=
TComp=0 | 1
Tmode=Celsius | Fahren
```

OLED [128x64 16characterx8lines 0.96" SSD1306 or 1.3" SSH1106]

Oled page 1
```
Position=
Target  =
Coil.Pwr=0 | 1
Rev.Dir =0 | 1
StepMode=
Temp    =
MotorSpd=0 | 1 | 2
MaxSteps=
```

Oled page 2
```
TComp Steps =
TComp State = 0 | 1
TComp Dir   = 0 | 1
Backlash In = 0 | 1
Backlash Out= 0 | 1
Backlash In#=
Backlash Ou#=
Serial Mode | Bluetooth Mode
```

NOKIA [5110]

Nokia page 1
```
Posn=
Targ=
CPwr= Y | N
RDir= Y | N
Temp=
MSpd= Slow | Med | Fast
```

Nokia page 2
```
StepMd= Full | 1/2 |
StepSi=
Maxpos=
TComp =Yes | No
TCoeff=
```

UTFT [2.2" UTFT SPI Display with ILI9341]

Utft page 1
```
CURRENT POSITION  =
TARGET POSITION   =
COIL POWER        =
STEP MODE         = FULL | 1/2 | 1/4 | 1/8 | 1/16 | 1/32
TEMPERATURE       =
MOTOR SPEED       = SLOW | MEDIUM | FAST
MAX STEPS         =
```

TFT [1.8" SPI TFT LCD 128X160 ILI9163/ST7735]

Tft page 1
```
myFocuserPro2
CurrentPos=
Target Pos=
MaxSteps  =
CoilPwr   = ON | OFF
Reverse   = ON | OFF
Stepmode  = FULL | 1/2 | 1/4 | 1/8 | 1/16 | 1/32
MotorSpd  = Slow | Medium | Fast
```
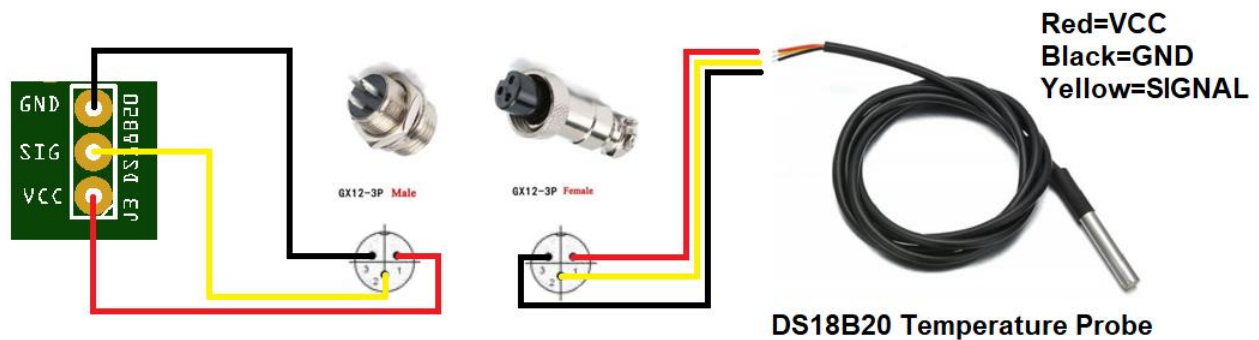
Note:

UTFT   Stripboard HW203-DRV8825

TFT   Stripboard HW203-DRV8825

Nokia   L283DMini, ULN2003

TEMPERATURE PROBE [Optional]

One temperature probe (DS18B20) is supported. The default resolution is set by the controller to 10-bit giving 0.25 degree of resolution (the resolution can be changed). The accuracy of the measured temperature is within 0.5 degrees Celsius. The controller uses a GX12 connector for the temperature probe.

Disconnection or reconnection of the probe must be done when power is OFF. VCC and GND are wired separately.

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

Temperature Probe Placement

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

Temperature Probe Calibration

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

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

Note: The temperature probe is NOT used for reading ambient temperature. The probe is used to read the temperature of the focuser tube. This reading can then be used to adjust the position of the focuser as the tube temperature changes, keeping the image in focus. This is called temperature compensation. The amount by which the focuser tube expands or contracts according to temperature change is called the temperature coefficient, and is measured in motor steps per degree Celsius.

You need to secure the temperature probe where it can actually measure the temperature of the focuser tube. For a refractor this would be about half-way down the tube length.

ENABLING THE TEMPERATURE PROBE IN THE FIRMWARE

If a temperature probe is fitted to the controller, it needs to be enabled in Section 1 of the file *focuserconfig.h*

```
// ---------------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// ---------------------------------------------------------------------------

// To enable the temperature probe, uncomment next line
#define TEMPERATUREPROBE 1
```

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 connected to a single analogue pin (A0).

12mm Waterproof Lockless Momentary Push button Mini Round Switch

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

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

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

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

ENABLING THE PUSH BUTTONS IN THE FIRMWARE

If push buttons are fitted to the controller, they need to be enabled in Section 1 of the file *focuserconfig.h*

The example below shows the push buttons enabled in the firmware

```
// ---------------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// ---------------------------------------------------------------------------
// Caution: Do not enable a feature if the associated hardware circuits are not fitted
// on the board
// Enable or disable the specific hardware below

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

INFRARED REMOTE CONTROLLER [Optional]

The infra-red remote controller is only available on the following PCB

- DRV8825HW203_FIRBT

The Infra-red remote controller is NOT compatible with the following options

- Rotary encoder
- Bluetooth HC-05

The IR remote option is an alternative to the push button switches. The focuser can move +1, -1, +10, -10, +50, and 50 and reset position to 0. If the code is changed more functions could easily be added.
https://www.ebay.com/itm/HX1838-NEC-Code-Infrared-Remote-Control-module-DIY-Kit-NEW/251895571659

RECOMMENDED CONTROLLER
ONLY THE SUGGESTED HX1828NEX REMOTE CONTROLLER IS SUPPORTED

CONNECTIONS
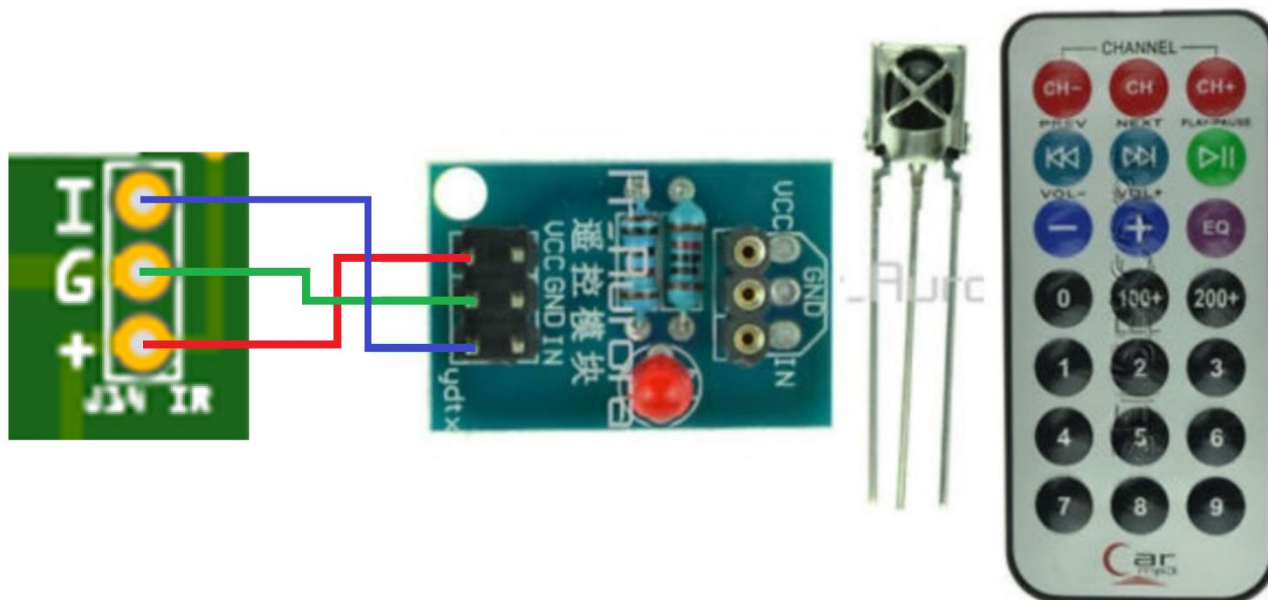
Note: If the IR is not supplied with a little PCB daughter-board, you will need to use a 10KΩ resistor between OUT and VCC.

① OUT
② GND
③ VCC
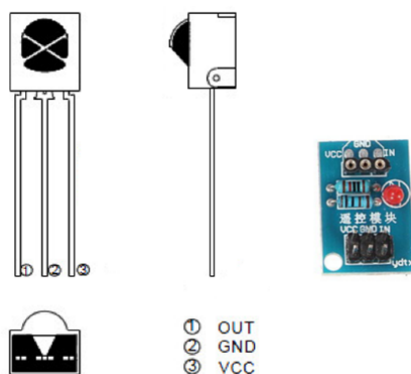
ENABLING THE IR REMOTE IN THE FIRMWARE

To enable the push buttons, you must uncomment the #define INFRAREDREMOTE statement Section 1 of the file *focuserconfig.h*

```
// ----------------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// ----------------------------------------------------------------------------

// To enable infra-red Remote, uncomment the next line (only available on some boards)
#define INFRARED 1
```

IR REMOTE KEY MAPPINGS

CH-	SLOW
CH	MEDIUM
CH+	FAST
1	IN 1 STEP
3	OUT 1 STEP
4	IN 10 STEPS
6	OUT 10 STEPS
7	IN 50 STEPS
8	HALT
9	OUT 50 STEPS
-	IN 100 STEPS
EQ	OUT 100 STEPS
0	RESET POSITION TO 0

It is possible to map other keys for specific tasks. This would require changes to key-mappings defined the firmware file *myIRremotemappings.h*

```
#define IR_SLOW        KEY_CHMINUS
#define IR_MEDIUM      KEY_CH
#define IR_FAST        KEY_CHPLUS

#define IR_IN1         KEY_ONE
#define IR_OUT1        KEY_THREE
#define IR_IN10        KEY_FOUR
#define IR_OUT10       KEY_SIX
#define IR_IN50        KEY_SEVEN
#define IR_OUT50       KEY_NINE
#define IR_IN100       KEY_MINUS
#define IR_OUT100      KEY_EQUALS
#define IR_SETPOSZERO  KEY_ZERO
#define IR_HALT        KEY_EIGHT
```

USING A NON-RECOMMENDED IR REMOTE CONTROLLER

The IR Remote library MAY support other controllers. The test program IR.INO can be used to find out the codes generated by a different IR Remote, and then those codes can be inserted into the *myIRremotemappings.h* file

For example, if you connected your IR remote controller, ran the IRREMOTE.INO test program and pressed the 0 key, and a value of 1760214 was displayed, then you can edit the *myIRremotemappings.h* file as shown below to change the value of the 0 key to your value.

```
// #define KEY_ZERO       16738455         // 0
#define KEY_ZERO       1760214 // 0
```

You would then repeat this for the following keys, updating the values in the *myIRremotemappings.h* file. Once you have defined all the new values you can reprogram the controller. No changes need to be made to the main myFP2ESP.ino file.

Note: *You would need to keep a copy of the modified myIRremotemappings.h file so that you can use it for future firmware upgrades.*

ROTARY ENCODER [Optional]

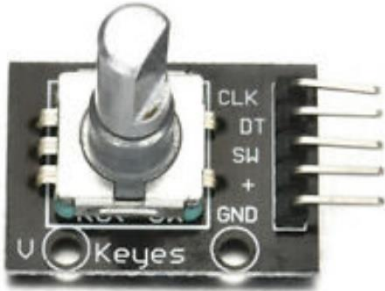The Keyes-040 rotary encode is supported ONLY on some PCB boards.

- DRV8825HW203_FRE
- EasyDriver F-RE

The rotary encoder is NOT compatible with the following options

- Infra-red remote controller
- Bluetooth HC-05

RECOMMENDED ENCODER

ONLY THE RECOMMENDED KEYES-040 ROTARY ENCODER IS SUPPORTED

CONNECTIONS
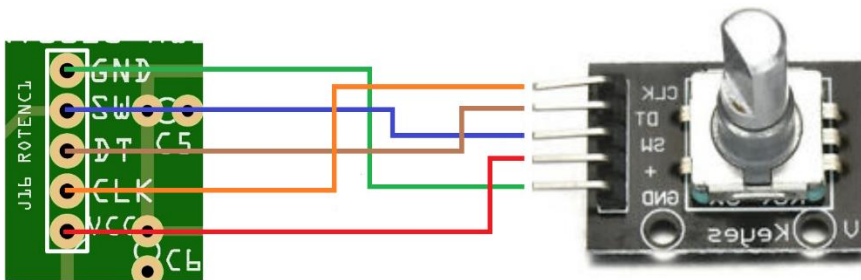
ENABLING THE ROTARY ENCODER IN THE FIRMWARE

To enable the rotary encoder, you must uncomment the #define ROTARYENCODER Section 1 of the file *focuserconfig.h*

```
// -----------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// -----------------------------------------------------------------------
```

// To enable a Rotary encoder. uncomment the next line (only available on some boards)
#define ROTARYENCODER 1

IF using a rotary encoder, PLEASE WAIT for the motor to STOP before turning the encode again.

KEYPAD 4x4 [Optional]

The 4x4 matrix soft membrane keypad is supported ONLY on the following board

- DRV8825HW203_M_MT_F

and requires the use of a special daughter board.

RECOMMENDED KEYPAD

ONLY THE RECOMMENDED 4x4 KEYPAD IS SUPPORTED

CONNECTIONS

The 4x4 keyboard is attached to a daughter board, which connects to the main board via an I2C connector.

ENABLING THE KEYPAD IN THE FIRMWARE

To enable the keypad, you must use the myFP2_DRV8825HW203_KEYPAD_301 firmware file

For a demonstration of the keypad in operation, please see
https://youtu.be/Xws4XfaG-3s

BLUETOOTH HC-05 [Optional]

The HC-05 bluetooth module is supported on most PCB boards.

- A4998_M-MT-F
- DRV8825_M-MT-F-BT
- DRV8825HW203_FIRBT
- DRV8825HW203_M-MT-F-BT
- EasyDriver M-MT-F-BT
- EasyDriver F-RE

Bluetooth is NOT compatible with the following options

- Infra-red remote controller
- Rotary encoder

RECOMMENDED MODULE

ONLY THE SUGGESTED HC-05 BLUETOOTH MODULE IS SUPPORTED

CONNECTIONS
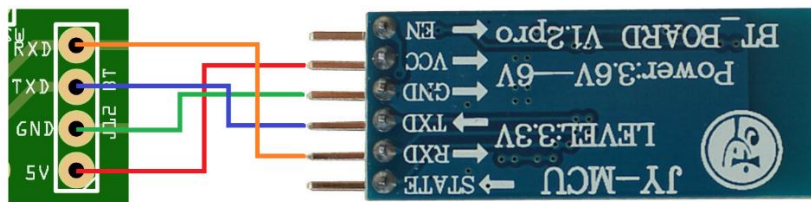
ENABLING THE BLUETOOTH HC-05 ADAPTER IN THE FIRMWARE

To enable the Bluetooth adapter, you must uncomment the #define BLUETOOTH statement in Section 1 of the file focuserconfig.h

```
// ----------------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// ----------------------------------------------------------------------------
```

// To enable bluetooth, uncomment the next line (only available on some boards)
#define BLUETOOTH 1

MANUAL CONTROL OF FOCUSER

There are a number of options for manual control of the focuser
- Push buttons
- Infra-red remote controller
- Rotary encoder
- Keypad

There is a very good reason why push buttons are not available on commercial focusers..............

In normal operation, the focuser will be at near focus. This means you only need to go a few steps either side to get good focus. You need the steps to be small and slow. If you have large steps and fast, then you will overshoot best focus, so then press other button, then that overshoots again and you play ping pong either side of focus and just pure luck to get good focus.

The buttons are NOT designed to travel long distances. What one normally does is probably start from home, tell the focuser to move to near focus position, then use computer or buttons slowly to get good focus. If you want to use buttons for both LONG and FAST and then SMALL and SLOW that is not going to work well. All focusers, not matter what versions or authors will have this restriction.

Another method is fit a home position switch, then in the final part of setup() tell the focuser to move to home, and then, follow with a statement that will move the focuser to near focus for your setup. That way the focuser is at near focus and steps to move using the push buttons will only be small and slow because if you are using FWHM or Bahtinov mask then you do not want quick movement.

If you are manually focusing using buttons, always best to start near focus position.

COMPARING PUSH BUTTONS, IR REMOTE AND JOYSTICK OPTIONS

	Push Buttons	IR-remote	Rotary Encoder	Keypad
Relative Speed	Slowest	Faster at multiple moves	Faster at multiple moves	Faster at multiple moves
Speed adjustable	Need to issue a command to focuser	Just press a key	Need to issue a command to focuser	Press key sequence to move
Moves	One step at a time	Can do multiple steps	Can do multiple steps	Can do multiple steps
User Location	Must be near the controller	Can operate up to 5m away	Must be near the controller	Must be near the controller
Focusing	Poor at focusing when not at near focus	Good for focusing when not at near focus	Poor at focusing when not at near focus	Good for focusing when not at near focus
Cost	Low cost	Low Cost	Low Cost	Low Cost
Batteries	Not needed	Needed for remote	Not needed	Not needed
Other			Must wait for motor to stop before turning again	Requires display Requires extra PCB

myFP2 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. [myFP2 Windows](#) application that can remotely control the focuser
3. [myFP2 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](#))
4. [myFP2Linux](#) application that runs on Linux platforms
5. [myFP2Android](#) app (for Bluetooth)

myFP2 RECOMMENDED CLIENT APPLICATIONS

As well as free Windows application and ASCOM drivers (which is fully ASCOM compliant), myFocuserPro2 controllers have been tested and work with the following software applications

[FocusMax v4.x](#)   [Nebulosity 4.x](#)   [NINA](#)   [APT v3.x](#)

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

MAXINCREMENT

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

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

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

The myFP2 controller will set the value of MaxIncrement to maxStep on connection.

The setting for MaxIncrement is NOT remembered by the controller.

BOUNDARY RULES FOR maxStep AND maxIncrement

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

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

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

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

CONNECTING THE STEPPER MOTOR TO THE FOCUSER

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

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

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

A direct connection to the single focuser shaft is preferred as this will have the least backlash and generate the lowest axial loads.

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

Once connected with a flexible coupler, you cannot move the focuser manually. You must remove the Focus Lock screw or keep the screw loose.
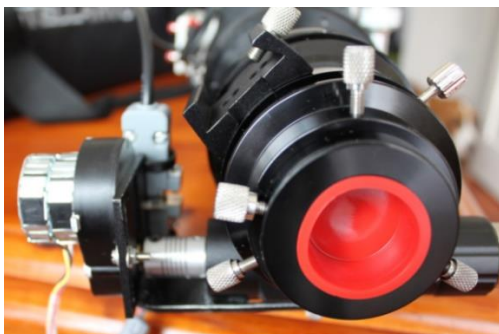
The photo shows an L-bracket (bolted to bottom of focuser) that allows the direct connection of the stepper motor to the focuser shaft using a flexible coupler.

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

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

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

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

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

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

However, using a belt drive significantly increases the axial loads placed on the focuser and over time will lead to significant wear and tear on the focuser mechanism.

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

HOW DO I CONNECT THE STEPPER MOTOR TO THE FOCUSER?

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

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

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

If you are using a geared NEMA stepper motor directly connected to the main focuser shaft, one way to offset the extra weight is to rotate the focuser around so that the focuser shaft runs vertically instead of horizontally with the stepper motor above the tube. This will make balancing easier.

FOCUSER BRACKETS 3D PRINTED

Attaching Stepper Motor direct to focuser shaft using a flexible coupler

The stepper motor fits to one side of the focuser shaft and the motor uses a flexible couple to attach direct to the focuser shaft. Slots in the bracket allow the stepper motor to positioned at the same height as the focuser shaft as well as the correct distance apart to allow for fitting of the flexible coupler.

Attaching Stepper Motor to focuser shaft using a pulley and belt

The stepper motor fits underneath the focuser. The focuser shaft has a large pulley and is connected to the stepper motor via a belt. Slots in the bracket allow the stepper motor to move up and down providing tension adjustment.
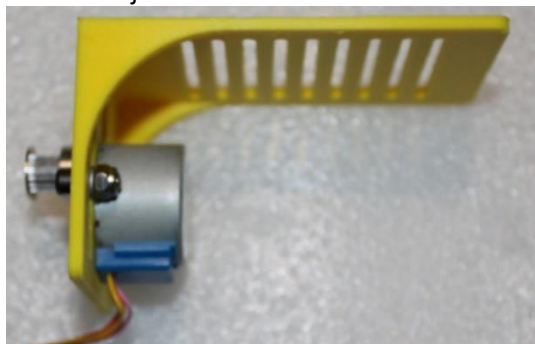
28BYJ-48 BRACKETS
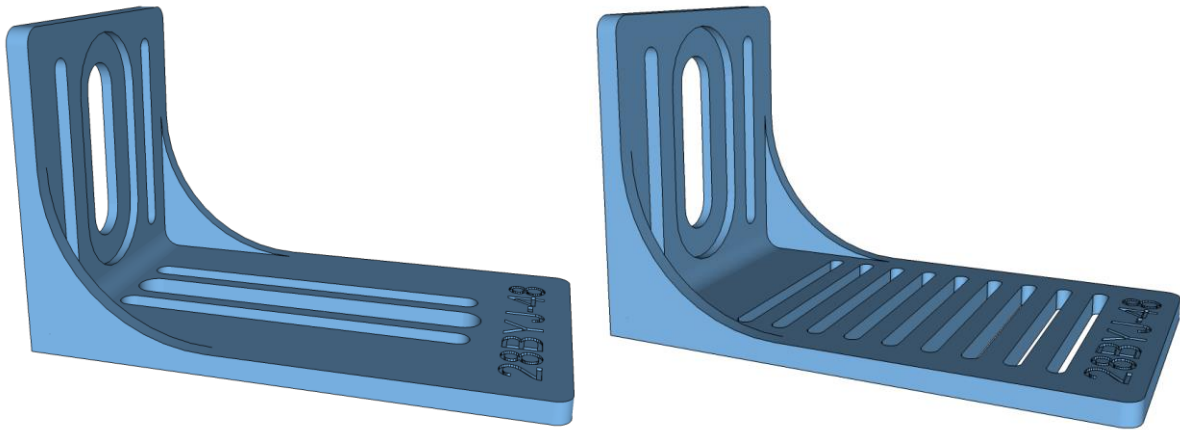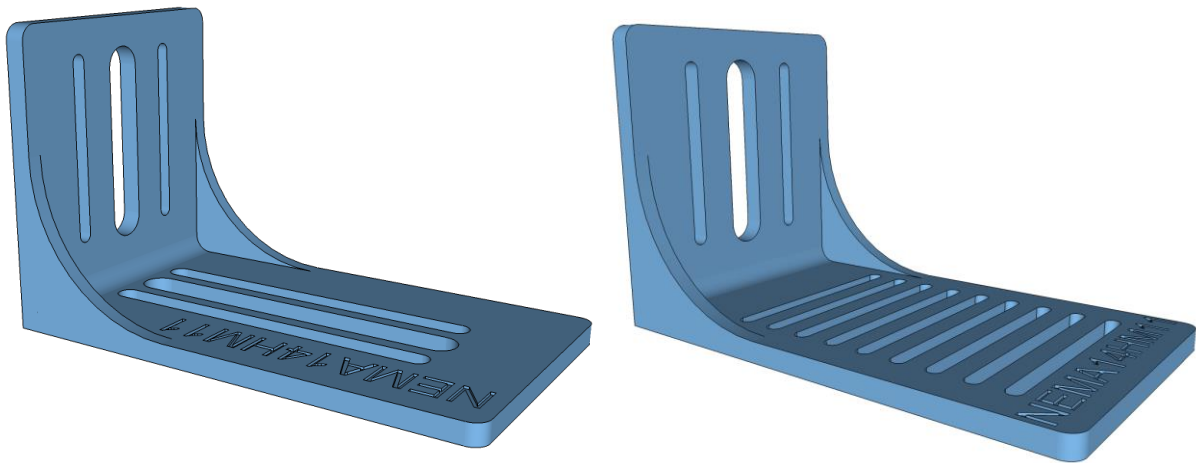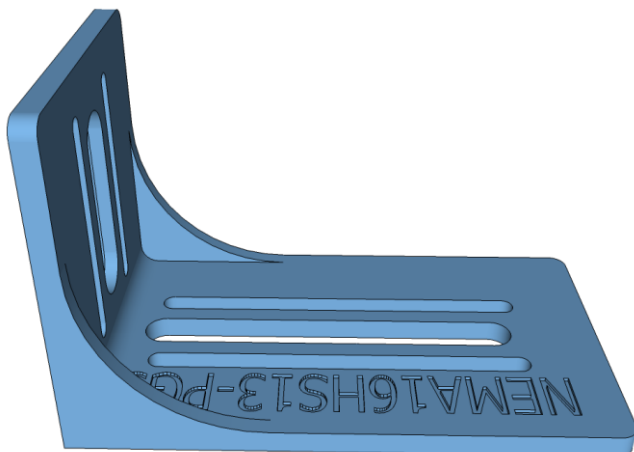
https://www.thingiverse.com/thing:3731843

NEMA14 14HM11-0404S

https://www.thingiverse.com/thing:3731855

16HS13-0604S-PG5

https://www.thingiverse.com/thing:3731917

NEMA17-17HS15-0404S

https://www.thingiverse.com/thing:3731927

NEMA17-PG5-0404S

https://www.thingiverse.com/thing:3731932

NEMA17-PG27-0404S

https://www.thingiverse.com/thing:3731940

https://www.thingiverse.com/thing:3731836
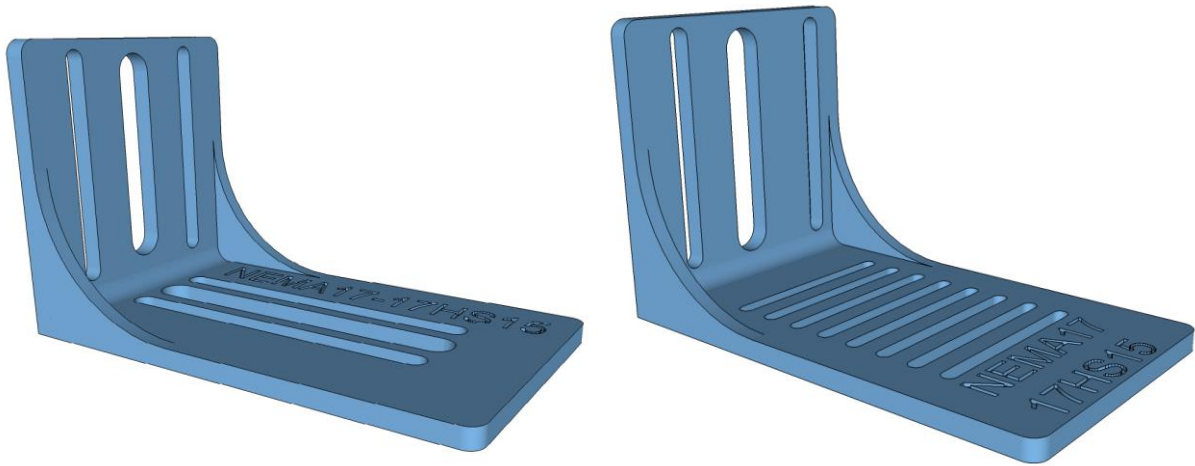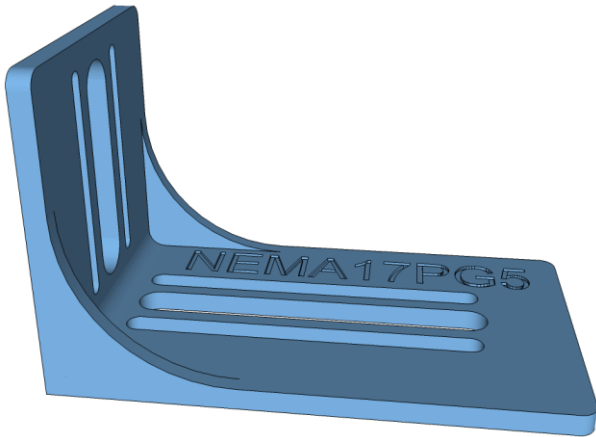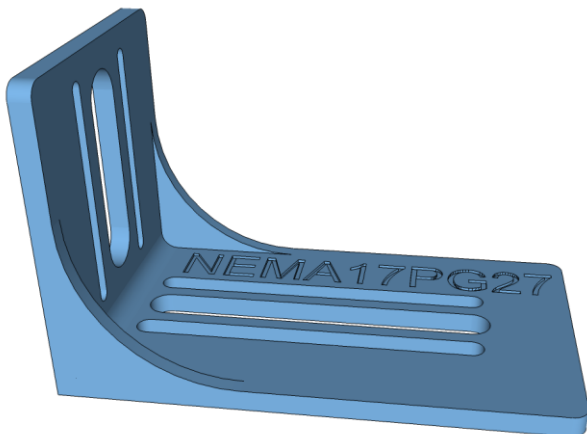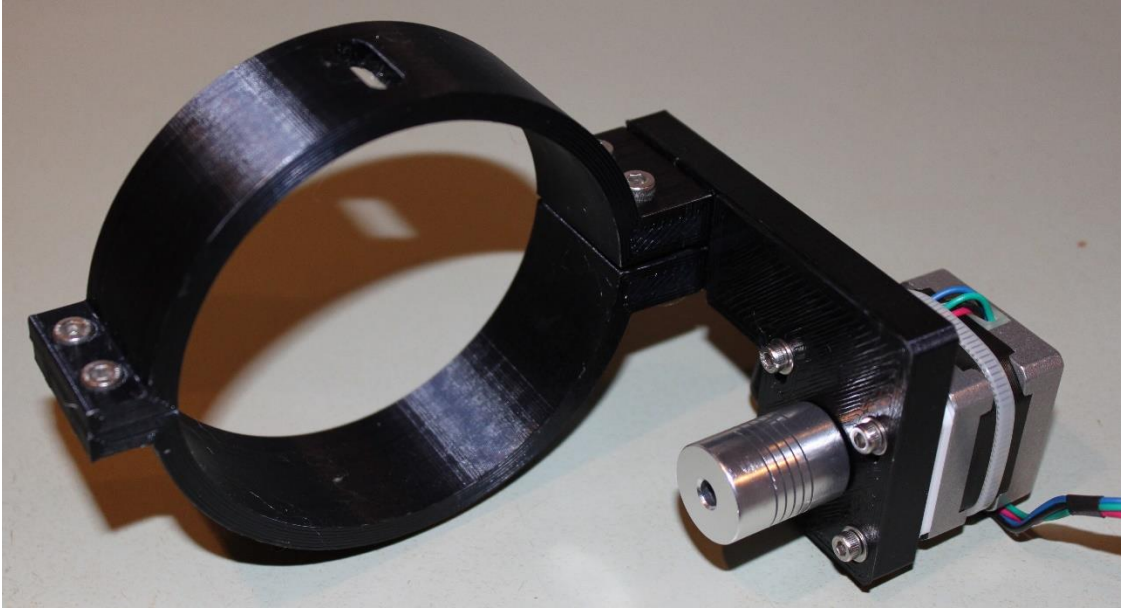
USING GEARS. WHAT OPTIONS DO I HAVE?

Gears are normally used to provide a reduction, so the small gear is fitted to the stepper motor and the larger gear (more teeth) is fitted to the focuser.

This provides a reduction in step size as well as an increase in torque.

© Clive Garner 2017

Gears are normally using in ratios like 1:2, 1:3 or 1:4
This means

1:1	1:2	1:3
15T	30T	45T
16T	32T	48T
18T	36T	54T
20T	40T	60T
30T	60T	
40T	80T	

3D PRINTING GEARS

Infill 100%
Nozzle 0.4mm
Layer Height 0.2mm
Speed SLOW

GEARS FOR NEMA MOTOR

NEMA motors have shaft sizes like 5mm and 6mm in diameter. In 3D printing, the bore size in the gear is made slightly large to accommodate shrinkage in the printing process. Gears are provided with a M3 hex grub screw [which interlocks with an M3 nut inserted into the base of the gear] to secure the gear to the stepper motor shaft.

All measurements in mm, designed to fit standard Reprap 3D printer GT2 belt (2mm pitch)

height of tooth part = 8mm
height of hub part = 6mm

measurements below are approx

	Pulley D	Hub D
15T	9	20
16T	10	20
18T	11	20
20T	12	20
30T	18	20
32T	19	20
36T	22	20
40T	24	25
45T	26	25
48T	28	25
54T	34	25
60T	38	30
72T	45	30
80T	50	30

30T-80T has 2xM3 hex screws

3D PRINTED GEARS NEMA 5mm and 6mm shaft
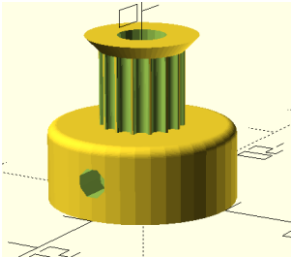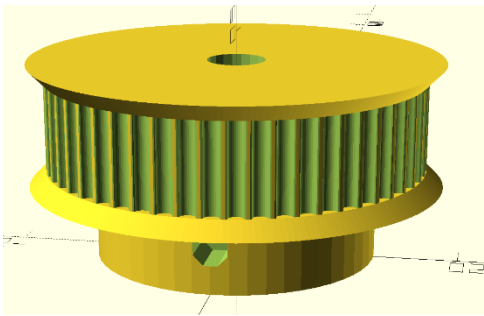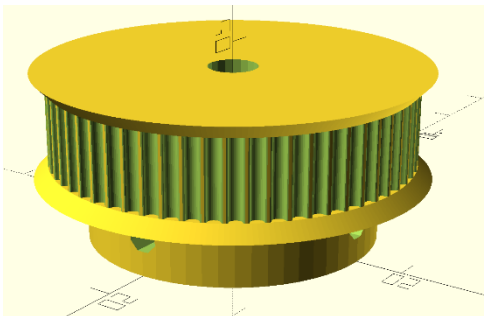
15T
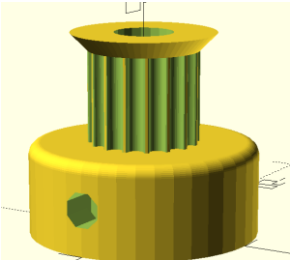
16T
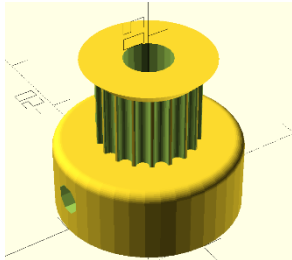
18T
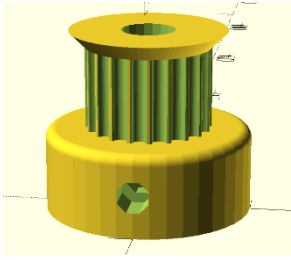
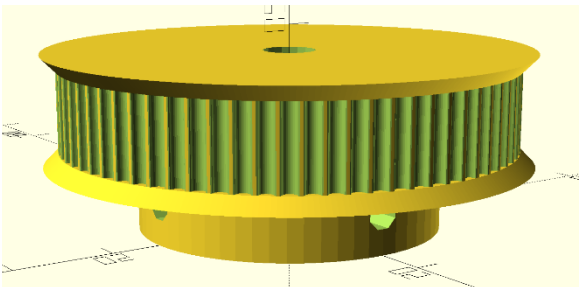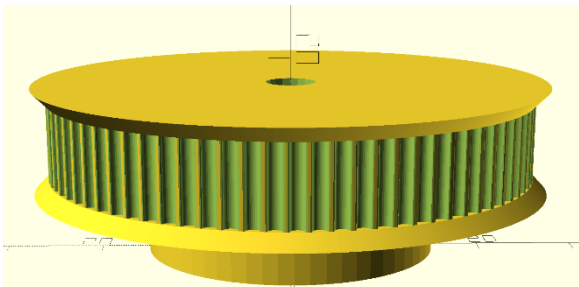20T
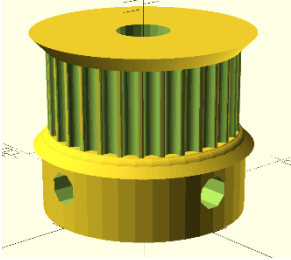
30T
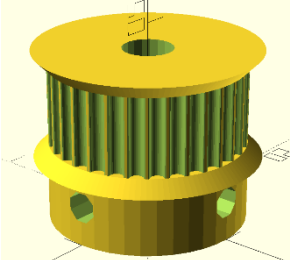
32T
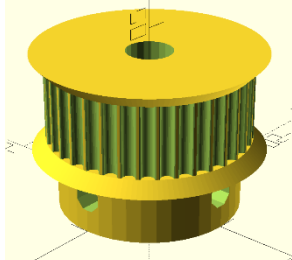
36T
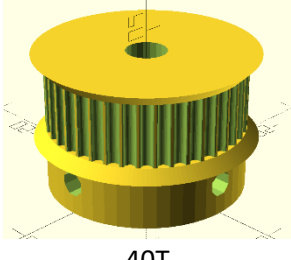
40T
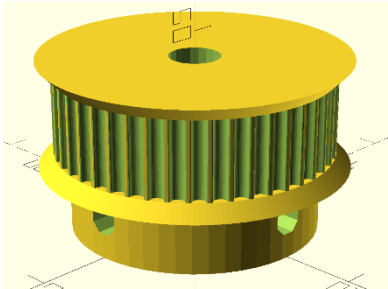
45T
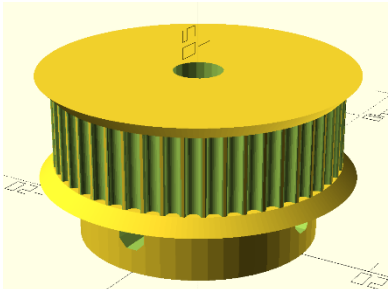
48T

54T

60T

72T

80T

# 3D PRINTED GEARS FOR CELESTRON C8/9.25 SCT

For the C8 there are a number of different knob sizes. Newer Celestron telescopes have a rubber knob which can be pulled off. Older 8" telescopes will have a metal focus knob which must be removed using a 1/16" hex-head wrench to loosen a small set screw. Remove the knob to expose the brass focus shaft.
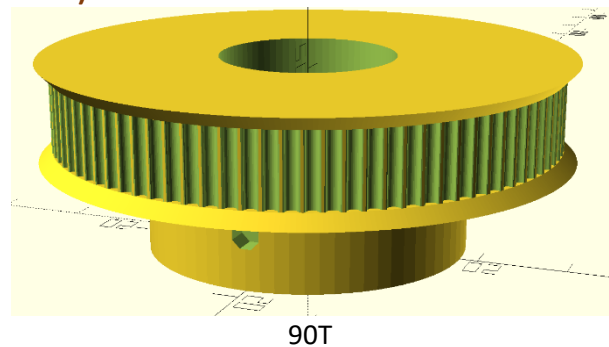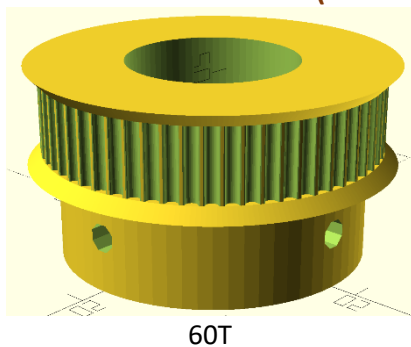
The length of the knob part is about 21mm

The brass shaft size is 13mm.

The knob size is either 0.750" or .7750"

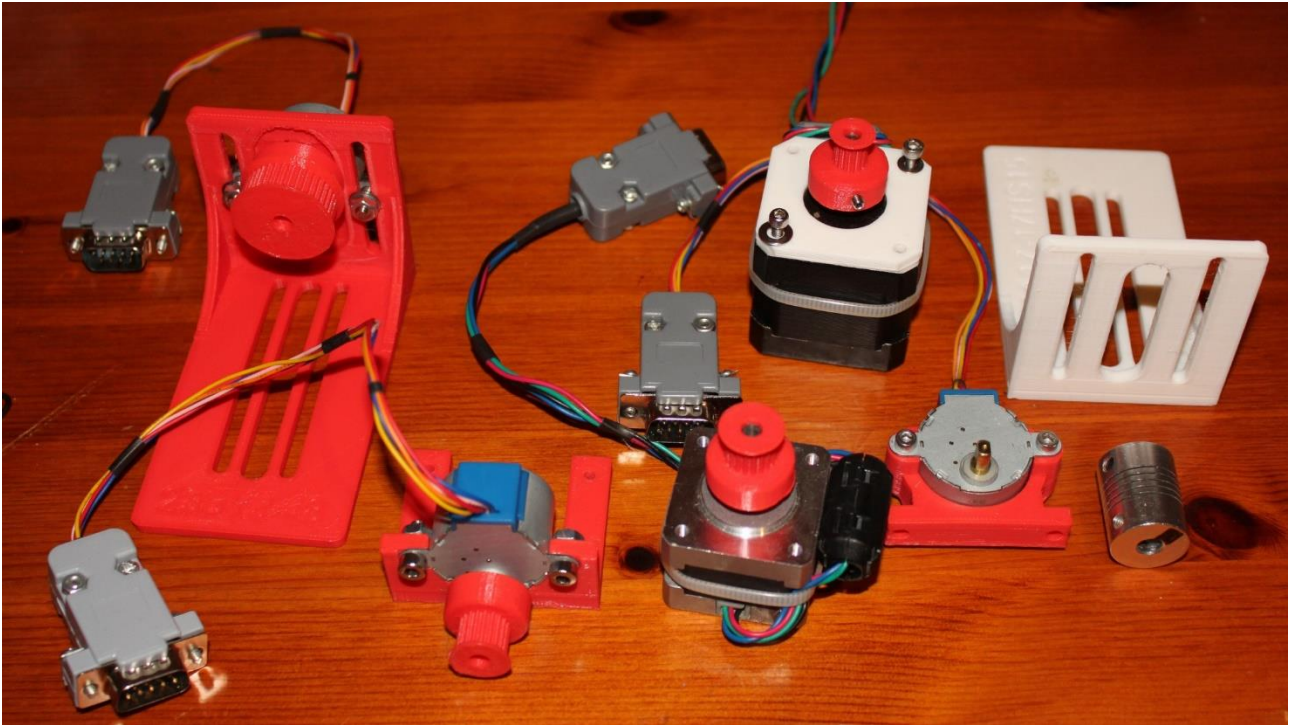## Gears to fit over the brass focus shaft (OD13)

60T

90T
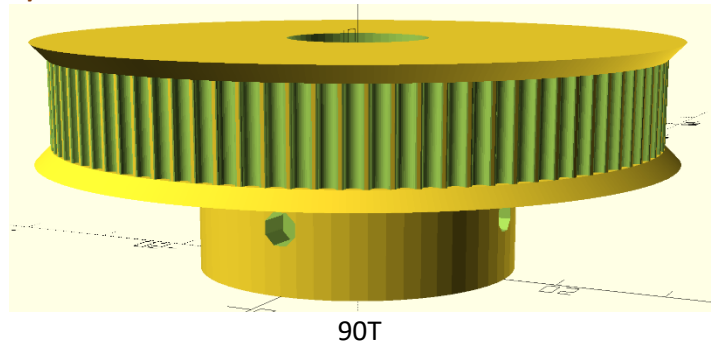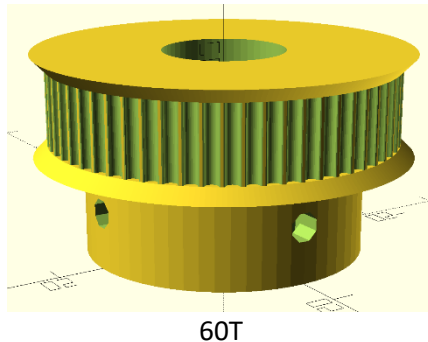
Gears to fit over the focus knob (OD0750 and OD07750)

60T

90T

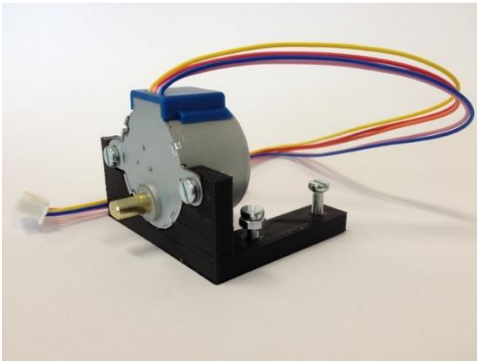To download a zip file containing STL files for all NEMA, 28BYJ-48 and Celestron™ SCT gears,

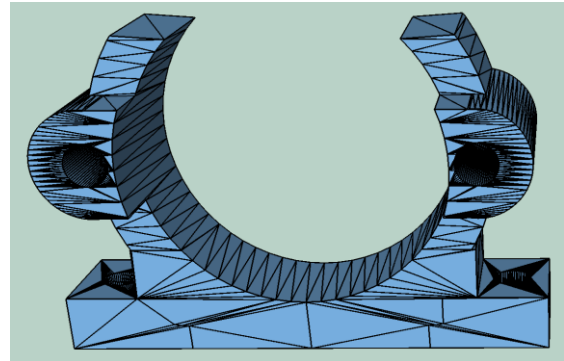https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/3D%20Printed%20Parts/Gears/

3D PRINTED GEARS AND BRACKETS

3D PRINTED BRACKET 28BYJ-48

https://www.thingiverse.com/thing:2514652
ewoud_design

http://www.thingiverse.com/thing:1542952
Druidosh

CALCULATING THE BELT LENGTH REQUIRED FOR GEARS

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

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

HOW DO I CONNECT MY STEPPER MOTOR ON AN SCT?

There are many images on the Internet showing users innovative solutions to attaching a stepper motor to an SCT. Download this file

https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/MOUNTING%20BRACKETS%20AND%20IDEAS/User%20Builds%20and%20Bracket%20Examples.zip/download

and unzip it. In the folder SCT-MAK you will find a number of examples.

Here is an example of the commercially available Pegasus Astro controller bracket
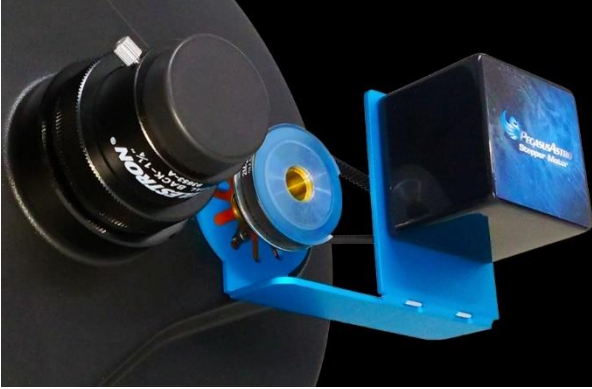
myFP2 PROTOTYPE BUILD PICTURES

Initial prototype and breadboard to test LCD, push button switches, LEDS, L293D shield and stepper motor
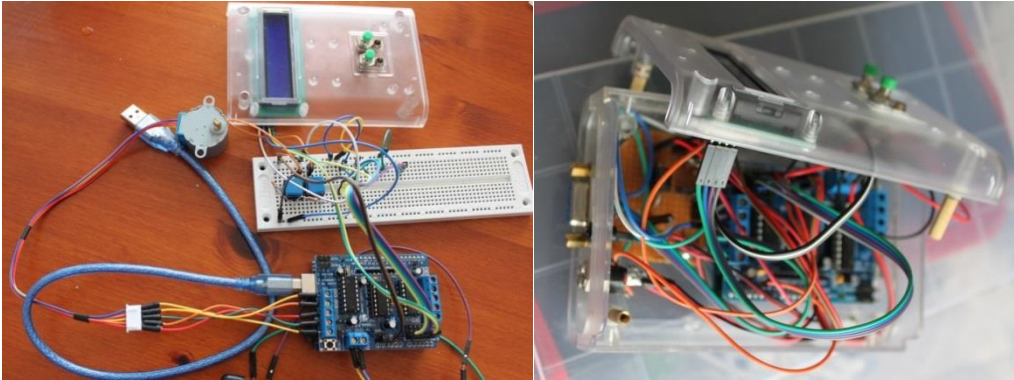
Arduino, L293D shield, LCD, switches enclosed in Link-Sprite case

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

ED80T-CF Mounting bracket

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

DO NOT MANUALLY MOVE THE FOCUSER ONCE SETUP

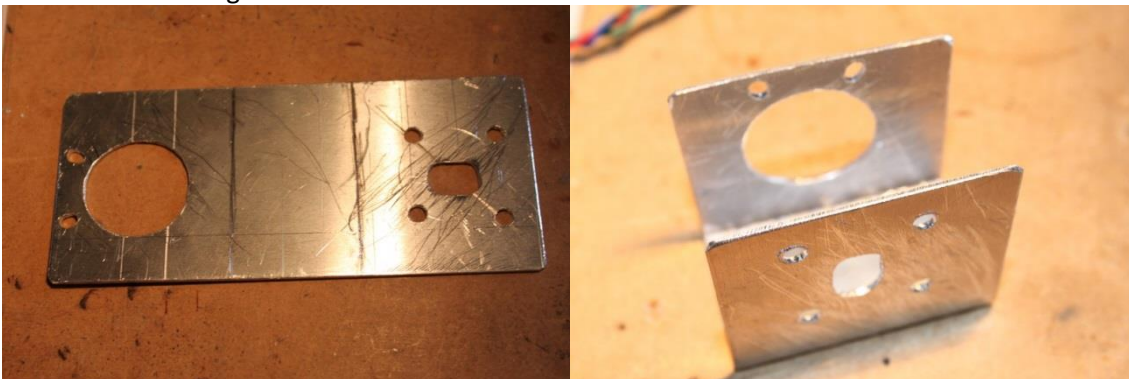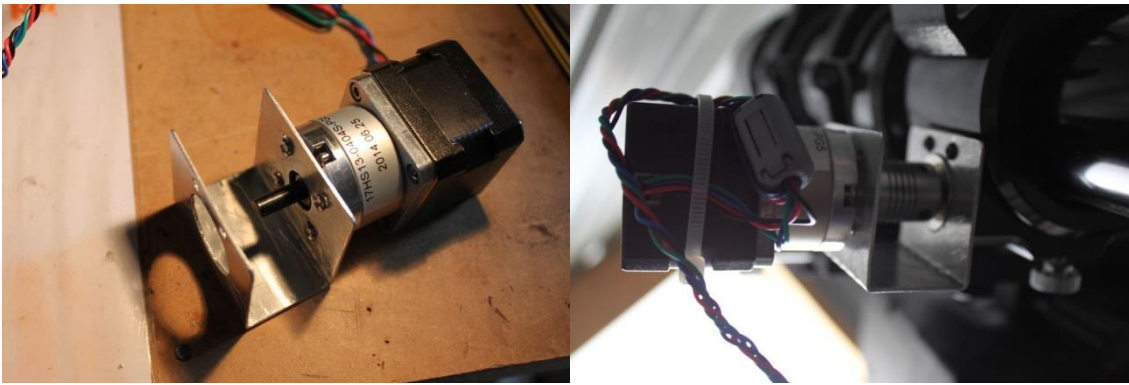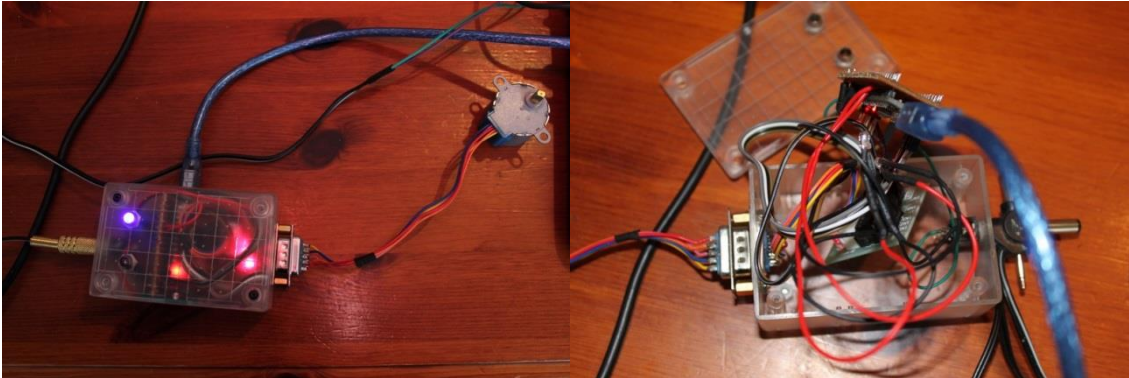Manually moving the focuser position between sessions invalidates the saved focus position in the controller. For example, a user uses the Windows application to set the focus position as 4605. That position is saved by the controller as the last known position when the application is closed.

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

DO NOT CHANGE THE 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.
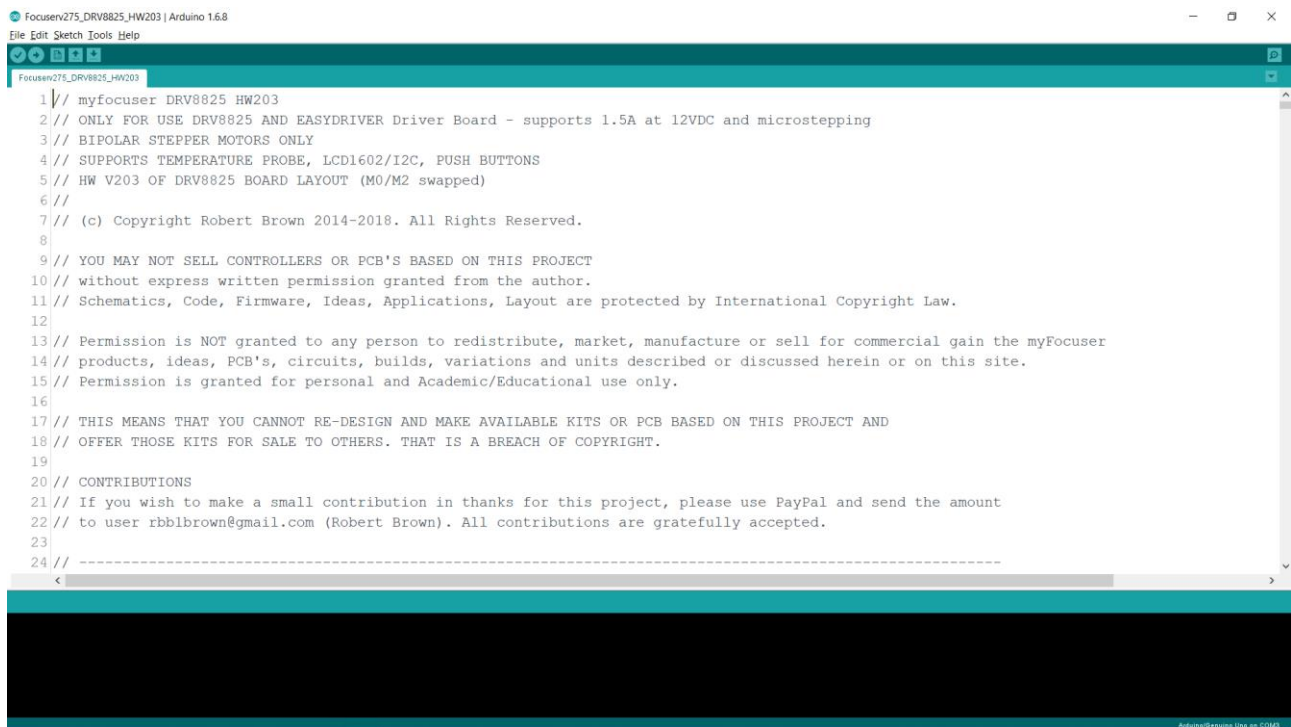
Consider the case where the focuser has been set up as 0 to 6000 maxStep and the focuser is currently at position 4000. The step mode is Half steps. Using some maths, this means there are 2000 half-steps available before the maximum position is reached (or in terms of full-steps, 1000).

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.

myFP2 FIRMWARE

Create a project folder on your computer that will hold all the files required for the controller. Download the firmware zip file from the Sourceforge site.

https://sourceforge.net/projects/arduinoascomfocuserpro2diy/files/CODE%20ARDUINO%20FIRMWARE/

HOW TO PROGRAM THE FIRMWARE

Programming is the same as for any other myFocuserPro controller. The controller is connected via a USB cable to the computer. The Arduino IDE is started and the firmware file and com port selected. The target is set to Arduino Nano [old bootloader] and then the upload button in the IDE is selected.

The Arduino IDE will automatically place the Nano chip into flash mode and once the upload is completed, the Nano is reset and the controller starts running the firmware.

INSTALL THE ARDUINO IDE

To reprogram a controller requires the Arduino IDE software application. Please **download and install** the latest IDE software from. https://www.arduino.cc/en/main/software [as at writing v1.8.9 – do NOT use the web editor based version]. Once the Arduino IDE is installed, you can start installing the necessary libraries.

COPY THE LIBRARIES FIRST

Unzip the firmware file for myFP2 that you downloaded from the Sourceforge site. Navigate to the **myFP2libraries** folder. Copy **ALL** the folders to the Documents\Arduino\libraries folder (for Windows, for other OS's please consult were the Arduino IDE stores user libraries).

Do not edit or replace any of these provided library files. They have been edited and modified to use with the myFP2 firmware.

ARDUINO IDE CONFIGURATION

Under File, Preferences, turn on line numbers and Show verbose output during compilation.
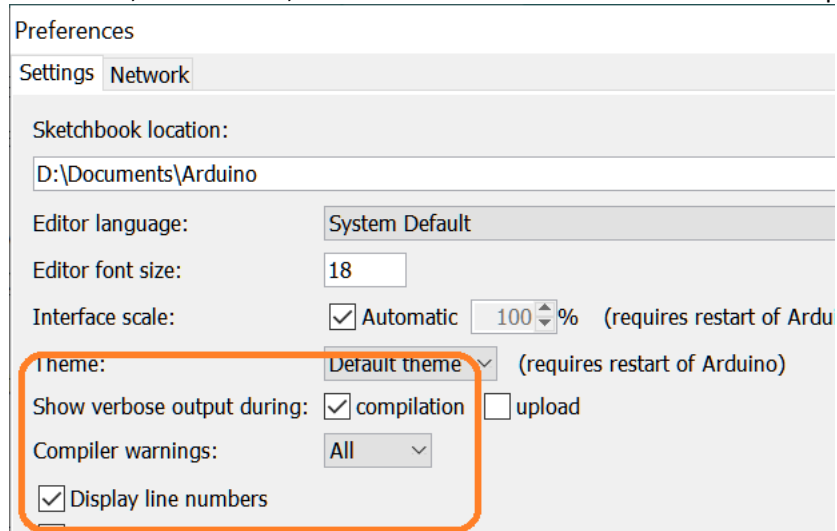
FIRMWARE FILES

Each driver board has its own firmware file. For example, the A4998 driver board has the folder myFP2_A4998_30xx and in that folder is the file used to program a controller with a A4998 driver board.

MYFP2 FIRMWARE OPTIONS

The firmware files that are used to program the controller support different hardware driver board options as well as focuser options such as temperature probe, display and backlash.

Features like the OLED display, Temperature Probe, Backlash are all options. These are either enabled or disabled by the user.

The idea is the user configures the firmware options to match the hardware they have purchased.

These options are in the focuserconfig.h file and most options are disabled by default.

CONFIGURING THE MYFP2 FIRMWARE

The firmware file is comprised of a number of sections in which you will need to change some things.

CHANGES YOU MUST MAKE TO THE MYFP2 FIRMWARE CODE

Before you program the controller, you must make a few changes to the firmware. If you have already programmed the controller, simply make the required changes and then re-program the module again.

A NOTE ABOUT OPTIONS

Any line that begins with two slashes // indicate the line is commented out – ignored by the compiler. To enable an option means making sure that your choice does NOT start with two slashes and the other choices are commented out (begin with two slashes)

SPECIFY THE DRIVER BOARD

You first need to specify the Driver board you are using. This is done in section 1 of the main firmware ino file. The following example shows the driver board as DRV8825TFT22.

```
// ----------------------------------------------------------------------------------------------
// SECTION ONE: SPECIFY DRIVER BOARD HERE
// ----------------------------------------------------------------------------------------------

// Set DRVBRD to the correct driver board, ONLY ONE!!!! - uncomment the correct line for your controller
//#define DRVBRD DRV8825HW203
#define DRVBRD DRV8825HW203TFT22          // Wiring details in myBoards.h
```

Some driver boards may also require you to change an option in the myBoards.h file. If this is a case, a comment will be in this section 1 of the main firmware ino file instructing you to do this.

L293D, L293DMini, L298N, TB6612FNG BOARDS

When using stepper motors with the L293D Motor Shield, L293D-Mini, L298N and TB6612FNG driver boards the number of steps per revolution must be specified in the Arduino firmware file myBoards.h

Enable the correct line which matches the stepper motor configuration you are using. For example, if using a NEMA14 0.9° stepper motor running at half steps, the correct setting is shown below

```
// stepper motor steps per full revolution using full steps, uncomment the correct one that matches your stepper motor
#define STEPSPERRVOLUTION 2048          // 28BYJ-48 stepper motor
#define STEPSPERREVOLUTION  800         // NEMA 14 0.9 running at half steps
// #define STEPSPERREVOLUTION  400       // there are 400 steps for one rotation of the NEMA14 0.9 motor at Full steps
// #define STEPSPERREVOLUTION 1028      // NEMA17-PG5 motor, if half stepping multiply by 2
// #define STEPSPERREVOLUTION 200       // NEMA17 motor, if half stepping multiply by 2
// #define STEPSPERREVOLUTION 5370      // NEMA17-PG25 motor, if half stepping multiply by 2
```

Once changed, upload the modified firmware to your controller.

L293D MOTOR SHIELD DRIVER BOARD

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

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

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

THE focuserconfig.h FILE

This is the main configuration file for hardware options. Section 1 of this file lets you enable options like a display or temperature probe.

HARDWARE OPTIONS

The following example enables the LCD display and the temperature probe, but disables the pushbutton switches

```
// -----------------------------------------------------------------------
// SECTION ONE: SPECIFY HARDWARE OPTIONS HERE
// -----------------------------------------------------------------------
// Caution: Do not enable a feature if the associated hardware circuits are not fitted on the board
// Enable or disable the specific hardware below
```

// To enable the temperature probe, uncomment next line
#define TEMPERATUREPROBE 1

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

// To enable the LCD DISPLAY uncomment the next line (LCD1602, I2C)
#define LCDDISPLAY 1

LCD DISPLAY TYPE

If you are using an LCD display, then you need to set the LCD type in this section. The following example enables the display type LCD2004. You cannot have more than one display type.

```
// -----------------------------------------------------------------------
// SECTION TWO: SPECIFY LCD DISPLAY TYPE
// -----------------------------------------------------------------------
// only uncomment one of the following LCDxxxx lines depending upon your lcd type
//#define LCD1602      1              // 16 character, 2 lines
//#define LCD1604      2              // 16 character, 4 lines
#define LCD2004       3              // 20 character, 4 lines
```

// SECTION OLED DISPLAY TYPE

If you are using an OLED display, this defaults to an 0.96: display using the SSD1306 driver chip. If this is the OLED display you are using, then no change is necessary.

If however you are using a 1.3" OLED display using the SSH1106 driver chip, you will need to make the following change.

```
// ----------------------------------------------------------------------------
// SECTION THREE: SPECIFY OLED DISPLAY TYPE
// ----------------------------------------------------------------------------
// uncomment one of the following USE_SSxxxx lines depending upon your lcd type
// For OLED 128x64 0.96" display with the SSD1306 driver, uncomment the following line
//#define USE_SSD1306    1

// For OLED 128x64 1.3" display with the SSH1106 driver, uncomment the following line
#define USE_SSH1106    2
```

There are no further changes to make.

Once you have made the required changes, reprogram the controller with the revised firmware.

GENERAL RULE FOR OPTIONS

Do NOT enable an option if the hardware for that option does not exist.

For example, if you enable the OLED display in the firmware and the hardware has not been added to the controller - then you can expect some pretty annoying and frustrating problems to occur.

Important settings are stored in the controller EEPROM; are remembered from session to session; and can be configured using the Windows Application as part of the Initial Setup of the myFP2 controller.

Values are retrieved from the controller when connecting using the Windows Application or ASCOM driver. Some of these settings can be updated when connecting to the myFP2 controller.

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

UPLOADING ARDUINO FIRMWARE CODE TO THE UNO/NANO

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 myFP2 controller.

Note: Depending on the Arduino Nano chip you are using, with the newer Arduino IDE v1.8.8+ you may need to specify the Old bootloader to reprogram the chip.
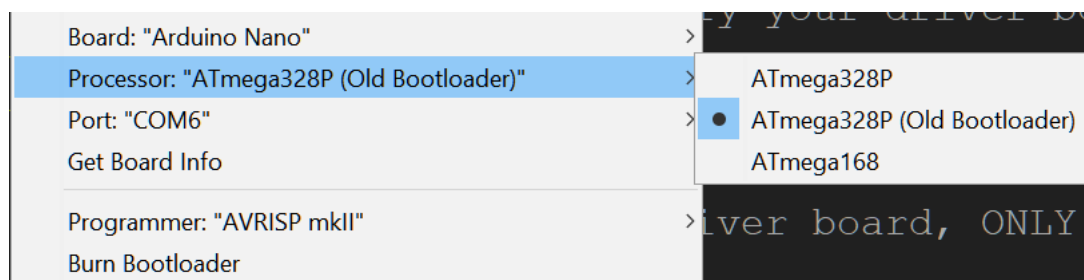
INITIAL SETUP FOR ALL DRIVER BOARDS

By now you should have calculated

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

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

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

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

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

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

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

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

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

1-1 FIRMWARE CHANGE: LCD ISSUES

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

2: SERIAL PORT BAUD RATE

If you want to change the baud rate to a higher value, then it must first be changed in the firmware file myDefines.h 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 SERIALPORTSPEED   9600   // 9600, 14400, 19200, 28800, 38400, 57600

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   // 9600, 14400, 19200, 28800, 38400, 57600

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

3: USING A FULL BOARD WITH NO PUSH BUTTONS?

3-1 Disable Pushbutton code

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

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

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

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

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

EasyDriver v4.4 Pins
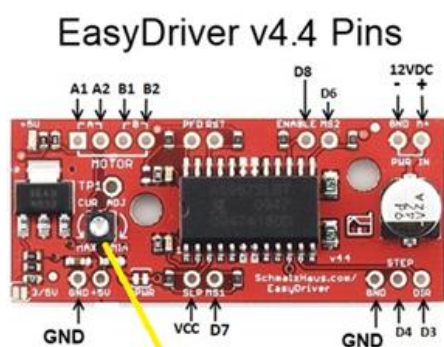
Adjust this Pot to vary the current in the stepper motor coils.

DRV8825
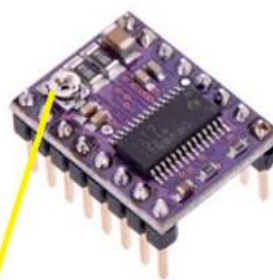
RAPS128

Adjust this pot to set current

Adjusting the Stepper Motor manually

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

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

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

5: SET STEPSIZE AND MAXIMUM PERMISSIBLE VALUE FOR MAXSTEP

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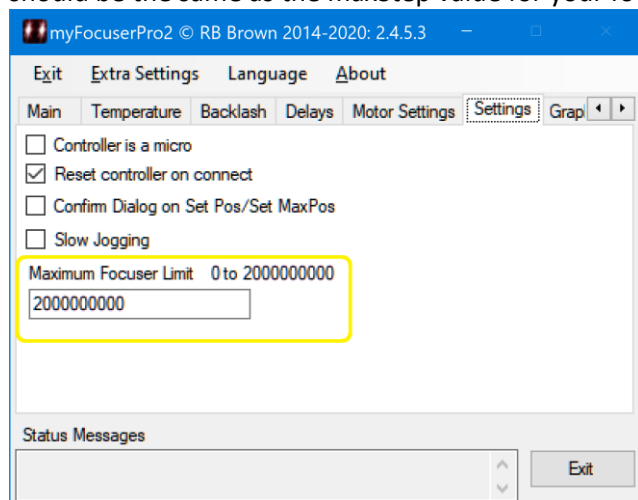c) Start the windows application BUT do NOT connect to the focuser yet

5-1: SET STEP SIZE

Select the MotorSettings tab. Enter the correct settings for step size.
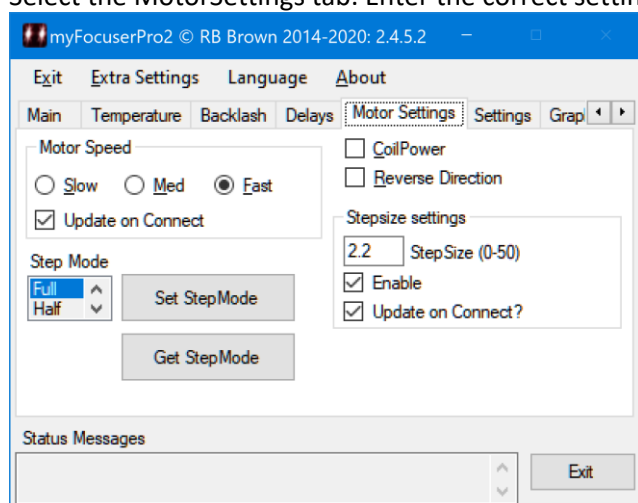
5-2: SET MAXIMUM FOCUSER LIMIT

Select the Settings tab. This displays the following (note that your values can be different). In all cases, this should be the same as the maxStep value for your focuser.


5-3: CONNECT TO THE CONTROLLER

Select the Connection tab. Select the correct COM Port and Port Speed then click the Connect button to connect to the controller.

5-4: SET STEP MODE

When the controller connects, the step mode is retrieved from the controller. Select the Motor Settings tab. From the drop down list, choose the correct step mode, then click the Set Step Mode button to update the controller. The step mode is now saved in the controller and will be remembered.
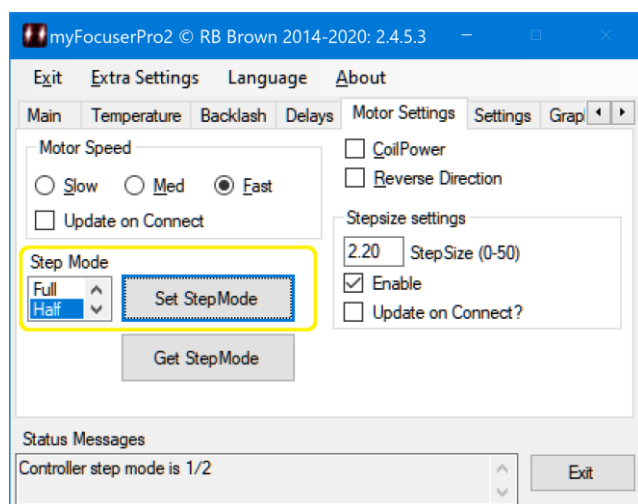
The maximum value of step mode is 128. However, only the RAPS128 and ST6128 driver boards support this level. Each driver board type has its own limit for micro-stepping. If you try to send an invalid step-mode to any driver-board it is ignored by the controller firmware. For instance, sending a step-mode of 64 to a DRV8825 controller will cause the controller to ignore the request and the controller will use the default of half-steps.

**THE FOLLOWING ONLY APPLIES TO ALL EASYDRIVER/DRV8825/RAPS128/A4988 FIRMWARE CODE**

If you are using a stepmode which is $\frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ or greater then you will need to have the diodes in place else the stepper motor may have trouble moving smoothly at the selected step mode.

5-5: SET FOCUSER POSITION TO 0
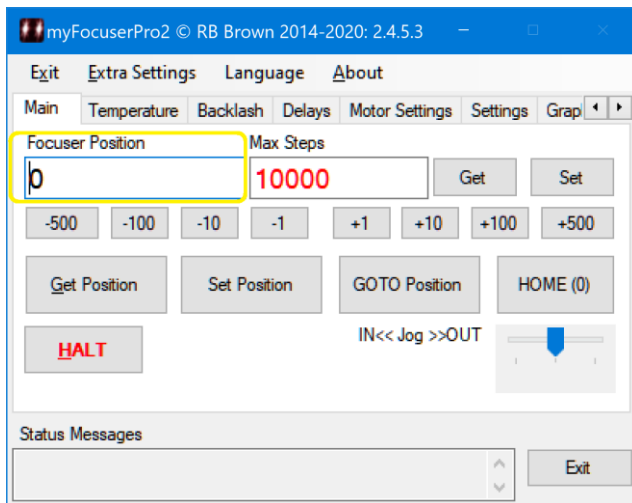
5-6: SET MAXSTEPS

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.
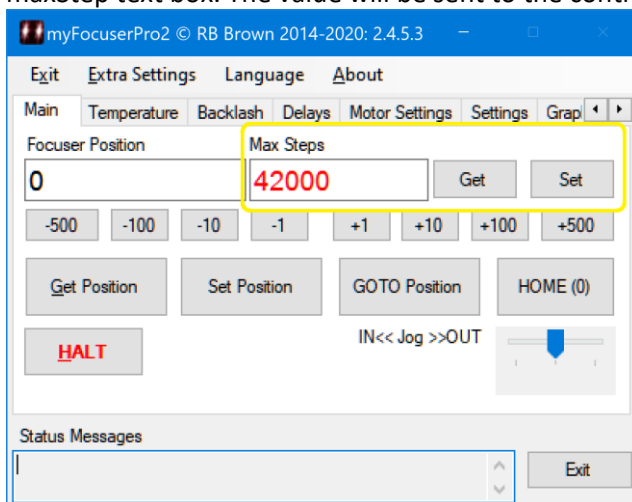
6: CHECK THE SPEED SETTINGS

The next step is to check the speed settings of Slow, Medium and Fast. Speed control is on the Motor Settings tab.

  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

7: TEST 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. Reverse direction is on the Motor Settings tab.

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

8: 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. Coil Power is on the Motor Settings tab.

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

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

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

If you need to set up backlash [on the Backlash tab], consider doing this once you have your focuser configured and working. Please see the section on backlash to determine how these settings can be determined and configured.

CONGRATULATIONS: Your focuser is now ready to use!

ADDITIONAL FIRMWARE SETTINGS

These settings appear in SECTION 1 of the file *focuserconfig.h*

SUPERSLOWJOGGING

This option allows an extra delay to be inserted in jogging moves to enable very slow jogging to take place.

// To enable the super slow jogging, uncomment the next line
//#define SUPERSLOWJOGGING 1

SHOW BOOTMESSAGES

This option will display a number of boot messages during startup (on the display) and should only be used for diagnostic purposes.

// To enable the start boot screen showing startup messages, uncomment the next line
//#define SHOWBOOTMSGS 1

DISABLE UPDATE OF DISPLAY DURING MOVES

This option is very useful for using the push buttons. It disables the updating of the display and temperature till 5 seconds has elapsed from the last move. This gives better response to the push buttons. The default is to have this option disabled. If defined, then this option is automatically enabled.

// To disable the update of the display till after a move has finished, uncomment the next line
//#define DELAYEDDISPLAYUPDATE 1

The downside of this option is that the display [if enabled] will only be updated after 5s has elapsed since the last pushbutton move occurred.

WINDOWS APPLICATIONS

myFP2 WINDOWS APPLICATION

The myFP2 application gives full control of the focuser (supports ALL build options). Note that the majority of settings are remembered/saved by the application or within the controller. The Menu provides access to Exit (quit the program), Settings, Language and About (Copyright message).

The application is arranged as a series of tabs. When first started, the application will ask for the location where data logfiles and error logfiles will be stored.
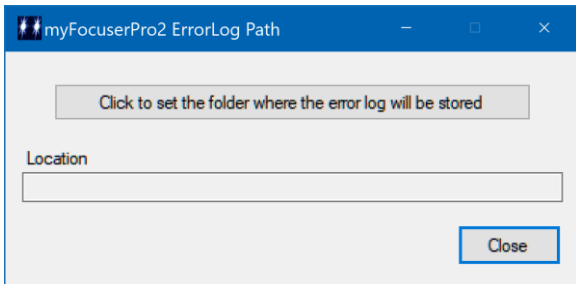
Click the button and select the desired location. Once selected, the form will look like
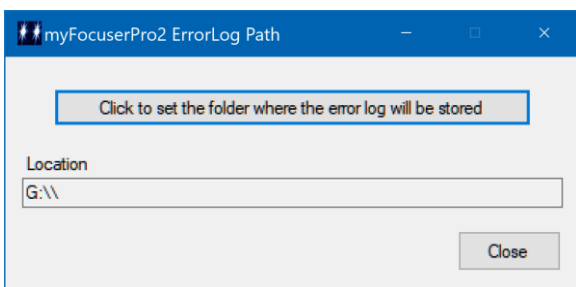
Click the Close button to continue. The application displays the connection tab.

TYPING NEW VALUES INTO TEXT FIELDS

After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.

CONNECTION TAB

The connection tab allows you to specify the serial port [COM port] and speed [9600] to which the controller is connected. Buttons are also provided for Connecting, Disconnecting and Refreshing the list of serial ports.
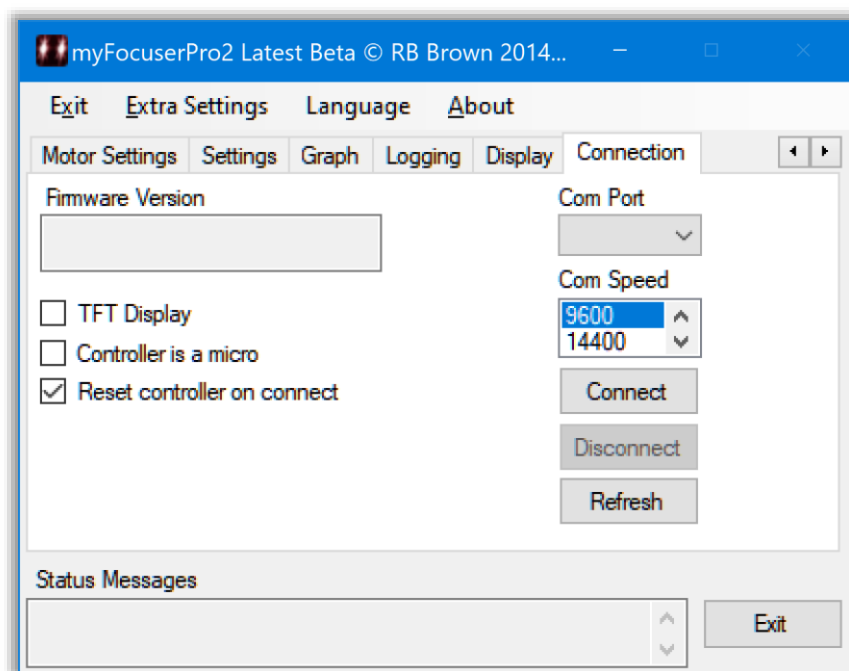
Select the correct COM port from the drop-down list and select 9600 and the speed. Click Connect to connect to the controller.

Once connection is established, controller parameters will be retrieved and then the Main tab will be displayed.

☑ **TFT Display**
When checked allows a 10 second delay on Connect to allow the controller to initialize the TFT display correctly.

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

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

COM Port - Use the dropdown list to select the correct comport that the myFP2 is connected to

Com Port Speed - Select the baud-rate that matches the firmware in the controller (default 9600)

Connect - After selecting the correct comport, click Connect to connect to the myFP2 controller.

Disconnect - disconnects the myFP2 controller.

Refresh - refreshes the list of available comports.

MAIN TAB

The main tab provides basic control of the focus controller, with operations such as move, halt, go and home. The jogging control is also on the main tab. Status messages are displayed in the fields at the bottom of the application. The Exit button will exit the application.
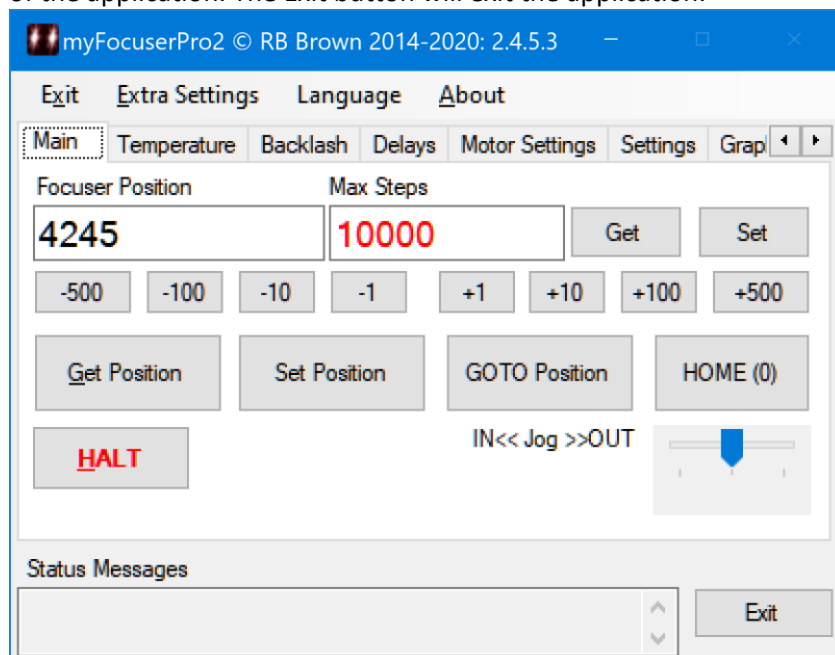
Focuser Position

This text fields displays the current focuser position. This filed can be changed by typing a new value in the text field. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.* If a new value is entered, you can then goto the desired position by clicking the GOTO Position button, or you can change the current focuser position to the new value [without moving the focuser] by clicking the Set Position button.

Goto Position

Type in the desired focus position in the Focuser Position text box (digits only, < maxStep, 0 or > 0) and press the ENTER key to finish the text entry. Next click the GOTO Position button. The focuser will move to the specified position.

Get Position

Returns the current myFP2 controller position in the Focuser Position text box.

Set Position

Type in the desired focus position in the Focuser Position text box (digits only, less than Maximum Position, 0 or greater than 0), press ENTER to terminate the text entry, then click the Set Position Button. The focuser will NOT move but the focuser position is updated.

HOME

The focuser will move to position 0 and stop.

Get Maximum Position

Returns the current myFP2 maxStep value in the Maximum Position text box.

Set Maximum Position

Type in the desired maxStep value in the Maximum Position text box (digits only, greater than FocuserPosition) and terminate the text entry by pressing the ENTER key, then click the Set Button.

HALT

Halts the focuser if currently moving.

-500 to 500

These buttons provide a means to move the myFP2 controller by a specified number of steps. For example, clicking +10 will move the focuser +10 steps. Moves are not permitted when temperature compensation is enabled.

Jogging

This control lets the user move the focuser under jog control. Dragging the control to the left moves the focuser IN whilst dragging the control to the right moves the focuser OUT. The speed is controlled by the motor speed setting. Jogging continues as long as the mouse is held down or the control is NOT at the midpoint of the control. When the mouse is released, the control is set to 0 and de-activated, jogging stops and the current focuser position is retrieved from the controller. Jogging also stops if the HALT button is clicked. A single click on the control stops the focuser moving by sending a HALT command to the controller.

Status Messages

Text box used to indicate status messages and responses from myFP2 controllers.

EXIT

Exits the application.

TEMPERATURE TAB

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

Get Temperature
Requests the temperature from the myFP2 controller (if a temperature probe is supported and attached) and display the adjusted value in the Temperature Text Box.

☑ Celsius
When checked displays temperature value in Celsius

☑ TC Available
Updated when connected to a myFP2 controller. If checked, the controller supports temperature compensation

☑ TC ON/OFF
If the controller supports temperature compensation, this checkbox can be used to turn temperature compensation in the controller ON or OFF

Temperature Compensation

Steps per degree (0-400) is the coefficient value, which determines how many steps the focuser will move on a one-degree temperature drop (each focuser is different and this value must be calculated). *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

☑ **Direction**

When checked means that the focuser temperature compensation move will be applied inwards. When unchecked, the focuser temperature compensation move will be applied in an outwards direction.

☑ **Update on Connect**

When checked means that the entered coefficient value and the Direction value will be sent to the controller when connecting

Temperature Precision

The drop down box provides a list of the currently available temperature precision settings, from 0.5 degree to 0.06125 degree resolution. The current highlighted value is shown.

☑ **Update on Connect**

When checked means that the selected temperature precision setting will be sent to the controller when connecting. If unchecked, then the temperature precision setting will be updated once connection to the controller occurs.

BACKLASH TAB
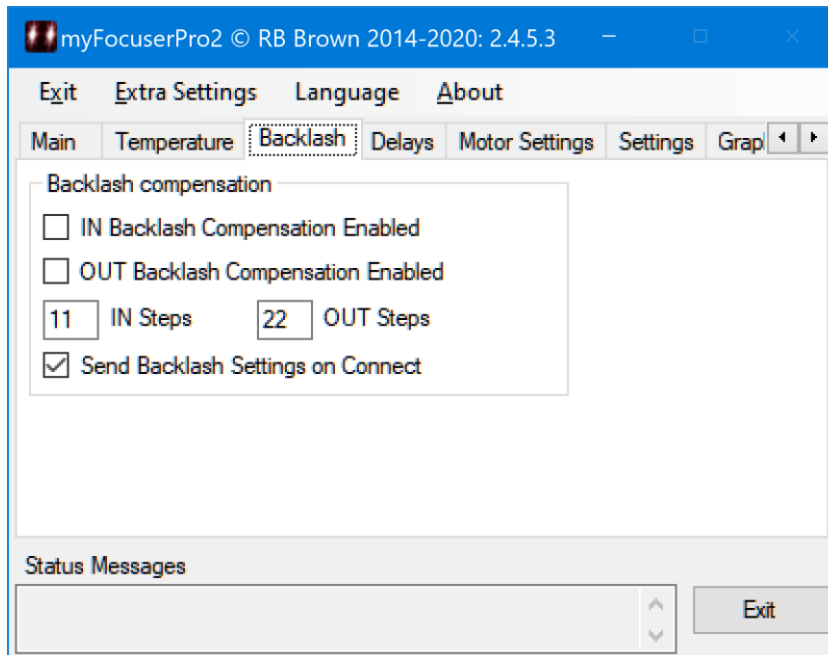
The myFP2 supports backlash for both IN and OUT directions. Backlash is only applied if the move is in the OPPOSITE direction to the previous move, and the number of backlash steps is applied BEFORE the move is executed.

☑ **IN Backlash Compensation Enabled**
Enable or disable backlash compensation of IN moves

☑ **OUT Backlash Compensation Enabled**
 Enable or disable backlash compensation of OUT moves

IN Steps (0-255)
The number of steps of backlash to apply after the IN move has completed. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

OUT Steps (0-255)
The number of steps of backlash to apply after the OUT move has completed. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

☑ **Send Backlash Settings on Connect**
Check this box to send the backlash settings to the controller on connect. If unchecked, the values are updated from the controller when connection is made.

DELAYS TAB

Delay (s) 2-10 on Connect
Specify the delay in seconds after connecting that the driver will wait before sending a command to the myFP2 controller. Valid values are 2-10 seconds. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*
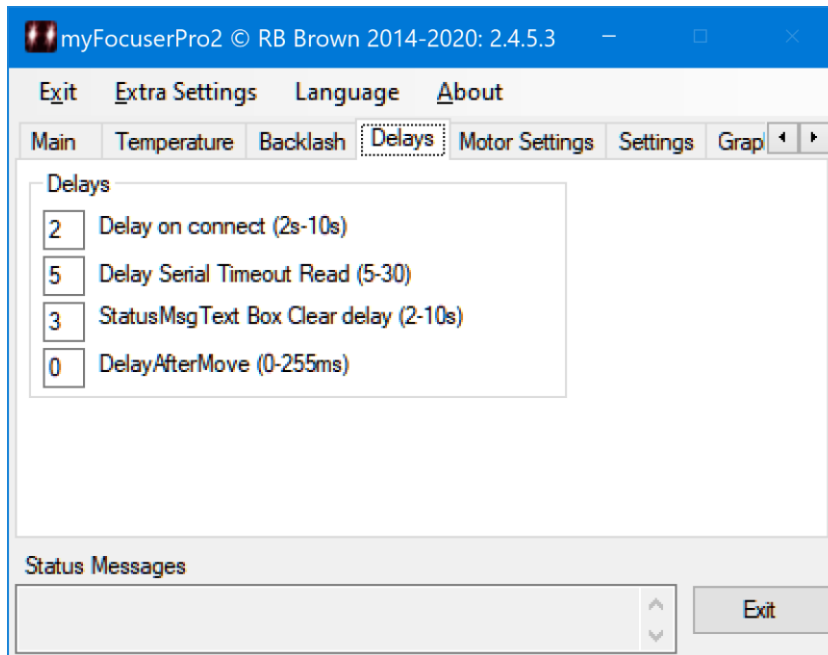
Delay (s) 5-30 timeout on Serial Read
Specify the delay in seconds that the application will wait when attempting to read from the serial port after sending a command to the myFP2 controller (default = 5). For Bluetooth or slower devices, a value of 8 or 10 may suffice. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

StatusMsg Text Box Clear after delay
After writing text to the Status messages text box, this delay specifies the number of seconds that will elapse before the text box is automatically cleared. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

DelayAfterMove
Specifies the delay in milliseconds (0-255) that is applied after move to prevent any blurring of star caused by any mechanical residual movement or vibration of the stepper or focuser during focusing. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*
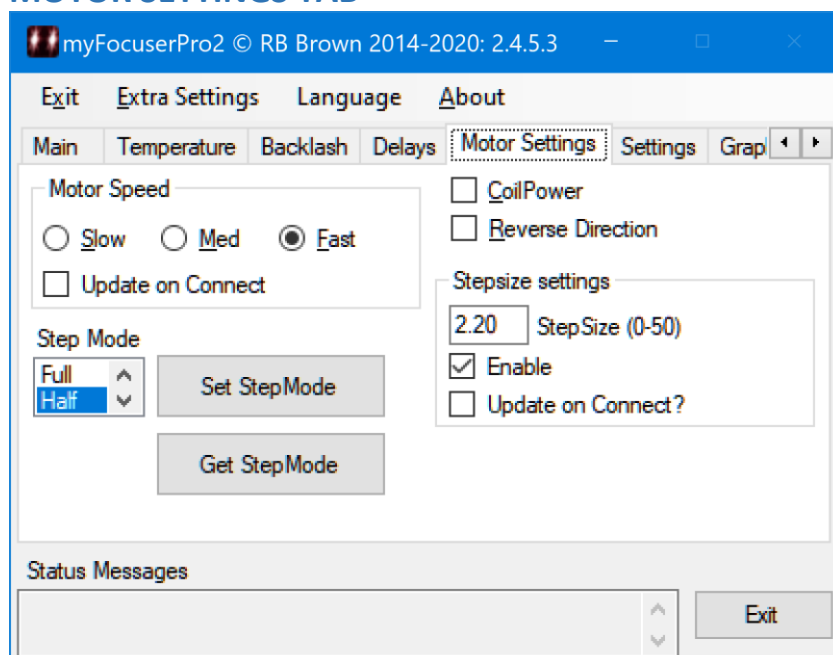
MOTOR SETTINGS TAB

Set StepMode

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

Get StepMode

Click the Get StepMode to get the current stepping mode from the myFP2 controller

MotorSpeed

This controls the relative speed of the stepper motor. The speed can only be changed when the focuser is NOT moving. If the Update on Connect option is checked, the motorSpeed is sent to the controller when connecting, otherwise the speed setting will be updated after connection to the controller occurs.

☑ Coil Power

When ON coil power is ON [checked] and the stepper coils remain powered after the move is completed

☑ Reverse Direction

When ON [checked] the motor moves in the opposite direction (IN means OUT and OUT means IN)

Step Size in microns (0-50)

Defines the StepSize in microns. You can enter your measured / calculated value for your focuser and this will be sent to the myFP2 controller when connecting. This allows client applications using the ASCOM driver to retrieve the StepSize setting from the myFP2 controller. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.* Please see the section in this PDF on determining your focuser step size value.
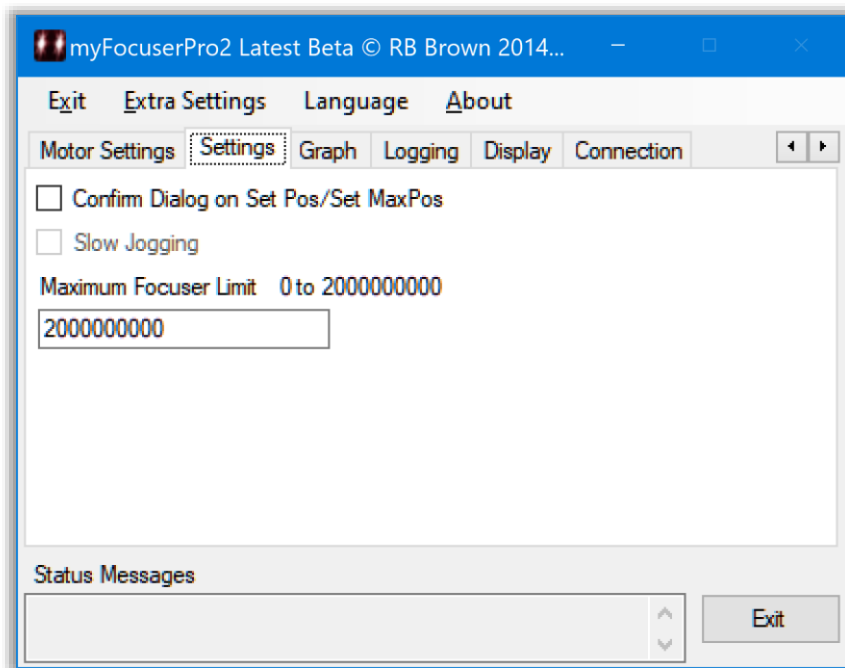
☑ Enable in controller?

Turns ON or OFF the reporting of the StepSize value from the myFP2 controller. If you do not know your step size value, DO NOT enable this feature.

☑ Update on Connect

Enable the checkbox to send these values to the myFP2 controller on Connect.

SETTINGS TAB

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

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

Maximum Focuser Limit
Specify the maximum number of steps for the focus controller and will vary depending upon each user's configuration. You should set this to the maximum number of steps that you have calculated for your focuser [see determining maxStep]. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.* This setting must be specified before connecting to the controller, and will be remembered by the application. See the section on the initial setup of the focuser for further information.

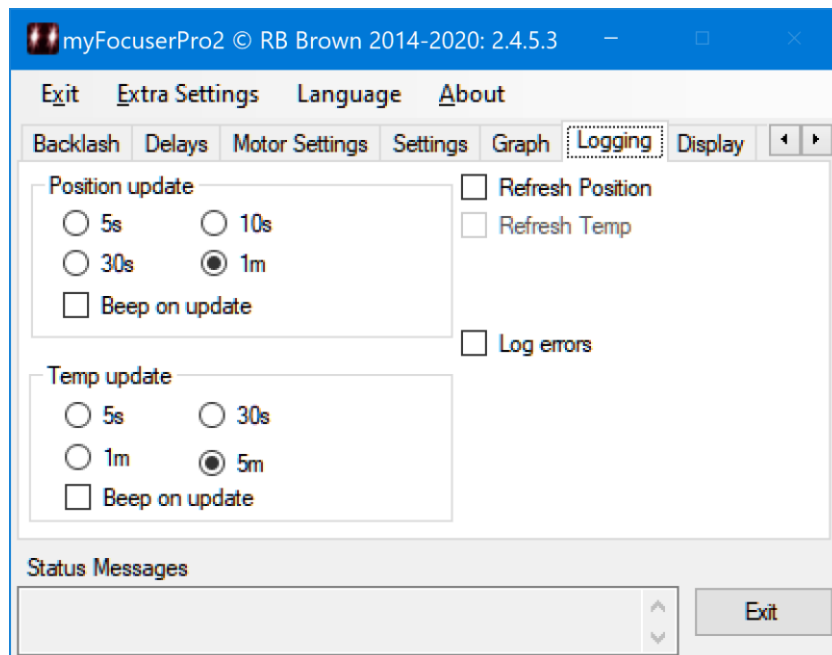## ☑ SlowJogging
When checked enables super slow jogging

GRAPH TAB
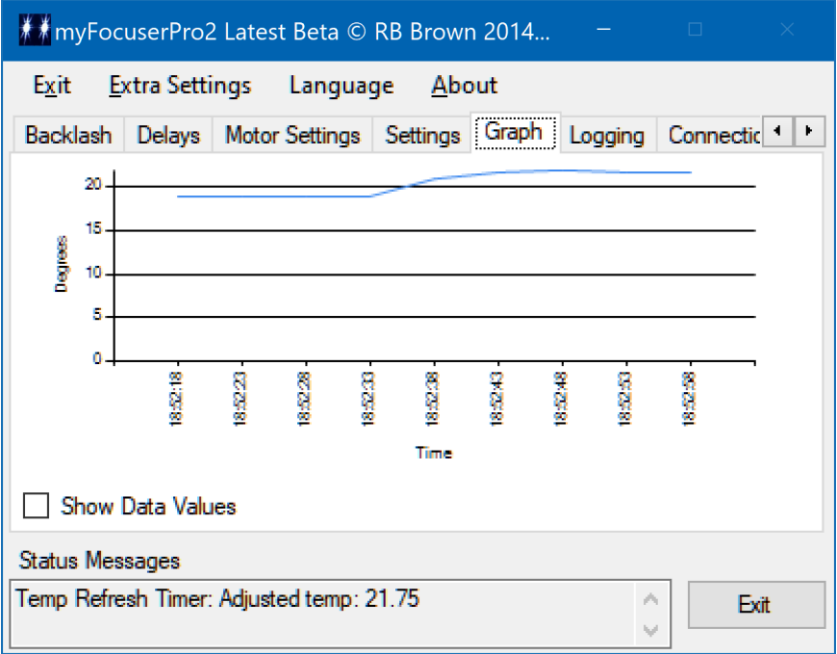
LOGGING TAB

☑ **Refresh Temp**
When ON [checked] the application will periodically poll the myFP2 controller and request a temperature update. The polling interval is set under the settings menu

☑ **Refresh Position**
When ON [checked] the application will periodically poll the myFP2 controller and request a position update. The polling interval is set under the settings menu

Refresh Position
Select the desired (5s, 10s, 30s, 1m) refresh interval for getting the current focuser position from the controller (when connected).

Refresh Temp
Select the desired (5s, 30s, 1m, 5m) refresh interval for getting the current focuser temperature from the controller (when connected). In the Windows app we now have a separate timer for Position and Temperature. For Temperature compensation, you must set the temperature coefficient AND start the Refresh Update timer BEFORE enabling temperature compensation.

☑ **Log Errors**
To generate a log file for debugging and trouble-shooting purposes, check the Log Errors checkbox.

DISPLAY TAB
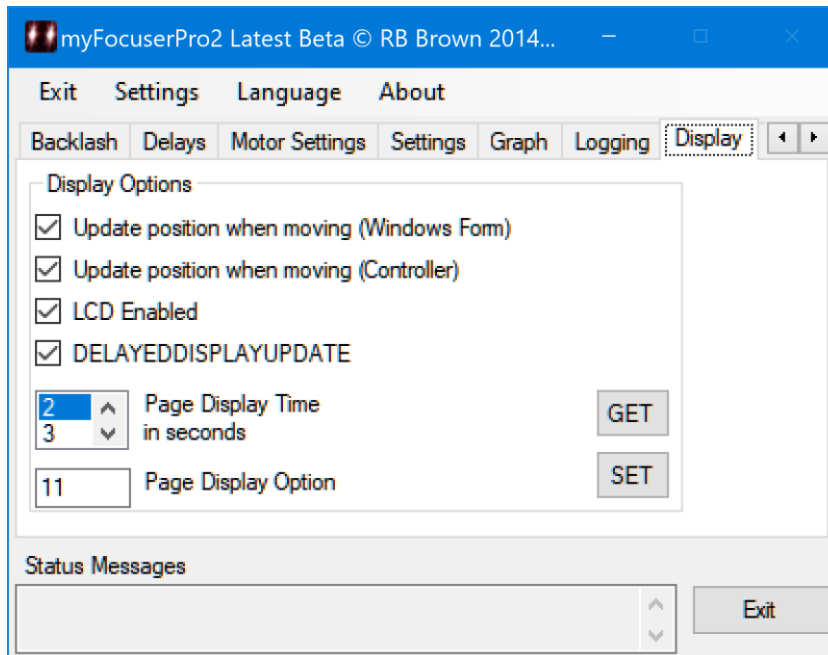
☑ **Update position when moving (Windows)**
When checked, will update the focuser position on the main form when a long move is sent to the controller. The focuser position on the form will be updated during the move.

☑ **Update position when moving (Controller)**
When checked, will update the focuser position on the controller display when a long move is sent to the controller. The focuser position will be updated once per second. On some builds, this might cause the focuser to slightly pause every second, so the default value is unchecked.

☑ **LCD Enabled**
When checked enables the controller's display

☑ **DELAYDISPLAYEDUPDATE**
 When checked disables updating of the display and temperature till 5s has elapsed after a move has completed.

Page Display Time
The time in seconds that a display screen is displayed for can be adjusted using this menu option, from 2s-10s delay. If I want the pages to show for 3 seconds, I would select 3 from the dropdown list and click the SET button.

Page Display Option
This option provides a means of enabling or disabling each display page. The text box accepts a string of binary digits [0 or 1]. If the bit=1 the page is shown; if the bit=0 the page is not displayed.

The least significant bit = page 0

Example
I want pages 0,1 showing I would write
0011

then click the SET button. The default is 11 – only the first two pages are shown.

Note the following for the number of pages for each display type

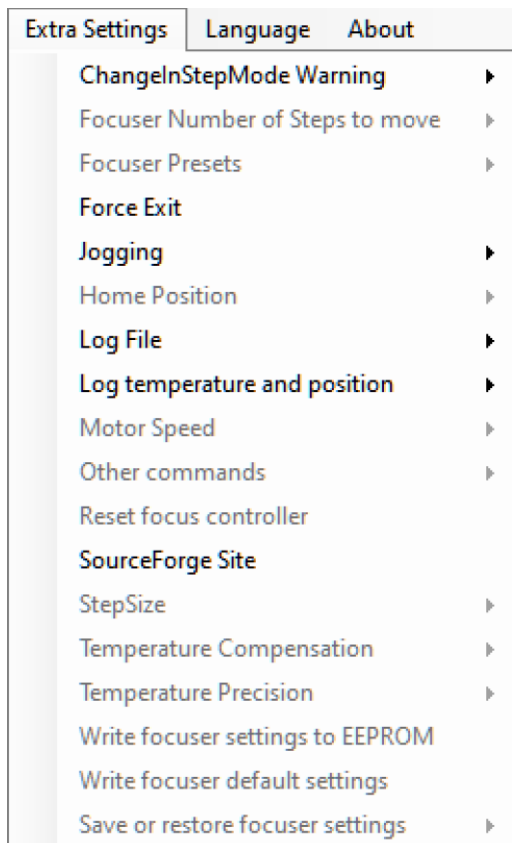| Display Type	Number of pages	Binary
LCD1602	5	11111
LCD1604	2	11
LCD2004	2	11
OLED	2	11
Nokia	2	11
UTFT	1	1
TFT	1	1

GET
Get the settings from the controller.

SET
Send the current values to the controller.

THE SETTINGS MENU

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

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

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

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

Force Exit
In the event of problems, this provides a clean method of exiting the program **Jogging** - Get the jogging status and the last jogging direction

Get Controller Firmware Version
Get the firmware version of the connected myFP2 controller

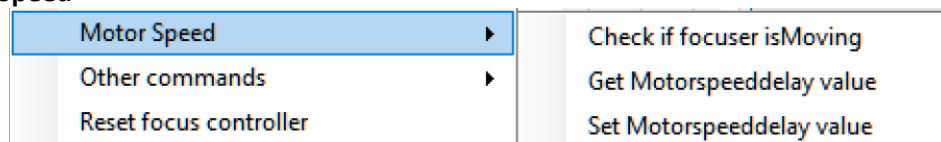**Home Position**
Get status of home position switch

Log File
  - Reset Error Log File path displays a dialog box to specify the drive and directory where the error log file and other log files are to be stored.

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

Motor Speed

Check if focuser isMoving
Gets the moving status of the controller

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

SourceForge Site

Opens a web browser and display the myFP2 website where you can download the latest drivers, software and documentation.

Step Size

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

Temperature Compensation

Please read the section on temperature compensation.

Temperature Precision

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

Write Focuser Settings to EEPROM

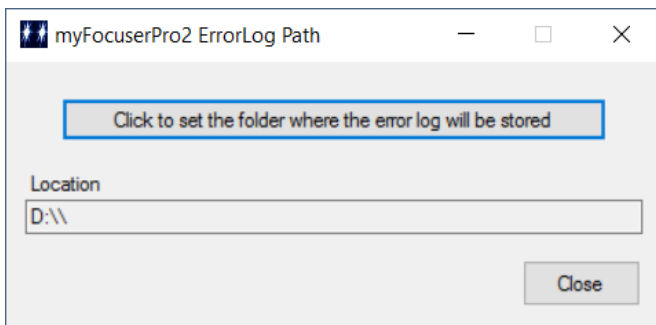Save the controller settings in EEPROM.

Write focuser default settings

Used to write a standard configuration setting to the myFP2 controller. The default configuration is specified within the Arduino firmware file. This may be useful after upgrading the firmware file to a newer version. For more information, click here.

Save and Restore Focuser/App settings

Provides for the saving and restoring of focuser and application settings. This is designed to be used before and after upgrading the Windows application and myFP2 controller firmware.
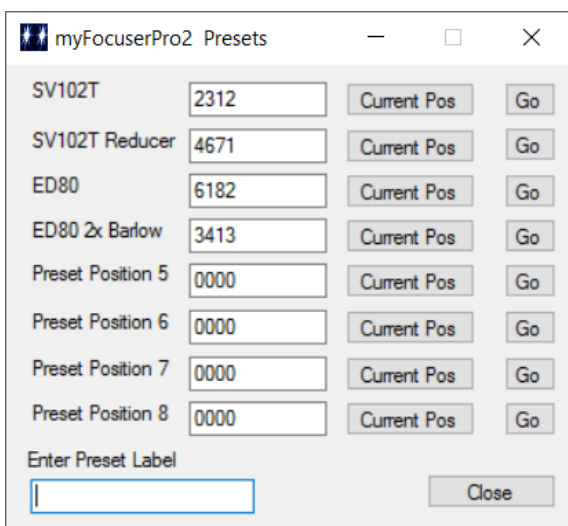
THE ERRORLOGPATHNAME FORM

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

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

THE FOCUSER PRESETS SETTINGS FORM

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

You can enter up to 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 Status messages textbox and the move cancelled).

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

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

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

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

Preset settings and Labels are remembered by the application.

USING THE REFRESH TIMER SOUNDS

With Windows, the refresh timers can generate PC system sounds when the update occurs. For the Temperature refresh timer, the Windows system sound is Beep, and Exclamation is used for the Position refresh timer. The sound for each refresh timer is independent and can be enabled or disabled.

These sounds can be configured in the Windows environment by accessing the Sound control.
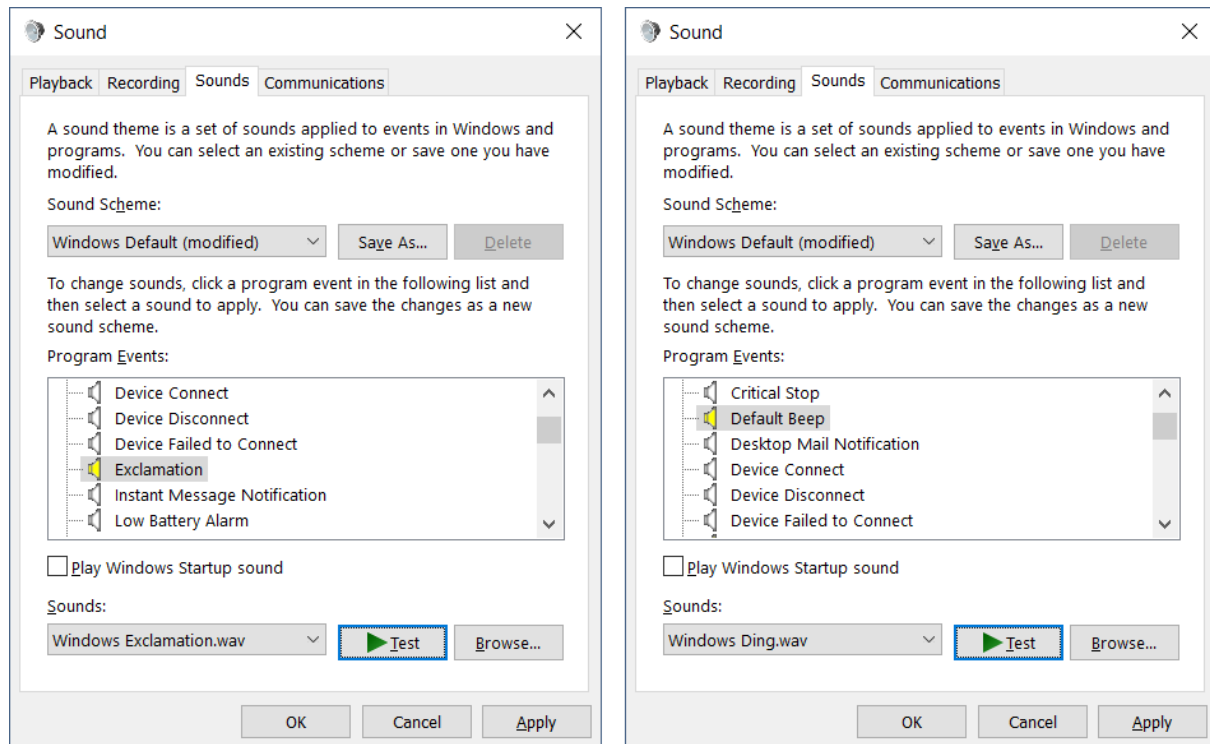
LANGUAGE SUPPORT
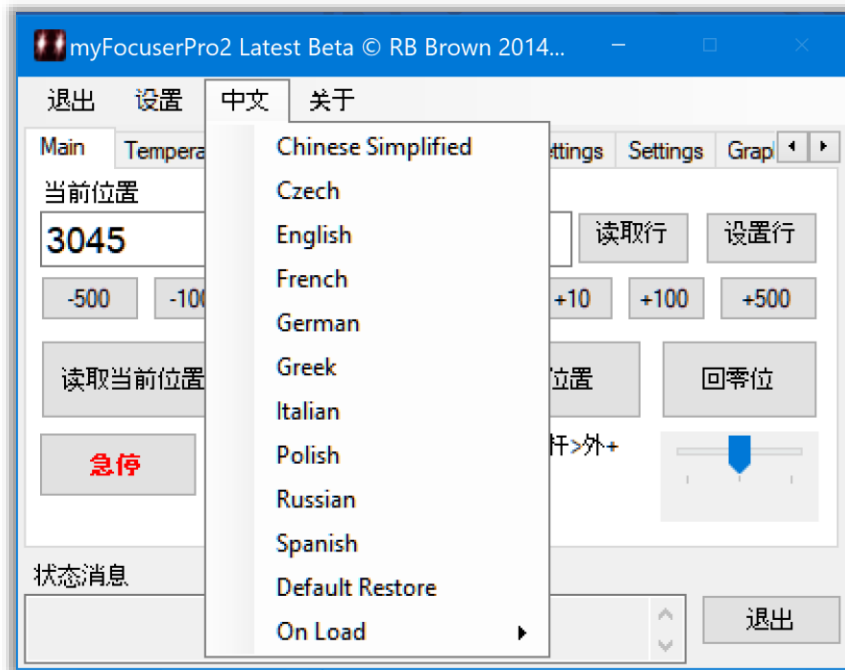
Current supported languages are

        Chinese (Simplified)
        Czech
        English
        French
        German
        Greek
        Italian
        Polish
        Russian
        Spanish

Other languages can be added. Translations have been done using online translation sites and may not be a true or accurate translation.

THE BACKLASH SETTINGS

Notes about backlash

Backlash is only applied if
1. Backlash is enabled, and
2. The new focuser direction is not the same as the previous focuser direction

The backlash compensation in myFP2 works as follows.
1. When a move is requested, the direction of the move is determined to see if it is the opposite direction to the previous move.
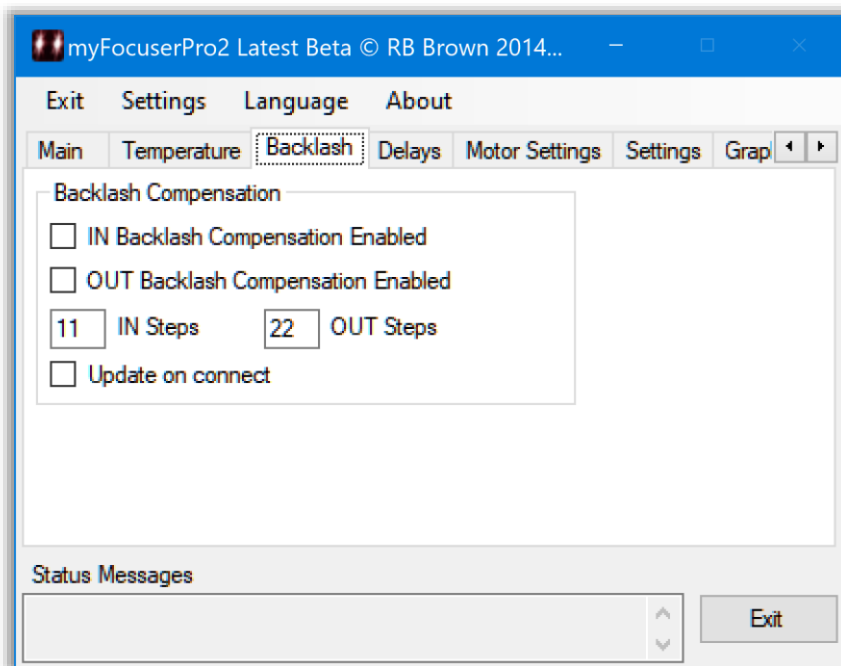2. If the answer is yes, then (if backlash compensation is enabled for the direction being requested) the focuser firmware performs the backlash steps without adjusting the target position (but ensures that it does not exceed the boundaries of 0 and maxSteps).
3. The focuser then performs the move to the desired target position

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

How to determine and set Backlash

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

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

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

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

Open the Backlash tab and set the backlash as indicated on the left.

Click on the Connection tab and connect to the controller.

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

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

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

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

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

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

Backlash and ASCOM drivers
As ASCOM provides no means for handling backlash, the settings will only be able to be configured on the extra settings form from the setup-dialog form for ASCOM (in the same way that MotorSpeed and other

values are handled). Any backlash setting can then be sent to the controller when connecting and would then be set and remembered.

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

Do Not Be Fooled

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

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

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

THE SETTINGS SAVE/RESTORE OPTIONS

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

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

Save and Restore Focuser/App settings ▶	Restore from file
	Save to file
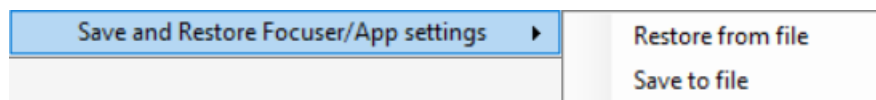
Restore from file - Restores the previously saved myFP2 controller and Windows application settings from a text file in the specified folder. After restoring the settings, the application will save the myFP2 controller settings to EEPROM, reboot the myFP2 controller, and exit the application. Next time the application is started the restored settings will be in effect.

Save to file - Saves the current myFP2 controller and Windows application settings to a text file in the specified folder (the name of the file is automated)

JOGGING

The Windows application provides an ability to jog the focuser in either direction.Dragging the control to the left moves the focuser IN whilst dragging the control to the right moves the focuser OUT.

The speed is relative to that of the motor speed setting.

Jogging continues as long as the mouse is held down or the control is NOT at the midpoint of the control. When the mouse is released, the control is set to 0 and de-activated, jogging stops and the current focuser position is retrieved from the controller.

A single click on the control stops the focuser moving by sending a HALT command to the controller.

ISSUES WITH JOGGING

If BACKLASH is enabled, then backlash will be applied to jogging moves.

If backlash out is enabled, and backlash out steps is set to 50, then consider the following scenario.

1. The user sets the speed to SLOW and then drags the jogging control to move the focuser in
2. The focuser begins to move inwards
3. The user sees that they have gone past best focus so they drag the control to move out
4. The focuser immediately applies backlash out and moves the focuser by 50 steps out (the backlash out amount), then starts jogging one step at a time.

If you intend to use jogging for MANUAL FOCUS, it is best to disable backlash.

SUPER-SLOW JOGGING

Firmware 289+ introduces super slow jogging, where the motor steps are much slower. The option for slow jogging applies to jogging only and is specified in the firmware section 2: SPECIFY HARDWARE OPTIONS HERE

To enable super slow jogging, uncomment the line in the firmware as shown below

// To enable the super slow jogging, uncomment the next line
#define SUPERSLOWJOG 1

THE DELAY USED BY SLOWJOGGING

Super slow jogging inserts a delay onto the timing loop for motor steps when jogging. This delay is specified in the *myDefines.h* file. The maximum value for this should be 250 (values are milliseconds).

#define SLOWSPEEDJOGDELAY   250   // milliseconds delay for slow speed jogging

INITIAL CONNECTION TO THE myFP2 CONTROLLER?

Some values are sent to the controller when connecting and some values are retrieved from the controller when connecting. Please see the following video's
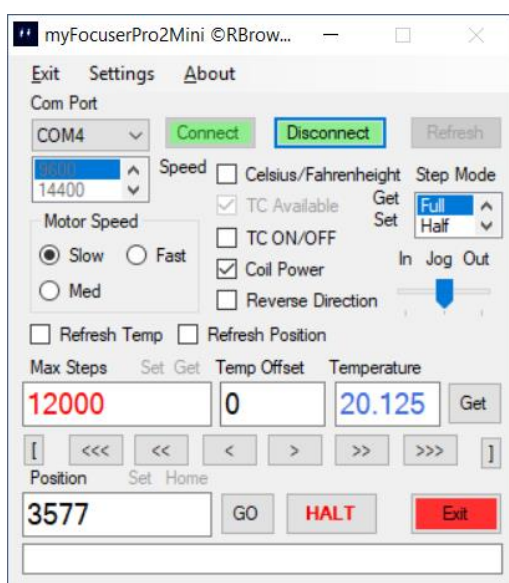
https://www.youtube.com/watch?v=mZmWG_vQ1uM

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

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

myFP2 MINI APPLICATION

The myFP2Mini is a slimmed version of the myFP2 Windows Application program (and several revisions behind), but retains most of the same features. It is designed to be compact and save on screen real-estate.


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

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

The main differences are as follows

- Some settings are moved to the Settings form and must be set before connecting to the myFP2 controller
- After connection is established, the Disconnect button will flash and the computer will beep (if sound enabled) two times to indicate that the application has successfully connected
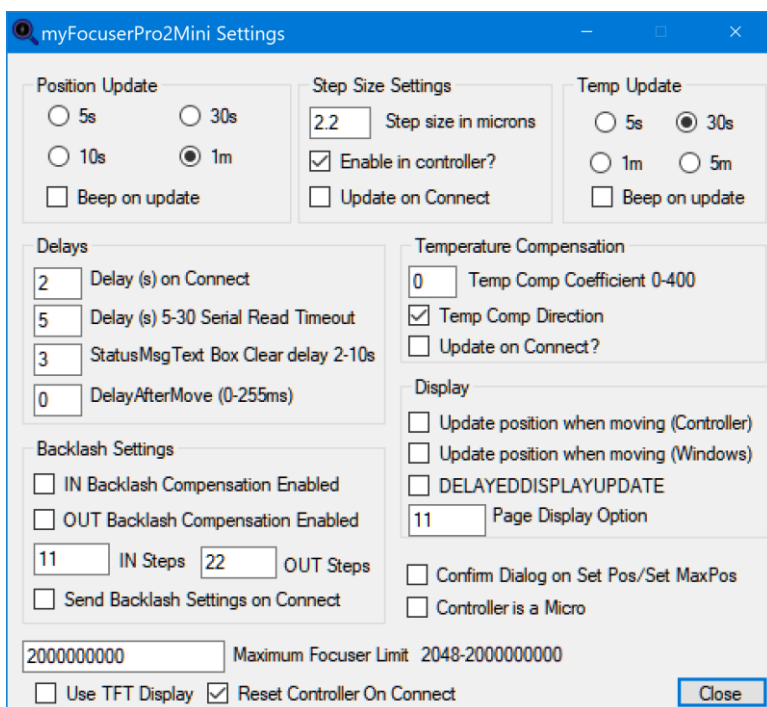
DATA LOG VIEWER

The option "**Log Temperature and Position**" allows the user to save the temperature probe reading and the focuser position value to a log file for later analysis (such as determining temperature compensation values for your focuser setup).

This feature is **enabled** only when the **automate** feature is enabled. During the automatic update (when the automate check box is checked), the routine will get the current temperature and focuser position from the myFP2 controller. If the temperature logging file is enabled, then these values will also be written to the associated log file.

Below is an example of the application **myLogViewerPro**, which can display both myDewControllerPro3 and myFP2 data log-files for analysis.
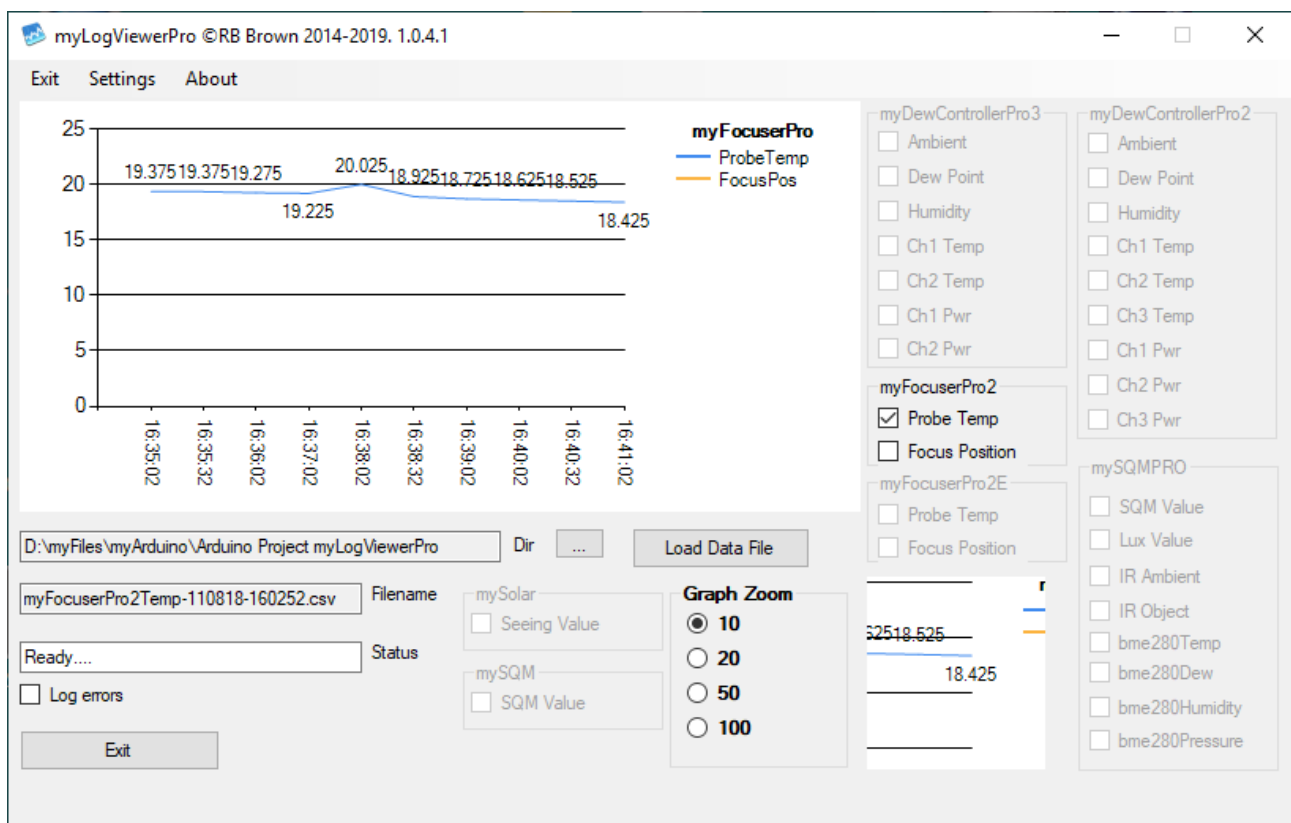
myFP2-ANDROID-BLUETOOTH

The provided Android application was developed and tested on a Samsung S9+ phone and Samsung Tablet A4.

Tips for best operation before starting the application

1. It is best to start with the Bluetooth enabled
2. Turn off screen rotation
3. Disable any screen saver or screen display timeout
4. Disable any running apps which might interrupt the application (such as email, notifications etc)
5. As GPS and Bluetooth consume a lot of battery power, it is best to have the phone connected to a charger when using this application

myFP2 Bluetooth Android App
https://www.youtube.com/watch?v=I7mUmHR_bx8

How to pair the BT Controller - 1
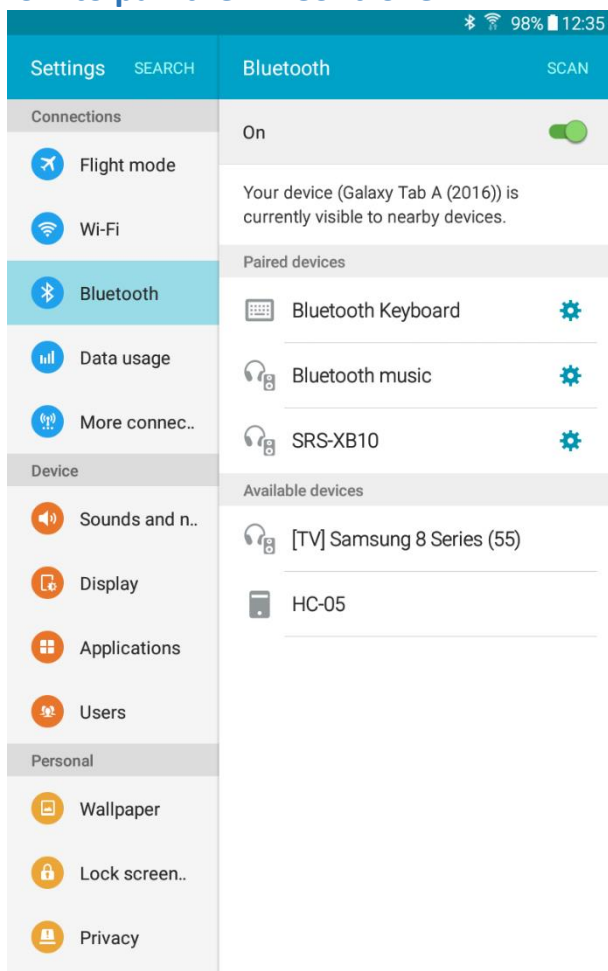
To first setup the HC-05 adapter to be recognized and paired to the Android device, turn on the Bluetooth enabled adapter.

Open the Bluetooth settings of the Android device and you should see a device which is a series of digits separated by colons or might appear as HC-05.

How to pair the BT Controller – 2
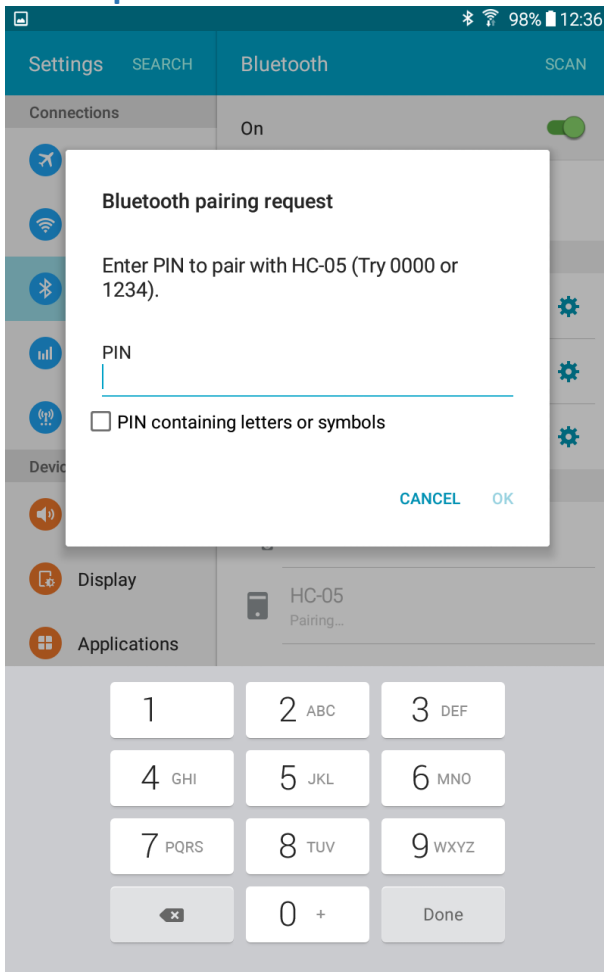
If you tap on the HC-05 Bluetooth device, a pairing request will be established and you will need to enter a pin number to pair the HC-05 device to the Android device.

The pin number is 1234

After entering the pin number, tap the OK text on the dialog box.

How to pair the BT Controller - 3
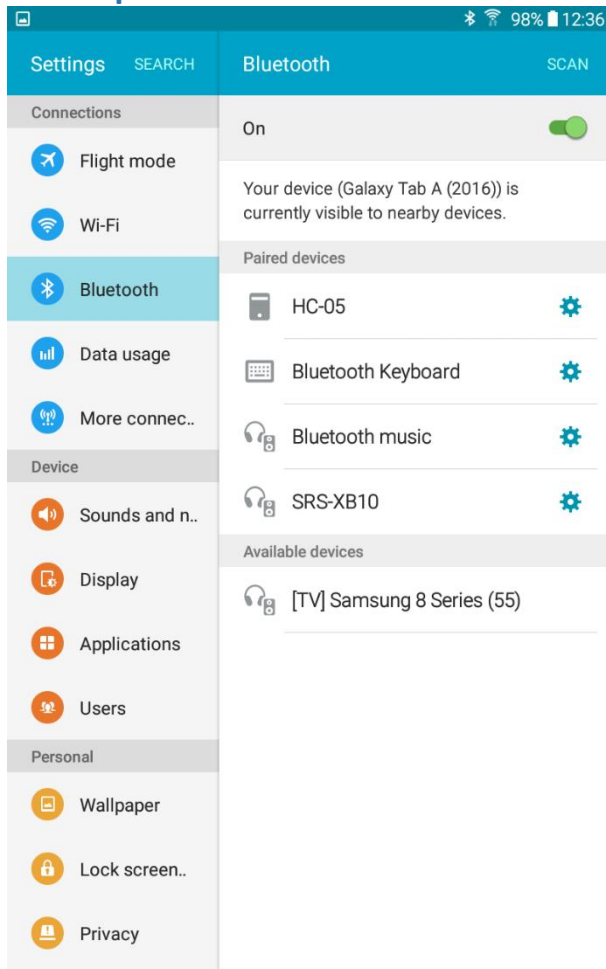
The picture shows the HC-05 as paired.

Once paired, you do not have to repeat the pairing unless you delete the HC-05 device from your Android device.

Next time, all you need to do is power on the controller, enable Bluetooth on your Android device, then run the application.

Main Activity Screen

It is best to have the HC-05 enabled controller powered on and paired with the Android device before starting the application.

The initial screen gives a number of options
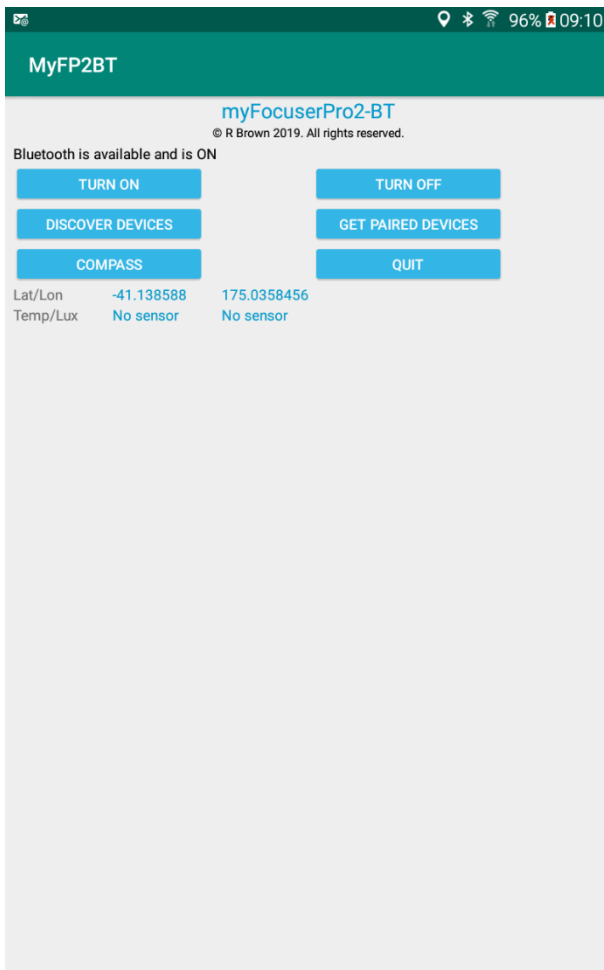
- **Turn On** provides a means to enable the Bluetooth device on the Android phone
- **Turn Off** provides a means to disable the Bluetooth device on the Android phone
- **Discover Devices** scans and displays a listed of discovered Bluetooth devices found by the Android phone
- **Get Paired Devices** lists the currently paired Bluetooth devices for the Android phone
- **Compass** opens a compass to give a bearing heading. Accuracy depends on whether the location sensor on the phone has been calibrated correctly
- **Quit** exits the application

- Latitude and Longitude are the current coordinates of the phone. If shown as 0, this indicates that the Location button has not yet been selected. The values are only updated IF the location button is selected
- Temp/Lux displays the temperature and Lux (light) values if those sensors are present

Turn On
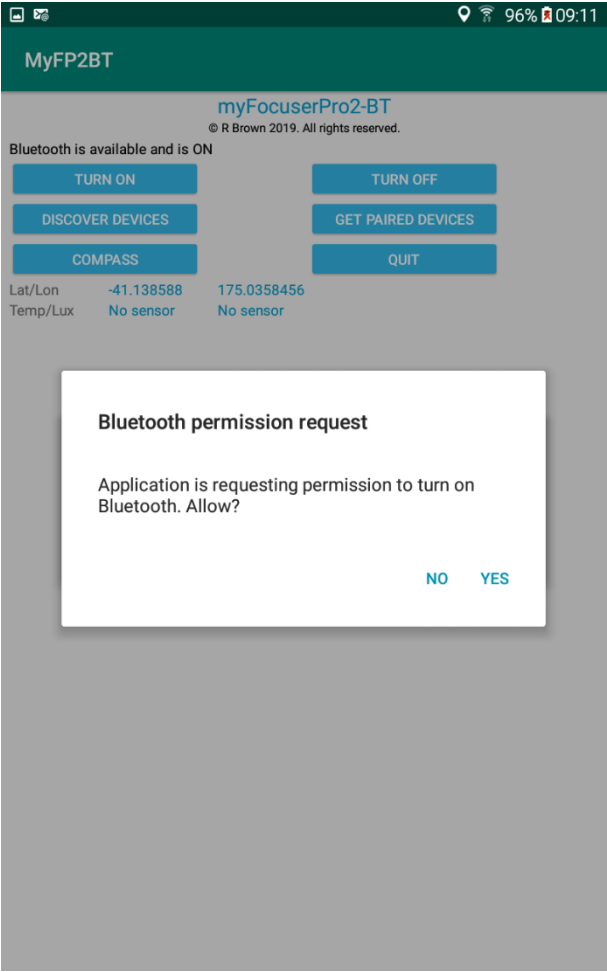
Turns on Bluetooth.

A dismiss dialog is shown.

Tap YES to enable Bluetooth.

Turn Off
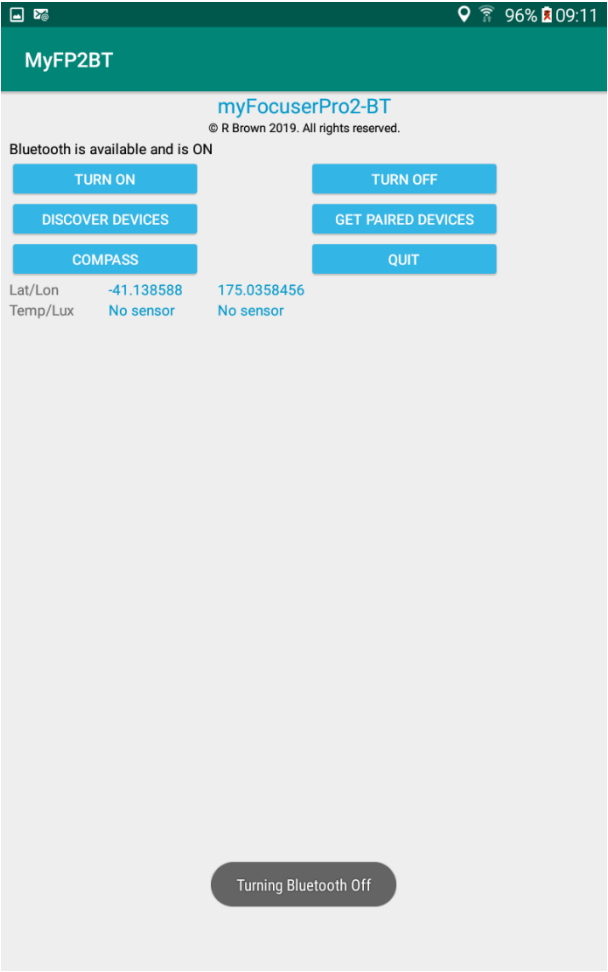
Turns off Bluetooth.

Bluetooth will be turned off.

A toast message confirms Bluetooth is being turned off.

When Bluetooth is OFF, you cannot communicate with the myFP2 Bluetooth controller.

Discover Devices
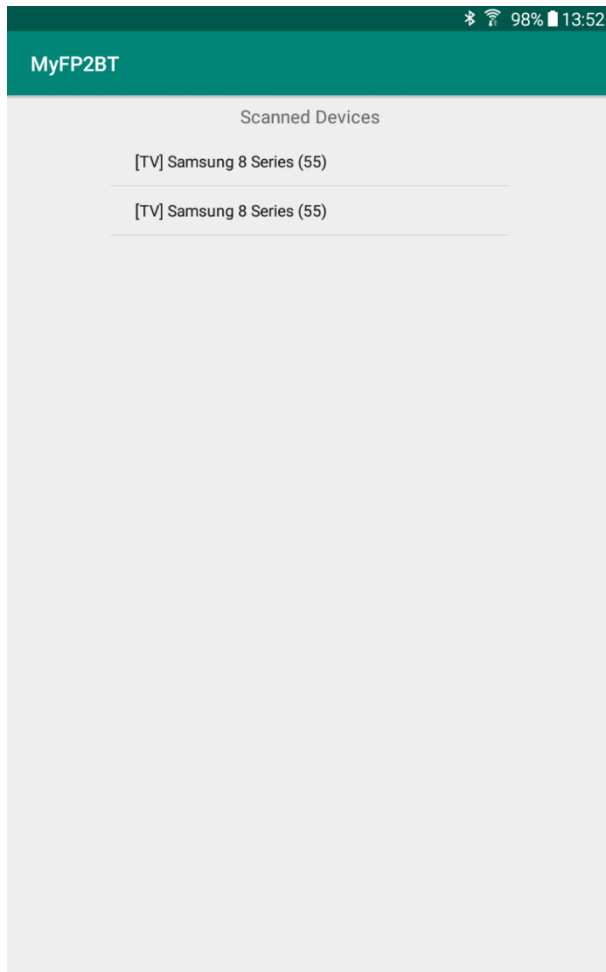
List all available Bluetooth devices that are not paired.

Bluetooth must be turned on for this to function.

Get Paired Devices
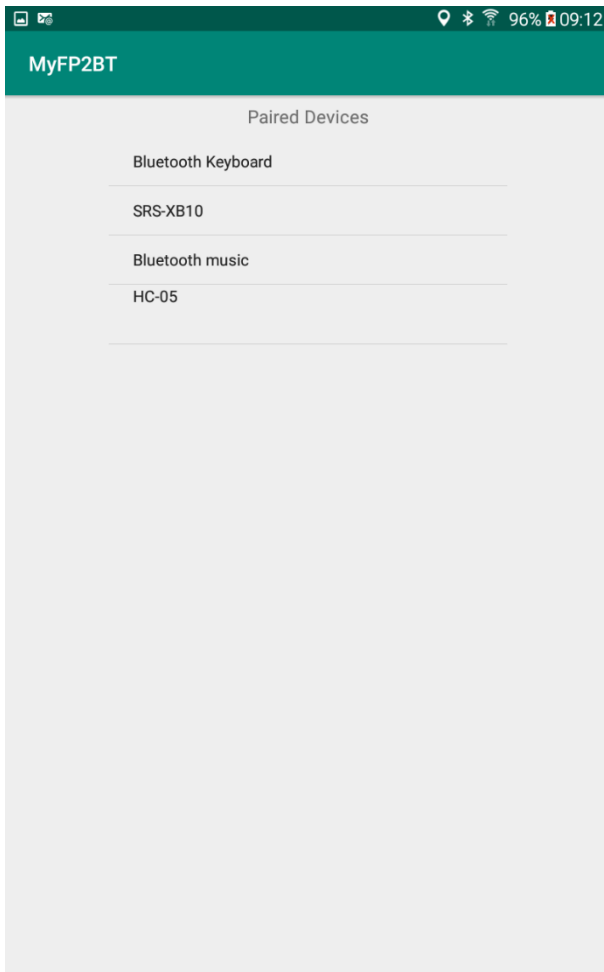
Get all paired Bluetooth devices.

The HC05, if paired, will show in this list.

To connect to the HC-05 device, tap on the item in the list.

Bluetooth must be turned on for this to function.

Compass
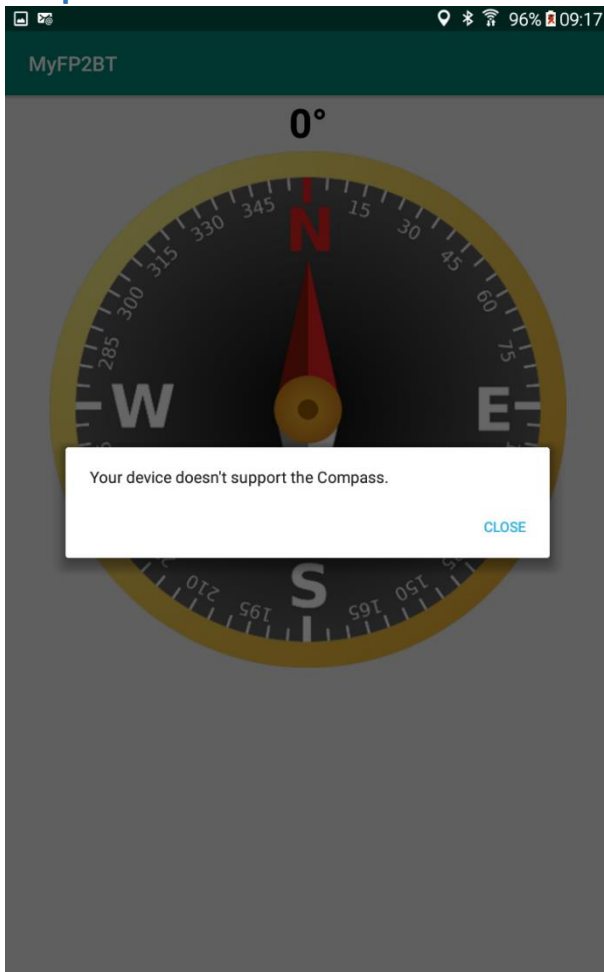
Tapping the Compass button will display a compass based on the user's current phone heading.

To return to the main screen tap the Back arrow.

The phone/tablet must have a magnetic sensor for this to function.

Quit
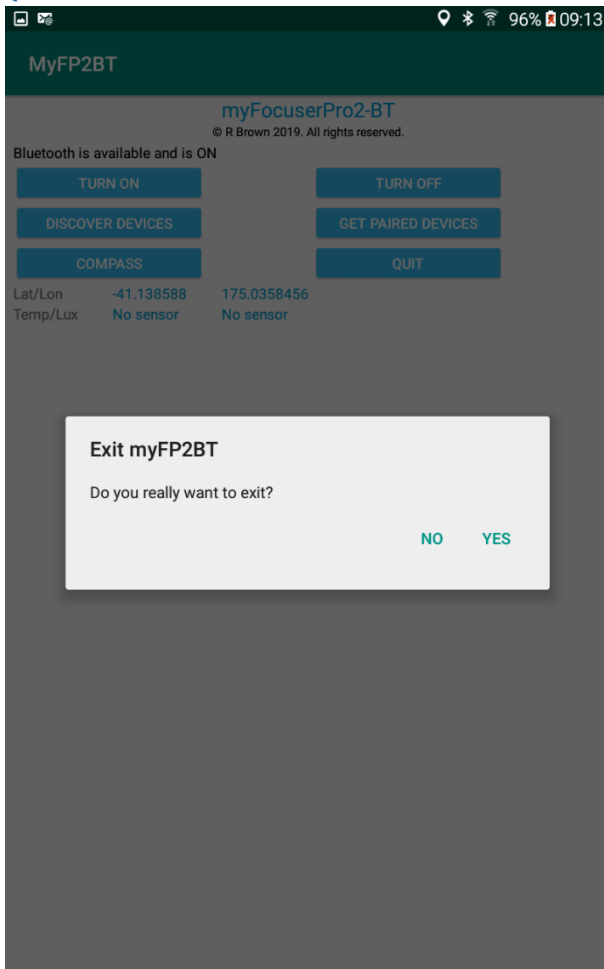
When the QUIT button is tapped, a dialog box will appear asking to quit (Yes or No).

If the user taps YES, then the application will exit.

If the user taps NO, the application will return to the Main Activity.

Connect-1

The Connect activity is displayed when the user has done Setup, followed by tapping the Connect button.

Before any commands can be sent to the controller, the user MUST tap the TEST CONNECTION button.

Once connection has been made, the values can be obtained from the controller by tapping the GET ALL button.

Connect-2

This example shows an established connection to the Bluetooth controller and the user has tapped the GET ALL button.
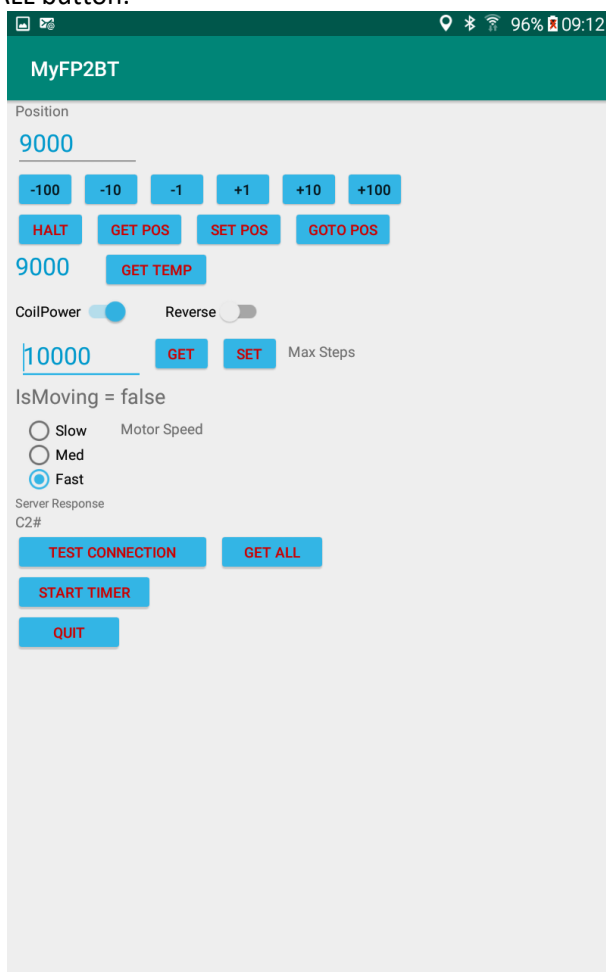
All buttons are self-explanatory. For example, to get the current controller temperature, tap the GET TEMP button. To move the focuser 100 steps out, tap the +100 button.

If the user taps in the Position field, the soft-keyboard will appear and the user can change the current position value. Once this is done, the new value can be sent to the controller by tapping the SET POS button.

If the user wants to have the values regularly updated, the START TIMER button is tapped. This starts a timer which will periodically get the position, temperature and ismoving status from the controller.

When the timer is running, the STOP TIMER button will be displayed.

To exit the Connect Activity and return to the Paired Activity tap the QUIT button.

How to Install the Android Bluetooth Application

Currently the Android BT application is an APK file that is provided for download as an unsigned app and not via Google Play.

Setting up your device

From your smartphone or tablet running Android 4.0 or higher, go to Settings, scroll down to Security, and select Unknown sources. Selecting this option will allow you to install apps outside of the Google Play store. Depending on your device, you can also choose to be warned before installing harmful apps. This can be enabled by selecting the Verify apps option in the Security settings

How to install the APK from your Android device?

Just open your browser, find the APK file you want to download, and tap it – you should then be able to see it downloading on the top bar of your device. Once it's downloaded, open Downloads, tap on the APK file, and tap Yes when prompted.

Where do I put an APK file on my Android?

To get started, download an APK file using either Google Chrome or the stock Android browser. Next, go to your app drawer and click Downloads; here you will find the file you just downloaded. Open the file and install the app. If you downloaded the APK file on your computer, the process is slightly different.

Installing an APK File from the File Manager

Download the APK file if necessary. If you haven't yet downloaded the APK file onto your Android, do the following: ...

    Open your Android's file manager app. ...
    Select your Android's default storage. ...
    Tap Download. ...
    Tap the APK file. ...
    Tap INSTALL. ...
    Tap DONE when prompted.

How to calibrate the Compass for the Samsung Galaxy S9+ Android Phone

1. When you are on the start screen, open the phone app
2. Select the dial keypad to manually enter numbers.
3. Next, type *#0*# on your Samsung Galaxy S9 or S9 Plus smartphone.
4. The service menu will open up. Select "Sensor" and scroll down to the bottom of the next menu.
5. You will see "Magnetic Sensor" and below it a circle with a line and a number behind it. This has the following meaning: Red line with a 0 = The compass sensor must be calibrated Blue line with a 3 = The magnetic sensor is calibrated 6.

To calibrate the compass, rotate the Samsung Galaxy S9 phone once around all three axes. Do this until the circle shows a blue line and a 3. Only then is your compass calibrated.

Source: https://www.solvemix.com/index.php/android-smartphones/17-year-2018/2522-samsung-galaxy-s9-google-maps-arrow-shows-wrong-direction-compass
Android Smartphones, © Solvemix - quick tips, tricks and solutions

myFP2LINUX

This application has been tested on Ubuntu-Mate. Some features present in the Windows application may not be available under Linux.
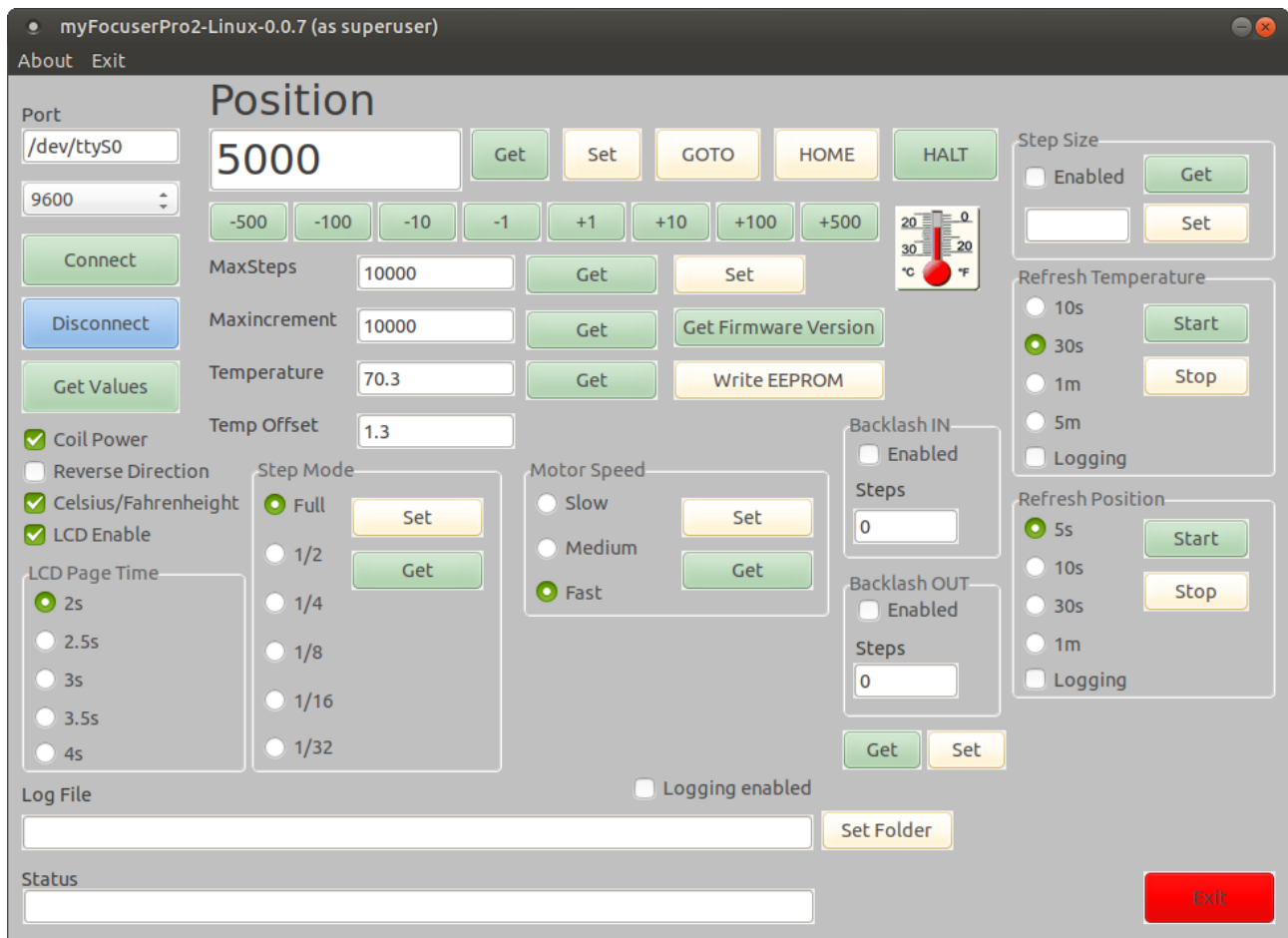
Copy the **myFocuserPro2Linux** AND **myFocuserPro2Linux.ini** files to a user folder on your Linux system (like a folder under the desktop)

==
UBUNTU-MATE
You must be a member of the group tty
Use menu, administration, users and groups, manage groups, tty, properties
and ensure that your name is checked, then click OK

chmod +x myFocuserPro2Linux
or you can right mouse click the file, properties, permissions and check allow executing file as program

I had to right mouse click file and chose run as adminstrator
It seems it cannot connect to tty0 device without admin priviledge

ASCOM

myFP2 ASCOM DRIVER

The [myFP2 ASCOM driver](#) provided comes with an installer program. The ASCOM driver works with ALL driver build options. Half steps are ignored when using the ULN2003 build option.

Most settings must be set before connecting to the controller.

When run the first time, the settings you see are default settings from the install program. They are NOT the settings of the controller.

In ASCOM, there is no way to retrieve the settings from the controller BEFORE showing the setup dialog form. The values for the settings either show the default settings from the installation, OR, if you have connected previously, the last known state of the ASCOM driver settings.

Settings are arranged as a group, for which there will be an **Update on Connect** checkbox. If this checkbox is checked, then the values in that group will be sent to the controller when connected. If this checkbox is unchecked, then the ASCOM driver will retrieve those settings from the controller and update the ASCOm settings to match, at which point they will be saved for later use.

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

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

The ASCOM DRIVER has a setup dialog form that is displayed when accessing the Drivers Properties. This form is arranged as a series of tabs, each tab is accessible by clicking on the name of tab.

Controls must be SET to their correct state BEFORE connecting to the controller, if you want to send those settings to the controller when connecting.

If you want the group of settings to be sent to the controller when connecting, the **Update on Connect** checkbox for those settings must be checked before connecting to the focuser.

ASCOM DRIVER CONNECTION TAB

The connection tab provides the main connection point for the driver to connect to the controller.
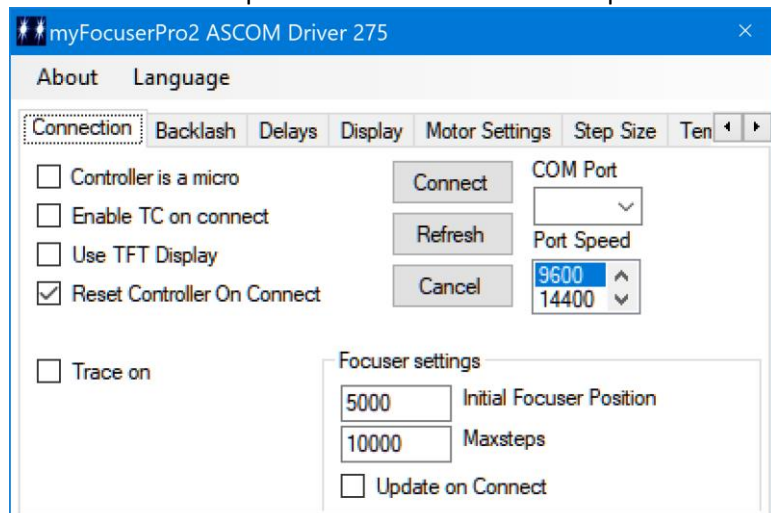
Controller is a Micro

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

Enable TC on connection

When checked, temperature compensation will be turned on in the controller once a connection is established.

Use TFT Display

Enable this if the controller is fitted with a TFT display

Reset Controller on Connect

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

Trace On

Debug messages are written to a trace file

Com Port and Com Port Speed

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

FocuserSettings Initial Position

This setting specifies the initial position of the focuser. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

FocuserSettings MaxSteps

This setting specifies the maximum number of steps for the focus controller and will vary depending upon each user's configuration. You should set this to the maximum number of steps that you have calculated for your focuser [see determining maxStep]. This setting must be specified before connecting to the controller and will be remembered by the application. See the section on the initial setup of the focuser for further information. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

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

CONNECT
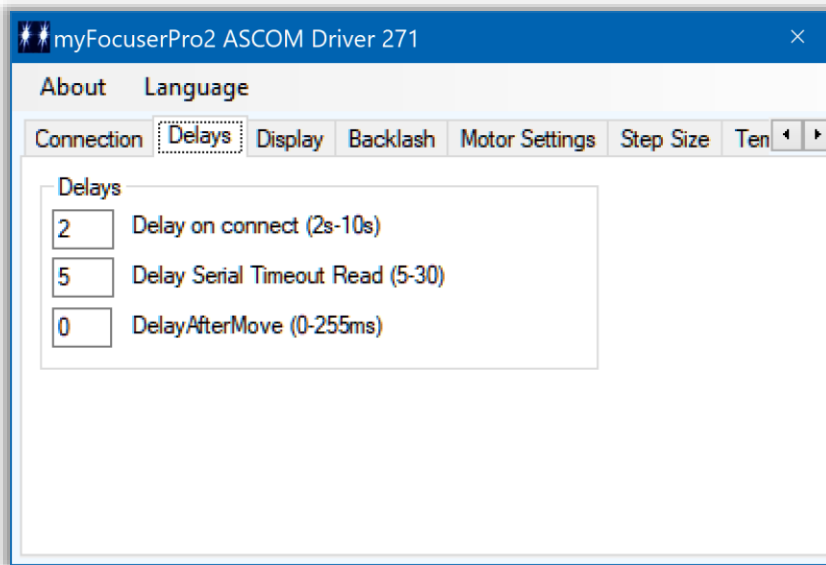When all the settings have been set to their desired states, click the CONNECT button.

REFRESH
The refresh button updates the list of com ports shown in the drop-down list of available com ports.

CANCEL
Aborts the setup dialog form

ASCOM DRIVER DELAYS TAB

Delay (s) 2-10 on Connect
Specify the delay in seconds after connecting that the driver will wait before sending a command to the myFP2 controller. Valid values are 2-10 seconds. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

Delay (s) 5-30 timeout on Serial Read
Specify the delay in seconds that the application will wait when attempting to read from the serial port after sending a command to the myFP2 controller (default = 5). For Bluetooth or slower devices, a value of 8 or 10 may suffice. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

DelayAfterMove (0-255)
Specifies the delay in milliseconds that is applied after move to prevent any blurring of star caused by any mechanical residual movement or vibration of the stepper or focuser during focusing. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*
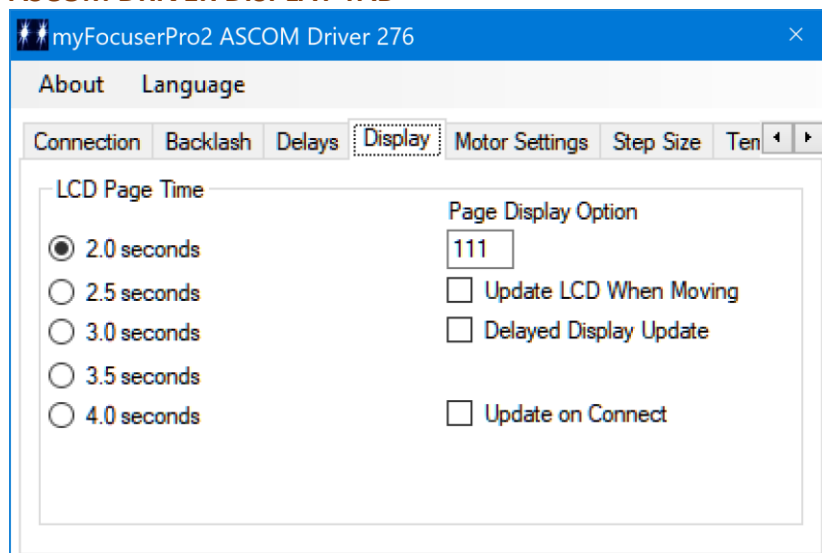
ASCOM DRIVER DISPLAY TAB

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

Page Display Option
This option provides a means of enabling or disabling each display page. The text box accepts a string of binary digits [0 or 1]. If the bit=1 the page is shown; if the bit=0 the page is not displayed.

The least significant bit = page 0

Example
I want pages 0,1 showing I would write
0011

then click the SET button. The default is 11 – only the first two pages are shown.

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

Delayed Display Update
This is most useful when using the Push Buttons to move the motor. When checked, this option delays updating the display on the controller till after the move has ended.

Update on Connect
If checked, the ASCOM driver will send the LCD Page Time and Update LCD When Moving settings to the controller when connecting. If unchecked, the LCD Page Time and Update LCD When Moving settings will be retrieved from the controller when connecting and saved for future use.

ASCOM DRIVER BACKLASH TAB

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

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

IN Steps (0-255)
The number of steps of backlash to apply after the IN move has completed. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

OUT Steps (0-255)
The number of steps of backlash to apply after the OUT move has completed. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*
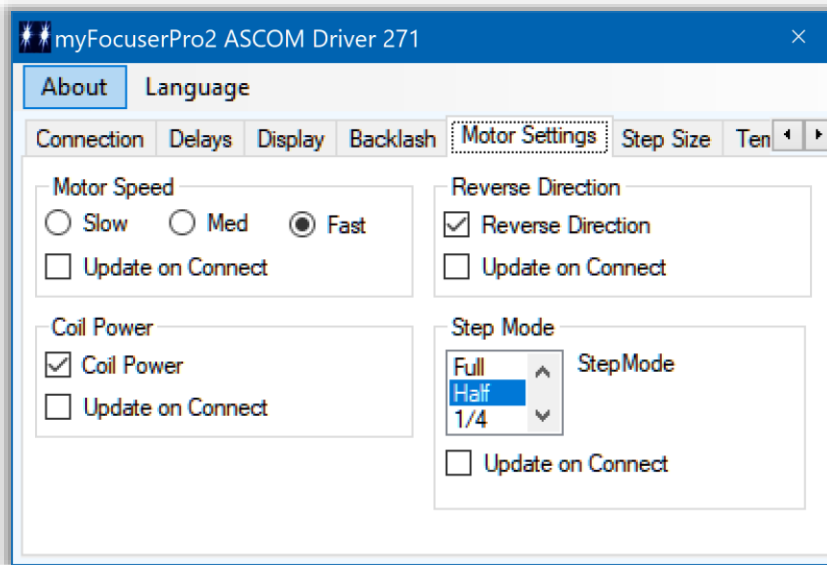
Update on Connect
If checked, the ASCOM driver will send the Backlash Settings to the controller when connecting. If unchecked, Backlash settings will be retrieved from the controller when connecting and saved for future use.

ASCOM DRIVER MOTOR SETTINGS TAB

Motor Speed
The speed of the stepper (delay between steps) can be adjusted in the range Slow-Medium-Fast. The speed setting is remembered by the controller. If the Update on Connect checkbox is checked, then the value of motor speed will be sent to the controller when connecting. If the checkbox is not checked, then the controller will use the last remembered motor speed setting, and the ASCOM driver will retrieve this value when connecting to the controller and save it for future use.

Coil Power
When enabled (ticked), indicates that coil power is ON and the stepper coils are powered after the move is completed. If the Update on Connect checkbox is checked, then the value of Coil Power will be sent to the controller when connecting. If the checkbox is not checked, then the ASCOM driver will retrieve this value when connecting to the controller and save it for future use.

Reverse Direction
When enabled (ticked), indicates that the motor moves in the opposite direction (IN means OUT and OUT means IN). If the Update on Connect checkbox is checked, then the value Reverse Direction will be sent to the controller when connecting. If the checkbox is not checked, then the ASCOM driver will retrieve this value when connecting to the controller and save it for future use.

Step Mode
The desired micro stepping mode.

Update on Connect
If the Update on Connect checkbox is checked, then the value Step Mode will be sent to the controller when connecting. If the checkbox is not checked, then the ASCOM driver will retrieve this value when connecting to the controller and save it for future use.
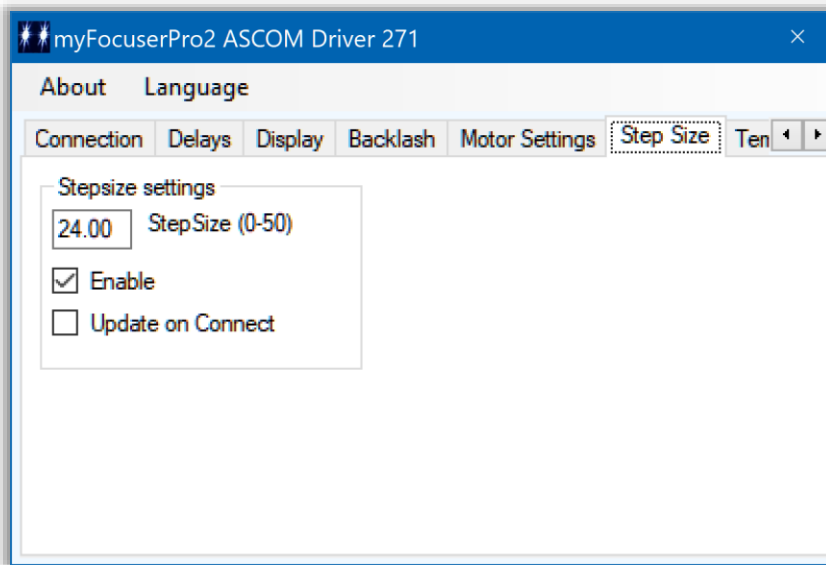
ASCOM DRIVER STEPSIZE TAB

Step Size in microns (0-50)

This setting defines the Step Size in microns. You can enter the calculated value for your focuser and this will be sent to the myFP2 controller when connecting. This allows client applications using the ASCOM driver to retrieve the Step Size setting from the myFP2 controller. Please see the section in this PDF on determining your focuser step size value. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

Enable

This setting turns ON or OFF the reporting of the StepSize value from the myFP2 controller. If you do not know your step size value, DO NOT enable this feature.

Update on Connect

If checked, the ASCOM driver will send the Step Size Settings to the controller when connecting. If unchecked, Backlash settings will be retrieved from the controller when connecting and saved for future use.

ASCOM DRIVER TEMPERATURE TAB
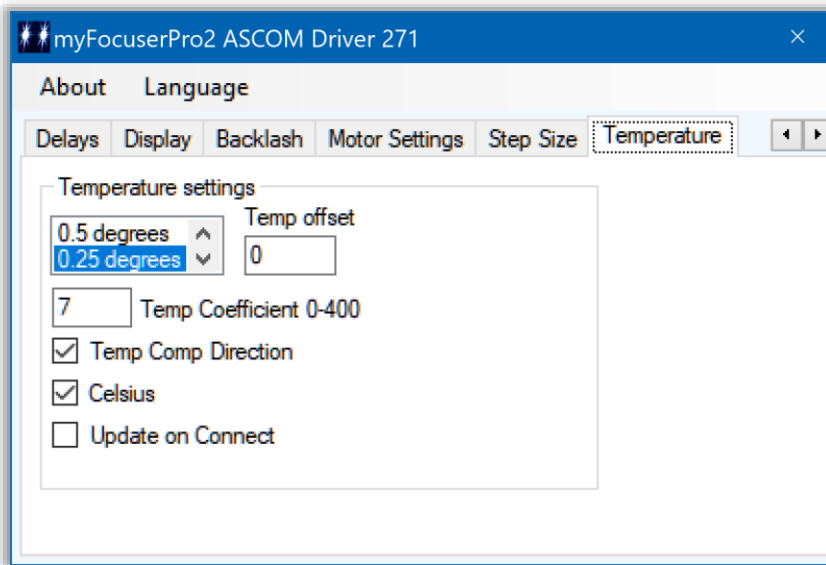
Temperature Resolution
Allows the user to control the resolution of the DS18B20 temperature probe from 0.5 degrees down to 0.0625 degrees (9=0.5, 10=0.25, 11=0.125, and 11=0.0625). This setting is remembered by the myFP2 controller.

Temp Offset
The value to subtract from the temperature reading returned from the myFP2 controller before returning it to the requesting application. Adjustment values range from –3 to +3. For example, typing -1.5 into the entry-box will subtract 1.5 degrees C from the returned temperature value. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

Temperature Coefficient (0-400
The value in steps per degree for temperature compensation, and is specific to each focuser and must be calculated by the user. The myFP2 Windows application has special support for calculating this value. *After typing the new value in the field, you must use the ENTER key to ensure the value is validated. If the value you entered is invalid, then the field is updated with a default value.*

Temperature Compensation Direction
Use this setting to set the direction in which temperature compensation moves are applied. Checked means moves are make Inwards (normal).

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

Update on Connect
If checked, the ASCOM driver will send the Temnperature Settings to the controller when connecting. If unchecked, Backlash settings will be retrieved from the controller when connecting and saved for future use.

ASCOM DRIVER CONFORM VALIDATION

Every release of the myFP2 ASCOM drivers are tested against the CONFORM utility. This program tests to ensure that the functions/methods/properties provided by the ASCOM driver are compliant to the ASCOM standard.

All myFP2 drivers are ASCOM compliant and the validation reports are available on the myFP2 site.

Temperature Compensation and the ASCOM driver

An **ASCOM client** has the ability to send a command to an ASCOM driver and enable/disable temperature compensation. The myFP2 ASCOM driver and firmware supports temperature compensation.

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

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

Here is the part of the ASCOM Conform report for the myFP2 related to temp compensation

```
17:30:58.416 TempCompAvailable       OK       True
17:30:58.456 TempComp Read           OK       False
17:30:58.465 TempComp Write          OK       Successfully turned temperature compensation on
17:30:58.473 TempComp Write          OK       Successfully turned temperature compensation off
17:31:06.998 Move - TempComp True             Moving to position: 61748
17:31:07.005 Move - TempComp True     OK       .NET InvalidOperation Exception correctly raised as expected
Conformance test complete
No errors, warnings or issues found: your driver passes ASCOM validation!!
```
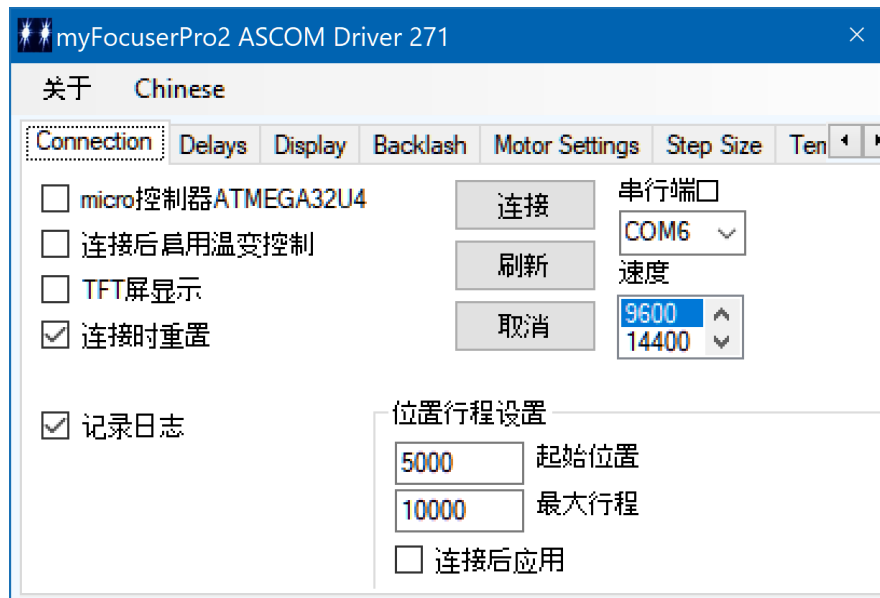
If the client application implements its own temperature compensation support, please disable it. Do not enable the client support if temperature compensation is implemented in the controller.

myFP2ASCOM LANGUAGE SUPPORT

Support is provided for multiple languages. Translation has been done by Google translate. Buttons, Labels, menu items and Hover messages are all translated. If the menu option Language, On Load, Enable is checked then the ASCOM driver will remember the language setting and reapply the setting each time the ASCOM driver is loaded.

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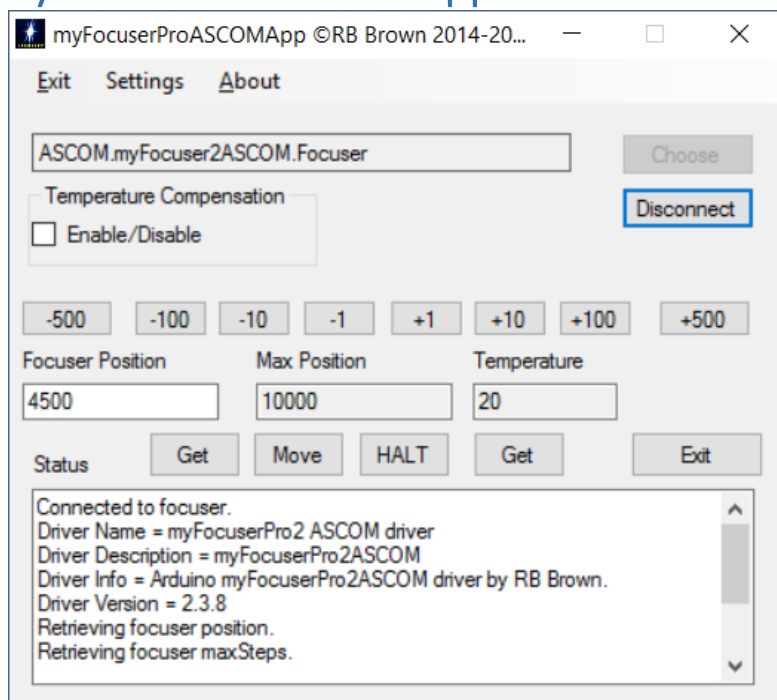
The myFocuserASCOMAPP is a Windows software application that is actually an ASCOM client and will talk to any ASCOM focuser driver and allow you to control the focuser.

The settings menu has a number of options, similar to those provided by the Windows Application and the Windows Mini Application for controlling the myFP2 controller.

This is a client program and you can use it to test any of the features of the myFP2 ASCOM driver, including temperature compensation.

HOW TO RUN TWO FOCUSERS

What happens if you have two telescopes and want to have two myFP2's, one on each scope?

There are some simple rules.
1. You can only run one instance of the myFP2 Windows application at any time
2. You can only run once instance of the myFP2Mini application at any time
3. You can only run one instance of the ASCOM driver at any one time (but you can run both the ASCOM and ASCOM1 drivers at the same time)

Here are some options for running two focusers

Scope	Option 1	Option 2	Option 3
Scope 1: Orion ED80	myFP2 App	myFP2 App	ASCOM driver
Scope 2: SV102T	MyFP2Mini App	ASCOM Driver	ASCOM1 driver

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

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

INDI

Thanks to Alan Townshend myFP2 is supported by INDI.

https://indilib.org/devices/focusers/myfocuser-pro-2.html

Current features are:
- Sync
- Absolute and relative position
- Preset positions
- Temperature compensation
- Coil power control
- Display on or off
- Reverse direction
- Joystick control
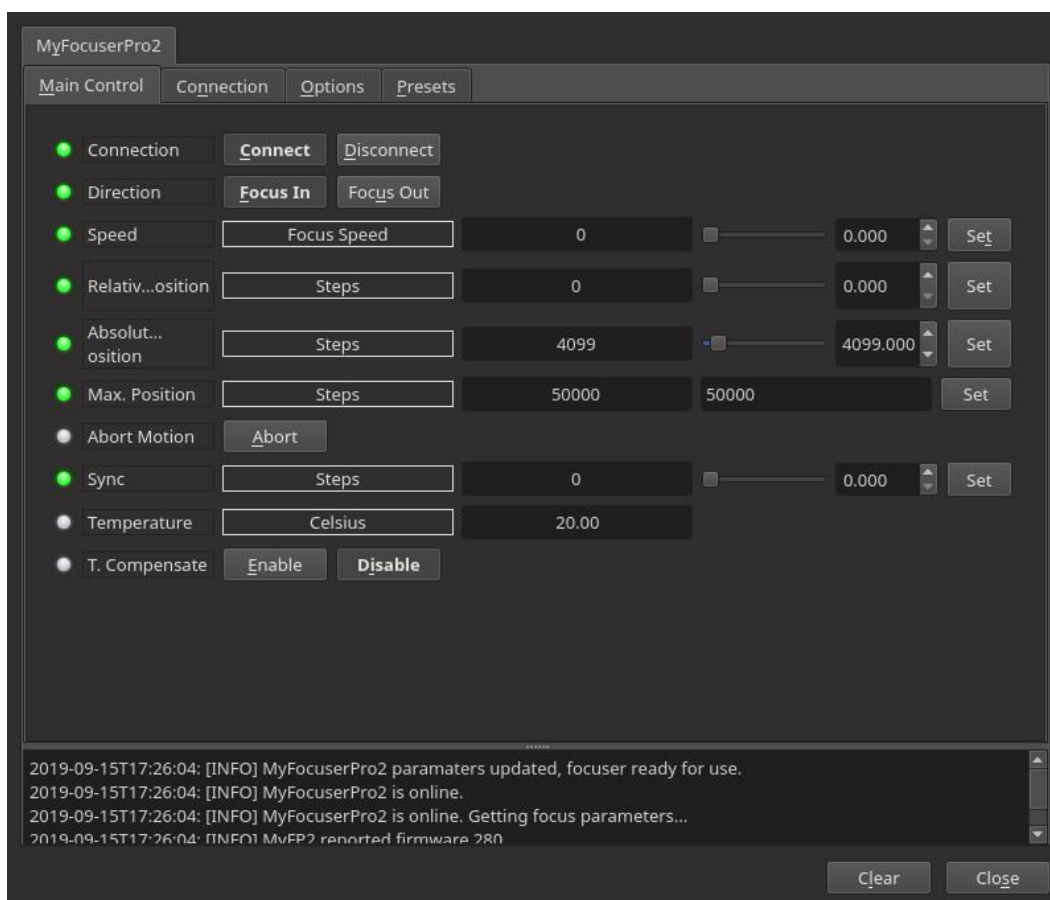
INSTALLATION

The INDI myFocuserPro2 driver is included with libindi >= 1.8.2 Under Ubuntu, you can install the driver via:

> sudo add-apt-repository ppa:mutlaqja/ppa
> sudo apt-get update
> sudo apt-get install indi-bin

TEMPERATURE COMPENSATION

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

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

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

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

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

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

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

The following pages outline how to do this with the myFP2 controller and application software.

Step 0 Read the Instructions
The following assumptions are
1. The temperature probe is located near the optics of the telescope (for a refractor this would be near the front lens cell)
2. The telescope has reached thermal equilibrium. This may take up to 30 minutes or more to occur and varies depending on the conditions and telescope type and size
3. The myFP2 controller is connected and the Windows Application is running
4. The telescope is at optimal focus and tracking the target star

The first step is to access the **Read Instructions** option from the Temperature Compensation menu of the myFP2 windows application. Select the Read Instructions option
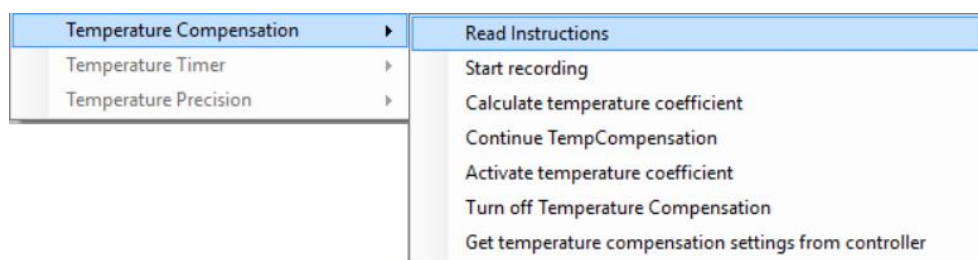
This displays the following MessageBox of the steps you must follow to determine and apply a temperature coefficient value and enable temperature compensation for your myFP2 controller. The steps are performed one after the other in order. If you make a mistake, simply start again from Step 1.
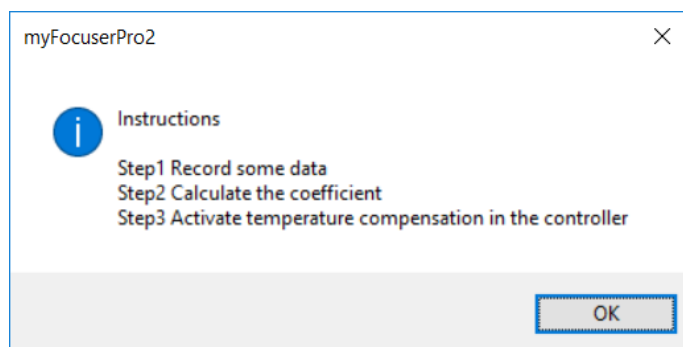
Step 1 Start Recording

The next step is to access the **recording** option from the Temperature Compensation menu of the myFP2 windows application. Select **Start recording**
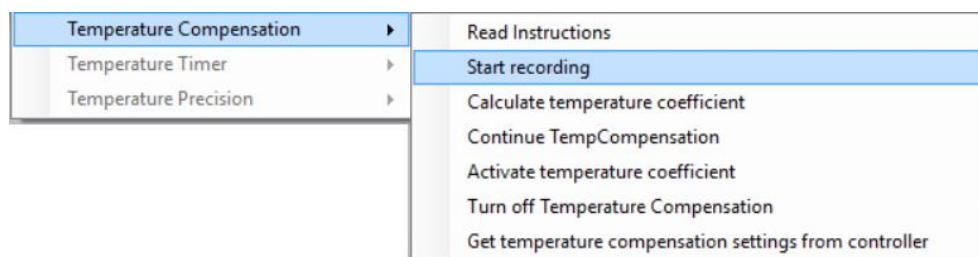
The following dialog box appears,
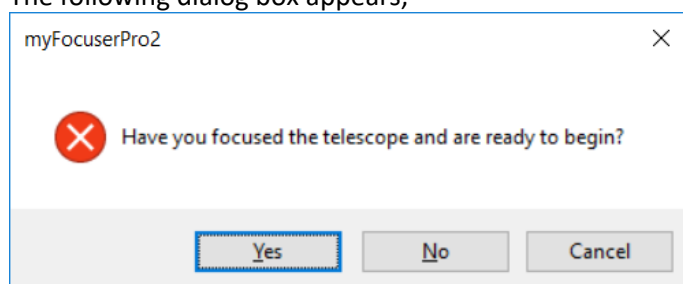
If you have already focused the telescope within the last few minutes, click **Yes**, else click No and refocus the telescope before starting again.

After clicking **Yes**, the following dialog box appears,
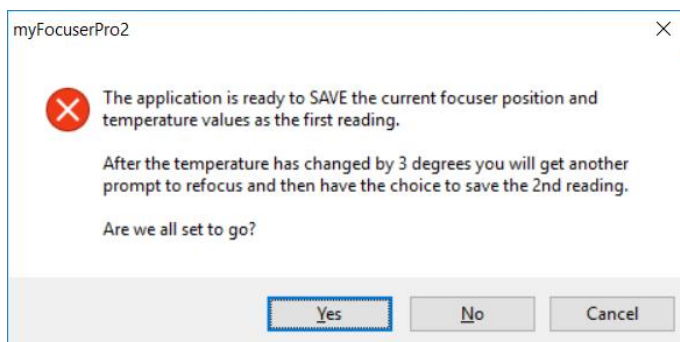
When you click **Yes**, you will be automatically taken to the record dialog option and the following MessageBox appears.
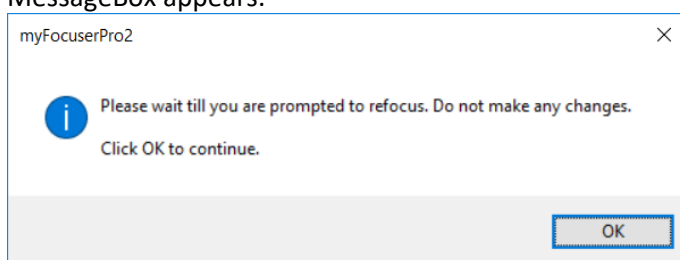
At this point, the program has automatically
- saved the current temperature and focuser position
- started the auto refresh timer for the temperature updates and enabled it to a 10s refresh cycle
- started to monitor the temperature change

Just wait for the application to monitor the temperature and prompt you when ready. During this time interval DO NOT MAKE ANY CHANGES TO ANY SETTINGS. You will be able to see the progress in the Status messages textbox, as indicated below
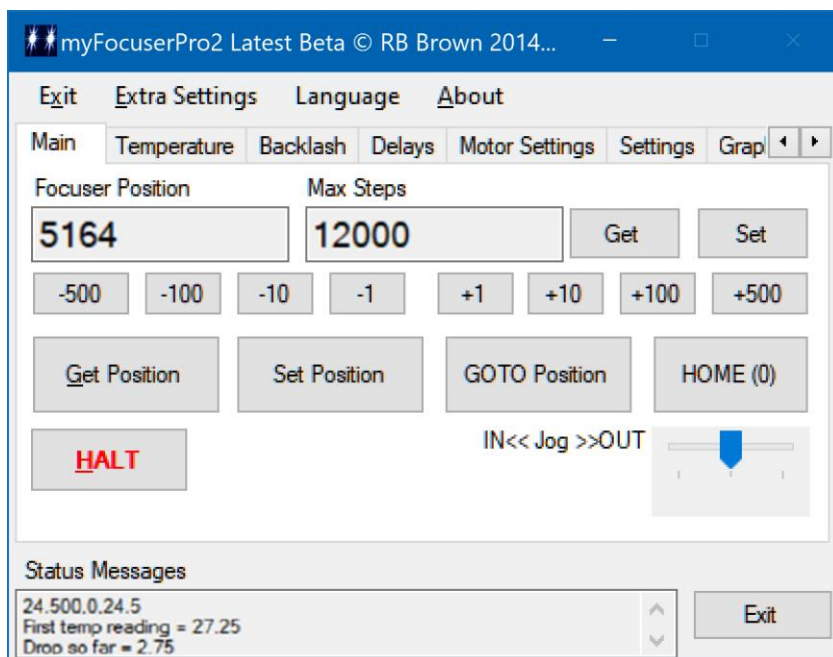
Once the program detects a 3 degree change in temperature, a new dialog MessageBox will automatically appear asking you to refocus the telescope. Click **OK** and then refocus the telescope.
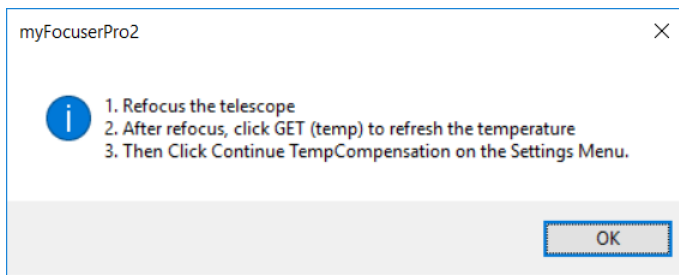
After refocusing, get the latest temperature reading by clicking the GET button for temperature. Once the new temperature value is displayed, click the **Continue TempCompensation** menu option on the settings menu under Temperature Compensation.

The application will automatically calculate the temperature coefficient and display the value in a MessageBox (example below) and include an option for you to now update the controller and enable temperature compensation.
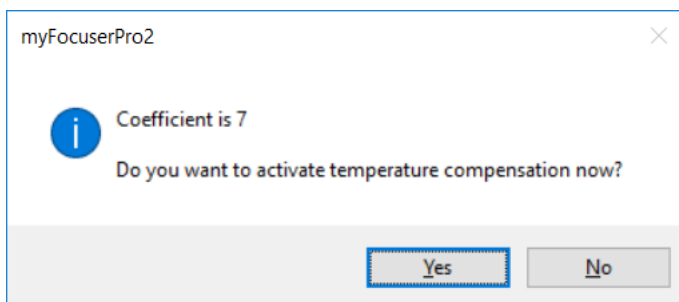
Click **YES** to Goto the next step. The application will now display the Apply Coefficient Form as shown below.
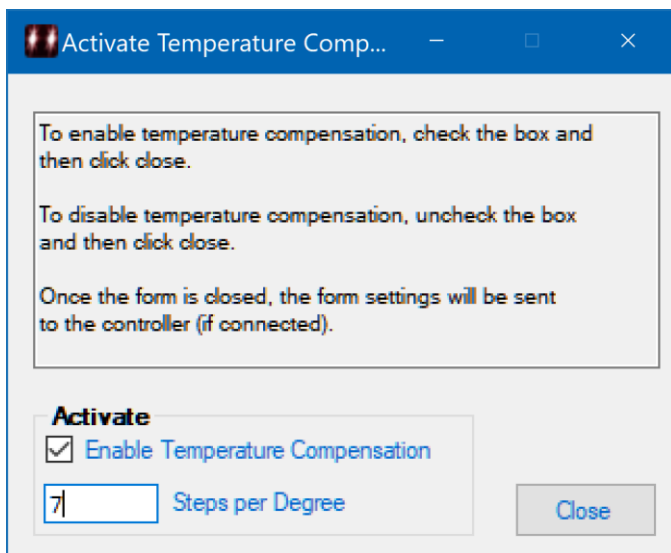
The Steps per Degree is automatically preloaded from the previous calculation. To send this value to the controller, and enable temperature compensation, check the box **Enable Temperature Compensation** and then click the **Close** button.

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

The values are sent to the myFP2 controller and temperature compensation is enabled. The push buttons are disabled when temperature compensation is enabled in the controller.

Display the Current myFP2 Controller Temperature Compensation Settings

To display the current temperature compensation settings, select the **Get temperature compensation settings from controller** option
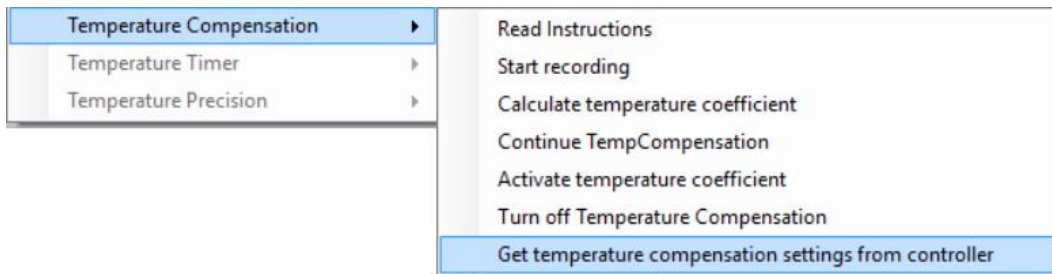
The application will query the controller and display the current settings in the status area.
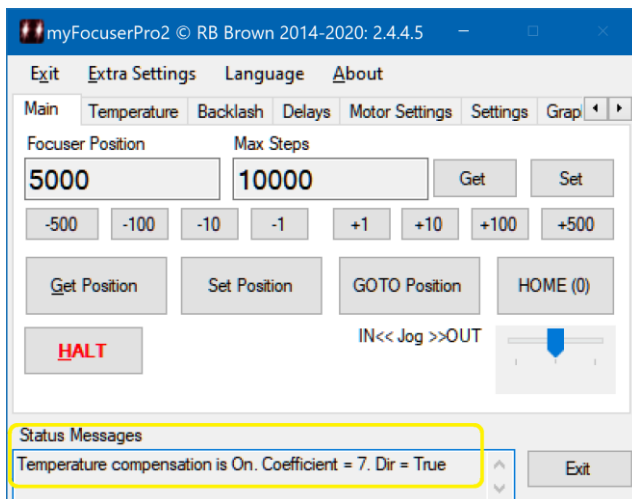
The Windows Application and ASCOM driver support temperature compensation. However, the controller handles the changes internally when enabled.

AT THE BEGINNING OF THE EVENING SHOULD YOU FOCUS AT LEAST TWO OR THREE TIMES WHEN THE TEMPERATURE CHANGES?

That would depend upon your temperature coefficient setting compared to step-size.
If the temperature coefficient setting was 28 steps, and your step size was 4.2, then this means that for every degree of temperature change, the focuser must move by 28 steps to compensate. This pdf explains how to calculate step size, critical focus zone and temperature coefficient steps.

If the temperature drops by 1 degree, this means we need to move the focuser 28 steps to stay in focus [the temperature coefficient].

The step size is 4.2 and we have 10 steps in the critical focus zone. So that means the focuser has 10 steps where it is in focus. But we need to move 28 steps. which is way outside the critical focus zone and so the image is out of focus and we must move the focuser to stay in focus.

This would mean adjusting the focus every time there is a temperature change of 1 degree.

If you are an imaging taking photos, then you need to
1. wait till thermal equilibrium of the scope has occurred
2. determine your coefficient setting [pdf explains how]
3. enable step size in the focuser and set the step size value
4. set the temp coefficient value in the focuser
5. before doing an imaging run, do an auto focus
6. enable temperature compensation
7. start the imaging sequence

TEMPERATURE COMPENSATION DIRECTION

If you find your images are slowing becoming out of focus as the imaging run capture sequence advances, this means the temperature compensation direction value is set wrong and should be changed to the other direction.

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

EEPROM USAGE

The myFP2 controller remembers important focuser settings 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.

Whenever you make an important change, the controller will wait 30s before saving the value in EEPROM. You have to wait 30s before disconnecting and reconnecting or power off.

If you cannot wait 30s, then use the settings menu - option is - Write focuser settings to EEPROM - this does the save immediately.

The myFP2 controller code, on power up or reset, restores the focuser settings from EEPROM.

You need an **ATMEGA328** Nano chip which has 1024 bytes of EEPROM

You need to copy the library folders as outlined in the instructions.

It is very unlikely that the contents of the EEPROM will wear out even after years of use.

WHAT TO DO IF YOU LOSE YOUR FOCUSER SETTINGS

1. Unclamp the focuser coupler connecting the stepper motor to the focuser (or remove belt if using a pulley system)
2. Manually move focuser to initial 0 position (1/2 turn out as described above)
3. Power focuser and start myFP2 Windows application software (not ASCOM driver)
4. Set the step mode to what you used in the initial setup of the controller
5. Enter 0 as the focuser position and click the SET POSITION button
6. Clamp the focuser coupler so that the focuser motor can now drive the focuser
7. Enter your determined maxStep value into the Maximum Position text box and click the Set Button
8. The focuser is now setup. Enter the position for reasonable focus into the Focuser Position text box and click the GOTO POSITION button to move the focuser to the focus position

UPGRADING FIRMWARE AND SOFTWARE – METHOD 1

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

New releases often occur after you have built and set-up your myFP2 controller.

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

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

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

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

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

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

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

STEP 1: SAVE SETTINGS

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

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

STEP 2: UPDATE FIRMWARE OR APPLICATION

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

STEP3: RESTORE SETTINGS

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

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

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

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

STEP 1: SAVE SETTINGS

UPGRADING FIRMWARE AND SOFTWARE – METHOD 2

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

What to do first before upgrading

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

The important settings are (some of these are recent so might not be on your system)
- Backlash settings
- Focuser Position
- Max Steps
- Coil Power
- Reverse Direction
- Motor speed
- Step mode
- Temp offset
- Step Size
- Temperature Precision
- Temperature Compensation value (if you have undertaken to calculate this)
- Refresh rate of display – how long each LCD page is displayed
- Backlash

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

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

Download the required files

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

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

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

Update the controller firmware

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

If you built the myFP2 controller with the power-on reset circuitry, remember to slide the switch into the program position first before turning on power to the controller.
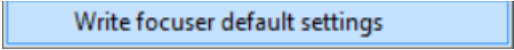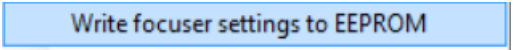
What I do is just remove the controller from the telescope and then reprogram the controller on the work-bench. Once you have finished re-programming the controller, remember to slide the power-on reset switch back to its normal setting (if fitted).

Update the Windows application and ASCOM driver

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

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

Run the Windows Application and set the controller default settings

1. Connect to the controller
2. Select this option [Write focuser default settings] from the Settings menu
3. Disconnect
4. Wait 10s
5. Reconnect to the controller
6. Now set all the focuser settings as per the settings you wrote down earlier (such as focuser position, maxSteps, stepmode etc)
7. Select this option [Write focuser settings to EEPROM] from the Settings menu to write these new settings to the controller

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

IT IS IMPORTANT THAT TO REALIZE THAT THE ARDUINO FIRMWARE AND WINDOWS APPLICATIONS OFTEN INTRODUCE NEW FEATURES AT THE SAME TIME. WHAT THIS MEANS IS YOU MUST UPDATE THE FIRMWARE AS WELL AS THE WINDOWS APPLICATION AND ASCOM DRIVER TOGETHER. YOU CANNOT RUN THE LATEST WINDOWS APPLCIATION OR ASCOM DRIVER ON ARDUINO FIRMWARE THAT MAY BE SEVERAL VERSIONS EARLIER.

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

BRACKET MOUNTING IDEAS

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

There are lots of pictures from contributors in how to mount the stepper motor on various telescopes. These are is in the folder **Mounting Bracket Examples** included with the firmware zip file. Included are some 3D printer bracket designs.
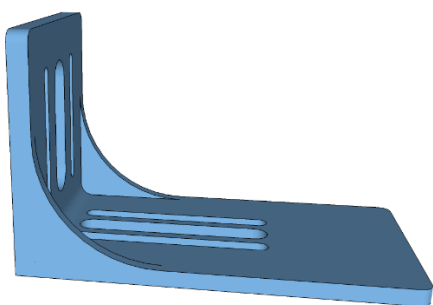
NEMA14HM11-0404S-Bracket
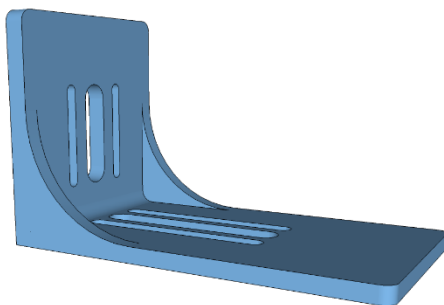
NEMA17-17HS13-0404S-PG5-PG27-Bracket
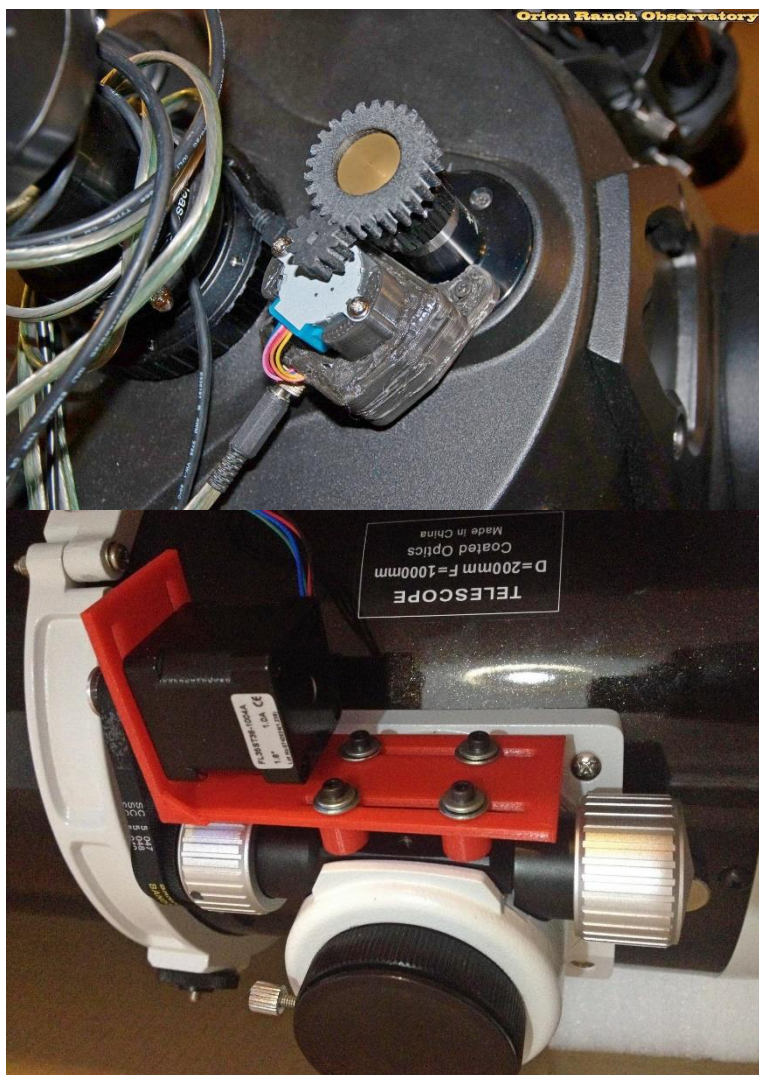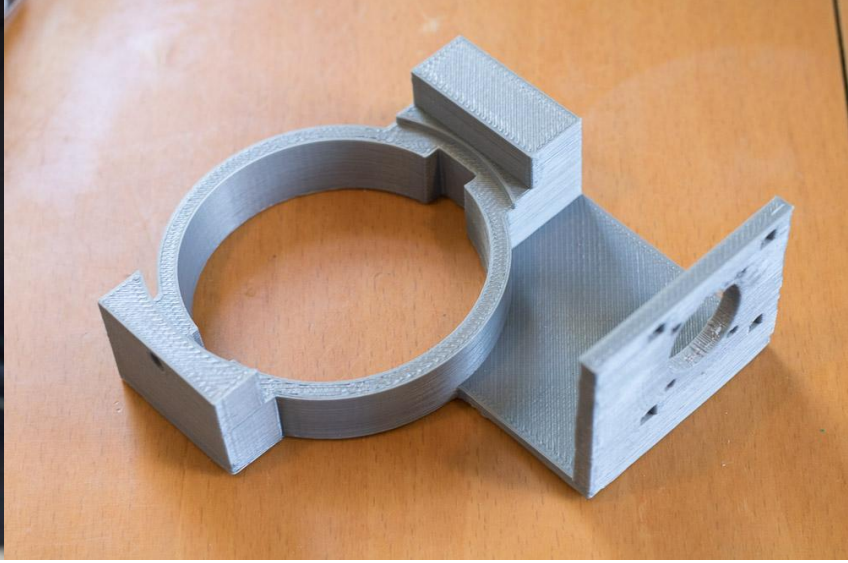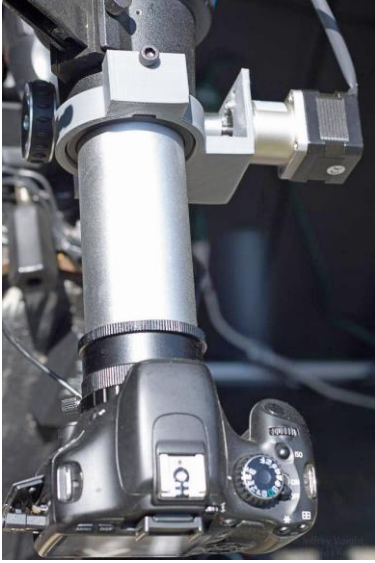
RECOMMENDED PCB BUILD IS NANO+DRV8825 HW203 BOARD
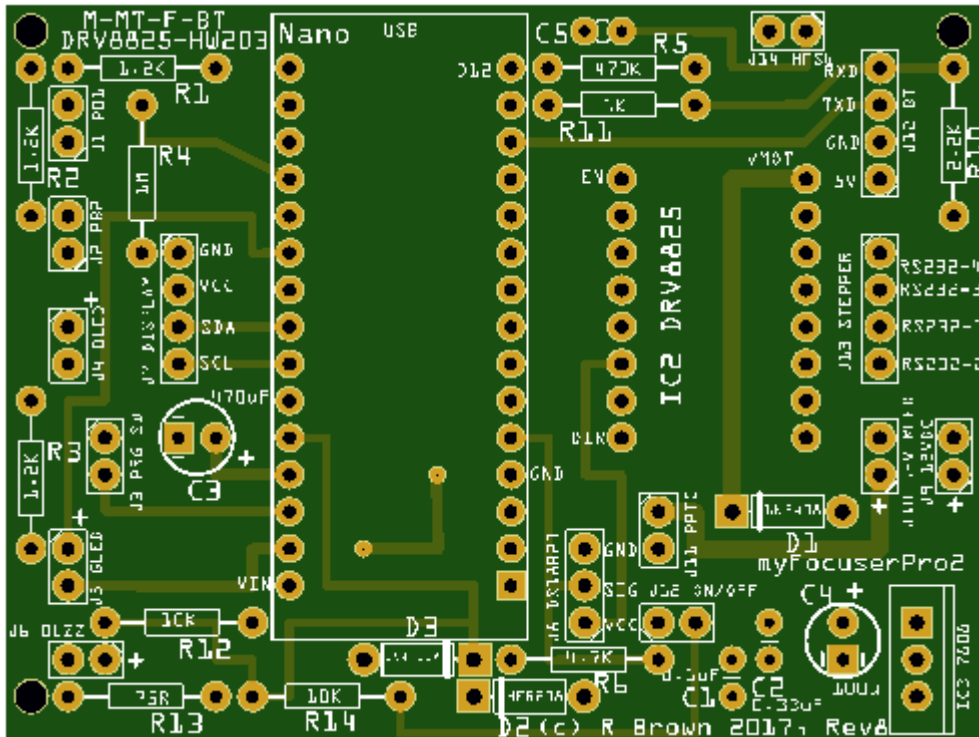
RECOMMENDED STRIPBOARD BUILD IS NANO+DRV8825 HW203 BOARD
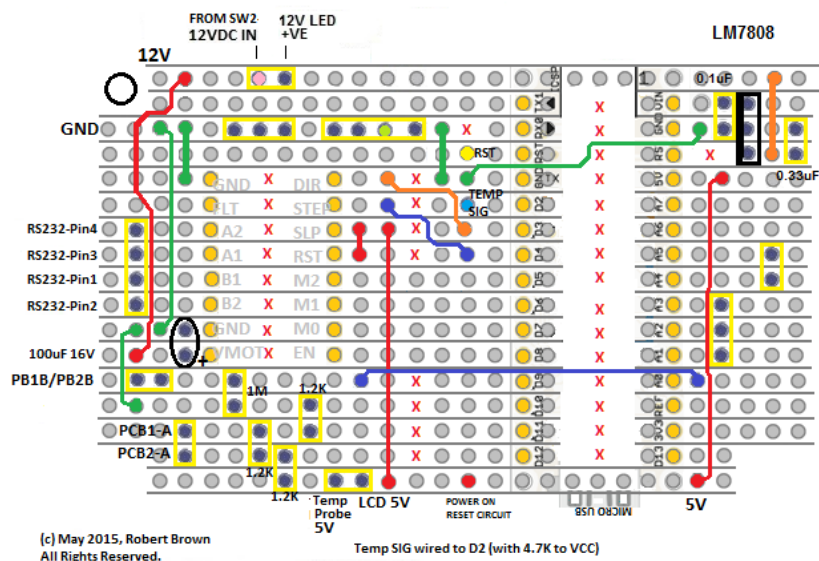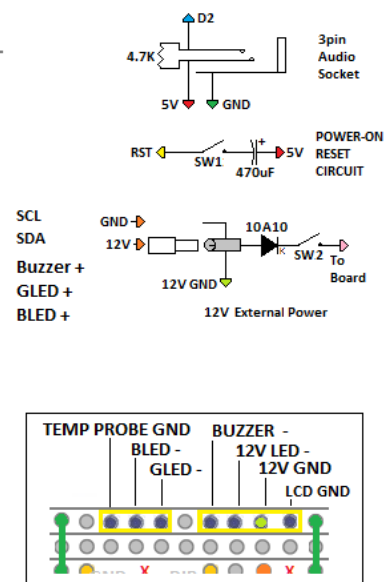
Note: Stripboard layouts are found in the STRIPBOARDS folder.

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 myFP2 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

Bahtinov Mask Focusing using Bahtinov Grabber And Nebulosity

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

Automated Focusing

Right now, out of the box, you can do automated focusing on stars with any myFP2 controller fitted to a telescope. You can use the controller with MaximDL, APT and Scopefocus. You could also use FocusMax v3 which was the last free version of Focusmax before it became commercial. We recommend FocusMax v4.

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

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

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

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

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

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

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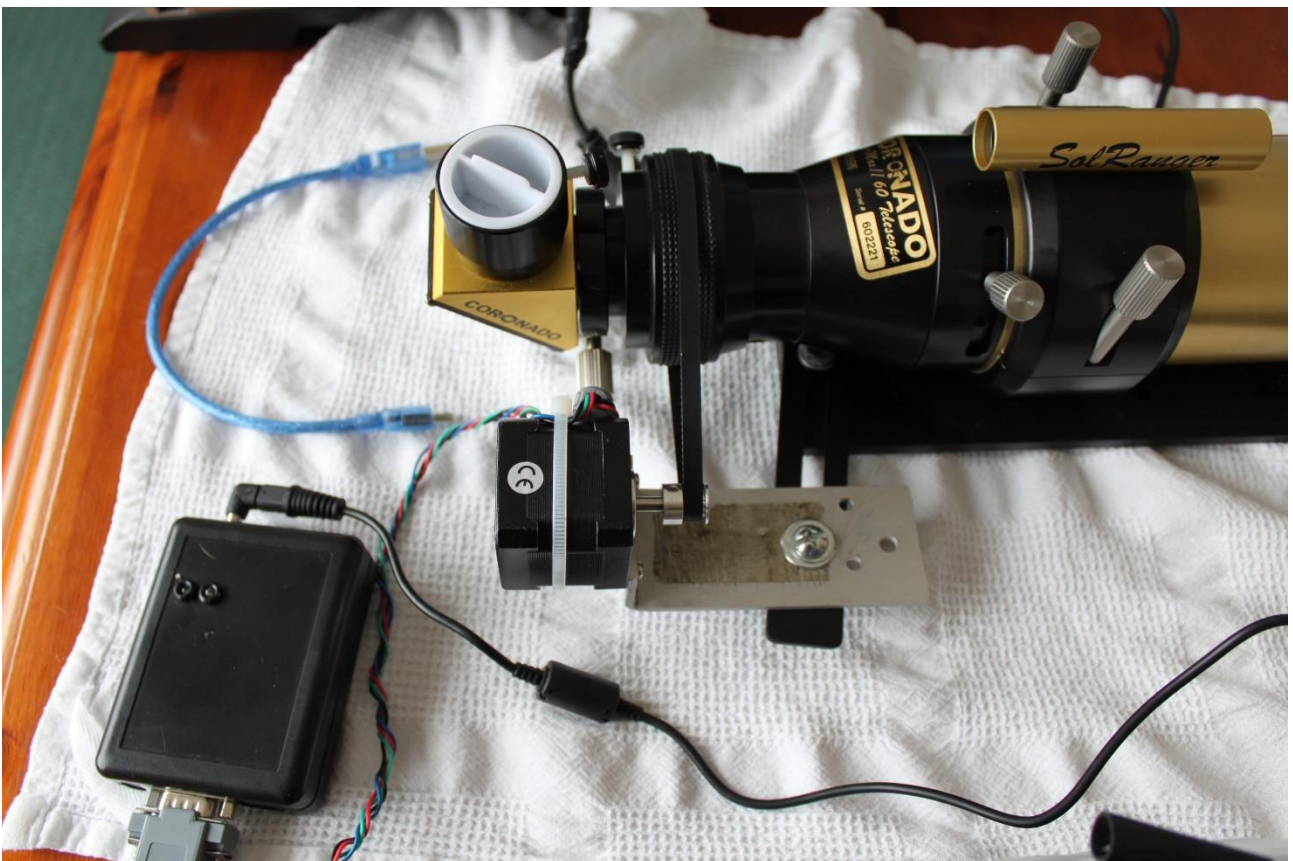
Nema 17 with belt drive as a focuser for a Coronado SolarMax telescope

SYMBOLS AND MEANINGS USED IN BOARD LAYOUTS

Stripboard layouts are found in the STRIPBOARDS folder.

Resistor

3 pin header

Transistor

Capacitor (Ceramic)

LED

OUT 78XX IN — Voltage regulator

Bipolar/Electrolytic Capacitor

Diode

12V — 12V external power

Toggle or Slide Switch

Buzzer

5V — 5V, normally from Nano

Fuse and Fuse Holder

Ground
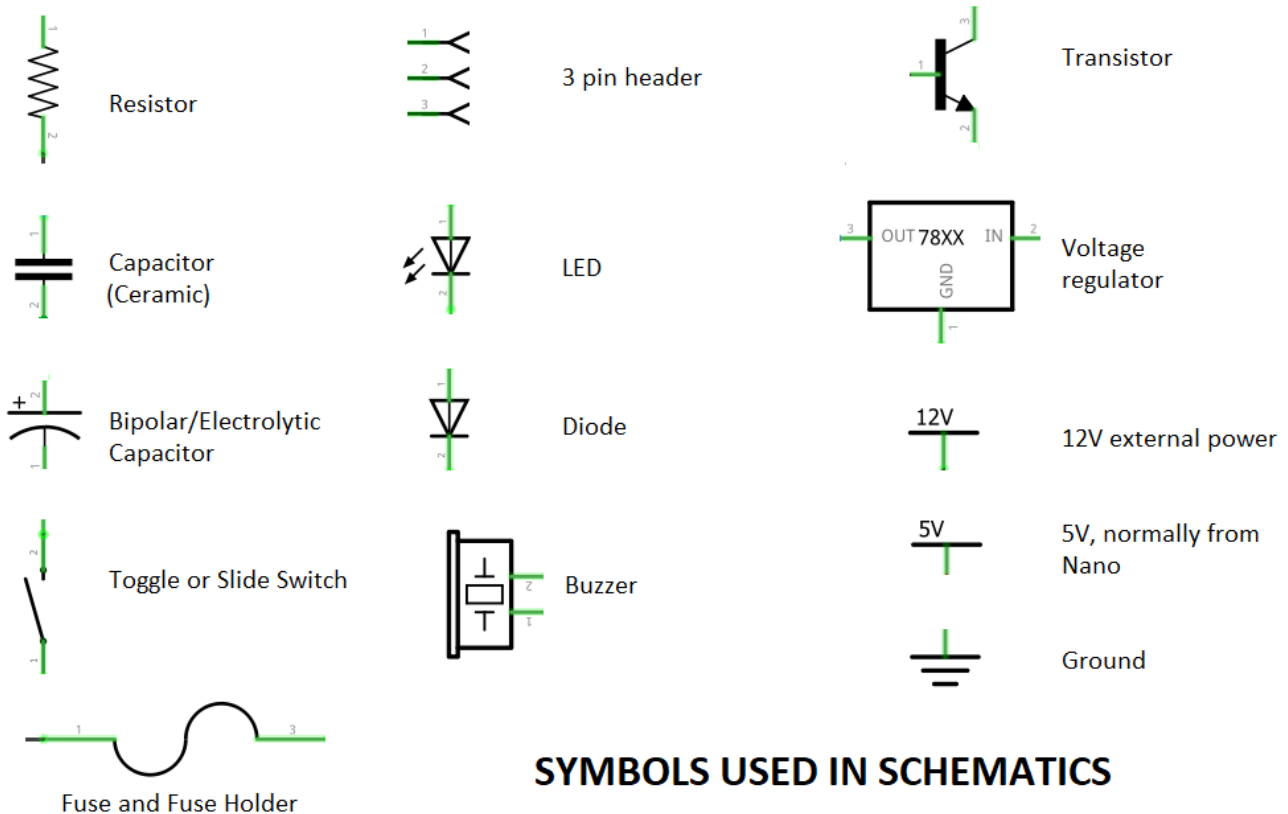
SYMBOLS USED IN SCHEMATICS

A single (1-pin) header

Arduino header pins, 40 pins, 2.54mm spacing - break off the required number and solder to board

A 2P (2-pin) header

A 3P (3-pin) header

A 4P (4-pin) header

A jumper wire

A cut track on underside of board

A Transistor or Voltage regulator. The thick bar represents the metal back of the component

1.2K   1M   Examples of resistors to be soldered between designated pins

A Capacitor

On a stripboard this is just the hole in each corner of the stripboard that is used for mounting the stripboard in a case

STRIP BOARD SYMBOLS USED IN DIAGRAMS AND LAYOUTS

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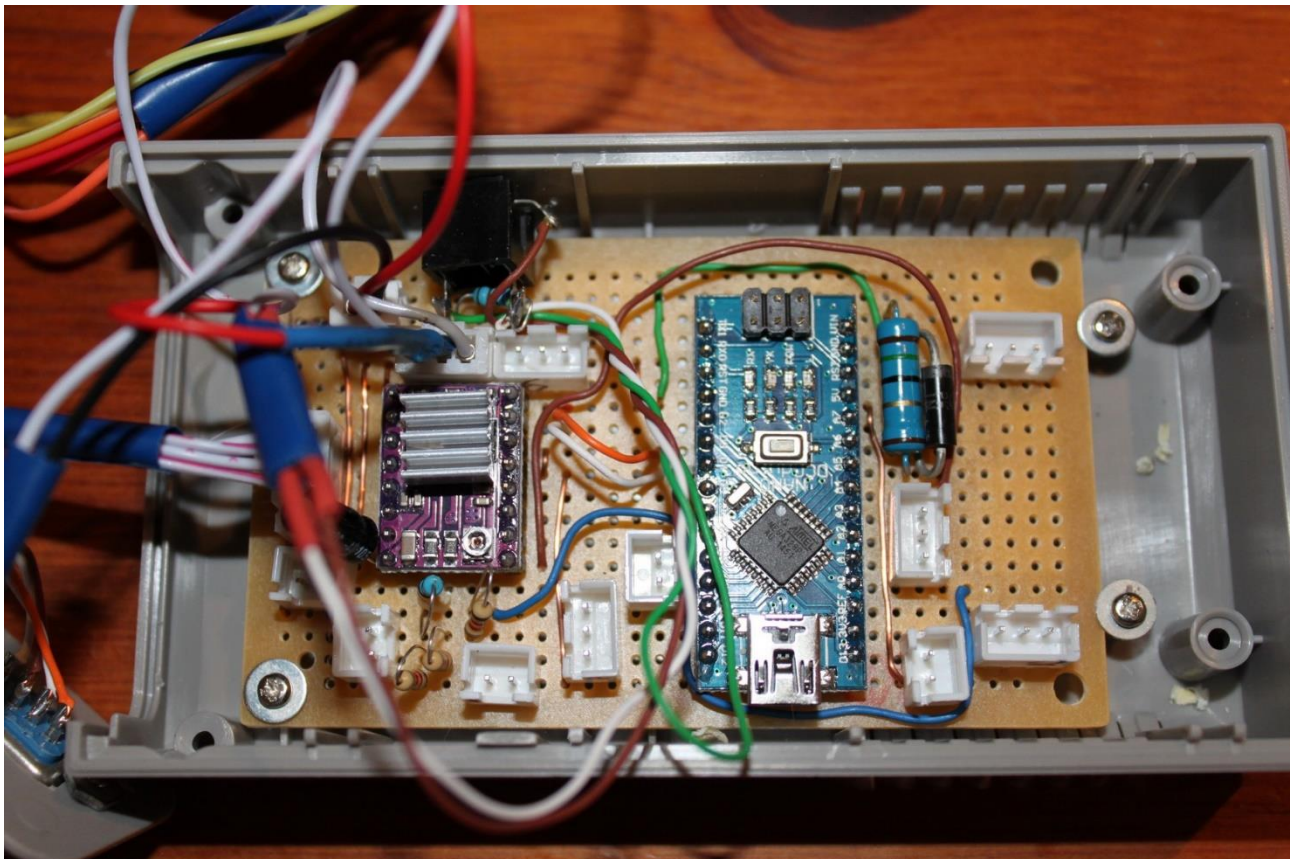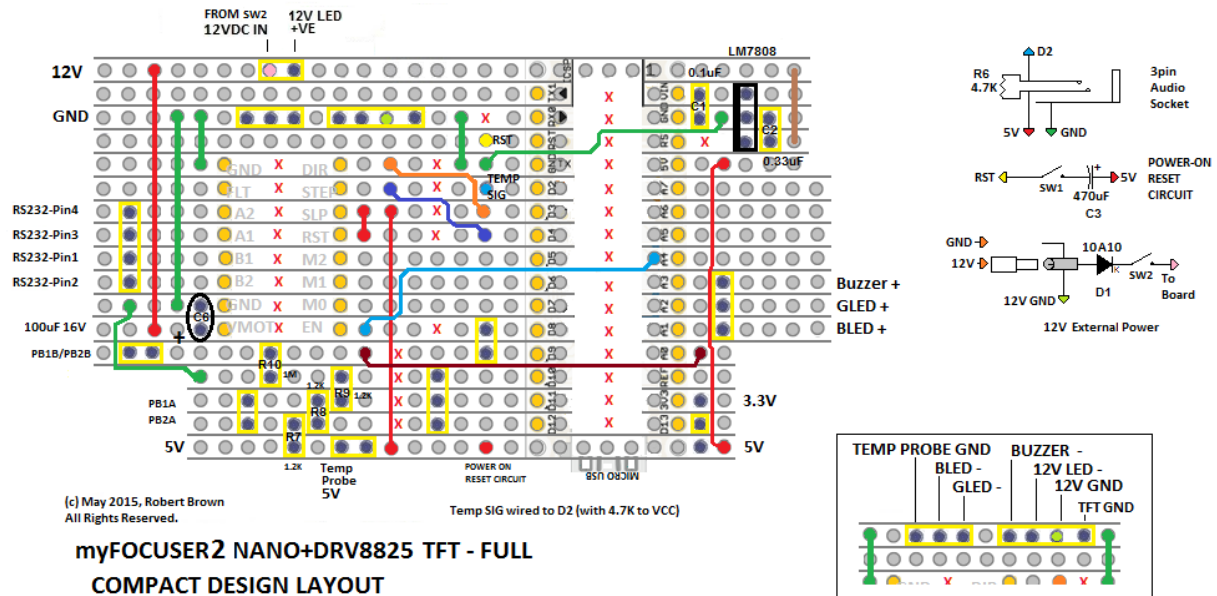
Note: This is a previous version stripboard which used a Zener diode regulator which has since been replaced with a LM7808 voltage regulator (top right of board).

Note: Not all software features may be available on Stripboards due to missing hardware components.

THE ULTIMATE FOCUSER SOLUTION

The closed loop focuser solution uses a special closed loop NEMA17 stepper motor and closed loop driver. This is especially suited for Rack and Pinion focusers [not recommended for Crayford focusers dues to slippage issues]

Because this system uses an encoder, the feedback ensures that there are **zero** lost stepper steps.

https://www.omc-stepperonline.com/closed-loop-stepper-kit/tp-series-1-axis-closed-loop-stepper-cnc-kit-48ncm-67-99oz-in-nema-17-motor-and-driver.html

The PCB supports the following options [if components fitted to PCB]
- Push buttons switches
- Display [LCD or OLED]
- Home Position Switch
- Bluetooth [HC05]
- Temperature Probe [DS18B20]

CLOSED LOOP SCHEMATIC

CLOSED LOOP
LCD1602I2C OR OLEDI2C
(c) R Brown 2020
Rev 01, May 14 2020

DRIVER: Closed Loop Stepper Driver 0-3.0A 24-48VDC for Nema 17 Stepper Motor
STEPPER: P Series Nema 17 Closed Loop Stepper Motor 48Ncm/67.99oz.in with Encoder 1000CPR
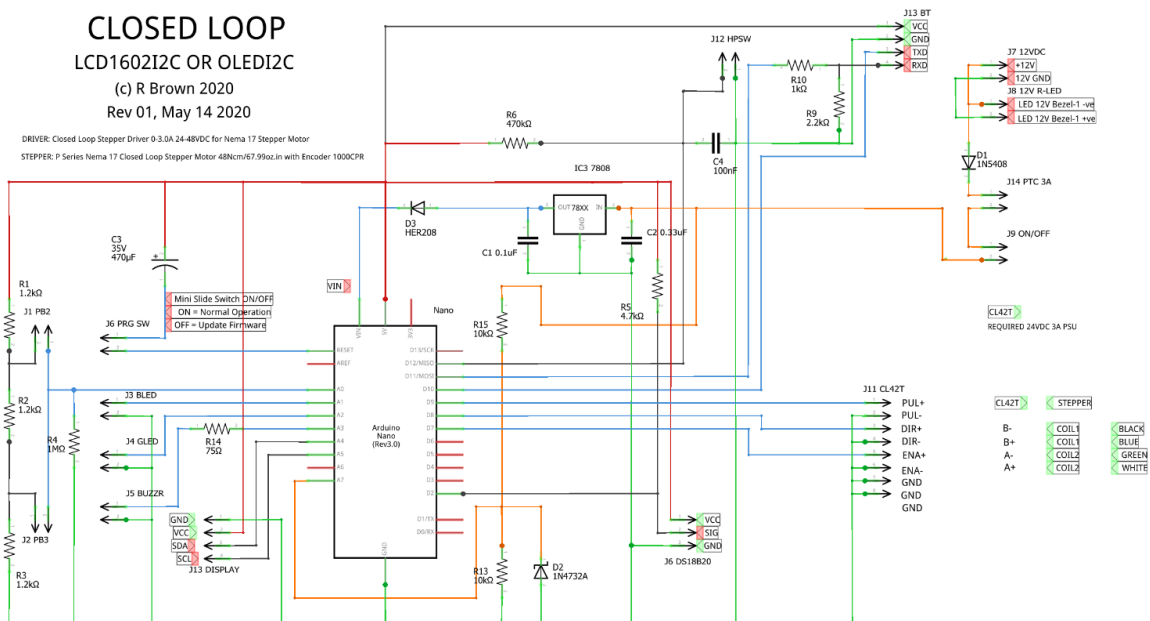
CLOSED LOOP PCB
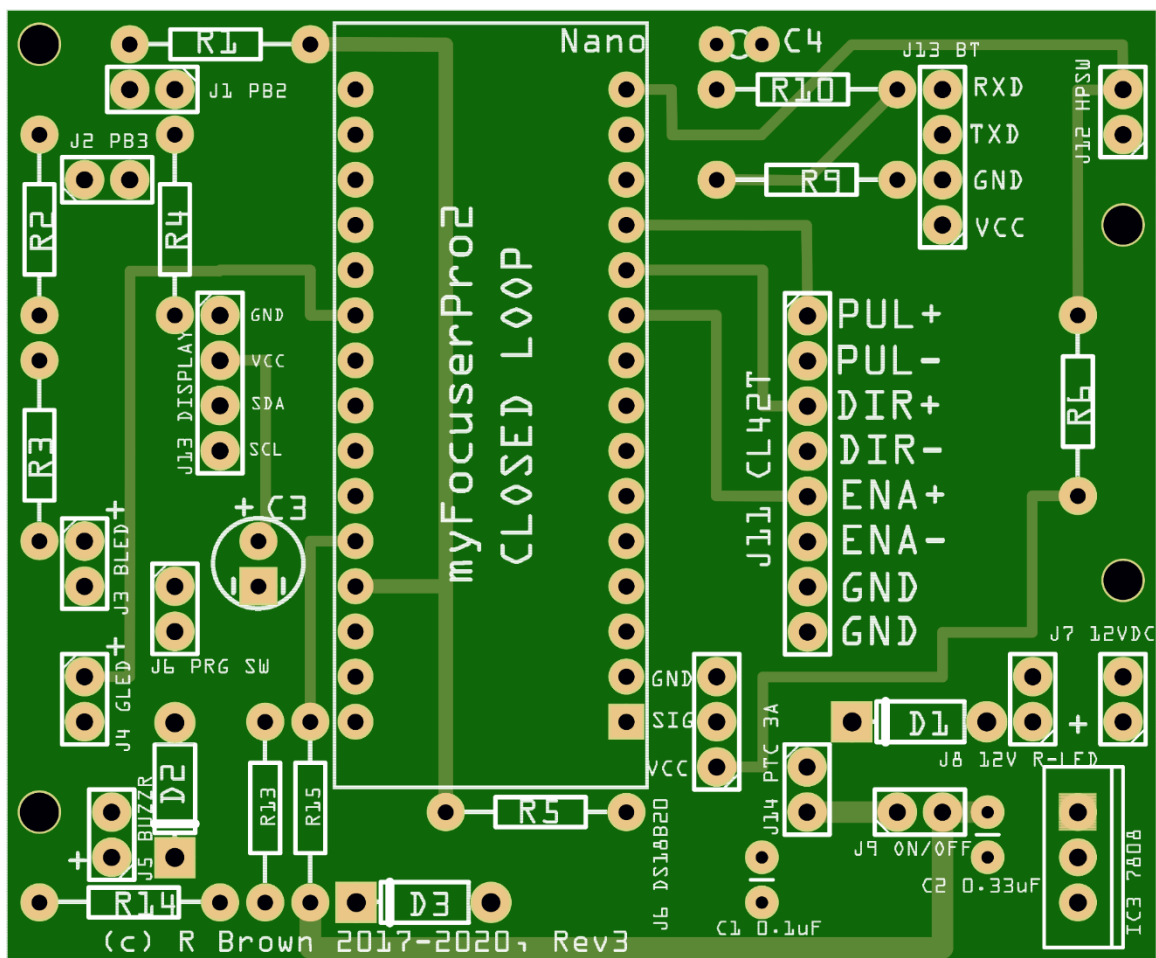
CLOSED LOOP FIRMWARE

The myFP2 firmware file to support the closed loop PCB and system is myFP2_CLOSEDLOOP_xxx

MICROSTEPPING

The micro-stepping settings are done by hardware switches [SW1-SW4] on the CT42T driver unit]

PACKAGE KIT

The stepper motor and driver package can be purchased from stepperonline

https://www.omc-stepperonline.com/closed-loop-stepper-kit/tp-series-1-axis-closed-loop-stepper-cnc-kit-48ncm-67-99oz-in-nema-17-motor-and-driver.html

24VDC PSU REQUIRED FOR CT42T DRIVER UNIT

https://www.omc-stepperonline.com/switching-power-supply/100w-24v-45a-115230v-switching-power-supply-stepper-motor-cnc-router-kits-s-100-24.html

APPENDIX A   ASCOM SUPPORT

The following lists the ASCOM support provided by myFP2 ASCOM Driver. The myFP2 ASCOM driver has passed the CONFORM report.

Property

Absolute	Implemented
Connected	Implemented
Description	MyFP2 ASCOM Driver
DriverInfo	Implemented
DriverVersion	Implemented
InterfaceVersion	2
Halt	Implemented
IsMoving	Implemented
Link	Implemented
MaxIncrement	Implemented
maxStep	Implemented
Move	Implemented
Name	myFP2 ASCOM driver
Position	Implemented
StepSize	Implemented
SupportedActions	returns a NULL list as not implemented
TempComp	Implemented
TempCompAvailable	Implemented
Temperature	Implemented

APPENDIX B   TESTING THE myFP2 CONTROLLER

The firmware zip file, when extracted, creates a Tests folder that contains all the test programs needed to test each function of the myFP2 controller.

If you have a specific issue, you should look in the Tests folder for the specific test program related to your issue.

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

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

All programs use the serial port monitor at 57600bps.

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

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

Test programs are available on the [Sourceforge](#) site

Testing the Home Position Switch

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

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

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

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

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

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

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

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

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

APPENDIX C: RUNNING TWO FOCUSERS

The ASCOM driver is not re-entrant so multiple instances cannot be run. This means that to run two myFP2 controllers on the same computer requires two separate ASCOM drivers.

The **first** ASCOM driver is known as **myFP2ASCOM** in the chooser.
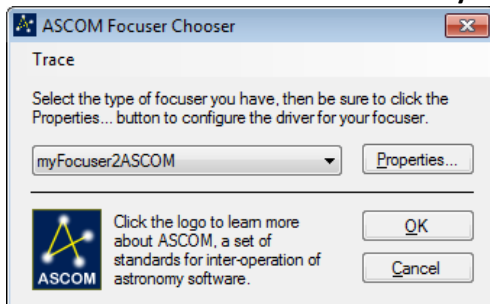
The **second** ASCOM driver (if installed) is known as **myFP2ASCOM1** in the chooser (and must connect to a different focuser controller than the first driver).
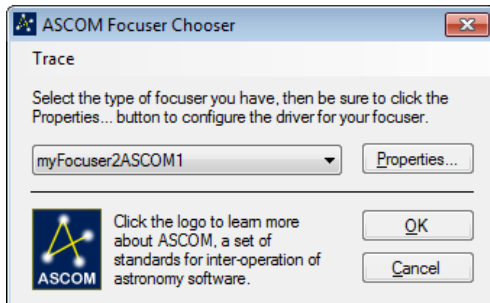
Consider the case where you have a DRV8825 controller on COM PORT3 and a second ULN2003 controller on COM PORT 4. To set this up, you would connect the myFP2ASCOM to the DRV8825 controller by specifying COM PORT3 under properties, and specify myFP2ASCOM1 to the ULN2003 controller by specifying COM PORT4 under properties for that driver.
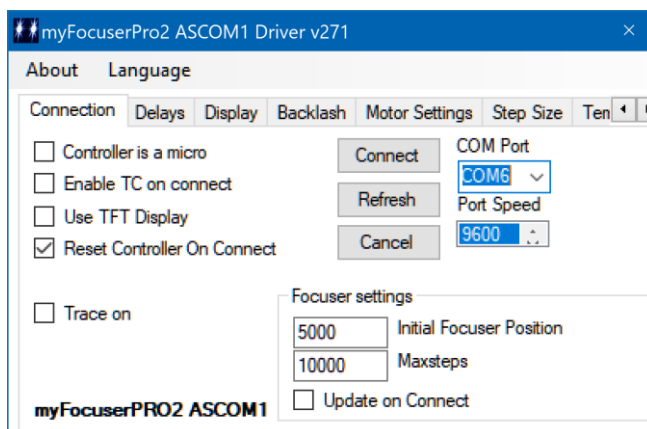
You can then work with both focusers independently.

To install the second ASCOM driver, run the setup program for the second ASCOM driver (myFP2ASCOM1Setupxxx.exe)

The second ASCOM driver setup dialog box is in a different colour and labelled so you can easily identify it is the second driver as is called **myFP2ASCOM1**.

Remember that there are additional settings accessible from the other tab pages.

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

APPENDIX D: WHAT ABOUT STEP SIZE?

Step size is the amount in microns that the focuser travels for a single step. The myFP2 controller supports the implementation of step size (ASCOM driver can return the step size if enabled, else the driver returns a not implemented exception).

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

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

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

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

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

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

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

Note: 1mm = 1000 microns.

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

The settings for StepSize and whether it is enabled in the controller is specified in the Extra Settings form of the Windows Application or the Settings form for the ASCOM driver.
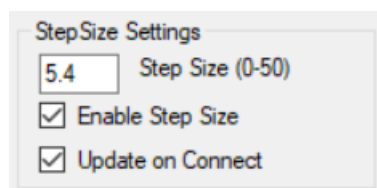
To enable stepsize in the controller, you must specify the step size value, check the Enable Step Size box and check the Update on Connect box before connecting to the controller.

STEP SIZE IN SGPRO

SGPRO has a "step size" setting in the focuser properties. This is **NOT** the same as the focuser step size.

To calculate the "step size" value for SGPRO
- Adjust for best focus
- Set the number of focus points to 9
- Note focuser position (example 12321)
- Note HFR value (example 1.9)
- Cycle subs continuous (image capture) - Move focuser out till HFR value is 3 to 5 times the best HFR value (8)
- When this is reached note focuser position (example 12571)
- Deduct original focuser position from current focuser position (12571-12321=250)
- Step size = steps moved (250) * 2 / focus points (9) - 1
- Step size = (250) * 2 / 8
- Step size = 500 / 8 = 55
- Stop subs
- Open equipment profile, focuser section, enter in "step size"

STEPSIZE IN APT

In APT under GEAR, Stepsize does not refer to the stepsize of the controller. In APT stepsize refers to when APT issues a step, how many steps does the focuser move. In our case it should be set to 1.

In focus-aid settings,
FWM Threshold - take about 4-5 images and measure the FWHM. Let's say it came as 4.5, 4.65, 4.4 and 4.45
This means that FWM is varying through seeing by 0.25 so you would set this threshold to a little higher than the variation, say 0.3 or 0.35. Note that this will vary from night to night depending upon seeing.

Fine step - this is closely related to how many steps there are in the CFZ. If you calculated say 10 steps in CFZ then I would set this to maybe slightly less like 7 or 8.

Coarse step - set this to at least fine step * 2, a good number should be number of steps in cfz * 2 (rounded to whole number)

Max moves - 100 should be fine.

YOU NEED TO BE NEAR FOCUS WHEN RUNNING THE AUTO-FOCUS AID - within say 50 steps or so of best focus, unless you change Max Moves.

Note: This information has been gleaned from user feedback and the authors own testing. This might not apply to your particular imaging setup, but may serve as a starting point. All feedback most welcome.

APPENDIX E: STEPPER MOTOR TO CONTROLLER CONNECTION

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

The stepper motor wires are terminated using a RS232 female connector. A TDK ferrite core clip-on cable clamp is clamped over the stepper wires (close to the stepper motor) to minimize back EMF.
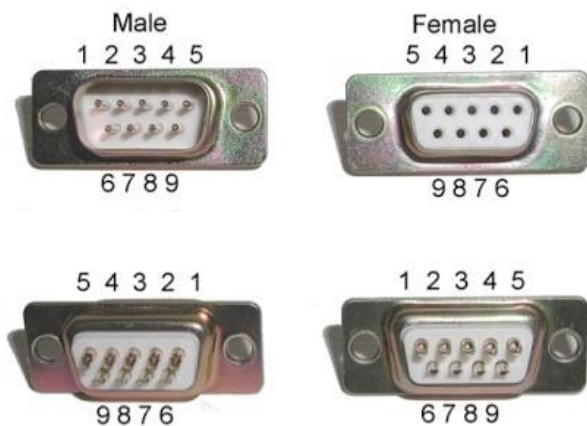
On the controller box, the M1/M2 (or M3/M4 in the final version) wires from the L293D shield are wired to a RS232 female socket mounted on the case.

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

The maximum length of the cable for reliable operation depends on the characteristics of the cable wire, but should be kept as short as practical for reliable operation.
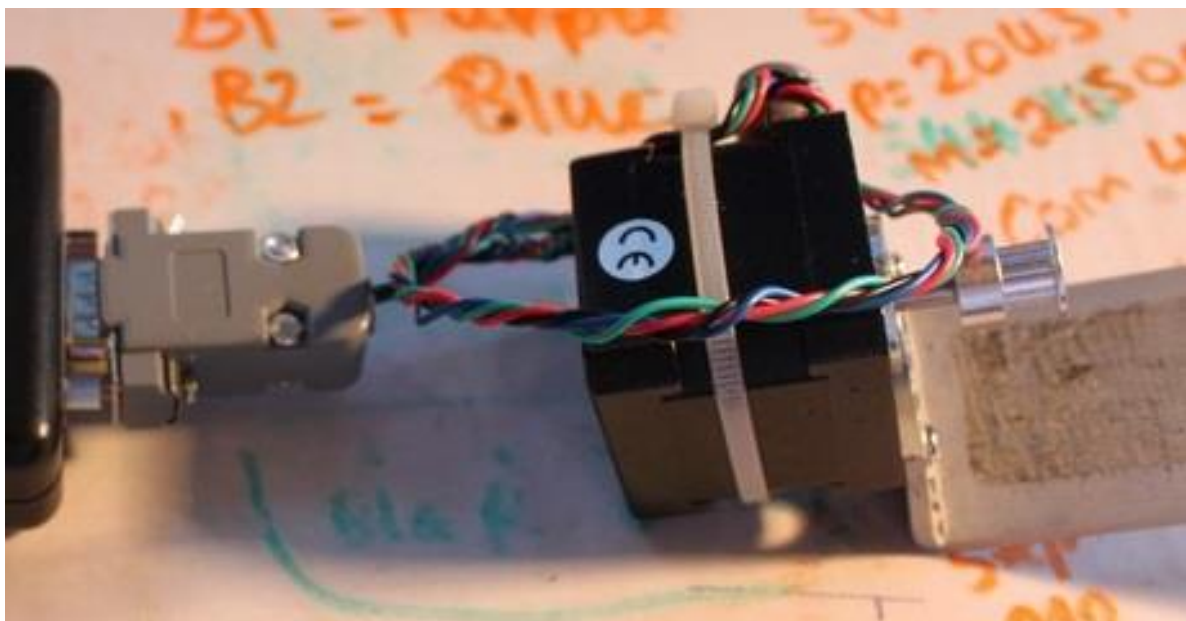
I have used 6 feet RS232 cables between the myFocuser2Pro controller (mounted on a Pier/tripod) to the stepper motor (mounted on the telescope) without any loss in performance.

WIRING THE NEMA17 COILS TO RS232 CONNECTOR

NEMA17-PG5
NEMA17-PG27

NEMA17

BLK
GRN
RED   BLU

BLK
RED
GRN   BLU

RS232-1	BLUE
RS232-2	RED
RS232-3	GREEN
RS232-4	BLACK

RS232-1	BLUE
RS232-2	GREEN
RS232-3	RED
RS232-4	BLACK
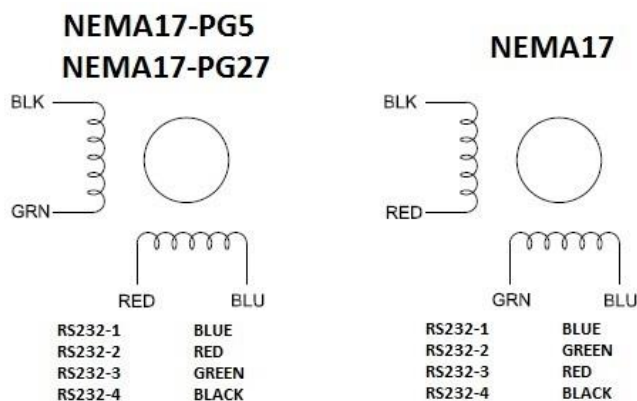
For wiring of the L293D shield to the RS232 connector

MotorPort-1	RS232Pin	Nema17-PG5	28BJY-48	Stepper Motor -> RS232 DB9pin Female Connector
M2	1	Blue	Yellow	
M2	2	Red	Blue	
GND	5	-	Red	L293D Motor port -> RS232 DB9pin Female connector
M1	3	Green	Pink	
M1	4	Black	Orange	
MotorPort-2	**RS232Pin**	**Nema17-PG5**	**28BJY-48**	To connect the myFP2 controller to a stepper motor use a RS232-DB9 Male-to-Male cable (straight through extension cable)
M3	4	Black	Orange	
M3	3	Green	Pink	
GND	5	-	Red	
M4	2	Red	Blue	
M4	1	Blue	Yellow	

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.

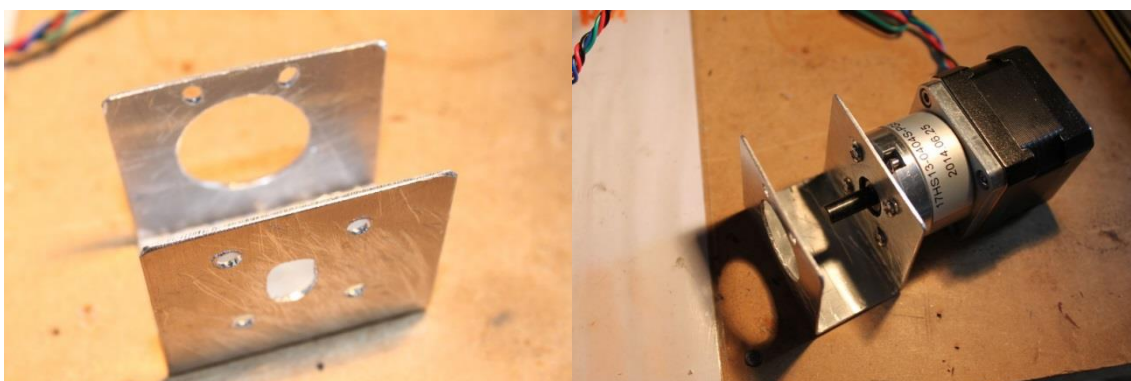# APPENDIX F: NEMA17-PG5 STEPPER MOTOR BRACKET

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.
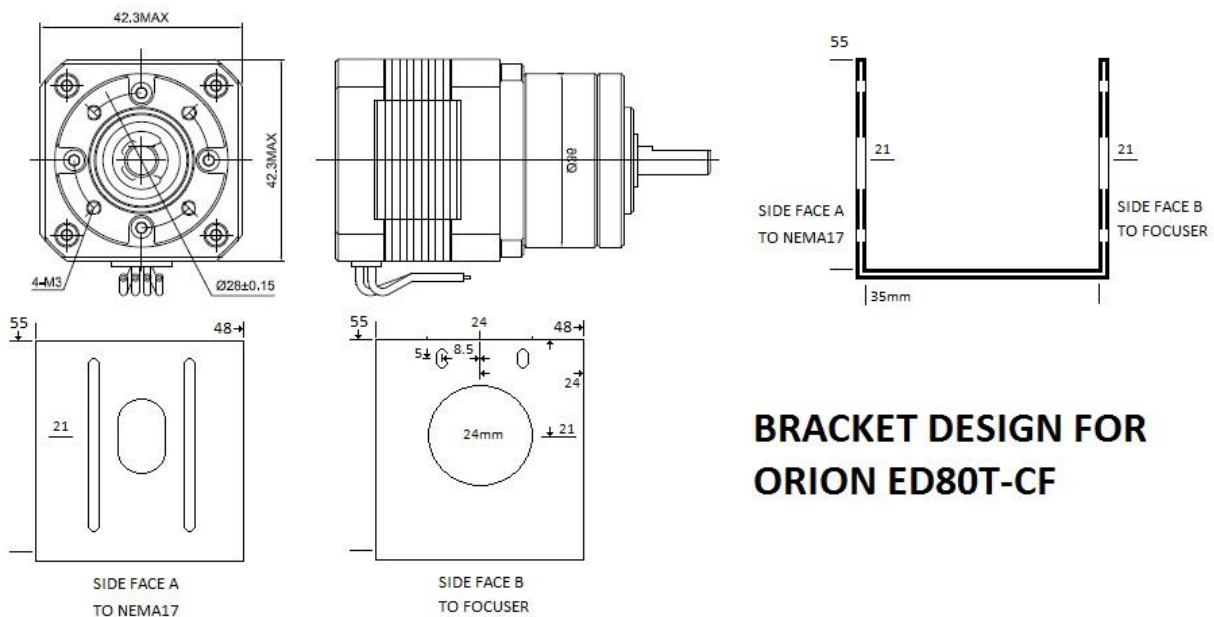
BRACKET DESIGN FOR ORION ED80T-CF
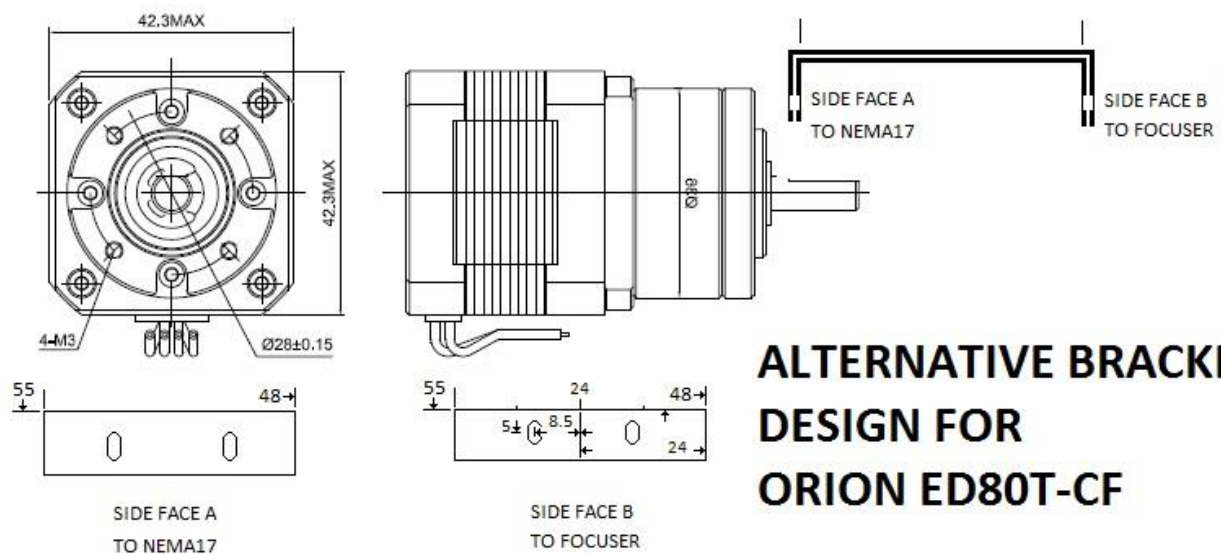
ALTERNATIVE BRACKET DESIGN FOR ORION ED80T-CF

APPENDIX G: USING A PULLEY AND BELT DRIVE

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

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

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

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

APPENDIX H: DRV8825 DRIVER BOARD

PRECAUTIONS

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

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

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

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

For the recommended NEMA17PG27, most solutions would use FULL or ½ stepping to get the desired resolution (number of stepper motor steps within the Critical Focus Zone).
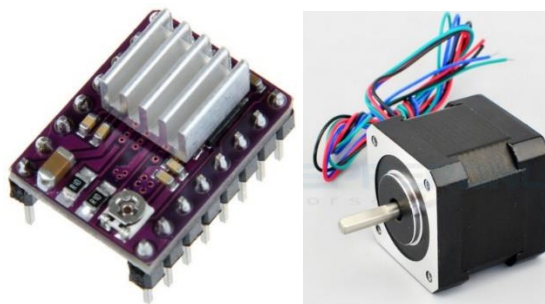
myFP2 DRV8825 PURCHASE LIST
Please refer to the separate spreadsheet for a full parts list

myFP2 DRV8825 CONTROLLER DEFAULTS

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

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

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

MICROSTEPPING

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

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

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

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

ADJUSTING THE POT ON THE DRIVER BOARD FOR CURRENT MAXIMUM

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

Adjust this Pot to vary the current in the stepper motor coils.
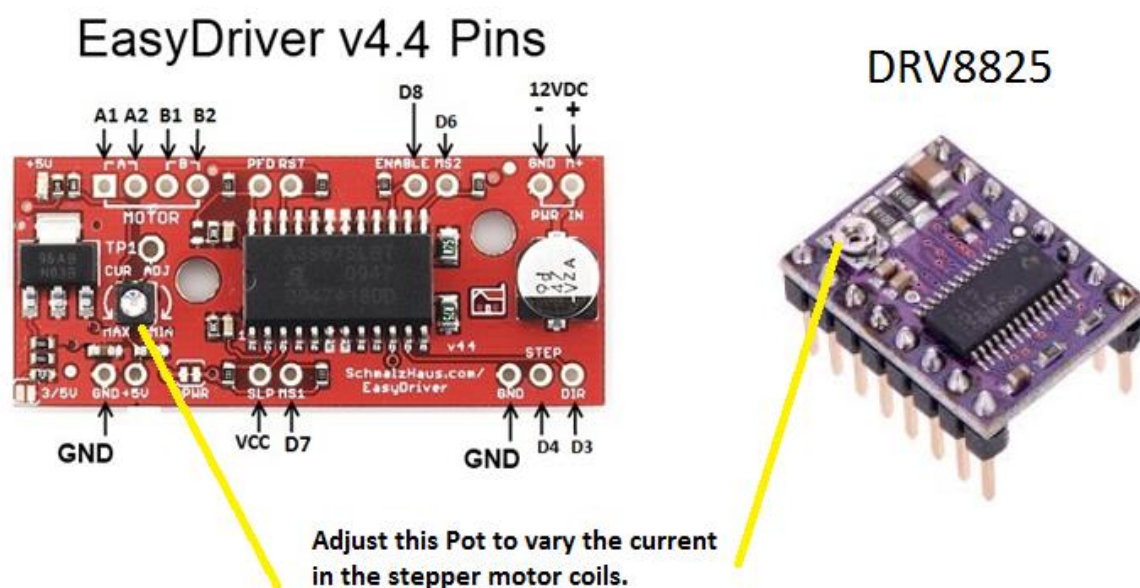
Adjusting the Stepper Motor manually

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.

myFP2 DRV8825 PROTOTYPE BUILD PICTURES

Initial prototype and breadboard (DRV8825 sub-board) to test DRV8825 driver board with NEMA17 hybrid bipolar stepper motor using Arduino Uno
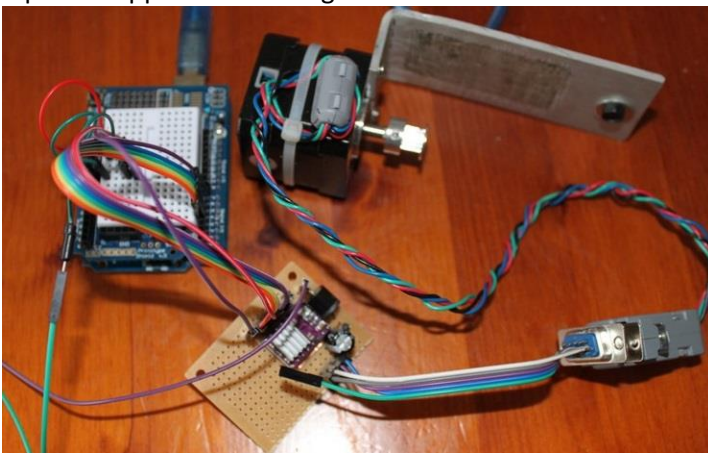
Arduino Nano, DRV8825 sub-board housed in small plastic case (minimal solution) and RS232 connector for stepper motor, 12VDC panel mount socket, two LEDS for indicating stepper direction (IN/OUT)

Finished DRV8825 controller

WIRING THE DRV8825 BOARD TO RS232 CONNECTOR
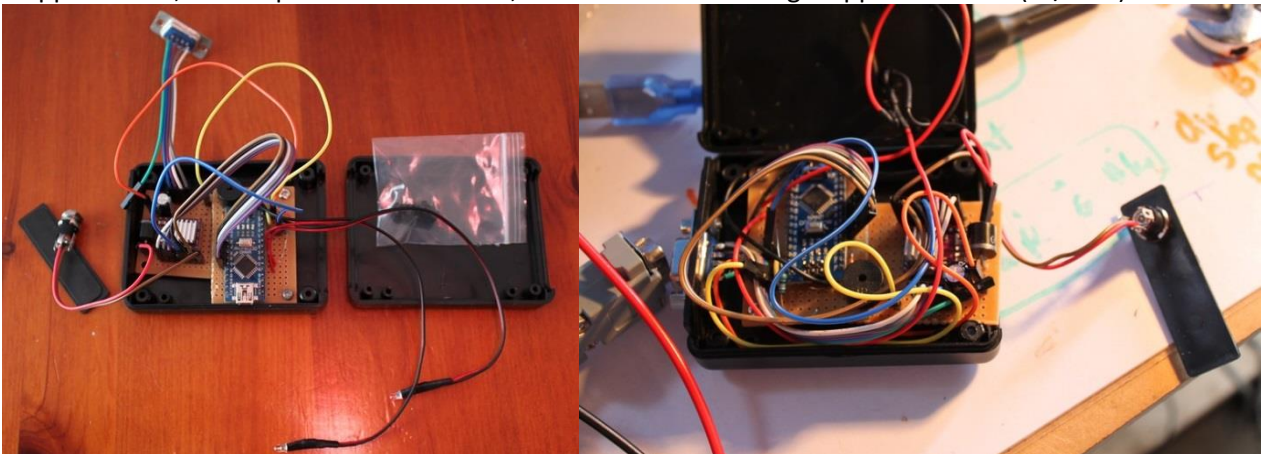
The DRV8825 driver board has four outputs for driving a bipolar stepper motor. These are connected as shown below

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

Note: The stepper motor wires can differ from model to model. You may need to identify which pairs are which with a multi-meter. Pair one is wired to the pins 1 and 2 of the connector, with pair two wired to pins 3 and 4. If you find the motor is turning the wrong way (when you move inwards to a lower focuser position the motor moves the focuser out, you should reverse the wires on pin 1 and pin 2 of the connector. Then you are good to go.

APPENDIX I: WIDE FIELD IMAGING WITH CANON EOS LENS

This article discusses how the myFP2 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 myFP2 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 myFP2 application is started. As the focuser is already near focus, any fine adjustment of focus will be relatively small, so the initial focus position is set to 5000, maxStep set to 10000 and half-steps enabled. This should provide more than enough steps to achieve good focus. Focus is then achieved by moving the myFP2 controller IN or OUT as required.

APT WITH CANON EOS 500D WITH F2.8 70-200L LENS FITTED WITH myFP2 DRV8825 and NEMA17
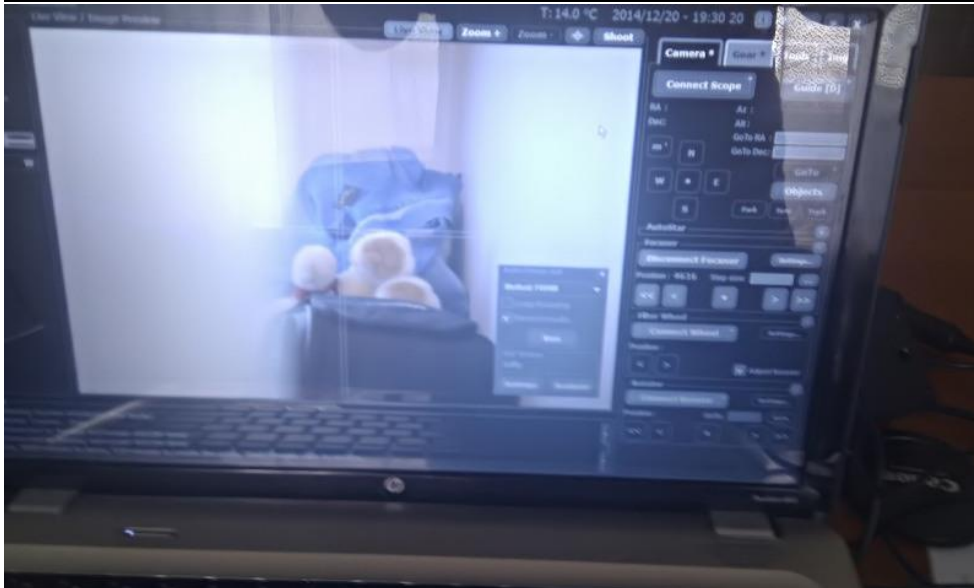
Focusing done in LIVEVIEW WITH ZOOM and ASCOM focuser jogged till best focus achieved.

APPENDIX J: TEMPERATURE DS18B20 PROBE PRECISION

The resolution of the temperature sensor is user-configurable to

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

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

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

Both the Windows application (via the Settings, Temperature Precision menu) and ASCOM driver (on the main Setup dialog form) lets the user specify the temperature precision when connecting to a myFP2 controller.

The temperature precision value is saved in EEPROM of the myFP2 controller, thus is remembered from session to session. For the ASCOM driver, this is found on the Settings form.
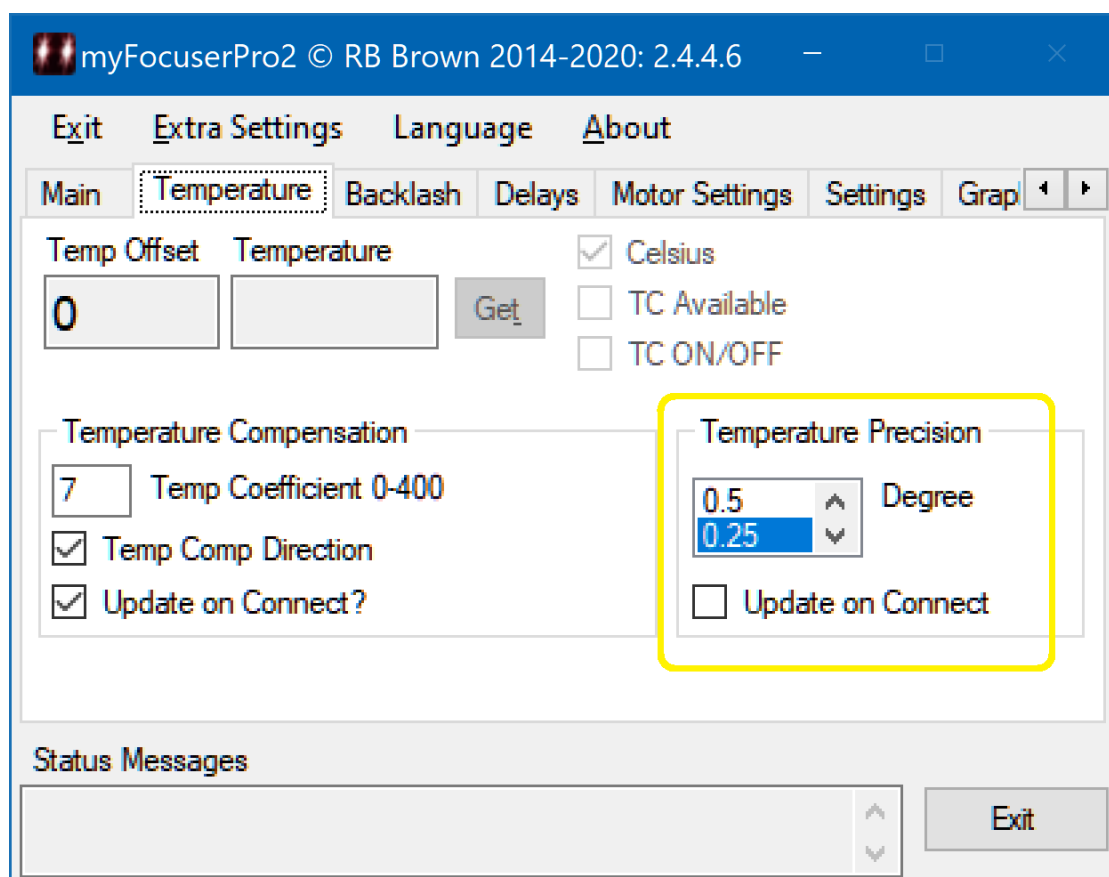
APPENDIX K: SMALL STEP MODES AND DIODES

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

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

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

The diodes used are 1N5404.

Original Article

http://cabristor.blogspot.co.nz/2015/02/drv8825-missing-steps.html
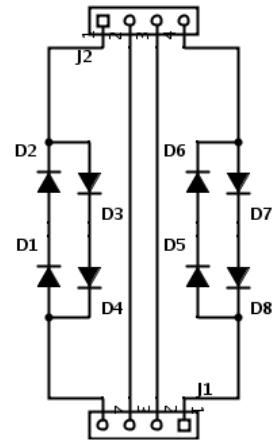
An alternative would be to use 1N5822 Schottky diodes which have a similar rating but react faster and have a lower forward voltage drop than the 1N5404 diodes.

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

APPENDIX L: myFP2 COM SERVER

This program is a myFP2 controller emulator which I use for testing purposes. To use the application first install Com0Com from http://com0com.sourceforge.net/
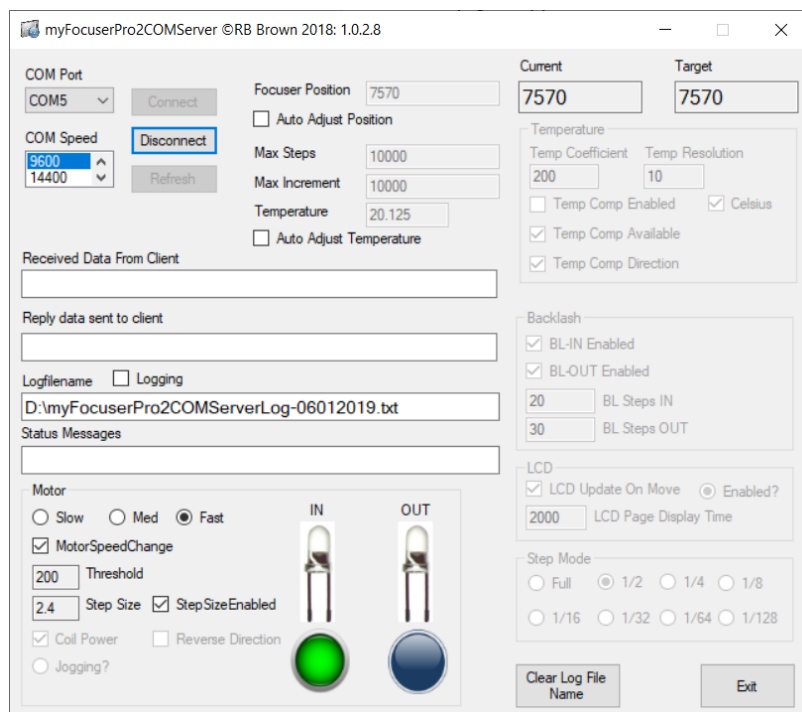
The output to CNCA0 port should be the input from CNCB0 port and vice versa. But you have to use the COMx and COMx associated with the CNCA0 and CNCB0 ports, which on my system CNCA0=COM7 and CNCB0=COM6.

To run this
1. Start this program
2. Select COM6 as the com port and connect
3. Start the myFP2 windows app and connect to COM7 as the com port

and away it should go

Avoid trying to change parameters in this program when it is running

COM0CM SETTINGS