

JMI Smart Focus Controller

Software Version 3.02
2-15-2002

© Jim's Mobile, Inc. (JMI Telescopes)

This focuser controller will employ an RS-232 interface to a Host PC, using a 9 pin female D-Sub connector. A six wire modular cable connects the focus controller to a JMI DRO focuser fitted with an optical encoder. The focuser motion is then under computer control.

Additionally, there will be pushbuttons on the focuser to manually move the focuser in both directions. These are disabled during go to operations. When one of these pushbuttons is pressed, the focus motor will move slowly for about 3 seconds, and then accelerate to a high speed, allowing for long shuttle distances. When doing fine adjustments, which last less than 3 seconds, all motion will be at slow speed in both directions.

During Go To Zero Position, and Go To Specified Position commands, the pushbuttons are disabled. At times when the pushbuttons are disabled, the power on indicator LED will blink. The pushbuttons are enabled when the LED is constantly on.

The unit is designed to operate at 12 to 14 volts DC. A DC power input jack will allow connection to an AC adapter, or a 12 volt battery. The unit will work from a 9 volt battery, but battery life will be short. The power supply current will be a constant 10 to 15 mA, draining a battery in less than 10 hours.

There will be 3 DIP switches used to configure the unit, as follows:

- **Baud Rate Select:** This switch selects the RS-232 port baud rate. The two selections are 2400 and 9600 baud.
- **Motor Direction Select:** This setting allows for selecting the motor polarity, in the event the motor rotation is reversed. Furthermore, with this switch, the Zero position can be selected to be either full in, or full out. This direction is transparent to the host computer. The encoder position will always be a positive number, and will increase as the focuser moves away from Zero position.
- **Motor Voltage Select:** This setting allows for running focusers with different drive motors. One setting will allow operation with 6 to 9 volt motors, the other with 3 to 4.5 volt motors.

Due to the wide variety of focuser configurations that may be employed, the focuser controller will have no knowledge of the physical properties of the focuser. Only the host PC will have information about the physical characteristics of the focuser, specifically, the drive shaft diameter, number of encoder lines per revolution, and focuser travel distance. The host PC has a configuration program which allows the user to input this data. The host PC program will use this information to calculate the maximum travel setting, and settings for motor speeds. These values are then transferred to the focuser, where they are stored in EEPROM memory. Every time the focuser position is moved, the new position value is also saved to EEPROM. Thus, all necessary settings are retained when the unit is turned off, and are recalled when it is turned back on.

The host PC will perform the translation of position information to dimensional values (inches, mm, etc). The focuser unit will simply move the focus shaft by controlling the drive motor, and keep track of position by counting encoder pulses.

The focuser position and maximum travel values use an unsigned 16-bit binary word format, based on encoder line counts. The focuser will always start at Zero position, and the Position value, always a positive number, will increment as the focuser moves away from the Zero position. Once properly set, the Maximum Travel value provides a programmed stop to prevent the motor from running on at the end of travel, which would corrupt the position register. There is no programmed stop at the zero position. The motor will, however, move only at its slow speed. This allows the user to manually set the zero position.

SERIAL INTERFACE

The focus controller serial interface will respond to the following commands from a host computer.

GO TO SPECIFIED POSITION	g
REINITIALIZE FOCUSER	h
READ POSITION REGISTER	p
READ DEVICE STATUS REGISTER	t
READ DEVICE IDENTIFY REGISTER	b
MOVE IN	i
MOVE OUT	o
STOP	s
SET ZERO POSITION	z
WRITE MAXIMUM TRAVEL REGISTER	w
WRITE POSITION SPEED REGISTER	d
WRITE MOVE SPEED REGISTER	e
WRITE SHUTTLE SPEED REGISTER	f

Detailed descriptions of the serial commands follow. All command characters are lower case ASCII letters. Additional data is sent in binary format. The serial protocol will be No parity, 8 data bits, 1 stop bit. Baud rate is selectable at 9600 or 2400. (The 2400 baud option allows for a longer cable length between the computer and focuser.)

GO TO SPECIFIED POSITION

The unit will move the focuser to the position specified in the command. The pushbuttons will be disabled during the go to operation. The LED will blink to indicate the command is in progress, and the pushbuttons disabled.

This is a three byte command. The command character is 'g'. This is followed by the desired sixteen bit position value, sent as two bytes, most significant byte first.

After the focus controller receives all three bytes, it will echo the command character back to the host, and attempt to perform the operation. Once the operation is complete, the focus controller will send a 'c' character to the host, indicating the command is complete.

If at any time it is determined that the motor or encoder is not working, the focus controller will send an 'r' character to the host, and terminate the operation. The Motor/encoder error bit will be set in the Status Register.

A Go To Specified Position command can be prematurely terminated by sending a 's' character via the serial port. (Note that the 's' character will not be echoed by the focuser controller, only the 'c' character, indicating command complete, will be sent.)

REINITIALIZE FOCUSER

The unit will determine speed settings, store all data to the EEPROM, and move the focuser to the zero position.

This is a one byte command. The command character is 'h'.

After the focus controller receives the command byte, it will echo the command character back to the host, and attempt to perform the operation. Once the operation is complete, the focus controller will send a 'c' character to the host, indicating the command is complete. Note that this command may take a minute, or more, to complete.

If at any time it is determined that the motor or encoder is not working, the focus controller will send an 'r' character to the host, and terminate the operation. The Motor/encoder error bit will be set in the Status Register.

The operation of the Go To Zero Position command is complex. Initially, the focuser will move in both directions, while the controller performs speed tests on the motor. During this phase of the operation, the LED will blink, and the pushbuttons are disabled, as is the serial port. Once the controller has determined the proper speeds to run the motor, the focuser will move towards the zero position. The focuser will move until the number of encoder

pulses exceeds the value programmed into the maximum travel register. Since this number is initially set to a very large value, the motor will run well past the time required to reach the zero position.

During this phase of the operation, the pushbuttons, and serial port are enabled, allowing the user to terminate the operation early, since the focuser will reach the zero position before the motor automatically shuts off. The LED will be constantly lit to indicate this phase. Depressing either pushbutton will stop the motor, and cause the command to complete. Likewise, sending a 's' character via the serial port will cause the same result. (Note that the 's' character will not be echoed by the focuser controller, only the 'c' character, indicating command complete, will be sent.)

At the completion of this command, all stored data in the EEPROM is updated.

READ POSITION REGISTER

This command will allow the host to read the value of the encoder position register.

This is a one byte command. The command character is 'p'.

After the focus controller receives the command byte, it will echo the command character back to the host, followed by the sixteen bit position value, sent as two bytes, most significant byte first.

READ DEVICE STATUS REGISTER

This command will allow the host to read the focus controller status byte.

This is a one byte command. The command character is 't'.

After the focus controller receives the command byte, it will echo the command character back to the host, followed by the eight bit status byte. All bits are active high; a 1 indicates the condition is true, a 0 indicates it is false. Reading the device status resets the error conditions (bits 1 through 3 only).

Reflected in the status byte are the following conditions:

- Bit 0: Unused
- Bit 1: Serial receiver framing error
- Bit 2: Serial receiver overrun error
- Bit 3: Motor/encoder error
- Bit 4: Unused
- Bit 5: Unused
- Bit 6: Focuser at zero position
- Bit 7: Focuser at maximum travel position

READ DEVICE IDENTIFY REGISTER

This command will allow the host to read the focus controller identify byte.

This is a one byte command. The command character is 'b'.

After the focus controller receives the command byte, it will echo the command character back to the host, followed by the eight bit identify byte. This identify byte is a lower case 'j' (6Ah).

WRITE MAXIMUM TRAVEL REGISTER

This command will allow the host to write the focus controller maximum travel register. This command is used by the host PC to configure the focuser controller, based on the focuser characteristics defined by the user. When this command is received, all stored data in the EEPROM is updated.

This is a three byte command. The command character is 'w'. This is followed by the desired sixteen bit maximum count value, sent as two bytes, most significant byte first.

After the focus controller receives all three bytes, it will echo the command character back to the host.

WRITE POSITION SPEED REGISTER

This command will allow the host to write the focus controller position speed register. This command is used by the host PC to configure the focuser controller, based on the focuser characteristics defined by the user. Position speed is used to move the focuser slowly when it is close to the desired Go To position.

This is a three byte command. The command character is 'd'. This is followed by the desired sixteen bit speed value, sent as two bytes, most significant byte first.

After the focus controller receives all three bytes, it will echo the command character back to the host.

WRITE MOVE SPEED REGISTER

This command will allow the host to write the focus controller move speed register. This command is used by the host PC to configure the focuser controller, based on the focuser characteristics defined by the user. Move speed is used to move the focuser slowly when a Move In or Move Out command is issued. It is also the speed the focuser moves when a pushbutton is pressed.

This is a three byte command. The command character is 'e'. This is followed by the desired sixteen bit speed value, sent as two bytes, most significant byte first.

After the focus controller receives all three bytes, it will echo the command character back to the host.

WRITE SHUTTLE SPEED REGISTER

This command will allow the host to write the focus controller shuttle speed register. This command is used by the host PC to configure the focuser controller, based on the focuser characteristics defined by the user. Shuttle speed is used to move the focuser rapidly during a Go To Position command, when far from the desired position. It is also the speed the focuser moves when a pushbutton is pressed for more than three seconds, and the motor speeds up.

This is a three byte command. The command character is 'f'. This is followed by the desired sixteen bit speed value, sent as two bytes, most significant byte first.

After the focus controller receives all three bytes, it will echo the command character back to the host.

SET ZERO POSITION

This command will set the Position register value to zero, regardless of the actual drawtube position. The position value stored in the EEPROM is also updated.

This is a one byte command. The command character is 'z'.

After the focus controller receives the command byte, it will echo the command character back to the host.

MOVE IN

This command will cause the focuser to move in at a slow speed.

This is a one byte command. The command character is 'i'.

After the focus controller receives the command byte, it will start the motor, and then look for encoder changes. When motion is detected, the command character is echoed, indicating the focuser is in motion. If it is determined that the motor or encoder is not working, the focus controller will send an 'r' character to the host. The focus motor will continue to move in, until a STOP or MOVE OUT command is sent. Depressing either pushbutton will terminate the motion.

MOVE OUT

This command will cause the focuser to move out at a slow speed.

This is a one byte command. The command character is 'o'.

After the focus controller receives the command byte, it will start the motor, and then look for encoder changes. When motion is detected, the command character is echoed, indicating the focuser is in motion. If it is determined that the motor or encoder is not working, the focus controller will send an 'r' character to the host. The focus motor will continue to move out, until a STOP or MOVE IN command is sent. Depressing either pushbutton will terminate the motion.

STOP

This command will cause the focuser to stop.

This is a one byte command. The command character is 's'.

After the focus controller receives the command byte, it will stop the motor, and echo the command character.

It is possible to use the MOVE IN, MOVE OUT, and STOP commands for timed focuser moves. A minimum practical value is 100 milliseconds. After either MOVE command, wait at this long before sending a STOP command.