

EXPLORE[®]
SCIENTIFIC



PRECISION MOTION CONTROLLER



PMC-Eight Programmer's Reference

DOC-ESPMC8-002 Release 1
2018_February_27

Prepared by: Jerry Hubbell, Director Electrical Engineering, Explore Scientific, LLC.
Reviewed by: Alex Sanchez, Director Quality Explore Scientific, LLC.
Reviewed by: Dan Dickerson, APh Technological Consulting
Approved by: Scott Roberts, President Explore Scientific, LLC.

This document is copyright ©2016-2018 Explore Scientific, LLC. All Rights Reserved.

If you have any comments or questions about this document, please contact:

Director Electrical Engineering
Explore Scientific, LLC.
1010 South 48th Street
Springdale, AR 72762
(866)252-3811

The PMC-Eight® and OpenGOTO® Name and Logo are registered trademarks of Explore Scientific, LLC., All Rights Reserved. SBIG ST-4 standard copyright Santa Barbara Instruments Group, Cyanogen, LLC. RN-131 module copyright Microchip, Inc. G-11 mount copyright Losmandy Astronomical Products, Hollywood General Machining, Inc.

This is the second in a series of documents for the PMC-Eight® System:

Document Number	Title
DOC-ESPMC8-001	PMC-Eight® Owner's Reference Manual (Customer Use)
DOC-ESPMC8-002	PMC-Eight® Programmer's Reference (Customer Use)
DOC-ESPMC8-003	PMC-Eight® JOC Command Language Reference (Internal Use Only)
DOC-ESPMC8-004	PMC-Eight® Datasheet and Design Basis Document (Internal Use Only)
DOC-ESPMC8-005	PMC-Eight® Firmware and Software Design Reference (Internal Use Only)
DOC-ESPMC8-006	PMC-Eight® Firmware and Software Flowcharts (Internal Use Only)
DOC-ESPMC8-007	PMC-Eight® Hardware Reference Schematics and Datasheets (Internal Use Only)

Document Notes

Revision History

- 0.0 DRAFT Initial Draft.
- 0.1 DRAFT Added Sections VI and VII.
- 0.2 DRAFT Revised Section V to add "ESR!" Communications Reset command.
- 0.3 DRAFT Revised Section III to correct and expand on Meridian Flip discussion.
- 0.4 RELEASE 0 Miscellaneous cleanup of text and sections for release.
- 0.5 RELEASE 1 Revised Section II to add board layout information, and LED and test point information. Miscellaneous revisions and formatting changes.


This document uses the following convention for units of measure. Instead of stating: 30 counts/second or 20 arc-min/count, or 20 arc-min per count, the units are stated as: 30 counts sec⁻¹ or 20' arc count⁻¹. The superscript -1 indicates that the unit is in the denominator. You will see °, ', and " for degrees, minutes, and seconds for angular values.

Motor count values are stated in decimal (115200) and hexadecimal (0x9C40) notation. All the values returned by the controller are hexadecimal strings such as "1F92CA". Actual commands are shown in Courier New font with quotes around the command string, i.e., "ESGp0!". Do not include the quotes when typing the commands at a terminal connected to the controller.

The PMC-Eight® Controller is warranted by Explore Scientific, LLC. and you should contact Explore Scientific Customer Support at 866-252-3811 if you have any questions or issues when using this reference with your PMC-Eight® system.

EXPLORE
SCIENTIFIC

Designed by Explore Scientific® in Arkansas. Assembled in China.
Model: ESOGT-01B
©2016 Explore Scientific®, LLC. All rights reserved. Made in China.
explorescientificusa.com - 866.252.3811

 Contains FCC ID: U3O-G2M5477

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Table of Contents

I.	Introduction	4
II.	System Description	4
III.	Controller Theory of Operation and Command Language	6
IV.	PMC-Eight® ASCII Command Language	10
V.	PMC-Eight® ASCII Command Language Syntax	14
VI.	PMC-Eight® Interface Specifications	20
VII.	Miscellaneous Programming Application Notes	20

Glossary

Axis – The rotational axis that the motor is driving, either RA or DEC.

Right Ascension (RA) – The rotational axis that parallels the Earth's rotational axis.

Declination (DEC) – The rotational axis that allows the scope to move North or South.

German Equatorial Mount (GEM) – A type of mount that positions the telescope to move on one axis to counteract the rotation of the Earth.

Hexadecimal – The base 16 number system using characters "0" through "9" and "A" through "F".

Decimal – The normal base 10 number system used in everyday life using characters "0" through "9".

Rate – The rotational speed the motors are driving the mount.

Position – The number of micro-steps the motor has moved since startup.

Direction – The direction of the motor rotation (clockwise, or counter-clockwise) when looking at the shaft end.

Tracking Rate – The precision rotational speed of the RA axis motor at a very slow rate. Used to counteract the rotation of the Earth.

Tracking – The act of counteracting the rotation of the Earth to keep an object in the center of the telescope field-of-view (FOV).

North Celestial Pole (NCP) – The point in the Celestial Sphere where the Earth's rotational axis points in the Northern Hemisphere.

Firmware – The computer instructions that are installed in the permanent memory of the processor that communicate directly with the hardware I/O on the system. The firmware is compiled on an external computer and uploaded via the serial RS232 communications port.

RS232 – A communications hardware standard for serial communications.

Acronyms

RA	Right Ascension
DEC	Declination
HMI	Human-Machine Interface
GEM	German Equatorial Mount
NCP	North Celestial Pole
PMC	Precision Motion Controller
LMST	Local Mean Sidereal Time
FOV	Field-of-View
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

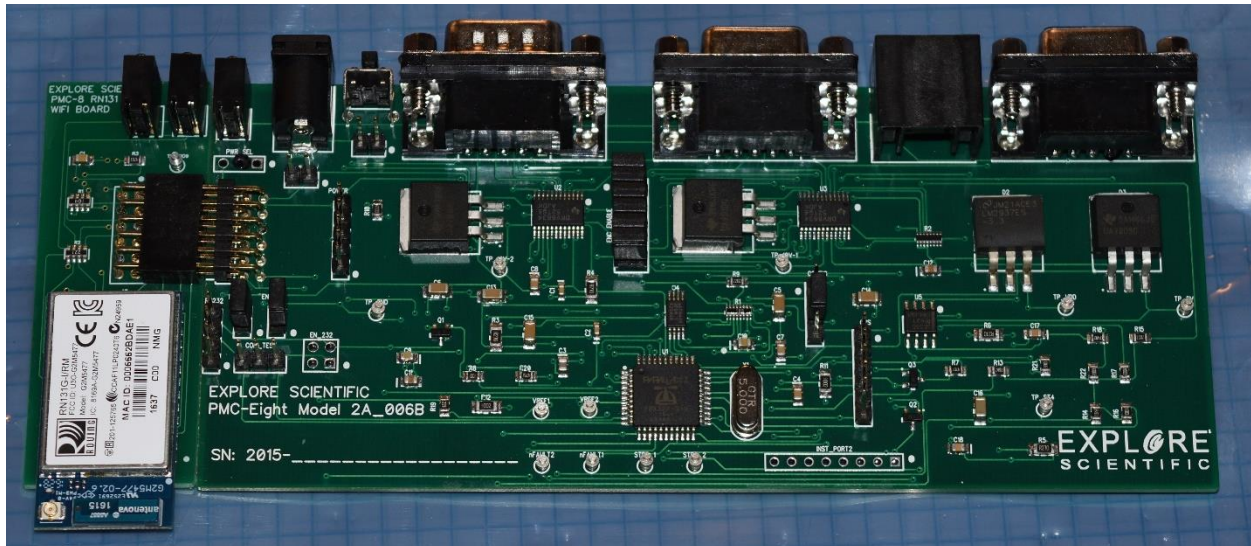
I Introduction

The Explore Scientific PMC-Eight® Controller Programmer's Reference contains the information needed to communicate and use the base controller command language. This document contains the system description and other data used to create programs to control the PMC-Eight®. Information about the PMC-Eight® controller can be found on the PMC-Eight® webpage at ExploreScientificUSA.com. Additional information is included in Application Notes available on the PMC-Eight® webpage.

One of the key elements of the PMC-Eight® System is the Explore Scientific PMC-Eight® Command Language. This language is designed to be both flexible and powerful enough to instruct the controller to perform any task the controller can perform. This language is different from most mount controller languages in that it does not contain astronomy or telescope mount specific commands. This language contains more generic motion control commands that will enable the user to quickly learn and use the system. The PMC-Eight® System also has a built-in JOC Command Interpreter used by the OpenGOTO™ Explore Stars App available at the Microsoft App Store.

II System Description

The PMC-Eight® is a high-precision stepper motor controller used to very accurately control the pointing and tracking of a telescope mount. The design basis for this Precision Motion Controller (PMC) includes features used to increase the reliability and flexibility of the system. The micro-controller selected is a multi-processor design that uses 8 processors all working as independent units and communicate through a shared memory space. Each of these processors are absolutely deterministic in their operation and DO NOT utilize interrupts. This ensures that each task assigned to each processor runs independently and any failure of any one task does not cause a cascading failure in any of the other processors.



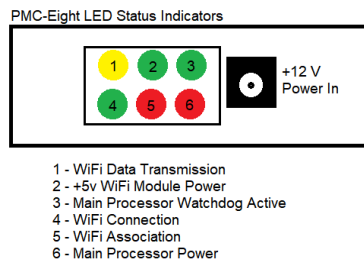
PMC-Eight® Controller Model 2A-006B

The PMC-Eight® system consists of several components and is designed to interface with the Host Computer in different ways. The primary interface to the PMC-Eight® Model 2A-006B is through a

dedicated RS-232 interface (DB-9 connector) that serves both as the programmer's (PGMR) interface for loading updated firmware into the system, and as the interface for sending data request commands and receiving data. A wireless network interface (Wi-Fi) is available also for sending data request commands and receiving data. This wireless interface is the primary HMI interface for the OpenGOTO® Explore Stars application available for the Microsoft Windows OS and Google Android OS. For details see section VII, Miscellaneous Programming Application Notes. The controller communicates at 115,200 BAUD with No Parity, 8 Bits, 1 Stop Bit over the network and through the RS232 connection.

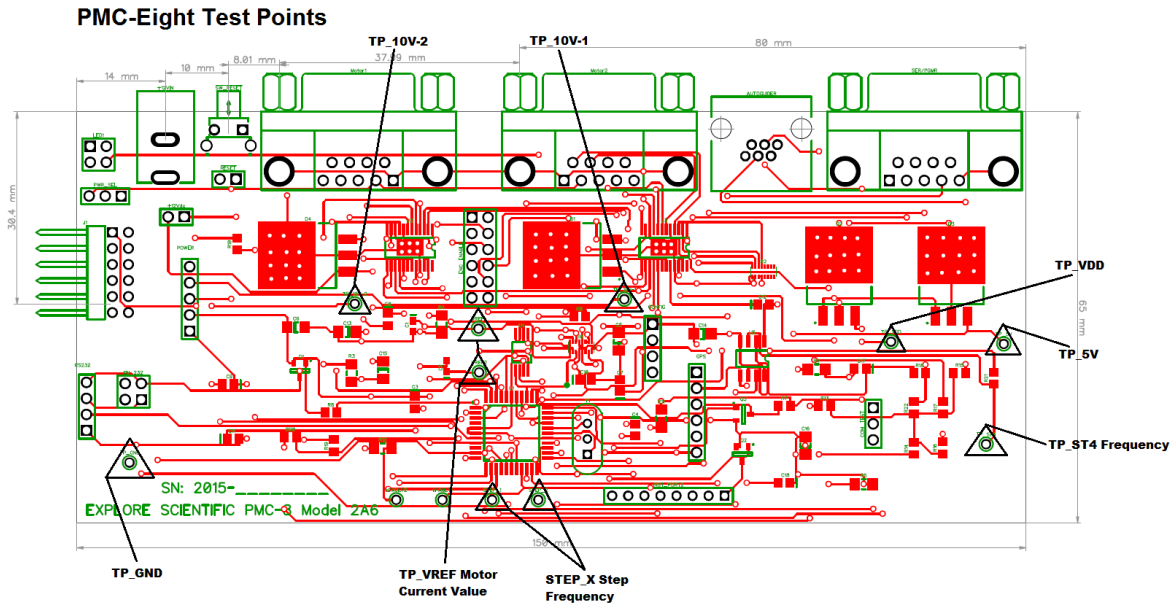
The main PMC-Eight processor board has 2 LEDs; the RED LED (6) is ON when power is applied to the board, and the GREEN LED (3) is ON indicating the processor status after the processor firmware boots up and active communications is going on with the processor, and the watchdog process is active monitoring the communications.

The RN-131 daughter board has 4 LEDs; The YELLOW LED (1) is FLASHING when there is data transmission between the PMC-Eight and the host computer application, either the ExploreStars application, or the ASCOM Driver. The RED LED (5) is the WiFi association status and flashes until a connection to the SSID (PMC-Eight-xx) of the PMC-Eight is made by the host computer. The GREEN LED (2) is ON when power is applied to the daughter board from the main board. The GREEN LED (4) FLASHES FAST when a connection is ACTIVE. LED (4) Flashes SLOW when the module is waiting on a connection.



PMC-Eight Status LEDs

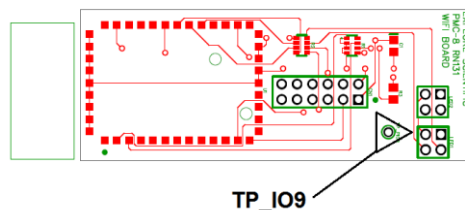
There are several test points on the circuit board assembly (indicated below with a black triangle) labeled TP_XXXX, such as: TP_GND, TP_VDD, TP_10V-1, TP_10V-2, TP_5V. There are also test points to measure the reference voltages for the current limits and the ST4 port frequency and stepper motor pulses. There are headers on the assembly which provide access to the RS-232 signals, the power supply voltages (3.3 Vdc, 5.0 Vdc, 10.0 Vdc, and GND), and to the WiFi serial communications signals (COM_TEST header).



PMC-Eight® Controller Version 2A-006B Test Points

The WiFi module (Microchip RN-131G) daughter-board has a test point (TP_IO9) used in resetting the module to the Microchip factory defaults. The “Explore Scientific PMC-Eight® Application Note PMC8-AN002: Connecting to the PMC-Eight with a Terminal Program to Configure the RN-131 WiFi Interface and Switching Between the WiFi Interface and the Serial Interface” (available at www.explorestars.com/pmc-eight) describes in detail how to connect to the Microchip RN-131 Module and restore the PMC-Eight® default configuration. You can also use Application Note AN002 to configure the Microchip RN-131 Module to communicate on your Local Area Network (LAN). The Microchip RN-131 WiFly Command Reference Manual is located at <http://ww1.microchip.com/downloads/en/DeviceDoc/50002230B.pdf>.

CAUTION: Changing the WiFi module configuration from the default factory configuration is at your own risk.



PMC-Eight® Controller WiFi daughter-board

In some instances, the WiFi environment is very busy with several SSIDs and if the PMC-Eight is using the same channel as others this can disrupt the communications regardless of the signal strength and may make the PMC-Eight WiFi drop out. The solution is to change the WiFi channel. You can do this with the little ST4 port (RJ12) dongle included with your PMC-Eight system.

Here is the procedure to increment the channel number:

1. Power up the PMC-Eight and let it go through the boot process (the lights will settle out).
2. Insert the dongle into the auto-guider port (ST4 RJ12).
3. Watch the lights cycle and the system should reboot itself.
4. Wait until it settles down again, then try and connect the ExploreStars or ASCOM Client application again.

It is recommended to power down the system then power it back up between steps 3 and 4, but this shouldn't be necessary. If the new channel still gives problems, the procedure can be repeated until successful (There are 10 channels available in total, channels 1-10).

III PMC-Eight® Controller Theory of Operation and Command Language

The PMC-Eight® Precision Motion Controller is designed to enable the quick and reliable movement of the telescope OTA to point to any object of interest on the celestial sphere. The controller is made up of 3 main electronic systems: a) Motor Driver, b) Communications Interface, c) Computer Processor and Memory system.

The Motor Driver section is designed to generate the required low level stepper motor control signals at the proper voltage and current levels needed to drive the mount as required. There is performance margin built into the system to enable the mount to operate over the design temperature range. The motor driver chips are low power and very efficient at delivering the power to the stepper motors reliably.

There are 3 communication interfaces on the system. The first is the base serial interface using the RS232 hardware specification to provide basic communications to transfer firmware and data to the permanent memory of the Computer Processor and configure the wireless network Wi-Fi processing module. The serial command processor/interpreter can be accessed via the RS232 port and also via the second interface, the wireless network port. The third communications channel is via the RJ 6P6C connector configured as an SBIG ST4 Autoguider Port. This port has dedicated interface circuitry to provide contact input commands to slowly move the mount in the four cardinal directions to correct for any tracking errors due to drift in RA or Declination drift caused by less than perfect polar alignment. This port is also used to change the WiFi Channel when needed using the supplied ST4 port Dongle.

The Computer Processor and Memory System is used to interpret and process the incoming commands and provide data output to the communications channels. There are 8 processors integrated into the microcontroller chip each with its own dedicated memory space and also memory space used to share information between the processors. There are 2 processors dedicated to communicate with each motor, read the real-time motor parameters, and adjust the motors as needed to generate the required DIRECTION and RATE values. There is one processor dedicated to communicating via the RS232 Serial Port via the port driver. There is one processor dedicated to taking the commands and interpreting them into executable instructions. There is one processor dedicated to monitoring the motor status and reading the Autoguider port. There are 2 processors for the motors used only when monitoring the status of the real-time SLEW process. The processors run independently so that if any one fails, it will not impact the operation of the others.

The commands are used to send data requests and instructions and receive data to control the motors. There are 4 basic types of commands used to interrogate the controller, 2 general purpose commands, and 2 special purpose commands. The GET and SET commands are used to get and set real-time motor parameters. These are general purpose commands to get and set a variety of parameters. The 2 special purpose commands consist of the POINT and TRACK commands. Each of the commands apply to a given motor axis defined by the AXIS value. With a telescope mount there are 2 predefined axes: AXIS 0: RA/AZ, AXIS 1: DEC/ALT. This command language can support multiple axis controllers with any number of axes.

The GET and SET commands have several primary parameters dealing with motor operation. These parameters are: DIRECTION, POSITION, RATE, and TARGET. The three main parameters, DIRECTION, POSITION, and RATE, are set statically, and are fixed until the next time they are changed with a SET command. The SET commands for DIRECTION and RATE are immediate commands and update the values in real-time. The SET POSITION command is used when adjusting the motor position coordinates when calibrating the position in reference to the celestial coordinate system, or when SYNCING with the celestial coordinates of a given object.

The GET commands allow you to interrogate the controller for various real-time parameters including: DIRECTION, POSITION, RATE, and TARGET. Other parameters that are available are FIRMWARE VERSION, SYSTEM INFORMATION via an Index value, and the current TRACKING RATE value.

The controller is designed to handle the requirement for fast slews and very slow, precise tracking rates by implementing 2 rate types, similar to a HIGH and LOW in a four-wheel drive vehicle. Standard Rate (SLEW) and Precision Track Rate (TRACK) commands are provided. The SLEW Rates are 25x faster than the TRACK Rates. The rate values can be set to any value between 0 – 40000 (decimal), 0x9C40 (hexadecimal). For the Losmandy G-11 this equates to SLEW rates up to $3.125^{\circ} \text{ sec}^{-1}$ on each axis¹. This allows the system to SLEW the telescope 180° across the sky in 60 seconds. The TRACK rate allows for setting the RA motor rate up to $450.00'' \text{ arc sec}^{-1}$ with an accuracy of $\pm 0.006'' \text{ arc sec}^{-1}$. The standard GET/SET RATE command sets the SLEW rates. The TRACK command and GET TRACKING RATE command are used to SET/GET a more precise tracking rate value. This is an equivalent rate selection range of 1:1000000

The POINT command is a higher level FIRMWARE command that automatically calculates the necessary rates used to efficiently SLEW to a TARGET. The process algorithm handles the ramping up and down of the motor rate to manage the inertial load placed on the stepper motors. Additionally, the motor current is carefully managed to provide enough torque while slewing to accurately position the telescope on the object desired without exceeding the motor capability.

The GET/SET commands for FIRMWARE VERSION, and SYSTEM INFORMATION are used to query and set various values such as communications BAUD rate, Sync Offset Positions, and Axis Scale values.

The PMC-Eight® controller can be remotely rebooted using the BOOT SYSTEM command. This will reinitialize the controller but will not reboot the wireless communications system. You can also use the BOOT COMMUNICATIONS command to remotely restart the wireless communication system. This also

¹ This angular rate is for the Losmandy G-11 mount, other mounts will slew faster or slower depending on the total motor counts for the drive.

independent of the system controller and does not reset the motor drivers or the current motor counts. To reboot the entire system, you must remove power and reapply power to the controller system.

The Get Motor Position Command returns the motor position string in Hex Value. The string is in the format "FFFFFF". The value returned is in the range 0x000000 to 0xFFFFF where positive values range from 0x000000 to 0x7FFFFF and the negative values range from 0xFFFFF to 0x800000. The scaling of the motor gear position will limit the actual values used to slew the motors. The 2 motors are on the Right Ascension Axis (RA) and the Declination (DEC) Axis of a German Equatorial Mount (GEM). The angular range for counts depends on the motor count scaling value and is calculated as follows (this example uses the Losmandy G-11).

The Losmandy G-11 uses direct coupled stepper motors where the motor scaling value only uses the wheel tooth count, the stepper motor steps, and the driver microstep value. The Losmandy G-11 uses a 360-tooth wheel, and 400 step motors. The driver uses 32 microsteps step⁻¹. So the total counts for 360-degree rotation of both the RA and DEC axis is equal to 360 teeth * 400 steps * 32 microsteps, which is equal to 4,608,000 counts per 360-degree rotation. The angular motion per step in "arc is equal to the total number of "arc/4,608,000 counts. 360 degrees is equal to 360 x 60 x 60" arc or 1,296,000" arc. The "arc count⁻¹ is equal to 1,296,000/4,608,000 or 0.28125" arc count⁻¹.

So, in the case of the Losmandy G-11, one rotation in RA, or 24 hours (sidereal time) is equal to 4,608,000 counts or in Hex, 0x465000. The RA value counts positive and negative from the Meridian+6 hours or Local Mean Sidereal Time (LMST)+6 hours. On a GEM mount, the way the telescope is pointing is an important consideration when determining the motor counts. When the GEM is tracking when the telescope gets to the Meridian a maneuver called a "Meridian Flip" is performed. To help explain this here is the sequence of events.

NOTE1: This description is only valid for mounts located in the Northern Hemisphere.

NOTE2: This description only applies to the Losmandy G-11 as the description of the motor counts increasing or decreasing is dependent on the gear direction for the mount.

1. The mount is pointing the telescope to an object in the EAST. The mount is tracking at SIDEREAL rate towards the WEST and the RA motor counts are POSITIVE and INCREASING. The RA motor counts are in the range 0 (0x000000) moving towards 1152000 (0x119400). The object the mount is pointing to is also NORTH of the ECLIPTIC or has a positive Declination value between +0.000 and +90.000 degrees. This is a fixed DEC motor count value between 0 (0x000000) and 1152000 (0x119400) when WpE. The DEC axis counts INCREASE going SOUTH from the NCP. Objects that are SOUTH of the ECLIPTIC have values between 1152000 (0x119400) and 2304000 (0x232800). These values are from -0.000 to -90.000 degrees. This pointing position is called "West Pointing East" or WpE. This means that the telescope is WEST of the PIER and pointing EAST of the MERIDIAN. In the ASCOM driver, this is referred to as "PierWEST" where the Hour Angle (HA) is less than zero (HA<0) and DECREASING in value.
2. When the mount RA axis finally arrives at the LMST value of 1152000 (0x119400) then the mount should execute a "Meridian Flip". The flip maneuver SLEWS the RA axis back to the EAST equal to approximately 12 hours of RA or 180 degrees. This means that the RA motor counts will DECREASE from 1152000 (0x119400) moving towards 0 (0x000000) and then

continue on to -1152000 (0xEE6C00). At the same time the DEC axis will SLEW to the NORTH, DECREASING the motor counts passing through the NCP while crossing 0 (0x000000) counts and continuing on down towards the object's declination motor count value between 0 (0x000000) and -2304000 (0xD80000) counts.

3. At the completion of the flip maneuver, the RA tracking begins to drive at the SIDEREAL rate towards the WEST again with the RA axis counts again INCREASING in value. The RA axis counts will be in the range from -1152000 (0x119400) moving towards 0 (0x000000). In this mode the DEC axis counts will INCREASE going NORTH from -90.000 to +90.000 degrees or from -2304000 (0xD80000) to 0 (0x000000) counts when EpW. This position, post Meridian Flip, is called "EAST Pointing WEST" or EpW. This means the telescope is EAST of the PIER and pointing WEST of the MERIDIAN. In the ASCOM driver, this is referred to as "PierEAST" where the Hour Angle (HA) is more than zero (HA>0) and INCREASING in value.

In summary, the motor count ranges for the different pointing quadrants are defined as specific values for each quadrant pointed to. The Sidereal Past is defined as object RA values that are LESS than the LMST value (RA<ST) and is WEST of the MERIDIAN. The Sidereal Future is defined as object RA values that are GREATER than the LMST value (RA>ST) and is EAST of the MERIDIAN. For example, if the LMST is currently 14:32:26.0, then 14:35:00.0 would be in the Sidereal Future, and 14:30:00.0 would be in the Sidereal Past.

When telescope is West pointing East (WpE) (target object RA is in the Sidereal Future):

1. The counterweight is East of the Pier
2. The telescope is West of the Pier
3. RA motor counts Increase tracking West: $0 < RA < 1152000$
4. DEC motor counts Increases from NCP: $0 < N < 1152000 < S < 2304000$: SCP

When WpE, LMST is at motor count value 1152000 and is the position where the counterweight bar is horizontal to the horizon.

When telescope is East pointing West (EpW) (target object RA is in the Sidereal Past):

1. The counterweight is West of the Pier
2. The telescope is East of the Pier
3. RA motor counts increase tracking West: $-1152000 < RA < 0$
4. DEC motor counts Increase from SCP: $-2304000 < S < -1152000 < N < 0$: NCP

When EpW, LMST is at motor count value -1152000 and is the position where the counterweight bar is horizontal to the horizon.

IV PMC-Eight® ASCII Command Language

The 16 base PMC-Eight® ASCII Commands allow you to control the mount in various ways to set the tracking rates, Slew the mount to a position, and various other things. Each of the command strings is ended by the "!" (bang) symbol except when entering and exiting the Diagnostic Mode. The Diagnostic Mode is used to enable the Simple Base ASCII Command Interpreter within the FIRMWARE. These commands are used internally by Explore Scientific employees for diagnosing low level motor functions.

The default Command mode enables both the JOC and ES Command Interpreters when the PMC-Eight® Controller boots. The Simple Base ASCII Commands are defined later in this document. The JOC Command Language is a BINARY based command system that uses a pre-defined header plus command and argument values. The JOC Command Language Reference will discuss that language.

Enter ASCII Diagnostic Mode:

Command: "%%%"
Response: "Diagnostic Mode ENABLED%"

NOTE: In the current version of the firmware (09T10 or later) entering the ASCII Diagnostic Mode will continuously display the analog value (Hex) for the ST4 Interface. This is used to acquire the data for calibrating the ST4 interface with a user's camera. The other low-level motor commands can be entered while the display is continuously showing the ST4 values.

Exit ASCII Diagnostic Mode:

Command: "###"
Response: "Diagnostic Mode DISABLED#"

Explore Scientific Controller Command Language Syntax

The Explore Scientific Controller Command Language is an ASCII based command system that can be parsed to determine the desired action. This is implemented in the controller FIRMWARE version ES06B9T2 and later.

NOTE1: ASCII Commands are CASE SENSITIVE. Data values are specified as Hex String Values, e.g. "FFD47A". These values are converted to Long Integers (32-bit) within the FIRMWARE. Generally, when a REQUEST command is sent, the RESPONSE includes the corresponding request syntax so that future asynchronous capabilities can be accommodated in the FIRMWARE.

NOTE2: The header "ES" is required, items in brackets {} are required, items in parenthesis () are optional depending on the command, and the "!" or shriek is required as an End-Of-Line (EOL) marker. ONLY 1 (one) of each command identifier is to be specified in each {} grouping.

Request Syntax: "ES{GSTPBR}{prtdvix} (A) (D) (XXXX) (YYYYYY) (ZZZZZZ) !"
Response Syntax: "ES{GSTPBR}{prtdvix} (A) (D) (XXXX) (YYYYYY) (ZZZZZZ) !"

Header String: "ES" Explore Scientific Language Command
Base Commands: "G" Get command
 "S" Set Command
 "T" Track Command
 "P" Point Command
 "B" Boot Command
 "R" Reset Communications Command

NOTE3: Command Parameter Codes only apply to the Get "G" and Set "S" Base Commands.

Command Parameter Code: "p" Position value
 "r" Slew Rate value
 "t" Target value
 "d" Direction value

"v" Version value
"i" System Information value

Axis Values: "0" Right Ascension / Azimuth Axis
"1" Declination / Altitude Axis

Direction Values: "0" Clockwise
"1" Counter-Clockwise

NOTE4: Axis Direction value is determined by looking at the motor from the shaft end.

NOTE5: The following System Information Values are not fully implemented in firmware versions 09t10 and previous. Only values "0x00" and 0x01" are implemented but they are not queryable in these firmware versions.

System Information Values: "0x00" WIRELESS Enabled {True=1, False=0}
"0x01" BAUD Rate {2400,9600...115200}
"0x02" AXIS 0 SCALE Counts (RA/AZ Axis) {0xFFFFFFFF}
"0x03" AXIS 1 SCALE Counts (DEC/ALT Axis) {0xFFFFFFFF}
"0x04" AXIS 0 OFFSET Counts (RA/AZ Axis) {0xFFFFFFFF}
"0x05" AXIS 0 OFFSET Counts (DEC/ALT Axis) {0xFFFFFFFF}
"06" TBD ... "0F" TBD

NOTE6: The AXIS SCALE Counts value is related to the physical configuration of the mount's gear ratio, stepper motor configuration, and controller motor driver configuration. The Losmandy G-11 mount's AXIS SCALE Count values are calculated using the following parameters: Worm/Wheel Ratio 1:360, Stepper Motor Steps/Revolution 400, Motor Attachment Gear Ratio 1:1, Motor Driver Micro-stepping ratio: 1:32. These parameters apply to both the RA and DEC axis on the Losmandy G-11 mount. The calculation for the RA and DEC AXIS SCALE Counts is: $360 \times 1 \times 400 \times 32 = 4,608,000$ (0x465000)². The mount's movement in "arc/count can be calculated by dividing 1,296,000 by the AXIS SCALE Counts. In this case the mount's resolution is (1,296,000/4,608,000)" arc count⁻¹ or 0.28125" arc count⁻¹. Since the PMC-Eight® Controller utilizes the Sidereal Second as the standard for internal hardware time-keeping, the Sidereal Tracking Rate of 15.000" arc sec⁻¹ would be equivalent to a COUNT RATE value of (15.000" arc sec⁻¹) / (0.28125" arc count⁻¹) = 53.333 counts sec⁻¹. The mean **Lunar Tracking Rate** of 14.453" arc sec⁻¹ equates to a **COUNT RATE of 51.387 counts sec⁻¹**. The mean **Solar Tracking Rate** of 15.041" arc sec⁻¹ equates to a **COUNT RATE of 53.479 counts sec⁻¹**.

² See Table for the count calculation parameters for each of the Explore Scientific mounts.

Explore Scientific Mount Stepper Motor Drive Parameters

Mount	UNITS	Losmandy G-11	EXOS I	EXOS II	EXOS III
Total Counts	micro-steps	4608000	1792000	4147200	4147200
Motor Counts	steps/revolution	400	400	200	400
Motor Model	number	17HM19-1684S	17HM19-1684S	11HS20-0674S	17HM19-1684S
Motor Frame	type	NEMA 17	NEMA 17	NEMA 11	NEMA 17
Primary Gear Teeth	count	360	140	144	144
Secondary Gear Ratio	NA	1	1	4.5	2.25
Motor Current	mA	900	900	600	900
Sidereal Tracking Rate	micro-steps/second	53.333	20.741	48.000	48.000
Motor Scale	arc-seconds/micro-step	0.28125	0.72321	0.31250	0.31250
Maximum Step Frequency	Hz	40000	40000	40000	40000
Maximum Slew Rate	°/second	3.125	8.036	3.472	3.472

The following tables list the command language and the command request/response syntax. A description of all the commands is listed here for your reference and usage. Pay close attention to the command notes for any special requirements or restrictions on the command's usage.

List of PMC-Eight® ASCII Commands

GET Command	Name	Parameters
"ESGp!"	Get Axis Current Position	Axis
"ESGr!"	Get Axis Current Rate	Axis
"ESGt!"	Get Axis Current Target	Axis
"ESGd!"	Get Axis Current Direction	Axis
"ESGv!"	Get System Firmware Version	None
"ESGi!"	Get System Information	Value Index
SET Command	Name	Parameters
"ESSp!"	Set Axis Current Position	Axis, Position
"ESSr!"	Set Axis Current Rate	Axis, Rate
"ESSd!"	Set Axis Current Direction	Axis, Direction
"ESSi!"	Set System Information	Value Index
TRACK Command	Name	Parameters
"ESTr!"	Set Precision Tracking Rate	Axis, Rate
POINT Command	Name	Parameters
"ESPt!"	Point to Target Position	Axis, Position
SWAP Command	Name	Parameters
"ESX!"	Swap Primary Serial Interface	None
RESET Command	Name	Parameters
"ESR!"	Reset WiFi Comm Controller	None
BOOT Command	Name	Parameters
"ESB!"	Reboot Motor Controller	None

The firmware code modules have the following information as a ready reference to the command syntax.

NOTE: The firmware (version 09T10) source code is proprietary and is not available for inspection.

```
-----
'
' Explore Scientific Command Language
' See programmers reference manual for details on command syntax
'
' ES Internal Command Return Strings
'-----
' Internally, HEX Return Strings are used for the ES commands to separate out the
' different ones. HEX values from 0x00 to 0x3F are used for the Explore Scientific
' commands. Items in {} are required, items in () are optional depending on the
' command. A=axis, D=direction, XXXX=rate, YYYYYY=position, ZZZZZZZZ=parameter value
'
' ES{G,S,T,P,B,X}{p,r,t,d,v,i,x}(A)(D)(XXXX)(YYYYYY)(ZZZZZZZZ)!
'
' G_! 0x00      FUTURE                               ESG_!
' Gp! 0x01      get current axis position value       ESGpA!
' Gr! 0x02      get current axis rate value           ESGrA!
' Gt! 0x03      get current axis target value         ESGtA!
' Gd! 0x04      get current axis direction value      ESGdA!
' Gv! 0x05      get current firmware version          ESGv!
' Gi! 0x06      get current system information         ESGiZ!
' Sp! 0x11      set axis position value               ESSpAYYYYYY!
' Sr! 0x12      set axis slew rate value              ESSrAXXXX!    (Allows full range of rates)
' Sd! 0x14      set axis direction value               ESSdAD!
' Si! 0x16      set system information                 ESSiAZZZZZZZ! (A is the parameter index)
' Tr! 0x22      set tracking rate value                ESTrXXXX! (Track rate only applies to RA)
' Pt! 0x31      point (slew) to target using ramps    ESPtAYYYYYY!
' X!  0xFD      Switch Communications Mode (WIFI/SER) ESX!
' R!  0xFE      Reset Communications Controller       ESR!
' B!  0xFF      Boot PMC-Eight Controller            ESB!
'-----
```

V PMC-Eight® ASCII Command Language Syntax

“G” GET Commands:

“ESGp” GET AXIS CURRENT POSITION VALUE

```
REQUEST:  "ESGpA!"
RESPONSE: "ESGpAYYYYYY!"
```

Example: Get Current RA Axis Position Value

```
REQUEST:  "ESGp0!"
RESPONSE: "ESGp0FF37DA!"
```

NOTE: Data values are specified as Hex String Values. The response value can be

“ESGr” GET AXIS CURRENT RATE VALUE

```
REQUEST:  "ESGrA!"
RESPONSE: "ESGrAXXXX!"
```

Example: Get Current RA Axis Rate Value

```
REQUEST:  "ESGr0!"
RESPONSE: "ESGr037DA!"
```

NOTE: Data values are specified as Hex String Values.

"ESGt" GET AXIS CURRENT TARGET VALUE

REQUEST: "ESGtA!"
RESPONSE: "ESGtAYYYYYY!"

Example: Get Current RA Axis Target Value

REQUEST: "ESGt0!"
RESPONSE: "ESGt062E4D7!"

NOTE: Data values are specified as Hex String Values

"ESGd" GET AXIS CURRENT DIRECTION VALUE

REQUEST: "ESGdA!"
RESPONSE: "ESGdAD!"

Example: Get Current RA Axis Direction Value

REQUEST: "ESGd0!"
RESPONSE: "ESGd01!"

NOTE: Data values are specified as Hex String Values

"ESGv" GET CURRENT FIRMWARE VERSION VALUE

REQUEST: "ESGv!"
RESPONSE: "ESGvZZZZZZZZ!"

Example: Get Current FIRMWARE Version Value

REQUEST: "ESGv!"
RESPONSE: "ESGvES6B09U0!" (version ES6B09U0)

NOTE1: Data values are specified as character string values.

NOTE2: The returned string is a combination of the hardware version "6B" and the firmware version "09U0". This is currently hard coded in the firmware for each firmware release.

"ESGi" GET CURRENT SYSTEM INFORMATION VALUE

NOTE: Not Implemented in this version.

REQUEST: "ESGiX!"
RESPONSE: "ESGiXZZZZZZZZ!"

Example: Get Current System Information 1 (BAUD Rate) Value

REQUEST: "ESGi1!"
RESPONSE: "ESGi10001C200!" (decimal 115,200 BAUD)

Example: Get Current System Information 03 (DEC Maximum) Value

REQUEST: "ESGi3!"
RESPONSE: "ESGi300465000!" (decimal 4,608,000 DEC MAX)

NOTE: Data values are specified as Hex String Values.

“ESGx” GET CURRENT PRECISION TRACKING RATE VALUE

REQUEST: “ESGx!”
RESPONSE: “ESGxXXXX!”

Example: Get Current Precision Tracking Rate Value

REQUEST: “ESGx!”
RESPONSE: “ESGx0535!”

NOTE1: Data values are specified as Hex String Values.

NOTE2: The Get Tracking Rate value only applies to Axis 0, or the RA Axis.

The Precision Tracking value is calculated by multiplying the nominal calculated Rate Counts (for Losmandy G-11) by 25, i.e.,
Rate Counts = rate in “arc sec⁻¹/0.28125” arc count⁻¹

The calculated precision tracking value for a sidereal rate of 15.000” sec⁻¹ sidereal is:
Precision Tracking value = (15.000 / 0.28125) * 25
= 53.333 * 25
= 1333 (decimal)
= 0x0535 (hexadecimal)

The Tracking Rate value is accurate to ±0.02 counts sec⁻¹ so the resulting count rate is equal to 53.32 counts sec⁻¹. This command will accommodate precision count rates from 0.00 to 2621.44 counts sec⁻¹ which is equivalent to precision rate values of 0 to 7.5’ arc sec⁻¹ or about 0.1° sec⁻¹.

“S” SET Commands:

The RESPONSE to the SET Commands in general mimic the value returned when using the GET Commands. The RESPONSE syntax will be identical to the corresponding GET Command. This is implemented this way in order to simplify the coding of any function that is written to process the RESPONSE string.

“ESSp” SET AXIS POSITION VALUE

REQUEST: “ESSpAYYYYYY!”
RESPONSE: “ESGpAYYYYYY!”

Example: Set RA Axis Position Value

REQUEST: “ESSp0FF37DA!”
RESPONSE: “ESGp0FF37DA!”

NOTE1: Data values are specified as Hex String Values.

NOTE2: Setting the Axis Position Value UPDATES it to a NEW value on the fly without slewing the mount in any way. This can be used to recalibrate the axis pointing position when syncing the axis to an external reference point provided by a reference source such as a star catalog or planetarium program. The Axis Position Value is only set to 0 (zero) when the axis is in its PARK position. When used in the ASCOM Telescope driver, the Position value is converted back and forth between the Motor Count Hex Value and the Degrees/Minutes/Seconds System. The defined PARK position for a German Equatorial Mount (GEM) is the North Celestial Pole (NCP).

This position NCP position has coordinates: RA LST (07:43:22.00) + 06:00:00.00, DEC +90d 00m 00.0s.

"ESSr" SET AXIS TRACKING RATE VALUE

REQUEST: "ESSrAXXX!"
RESPONSE: "ESGrAXXX!"

Example: Set Current RA Axis Rate Value

REQUEST: "ESSr037DA!"
RESPONSE: "ESGr037DA!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: Setting the Axis Tracking Rate Value UPDATES it to a NEW value on the fly *without initiating a ramp to rate process*. This can be used to adjust the Tracking Rate for Sidereal, Lunar, Solar, and any other object that you may wish to track (within reason). The nominal Sidereal Rate for the RA Axis is 53.333 counts sec⁻¹ for the Losmandy G-11. The nominal Lunar Rate for the RA Axis is 51.387 counts sec⁻¹. To improve the resolution for setting the Axis Rate, the floating point value for counts per second needs to be multiplied by 25 and rounded to the nearest integer to develop the value used in sending the REQUEST, e.g. Sidereal Rate = 53.333 Counts sec⁻¹, 25 x 53.333 = 1333.325 or 1333 integer. Finally convert the value 1333 Rate to Hex: 0x0535.

"ESSt" SET AXIS TARGET VALUE

REQUEST: "ESStAYYYYY!"
RESPONSE: "ESGtAYYYYY!"

Example: Set Current RA Axis Target Value

REQUEST: "ESSt062E4D7!"
RESPONSE: "ESGt062E4D7!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: Setting the Axis Target Value UPDATES it to a NEW value on the fly and *initiates a ramping rate process*. (Ed: This needs to be tested). This can be used to reposition the Target in the middle of a POINT command. When used in the ASCOM Telescope driver, the Target value is converted back and forth between the Motor Count Hex Value and the Degrees/Minutes/Seconds System. See the Supplemental Command Discussion section for details.

"ESSd" SET AXIS DIRECTION VALUE

REQUEST: "ESSdAX!"
RESPONSE: "ESGdAX!"

Example: Set Current RA Axis Target Value

REQUEST: "ESSd01!"
RESPONSE: "ESGd01!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: The direction values as looking at the shaft end of the motor are 0x0 for clockwise, 0x1 for counter-clockwise.

NOTE3: Setting the Axis Direction Value UPDATES it to a NEW value on the fly. In general, this should only be done when the motor RATE value is zero or close to zero as it puts inertial stress on the motor and gear system of the mount and may result in lost motor counts and slipping of the motor position.

"ESSi" SET CURRENT SYSTEM INFORMATION VALUE

NOTE: Not Implemented in versions 09T10 and previous.

REQUEST: "ESSiXZZZZZZZZ!"

RESPONSE: "ESGiXZZZZZZZZ!"

Example: Set Current System Information #1 (BAUD Rate) Value

REQUEST: "ESSi10001C200!"

RESPONSE: "ESGi10001C200!" (decimal 115,200 BAUD)

Example: Set Current System Information #3 (DEC Maximum) Value

REQUEST: "ESGi300465000!"

RESPONSE: "ESGi300465000!" (decimal 4,608,000 DEC MAX)

NOTE: Data values are specified as Hex String Values.

"T" TRACK Command:

"ESTr" SET PRECISION TRACKING RATE VALUE

REQUEST: "ESTrXXXX!"

RESPONSE: "ESGxXXXX!"

Example: Set Precision Tracking Rate Value

REQUEST: "ESTr0535!"

RESPONSE: "ESGx0535!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: The Set Tracking Rate value only applies to Axis 0, or the RA Axis. The Precision Tracking value is calculated by multiplying the nominal calculated Rate Counts (for Losmandy G-11) by 25, i.e.,

$$\text{Rate Counts} = \text{rate in "arc sec}^{-1}\text{/0.28125"} \text{ arc count}^{-1}$$

The calculated precision tracking value for a sidereal rate of 15.000" sec⁻¹ sidereal is:

$$\begin{aligned} \text{Precision Tracking value} &= (15.000 / 0.28125) * 25 \\ &= 53.333 * 25 \\ &= 1333 \text{ (decimal)} \\ &= 0x0535 \text{ (hexadecimal)} \end{aligned}$$

The Tracking Rate value is accurate to ±0.02 counts sec⁻¹ so the resulting count rate is equal to 53.32 counts sec⁻¹. This command will accommodate precision count rates from 0.00 to 2621.44 counts sec⁻¹ which is equivalent to precision rate values of 0 to 7.5' arc sec⁻¹ or about 0.1° sec⁻¹.

"P" POINT Command:

"ESPt" SET POINT TARGET VALUE

REQUEST: "ESPtAYYYYYY!"
RESPONSE: "ESGtAYYYYYY!"

Example: Set Current RA Axis Target Value

REQUEST: "ESPt006FAE4!"
RESPONSE: "ESGt006FAE4!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: The Point Command is used to slew the mount from the current target to a new target specified by the YYYYYY value. The command returns the target value immediately. You must use the Get Motor Position Value command "ESGp0!", "ESGp1!" while the mount is slewing to get the current position of the mount. The controller automatically ramps the slew rate up to the maximum rate commensurate with the distance between the current position and the target position. The controller also calculates the start of the ramp down to stop the mount movement when it reaches the target value. Both axes may be slewed concurrently as desired as these are independent processes.

"X" Swap Command:

"ESX" Swap Primary Interface

REQUEST: "ESX!"
RESPONSE: "ESXY!"

NOTE1: Data values are specified as Hex String Values.

NOTE2: The swap command will toggle the primary interface between the programmer's hard-wired serial port and the WiFi network serial interface. The response value "Y" is either "0" or "1" denoting the port. Port "0" is the programmer's RS232 port and Port "1" is the WiFi wireless port.

"R" RESET Command:

"ESR" Reset Communications Controller

REQUEST: "ESR!"
RESPONSE: "ESR!"

NOTE1: The communications reset command will restart the wireless communications controller. The system controller will be maintained while the wireless communications controller restarts.

"B" BOOT Command:

"ESB" Reboot System Controller

REQUEST: "ESB!"
RESPONSE: "ESB!"

NOTE1: The system reboot command will restart the controller independent of the wireless communications controller. The wireless communications will be maintained while the motor controller restarts.

VI PMC-Eight® Interface Specifications

The PMC-Eight® controller has several interfaces for power, motors, and communications. The system is capable of communicating via the wireless Wi-Fi interface and through the RS232 DB-9 interface. The power connection uses a standard 2.1mm barrel type connector. The RA and DEC motors connect via a DB-9 connections that are gender specific for each motor so that they cannot be connected wrong. The Autoguider Port uses a standard RJ connector of type 6P6C wired as per the SBIG ST-4 standard for contact inputs for each direction, N, S, E, and W. The PMC-Eight® controller limits the motor current provided to each motor to maximize torque while slewing and minimizes the current to extend the battery life. The controller requires a minimum of 12 VDC @ 2 ADC power supply for proper operation. The controller current draw is approximately 800 mA while power up and tracking. The controller current draw is approximately 1000 mA while slewing both axes.

The PMC-Eight® controller communications BAUD rate default is 115,200 BAUD which provides near instantaneous response to incoming commands. The WiFi module used is IEEE 802.11 compliant for use in consumer equipment.

VII Miscellaneous Programming Application Notes

The PMC-Eight® provides a very robust and flexible platform for building custom applications to control and automate the operation of your telescope mount. Through the use of open source OpenGOTO™ Community applications the performance and full range of operations can be realized. Remote operation of the system is also possible through the use of the wireless Wi-Fi network interface.


The OpenGOTO™ Explore Stars application for Microsoft Windows™ Tablets, Laptops, and PCs is the perfect “first app” for exploring the operation of the PMC-Eight® System. The Explore Star App is available at the Microsoft Windows App Store. Example source code is available written in Microsoft C# for use when developing your own software for the PMC-Eight® System. The example code can be compiled and installed on any Windows based system. The Explore Stars app uses the JOC command language used in the JOC GOTO controller also and can be used with some work to control the EXOS II with the JOC GOTO controller.

The open source ASCOM driver is built using the ASCOM-Standards Platform 6.2 templates for .NET programs and all the source code is available along with the driver for those who wish to use it for their client applications. The Explore Scientific PMC-Eight® ASCOM driver has been testing using the ASCOM Platform Conformance tool. The driver has been testing successfully using the ASCOM Platform client applications and also MaximDL and Cartes du Ciel. The driver is a work in progress and will continue to be updated and the performance upgraded as time goes on. You can download the ASCOM platform and find the link to the Explore Scientific PMC-Eight® ASCOM driver at ASCOM-Standards.org.



EXPLORE[®]
SCIENTIFIC

Designed by Explore Scientific® in Arkansas. Assembled in China.
Model: ESOGT-01B
©2016 Explore Scientific®, LLC. All rights reserved. Made in China.
explorescientificusa.com - 866.252.3811

CE FC  Contains FCC ID: U30-G2M5477

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.