GreenEye Monitor

For GEM model: **All**

Document Code: **CMD**
Ver 10.0
2013-03-09

TABLE OF CONTENTS

GEM COMMANDS, PACKET INFO AND API

COMMANDS

ESCAPE SEQUENCE

All GreenEye commands are executed beginning with the escape sequence "^^^". Once the GEM receives these three characters, it will pause normal operation and listen for a valid command sequence.

The quotations are not included in the command.

The brackets "<" and ">" used in this document are not part of the command. They are used to indicate where a given variable is inserted. Many of the variables are specified by the "x" character, each "x" value representing one character.

COMMAND CLASSIFICATIONS

These commands are broken into groups:

SYS system
RQS request
Cxx channel
TMP temperature
PLS pulse counter
API data request

System "**SYS**" commands are used to read or modify parameters which apply to the entire system.

Request "**RQS**" commands provide the ability to read the current settings.

Channel "**Cxx**" commands allow setup of parameters relating to any of the 32(48) power monitoring channels.

Temperature "**TMP**" commands are used for setting up the temperature sensor parameters.

Pulse "**PLS**" commands are used to setup the pulse counter parameters.

API data "**API**" gets monitored data. Typically used for "polling" data when realtime packets are not used.

SYSTEM COMMANDS "SYS"

All System Commands begin with the following sequence: **"^^^SYS"**
NOTE:
- Quotes are not included
- Characters are cases sensitive

"_ON"	RealTime "ON"	Causes the GEM to start sending real-time packets based on the user selected packet format and interval. Ex: Sending "^^^SYS_ON" GEM responds with _ON and begins sending the selected packet format after the packet send interval time has elapsed.
"OFF"	RealTime "OFF"	Causes the GEM to stop sending real-time. If you are sending this command remotely, you should enable the "keep alive" heartbeat so that the GEM doesn't lose TCP/IP connection with the server. Ex: Sending "^^^SYSOFF" GEM responds with OFF
"KAI"<*x*>	Keep-Alive Enable and Interval	X = 0 to 7 0= Disable Keep=Alive 1 = every 4 seconds 2 = every 8 seconds 3 = every 16 seconds 4 = every 32 seconds 5 = every 64 seconds 6 = every 128 seconds 7 = every 256 seconds Keep-Alive is functional only if Real-Time is off. The "keep-alive" heartbeat is used to maintain a TCP/IP connection when using API method of collecting data from the GEM when the GEM is configured as TCP/IP client. The keep-alive string is configured by the KAS command and has a maximum size of 8 characters. KAI<x> is responded with OK or ER
"KAS"<*string*><CR>	Keep-Alive String	<*string*> is a set of one to eight characters which is sent as the "keep-alive" text. The "keep-alive" heartbeat is used to maintain a TCP/IP connection when using API method of collecting data from the GEM when the GEM is configured as TCP/IP client. It is configured using the KAI command.

"PKT"<_xx_>	Set Packet Format	xx = 00 to 99 The value sent must be two numeric characters long. This means that a value such as "2" should be sent like this: "^^^SYSPKT02" Notice the leading zero padding. GEM responds with "PKT" if executed. The packet formats are described in the GEM's Packet Format manual.
"PKF"<xx>	Set Secondary Packet Format	<xx> is "00" to "14" The secondary packet format is disabled by setting it to "00" Ex: "^^^SYSPKF00" command turns off the secondary packet format. As of COM firmware version 1.95 the GEM can send a different packet format to each of its two Com ports. The primary port (Com1) packet format is determined by the "PKT" command as described above. The new "PKF" applies only to the packet format sent via Com2. Only certain packet formats may be specified as the secondary format. These formats are those that do not require HTTP POST, GET or PUT. If an invalid packet format is specified as the secondary packet, the GEM will respond with an "ER" when the command is sent. Currently the secondary packet interval is the same as that of the primary packet. The primary packet is sent to COM1 first then the secondary packet is sent to Com2 immediately after. When the secondary packet is disabled, the primary packet will be sent to both Com ports simultaneously.
"IVL" <xxx>	Set Packet Send Interval	xxx = 001 to 256 This must be three numeric characters long. GEM responds with "IVL" once executed. The value represents the number of seconds between each packet send. Realtime must be enabled (_ON) for the packets to be sent.
"PKS"<val><CR>	Set Packet Chunk Size	<val> is a numeric value between 80 and 65000 A carriage return (CR) is required at the end of the command. This parameter sets the number of bytes to be sent in each chunk. Since the GEM's packet size can be fairly large, the packet sometimes need to be broken into small sections (chunks) with slight pauses (milliseconds) between each transmission in order not to surpass the

		throughput of certain communicating devices. Ex: ^^^PKS240<CR> will cause the packet data to be sent in chunks of 240 bytes. The GEM responds with PKS
"PKI"<val><CR>	Set Packet Chunk Interval Time	<val> is a numeric value between 16 and 65000 A carriage return (CR) is required at the end of the command. The GEM responds with PKI
"PDL"	Delays next packet send by 15 seconds	This command produces no response! This command is used to prevent command responses from being mixed in with a packet if a command is issued while the packet is mid-stream. Sending this command a couple of seconds before issuing the desired command allows the GEM to complete the current packet if one is in process. The GEM will then refrain from sending a packet for 15 seconds. *Added to firmware version 2.12*
"BFF"<val><CR>	Set Max Buffer Size	<val> is a numeric value between 10 and 1700 A carriage return (CR) is required at the end of the command. The GEM responds with BFF
"BD1"<x>	Set Com1 Baud Rate	<x> = "0" for 19200 baud or "1" for 115,200 baud Beginning with COM firmware version 2.0, the standard baud rate for the GEM was increased from 19,200 to 115,200 baud. This greatly improves the performance of the web setup interface. Unfortunately the bootloader which resides in the processor is set to operate at 19,200 baud. Because of this, older GEM units require the baud rate to be reduced for firmware upgrades, then increased afterwards. Changing the GEM's baud rate also requires changing the baud rate of the communication module. This requires more time since the module must reboot after the baud rate change. Depending on the communication module, this can take anywhere between 10 to 32 seconds after the command is issued.
"BD2"<x>	Com2 Baud Rate	Currently Com2 baud rate remains set to 19,200 and <x> should always be "0".

"FL1"<x>	Enable Wifi Hardware Flow Control Affecting Packet Send Requires RN-171 firmware version 1.44 or greater	<x> = 0 for disabled or 1 for enabled. Make sure RN-171 is set for "set uart flow 1". Option 2 uses the RTS line. This option is implemented to prevent buffer issues with the Wifly module. The GEM responds with FL1 OR "Hardware Flt" which signifies that the GEM is not configured for Wifi module.
"FL2"<x>	Enable Zigbee Flow Control (COM2).	<x> = 0 for disabled or 1 for enabled. Make sure that XBee flow control is enabled ATD71 (CTS). This option applies to realtime packet data transfer from the process buffer only. Commands are issued regardless of flow status. This flow control also stops data from being sent to the Wifi module or other COM1 communication module The GEM responds with FL2 OR "Hardware Flt" which signifies that the GEM is not configured for XBee module.
"TPT" <xxx>	PT Type	xxx= 001 to 255 Value must be three numeric characters long. GEM responds with "TPT" This value is a constant based on the Potential Transformer's (PT) voltage ratio. A value for this constant is chosen so that the GEM displays the proper AC line voltage. If the GEM displays 116V when a multimeter measures 118V at the power line, this value may be increased slightly to cause the GEM to display 118V. See manual for more information.
"VRG" <xx>	PT Range	xx= 00 to 10 Value must be two numeric characters long. GEM responds with "VRG" This parameter works in conjunction with the "PT Type" described above. The interpreted voltage is doubled every time the value is reduced by one. For example: Range = 3 Displayed voltage = 117.0 V Range = 2 Displayed voltage = 234.0 V

		Refer to the GEM User's Manual for more details.
"RSTH"	Reset Wh Counters and Wh minutes (firmware 1.86 or greater)	Zeroes Watt-Hour counters and Wh elapsed minutes. GEM responds with "RSTH" after execution.
"RSP"<x>	Reset a Specific Pulse Counter	x = 1 to 4 <x> is a numeric value representing the pulse counter channel to be reset. The reset value is zero. Each counter channel is 3-bytes long, therefore can count to 16,777,215 pulses. GEM responds with "RSP" if executed.
"RSTP"	Reset All Pulse Counters	Resets all four pulse channel counters to zero. GEM responds with "RSTP" if executed.
"RSTA"	Reset All Counters	Zeroes all counters: Channel 1 to 48 watt-second All pulse counters Seconds counter GEM responds with "RSTA" after execution.
"RSTS"	Reset Seconds Counter	Zeroes the seconds incremental counter. This counter increments every second as long as the system is monitoring. GEM responds with "RSTS" It is used along with watt-second values to calculate Watt. Refer to GEM manual.
"RSTC"<xx>	Reset the Watt-Second counter for a specified channel.	xx = 01 to 48 <xx> must be a two character value. GEM responds with "RSTC" This command zeroes the watt-second counter for the specified channel. Absolute and Polarized counters are both reset.
"HZ"<x>	Set the Line Frequency to 50Hz or 60Hz	<x> = "0" (60Hz) OR "1" (50Hz) GEM responds with "HZ"

"DTM"<yy,mm,dd,hh,mm,ss><CR>	Set Date and Time	Set the date and time of the GEM. For the most part, this date clock is not required. It is there for future use. Ex: "DTM12,08,23,13,30,28"<CR> will set the GEM for August 23rd, 2012 1:30:28 pm GEM responds with "DTM"
Reserved	Bootload COM processor	
Reserved	Bootload Engine processor	Wait 500 ms after B87 response
"URL"<url><CR>	Set the HTTP post URL	<url> is the URL text such as http://mysite.com The last byte must be a CR to indicate the end to the URL data. The URL text is limited to 63 characters. GEM responds with "URL"
"URE"<url ext><CR>	Set the HTTP post URL extension	<url ext> is the URL extension text such as /mycode.php The last byte must be a CR to indicate the end to the URL extension data. The URL extension text is limited to 63 characters.

		GEM responds with "URE"
"URT"<token><CR>	Set Data Host Token/Key1	<token> is the assigned token from the data hosting site. Max 63 characters. The last byte must be a CR to indicate the end the token string. The token string is limited to 63 characters (not including the CR). GEM responds with "URT"
"URK"<x><key><CR>	Set Additional Post Keys	<x> is the key number from 2 to 8 <key> is the assigned key from the data hosting site. Max 31 characters. The last byte must be a CR to indicate the end the token string. The token string is limited to 63 characters (not including the CR). GEM responds with "URK"
"URN"<node><CR>	Set Data Host Device Node	<node> is the assigned token from the data hosting site. Max 63 characters. The last byte must be a CR to indicate the end the token string. The token string is limited to 63 characters (not including the CR). GEM responds with "URN"
"ZBR"	Reset XBee Module	
"RSW"	Reset Wifi Module	
"WAR"<x><CR>	Wifi Auto Reset Interval	Automatically sends a reset pulse to Wifi module after "x" seconds if data has not been received from COM1 port. If x=0 then this feature is disabled Valid Range: 30 to 255 seconds

		If a value between 1 and 29 is entered, the value of 30 will automatically be used. The last byte must be a CR to indicate the end. This feature has been implemented to deal with issues of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. Refer to GEM manual for more information. GEM responds with "WAR"
"WRR"<x><CR>	Wifi No-HTTP-Response Reset Interval	Automatically sends a reset pulse to Wifi module after "x" number of missing response after sending an HTTP "GET", "POST" or "PUT. Used in cases where the Wifi module may have locked up. Valid Range: 2 to 255 missing responses The last byte must be a CR to indicate the end. This feature has been implemented to deal with issues inherent of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. This method is preferred over the "WAR" method when sending packet data using HTTP GET or PUT method. Refer to GEM manual for more information. GEM responds with "WRR"
"ZAR"<x><CR>	ZigBee Auto Reset Interval	Automatically sends a reset pulse to Wifi module after "x" seconds if data has not been received from COM1 port. If x=0 then this feature is disabled Valid Range: 30 to 255 seconds If a value between 1 and 29 is entered, the value of 30 will automatically be used. The last byte must be a CR to indicate the end. This feature has been implemented to deal with issues of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. Refer to GEM manual for more information.

		GEM responds with "WAR"
"ZRR"<x><CR>	ZigBee No-HTTP-Response Reset Interval	Automatically sends a reset pulse to Wifi module after "x" number of missing response after sending an HTTP "GET", "POST" or "PUT". Used in cases where the Wifi module may have locked up. Valid Range: 2 to 255 missing responses The last byte must be a CR to indicate the end. This feature has been implemented to deal with issues inherent of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. This method is preferred over the "WAR" method when sending packet data using HTTP GET or PUT method. Refer to GEM manual for more information. GEM responds with "WRR"
"HDW"<xx>	Set Installed Hardware/Module	<xx> first "x" represents modules on Com1 and second "x" represents XBee module on Com2 **First "x" value** 0 = No modules installed 1 = Wifi only module (RN-171) 2 = Old Ethernet module with ribbon cable 3 = New Ethernet module with connector PCB 4 = Wifi/Ethernet combination mdule **Second "x" value** 0 = No XBee module 1 = XBee module installed **Example:** **"14"** = Wifi/Ethernet combo module and XBee module installed. Informs GEM's processor as to which hardware communication module(s) is installed. If this value is not configured, the green system LED will flash every second.

		This setting is important to reduce battery drain during power outages. GEM responds with "HDW"
"WBR"	Wifi Baud Reset to 19200	
"RBC"	Re-Boot COM Processor	This command restarts the main processor. Used for debugging. GEM responds with "RBC"
"RBE"	Re-Boot Engine1 Processor	Resets the power engine processor. Generally not required. GEM responds with "RBE"
"PH"<x>	Master Polyphase ON/OFF Setting (New 09/10/12) Requires ENG firmware ver 1.37 or higher.	X = 0 Polyphase is OFF all phases assume phase A regardless of phase settings. X= 1 Polyphase is ON all channels are assigned the user defined phase setting. X= 2 Three-phase delta with 30degree offset X = 3 Single Phase with CT phase compensation based on CT setting GEM responds with "PHx" where "x" is the selected value
"CLC"	DO NOT USE! Experimental mode. Calibrate Clock Calender	**** Requires PT signal to be connected.
"EPC"<xxx>,<xxx>,<xxx>, **<xxx>,<xxx>,<xxx><CR>**	Enable Channels to be included in the HTTP PUT packet	xxx = 000 to 255 Six bit mapped bytes sets which of the 48 channels will be sent with the HTTP post. First value = channels 1 to 8 Second value = channels 9 to 16 …… and so on Ex: EPC007,001,128,000,001,000<CR> 007(1-8): chan 1,2 and 3 are enabled 001(8-16): chan 9 enabled 128(17-24): chan 24 enabled 000(25-32): none enabled 001(33-40): chan 33 enabled 000(41-48): none enabled

"EPP"<xxx><CR>	Enable Pulse Counters included in the HTTP PUT packet	xxx = 000 to 015 Single bit mapped bytes sets which of the 4 pulse counting channels will be sent with the HTTP post. Counter1 = bit0 Counter2 = bit1 Counter3 = bit3 Counter4 = bit4 Ex: value 004 = CNTR 3 enabled for POST
"EPT"<xxx><CR>	Enable Post Temperature	xxx = 000 to 255 Single bit mapped bytes sets which of the 8 temperature channels will be sent with the HTTP post. Temperature 1 = bit0 Temperature 2 = bit1 Temperature 3 = bit2 Temperature 4 = bit3 Temperature 5 = bit4 Temperature 6 = bit5 Temperature 7 = bit6 Temperature 8 = bit7 Ex: value 017 = T1 and T5 enabled for POST
"CMX"<xx><CR>	Set Max Number of Channels To Be Used	X = 01 to 48 (two characters) Prevents from processing un-used channels.
"TSB"<x>	Set Number of ThingSpeak Block	<x> is the number of blocks from 1 to 8 Each TS blocks contain 8 channels. Each block has its own unique post "Key". The first key (block1) resides in "Token/Key1". Subsequent keys reside in Key2-8.
"GSN"	Read and Store SN from Bootload (Get Serial Number)	
"LU1"	Load User1 WiFly File	Adhoc

"LU2"	Load User2 WiFly File	Host
"BR1"	Enter Bridge Mode	Links COM1 directly to COM2 and is baud independent. The serial port or module connected to COM1 has a crossover connection directly to any device on COM2 or COM2's serial port.

The GEM pauses regular processes while in this mode. The date-time clock is also not functional at this time, therefore if the realtime clock is used for any reason, it should be updated after exiting bridge mode.

This function is very handy for accessing installed communication devices via a serial port or other communication module. This mode is also useful for upgrading firmware or configuring an XBee module (for example) using COM1 serial port.

A special sequence is required to exit bridge mode. |
| NOTE:

Do not send "^^^" <SYS>

Only send the sequence described in the paragraph (right) | Exit Bridge Mode

NOTE: Bridge mode may also be exited by pressing the GEM's PB briefly. | Since the bridge1 mode is baud independent, the command to exit from this mode must also be baud independent. For this reason, a timed sequence of any single character is sent to exit this mode.

When in bridge mode, the System LED will be ??? . To exit bridge1 mode, send the following sequence:

<any single character> pause 1.75 second
<any single character> pause 1.75 second
<any single character> pause 750 millisecond
<any single character> pause 750 millisecond
<any single character> |
| | | |
| "HF1" | Preset HF module | Work in progress. |
| "FP"<x> | Factory Preset | <x> is 0-3
0 = no com modules installed
1 = Wifi module installed
2 = XBee module installed
3 = Both modules installed
4 = Wiznet Only on COM1

Presets all default values and sets up the hardware settings including flow control. |
| | | |

CHANNEL COMMANDS "CXX"

INDIVIDUAL CHANNEL SETTINGS

These commands affect only the specified channel. One command is required for each channel.

All individual channel settings begin with the following sequence: "^^^" "Cxx", where "xx" is the channel number 01 to 48.

TYP\<xxx\>	Set channel CT Type	xxx = 001 to 255		
RNG\<x\>	Set channel CT Range	x = 0 to 9		
PH\<x\>	Set channel Phase	x = A or B or C		
PC\<x\>	Set channel Phase Compensation (Added Sept 10,2012) Requires ENG firmware version 1.37 or higher	X = 0,1,2 0 = no phase compensation 1 = 4 degrees compensation 2 = 8 degrees compensation		
PL\<p\>	Change channel polarity bit *(revised Sept 18, 2012)* Requires ENG firmware version 1.37 or higher	\<p\> is "0" or "1" polarity \<xx\> is two character channel number		
TOG	Toggle channel polarity *(revised Sept 18, 2012)* Requires ENG firmware version 1.37 or higher	\<p\> is "0" or "1" polarity \<xx\> is two character channel number		

ALL CHANNEL SETTINGS

Sets all channels to a common setting defined by a single value.

The "All Channel" settings begin with the following sequence: "^^^" "CHE".

TYP <xxx>	Set All channel CT Type	xxx = 001 to 255
RNG <x>	Set All channel CT Range	x = 0 to 9
PHx	Set All channel Phase	x = A or B or C
POL0	Set All Channel Pol 0 (default)	
POL1	Set All Channel Pol 1	

GROUP CHANNEL SETTING

Set all channels to an individually defined value. All values are sent in a single block.

The "Group Channel" settings begin with the following sequence: "^^^" "CHG".

TYP <xxx>,<xxx>, <xxx>, ... <CR>	Set CH1 to CH48 CT Type	xxx = 001 to 255 Send 48 comma delimited 3-character values corresponding to the desired CT type for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER"
RNG <x>,<x>,<x>, ... <CR>	Set All channel CT Range	x = 0 to 9 Send 48 comma delimited single character values corresponding to the desired CT range for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER"
PH<x>,<x>,<x>, ... <CR>	Set All channel Phase	x = A or B or C Send 48 comma delimited single character values corresponding to the desired CT phase for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER"

TEMPERATURE SENSOR COMMANDS "TMP"

All temperature sensor commands begin with the following sequence: **"^^^" "TMP"** (quotes are not included, cases sensitive)

"ROM"<c>, <val1> <val2> <val3> <val4> <val5> <val6> <val7> <val8><CR>	Save Temperature Sensor ROM code	c = temperature channel number "1" to "8" val = three character byte value "000" to "255" val1 = the value after <val> is the byte number 1 to 8
"EN"<x>	Enable Temperature Channel	x = Temperature Channel 1 to 8 When enabled, temperature is processed and read. NOTE: Leave disabled if not required to reduce required processing.
"DS"<x>	Disable Temperature Channel	x = Temperature Channel 1 to 8
R<x><r>	REMOVED! Set Resolution (9,10,11 or 12 bit)	x = Temperature Channel 1 to 8 r = resolution (0,1,2,3 = 9bit,10bit,11bit,12bit)
"ROW"<x>	Read then Write ROM code for SINGLE 1-Wire Device connected.	x = Temperature Channel 1 to 8 NOTE: Only one 1-wire device may be connected to the bus when using this command. Reads the ROM code of the connected 1-wire device and saves it to the temperature channel indicated by the value of <x>.
"DGC" or "DGF"	Set Temperature Units (C or F) Sept 12,2012	DGC sets to Celsius DGF sets to Fahrenheit
"SAS"	Search All and Save to Settings	Finds all 1-wire device ROM code (8max). Currently, this function saves all devices in the settings beginning with degree channel 1. Subsequent "search all" commands will always find and save the devices in the same order. If one device is removed, all other devices will shift channels to replace the missing device/channel. This behaviour may be modified in the future to prevent re-assigning device channels. Nov4, 2012
"SAL"	Search and Display All 1-Wire Devices	Finds all 1-wire device ROM code (8max) and returns the 8-byte ROM code for each separated by a comma.

PULSE COUNTER COMMANDS "PLS"

All temperature sensor commands begin with the following sequence**: "^^^" "PLS"** (quotes are not included, cases sensitive)

"EN"<x>	Enable Pulse Counter	x = Counter number 1 to 4
"DS"<x>	Disable Pulse Counter	x = Counter number 1 to 4

REQUEST INFORMATION COMMANDS "RQS"

All data Request commands begin with the following sequence: "^^^" "RQS"
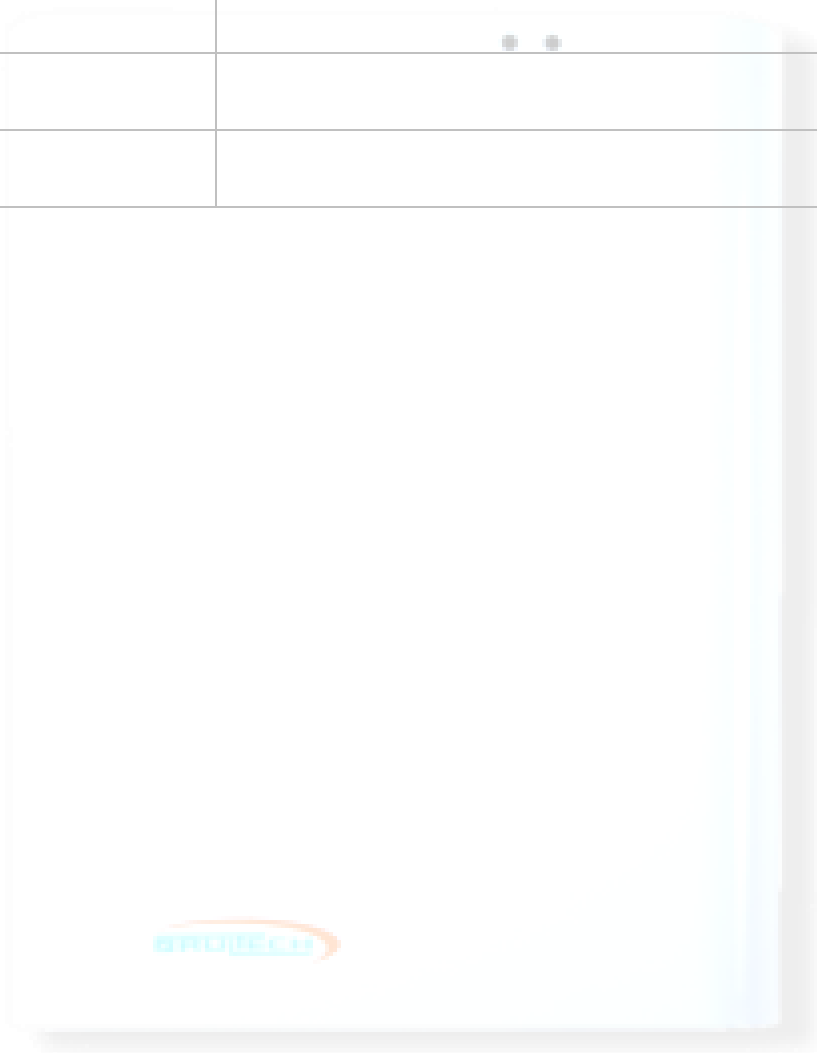
| "ALL" | Read all settings | Response: "ALL" <CRLF> Then sends Hex values comma delimited:

1 byte (spare)
48 bytes channel option
48 bytes CT types (ch1 to ch48)
24 bytes CT range (One nibble/channel. Two nibbles/byte.) ex: First CT range byte = 0x23, CH1 CT range = 3 and CH2 CT range = 2
1 byte = PT type
1 byte = PT range
1 byte = packet format selection
1 byte = packet send interval
1 byte: Wifi Auto Reset Timer Value ("0" = off)
1 byte:Wifi Missing Resp Rst Value ("0" = off)
1 byte: Wifi Options
1 byte: Xbee Auto Reset Timer Val ("0" = off)
1 byte: Xbee Missing Resp Rst Val ("0" = off)
1 byte: XbeeWifi Options
1 byte: Primary COM port (not yet implemted)
48 bytes = packet options / channel
1 byte = counter options bits 0 to 3, set for enabled (selects if included in certain packet formats)
1 byte = temperature enable 8bits, one per temperature channel (selects if included in certain packet formats)
8 bytes = Temperature 1 ROM code
8 bytes = Temperature 2 ROM code
8 bytes = Temperature 3 ROM code
8 bytes = Temperature 4 ROM code
8 bytes = Temperature 5 ROM code
8 bytes = Temperature 6 ROM code
8 bytes = Temperature 7 ROM code
8 bytes = Temperature 8 ROM code
1 byte = Temperature channel enabled to be converted and read.
1 byte = Additional 1-Wire device options
1 byte = Hardware Modules Installed (required to reduce battery consumption)
1 byte = Maximum channels processed.
48 bytes = reserved for future
8 bytes = System Settings
6 bytes = bitwise selects which of the 48 channels sent during an HTTP PUT
1 byte = bitwise selects which of the 4 pulse counter channels sent during an HTTP PUT .
1 bytes = bitwise selects which of the 8 temperature channels sent during an HTTP PUT
2 bytes = General packet options
1 byte = NET metering options (bi-directional counter?)
2 bytes = Chunk size |

		1 byte = Enabled Pulse counters... which counters are processed. Others..???? --------- total ??? (was 374) bytes End with <CRLF>
"ATP"	Read all CT Types	Response: Hex values comma delimited Packets sent: 48 comma delimited Hex values <CRLF> carriage return (ascii 13d,10d)
"ARG"	Read all CT Ranges	Response: Hex values comma delimited Data sent: 24 comma delimited Hex values <CRLF> carriage return (ascii 13d.10d) Each range is represented by a single nibble of each Hex value. The LSB nibble represents the odd channels and MSB nibble the even channels. For example: Hex values: 12,32,34,45........ would represent CH1 range = 2 CH2 range = 1 CH3 range = 2 CH4 range = 3 CH5 range = 4 CH6 range = 3 CH7 range = 5 CH8 range = 4
"AOP"	Read All channel Options	Response: Comma delimited Hex values. One value for each channel. The "phase" setting is part of this byte. Packets sent: 48 comma delimited Hex values <CRLF> carriage return (ascii 13d,10d)
"PH"<xx>	Read Channel Phase Setting	<xx> is 01 to 48 channel value Responds with "A", "B" or "C"

"RTL"	Read Realtime Status	Responds with "OFF" or "ON" followed by <CRLF>
"ITV"	Read Realtime Send Interval	Responds with a single hex value representing 0 to 255 seconds followed by <CRLF>
"RTF"	Read Realtime Packet Format	Responds with a single hex value representing format value followed by <CRLF>
"TST"	Read Temperature ON/Off Status	Responds with "xxxxxxxx" where x = 0 or 1 (off/on) followed by <CRLF> Ex: 00010100 temp channel 3 & 5 enabled
"PST"	Read Pulse ON/Off Status	Responds with "xxxx" where x = 0 or 1 (off/on) followed by <CRLF> Ex: 0100 pulse counter 3 enabled
"PKS"	Read Packet Chunk Size	Responds with numeric value (two byte size)
"PKI"	Read Packet Chunk Interval	
"BFF"	Get Max Buffer Size	Responds with a value between 10 and 1700. This value is the maximum number of bytes to be buffered and must be set to a value greater than the packet size and chunk size. If set to great, problems may occur trying to transfer too much buffered data should a connection be lost then restored.
"Cxx"	Read Specified Channel CT Type and Range	xx= 01 to 48 Channel number Responds with two byte hex value representing CT Type and Range values <CRLF>
"RPT"	Read PT Type and Range	Responds with two byte hex value representing PT Type and Range values <CRLF>
"HZ"	Read Hertz Setting	Responds with 60Hz or 50Hz
"RTR"<x>	Read 8 Temperature ROM byte setting	x = 1 to 8 Responds with 8 comma delimited hex bytes followed by CRLF
"CMX"	Read Max Channel Setting	A value of 1 to 48 is returned. This represents the value that the maximum number of channels to be processed is set to.
"ENM"<x>	Master Enable Polarized WS counters for NET metering	.x = "0" or "1" When set to "1", polarized ws counters are used. If NET metering is not required, set this value to zero. Future updates will provide the ability to select which channel polarized counters are enabled.
"TSB"	Read ThingSpeak number of blocks	Returns the number of ThingSpeak blocks. Each block contins 8 channels.
"URL"	Read URL string	<url string> Followed by CRLF
"URE"	Read URL extension	<url extension string> Followed by CRLF
"TE"<x>	Read Temperature	x is the temperature channel

"WAR"	Read Wifi Auto-Reset Value	Returns Wifi Auto-Reset value
"WRR"	Read Wifi Missing Response Counter Value	Returns Wifi Missing Response value
"VR1"	Read Firmware Version of Master Processor	Read the firmware version for the communication processor. Ends with CRLF
"VR2"	Read Firmware Version of Engine Processor	Read the firmware version for CH1 to CH32 engine processor. Ends with CRLF
"VR3"	Read Firmware Version of Expansion Processor (if 16 channel expansion is present)	Read the firmware version for CH33 to CH48 engine processor. Ends with CRLF
"DTM"	Request time	Read realtime clock yy-mm-dd hh:mm:ss
"SRN"	Read Serial Number	
"EPC"	Read Enabled Post Channels	Returns: <xxx>,<xxx>,<xxx>,<xxx>,<xxx>,<xxx> xxx = 000 to 255 Six bit mapped bytes sets which of the 48 channels will be sent with the HTTP post. First value = channels 1 to 8 Second value = channels 9 to 16 …… and so on Ex: EPC007,001,128,000,001,000<CR> 007(1-8): chan 1,2 and 3 are enabled 001(8-16): chan 9 enabled 128(17-24): chan 24 enabled 000(25-32): none enabled 001(33-40): chan 33 enabled 000(41-48): none enabled
"EPP"	Read Enabled Post Counters	Returns bit mapped value representing the counters included in HTTP post.
"EPT"	Read Enabled Post Temperature Channels	Returns bit mapped value representing the temperature channels included in HTTP post.
"KAI"	Keep Alive Options	Returns a value between 0 and 7 corresponding with the keep-alive options.

	Keep Alive String	String between 1and 8 characters sent as keep-alive character.
"KAS"		

DATA POLLING API "API"

All data Request commands begin with the following sequence: "^^^" "API"

"WAT"<xx>	Read Single Channel Power (Watt)	xx= Channel number 01 to 48 Numeric watt data followed by CRLF
"WRG" <ss><rr>	Read Power for a Range of Channels (Watt)	ss = start channel (01 to 48) rr = number of consecutive channels Numeric Data sent is comma delimited. Ends with CRLF
"ENR"<xx>	Read Single Channel Energy (WattSecond)	xx= Channel number 01 to 48 Numeric watt-second data followed by CRLF
"EGR" <ss><rr>	Read Absolute for a Range of Channels (WattSeconds)	ss = start channel (01 to 48) rr = number of consecutive channels Numeric Watt-Second Data sent is comma delimited. Ends with CRLF
"EPC"	Read Enabled Channels for POST	Returns six byte values. These bit-mapped values represents which channels are included during a POST Returned values are comma delimited. Ex: value 5 = counter 1 and counter 3
"EPP"	Read Enabled Pulse Counters for POST	Returns single byte value 0 to 15 This bit-mapped value represents which pulse counter channels are included during a POST Ex: value 5 = counter 1 and counter 3
"EPT"	Read Enabled Temperature Channels for POST	Returns single byte value 0 to 255 This bit-mapped value represents which temperature channels are included during a POST Ex: value 5 = temp 1 and temp 3
"VLT"	Read Voltage	Sends Numeric Value = Volt x 10. This value should be divided by ten to obtain one decimal point accuracy. Ends with CRLF
"SEC"	Read Incremental Seconds Counter	Send Numeric 3-byte value. Ends with CRLF
"VAL"	Send all recent power, temperature and counter values. The purpose of this API is to provide updated data to the setup program.	Responds with "VAL"<CRLF> Sends comma delimited numeric values. c1=<watt>,c2=<watt> ....,t1=<degrees>,t2=<degrees>,…… p1=<pulse count lsb value>, …… "END"<CRLF>

"TP"<x>	Read Temperature of Specified Temperature Channel	"x" is the temperature channel number.
"SPK"	Send One Packet	Sends using the currently selected packet format. Realtime must be OFF in order for this cmd to be functional.

COMMANDS USING HTTP GET METHOD

All data Request commands begin with the following sequence: "^^^"

"RQS" "WEB"	Send All GEM Settings Info	index.php/ecmEngine/gsettings/?SN=&cm=&t1=&t2=&t3=&t4=&t5=&t6=&t7=&t8=&t9= &t10=&t11=&t12=&t13=&t14=&t15=&t16=&t17=&t18=&t19=&t20=&t21=&t22=&t23= &t24=&t25=&t26=&t27=&t28=&t29=&t30=&t31=&t32=&t33=&t34=&t35=&t36=&t37= &t38=&t39=&t40=&t41=&t42=&t43=&t44=&t45=&t46=&t47=&t48=&r1=&r2=&r3=&r4= &r5=&r6=&r7=&r8=&r9=&r10=&r11=&r12=&r13=&r14=&r15=&r16=&r17=&r18=&r19= &r20=&r21=&r22=&r23=&r24=&r25=&r26=&r27=&r28=&r29=&r30=&r31=&r32=&r33= &r34=&r35=&r36=&r37=&r38=&r39=&r40=&r41=&r42=&r43=&r44=&r45=&r46=&r47= &r48=&p1=&p2=&p3=&p4=&p5=&p6=&p7=&p8=&p9=&p10=&p11=&p12=&p13=&p14= &p15=&p16=&p17=&p18=&p19=&p20=&p21=&p22=&p23=&p24=&p25=&p26=&p27= &p28=&p29=&p30=&p31=&p32=&p33=&p34=&p35=&p36=&p37=&p38=&p39=&p40= &p41=&p42=&p43=&p44=&p45=&p46=&p47=&p48=&pt=&pr=&dp=&si= SN = serial number cm = maximum channels used t = channel type r = channel range p = channel phase pt = PT Type pr = PT Range dp = device phase si = send interval

SEND COMMAND TO SPECIFIC GEM USING SERIAL NUMBER

For systems with multiple GEM monitors, a command may be directed to a specific device based on its serial number. This requires the following characters between the "^^^" and any of the commands above:

"NMBxxxxx" where "x" is the last five characters of the serial number.

Example:

To send a real-time off command (^^^SYSOFF) to GEM serial number "01002345" the following command would be used:

"^^^NMB02345SYSOFF"

This will issue the command to that specific device.