mySQM+ PROTOCOL 026

31 December 2021

INTERFACE COMMANDS - SERIAL OR TCP/IP

All send commands begin with a :
All send commands end with a #
All send commands are two digits followed by optional values (if setting a value)

RETURN VALUES

All return values are terminated with a #
All return values begin with a single character code to represent what the command was
All return values are string characters so numbers are converted to strings before sending

Send	Return Code	Meaning
:03#	C	Get System up-time
:41#		Reboot controller
:71#	Q	Get MAC Address
:01#	A	Get magnitude value (as float)
:21#	U	Get LUX value
:31#	Z	Get NELM
:A0#	^	Get make hay
BME280		
:32#	a	Get BME820 humidity (app can calculate dew point from temp and humidity)
:33#	e	Get BME820 pressure
:34#	c	Get BME820 temperature
:35#	d	Get BME280 dewpoint
:A7#	$	Get BME280 defined altitude (in meters)
:A8x#		S et BME280 defined altitude (in meters)
:A9#	>	Get BME280 Pressure adjusted to sea-level
DISPLAY		
:06#	F	Get display enabled state
:17x#		Set display enabled [0 | 1]
:78#	1	Get page display option [String]
:79xxx#		Set page display option [String]
:80#	2xxxx#	Get page display time [unsigned int]
:81xxx#		Set page display time [unsigned int 2000-10000] in milliseconds
DISTANCE MODE		
:44#	j	Get metric/imperial mode 0 = metric 1 = Imperial
:45x#		Set metric/imperial mode 0 = metric 1 = Imperial

FIRMWARE

:04#	D	Get firmware version number
:05#	E	Get firmware filename
:40#	h	Get firmware hash value

GPS

:08#	H	Get local date
:09#	I	Get local time
:10#	J	Get longitude
:11#	K	Get latitude
:12#	L	Get altitude (height in meters)
:13#	M	Get number of satellites
:16#	N	Get GPS fix
:18#	%	Get if GPS using static co-ordinates [0=No 1=Yes]
:30x#		Set GPS to use static co-ordinates [0=Off 1=On]
:50#	@	Get if GPS using truncated co-ordinates
:92x#		Set GPS truncated [0=No 1=Yes]
:A1#	[	Get GPS Static Latitude setting
:A2#	]	Get GPS Static Longitude setting
:A3x#		Set GPS Static Latitude setting
:A4x#		Set GPS Static Longitude setting

MLX90614

:19#	O	Get IR Sensor object temperature
:20#	P	Get IR Sensor ambient temperature
:27#	V	Get cloudstate, 0=clear 1=partly cloudy 2=cloudy
:48#	t	Get % cloud cover
:49#	u	Get corrected sky value
:51#	m	Get k1 cloud model value
:52#	n	Get k2 cloud model value
:53#	o	Get k3 cloud model value
:54#	p	Get k4 cloud model value
:55#	q	Get k5 cloud model value
:56#	r	Get k6 cloud model value
:57#	s	Get k7 cloud model value
:58#	v	Get tempclear
:59#	w	Get tempovercast
:60#	x	Get cloudflagpercent
:61xx#		Set k1 cloud model value
:62xx#		Set k2 cloud model value
:63xx#		Set k3 cloud model value
:64xx#		Set k4 cloud model value
:65xx#		Set k5 cloud model value
:66xx#		Set k6 cloud model value
:67xx#		Set k7 cloud model value
:68xx#		Set temp clear (-8.0)
:69xx#		Set tempovercast (0.0)
:70xx#		Set cloudflagpercent (30.0)

MQTT

:07# G	Get MQTT subscribe topic
:15String#	Set MQTT subscribe topic
:83# 4String#	Get MQTT broker IP address
:84String#	Set MQTT broker IP address
:97# !String#	Get MQTT Publish Topic
:98String#	Set MQTT Publish Topic
:A5# <String#	Get MQTT Publish Interval
:A6String#	Set MQTT Publish Interval

NTP

:00# z	Get NTP time zone string
:74# b	Get NTP UTC date/time string [dd/mm/yr,hh:mm:ss]
:75# y	Get NTP LOC [UTC adj by TZ] date/time string [dd/mm/yr,hh:mm:ss]
:77# 0	Get NTP servername
:85String#	Set NTP Servername
:88# 6xx#	Get NTP minutes interval between RTC is synced to NTP
:89xx#	Set minutes between syncing RTC to NTP
:99String#	Set NTP time zone

RAIN

:22# 3	Get rain total in mm over previous hour
:23# R	Get Raining (Boolean)
:24# S	Get raining analogue value (int)
:28# W	Get cumulative rainfall for this current hour
:29# X	Get cumulative rainfall for this current day
:82#	Sync Rain Bucket Gauge min/hour to rtc

RTC

:72# T	Get RTC date/time string dd/mm/yr,hh:mm:ss]
:94String#	Set RTC date/time mm,dd,yyyy,hr,mn,ss
:90# 7x#	Get SYNC RTC to NTP, if NTP should sync date/time to RTC
:91x#	Set SYNC RTC to NTP, if NTP should sync NTP date/time to RTC

TEMPERATURE MODE

:42# i	Get temperature mode, 0=Celsius 1=Fahrenheit
:43x#	Set temperature mode, 0=Celsius 1=Fahrenheit

TLS2591

:25# 8	Get tlscorrectionfactor
:26#	Set TLS sensor correction factor
:36xx#	Set TSL2591 Gain
:37xx#	Set TSL2591 Integration time
:38# f	Get TSL2591 Gain
:39# g	Get TSL2591 Integration time

WIND

:46#	k	Get wind speed (in meters per second)
:47#	l	Get wind direction
:73#	Y	Get Wind Chill Factor
:86#	5	Get average windspeed last 30s
:87#	z	Get maximum wind speed gust last 30s

WEBSERVER

:02#	B	Get webserver port
:14xxxx#		Set webserver port
:95#	*x#	Get webserver state (0=Stopped 1=Running)
:96x#		Start/Stop webserver (0=stop 1=start)

UNUSED

76
93
A8

BY NUMBER

Send	Return Code	Meaning
:00#	z	Get NTP time zone string
:01#	A	Get magnitude value (as float)
:02#	B	Get webserver port
:03#	C	Get System up-time
:04#	D	Get firmware version number
:05#	E	Get firmware filename
:06#	F	Get display enabled state
:07#	G	Get MQTT subscribe topic
:08#	H	Get local date
:09#	I	Get local time
:10#	J	Get longitude
:11#	K	Get latitude
:12#	L	Get altitude (height in meters)
:13#	M	Get number of satellites
:14xxxx#		Set webserver port
:15String#		Set MQTT subscribe topic
:16#	N	Get GPS fix
:17x#		Set display enabled [0,1]
:18#	%	Get if GPS using static co-ordinates [0=No, 1=Yes]
:19#	O	Get IR Sensor object temperature
:20#	P	Get IR Sensor ambient temperature
:21#	U	Get LUX value
:22#	3	Get rain total in mm over previous hour
:23#	R	Get Raining (Boolean)
:24#	S	Get raining analogue value (int)
:25#	8	Get tlscorrectionfactor
:26#		Set TLS sensor correction factor
:27#	V	Get cloudstate 0=clear 1=partly cloudy 2=cloudy
:28#	W	Get cumulative rainfall for this current hour
:29#	X	Get cumulative rainfall for this current day
:30x#		Set GPS to use static co-ordinates [0=Off 1=On]
:31#	Z	Get NELM
:32#	a	Get BME820 humidity (app can calculate dew point from temp and humidity)
:33#	e	Get BME820 pressure
:34#	c	Get BME820 temperature
:35#	d	Get BME280 dewpoint
:36xx#		Set TSL2591 Gain
:37xx#		Set TSL2591 Integration time
:38#	f	Get TSL2591 Gain
:39#	g	Get TSL2591 Integration time
:40#	h	Get firmware hash value
:41#		Reboot controller
:42#	i	Get temperature mode 0=Celsius 1=Fahrenheit
:43x#		Set temperature mode 0=Celsius 1=Fahrenheit
:44#	j	Get metric/imperial mode 0 = metric 1 = Imperial
:45x#		Set metric/imperial mode 0 = metric 1 = Imperial

:46#	k	Get wind speed (in meters per second)
:47#	l	Get wind direction
:48#	t	Get % cloud cover
:49#	u	Get corrected sky value
:50#	@	Get if GPS using truncated co-ordinates
:51#	m	Get k1 cloud model value
:52#	n	Get k2 cloud model value
:53#	o	Get k3 cloud model value
:54#	p	Get k4 cloud model value
:55#	q	Get k5 cloud model value
:56#	r	Get k6 cloud model value
:57#	s	Get k7 cloud model value
:58#	v	Get tempclear
:59#	w	Get tempovercast
:60#	x	Get cloudflagpercent
:61xx#		Set k1 cloud model value
:62xx#		Set k2 cloud model value
:63xx#		Set k3 cloud model value
:64xx#		Set k4 cloud model value
:65xx#		Set k5 cloud model value
:66xx#		Set k6 cloud model value
:67xx#		Set k7 cloud model value
:68xx#		Set temp clear (-8.0)
:69xx#		Set tempovercast (0.0)
:70xx#		Set cloudflagpercent (30.0)
:71#	Q	Get MAC Address
:72#	T	Get RTC date/time string dd/mm/yr,hh:mm:ss]
:73#	Y	Get Wind Chill Factor
:74#	b	Get NTP UTC date/time string [dd/mm/yr,hh:mm:ss]
:75#	y	Get NTP LOC [UTC adj by TZ] date/time string [dd/mm/yr,hh:mm:ss]
:76#		
:77#	0	Get NTP servername
:78#	1	Get page display option [String]
:79xxx#		Set page display option [String]
:80#	2xxxx#	Get page display time [unsigned int]
:81xxx#		Set page display time [unsigned int 2000-10000] in milliseconds
:82#		Sync Rain Bucket Gauge min/hour to rtc
:83#	4String#	Get MQTT broker IP address
:84String#		Set MQTT broker IP address
:85String#		Set NTP Servername
:86#	5	Get average windspeed last 30s
:87#	z	Get maximum wind speed gust last 30s
:88#	6xx#	Get NTP minutes interval between RTC is synced to NTP
:89xx#		Set minutes between syncing RTC to NTP
:90#	7x#	Get SYNC RTC to NTP, if NTP should sync date/time to RTC
:91x#		Set SYNC RTC to NTP if NTP should sync NTP date/time to RTC
:92x#		Set GPS truncated [0=No 1=Yes]
:93#		

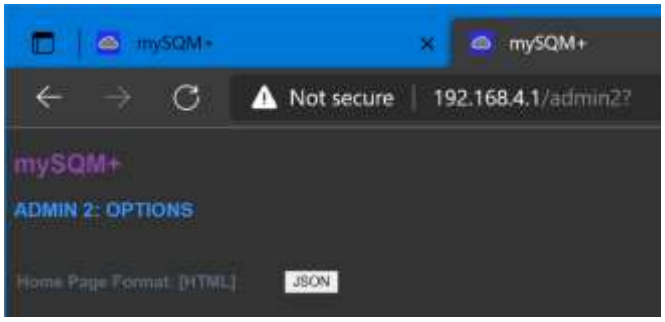
:94String#	Set RTC date/time mm,dd,yyyy,hr,mn,ss
:95# *x#	Get webserver state (0=Stopped 1=Running)
:96x#	Start/Stop webserver (0=stop> 1=start)
:97# !String#	Get MQTT Publish Topic
:98String#	Set MQTT Publish Topic
:99String#	Set NTP time zone
:A0# ^	Get make hay
:A1# [	Get GPS Static Latitude setting
:A2#]	Get GPS Static Longitude setting
:A3x#	Set GPS Static Latitude setting
:A4x#	Set GPS Static Longitude setting
:A5# <String#	Get MQTT Publish Interval
:A6String#	Set MQTT Publish Interval
:A7# $	Get BME280 defined altitude (in meters)
:A8xx#	Set BME280 defined altitude (in meters)
:A9# >	Get BME280 Pressure adjusted to sea-level

WEBSERVER JSON SUPPORT

There is significant JSON support built into the web server functionality.

HOME PAGE SUPPORT FOR BOTH HTML AND JSON

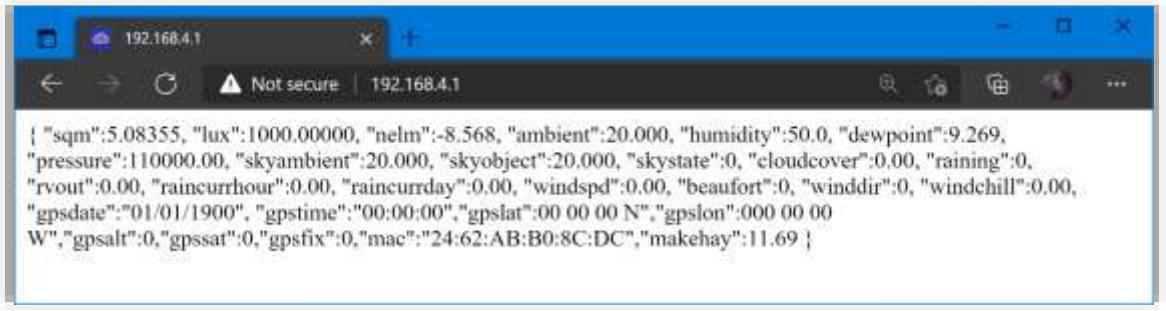
The home page of the web server can be sent in either HTML format or in JSON format. The admin page responsible for this is ***/admin2***



In the screenshot, the current format setting for the home page is HTML (the current format state is shown between [].

The button shows what the home page format state will change to when clicked by the user.

Clicking on the JSON button will change the home page format to JSON, and any request for the home page will be sent a JSON string.



This can be useful if you want to write your own applications to communicate with the controller. The client application is responsible for connecting to the controller and sending a GET root page **/** request to the webserver via HTTP.

All parameters are lowercase and values are returned in CELSIUS and METRIC. The client application is responsible for converting any values to FAHRENHEIHT or IMPERIAL units of measurement.

This call obeys the rules for truncate gps and static gps. The gps co-ordinates are derived from the static gps co-ordinates, use static gps is enabled, and truncate gps is enabled on the webserver admin page ***/admin2***

Using the Home Page customizations, you can decide what values/settings appear on the webserver's Home Page, for both HTML and JSON formats.

JSON SUPPORT FOR CLIENT APPLICATIONS

Client applications can establish a web connection on port 80 to the controller. Using this port, the application can request the groups of data corresponding to various functions of the controller. A request will return a JSON formatted string.

/about REQUEST

An application sending a request **/about** to the controller will get a JSON string response of the *controller* details.

{ "product":"mySQM+","Author":"R. Brown, 2020-2021.","ver":"120" }

/d1 REQUEST

An application sending a request **/d1** to the controller will get a JSON string response of the "*DARKSKY*" variables.

{ "sqm":25.08356,"nelm":7.536,"lux":0.00,"skystate":1,"cloudcover":100.00 }

/date REQUEST

An application sending a request **/date** to the controller will get a JSON string response for the *local date and time*. Which date and time is sent to the client is subject to the LOCALDATETIMESOURCE on the /admin web page.

{ "local":"02/12/2021,21:28:35","source":rtc" }

/g1 REQUEST

An application sending a request **/g1** to the controller will get a JSON string response of the "*GPS*" variables. This call obeys the rules for truncate gps and static gps. To convert UTC to LOC, firmware uses the *TimeZone* and *TZ* string values from the */admin5* web page (NTP settings).

{ "gpsdate":"01/01/1900","gpstime":"00:00:00","gpslat":"00 00.000000 N","gpslon":"00 00.000000 E","gpsalt":0, "gpssat":0,"gpsfix":0 }

/mac REQUEST

An application sending a request **/mac** to the controller will get a JSON string response of the *MAC address* value.

{ "mac":"24:62:AB:B0:8C:DC" }

/makehay REQUEST
An application sending a request **/makehay** to the controller will get a JSON string response of the *vpd value* (make hay < 10 is good) value.

{ "makehay":7.12 }

/mqtt REQUEST
An application sending a request **/mqtt** to the controller will get a JSON string response of the *MQTT broker IP and connection details* [if enabled].

{
"brokerip":"192.168.2.129","publishtopic":"mySQM/rufus","subscribetopic":"mySQM/cmdset",”mqrrpubtime”:30,"mqttconnectstate":"UNKNOWN","mqttclientstate":"false" }

/ntp REQUEST
An application sending a request **/ntp** to the controller will get a JSON string response of the *NTP local date and time* dynamic values [if enabled].

{
"ntpldate":"","ntpltime":"","ntpudate":"","ntputime":"","ntpserver":"pool.ntp.org","ntptz":"TZ_Asia_Kuaka_Lumpur","ntptz": “<+08>-8” ,"rtcsyncntp":0,"rtcsynctime":30 }

/rain REQUEST
An application sending a request **/rain** to the controller will get a JSON string response of the *rain information* [if enabled].

{
"raining":0, "rvout":2.16,"raincurrhour":0.00,"rainprevhour":0.00,
"raincurrday":0.00,"rainprevday":0.00
}

/rd REQUEST
An application sending a request **/rd** to the controller will get a JSON string response of all dynamic values except GPS.

{
"sqm":12.21815,"nelm":-1.461,"ambient":20.000,"humidity":50.0,"dewpoint":9.269,
"pressure":1100.00,"skyambient":20.000,"skyobject":20.000,"lux":1.40011,"skystate":0,
"cloudcover":0.00,"raining":0,"rvout":0.00,"windspd":0.00,"beaufort":0,"winddir":0
}

/rtc REQUEST
An application sending a request **/rtc** to the controller will get a JSON string response of the *Real Time Clock* dynamic values [if enabled].

{ "rtcdate":"19/07/2021","rtctime":"18:16:23" }

/t1 REQUEST
An application sending a request **/t1** to the controller will get a JSON string response of the "*TEMPERATURE*" variables.

{ "ambient":20.000,"dewpoint":9.269,"skyambient":23.850,"skyobject":23.690 }

/tlscf REQUEST
An application sending a request **/tlscf** to the controller will get a JSON string response of the *TLSCorrectionFactor* variable.

{ "tlscf":1.8 }

/uptime REQUEST
An application sending a request **/uptime** to the controller will get a JSON string response of the *system uptime.*

{ "uptime":"00:00:06" }

/w1 REQUEST
An application sending a request **/w1** to the controller will get a JSON string response of the "*WEATHER*" variables.

{ "humidity":50.0,"pressure":110000.00,"raining":0,"rvout":0.00,"raincurrhour":0.00,
"raincurrday":0.00,"windspd":0.00,"beaufort":0,"winddir":0 }

WEBSERVER SERVICE MANAGEMENT SUPPORT

Additional administration requests are available. These are shown below.

/rebootws REQUEST

An application sending a request **/rebootws** to the controller will cause the controller to stop and then restart the web-server service.

/reboottcp REQUEST

An application sending a request **/reboottcp** to the controller will cause the controller to stop and then restart the TCP/IP-server service [used by TCP/IP applications in Windows and LINUX on port 2121].

/reboot REQUEST

An application sending a request **/reboot** to the controller will cause the controller to reboot. There is no response to this request.

/rbgsync REQUEST

An application sending a request **/rbgsync** to the controller will cause the controller to update the **minute** and **hour** values used by the Rain Bucket Gauge to that of the Real Time Clock. There is no response to this request.

WEBSERVER XHTML SUPPORT

The webserver supports direct XJTL requests, where the value of the requested item is returned. The home page of the webserver as an HTML page has embedded scripts in the page that send XHTML requests to the server, and the responses are used to dynamically update the web page contents.

This means that you do not need to use REFRESH to get an updated page with new values, because the home page does this dynamically without having to request the home page again.

XHTML Request	Value returned
/ba	ambient
/bh	humidity
/bd	dewpoint
/bp	pressure
/ma	skyambient
/mo	skyobject
/sq	sqm
/ne	nelm
/lu	lux
/ss	skystate
/cc	cloudcover
/ra	raining
/rv	rvout
/rh	raincuurhr
/ry	raincurrday
/ws	windspd
/bs	beaufort
/wd	winddir
/wc	windchill
/ut	system_up_time
/mh	makehay