

1 QTFM2 Logger Getting Started

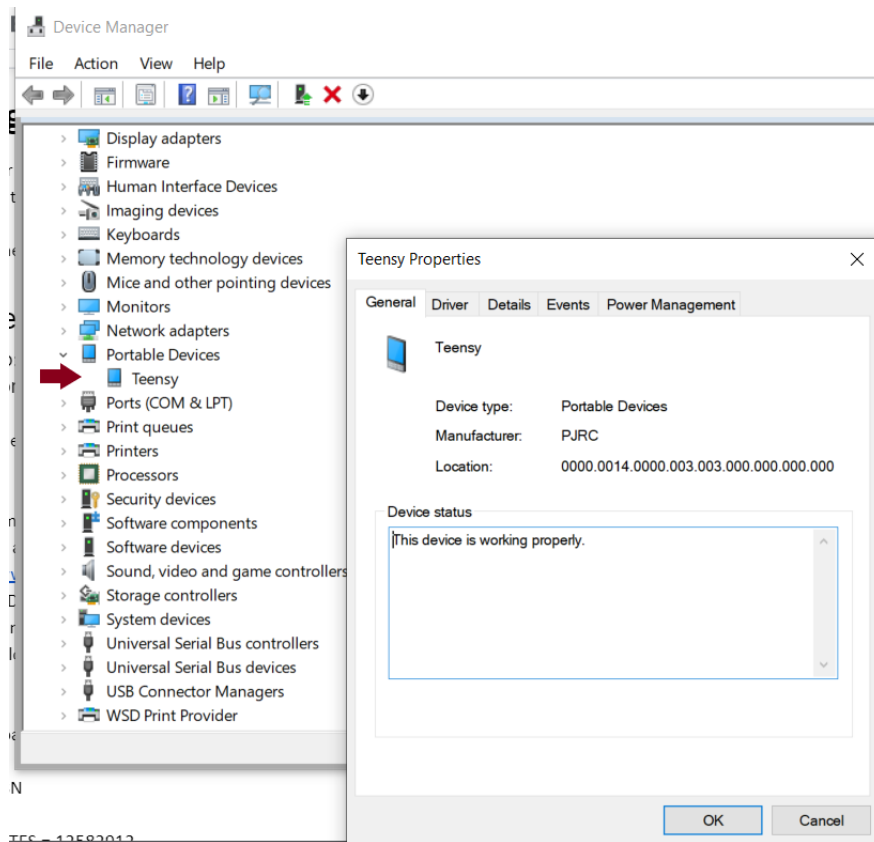
The QTFM2 Logger allows for recording and storing QTFM sensor output synchronized with GPS position data.

This document describes the operation of the Logger and QLogMonitorConsole (QLogMonCon) host-side application software.

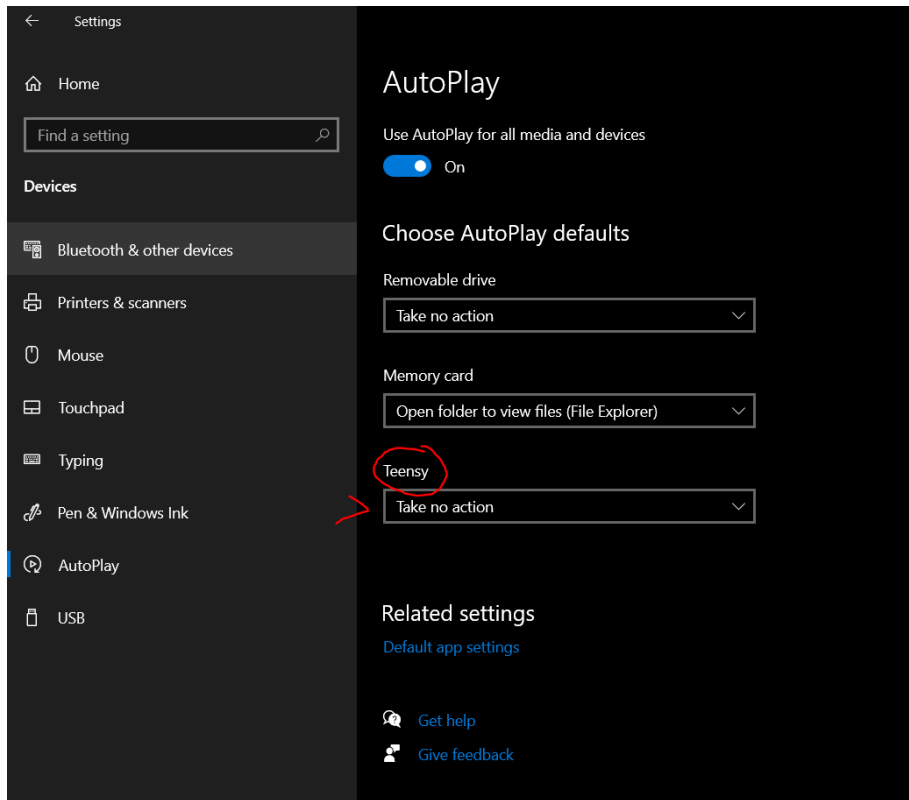
1.1 Connect PC to Logger

Required hardware setup: QTFM ACB2 hardware with a USB cable attached from PC host to the exposed USB port connector.

1. After connecting the USB cable, the logger will be automatically recognized by the PC host as a Portable Device. You can verify this in Device Manager as shown below.



2. The Windows system may prompt to take action whenever the device is plugged in. It is strongly recommended that you change the Win10 Settings > AutoPlay default action to "Take no action" for the Teensy Portable Device. The PC host applications provide a full set of features to control the logger operation and transfer logging files.



3. Launch the QLogMonitorConsole (QLogMonCon) PC application to interact with the logger.

1.2 Install and run QLogMonitorConsole PC app

QLogMonCon is a utility application that interacts with the logger to perform utility operations. It is particularly useful for transferring log files from the logger SD card storage to a PC repository. This is described in detail in the QLogMonCon App section.

If you have not yet installed QLogMonCon, use this link to download the installer:

[QLogMonitorConsole Installer](#)

NOTE: QLogMonCon will always check for updates available and offer to install the new version.

If you have already installed QLogMonCon, you can launch it from the Start Menu under the QuSpin folder.

This screen is presented if QLogMonCon has successfully opened a connection to the logger:

```
----- Qtfm Logger Monitor Console App      v1.1.1.0  Deployment: 1.0.5.66 -----
Checking and installing prerequisites
Finding QTFM logger and launching session over USB virtual com port...
Using QLogMonitor DLL version 1.0.8.0
Device found, opening session...
COM port session opened

Logger error status is 0
Running ACB firmware version 4.4.0
Set ACB to idle state
Waiting for logger...
2 files found, 'f' to list them
Get logger configuration  successful.

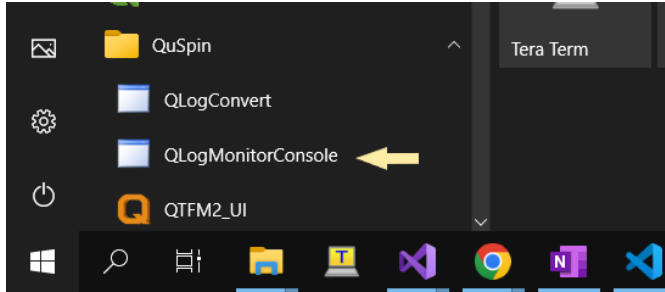
Primary commands:
f      list files with current filename prefix
<num>  transfers a specific file <FILE_NAME_PREFIX>_0<num>.txt
x      to exit
u      update logger firmware
g      show/change USE_GPS
p      show/change PPS_TIMESTAMP
m      show/change MAGDATA_PULSE_TIMESTAMP
c      show/change LOG_AT_STARTUP (continuous logging)
t      show/change GPS_TYPE
b      show/change MAX_FILE_SIZE_BYTES
s      show/change FILE_NAME_PREFIX string

Secondary commands:
e      get error word
v      firmware version
r      logger reset
n      name of next log file
? / h  show this command list
l      start logging
o      stop logging
y      print internal temp over BT
Enter:
```

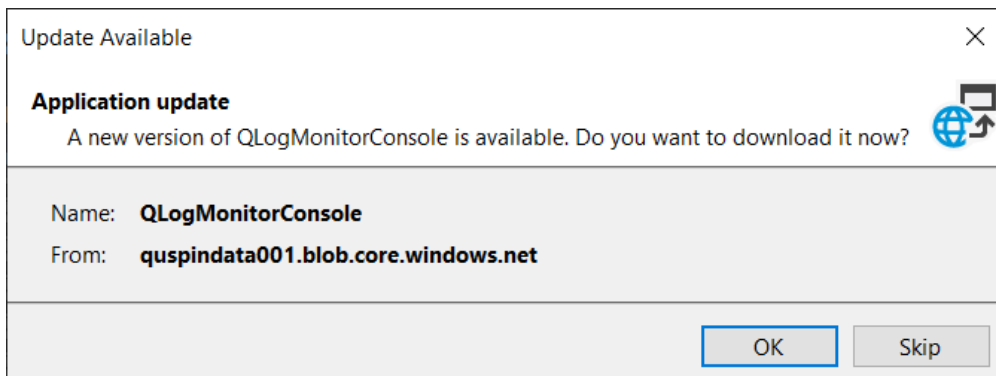
1.3 QLogMonitorConsole installation and update

First time install of QLogMonitorConsole by [downloading and running setup.exe installer located here](#).

After installation, the app can be launched from the Start Menu in the QuSpin folder:



Once installed, the app automatically searches for available updates if an Internet connection is available. If a new version is found, the user is presented with an opportunity to update the app:



2 QLogMonitorConsole App

QLogMonitorConsole is a Windows console utility that interacts with the logger to perform configuration and maintenance operations. It is particularly useful for transferring/converting log files from the logger SD card storage to a PC repository without the need to physically move the SD card from the logger. It requires a USB cable connection between the exposed port on the logger and the PC.

When the QLogMonCon app starts, it puts the logger in an idle mode to enable file management features. In this mode, normal logging functionality is suspended until the logger is reset or power cycled. The logger is reset automatically the the app when the app exits.

When the logger is recognized by a Windows PC as a Portable Device, the logger exposes a Media Transport Protocol (MTP) interface that is used to list and transfer files from the SD card. In addition, the logger implements USB virtual COM port functioning as a bidirectional serial port. These two interfaces enable the QLogMonCon app and logger firmware to provide the configuration and file transfer functionality.

Running the QLogMonCon app:

1. Connect a PC host to the exposed USB port on the QTFM2 ACB2 hardware.
2. Launch QLogMonitorConsole from the Windows Start menu which searches for the logger interface and opens a session

Utility functions are accessed by single-character commands:

```
Primary commands:
  f      list files with current filename prefix
<num>   transfers a specific file <FILE_NAME_PREFIX>_0<num>.txt
  x      to exit
  u      update logger firmware
  g      show/change USE_GPS
  p      show/change PPS_TIMESTAMP
  m      show/change MAGDATA_PULSE_TIMESTAMP
  c      show/change LOG_AT_STARTUP (continuous logging)
  t      show/change GPS_TYPE
  b      show/change MAX_FILE_SIZE_BYTES
  s      show/change FILE_NAME_PREFIX string
Secondary commands:
  e      get error word
  v      firmware version
  r      logger reset
  n      name of next log file
  ?      show this command list
  h      show this command list
  y      print internal temp over BT
Enter:
```

2.1 Working with logger files

QLogMonCon can display logger files residing on the SD Card storage. Files can be transferred to the QuSpin Data repository on the host PC, and simultaneously converted and processed for input to the user's analysis and display applications.

The file created during a logging session follows a strict naming convention, where the logger filename consists of the following:

```
<FileNamePrefix>_<SequenceNumber>.txt
```

when start of logging is requested, the logger pre-allocates a file containing null characters that are replaced by log entries as logging progresses. The size of the file is determined by configuration setting, and is truncated to only contain logged data records when the logging session is terminated. To name the file, the logger forms a filename using the "current" value of *FileNamePrefix* (default is "TFM") and a *SequenceNumber* which is one more than the sequence number of the last log file containing the *FileNamePrefix*.

The following commands can be used to display and manipulate files on the SD card:

'f'	Displays a list of files beginning with the "current" FileNamePrefix.
's'	Display or change the value of the FileNamePrefix string.
<num>	Transfer and convert the single file with the following name: "<FileNamePrefix>_<num>.txt" <num> may be 1-4 digits (no need to specify the leading zeros that will exists in the name)
'n'	Displays the name of the log file for the next logging session.
'a'	Display a list of all files in SD card storage

QLogMon displays the logger files with the current filename prefix.

```
Enter: f
Num      Name           DateTime           Size
----      -
0        TFM_0000.txt      3/1/2024 10:00:12 AM  457600
1        TFM_0001.txt      3/18/2024 6:16:10 PM  322400
2 files of 2
Enter:
```

Note that the first column in the file list shows the integer *SequenceNumber* assigned to each file stored by the logger. The actual name of the file is in the second column. Only files beginning with the current value of *FileNamePrefix* are shown. Since files cannot be deleted from the SD card to safeguard data, the list will always show file names with indices from 0 to the last sequence number. The third column shows the date and time (UTC) that the file was preallocated. This timestamp will be accurate only if a GNSS fix was available that sets the logger system time.

Destination for the transfer/convert operations is in the `\Users\Public\Documents\QuSpin Data` folder.

Enter the integer at the prompt, and the file is transferred to the *repository* folder and used as input to the QLogConvert app which will generate the output .CSV files after conversion. The filename shown in the second column is the base filename for the output .CSV files which are stored in the *archive*, *flatCSV*, and *geospatial* sub-folders. A log of the operations is continually appended to a readable text file found in the *log* sub-folder.

The next section contains more detail about the output files generated during transfer.

2.2 Transfer and convert logger files

Log files are written to the SD card storage as ASCII text and will record verbatim with timestamp the QuSpin sensor data. QuSpin sensor responses to commands, status messages, star codes, etc are also timestamped and interleaved with mag data packets. If a GNSS module is connected, enabled in the config file, and operational, then time and position data is also logged. The logging file format is chosen for efficiency and reliability of logging, but is not so convenient for viewing or analysis. For this reason, post-processing of the log file is carried out during the process of transferring the file from SD card storage to PC host.

The following conversions are performed on the logger .txt file in the process of transferring to PC host:

- Strip null characters from the .txt file
- Sort all records by the microsecond timestamp and adjust for timestamp rollover
- Reformat into a flat file in .csv format
- Post-process data with geospatial conversion algorithm

The output of these conversions are written to ancillary files in separate sub-folders in the QuSpin Data folder. A verbatim copy of the original .txt file is retained in the Repository sub-folder of QuSpin Data. The original file also remains in SD card storage.

Example of output files produced in host folder **C:\Users\Public\Documents\QuSpin Data** by transfer/conversion of SD log file TFM_0014.txt. Note that each output file name contains the GPS datetime from the log file corresponding to the start of logging.

<sub-folder>\<file name>	Description
Repository\TFM_0014.txt	Copy of unmodified log file
.\TFM_0014_20230226_1807_UTC-090752.csv	Processed log file sorted by uS timestamp, one record per line
archive\TFM_0014_20230226_1807_UTC-090752_raw.csv	Original log file stripped of null characters.
flatCSV\TFM_0014_20230226_1807_UTC-090752_flat.csv	Output .csv file in flat format (single data table with unique columns)
geospatial\GPS.Interpolated _TFM_0014_20230226_1807_UTC-090752_flat.csv	Geospatial file with heading error correction, processed from flatCSV file input.

2.3 Configuring the logger firmware

The following commands can be used to configure the operation of the logger:

'g'	Show/change whether an attached GPS module should be used to track location during logging and timestamp log entries. If true, the attached GPS module will be used to set the realtime clock for file date/time, and also time and location information will be logged with the mag data.
'p'	Show/change whether the GNSS PPS (pulse per second) signal edge from the GNSS module should be used to establish the precise timing of incoming GNSS packets. Use of the PPS signal for timing is more accurate than the receipt of the first byte of a GNSS record. Set this to TRUE only if the GNSS module has a dedicated PPS output pin and it is wired to the ACB. The AdaFruit GPS devices have a dedicated PPS pin.
'm'	Show/change whether the sensor's magdata pulse edge should be used to timestamp the incoming sensor data. Use of the mag data pulse for timing is more accurate than the receipt of the first byte of a sensor data record.
'c'	Show/change the value of the LOG_AT_STARTUP flag. It is recommended to keep this flag set to FALSE to avoid unnecessarily filling up the SD card storage. Set this to "true" only if continuous logging is a requirement for the sensor application. If true, logging starts automatically on bootup and stops with a button press or software command.
't'	Show/change the GPS type. This value must be set accurately according to the specific GNSS module attached to the sensor. Use one of the following setting to specify the GPS chip contained on attached GPS module: 1 - Adafruit GPS devices PA1616D, PA6H 2 - NEO-M8N 3 - NEO-M9N 4 - Adafruit GPS device PA1616S
'b'	Show or change the value of the MAX_FILE_SIZE_BYTES parameter. File of this size is preallocated and filled with nulls and replaced with data as logging progresses. File is truncated when logging is stopped. Logging stops if this size is reached. Default max size is 0.5G if this key is not specified. MAX_FILE_SIZE_BYTES = 67108864 allows logging for 3.5 hrs at max mag rate of 5290 B/sec.

2.4 Update logger firmware to the latest version

Every installation of the QLogMonitorConsole application is packaged with the latest logger firmware image.

'v'	Displays the firmware version currently running in the logger.
'u'	Update the logger firmware with the packaged firmware image. The process is completely automated including the required restart of the QLogMonitorConsole app, which is required to reestablish the session..

2.5 Reset the logger

A reset command is provided to force the logger firmware to reboot.

'r'	Launches a process to reboot the logger firmware. This command requires a relaunch of the host app to re-sync the session. The process is completely automated including the required restart of the QLogMonitorConsole app.
-----	--

2.6 Viewing the Logger Error Status

Logger Error Status is a hexadecimal value that can be observed 1) from the QLogMonCon app, and 2) over the BlueTooth serial link where it appears along with the Logger state, also a number.

The Logger Error Status hex value represents a bitfield, where each bit is dedicated to a logger status item. In the Error Status word, if the bit is set, it indicates that there is an error associated with that item. These items are checked at startup and during prearm checks. An Error Status of 0 is required for the logger to move to the READY state. Therefore the error condition must be cleared before logging can begin.

Note that some of the items in the bitfield are masked if not configured. Therefore, error conditions can be caused by improper configuration of the logger. For example, the configuration of the GNSS PPS signal must not be set to TRUE unless a GNSS module providing an external PPS signal is connected, is powered on, and a fix is obtained. If this configuration item is set to TRUE, then the logger checks for the presence of this signal and sets the ppsSignalDetected bit in the Error Status word. If this configuration item is set to

FALSE, the bit is masked and will not indicate an error, regardless of the state of the physical signal.

The following table indicates the correspondence between bit position and error indication. Bit 0 is the least-significant bit in the bitfield.

Bit position	Description
0	configFileOk
1	masterDataDetected
2	slaveDataDetected
3	sensorSyncDetected
4	ppsSignalDetected
5	magdataPulseDetected
6	sdCardWriteable
7	sdCardSpaceAvailable
8	gpsStatusStdOrBetter
9	gpsTimeValid
10	gpsDateValid
11	gpsFixValid

3 Logger firmware functional description

The logger operation is controlled by a state machine that moves through initialization, startup checks, and configuration.

If any errors occur during this process they are recorded in the Logger Error Status, and the error is communicated via the LED indicator as well as the communications port. (See sections on Monitoring Logger Status and Viewing the Logger Error Status) Then logger firmware moves to an error recovery state depending on the nature and severity of the error.

If this process proceeds without error, the firmware moves to an “armed” state where resources are fully prepared and the device is ready to begin logging. This state is referred to as the READY state.

On applying power or reset, the logger firmware flashes the LED and goes through the following startup process in preparation for logging of sensor data.

1. Initialize communications ports including connection to sensor data stream
2. Verify the ability to create and update files on the built-in SD card
3. Acquire and apply configuration settings
4. Set the system clock. If the logger is configured to use GPS, the realtime clock and file date/time is UTC
5. Configure and start periodic tasks such as status updates, sanity checks, etc
6. Check all required pre-arm conditions before arming. For example, if the logger is configured to use the GPS PPS, firmware checks for its presence before arming. Similar checks are done for GPS location, sensor data stream, etc.
7. Move to READY state and monitor for a command to start logging.

Once the logger is in the READY state, logging may be initiated, either by software command or by physical button press. Once logging is started, the firmware continually monitors and records the data streams to SD card file until a command to stop logging is received.

The logger is able to respond to software requests to display and/or transfer log files in all states except when actively logging sensor data. Host software and firmware structure prevents such requests from being acted upon if actively logging.

4 Monitoring Logger status

The status of QTFM logger firmware operation is indicated by a dedicated LED on the ACB2 PCB, as described in the next section.

The status of the QTFM logger can also be monitored via the message stream sent over the BlueTooth port.

Pair a host to the Bluetooth serial module and the stream of ASCII serial messages may be visible in a Terminal emulator such as TeraTerm. This status information is not required for normal operation but may be useful to verify or troubleshoot operation.

4.1 LED Indicator

A dedicated LED on the ACB2 PCB indicates the status of the logger as well as any error conditions that exist.

LED state	Indication
Steady YELLOW	In a prearm state, executing prearm checks
Flashing RED	Specific error condition is indicated by the number of red flashes (see table below)
Flashing BLUE	Ready to begin logging, prearm checks have passed.
Steady BLUE	Actively logging
BLUE-RED-BLUE-RED-BLUE	Firmware boot

Logger error indications:

RED FLASHES	INDICATION
2	Configuration error
3	Sensor data not detected
4	PPS signal configured but not detected
5	Mag data pulse configured but not detected
6	GPS issue or no fix

4.2 BlueTooth message stream from logger

In the READY state, the logger automatically configures the BlueTooth module to continuously stream sensor data over the BlueTooth serial port. However if the logger prearm checks detect any errors preventing the logger from being READY to begin logging, the logger will set the BlueTooth serial port source to stream informative logger status messages.

5 Viewing the log files using legacy QLogConverter

Log files written as text will contain a verbatim dump of everything the QuSpin mag transmitted. Responses to commands, status messages, star codes, etc are timestamped and interleaved with mag data packets. If a GPS module is connected, enabled in the config file, and operational, then time and position data is also logged. The logging file format is chosen for efficiency and reliability of logging, but is not so convenient for viewing or analysis. For this

reason, some post-processing of the log files is necessary before viewing the mag data, and a utility is provided for this purpose.

To view the log file(s)

1. Terminate logging by momentarily pressing the blue switch **before** removing the card
2. Transfer the logger files to a repository folder on the PC
3. Locate and open the install folder for the QLogConverter application
(C:/Users/Public/Public Documents/QuSpin Data) Drag and drop a log file (.txt) onto the QLogConverter.exe file (or a shortcut on your desktop). A .csv version (comma-delimited) of the file will be created that is readable by Excel.

Once the file is in CSV format, another issue arises for large files: Excel maximum worksheet size of 1048576 ($=2^{20}$) rows by 16384 columns. QTFM logger can log very large files but they cannot be handled by Excel which just truncates them.

For example, a log file size of ~54.2MB, corresponds to 4,169,230 records or spreadsheet rows and Excel throws out the last 3M rows.

[Solutions, workarounds actually, to the problem of large CSV/xls files](#)