

This manual is for the QLG2 GPS/GNSS Receiver module
Firmware version **1_00.QG2, 1_00a.QG2 and 1_00b.QG2**

Contents

1. Introduction.....	2
2. Getting to know your QLG2.....	2
3. Assembly.....	5
3.1 Assembly for basic operation.....	5
3.2 Use with QCX, U3S, Clock, VFO, ProgRock etc. kits.....	6
3.3 USB connector.....	6
3.4 USB to Serial converter.....	7
3.5 Powering the board from the USB port.....	10
3.6 Status LEDs.....	11
3.7 Fitting an ultracap (a.k.a. supercap).....	12
3.8 6-pin interface header.....	13
3.9 Jumper wire configuration.....	14
3.10 Interface configuration.....	14
3.11 Microcontroller configuration.....	16
3.12 Connecting an LCD module.....	17
3.13 QLG2 Commands.....	19
4. Firmware update procedure.....	20
5. Circuit diagram (schematic) and description.....	23
6. Testing.....	26
7. Acknowledgments.....	27
8. Resources.....	27
9. Document Revision history.....	27



1. Introduction

The QLG2 GNSS (Global Navigation Satellite System) receiver module is an upgrade over the former QLG1 with higher performance and highly flexible features:

- Multi GNSS satellite constellation receiver supports GPS (US), Galileo (Europe), GLONASS (Russia) and Beidou (Chinese) - (default GPS + Beidou) - giving a faster more accurate position fix – Module is E108-GN01 (GK9501 GNSS chip set)
- Serial to USB converter, using onboard USB B connector, for convenient interface to PC software if required - no drivers required. Jumpers select either the GPS serial data, or your own externally connected serial data, for example you could use this as a USB to Serial data converter for CAT control of QCX, QCX+ and QCX-mini transceivers.
- Can optionally be connected to a standard 1602 LCD, in the same style as the Ultimate3S, VFO and Clock kits, to GPS date, time and satellite information using the onboard microcontroller.
- Supplied with included magnetic-mount active antenna, approx 2m coax and SMA connectors
- Supply voltage range 3.3 to 6V.
- Board 80 x 37mm (Same as, and compatible with, Ultimate3S, VFO and Clock kits).
- Onboard LEDs for status indication: Red (Power), Yellow (Serial data) and Green (PPS).
- 2.8 to 5V logic level conversion to provide 5V serial data for full compatibility with all QRP Labs kits.
- Supports 2.8/3.3V logic OR 5V logic (jumper wire selected)
- SMD assembly already undertaken by factory facility - only SMA connector, and optional USB B connector to solder.
- Space provided for optional ultra-capacitor for faster hot-start

All QLG2 GPS modules are comprehensively tested before shipment including:

- Active antenna with 2m coax and SMA connector functions correctly
- Satellite reception and Time To First Fix is under 1 minute
- Serial data and 1pps outputs are correctly present
- USB to Serial data converter operates properly (view serial data in PC terminal)
- Microcontroller firmware update procedure test

PLEASE READ THE BASIC ASSEMBLY AND USE INSTRUCTIONS IN THIS MANUAL VERY CAREFULLY BEFORE APPLYING POWER TO THE BOARD!

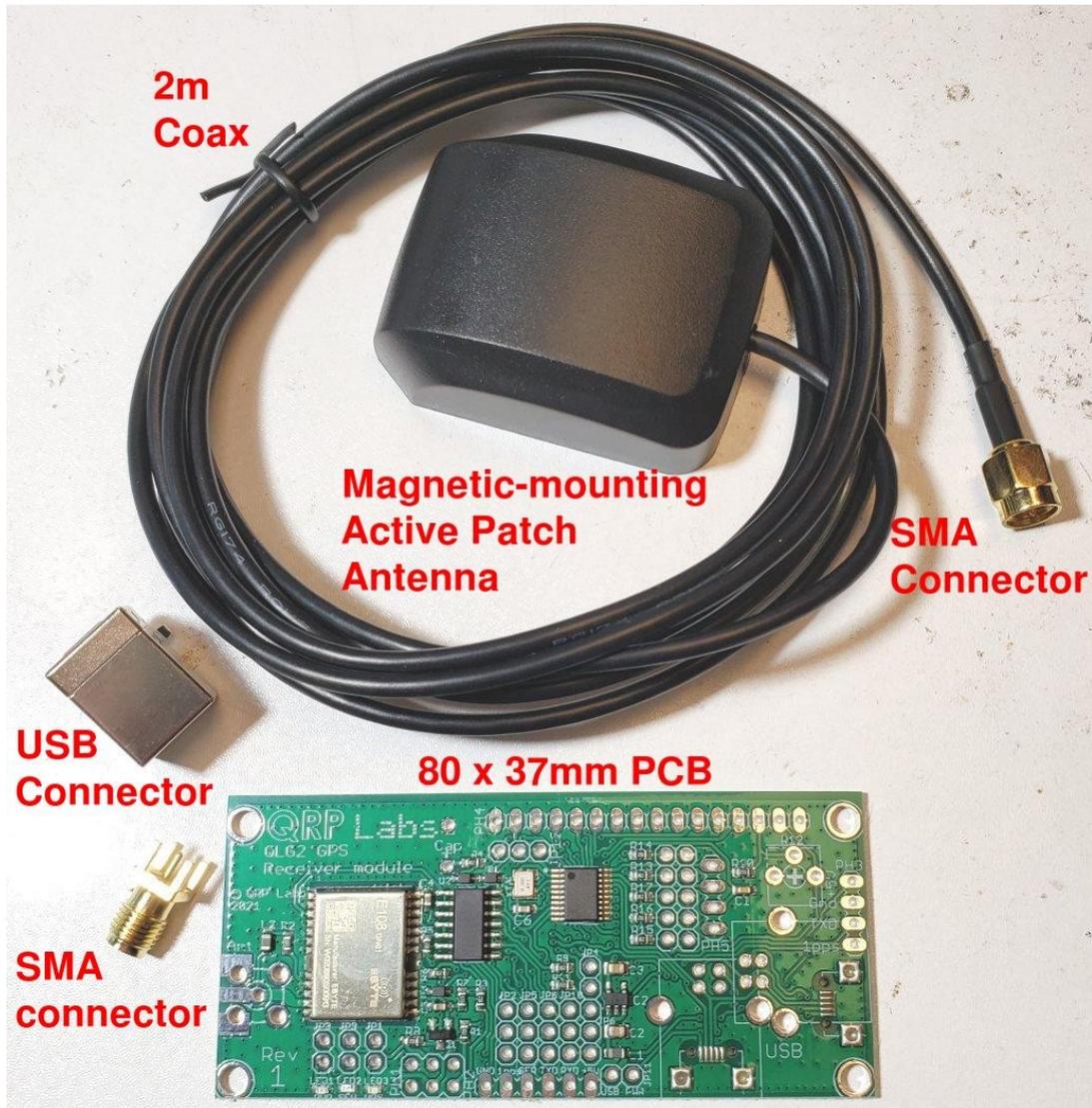
2. Getting to know your QLG2

There are only FOUR parts in the QLG2 kit:

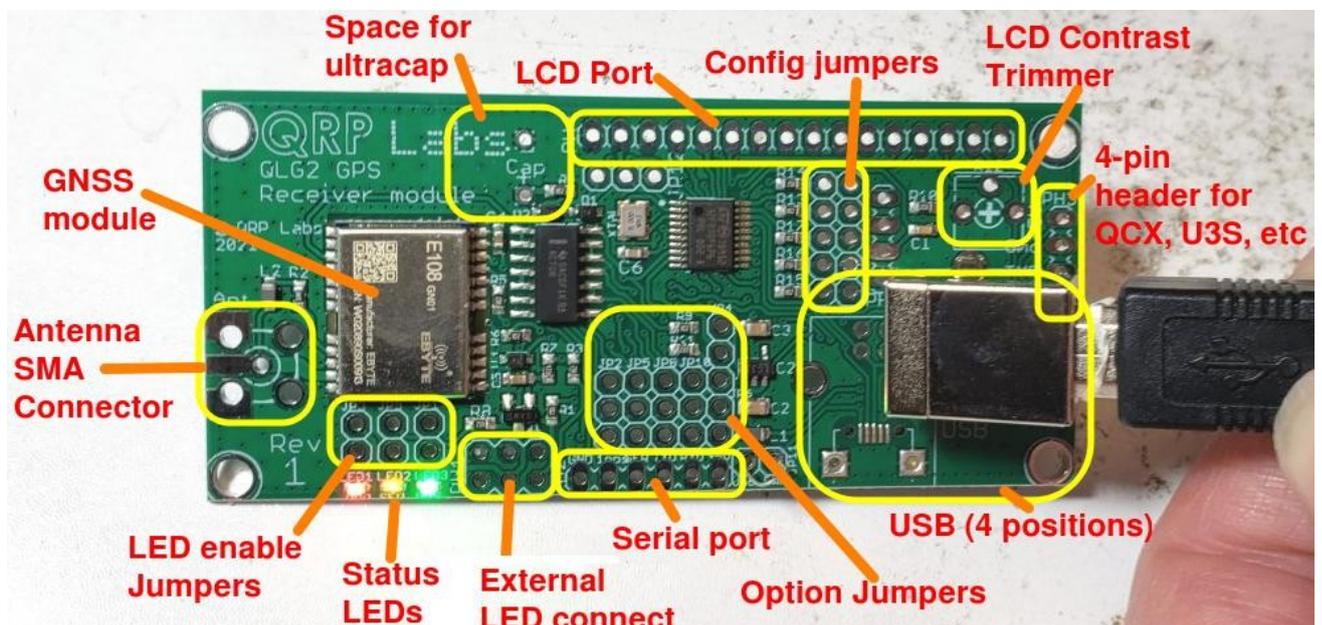
1. 80 x 37mm PCB with assembled SMD
2. Active antenna including 2m coaxial cable and SMA connector
3. SMA connector for installation on the PCB
4. USB-B type connector for optional installation on the PCB-mount

The parts list of the PCB is provided later in this manual in the circuit description section.

Kit contents:



Know your QLG2:



With reference to the above photograph, looking anti-clockwise from left:

- GNSS Module: The E108-GN01 GNSS receiver module, multi-satellite system capable (GPS and Beidou enabled by default).
- Antenna SMA Connector (supplier): Can be board edge-mounted or at 90-degrees standing up on the board (**DO refer to important assembly instructions later**).
- LED enable jumpers: these are wired by default, and connect the signals to the onboard LEDs. If you wanted to use external LEDs for example in a front panel, you would cut the tiny traces under these jumpers (see later section).
- Status LEDs:
 - Power (RED): always lit
 - Serial Data (Yellow): flashes once per second in time with the serial data burst
 - 1 PPS (Blinding Green): flashes once per second when a 3D satellite lock is acquired
- External LED connect: Three LEDs may be connected here using wires if you prefer external LEDs to the onboard SMD LEDs.
- 6-pin interface: this provides six pads:
 - GND
 - GNSS 1pps
 - GNSS serial data
 - TxD serial data (for USB to Serial converter)
 - RxD serial data (for USB to Serial converter)
 - +5V
- Option Jumpers: configure how the six serial port pads are connected, allowing either 2.8/3.3V logic or 5V logic levels
- USB (4 positions): a Type B USB connector is supplied, suitable for a standard USB A – B cable. It can be installed either facing to the right side of the board, or facing the bottom side of the board. Additionally there are pads for a micro-USB connector if you should happen to prefer that.
- 4-pin interface: pads for QCX, U3S etc: the standard QRP Labs 4-pin connector: +5V, Gnd, Serial Data (TxD) and 1PPS
- LCD Contrast trimmer: When installing the optional 1602 alphanumeric LCD module, a 22K trimmer potentiometer is fitted here to permit contrast adjustment
- Config jumpers: these are read by the microcontroller at power-up, and determine configuration options such as baud rate; they can also be used to enter the bootloader (firmware update) mode
- LCD Port: The pinout is in the exact same position as on the QRP Labs Ultimate3S, Clock or VFO kits; a standard 1602 alphanumeric LCD module can be bolted onto the board here, using 16-pin male/female pin header connectors, 12mm plastic hex spacers and eight M3 6mm plastic screws (as on the aforementioned QRP Labs kits).
- Space for Ultracap: 5mm-spaced pads suitable for installation of an Ultracap for ephemeris backup purposes; this provides faster hot-start.

The board size is 80 x 37mm, and the holes in the four corners are at exactly the same positions as on the Ultimate3S, Clock and VFO kits; therefore the board can be stack-mounted behind these kits on suitable spacers (12 or 25mm).

These features will be described in more detail in subsequent sections.



Compatibility

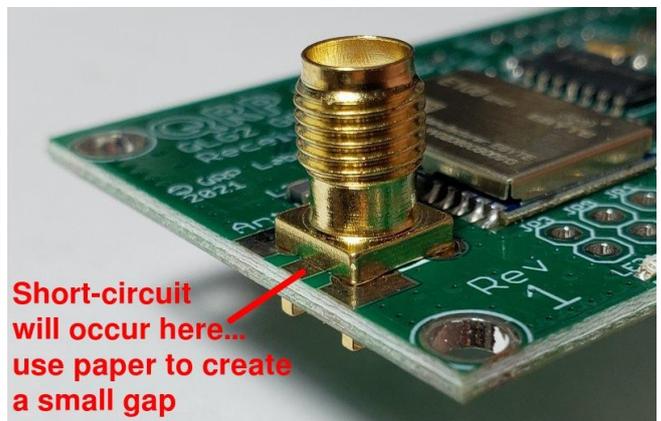
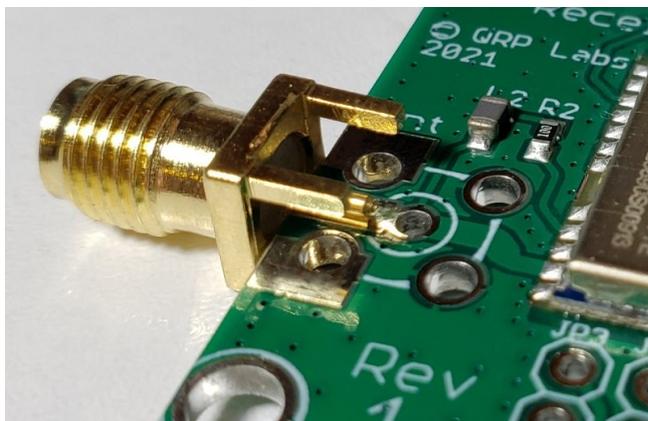
Straight out of the box, if you simply install the coaxial connector and plug in the antenna, the QLG2 is backwardly compatible by default with the former QLG1 kit. It is therefore directly compatible with all other QRP Labs kits:

- QCX-series CW transceivers (QCX, QCX+ and QCX-mini)
- Ultimate3S (and preceding kits in the Ultimate- series)
- Clock kit
- VFO/Signal generator kit
- ProgRock kit

3. Assembly

3.1 Assembly for basic operation

Assembly of this kit is EXTREMELY straightforward; at the minimum, you need only solder in the SMA connector. This can be installed horizontally or vertically (below left and right, respectively), to suit your particular application.



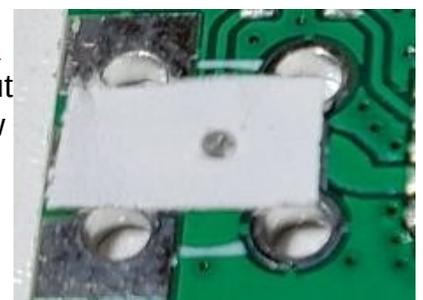
Vertical installation: IMPORTANT! The body of the SMA connector WILL DEFINITELY short to the center pad of the component footprint on the board. This will not only short out the antenna, rendering it completely ineffective, it will also draw high current from the GNSS module through the 10-ohm resistor R2. You will soon note the temperature rise of the GNSS module as well as the 3.3V voltage regulator chip (you don't need to ask me how I know...).

Therefore it is essential to create a little gap between the body of the SMA connector and the board.

The best way to do this is via small piece of paper. It's a simple, cheap and effective solution.

Cut a small piece of paper as shown (above right), place it on the board and push the center pin of the SMA connector through the paper.

Solder the SMA connector in place as usual. Don't overdo the heat and burn the paper.



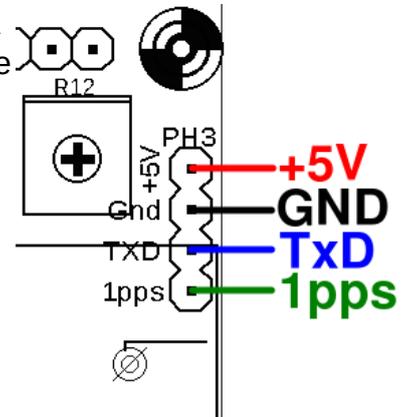
Horizontal installation: The SMA connector happens to fit the board nicely too in the horizontal, edge-mounted position. Place it with the center pin on the top side of the board, and two of the corner pins on the bottom side, and solder in place. The other two ground pins can be ignored. This works very well.

3.2 Use with QCX, U3S, Clock, VFO, ProgRock etc. kits

In this application, you need only plug in the antenna coax SMA connector and the QLG2 is ready to use. It is entirely compatible with the former QLG1 kit and the 4-way pin header pads at the right side of the board can be used in the same way as a QLG1 kit.

+5V power can be supplied to the module via the +5V connection if desired.

These signals are also available on the 6-way pin header pads at the bottom of the board edge. If you install the USB-B connector in the position facing out from the right-hand edge of the board, the body of the connector will block access to the 4-pin header. In this case, if you wish to connect to these 4 signals, you can do so at the 6-pin interface



NOTE: The 4-pin header “TXD” connection is not the same as the “TXD” connection on the 6-pin header at the bottom board edge. On the bottom edge connector (6-pin) the signals named TXD and RXD are the microcontroller serial ports that access the USB to Serial converter. On the 6-pin connector, the pin labeled “SER” on the PCB silkscreen, and “GPS_TXD” on the schematic, is actually connected to the pin labeled “TXD” on the 4-pin header.

3.3 USB connector

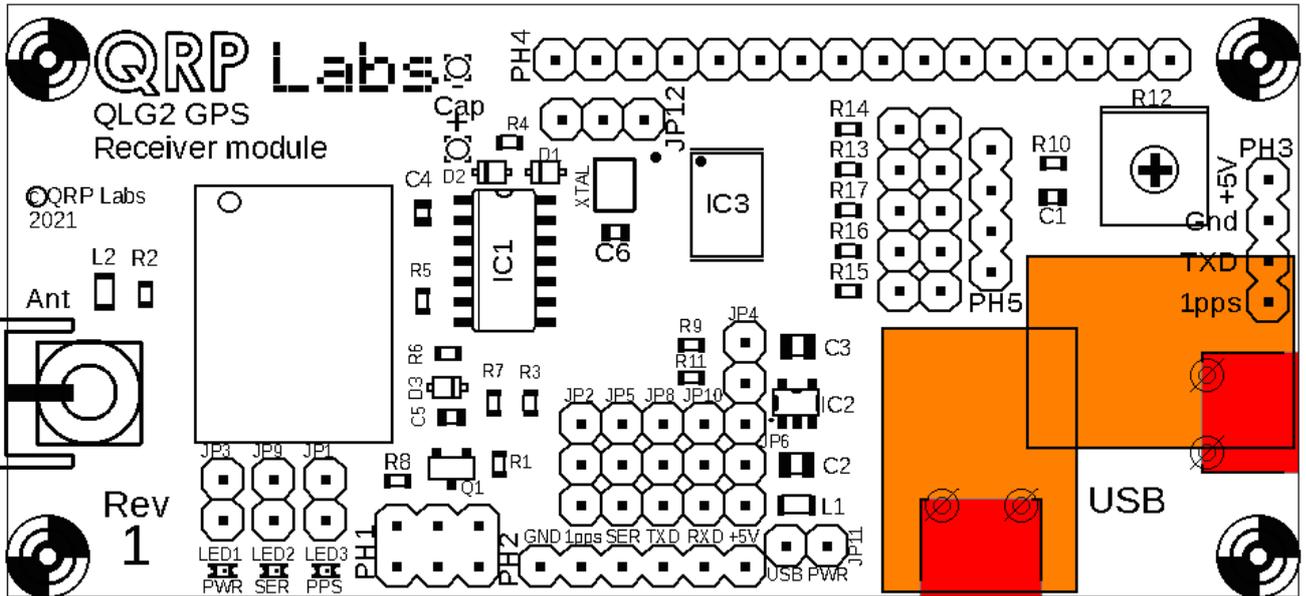
For all but the most basic use of the QLG2, you will likely wish to install the USB B connector (supplied) or if you prefer, a micro-USB connector (not supplied).

The USB connector is used for two purposes:

1. To provide a USB to Serial converter, which can be used either to stream the GNSS NMEA serial data to the PC, or as a separate USB to Serial converter for other purposes, for example, for providing CAT control to a QCX-series CW transceiver kit. (This is described in a later section).
2. In bootloader mode, the USB connector enables the functionality of the QRP Labs Firmware Update (QFU) procedure. (This is described in a later section).

In the diagram below, you will note that the supplied USB-B connector can be installed with its socket facing out from the bottom of side of the PCB, or from the right side. These are coloured orange in the diagram. There are also pads provided for a micro-USB connector (not supplied), again facing down or to the right; these are coloured red in the diagram.

Which of these positions you choose to install the connector in, has no bearing on the operation of the unit; it is only a matter for you to choose whatever will be most suitable in your enclosure or application.



3.4 USB to Serial converter

The primary application of the microcontroller on the QL2 board, is to provide a USB to Serial converter.

The microcontroller is programmed to appear as a standard Virtual COM Port USB device class, which means that no additional drivers are required on any modern operating system. The standard drivers are always already included. This greatly simplifies operating the device. In comparison to some other Serial to USB converters which end up in an endless fight with driver issues, QL2 “just works”.

There are many Terminal Emulator programs available for PCs. Personally I normally use PuTTY, which is available for both Windows and Linux operating systems. But it does not matter, any favourite Terminal Emulator will work fine. The Serial Monitor of the Arduino IDE is another possibility which I use sometimes.

On opening the Terminal Emulator you will need to choose the USB Virtual COM port to connect to; this is different depending on whether you are using Windows or Linux.

1. On Linux, the QL2 will appear as **/dev/ttyACM0** – or perhaps ACM1, ACM2 etc if you already have similar devices connected to your system.
2. On Windows, the QL2 will appear as a COM port for example, COM1, COM2 etc. I don't know of any easy way to find out which COM port number Windows has assigned, except to open the device manager and look at the listed USB devices.

Note that it does not matter what baud rate you select in the terminal emulator, since USB does not run at any particular rate, and the baud rate of the QL2 USB to Serial converter is determined by jumpers on the board (see later section).

On connecting the terminal emulator, you will see lots of text scroll by, in bursts once per second; these are the NMEA sentences output by the QL2's GNSS module. By default, the QL2 jumpers are configured to route the GNSS module's serial data to the microcontroller's USB to Serial port.

The following screenshot shows an example of what it could look like.

```
/dev/ttyACM1 - PuTTY
$BDGSA,A,3,11,09,06,16,14,21,26,28,,,,,1.45,0.79,1.21*1F
$GPGSV,3,1,12,01,78,174,48,22,67,042,49,03,60,329,44,21,58,162,46*76
$GPGSV,3,2,12,04,52,234,36,40,39,140,35,31,37,075,47,17,25,294,35*7A
$GPGSV,3,3,12,09,17,233,13,19,14,316,23,08,02,187,,32,01,050,*7C
$BDGSA,A,3,1,09,14,57,300,38,11,42,208,39,09,37,047,43,21,29,063,50*6B
$BDGSA,A,3,2,09,26,26,117,46,28,26,309,31,16,16,043,47,06,11,043,44*66
$BDGSA,A,3,3,09,24,09,162,32*5A
$GNRMC,194456.000,A,3650.30309,N,02838.45420,E,0.000,38.33,190521,,,A*71
$GNVTG,38.33,T,,M,0.000,N,0.000,K,A*18
$GNGGA,194457.000,3650.30309,N,02838.45420,E,1,16,0.79,2.481,M,34.507,M,,*74
$GPGSA,A,3,04,31,19,17,01,22,03,21,,,,,1.45,0.79,1.21*07
$BDGSA,A,3,11,09,06,16,14,21,26,28,,,,,1.45,0.79,1.21*1F
$GPGSV,3,1,12,01,78,174,48,22,67,042,49,03,61,330,44,21,58,162,46*7F
$GPGSV,3,2,12,04,52,234,36,40,39,140,35,31,37,075,47,17,25,294,36*79
$GPGSV,3,3,12,09,17,233,14,19,14,316,24,08,02,187,,32,01,050,*7C
$BDGSA,A,3,1,09,14,57,300,38,11,42,208,39,09,37,047,43,21,29,063,50*6B
$BDGSA,A,3,2,09,26,26,117,46,28,26,309,31,16,16,043,47,06,11,043,44*66
$BDGSA,A,3,3,09,24,09,162,32*5A
$GNRMC,194457.000,A,3650.30309,N,02838.45420,E,0.000,38.33,190521,,,A*70
$GNVTG,38.33,T,,M,0.000,N,0.000,K,A*18
$GNGGA,194458.000,3650.30309,N,02838.45420,E,1,16,0.79,2.481,M,34.507,M,,*7B
$GPGSA,A,3,04,31,19,17,01,22,03,21,,,,,1.45,0.79,1.21*07
$BDGSA,A,3,11,09,06,16,14,21,26,28,,,,,1.45,0.79,1.21*1F
$GPGSV,3,1,12,01,78,174,48,22,67,042,49,03,61,330
```

Using a Terminal Emulator for testing CAT on a QCX-series CW transceiver

Remember that in any serial communications operation of this type, the TxD (transmit) output of the host controller, has to be connected to the RxD (receive) input of QCX; correspondingly the RxD (receive) input of the host controller has to be connected to the TxD (transmit) output of QCX. In other words, you have to get your wires crossed – correctly.

An important note is required here. QCX firmware requires that incoming CAT commands are a rapid succession of characters arriving at 38400 baud. The permissible delay between characters of a command is a fraction of a millisecond. You certainly cannot type that fast. Even I can't type that fast. Therefore, any text you type into a PuTTY (or most other) terminal emulators will not produce the desired result on the QCX since the QCX firmware will time while out waiting for the next character. (Note that normal CAT host programs do not have this issue, they send commands as a rapid burst of characters).

There are two possible ways around this:

1. Type the command you want to send to QCX, in another window such as a text editor of some sort; then copy-and-paste it into the Terminal Emulator window. This will ensure that the whole text sentence is sent with the characters in quick succession, not leaving gaps where the QCX times out.
2. Use the Serial Monitor application inside the Arduino IDE – this also functions as a very basic, very limited, Terminal Emulator – but it has the quirk (useful in this case) that there is a separate box at the top of the window where you type the text to be sent, and it does not actually get sent until you hit return or click the “Send” button. So again, this ensures that the text is all sent in one rapid lump, which is acceptable to the QCX.

An example screenshot of the Arduino Serial monitor is shown below, again showing the NMEA sentences scrolling past.

```
/dev/ttyACM1
$GPGSV, 3, 2, 10, 21, 53, 164, 46, 40, 39, 140, 35, 31, 36, 069, 50, 17, 25, 289, 34*78
$GPGSV, 3, 3, 10, 09, 22, 237, 23, 19, 16, 311, 14*7F
$BDGSV, 3, 1, 12, 14, 59, 307, 36, 11, 37, 206, 38, 05, 37, 136, 39, 09, 35, 046, 44*68
$BDGSV, 3, 2, 12, 26, 29, 113, 48, 28, 28, 306, 32, 21, 26, 067, 49, 02, 19, 112, 38*66
$BDGSV, 3, 3, 12, 16, 15, 044, 45, 24, 12, 159, 39, 06, 10, 044, 43, 12, 01, 170, *6F
$GNRMC, 195541.000, A, 3650.30309, N, 02838.45420, E, 0.000, 38.33, 190521, , , D*72
$GNVTG, 38.33, T, , M, 0.000, N, 0.000, K, D*1D
$GNGGA, 195542.000, 3650.30309, N, 02838.45420, E, 2.19, 0.60, 2.481, M, 34.507, M, , *74
$GPGSA, A, 3, 04, 31, 19, 17, 09, 01, 22, 03, 21, , , , 0.92, 0.60, 0.70*08
$BDGSA, A, 3, 02, 11, 09, 24, 16, 14, 21, 26, 28, 05, , , 0.92, 0.60, 0.70*1E
$GPGSV, 3, 1, 10, 01, 73, 172, 45, 22, 65, 054, 50, 03, 63, 339, 42, 04, 57, 240, 38*7F
$GPGSV, 3, 2, 10, 21, 53, 164, 46, 40, 39, 140, 35, 31, 36, 069, 50, 17, 25, 289, 34*78
$GPGSV, 3, 3, 10, 09, 22, 237, 23, 19, 16, 311, 15*7E
$BDGSV, 3, 1, 12, 14, 59, 307, 36, 11, 37, 206, 38, 05, 37, 136, 39, 09, 35, 046, 44*68
$BDGSV, 3, 2, 12, 26, 29, 113, 48, 28, 28, 306, 32, 21, 26, 067, 49, 02, 19, 112, 38*66
$BDGSV, 3, 3, 12, 16, 15, 044, 45, 24, 12, 159, 39, 06, 10, 044, 43, 12, 01, 170, *6F
$GNRMC, 195542.000, A, 3650.30309, N, 02838.45420, E, 0.000, 38.33, 190521, , , D*71
$GNVTG, 38.33, T, , M, 0.000, N, 0.000, K, D*1D
$GNGGA, 195543.000, 3650.30309, N, 02838.45420, E, 2.19, 0.60, 2.481, M, 34.507, M, , *75
$GPGSA, A, 3, 04, 31, 19, 17, 09, 01, 22, 03, 21, , , , 0.92, 0.60, 0.70*08
$BDGSA, A, 3, 02, 11, 09, 24, 16, 14, 21, 26, 28, 05, , , 0.92, 0.60, 0.70*1E
$GPGSV, 3, 1, 10, 01, 73, 172, 45, 22, 65, 054, 50, 03, 63, 339, 42, 04, 57, 240, 38*7F
$GPGSV, 3, 2, 10, 21, 53, 164, 46, 40, 39, 140, 35, 31, 36, 069, 50, 17, 25, 289, 34*78
$GPGSV, 3, 3, 10, 09, 22, 237, 23, 19, 16, 311, 12*79
$BDGSV, 3, 1, 12, 14, 59, 307, 36, 11, 37, 206, 38, 05, 37, 136, 39, 09, 35, 046, 44*68
$BDGSV, 3, 2, 12, 26, 29, 113, 48, 28, 28, 306, 32, 21, 26, 067, 49, 02, 19, 112, 38*66
$BDGSV, 3, 3, 12, 16, 15, 044, 45, 24, 12, 159, 39, 06, 10, 044, 43, 12, 01, 170, *6F
$GNRMC, 195543.000, A, 3650.30309, N, 028
```

Using a Terminal Emulator for sending commands to the E108-GN01 GNSS module

By default, the USB to Serial converter is connected to the GNSS receiver module. The module accepts a variety of commands which can be used, among other things, for:

- Changing the baud rate (default is 9600 baud)
- Changing which NMEA sentences are output
- Enabling or disabling the various GNSS systems (GPS, Galileo, GLONASS and Beidou)

You can type the command in full on the Terminal Emulator, and it will be transferred to the GNSS module by the QLG2 USB to Serial converter.

A common error is to not calculate the checksum properly; these are the last two characters of the command string. The actual command is prefixed by a \$ and terminated by a * followed by the two checksum characters. All the ASCII byte codes of the command between the \$ initiator and * terminator, but not including the \$ and the *, are logically XOR'ed together, which produces the checksum.

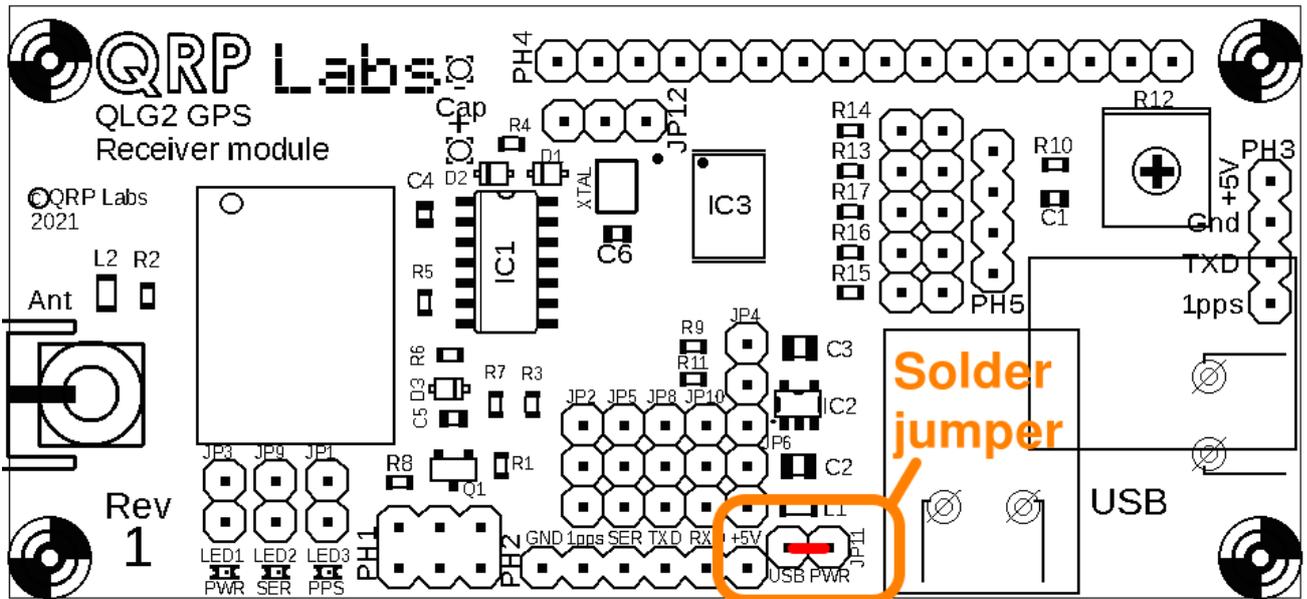
If this seems like too much of a headache, *and actually it really IS too much of a headache*, then asking Google about “NMEA checksum calculator” will find you lots of easy calculators that do this for you. For example <https://nmeachecksum.eqth.net/> .



3.5 Powering the board from the USB port

In some applications, you may find it convenient to power the QLG2 board from the USB port.

In this case, simply solder a jumper wire across JP11 as shown in the diagram below.

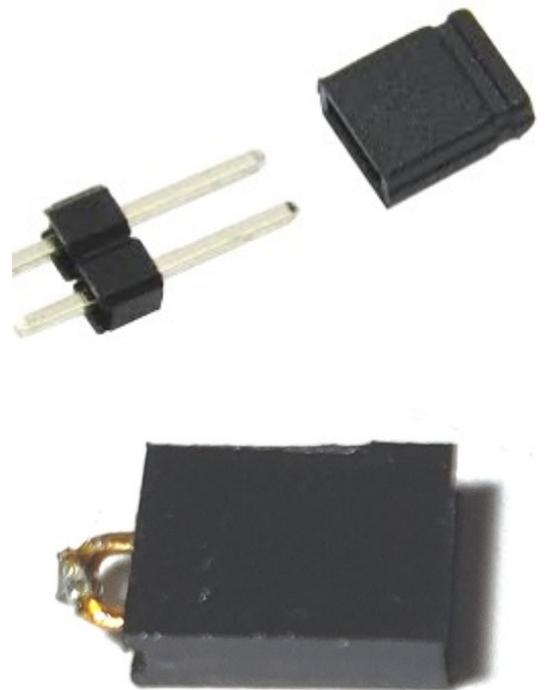


Tip: I am in the habit of making my soldered jumper wires as a little loop, standing up about 5mm from the board surface; this is because even if I am 100% sure now, that this will be a permanent connection... all too often I find for whatever reason, I will change my mind later. To make it easier to remove, or to temporarily disable, a jumper wire – I just make a 5mm-high loop rather than a wire laying near the PCB surface. **The 5mm loop is a lot easier to cut and if necessary, to remove, than a wire tight flat against the board surface.**

If you already feel now that it will not necessarily be a permanent connection, then fit a 2-way pin-header and use a header jumper, as pictured (right).

Now you might be thinking, “I have pin headers, but I don’t have those fancy jumpers you used to find (and maybe still do) on old PC Motherboards, and I don’t fancy waiting for Amazon, eBay, whoever”... well never fear, for necessity is the Mother of invention – I don’t have those fancy jumper plugs either, I just improvise using female pin header connectors. Which aren’t even the right right size (2-way) but can be cut to leave just two pins... and then gently bend over the two pins towards each other and solder in the middle (see photo, right).

IMPORTANT: never connect the jumper to power from USB, and at the same time supply +5V power from some other source to the other connection pads! NEVER, EVER!



3.6 Status LEDs

Just like its predecessor the QLG1, the QLG2 module has the same three LED status LEDs.

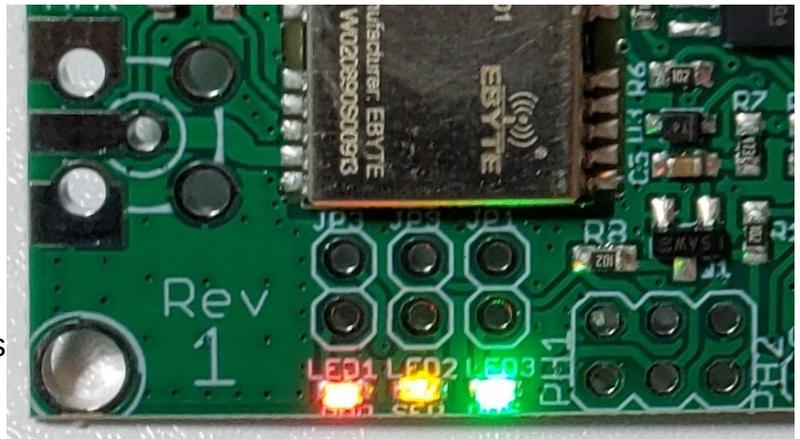
In the case of QLG2, these are 0603-size SMD LEDs installed on the PCB near the GNSS receiver module.

LED 1 (red) is the Power LED and is always lit when QLG2 is powered.

LED 2 (yellow) is the Serial data LED and pulses in time with the serial data. Note that in a departure from the QLG1, this LED is actually OFF during the data burst (whereas on QLG1 it is ON during the data burst). The reasons for this are explained later, in the circuit explanation section.

LED 3 (blinking green) is the 1pps indicator and flashes once per second, for 0.1 seconds, coincident with the 1pps pulse whose leading edge indicates the exact UT second. This LED only starts blinking once a satellite lock (3D fix computation) has been achieved.

Together, these three LEDs provide an accurate diagnosis of correct operation of the QLG2.



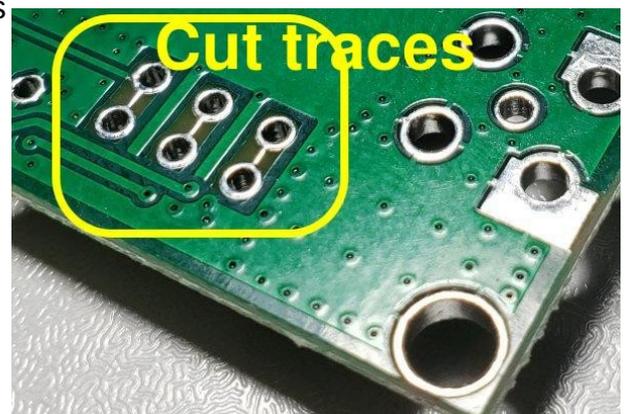
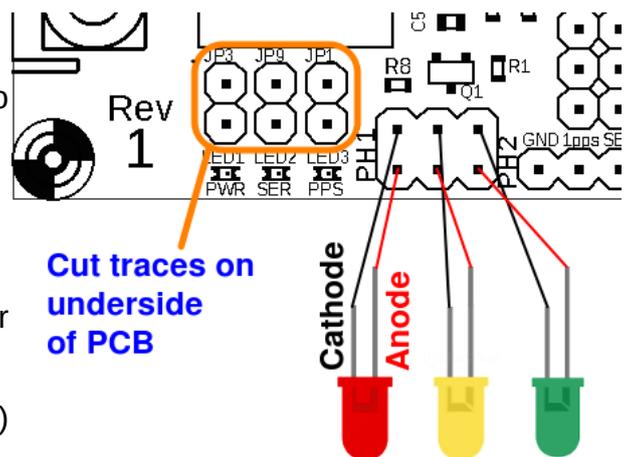
Connecting external LEDs

You may wish to use external LEDs instead of the onboard ones. For example, you may wish to make the LEDs visible on the front panel of your equipment. After all, which of us does not like blinking lights on his equipment, and generally the more the merrier.

In this case, you can connect wires at pin header PH1, as follows in the diagram (right).

The cathodes (negative, indicated by black lines) of the LEDs are connected to the pads nearer the center of the board; the anodes (positive, indicated by red lines) are connected to the pads nearest the board edge. The sequence of the pads, is as three pairs from left to right, as Power (RED), Serial data (YELLOW) and 1pps (GREEN) – in other words, the same order as that of the SMD LEDs on the board.

You would also need to cut the thin exposed tinned copper traces on the underside of the PCB, in order to disconnect the onboard LEDs. This should be done with a sharp knife, carefully so as not to damage any other nearby traces.



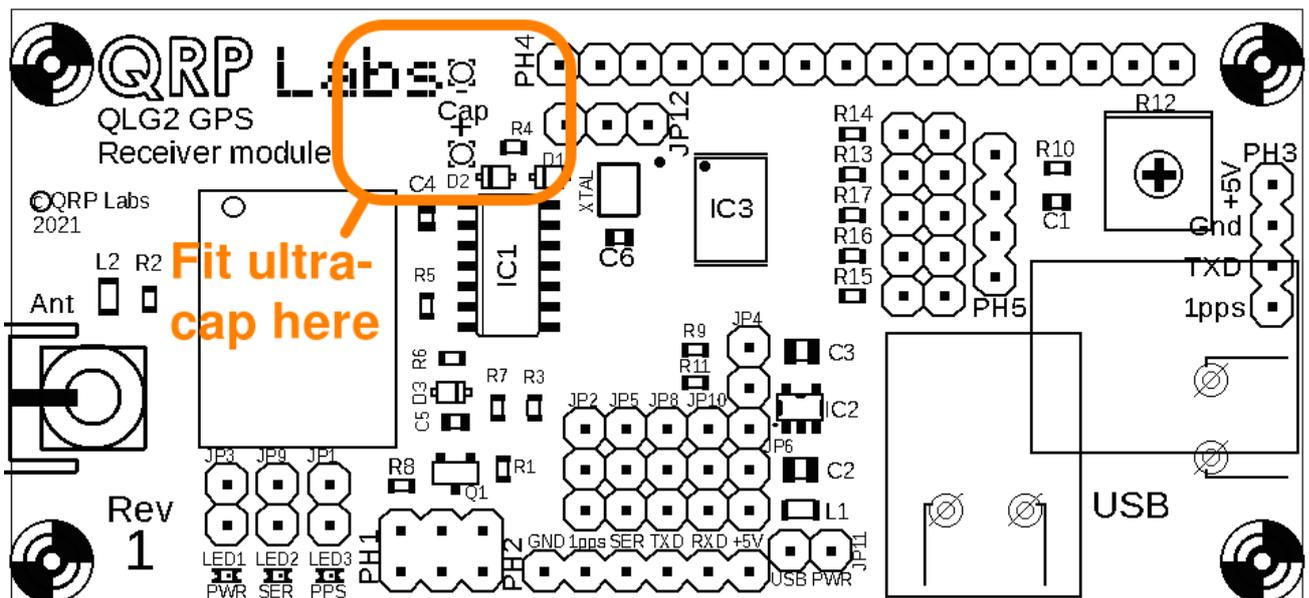
3.7 Fitting an ultracap (a.k.a. supercap)

There is a place on the QLG2 board to fit an ultracap (a.k.a. supercap). This is a lot more reliable means of providing backup power, than the little rechargeable battery on the QLG1.

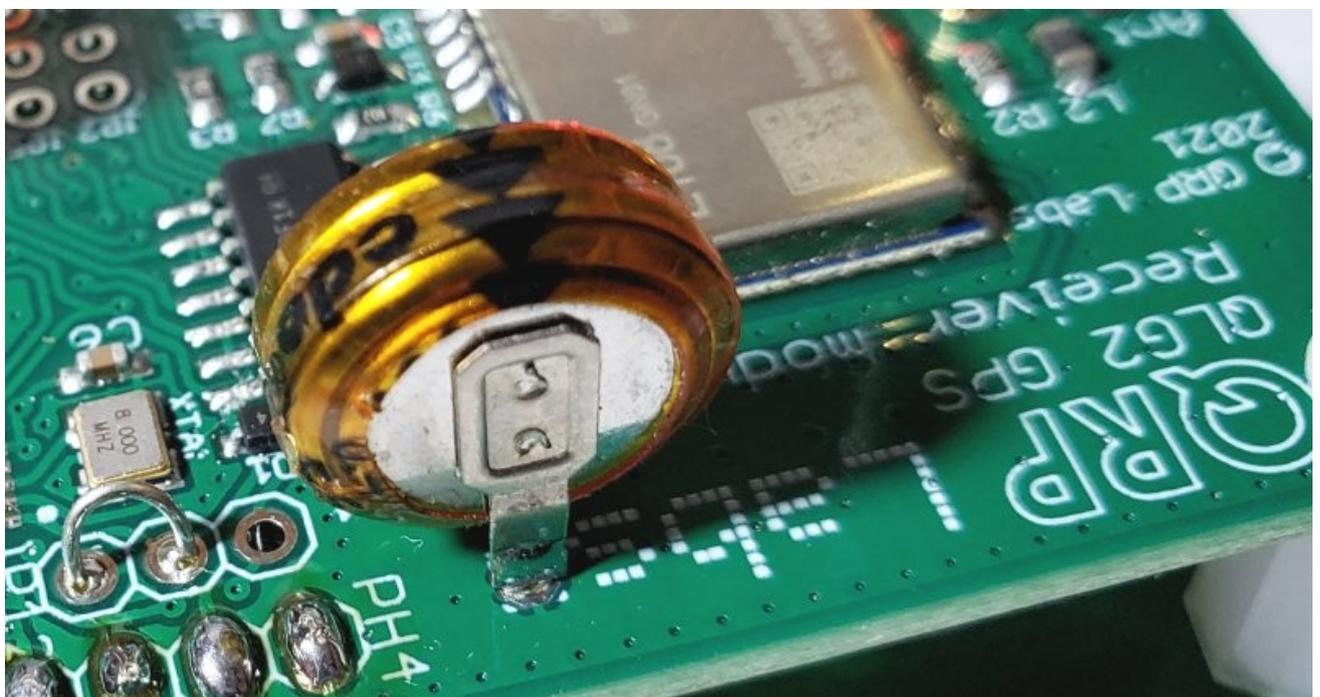
Backup power permits the operation of the internal Real Time Clock (RTC) in the GNSS module and also retains the downloaded satellite ephemeris data, which will enable a hot-start (satellite fix computation) within a second or two of power-up.

Most amateur radio applications will not need this, but the pads are made available on the board in case you do wish to use it.

Be careful to observe the correct polarity to avoid a possible bang and a nasty chemical mess (don't ask me how I know about this, either).

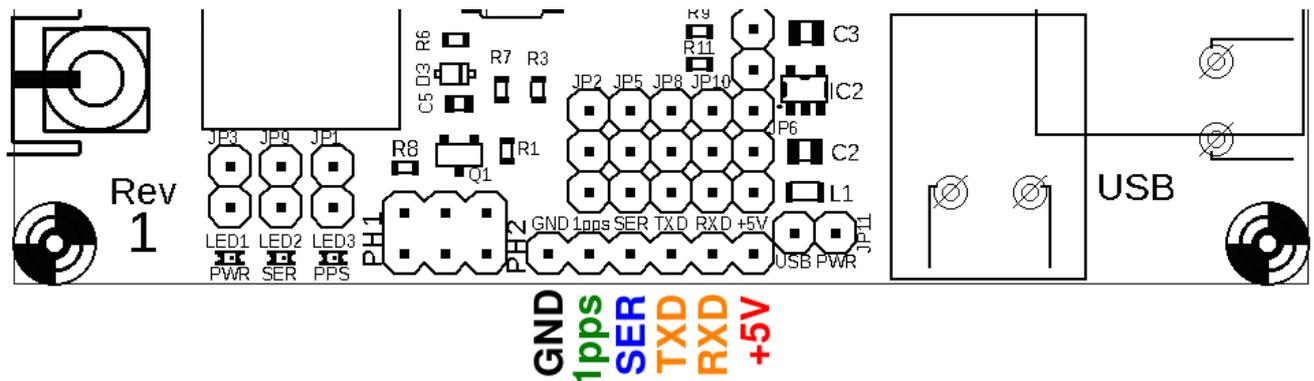


The photograph shows a 0.47 Farad, 5V capacitor installed on the QLG2 board. The pads are separated by 0.2-inches (5.08mm) and the capacitor fits perfectly. **This capacitor appears sufficient to backup the QLG2 GPS for at least 12 hours.**



3.8 6-pin interface header

At the center of the QLG2 board's bottom edge, is a 6-pin header that provides access to the power supply rails, the GNSS outputs, and the Serial port of the USB to Serial converter.

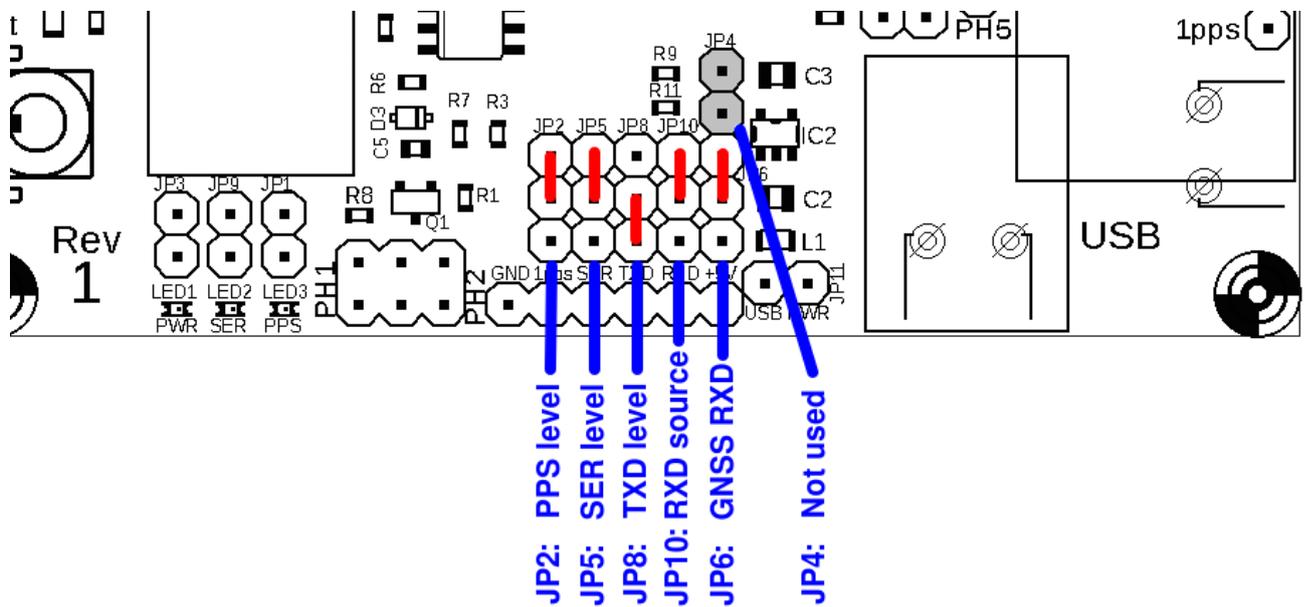


The 1pps, Serial data, TXD and RXD signals can be configured to be either 2.8V/3.3V logic level, or 5V logic level. The configuration is done using jumper wires that will be described in the next section. Note that 2.8V logic will be compatible with 3.3V systems, without issue.

From left to right:

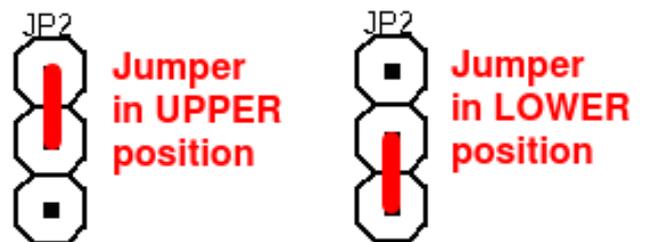
- Power supply ground.
- 1pps: 1 Pulse Per Second output from the GNSS module; by default this pulse is 0.1 seconds wide and it is sent once per second. By default, the voltage level is 5V which is suitable for use with QRP Labs products such as QCX, Ultimate3S, VFO, ProgRock and Clock kits.
- SER: The NMEA serial data output from the GNSS module. By default this is at 9600 baud and is 5V logic level, which is suitable for use with QRP Labs products such as QCX, Ultimate3S, VFO and Clock kits. Note that this signal is connected to the “TXD” signal of the 4-pin header on the right side of the board.
- TXD: Serial port transmit signal; anything sent from the USB Terminal Emulator (for example), is delivered to this transmit output pin. By default the baud rate is 9600 baud and the voltage level is 5V.
- RXD: Serial port receive signal; when the jumper wires are configured to connect this signal to the Serial to USB Converter, data arriving on this port is transferred to the USB host. By default this signal is NOT connected to the Serial to USB port, which is by default connected to the GNSS module. This RXD (receive) pin is tolerant of 2.8, 3.3 or 5V logic levels and automatically converts this to 3.3V for the microcontroller that implements the serial to USB converter.
- +5V: Power supply positive. It is nominally 5V, but in fact you could power the QLG2 module from any positive supply voltage in the range 3.3V to 6.0V, and the onboard voltage regulator will provide 3.3V to the GNSS module which is its correct supply voltage. **NEVER connect both USB power supply via JP11, and an external power supply.**

NOTE that the 5V logic levels referenced above, assume a 5V supply voltage to the QLG2; if you use a different supply voltage in the range 3.3V to 6.0V then the “high” logic level will be at that different supply voltage, not 5.0V.



In the following description, the nomenclature adopted is – as illustrated in the diagram (right):

UPPER: means the jumper wire is connected from the center of the group of three pads, to the top pad.



LOWER: means the jumper wire is connected from the center of the group of three pads, to the lower pad.

1. PPS voltage level:

- UPPER: 5V logic level (DEFAULT)
- LOWER: 2.8V logic level

2. SER voltage level (GNSS module serial data output, which is the TXD pin on the 4-pin header at the right of the board):

- UPPER: 5V logic level (DEFAULT)
- LOWER: 2.8V logic level

3. TXD voltage level (the USB to Serial converter transmit output pin):

- UPPER: 2.8V logic level
- LOWER: 5V logic level (DEFAULT)

4. RXD source (where the USB to Serial converter receive signal is connected to):

- UPPER: Receive signal is connected to the GNSS Serial data output (DEFAULT)
- LOWER: Receive signal is connected to the RXD pin for external use

5. GNSS RXD source (where the GNSS receive signal is connected to):

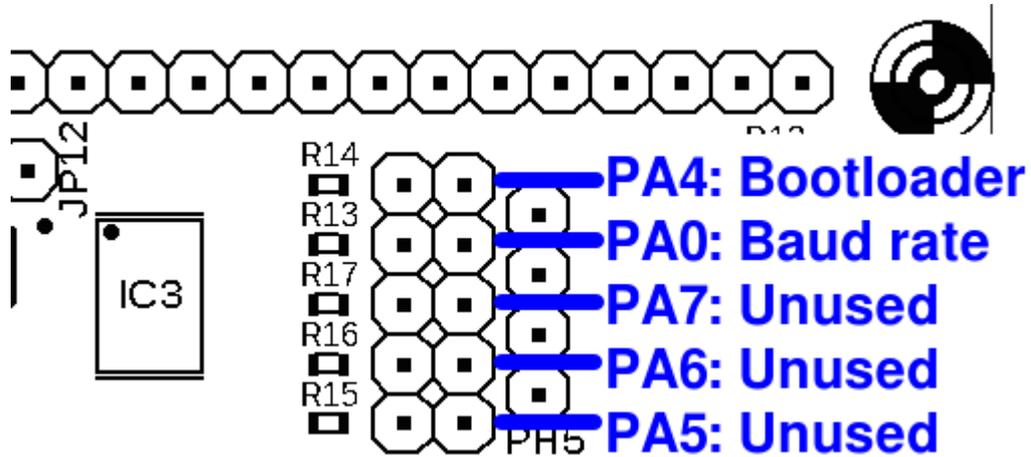
- UPPER: GNSS RXD is connected to the USB to Serial converter transmit signal (DEFAULT)
- LOWER: GNSS RXD is connected to the RXD pin for external use

6. JP4 is not used currently

NOTE: External RXD pin may be 2.8V, 3.3V or 5V logic and is automatically converted.

3.11 Microcontroller configuration

The 2 x 5 pads near the microcontroller chip, configure the operation of the microcontroller. These jumpers are “read” by the microcontroller at power-up in order to determine the configuration. Thereafter, they are ignored; therefore if a jumper plug is removed (for example) after power-up, nothing will happen until the next time the power is removed and applied again to the board.



By default, none of these pads have any connection (no tinned copper traces on the underside of the board). The nomenclature PA4, PA0 etc is a reference to the port pins of the microcontroller. PA7, PA6 and PA5 are currently unused.

The operation of these configuration jumpers is:

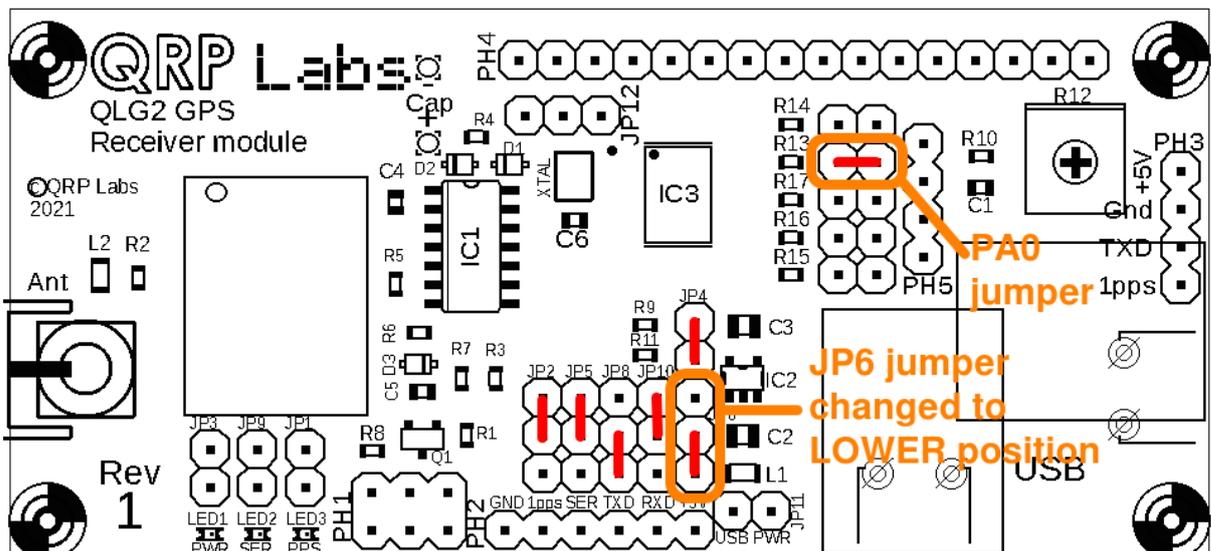
PA4: Bootloader

- JUMPER: Causes the microcontroller to enter the bootloader (see section below)
- OPEN: Causes the microcontroller to run the normal USB to Serial converters

PA0: Baud rate

- JUMPER: Baud rate is set to 38,400 (suitable for QCX CAT)
- OPEN: Baud rate is set to 9,600 (suitable for default GNSS module communication)

Example configuration for CAT control of a QCX-series CW transceiver:



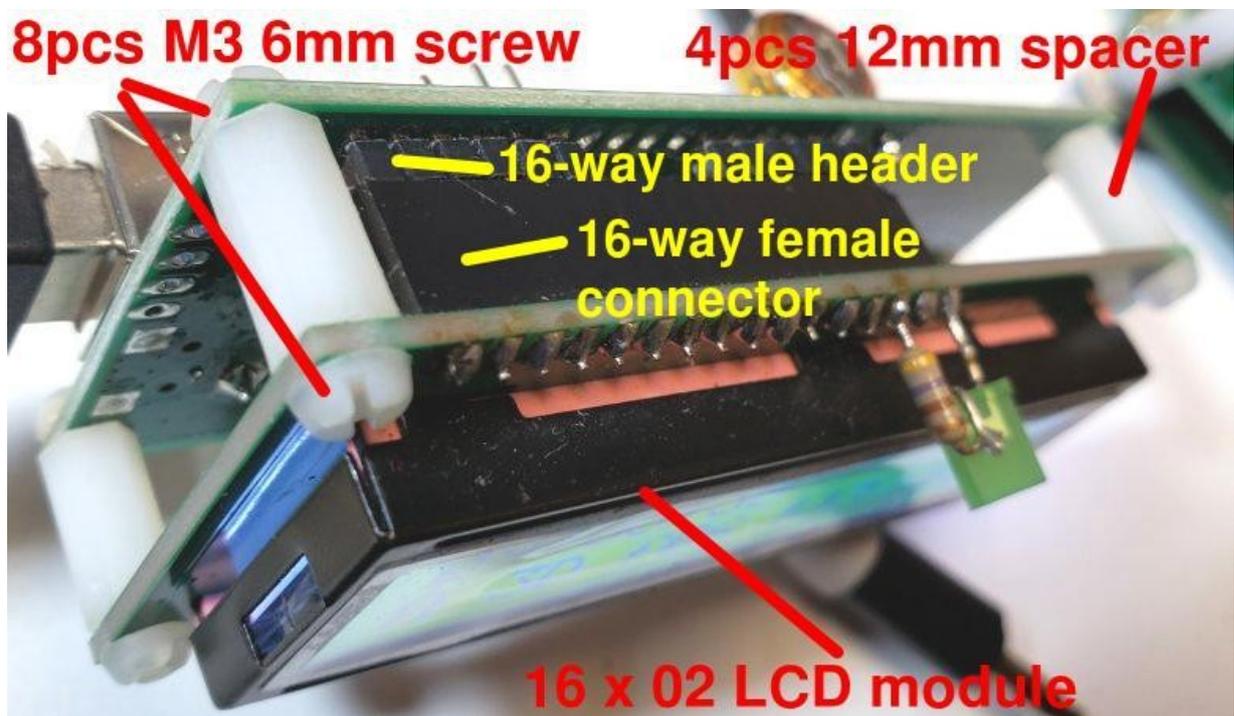
The diagram above shows an example interface and microcontroller configuration, to use the USB to Serial converter for connecting a host computer to a QCX-series transceiver for CAT control. The JP6 jumper must be cut from its default UPPER position, and install a connection (jumper) at the LOWER position. This selects the external RXD pin as input to the USB to Serial converter receive signal. Additionally a jumper must be installed at PA0 to configure the microcontroller USART peripheral for 38400 baud operation to suit QCX.

3.12 Connecting an LCD module

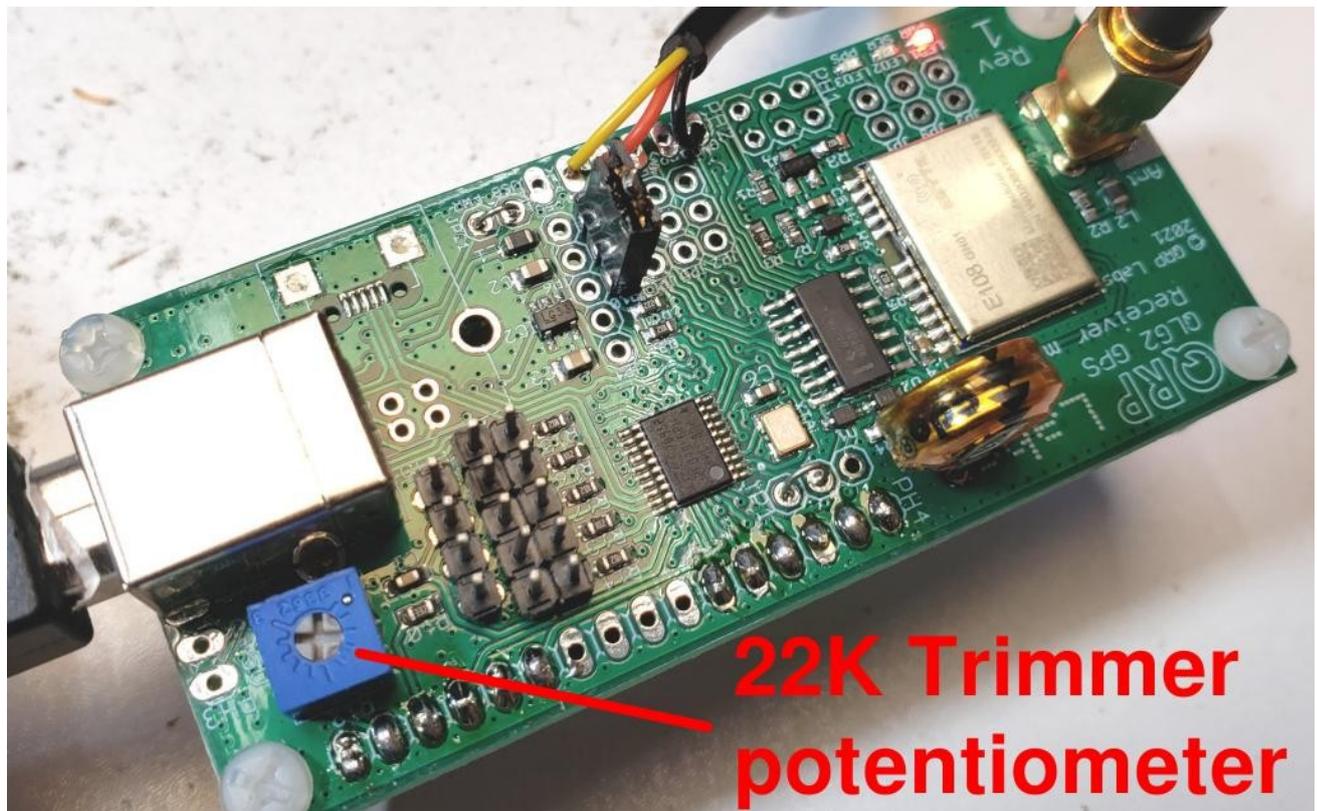
An LCD module may be connected to the QLG2 module directly. The standard 16 x 2 alphanumeric LCD module such as used in the Ultimate3S, VFO, Clock, QCX and QCX+ kits (blue colour) or the QCX-mini (yellow/green colour) can be used here; these types of LCD are available from many vendors. If it is HD44780 compatible (99% are) and has the row of 16 pads along the top edge, it will work. **The kit is not designed to drive display sizes other than 16 x 2.**

Installation of the LCD module is on the UNDER-side of the QLG2 module and is done in the same way as on the Ultimate3S, VFO and Clock kits and with the same components, namely:

- 16 x 2 LCD module, 80 x 36mm HD44780 compatible
- 16-way male pin header connector (soldered with the short side of the pins to the underside of the QLG2 board)
- 16-way female pin header connector (soldered to the underside of the LCD module as shown)
- 4pcs 12mm hex nylon M3-threaded spacer
- 8pcs 6mm M3 nylon screw



Additionally a 22K trimmer resistor is necessary and is installed in the provided position on the QLG2 board as shown below:



On power-up, the LCD briefly shows the firmware version of the program file that is held in the Flash memory of the microcontroller chip, for example 1_00.QG2 (the QG2 extension is for QLG2, but because it is a FAT16 file system file names are limited to ye olde 8.3 format).

Thereafter a display is shown such as the one below. It updates once per second and shows time (UT, GMT) on the top row, and GPS information on the bottom row.

NOTE: The green LED is just left over from a debugging development session.



The GPS information display contains five fields as follows:

1. Validity:
 - A: Valid fix
 - V: No valid fix is available yet
2. Nature of fix computation
 - 2D: 2-D fix (requires a minimum of 3 satellites)
 - 3D: 3-D fix (requires a minimum of 4 satellites)
3. f (Fix): the number of satellites used in the position fix computation
4. t (Tracking): the number of satellites being tracked
5. s (SNR): The average signal to noise ratio of the satellites being tracked – higher numbers indicate stronger signal strength

3.13 QLG2 Commands

QLG2 normally converts all incoming serial data from the USB host, converts it to standard Serial, and transmits this to the GNSS module (or via the TXD interface pin).

However, there are also several commands which you can type into the Terminal Emulator window on the host PC, as follows:

\$\$QFU immediately causes QLG2 to restart in the bootloader mode (firmware update)

\$\$VER returns the current firmware version file name, for example, 1_00.QG2

4. Firmware update procedure

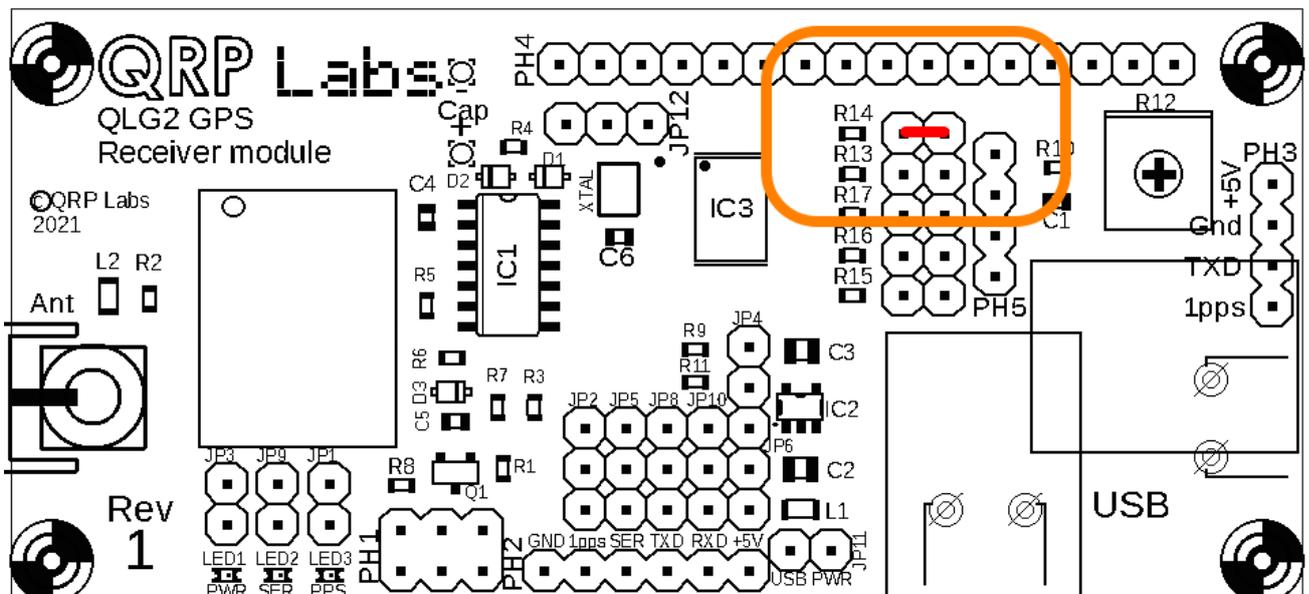
The QLG2 module contains a new firmware update procedure for STM32-series microcontrollers, called QFU (**Q**RP Labs **F**irmware **U**ppdate) which provides the following features:

- **Easy** – anyone can do the firmware update
- **No additional hardware required:** only a standard USB A-B cable (or micro-USB cable if you have installed a micro-USB connector)
- **No additional software required:** just the standard file manager application that is already available on any PC
- **No drivers:** no special drivers need to be installed, the existing drivers on any modern operating system are used
- **Works on any PC Operating System:** and in the same way: Windows, Linux, Mac
- **Secure:** firmware files are published on the QRP Labs website and are encrypted using 256-bit AES encryption technology

Entering bootloader (firmware update) mode:

In the QLG2 there are two possible ways to initiate a firmware update:

- 1) Issue the command **\$\$OFU** by typing these letters into the host PC Terminal Emulator connected to QLG2
- 2) Make a temporary jumper connection as shown:

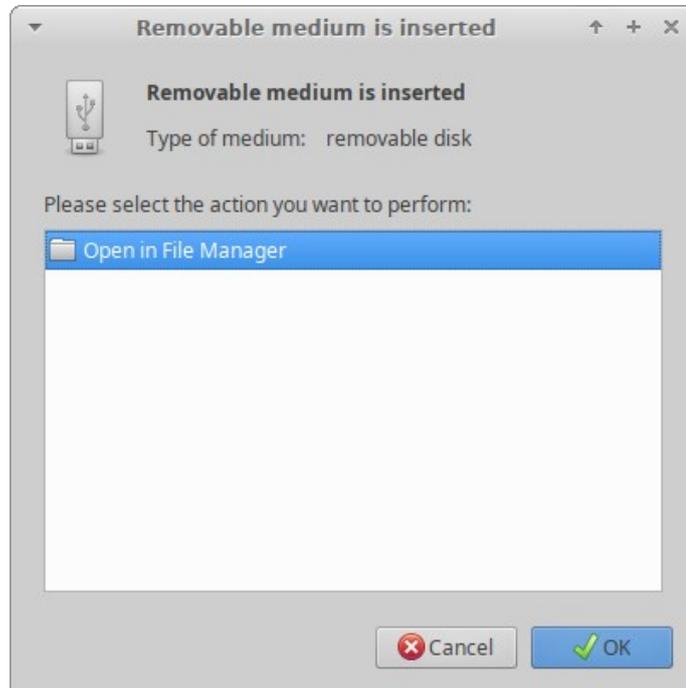


The connection can easily be made just with a small piece of bare tinned copper wire temporarily inserted in the jumper holes and not soldered; the connection only needs to be made at the moment of power-up of the QLG2, which is when the microcontroller configuration jumpers are read and actioned.

USB Flash memory stick emulation:

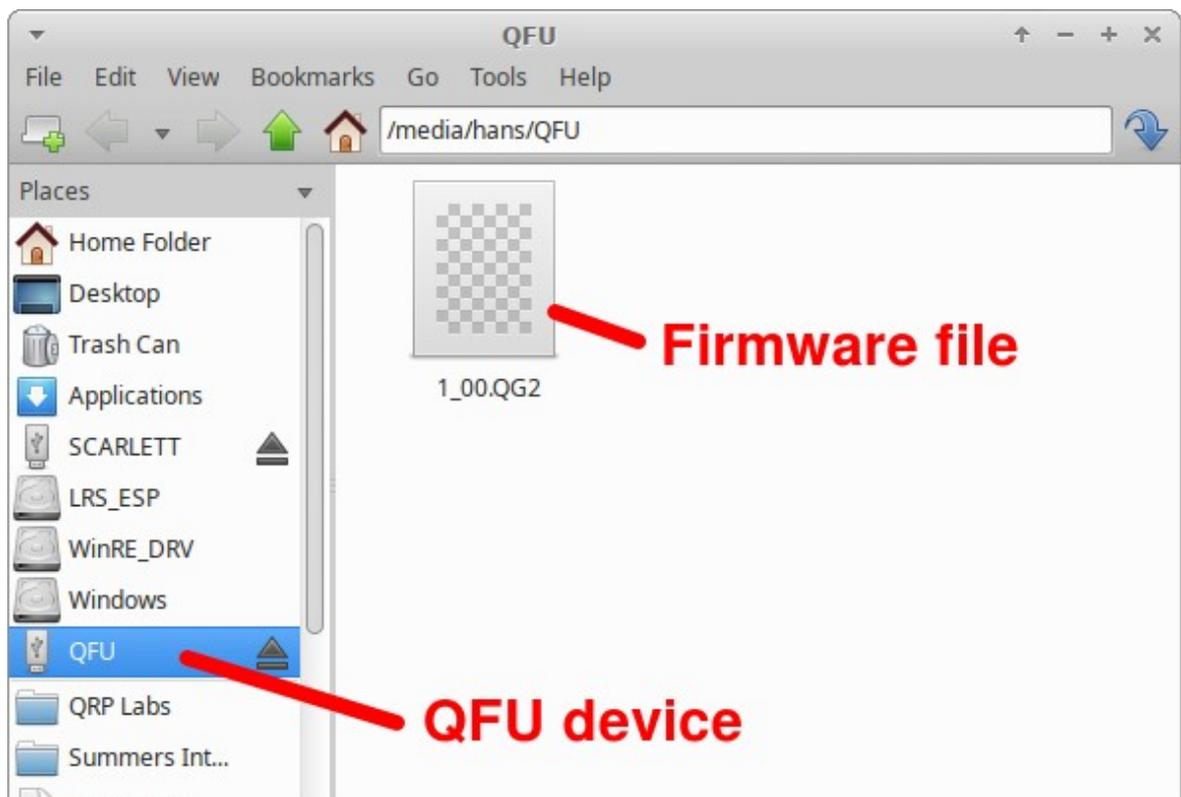
In the firmware update mode, the QLG2 pretends to be a USB Flash memory stick, having a 4MByte capacity and implementing a FAT16 file system. This virtual “Flash stick” contains a single file, the firmware program file in the QLG2 microcontroller. You may read the file from the QLG2, or write a new one, just by dragging files in your file manager application.

On entering the Firmware update procedure, a pop-up window should appear on your PC. On my system (Linux XUbuntu 18.04) it looks like this:



Click the OK button.

The File Manager window will then open, and on my system looks like this:



The QLG2 appears as a removable USB Flash device named “QFU”, and the folder shows a single file which is the firmware version file, 1_00.QG2 in this example.

The file name must not be longer than 8 characters, and cannot contain punctuation or spaces; the file extension must be no more than 3 characters (hence “QG2” not the preferred “QLG2”). This is because the file system emulation is FAT16 and these are the specifications of the FAT16 format.

You may check the properties of the file and will note that it is a 23.5K file. QLG2 firmware images are always a 23.5K file. The creation date and modification date etc. have not been set, because it was important to minimize the size and complexity of the QLG2 QFU bootloader, in order to maximize the space available to the application firmware.

You may copy the existing firmware file to another directory of your computer. Crucially, to do the firmware update, all you need to do is copy the new firmware file to this QFU “Flash disk”. Download the new firmware file from the QRP Labs website, and simply drag it into the folder where the existing firmware file version is shown. Or copy and paste it, however you wish.

As soon as you copy the new file to the QLG2 QFU “flash drive”, the QLG2 QFU bootloader erases the current program from its memory and installs the new one.

The QLG2 firmware is 256-bit AES encrypted and this means:

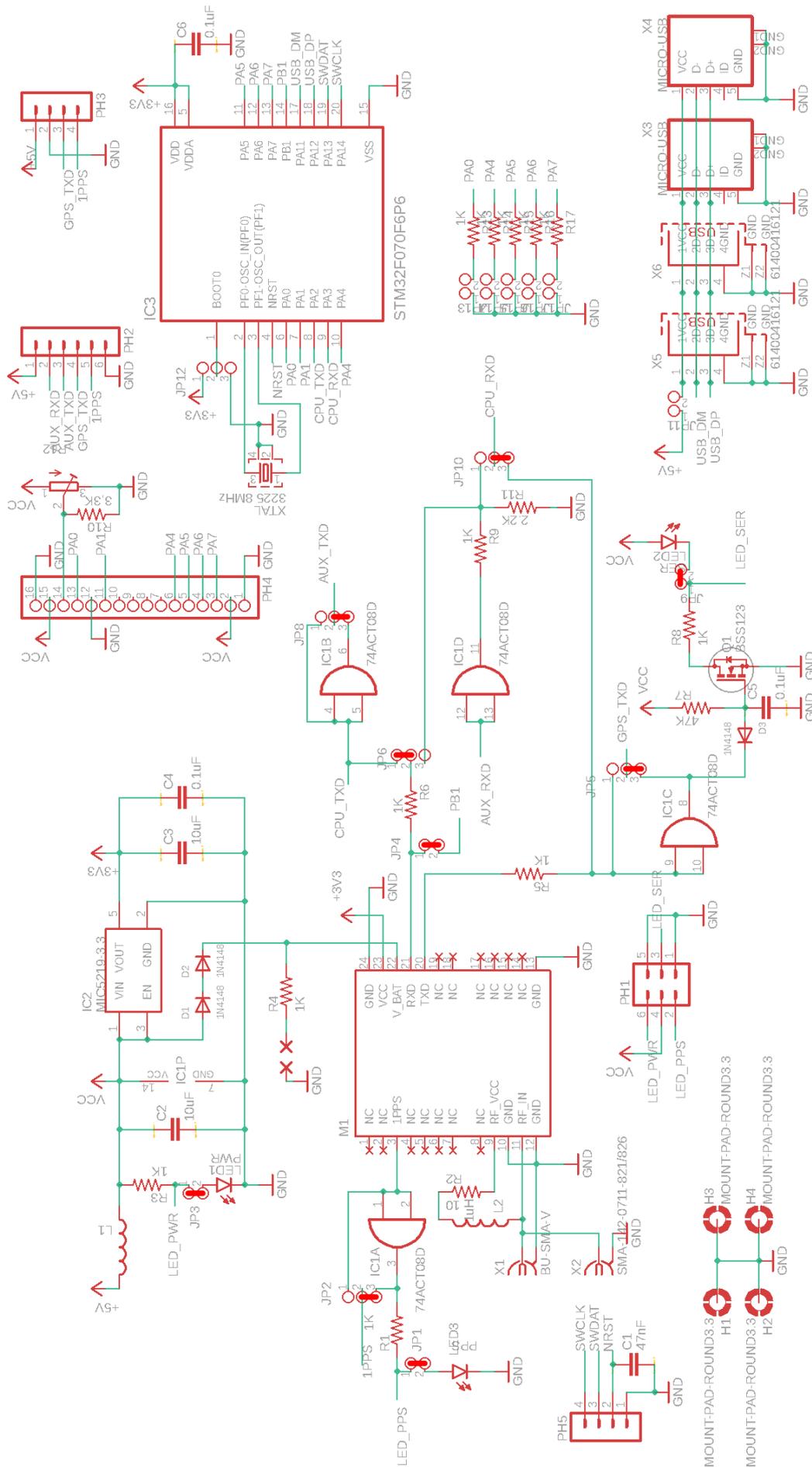
- The encrypted QLG2 firmware file will only work on a QRP Labs QLG2 board, it cannot be installed on any other board, even one containing the same processor.
- No other firmware file will work on the QRP Labs QLG2 board except an official QRP Labs encrypted QLG2 firmware file.

The procedure will vary slightly for different Operating systems but in all cases is just a simple matter of copying the new firmware file to the emulated QLG2 QFU USB Flash drive.

The above firmware update procedure works on ANY modern OS because the QFU bootloader emulates a USB Flash memory stick with the USB Mass Storage Device (MSD) class, for which drivers are already present.

The QFU bootloader implements a USB device stack (Mass Storage Device class), emulated FAT16 file system, Flash erase/write, and 256-AES encryption. It occupies the first 10K of Flash memory leaving 22K for the application itself.

5. Circuit diagram (schematic) and description



In the above circuit diagram, the position of the default jumper wires, which are implemented by tinned copper traces on the PCB underside, are indicated with a thick red line.

SMD Components parts list:

Part	Value	Type	Package
C1	47nF (0.047uF)	Capacitor	0603
C2, C3	10uF	Capacitor	0805
C4, C5, C6	0.1uF	Capacitor	0603
D1, D2, D3	1N4148	Diode	SOD-323
IC1	74ACT08	IC	SOIC-14
IC2	MIC5219-3.3	IC	SOT23-5
IC3	STM32F070F6P6	IC	TSSOP-20
L1, L2	1uH	Inductor	2012
LED1	Red	LED	0603
LED2	Yellow	LED	0603
LED3	Green	LED	0603
M1	E108-GN01	GNSS Module	16 x 12 mm
Q1	BSS123	Transistor	SOT23
R1, R3, R4, R5, R6, R8, R9, R13, R14, R15, R16, R17	1K	Resistor	0603
R2	10-ohms	Resistor	0603
R7	47K	Resistor	0603
R10	3.3K	Resistor	0603
R11	2.2K	Resistor	0603
XTAL	8MHz	Crystal	3225

The QLG2 circuit is based around the E108-GN01 GNSS module. This module is capable of reception of multiple satellite systems, not just GPS. This aids faster satellite acquisition and higher sensitivity in difficult reception locations. The E108-GN01 module uses the GK9501 GNSS chipset. Links to the datasheets for both of these are on the QRP Labs QLG2 page <http://qrp-labs.com/qlg2> .

An active antenna module is used, having a magnetic mount, antenna patch and Low Noise Amplifier in a weather-resistant enclosure. It is supplied with 2m of coaxial cable and an SMA connector.

Compared to the QLG1 (former QRP Labs GPS module), which was already a highly sensitive receiver, the QLG2 is even more sensitive.

The E108-GN01 GNSS module and the microcontroller require a 3.3V power supply; this is provided from the host PC over USB, or from the host device (Ultimate3S, QCX-series transceiver etc) which provides a 5V supply. The chosen voltage regulator is MIC5219-3.3.

The backup ultracapacitor (if installed) is charged via two 1N4148 diodes and a 1K resistor (R4). When power to the board is removed, the ultracapacitor powers the E108-GN01's V_BAT pin, providing retention of the downloaded satellite ephemeris data and Real-Time Clock. A 0.47F 5V capacitor is suitable in this application (see section above).

Level conversion from the 2.8V output of the E108-GN01 module, to the 5V logic levels required by other QRP Labs kits, is done using a 74ACT08 quad AND-gate chip; it is not used as an AND gate (all gates have their two inputs tied together), it is simply used as a logic level converter. The T in the part number 74ACT08 indicates that the device has TTL voltage threshold compatibility, and this means a "1" (logic high) is 2.4V so it is ideally suited for accurately converting GNSS module 2.8V logic to 5V levels.

Q1 implements a simple timer which causes the yellow LED to be extinguished whenever a '0' appears on the serial signal. When the signal returns to '1', the capacitor C5 must charge up via R7, until the turn-on threshold voltage of the Q1 MOSFET is reached, at which point the yellow LED is switched on again. This simple circuit means that the data transmission does not directly modulate the yellow Serial data LED, the LED is only lit a small delay after the end of the data burst. This avoids generating any power line noise due to toggling the LED at 9600 baud rate, which could be radiated into sensitive radio receivers nearby.

PH5 implements a 4-pin header for programming the microcontroller, however, once the microcontroller is programmed, this programming interface is disabled so the header has no further use, none at all, and is effectively only for use during manufacture.

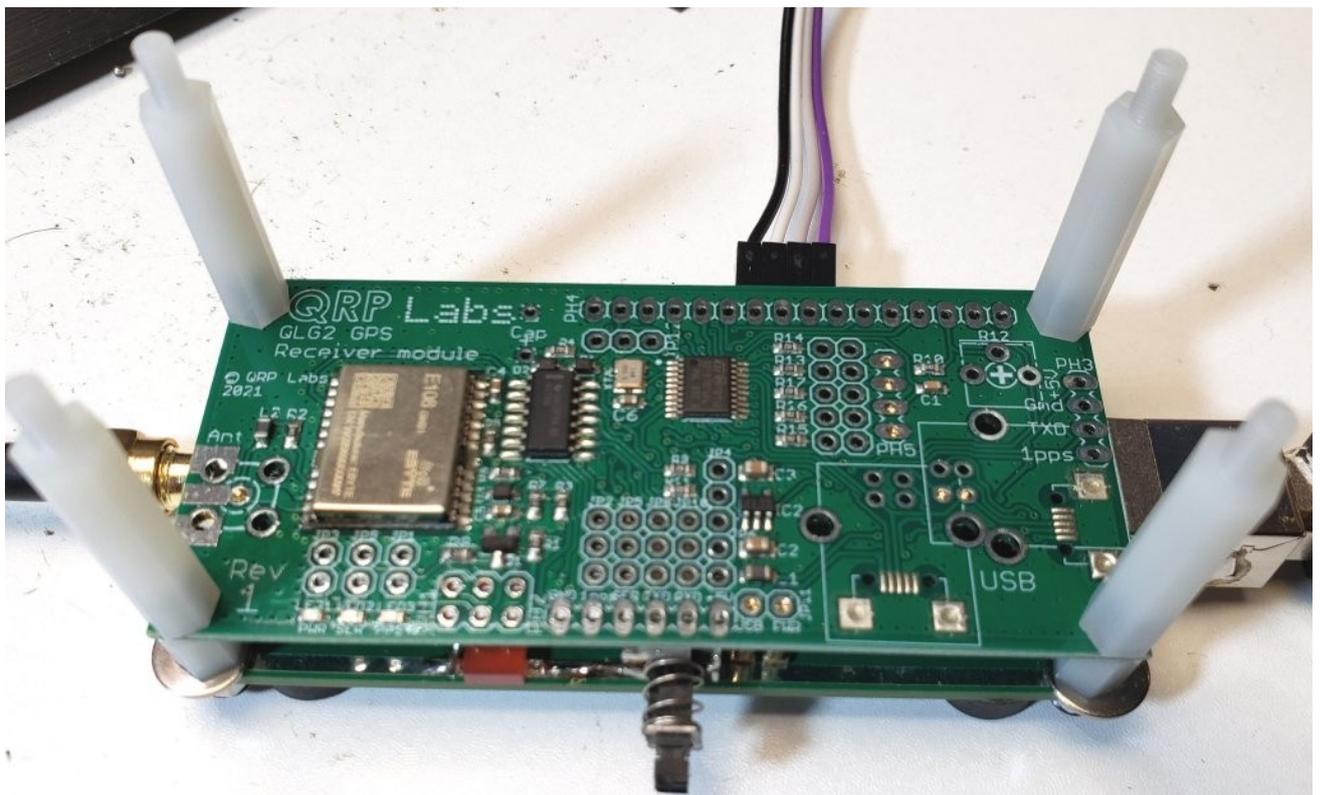
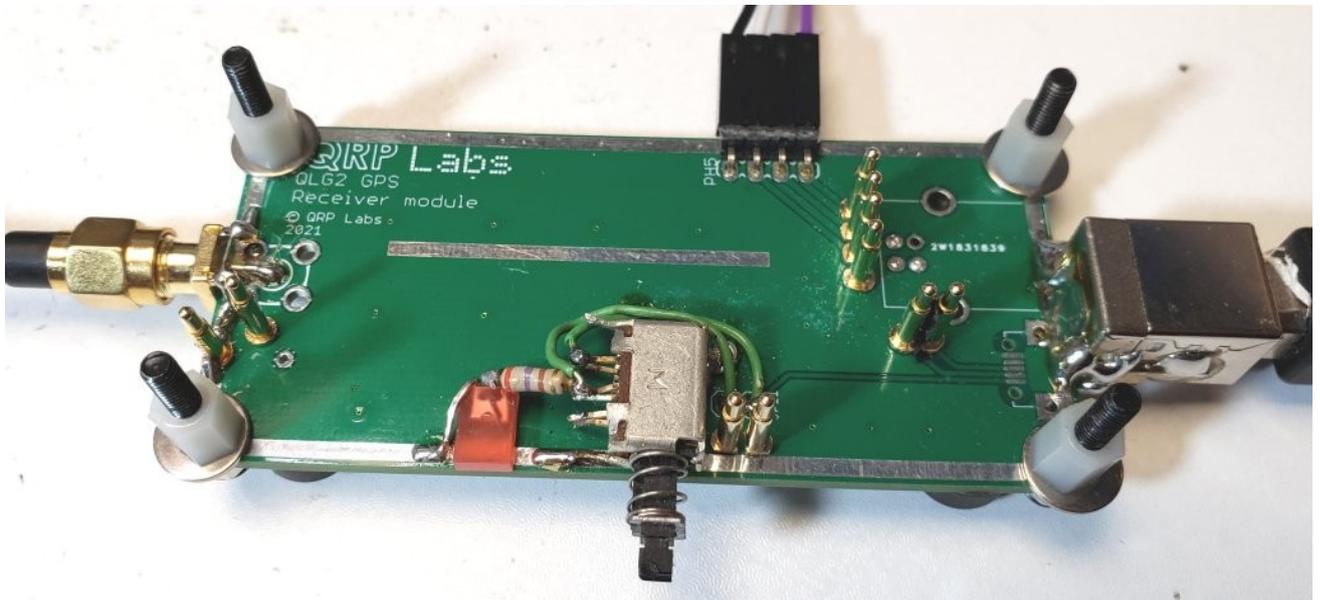
QLG2 uses a 32-bit STM32F070F6P6 ARM Cortex M0 microcontroller running at 48MHz derived from an 8MHz crystal. This processor has a USB Device peripheral, which is used in QLG2 to implement a Virtual COM Port for USB to Serial conversion, and during the bootloader mode, implement a USB Flash device (Mass Storage Device class). Both use standard USB device drivers that are already present on all modern operating systems.

The five microcontroller configuration jumpers are connected via 1K resistors to port pins PA4, PA0, PA5, PA6 and PA7. These pins are also connected to the LCD module (if installed). At power-up the pins are configured briefly as inputs so that the state of the jumper wires can be read; then re-configured as outputs in order to operate the LCD module.

6. Testing

All QLG2 modules are tested by QRP Labs in a custom-designed test jig. The test jig is used for firmware flashing, and connecting to the GPS antenna and USB connector. These connections are achieved by ten pogo pins installed on a special PCB. Four 3mm screws in each corner of the boards ensure accurate alignment, and the PCB under test is secured in place and under slight pressure (to press the pogo pins) using four 25mm nylon spacers. There is a push switch to connect power from the USB connector.

The following pictures show the test jig, and then the test jig with QLG2 board under test.



The test suite includes:

- Install QFU Bootloader program via ST flash tools (STM32CubeProgrammer).
- Use USB connection and Host PC to install the application firmware program (USB to Serial Converter and LCD display driver).
- Test E108-GN01 module using the lab GNSS antenna which is the same model as supplied in the QLG2 kit, and is lying outside on a sloping tile roof that slopes towards the East.
- Ensure time-to-first-fix (TTFF) is under 1 minute; on a sample of 20 units, the TTFF was found to range from 23 seconds to 52 seconds, with an average of 36 seconds.
- Check correct operation of Serial data and 1pps signals via status LEDs
- Check application firmware was correctly installed and microcontroller is running properly, by connecting a Terminal Emulator via USB and observing the NMEA serial data.

These tests give a very high confidence that all QLG2 modules shipped by QRP Labs will be in proper working order.

7. Acknowledgments

QRP Labs wishes to sincerely thank ST Microelectronics <https://www.st.com/> and in particular the CPU division for their assistance in the development and production of this project.



8. Resources

For updates relating to this kit please visit the QRP Labs QLG2 kit page <http://qrp-labs.com/qlg2>. This web page also contains links to the E108-GN01 datasheet and the GK9501 command reference

For any questions regarding the assembly and operation of this kit please join the QRP Labs group, see <https://groups.io/g/qrplabs> for details

9. Document Revision history

1.00	20-May-2020	First version for QLG2 firmware version 1_00.QG2
1.00a	21-May-2020	Minor edits to clarify a couple of points