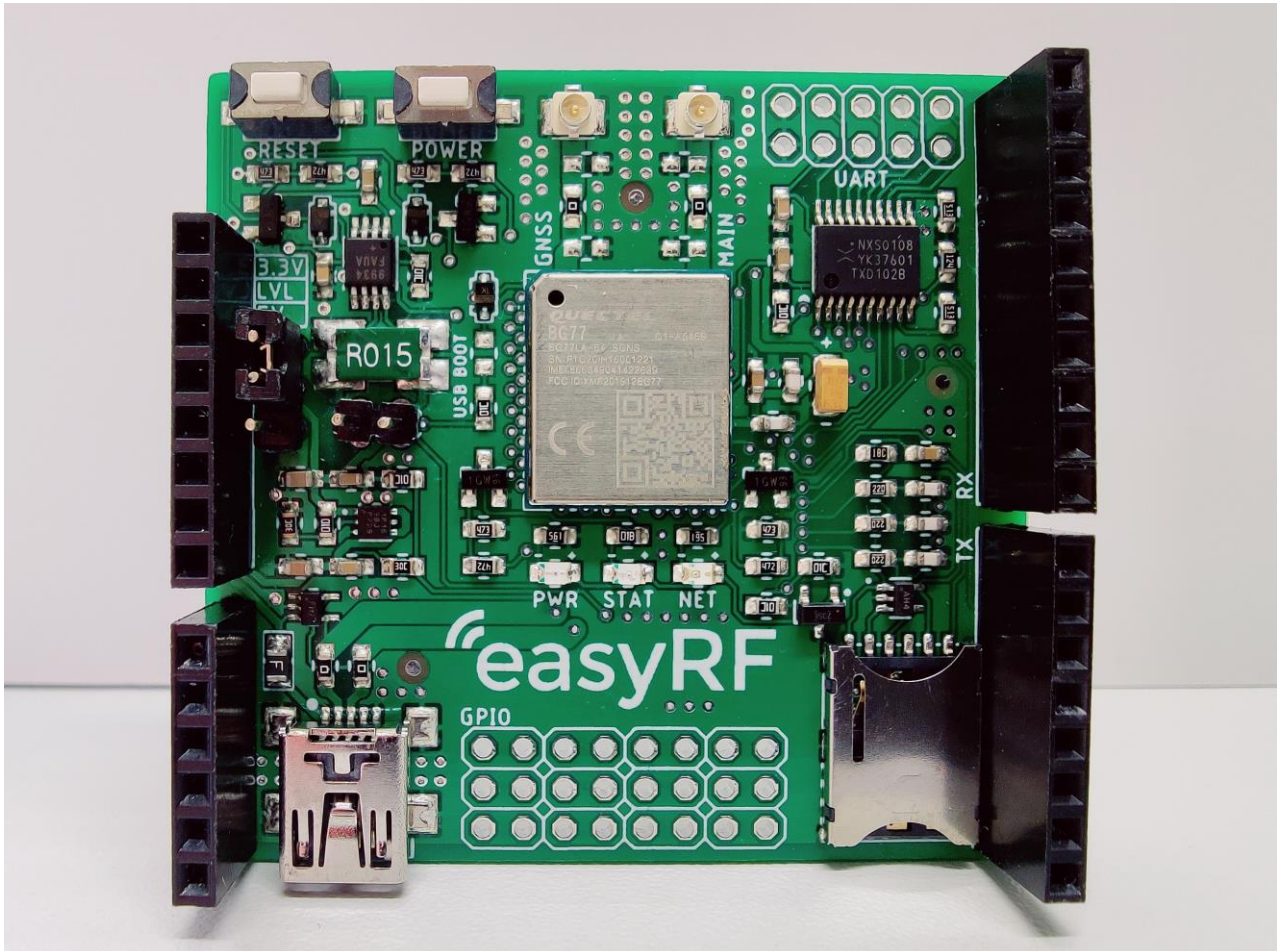




ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield





ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

Document Information

Title	ERF3002
Subtitle	LTE Cat. M1, Cat. NB2 & GNSS Arduino Shield
Document type	Preliminary User Guide
Document number	2
Revision and date	R1 29-07-2021

Product status

Mass Production

This document applies to the following products:

Name	Type number	version	Product status
ERF3002	ERF3002	1.0	Mass Production



Contents

Introduction	4
Set-up	4
1 Network connection	7
1.1 AT – commands	7
1.2 Examples	8
2 Sensors	9
2.1 Temperature sensor	9
2.2 Current measurement	10
3 GNSS	11
3.1 XTRA feature	11
3.2 XTRA example	12
4 Low-Power Application	13
4.1 PSM	13
4.2 eDRX	13
4.3 Antenna design	14
5 Firmware update	15
5.1 USB	15
5.2 DFOTA	17
6 Arduino Programming	18
6.1 Setup	18
6.2 Examples	19



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

Introduction

This document functions as a Quick start guide / User guide. Several basic processes and features are described in the document to help speed up the development and evaluation time. Several of these basic processes are: Getting a network connection, performing a firmware update, creating an Arduino software example and explaining how to create a Low power Internet Of Things (IoT) application.

Set-up

To start evaluating the shield, please follow below steps:

Step 1: Connect the GSM antenna.

Step 2: Connect the GNSS antenna.

Step 3: Plug in a Nano SIM card.

Step 4: Plug in a USB mini cable and connect it to a pc/laptop.

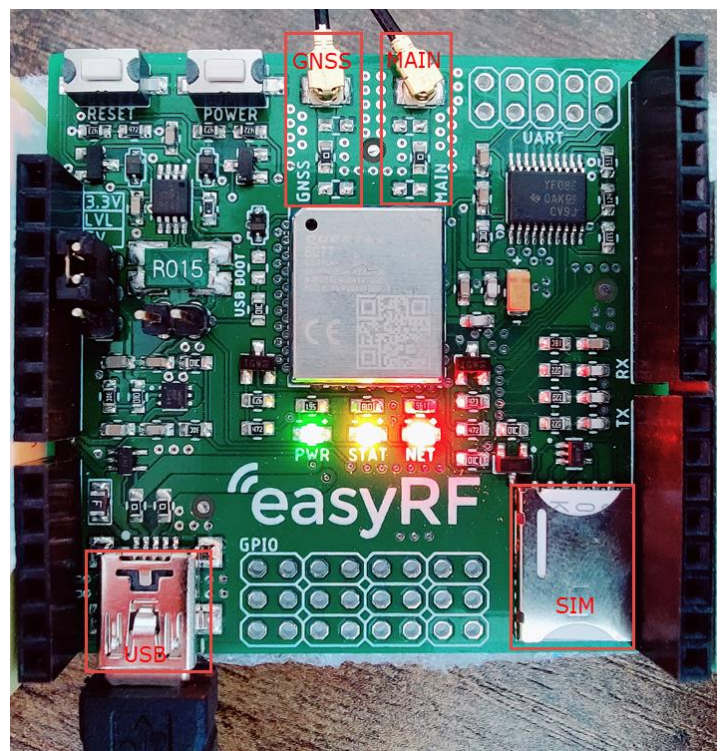


Please see chapter 3.1 Power of the [ERF3002 Datasheet](#) for the specifications.

Step 5: When power is supplied to the shield the green LED marked with PWR should light up.

Step 6: Press the POWER button to start the shield.

Step 7: After the module is done with start-up, the yellow LED marked with STAT should light up, and the red LED marked with NET will start blinking.



When above steps are followed, the shield is ready to receive AT-commands. Before a serial connection to the shield can be opened, the correct drivers need to be installed on your PC/laptop.

Please download them from the [ERF3002 Github page](#).

When the download is done, open and extract the .zip file.

Inside the ERF3002/Driver download either the Linux or Windows drivers and install this on your device.

When the installation is done, the PC will recognize the shield.



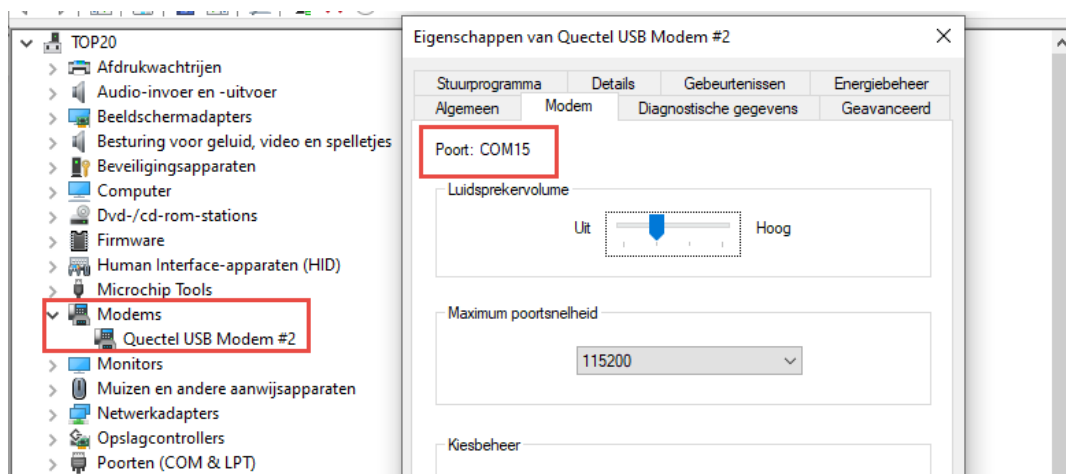
ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

Windows:

To verify if the drivers are installed go to, device management and look for the following 2 ports.

- ▼ Poorten (COM & LPT)
 - Quectel USB DM Port (COM16)
 - Quectel USB NMEA Port (COM17)

Due to a driver change by Qualcomm, the Quectel USB AT Port is found under Modems instead of Ports(COM & LPT):



After the drivers are installed, open a serial program like Putty, Realterm, etc.. These can be used to send AT-commands to the BG77. Quectel also developed a serial communication tool called QCOM. This program can be downloaded from the [ERF3002/Tools](#) folder.

In QCOM you have the option to write “scripts” that when started will automatically send a predefined AT-sequence to the BG77 module.

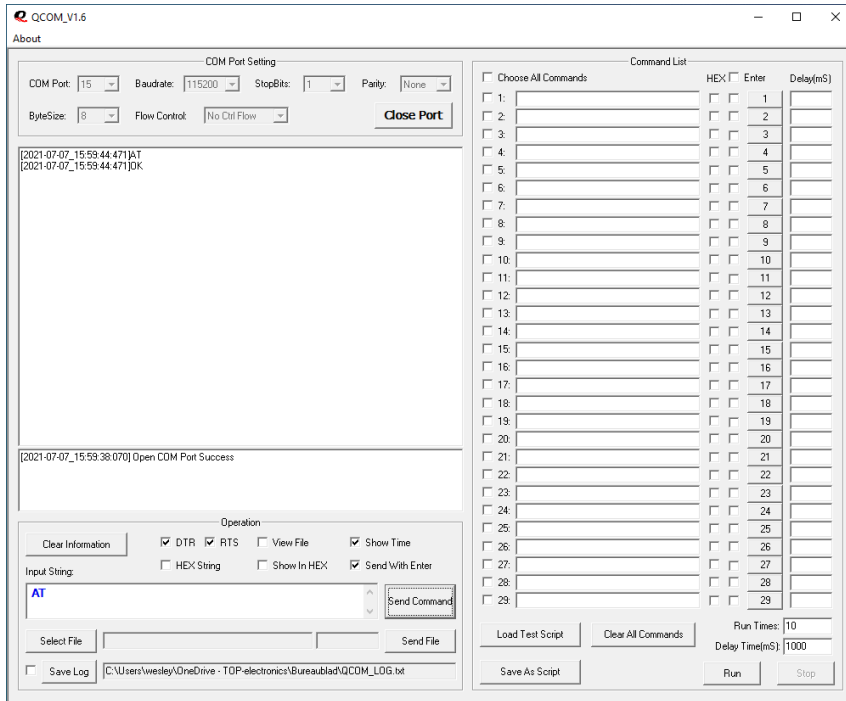
Open your serial communication tool and set the COM Port settings to:

- 115200 baud
- Stopbits: 1
- Parity: none
- Bytesize: 8
- Flow control: no flow control

Set the COM Port to the USB AT Port, the number of the port can be found in your device management. Now open the port and send AT, the module should respond with OK.



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield



Linux:

To install the drivers for Linux, please follow the [Quectel LTE&5G Linux USB Driver User Guide](#).



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

1 Network connection

1.1 AT – commands

In order to establish a connection, the module must be initialized with AT-commands. Please note, most providers use different settings for all below commands. See chapter [1.2 Examples](#) for example settings of the Dutch network providers.

When changing the network search settings of the module, start by turning off the radio:

Command	Description
AT+CFUN=0	Disable all phone functionality of the module

When the phone functionality is disabled set below settings according to your providers network.

Command	Description
AT+QCFG="band"	This command specifies the frequency bands allowed to be searched of UE
AT+QCFG="nwscanseq"	This command specifies the searching sequence of RATs
AT+QCFG="iotopmode"	This command specifies the network category to be searched under LTE RAT
AT+CGDCONT	This command specifies the PDP context parameters for a specific context

When the commands are set, the phone functionality can be activated again, and a connection to the network can be forced.



Before forcing a connection to a network, please check with your provider if IMEI registration of the Quectel hardware is necessary. If you don't register the IMEI number with your provider and still try to connect to their network you may be rejected, and access to the network will be blocked. The IMEI number can be found on the metal casing of the Quectel module, or can be read out using AT+GSN.

Command	Description
AT+CFUN=1,1	Will activate all phone functionality of the module, and reset the module
AT+COPS=...	This command forces an attempt to select and register the network operator

When AT+COPS returns OK. A connection has been established. Information about the connection can be gained with the following commands:

Command	Description
AT+QNWINFO	This command indicates network information such as the access technology selected, the operator, and the selected band.
AT+QSPN	This command will display the name of the registered network
AT+CSQ	This command indicates the received signal strength (RSSI) and the channel bit error rate
AT+QCSQ	This command is used to query and report the signal strength of the current network service. Dependent on the network this will return: RSSI, RSRP, SINR & RSRQ
AT+QENG="neighbourcell"	This command will return the neighboring cell towers information
AT+QENG="servingcell"	This command will return the serving cell tower information



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

1.2 Examples

Below are examples of network settings for all 3 major Dutch network providers:

KPN NL Cat. M
AT+CFUN=0
AT+QCFG="band",1,80000,1,1
AT+QCFG="nwscanseq",020202,1
AT+QCFG="iotopmode",2,1
AT+CGDCONT=1,"IP","APN"
AT+CFUN=1
AT+COPS=1,2,"20408",8

T-Mobile NL Cat. M	T-Mobile NL Cat. NB1
AT+CFUN=0	AT+CFUN=0
AT+QCFG="band",1,4,80,1	AT+QCFG="band",1,4,80,1
AT+QCFG="nwscanseq",030303,1	AT+QCFG="nwscanseq",030303,1
AT+QCFG="iotopmode",2,1	AT+QCFG="iotopmode",2,1
AT+CGDCONT=1,"IP","APN"	AT+CGDCONT=1,"IP","APN"
AT+CFUN=1	AT+CFUN=1
AT+COPS=1,2,"20416",8	AT+COPS=1,2,"20416"

VodafoneZiggo NL Cat. M	VodafoneZiggo NL Cat. NB1
AT+CFUN=0	AT+CFUN=0
AT+QCFG="band",1,80000,80000,1	AT+QCFG="band",1,80000,80000,1
AT+QCFG="nwscanseq",020202,1	AT+QCFG="nwscanseq",030303,1
AT+QCFG="iotopmode",2,1	AT+QCFG="iotopmode",2,1
AT+CGDCONT=1,"IP","APN"	AT+CGDCONT=1,"IP","APN"
AT+CFUN=1	AT+CFUN=1
AT+COPS=1,2,"20404",8	AT+COPS=1,2,"20404",9

For

more information on above AT-commands please take a look at the [BG77 AT Manuals](#).



2 Sensors

2.1 Temperature sensor

The TMP235 Temperature sensor is connected to ADC0 on the BG77, and can be read out using AT+QADC=0.

Because the TMP235 has a 3.3V max analog output signal, a resistor divider is used to lower the output voltage.

The resistor divider consists of 2x 10 KΩ +/- 0.1% resistors to effectively cut the analog output voltage in half and keep it from crossing the 1.8V boundary.

To convert the voltage read from the ADC0 pin on the BG77 module into temperature, please use the following table and formula:

$T_{A,RANGE}$ (°C)	V_{RANGE} (mv)	T_{INFL} (°C)	T_C (mV/°C)	V_{OFFS} (mV)
-40 to +100	< 1500	0	10	500
+100 to +125	1500 to 1752.5	100	10.1	1500
+125 to +150	>1752.5	125	10.6	1752.5

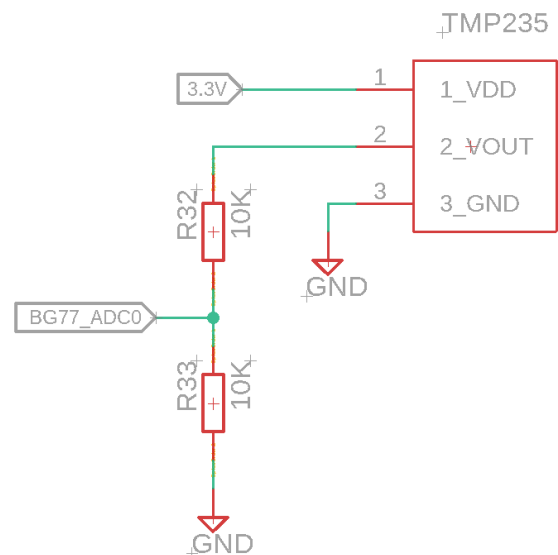
$$T_A = \frac{((2 * V_{OUT}) - V_{OFFS})}{(T_C + T_{INFL})}$$

Example:

AT+QADC=0
+QADC: 1,390

OK

$$T_A = \frac{((2 * 390) - 500)}{(10 + 0)} = 28^{\circ}\text{C}$$





ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

2.2 Current measurement

The MAX9934T current sense chip can be used to measure the current consumption of the BG77 module during the different modes. When this data is plotted vs. time you can get a good graphical image of the consumed current by the module.

To convert the voltage measured on the ADC pin into current, please use the following table and formula:

Parameter	Value	Unit	Description
V_{CC}	3.3	V	Voltage coming from LDO
R_{Sense}	0.015	Ω	Sense resistor
G_M	25	$\mu\text{A}/\text{mV}$	Gain of the current sense chip
R_{Out}	16.9	$\text{k}\Omega$	Output resistor Current sense chip
ADC_{res}	1024	bits	Arduino UNO ADC resolution is 10 bit ($2^{10} = 1024$)
ADC_{read}	-	bits	ADC value read from ADC pin A0 on the Arduino header
I_{Load}	-	A	Current consumption

$$I_{Load} = \frac{V_{CC} \cdot ADC_{read}}{(R_{Sense} \cdot R_{Out} \cdot G_M \cdot ADC_{res})}$$

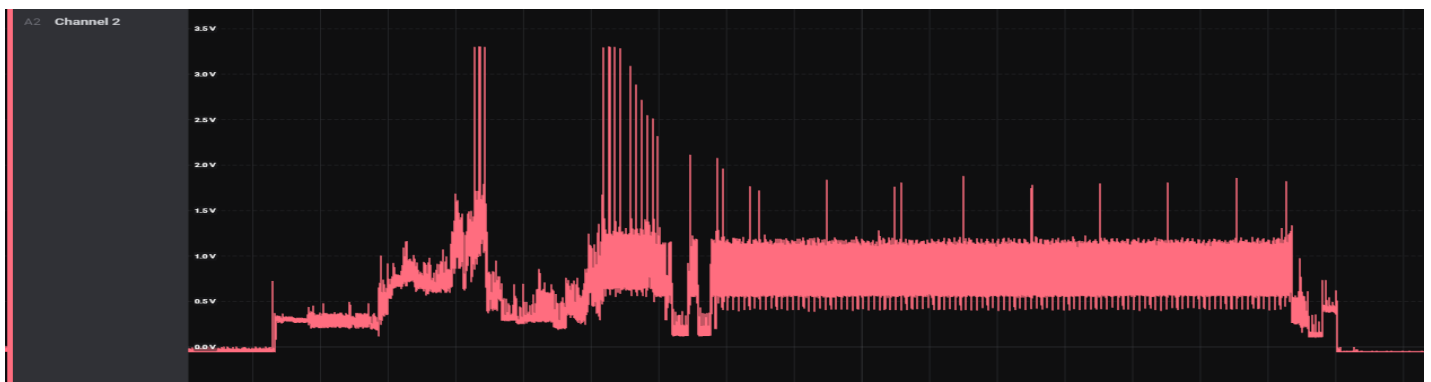
Examples:

$$I_{Load,min} = \frac{3.3 \cdot 1}{(0.015 \cdot 16.9 \cdot 25 \cdot 1024)} = 0.509 \text{ mA}$$

$$I_{Load,max} = \frac{3.3 \cdot 1024}{(0.015 \cdot 16.9 \cdot 25 \cdot 1024)} = 520.71 \text{ mA}$$

Example of plotted data:

*Captured on ERF3002 shield with Saleae Logic Analyser





ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

3 GNSS

The BG77 includes a fully integrated global navigation satellite system solution that supports Gen9 VT of Qualcomm (GPS, GLONASS, BeiDou, Galileo and QZSS).

The module supports the standard NMEA-0183 protocol, and outputs NMEA sentences at 1Hz data update rate via USB interface by default , or UART.

GNSS and cellular connection(WWAN) cannot work simultaneously, either WWAN or GNSS takes priority during implementation.

More information on the GNSS engine can be found in the [GNSS Application Note V1.2.pdf](#)

3.1 XTRA feature

The XTRA technology enhances the performance of GNSS and provides simplified GNSS assistance delivery, including ephemeris, almanac, ionosphere, UTC, health and coarse time assistance for GNSS engine. After activating XTRA, the TTFF (Time to First Fix) can be reduced to about 10 seconds in the open sky.

The XTRA file needs to be downloaded and loaded into the GNSS engine. There are multiple file types available choose whichever one suits your application best:

File Type & Duration (Validity)	7-Day File (kB)	3-Day File (kB)	1-Day File (kB)
GPS + QZSS	16.1	9.7	6.5
GPS + GLONASS	24.5	14.7	9.7
GPS + BeiDou	23.9	14.6	9.9
GPS + Galileo	25.8	15.2	9.9

NOTE: The sizes listed are illustrative only.



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

3.2 XTRA example

enable XTRA:

```
AT+QGPSXTRA=1  
OK
```

perform a reset of the module.

After the module restarts, get an active network connection:

```
AT+CEREG?  
+CEREG: 0,1  
OK
```

```
AT+QNWINFO  
+QNWINFO: "eMTC","20408","LTE BAND 20",6400  
OK
```

Check whether the XTRA file has expired:

```
AT+QGPPSCFG="xtra_info"  
+QGPPSCFG: "xtra_info",0,...  
OK
```

Enable the GNSS engine:

```
AT+QGPS=1  
OK
```

```
+QGPPSURC: "XTRA_DL",0 //XTRA file downloaded
```

If this URC does not show up, please use `AT+QGPPSCFG="xtra_download",1` to trigger the XTRA file download manually.

Check if the XTRA data is loaded:

```
AT+QGPPSXTRADATA?  
+QGPPSXTRADATA: 4320,"2021/07/12,11:00:00" //the XTRA file expires at 2021/07/12, 11:00:00  
OK
```

after XTRA is enabled and the GNSS engine is on. The time to fix is depending on the antenna used, the environment and the number of available satellites in view.

More information on the GNSS engine can be found in the [GNSS Application Note V1.2.pdf](#)



4 Low-Power Application

4.1 PSM

Power Saving Mode (PSM) is a key feature for eMTC/NB-IoT devices and applications with the following requirements:

- Infrequently active
- Short duration of active communication
- Data transmission is mainly originated by eMTC/NB-IoT devices, but also possible to engage in DL receiving during the active duration
- Power constraint, running on battery
- Require long battery life

PSM is to make an IoT device inactive or power-down most of the time to save power and wake up the device only for a short period of time for data transmission.

The Power saving mode (PSM) is the most important feature when creating a low-power application. This feature allows the module to enter a deep sleep mode for a specified time. When the specified time expires, the module will come back online. During the Power saving mode the current consumption of the BG77 drops to 3.2uA. The PSM can be configured with the following commands:

Command	Description
AT+QCFG="psm/enter",1	This command is used to trigger the module to enter into PSM mode immediately after the RRC released. When this function is enabled and RRC connection release is received, the module will skip active timer (T3324) and enter into PSM mode immediately
AT+QCFG="psm/urc",1	This command will allow the +QPSMTIMER URC message to be displayed, when the module receives the RRC release
AT+CPSMS=1,,,"TAU","active"	This command sets the TAU (T3312) and active (T3424) timers
AT+QPSMS=1,,,"TAU","active"	This command extends the previous set AT+CPSMS active and sleep timers

Please see the [BG77 AT-manuals](#) for more details on these AT-commands.

4.2 eDRX

The extended discontinuous reception (eDRX) is a feature to lower current consumption while connected to a network. Using the eDRX settings the interval between communication to the network can be manually set. The default value for M2M communication is 10.24 seconds. For IoT applications this time can be prolonged to 2,9 hours. In between the eDRX paging cycles, the module will turn off its radio and thus lowering the consumed current. Using the below commands the eDRX settings can be tuned.

Command	Description
AT+CEDRXS	This command will initialize the eDRX settings
AT+QPTWEDRXS	This command can be used so set advanced eDRX settings
AT+CEDRXRDP	This command will report the dynamic parameters of the extended eDRX settings

Please see [BG77 AT-manuals](#) for more details on these AT-commands.



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

4.3 Antenna design

The antenna design also has great influence on the total power consumption of the module. The consumed power is in direct correlation with the RSSI value. The RSSI value of the connection can be retrieved with the following command:

Command	Description
AT+CSQ	This command will return the signal strength (RSSI) of the connection
AT+QCSQ	This command returns the RSSI value as well as RSRP, SINR and RSRQ

The RSSI value is in its place in correlation with the Impedance and the return loss of the antenna (design).

In order to measure the return loss and impedance of an antenna (design), a Vector Network Analyzer (VNA) is needed. Using the VNA you can measure the impedance of your antenna design and get a representation of your antenna performance. To give an example of how important the antenna design is, see below test results:

The following tests are performed on the KPN LTE Cat M1 network. This network uses LTE B20 band with a center frequency of 800MHz.

Antenna type	Return Loss @800 MHz (dB)	RSSI (dBm)	Avg current (mA)	Time to connect (s)	Total (mA)	% increase in current consumption
Antenna 1	-14,12	-63dBm	117,309	4,423	518,8577	100%
Antenna 2	-5,02	-69dBm	119,84	5,883	705,0187	135,90%
Antenna 3	-3,51	-75dBm	128,294	6,43	824,9304	158,98%

As seen in the table the antenna design is a big part of creating an optimal low power application. In order to get an optimal antenna performance, measurements with a VNA are necessary.

If you need help tuning your antenna and/or antenna design for your low power application, please contact us through info@easyRF.eu and we will be able to help you.



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

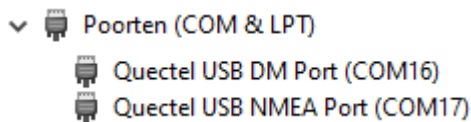
5 Firmware update

5.1 USB

For the update through the USB interface the tool Qflash is needed, this tool can be found in the [ERF3002/Tools](#) folder. In order to get a firmware package please contact TOP-electronics via: support@top-electronics.com.

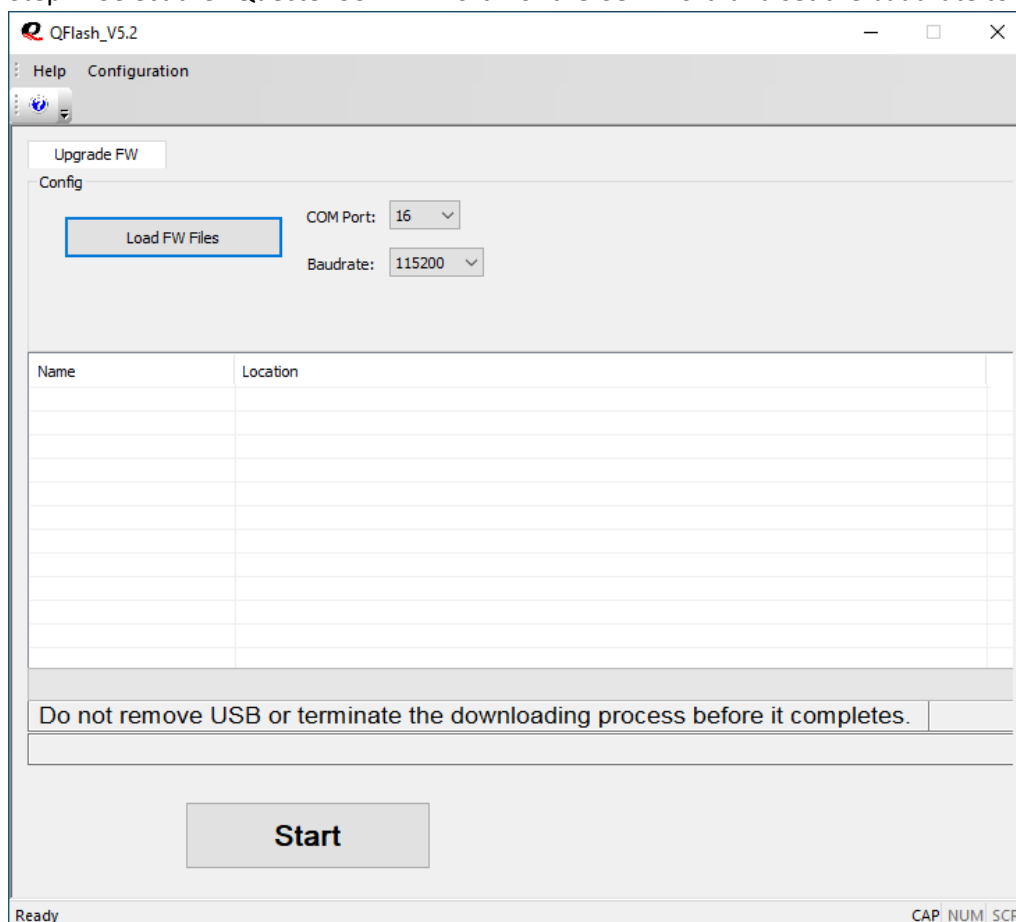
Step 1: Turn on the ERF3002 shield by connecting it to a Laptop/PC using the USB interface and pressing the PWR button.

Step 2: On your PC/Laptop look at your device management and find the “Quectel USB DM Port”.



Step 3: Open the Qflash software

Step 4: Select the “Quectel USB DM Port” for the COM Port: and set the baudrate to 115200.





ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

Step 5: Select the Firmware with the “Load FW Files”, This will open an explorer tab.

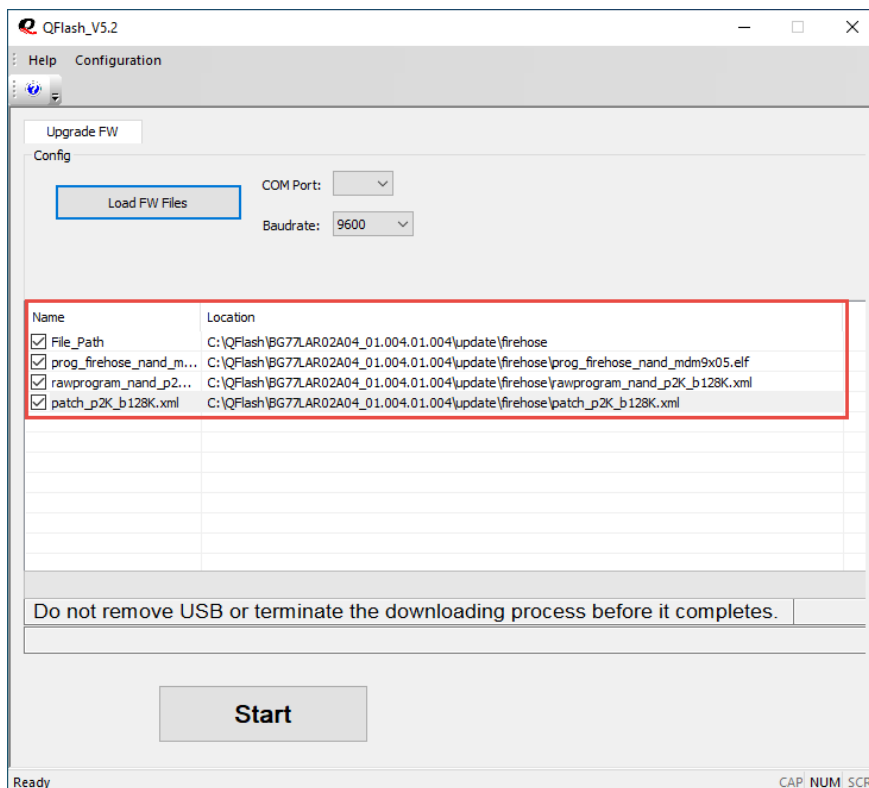
Navigate to the firmware folder and into the “update” folder.

BG77 > **BG77LAR02A04_01.004.01.004 > update > firehose**

Naam	Gewijzigd op	Type	Grootte
partition.mbn	17-3-2021 07:39	MBN-bestand	1 kB
partition_nand	17-3-2021 07:39	XML-document	8 kB
patch_p2K_b128K	17-3-2021 07:39	XML-document	1 kB
prog_firehose_nand_mdm9x05.elf	17-3-2021 07:39	ELF-bestand	439 kB
rawprogram_nand_p2K_b128K	17-3-2021 07:39	XML-document	9 kB

Select the “partitions.mbn” file and click on “Open”. The software will automatically select all relevant files for the update.

Step 7: Click on the start button to begin the update.



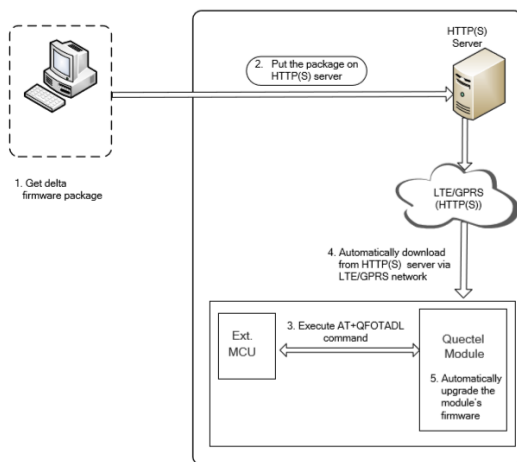


ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

5.2 DFOTA

In order to perform a DFOTA update, you will need to get a special DFOTA package. This can be obtained through contacting TOP-electronics via: support@top-electronics.com. Please state your current firmware and specify what firmware version you would like to upgrade/downgrade to.

When the DFOTA package has been received please follow these steps:



Step 1: Get the delta firmware package.

Step 2: Put the delta firmware on a HTTP(s) server.

Step 3: Power on module and make sure a network connection is established.

Step 4: Execute AT+QFOTADL=<http(s)URL> command. Then the module will automatically download the package from HTTP(S) server via LTE Cat. NB or Cat. M network.

Command	Description
AT+QFOTADL=<httpURL>	This command will start the DFOTA Download. (example AT+QFOTADL="https://www.quectel.com:100/update.zip")

Step 5: After the download is complete, start the update with AT+QFOTADL=1

Command	Description
AT+QFOTADL=1	This command will start the DFOTA process.

For more information please take a look at the [DFOTA User Guide.pdf](#)



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

6 Arduino Programming

6.1 Setup

To evaluate the shield with an Arduino board, the Arduino software is needed. This can be downloaded from the [Arduino website](#). After downloading the software, install it.

To communicate with the shield, a standard Arduino library called “SoftwareSerial” is needed. This is already installed with the Arduino IDE. Information about this library can be found [here](#).

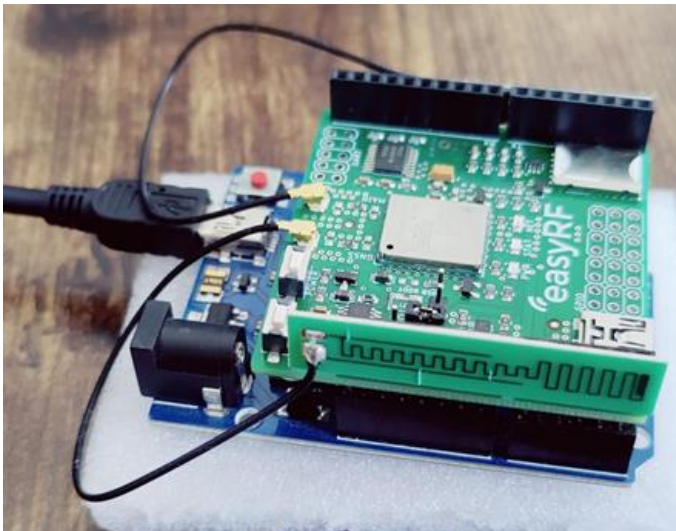
When the Arduino software is installed, open the software.

To speed up development and evaluation time, a few examples are provided on the [ERF3002 Github page](#).

In order to test the ERF3002 shield with an Arduino, place the ERF3002 on the Arduino headers, as can be seen in below image.

Now connect the following before inserting the USB cable in the Arduino:

- GSM antenna
- GNSS antenna
- Insert SIM card
- Verify the Arduino Voltage and set the LVL shift header accordingly to 3.3V or 5.0V
- Connect the USB cable to the Arduino



Verify the Arduino board is recognized by the PC/Laptop, and upload an example of your choosing to the Arduino board. When opening the COM port in the Arduino software, make sure to put it on 9600 baud.



ERF3002, LTE Cat. M1, Cat. NB2 & GNSS Arduino shield

6.2 Examples

The Examples provided are:

Name	Special function	Description
ERF3002_CAT-M_Connect	CAT-M	Connect to CAT-M network
ERF3002_Temp_sensor	Temp Sensor	Read out Temp sensor connected to the BG77
ERF3002_GNSS.ino	GNSS	Start the GNSS service and get a fixed location with 1Hz interval. Please do note the GNSS doesn't work indoors, and it could take some time until a location fix is acquired.