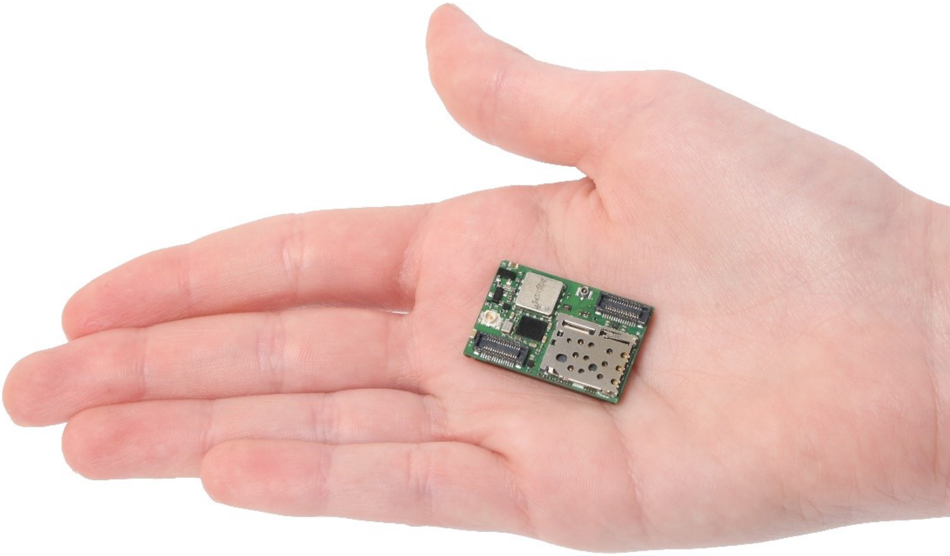


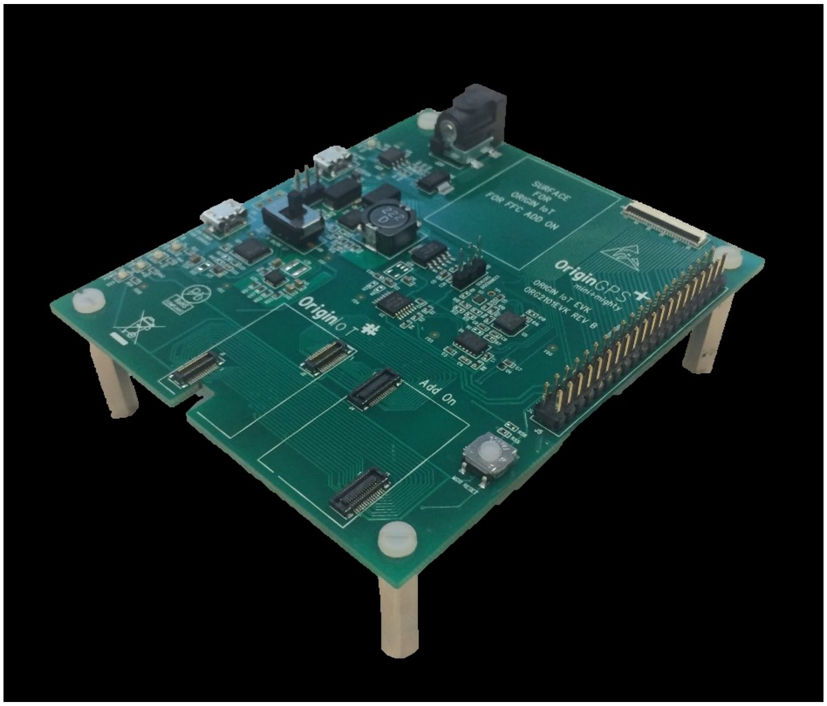
OriginIoT ORG2101 - EVK

USER GUIDE

OriginGPS.com

INDEX

FIGURE INDEX


SURFACE FOR ORIGIN IoT FOR FFC ADD ON
OriginGPS mini+mighty
ORIGIN IoT EVK ORG2101EVK REV B
OriginIoT
Add On

1. SCOPE

This document is a user guide for miniature cellular IoT system evaluation kit (OriginIoT-EVK).

2. DISCLAIMER

All trademarks are properties of their respective owners.

Performance characteristics listed in this document do not constitute a warranty or guarantee of product performance. OriginGPS assumes no liability or responsibility for any claims or damages arising out of the use of this document, or from the use of integrated circuits based on this document.

OriginGPS assumes no liability or responsibility for unintentional inaccuracies or omissions in this document.

OriginGPS reserves the right to make changes in its products, specifications and other information at any time without notice.

OriginGPS reserves the right to conduct, from time to time, and at its sole discretion, firmware upgrades.

As long as those FW improvements have no material change on end customers, PCN may not be issued.

OriginGPS navigation products are not recommended to use in life saving or life sustaining applications.

3. SAFETY INFORMATION

Improper handling and use can cause permanent damage to the product.

4. ESD SENSITIVITY

This product is ESD sensitive device and must be handled with care.

5. CONTACT INFORMATION

Support - iot@origingps.com or Online Form

Marketing and sales - marketing@origingps.com

Web – www.origingps.com

www.origin-iot.com

6. RELATED DOCUMENTATION

№	DOCUMENT NAME
1	ORG2101 Datasheet
2	OriginSmart™ API specifications
3	OriginIoT debug console user guide
4	Spider and Hornet - NMEA Protocol Reference Manual
5	Spider and Hornet - One Socket Protocol Reference Manual
6	Spider and Hornet – OSP GNSS Extension Reference Manual

TABLE 1 – RELATED DOCUMENTATION

7. REVISION HISTORY

REVISION	DATE	CHANGE DESCRIPTION
1.0	March 26, 2018	First Edition – Preliminary
1.1	May 13, 2018	New web application guidance. Mating cycles of connectors. Change in P/n. Minor corrections and clarifications

TABLE 2 – REVISION HISTORY

8. ABOUT ORIGINIOT™

8.1. MODULE DESCRIPTION

The OriginIoT module ORG2101 is a miniature generic cellular IoT system with GNSS tracker and interfaces to peripheral sensors and devices. The data is transferred via cellular communication, stored in a remote server (cloud), and displayed on a web GUI.

The module is communicating through GSM or LTE interface with a cloud-based application.

The multi-purpose IoT sensor platform can accommodate sensors and other peripheral devices via UART, SPI, I2C, USB, ADC or GPIO, with superior positional accuracy of stand-alone GNSS. Data is configured over a web interface, eliminating additional embedded FW efforts. The ease and flexibility of adding the mini+mighty OriginIoT to a vast array of applications quickens time to market while minimizing the size of your IoT sensor device.

8.2. ARCHITECTURE

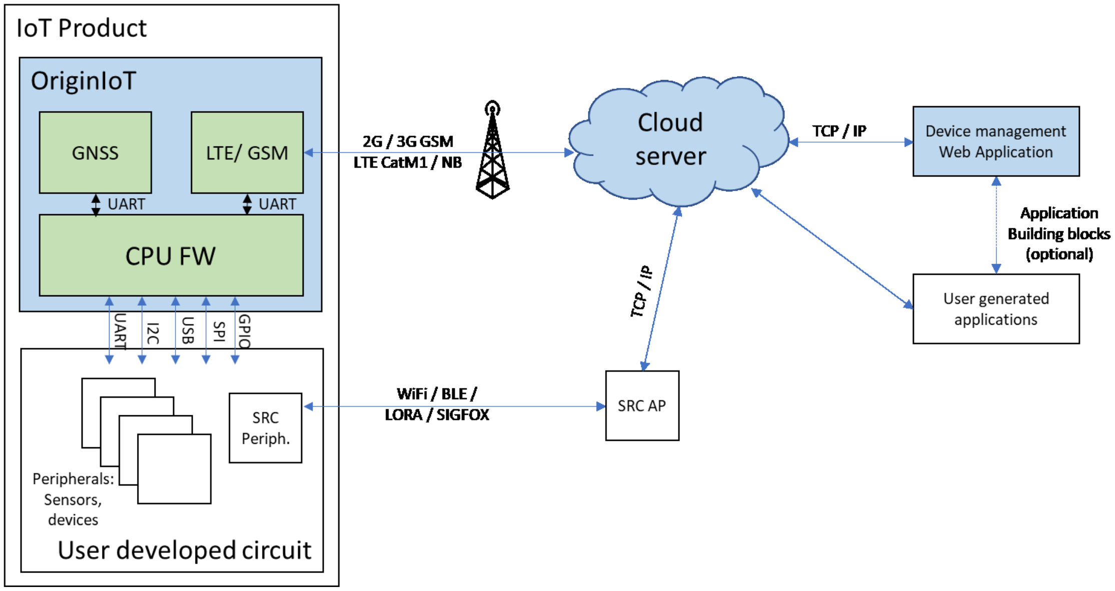


Figure 1 –OriginIoT Architecture

8.3. ORIGINIOT EVK KIT INCLUDING

- OriginIoT ORG2101 module and EVK
- GNSS antenna 12x12x4.3 mm with W.FL connector
- Penta-band GSM antenna with SMA connector (designed to support of all frequency bands, may be varied upon customer application)
- SMA to U.FL connector cable
- Power adapter
- Micro USB cable
- OriginIoT add-on with EVK sensors, DC/DC and charging circuit (optional)

8.4. COMPONENTS ON THE EVK

- Sensors:
  - Accelerometer- ADXL345, U1, by Analog Devices.
    Datasheet: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf
  - Gyroscope- DM00168691, U5, by STMicro Electronics.
    Datasheet:http://www.st.com/content/ccc/resource/technical/document/datasheet/e4/b1/d1/62/1a/e6/44/2f/DM00168691.pdf/files/DM00168691.pdf/jcr:content/translations/en.DM00168691.pdf
  - Electronic Compass- IST8304, U3, by iSentek.
    Datasheet: http://www.isentek.com/cn/dlf.php?file=../ISENTEK/%28201703-13%29IST8304%20Datasheet%20v2.3_brief.pdf
  - Digital Temperature- TMP104, U4, by Texas Instruments. Datasheet: http://www.ti.com/lit/ds/symlink/tmp104.pdf
- FTDI to USB serial chip: FT232RQ, U21.
- Voltage Translator 4bit bi-directional: LSF0204PWR, U2.
- Common mode filter for USB/HDMI: ACM2010-900-2p-T002, U23.
- Buck-Boost Switching Regulator IC Positive Adjustable 1.8V 1 Output 3A: LTC3113IDHD#PBF, U24
- Power Switch ICs- Power Distribution Single 1A Current-Limited, TPS2051B, U25
- 1A Low Noise Fast Transient Response Low Dropout Regulator, TPS3801DCQR

9. GETTING STARTED

9.1. REQUIRED EQUIPMENT

Items needed:

- OriginIoT EVK
- Nano SIM card with data package
- Server with a fixed IP.

OriginIoT module allows the user to communicate with GNSS receiver and sensors through web GUI on a cloud. The communication between OriginIoT and the cloud server is done through cellular network via data sim card. OriginGPS provides a cloud server for evaluation: http://18.184.85.107*
* Server IP can be changed in the future. Please make sure you downloaded the latest publication of this document from www.origingps.com

- PC with a terminal emulator (such as free – open source Tera-Term)
  Tera Term can be downloaded here:
  http://download.cnet.com/Tera-Term/3000-20432_4-75766675.html
- Micro USB cable (included in the EVK kit)

9.2. CONNECTING THE EVK TO PC VIA DEBUG PORT

In order to configure the OriginIoT to connect to GSM/LTE Network and web server, it is needed to configure the module using DEBUG port connection to PC (J22).
Please follow these steps:

1. Insert nano SIM card to its slot. See figure 1.

Note: skip this step if using ORG2101-XXXX-E (embedded SIM)

2. Connect GSM/LTE antenna to the U.Fl connector OriginIoT, See figure 1.

3. Connect GNSS antenna to the W.Fl connector on OriginIoT, See figure 1. (Required only if user wants to evaluate GNSS functionality)

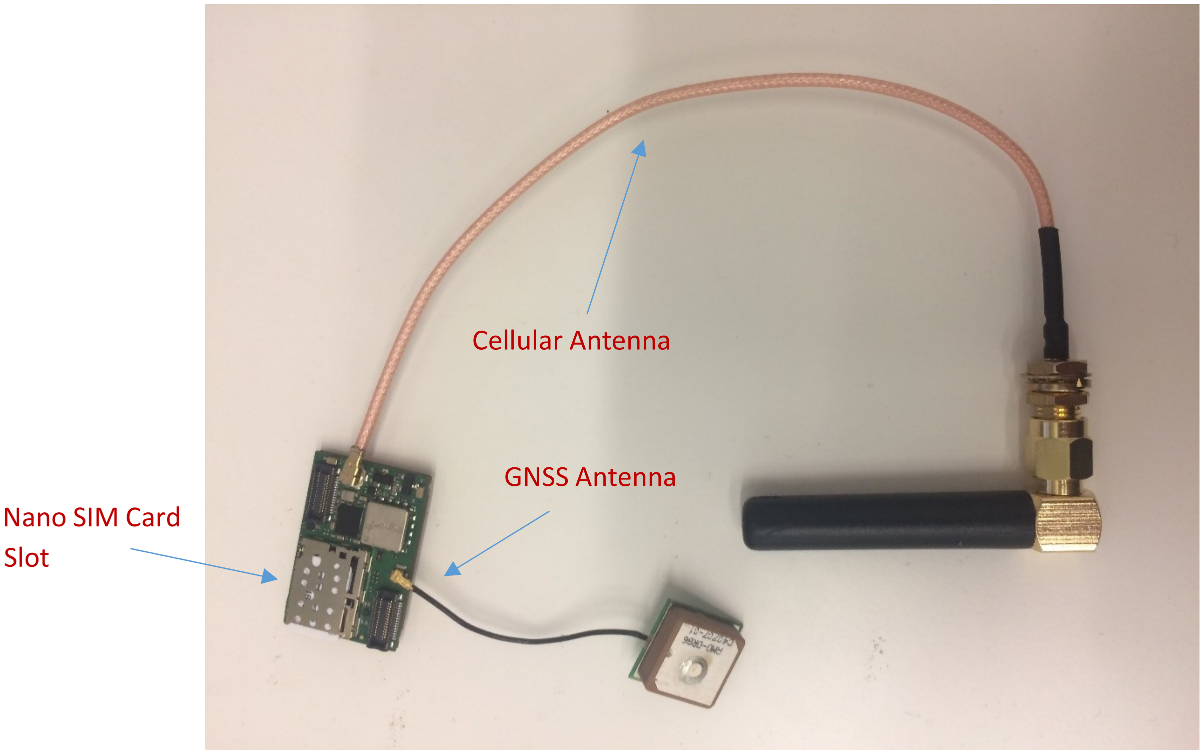


Figure 1 – connection of ORG2101 antennas and SIM card

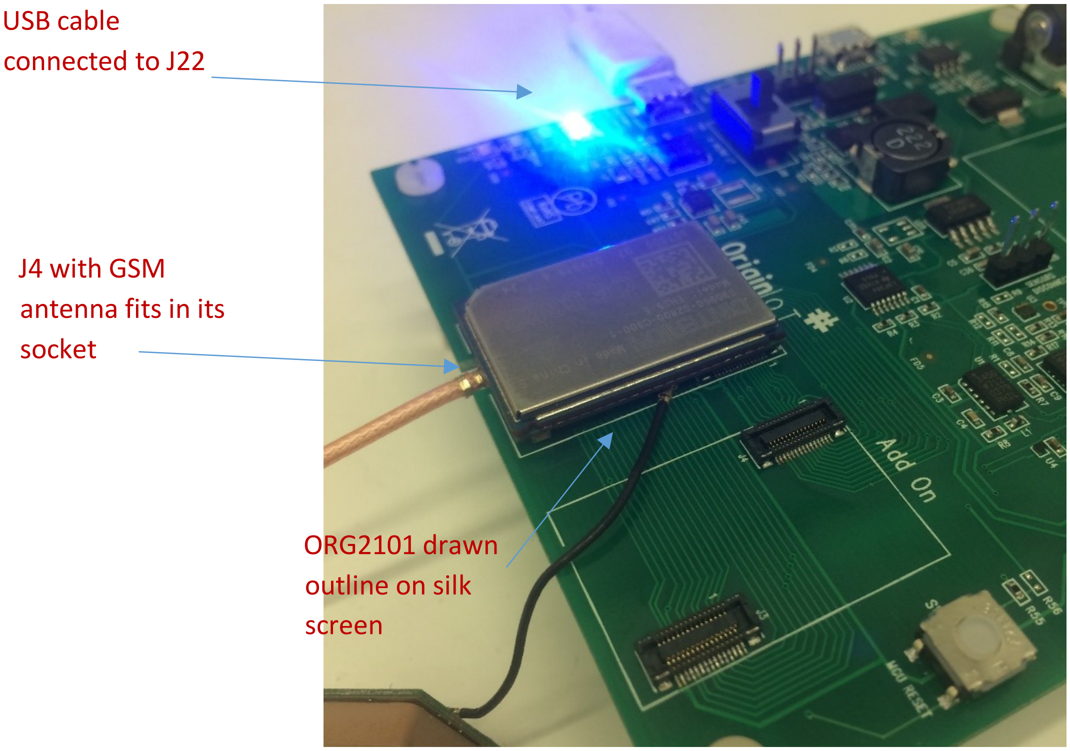
4. The OriginIoT module should be connected to EVK board up-side down (components' side down) to the area near the OriginIoT logo on the EVK, see figure 2.

Please notice: connection of OriginIoT to EVK is unidirectional – J1 of the module (closer to MCU) should connect to J1 on the EVK.

Note: Extra precaution is required upon insertion / withdrawal of ORG2101 to EVK. Incautious handling can alter the performance of the connectors.

You know that you are connected correctly if:

1. J4 for U.FL antenna fits inside its socket on the EVK
2. OriginIoT fits in white outlines drawn on the EVK (silk screen)



USB cable connected to J22

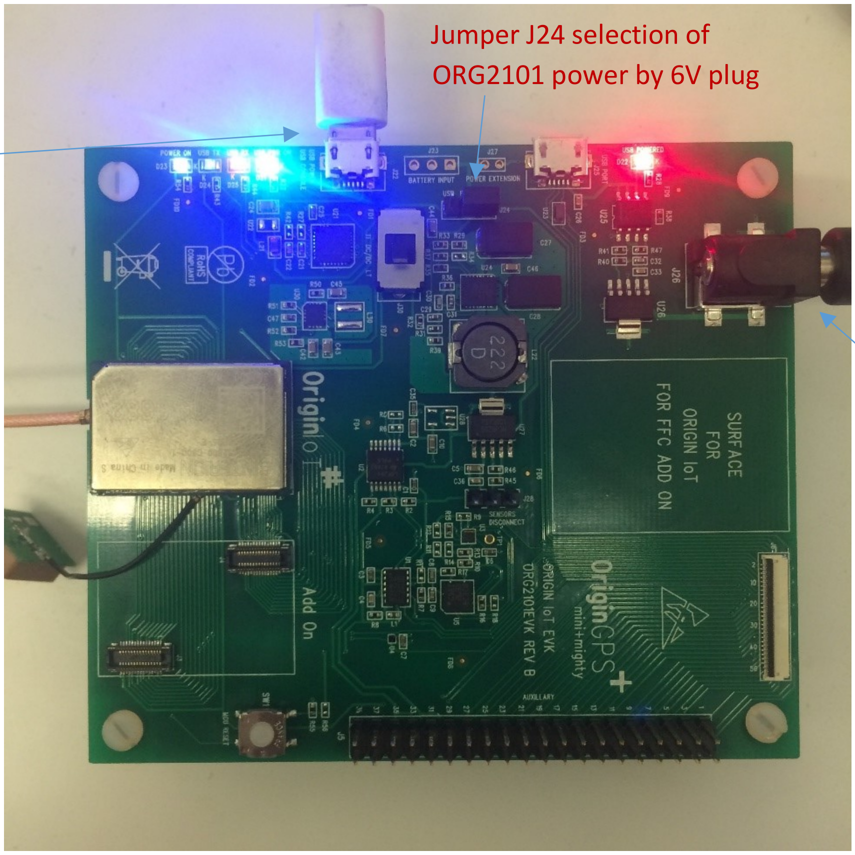
J4 with GSM antenna fits in its socket

ORG2101 drawn outline on silk screen

Figure 2 – mating ORG2101 to EVK correctly

5. Connect USB Power/Console of the EVK to PC via micro USB cable (J22). See figure 3.

6. Supply power to EVK - Connect the power adapter to the EVK (J26). See figure 3.



Jumper J24 selection of ORG2101 power by 6V plug

USB connector to PC via J22

EVK powered by external 6V power supply

Figure 3 – powering up EVK and ORG2101

7. Open terminal on your PC and connect to corresponding COM port of the OriginIoT with a baud rate of 460800 bps.
In Tera Term – go to Setup - > Serial Port (See figure 3a), select the COM port and set baud rate to 460800 bps. Press “OK”. See figure 3b.
Full communication parameters:

- Baud Rate: 460800
- Data Bits: 8
- Stop Bits: 1
- Flow Control: None
- Parity: None

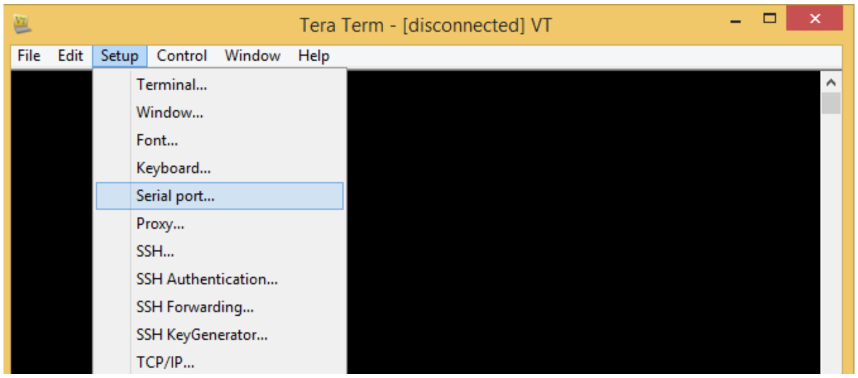
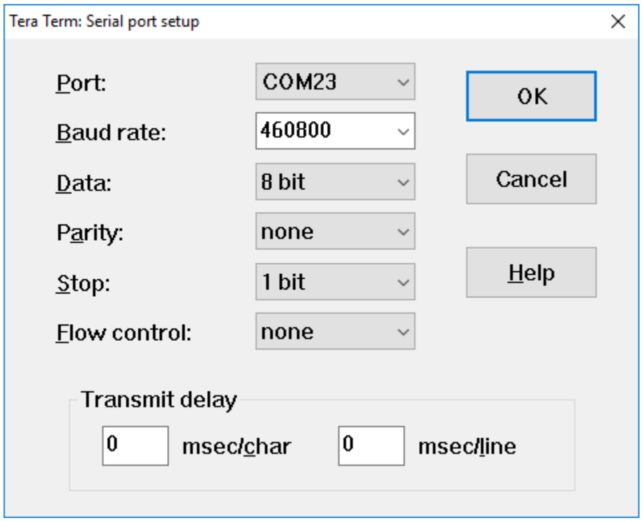


Figure 3a – Serial port in Tera Term


Tera Term: Serial port setup

Port: COM23

Baud rate: 460800

Data: 8 bit

Parity: none

Stop: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Figure 3b – COM settings in Tera Term

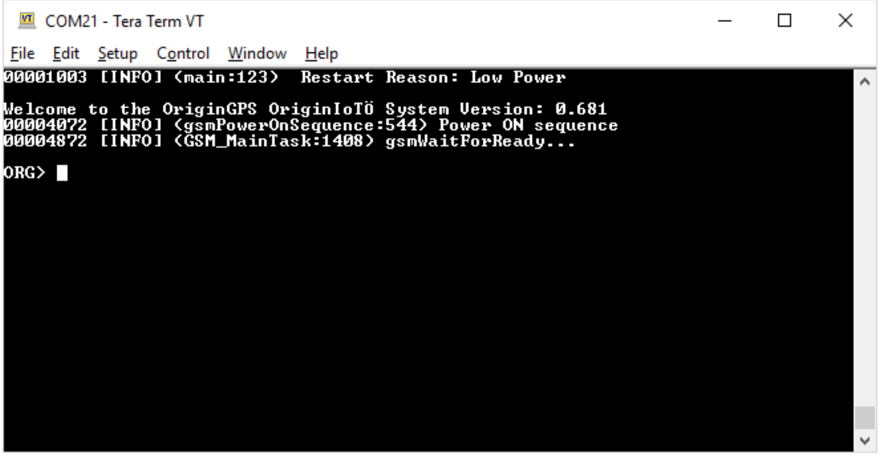
8. Supply power to OriginIoT via DC/DC circuit by placing jumper J24. See figure 3.

Note: the EVK can be powered by USB (J22) or Plug (J26) determined by jumper J24.

Note: you can choose one of two DC/DC circuits provided on the EVK – one by Texas Instruments TPS63070 or one by Linear Technology LTC3113. To switch between them use slide switch J30

After supplying power to OriginIoT you should see a welcome message and GSM/LTE start-up messages.

8. While connected, press Enter key in the terminal.
You should see the command line “ORG>” in the terminal window. See Figure 3c.



```
00001003 [INFO] <main:123>  Restart Reason: Low Power

Welcome to the OriginGPS OriginIoTÖ System Version: 0.681
00004072 [INFO] <gsmPowerOnSequence:544> Power ON sequence
00004872 [INFO] <GSM_MainTask:1408> gsmWaitForReady...

ORG>
```

Figure 3c – OriginIoT connected to Debug Console

Now the module can be accessed via the debug console. Please refer to document "OriginIoT debug console user guide" for the complete list of functions available via the debug console.

10. CONNECTING ORIGINIOT TO A SERVER

10.1. CELLULAR NETWORK PARAMETERS

In order to allow OriginIoT sending data via cellular network, initial cellular configuration should be defined. This is done one time only for your specific SIM card, and includes several parameters of your SIM network:

- APN – Access Point Network
- Username**
- Password**

**Username and Password are required only by several cellular providers and sometimes they are not needed at all.
This data can be found on the cellular providers' website or with the SIM card you received from your operator.

10.2. ATTRIBUTING ORIGINIOT TO A SERVER

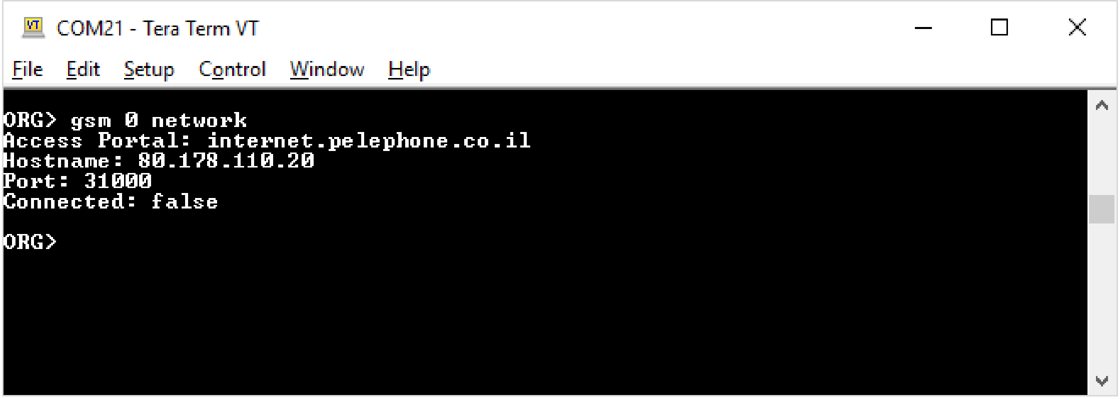
OriginIoT™ comes preconfigured to connect to OriginGPS's evaluation server at http://18.184.85.107. Please contact our support team at iot@origingps.com if you need your products preconfigured with a different server and network configuration.

In order to attribute the OriginIoT to this server or a different server, please follow these steps:

1. To see current OriginIoT network configuration, enter the following command:

```
gsm 0 network
```

See figure 4 for output example.



```
ORG> gsm 0 network
Access Portal: internet.pelephone.co.il
Hostname: 80.178.110.20
Port: 31000
Connected: false

ORG>
```

Figure 4 – OriginIoT Configuration example

2. The “Access Portal” is the APN of your SIM card provider’s network.

The “Hostname” is the hostname or IP address of the Cloud based Application – the server to which the module is connecting to.

The “Port” is the port used by the Cloud Application to receive all messages from the module.

The “Connected” line is set to “true” if the GSM/LTE has successfully connected to the Access Portal. Otherwise it is set to “false”.

3. To set these fields use the command:

```
gsm <instance> set <portal|hostname|port> ...
```

For example, the command to change the APN is:

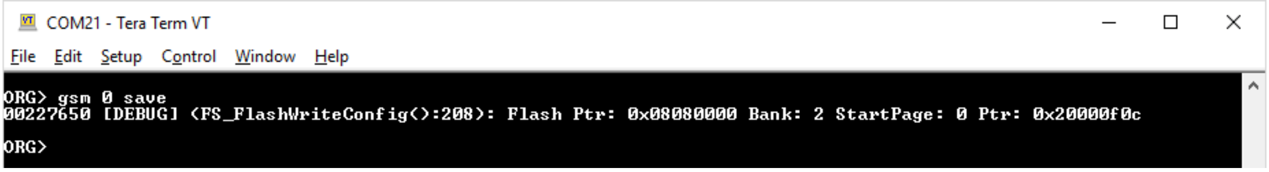
```
gsm 0 set portal
```

The operator will be prompted with a “→” and must fill in the string that is the new portal setting. The same must be done for each of the other network settings if required to be changed. The same method applies if changing the server address or the port is required.

4. Once the changes are made, the operator must save them to flash using the command:

```
gsm 0 save
```

See figure 5 for example.



```
ORG> gsm 0 save
00227650 [DEBUG] <FS_FlashWriteConfig():208>: Flash Ptr: 0x08080000 Bank: 2 StartPage: 0 Ptr: 0x20000f0c

ORG>
```

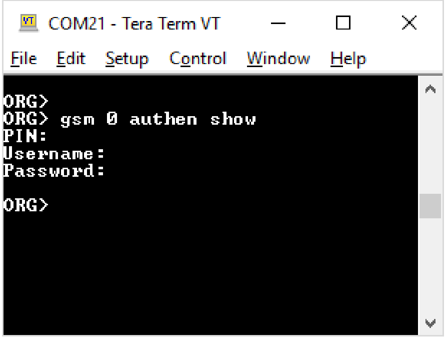
Figure 5 – Setting OriginIoT network configuration and saving to flash

10.3. ENTERING CELLULAR-NETWORK PARAMETERS – GSM/LTE AUTHENTICATION COMMAND

In order to configure your SIM card, use in the debug console the “authen” command. To see current configuration, input:

```
gsm 0 authen show
```

You will receive the following output:



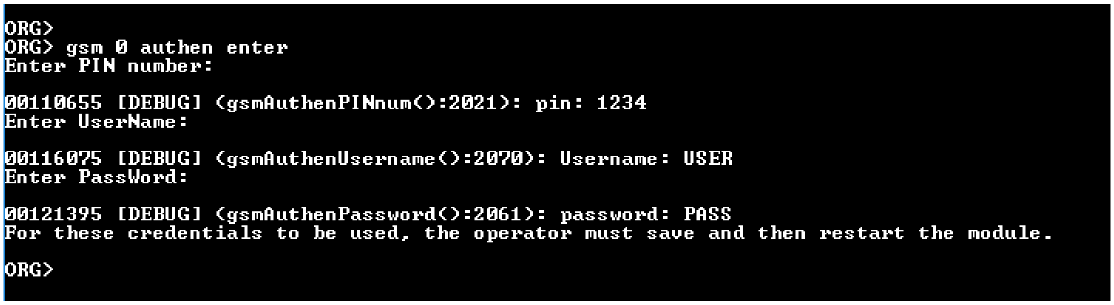
```
ORG>
ORG> gsm 0 authen show
PIN:
Username:
Password:

ORG>
```

Since the module is shipped without authentication settings, they are shown as empty fields here. The "authen" sub-command lets the operator set 3 fields: PIN, Username and Password. The PIN is a 4-8 digit number provided with the SIM card, if required by the operator. If Username and Password are required (see section 9.3) then the values of these fields are provided by the network operator as well. If there is no requirement for them then these fields can remain blank.

To set these fields use the command:

```
gsm 0 authen enter
```

The operator will be prompted for each of the 3 fields. If the field is not required, exit the field by typing the "Enter" key.



```
ORG>
ORG> gsm 0 authen enter
Enter PIN number:

00110655 [DEBUG] (gsmAuthenPINnum():2021): pin: 1234
Enter UserName:

00116075 [DEBUG] (gsmAuthenUsername():2070): Username: USER
Enter PassWord:

00121395 [DEBUG] (gsmAuthenPassword():2061): password: PASS
For these credentials to be used, the operator must save and then restart the module.

ORG>
```

Once completed, the operator can once again use the command " gsm 0 authen show" to check that the values entered are correct. If they values are correct, they must be saved to flash using the following command:

```
gsm 0 authen save
```

The next time the system boots these fields will be used in the wireless network connectivity.

5. In the end of the process, while the EVK is successfully connected to a server, the output should look like this:

```
ORG> gsm 0 network
Access Portal: internet.com
Hostname: 80.178.110.20
Port: 31000
Connected: true
```

IP Address: 25.48.187.64 (the IP is dynamic)

The last entry "IP Address" is the GSM/LTE's IP address given to the module by the Internet providers DHCP servers and is not a configurable item.

11. ORIGINIOT WEB APPLICATION

11.1. OVERVIEW

This chapter applies to users who choose to use OriginIoT evaluation server. The OriginIoT™ web application server is an interface to configure and view modules with associated interfaces, devices, messages, and commands. User-created and System commands can be executed via user friendly GUI interfaces.

11.2. USER LOG IN

Website address of the web application is: http://18.184.85.170 (resides on Amazon Web Services cloud).

When entering this address, you will be automatically redirected to the login page. See figure 6.

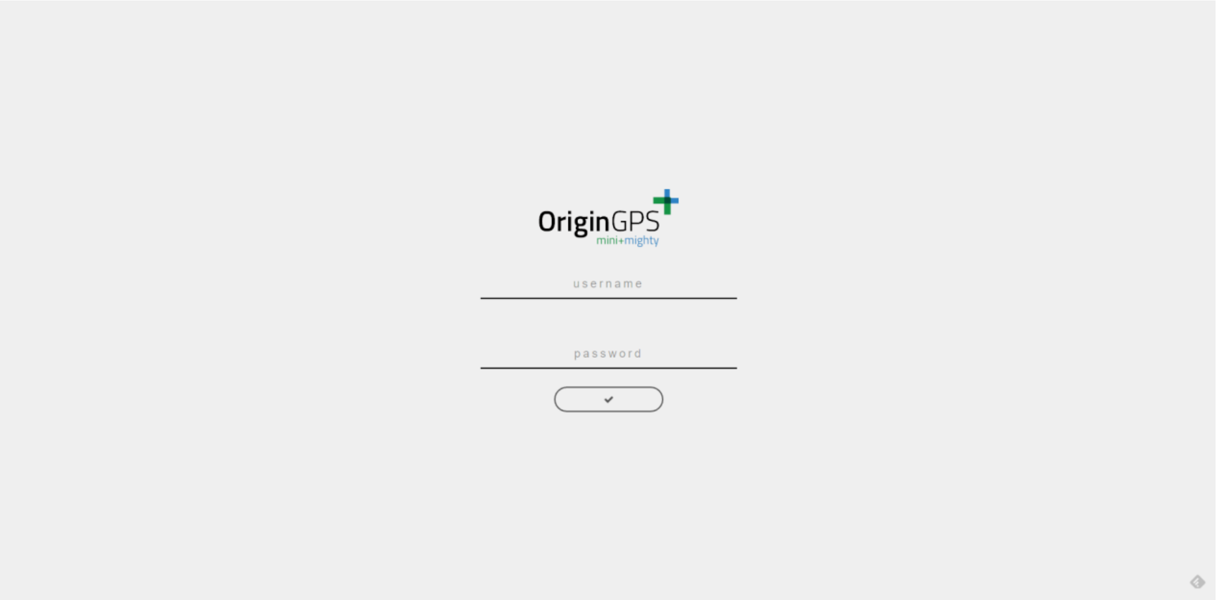
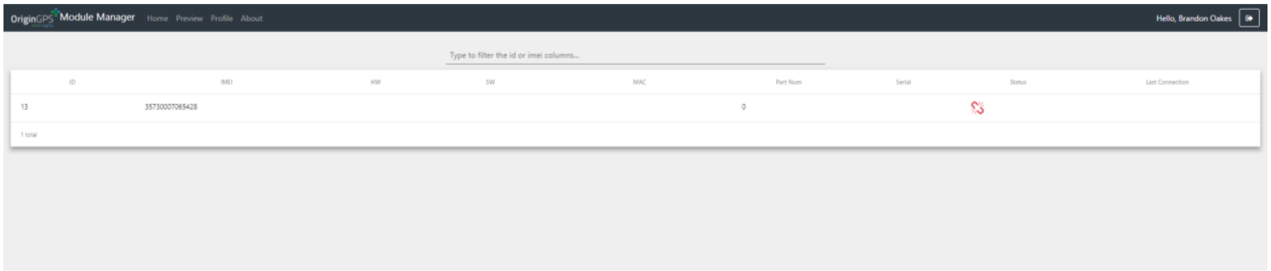


Figure 6 – OriginIoT Web Application, login page

Please enter the credentials you received from your OriginGPS representative. You will be able to change the password from the Profile tab. Clicking the ✓ symbol will redirect you to the Home tab.

11.3. WEB HOME TAB

Each OriginIoT module has an ID number associated to it by this specific server. The unique ID in the server is associated with the module upon its first registration to the server. The unique ID is linked to the module's IMEI (International Mobile Station Equipment Identity). The modules that you purchased are added by us to your profile and will appear once you are logged in.



Type to filter the id or imei columns...

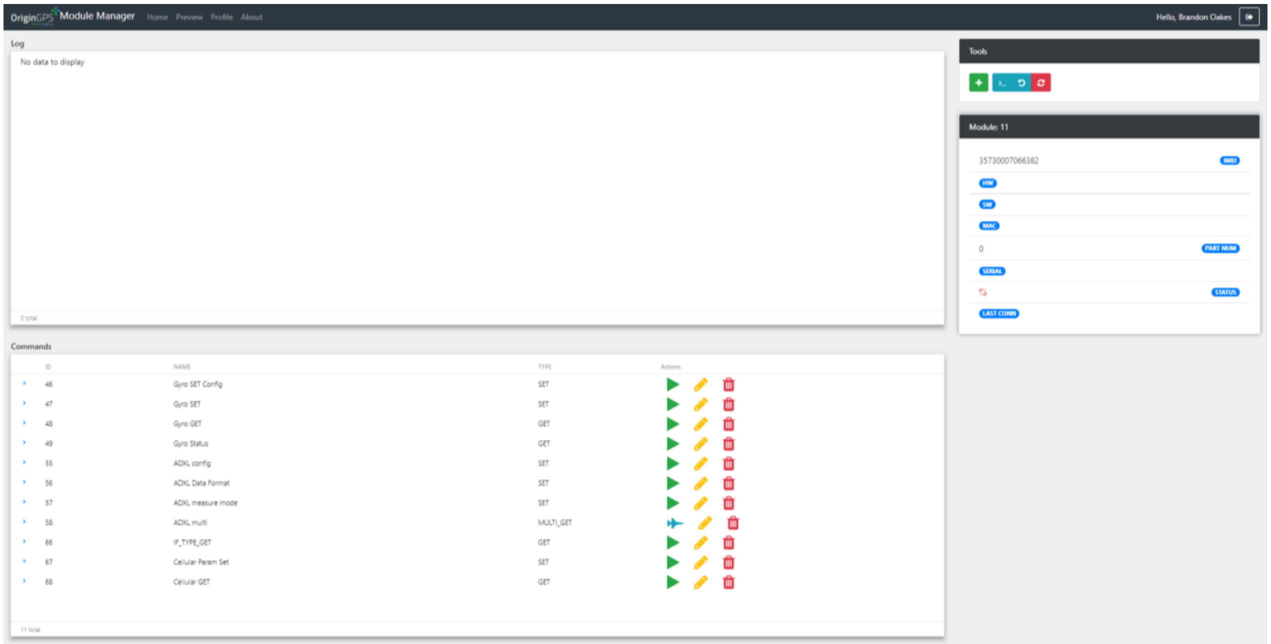
ID	IMEI	HW	SW	MAC	Part Num	Serial	Status	Last Connection
13	35730007065428				0		⛓	

1 total

Figure 7 – OriginIoT Home Tab

Clicking the row of the module will forward you to the device details page. Red link symbol in the status column shows the connection status of your OriginIoT. Red symbol means no connection, green means good connection between module and server. Note you will need to refresh the page to get the latest connectivity status from the modules.

11.4 MODULE DETAILS TAB



Log

No data to display

0 total

35730007066382	IMEI
	HW
	SW
	MAC
0	PART NUM
	SERIAL
⛓	STATUS
	LAST CONN

Commands

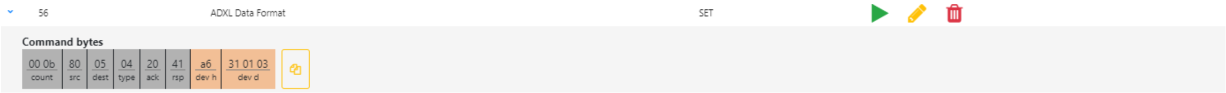
	ID	NAME	TYPE	Actions
›	46	Gyro SET Config	SET	▶ ✎ 🗑
›	47	Gyro SET	SET	▶ ✎ 🗑
›	48	Gyro GET	GET	▶ ✎ 🗑
›	49	Gyro Status	GET	▶ ✎ 🗑
›	55	ADXL config	SET	▶ ✎ 🗑
›	56	ADXL Data Format	SET	▶ ✎ 🗑
›	57	ADXL measure mode	SET	▶ ✎ 🗑
›	58	ADXL multi	MULTI_GET	✈ ✎ 🗑
›	66	IF_TYPE_GET	GET	▶ ✎ 🗑
›	67	Cellular Param Set	SET	▶ ✎ 🗑
›	68	Cellular GET	GET	▶ ✎ 🗑

11 total

Figure 8 – Module Details Tab

There are 4 windows in the module details tab: Log, Commands, Tools and Details.

- Log – showing data sent and received to/from the module. The log is saved locally, it will be deleted if page is refreshed.
- Commands – showing all commands that were entered in the system. Clicking on the command row will open up the command details line that shows the byte string that will be sent to the module upon execution of the command (see figure 8). Clicking on the yellow copy button will copy the command byte string to the clipboard as a string (00 0b 80 05 04…).



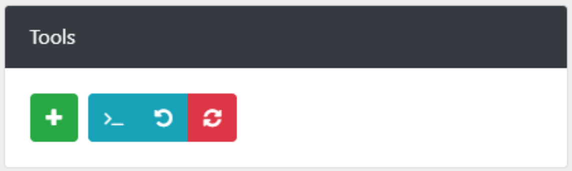
56 ADXL Data Format SET

Command bytes

00 0b	80	05	04	20	41	a6	31 01 03
count	src	dest	type	ack	rsp	dev h	dev d

Figure 8 – Command details

Each command has three buttons in its line: play, edit and delete. Play takes the form of a blue airplane symbol when a mutli_GET or Multi_BUF command is entered. The green play symbol executes the command, you will see the sent command in the log. The blue airplane button will route the user to the preview tab, where he can see the data in the widgets. The yellow edit button will open the new message windows pre-loaded with the data of the edited message. The red delete button remove the command from the list and from the database.



- Tools – allows the user to choose from 3 available options (red reset button not implemented yet).
  - The green “+” button opens a new message window and allows the user to add a new command.
  - The “>_” button allows the user to write a custom message, bypassing the new message window restrictions.
  - The blue “loopback” button ↺ initiates a loopback test to test the link status of the module to the server. By executing a loopback test you can ping the module in order to check its connectivity status (this test only checks the connectivity of the module to the server, not the link state or status of peripheral sensors/devices). The structure of the loopback test message is described in API specifications document. The created sent message and the message received from the module will show up on the log area. If unsuccessful, you will not receive any response.
    Note: you will not receive a success notification if the module is not yet connected to the server.
    If you are connected to debug console, please wait until you receive:

    ```
    00021788 [INFO] (MCU_ProcessAdminMsg:1086) Module is registered with
    host application.
    ```

    In the debug console, successful loopback message will print:
    ORG> (MCU_ProcessSetMsg:1027): Received App Loopback message

    ```
    0x000000: 00 0d 80 00 04 20 41 00 64 01 02 03 04              .....
    A.d....
    (MCU_ProcessSetMsg:1043): Outgoing message...
    0x000000: 00 0c 00 80 10 20 00 65 01 02 03 04                  .....
    .e....
    ```

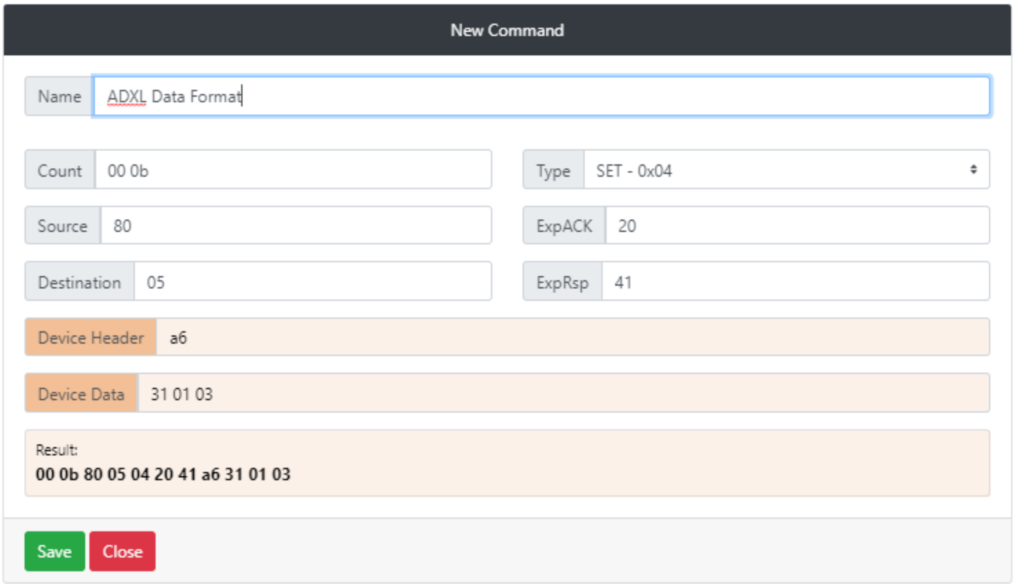
- Module Details – shows the IMEI and the link state of the module, with place holder for future HW and SW versions, last connection time, MAC address and module serial number.

11.5 ADDING A NEW COMMAND

For the user convenience, the New Command window allows to easily define a message via online form. There are no format constraints for the inputs so user has to make sure data is inserted correctly, two characters for bytes (Hexadecimal) with space between bytes. The count field is updated automatically when new bytes are added/removed.

Please consult the OriginSmart API Specification document to learn what inputs are required in Type, Source, Destination, ExpAck, ExpRsp, Device Header and Device Data.

The result command is the actual string that will be sent to the module.



New Command

Name: ADXL Data Format

Count: 00 0b

Type: SET - 0x04

Source: 80

ExpACK: 20

Destination: 05

ExpRsp: 41

Device Header: a6

Device Data: 31 01 03

Result:
00 0b 80 05 04 20 41 a6 31 01 03

Figure 9 – New Command

11.6 ADDING A CUSTOM COMMAND

The user can send any string he wants to the device using this custom command feature. If the command cannot be interpreted by the OriginIoT module, no response will be issued. If the first two bytes in the system, which are dedicated for message length, are not correct, the message will not be sent to the module.

The user can reach the custom command window by clicking on the “>_” button in the Tools section. Clicking on Close will close the window without sending. Clicking on Send will send the message and close the window.

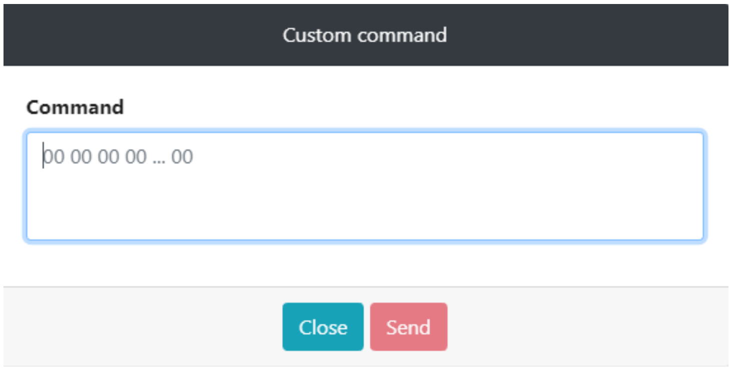
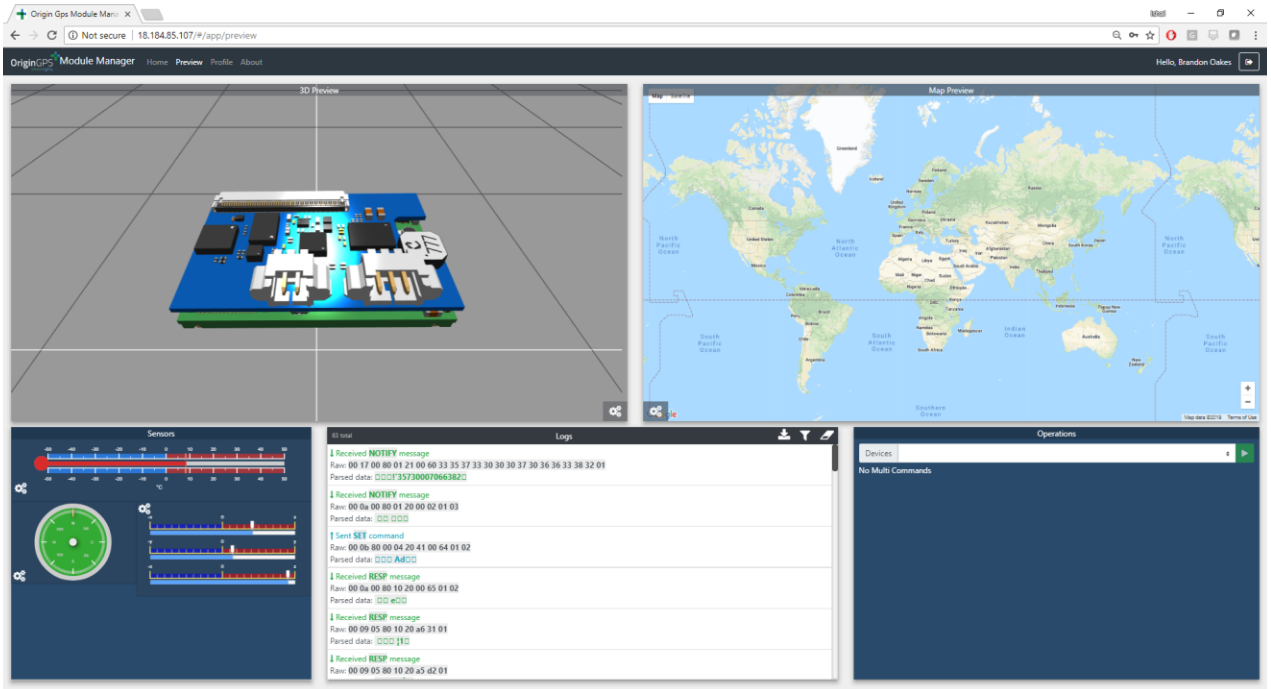


Figure 9 – Command details

11.7 WIDGET PREVIEW TAB



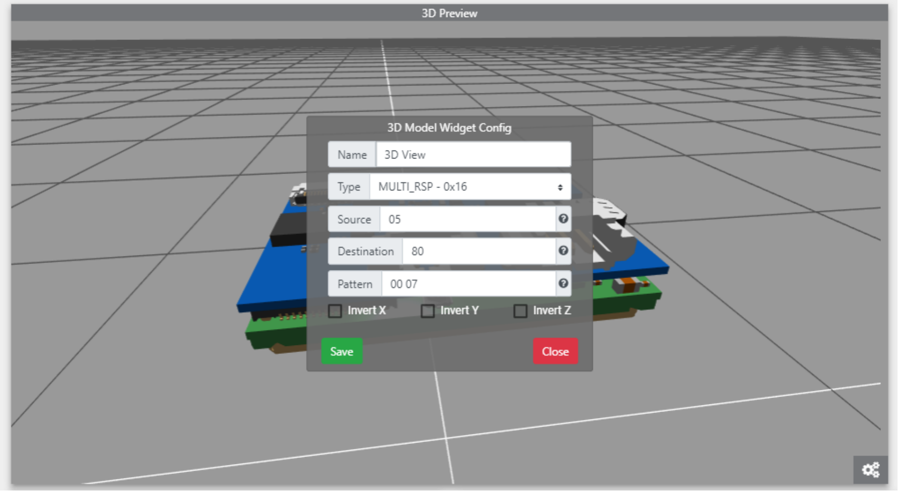
```
63 total
↓ Received NOTIFY message
Raw: 00 17 00 80 01 21 00 60 33 35 37 33 30 30 30 37 30 36 36 33 38 32 01
Parsed data: □□□!`357300070663820
↓ Received NOTIFY message
Raw: 00 0a 00 80 01 20 00 02 01 03
Parsed data: □□ □□□
↑ Sent SET command
Raw: 00 0b 80 00 04 20 41 00 64 01 02
Parsed data: □□□ Ad□□
↓ Received RESP message
Raw: 00 0a 00 80 10 20 00 65 01 02
Parsed data: □□ e□□
↓ Received RESP message
Raw: 00 09 05 80 10 20 a6 31 01
Parsed data: □□□ ¦1□
↓ Received RESP message
Raw: 00 09 05 80 10 20 a5 d2 01
```

Figure 10 – Widget preview tab

OriginGPS has developed some sample widgets to demonstrate the functionality of the module. These widgets include:

- 3D Preview of the module+addon1 assembly, controllable by accelerometer
- Google maps preview, showing GNSS locations
- Sensors preview, showing temperature, compass, and gyro data.
- Logs, showing raw data transferred to and from the module. The log widget allows you to filter some of the messages (GNSS, Registration, Loop Back, Multi_Get). The log widget also allows you to download the data of the log as a .log file.
- Operations, allowing the user to select which device to operate according to IMEI; start/stop listening to selected device, and start/stop Multi_Get and Multi_Buf messages.

11.8 WIDGET CONGIFURATION



3D Model Widget Config

Name: 3D View

Type: MULTI_RSP - 0x16

Source: 05

Destination: 80

Pattern: 00 07

□ Invert X □ Invert Y □ Invert Z

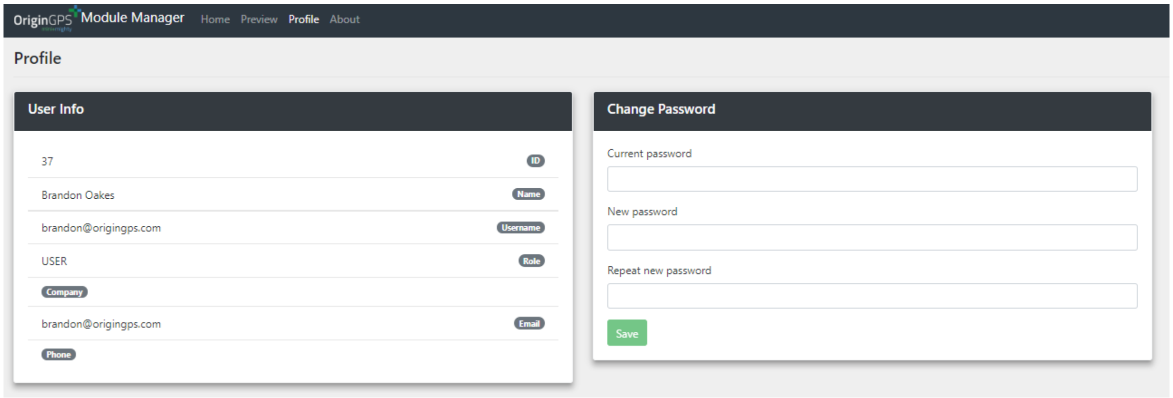
Figure 10 – Widget configuration window

Each widget can be configured by clicking the ⚙ button in its corner. In order for the widget to present the required information, the user needs to set the type of message to listen to. The fields name, type, source, destination and pattern are common to all widgets. Widgets that are planed to be used with 3 dimensional data have invert axis function that adds a (-) sign to all raw data (in case your sensor is physically oriented in negative direction).

In “name” you can put any string as input. “type” allows you to choose from a drop-down menu. Under “source” and “destination” you should choose the hexadecimal interface ID from which and to where your message is sent (the message you want to present with the widget).

“Pattern” allows you to choose which of the messages from the chosen interface to listen to. The pattern should be in Hexadecimal byte sting (not a string). For example, if you look for a $GNRMC NMEA message, you should enter 24 47 4e 52 4d 43.

11.9 PROFILE TAB



Profile

User Info

37 ID

Brandon Oakes Name

brandon@origingps.com Username

USER Role

Company

brandon@origingps.com Email

Phone

Change Password

Current password

New password

Repeat new password

Figure 11 – Profile Tab

The profile tab allows the user to see its info: ID, Name, Username, role (user or admin), company, email and phone number. This tab also allows the user to change his password.

12. OPERATING THE SENSORS ON THE EVK

The EVK has 5 sensors (including the GNSS sensor on the OriginIoT module) connected via 3 interfaces: GNSS and temperature sensors connected via two UART interfaces (IF_20 and IF_03 respectively) Accelerometer, Compass and Gyroscope connected via I2C (IF_05).

12.1 PARAMETER SETUP

All sensors must to be configured (“Committed”) on the device using an IF_PARAM_SET message has specified in the API spec document.

The required parameters for each peripheral device can be found in the device datasheet. OriginIoT module does not restrict the developer with the specific devices that he can connect to the module as long as they interface with I2C, UART, SPI or GPIO. Every device has its own parameters that are required in this step. For convenience, table 3 below lists all the parameters that are required for configuring the devices that the EVK equips.

<table>
<tr><th>Device</th><th>Device P/N</th><th>IF_ID</th><th colspan="7">Parameters</th></tr>
<tr><td colspan="3">UART Type Interface</td><td>Baud Rate</td><td>Flow Control</td><td>Stop Bit</td><td>Parity</td><td>Data Bits</td><td>Data Ready IF_ID</td><td>Data Over Run Action</td></tr>
<tr><td>GPS</td><td>ORG4572</td><td>03</td><td>4800-230400</td><td>Disable</td><td>1</td><td>None</td><td>8</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Temperature</td><td>TMP104</td><td>20</td><td>Auto-baud mode 3</td><td>Disable</td><td>1</td><td>None</td><td>7</td><td>N/A</td><td>N/A</td></tr>
<tr><td colspan="3">I2C Type Interface</td><td>Addressing Mode</td><td>Bit Rate</td><td colspan="2">Register Type</td><td>Data Ready IF_ID</td><td>Data Over Run Action</td><td>Device Address (0x)</td></tr>
<tr><td>eCompass</td><td>IST8304</td><td>05</td><td>7-bit</td><td>100 kbps</td><td colspan="2">BYTE</td><td>N/A</td><td>N/A</td><td>18</td></tr>
<tr><td>Gyroscope</td><td>DM00168691</td><td>05</td><td>7-bit</td><td>100 kbps</td><td colspan="2">BYTE</td><td>N/A</td><td>N/A</td><td>d2</td></tr>
<tr><td>Accelerometer</td><td>ADXL345</td><td>05</td><td>7-bit</td><td>100 kbps</td><td colspan="2">BYTE</td><td>N/A</td><td>N/A</td><td>a6</td></tr>
</table>

Table 3 – Configuration parameters for EVK sensors

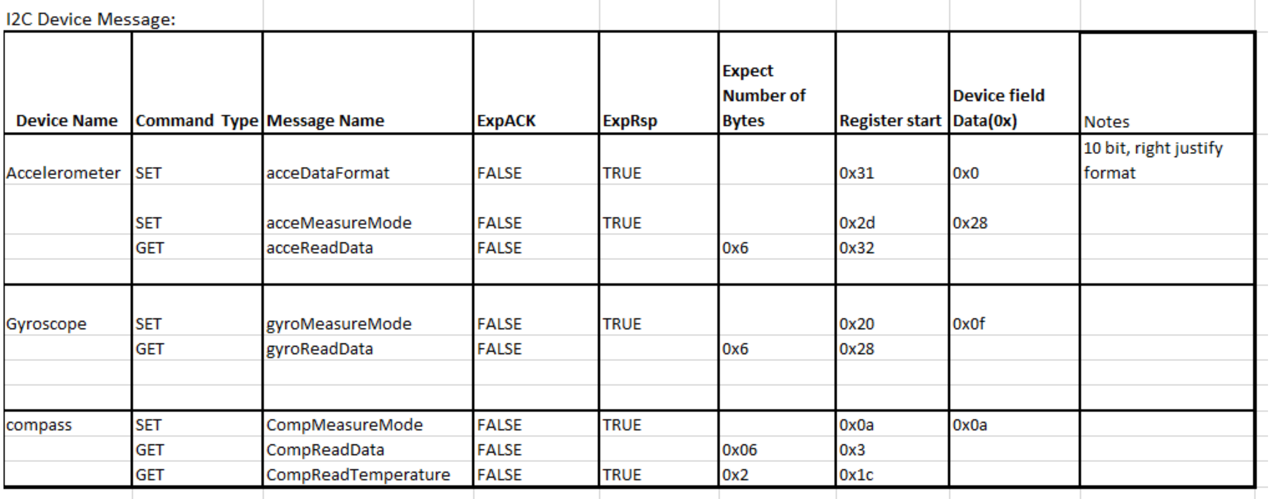
If you are using a debug console, you will see this message printed after successful configuration message (in this example we commit the accelerometer ADXL345):

```
(MCU_ProcessParamSetMsg:575): Parameter Set Dump...
0x000000: 00 13 80 00 04 20 41 00 06 05 03 00 01 01 53 00 .....
A.......S.
0x000010: 00 00 00                                        ...
(I2C_ProcessParamSetMessage:360): PARAM_SET command...
0x000000: 00 13 80 00 04 20 41 00 06 05 03 00 01 01 53 00 .....
A.......S.
0x000010: 00 00 00                                        ...
(I2C_ProcessParamSetMessage:413): Outgoing I2C Response message...
0x000000: 00 0b 00 80 10 20 00 07 05 03 01                ..... .....
```

12.2 ACTIVATING THE SENSORS

To understand what messages shall be sent to achieve a certain functionality for a specific device one can refer to the datasheets of the devices. Usually under “Register map” you will find the description of the functionality each register address is responsible for and what kind of data to set in it to achieve this functionality. This will help you fill the register start, device field data and expected number of bytes fields in SET and GET messages (please refer to API spec document to see the complete message structure). The full register map of the devices on the EVK can be found in appendix C.

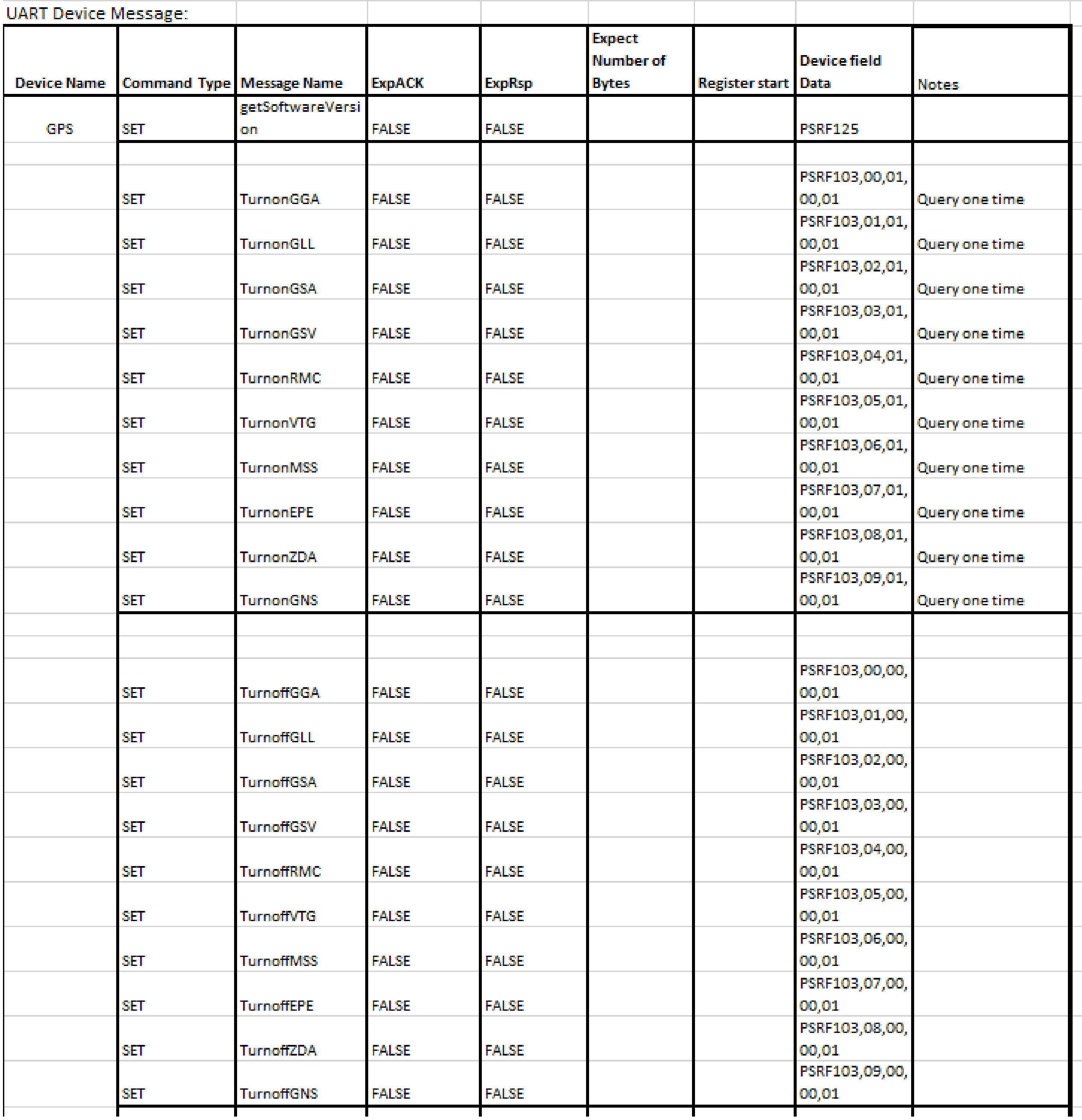
For your convenience, tables 4 and 5 offer sample set and get message configurations for the sensors.



I2C Device Message:

Device Name	Command Type	Message Name	ExpACK	ExpRsp	Expect Number of Bytes	Register start	Device field Data(0x)	Notes
Accelerometer	SET	acceDataFormat	FALSE	TRUE		0x31	0x0	10 bit, right justify format
	SET	acceMeasureMode	FALSE	TRUE		0x2d	0x28	
	GET	acceReadData	FALSE		0x6	0x32		
Gyroscope	SET	gyroMeasureMode	FALSE	TRUE		0x20	0x0f	
	GET	gyroReadData	FALSE		0x6	0x28		
compass	SET	CompMeasureMode	FALSE	TRUE		0x0a	0x0a	
	GET	CompReadData	FALSE		0x06	0x3		
	GET	CompReadTemperature	FALSE	TRUE	0x2	0x1c		

Table 4 – Configuration for I2C Messages



UART Device Message:

Device Name	Command Type	Message Name	ExpACK	ExpRsp	Expect Number of Bytes	Register start	Device field Data	Notes
GPS	SET	getSoftwareVersion	FALSE	FALSE			PSRF125	
	SET	TurnonGGA	FALSE	FALSE			PSRF103,00,01,00,01	Query one time
	SET	TurnonGLL	FALSE	FALSE			PSRF103,01,01,00,01	Query one time
	SET	TurnonGSA	FALSE	FALSE			PSRF103,02,01,00,01	Query one time
	SET	TurnonGSV	FALSE	FALSE			PSRF103,03,01,00,01	Query one time
	SET	TurnonRMC	FALSE	FALSE			PSRF103,04,01,00,01	Query one time
	SET	TurnonVTG	FALSE	FALSE			PSRF103,05,01,00,01	Query one time
	SET	TurnonMSS	FALSE	FALSE			PSRF103,06,01,00,01	Query one time
	SET	TurnonEPE	FALSE	FALSE			PSRF103,07,01,00,01	Query one time
	SET	TurnonZDA	FALSE	FALSE			PSRF103,08,01,00,01	Query one time
	SET	TurnonGNS	FALSE	FALSE			PSRF103,09,01,00,01	Query one time
	SET	TurnoffGGA	FALSE	FALSE			PSRF103,00,00,00,01	
	SET	TurnoffGLL	FALSE	FALSE			PSRF103,01,00,00,01	
	SET	TurnoffGSA	FALSE	FALSE			PSRF103,02,00,00,01	
	SET	TurnoffGSV	FALSE	FALSE			PSRF103,03,00,00,01	
	SET	TurnoffRMC	FALSE	FALSE			PSRF103,04,00,00,01	
	SET	TurnoffVTG	FALSE	FALSE			PSRF103,05,00,00,01	
	SET	TurnoffMSS	FALSE	FALSE			PSRF103,06,00,00,01	
	SET	TurnoffEPE	FALSE	FALSE			PSRF103,07,00,00,01	
	SET	TurnoffZDA	FALSE	FALSE			PSRF103,08,00,00,01	
	SET	TurnoffGNS	FALSE	FALSE			PSRF103,09,00,00,01	

Table 5 – Configuration for UART Messages

If you are using a serial debug console you will be able to see the commands received on the module and the responses issued by the module. For example, this is what you will receive on the console after successful dataFormat SET message, measureMode SET message and readData GET messages for the accelerometer:

```
(i2cProcessSetMsg:759): Set message...
0x000000: 00 0b 80 05 04 20 41 53 31 01 00                .....  AS1..
```

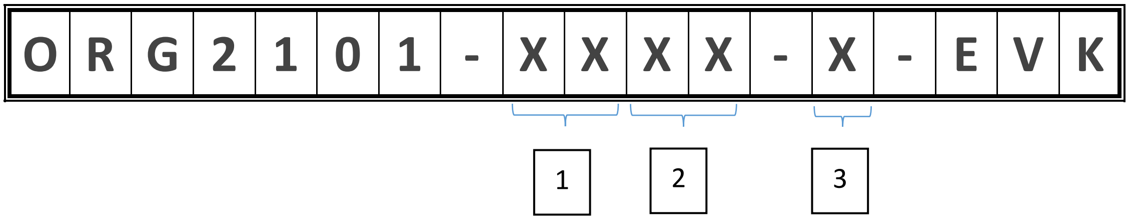
```
00260504 [DEBUG] (i2cProcessSetMsg():764): the Ack flag: 0x20 the RSP
Flag: 0x41
00260505 [DEBUG] (i2cProcessSetMsg():769): Device Address: 0x53
00260520 [DEBUG] (i2cProcessSetMsg():784): The Start Reg: 0x31
00260532 [DEBUG] (i2cProcessSetMsg():797): Number of bytes: 1
(i2cProcessSetMsg:834): Outgoing Set msg Response...
0x000000: 00 09 05 80 10 20 53 31 01                       ..... S1.

(i2cProcessSetMsg:759): Set message...
0x000000: 00 0b 80 05 04 20 41 53 2d 01 28                 ..... AS-.(
00275941 [DEBUG] (i2cProcessSetMsg():764): the Ack flag: 0x20 the RSP
Flag: 0x41
00275942 [DEBUG] (i2cProcessSetMsg():769): Device Address: 0x53
00275957 [DEBUG] (i2cProcessSetMsg():784): The Start Reg: 0x2d
00275969 [DEBUG] (i2cProcessSetMsg():797): Number of bytes: 1
(i2cProcessSetMsg:834): Outgoing Set msg Response...
0x000000: 00 09 05 80 10 20 53 2d 01                       ..... S-.

(i2cProcessGetMsg:443): Get Message...
0x000000: 00 09 80 05 02 20 06 53 32                       ..... .S2
(i2cProcessGetMsg:503): Outgoing Get response msg...
0x000000: 00 0f 05 80 10 20 53 32 06 0a 00 07 00 e7 00    .....
S2.......
```

12.2 WIDGET CONFIGURATION

For your convenience we added table 5 below to help in configuring the widgets to listen to the sample sensors messages.

Widget	Sensor	Type	Source	Destination	Pattern
3D Preview	**ADXL345**	**MULTI_RSP – 0x16**	**05**	**80**	**00 06**
Map Preview	**GNSS (Internal)**	**NOTIFY – 0x01**	**03**	**80**	**24 47 4e 52 4d 43**
Temperature	**IST8304**	**MULTI_RSP – 0x16**	**05**	**80**	**00 02**
Compass	**IST8304**	**MULTI_RSP – 0x16**	**05**	**80**	**00 06**
Gyro	**DM00168691**	**MULTI_RSP – 0x16**	**05**	**80**	**00 06**

13. ORDERING INFORMATION



1. Selection of communication technology:

   2G = 2G GPRS

   3G = 3G HSPA

   C1 = LTE Cat 1

   CM = LTE Cat M*

   NB = LTE Cat NB1 *

2. Selection of communication:

   EU= EMEA/APAC

   GL=Global

   US = North America

   JA = Japan

   CH = China

3. Selection of SIM type

   T = Traditional plastic SIM card holder

   E = Embedded SIM

* Please contact iot@origingps.com

List of orderable parts			
ORG2101-2GEU-T-EVK	ORG2101-3GEU-T-EVK	ORG2101-C1US-T-EVK	
ORG2101-2GEU-E-EVK	ORG2101-3GEU-E-EVK	ORG2101-C1US-E-EVK	
ORG2101-2GGL-T-EVK	ORG2101-3GUS-T-EVK	ORG2101-C1JA-T-EVK	
ORG2101-2GGL-E-EVK	ORG2101-3GUS-E-EVK	ORG2101-C1JA-E-EVK	

Table 7: list of orderable parts

Please refer to document “ORG2101 Datasheet” for complete details of the devices.

APPENDIX A – PORTS AND POWER

Power input connector – J26. 6V/3A. or J22 USB. To determine which is the power supply to be used use jumper J24.

J22 is also used for serial debug console interface

USB 2.0 for input from external devices – J25.

MCU cold restart button – SW1

[Note: push the restart button for at list one second to allow GNSS module to fully restart]

Disconnect EVK sensors – Jumper J28

Auxiliary mezzanine connector - J5:

PAD	NAME	FUNCTION	DIRECTION
1	LPUART1_RX	UART Receive	Input
2	I2C1 SCL	I2C Serial Clock	Input
3	LPUART1_TX	UART Transmit	Output
4	I2C1 SDA	I2C Serial Data	Bi-directional
5	SWDIO	JTAG / Firmware Loading Pin	Bi-directional
6	SWCLK	JTAG / Firmware Loading Pin	Bi-directional
7	SPI1_MISO	SPI Data Out	Output
8	SPI1_SCK	SPI Serial Clock	Input
9	WAKEUP_2_PIN	Power Status	Output
10	SPI1_MOSI	SPI Data In	Input
11	WAKEUP_1_PIN	Power Status	Output
12	RST_MCU	Asynchronous Reset	Input
13	SW_UPDATE	Shutdown comm. module after SW update	Input
14	1PPS	UTC Time Mark	Output
15	GPIO_OUT_1	GPIO Output	Output
16	GPIO_OUT_2	GPIO Output	Output
17	GPIO_OUT_4	GPIO Out	Output
18	TIM1_CH1	Timer Function	Input
19	ADC1_IN	Analogue to Digital Input	Input
20	DAC1_OUT1	Digital to Analogue	Output
21	GPIO6	GPIO	Bi-directional
22	NC	Not Connected	
23	Do not use	Not Connected	
24	Do not use	Not Connected	
25	GPIO12	GPIO	Bi-directional
26	GPIO8	GPIO	Bi-directional
27	GPIO14	GPIO	Bi-directional
28	GPIO3	GPIO	Bi-directional

29	VSW	Battery Power Input	Power
30	VIN	EVK Power Supply	Power
31	VSW	Battery Power Input	Power
32	VIN	EVK Power Supply	Power
33	VCC3V3AUX	3.3V Power Supply	Power
34	VIN	EVK Power Supply	Power
35	VCC3V3AUX	3.3V Power Supply	Power
36	GND	System Ground	Power
37	GND	System Ground	Power
38	GND	System Ground	Power
39	GND	System Ground	Power
40	GND	System Ground	Power

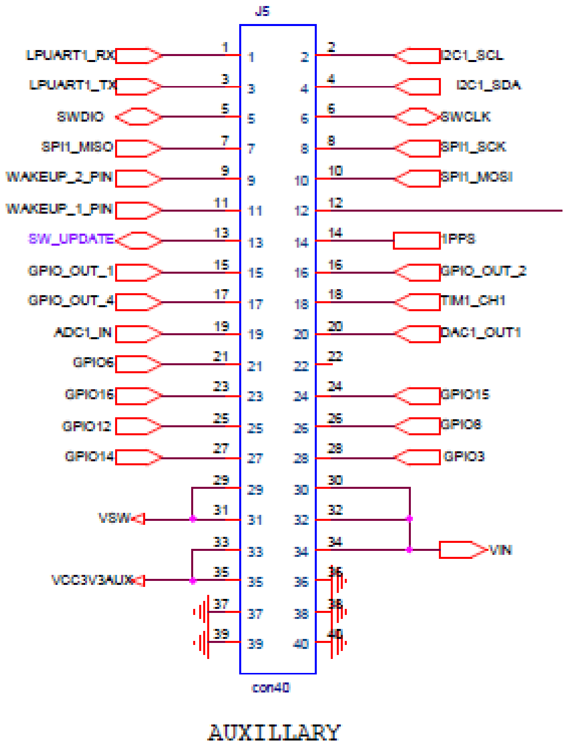


Figure 32 –J5 connector

APPENDIX B – FIRMWARE UPLOAD (FLASHING) FOR ORIGINIOT MODULE

OriginIoT module in the EVK is supplied with the company proprietary OriginSmart™ FW, designed to allow codeless configuration of the remote module without any FW changes.
If for any reason there is a need to upload firmware, this section explains how this is done.

Required equipment for FW upload:
- ST-Link/v2 ISOL STM8 & STM32 controller
- ST-LINK/V2 ISOL MCU STM8 & STM32 and STM32-ST-LINK Utility. This application can be downloaded here:
http://en.freedownloadmanager.org/Windows-PC/STM32-ST-Link-Utility-FREE.html

1. The plug supplied along with ST-LINK/V2 controller is suggested for FW upload. Connect the plug to the controller in orientation as shown in figure 33.

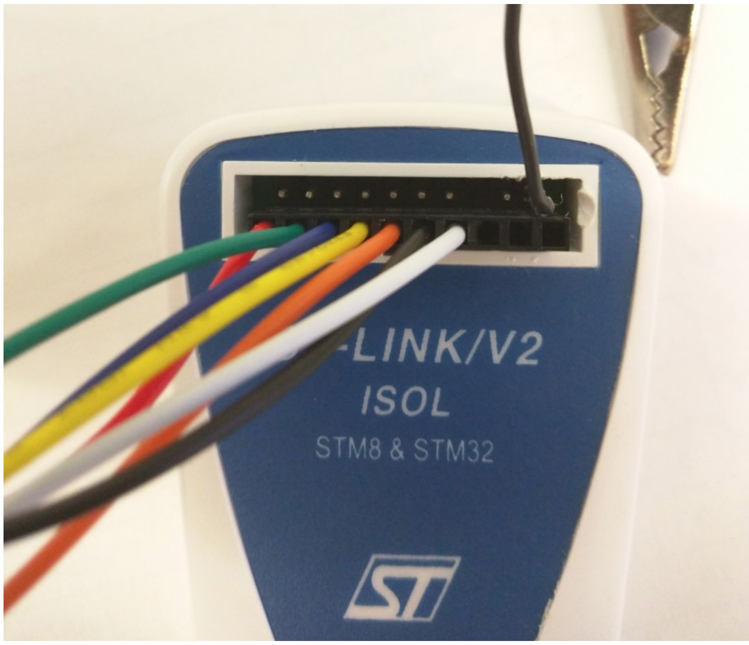


Figure 33: ST-Link plug orientation

2. Connect the wires to the J5 connector on EVK board in a method that described in table 8 and figures 34.

Wire Color	Function	Pin Number on J7 connector

Black	Ground	38
Red	Vcc	33
Orange	JTAG / Firmware Loading Plug	6
Yellow	JTAG / Firmware Loading Plug	5
Black – Separate wire on top right pin of controller	Ground	40

Table 8 – Wires Connection to EVK board

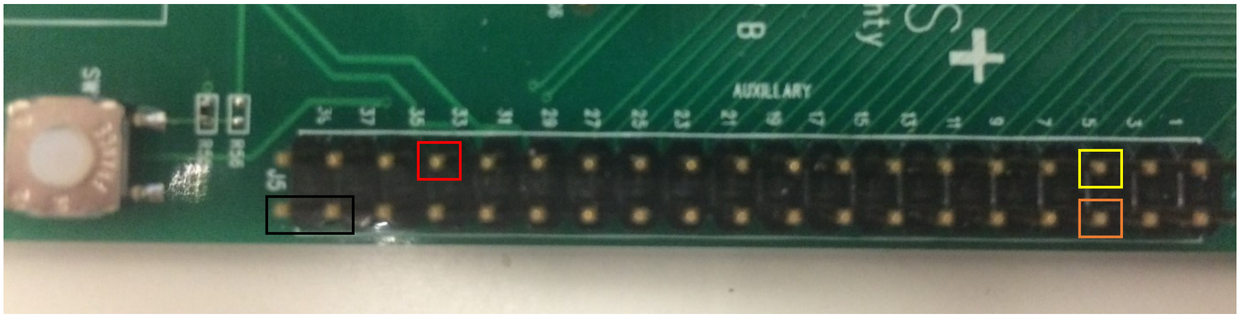
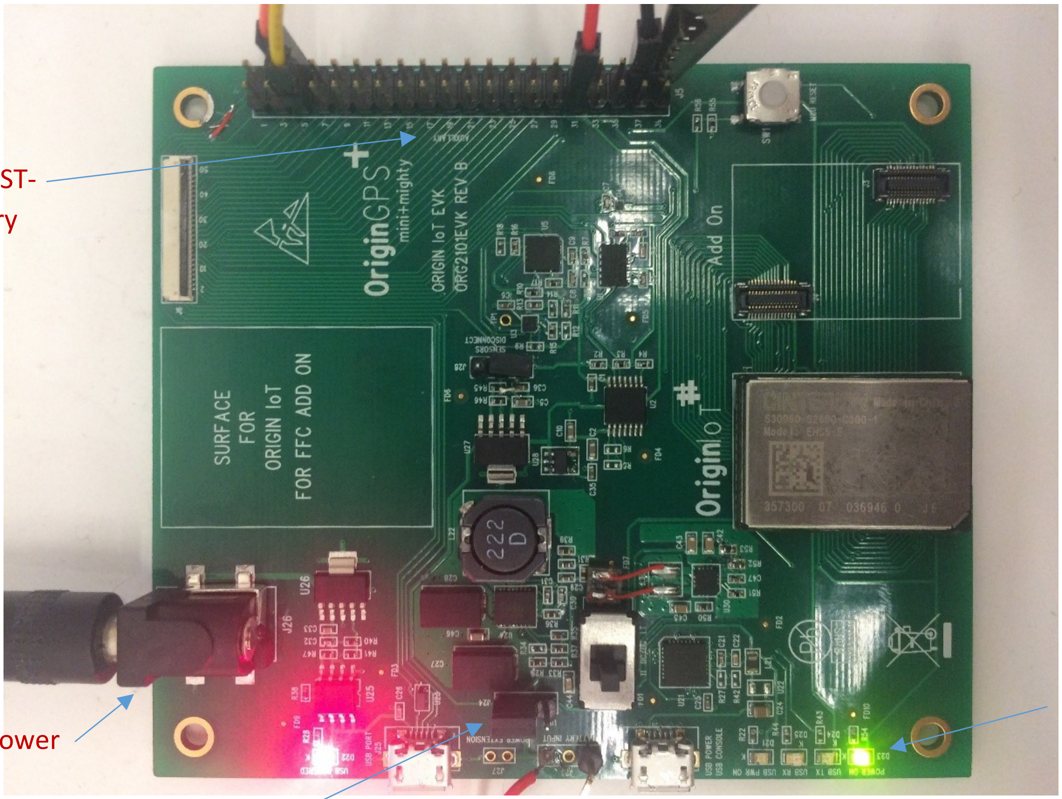


Figure 34: Wire Colors on J7 connector pins

All the other wires in the plug (green, blue, white) are not relevant and should not be connected.

3. Connect the ST-Link to PC via USB.
   Power on the EVK and the module – plug power connector J26 or USB connector J22, place jumper J24 accordingly.



Connection of ST-Link to Auxiliary connector J5

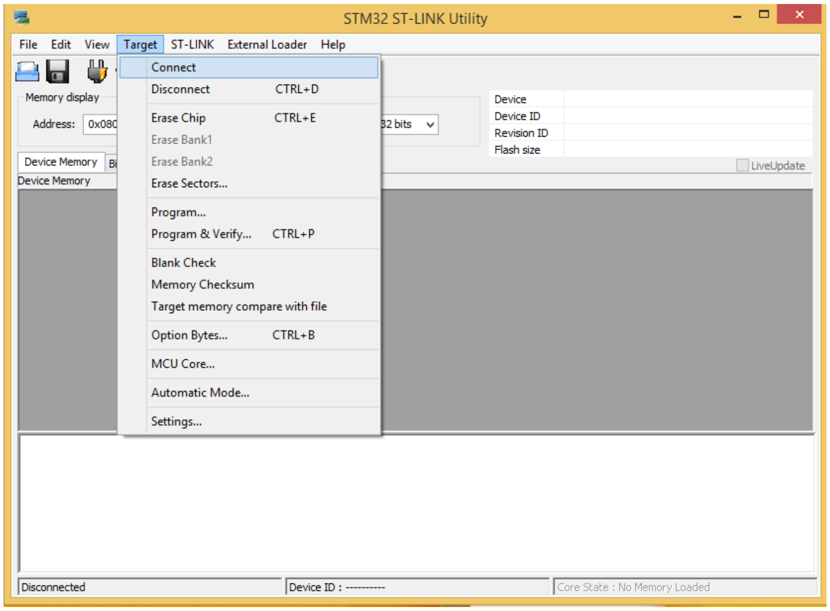
J26 EVK power

Led Indicating power OK to OriginIoT module

J24 Jumper choosing J26 as power supply to Origin IoT module

Figure 35: Power set up for FW upload

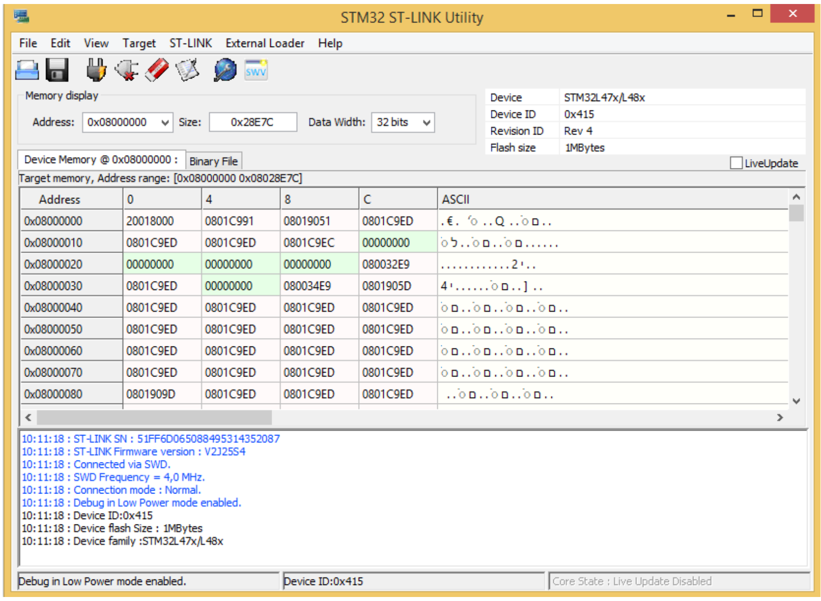
4. Open STM32 ST-LINK Utility.
   Press Target -> Connect.



Disconnected | Device ID : --------- | Core State : No Memory Loaded

Figure 36: STM32 ST-Link Utility

5. In case the connection is fine, you should see the following screen, with message: ”Connected via SWD”. See figure 37.
In case you get an error and cannot connect – please check your wires connection.



Memory display
Address: 0x08000000 Size: 0x28E7C Data Width: 32 bits

Device	STM32L47x/L48x
Device ID	0x415
Revision ID	Rev 4
Flash size	1MBytes

Device Memory @ 0x08000000 :

Target memory, Address range: [0x08000000 0x08028E7C]

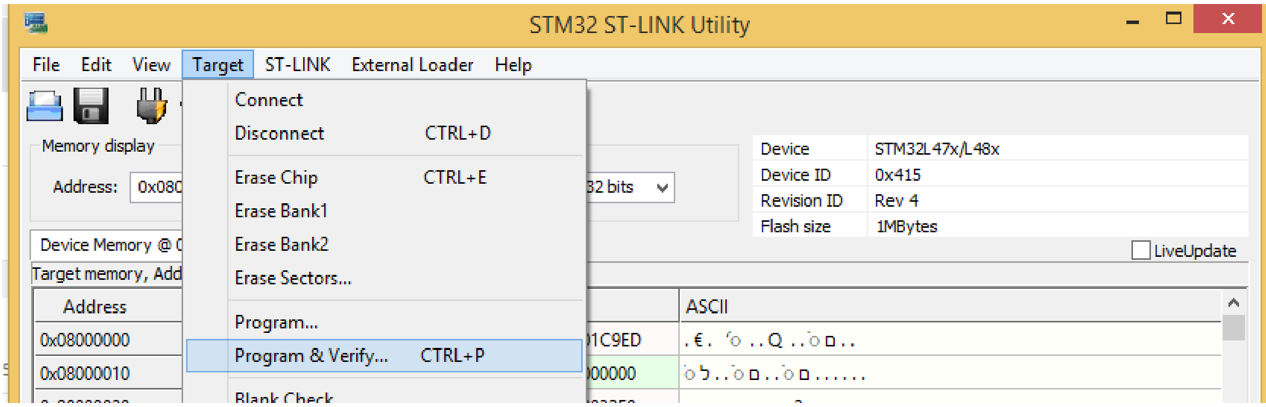
Address	0	4	8	C	ASCII
0x08000000	20018000	0801C991	08019051	0801C9ED	.€. ˙ó ..Q ..˙ó □..
0x08000010	0801C9ED	0801C9ED	0801C9EC	00000000	˙ó ל..˙ó □..˙ó □......
0x08000020	00000000	00000000	00000000	080032E9	............2'..
0x08000030	0801C9ED	00000000	080034E9	0801905D	4'......˙ó □..] ..
0x08000040	0801C9ED	0801C9ED	0801C9ED	0801C9ED	˙ó □..˙ó □..˙ó □..˙ó □..
0x08000050	0801C9ED	0801C9ED	0801C9ED	0801C9ED	˙ó □..˙ó □..˙ó □..˙ó □..
0x08000060	0801C9ED	0801C9ED	0801C9ED	0801C9ED	˙ó □..˙ó □..˙ó □..˙ó □..
0x08000070	0801C9ED	0801C9ED	0801C9ED	0801C9ED	˙ó □..˙ó □..˙ó □..˙ó □..
0x08000080	0801909D	0801C9ED	0801C9ED	0801C9ED	..˙ó □..˙ó □..˙ó □..

```
10:11:18 : ST-LINK SN : 51FF6D065088495314352087
10:11:18 : ST-LINK Firmware version : V2J25S4
10:11:18 : Connected via SWD.
10:11:18 : SWD Frequency = 4,0 MHz.
10:11:18 : Connection mode : Normal.
10:11:18 : Debug in Low Power mode enabled.
10:11:18 : Device ID:0x415
10:11:18 : Device flash Size : 1MBytes
10:11:18 : Device family :STM32L47x/L48x
```

Debug in Low Power mode enabled. | Device ID:0x415 | Core State : Live Update Disabled

Figure 37: Connecting to module in STM32 ST-Link Utility

6. Click on “Target->Program & Verify”. See figure 38.



Device	STM32L47x/L48x
Device ID	0x415
Revision ID	Rev 4
Flash size	1MBytes

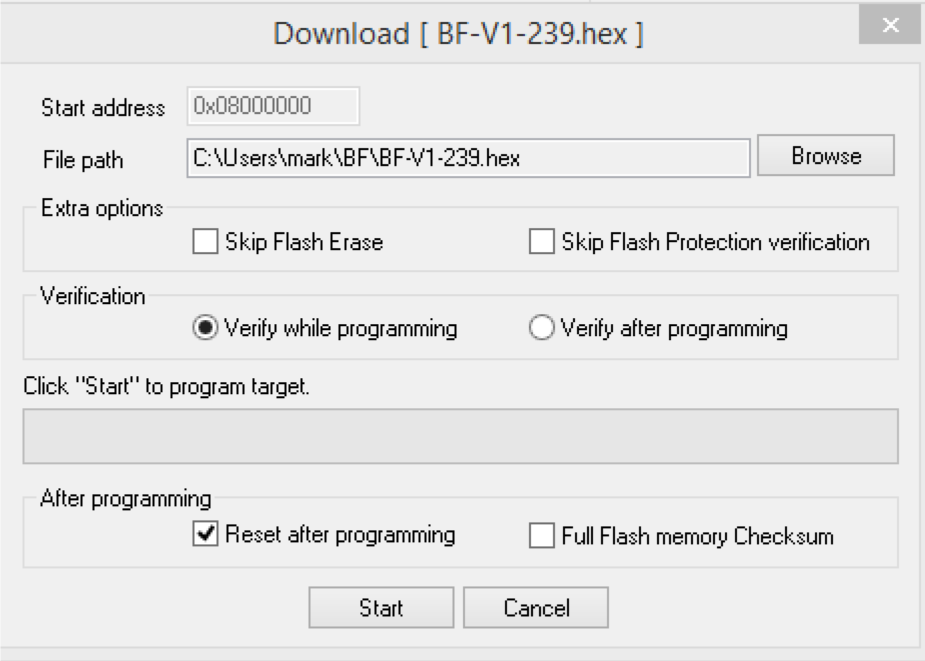
Address: 0x080
32 bits

Address		ASCII
0x08000000	01C9ED	.€. ό..Q..όם..
0x08000010	000000	όϿ..όם..όם......

Figure 38: Program & Verify in STM32 ST-Link Utility

Open file dialog will open. Select your firmware file. This should be a .hex /.bin file.

7. After selecting the file, you will see “download” window. See figure 39. Press Start.



Download [BF-V1-239.hex]

Start address: 0x08000000

File path: C:\Users\mark\BF\BF-V1-239.hex

Extra options
☐ Skip Flash Erase
☐ Skip Flash Protection verification

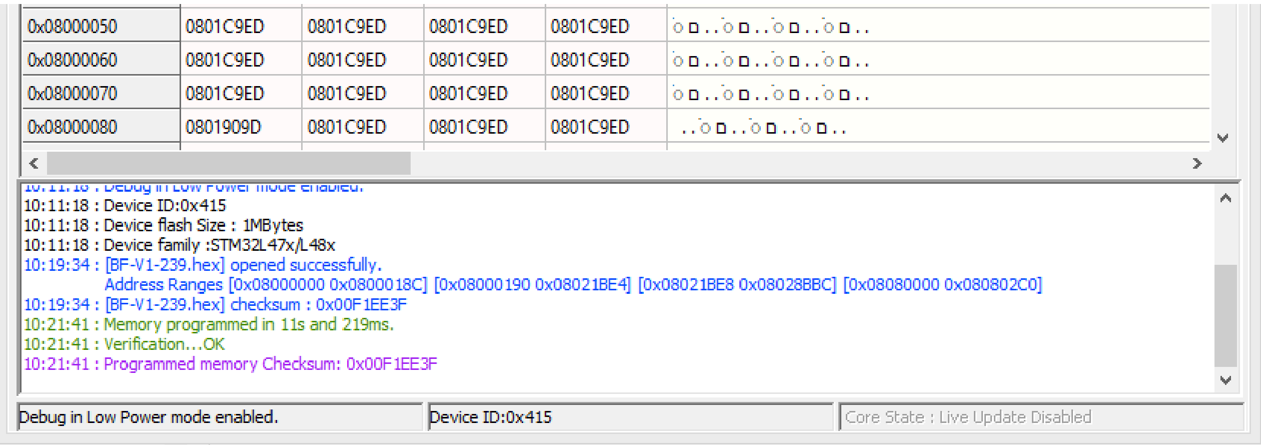
Verification
● Verify while programming
○ Verify after programming

Click "Start" to program target.

After programming
☑ Reset after programming
☐ Full Flash memory Checksum

Figure 39: Download window

8. After finishing the firmware upload, you should see “Verification…OK” message. See figure 40.



```
0x08000050   0801C9ED   0801C9ED   0801C9ED   0801C9ED   ȯ□..ȯ□..ȯ□..ȯ□..
0x08000060   0801C9ED   0801C9ED   0801C9ED   0801C9ED   ȯ□..ȯ□..ȯ□..ȯ□..
0x08000070   0801C9ED   0801C9ED   0801C9ED   0801C9ED   ȯ□..ȯ□..ȯ□..ȯ□..
0x08000080   0801909D   0801C9ED   0801C9ED   0801C9ED    ..ȯ□..ȯ□..ȯ□..
```

```
10:11:18 : Debug in Low Power mode enabled.
10:11:18 : Device ID:0x415
10:11:18 : Device flash Size : 1MBytes
10:11:18 : Device family :STM32L47x/L48x
10:19:34 : [BF-V1-239.hex] opened successfully.
           Address Ranges [0x08000000 0x0800018C] [0x08000190 0x08021BE4] [0x08021BE8 0x08028BBC] [0x08080000 0x080802C0]
10:19:34 : [BF-V1-239.hex] checksum : 0x00F1EE3F
10:21:41 : Memory programmed in 11s and 219ms.
10:21:41 : Verification...OK
10:21:41 : Programmed memory Checksum: 0x00F1EE3F
```

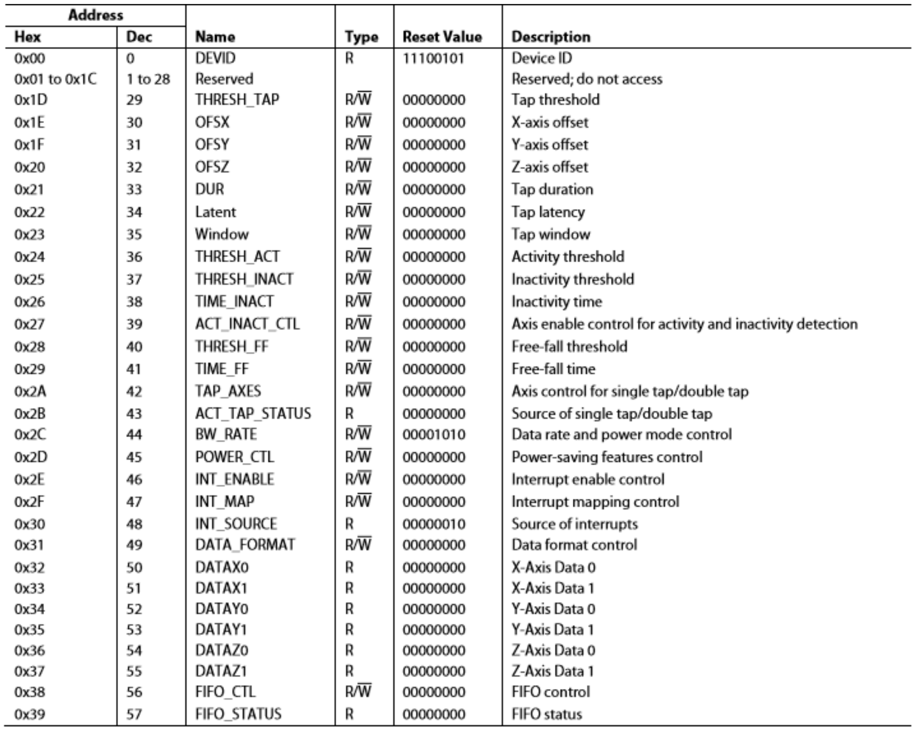
```
Debug in Low Power mode enabled.   Device ID:0x415   Core State : Live Update Disabled
```

Figure 40: Successful Firmware Download

9. Press “Target-> Disconnect” to disconnect the module.

APPENDIX C –REGISTER MAPS FOR SENSORS

Accelerometer -



Address					
Hex	Dec	Name	Type	Reset Value	Description
0x00	0	DEVID	R	11100101	Device ID
0x01 to 0x1C	1 to 28	Reserved			Reserved; do not access
0x1D	29	THRESH_TAP	R/W	00000000	Tap threshold
0x1E	30	OFSX	R/W	00000000	X-axis offset
0x1F	31	OFSY	R/W	00000000	Y-axis offset
0x20	32	OFSZ	R/W	00000000	Z-axis offset
0x21	33	DUR	R/W	00000000	Tap duration
0x22	34	Latent	R/W	00000000	Tap latency
0x23	35	Window	R/W	00000000	Tap window
0x24	36	THRESH_ACT	R/W	00000000	Activity threshold
0x25	37	THRESH_INACT	R/W	00000000	Inactivity threshold
0x26	38	TIME_INACT	R/W	00000000	Inactivity time
0x27	39	ACT_INACT_CTL	R/W	00000000	Axis enable control for activity and inactivity detection
0x28	40	THRESH_FF	R/W	00000000	Free-fall threshold
0x29	41	TIME_FF	R/W	00000000	Free-fall time
0x2A	42	TAP_AXES	R/W	00000000	Axis control for single tap/double tap
0x2B	43	ACT_TAP_STATUS	R	00000000	Source of single tap/double tap
0x2C	44	BW_RATE	R/W	00001010	Data rate and power mode control
0x2D	45	POWER_CTL	R/W	00000000	Power-saving features control
0x2E	46	INT_ENABLE	R/W	00000000	Interrupt enable control
0x2F	47	INT_MAP	R/W	00000000	Interrupt mapping control
0x30	48	INT_SOURCE	R	00000010	Source of interrupts
0x31	49	DATA_FORMAT	R/W	00000000	Data format control
0x32	50	DATAX0	R	00000000	X-Axis Data 0
0x33	51	DATAX1	R	00000000	X-Axis Data 1
0x34	52	DATAY0	R	00000000	Y-Axis Data 0
0x35	53	DATAY1	R	00000000	Y-Axis Data 1
0x36	54	DATAZ0	R	00000000	Z-Axis Data 0
0x37	55	DATAZ1	R	00000000	Z-Axis Data 1
0x38	56	FIFO_CTL	R/W	00000000	FIFO control
0x39	57	FIFO_STATUS	R	00000000	FIFO status

TABLE 9– Accelerometer Register Map

eCompass -

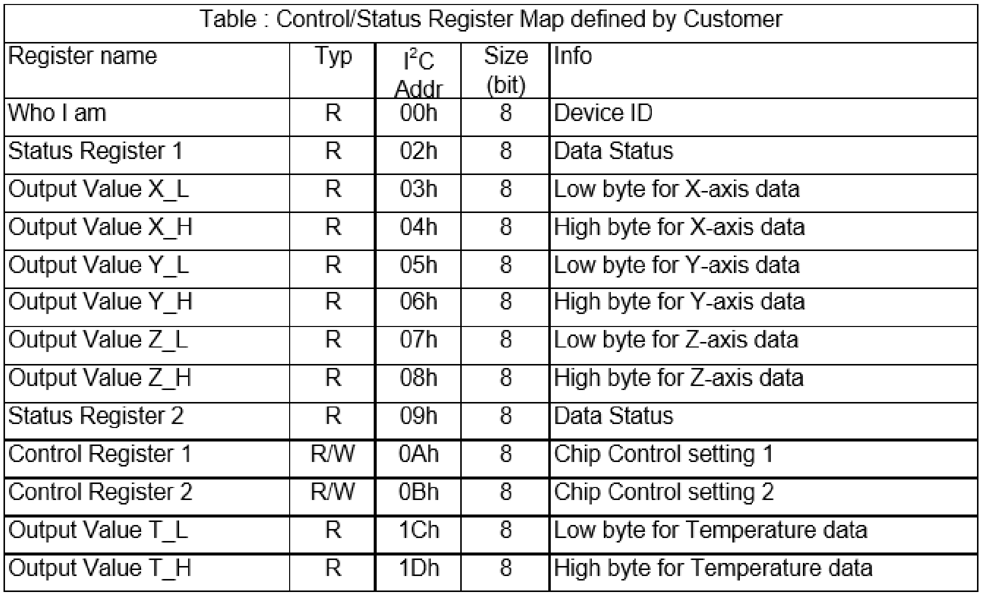
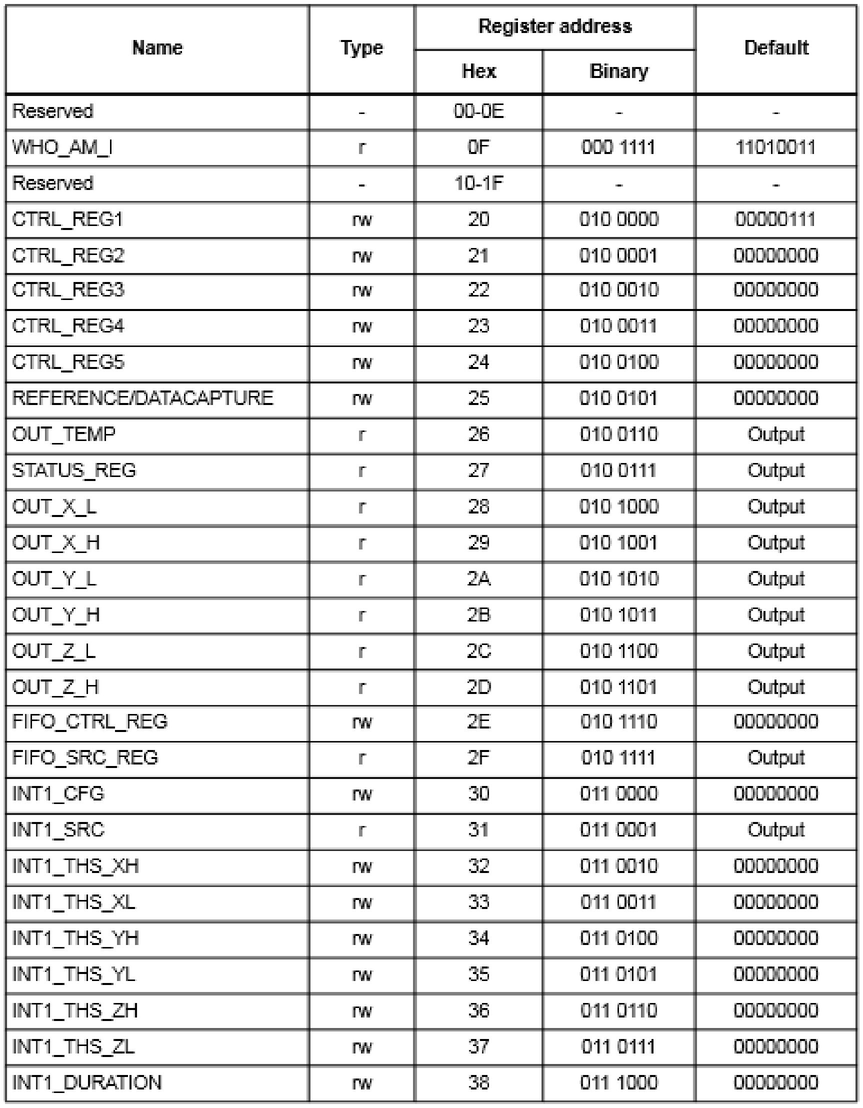


Table : Control/Status Register Map defined by Customer				
Register name	Typ	I^{2}C Addr	Size (bit)	Info
Who I am	R	00h	8	Device ID
Status Register 1	R	02h	8	Data Status
Output Value X_L	R	03h	8	Low byte for X-axis data
Output Value X_H	R	04h	8	High byte for X-axis data
Output Value Y_L	R	05h	8	Low byte for Y-axis data
Output Value Y_H	R	06h	8	High byte for Y-axis data
Output Value Z_L	R	07h	8	Low byte for Z-axis data
Output Value Z_H	R	08h	8	High byte for Z-axis data
Status Register 2	R	09h	8	Data Status
Control Register 1	R/W	0Ah	8	Chip Control setting 1
Control Register 2	R/W	0Bh	8	Chip Control setting 2
Output Value T_L	R	1Ch	8	Low byte for Temperature data
Output Value T_H	R	1Dh	8	High byte for Temperature data

TABLE 10– eCompass Register Map

Gyroscope



Name	Type	Register address		Default
		Hex	Binary	
Reserved	-	00-0E	-	-
WHO_AM_I	r	0F	000 1111	11010011
Reserved	-	10-1F	-	-
CTRL_REG1	rw	20	010 0000	00000111
CTRL_REG2	rw	21	010 0001	00000000
CTRL_REG3	rw	22	010 0010	00000000
CTRL_REG4	rw	23	010 0011	00000000
CTRL_REG5	rw	24	010 0100	00000000
REFERENCE/DATACAPTURE	rw	25	010 0101	00000000
OUT_TEMP	r	26	010 0110	Output
STATUS_REG	r	27	010 0111	Output
OUT_X_L	r	28	010 1000	Output
OUT_X_H	r	29	010 1001	Output
OUT_Y_L	r	2A	010 1010	Output
OUT_Y_H	r	2B	010 1011	Output
OUT_Z_L	r	2C	010 1100	Output
OUT_Z_H	r	2D	010 1101	Output
FIFO_CTRL_REG	rw	2E	010 1110	00000000
FIFO_SRC_REG	r	2F	010 1111	Output
INT1_CFG	rw	30	011 0000	00000000
INT1_SRC	r	31	011 0001	Output
INT1_THS_XH	rw	32	011 0010	00000000
INT1_THS_XL	rw	33	011 0011	00000000
INT1_THS_YH	rw	34	011 0100	00000000
INT1_THS_YL	rw	35	011 0101	00000000
INT1_THS_ZH	rw	36	011 0110	00000000
INT1_THS_ZL	rw	37	011 0111	00000000
INT1_DURATION	rw	38	011 1000	00000000

TABLE 11– Gyroscope Register Map

APPENDIX D – WORK WITH ORG2101 ADD-ONS

ORG2101 EVK allows developers to test their add-ons (or add-ons purchased directly from OriginGPS) in two manners:

1. Dedicated connectors on EVK - J3 and J4
2. Flat Flexible Cable (FFC) connector – J6

To use this EVK functionality, additional required equipment is:

1. Add-on developed according to guidelines in ORG2101 Datasheet document, or purchased directly from OriginGPS (For FCC option it shall have FFC connector compatible with Molex 150150451)
2. Flat Cable connector Molex 150150451 or compatible (only for FFC connection option)
3. Battery compatible with the add-on used (only for FFC connection option)

In this appendix we will demonstrate using ORG2101-AD01 developed by OriginGPS as implementation example.

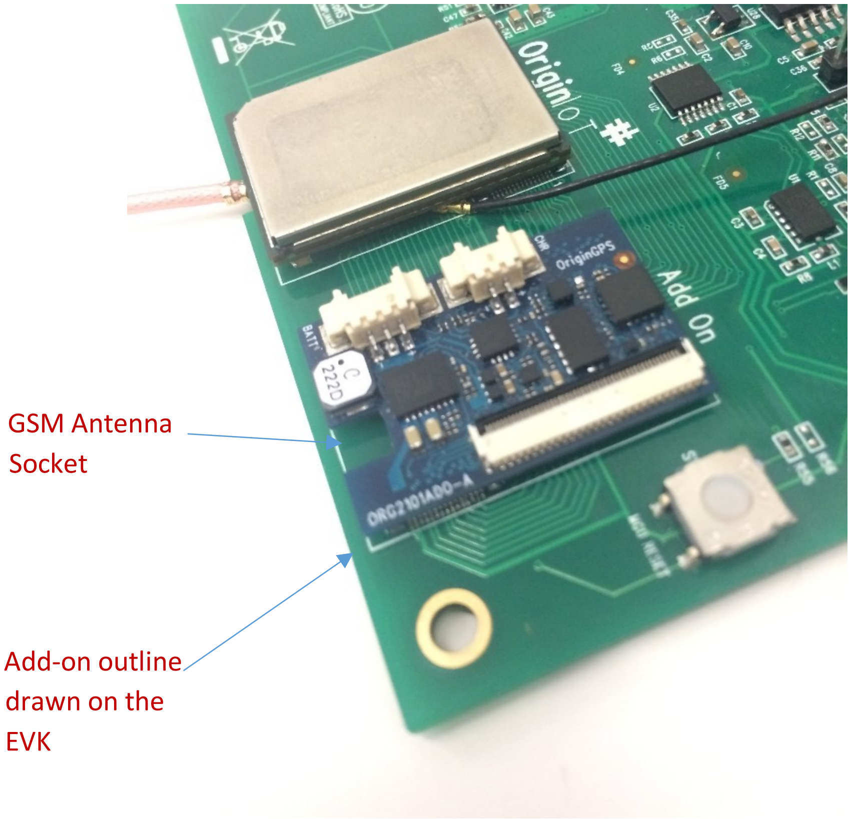
D.1 Using Add-on via connectors J3 and J4

In this mode of operation, one can test the functionality of the add-on together with ORG2101 module without connecting the add-on to external battery. The addon is powered by the power supply circuits in the EVK.

Note: In this mode you will be able to use both the sensors on the EVK and the also the sensors on the add-on. Make sure that sensors are not occupying the same I2C address as the sensors on the EVK (see table 3)

Please notice: connection of the add-on to EVK is unidirectional – J1 of the add-on (closer to GSM/LTE antenna socket) should connect to J3 on the EVK.

You know that you are connected correctly if add-on fits in white outlines drawn on the EVK (silk screen)



GSM Antenna Socket

Add-on outline drawn on the EVK

Figure 41 – mating Add-on to EVK correctly

D.2 Using Add-on via FFC connector J6

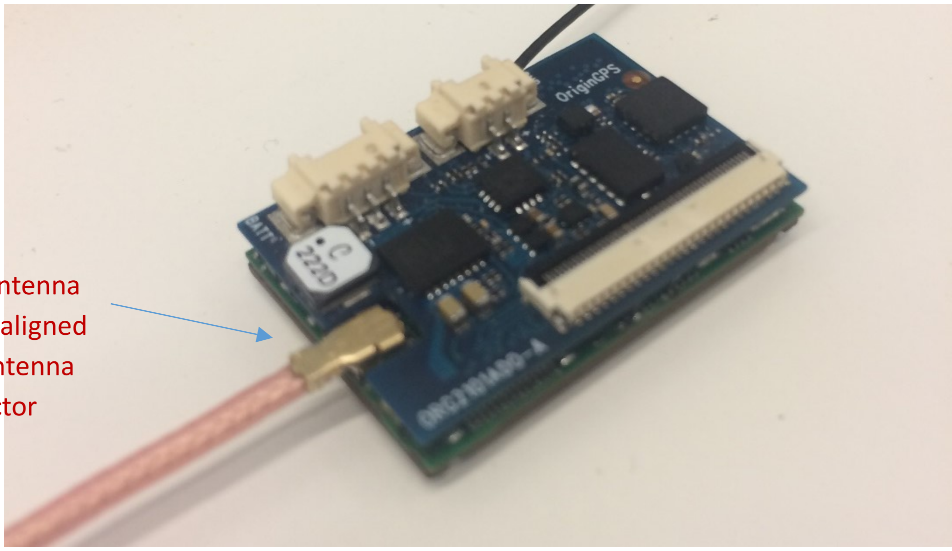
Using add-on via FFC connector is the closest simulation for the real-life functionality of the add-on in a product. In this mode the addon and the ORG2101 module are powered by a battery (and not from the EVK). The FFC cable allows debugging via the serial console as described in section 9.2.

To use this mode your add-on should include FFC connector which is compatible with Molex FFC 150150451. Note: the flat cable is not included in the EVK.

The user can choose if he wants to operate the sensors on the EVK using Jumper J28. Disconnecting the Jumper cuts power supply to the sensors on the EVK and allowing the user to work solely with the sensors on the Add-on.

To connect Add-on and ORG2101 module through connector J6:

1. Prepare ORG2101 - Insert nano SIM card to its slot, Connect GSM/LTE antenna to the U.Fl connector OriginIoT, Connect GNSS antenna to the W.Fl connector on OriginIoT, See figure 1 (section 9.2).
2. Mate ORG2101 with Add-on – connect add-on connectors J1 and J2 to ORG2101 connectors J1 and J2 as shown in figure 42.
   Note – mating ORG2101 with Add-on is unidirectional. Make sure that antenna socket is aligned with connector J4.



GSM Antenna socket aligned with antenna connector

Figure 42 – mating Add-on to ORG2101 correctly

3. Connect assembly to EVK with FFC – place FFC cable inside the FFC connector (J6 in our example), see figure 43a.
4. Close lever of FFC connector as shown in figure 43b.

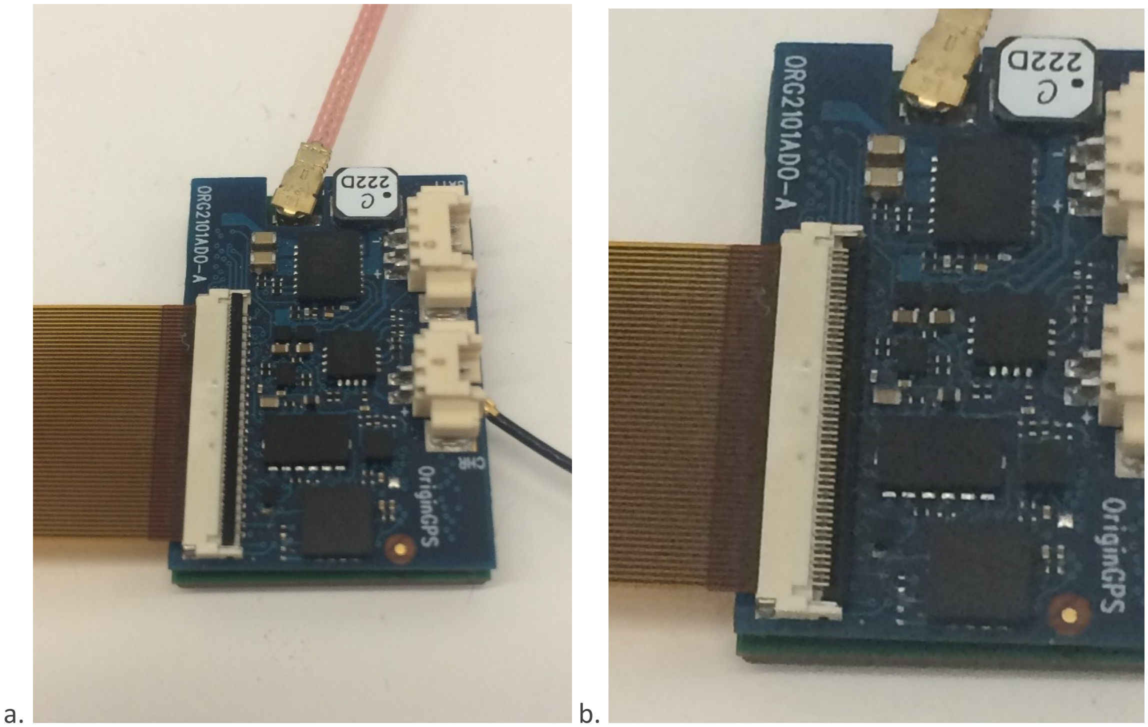


Figure 43 – attaching FFC cable to add-on

5. In the same manner, connect FFC to FFC connector on the EVK (J6), see figure 44. Pay close attention to the direction of the FCC.
   Note: repeated insertion of FFC, closing and open of actuator can be harmful to the performance of connector J6.
6. Supply power to EVK by connecting USB cable from PC to J22 and placing Jumper J24 in USB position, see figure 44.

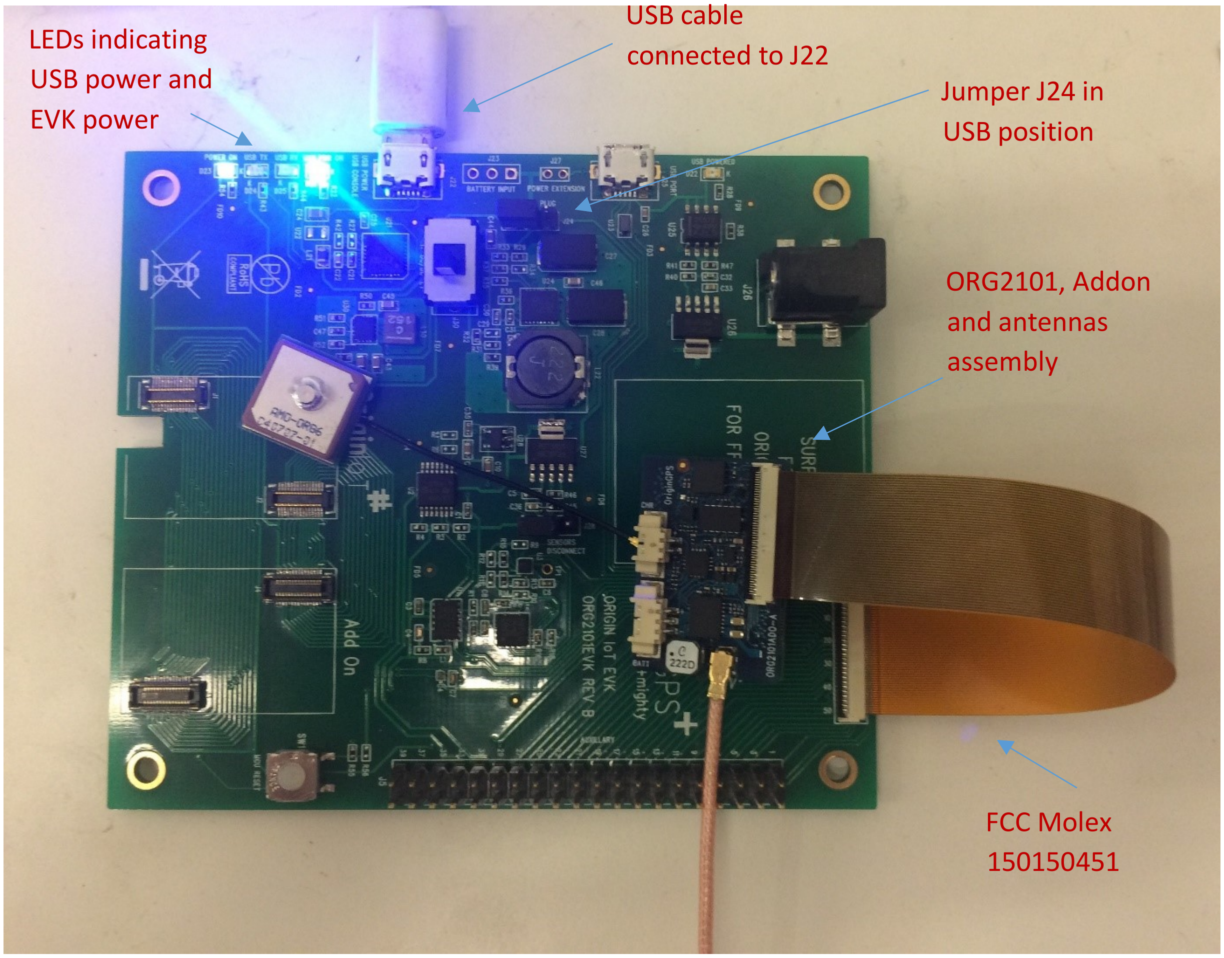

LEDs indicating USB power and EVK power
USB cable connected to J22
Jumper J24 in USB position
ORG2101, Addon and antennas assembly
FCC Molex 150150451

Figure 43 – attaching FFC cable to EVK

7. Open terminal on your PC and connect to corresponding COM port of the OriginIoT with a baud rate of 57600 bps.

In Tera Term – go to Setup - > Serial Port (See figure 3a, section 9.2), select the COM port and set baud rate to 460800 bps. Press "OK". See figure 3b.

Full communication parameters:

- Baud Rate: 460800
- Data Bits: 8
- Stop Bits: 1
- Flow Control: None
- Parity: None

8. Supply power to OriginIoT via the addon by connecting the battery.

After supplying power to OriginIoT you should see a welcome message and GSM/LTE start-up messages.

9. While connected, press Enter key in the terminal.

You should see the command line "ORG>" in the terminal window. See Figure 3c, section 9.2.

Now the module can be accessed via the debug console. Please refer to document "OriginIoT debug console user guide" for the complete list of functions available via the debug console.

Note: you can choose to operate or disconnect the sensors on the EVK using Jumper J28.