User Manual

TBSL1 and Console

Version 04.00.01.38

January 2024

Table of Contents

1. Introduction

The TBSL1 from Tekbox is a high capacity data logger with integrated telemetry. It is fitted with an SDI-12 input channel as well as an analogue and pulse input. The TBSL1 may be fitted with a communications board for either LoRa WAN (TBSL1-LO) mobile phone/cellular (TBSL1-4G) or WiFi (TBSL1-WF) communications. The unit will automatically switch between 4G and LoRa modes, but the WiFi option requires installation of WiFi specific firmware.

The TBSL1 GUI is used to program the TBSL1 via a serial connection. It can be used to set all the board's operational parameters (time settings, SDI-12 commands, LoRaWan settings) and send commands to SDI-12 sensors or the LoRaWAN module on the board and to display the received responses.

1.1. Hardware setup

The picture below shows the designations of the various connectors on the TBSL1.

The TBSL1 connects to a PC using a Type B USB connection. In addition to the connector on the case, there is a second connector on the PCB.

Using a Number 1 Philips Screwdriver, undo the 4 screws holding the cover to the TBSL1 case and remove the cover.

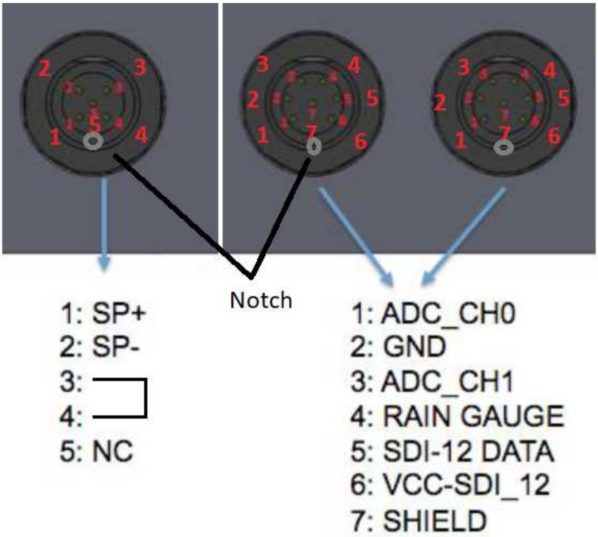
Before fitting the battery, make sure that the antenna is screwed on to the TNC connector on top of the case and that the UFL cable from the antenna is fitted to the upper most socket on the LoRaWAN PCB.

For the standard TBSL1 the battery is a 3.6V Lithium-Ion cell with internal thermistor and 3 pin JST connector. The battery is secured to a plastic holder which sits on 4 standoffs installed on top of the PCB. Connect the battery lead to the matching socket on the upper left of the TBSL1 PCB.

If your TBSL1 is going to be powered via a solar cell, wire the solar panel up according to the diagram below (Port 3). If the solar panel has been supplied with your unit it will be pre-wired with the required connector. In most applications a 6V 2W solar panel will operate the TBSL1 although a 5W unit can be used if you have concerns about the power budget or have sensors which draw high current or if you are using the cellular version with 15 minute logging/transfer.

CAUTION: Do not use the TBSL1 with 12V solar panels. The maximum permissible open circuit panel voltage is 7.2V. Using a panel with a higher voltage rating will damage the TBSL1.

The image below shows the input connectors when viewed from the bottom or the TBSL1. The orientation is the same as this picture when looking down on the Plugs i.e. if you put the 7 pin plug in to the connector, do up the lock nut and then remove the back-shell from the connector, when you look down on the solder buckets they will be in the same order.



The two input connectors are wired in parallel. This allows you to easily connect two sensors. Should you need to connect more than two cables, you can add a TBS04 Junction Box.

The back of the TBSL1 case is fitted with a mounting bracket. This can be used to secure the unit to a 40 to 50mm post using a stainless steel hose clamp.

The colour code for the cables supplied with the TBSL1 is as follows:

Pin	**Solar (5 pin)**	**Sensor (7 pin)**
1	Blue – SP +	Analogue 1
2	Brown – SP -	Brown - GND
3	Link to Pin 4	Analogue 2
4	Link to Pin 3	Pulse
5	-	Green/Yellow - SDI-12
6	-	Blue - PWR
7	-	Not used

If you are using either of both of the Analogue inputs or the Pulse Input, you may need to use a 6 or 7 core cable. Or you can re-wire one socket to carry the analogue/pulse inputs and the other the SDI-12.

When using the TBSL1 with a solar panel, mount the solar panel facing North (south in the Northern Hemisphere) and the fit TBSL1 on the opposite side of the pole, facing south. This way the solar panel helps shade the TBSL1 from the sun. If you are fitting the TBSL1 with an external antenna, you can mount the antenna as high as possible on the post and the TBSL1 unit directly below the solar panel.

Radio range is governed by a number of factors including terrain. Radio range can be maximised by:

- raising the height of the Gateway antenna
- raising the height of the TBSL1 antenna.

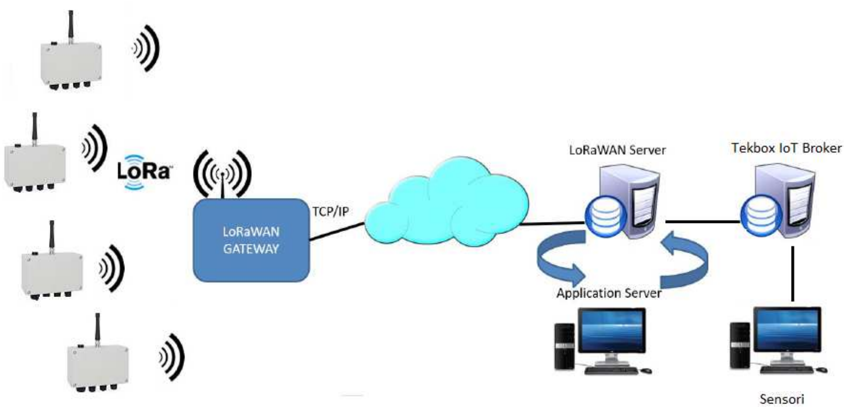
Often a compromise must be sought between raising the antenna height for best coverage versus keeping it low to allow machinery to pass over the site. If regular machinery movement is an issue, consider using a tilt down base pole.

The following factors will cause a decrease in range:

- hills between the TBSL1 and the Gateway
- tree canopy between the TBSL1 and the Gateway, particularly trees with fine needle leaves which act can absorb signal when they are wet
- metal objects placed close to the TBSL1 antenna.

1.2. TBSL1 Data Path

The path of data from a LoRa WAN Node to the Presentation software (the default is to use Sensori) is shown in the image below.



The readings from the sensors connected to the TBSL1 are saved in the unit's memory. At the programmed Send Interval (e.g. 15 min) the readings are transmitted via LoRa WAN to a radio base station – which in a LoRa WAN environment is called a Gateway.

The Gateway sends all the received messages (or packets) to a LoRa WAN Server.

A LoRa WAN Server will contain one or more “Applications” which belong to different users. Each Application will in turn contain a number of “Devices” (telemetry units or RTUs). Each Device is has a unique identifier.

When devices are created on the LoRa WAN Server, they are allocated a set of encryption keys: one is used to signal that a device has a legitimate right to use the network, the other is used to encrypt the data.

When the LoRa WAN Server receives a message, it checks to see if it recognises the Devices unique identifier (its EUI). If not, it rejects the message. If it does, it unencrypts the message and saves the contents in its database.

Over time the LoRa WAN Server will accumulate sets of readings for each Device. These can then be collected by the programs used to display and interpret the readings

- the servers typically offer two or more ways to access this data
- the first is using a proprietary URL based XML format
- the second is by using an open protocol called MQTT.

One of the complications of LoRa WAN is that the LoRa WAN Server does not know anything about the sending device and hence what type of sensors(s) are connected. So if you are working with devices whose configuration can vary, you must manually unpick the contents of each data packet based on information about the values it contains.

This can get very complex if large numbers of devices are in use or if the configuration changes.

To deal with this complexity, Tekbox offers an IoT Broker which does this work for you.

The Broker is configured with the details of each device including the LoRa WAN server to which the device belongs and the sensors attached to it. When packets are received, the Broker then unpicks them (parses the packets). The readings are then saved along with any there relevant information, such as the units, the type of sensor etc.

The biggest benefit of this approach is that it scales easily to hundreds or thousands of devices. It also means that the presentation software can be given other relevant information about each device: such as its type, its GPS coordinates etc. Similarly it gets the “meta-data” for each sensor.

In the Cellular and WiFi units, the data is sent to the Tekbox IoT Broker using MQTT, which in turn passes it to Sensori. If you do not wish to pass the data to Sensori, you can configure the units to send the data to any other presentation package however you will need to add your own packet decoder.

2. Software Installation

2.1. Serial Port Drivers and Windows 10

The TBSL1 uses a serial port chip manufactured by ST Microelectronics. Windows 8 and earlier versions employed a serial port driver provided by STM. But the STM driver is incompatible with Windows 10

- if you use the STM driver when you put the tBSL1 in to Logging mode the USB port will toggle on and off continually and you will not be able to re-connect.

If you have an STM driver on your PC, delete it before installing or updating the TBSL1 Console. The easiest way to do this is to connect the TSBL1 to your PC and open Device Manager. Select the COM port allocated to the device and open its Properties. Then select Disable Device and when prompted select Delete Driver. Disconnect the TBSL1, wait a couple of minutes then re-connect it. Windows will load its own generic driver for the COM port.

2.2. New Installation

The Console GUI is loaded using an MSI Installer. If you are upgrading from an older version, follow the instructions in section 7.2 . For a new installation:

- double click on the installer TBSLnConfigurationInstaller.msi
- when prompted, select install for “Everyone”
- when prompted use the default installation path

  - this will install the program to C:/Program Files (X86)/TekBox Digital Solutions Viet Nam/TBSL1 Configuration Tool/TBSL1-04.00.0a.bb
  - the location for configuration files, log files and downloaded measurements will be set to: C:/Users/Public/Public Documents/TBSL1 Configuration Tool/TBSL1-04.00.0a.bb

Once you have completed the installation, open the GUI and edit the default configuration files so that they represent your typical setup:

- double click on the Icon for the TBSL1 Console to open it
- when the program opens it will load the configuration stored in the file C:/Users/Public/Public Documents/TBSL1 Configuration Tool/TBSL1-04.00.0a.bb/JsonFile/Settings.json
- from the menu select Settings/TBSL1 Configuration
- go through the settings and set the values to suit your region, channel preferences, join type etc
- save the settings with File Save
- write the settings to the Default config with File / Save As / Default Settings.Json.

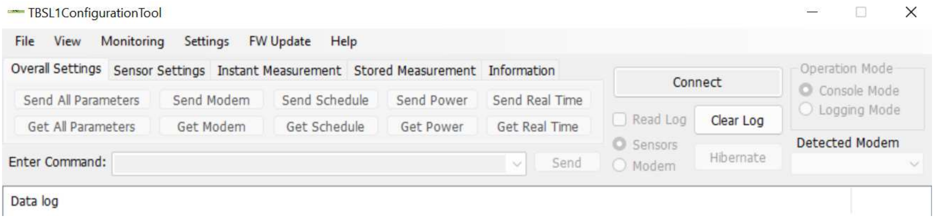
The installer will create a shortcut on the Desktop. To ensure that you can create and edit configuration files, edit the Shortcut and set the properties to “Run as administrator”.

The TBSL1 GUI program and TBSL1 Firmware must be maintained at the same version

- the GUI software checks the TBSL1 firmware version whenever it connects
- if the versions do not match, an error will display on the screen

3. Using the TBSL1 and GUI

To open the GUI application, run the program from the start menu or by clicking on the file TBSLnConfigurationTool.exe from the directory where the application files are located, or by clicking on the desktop shortcut you created during installation.



3.1. *Install Battery*

Using a Philips Head screw driver, loosen the 4 screws that hold the lid in place and remove it from the case.

The battery has a 3 pin connector in it. Plug this in to the socket labelled CON5 on the top left of the printed circuit board (PCB).

The battery should then be secured in to place on the top of the plastic battery holder using a cable tie.

3.2. *Connecting to the TBSL1*

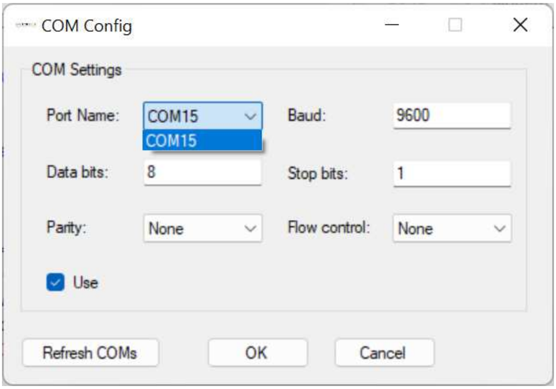
In order to configure the TBSL1, the unit must have the Solar Panel connected. This is because the start link (between Pins 3 & 4) on the socket is used to activate the unit. If you are regularly setting up TBSL1 units you can use a spare 5 pin plug to make up a dummy plug: simply solder a link between pins 3 and 4 on the back of the plug. Note that you can also solder wires to pins 1 (+) and 2 (-) to allow you to charge the battery from a 6V plug-pack or bench power supply.

If the lid of the TBSL1 case is open, the blue LED in the middle of the PCB should flash every half second.

To work with the TBSL1, you must connect to the unit using the COM port allocated to the device by your PC when the unit is plugged in. Use the Windows Device Manager to identify the port.

To select the Port:

- From the menu select ***Settings / Com Port Configuration***
- When the communications dialogue box opens, click on COM Port and select the port allocated to the device form the drop down list box



- If the port has not displayed, click on REFRESH COMMS to have the program re-scan for the available ports
- Leave all the other settings on the default values

Once you have selected the port, the GUI will attempt to communicate with the TBSL1. The result of the communications between the GUI and the device are shown in the Data Log section of the screen

- Transmitted commands display in red >>
- Responses from the TBSL1 are shown in blue <<
- If you want to see more detailed information about the log information, open the log file from the folder :

C:\Users\<User name>\AppData\Local\TBSL1\Logs

3.3. Create and Save a Configuration File for the Device

The TBSL1 GUI will open with a default configuration, which is read from the file:

> C:\Users\Public\Public Documents\TBSL1 Configuration Tool\JsonFile\defaultsettings.json.

To enable you to easily recall and modify configurations for specific TBSL1 units, you should save a configuration file for each device you install:

- To create a new Configuration file select ***File / Save As*** and then give the file a suitable name e.g. TBSL1_SN_13423 or TBSL1_Jones_Block_34
- You may like to include the Serial number in the name. To identify the Serial Number of the TBSL1, from the menu select ***Settings / TBSL1 Configuration*** and then click on the INFO tab
    - the serial number is shown in the Board ID field.

Once you have saved configurations, you can re-load them at any time. In this fashion you can create one device configuration file and then use it as a template for future devices. You can also save a file for each of the standard configurations you use (for instance your standard soil moisture probe site or standard weather station).

3.4. Changing the Configuration Defaults

The default configuration file has settings compatible with those used in Asia. To save you from having to change all the settings each time you start a fresh RTU, you can change the default settings to suit your preferences. From time to time Tekbox may add parameters to the configuration file or change the way parameters are stored. This may cause the GUI to raise an error when it is saving a configuration file. When such changes are made Tekbox will update the default Configuration file. You will then need to update the default file again:

- from the menu select File / Load Default Settings

- once the new configuration file opens, select Modem / LoRa WAN
- you should change the key settings to suit the Channel plan you use
    - Select Version 1.03
    - Mode ABP (preferred over OTAA)
    - Duty Cycle, ADR State OFF (unchecked)
    - Wait ACK ON Retry times 3
    - RX Window Delay RX1 1000, RX2 2000
    - Freq Scheme AU915
    - SET CH: select and set the frequencies for AU 915 Sub Band 1 i.e. 916.8, 917.0… 918.2 (steps of 0.2)
    - RXWIN 2 Freq 923.3 DR 8
    - Tx Power 14 Port 8
    - you can also update the Interval and SDI-12 settings
- now use File / Save As and overwrite the file “DefaultSettings.json”
- then repeat, this time overwriting "Settings.json"
- the next time you need a new configuration file it will already have the key values pre-set.

You should then re-create the configuration files for your devices using the “DefaultSettings.json” as the template.

3.5. Changing Operating Mode

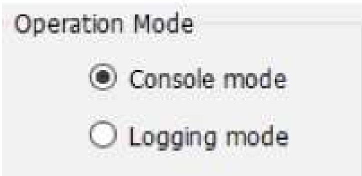
The TBSL1 can at any time be in one of two modes: **Console** and **Logging**:

- Console Mode: this mode is the default and is used to configure and test the TBSL1
    - In console mode, the board doesn’t perform any measurement/transmission operations. It only receives the

    commands from the application, executes the commands on sensor/LoRa module and send responses to the application.

- Logging Mode: units must be set back to this mode to enable logging
    - This is the normal mode of operation for devices in the field
    - While in logging mode the GUI displays the results of commands as they are executed on the TBSL1.

When powered-up, the board will send a “PING” command to the PC and wait for a response from the application. If the application has started and the COM port is open, it should answer with an “OK” response, after which the date and time are automatically programmed into the unit and the board will switch to “C***onsole***” mode. When the TBSL1 first starts, it will stay in **“Console”** mode. In this mode, the board will wait for commands to be sent – it will not read the sensors or transmit the readings. Any settings sent from the application at this time will only be saved in the board memory: they will not be activated.



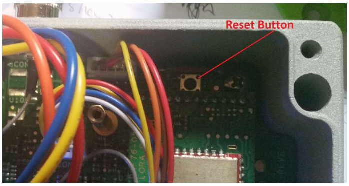
When you click on **“Logging mode”** button, the application will send a command to the board, telling it to switch to **“Logging”** mode. Once in **“Logging”** mode, the board will activate the new settings. After that, the board will go to sleep – and will wake up according to its logging and transmission settings.

From here you have two options:

- Switch to **“Console”** mode by clicking on the **“Console mode”** button, which will send a command to the board. When the board next wakes up, it will receive that command and switch to **“Console”** mode.
- Stay in **“Logging”** mode. In this state, the board will stay in its normal logging mode of *measure data -> go to sleep -> wake up -> measure*

again and/or transmit data -> go to sleep again and go on. The time between each sleep and wake up depends on the Interval settings in the application.

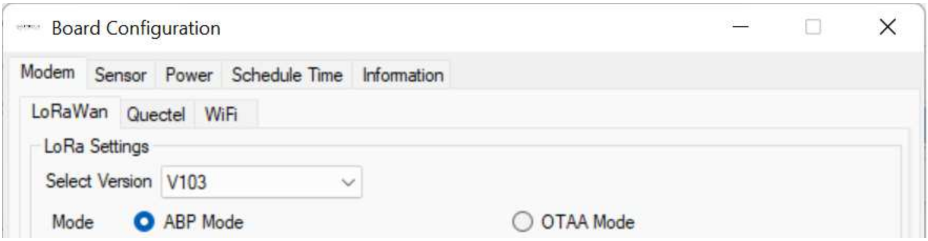
NOTE : to save power, the TBSL1 shuts down the USB port if it has not been used for a few minutes. If you cannot get a response on the USB port, press the Reset button, which is located on the top of the PCB, below the modem. Alternatively, hold a magnet next to the top right corner of the case – this will trigger the Reset reed switch.



4. Configuring the TBSL1

When you open a configuration file, you will see the various options::

- Modem (Cellular, WiFi or LoRaWAN) settings
  - for Cellular (4G and Cat M1) configuration, refer to section 4.1.2 .
- Sensor Settings
- Power: to set any power related functions
- Schedule Time: to set log and send intervals
- Information: shows info about the TBSL1



4.1. Modem Settings

The Modem tab has 3 subordinates tabs, one for each of the modem options. When you are connected to a TBSL1, the unit checks to see what modem is installed and only the tab for that modem type will be available – the others will be greyed out.

4.1.1. LoRaWAN Modem Settings

Before setting up the TBSL1 you must obtain all the relevant information on the network to which it is being connected and the Join type to be used. You must also pay attention to the LoRa WAN MAC version used on the RTU and make sure you select that version on the LoRa Network Server (LNS) e.g. Things Network, Chirpstack, Loriot.

LoRa WAN Devices can use one of two modes: ABP or OTAA

- ABP or Activation: the device has all of its security keys pre-loaded and can send data at any time
- OTAA: requires less setup as the security keys are sent over the network. But can be a problem in large networks after a Gateway outage as all devices will attempt to rejoin
- when using OTAA, your device must JOIN the network before it can send data e.g. AT+JOIN
  - when the LoRa Network Server (LNS) receives the JOIN request, it replies with the security keys for the device and a list of the channels to use

To check the TBSL1's LoRa WAN MAC version:

- connect to the unit in Console mode
- set the Command radio button to MODEM
- in the command window type AT+LW=VER
- the device will show the MAC version it is set to

```
<<07:56:26 AT+LW=VER
>>07:56:26 +LW: VER, V103
```

- If you are using Over the Air Activation (OTAA) you will need:
  - Device EUI
  - APP EUI
  - APP Key
  - you will also need to confirm the settings for the RX1 delay used by the LNS (LoRa Network Server).
- If you are using Activation by Personalisation (ABP) you will need
  - Device Address

  - Device EUI
  - Network Security Key
  - Application Security Key

To configure the LoRaWAN parameters:

- From the menu select ***Settings / TBSL1 Configuration***
- When the Board Configuration menu, click on the ***Modem*** / **LoRaWAN** tab
- In the LoRaWAN Settings section:
  - Select the activation type – ABP or OTAA
  - The fields required for each activation type will be editable and those not needed will be greyed out
  - Fill in the required information with the values given to you by your LoRaWAN provider
  - Note that all fields are in “Big Endian” format i.e. with high order bytes first

4.1.1.1 Packet Acknowledgement Settings

LoRaWAN offers two methods to reduce the risk of losing packets of data (for instance if two or more devices transmitting at the same time)

- **No Ack from Server** : This is the simplest method : each packet is simply transmitted multiple times in the hope that one gets through. The LoRa Server then deletes any duplicated messages
  - The **Repeat Times** setting controls how many times each message will be sent: initially try a value of 2, but if signal strength is very good use 1
- **Wait Ack from server** : The recommended approach is to have each Packet acknowledged by the LoRaWAN Server. If this option is selected, the device will continue trying to send a packet until it

receives an Acknowledgement. This is done by maintaining a Frame Counter which is incremented each time a new packet is sent

  - the **Retry Times** setting controls how many times to try repeating the message before going on to the next packet
  - after successful transmission of a subsequent packet, the unit will come back and retry the missed packet(s)
  - set the Retry Times to a value of 3

- For simple devices such as water meters, you can use No Ack and 3 retries
- For any application which requires a contiguous data set, such as Soil moisture and weather monitoring, you should use the Acknowledged packets option.

4.1.1.2 LoRa Frequency Settings

The LoRaWAN protocol defines a number of different channel plans based on the region in which the devices will be operated. The standard channel plan for Australia is AU915. This channel plan supports 64 channels broken up in to 8 sub bands, each with 8 channels. This limitation is imposed because the standard Gateway chip set only supports 8 channels. While some early networks used sub band 0, most now use sub band 1 (Ch 8 to 15).

Note also some operators in Australia and New Zealand have also elected to use the Asian region channel plan (AS923). This channel plan is supported by the TBSL1 and can be selected from the TBSL1 GUI.

The RF parameters are set in the LoRa RF section of the screen

- **ADR State -** Adaptive Data Rate (ADR) set this to OFF
  - ADR helps maximise the capacity of large networks by allowing devices close to the Gateway to transmit at a high rate and those further away to use a lower data rate. But the low data rates do not support long packets so can not be used with multi-parameter sensors. You can enable ADR if you are using simple sensors (e.g. water level)

- **Freq Scheme**: this is where you choose the channel plan which is being used by the Gateway
  - if you are setting up an on farm network, our preference is touse AU915 SB1. But if you are joining an existing network you must first find out whether it is using AU915 or AS923 and then select the channel plan accordingly. The Things Network uses AU915 SB2 (CH 8-15)
  - click on the SET CH button to bring up a list of the available channels and their operating frequencies
  - Clicking on Default Setting will reload the Sub band 0 settings (CH 0-7)
    - each sub-band just occupies the next 8 channels, with the frequency increasing by 0.2 MHz from one channel to the next
    - make sure that you choose the corresponding sub-band when setting up the Gateway
  - if you need any extra information on the LoRa WAN channel plans, consult the "LoraWAN Regional Parameters" document.
  - the channel plans used most commonly are as follows:

Channel Plan	**AU915 SB1 (preferred)**	**AU915 SB2 (TTN)**	**AS923**
CH0	915.2	916.8	923.2
CH1	915.4	917	923.4
CH2	915.6	917.2	923.6
CH3	915.8	917.4	923.8
CH4	916	917.6	924
CH5	916.2	917.8	924.2
CH6	916.4	918	924.4
CH7	916.6	918.2	924.6

  - the Data Rate range controls the range of data rates used when ADR (adaptive data rate) is active. This needs to match the settings in the LNS. For Chirpstack, this is held in the Service Profile (e.g. 3 to 8)

- **Data Rate:** if ADR is not being used, you must select the data rate to use on each device
    - if you are using simple sensors use DR 3
    - if you are using soil moisture probes or other multi-parameter sensors use DR 5
    - when setting up the device in the LoRa Network Server (LNS) make sure that the Downlink data rate matches the uplink rate
        - for DR5 uplink, use DR13 (SF7 / BW 500kHz) which requires a Data Rate Offset of 0
- **TX power**: this setting controls the power of the transmitter in the unit
    - This must be left on the default (max) setting i.e. 14
- **Port:** sets the port allocated for LoRaWAN traffic. Set to 8

4.1.2. 4G / Cellular Modem Settings

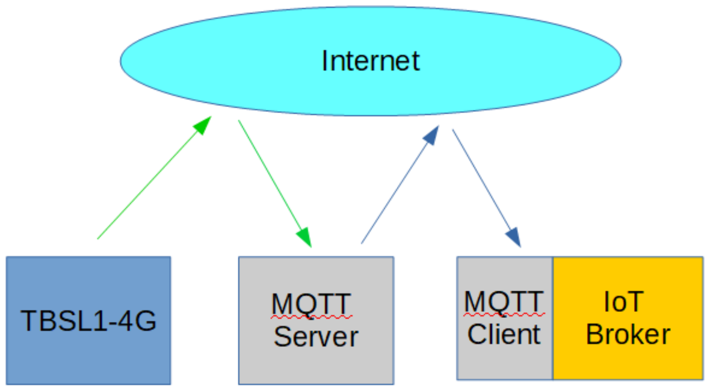
This tab is provided for the Cellular version of the TBSL1 and is used to control the transmission of readings from the unit to the Tekbox Broker

- a LoRaWAN unit may be converted to cellular operation by removing the LoRaWAN sub board and replacing it with the Cellular module
- the “Detected Modem” field on the GUI will display the type of cellular modem fitted
- the LoRa WAN tab on the Console will be greyed out and the Cellular tab activated.

The TBSL1-RFB-4G holds a Quectel universal Cat M1 / NB2 modem. It uses the MQTT protocol to send data to an MQTT Server. Any MQTT Client can then read the data which means that you can “publish” the data to any presentation software which offers an MQTT input module.

The Tekbox IoT Broker contains an MQTT client, which used to retrieve the readings from the MQTT Server. MQTT is a bi-directional protocol: it is used to send data to the server but may also send firmware updates and configuration changes to the unit. A server “publishes” its data to any “clients” who “subscribe” to one or a number of “topics”.

The flow of data between the TBSL1-4G and the Tekbox IoT Broker is as follows:



The TBSL1-4G “publishes” its readings to a “Topic” on the MQTT server. The Topic is the ID number assigned to the RTU

- make sure that you set the BOARD ID to a unique value for this telemetry unit. It should be read from the PCB
- you should also set the Downlink Path to the same ID
  - e.g BOARD ID 01041048
  - DOWNLINK 01041048/Data

The MQTT Client in the Tekbox IoT Broker, then “Subscribes” to this topic. Any new values published by the RTU are automatically sent to the MQTT client in the Broker.

The Topic for the TBSL1 can include two types of information: the readings themselves and the Metadata, which describes the RTU and sensor configuration. A new Metadata file is written each time the configuration of the RTU is changed. The presentation software can check the Metadata and, if the file has changed, use it as a trigger to update the device’s configuration.

4.1.2.1 Using Secure TLS Connections

The connection to the server can be insecure (MQTT, FTP, HTTP) or Secure (MQTTS, FTPS, HTTPS) with the latter using TLS to encrypt the information sent over the connection. To use TLS you must upload security certificates for the server to the TBSL1 modem:

- contact the organisation hosting the server and ask for them to EMail you a copy of the TLS Certificate files for the server. Save the files on your PC
- Open the GUI and load the relevant device configuration
- open the configuration by selecting ***Settings / TBSL1 Configuration***
- select the ***Cellular*** tab
- activate the **SSL Enable** check box
- select the appropriate security level e.g. Sever and Client Authentication
- click on the **UPLOAD CERTIFICATE** button and then select the certificate file you saved earlier.

4.1.2.2 MQTT Configuration

The Tekbox IoT Broker includes an MQTT Client which can collect the readings from the MQTT server to which the RTU is publishing the data.

TOIP operates two MQTT servers for this purpose: one on a cloud server for operational use and another on a local server for testing.

Operational cloud server:

URL toip-server.net.au

Port	1883	(unencrypted)	8883 (encrypted)
User	toip		
Password	p10t		

The settings for the Cellular tab should be set as follows:

- Select RAT: this controls the data mode used
  - select either Cat M1 or NB2. NB2 allows a greater range but Cat M1 is more suitable for longer packets such as found with multi-parameter SDI-12 sensors
- Connection Type: select IP4 unless you are sure the server supports IP6 protocol
- Client ID: enter the name of the MQTT broker you will be using e.g. MQTT_CLIENT_A
- Network APN: the APN being used by the SIM card e.g. telstra.internet
- Network User: the APN User Name (blank for telstra.internet)
- Network Pass: the APN Password (blank for telstra.internet)Board ID: set to the Serial Number of the TBSL1
- Board ID: set this to the serial number of the TBSL1
- Protocol: set this according to the server you are sending to
  - for Chirpstack/Tekbox IoT Broker select MQTT
- Protocol Host: enter the URL or IP address of the server
  - for Chirpstack/Broker use toip-server.net.au
- Port: select a port to match the protocol
  - for MQTT 1883 (for unencrypted sessions, 8883 for encrypted)

- Protocol User: The user name for the server connection
    - for Chirpstack/Tekbox IoT Broker use toip
- Protocol Pass: The password for the selected user
    - for the Tekbox IoT Broker use p10t
- SSL Enable: activate this box if you are using MQTTS, FTPS or HTTPS
- SSL Level: if using SSL, select the level of security
    - if you have selected Server Auth or Server and Client Auth, you will need to click on the UPLOAD CERTIFICATE button and upload the security certificate for the server. Contact TOIP and request a copy of the certificates for the Chirpstack server
- Topic Publish: the name used to identify this feed e.g. TOIP
- MQTT Client ID: set to MQTT_CLIENT_1
- MQTT Publish: set to TOIP
- Topic Downlink: the path to the folder in which the data will be stored
    - this is made up of the TBSL1 ID followed by the topic name (from the field Topic Publish)

      e.g. for TBSL1-4G with ID 40000011 use /40000011/Data
- MQTT QoS: set to 2, this ensures that every packet is delivered

After setting up the RTU, you must make sure that you the device is added to the Tekbox IoT Broker (refer Tekbox IoT Broker User Manual) – TOIP will normally perform this setup task for you.

4.1.2.3 Testing 4G Connectivity

Once the configuration has been set, write it to the RTU Overall Settings / Send All Parameters

You can then test the connectivity:

- to check that the SIM is active   Monitoring / Cellular Monitoring / Check SIM Status
- to check that the unit has registered to the network   Monitoring / Cellular Monitoring / Test Network Registration
- to check that the unit has registered to the network   Monitoring / Cellular Monitoring / Test Connection and Transmission
    - the unit will show the status of the various steps in making the connection
- You will not be able to place the unit in to Logging mode until this test has successfully completed.

4.1.3. WiFi Modem Configuration

When the TBSL1 is fitted with a WiFi modem board, it is designated as the TBSL1-WF. The TBSL1-WF uses a local WiFi service to make the data connection to an MQTT server – typically the one in the Tekbox Broker. The data path is thus the same as that for the TBSL1-4G other than the fact that the initial connection is to a wireless access point rather than a mobile phone tower.

4.1.3.1 WiFi Configuration

When a WiFi modem board is installed, the WiFi tab in the configuration will be active and the 4G and LoRa WAN tabs greyed out. With the configuration open, click on the Modem tab and select WiFi. Make sure that the TBSL1 has been programmed with the WiFi specific firmware (04.02.0a.bb where the 02 indicates the WiFi firmware).

The first step is to scan for available networks. To do this click on the **SCAN** button

- the RTU will scan the network and respond with a list of the networks in range

- select a network from the drop down list and then enter the password in the **WiFi Pass** field.

The next step is to configure the Host used for the MQTT transfer. The settings below are for TOIP's Chirpstack Server

- SSL Enable: check this box if the MQTT server uses a secure connection (leave it off for Chirpstack)
- MQTT Host: set this to the URL or IP address of the server e.g. 120.150.31.37
- MQTT USER: set this to the user name for the MQTT session e.g. toip
- MQTT PASS: set this to the MQTT password e.g. p10t
- MQTT PORT: set this to match the server port 1883 (1883 for insecure and 8883 for MQTTS)
- MQTT Board ID: set this to 0540nnnn where nnnn is the serial number for the TBSL1
- MQTT Client ID: set this to a random string of characters
- MQTT Publish: this is the topic to which the data will be published. Set it to TOIP
- MQTT Downlink: set this to 0540nnnn/data
- MQTT QOS: set this to 1.

When you have checked the settings, save the file and send them to the RTU.

To test Connectivity:

- Select Monitoring / WiFi Monitoring / Connect to the WiFi and confirm that the unit can connect
- Select Monitoring / WiFi Monitoring / Check Current WiFi status and confirm the signal strength is acceptable (> -80 dBm)
- Perform an MQTT test with Monitoring / WiFi Monitoring / Test MQTT Connection and Transmission

- if the test fails, double check the credentials.

4.2. Time Settings

The TBSL1 units are intended to be used with the time set to UTC (Universal Coordinated Time, which was referred to for a long period as GMT or Greenwich Mean Time). It is also the standard used by Sensori. This allows the data from devices anywhere in the world to be compared and avoids issues with time stamps changing when daylight savings starts and finishes (which causes a 1 hour gap in the data).

The TBSL1 may be configured to synchronise its time with that of the Gateway:

- select the **Schedule – Time** tab then **Time Synch from Modem** and activate the checkbox
- set the time to 5 sec
  - if the RTU time is more than 5 seconds different from that of the downlink packet, the RTU will adjust its time to match the Gateway time

The time zone settings are located in the ***Schedule Time / RTC Settings*** of the configuration screen:

- First click on the **Time Zone** drop down list and then set the Time Zone to **UTC**
- If you want to fetch the time zone from the PC you are using, click on the **Get User PC Time Zone** button
- To write the time to the TBSL1 select the Overall Settings Tab and the click on **Send Real Time**

After setting the time, you should fetch the time from the TBSL1 and check it to ensure it is correct:

- To read the time from the TBSL1, select the Overall Settings tab and then click on ***View / Get Current Date***

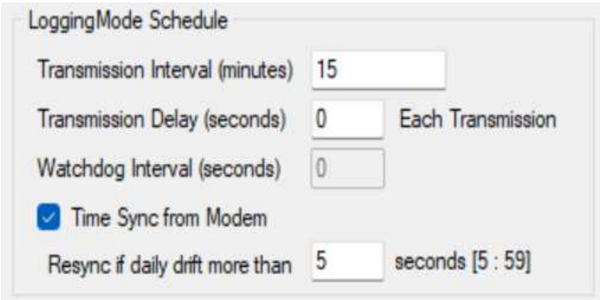
- The time will be displayed on the log screen. Using the offset to local time for your local, confirm that the time is correct
    - e.g. the image below shows the time from the TBSL1

```
<<16:09:52 Getting Ext.RTC: Tuesday, 3/12/2019, 5:39:50
>>16:09:52 ---------------Getting Ext.RTC Parameter: DONE---------------
```

- The command was sent in Adelaide during Daylight Savings, where the time is 10:30 ahead of UTC. Adding 10:30 to 05:39:50 indicates a local time of 16:09:50.

4.2.1. Time Synchronisation

The TBSL1 can synchronise its time to the time set in the LoRa WAN Gateway(s) – if the LoRa WAN Gateways are in turn set to synchronise their time to an NTP server, the TBSL1 clock will always be very close to the true time. This is achieved using a Time Synch MAC command which has been added to the LoRa WAN protocol.



To activate Time Synchronisation:

- activate the “Time Sync from Modem” checkbox. Then set a value for how many seconds drift will be allowed before the clock is updated.

4.3. Schedule Settings

These control how often measurements are made and transmitted. Normally the Measurement Interval (found in the SDI-12, Analogue or Pulse sensor settings) and Transmit Interval will be set to the same value

- Transmission Interval: set to suit your application e.g. 15 min for soil moisture and weather sites

- Watchdog Interval (Seconds): this is used to detect fault conditions in the TBSL1
  - It is currently not implemented and will be enabled in a future firmware release
- Battery Information: to save air time, you can reduce the rate at which the battery information is sent. Changing the rate at which the Battery is sent also changes how often the C or Common Data packet is sent
  - For example with 15 min logging, if this is set to 4 cycles both a B and C packet will be sent once per hour
- Transmission Delay:
  - you can leave this on zero or add a delay to each unit
  - in large networks this will help reduce the number of collisions (where two devices send on the same channel at the same time)
  - if you are going to use the Delay feature, create a spreadsheet in which you can record the delay for each device
- Power Management: this setting is currently disabled and will be activated in a future firmware release.

4.4. SDI -12 Sensor Settings:

Prior to configuring the SDI-12 sensors in the GUI, connect the sensors to the input ports on the TBSL1. This is because, when the RTU assembles its configuration, it sends the measure command to each sensor and records how many values are to be returned

- The RTU sums up the total number of measurements (SDI-12, Pulse and Analogue) and sends them as a parameter in the Common Data or "C" parameter (See section 6.2)
- This will be updated when the configuration is sent and whenever the RTU is changed from Console to Logging mode.

Note that the SDI-12 pins on the two connectors are wired in parallel so it does not matter whether you use Connector 1 or 2.

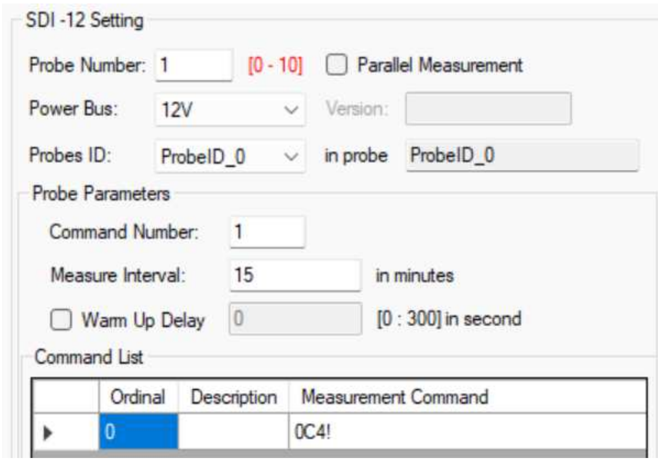
In the SDI tab you add details of the SDI-12 commands used by the various sensors connected to the TBSL1.

Because SDI-12 supports multiple sensors, each of which may make multiple measurements, the TBSL1 allows you to define “Probes” (defined as a sensor on a give address) and Commands (the individual measurements taken)

- A 120cm 12 sensor Soil Moisture Probe will count as a Single Probe but will have 2 Command entries (1) for Soil Moisture on 0C! And (2) for Soil Temperature on 0C2!
- An HT03 air temperature and relative humidity sensor will count as a single Probe with 1 command entry (0C1! for RH & Temp).

To add your SDI-12 sensors:

- first locate the **Number of Probes** field and enter the number of “Probes” you will connect
    - this will be 1 if you have a single sensor (e.g. Air Temperature and Relative Humidity)
    - a weather station with Air Temp/RH, Wind Speed, Wind Direction would have a count of 3



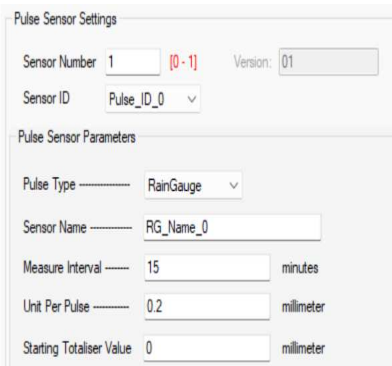
- set the POWER BUS Option to 12V

- then, for the first Probe, click on the **SDI-12 Probes ID** field and select 1 to enter the settings for this sensor
- Click on **Measurement Interval** and set it to the desired rate e.g. 15 min
- for sensors which need a warm up time in addition to the standard SDI-12 time (for example with WS01 sensors operated in Advance mode or with ultrasonic wind sensors) you can specify a **Warm Up Delay** for each Probe. Tick the check box to activate the delay, then enter the number of seconds to wait
- in the **SDI-12 Sensor Settings** field, change the Description to one appropriate for the Sensor e.g. Otto SM or HT03
- then set the **Measurement Command** to match the SDI-12 command used to make this measurement e.g. 0C! For soil moisture on the Otto probe, 0C1 for a HT03
- repeat this process for each additional sensor (each will be on a different address)
- if your sensor requires permanent power, in the Power Management section, click on the checkbox labelled "**12V Always On**".

To add a second sensor, make sure the Number of Probes is set to 2 (or more), then select the 2nd Probe from the SDI-Probes ID list (the second sensor is ProbeID_1) and then repeat the process.

4.5. Adding Pulse Sensors

Pulse inputs are normally used to connect to flow meters and rain gauges. The TBSL1 can support up to two pulse inputs although only one is active in the current firmware.



To add a Rain Gauge or flow meter:

- click on the **Pulse** tab
- leave the **Number of Pulse Sensors** to 1 (more will be supported in a future firmware release
- leave the **Pulse ID** list box at 1
- in the **Pulse Sensor Parameters** section, record the configuration for this sensor:
    - Pulse Type: Select Rain Gauge or Flow Meter
        - Rain Gauge mode: RTU records the number of pulses and the rain (mm) based on the units per pulse
        - Flow Meter: RTU records the number of pulses and the Flow Totaliser
    - Name: give the sensor a name e.g. Precipitation
    - Pulse Measurement Interval: set to the same rate as the SDI-12 sensors
    - Units per pulse: set to match the sensor e.g. 0.2mm per pulse for rain gauge
    - Starting Totaliser value: if you are using the TBSL1 on a flow meter and want the Totaliser value to match the value on the meter's mechanical dials, set the totaliser to the current value shown on the dials.

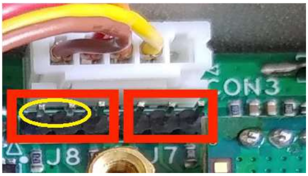
4.6. Adding Analogue Sensors

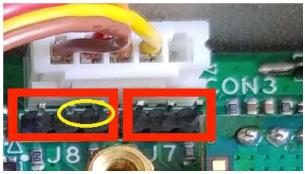
The TBSL1 carries two analogue inputs (although only one is enabled in the current firmware).

The analogue input may be used in one of two modes: 0 - 2.5V or 4 - 20mA. The selection of which to use is made according to the state of jumpers J8 (CH 0) and J7 (CH 1). This is located below the analogue connector CON3.

The three pins on the jumper number from left to right and there is a small triangle on the PCB to indicate pin 1.

Input Type	Jumper Position
4-20mA	Pins 1 -2 (Centre and left)
Voltage (0-2.5)	Pins 2 - 3 (Centre and right)



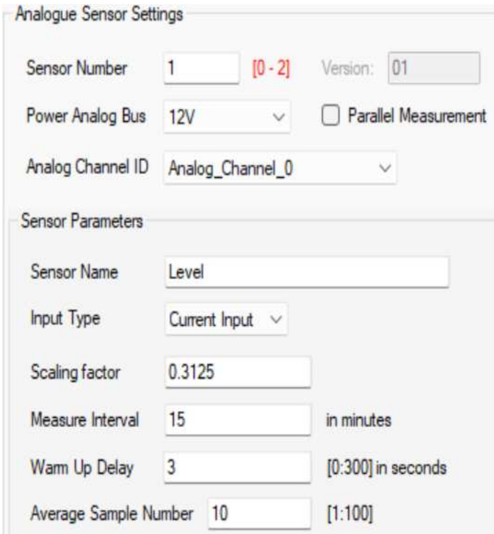


CH 0 Links for 4-20mA Input   **CH 0 Links for Voltage Input**

The analogue sensors are configured in the Analog tab.

To add an Analogue sensor to the configuration:

- click on the **Analog** tab
- click on the **Number Analogue Sensors** field and set it the total number of analogue sensors you will connect (0, 1, 2)
- in the **Analogue Channel ID** list box, select the first sensor
- Power Analogue Bus: leave this set to 12V



- Parallel Measurement: leave this set to off
- Analogue Sensor Name: give the sensor a suitable name
- Analogue Measurement Interval: set this to a rate suitable for the sensor e.g. 15 min
    - Port: the first sensor will be on Port 1
- Warm Up Delay: the time needed to wait before the sensor will give a stable reading (use 1 or 2 sec if unsure)
- Number of Samples: this setting allows you to apply averaging over a number of readings which will help reduce noise. But more samples mean higher power consumption
- Scaling Factor: you can enter a multiplier here in order to scale up your values to represent the true reading e.g. if you have a voltage sensor where full scale of 2.5V represents 5m, enter a scaling factor of 2.

You can make an immediate analogue measurement as follows:

- click on the **Others** tab
- click on the **Analogue Measure** button
- the RTU will read the analogue sensors and display the result on the log screen
- when the mode is set for voltage sensors, the reading will display in volts
- when the mode is set for 4-20mA sensors, a 125 ohm precision resistor is switched in to the circuit and the unit measures the voltage across this resistor

```
>>15:10:07 --------------Sending Analog Sensor Parameter: DONE--------------
<<15:11:07 Instant Analog Measurement...
<<15:11:08 Analog CH0 Voltage: 0.564mA
<<15:11:08 Instant Analog Measurement: DONE
```

- to convert the voltage to the equivalent current, divide the voltage by 125 e.g Voltage = 1.183 → 1.183/125 = 9.46mA

Measured Voltage	**Current**
0.5V	4mA
1.25V	10 mA
2.5V	20 mA

Alternatively if you want to see the results of all the individual samples taken on the analogue port use the Get function:

- from the menu select View / Getting Sensor Measurement / Getting Analogue Measurement
- this will read the sensors and write the results to a file called AnalogMeasurement.txt which is stored in the “Measurements” directory under the installation directory
- 256 samples will be taken and written to the file.

4.7. Setting up Alarms

The TBSL1 allows you to create an alarm if a parameter goes outside of preset limits. Alarms can be generated based on the value of any of the SDI-12, analogue or pulse sensors.

To create an alarm:

- if you have changed any configuration settings, click on Save to update the stored configuration
- from the Board Configuration dialogue box, click on the Alarm tab

- now, click on the Update Sensor List button so the GUI reads in a list of all of the sensors currently configured on this device
- click on the Choose Sensor list box and select the sensor you wish to add the alarm to e.g. Water Level
    - the Alarm Type will change to match the type of sensor
- Alarm name: set this to an appropriate descriptive name e.g. High Level
- Alarm Conditions: from the list box select the type of alarm to be generated and then the limits to apply according to the alarm type
    - Over: generates an alarm if the value exceeds the Max Value e.g. over 40
    - Under: generates an alarm if the value is below the Min Value e.g. under 20
    - Inside Limit: generates an alarm if the value is below the Max Value and above the Min Value e.g. between 20 and 40
    - Outside Limit: generates an alarm if the value is above the Max Value and below the Min Value e.g. is not between 20 and 40
- Hysteresis: this is the range the value must recover before a new alarm can be triggered and is designed to prevent multiple alarms being generated if a value is sitting close to the alarm point
    - for a high alarm, if the Max Value is 40 and the hysteresis 4, the value must fall to below 36 before a new alarm will be recognised
- Enable Alarm: activate this check box to enable processing of the alarm
- Actions: the method by which alarms can be notified varies according to the type of communications module fitted to the TBSL1
    - SMS to mobile Phone – only available on cellular unit. If this option is selected, enter the phone number in the field below the action type

    - Email – only available on cellular unit. If this option is selected, enter the recipient's Email address in the field below the action type
    - Data to Packet Server – available on LoRaWAN and Cellular units. This will send a data packet to the server where it can be actioned by the limit testing functionality provided by the vendor
- Toggle GPIO: this function enables you to turn on one of the TBSL1 outputs in response to the alarm condition
    - note that in the present firmware only PC0 is available.

4.7.1. Setting the Board ID

You can allocate your own Serial Number or ID to each unit as it is programmed. This can be helpful in keeping track of units, particularly if you move them from one LoRa App Server to another.

The Serial Number is set in the Info tab

- after selecting **TBSL1 Configuration**, click on the ***Info*** tab
- the **Board ID** will display the default value
- change this to the required number
- click on the ***Save*** button to save the changes.

4.8. Writing a Configuration to the TBSL1

Once configuration is complete, you need to write it to the TBSL1

- make sure that the TBSL1 is in Console mode and that it has connected to the GUI
- click on the **SEND ALL SETTINGS** button
    - the GUI will send the settings to the TBSL1
    - if the GUI can send the commands, you will see a message in the Data Log section showing that each command was successful

   - if you receive an error with every command, the TBSL1 is not yet active (wait or cycle the power and retry)
   - if you receive an error on 1 or 2 commands, the GUI and TBSL1 may be set to different firmware versions. Contact us to organise an update to the GUI and / or TBSL1 firmware

- you can send settings for specific functions by selecting the tab corresponding to that area and then clicking on the relevant send button
   - Overall Settings: send All, Modem, Schedule or Clock settings
   - Sensor Settings: send All, SDI-12, Pulse or Analogue settings
   - Others: make an immediate Analogue measurement, get board version and serial number.

4.9. Testing Modem Connectivity

For each type of Modem used on the TBSL1, you will need to check end to end connectivity with the server.

4.9.1. Testing LoRaWAN Connection

4.9.1.1 RSSI Check

The ***Monitoring / LoRa WAN Monitoring / Check RSSI*** command sends a null message to the LoRa WAN Server and displays the signal strength (RSSI). This command sends a “Link Check” packet, which is just an empty packet. The LoRa Server will always acknowledge a Link Check packet.

```
<<11:56:51 AT+MSG
>>11:56:51 +MSG: Start
>>11:56:51 +MSG: Link 16, 1
>>11:56:51 +MSG: RXWIN1, RSSI -71, SNR 6.0
>>11:56:51 +MSG: Done
```

If you do not see a reply, the Acknowledgement message was not received by the TBSL1. Double check your settings. Then track the message from the

Node to the Gateway then to the Application. You should also be able to see the ACK (acknowledgement) being sent by the Gateway back to the TBSL1.

4.9.1.2 Test Transmission

The ***Monitoring / LoRa WAN Monitoring / Check Transmission*** command sends a test message with the contents "test transmission"

- after selecting this option, watch on the Gateway and Application to see if the message was received
- Gateways are promiscuous and the traffic monitor will show all messages
- the traffic monitor in the Application (and Node) will only display messages where the App and NWK session keys are correct

```
<<11:59:19 AT+MSG="test transmission"
>>11:59:19 +MSG: Start
>>11:59:19 +MSG: Done
```

4.9.2. Testing 4G Connectivity

When the TBSL1 is powered up and connected to the GUI, check that the "Detected Modem" displays as **BG95-M6**

You can confirm whether the SIM card is active during startup:

- if the unit is already running, press the Reset button to restart the TBSL1
- watch the console for messages as the TBSL1 starts and look for a message labelled Get Modem Hardware ID
- this should be followed with BG95_M6

Check the SIM status:

- from the menu: **Monitoring / Cellular Monitoring / Check SIM Status**

Check the unit has registered on the 4G network

- from the menu: **Monitoring / Cellular Monitoring / Test Network Registration**

Perform an end to end MQTT Test

- this can only be completed once you have entered the details for the server
- from the menu: **Monitoring / Cellular Monitoring / Test Connection and Transmission**

```
12:03:19 Checking SIM Status ====> Ready, not pending for any password.
12:03:20 Set radio access technology ====> PASSED
12:03:20 Checking Network Register Status ====> Registered, home network.
12:03:21 Check network information: Operator ====> "Telstra Mobile".
12:03:21 Check network information: Access technology ====> Unknown.
12:03:24 Checking TCP connection ====> PASSED
12:03:30 MQTT open connection ====> PASSED
12:03:32 MQTT connect to server ====> PASSED
12:03:34 Publish packet to topic "TOIP" ====> PASSED
12:03:36 Connection and transmission test ====> PASSED.
12:03:36 --------------- Connection and Transmission Test: DONE ---------------
```

If your connection fails (i.e. the response shows ERROR):

- check that the SIM Card is active
- check that the MQTT server is on line.

4.9.3. Testing WiFi Connectivity

You can now test connectivity to the APN and to the MQTT server

- to test APN connectivity select **Monitoring / WiFi Monitoring / Connect to WiFi**
    - the screen will advise whether the connection was successful or not
    - make sure you have an antenna on at this stage

- to check the WiFi signal strength **Monitoring / WiFi Monitoring / Check current WiFi status**
    - the screen will display the APN, MAC address, channel and signal strength
- to check connectivity to the broker **Monitoring / WiFi Monitoring / Test MQTT Connection and status**
    - the unit will attempt to connect to the MQTT server and send a test packet

You can monitor traffic on the WiFi connection using an MQTT client as shown earlier for the 4G units.

4.9.4. Monitoring MQTT Server Traffic (All Modems)

Once the RTU is logging, you can monitor traffic through the MQTT server using an MQTT Client such as MQTT-FX.

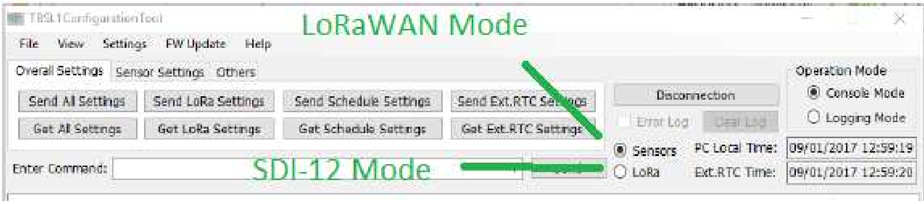
After installing MQTT-FX, create a new Connection Profile for the server:

- select Extras / Edit Connection Profiles
- at the bottom left of the screen click on + to create a new profile
- give the Profile a name e.g. TOIP LoRa Server
- set the Profile Type to MQTT Broker
- set the Broker Address e.g. toip-server.net.au
- set the Broker Port e.g. 1883 for unencrypted connections
- type in a Client ID e.g MQTT-Test
- click on the User Credentials tab
    - set the User Name to toip
    - set the Password to p10t
- Click on Apply and then OK to save the changes

- open this session by selecting it from the list of available connections and click on CONNECT
    - the list will only be accessible if you are not already connected
- click on the SUBSCRIBE tab
- in the Topic text box type in the name of the Topic to use
    - in MQTT the "/" character is a separator, the "#" is a wildcard
    - if you leave the topic as # it will show all traffic
    - to show just your topic, set it to the Topic Name used in the TOPIC PUBLISH field in the configuration
        - for LoRa WAN nodes application/<application number>/device/+/event/up
        - for 4G and WiFI nodes +/TOIP/#
- click on SUBSCRIBE to activate this feed
- any traffic from the Node(s) will be displayed in the traffic window
- if you inspect the packet contents you should be able to recognise the various packet types: C (Common data), PB (Battery), PS (SDI-12), PP (Pulse) and PA (Analogue)
- if you have an SDI-12 sensor connected and being read, you will see PS packets with the date/time, Sensor ID, Sub Sensor ID, number of values and finally the values
- the packet also includes the modem type and device ID (which is used as the EUI in the Broker).

5. Sensor Testing

The TBSL1 GUI allows you to send commands to either the SDI-12 sensors or to the LoRaWAN modem. You select which to use by clicking on either the **Sensors** or **LoRa** radio buttons.



5.1. SDI-12 Sensor Testing

You can use the GUI to send commands to the attached SDI-12 sensor. This can be viewed as a transparent command mode: commands you type must be terminated with the "!" used by SDI-12. When you click on Send, the command will be sent over the SDI-12 bus. The Log screen will then echo any response given by the sensor

- in the SEND COMMAND TO section, click on the SENSORS tick box
- in the ENTER COMMAND line, type in the command to send to the sensor
  - to make a test measurement on the third sensor on the probe (address 2) type 2C1! And press enter
  - to read the command result type 2D0! " "

```
>>11:03:50.563 CMD,SDI12,2C1!
<<11:03:50.762 RES,SDI12,200102
>>11:03:55.509 CMD,SDI12,2D0!
<<11:03:55.754 RES,SDI12,2+0.00+25.92
```

5.2. *Starting the TBSL1 Logging*

5.2.1. Starting the TBSL1 immediately

After completing the setup of the TBSL1, make sure you save any changes to the configuration file. You must then switch the unit from Console mode to Logging Mode:

- with the TBSL1 GUI open, click on the Logging Mode radio button in the Operation Mode section at the top right of the screen
- the console log will display a message indicating that the unit has switched to Logging mode
- you can now disconnect the USB terminal
- note that if you leave the USB terminal connected, you can watch as the unit goes through its measurement and log cycle.

<<13:32:29 Platform is switching to Logging Mode...
>>13:32:29 Closing COM Port: PASSED

5.2.2. Using the Hibernate Function

The Hibernate function allows you to configure a TBSL1 then put it to sleep. It will wake up and go to Logging mode as soon as either the Solar Panel or USB connector is plugged in. The link between pins 3 and 4 on the Solar panel plug are used to signal the unit to come out of hibernation.

To set a unit to Hibernate mode:

- complete the configuration of the unit
- save any changes to the configuration file for the unit
- write the updated configuration to the unit
- on the GUI menu, click on the Logging button and wait until the TBSL1 shows that it has finished writing its settings and is in log mode
- unplug the Solar Panel

- the unit will go to sleep mode
- when you connect the solar panel again, the TBSL1 will wake and go back to Logging mode.

6. TBSL1 Packet Format

The TBSL1 sends data to the LoRa WAN Server as one or more packets. Each packet is time stamped and includes an identifier to signify the type of information it holds. This flexibility is important because each of the different measurement types may be sent at a different rate: e.g. SDI-12 soil moisture data every 15 minutes, battery voltage every 4 hours.

The TBSL1 defines a number of different message types:

- C Common data messages
  - includes sensor count and RSSI
- P Parameter messages (hold data) with several sub-types
  - B Battery voltage
  - P Pulse counter values
  - A Analogue sensor values
  - S SDI-12 sensor values

6.1. Order of Transmissions

The packets are always sent in the same order:

Link Check	AT+MSG	-> always acknowledged
C	Common Data Packet	→ sent with B
B	Battery packet	-> sent at rate set in config
P	Pulse packet	-> sent if in configuration
A	Analogue packet	-> sent if in configuration
S	SDI-12 packets	-> sent if in configuration

In a LoRa WAN system the Link Check packet is used to check the path from the Node to the Gateway. It will always be acknowledged (ACK) by the LNS. If Wait For Acknowledgement is selected, data packets will only be sent of the

Link Check is acknowledged. If Wait for Acknowledgement is turned off, the unit does not wait for the ACK, it just sends every packet the selected number of times.

6.2. Common Data Messages

The content of the Common Data Messages is shown below:

Message for Common Data Report							
Report ID	Device ID	FW Version	Power Supply ID	Sensor Number	Board Status ID	Space character	RSSI
1 byte	*8 bytes*	*8 bytes*	*1 byte*	*1 byte*	*1 byte*	*1 byte*	*RSSI format*

The fields hold the following:

Report ID	C	Signifies a Common Data message
Device ID	8 bytes (hex)	The ID for the TBSL1. This is in hexadecimal format
FW Version	8 bytes (hex)	The current firmware version
Power Supply ID	1 byte	The type of power option in use 1 Solar & NiMH 2 External 6-12V DC (no battery) 3 Internal single use Lithium cell
Sensor Numbers	1 byte	The total number of sensors in the configuration e.g. total of SDI-12, pulse and analogue
Board Status	1 byte	The current state of the TBSL1 e.g. R=Running
Space	-	To separate status from RSSI
RSSI	1 to 7 bytes	The signal strength in dBm. May be 1 to

7 digits with a range of +20 to - 130

Sample Message:		C4AC140100073.851 -3	
Content	C		Common message
		00004AC1	Device ID 4AC1
		0400010E	Firmware version 04.00.01.0E
		0	Solar power supply with Li-Ion battery
		07	7 sensors in config
		R	Unit is running
		-3	Signal strength -3 dBm

6.3. Data or Parameter Messages

The Data messages contain sensor readings and are always time stamped. They also include an identifier for the type of sensor.

6.3.1. Battery Messages

The battery voltage may be sent with each poll or after a given number of polls/transmissions (e.g. 4).

The format of the Battery packet is as follows:

Message Format of Battery Parameter Report				
Report ID	Sensor Type ID	Timestamp	Space Character	Battery Voltage
1 byte	*1 byte*	*17 bytes*	*1 byte*	*5 bytes*

The fields hold the following:

Report ID	P	Signifies a Parameter message
Sensor Type ID	B	To signify a Battery message

Timestamp	e.g. 18:08:28:09:15:00	The date and time of the reading in the format YY:MM:DD:hh:mm:ss
Space		Separates sensor index from value
Battery V	5 Bytes	The measured battery voltage

Example:

UEIwMDowMTowMToxNTozMDowMCA0LjUxMg==

P B 2 3 : 0 1 : 2 1 : 1 5 : 3 0 : 0 0 4 . 5 1 2

6.3.2. Pulse Data Messages

The format of the data for the pulse sensors is as follows:

Message for Pulse Count Sensors Report									
Report ID	Sensor Type ID	Timestamp	SensorID	SubSensorID	Number of Parameters	Space Character	Parameter 1	Space Character	Parameter 2
1 byte	*1 byte*	*17 bytes*	*1 byte*	*1 byte*	*1 byte*	*1 byte*	*8 bytes*	*1 byte*	*12 bytes*

The fields hold the following:

Report ID	P	Signifies a Parameter message
Sensor Type ID	P	To signify a Pulse sensor message
Timestamp	e.g. 18:08:28:09:15:00	The date and time of the reading in the format YY:MM:DD:hh:mm:ss
Sensor Index	e.g. 0	The index for this sensor. The range is 0 to 1 (currently only 0 supported for first pulse counter)
Space		Separates sensor index from value

Sub Sensor Index	e.g. 0	This indicates whether the input has been configured for rainfall or flow meter logging: 0 = rainfall 1 = flow meter
Space		Separator
Parameter 1	8 Bytes Hexadecimal e.g. 0000000C	The number of pulses recorded in the current log period
Space		Separates values
Parameter 2	12 Bytes Exponential format e.g. 2.400000E+00	The rain value as calculated by multiplying the pulse count by the resolution and reported in exponential format

Sample Message: PP18:08:28:09:23:562 1204 4876832

Content	P	Parameter message
	P	Pulse counter
	18:	Year
	08:	Month
	28:	Day
	09:	Hour
	23:	Minute
	56:	Seconds

		Space
	000000C	Pulse counter value (0Cx = 12)
		Space
	2.400000E+00	Rainfall mm

6.3.3. Analogue Sensor Messages

The format of the data for the Analogue sensors is as follows:

Message for Analog Sensors reporting									
Report ID	Sensor Type ID	Timestamp	Sensor Index	Space	Min Parameter	Space	Avg Parameter	Space	Max Parameter
1 byte	*1 byte*	*17 bytes*	*1 byte*	*1 byte*	*variable*	*1 byte*	*variable*	*1 byte*	*variable*

The fields hold the following:

Report ID	P	Signifies a Parameter message
Sensor Type ID	A	To signify an Analogue sensor message
Timestamp	e.g. 18:08:28:09:15:00	The date and time of the reading in the format YY:MM:DD:hh:mm:ss
Sensor Index	e.g.0	The index for this sensor. The range is 0 to 1 (Currently only 0 supported for first analogue sensor)
Space		Separates sensor index from value
Min Parameter	e.g. 0.23541	The minimum value recorded in the measurement interval (1 to 7 digits)
Space		Separator
Avg Parameter	e.g. 1.65567	The average value recorded in the measurement interval (1 to 7 digits)
Space		Separator

Max Parameter	e.g. 2.01339	The maximum value recorded in the measurement interval (1 to 7 digits)

Sample Message: PA18:08:28:09:23:560 0.23541 1.65567 2.01339

Contents	P	Parameter message
	A	Analogue Sensor
	18:	Year
	08:	Month
	28:	Day
	09:	Hour
	23:	Minute
	56:	Seconds
		Space
	0.2354	Min value
		Space
	1.65567	Avg value
		Space
	2.01339	Max value

6.3.4. SDI-12 Sensor Messages

The format of the data for the SDI-12 sensors is as follows:

Parameter Report Message of SDI-12 Sensor										
Report ID	Sensor Type ID	Time stamp	SDI-12 Probe ID	SDI-12 Ordinal	Number of Parameters	Space Character	Parameter 1	....	Space Character	Parameter n
1 byte	*1 byte*	*17 bytes*	*1 byte*	*1 byte*	*2 bytes*	*1 byte*	*variable*	...	*1 byte*	*variable*

The fields hold the following:

Report ID	P	Signifies a Parameter message
Sensor Type ID	S	To signify an SDI-12 sensor message
Timestamp	e.g. 18:08:28:09:15:00	The date and time of the reading in the format YY:MM:DD:hh:mm:ss
SDI-12 Probe ID	e.g. 1	The position in which this probe appears in the SDI-12 configuration for the device (range 0 to 9 where 0 is first probe). This typically indicates a given SDI-12 address
SDI-12 Ordinal	e.g. 0	An identifier for the measurement command for this probe. For a soil moisture probe on address 0 which returns soil moisture (SM) and soil temperature (ST), SM will be ordinal 0 and ST ordinal 1
Number of Parameters	e.g. 3	The number of sensor values which follow where n = 01 to 99
Space		Separates sensor index from value
Parameter 1	e.g. +0.23541	The first sensor value (1 to 7 digits)
Space		Separator
Parameter 2	e.g. +1.65567	The second sensor value (1 to 7 digits)

Space	Separator
Parameter 3 e.g. +2.01339	The third sensor value (1 to 7 digits)

Sample Message: PS1018:08:28:09:23:561003 +0.23541 +1.65567 +2.01339

Content	P	Parameter message
	S	SDI-12 data
	18:	Year
	08:	Month
	28:	Day
	09:	Hour
	23:	Minute
	56:	Seconds
	1	Probe ID 1
	0	Ordinal 0
	03	Number of parameters =3
		Space
	+0.2354	First sensor value
		Space
	+1.65567	Second sensor value
		Space
	+2.01339	Third sensor value

Missing Sensor

- if the TBSL1 can’t communicate with an SDI-12 sensor on a given address, the packet will be send with the text “No response”

  e.g. No response from sensor on address 0

7. Firmware and Console Update

The version of the Console GUI and RTU must be kept the same to ensure that the device operate correctly. If the versions are different, you will see a nag message when the TBSL1 connects to the GUI.

Contact TOIP to obtain a copy of the latest build of the Application firmware and GUI.

7.1. To Check the Current Version

7.1.1. Checking the Console GUI Version

To check the GUI version:

- from the menu select Help / About
- the GUI version will display in the field “Application Version” at the bottom of the dialogue box.

7.1.2. Checking RTU Firmware Version

To check the RTU firmware version:

- with the RTU connected to the GUI, click on View / Get Firmware Version

7.2. Upgrading the Console GUI

Before upgrading, backup your configuration files:

- locate the Directory in which your existing copy of the program is installed e.g. C:/Users/Public/Public Documents/TBSL1 Configuration Tool
- copy the contents of the JSON directory to a backup directory: the format of the configuration files may change from version to version as new features are added. This makes it mandatory to start with a new, clean configuration file for each device. When creating a new config

file, open the old file with a Text Editor (e.g. Notepad++) and copy and past the relevant fields form the text editor to the configurator.

The old version must be Un-installed via the Programs and Features option. After removing the old version from Control Panel / Programs and Features. To install the program, double click on the exe file located in the installation directory

- when prompted set the installation path to: C:/Users/Public/Public Documents/TBSL1 Configuration Tool
- when prompted, select install for “Everyone”.

Once you have completed the installation, open the GUI and edit the default configuration files so that they represent your typical setup:

- double click on the Icon for the TBSL1 Console to open it
- when the program opens it will load the configuration stored in the file C:/Users/Public/Public Documents/TBSL1 Configuration Tool/TBSL1-04.00.0a.bb/JsonFile/Settings.json
- from the menu select   Settings/TBSL1 Configuration
- go through the settings and set the values to suit your region, channel preferences, join type etc
- save the settings with   File Save
- write the settings to the Default config with File / Save As / Default Settings.Json.

7.3. Upgrading the TBSL1 Firmware

Firmware updates will be made available from time to time and will be available via TOIP’s FTP server and from Dropbox.

> **Note:**
>
> **After upgrading the firmware, make sure that you set the RTU time again.**

The firmware files are numbered in the format: aa.bb.cc.dd

<major build><production/debug><Release><version>

For example	04.00.01.33		
Where	04	major build	
	00	indicated the production build (01 = Debug)	
	01	Release	where 01 = LoRa/4G and
			03 = WiFi
	33	Version	

You should always install the aa.00.cc.dd version in the units

- the aa.01.cc.dd version is a debug build which includes additional logging and diagnostics
- the Debug version is not suitable for in field use as it will cause the RTU to draw much more power
- it should only be installed if you are specifically instructed to do so as part of tracking down a bug.

The TBSL1 firmware can be upgraded using two different methods:

- using the DFUse utility from ST Microelectronics
- using an ST-Link programmer

Minor upgrades can be performed through the Console or using an ST-Link programmer (Section 7.3.1 and 7.3.2). Major upgrades (e.g. upgrading to V04.00.01.33) which require an update to the Bootloader, can only be performed via the ST-Link programmer (Section 7.3.3)

7.3.1. Upgrading Via Console GUI

To upgrade the firmware you first need to download the “DFUse” utility from ST Micoelectronics:

https://www.st.com/en/development-tools/stsw-stm32080.html

After downloading the software, install it on your PC.

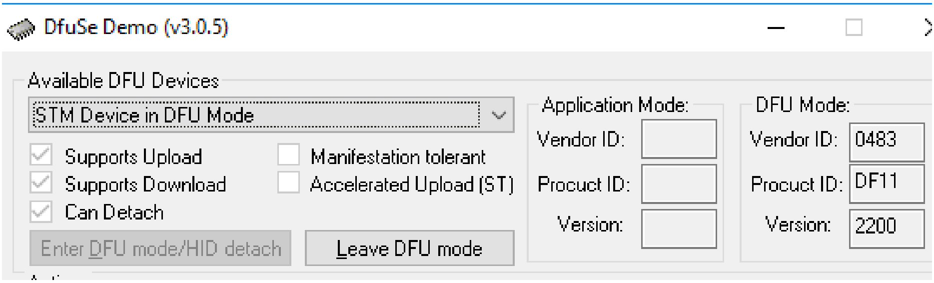
Next you will need to obtain a copy of the updated firmware. This can be obtained from TOIP and is usually supplied in conjunction with the GUI update. Save the firmware in a convenient location on your PC.

Performing the Upgrade:

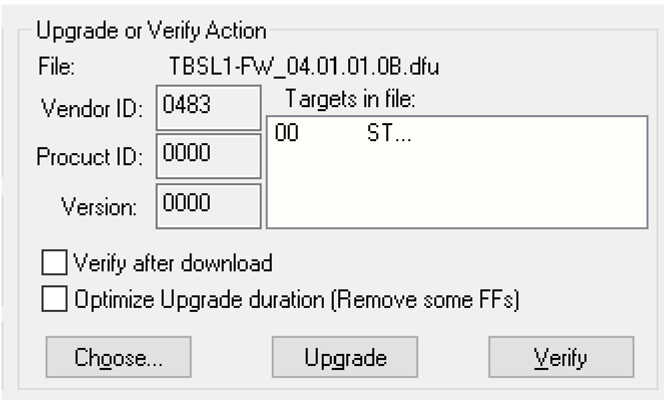
- connect the TBSL1 to your PC and open the GUI in Console mode
  - do not attempt the upgrade when in Logging mode as it will not succeed: you must confirm that the unit has switched to Console mode
- from the Menu select **FW Update / Update**
- you will receive a message indicating that the unit has been set to DFU Bootloader mode: this mode allows the unit’s firmware to be upgraded over the USB connection

```
<<12:30:30 Switching platform to DFU Bootloader mode
>>12:30:30 Switch to DFU Bootloader mode: OK
<<12:30:30 Closed COM port: OK
<<12:30:30 COM Port connection is not available during DFU Bootloader mode!!!
<<12:30:30 Please open DFUse Tool to perform firmware update!!!
<<12:30:30 Unplug and plug USB cable again to exit DFU Bootloader mode!!!
<<12:30:30 Detect removal of USB device
```

- once you see this message, open the **DF Use Demo** program
- the “Available DFU Devices” list box should show an “STM Device in DFU Mode”. If not, then the program has not correctly detected the TBSL1. If this happens, repeat the process
- In the lower right of the screen, you will see the **Upgrade or Verify Action**
- use the **Choose** button to select the file to load



- Click on **Upgrade** and then select the new firmware file from the location you saved it in the previous step
- now click on the **Upgrade** button and, when prompted, confirm that you wish to perform the upgrade
- a progress indicator will display showing the status of the update
- when the update completes, close the DFU program
- disconnect the USB Port and battery from the TBSL1
- wait 5 seconds, then reconnect the battery and USB port
- then close and re-open the GUI and verify that the GUI and firmware now have the same version.



7.3.2. Upgrading Via ST-Link

This option is only available if you have an ST-Link module and an appropriate header connector

- obtain the “hex” file for the upgrade
  - normally you will not need to update the Bootloader (the low level program which gives the microprocessor its basic functionality) only the operating program (which makes the unit a TBSL1)
- connect the ST-Link cable to the connector labelled J6 which is located below the bottom right corner of the battery (remove the battery to make the task easier)
- open the ST-Link program and load the hex file
- set the Memory Display Address to suit the file you are loading
  - for TBSL1 main program 0x08010000
  - for the Bootloader 0x080000000
- select Target / Program and then click on OK to start the update
- when the update has completed, remove the ST-Link cable
- the RTU will reboot with the new firmware.

Contact TOIP if you want to borrow an ST-Link programmer and cable.

7.3.3. Major Upgrades

Major upgrades (e.g upgrading to version 04.00.01.35) can only be performed using the ST-Link programmer. This is because the full chip memory must be erased prior to the upgrade and this can’t be done from the DFUseDemo program, only from ST-Link/Cube.

The sequence shown below must be strictly adhered to. Failure to do so could render the board inoperative:

- ensure the battery in the unit is fully charged
- connect a solar panel or dummy plug to the Solar Panel socket

- remove the battery holder from the mounting posts to give clear access to the programming socket
- connect the ST-Link programmer to the programming socket (J6): this is the 5 pin socket closest to the lower right brass post and has “J6” silk screened by its top left edge
- open the “STM32 ST-Link” software
- erase the CPU memory by selecting Target / Erase Chip and wait until the chip contents are cleared
- select File / Open and choose the new bootloader file e.g. TBSL1-BL_04.00.01.35.hex
- program the Bootloader by selecting Target / Program
- select File / Open and choose the new Application file e.g. TBSL1-FW_04.00.01.35.hex
- program the Application by selecting Target / Program.

When the firmware upgrade is complete, connect your PC to the TBSL1 via a Type 2 USB cable and open the GUI. Complete the following steps to finalise the upgrade:

- activate Manufacturer Mode by selecting Help / MFR_Admin
- when prompted, enter the security key:

  TBOX - FWTM - 1234 - ADMN

- if the key is accepted the screen will show a prompt “Manufacturer permission is enable”
- check the module versions by clicking on the Information tab and then selecting GET DEVICE INFORMATION. Confirm that it is correct

```
09:15:25 get device information ==> OK
09:15:25 Pulse Version: 01
09:15:25 Relay Version: 00
09:15:25 Power Version: 01
09:15:25 Analog Version: 01
09:15:25 SDI-12 Version: 0e
09:15:25 Camera Version: 00
09:15:25 Board Version:  00000101
09:15:25 Board Serial Nbr: 000004d2
09:15:25 Initialized FW Ver: 04000133
09:15:25 --------------- Getting versions on platform: DONE ---------------
```

The

Temperature calibration on the board must be set to provide the best accuracy for the internal clock:

- using an infra-red thermometer, make a measurement of the temperature of the processor (the large chip above the programming port)
- select Settings / TBSL1 Configuration / Info, then at the bottom of the window locate the ***MCU Temperature Calibration*** section
- type the temperature recorded by the Thermometer into the text box labelled ***Enter MCU Temperature*** and click on ***Start Calibration***
- wait while the TBSL1 performs the calibration and then take note of the value shown to the right of the calibrate button
    - if this does not match the reference temperature, press ***Start Calibration*** again
    - repeat until the two values agree
- close the config screen and from the main page select the ***INFORMATION*** tab. Then click on ***MCU Temperature*** and verify that the displayed temperature matches that from the thermometer. If not, repeat the measurement and calibration step.

7.4. Using the Debug Firmware

Each firmware build includes 3 hex files:

- TBSL1-BL_aa.bb.cc.dd.hex the bootloader: a low level file which is responsible for starting the RTU and checking for firmware updates
- TBSL1-FW_aa.00.cc.dd.hex the normal firmware: which is what is used on a day to day basis
- TBSL1-FW_aa.01.cc.dd.hex the Debug Firmware: this may be loaded for troubleshooting purposes and write additional information to the Logs

If you want to load the Debug firmware, perform the Upgrade using ST-Link, but instead of selecting the normal firmware file, select the file with the “01” mid way through its name

- when you load the GUI you will receive a nag message advising that the GUI and RTU firmware versions do not match
- the LED inside the RTU will also flash to signal that you are in Debug mode.

8. Diagnostics and Troubleshooting

8.1. TBSL1 GUI Log Files

The TBSL1 GUI program maintains a log file which can be used for diagnostic and trouble shooting purposes. A new log is created each time the program is opened.

The Log can be found in the directory:

C:\Users\<user name>\App Data\Local\TBSL1\Logs

The file will be named TBSLnConfigurationTool.txt

The log can be opened using a text editor such as Notepad ++.

To prevent the text editor regularly warning you that the file has been changed, make a copy of the file and then open the copy. If you need to later see more recent information in the log, simply repeat the process.

8.2. Diagnostic Mode – Error Log

The TBSL1 also provides a Debug or Diagnostic mode which can be selected from the console. The option is only available when the RTU is in ***Logging*** mode. Once selected, the RTU will remain in Debug mode until you next select Console.

To enable Diagnostic Mode from the console:

- below the CONNECT button you will see a check box labelled “Error Log”
- this is greyed out unless the RTU is logging
- with the RTU connected to your PC, click on the Logging button
- watch to see that the **Error Log** button is active
- click on the check box to activate it
    - the RTU will enable Diagnostic mode and resume logging
- the RTU’s error log will be written to the log file, which is stored in:

    C:\Users\Public\Public Documents\TBSL1 Configuration Tool\Logs\TBSLnConfigurationTool.txt

- you can open this file with a Text Editor (e.g. Notepad ++) to review the error messages
- To exit Diagnostic Mode, set the Operating Mode to Console

8.3. Clearing the RTU Memory

If a device is moved from one location to another of the configuration change significantly, you may need to delete the data stored in the unit’s memory to prevent it from being re-loaded.

To delete the stored data:

- connect to the RTU and select Console Mode
- from the menu click on the ***Stored Measurement*** tab

- click on the ***Clean All Parameters*** button
- wait while the data is deleted.

```
>>10:52:28 --------------Sending SDI-12 Sensor Parameter: DONE--
<<10:52:43 Send erasing measurement parameters:
<<10:52:45 Clear all battery parameters in bank A ==> OK
<<10:52:46 Clear all battery parameters in bank B ==> OK
<<10:52:49 Clear all pulse parameters in bank A ==> OK
<<10:52:52 Clear all pulse parameters in bank B ==> OK
<<10:52:56 Clear all analog parameters in bank A ==> OK
<<10:52:59 Clear all analog parameters in bank B ==> OK
<<10:53:10 Clear all SDI-12 parameters in bank A ==> OK
<<10:53:20 Clear all SDI-12 parameters in bank B ==> OK
```

8.4. Downloading Readings from Memory

The TBSL1 Console allows you to download the readings from the units memory to a text file for analysis. You can download the values for all sensor types or from a specific sensor type.

The downloaded readings will be saved to files in a sub directory called **Measurements** under the TBSL1 installation directory

e.g. C:\users\<user name>App Data\Local\Programs\TBSL1\Measurements

To download the data for all Sensors:

- connect to the TBSL1 from the GUI
- set the unit to ***Console Mode***
- from the menu select ***View / Get Sensor Measurement / Get All Sensor Measurements***

To download the data for a specific sensor:

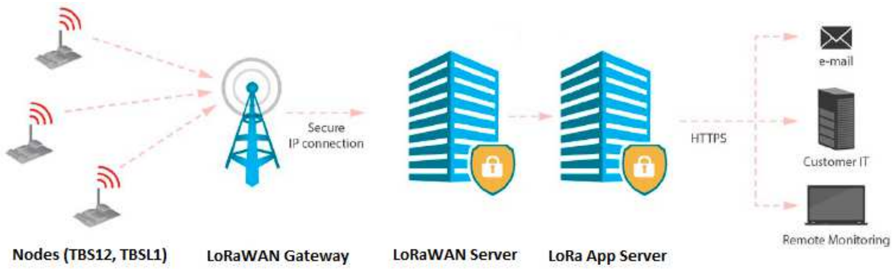
- connect to the TBSL1 from the GUI
- set the unit to ***Console Mode***
- from the menu select ***View / Get Sensor Measurement*** *and then select the measurement type:* battery, pulse, analogue, SDI

You can then open the file with a text editor and view the readings

- the files are saved with the suffix Measurement.txt and a prefix which indicates the measurement type e.g Sdi12.

9. Backend Setup

The figure below shows the major components in a LoRaWAN system.



In the LoRa WAN ecosystem, sensor readings take the following path:

- the LoRaWAN Node reads the sensors and builds the values in to a data packet
- the Node then transmits the values over its LoRaWAN radio module
- the transmission from the Node is processed by a LoRaWAN Gateway
- in the LoRa WAN system, Gateways are relatively primitive devices: after receiving a packet of data, they simply forward it on (the program running on the Gateway is thus referred to as a Packet Forwarder)
- as LoRaWAN is intended as a wide area network, it is possible that two or more Gateways may be in range of the Node
- each Gateway processes the packets of data and passes them on to a LoRa WAN Server
- the LoRaWAN server processes the LoRaWAN packets and send them on to the nominated LoRa App Server
- since it may receive the same data from two or more Gateways, the LoRaWAN Server picks the one with the best signal strength to on-forward and rejects the others

- the LoRa App Server receives the data and stores it in a temporary database
- the LoRa App Server will then make the data available via one or more standard formats (protocols) to an Application Server
- most LoRa App Servers support a version of an XML or JSON format for accessing the data
- many also provide support for the MQTT (Message Queuing Telemetry Transport) protocol, which is gaining wide use in the internet of things (IoT) sector
- the Application Server is the program responsible for displaying and analysing the readings.

Normally the LoRaWAN Server and LoRa App Server reside on Internet connected servers. In remote locations, the time taken for messages to travel back to the LoRaWAN server can cause issues with processing the data packets, so on some Gateways it is possible to run the LoRa WAN Server and LoRa App server on the Gateway.

The TBSL1 units have been tested with the following LoRa Suites:

- The Things Network (free service)
- Loriot (subscription service)
- Brocaar LoRa APP Server (Open Source).

To make it easier for older (legacy) systems to bring in data from a LoRa WAN system, Tekbox have implemented a LoRaWan Broker. The Broker collects the data from the LoRa App Server, stores it in a short term database and then makes it available via a number of standard formats to third party systems.

9.1. Configuration Requirements

When you add a new node to a LoRa WAN Server you will normally use the Node Creation function on the server to build all of the keys and IDs used by the node.

The list below shows the information you will need to obtain from the server prior to configuring the node:

- Activation Type (ABP or OTAA)
    - Activation By Personalisation: when you add devices to a network, they are registered in advance and to not need to “Join” again in order to transmit data
    - Over the Air Update (OTAA): devices will Join the network when the come on line. They send a Join Request, which must be acknowledged by the server
- each activation type requires different information to be written to the Nodes.
- ABP: for ABP devices you will require the following:
    - Device Address (Dev Add): a unique address used to identify a LoRaWAN node
    - Device EUI (Dev EUI): an identifier allocated to the device by the manufacturer. This is normally created to ease configuration: all devices with an identical configuration can be given the same Device EUI
    - Network Security Key (Nwk Sec Key): an encryption key used to encrypt attempts by devices to Join the network. It ensures any messages sent on the network are valid
    - Application Security Key (App Sec Key): this key is used to encrypt the data so that nobody else can view the contents of the messages
- OTAA: for OTAA devices you will require the following:
    - Application EUI (App EUI): this controls the LoRa APP server to which the data will be sent
    - Device EUI (Dev EUI): an identifier allocated to the device by the manufacturer (same as for ABP)

    - Application Key: the encryption key used for encrypting data in ABP mode.

- Acknowledgement Type (ACK or NACK)
    - Acknowledged Packets (ACK): in this mode, each time a packet of data is received by the Gateway it sends back an acknowledgement. This is done through the use of a pair of Counters: an Uplink Packet Counter and a Downlink Packet Counter. The counters are incremented each time a packet is sent. In Acknowledged Packets mode, the counter will not be incremented until the device at the other end sends its acknowledgement
    - No Acknowledgements (NACK): in this mode, there is no formal confirmation that a packet has been received. Instead, users can program a node to transmit the data multiple times. That way, if there is a collision (caused by two devices transmitting at the same time), one of the following transmissions may still succeed. NACK mode is simpler but not as reliable.

If you are setting up a Gateway, LoRaWAN Server and LoRa App Server, you may also need to determine the correct settings for the operating frequencies:

- Channel Plan: this is set at the country level. Most countries have allocated LoRaWAN frequencies in either the 868 or 900 MHz ranges
- Sub Bands: channel plans typically allow for up to 64 transmission channels. However since most Gateways only support 8 channels, the Nodes and Gateway must be configured to operate on the same Sub-Band (i.e. a sub-set of 8 channels)
- The nominal Channel Plan for Australia is referred to as AU915, with most networks using Sub Band 2 (Channels 8 to 15) which corresponds to frequencies of 916.8 to 918.2 MHz (channels are 0.2 MHz apart). The Things Network (TTN) was allowing AS923 to be used but has moved to standardising on AU915 SB2.

9.2. Passing Data to the Tekbox LoRa WAN Broker

One of the issues with working with LoRa WAN, is that not only does each device manufacturer decide on their own data format, each LoRa Server suite uses a different format with which to extract the node data. End users must thus write code for each type of device and server they need to support.

The TekBox LoRa WAN Broker has been written to simplify the task of extracting readings from LoRa WAN servers and works with all of the Tekbox LoRa WAN and cellular telemetry devices.

The Broker can be configured to regularly scan the LoRa Server for new data. It then stores the data in a temporary database and makes it available in industry standard formats.

The TBS12 units have been tested with the following LoRa Suites:

- The Things Network (free service)
- Loriot (subscription service)
- Brocaar LoRa APP Server (Open Source).

The Broker can make data available in the following formats:

- MQTT: clients can subscribe to the broker to pull down data
- FTP: the broker can periodically export the data in CSV format to a nominated server.

Support for XML and JSON formats is being added and should users require a custom format, it can also be supported.

For further details on using the Broker, refer to the Tekbox LoRaWAN Broker manual.

9.3. Accessing the Broker

Most users will not need to access the broker. Instead, devices will be added by your distributor or by TOIP.

URL	https://iot.tekbox.com
User Name	____________________
Password	____________________