

TOIP Pty
Ltd
Telemetry Over Internet Protocols

TBS12 Console



Version 1.05
February 2020

Copyright TOIP Pty Ltd

Table of Contents

1. Introduction.....	4
2. Software installation.....	5
3. Hardware setup.....	6
3.1. Connect COM Port.....	6
3.1.1. TBS12H (T shaped PCB for Otto SMP).....	6
3.1.2. TBS12S.....	7
3.1.3. TBS12B.....	7
3.1.4. TBS12 PC-FS.....	8
3.2. Install Batteries.....	8
3.2.1. TBS12B.....	8
3.2.2. TBS12S.....	9
3.2.3. TBS12H.....	9
3.2.4. TBS12 PC-FS.....	9
3.3. Sensor Connection.....	10
3.3.1. TBS12H.....	10
3.3.2. TBS12B and TBS12S.....	10
3.3.3. TBS12PC-FS.....	10
4. Connecting to the TBS12.....	12
4.1. Operation Mode.....	13
5. TBS12 Configuration Sequence.....	15
5.1. Board Information.....	16
5.2. Time Settings.....	17
5.3. Schedule Settings.....	17
5.4. LoRa Module Settings.....	18
5.4.1. LoRa RF.....	18
5.5. Sensor Configuration.....	19
5.5.1. TBS12 - SDI Commands.....	19
5.5.2. TBS12PC Pulse Inputs.....	20
5.5.3. TBS12FS Float Switch Inputs.....	21
5.6. Save Config for This Device.....	23
5.7. Writing a Configuration to the TBS12.....	23
5.8. Using the Direct Command Mode.....	24
5.8.1. LoRa WAN Commands.....	24
5.8.2. SDI-12 Sensor Testing.....	26

5.9. Checking Firmware Revision.....	26
1.1. Starting the Unit Logging.....	28
6. Other Functions.....	29
6.1. View Options.....	29
6.1.1. Clearing the Log.....	29
6.1.2. Displaying the Time/Date of events.....	29
6.1.3. Displaying Log Events.....	29
6.2. Reading Settings from TBS12.....	29
6.2.1. Displaying the Battery Voltage.....	29
6.2.2. Viewing the Device ID.....	30
6.2.3. Viewing the Firmware Version.....	30
6.2.4. Reading the Date-Time.....	30
6.2.5. Reading the LoRa Settings.....	30
6.2.6. Reading the Sensor Settings.....	30
6.2.7. Reading the Schedule Settings.....	30
7. TBS12 Data format.....	31
7.1. Battery data format.....	31
7.2. Sensor data format.....	31
7.2.1. TBS12S, TBS12B, TBS12SM.....	31
7.2.2. TBS12PC-FS Pulse Count Mode.....	32
7.2.3. TBS12PC-FS Flow Switch Mode.....	33
8. Backend Setup.....	34
8.1. Configuration Requirements.....	36
9. TBS12 Console Log Files.....	38
10. TBS12 Firmware Upgrade.....	39
10.1. Equipment Required.....	39
10.2. Preparation.....	40
10.2.1. TBS12B, TBS12S, TBS12PC.....	40
10.2.2. TBS12PC/FS.....	41
10.3. Firmware Update.....	41
11. Rising HF Modem Firmware Upgrade.....	43
11.1. Preparation.....	43
11.1.1. TBS12B & TBS12S.....	44
11.1.2. TBS12PC/FS.....	44
11.2. Upgrade Process.....	45

1. Introduction

The SDI12-LoRa Console is used to program the various versions of the TBS12, SDI-12 to LoRaWAN Bridge which are manufactured by Tekbox.

These include:

- TBS12H Head for Otto Soil Moisture Probe (Otto-LO)
- TBS12B Enclosure mounted unit with dry cell batteries
- TBS12S Enclosure mounted solar powered unit with rechargeable battery
- TBS12PC dual input pulse counter
- TBS12FS dual input flow switch (alarm)
- TBS12T LoRa WAN Temperature Tracker

The TBS12 must be connected to a computer via a USB to FTDI adaptor. Once connected the Console 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.

This manual is written around version 1.04 of the TBS12 firmware and Console program.



2. Software installation

To avoid issues with permissions, copy the Console application files to the “AppData/Local/Programs” directory for the current user:

C:/<user name>/AppData/Local/Programs/TBS12

After creating the folder and copying the files, make a new desktop shortcut, linked to the file “ConsoleApplication.exe”.

The TBS12 Console is written in the C# (C Sharp) programming language and utilises the Microsoft “.NET Framework 4”. This is supplied with the package (Newtonsoft.json.dll).

The application consists of 4 files plus an optional default file. During operation, it will automatically create a “Logs” directory to store logging information and errors.

Configuration files should be stored in the **JsonFile** directory.

Name	Date modified	Type	Size
 JsonFile	8/07/2019 4:47 PM	File folder	
 Logs	8/07/2019 4:35 PM	File folder	
 Configuration_Tool_User_Manual.pdf	8/07/2019 4:35 PM	Adobe Acrobat D...	395 KB
 ConsoleApplication.exe	8/07/2019 4:36 PM	Application	189 KB
 DefaultSetting.json	8/07/2019 4:55 PM	JSON File	3 KB
 Newtonsoft.Json.dll	8/07/2019 4:35 PM	Application extens...	478 KB
 Settings.json	9/07/2019 10:12 AM	JSON File	3 KB
 TracerX-Logger.dll	8/07/2019 4:35 PM	Application extens...	136 KB

3. Hardware setup

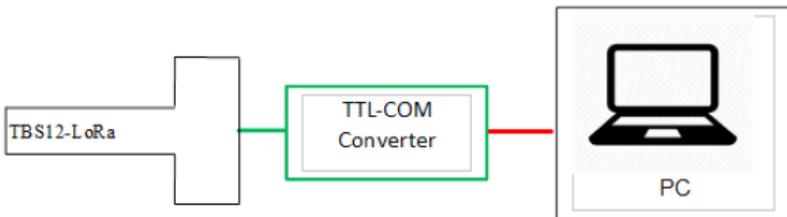
To ensure that the TBS 12 connects to the Console without you having to wait a full log cycle, connect the COM port prior to fitting the battery.

3.1. Connect COM Port

To keep the TBS12 design simple and to minimise power consumption, the serial interface on the unit utilises “TTL” signal levels. A level conversion circuit is thus needed to translate these signals to the USB format used on a PC. These may be called a USB to FTDI converter or USB to TTL converter.

All TBS12 variants provide a 3 pin JST socket for the communications / programming port.

Please ensure that your TBS12 unit has been supplied with the correct adaptor. If you are not sure, contact TOIP or your distributor. Most USB to FTDI adaptors utilise the FTDI driver which can be downloaded from:
<http://www.ftdichip.com/Drivers/VCP.htm>



3.1.1. TBS12H (T shaped PCB for Otto SMP)

Current TBS12SM boards have a JST socket fitted. If you have an old board with the 3 pin header, you will need an adaptor cable.

If you hold the PCB so that the pins on the header are facing you, the pin on the right is the ground pin.

The 3 pin header is not polarised: When you connect the plug from the USB Adaptor make sure it is lined up so the black mark is on the side with the dot on the PCB.



3.1.2. TBS12S

To open the TBS12S, turn the unit upside down and remove the 6 screws holding the cover to the base. Slowly turn the unit back over and collect the screws as they fall out.

Now remove the cover

- the cover is connected to the TBS12S PCB via the solar panel cable
- to remove the cover completely, unplug the 2 pin header (CON2)

The TBS12S PCB is fitted with a 3 pin JST header which is labelled CON3. If the unit is held with the antenna on top, the connector is 2/3 the way down the PCB on the right side.



The JST plug is polarised so it will only go in one way. Plug the header from your adaptor cable in to this socket and open the Console.

3.1.3. TBS12B

To access the programming port, undo the 4 Philips head screws holding the lid in place, then lift the cover off.

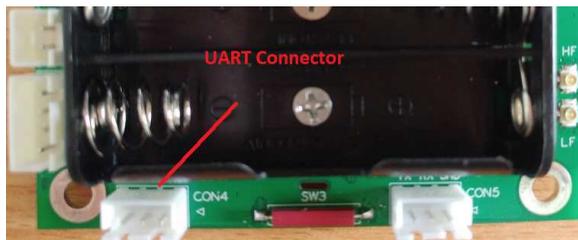
The TBS12S PCB is fitted with a 3 pin JST header which is labelled CON3. If the unit is held with the antenna on top, the connector is just below the bottom left corner of the battery holder.



The JST plug is polarised so it will only go in one way. Plug the header from your adaptor cable in to this socket and open the Console.

3.1.4. TBS12 PC-FS

The TBS12PC/FS is fitted with a 3 pin JST header which is labelled CON 4. This connector is located on the lower left of the PCB.



3.2. Install Batteries

3.2.1. TBS12B

The TBS12B is powered by a set of 2 off 1.5Volt alkaline D cells. Please ensure you use high quality cells (Energizer, Duracell) in order to obtain maximum battery life.

When inserting the cells into the holder, take care to align them according to the polarities shown on the holder.

3.2.2. TBS12S

The TBS12S battery is a type 18650 rechargeable Lithium Ion cell with a nominal voltage of 3.7V. It sits in a holder in the centre of the PCB.



To make it easier to remove the battery, a Puller can be made up using a piece of ribbon: cut an 80mm length of 8 to 12mm wide ribbon. Fold in half and glue the last 20mm of each end together with superglue. Once the glue is dry, you can sleeve the ribbon over the battery. The end of the ribbon can then be used to lift the battery out of the holder.

3.2.3. TBS12H

The battery for the TBS12H is fitted in to the slot cut into the PCB and secured with a zip tie. Plug the battery cable in to the matching socket on the PCB.

3.2.4. TBS12 PC-FS

The TBS12PC/FS may be powered using the on board batteries or an external supply

- if using the on board batteries, load 2 off Alkaline AA cells in to the holder
- if using an external supply (3.6V lithium cell or regulated 3.3V DC power supply) connect the supply to CON 6 on the top left of the PCB.

CAUTION: The Battery plug on the TBS12SM head for the Otto soil probe is wired reverse to that on the TBS12PC-FS. Please check the wiring before using any pre-wired batteries with these devices.

3.3. Sensor Connection

With the exception of the TBS12H, the sensors connect via an M14 socket.

3.3.1. TBS12H

The TBS12H (T-Head) connects to the Otto soil moisture sensors via 5 Pin JST connectors. For testing you should ensure that at least one sensor (on address 0) is connected.

3.3.2. TBS12B and TBS12S

The TBS12B and TBS12S connect to the SDI-12 sensors via a 7 pin waterproof M14 connector. The socket, which is on the bottom of the enclosure, is wired as follows:



This image is taken looking at the socket on the TBS12 body face on. If you plug the 7 pin plug into the socket and stand the unit on its head so the connector is on top, the orientation will be the same when you look down on the back of the plug from above (after taking off the backshell)

You can test connectivity to and functionality of, the sensors using the Direct Command mode (Refer section 5.8.2).

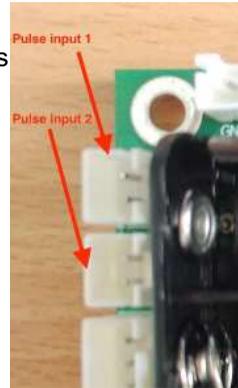
3.3.3. TBS12PC-FS

The TBS12 PC-FS is used with switches which provide a closed contact when they are active. These may be reed switches fitted to a rain gauge or flow meter, or switches fitted to an alarm contact or float switch. In all cases, the switch must make two contacts available: a Common (which is wired to the ground terminal on the inputs) and the Switch input. When the

switch operates (i.e. the contacts close, the input line is connected to the ground terminal).

With the TBS12PC (pulse count) module, the switch closes for a brief period (typically 20 mSec). Whereas with the TBS12FS, the switch may be closed for minutes or hours.

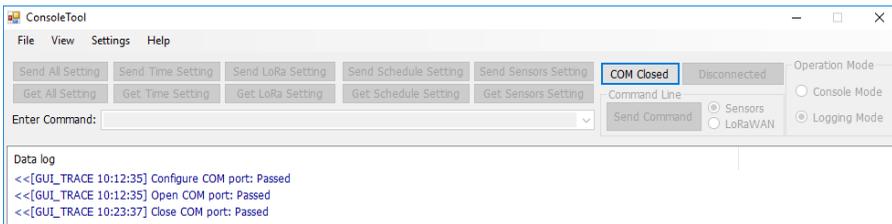
The sensors on the module connect via the two pin headers labelled CON 2 (input 1) and CON 6 (Input 2). These are located on the upper left side of the PCB.



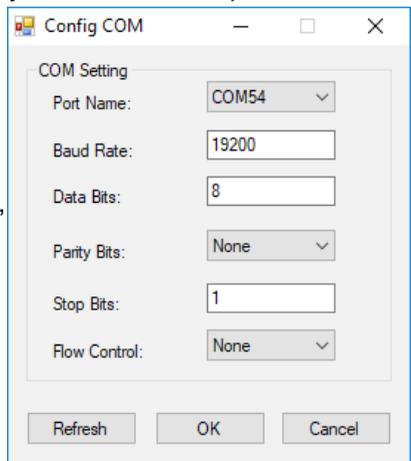
4. Connecting to the TBS12

To operate the TBS12 and Console together:

- first connect the serial to USB adaptor as shown in Section 3
- open the Console by clicking on the desktop shortcut you created in Section 2
- menu functions which send commands to the device will be greyed out at this stage
- those functions that read or write to a configuration file will be active as will those to connect to the device
- the buttons to control communications show the current state i.e. “COM Closed”. Click on the button to Open the communications port
 - the label will change to COM Open



- from the menu select **Settings / COM Port**
- when the **Config Com** menu displays, click on the drop down list box labelled **Port Name** and choose the port allocated by Windows to your FTDI to USB adaptor
 - if you can't see the port, click on the **Refresh** button
 - if you are still having problems, open Windows Device Manager, then check to see if a new COM port appears as you plug and unplug the adaptor. If you do not see the adaptor, check that the driver is installed and that Windows is not blocking the Driver



- click on the **OK** button
- then, at the Main Console screen, click on the button labelled **COM Closed**
 - this opens the connection to the USB-TTL adaptor
 - the log screen should display the text *Open Com Port : Passed*
 - click on the **Disconnected** button to Connect to the TBS12
 - the button label will change to “Connecting” while the Console attempts to communicate with the TBS12
 - the label will change to Connected once successful.

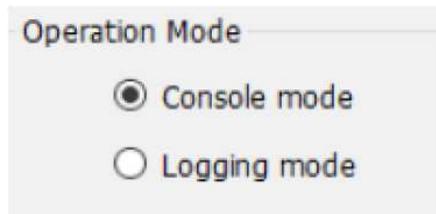
Connection Sequence:

- the two buttons manage the connection firstly to the USB-TTL adaptor and secondly to the TBS12
- during a normal connection sequence, you will move the unit through the following states:

		COM	CONNECTION
COM Closed	Disconnected	Port closed	
COM Open	Disconnected	Port Open	Not connected to TBS12
COM Open	Connecting	Port Open	Waiting for TBS12 to respond
COM Open	Connected	Port Open	Connected to TBS12

- once the Port is Open and the TBS12 connected, the Operation Mode buttons will be activated

4.1. Operation Mode



The TBS12 has two operating modes:

- **Console mode:** this mode is used when you configure and test the TBS12. Any changes you make to the configuration and send to the device will not be activated until it is set to logging mode
- **Logging mode:** when in this mode the TBS12 will automatically read the attached sensors and transmit the values to the LoRa Server. In between readings the unit goes in to sleep mode.

Once the TBS12 is in logging mode, it will regularly check to see if a USB-TTL adaptor is plugged in to the COMMS port. If so it will activate the port and check for commands. The unit should thus respond within 20 or 30 seconds to a change back to Console mode.

If the unit does not respond, remove the battery and restart it:

- The TBS12 units have large storage capacitors on the PCB which will retain power for a while after the battery is removed. To ensure the unit restarts:
 - connect the USB to TTL converter and Open the Console
 - remove the battery
 - short the terminals on the battery holder on the PCB for 1 second
 - replace the battery and wait for a message on the Console.

5. TBS12 Configuration Sequence

When setting up a new TBS12, you need to perform the following steps:

- open the TBS12 Console program
- connect the Serial to USB Converter to your PC and identify which COM port it is allocated (via Device Manager)
- connect the other end of the USB converter to the TBS12
- from the TBS12 Menu select **Settings / Com Port** and then set the Port Name to match the port you identified for the converter
 - if the converter was recognised before you start the Console, the port should show in the drop down list
 - leave the other parameters at the default setting then click on OK
- click on the **COM Closed** button to connect to the USB-TTL adaptor: the button label will change to **COM Open**
- apply power to the TBS12 (i.e. plug in the battery connector)
- then click on the **Disconnected** button to connect to the TBS12
 - depending on the settings of the board, it may take a while for the board to wake
 - If you don't want to wait, select Com Open again to close the connection, remove the battery and follow the process shown earlier to restart the unit
- check the **Operation Mode** displayed on the upper right of the screen
 - if it is showing **Logging Mode** click on the **Console Mode** radio button
 - wait while the unit connects and changes modes
 - a message will display on the console to indicate that the device has connected.

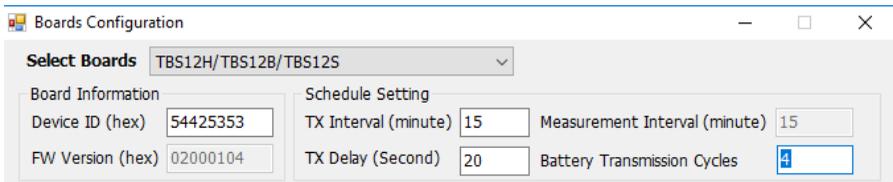
Note:

It is possible for the Console and TBS12 to get out of step if the Console thinks the board is in one mode when it is really in the other. This will be evident as an inability to connect to the TBS12. If this occurs, try the following:

- **make sure the Console open and connected,**
- **remove the RTU battery**
- **toggle the Mode setting from Console to Logging or Vice Versa**
- **replace the RTU battery and allow the unit to re-start**
- **you may need to try this a couple of times to get the two back in step with one another.**

Once the connection to the TBS12 is established, you can create a configuration file for your device

- to start with a default configuration, click on File / Load Default Settings
- if you have an existing file you want to open or to use as the template for the new device, select File / Load and then choose the desired file
- from the TBS12 Menu select **Settings / Boards**
- the Board configuration screen will display
- from the Select Boards list box, choose the TBS12 type you are working with e.g. TBS12H/B/S
- you can the progress to setting up the various parameters.



5.1. Board Information

The Board Information area shows you the ID number allocated to the unit and the firmware version

- Device ID: this is a hexadecimal number used to uniquely identify the device
 - this can be used to track the location and history of a PCB
 - the Device ID should be set when the device is new and then remain with the device even if it is moved from one location to another
- FW Version: this shows the version of the firmware loaded in to the board. You can also fetch this from the menu using View / Get Firmware Version
 - the TBS12 firmware version must always match the Console firmware version.

5.2. Time Settings

All of the LoRa WAN nodes should be operated with the time set to UTC (Universal Coordinated Time, also often called GMT). This matches with the settings in the Sensori platform and avoids issues with having to change the time when daylight savings starts and ends (which always causes an hour data gap). When the data is viewed, a time offset will be applied according to the time zone set for the location in which the unit is installed.

To set the time:

- from the **Boards Configuration** screen, click on the **Time Zone** tab
- from the Time Zone Selection list box, select **UTC – Universal Coordinated Time**
- to set the time in the board, click on the **Set Time for External RTC** button.

5.3. Schedule Settings

These control how often measurements are made and transmitted. Normally the Measurement Interval and Transmit Interval will be set to the same value

- **Transmit Interval:** set this to how often you want the values to be sent. Normally this is the same as the Measurement Interval e.g. 15 min for soil moisture and weather, 8 hours for groundwater
- **Measurement Interval:** set to 15 or 30 minutes for soil moisture sensors
 - on the TBS12PC-FS the measurement interval must equal the Transmit Interval
- **Transmission Delay:** adds a delay after the read is completed before the unit sends its data
 - within the LoRaWAN ecospace, it is assumed that nodes will wake at random and send values. This approach is fine when nodes only transmit a few packets once a day
 - But in our applications (with lots of data and frequent transmission) we need to manage transmissions so that they don't all occur at once (collisions)
 - First estimate how long it will take for the node to read the sensors (e.g. 10 seconds), then allow another 10 second buffer and start to add your Nodes with a 10 or 20 second offset: first

Node Delay = 30, 2nd Node Delay = 40, 3rd Node Delay = 50 etc. You may need to maintain a separate list of what Delay you allocate to each node (if you save each configuration file you can easily recall the settings at a later time)

- Battery Transmission Cycles: to save air time, you can reduce the rate at which the battery information is sent. For example set this to every 4 cycles (2 hours for 30 min log, 1 hour for 15 min log)

5.4. LoRa Module Settings

You now need to set up the LoRaWAN parameters to match the network you are connecting to. This process begins at the level of the LoRaWAN Application Server. Add each node on the server and record all of the relevant details, then transfer them to the TBS12 Console:

- LoRaWAN Mode: In closed (private) networks, you will normally use ABP but if you are using a public network, the provider may stipulate the use of OTAA. The Network settings you need to program will change depending on whether you are using ABP (Activation by personalisation) or OTAA (Over the air activation)
 - ABP: uses NWKSKY and APPSKY
 - OTAA: uses APPLICATIONEUI and APPLICATIONKEY
- Device Address, Device EUI, NwkSKey, AppSKey, ApplicationKey: set to value from LoRaApp Server
 - copy and paste these exactly as they appear in the backend LoRa Server
- ACK Options: there are two approaches to making sure transmissions get through
 - The simple approach is to send them multiple times (No Ack from Server). This can waste a lot of air time: if the first packet is received every time, sending it a 2nd or 3rd time is a waste of time and power
 - The more complex (and more robust) approach is to wait for the server to acknowledge or confirm that it has received each packet. The Node and Gateway then maintain a Packet Counter which is incremented each time a packet is sent
 - Acknowledged Packets should be used on all soil moisture and weather monitoring stations i.e. enable the checkbox labelled "Wait ACK from server"

5.4.1. LoRa RF

In this area you set up all of the radio parameters to match the country in which you are operating

- ADR: the adaptive data rate option allows nodes closer to a Gateway to transmit at a higher speed than nodes further away. This helps maximise the time window available for nodes to transmit. But the feature is most benefit in large public networks. The default should be to set ADR OFF
 - the major limitation of ADR is that it reduces the data rate to low values when signal strength is poor (i.e. a node is a long way from the Gateway). But under the LoRa Protocol, the low data rates do not allow long packets, so those from multi-sensor soil moisture probes will be blocked. So with these devices you are limited to data rates of 4 and above
 - so if you are planning on using ADR, make sure that you set the "DR MIN" to a value of 4 and the DR MAX to a value of 5
- PORT: this defines the port used by the modem internally for LoRa
 - it must be set to a value of 8
- TX POWER: sets the maximum transmitter power: set to 14
- DR: this setting controls the default data rate used when sending the packets
 - Slower data rates will provide longer range but tie up the network for longer. The default should be to use a DR of 5 and you can then try 4 and 3 if you want to get more range
- FREQ SCHEME: click on the drop down list and choose the AU915 entry, which will set the broad channel plan to match the Australian 915 to 928 MHz scheme
- SET CH: now click on the SET CH button to view a list of the current channel settings
 - The default for Australia is sub-band 1, with the channels set on 0.2MHz increments from 916.8 to 918.2
 - If the network in your area is congested, you may wish to choose one of the other sub-bands (this will have to be set in the Gateway and on each node).

5.5. Sensor Configuration

Providing you have selected the correct Board when you began this process, the Sensor tabs will show the options available for your TBS12: SDI on the TBS12SM/B/S, Pulse for the TBS12PC and Float Switch for the TBS12FS.

5.5.1. TBS12 - SDI Commands

In the SDI Commands area, you add details of the SDI-12 commands used by the various sensors connected to the TBS12. A separate data packet is transmitted for each Ordinal or line added to the list.

SDI-12 Config		
Pulse Counter Config		
Float Switch Config		
SDI12 Commands		
	Ordinal	Measurement Command
	1	0C!
	2	0C2
	3	1C1!
	4	
	5	
	6	
	7	

Add a line for each individual measurement command you will be using.

- for an EnviroPro or Otto probe you will use 0C! for the soil moisture and 0C2! for soil temperature. If you are using an EnviroPro EC, you add 0C! for the EC
- for a 3 sensor “Naked” or Otto LO probe, you will need 3 command entries in the table: 0C1!, 1C1!, 2C1!
- When you have finished adding the settings, click on the SAVE button.

Take note of the order in which you add your sensors as this must be reproduced in the same format when you add the device in the Tekbox IoT Broker

- the TBS12 firmware reviews the commands which have been entered and uses the information to assign the Sensor ID and Sub Sensor ID
- sensors on the first address (e.g. 0) are sent with Sensor ID 0
 - the first measurement command (0C) which is shown as Ordinal 1 in the example above, will be sent as Sub Sensor ID 0
 - the 2nd command (ordinal 1 in the example) as Sub Sensor ID 1
- the first sensor on the next address (ordinal 3 in the example) will be sent as Sensor ID 1, Sub Sensor ID 0.

This information will also help you if you are monitoring the Data Packets as they are received in the Application Server on TTN / Lorient.

5.5.2. TBS12PC Pulse Inputs

If you are working with a TBS12PC, click on the Pulse Counter Config tab.

The screenshot shows the 'Pulse Counter Config' tab with the following configuration:

- Pulse Number [0:2]: 2
- Pulse Channel: 0
- Pulse Parameters
 - Pulse Types: Rain Gauge
 - Unit Per Pulse: 0.2
 - Starting Totaliser: 0

When the Pulse Counter Config displays, edit the configuration as follows:

- Pulse Number: set this value to the number of pulse sensors you will have connected: either 1 or 2
- Pulse Channel : once you have told the Console the number of sensors you have connected, you must select each one in turn and configure it
 - select the first channel (Pulse Channel 0)
 - Pulse Types : in the Pulse Types list box, select whether the sensor is a Rain Gauge or Flow meter
 - Units per Pulse : set the Units per Pulse to match the sensor e.g. 0.2mm per tip for a rain gauge; 100 litres per pulse for a flow meter
 - Starting Totaliser: if you are working with a flow meter which has a totaliser display on it, set the Starting Totaliser value to the value on the meter's display.

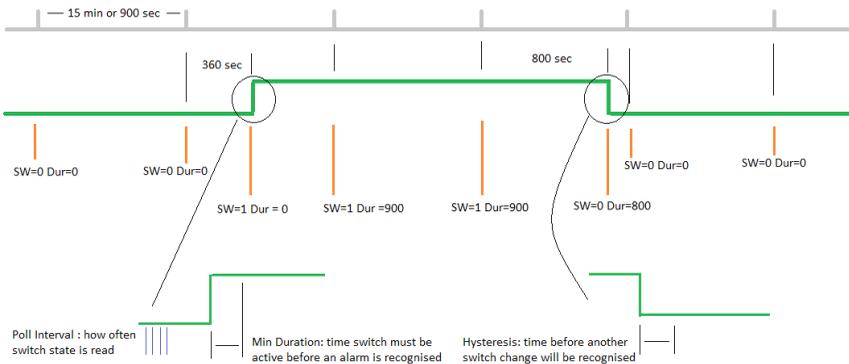
5.5.3. TBS12FS Float Switch Inputs

The TBS12FS can accommodate up to 2 switch inputs. Each can be configured separately.

The unit will send a packet out under 3 different circumstances:

- at the end of each log period
 - here the unit sends the switch state and the duration
 - if the switch is off it will send a state of 0
 - if the switch is on it will send a state of 1

- if the switch was OFF for the whole interval, the duration will equal 0
- if the switch was ON for the whole interval, the duration will equal the Log Interval (e.g. 900 for 15 min interval)
- when the switch turns on
 - a packet will be sent with the new switch state i.e. 1
 - the duration will be 0
- when the switch turns off again
 - the state will be 0
 - the duration will be set to the number of seconds since the start of the log period before the switch released. For instance if the switch releases 6 minutes in to the log period, the duration will be sent as 360.



SDI-12 Config
Pulse Counter Config
Float Switch Config

Float Switch

Number of Float Switches [0:2]

Float Switch Index: ▼

Parameters

Polls Interval (seconds): [10 : 300]

Min Duration (seconds): [1 : 300]

Hysteresis Time (seconds): [1 : 300]

Send Release Packet

When configuring a TBS12FS, select the Float Switch Config tab.

You can then set the unit up according to how you wish to use the switches:

- Number of Float Switches: select 1 or 2
- Float Switch Index: select each sensor in turn and then set the remaining parameters for each
 - Poll Interval: this setting controls how often the board will sample the input. A good guide is to set this to half what you expect will be the shortest switch duration e.g. if your switch will stay on for a minimum of 30 sec, set the Poll Interval to 15 sec
 - Min Duration: set this setting to the minimum time the switch contacts need to be closed in order to trigger the system
 - Hysteresis Time: set this to the number of seconds you want to wait after the switch is released before a new alarm can be recognised
- Send Release Packet: the TBS12FS will send an alarm packet when the input triggers. If you activate the “Send Release Packet” check box, an additional packet will be sent when the switch opens again.

5.6. Save Config for This Device

Once you have completed all of the setup, you should save the settings in a new config file. To do this, from the menu select **File / Save As**

- then choose both the location to where you want to save the file (the default is fine) and the name
- the file name can be set to match the Device Address or the Name you allocate to the site
- once you have a standardised configuration, you can use it as a template for each new device.

5.7. Writing a Configuration to the TBS12

Now that the configuration is complete, you need to write it to the TBS12

- make sure that the TBS12 is in Console mode and that it has connected to the Console
 - if you don't want to wait for the unit to poll, you can cycle the power as described earlier
- click on the **SEND ALL SETTINGS** button
 - the Console will send the settings to the TBS12
 - if the Console can send the commands, you will see a message in the Data Log section showing that each command was successful

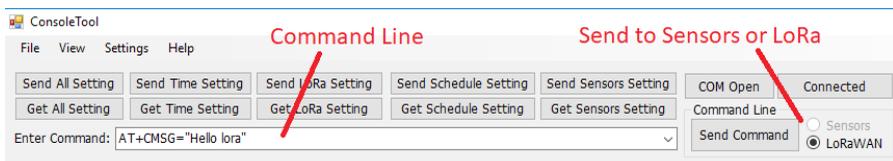
- if you receive an error with every command, the TBS12 is not yet active (wait or cycle the power and retry)
- if you receive an error on 1 or 2 commands, the Console and TBS12 may be set to different firmware versions. Contact us to organise an update to the Console and / or TBS12 firmware
- later, if you edit individual settings, you can send each type of items separately by clicking on the appropriate button.

WARNING

- The configuration will only be Activated on the TBS12 once the unit is set to **Logging Mode**
- If you do not switch to Logging (Run) Mode, the unit will retain its old settings
- If you need to do further testing, set the unit to Logging Mode, wait while the changes are activated and then go back to Console Mode
- if the battery has been removed, the unit's time will be lost: you must reset the time before going to Logging Mode. This can be done by selecting **Send Time Settings**.

5.8. Using the Direct Command Mode

The Direct Command function allows you to send commands directly to the attached SDI-12 sensors (TBS12SM/S/B) or to the LoRa WAN Modem (all units).



5.8.1. LoRa WAN Commands

To send commands to the LoRa WAN Modem:

- in the Command Line radio buttons, activate the **LoraWAN** button
- now type the command you wish to send in to the Command Line
 - e.g to send a confirmed test message type `AT+CMSSG="Hello"`
 - to check the modem version type `AT+VER`

- then press on the Send Command button or press Enter
 - the command will be sent to the modem
 - after a delay the result will be displayed in the Log screen
 - outgoing commands are shown prefixed with ">>"
 - responses from the modem are prefixed with prefixed with "<<".

To check the LoRaWAN Join mode

AT+MODE=?

- The unit should return either LWABP (ABP) or LWAOTAA (OTAA)

To check the Node settings

AT+ID

To check the APP EUI (Application EUI)

AT+ID=APPEUI

Network Session Key (NWKSKEY) and Application Session Key (APPSKEY)

- the commands for both keys are set only
- for security reasons, you can not send a command to check either
- the Network Session Key or Application Session Key

To send a Link Check Message

AT+MSG

- the Link Check message is a special form of message which is sent with no payload
- when a Link Check message is received by the LoRa Server, it will always be acknowledged
- The Link Check message can thus be used to check your signal strength
- the TBS12 units send regular Link Check messages to ensure that the transmit path is functional before sending data.

To send a confirmed test message

AT+CMSG="abcdefg..."

- when a confirmed message is sent, the receiving Gateway will send a message acknowledging receipt of the packet
- after sending the message monitor traffic on the Gateway to see if it has been received.
e.g. AT+CMSG="11 22 33 AA BB FF"

To send an unconfirmed message

AT+MSG="abcdefg"

- in a system using unconfirmed messages, messages are sent multiple times to ensure they get through
- after sending the message monitor traffic on the Gateway to see if it has been received.
e.g. AT+MSG="Hello"

To send an unconfirmed message in hex format

AT+MSGHEX="xx yy zz"

- in a system using unconfirmed messages, messages are sent multiple times to ensure they get through
- after sending the message monitor traffic on the Gateway to see if it has been received.

e.g. AT+MSGHEX="11 22 33 AA BB FF"

5.8.2. SDI-12 Sensor Testing

You can also use the Console to send command to the attached SDI-12 sensors.

To send commands to the SDI-12 sensors:

- in the Command Line radio buttons, activate the **Sensors** button
- now type the command you wish to send in to the Command Line
 - to make a test measurement a sensor on address 2, type
2C!
And press enter
 - to read the command result type 2D!
" "
 - to retrieve the Information string for a sensor on address 0 type
0!

5.9. Checking Firmware Revision

The Console and TBS12 need to be maintained at the same firmware version. Using different versions can lead to incompatibilities in the configuration files which could lead to problems in operating the devices.

If the versions do not match you will receive a warning message from the Console.

To check the Console and Firmware Version:

- from the menu select **Help / About**
- the Application Version and Device Firmware version will display in the help window

Rising HF also release updates to the firmware in the Modem module used on the devices. To check the Modem firmware version:

- activate the Send Commands to Lora radio button
- in the command field type **AT+VER**
- the version will be displayed
- this should be version 2.1.19 or later
- contact TOIP for assistance with updating the firmware.

1.1. Starting the Unit Logging

When you have finished your configuration, set the Console to Logging Mode. Before you do this you may wish to perform a few final checks:

- using the Direct Command Mode, use an "I" command to check that the sensor is connected
- using the Direct Command Mode, send a test message AT+MSG and check that an acknowledgement is returned
 - you should also be able to see the message and its reply on the Data / Traffic monitor on the LoRa Server
- when you are comfortable with these tests, set the unit to Logging.

Once the unit is set to Logging mode, it will activate the configuration and send a Link Check Message.

You can then remove the USB-TTL adaptor and fit the cover to the case.

6. Other Functions

The TBS12 Console supports a number of other functions to help you configure and test your Nodes.

6.1. View Options

You can control what information is available in the Log screen in the Console.

6.1.1. Clearing the Log

To clear the log screen select **View / Clear Log**

6.1.2. Displaying the Time/Date of events

When information sent to or received from the Node is shown in the Log screen, you have the option of controlling whether the Date & time are also shown.

To display the Date and Time in the log screen select **View / Toggle Tag Time**

- This function acts as a Toggle: each time you select it the mode will be reversed.

6.1.3. Displaying Log Events

The Console can also be set to display additional log information. Logs are also written to a log file which can be read using a Text Editor such as Notepad++.

To display the event information select **View / Toggle Show Logs**

- This function acts as a Toggle: each time you select it the mode will be reversed.

6.2. Reading Settings from TBS12

From the View Menu, you can also read the values of the various settings on the TBS12. This function is only available if you are connected and the Console is on Console mode.

6.2.1. Displaying the Battery Voltage

To read the TBS12 battery voltage select **View / Get Battery**

6.2.2. Viewing the Device ID

Each TBS12 PCB is allocated a unique device ID. This can be used to manage your inventory and repair records. The Device ID can be altered by setting a new value in the configuration.

To display the Device ID select **View / Get Device IDs**

6.2.3. Viewing the Firmware Version

Once connected you can read the Firmware version from the TBS12. This should match the version displayed in the Console:

To display the TBS12 firmware version **View / Get FW Version**

To display the Console version **Help / About**

6.2.4. Reading the Date-Time

To read the time from the TBS12 select **View / Get Time**

- the time will be displayed as below on the Console
- the format of the display is:
YYYY MM dd HH mm ss (year month day hour minute second)

6.2.5. Reading the LoRa Settings

You can read the LoRa WAN settings from the TBS12 from the Console menu by selecting: **View / Get Lora Settings**

6.2.6. Reading the Sensor Settings

To read the current sensor configuration from the TBS12 select **View / Get Sensor Settings**

6.2.7. Reading the Schedule Settings

To read the current schedule settings from the TBS, select **View / Get Schedule Settings**

7. TBS12 Data format

To help keep the individual LoRaWAN transmissions (packets) a manageable size, two things are done by the TBS12:

- firstly, each SDI-12 measurement (combination of Address + Measurement number) is sent as a separate packet
- secondly, diagnostic information (battery voltage etc) is sent in a separate packet. The transmission interval for the diagnostic data can be set independently of that used by the logged data.

7.1. Battery data format

Format PB:yy:mm:dd:hh:mm:ss<space>aa.bbb

Data format for battery consists of the following:

- a PB header
- the timestamp (*YYYY MM dd HH mm ss - year month day hour minute second*)
- the battery voltage.

PB18:06:01:17:00:00 3.600

5:00 pm 6/01/2017 3.6V

7.2. Sensor data format

Sensor data is always identified with a P or parameter character and then a second character which identifies the sensor type.

7.2.1. TBS12S, TBS12B, TBS12SM

The S, B and SM variants all support SDI-12 readings and the packets are identified with a P for parameter then an S to indicate SDI-12

Format PS:yy:mm:dd:hh:mm:ss<space><Index><No. Values><space><sign>aa.bbb

The sensor (tag) data consists of the following:

- the PS header
- the timestamp (*YY MM dd HH mm ss - year month day hour minute second*)

- the index of the SDI-12 sensor (0, 1, 2 etc)
 - the first entry in the SDI-12 command list (Ordinal 0) will be Sensor ID 0
 - if all subsequent entries (Ordinal 1, 2 etc) are on the same address they will all have Sensor ID 0
 - if a command (Ordinal) is entered with a different address, it will become Sensor ID 1
- the sub sensor ID (0, 1, 2 etc)
 - the first command (Ordinal 0) will be allocated Sensor ID and Sub Sensor 0
 - if the 2nd command is for the same address it will be Sensor ID 0, Sub Sensor 1
- the number of values which follow (01 to 99)
- the sensor / tag value.
 - If there are multiple measurements, they are separated with a space character

PS18:06:01:16:25:003202 +37.37 +29.65

- Time stamp: 1rst of June 2018 (date) 16:15:00 (time)
- '3': Sensor_ID is 3
- '2': SubSensor_ID is 2
- '02': 2 SDI-12 measurements returned by the sensor
- SDI-12 measurements: +37.37 and +29.65

7.2.2. TBS12PC-FS Pulse Count Mode

TBS12 PC-FS pulse count data is sent with a P or Pulse identifier.

PP18:06:01:17:00012 15 350

Format PP:yy:mm:dd:hh:mm:ss<SID><SSID>n pr tot

PP	header
YYYY MM dd HH mm ss	timestamp
SID	Sensor ID 0 = P1 1 = P2
SSID (Mode)	Sub Sensor ID 1 = Pulse Count mode
n	number of parameters (always 2)
pr	precipitation or elapsed flow
tot	flow totaliser

7.2.3. TBS12PC-FS Flow Switch Mode

When the TBS12 is used in Flow Switch Mode, the packet is still identified with a P type

PP18:06:01:17:00011 1 90

Format **PP:yy:mm:dd:hh:mm:ss<SID><SSID>n a dur**

PP	header
YYYY MM dd HH mm ss	timestamp
SID	Sensor ID 0 = P1 1 = P2
SSID (Mode)	Sub Sensor ID 2 = Flow switch mode
n	number of parameters (always 2)
a	alarm state
dur	alarm duration in seconds

8. Backend Setup

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

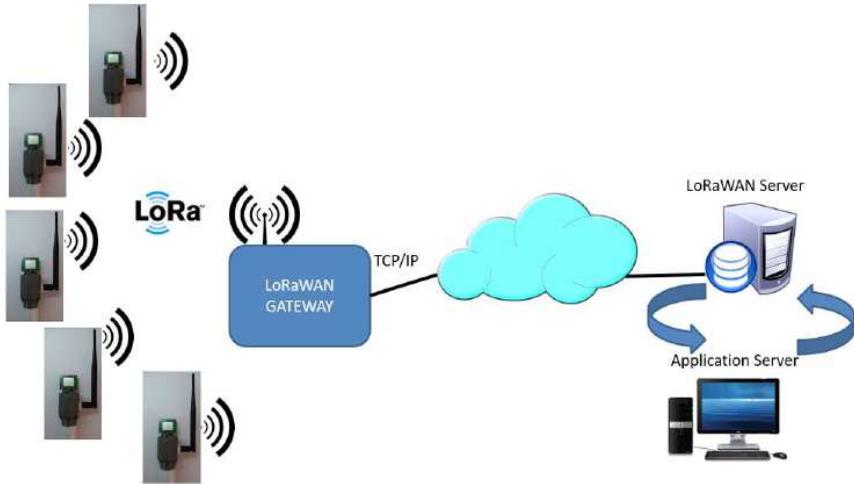


Fig 23. LoRa WAN Environment

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 TBS12 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. The Broker currently supports the LoRa Suites shown above and 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.

8.1. 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 do not need to “Join” again in order to transmit data
 - Over the Air Update (OTAA): devices will Join the network when they 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 1 (Channels 8 to 15) which corresponds to frequencies of 916.8 to 918.2 MHz (channels are 0.2 MHz apart)
 - some providers however are promoting the use of the Asian sub band, referred to as AS923. This is not for any technical reason and simply allows them to sell the same equipment in Australia and New Zealand as they sell in Asia.

9. TBS12 Console Log Files

The TBSL1 Console 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\\App Data\Local\TBS12\Logs

The file will be named TBS12_Console.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.

10. TBS12 Firmware Upgrade

The firmware in the TBS12 will be upgraded from time to time to add features or to fix problems. To load the updated firmware you must have access to an ST-Link programming module and a 4 pin JST header wired to suit the TBS12. These are available through TOIP.

10.1. Equipment Required

To upgrade the TBS12 you will require the following:

- one or more TBS12 units fitted with a battery
- an ST Microelectronics ST-Link programmer fitted with an adaptor cable with plug for the TBS12 (4 pin). This will be supplied to you by TOIP
- a copy of the STM32 ST-Link program which can be downloaded from the ST Microelectronics web site – you need to register on the site to complete the download
- copies of the TBS12 firmware
- a PC with two available USB ports
 1. one which will connect via the Type B USB cable to the TBS12
 2. a second which will connect to the cable on the ST-Link programmer.

If you are making your own adaptor cable for the TBS12 units, you can make the header for the TBS12B/S/PC using either a 4 pin 2.54mm (0.1”) spacing JST connector or a 2.54mm spacing PCB header.

Wire the socket/header as follows:

Function	ST-Link	Header
Ground	5	1
Reset	1	2
SWIM	3	3
VCC (3V)	7	4

- Programming mode Set to SWIM (Single Wire Interface Mode)
- Device Set to STM8L15XC8

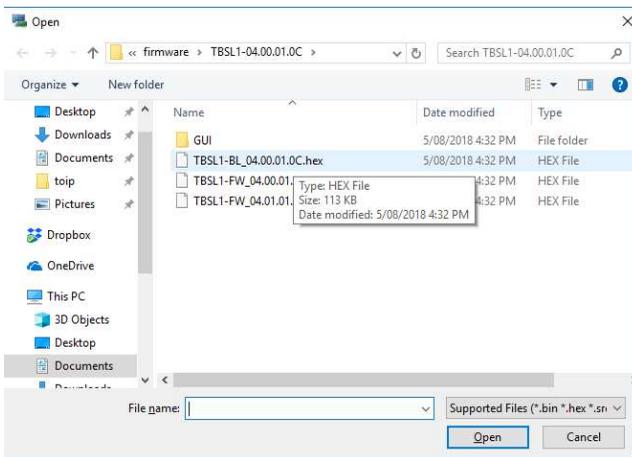
10.2.2. TBS12PC/FS

The programmer connects via CON 1, a 4 pin header on the lower left edge of the PCB.

10.3. Firmware Update

To update the RTU firmware:

- from the menu select File / Open
- now locate the folder in which you copied the TBS12 firmware
- select the file TBS12-x_nn.nn.hex
 - make sure that the “x” matches the model number of your device (e.g S for TBS12S, B for TBS12B)



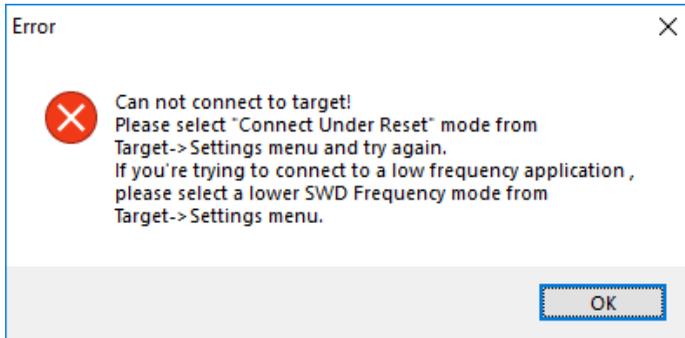
- from

the menu select **Program / Current Tab**

- look in the status display on the screen for a message showing the progress with the update
 - if the last line shows “.. successfully verified” you know that the new firmware has been written and verified

```
> Programming PROGRAM MEMORY area...
Cut Version and Revision of device: 1.1
< PROGRAM MEMORY programming completed.
> Verifying PROGRAM MEMORY area...
Cut Version and Revision of device: 1.1
< PROGRAM MEMORY successfully verified.
```

- if a message displays advising that the ST Link module can't communicate with the target, double check that the TBS12 is powered up and that everything is connected properly
 - the TBS12 must be powered up
 - if you still can't resolve the error contact TOIP for assistance.



Once programming is complete, you can remove the ST-Link connector plug from the TBS12 and disconnect the ST-Link module from your PC. Then close the TBS12 Console.

You should now restart the TBS12

- do this by removing the battery, waiting a couple of minutes and then plugging it back in
- now re-open the Console, select the correct COM port, select Connect and Console Mode
- to check the firmware version in the RTU, from the menu select View / Get Current Firmware Version
 - the firmware version will display on the bottom of the screen
 - if you receive from the Console that the version is different from that in the TBS12, check the Console version by selecting Help / About
 - if the Console is on an earlier version, update it using the copy supplied with the firmware.

11. Rising HF Modem Firmware Upgrade

The firmware in the LoRa WAN modem can be upgraded to add new features or in response to requests to fix problems. The upgrade is initiated from the Console and is performed using a second USB to Serial adaptor.

Items Required:

- TBS12S / TBS12B / TBS12PC-FS / TBS12SM
- USB-Serial adaptor (normal unit for Console0)
- USB to serial adaptor (second unit for modem programming)
- 3 pin “Pogo” pin header
 - this is available from TOIP and is fitted with a 3 pin header to connect to a standard TBS12 programming cable
- Rising HF modem firmware for RHF 076-052-am modem e.g. rhf76-052am-v2.1.19-20180525.ebin.bin
- Extra Putty – terminal emulation software with enhanced file upload functionality (download from <http://www.extraputty.com/download.php>)

11.1. Preparation

First off, check the modem firmware version to ensure that it does need upgrading:

- start the Console and connect the USB-TTL adaptor
- once you are in Console Mode, make sure the “send Command to” function is set to LORA
- in the command window type AT+VER
- if the version shows 2.1.18 or earlier it needs to be upgraded

Plug the 2nd adaptor to a USB port on your PC and identify the COM port it is allocated

Open Extra Putty and create and save a session on the port you identified for this device (speed 115200 bps).

Locate the modem Programming port connections: these are brought out close to pin 1 of the LoRa WAN modem unit and will be either on PCB pads or on a dedicated 3 pin header.

11.1.1. TBS12B & TBS12S

Connect the 2nd USB adaptor via a short length of cable to a pair of “Pogo” pins

- red (Rx line on adaptor) = Tx → right pad
- black (Tx line on adaptor) = Rx → left pad
- you will have to hold the pogo pins in place while you perform the upgrade



11.1.2. TBS12PC/FS

The TBS12PC/FS board has a socket connected directly to the RHF modem. It is labelled CON5.

Use an adaptor cable to connect the 3 pin JST plug on a normal TBS12 programming cable to the socket.

Function	Prog Cable	Adaptor
GND	1	3
TX	2	2

RX	3	1
----	---	---

11.2. Upgrade Process

Connect the TBS12 to your PC using the first USB to serial (FTDI) converter and open the TBS12 Console software

- click on Connect and then change to Console mode
 - rather than waiting for the unit to go through a logging cycle, disconnect the battery (and USB adaptor), wait 10 seconds, then plug it back in
- in the direct command section, select the LoRa radio button so that the commands will be sent to the RHF modem
- in the Direct Command window of the TBSL1 Console, type
AT+DFU=ON
 - this prepares the modem for a direct firmware upgrade
- switch back to Extra putty: you should see the “C” character being repeatedly written to the screen as the Modem waits for the downloaded file
 - from the menu select Files Transfer / Y-Modem and then choose the location in which you saved the modem firmware
 - select the file e.g. rhf76-052am-v2.1.19-20180525.ebin.bin
 - wait while the file is transferred
 - you can then close Extra Putty
- power down the TBS12 so it restarts (disconnect the USB cable, remove the battery lead from the socket on the PCB, wait 5 seconds than plug it back in)
- connect the Console to the TBS12 again and make sure Console Mode is selected
- check the modem firmware version now matches that of the updated file by typing the command AT+VER

- the modem will respond with its current firmware version
- remove the 2nd USB converter and modem upgrade lead