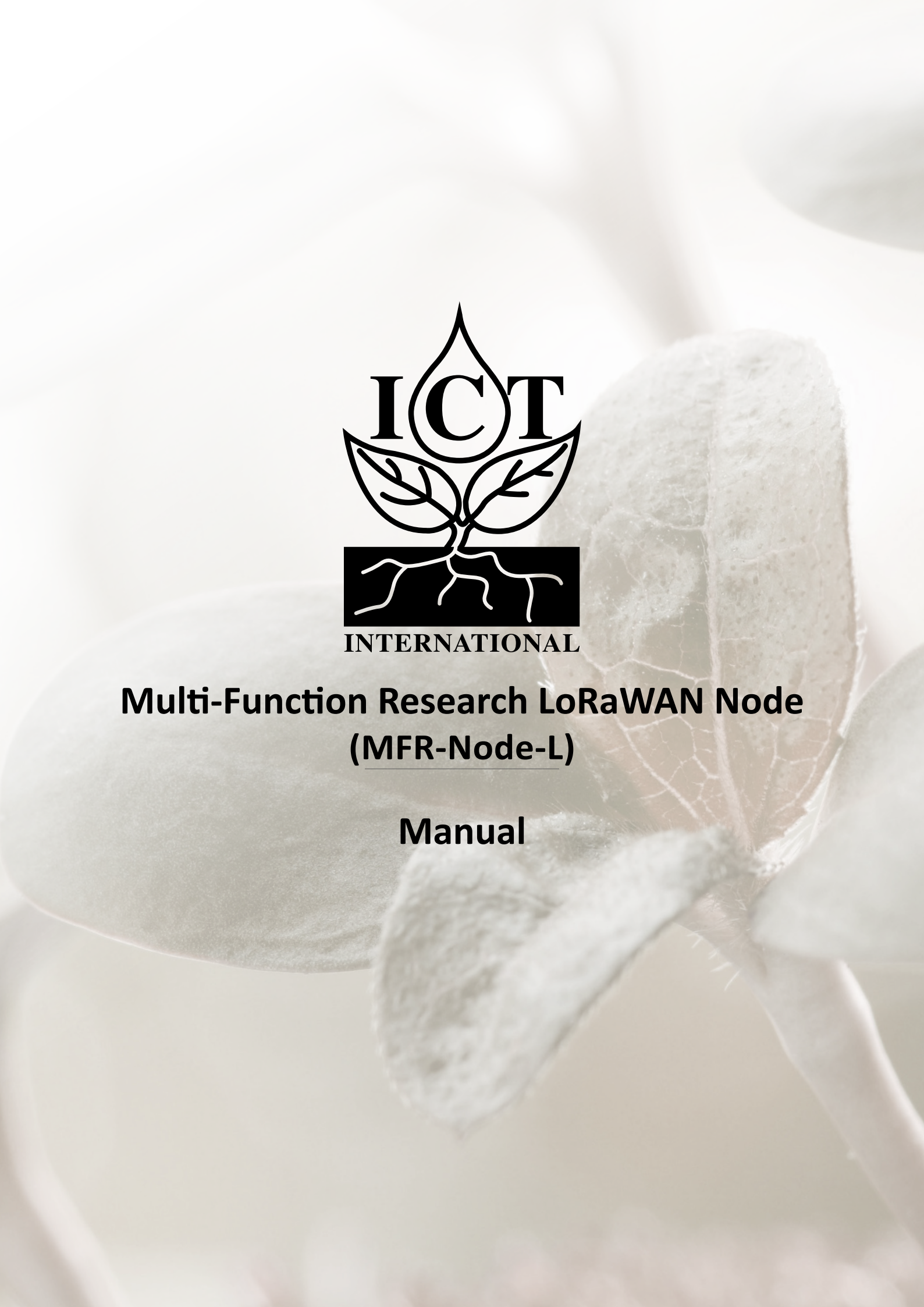


Multi-Function Research LoRaWAN Node (MFR-Node-L)

Manual

Contents

1. Introduction 4

4. Connecting Sensors 5

5. Connecting Battery 7

5. External Power 7

6. LED Status 7

7. Decoder Notes 8

2. Device Operation 9

2.1 SDI-12 Configuration 9
2.3 Analog Input Configuration 9
2.4 SD Card Logging 9
2.5 LoRaWAN Packet Structure 10

3. Device Configuration 11

3.1 Configuration Commands 12
3.1.1 Firmware Version – version 12
3.1.2 Save Current Device Configuration – save 12
3.1.3 Reload Saved Configuration – load 12
3.1.4 Reset to Factory Defaults – config reset 13
3.2 General LoRaWAN™ Configuration 13
3.2.1 LoRaWAN EUIs – lora eui 13
3.2.2 LoRaWAN Application/Join Key – lora key app 14
3.2.3 LoRaWAN Network Joined Status – lora joined 14
3.2.4 LoRaWAN Public Network Mode – lora net public 15

3.2.5 LoRaWAN Adaptive Data Rate – lora net adr 15
3.2.6 ADR Link Check Period– linkcheck period 16
3.2.7 LoRaWAN Default Data Rate – lora data rate 16
3.2.8 LoRaWAN Maximum TX Power – lora tx power 17
3.2.9 Confirmed Messaging – lora confirmed 17
3.2.10 LoRa Band 18
3.3 LoRaWAN Activation-by-Personalisation Mode 19
3.3.1 LoRaWAN Activation-by-personalisation – lora net abp 19
3.3.2 LoRaWAN Session Keys – lora key session 19
3.3.3 LoRaWAN Device Address – lora net dev addr 20
3.3.4 LoRaWAN Network Indentifier – lora net id 20
3.4 Timing 21
3.4.1 Time to retransmit – backoff initial 21
3.4.2 Maximum retransmit time after failed communication – backoff max 21
3.4.3 Period between reports – report period 22
8.5 SDI-12 23
8.5.1 SDI-12 add command – sdi12 add 23
8.5.2 Send SDI-12 Command – sdi12 send 24
8.5.4 Delete all SDI-12 Commands – sdi12 remove all commands 24
3.6 Analog Commands 25
3.6.1 Analog Channel Configuration – adc ch config 25
3.6.2 Analog Single Ended Test – adc single test 25
3.6.3 Analog Differential Test – adc diff test 25
3.6.4 Disable Constant Excitation – persistent pwr 26
3.6.5 Calibrate Analog Channel – adc ch calibrate 26
3.6.6 Disable voltage divider in ADC calculation 27
3.7 Digital Input Commands 27
3.7.1 Enable Digital Inputs - counter enable 27
3.7.2 Test Digital Inputs – counter test 27
3.8 General Commands 28
3.8.1 Enable SD Card Logging – sd enable 28
3.8.3 Set RTC offset – utc offset 29
3.8.4 Firmware Update Mode – bootloader 30
8.5.5 Set low battery mode threshold – battery threshold 31
8.5.6 Command List – help 31

1. Introduction

The ICT International MFR Node is a LoRaWAN data transmission device with on-board MicroSD card storage.

Data is stored in CSV format.

The device is powered by a lithium ion battery pack (6.7Ah or 13.4Ah) and is charged by external 12-24V DC input – typically a 10W or 20W solar panel.

Available sensor inputs are:

- SDI-12 (up to 5 sensors),
- Analogue (4 single-ended or 2 differential, 3V, 5V or 12V selectable excitation),
- and 4 digital pulse inputs.

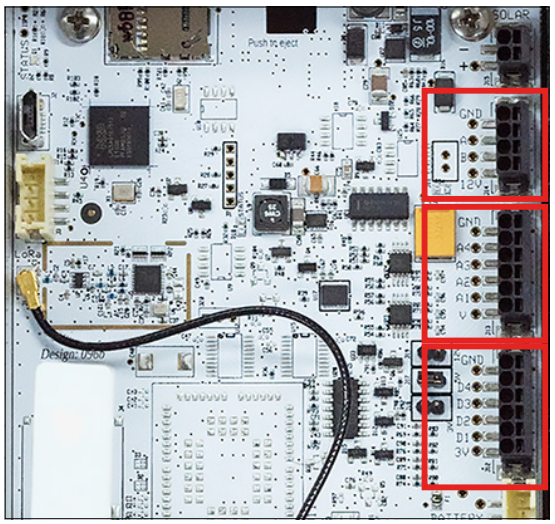
Figure 1. Inside The MFR-Node Box - The MFR-Node Board

Figure 2. The MFR-Node Box - With Antenna

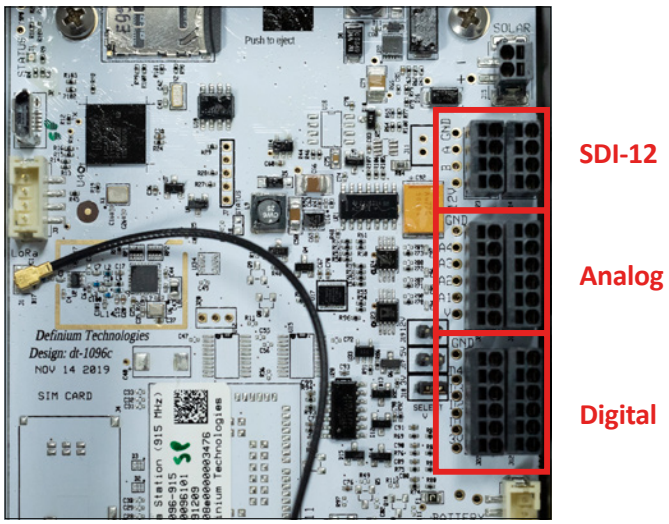
4. Connecting Sensors

4.1 Sensor Connector Locations on the MFR-Node board

A connector is available on the right-hand side of the board for each type of sensor. SDI-12 is below the solar input. Analog is below SDI-12 (A1 to A4). Digital is D1 to D4 .



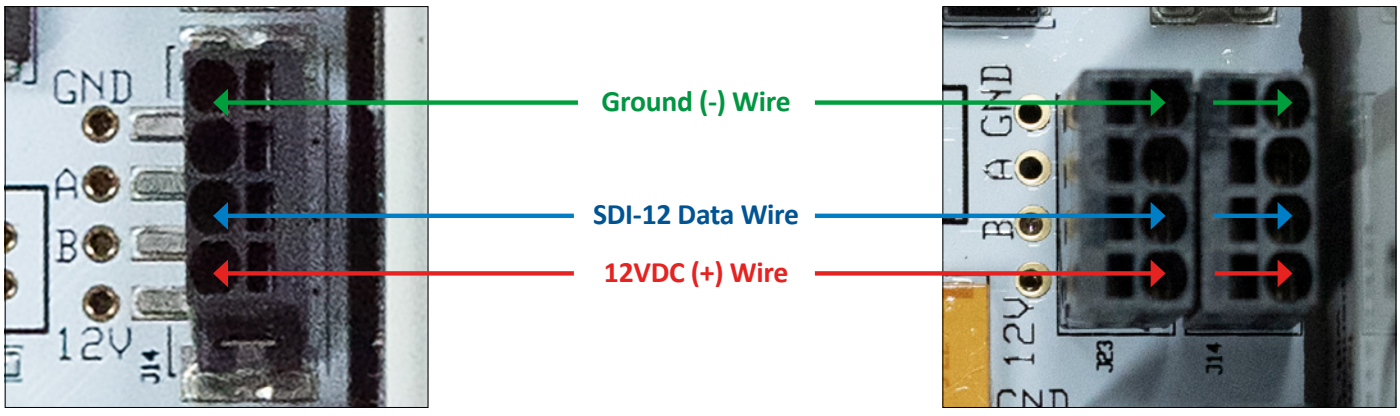
MFR-Node Hardware Variant 1 Serial Number Prefix MLNA1



MFR-Node Hardware Variant 2 Serial Numbers Prefix MLNA2

4.2 SDI-12

To connect an SDI-12 sensor, insert the Ground Wire of the sensor into the connector labelled GND. Insert the Data line into the connector labelled B. Insert the Power wire into the connector labelled 12V. Hardware variant 2 physically supports the connection of 2x SDI-12 sensors. For both hardware variants 1 & 2 additional SDI-12 sensors can be bussed off the board.



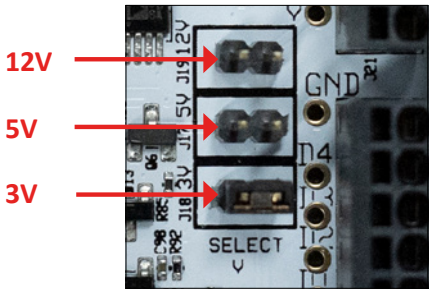
MFR-Node Hardware Variant 1

MFR-Node Hardware Variant 2

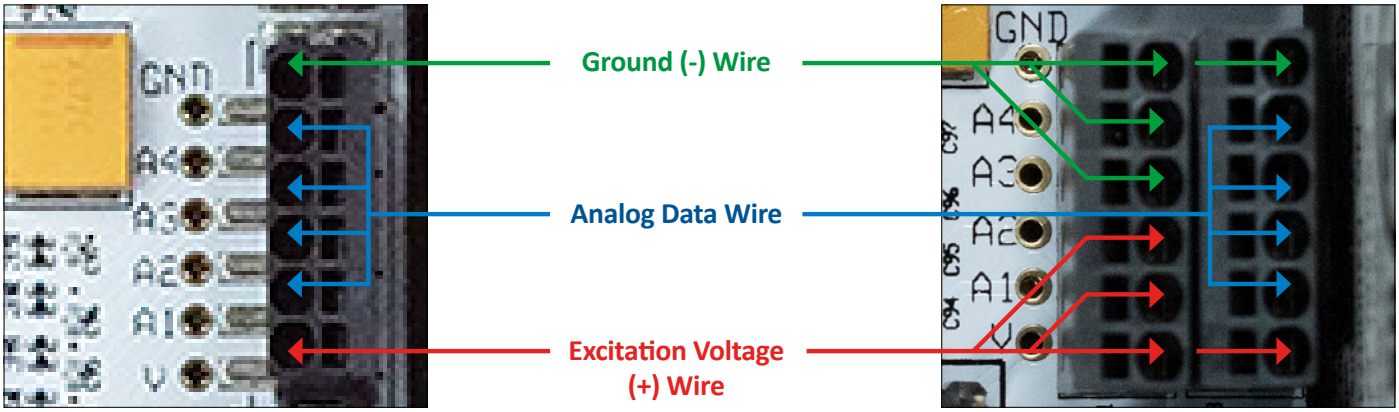
4.3 Analog Sensors

4.3.1 Analog Excitation Voltage Selection

To connect analog sensors, first ensure that the excitation voltage is set correctly. Available voltages are 12V (top), 5V (middle) and 3V (bottom). Put the jumper on the pins for the excitation required (shown below on 5V). Wire sensors according to the sensor manual. The V inputs supply the selected Excitation to the sensor; GND is ground. A1 to A4 are the analog channels. If using differential sensors, use A1 and A2 or A3 and A4.



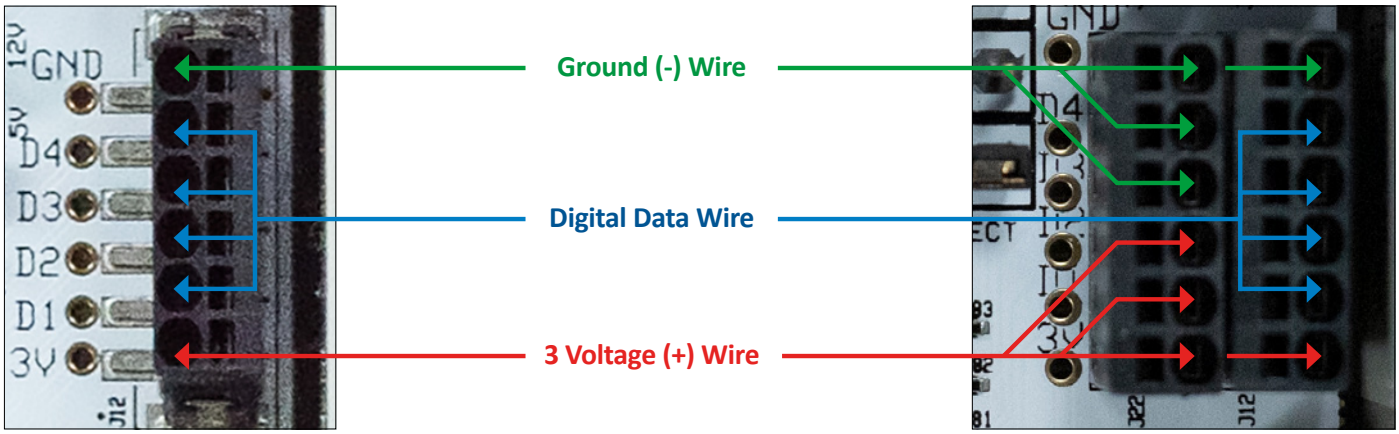
4.3.2 Wiring Analog Sensors



MFR-Node Hardware Variant 1 **MFR-Node Hardware Variant 2**

4.4 Digital (Pulse) Sensors

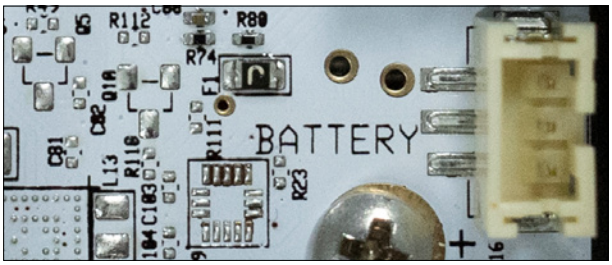
Wire the sensor according to the sensor manual. Most Digital Pulse sensors (rain gauges, anemometers, etc.) are passive and should be wired between an input (D1 to D4) and ground. If excitation is required, use the 3V pin.



MFR-Node Hardware Variant 1 **MFR-Node Hardware Variant 2**

5. Connecting Battery

It is important to ensure that the battery is connected to the board, before connecting to external power, such as a 12V or 24V solar panel.



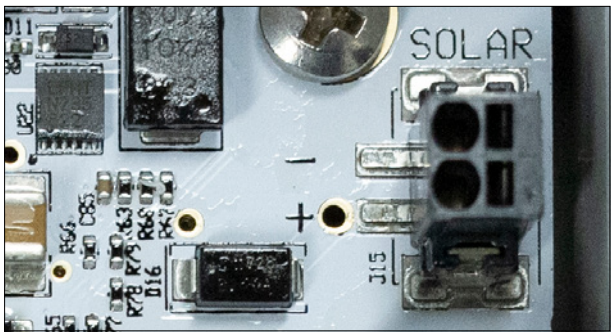
5. External Power

The input for external power is located on the top right of the board, labelled SOLAR.

This input is polarised, please ensure that positive is inserted in the + terminal and negative in the -.

The MFR Node has an on-board solar charge controller and can be directly connected to a 12V or 24V solar panel. Alternatively, a 12V to 24V mains DC power supply can be connected for indoors use.

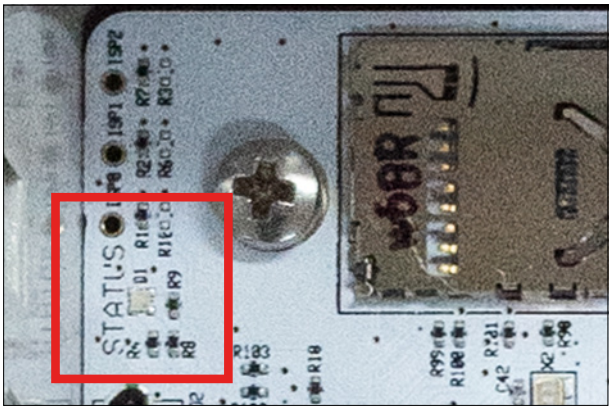
Please ensure that the MFR Node's internal battery is plugged in before (and when) using solar power.



6. LED Status

The status led is used to indicate the following:

- **LIGHT BLUE**: Joining Network
- **DARK BLUE**: Network Joined / Taking measurement
- **ORANGE**: Transmitting sensor data
- **PURPLE**: Measurement Complete
- **GREEN**: USB Idle
- **RED**: Failed to Join Network



7. Decoder Notes

Please contact ICT International for the MFR-NODE decoder applicable to your order and suitable for TTN (https://www.thethingsnetwork.org/):

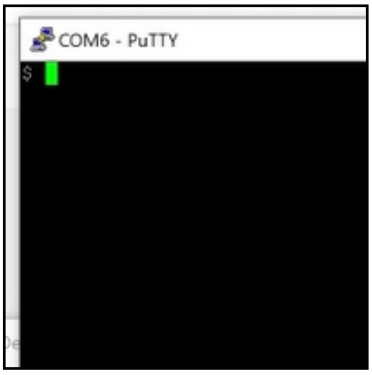
2. Device Operation

2.1 SDI-12 Configuration

The MFR Node's SDI-12 communications can be customised to allow for interfacing with any SDI-12 sensor. See section 3.5 for a detailed description of the configuration interface of the device.

Adding or modifying an SDI-12 command for taking a measurement from a sensor uses the sdi12 add command. Commands can be disabled/enabled using the sdi12 activate command – by default, commands are enabled when they are added or modified. Other commands can be sent directly to the sensor using the sdi12 send command.

For more information on SDI-12 related commands, see section 3.5.



Configuration Program Example

2.3 Analog Input Configuration

The MFR Node's analog inputs can be configured as 4 single ended, 2 differential, or a combination. This is done using the adc ch config command. A differential channel is a pair of single ended channels, as such, only channel 1 and 3 can be configured as differential.

For more information on analogue configuration, see section 3.6.

2.4 SD Card Logging

SD card logging is enabled by sd enable. The MicroSD card will be formatted by the MFR Node. Data will be logged at the report interval in standard CSV format. The data is timestamped with either the MFR Node's RTC, or a downlink can be sent to set the current time.

For more information on SD Card Logging, see section 3.8.

2.5 LoRaWAN Packet Structure

LoRaWAN packets sent by the device use the following format.

- The first byte is a header describing the packet.
- 0x10 is device information (battery voltage, external input voltage, charge indicator, fault indicator).
- 0x20 is analogue readings.
- 0x40 is digital readings.
- 0x80 is SDI-12 readings.

An example payload decoder is available in Section 7.
For more information on SD Card Logging, see section 3.8.

3. Device Configuration

The ICT International MFR Node is configured (i.e. identified and authenticated) over USB serial console using a terminal/terminal emulator. It is compatible with Windows 10, Mac OS and Linux.

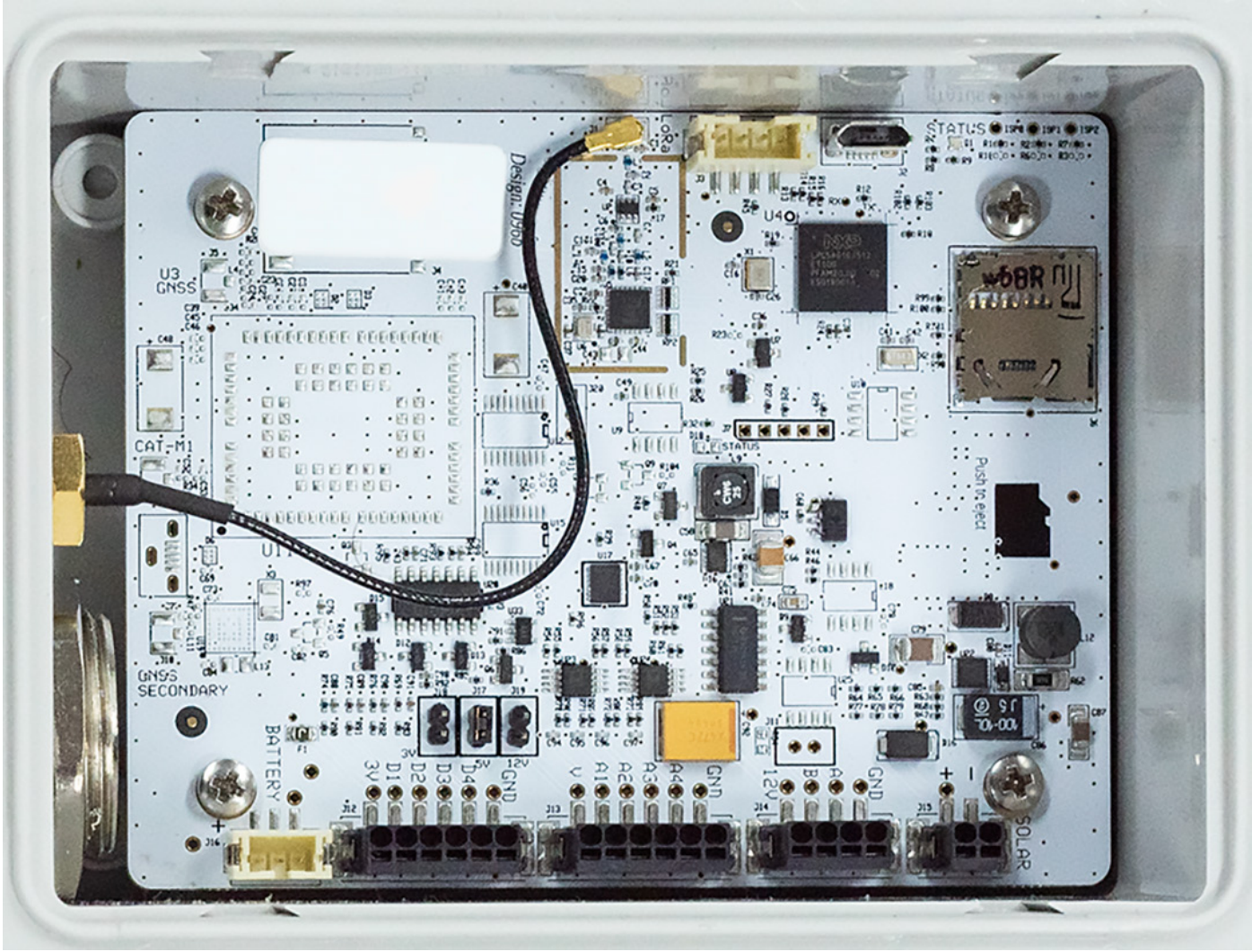
One terminal emulator we recommend is Putty, which can be downloaded from https://www.putty.org/.

All commands are entered as ASCII text and will return any response as ASCII text.

Connecting an MFR Node to a computer via the Micro USB port (top right, fig. 1) will provide a serial port for configuration.

Recommended settings are as follows:

- Baud Rate: 115200 baud
- Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: Disabled.



3.1 Configuration Commands

These commands are entered into a terminal or terminal emulator such as Putty, https://www.putty.org/, to action several types of commands to the MFR-Node.

3.1.1 Firmware Version – version

Command Input: `version`

Compatible: Device Firmware Versions > 1.2

Command Description: Returns information about the device firmware version and configured frequency.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`version`	<String>	`Definium Technologies Pty Ltd LoRaWAN Class-A Sensor 235c0e89-dirty Luna Station 4000096001-"AS923"`

3.1.2 Save Current Device Configuration – save

Command Input: `save`

Compatible: All Device Firmware Versions

Command Description: Saves the running configuration to permanent storage.

Command Type	Syntax Used	Response Type	Example Result of Command
Action	`save`	Saved config	`Saved config`

3.1.3 Reload Saved Configuration – load

Command Input: `load`

Compatible: All Device Firmware Versions

Command Description: Saves the running configuration to permanent storage.

Command Type	Syntax Used	Response Type	Example Result of Command
Action	`load`	Loaded config	`Loaded config`

3.1.4 Reset to Factory Defaults – config reset

Command Input: `config reset`

Compatible: Device Firmware Versions > 1.2

Command Description: Resets the running configuration factory defaults.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`config reset`	Reset app config to defaults	`Reset app config to defaults`

3.2 General LoRaWAN™ Configuration

These commands are entered into a terminal or terminal emulator such as Putty, https://www.putty.org/, to action several types of commands to the MFR-Node.

3.2.1 LoRaWAN EUIs – lora eui

Command Input: `lora eui`

Compatible: All Device Firmware Versions

Command Description: Manage the device's LoRaWAN EUIs (unique identifier), both device and application/ join (dependent on LoRaWAN version).

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command

Get	`lora eui <type>`	`lora eui app`
		`1122334455667788`
			<eui>	`lora eui app`
				`0x11 0x22 0x33 0x44`
				`0x55 0x66 0x77 0x88`

Set	`lora eui <type>`
	`<eui>`
		`lora eui dev`
		`1122334455667788`
			<eui>	`lora eui app`
				`0x11 0x22 0x33 0x44`
				`0x55 0x66 0x77 0x88`
				`0x11 0x22 0x33 0x44`
				`0x55 0x66 0x77 0x88`

Parameter	Type	Description
<type>	String	Type of EUI, options are: `dev`: Device EUI `app`: Application/Join EUI
<eui>	String	16 character Hexadecimal string representing an 8-byte EUI. Example: `0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88`

3.2.2 LoRaWAN Application/Join Key – lora key app

Command Input: `lora key app`

Compatible: All Device Firmware Versions

Description: Manage the device's LoRaWAN application/join key used for Over-the-Air Activation (OTAA).

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora key app`	`lora eui app 1122334455667788`	<key>	`lora key app 00112233445566778899AABBCCDDEEFF`
Set	`lora key app <key>`	`lora key app`	<key>	`lora eui app 1122334455667788AABBCCDDEEFFGGHH 1122334455667788AABBCCDDEEFFGGHH`

Parameter	Type	Description
<key>	String	32 character Hexadecimal string representing an 16-byte EUI. Example: `0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0xAA 0xBB 0xCC 0xDD 0xEE 0xFF 0xGG 0xHH`

3.2.3 LoRaWAN Network Joined Status – lora joined

Command Input: `lora joined`

Compatible: All Device Firmware Versions

Description: Manage the device's LoRaWAN application/join key used for Over-the-Air Activation (OTAA).

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora joined`	`lora joined`	<status>	`enabled`
Set	`lora joined <bool>`	`lora joined 0`	<status>	`lora joined 0 disabled`

Parameter	Type	Description
<bool>	Number	Network joined status to be set: `0`: Unjoined `1`: Joined
<status>	String	Network joined status string. Options: `enabled`: Joined `disabled`: Unjoined / Not Joined

3.2.4 LoRaWAN Public Network Mode – lora net public

Command Input: `lora net public`

Compatible: All Device Firmware Versions

Description: Enable/Disable public network mode. Change the sync word used by the LoRa radio between the public and private settings.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora net public`	`lora net public`	<status>	`lora net public enabled`
Set	`lora net public <bool>`	`lora net public 0`	<status>	`lora net public 0 disabled`

Parameter	Type	Description
<bool>	Number	Network joined status to be set: `0`: Disable `1`: Enabled

<status>	String	Public network status string. Options:
		`enabled`
		`disabled`

3.2.5 LoRaWAN Adaptive Data Rate – lora net adr

Command Input: `lora net adr`

Compatible: All Device Firmware Versions

Description: Devices using adaptive data rate will query the server periodically to determine the highest data rate they can transmit at and still be heard. The LoRaWAN specification states that stationary devices should use ADR, however use on mobile devise is discouraged as it is unlikely to work correctly.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora net adr`	`lora net adr`	<status>	`disabled`
Set	`lora net adr <bool>`	`lora net adr 1`	<status>	`lora net adr 1 disabled`

Parameter	Type	Description
<bool>	Number	Network joined status to be set: `0`: Disable `1`: Enabled

<status>	String	Public network status string. Options:
		`enabled`
		`disabled`

3.2.6 ADR Link Check Period– linkcheck period

Command Input: `linkcheck period`

Compatible: Device Firmware Versions > 1.2

Description: Determines the number of unconfirmed packets between link checks when using ADR. Typically, this is set automatically and should not need to be changed.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`linkcheck period`	`linkcheck period`	<period>	`64`
Set	`linkcheck period <period>`	`linkcheck period 0`	<period>	`linkcheck period 200 200`

Parameter	Type	Description
<period>	Number	Number of unconfirmed packets between link checks.

3.2.7 LoRaWAN Default Data Rate – lora data rate

Command Input: `lora data rate`

Compatible: All Device Firmware Versions

Description: Manage the device's default data rate. Used when ADR is not enabled.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora data rate`	`lora data rate`	<data rate>	`2`
Set	`lora data rate <data rate>`	`lora data rate 0`	<data rate>	`lora data rate 1 1`

Parameter	Type	Description
<data rate>	Number	LoRaWAN Data Rate, where 0 is minimum (LoRa spreading factor 12). Maximum value is region specific.

3.2.8 LoRaWAN Maximum TX Power – lora tx power

Command Input: `lora tx power`

Compatible: All Device Firmware Versions

Description: Manage the device's default maximum transmit power. This is relative to the maximum EIRP, which can be changed by the network server after the device is joined to the network.

The actual maximum transmit power will be the maximum EIRP minus 2 times this setting, in dBm, unless that value is above 20 dBm, as that is the maximum power of the device's radio.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora tx power`	`lora tx power`	<power>	`0`
Set	`lora tx power <power>`	`lora tx power 2`	<power>	`lora tx power 2 2`

Parameter	Type	Description
<power>	Number	Maximum transmit power, power below MAX_EIRP. TX_PWR = MAX_EIRP – 2 x power (dBm)

3.2.9 Confirmed Messaging – lora confirmed

Command Input: `lora confirmed`

Compatible: All Device Firmware Versions

Description: Enables/Disables LoRaWAN confirmed messaging.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora confirmed`	`lora confirmed`	<string>	`disabled`
Set	`lora confirmed <bool>`	`lora confirmed 0`	<status>	`lora net adr 0 disabled`

Parameter	Type	Description
<bool>	Number	Indicates whether to enable or disable confirmed messaging: `0`: Disable `1`: Enabled

3.2.10 LoRa Band

Command Input: `lora band`

Compatible: Device Firmware Versions

Description: Sets the LoRaWAN sub-band and default channel mask. Note that band should be set to 2 for AU915 and US915. Band selection is not applicable to AS923.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora band`	`lora band`	<string>	`lora band` `Band set to: 1`
Set	`lora band <band>`	`lora band 2`	<string>	`lora band 2` `Band set to: 2`

Parameter	Type	Description
<band>	Number	LoRaWAN Sub-band selection. Frequency plan dependant.

3.3 LoRaWAN Activation-by-Personalisation Mode

These commands are entered into a terminal or terminal emulator such as Putty, https://www.putty.org/, to action several types of commands to the MFR-Node. In Activation-by-Personalisation mode the device is pre-supplied with all keys and identifiers such that it does not need to undergo an over-the-air join process.

All settings in this section are required to be manually set when using Activation-by-Personalisation (ABP) and will be retrieved during the join process for OTAA devices. Do not change these if using OTAA on the device.

3.3.1 LoRaWAN Activation-by-personalisation – lora net abp

Command Input: `lora net abp`

Compatible: All Device Firmware Versions

Command Description: Enable/Disable Activation-by-Personalisation Mode.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora net abp`	`lora net abp`	<status>	`disabled`
Set	`lora net abp`	`lora net abp 1`	<status>	`lora net abp 1 enabled`

Parameter	Type	Description
<bool>	Number	Indicates whether to enable or disable abp mode: `0`: Disable / `1`: Enabled
<status>	String	ABP mode status string. Options: `enabled` / `disabled`

3.3.2 LoRaWAN Session Keys – lora key session

Command Input: `lora key session`

Compatible: All Device Firmware Versions

Command Description: Manage the device's application and network session keys.

Type	Syntax Used	Response Type	Example Result of Command
Get	`lora key session <type>`	<key>	`1122334455667788AABBCCDDEEFFGGHH`
Set	`lora key session <type> <key>`	<key>	`lora key session app 1122334455667788AABBCCDDEEFFGGHH` `1122334455667788AABBCCDDEEFFGGHH`

Parameter	Type	Description
<key>	String	32 character Hexadecimal string representing an 16-byte EUI. Example: `0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0xAA 0xBB 0xCC 0xDD 0xEE 0xFF 0xGG 0xHH`
<type>	String	Key Types include: `app` (Application Session Key) `nwk` (Network Session Key)

3.3.3 LoRaWAN Device Address – lora net dev addr

Command Input: `lora net dev addr`

Compatible: Device Firmware Versions

Command Description: Get/Set the device's LoRaWAN network address.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora net dev addr`	`lora net dev addr`	<eui>	`11223344`
Set	`lora net dev addr <eui>`	`lora net dev addr`	<eui>	`lora net dev addr 11223344 11223344`

Parameter	Type	Description
<eui>	String	8 character Hexadecimal string representing an 4-byte EUI. Examples: `0x11 0x22 0x33 0x44` `11223344`

3.3.4 LoRaWAN Network Indentifier – lora net id

Command Input: `lora net id`

Compatible: Device Firmware Versions

Command Description: Get/Set the network identifier of the connected LoRaWAN network.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`lora net id`	`lora net dev addr`	<eui>	`11223344`
Set	`lora net id <eui>`	`lora net dev addr`	<eui>	`lora net id 11223344 11223344`

Parameter	Type	Description
<eui>	String	8 character Hexadecimal string representing an 4-byte EUI. Examples: `0x11 0x22 0x33 0x44` `11223344`

3.4 Timing

Commands for managing the device's state transition timings.

3.4.1 Time to retransmit – backoff initial

Command Input: `backoff initial`

Compatible: Device Firmware Versions > 1.2

Command Description: Initial wait time on failed communications before retrying. Doubles each failure until it reaches backoff max.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`backoff initial`	`backoff initial`	<string>	`Initial Backoff Time: 60 sec`
Set	`backoff initial <time>`	`backoff initial 90`	<string>	`backoff initial 90 Initial Backoff Time: 90 sec`

Parameter	Type	Description
<time>	Number	Wait time in seconds.

3.4.2 Maximum retransmit time after failed communication – backoff max

Command Input: `backoff max`

Compatible: Device Firmware Versions > 1.2

Command Description: Maximum wait time on failed communications before retrying.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`backoff max`	`backoff initial`	<string>	`Maximum Backoff Time: 86400 sec`
Set	`backoff max <time>`	`backoff initial 90`	<string>	`backoff initial 90000 Initial Backoff Time: 90000 sec`

Parameter	Type	Description
<time>	Number	Wait time in seconds.

3.4.3 Period between reports – report period

Command Input: `report period`

Compatible: Device Firmware Versions > 1.2

Command Description: Initial wait time on failed communications before retrying. Doubles each failure until it reaches backoff max.

Command Type	Syntax Used	Example Command	Response Type	Example Result of Command
Get	`report period`	`report period`	<string>	`Report Period: 900 sec Current: 12 Last: 0 Next:0`
Set	`report period <period>`	`report period 600`	<string>	`report period 600 Report Period: 600 sec Current: 26 Last: 0 Next:0`

Parameter	Type	Description
<period>	Number	Time in seconds between reports.
<time>	Number	Current device timestamp
<last>	Number	Timestamp that state last triggered at
<next>	Number	Timestamp of next state trigger

8.5 SDI-12

8.5.1 SDI-12 add command – sdi12 add

Command Input: `sdi12 add`

Compatible: Device Firmware Versions

Command Description: Add or modify SDI-12 command in slot. Available SDI-12 commands are: Measurement (M!), Concurrent (C!) and Result (R!).

Measurement and Concurrent must use the Measure type command. Result must use the Data command

Older Firmware Examples:

```
sdi12 add 0 M 0C0! 3 3 0D0! 111
```

Use SDI-12 command id/slot 0 to send a Concurrent measurement command to SDI-12 sensor address 0, delay 3 seconds, value length 3, send a data command to sensor address 0, get 3 parameters. Older firmware models explicit delay and value lengths.

Newest Firmware Examples:

```
sdi12 add 0 M 0C0! 0D0! 111
```

Use SDI-12 command id/slot 0 to send a Concurrent measurement command to SDI-12 sensor address 0.

The sensor will return a response in the form of `atttnn` where a = the sensor address, ttt = the specified time in seconds until the sensor will have the measurements ready, and nn = the number of measurement values.

After the specified wait time, the node will send a data command `0D0!` to sensor address 0, it will return all available results but the `111` sensor measurement masking will only prepare and transmit the first three parameters. Newest firmware models don't explicit delay and value lengths.

Command Type	Syntax Used	Response Type
Measure/Data	`sdi12 add <id> M <measure command> <data command> <mask>`	<string>

Parameter	Type	Description
<id>	Number	0-9 Slot ID to add or modify.
<measure command>	String	SDI-12 measure command to execute on specified address.
<data command>	String	SDI-12 command to return data on specified address.
<mask>	Binary	Sensor measurement masking. Length of the mask can be equal to the number of readings. E.g: the mask to select the first and fourth reading for an 8 reading data command is: 10010000, where 1 is on and 0 is off.

8.5.2 Send SDI-12 Command – sdi12 send

Command Input: `sdi12 send`

Compatible: All Device Firmware Versions
Command Description: Send an SDI-12 command. Can be used for identifying sensors on a bus, configuring SDI-12 addresses, or any other sensor specific functions.

Type	Syntax Used	Example Command	Response Type	Example Result of Command
Action	`sdi12 send <command>`	`sdi12 send ?!`	<response>	`1`

Parameter	Type	Description
<command>	String	SDI-12 command to execute.
<response>	String	Response from SDI-12 command, command dependant

Commands	Description
`aI!`	Sends sensor identification request for the sensor at address `a`
`aAb!`	Change sensor address from `a` to `b`
`?!`	Query sensor address, can only be done when a single sensor is connected

8.5.4 Delete all SDI-12 Commands – sdi12 remove all commands

Command Input: `sdi12 remove all commands`

Compatible: All Device Firmware Versions. Immediately after using this command, you will need to enter the save command.
Command Description: Removes all configured SDI-12 command slots.

Type	Syntax Used	Response Type	Example
Action	`sdi12 remove all commands`	<string>	`sdi12 remove all commands` `SDI12 Commands Erased`

3.6 Analog Commands

3.6.1 Analog Channel Configuration – adc ch config

Command Input: `adc ch config`

Compatible: Device Firmware Versions > 1.2

Command Description: Enables and sets the configuration of the analog channels.

Command Type	Syntax Used	Example Result of Command
Set	`adc ch config <mask>`	`adc ch config DOSS`

Parameter	Type	Description
<mask>	String	State of each analog channel: `S`: Single Ended, available for channels 1-4 `D`: Differential, available for channels 1 and 3, channels 2 and 4 must be set to off respectively `O`: Off – disables channel

3.6.2 Analog Single Ended Test – adc single test

Command Input: `adc single test`

Compatible: All Device Firmware Versions

Command Description: Display readings for the 4 single-ended analog channels, in µV.

Command Type	Syntax Used	Response Type	Example Result of Command
Action	`adc single test`	<string>	`adc single test` `ADC SING=160,129,175,121`

3.6.3 Analog Differential Test – adc diff test

Command Input: `adc diff test`

Compatible: All Device Firmware Versions

Command Description: Display readings for the 2 differential channels, in uV.

Command Type	Syntax Used	Response Type	Example Result of Command
Action	`adc diff test`	<string>	`adc diff test` `ADC DIFF=39,50`

3.6.4 Disable Constant Excitation – persistent pwr

Command Input: `persistent pwr`

Compatible: Device Firmware Versions > 1.2

Command Description: Enables/disables constant sensor excitation for both SDI-12 and analog. Please contact ICT International to confirm that this is suitable for your application.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`persistent power`	<string>	`persistent power enabled`
Set	`persistent power <enable>`	<string>	`persistent power 0 disabled`

Parameter	Type	Description
<enable>	Number	Enable (1)/Disable (0)

3.6.5 Calibrate Analog Channel – adc ch calibrate

Command Input: `adc ch calibrate`

Compatible: Device Firmware Versions > 1.2

Command Description: Used to precisely calibrate analog channels. This is generally not necessary – Contact ICT International to confirm that this is applicable to your application. Calibration requires a power source that can cover 1mV to 1V accurately.

Procedure: (Repeat on all channels necessary.)

Set the power supply to 1mV and measure the voltage into the connector. Enter the following command:
`adc ch calibrate <channel> 0 <voltage measured as µV>`

For example: `adc ch calibrate 1 0 1024`

Set the power supply to 1V and measure the voltage into the connector. Enter the following command:
`adc ch calibrate <channel> 1 <voltage measured as µV>`

For example: `adc ch calibrate 1 1 1001024`

3.6.6 Disable voltage divider in ADC calculation

Command Input: `sdi12 remove all commands`

Compatible: Device Firmware Versions > 1.2

Command Description: Disables the resistor divider in the calculation and calibration of the ADC. Contact ICT International to see if this is applicable to your application. Requires hardware modification and can only be used with sensors with < 3V output.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`adc div`	<string>	`adc div` `enabled`
Set	`adc div <enable>`	<string>	`adc div 0` `disabled`

Parameter	Type	Description
<enable>	Number	Enable (1)/Disable (0)

3.7 Digital Input Commands

3.7.1 Enable Digital Inputs - counter enable

Command Input: `counter enable`

Compatible: All Device Firmware Versions

Command Description: Enable or disable logging and upload of digital inputs.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`counter enable`	<string>	`counter enable` `enabled`
Set	`counter enable <enable>`	<string>	`counter enable 1` `disabled`

Parameter	Type	Description
<enable>	Number	Enable (1)/Disable (0)

3.7.2 Test Digital Inputs – counter test

Command Input: `counter test`

Compatible: All Device Firmware Versions

Command Description: Displays current values of the digital inputs

3.8 General Commands

3.8.1 Enable SD Card Logging – sd enable

Command Input: `sd enable`

Compatible: All Device Firmware Versions

Command Description: Enable or disable logging of data to the onboard MicroSD card. Note that the log file uses instrument RTC time unless an offset has been set. See section 3.8.3.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`sd enable`	<string>	`sd enable` `enabled`
Set	`sd enable <enable>`	<string>	`sd enable 1` `enabled`

Parameter	Type	Description
<enable>	Number	Enable (1)/Disable (0)

3.8.3 Set RTC offset – utc offset

Command Input: `sdi12 remove all commands`

Compatible: Device Firmware Versions > 1.2

Command Description: Sets the RTC value to a current time value, based upon the unix epoch – or the number of seconds that have elapsed since January 1st, 1970. This can also be set using a downlink from the LoRaWAN network server. See below for the packet encoder.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`utc offset`	<offset>	`utc offset` `0`
Set	`utc offset` <offset>	<offset>	`utc offset 1560812278` `1560812278`

Parameter	Type	Description
<offset>	Number	Enable (1)/Disable (0)

3.8.4 Firmware Update Mode – bootloader

Command Input: `bootloader`

Compatible: All Device Firmware Versions

Command Description: Puts the device into firmware update mode. MFR-Node Firmware can be downloaded from: http://ictinternational.com/support/software/

To firmware update the node:

Install Python (make sure to add to path when prompted) - https://www.python.org/downloads/

Then run the following commands in a cmd window:

```
python -m pip install -U pip
pip install pyserial
```

Connect to the node using a terminal emulator (e.g: putty - https://www.putty.org/)

Disable the node by typing in: `enable 0`
Run command: bootloader

The device will stop flashing LED and appear to disconnect via USB.

Open windows cmd, type in: `cd (directory where you saved the firmware)`
Then (in cmd): `python windows_loader.py fw-4000097003-<frequency>.bin`
It will take 1-2 minutes, then once the firmware flash is done the USB will reconnect.

Unplug the node for ~30 seconds, then you can reconnect and reprogram the node.

Command Type	Syntax Used	Response Type
Action	`bootloader`	Node disconnects from serial interface

8.5.5 Set low battery mode threshold – battery threshold

Command Input: `battery threshold`

Compatible: All Device Firmware Versions

Command Description: Sets the battery threshold, below which the device will enter low power mode and cease regular transmission until the battery has charged above the threshold. By default, this is set to 3.4 volts or 3400 milivolts.

Command Type	Syntax Used	Response Type	Example Result of Command
Get	`battery threshold`	<voltage>	`battery threshold` `3400`
Set	`battery threshold <voltage>`	<voltage>	`battery threshold 3400` `3400`

Parameter	Type	Description
<voltage>	Number	Low power cutoff in mV.

8.5.6 Command List – help

Command Input: `help`

Compatible: All Device Firmware Versions

Command Description: Lists all available commands with brief descriptions of their functions.

Command Type	Syntax Used	Response Type
Get	`help`	List of commands

Enabling better global research outcomes in soil, plant & environmental monitoring.

www.ictinternational.com sales@ictinternational.com.au
Ph: +61 2 6772 6770 | 211 Mann St., Armidale, NSW 2350, Australia