Product revision: 1.4
2021-03-15

CrossLink TG

Technical Manual

crosscontrol
www.crosscontrol.com

Contents

Revision history

Rev	Date	Author	Comments
1.4	2021-03-15	Finn McGuirk	

1. Introduction

This Technical Manual describes how to install, set up, and configure the CrossLink TG.

The device can be provided with battery backup as an option and this version will be clearly identified to the user.

This first section provides safety information followed by a brief overview of the product followed by a start-up guide. Further sections will go into more detail on the hardware specifications and various firmware functions available to aid the development of user applications. Application notes, technical specifications, troubleshooting, and a peripheral devices reference, complete this manual.

1.1. Platform

The product provides a powerful platform for management of geographically distributed equipment. The unit integrates a GSM/GPRS communication system with embedded IP functionality, allowing an efficient and bi-directional transmission of information across the cellular wireless network. Additionally, the unit includes a GNSS receiver for reception of positioning information.

The key features of this product include:

- Cortex A8 with 512MB DDR3 and 1GB NAND Flash
- Linux Kernel 4.4.19 or 4.14.67
- Debian Filesystem
- User-friendly programming environment (with a complete set of APIs)
- UMTS/HSPA+: Five Band 800/850/900/1900/2100MHz
- GSM/GPRS/EDGE: Quad Band GSM 850/ EGSM900 / GSM1800/ GSM1900
- GNSS Receiver 72-channel with -160dBm tracking and navigation sensitivity
- Extensive range of external interfaces (RS-232, digital and analog I/Os…)
- CAN Interface
- Programmable 3 axis accelerometer
- MicroSD card holder

Optional feature include: Audio CODEC,Ethernet 10/100, Dead reckoning, Programmable 9 axis sensor, (accelerometer, gyroscope, magnetometer), Kline, IP67 enclosure, Internal antennas, Wi-Fi & BT module, Up to 4 CAN bus interfaces, USB

Check your data sheet for the exact features available in your device.

2. Safety

IMPORTANT: FOR THE EFFICIENT AND SAFE OPERATION OF YOUR CROSSLINK TG, READ THE INFORMATION BELOW BEFORE USE.

Care must be taken when handling the unit. It should not be dropped or exposed to excessive heat.

Only authorized staff can disassemble the product. In the case that the product was disassembled by un-authorized personnel, the warranty will be void.

Do not push foreign objects into the openings of your device. Doing so can cause fire or electric shock by shorting out interior components.

The product can withstand harsh conditions. But if the product is exposed to severe conditions beyond the limits specified in Environmental Specifications, the product could be damaged.

Keep the device away from radiators and extreme heat sources. Do not use the CrossLink TG submerged in wet or very humid environments.

The device may become hot during normal operation, to avoid burns, switch it off and wait for it to cool before handling it.

Do not clean the device when it is powered. Clean it with a soft cloth. Do not use liquid or aerosol cleaners, which may contain flammable substances.

2.1. Connections

The highest internal voltage applied to the unit can be 48Vdc (36Vdc for version R4) and complies with low voltage European directive.

The power supply can be made with a battery or continuous voltage supply with reinforced isolation, and limited in power to a maximum of 8A and 100VA.

Before you connect the device to a power supply, check the voltage and current rating to ensure that the required ones match the available power source. Exceeding the specified input range may cause unexpected operation and/or irreversible damage to the unit.

To remove the device from all power sources, turn the device off and disconnect it from the power supply.

Be sure that nothing rests on the connected cables and that the cables are not located where they can be tripped over or stepped on.

Applying loads outside of the range specified may result in unintended operation and/or possible permanent damage to the unit.

2.2. Protections

The unit inputs have some protections to avoid damages when erroneously connecting to a power supply for example or due to voltage peaks:

- Power supply input (V_IN) reverse voltage protection and overvoltage protection zener diode. External fuse of 2 A should be added.
- Power output (V_OUT) overcurrent protection.
- DIOs overcurrent protection.

2.3. EMC Instructions

Use shielded signal cables to ensure that you maintain the appropriate EMC classification for the intended environment.

Keep cables as short as possible.

2.4. Exposure to RF Energy

Minimize RF energy exposure by limiting the duration of GSM calls and operating the unit efficiently.

The antenna must be mounted in such a position that no part of the human body rests close to any part of the antenna. The product is intended to be used with an external GSM antenna, located at least at 20 cm. away from any part of the human body. Those installations not complying with this statement are responsible for providing SAR measurement reports and corresponding declaration.

Do not hold the antenna during a call since it affects call quality and can cause the module to operate at a higher power level than needed.

2.5. Antenna Care and Replacement

Do not use the product with a damaged antenna because when the antenna comes into contact with the skin, a minor burn may result. Replace the antenna immediately.

Use only antennas that are according to the technical specifications. Antennas that do not fulfil the specifications could damage the product and may contravene local RF emissions regulations or invalidate type approval.

2.6. Electronic Devices

Many electronic devices are shielded from RF energy. However RF energy may cause malfunction of improperly shielded electronic devices.

When the product is mounted in a vehicle, check your vehicle to determine that all on board electronic equipment is adequately shielded from RF energy.

In the same way, when the product is in the proximity of medical devices (hospitals, etc.) check with the manufacturer of medical equipment to determine if they are properly shielded.

This equipment should never be operated on an aircraft.

2.7. Blasting Areas

To avoid interfering with blasting operations, turn the unit OFF or in areas with “turn off your two way radio” instructions posted.

Construction crews often uses remote control RF devices to set off explosives so care should be used on site.

2.8. Explosive Atmospheres

Do not operate this product in environments containing explosive materials or vapour. This includes petrol service stations.

The unit accessories could generate sparks that can cause an explosion or fire resulting in bodily injury or even death.

To avoid interfering with blasting operations, turn the unit off in these areas posted

Do not transport or store flammable gas, liquid or explosives, in the same compartment of your vehicle which contains the unit or its accessories.

2.9. Handling the Device

Observe the following safe-handling guidelines to prevent damage to the unit:

- When setting up the device for work, place it on a flat level surface.
- Protect the device from environmental hazards such as dirt, dust, food, liquids, temperature extremes, and overexposure to sunlight.
- When you move your device between environments with very different temperature and/or humidity ranges, condensation may form on or within the device. To avoid damaging it, allow sufficient time for the moisture to evaporate before using the device.
- When taking the device from low-temperature conditions into a warmer environment or from high-temperature conditions into a cooler environment, allow the device to acclimate to room temperature before turning on power.
- When disconnecting a cable, pull on its connector or on its strain-relief loop, not on the cable itself. As you pull out the connector, keep it evenly aligned to avoid bending any connector pins. Also, before you connect a cable make sure both connectors are correctly oriented and aligned.

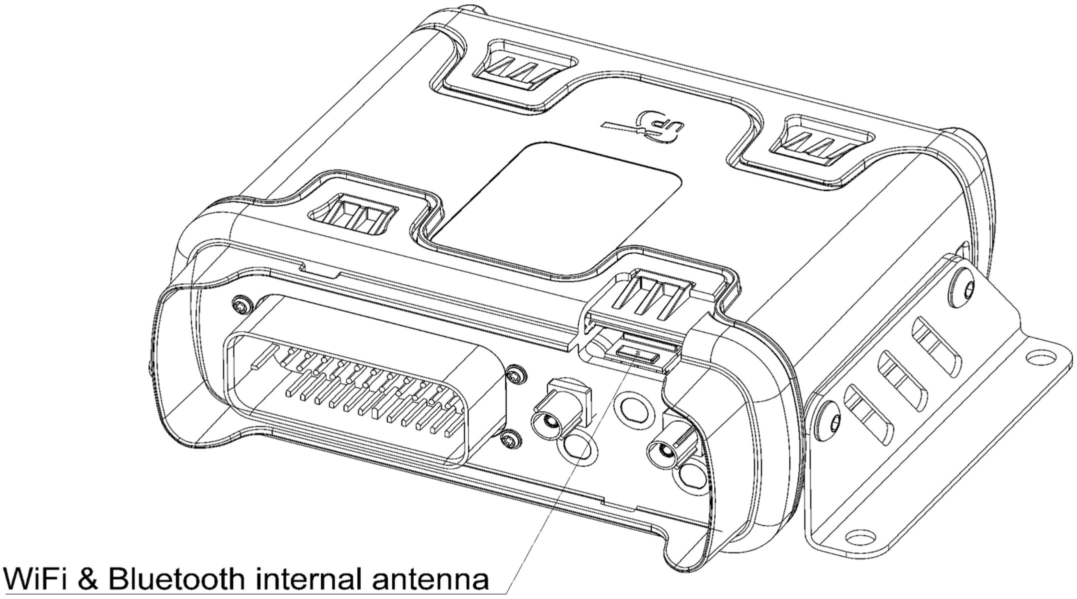
3. Getting Started

This section explains the steps needed to install and configure the development kit to begin with the set-up of the customer application.

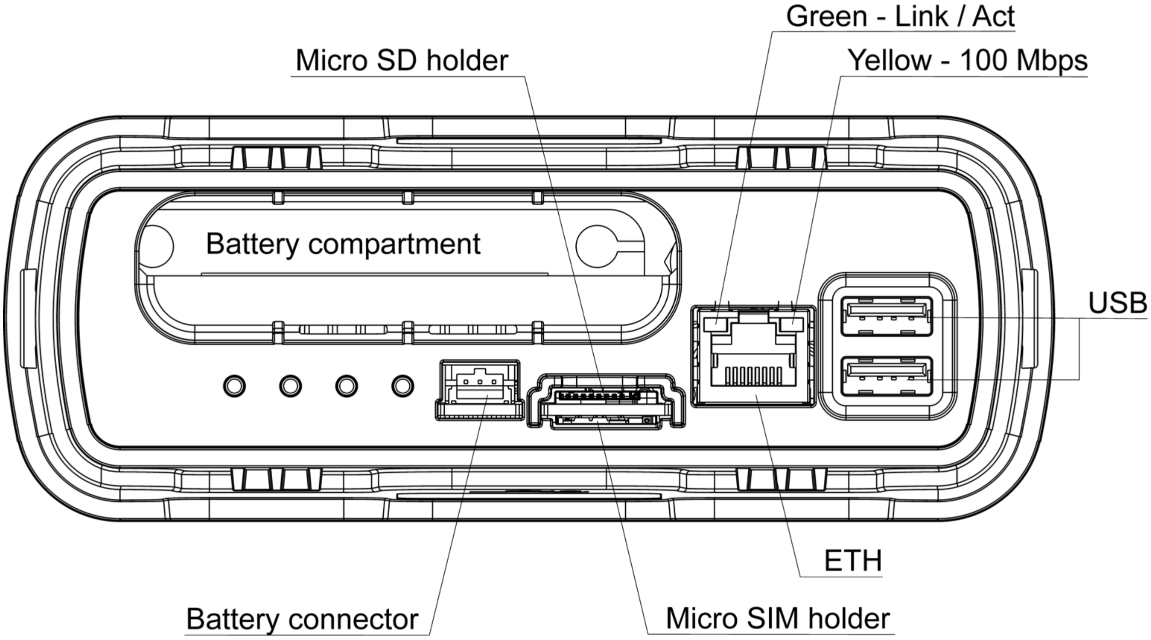
3.1. Unit Properties

The standard unit has GSM and GNSS external antenna connectors, while Bluetooth and WiFi antennas are internal, as shown in the following image.

The AUX LTE antenna is only available with HW R4, which mounts a LTE module.



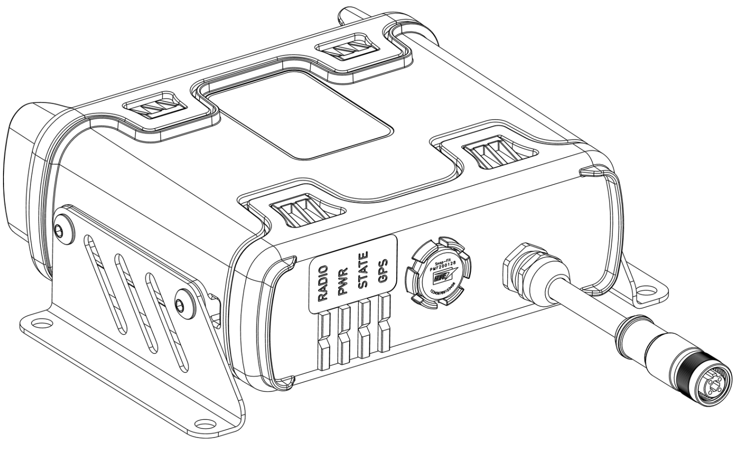
When using either Wi-Fi or Bluetooth functionality, try to make sure that the installation lets a free view of this side of the unit. The “UP” mark shows the side that must be upwards, and the front should also not be in contact or very close of metallic surfaces.



3.2. Unit Cover Options

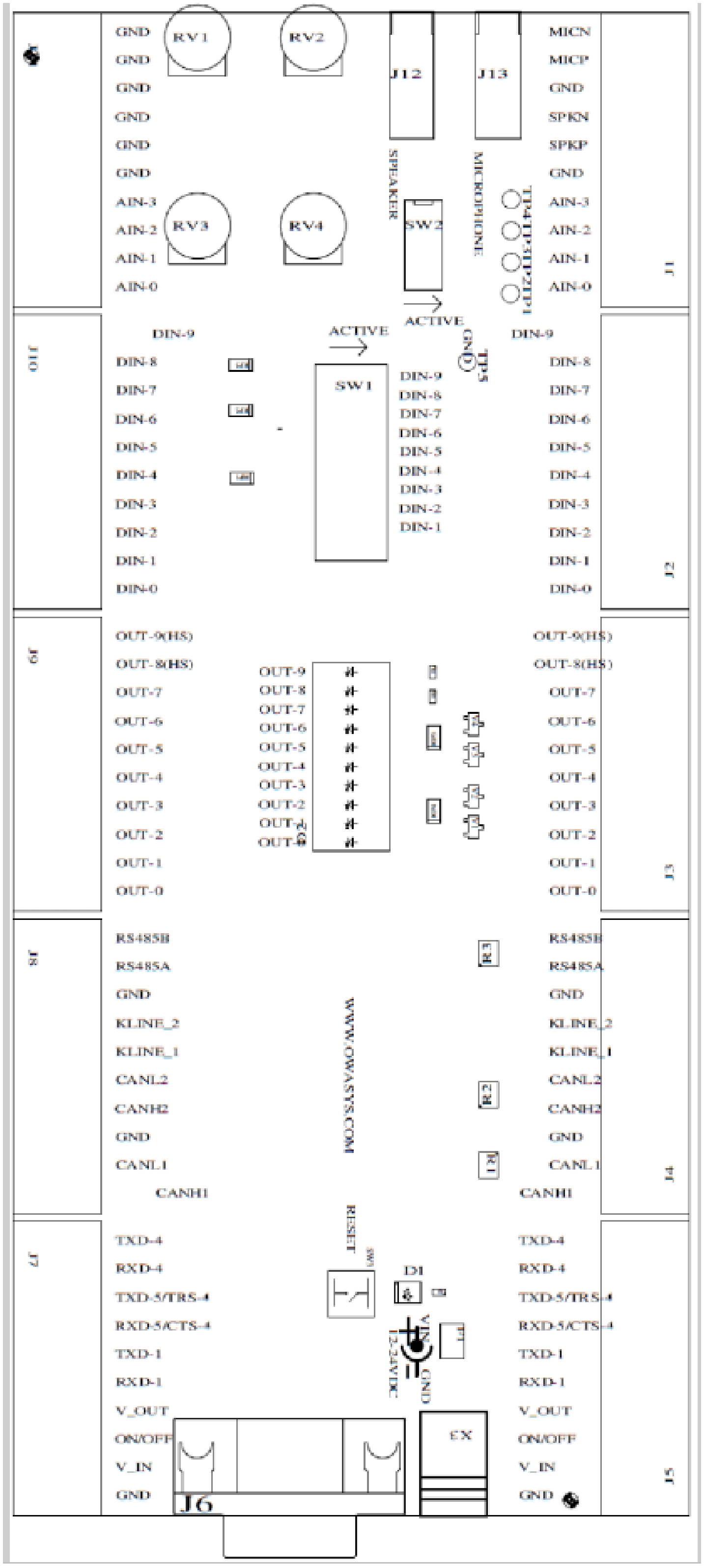
There are 3 possible rear back cover options, depending on customer requirements

- No rear cover – An open design for easier access
- IP67 rear cover – That meets IP67 requirement with no openings
- IP67 rear cover with cable gland - This back cover supports IP67 requirements, and has a cable gland to connect to an Ethernet or Rosenberger interface in the back side of the unit

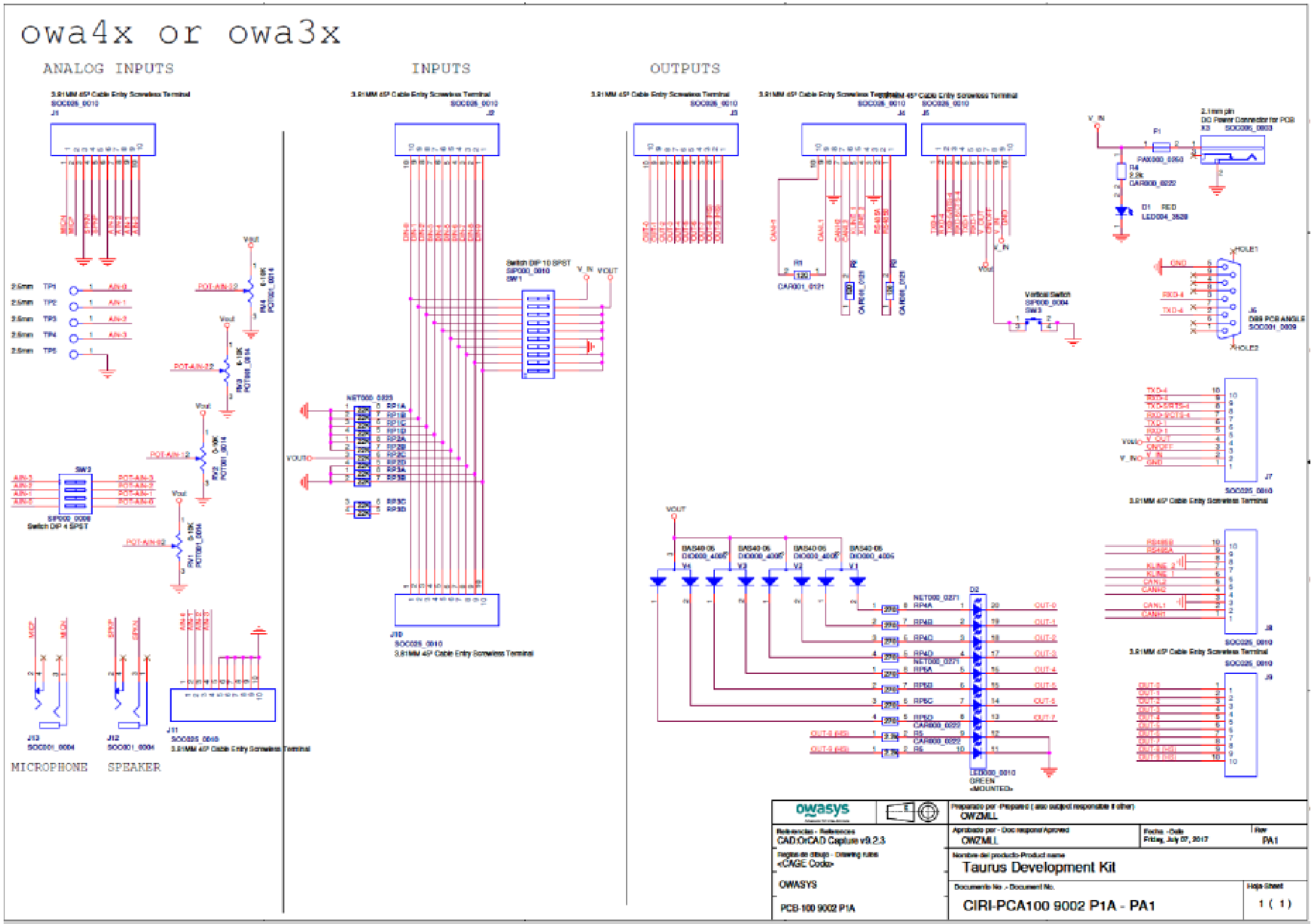


IP67 Rear cover with cable gland and M12 connection

3.3. Developer's Board Layout and Schematic



Developer's board layout



Developer’s Board Schematic

Number	Connector
J1	Audio and analog inputs
J2	Digital inputs
J3	Digital Outputs
J4	CAN, KLINE and RS485
J5	Power and RS232
X3	Jack for DC power supply, 12V or 24V
J6	DB9 for main uart, UART4
J7-11	External Devices
J12	Speaker
J13	Microphone

Power may be connected to either X3 or J7. The power input range is 9V – 48V or 9V – 36V for HW R4 or newer, and a fuse F1 is mounted in the DK board to prevent damages to other components in case of over-voltage.

Connectors J1 to J5 are used to connect the necessary signals to the unit. All the signals are available in the connectors mounted in the front in case they have to be connected to external devices.

In order to test the analog inputs, potentiometers are provided. The switch may select the potentiometer or leave the signal open for an external signal source to be connected.

Digital inputs have switches to change the default status, in order to test the behaviour of all them.

Be careful when connecting the DIO cables to the DIN J2 connector. If instead of using them as inputs, are activated the outputs that are multiplexed on them, the unit may be damaged. Have special care with DIO8 and DIO9, activating their output while connected to the J2 connector and closed switch, will damage the device.

Digital outputs have led indicators, to indicate when they have been activated.

Pin	Signal
J1-1	MICN
J1-2	MICP
J1-3	GND
J1-4	SPKN
J1-5	SPKP
J1-6	GND
J1-7	AIN-3
J1-8	AIN-2
J1-9	AIN-1
J1-10	AIN-0

Development Kit J1 Connectors

Microphone and speaker signals are connected to Jacks for the microphone and speaker connection.

Analog inputs are connected to potentiometers through switches. The switches should be in OFF position if external signal sources are to be applied.

Pin	Signal
J2-1	DIN-9
J2-2	DIN-8
J2-3	DIN-7
J2-4	DIN-6
J2-5	DIN-5
J2-6	DIN-4
J2-7	DIN-3
J2-8	DIN-2
J2-9	DIN-1
J2-10	DIN-0

Development Kit J2 Connectors

Digital inputs have pull downs in the Development kit and a switch to connect to a high level for the input testing.

Digital input 6 is inverted, so it has a pull up by default, and the switch connects the input to GND.

If external signal sources are to be connected the switches should be in OFF state.

Pin	Signal
J3-1	OUT-9(HS)
J3-2	OUT-8(HS)
J3-3	OUT-7
J3-4	OUT-6
J3-5	OUT-5
J3-6	OUT-4
J3-7	OUT-3
J3-8	OUT-2
J3-9	OUT-1
J3-10	OUT-0

Development Kit J3 Connectors

Digital outputs 0 to 7 are open drain, so they give a low value when active. Digital outputs 8 and 9 are High side so they provide a high value (V_IN) when active. All the signals have LEDs to provide a visual indication of the status of the output. The LED will be on when the output is activated.

Pin	Signal
J4-1	RS485B
J4-2	RS485A
J4-3	GND
J4-4	KLINE_2
J4-5	KLINE_1
J4-6	CANL2
J4-7	CANH2
J4-8	GND
J4-9	CANL1
J4-10	CANH1

Development Kit J4 Connectors

To use RS485 or CAN, a 120Ω termination resistor must be placed at each end of the network. In the development kit there is a termination resistor of 120 ohms soldered for each of the buses.

Pin	Signal	Function
J5-1	TXD-4	TX UART4 (Main uart)
J5-2	RXD-4	RX UART4 (Main uart)
J5-3	TXD-5/RTS-4	TX UART 5
J5-4	RXD-5/CTS-4	RX UART 5
J5-5	TXD-1	TX UART1
J5-6	RXD-1	RX UART1
J5-7	V_OUT	+5V voltage output 500mA MAX
J5-8	ON/OFF	Connect to ground to power down
J5-9	V_IN	Power input
J5-10	GND	Ground

UDK J2 Connectors

RS232 interface signals of the UART4 go to J6 DB9 connector too.

J7, J8, J9, J10, J11 connectors are directly connected to signals coming from the unit connectors in order to connect external devices or signal sources.

3.4. Connecting to the Development Kit

The steps for mounting the components in the unit are the following:

Antennas: Connect the antennas used for the required application, 3G, GNSS etc.

DB9 RS-232 Serial connector: Connect a RS232 connector, or USB to RS232 to the PC to debug the unit.

CrossLink TG Signals: Connect the signals needed to the development kit in the connectors J1 to J5. The development kit has specific use for each of the signal in each of the connections. DIOs should be connected to the DIN or DOUT depending on which feature is to be tested. RJ11 cable:

microSIM card: The microSIM card should be introduced in the SIM card compartment. The SIM card contacts shall be faced upwards. Push the SIM card until hearing a click and then lock it with the flap.

AC/DC Power supply: Provided AC/DC power supply adaptor can be connected to developers board, connector X3, and to a suitable AC main outlet. Alternatively a laboratory power supply can be connected to V_IN and GND at J7 connector.

3.4.1. Connecting to a PC

Either Linux OS or Windows OS can be used to connect to the device from a Personal Computer (PC) through serial port. The required configuration parameters are the following:

- Bit Rate: 115200 bps
- Data Bits: 8
- Parity: none
- Bit Stop: 1
- Flow Control: None

3.4.2. Windows Hyperterminal

Use Windows HyperTerminal to connect the unit to the PC configuring the serial port parameters to the values indicated in previous section.

Switch on the CrossLink TG. Once the Kernel is loaded in RAM memory and the system is up, the device waits for the user to enter a valid username to log in. This username is root and the password is root. There is also another user, “debian” and password “temppwd”.

Once logged in, the user is in the file system which has the directory structure of a usual Linux distro, in this case Debian.

To transfer a file from the PC to the unit, change to /home directory or to the directory where the file is to be stored (cd /home or cd /directory_name), type rz command and choose Transfer-> Send File… option of the HyperTerminal.

To transfer a file from the device to the PC, change to the directory where the file is, then type sz command indicating the name of the file (sz file_name) and choose the Transfer -> Receive File… option of the HyperTerminal. File transfer protocol is zmodem in both cases.

3.4.3. Linux Minicom

Run the minicom program and configure the serial port parameters to the values indicated in previous section. Minicom help is showed by typing Ctrl-A Z.

Serial port device files (/dev/ttyS0, /dev/ttyS1…) must have reading and writing permissions for all users. Log in as root and type chmod a+rw /dev/ttySx to change permissions.

Switch on the unit and wait until login prompt appears. Log in as root user with password “root” to enter into the device Operating system.

To transfer a file from the local PC to the device, change to /home directory (or to the directory where the file is to be stored), type rz command in the OS, type Control-A S so that the minicom knows the file that is to be transferred. The file transfer protocol is zmodem.

To transfer a file from the device to the local PC, change to the directory where the file to be transferred is stored, type sz command indicating the name of the file (sz file_name) and type Control-A R so that the minicom starts to receive the file. The file transfer protocol is also zmodem.

3.4.3.1. Ethernet Connections

To communicate with the unit a SSH connection can be stablished too using the Ethernet interface, if the unit features this option. The system get the SSH daemon up by default, and its configuration is:

IP: 192.168.10.1

Port: 22 (default port of SSH)

In order to use this connection, connect to this IP using these credentials:

- user: debian
- password: temppwd

By default the root user can not login to the system for security reasons, and it is not recommended to change this setting. To change any configuration, edit the file /etc/ssh/sshd_config

4. Main Hardware Features

The microprocessor is a CORTEX A8 at 800MHz clock speed supported by 512 MB of DDR3 RAM and 1GB of non-volatile NAND FLASH memory. These provide, in terms of available user space, 468 MB of RAM and 834 MB of Flash memory for the default configuration, with a basic version Debian as the operating system.

4.1. GSM/GPRS System

The unit provides GSM communication (Quad band GSM 850/900+1800/1900 as the default configuration). Audio calls, data calls and Short Message Service are the features supported by GSM.

GPRS is a widely deployed value added service of the cellular infrastructure that enables direct access to public and private data networks (Internet, corporate networks, private networks...). Using the GPRS service instead of simple GSM service significantly reduces traffic cost since resources are only allocated when data is to be sent/received.

GPRS service is class B and class 12. Four time-slots for the downlink and one for the uplink are available.

HSPA is also available as option with HSDPA Cat.8 / HSUPA Cat.6 data rates: DL 7.2Mbps and UL 5.7Mbps.

4.2. GNSS

GNSS (Global Navigation Satellite System) includes most of the available regional systems composed of a constellation of satellites orbiting the Earth, such as GPS, GLONASS, Galileo and Beidou, transmitting signals that allow the GNSS receivers to determine the receiver position (longitude, latitude and height) and time (Universal Time Coordinated, UTC).

With the GNSS module included in the unit, accurate position and time information is provided for Location Based Applications. The default datum used by the GNSS is WGS-84.

The GNSS can work in 2D navigation (viewing 3 satellites) or 3D navigation (viewing at least 4 satellites). When the GNSS starts up, it gives a valid position as soon as it sees 3 satellites, but it can only know the altitude once it sees 4 satellites. This is the reason why in the starting process, there is the possibility of a position jump. This position jump is more likely to happen the greater the altitude is.

The GNSS outputs the altitude as HAE (“Height Above Ellipsoid”) (i.e. WGS-84). But since an ellipsoid cannot model the shape of the earth perfectly, one can see some deviation from the so-called “Mean Sea Level” altitude. MSL refers to the actual sea level. The difference between these two altitudes can exceed 100m.

The supported GNSS receiver is Ublox NEO-M8N.

4.3. Hardware Interfaces

The unit is able to interface with many different devices through an extensive range of hardware signals, grouped into several connectors. Here every connector pin-out is depicted together with a brief description of every signal and electrical specifications and advice of use.

4.4. Battery Backup

The unit is provided with a small dedicated internal battery for RTC and GNSS data retention. In addition to this it is possible to install a high capacity battery to allow continuous operation of the unit without external power. Units that have this option fitted in the factory have the "/b" suffix.

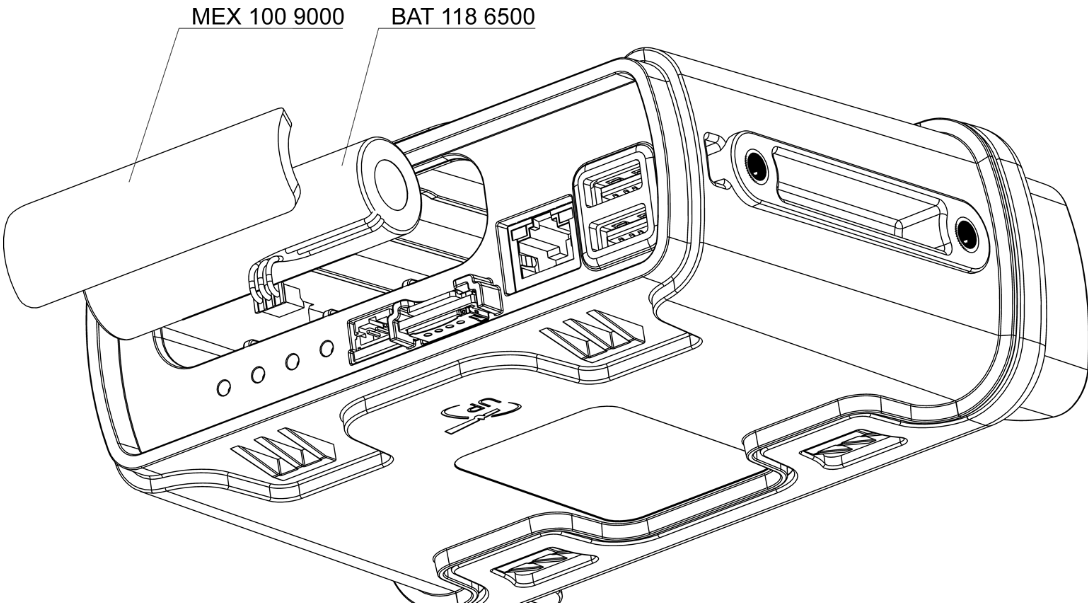
4.4.1.1. RTC and GNSS Data Battery

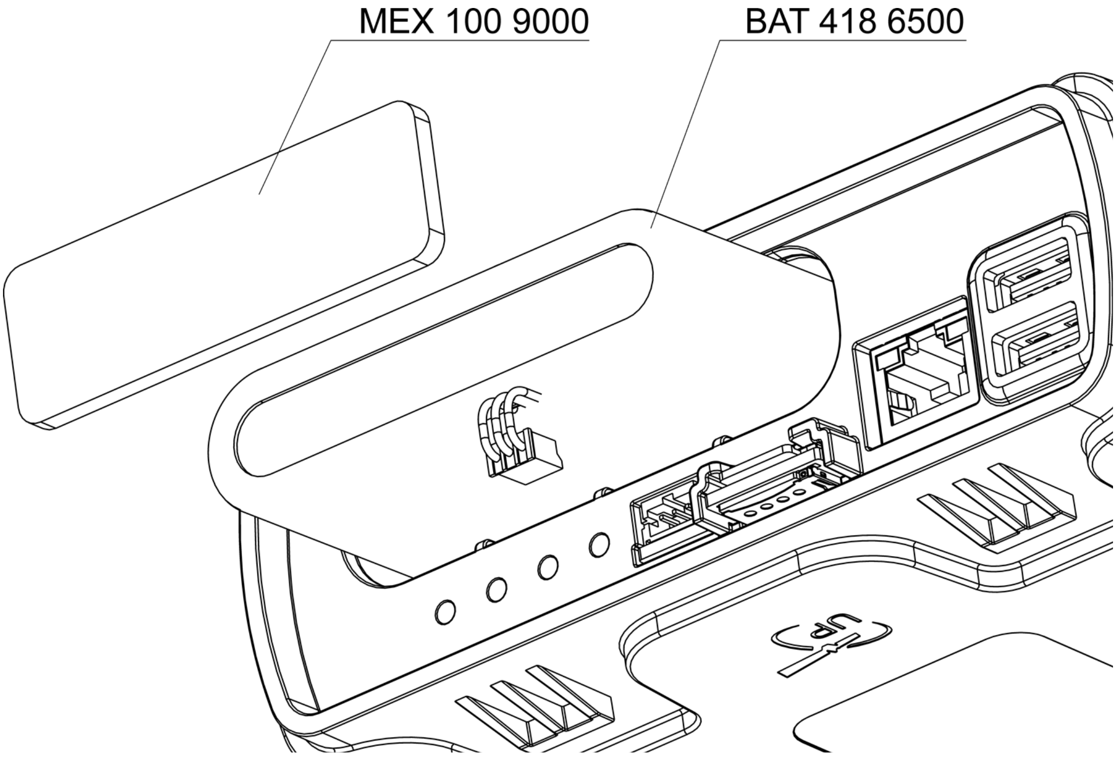
There is a small dedicated non-rechargeable battery which is supplied with all units to provide backup of the RTC and GNSS data. The RTC will be maintained for 10 years. The GNSS data is continuously backed-up.

4.4.1.2. Optional High Capacity Battery

Optional Li-ion batteries are available for the device. 3350mAh and 13400mAh high capacity battery back-up can be installed, which allows continuous operation when the main power is lost. This enables the version/b to e.g. make a final call before going into low power mode, or a similar procedure, as defined by the customer application software.

If the battery back-up is not pre-installed inside the unit, then insert it as it is shown in the figure below.

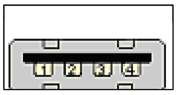




*The connectors may vary.

1.1.1.1 USB interface

USB connector with mass storage functionality.

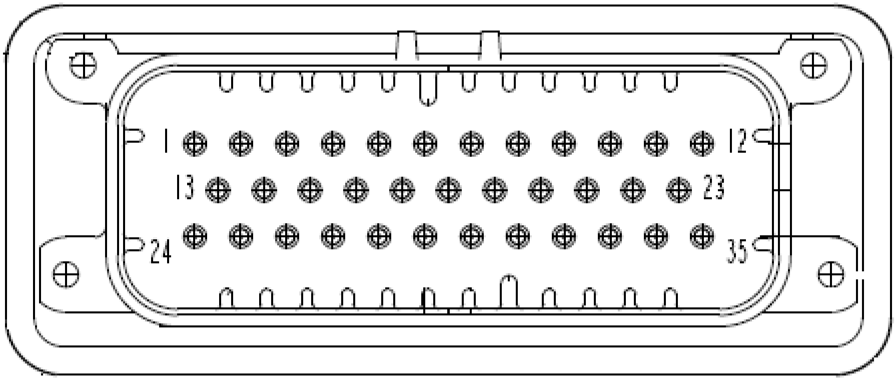


Pin	Signal	Function
1	VBUS	+5 V
2	D-	Data -
3	D+	Data +
4	GND	Ground
Connector type		USB type A
Mating connector		Male A type
Location		Back panel

1 USB connector pins

1.1.1.2 Machine Connector

To provide maximum flexibility, a 35 pin TE automotive connector is provided with power input, several RS232 interfaces, digital and analog I/Os and a CAN bus connection. The connector reference is **TE 776164-1**.



1	2	3	4	5	6	7	8	9	10	11	12
V_OUT (5V)	ON/OFF	AIN2	AIN3	CANH1	CANL1	CAN2H	CAN2L	RS485A	RS485B	MICN	MICP

13	14	15	16	17	18	19	20	21	22	23
GND	GND	AIN0	DIO-0	DIO-1 / AIN1	DIO-2 /CAN3L	DIO-3 / CAN3H	DIO-4/ CANL4	DIO-5 / CANH4	DIO-6 (iButton)	DIO-7/KLINE2

24	25	26	27	28	29	30	31	32	33	34	35
V_IN	DIO-8 (high side)	DIO-9 (high side)	TXD-4	RXD-4	TXD-5 / RTS-4	RXD-5 / CTS-4	TXD-1	RXD-1	KLINE1	SPKN	SPKP

Main connector pinout

Pin	Signal	Type	Low Level	High Level	Max Range
1	V_OUT 5V	Power output	4.5V	5.2V	
2	ON/OFF	Input	OFF 0V	ON Open	
3	AIN2	Input. Imp: 110K	0V	5.12V or 30.72V	0 to 50V
4	AIN3	Input. Imp: 110K	0V	5.12V or 30.72V	0 to 50V
5	CANH1	BUS			-36V to 36V
6	CANL1	BUS			-36V to 36V
7	CANH2	BUS			-36V to 36V
8	CANL2	BUS			-36V to 36V
9	RS485A	BUS			-70V to 70V
10	RS485B	BUS			-70V to 70V
11	MICN	AUDIO			
12	MICP	AUDIO			
13	GND	POWER			
14	GND	POWER			
15	AIN0	Input. Imp: 110K	0	5.12V or 30.72V	0 to 50V
16	DIO-0	Input. Imp: 220K	0-6V	9V -50V	0 to 50V

	DIO-0	Open Drain Output	0,6V@200mA		
17	DIO-1	Input. Imp: 180K	0-2V	3.3V -50V	0 to 50V
	DIO-1	Open Drain Output	0,6V@200mA		
	AIN1	Input. Imp: 68K	0	5.12V or 30.72V	0 to 50V
18	DIO-2	Input. Imp: 180K	0-2V	3.3V -50V	0 to 50V
	DIO-2	Open Drain Output	0,6V@200mA		
	CAN3L	BUS			-36V to 36V
19	DIO-3	Input. Imp: 180K	0-2V	3.3V -50V	0 to 50V
	DIO-3	Open Drain Output	0,6V@200mA		
	CAN3H	BUS			-36V to 36V
20	DIO-4	Input. Imp: 180K	0-2V	3.3V -50V	0 to 50V
	DIO-4	Open Drain Output	0,6V@200mA		
	CANL4	BUS			-36V to 36V
21	DIO-5	Digital Input	0-2V	3.3V -50V	0 to 50V
	DIO-5	Open Drain Output	0,6V@200mA		
	CANH4	BUS			-36V to 36V
22	DIO-6	Input. Imp: 1K to +5V	0-1V	1.6V -50V	0 to 50V
	DIO-6	Open Drain Output	0,6V@200mA		
	iButton	BUS			0V to 50V
23	DIO-7	Input. Imp: 180K	0-2V	3.3V -50V	0 to 50V
	DIO-7	Open Drain Output	0,6V@200mA		
	KLINE 2	BUS			-36V to 36V
24	V_IN	POWER			9V to 48V / 9V to 36V for unit R4
25	DIO-8(high side)	Input. Imp: 180K	0-2V	3.3V -50V	0 to Vin
	DIO-8(high side)	High Side Output 1A		VIN	
26	DIO-9(high side)	Input. Imp: 180K	0-2V	3.3V -50V	0 to Vin
	DIO-9(high side)	High Side Output 1A		VIN	
27	TXD-4	RS232			
28	RXD-4	RS232			
29	TXD-5	RS232			
	RTS-4	RS232			
30	RXD-5	RS232			
	CTS-4	RS232			

31	**TXD-1**	**RS232**
32	**RXD-1**	**RS232**
33	**KLINE1**	**BUS**
34	**SPKN**	**AUDIO**
35	**SPKP**	**AUDIO**
Mating connector manufacturer part number: TE 776164-1		
Crimps: TE 770520-1		

Machine connector pinout

4.4.1.3. Power Supply

Power supply in pin 24 of the machine connector is used to supply power to the unit. Signals used for this purpose are:

Pin	Signal	Type	Min level	Max level
24	Vin	Power in	9V	48V / 36V for R4
13	GND	Ground	-	-
2	ON/OFF	Input	OFF = 0 to 0.5V	ON = Leave OPEN

Power supply connector pins function

Vin and GND should be connected to a clean, stable supply between 9.0 and 48.0 Vdc. A cable with a current rating of more than 5A should be used.

The power supply can be made with a battery or continuous voltage supply with reinforced isolation, and limited in power to a maximum of 8A and 100 VA.

ON/OFF: Power Control Input signal. Leave open for ON, Connect to ground for OFF. Use an open collector transistor or switch to ground, but do not drive high. This input will turn the unit ON or OFF. When running from battery back-up, this signal can only be used to turn the unit OFF. If the signal is then released and there is no power on Vin, then the unit will remain OFF.

4.4.1.4. Digital Inputs/Outputs (DIO0..9)

The unit provides up to 10 configurable digital Input/Outputs, from DIO0 to DIO9.

These pins can be configured as inputs or as outputs. Note that if the pin is configured as an output it cannot be used as an input, and may be damaged if a voltage is applied while it is configured as an output. Hence ensure that the corresponding output pin is OFF before using it as an input. See the programming guide for more details.

The digital inputs are not TTL compatible, and they can withstand inputs up to 50V so that sensors and switches with higher voltages can be used. For example, in an automotive application, a switch may be connected to the positive supply giving an input of 14V or 28V. The input impedance is 180K, except for DIN0 with 220K and DIN6, which has 1K to +5V.

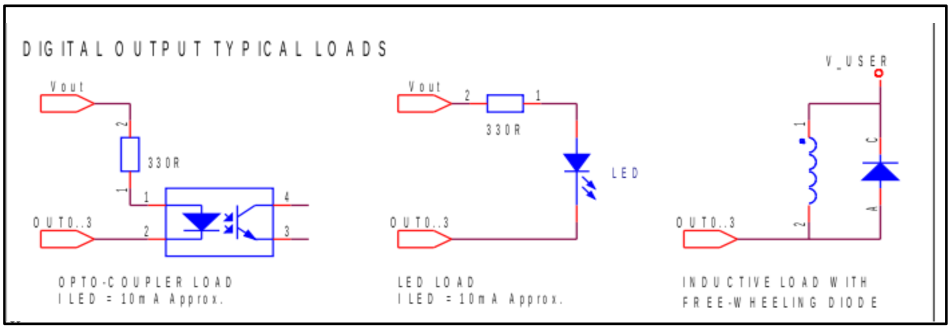
All inputs are inverted except DIO6, which is ready to take a iButton on it, and the range is also different when using this pin as input (see connector table).

If the digital output is set to 1 in the software it will be OFF and the transistor will pull the output pin to ground giving a low level. If it is set to 0 in the software it will be ON and the transistor will be open. Note that there are no pull-ups on these pins so to obtain a high level when the output is ON requires an external pull-up to a positive supply.

The DIO0 to DIO7 outputs are open collector transistor type capable of switching up to 50V and sinking up to 200mA.

With these open collector outputs the load should be connected between the output pin and a positive supply. The positive supply could be provided by the user, or from the pin Vout (pin 23). The maximum output current from Vout is 500mA, which should be sufficient to drive up to 8 LEDs, Opto-couplers or Solid State Relays at 10-12mA each. A typical connection for one output is shown below.

For an inductive load (such as a relay or motor) it is mandatory to connect a free-wheeling diode to provide a return path to the supply for the inductive energy, as shown below. Otherwise the resulting voltage spikes during switch off could damage the output circuitry.



DIO9 are high side switched digital outputs with Vin as input voltage and with a maximum of 1A current can be drawn from them. If loads with higher power are required, the user can connect them to these outputs. The maximum inductive load is 100 mH.

4.4.1.5. Analog Inputs

The 4 analog inputs AIN0 to AIN3 present the following characteristics:

Resolution	12 bits
Input range	0-5.12V (1.25mV per bit) or 0-30.72V (7.5mV per bit)
Accuracy	±1% typ. @ 25ºC
Input impedance	110K (AIN1 68K)

Characteristics of analog inputs

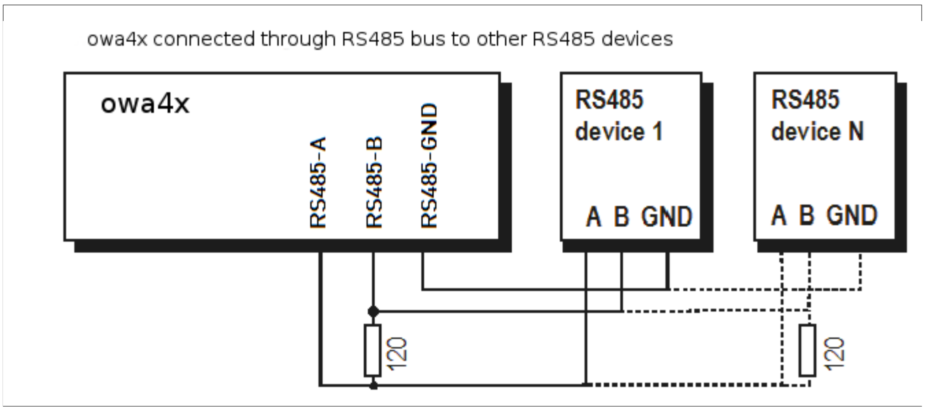
4.4.1.6. RS485

The unit provides an RS485 bus driver.

The number of devices that can be connected in parallel to this interface are 32.

In the following diagram an example of a set of devices connected to the owa2x through a RS485 bus is shown.

First and last devices in the bus should be equipped with terminating resistors of 120 ohms. the unit does not have this terminating resistor internally so it must be added externally as depicted in next figure.



4.4.1.7. CAN

The unit includes a CAN transceiver V2.0B at 1 Mbaud.
For more detailed information see the Software Programming Manual.

Optionally three more CAN transceivers can be mounted in the unit to have 2 different physical CAN drivers available simultaneously. This second CAN would be mounted in the optional board.

4.4.1.8. K Line

To connect to a tachograph or a K Line communicating device the unit provides optionally 2 K Line interfaces:

1 bidirectional K Line, Kline1, on pin 33 of the machine connector to transmit and receive K Line data.

1 unidirectional K Line, Kline2, on pin 23 of the machine connector to receive K Line data.

4.4.1.9. Ground

The unit provides a ground connection for all interfaces, digital I/Os, RS485, CAN and analog inputs.

4.4.1.10. Speaker connection

Audio output is of BTL type (Bridge Tied Load). That means none of the speaker signals are internally connected to ground.

Never connect any terminal of the loudspeaker output to ground or power supply as it may damage either the speaker or the unit.

The recommended external loudspeaker has the following features:

Impedance higher than 3 Ω

Required power handling: 1.5 W over 8 Ω and 2.5 W over 4 Ω

Frequency responser between 300 Hz – 3.4 Khz (recommended)

4.4.1.11. Microphone connection

Microphone input in of unbalanced type (signal is referred to ground) and accepts electret type microphones. Those are the type of microphones most widely used for consumer purposes. Yet they provide very good performance at a very low price. They need a clean, very stable power supply (usually called V_{BIAS}) that is provided by the unit. Signal Input and V_{BIAS} share the same pin in the connector, as this is the connection usually found in commercially available microphones.

Microphone sensitivity is usually expressed in terms of voltage/acoustic pressure (dB/μPa.). Higher value means better sensitivity, so a microphone rated at -46 dB/μPa is less sensitive than another at -36 dB/μPa

The recommended features of the microphone are:

- Use only electret type microphones
- High sensitivity is recommended for hands-free applications (in the range of -46 dB/μPa
- Impedance equal or higher than 1 KΩ
- Connector with signal and BIAS tied together

4.4.2. microSIM Card Holder

The SIM Card Holder is a push-push connector and it is for the GSM/GPRS service.

Connector type	Push push microSIM holder
Card type supported	3V and 1.8V
Location	Rear panel

4.4.3. GSM Antenna

Connector type	FAKRA Plug male type D(Bordeaux)
Mating connector	FAKRA jack female type D(Bordeaux)
Location	Front panel

To keep the IP67 waterproof features, a special fakra connection is needed, with code AIMMET 3659NTG14D1. Please contact your supplier to get more information on this connector.

4.4.4. GNSS Antenna

This product has a built-in GNSS module. Use only external GNSS active antenna connected to the GNSS connector provided in the front panel.

Connector type	FAKRA Plug male type C(blue)
Mating connector	FAKRA jack female type C(blue)
Location	Front panel

By means of internal circuitry, the unit is able to detect if the GNSS antenna is operating correctly, if there is an open circuit or a short circuit.

To keep the IP67 waterproof features, a special fakra connection is needed, with code AIMMET 3659NTG14C1. Please contact your supplier to get more information on this connector.

4.4.5. LEDs

There are 6 LEDs, 4 at the front of the unit, see 2, and 2 at the rear with the optional Ethernet RJ45 connector, see 2.

These are the LEDs colours and functions:

Name	Colour	Indication	Comment
GNSS	Orange	GNSS activity	Can be controlled by user
STATE	Red	As defined by application	Controlled by user
PWR	Green	As defined by application	Controlled by user
RADIO	Yellow	GSM activity	Can be controlled by user
LINK	Green	Ethernet Link	Controlled by ETH (optional)
100 Mbps	Yellow	Ethernet 100Mbps	Controlled by ETH (optional)

The green LED on the unit is user programmable. For more information about the control of the Red LED see the Software Programming Manual.

The yellow LED is used to indicate the status of the GSM Module as shown in the table below:

GSM Module State	Yellow LED
OFF	Off
ON without cellular coverage	On fast blinking
ON with cellular coverage	On slow blinking

The yellow LED is off until the device is switched ON and the GSM module is powered on and registered to the network.

This yellow LED may be configured by the user to follow its own state code. However, the user must enable it explicitly first, as by default is controlled by the GSM module and this has preference over the usage from the user application.

The orange LED on the unit is controlled by the GNSS module and gives one pulse per second when the GNSS module has coverage.

GNSS Module State	Orange LED
OFF	Off
ON without valid fix	Off
ON with valid fix	One pulse per second

The orange LED may also be configured by the user to follow its own state code. However, the user must enable it explicitly first, as by default is controlled by the GNSS module, and it has preference over the usage from the user application.

The red LED on the is user programmable. For more information about the control of the Red LED, see *the Software Programming Manual.*

Ethernet activity is indicated by means of the two LEDs.

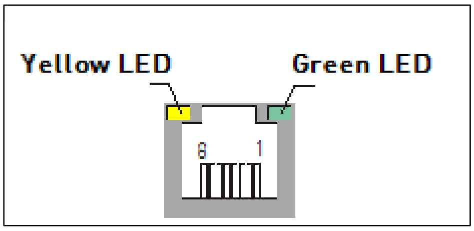
State	Yellow LED
OFF	OFF
Ethernet 100Mbps	ON

Yellow LED indications

State	Green LED
OFF	OFF
Link OK	ON
Activity	ON (blinking)

Green LED indications

4.4.6. RJ45



This optional connector is at the rear of the unit, is intended for an Ethernet communication.

Pin	Signal	Function
1	TX+	Transmission Positive
2	TX-	Transmission Negative
3	RX+	Reception Positive
4	-	Reserved
5	-	Reserved
6	RX-	Reception Negative
7	-	Reserved
8	-	Reserved
Connector type		RJ45, 8 pin socket
Mating connector		RJ45 jack
Location		Rear panel

Ethernet connector

4.4.7. SIM Card Holder

The SIM Card Holder provided is for the GSM/GPRS service. Both 3V and 1,8V SIMs are supported. Insert the SIM card with the connectors facing upwards in its slot, right under the microSD card slot, as shown in 2

4.4.8. Bluetooth/WiFi Antennas

The antennas for Bluetooth and Wi-Fi are internal.

4.4.9. uSD Card

The microSD card is inserted in its slot. This holder is also under the cover at the rear of the unit, right above the SIM card holder as shown in 2

4.4.10. Internal Temperature

The unit has an internal sensor to measure its internal temperature.

This temperature sensor cannot be used to measure external environment temperature due to the difference between this temperature and the internal temperature.

The temperature sensor is connected to an internal analog input. See the Software and Programming Manual to get further info in this input and how to read it.

- Temperature range: -40ºC to +85ºC
- Accuracy: ±5ºC

4.4.11. System Time and HW time

The unit is provided with two different clocks:

The **CPU system time**: This clock is able to maintain system time in normal operation but loses the time when the unit is powered off or is put into a low power mode.

The **RTC module:** This is a dedicated Real Time Clock (RTC) module which is battery backed up so that the time remains correct even when power is removed. The duration of the internal battery is 10 years.

When the unit returns from OFF or STANDBY modes, the CPU system time is automatically updated with the value in the RTC time. Hence any changes to the time should be performed on the RTC time as this is the 'master'.

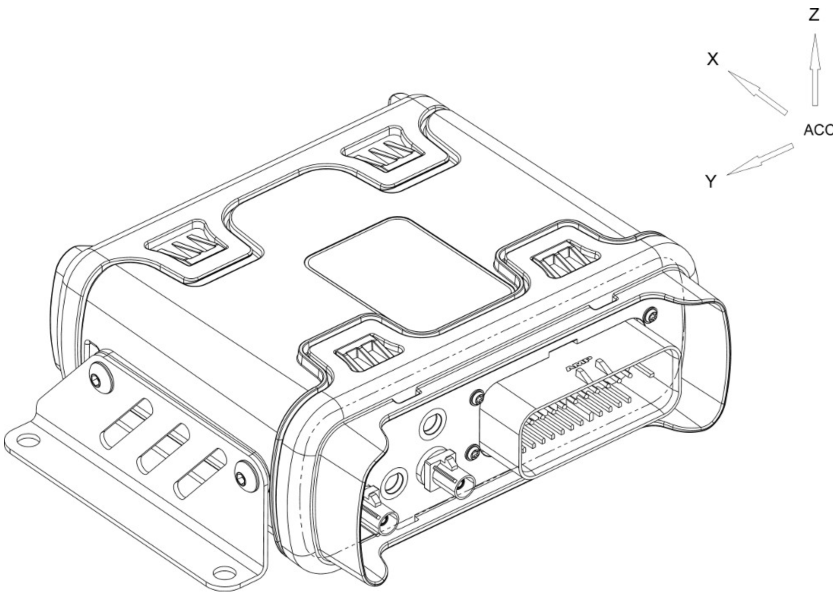
Some models have an additional and more precise time reference that is obtained from GNSS data. When available, use this as a reference to set the RTC value.

4.4.12. Accelerometer

The unit carries by default a 3 axis accelerometer with ±2g/±4g/±8g/±16g configurable ranges and 1mg/2mg/4mg/12mg sensitivity respectively.

The accelerometer can be set in the wake up mask of Standby or OFF mode, in order to wake up when there is a movement bigger than the preconfigured one.

The data from the axis registers can also be obtained using a function of the RTU library, with or without gravity acceleration filter, which can be of use to get the inclination of the device. See API for more information on these functions.



Accelerometer axis

4.5. Watchdog

There is a HW watchdog available under /dev/watchdog and a SW watchdog that can be controlled via systemd. You can get more information on how to use them in the Software and Programming Manual.

4.5.1. pmsrv

There is also a service that controls the low power modes and communication with the IOs, **pmsrv**, that must be always running. When for whatever reason this service is down for longer than one minute, the system will reboot.

4.5.2. Optional Features

The non standard unit may be provided with a number of optional features, which can customize the product to the user needs.

4.5.3. 3rd and 4th CAN

A third and fourth CAN bus can be provided optionally in pins 18 and 19 for CAN3 and pins 20 and 21 for CAN4.

HW R4 and onwards supports CAN-FD for the 3rd and 4th CAN interfaces.

4.5.4. Wi-Fi

Wi-Fi module can be integrated as an option.

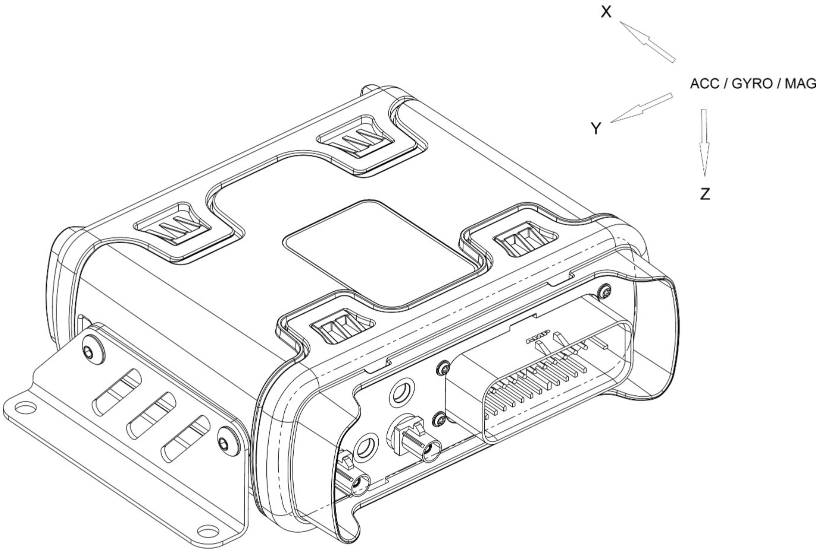
4.5.5. Ethernet

10/100 Ethernet can be provided with a RJ45 connector at the rear side of the unit.

The unit has also an optional back cover rated IP67 with a cable gland for the Ethernet cable. It has been designed and tested with Yamaichi Y-concable4 (external diameter 6.8mm) and Y-conplug-11 connector.

4.5.6. Gyro / accelerometer /magnetometer

The optional gyroscope/accelerometer/magnetometer works with the IO drivers. Check the Software Programming manual to get more information on the usage of this module.



Accelerometer / Gyroscope axis

4.5.7. Bluetooth / BLE

Bluetooth/BLE module can be provided as option.

4.5.8. USB hosts

One or two USB host connectors can be provided as option.

4.5.9. KLine2

A second KLine interface may be installed as option.

4.5.10. Trusted Platform Module (TPM)

The unit has the option to mount a TPM2.0 (ISO/IEC 11889).

This TPM is tamper resistant against invasive silicon, fault injection and side channel attacks, and provides protected storage, device hardware identity, secure generation of cryptographic keys and a random number generator.

5. Power management

From the functional point of view, the unit can work in one of the following functional modes:

State	Short Description	Active Circuits	Wakeup sources	Wake up time	Power sources
No Power	No power supply available.	RTC	V_IN Main bat	n.a.	Backup battery
OFF	Minimal power.	RTC Dig inputs Movement RS232-RX0	RTC Main bat DIN0..9 Movement RX0	Slow (boot)	V_IN, Main Battery
STANDBY	Low power with fast response to wakeup events.	RTC Movement Dig inputs CPU (halt) RAM (refresh) GSM RS232 CAN	RTC Main bat DIN0..9 Movement RX0 GSM event CAN1	Fast	V_IN, Main Battery
RUN	All circuits running.	All	n.a. All events monitored	n.a.	V_IN, Main Battery

functional modes

Once the unit is switched on, it starts executing the boot loader. Then, if the user stops the process, the unit will remain in Service Mode. Otherwise the unit continues loading the kernel and the file system.

The customer application is able to select the desired mode, usually based on the current consumptions.

In Run mode the microprocessor is running at 800 MHz. If the customer application wants or needs lower speed, it can send the unit to either Standby or Off mode.

From Standby and Off modes, the customer application can wake up the unit to Run mode with an event preconfigured with a mask. The difference between Standby and Off modes are the possible events that will be listened for as trigger conditions.

These modes offer the possibility to build powerful and flexible applications that can handle situations where power supply can be constrained.

5.1. System halt

The unit is designed to run at all times, as it will be installed in remote places. As such, the power management will reset the unit whenever the main system is not running, and that means that the standard “poweroff –halt” or “reboot –halt” will not work as expected, and in less than 3 minutes the system will be rebooted.

In case that the user wants to halt the system, a custom command, **OwaTurnOff**, must be used instead. This command makes use of the power management functionality to set the unit in STOP mode.

These are the options that can be passed to this command:

```
---

-t <secs>   (CPU is powered off after <secs>

-m <mask>   (<mask> selects events to retrun from OFF state)

               <mask> is the sum of the desired values

               1: moving

               2: power on

               4: console

               128: IO 0

               256: IO 1

               512: IO 2

               1024: IO 3

               2048: IO 4

               4096: IO 5

               8192: IO 6

               16384: IO 7

               32768: IO 8

               65536: IO 9

               524288: Remove VOUT in OFF state

-h          (show this help)

                 ---
```

6. Firmware Specifications

In order to manage the platform resources, a complete library of APIs, for GNSS management, Internet connection, management of the interfaces, GSM/GPRS functions and other services is made available. Thus, the developer does not need to consider about low level hardware drivers and protocols, and can focus on the application by means of user-friendly APIs.

6.1. Boot Loader

The boot loader is the booting system that manages the kernel and file system flashing.

It waits for 0 seconds before starting up Linux, but this time it can be changed with the variable *bootdelay*. During this time, if the boot loader receives a character different than SPACE through serial port, it starts up Linux immediately. If it receives SPACE, it enters in command mode; therefore the received characters are interpreted as commands. Once finished in command mode, send boot command for Linux start up.

The boot can be customized with the necessary u-boot environment variables, which can help the user to recover from unexpected crashes of the FS that prevents it to run, or remotely make an OTA upgrade of the FW. It follows an example to show how the u-boot environment variables can be used for this purpose:

```
fw_setenv update 1
fw_setenv availableupdate 'ubifsls /home/debian/update/ubi.img'
fw_setenv flashupdate 'ubifsload 0x82000000 /home/debian/update/ubi.img;
nand erase.part NAND.file-system; nand write 0x82000000 NAND.file-system
${filesize}; setenv update; saveenv'
fw_setenv updatecmd 'ubi part NAND.file-system; ubifsmount ubi0:rootfs; if
run availableupdate; then run flashupdate; else echo "Nothing to update";
fi'
fw_setenv bootcmd 'if test -n $update; then echo Running update!!!!...; run
updatecmd; fi; run nandboot'
```

fw_setenv and fw_printenv are the tools you need to set and get environment variables from the system.

As shown, u-boot variables can contain conditional statements that can be run by the user.

The example will flash the file system if a file /home/debian/update/ubi.img exists. **Take into account that this is an example, which could (should) be easily improved (e.g. from the security point of view), as writing the flash memory is a critical function.**

Following with the update example, a different approach would be to boot the backup filesystem, that could be ready configured to start an updater agent that would try to flash the update images from a different partition, or to connect to the server to retrieve the needed images to flash.

6.2. Linux Kernel

This is a Linux standard kernel, version 4.4.19. As this is a standard kernel, PC developed applications are easily made compatible with the platform. Also this kernel can be updated to follow standard kernel revisions.

6.3. File System

The file system is based in a Debian distribution.

6.3.1. Recovery File System

The default composition of the flash, includes a recovery file system based in busybox-1.27.2, or in buildroot for FW greater and 1.1.0. This file system can be booted in case the Debian file system can not be booted, for example after changing critical files that prevent the unit from booting up.

The default credentials are root / owasys.

6.4. Starting to develop the customer application

Once the cross compiler is installed, the user can use this compiler in the desired development environment, for example, Netbeans.

To use the available APIs library functions, add the includes to files in application header files and compile the application using the provided cross compiler.

Files to be included in customer header files are:

<owa4x/RTUControlDefs.h>

<owa4x/IOs_ModuleDefs.h>

<owa4x/ GSM_ModuleDefs.h>

<owa4x/ INET_ModuleDefs.h>

<owa4x/ GPS2_ModuleDefs.h>

<owa4x/owcomdefs.h>

<owa4x/owerrors.h>

All API module libraries have been written in C Language using the **KDeveloper** environment with gcc 5.3 cross-compiler tool, under LINUX Xubuntu 16.04 Operating system. It is recommended, to avoid problems from the development environment, to use the same language, compiler, and environment when possible.

Usually, a Client application is comprised of a main executable program, and a set of libraries. This set of libraries can be of two types: system libraries, both static and dynamically loaded; and API module libraries, always used in a dynamic way.

Once an application is defined, what the user has to do is select which libraries will be used and get pointers to the needed functions, using them as C language normal functions. For a better use of system resources, it is recommended to unload all functions of a library, as well as the library itself, when it is known that it will not be used any more in the program's scope. Also, it is not necessary to get all functions pointers in a library, if some of them will not be used in the program: it is a waste of memory, and system resources which can be useful for other tasks, and will improve the system behaviour in general.

Another alternative is to use the .h header of the library to compile the program against the library.

6.4.1. Available APIs

In this section there is an overview of the available APIs and an explanation of their functionality. For further information about the APIs and their functions, it is recommended to read through Programming Guide and API **Error! Bookmark not defined.**

6.4.1.1. API for Control (RTU)

This API provides functions to enter the low power modes and RTC.

6.4.1.2. API for GSM/GPRS

The communication with the GSM module is done with the GSM and GPRS APIs. Sending and receiving SMSs and making dial calls are some of the functions provided.

6.4.1.3. API for GNSS

Getting the time and positioning info are some developed functions included in this API.

6.4.2. API for IOs

The functions of this API facilitate the management of I/Os, UARTs and audio signals.

6.4.3. CAN Bus FMS API

There is an available FMS library, that can run on the unit under a license. This library provides the most important data for the FMS bus available through a couple of simple functions. This library must be first enabled to use it, so if desired please request this from your distributor.

6.4.4. RS232 Driver

RS232 driver is used to control all the RS232 interfaces at the machine connector.

6.5. Software Application Notes

Some application notes show the usage of the different available libraries:

- owa4x_AN3. Use of IO.
- owa4x_AN5. Use of GNSS.
- owa4x_AN8. Use of RTU.
- owa4x_AN24. Use of GSM.

7. Updating FW

The FW is composed of following images:

- MLO
- u-boot
- dtb
- Kernel
- Backup FS

Debian FS (the version is written in /etc/issue.owa)

Normally the MLO and u-boot and backup FS should not be updated as previously specified. In the other hand, the dtb, Kernel and Debian FS should be flashed together.

The update is possible in either of the two following ways:

7.1. uSD

A FW image can be flashed to the NAND of the unit from the bootloader, that previously has been stored in a uSD card.

These are the instructions to flash the MLO, u-boot, DTB, kernel, backup FS and Debian FS.

Power up the unit and press the space bar to enter the bootloader prompt

Insert the uSD card with the images to flash

Confirm a FAT32 formatted SD card is accessible

```
mmc rescan
```

List the files in the uSD card

```
fatls mmc 0
```

The following commands are the ones used to load the images, erase the partitions and write the images on them. The commands have been protected with an **if** clause to continue with the process only when the previous commands are successful.

MLO update. **ONLY IF U-BOOT HAS BEEN UPDATED!** Check the u-boot version with command version at the u-boot prompt, and flash the MLO ONLY if the u-boot version at the dev zone is newer. This step is critical and could brick the unit, so make sure of inserting the commands correctly.

```
if test $? = 0; then fatload mmc 0 0x82000000 MLO; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.SPL; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.SPL ${filesize}; else echo FAILED LAST
COMMAND!!; fi
```

u-boot update. **ONLY IF U-BOOT HAS BEEN UPDATED!** Check the u-boot version with command version at the u-boot prompt, and flash the u-boot ONLY if the u-boot version at the dev zone is newer. This step is critical and could brick the unit, so make sure of inserting the commands correctly.

```
if test $? = 0; then fatload mmc 0 0x82000000 u-boot.img; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.u-boot; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.u-boot ${filesize}; else echo FAILED
LAST COMMAND!!; fi
```

DTB update[1].

```
if test $? = 0; then fatload mmc 0 0x82000000 am335x-owasys.dtb; else echo FAILED
LAST COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.u-boot-spl-os; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.u-boot-spl-os ${filesize}; else echo
FAILED LAST COMMAND!!; fi
```

Kernel update.

```
if test $? = 0; then fatload mmc 0 0x82000000 zImage; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.kernel; else echo FAILED LAST COMMAND!!; fi
```

[1] The gyroscope needs a different dtb, take care when downloading it from the developers zone

```
if test $? = 0; then nand write 0x82000000 NAND.kernel ${filesize}; else echo FAILED
LAST COMMAND!!; fi
```

Backup FS update.

```
if test $? = 0; then fatload mmc 0 0x82000000 ubi.img.backup1; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.file-system.backup1; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.file-system.backup1 ${filesize}; else
echo FAILED LAST COMMAND!!; fi
```

FS update.

```
if test $? = 0; then fatload mmc 0 0x82000000 ubi.img; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part  NAND.file-system; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.file-system ${filesize}; else echo
FAILED LAST COMMAND!!; fi
```

7.2. TFTP server

The unit FW components can be saved in a TFTP server in the same LAN where the unit is connected, or which is accessible to it, and the unit can get them using a set of commands.

Take care with this process, over all with MLO and u-boot images reflashing, as writing incorrectly on these could brick the unit. It is important to check that each command has been correct.

These are the instruction steps for the whole process:

Power up the unit and press the space bar to enter the bootloader prompt

Connect the Ethernet cable to the RJ45 connector

Set the environment variables, for example when connecting the unit in a LAN with private IP network 192.168.100.255 and the TFTP server in 192.168.100.1:

```
env set ipaddr 192.168.100.2
env set serverip 192.168.100.1
```

The following commands are the ones used to load the images, erase the partitions and write the images on them. The commands have been protected with an **if** clause to continue with the process only when the previous commands are successful:

MLO update. **ONLY IF U-BOOT HAS BEEN UPDATED!** Check the u-boot version with command version at the u-boot prompt, and flash the MLO ONLY if the u-boot version at the dev zone is newer. This step is critical and could brick the unit, so make sure of inserting the commands correctly.

```
if test $? = 0; then tftp 0x82000000 MLO; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.SPL; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.SPL ${filesize}; else echo FAILED LAST
COMMAND!!; fi
```

u-boot update. **ONLY IF U-BOOT HAS BEEN UPDATED!** Check the u-boot version with command version at the u-boot prompt, and flash the u-boot ONLY if the u-boot version at the dev zone is newer. This step is critical and could brick the unit, so make sure of inserting the commands correctly.

```
if test $? = 0; then tftp 0x82000000 u-boot.img; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.u-boot; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.u-boot ${filesize}; else echo FAILED
LAST COMMAND!!; fi
```

DTB update

```
if test $? = 0; then tftp 0x82000000 am335x-owasys.dtb; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.u-boot-spl-os; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.u-boot-spl-os ${filesize}; else echo
FAILED LAST COMMAND!!; fi
```

Kernel update

```
if test $? = 0; then tftp 0x82000000 zImage; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.kernel; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.kernel ${filesize}; else echo FAILED
LAST COMMAND!!; fi
```

Backup FS update

```
if test $? = 0; then tftp 0x82000000 ubi.img.backup1; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand erase.part NAND.file-system.backup1; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.file-system.backup1 ${filesize}; else
echo FAILED LAST COMMAND!!; fi
```

Debian FS update

```
if test $? = 0; then tftp 0x82000000 ubi.img; else echo FAILED LAST COMMAND!!; fi
if test $? = 0; then nand erase.part  NAND.file-system; else echo FAILED LAST
COMMAND!!; fi
if test $? = 0; then nand write 0x82000000 NAND.file-system ${filesize}; else echo
FAILED LAST COMMAND!!; fi
```

7.3. Creating a customized UBIFS image from FS tree

The user can create a customized UBIFS, based on the rootfs available online, unit FS tree. After decompressing it in the PC and make the desired changes to it, *mkfs.ubifs* and *ubinize* commands may be used to create the image. The resulting image will have only one UBI volume storing UBIFS file-system.

It follows the commands and configuration files needed to generate the FS image for each possible FS version.

7.3.1. Commands to create a FS <= 1.0.8 and HW < R2

```
$ sudo mkfs.ubifs -F -q -r rootfs -m 4096 -e 520192 -c 1980 -o ubifs.img
$ ubinize -o ubi.img -m 4096 -p 512KiB -s 1024 ubinize.cfg
```

where ubinize.cfg contains:

```
[ubifs]
mode=ubi
image=ubifs.img
vol_id=0
vol_size=953MiB
vol_type=dynamic
vol_name=rootfs
vol_flags=autoresize
```

1.1.2 Commands to create a FS >= 1.1.0 and HW < R2:

```
$ sudo mkfs.ubifs -F -q -r debian_9.6_rootfs -m 4096 -e 520192 -c 1890 -o
ubifs9.6-524288.img
$ sudo ubinize -o ubi9.6-524288.img -m 4096 -p 512KiB -s 1024 ubinize9.6-
512k.cfg
```

where ubinize.cfg contains:

```
[ubifs]
mode=ubi
image=ubifs9.6-524288.img
vol_id=0
vol_size=910MiB
vol_type=dynamic
vol_name=rootfs
vol_flags=autoresize
```

1.1.3 Commands to create a FS >= 1.1.0 and with HW >= R2:

```
$ sudo mkfs.ubifs -F -q -r debian_9.6_rootfs -m 4096 -e 258048 -c 3780 -o
ubifs9.6-262144.img
$ sudo ubinize -o ubi9.6-262144.img -m 4096 -p 256KiB -s 1024 ubinize9.6-
256k.cfg
```

where ubinize.cfg contains:

```
[ubifs]
mode=ubi
image=ubifs9.6-262144.img
vol_id=0
vol_size=904MiB
vol_type=dynamic
vol_name=rootfs
vol_flags=autoresize
```

7.3.2. Configuration file description

The options used in the creation of the filesystem image mean:

```
-r <rootfs>: to create an UBIFS image containing the rootfs directory;
-m 4096: to set the minimum input/output unit size of the flash (NAND page in
this case);
-e 520192: to indicate the logical eraseblock size of the UBI volume;
-c 1980: to set the maximum file-system size in logical eraseblocks; the
resulting filesystem may be put on volumes up to about 982MiB
-p 512KiB: to set the physical eraseblock size of the flash chip;
-s 1024: to indicate that the flash supports sub-pages and sub-page size is
1024 bytes;
```

About the generated UBIFS [5], it is important to note that the command *df* does not get the same results as with EXT2 or EXT3. The flash space reported by tools like *df* is probably wrong, since UBIFS makes pessimistic calculations and reports its size as the absolutely minimum amount of space which it can fit in the worst-case scenario.

7.4. Creating a customized UBIFS image from current FS

The user can copy the filesystem that is currently in the unit executing the following command:

```
tar --exclude="/dev" --exclude="/run" \
--exclude="/tmp" --exclude="/temp" \
--exclude="/proc" --exclude="/mnt" \
--exclude="/sys" --exclude="/lost+found" \
--exclude="/etc/udev/rules.d/70-persistent-net.rules" \
--exclude="/etc/device.info" \
--exclude="/etc/pmsrv.info" \
--exclude="/var/log" \
--exclude="/var/lib/bluetooth" \
--exclude="/var/lib/dhcp" \
--exclude="/var/tmp" \
--exclude="/etc/ssh/ssh_host_*"  \
-cpzf ./rootfs.tar.gz /
```

And extract it with this command:

```
$ sudo tar --numeric-owner -zxvf rootfs.tar.gz -C rootfs
```

The original `/etc/udev/rules.d/70-persistent-net.rules` file should be copied in the rootfs folder to avoid using the same MAC address on multiple devices.

7.5. Starting the recovery image

In case of failure or misconfiguration of the Debian file system, a recovery image can be started. The purpose of this is to mount the Debian partition and to solve its problems, in order to be able of starting it again.

To load the recovery image, enter the u-boot prompt and type these commands:

```
setenv nandroot "ubi0:rootfs rw ubi.mtd=NAND.file-system.backup1,1024"
boot
```

Once booted, login in the recovery system, the default credentials are:

user: root

password: owasys

And check the partitions:

```
cat /proc/mtd
```

If the file /dev/ubi_ctrl doesn't yet exist, it has to be created using mknod.
The correct parameters should be read from /sys/class/misc/ubi_ctrl/dev.

```
cat /sys/class/misc/ubi_ctrl/dev
mknod /dev/ubi_ctrl c 10 58
```

Check what ubifs volume has been generated (e.g /dev/ubi1) and mount the partition (-m 10 is due to the mtd partition number):

```
ubiattach /dev/ubi_ctrl -m 10
mount -t ubifs /dev/ubi1_0 /tmp
```

After the changes, boot normally the device.

7.5.1. FW >= 1.1.x

With the FW 1.1.x a bigger backup FS has been included, that allows the user not only to mount the main FS as earlier, but also has the minimum needed features to be able to start a GPRS session and communicate with a server, which can be of help to recover from unknown states even in remote places, or deploy an update strategy for OTA updates of the FW images.

The programs can be started using system V, saving in etc/init.d the script to start and stop it. The following script example starts and stops a user program called gsm.

```
#!/bin/sh
#
# description: gsm server
#

# Start the service gsm
start() {
        initlog -c "echo -n Starting gsm server: "
        /usr/bin/gsm --start
        ### Create the lock file ###
        touch /var/lock/subsys/gsm
        success $"gsm server startup"
        echo
}

# Restart the service gsm
stop() {
        initlog -c "echo -n Stopping gsm server: "
        killall gsm
        ### Now, delete the lock file ###
        rm -f /var/lock/subsys/gsm
        echo
}

### main logic ###
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status gsm
        ;;
  restart|reload|condrestart)
        stop
        start
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|reload|status}"
        exit 1
esac

exit 0
```

In this backup FS all the libraries are available so the user program running on this FS could call to any API function as in the main FS, which can be of use in order to connect to the server and to try to recover the main system remotely.

8. U-boot environment

The u-boot environment variables can be saved permanently in flash, and accessed from the system using *fw_printenv* and *fw_setenv* command line tools.

```
root@arm:/# fw_printenv
arch=arm
...
```

8.1. System boot protection

The u-boot has a counter that can be enabled to protect undesired states, where an erroneous FS provokes a continuous reboot cycle of the system.

The following variables can be set to control the desired behaviour:

- bootcount**:**

This variable will be automatically created if it does not exist, and it will be updated at each reset of the processor. It is persisted in U-boot environment so after a power on reset it will keep the incremental value.

- bootlimit**:**

If this variable exists, its contents are taken as the maximum number of reboot cycles allowed.

- altbootcmd**:**

If, after a reboot, the new value of **bootcount** exceeds the value of **bootlimit**, then instead of the standard boot action (executing the contents of **bootcmd**) an alternate boot action will be performed, and the contents of **altbootcmd** will be executed.

The variable upgrade_available can be used to activate or disable this protection. If *upgrade_available* is 0, *bootcount* is always 0, if *upgrade_available* is 1 *bootcount* is incremented in the environment.

9. systemd - system and service manager

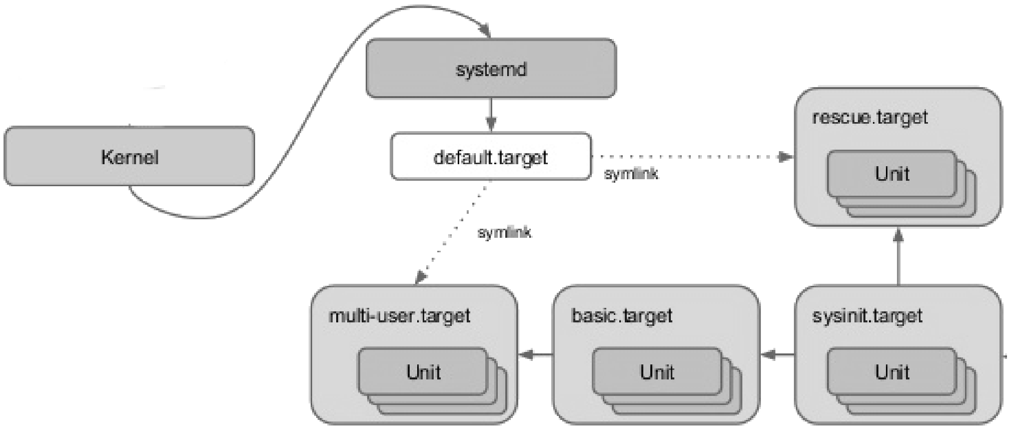
Systemd *(ref **Error! Bookmark not defined.**)* is currently the default init system for Debian and, therefore, for the unit.

systemd also serves as service manager, whose tasks are organized as *units*, and ***systemctl*** is the main tool used to control the state of *systemd*.

```
systemctl list-units              # List all units and their current status
systemctl stop unit_name          # Deactivate one or more units
systemctl start unit_name         # Activate one or more units
systemctl restart unit_name       # Stop and then start one or more units
systemctl reload unit_name        # Asks units to reload their configuration
systemctl status unit_name        # Show status information about units
systemctl show unit_name          # Show configuration properties of units
```

9.1. Boot process

In the unit, *systemd* uses the /etc/systemd/system/default.target configuration file to determine which state, into which it should boot. The *default.target* file is a symbolic link to the true target file. In particular, the default target in the unit is ***multi-user.target*** which is like runlevel 3 in *SystemV*. For example, the *multi-user.target* unit starts other essential system services, such as, *networking, D-Bus* or *systemd-timesyncd.*



For instance, an user can switch to single user mode by typing the following command:

```
systemctl isolate runlevel1.target
```

And to stop a serial console:

```
systemctl stop serial-getty@ttyO4.service
```

9.2. Unit files

systemd provides users an easy interface to configure services. These services are described by a **.service file**, called *unit*, whose syntax uses key-value pairs grouped between [section] headers. Unit files are stored under /lib/systemd/system/ and /etc/systemd/system/.

A *systemd* .service file describes a process managed by *systemd* in a declaratory way (e.g., the process it starts, its dependencies, and many other parameters). The following example configuration file summarizes the main options in unit files:

```
# location: /etc/systemd/system/
[Unit]
Description = Example unit that runs an application at boot
After      = syslog.target

[Service]
# make sure the shell script is executable (chmod +x $1)
ExecStart   = /home/debian/myowa4xd

# Tell the serive how to kill it / reload.
#ExecStop    = /home/debian/myowa4xd --exit

[Install]
# multi-user.target corresponds to run level 3
WantedBy    = multi-user.target
```

9.2.1. Enabling and disabling services

In the unit the user can enable multiple applications to automatically start at boot. This is achieved by the following command:

```
systemctl enable application.service
```

To disable the service from starting automatically, you can type:

```
systemctl disable enable application.service
```

9.2.2. Grouping services

Moreover, a systemd *target* file groups a set of services (units). Thus, the dependencies of a target can be listed either within the target file (in the Requires= line), or using a symbolic link to a service file in the /lib/systemd/system/targetname.target.wants/ directory. For instance, /etc/systemd/system/reboot.target.wants/ can contain links to multiple services that systemd will run before rebooting in order to reach reboot.target.

9.3. System logging

systemd provides a logging system, implemented with the *journald* daemon, for kernel and user-space processes. The *journalctl* utility is used to access (e.g., executed without any arguments, it simply spews all log messages that occurred since system boot) and filter (by time, priority, service, etc.) logs.

For example, to show only entries associated with the service, logged at the error level or above since January 10th, 2018 at 5:15 PM, you can type:

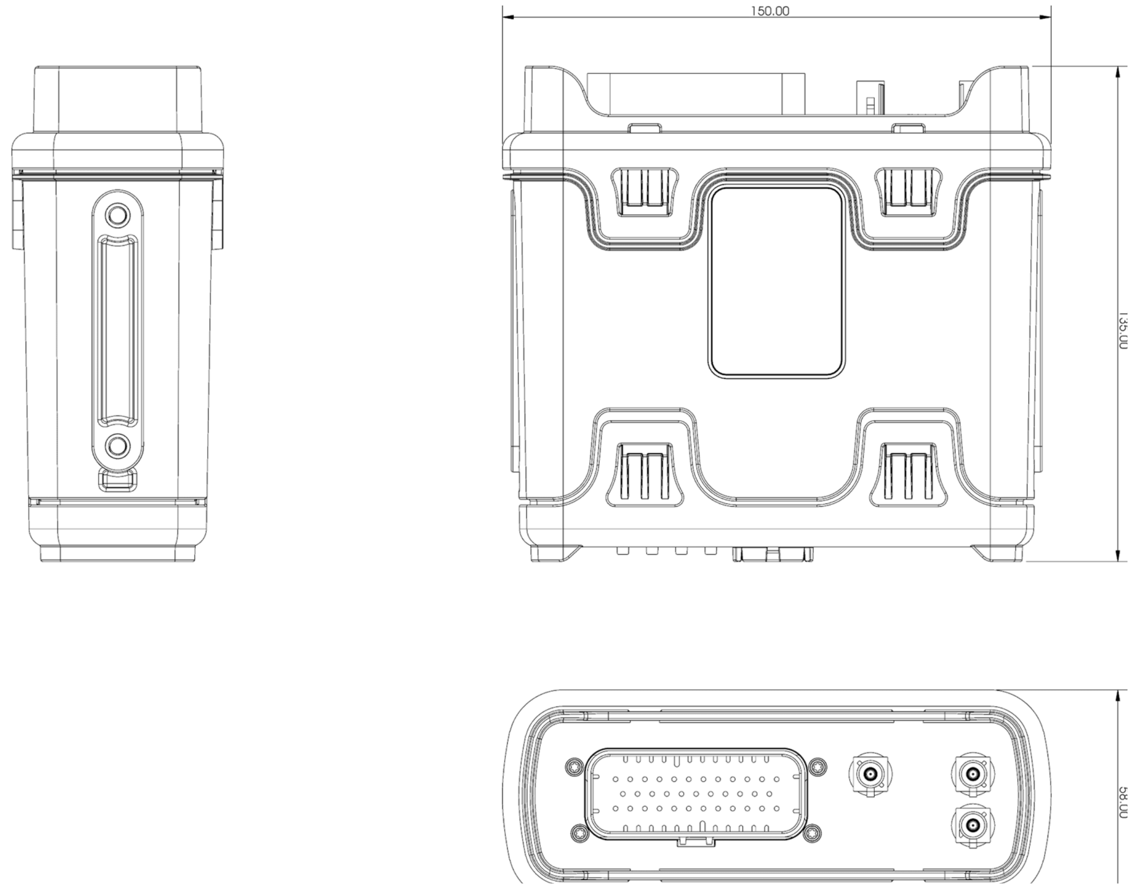
```
journalctl -u myowa4xd.service -p err --since "2018-01-10 17:15:00"
```

9.4. Backup FS

A backup FS is included that can be used to recover the system, either locally or remotely, as this small FS has all the libraries needed to open a data session with the 4G module, using the owasys GSM library.

10. Installation

Dimensions



10.1. Antenna Location and installation

Permanently installed antennas are preferred over magnetic, glass or body lip mounts for anything other than for low power or temporary installations. However, a magnetic mount antenna is a good tool for checking the proposed fixed antenna location for unwanted effects.

Glass mounted antennas are suitable for mounting on vehicles. In this case they should be kept as high as possible in the centre of the rear window or wind-shield. Some vehicles use glass that contains a thin metallic coating for defrosting or to control solar gain; glass mount antennas may NOT function properly when mounted on this type of glass.

If a magnetic mounted antenna is used, take care to locate the magnetic base in a location which avoids interference to the compass mechanism, since magnets may affect the accuracy or operation of the compass. If metallic panels are used, do not block the reception paths for installed antennas such as Global Position Satellite Receivers, if so equipped.

For an optimum performance of antennas, consider these aspects:

- The antennas should have an unobstructed view of the sky, especially for GNSS receivers. The antennas should not be shielded by any metal object or other impenetrable material.

- The antennas have to be safe from damage during normal vehicle operation and maintenance.
- GNSS antenna should not be shielded from satellite signals by metal objects or other impenetrable materials.
- Separate GSM and GNSS antenna at least 50 cm.
- Choose a location with access both above and below the antenna-mounting surface. This access is required for installing fasteners and for routing the antenna cable.
- Keep the antenna as separate as possible from the microphone and loudspeaker when hands-free option is used.

10.1.1. Antenna Tuning

It is important that the antenna is properly tuned and Voltage Standing Wave Ratio (VSWR) less than 2.0:1, and to avoid RF current on the antenna cable shield.

10.1.2. Antenna Cable Routing

Always use a high quality, one-piece coaxial cable. Connector quality and termination techniques are quite important.

The antenna cables should be treated in the same way as the control and power cables. Avoid sharp edges and pinches and keep the cable as short as possible. Also avoid routing the antenna cable in parallel with other control or power wiring over long distances. If it is necessary to cross over wiring, cross at right angles.

10.2. Power connection

To connect the cables follow these instructions:

- Allow enough cabling to enable the removal of the equipment.
- Ensure cables are not corrupted or rubbing against sharp objects.

10.3. Choosing a location

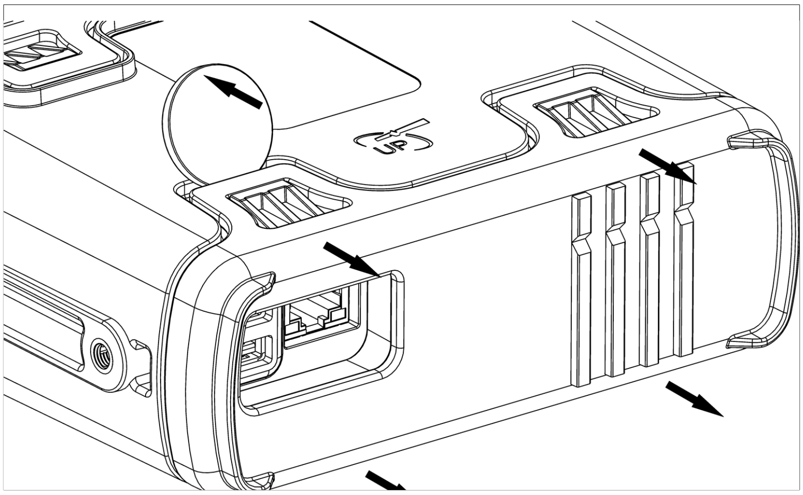
Choose a location for the unit which allows for convenient routing and connection of the antenna and interface cables, and which has access to power source and status LEDs. When selecting a mounting location, avoid the following hazards:

- Direct exposure to weather.
- Excessive heat or cold.
- Excessive humidity.
- High vibration areas.
- Corrosive fluids and gases.
- Direct exposure to water.
- Direct exposure to solar radiation.
- Do not obstruct drivers view or impede operation.

10.4. Opening the unit

In order to insert the SIM card and the microSD card, the rear part must be opened. There are 4 flaps that must be pulled one by one, until all 4 are up and the lid can be taken out.

For the opening process no special tool is needed, just a coin (for example 1€ coin) will do. In this picture the arrows show how to push the coin once it has been inserted under the flap.



The best procedure is opening one flap, then the flap in the other side, then the one near it and finally the remaining one. For example, and starting as shown in the figure, opening first the flap above the Ethernet / USB connectors in the left side, then the second should be the one under the Ethernet / USB connectors. Third and fourth flaps to open will be the ones above and under the LEDs in the right side of the unit.

10.5. Mounting and fixing the unit

Always keep cabling as short as possible.

The unit can be mounted horizontally, vertically, or in any convenient orientation, but it is advisable to maintain the indicator lights in sight, since that come as an advantage when troubleshooting the unit.

The 2 metal brackets are fixed in the sides of the unit, and the screws to mount it are not provided, as the installation scenario can vary significantly.

In the following image the distance between the holes is shown:

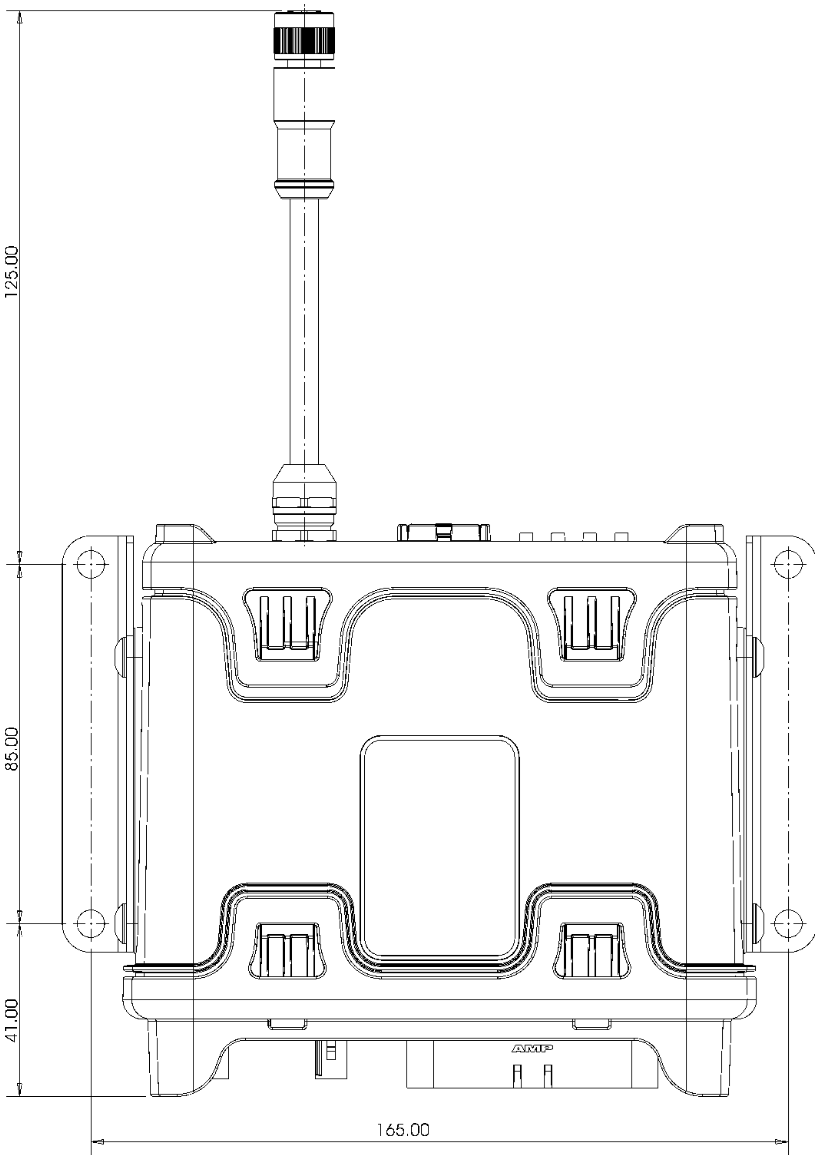


Illustration 29: Installation brackets measurements

For the option with IP67 lid and Ethernet in the M12 connector, the cable sticks out of the mechanics 125 mm, that is also to be considered when installing the unit.

For the Wi-Fi/BT option, take care not to blind the internal antenna, placing the brackets in the upper side of the unit, so that the antenna has an open view. See the “satellites up” signal in one the sides to guide on the installation of the fixing units, as this signal must have the open view.

11. Technical data summary

General Specification

Item	Specification
Processing Capacity	32 bits CORTEX A8 core at 800MHz
Available Memory for user	About 800 MB Flash About 450 MB RAM

Mechanical Description

Item	Specification
Dimensions (mm.)	150(W)x58(H)x135(L)
Weight	400 gr
Material	Glass reinforced plastic
Connectors	Machine Connector USB RJ45 SIM Card Holder Optional battery connector µSD card holder

Power Interface

Item	Specification	
Power supply	9 to 48 Vdc	
Power Consumption	**Mode**	**Consumption @ 24V (mA)**
(Typical average)	RUN	47 *
	RUN + ETH connected	59
	RUN + GNSS + ETH	67
	RUN + GSM + ETH	65
	RUN + GSM + GNSS	63
	RUN + GSM on call + ETH	92
	STANDBY	9.88
	OFF	0.335

* Consumption improvement can be achieved by switching down Ethernet in case it is not needed, 6mA@24V, in order to use Ethernet again, a system reboot is needed.

11.1. GSM/GPRS Specifications

HW < R4

Item	Specification
Frequency bands	Quad Band GSM 850/ EGSM900 / GSM1800/ GSM1900.
Power	2 Watt at GSM850/EGSM900 and 1 Watt at GSM1800/GSM 1900
GPRS	Class B Class 10 (4+2)
Operations	Audio calls, Data calls, Short Message Service
SIM	Integrated holder, 3 V
Antenna Connector	Fakra plug Male type D

HW = R4

Item	Specification
Frequency bands	LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28 LTE-TDD: B38/B39/B40/B41 UMTS: B1/B2/B4/B5/B6/B8/B19 GSM: 850/900/1800/1900MHz
Power (max ratings)	2.5 Watt at GSM850/EGSM900 1.8 Watt at GSM1800/GSM 1900 2.1 Watt at WCDMA 2.8 Watt at LTE
GPRS	multi-slot class 33
UMTS	DC-HSDPA: Max 42Mbps (DL) HSUPA: Max 5.76Mbps (UL) WCDMA: Max 384Kbps (DL), Max 384Kbps (UL)
LTE	LTE-FDD: Max 150Mbps (DL), Max 50Mbps (UL) LTE-TDD: Max 130Mbps (DL), Max 30Mbps (UL)
Operations	Audio calls, Short Message Service, data session
SIM	Integrated micro-SIM holder, 3 V. Do not user pre-cut mini/micro/nano-SIM cards, use true microSIM cards.
Antenna Connector	2 x Fakra plug Male type D

11.2. GNSS Specifications

Item	Specification
Receiver	72-channel GPS L1C/A SBAS L1C/A QZSS L1C/A GLONASS L1C BeiDou B1 GALILEO E1B/C SBAS: WAAS, EGNOS, MSAS, GAGAN
Update Rate	10Hz or 5Hz if received more than one satellite system
Accuracy	2 meters CEP
Signal Acquisition	Cold Start: 26 sec Hot Start: 1.5 s Signal Reacquisition: < 1 sec
Datum	WGS-84
Sensitivity	Tracking & reacquisition: -167 dBm Cold start: -148 dBm
Operational limits	Speed: 500 m/s (972 knots) Altitude: 50,000 m
Antenna Connector	FAKRA plug Male Type C
Antenna Requirements	Active antenna.[2] Impedance: 50Ω Active Antenna recommended gain:15 dB Frequency: 1575.42 MHz VSWR: Max 2.0:1
Active antenna detection circuitry	Active Antenna Power Supply: +3.0V @ 30mA current

[2] The use of a passive antenna would require to connect a DC block between the OWA connector and GNSS antenna.

Item	Specification
	• open circuit detection: < 0.5 mA • short-circuit detection: > 100 mA

11.3. Environmental Specifications

Item	Specification
Operating Temperature GSM off	-40ºC to +85ºC
Operating Temperature GSM on	-30ºC to +80ºC
Safety Purposes Temperature Range without Li-ion Battery	-40ºC to 65ºC
Safety Purposes Temperature Range with Li-ion Battery	-40ºC to +55ºC from external power supply -20ºC to +55ºC from Li-ion battery (-40ºC to -20ºC internally limited with battery protection) 0ºC to +45ºC Li-ion battery charge (internally controlled)
Storage Temperature	-40ºC to +85 ºC

12. Technical support

Additional sources of information are available on the CrossControl support site:
https://crosscontrol.com/support/
You will need to register to the site in order to be able to access all available information

Contact your reseller or supplier for help with possible problems with your device. In order to get the best help, you should have access to your device and be prepared with the following information before you contact support.

- The part number and serial number of the device, which you can find on the brand label.
- Date of purchase, which can be found on the invoice.
- The conditions and circumstances under which the problem arises.
- Status indicator patterns (i.e. LED blink pattern).
- Prepare a system report on the device, using CCSettingsConsole (if possible).
- Detailed description of all external equipment connected to the unit (when relevant to the problem).

13. Trademarks and terms of use

All trademarks sighted in this document are the property of their respective owners.

CCpilot is a trademark which is the property of CrossControl.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

CC Linux is an officical Linux distributuition pursuant to the terms of the Linux Sublicense Agreement

Microsoft® and Windows® are registered trademarks which belong to Microsoft Corporation in the USA and/or other countries.

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Qt is a registered trademark of The Qt Company Ltd. and its subsidiaries.

CrossControl is not responsible for editing errors, technical errors or for material which has been omitted in this document. CrossControl is not responsible for unintentional damage or for damage which occurs as a result of supplying, handling or using of this material including the devices and software referred to herein. The information in this handbook is supplied without any guarantees and can change without prior notification.

For CrossControl licensed software, CrossControl grants you a license, to under CrossControl's intellectual property rights, to use, reproduce, distribute, market and sell the software, only as a part of or integrated within, the devices for which this documentation concerns. Any other usage, such as, but not limited to, reproduction, distribution, marketing, sales and reverse engineering of this documentation, licensed software source code or any other affiliated material may not be performed without the written consent of CrossControl.

CrossControl respects the intellectual property of others, and we ask our users to do the same. Where software based on CrossControl software or products is distributed, the software may only be distributed in accordance with the terms and conditions provided by the reproduced licensors.

For end-user license agreements (EULAs), copyright notices, conditions, and disclaimers, regarding certain third-party components used in the device, refer to the copyright notices documentation.