

CrossLink AI Quick Start Guide

Revision 1.6



Contents

| | |
|---|-----------|
| 1. CrossLink AI | 3 |
| 1.1. Intended audience | 3 |
| 1.2. Introduction..... | 3 |
| 1.3. Supported devices status | 3 |
| 1.4. Physical Connectors | 4 |
| 1.5. Pin Connections | 4 |
| 2. Configuration and usage | 5 |
| 2.1. WLAN and 3G power up..... | 5 |
| 2.2. CrossLink AI module initialization | 6 |
| 2.3. WLAN station / client mode | 7 |
| 2.4. WLAN access point..... | 8 |
| 2.4.1. Sharing 3G connection | 9 |
| 2.5. 3G | 9 |
| 2.6. GPS | 11 |
| 3. Revision History | 12 |

1. CrossLink AI

1.1. Intended audience

This document is intended to help the reader to get the CrossLink AI unit up and running with a CrossControl display computer. Using the CrossLink AI unit with any other hosts is not supported. The document expects the reader to be familiar with Linux operation system basics such as using the shell to issue commands and edit files, connecting to remote devices using ssh and using Linux networking features from the shell.

1.2. Introduction

CrossLink AI is a wireless communications add-on module for CrossControl display computers. The add-on module provides WLAN station, WLAN access point, 3G functionality.

This document aims to give a brief introduction to the CrossLink AI telematics module software usage. All commands used to control the device are executed through a ssh connection in the display computer that has the CrossLink AI module connected to. Commands are visualized like this

```
# cp <file name> <new file name>
#
```

Parameters to the cp command in the above example are placeholder parameters that are to be replaced by the user. The hash sign (#) is not typed by the user. It is used to illustrate the root user command prompt. Lines that do not have the hash sign are output from the given command's execution. Plain output and configuration file snippets also use the same formatting but do not have the hash sign.

1.3. Supported devices status

CrossLink AI support with different CrossControl displays are presented below. Please note that a CrossLink AI enabled operating system is required.

| Display | WLAN | WLAN AP | 3G/GPS |
|---------------|------|---------|--------|
| CCpilot VC | yes | yes | yes |
| CCpilot VA | yes | yes | yes |
| CCpilot VS | yes | yes | yes |
| CCpilot VI | yes | yes | yes |
| CCpilot XM2 | yes | yes | yes |
| CCpilot XA/XS | yes | yes | yes |

1.4. Physical Connectors

| Connector | Description | Connector type | Mating cable |
|-----------|------------------------|---------------------------|------------------------|
| X1 | Power, 12-24 V | M12 5-pole B-coded male | C000082-94 |
| X2 | USB | M12 5-pole A-coded female | C0000138-56/C000124-85 |
| X3 | WLAN | RP SMA female | C000142-006 |
| X4 | 3G Main | RP SMA female | C000142-008 |
| X5 | GNSS (GPS and GLONASS) | RP SMA female | C000142-007 |
| X6 | 3G Diversity | RP SMA female | C000142-008 (optional) |

1.5. Pin Connections

| | X1 | X2 |
|-------|---|---|
| | <p>X1 Power Input M12 male 5-pole B-coded</p> | <p>X2</p> <p>SAACC-CHM12FS-5CON-L180-THR SH</p> |
| Pin 1 | Power IN | USB_VBUS |
| Pin 2 | Power IN | USB_D- |
| Pin 3 | GND | USB_D+ |
| Pin 4 | GND | NC |
| Pin 5 | NC | GND |
| Pin 6 | NC | GND (shield) |

2. Configuration and usage

The default configurations for WLAN and 3G most likely are not what one wants so this section describes where and how one can make changes to the settings.

2.1. WLAN and 3G power up

Currently in its default state the CrossLink AI's 3G and WLAN modules may not be powered on boot as default. To power on the modules and have them power on during startup in the future one needs to enable power on startup using the device's serial interface.

```
# microcom /dev/ttyACM0
#
```

Typing "help" in the serial console displays all available commands.

After opening the serial console you might see accelerometer data. Hit enter to break the data flow and issue the following commands to power on wireless LAN

```
# wlan power:1
# wlan startup power:1
```

To power on the 3G module issue the following commands

```
# 3g power:1
# 3g startup power:1
```

Press ctrl+x to exit microcom.

Wifi and 3G module power can also be controlled with `telematics_module_power.sh` script which is used by wlan and 3G initialization scripts in case the device is not seen when they are executed.

To turn on and off the wifi module manually using the script do

```
# telematics_module_power.sh wlan start
# telematics_module_power.sh wlan stop
```

To turn on and off the 3G module manually using the script do

```
# telematics_module 3g start
# telematics module 3g stop
```

To verify the modules powered on list the connected USB device

```
# lsusb
#
```

Following device ID's should be now present in the system

ID 1618:9113

ID 05c6:9003 Qualcomm, Inc.

Device with ID 1618:9113 is the WLAN module and the device with ID 05c6:9003 is the 3G radio module.

2.2. CrossLink AI module initialization

Files related to the module initialization configuration are listed below

| File | Description |
|-------------------------------|----------------------------------|
| /etc/init.d/init-wireless | WLAN init script |
| /etc/init.d/init-3g | 3G and GPS init script |
| /etc/telematics/wireless.conf | Wireless operating mode settings |

The operating mode of wireless LAN is set in wireless.conf. Value of the variable COEX_MODE sets the mode they operate in. Supported driver modes are shown below. By default the mode is set to 3.

- 2 WLAN access point (including multiple APs on different vaps)
- 3 WLAN access point + station mode (on multiple vaps)
- 5 WLAN station
- 6 WLAN access point

One can edit the file for example with the nano editor

```
# nano /etc/telematics/wireless.conf
#
```

After editing the file use ctrl+x to exit the editor. Press Y to confirm the changes.

To init WLAN (if enabled) modules do this

```
# /etc/init.d/init-wireless start
#
```

To enable the wireless module on boot issue the following command to create the symbolic link shown below. Please note that if the add-on module is not connected the init script will fail and the initialization will not be done.

```
# ln -s /etc/init.d/init-wireless /usr/local/etc/rc3.d/S96init-wireless
#
```

To Initialize the 3G modem's kernel modules with the init script

```
# /etc/init.d/init-3g start
#
```

To make start init the modem during boot create the symbolic link shown below.

```
# ln -s /etc/init.d/init-3g /usr/local/etc/rc3.d/S97init-3g
#
```

2.3. WLAN station / client mode

WLAN station configuration file(s) are listed below

| File | Description |
|---|-------------------------------|
| /etc/telematics/wpa_supplicant_station.conf | Station mode configuration |
| /etc/init.d/wlan-station | WLAN station mode init script |

To be able to connect to a wireless network one first needs to configure the access point details in the station mode configuration file. By default the template configuration does not have any network configuration enabled.

```
# nano /etc/telematics/sta_settings.conf
#
```

The file has example configurations for different network types. Uncomment the network type you want to use by removing “#” characters from the beginning of the lines so that the wanted configuration looks for example like this

```
network={
    ssid="NetworkName"
    pairwise=CCMP TKIP
    group=CCMP TKIP
    key_mgmt=WPA-PSK
    psk="network-passphrase"
    proto=WPA2 WPA
    priority=1
}
```

When the configuration is ready wifi is started in station mode with the following command

```
# /etc/init.d/wlan-station start
#
```

To enable the WLAN station on boot create the symbolic link for sysvinit shown below.

```
# ln -s /etc/init.d/wlan-station /usr/local/etc/rc3.d/S97wlan-station
#
```

After starting the station one still needs to acquire an IP address for the network interface. Do this by calling dhclient on the created interface.

```
# udhcpc wlan0
#
```

*On some devices - dhclient wlan0 - may be used instead.

2.4. WLAN access point

WLAN access point configuration files are listed below

| File | Description |
|--|---|
| /etc/telematics/wpa_supplicant_ap.conf | Access point mode supplicant configuration. |
| /etc/telematics/hostapd_open.conf | Access point open mode configuration |
| /etc/telematics/hostapd_open_5ghz.conf | Access point 5GHz open mode configuration |
| /etc/telematics/hostapd_tkip.conf | WPA encrypted access point configuration |
| /etc/init.d/wlan-access-point | Access point initialization script |
| /etc/telematics/dnsmasq.conf | Access point DHCP settings |
| /etc/telematics/wireless.conf | Wlan initialization configuration |

The Access point mode wpa_supplicant configuration file is only needed if one wants to run access point without hostapd. Using the access point like this is not covered in this document.

To use wlan in access point mode, set the appropriate COEX_MODE in wireless.conf first.

To configure the access point edit one of the hostapd configuration files shown in the table above. By default hostapd_tkip.conf is used.

```
# nano /etc/telematics/hostapd_tkip.conf
#
```

Change the network name and passphrase and exit saving your change by pressing ctrl+x and confirming the changes by pressing "Y".

If one wants to use another configuration file editing the the access point initialization script is needed

```
# nano /etc/init.d/wlan-access-point
#
```

Comment out the currently active configuration and uncomment or add the one to be used. Start the access point by issuing the following command.

```
# /etc/init.d/wlan-access-point start
#
```

To change the dhcp setting of the access point one needs to edit the access point DHCP setting configuration file

```
# nano /etc/telematics/dnsmasq.conf
#
```

Note that by default resolv.conf is not used in name resolution for the client connecting to the the access point.

To enable the WLAN access point on boot issue the following command

```
# ln -s /etc/init.d/wlan-access-point /usr/local/etc/rc3.d/S96wlan-access-point
#
```

2.4.1. Sharing 3G connection

3G internet connection can be shared for the clients connected to the wifi access point. First enable 3G networking based on instructions in chapter 2.6. After 3G connection is enabled the connection to internet is shared to clients connected to the wifi access point using iptables routing rules as follows.

```
iptables --flush
iptables --table nat --flush
iptables --delete-chain
iptables --table nat --append POSTROUTING --out-interface $OUT -j MASQUERADE
iptables --append FORWARD --in-interface $IN -j ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward

ip route change default via 10.64.64.64
```

2.5.3G

3G requires operator specific point-to-point connectivity scripts to be used. The ppp configuration scripts included in the default image are using Sonera configuration. If your service provider is something else you will need to change the scripts to match your operator's settings. Sonera scripts are provided as default for reference. The files are show in the table below.

| File | Description |
|--------------------------|---------------------------|
| /etc/ppp/peers/sonera | pppd configuration |
| /etc/ppp/connect-chat | Connection init settings |
| /etc/ppp/disconnect-chat | Connection close settings |

Before creating the connection one needs to unlock the SIM card (if locked). PIN number for the SIM card can be given with the help of a script provided in the image like this

```
# unlocksim.sh <pin code>
#
```

Please note that error will be returned if SIM card is already unlocked not just when the PIN code is not correct.

In case one needs to debug the 3G connection startup edit the pppd configuration file and uncomment the "nodetach" instruction in it.

To start the connection use the following command

```
# pppd call sonera
#
```

Details for the created ppp connection can be seen with ifconfig command. The created interface is named ppp0.

2.6.1 Serial interface

The 3G modem can be controlled with a serial interface. To connect to the interface issue the following command

```
# microcom /dev/ttyUSB2
#
```

If you see the following output you need to insert the SIM cards PIN code

```
+CPIN: SIM PIN
AT+CPIN
```

To insert the PIN number for the SIM card issue the following command

```
# AT+CPIN=<pin code>
#
```

If the PIN number was correct you will see the following output.

```
OK
+CPIN: READY
```

To see the currently selected operator (if any) issue the following command.

```
# AT+COPS?
#
```

If the command returns 0, it means that the 3G radio is not currently connected to any operator and the radio is on automatic mode.

To check the network registration status issue the following command.

```
# AT+CREG?
#
```

In case the radio is registered to its home network the answer will be 0,1 and if in case it is not registered but is searching for an operator the answer will be 0,2.

2.6. GPS

GPS device co-exists with the 3G modem so it is initialized with it. So to enable GPS one needs to execute the following init scripts. There is no need to run it again if it was already done for the 3G. GPS does not need additional configuration.

```
# /etc/init.d/init-3g start
#
```

A convenience script is provided to start the GPS engine.

```
# enablegps.sh
#
```

Alternatively one can enable GPS through the serial interface as follows.

Open the serial device.

```
# microcom /dev/ttyUSB2
#
```

Enter the following commands to the serial console

```
# AT+QGPSCFG="gpsnmeatype"  
# AT+QGPS=1
```

Close the serial console by pressing ctrl+x.

The GPS data is visible from another serial interface, /dev/ttyUSB1. To follow it use the following command.

```
# microcom /dev/ttyUSB1  
#
```

3. Revision History

| Date | Description |
|-----------|---|
| 25.1.2016 | First draft |
| 26.2.2016 | Removed unsupported driver modes, updated instructions, command highlighting and other small changes |
| 30.3.2017 | Changes configuration paths, added remount commands for places where writing to rootfs is needed and other needed changes |
| 12.4.2017 | 3G connection sharing via wifi, intended audience, known issues, power control, symbolic link creation |
| 24.8.2017 | Changed part number for the antennas in section 1.4. |
| 15.5.2019 | Added Pin Diagram, removed reference to unsupported feature. |