QUICK START GUIDE

AMOS-820

Android BSP v4.0.2

1.00-08112016-145000

Copyright

Trademarks

Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information (including use or connection of extra device/equipment/add-on card) in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right the make changes to the products described in this manual at any time without prior notice.

Revision History

Version	Date	Remarks
1.00	8/11/2016	Initial release

Table of Contents

1. Introduction

The purpose of this document is to provide an overview of getting started with the AMOS-820 system (Bare board: VAB-820 with NXP i.MX 6Quad Cortex-A9 processor) using Android 5.0.

The AMOS-820 Android BSP v4.0.2 is developed based on the NXP android_l5.0.0_1.0.0-ga (Android 5.0 Lollipop) and it enables hardware features that are defined on the AMOS-820 system.

1.1. Package Contents

There are three folders in the package as listed below.

BSP folder	Description
VAB-820_Android_ source_code_patch.zip	Android source code patch files and Image building script files
Smart_ETK_Demo_Program _Source_Code.zip	Smart ETK demo program source code
Document folder	**Description**
AMOS-820_Android_BSP_v4.0.2_Quick_Start_Guide _v1.00_20160811.pdf	Quick Start Guide
AMOS-820_Android_EVK_v4.0.2_Image_Installation_Guide_v1.00_20160811.pdf	Image Installation Guide
EVK folder	**Description**
Images_autoinstall_sd.zip	Android EVK system image and installation script files
OTA.zip	OTA upgrade packages
VAB-820_Smart_ETK_Demo_v0.0.16.apk	Smart ETK demo program

AMOS-820 Android BSP contents

1.1.1. BSP Folder Contents

VAB-820_Android_source_code_patch.zip: includes NXP Android patch file **android_L5.0.0_1.0.0-ga_core_source.tar.gz**, the VAB-820 Android source code patch file and Image building script files.

Smart_ETK_Demo_Program_Source_Code.zip: Smart ETK demo program source code.

1.1.2. Document Folder Contents

AMOS-820_Android_BSP_v4.0.2_Quick_Start_Guide _v1.00_20160722.pdf: The Quick Start Guide provides an overview of getting started with the AMOS-820 system using Android.

AMOS-820_Android_EVK_v4.0.2_Image_Installation_Guide_v1.00_20160722.pdf: The Image Installation Guide explains how to boot the Android EVK image on the AMOS-820 system in order to begin evaluating the platform.

1.1.3. EVK Folder Contents

Images_autoinstall_sd.zip: The Android EVK system image and installation script files.

OTA.zip: OTA testing upgrade packages.

VAB-820_Smart_ETK_Demo_v0.0.16.apk: The Smart ETK demo program.

1.2. Version Information and Supported Features

- U-Boot version: 2014.04
- Kernel version: 3.10.53
- Evaluation image: Android Lollipop 5.0
- Development based on NXP android_l5.0.0_1.0.0-ga (Android 5.0 Lollipop)
- Supports SPI with eMMC boot
- Supports HDMI display
- Supports HDMI audio output
- Supports CVBS
- Supports COM 1 DTE mode, and COM 2 as a debug port
- Supports two FlexCAN TX/RX
- Supports Gigabit Ethernet
- Supports Line-in, Line-out, and Mic-in
- Supports VNT9271 USB Wi-Fi dongle
- Supports EMIO-1541 miniPCIe Wi-Fi module
- Supports EMIO-2550 miniPCIe Mobile Broadband module
- Supports Smart ETK v0.0.16: Watchdog, GPIO, UART, and FlexCAN
- Support OTA (Over-the-Air technology)
- Support shutdown option in Quick Settings
- Support Ethernet configuration in Settings

2. Image Development

This section explains the setup requirements for booting from the SPI ROM with an eMMC. A precompiled image is provided in the "EVK" folder of the BSP.

2.1. Booting from the SPI ROM with an eMMC

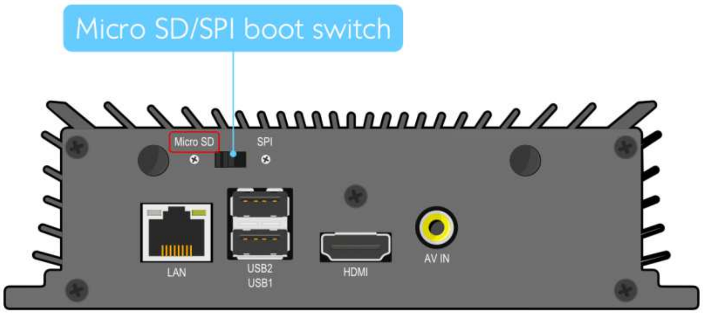
The **Images_autoinstall_sd.zip** includes the Android EVK system image and the installation scripts files.

The first step is to extract **Images_autoinstall_sd.zip** file to make the bootable Micro SD card. Insert a card into your Linux host machine and make sure it is not mounted. Open the terminal on your host machine, and run the script **mk_install_sd.sh** as shown below, replacing <device name> with the correct value for the card, for example "sdb".

Important: Make sure you are writing to the correct device or the host system environment could be damaged.

```
$ sudo ./mk_install_sd.sh /dev/<device name>
```

Next, on the AMOS-820, set the Micro SD/SPI boot switch to the Micro SD position as shown below.



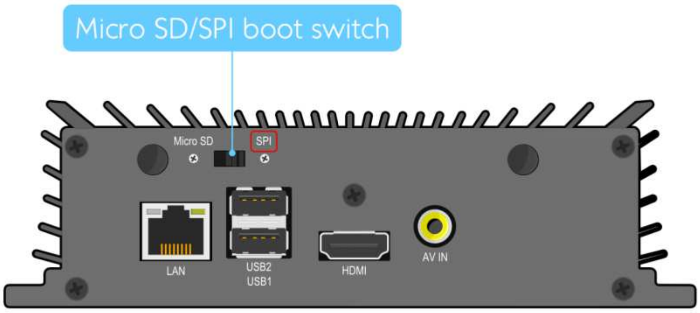
Micro SD/SPI boot switch diagram

Insert the prepared Micro SD card into the AMOS-820, connect an HDMI display, and power on the AMOS-820 to initiate the boot process. When the boot process is completed, you will see the command line of Linux environment.

To install the U-Boot and Android EVK image into SPI ROM and eMMC, use the following command:

```
$ ./fsl-eMMC-partition.sh –f imx6q /dev/mmcblk0
```

In order to boot from the SPI ROM make sure the Micro SD/SPI boot switch is set to SPI ROM boot.

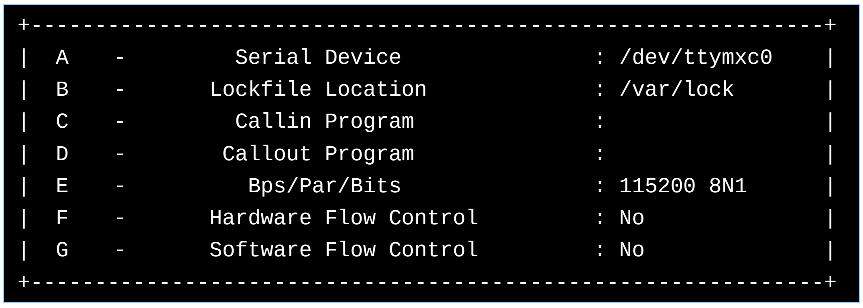


Micro SD/SPI boot switch diagram

Next, power on the device to initiate the boot process. When the boot process is completed, you will see the Android desktop.

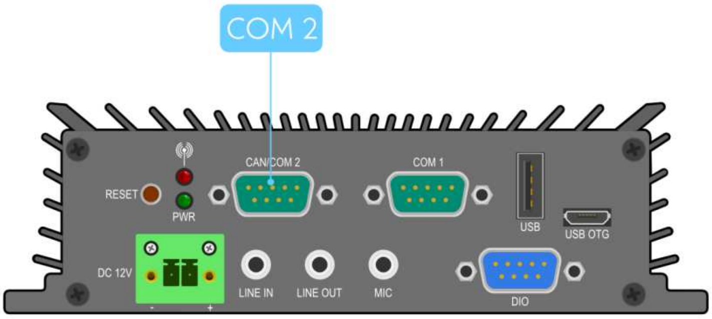
2.2. Setting Up U-Boot Parameters

The first step is to connect the host machine and the AMOS-820 through the COM 2 debug port. Use a serial port communication program such as PuTTY, GtkTerm, or Minicom, to configure the serial port setting and connect to the debug console. There you will be able to see the U-Boot boot log and adjust settings in the U-Boot console.



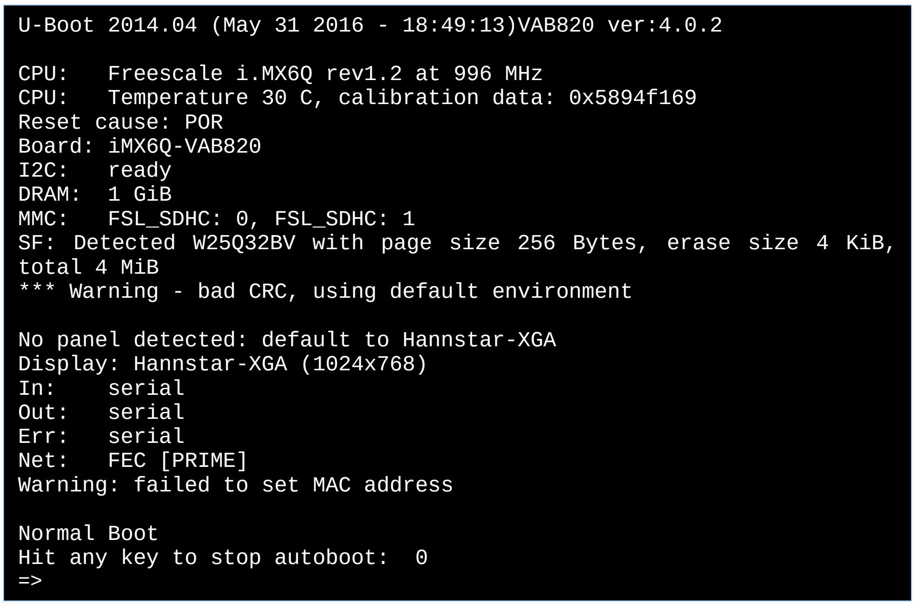
```
+-------------------------------------------------------------+
| A    -         Serial Device              : /dev/ttymxc0    |
| B    -       Lockfile Location            : /var/lock       |
| C    -         Callin Program             :                 |
| D    -        Callout Program             :                 |
| E    -          Bps/Par/Bits              : 115200 8N1      |
| F    -       Hardware Flow Control        : No              |
| G    -       Software Flow Control        : No              |
+-------------------------------------------------------------+
```

Serial port setting of host machine



COM 2 debug port diagram

Next, power on the AMOS-820 to initiate the boot process. When prompted, press any key to stop the boot process, and enter the U-Boot console as illustrated by the screenshot below.



```
U-Boot 2014.04 (May 31 2016 - 18:49:13)VAB820 ver:4.0.2

CPU:   Freescale i.MX6Q rev1.2 at 996 MHz
CPU:   Temperature 30 C, calibration data: 0x5894f169
Reset cause: POR
Board: iMX6Q-VAB820
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
SF: Detected W25Q32BV with page size 256 Bytes, erase size 4 KiB,
total 4 MiB
*** Warning - bad CRC, using default environment

No panel detected: default to Hannstar-XGA
Display: Hannstar-XGA (1024x768)
In:    serial
Out:   serial
Err:   serial
Net:   FEC [PRIME]
Warning: failed to set MAC address

Normal Boot
Hit any key to stop autoboot:  0
=>
```

Debug console view of boot process

To list the current U-Boot parameters, use the following command:

```
=> printenv
```

Make sure the printout message is as follows:

```
=> printenv

baudrate=115200
boot_emmc_root=mmc dev ${root_media_uboot};ext2load mmc ${root_media_uboot}:${root_partition}
${loadaddr} uImage;ext2load  mmc ${root_media_uboot}:${root_partition} ${dtbaddr} imx6q-
vab820.dtb;bootm ${loadaddr} - ${dtbaddr};
boot_media=booti mmc1
```

2.3. Restoring Default U-Boot Parameters

If the U-Boot parameters have been modified, the "destroyenv" command in the U-Boot console can restore the factory default settings.

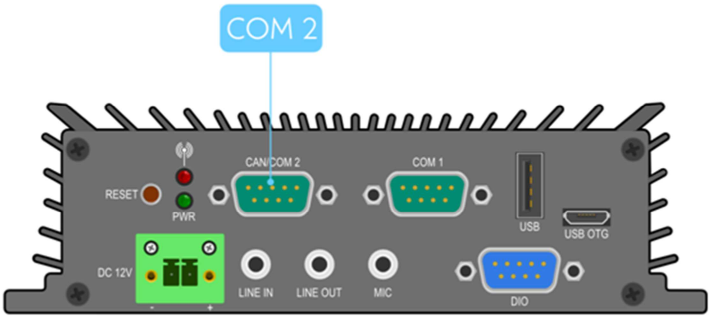
To restore the default U-boot settings, use the following command:

```
=> destroyenv
=> saveenv
=> reset
```

2.4. Using the Android Console

The first step is to connect the host machine and the AMOS-820 through the COM 2 debug port. Use a serial port communication program such as PuTTY, GtkTerm, or Minicom, to connect to the debug console. There you will be able to see the U-Boot boot log and adjust settings in the U-Boot console.

Next, power on the AMOS-820 to initiate the boot process. When the boot process is completed, you will be prompted to login.



COM 2 debug port diagram

When the boot process is completed, you will automatically log in to an Android console.

```
root@vab820_6dq:/ #
```

3. Build Environment Setup

This section guides you through setting up the build environment for development. All instructions are based on using Ubuntu 14.04 LTS.

To make sure that the build process completes successfully, we recommend at least 120GB of disk space and 15GB of combined memory and swap space on the host machine.

3.1. Configuring Ubuntu

The following packages are required for the Android development environment. The required packages can be installed using the commands below. To get more information, please visit Google Android website at http://source.android.com/source/initializing.html.

OpenJDK 7 is not included in the Ubuntu 14.04 default installation. The first step is to add a new server manually by using the following command:

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
```

To install the OpenJDK 7, use the following command:

```
$ sudo apt-get install openjdk-7-jre
$ sudo apt-get install openjdk-7-jdk
```

The following packages are required for the Android development environment. To install the required packages on the Ubuntu 14.04, use the following command:

```
$ sudo apt-get install git gnupg flex bison gperf build-essential zip curl zlib1g-dev libc6-
dev libncurses5-dev x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-
glx:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc
zlib1g-dev:i386 uuid-dev liblzo2-dev libz-dev libc6-dev-i386 lib32z1
```

4. Image Build

4.1. Extract the AMOS-820 BPS

Use the following command to extract the contents of the BSP.

```
$ unzip VAB-820_Android_source_code_patch.zip
```

4.2. Downloading the Source Code of Android 5.0.0

Downloading the sourced code of Android 5.0.0 to BSP/src folder will take several hours.

Use the following commands to download the source code of Android 5.0.0:

```
$ cd VAB-820_Android_source_code_patch/script
$ ./download_source.sh
```

4.3. Installing the Source Code Patch

Use the following commands to install the NXP and AMOS-820 patch source code:

```
$ ./patch_fsl.sh
$ ./patch_via.sh
```

4.4. Building the Android Image

Use the following commands to build the Android image, the compiling process will take several hours

```
$ cd ../src
$ build_bsp.sh
```

After the compilation, the **src/out/target/product/ vab820_6dq/** directory will contain the resulting binaries, as shown in the table below.

Binary	Description
u-boot-imx6q.imx	U-Boot boot loader for iMX6Q
boot-image6q.img	Kernel for iMX6Q
system.img	Android system image
recovery-imx6q.img	Recovery image for iMX6Q

Binary files generated

The BSP supports the following Wi-Fi modules

- EMIO-1533 USB Wi-Fi module
- EMIO-1541 miniPCIe Wi-Fi module
- EMIO-5531 USB Bluetooth and Wi-Fi combo module

Due to the EMIO-5531 Wi-Fi driver conflicts with the other two modules, specify the compiling parameter in order to build the EMIO-1533 and EMIO-1541 Wi-Fi driver into the Android system image.

```
$ ./build_bsp.sh --wifi-vendor=atheros
```

If you forgot to specify the parameter when building the Android image for the first time, you need to append the wpa-rebuild parameter as below.

```
$ ./build_bsp.sh --wifi-vendor=atheros --wpa-rebuild=true
```

5. Hardware Functions

5.1. Setting Up Display Device

The AMOS-820 Android BSP v4.0.2 supports the HDMI monitor.

5.1.1. Setting Up the HDMI U-Boot Display Parameter

The default display device is HDMI monitor.

To set the HDMI monitor as the display output, use the following command:

```
=> setenv display ‘run hdmi’
=> saveenv
```

The HDMI only supports CEA modes as shown in the table below.

Width	Height	Frame rate
640	480	60
720	480	60
1280	720	60
1440	240	60
1440	480	60
1920	1080	60
720	576	50
1280	720	50
1440	288	50
1440	576	50
1920	1080	50
1920	1080	24
1920	1080	25
1920	1080	30
1280	720	100
1280	720	120

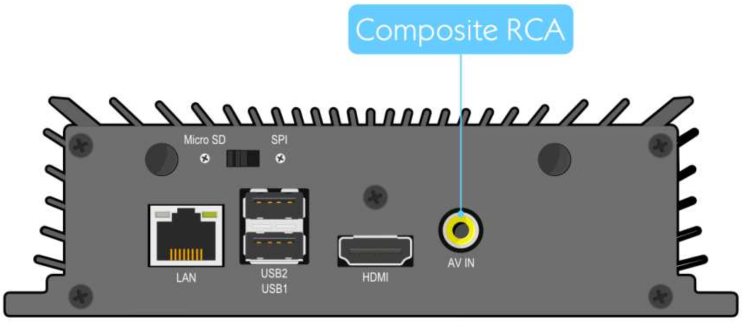
HDMI CEA mode support table

For example, if you want to change the HDMI resolution to 1280x720@60Hz, use the following command:

```
=> setenv hdmi_timing ‘1280x720M@60,bpp=32’
=> saveenv
```

5.2. Setting Up the Video-in

The AMOS-820 supports the video inputs. The composite RCA jack is a CVBS signal input and the S-VIDEO 1 pin header is an S-video signal input.



Composite RCA jack diagram

The CVBS is the default video input.

To set CVBS as video input source, use the following command:

```
root@vab820_6dq:/ # setprop adv7180_src 1
```

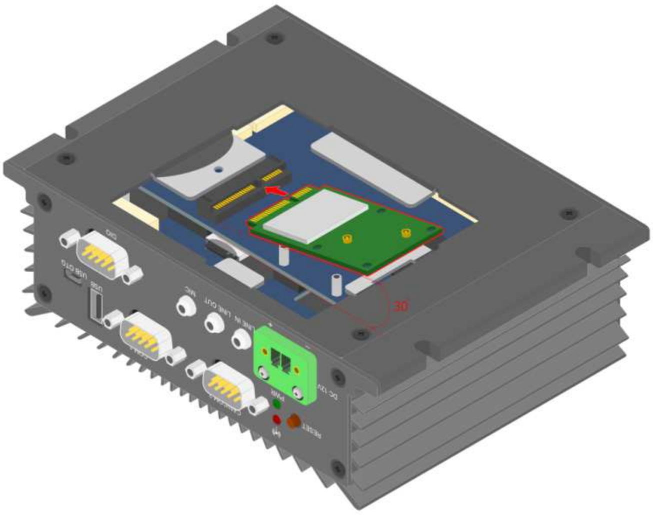
Please note that if the setting step is completed, you need to re-open the Camera program to apply this setting.

5.3. Configuring the EMIO-2550 MiniPCIe Mobile Broadband Module

The EMIO-2550 miniPCIe Mobile Broadband module supports 3G and GPS functions.

5.3.1. Connecting to the Internet

The first step is to insert an active SIM card into the EMIO-2550 module, then install the EMIO-2550 module into the AMOS-820. Next, power on the AMOS-820.



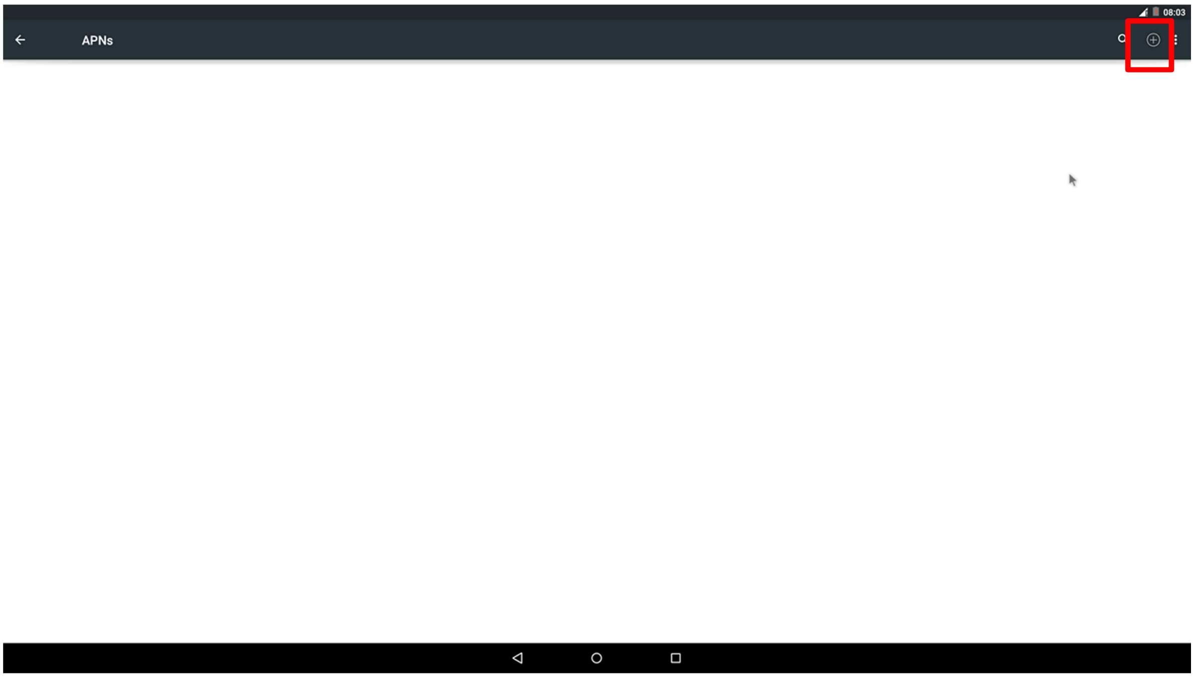
Inserting the EMIO-2550 module

To check that the system has correctly detected the EMIO-2550 module, use the following command:

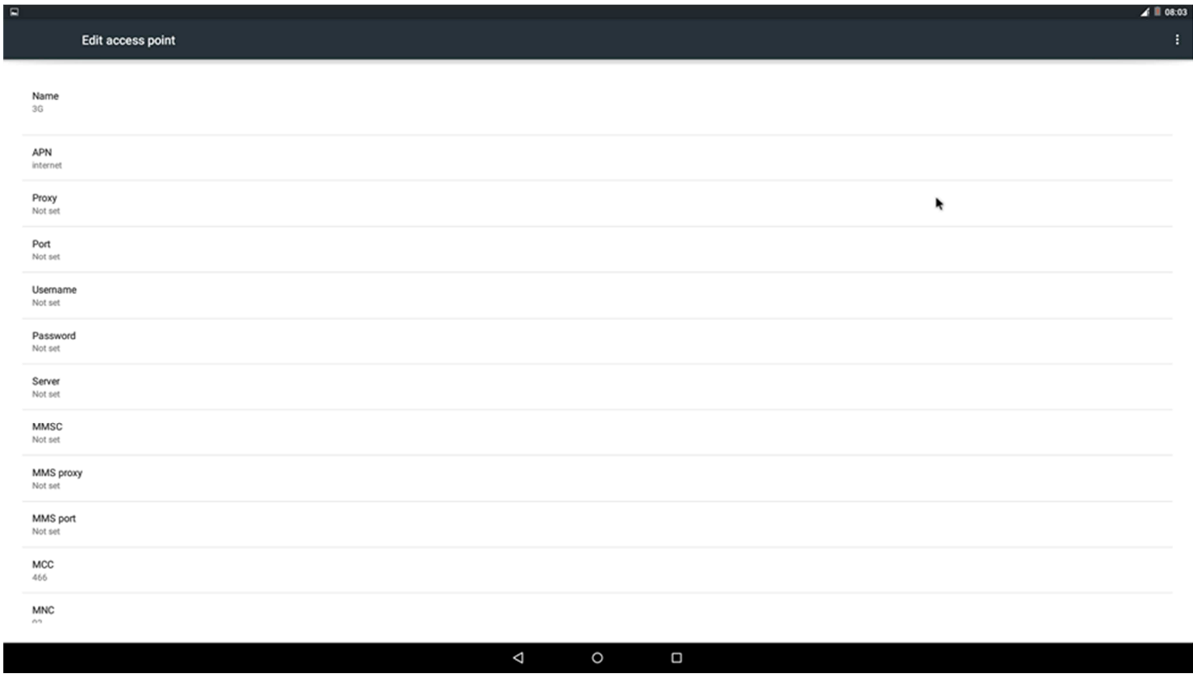
```
root@vab820_6dq:/ # busybox ifconfig
```

Make sure the printout message includes "ppp0 Link encap: Point-to-Point Protocol".

Click Settings->Wireless & networks-> More -> Cellular network ->Access Point Names. Next, click the **plus** button to add your APNs setting.



Fill in the required fields for APNs setting. If you are unsure of what the required fields and value are, check with your Mobile Broadband provider.



Name
3G

APN
internet

Proxy
Not set

Port
Not set

Username
Not set

Password
Not set

Server
Not set

MMSC
Not set

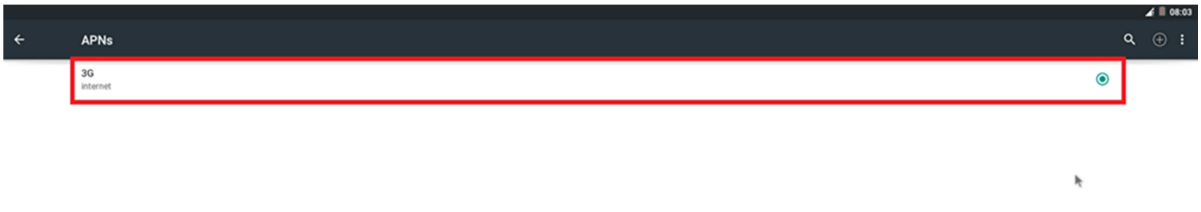
MMS proxy
Not set

MMS port
Not set

MCC
466

MNC

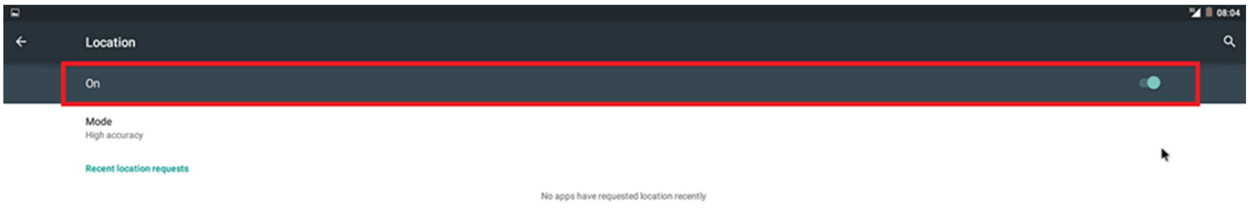
When the APNs setting is completed, click your APNs to enable the 3G network. Next, open the browser to connect to the Internet.



3G
internet

5.3.2. Enable GPS

Click Settings->Location to enable the option.



Location

On

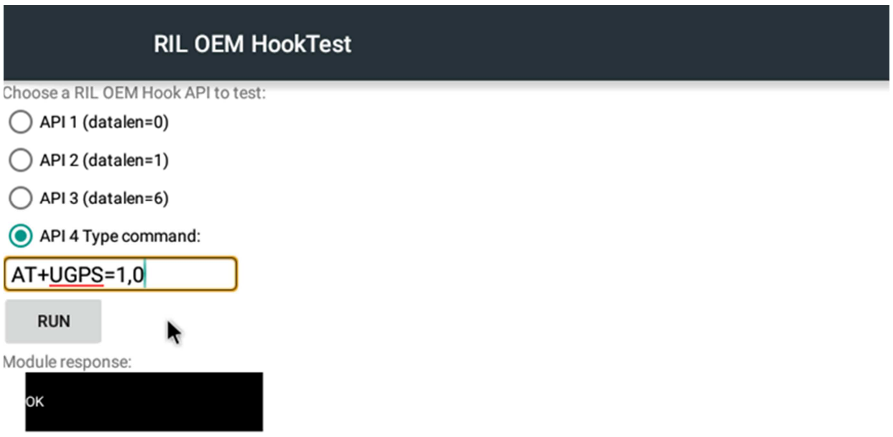
Mode
High accuracy

Recent location requests

No apps have requested location recently

Click Settings -> Wireless & networks-> More -> RIL OEM Hook Test to select "API 4 Type command. Full in "AT+UGPS=1,0" and click **RUN** button.

Wait for the "OK" message and open the GPS program.



RIL OEM HookTest

Choose a RIL OEM Hook API to test:

○ API 1 (datalen=0)

○ API 2 (datalen=1)

○ API 3 (datalen=6)

● API 4 Type command:

AT+UGPS=1,0

RUN

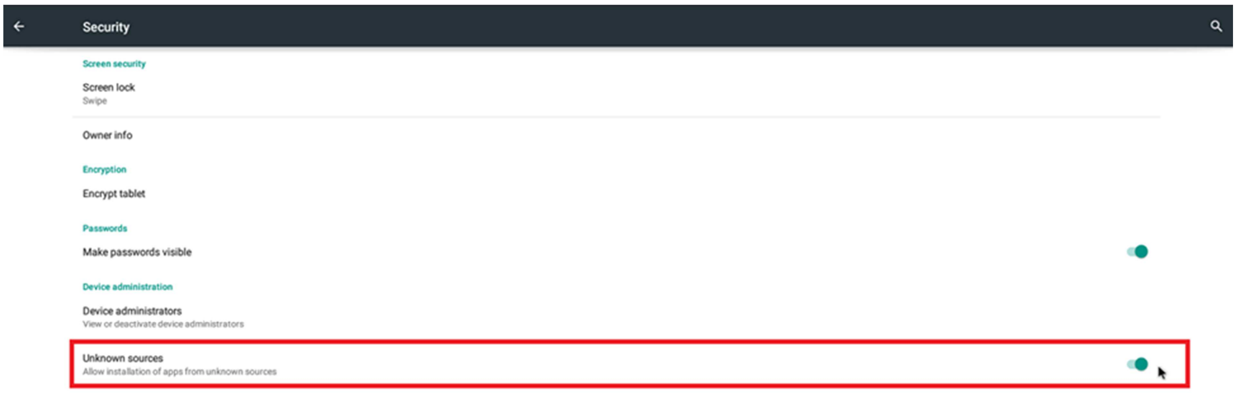
Module response:

OK

5.4. Smart ETK

The AMOS-820 Smart ETK supports Watchdog, GPIO, UART, and FlexCAN functions.

The first step is to copy the **VAB-820_Smart_ETK_Demo_v0.0.16.apk** onto a mass storage device such as USB thumb drive. Next, from the Settings screen, click Security -> Unknown sources to allow installation of non-Market apps.



Security

Screen security

Screen lock
Swipe

Owner info

Encryption

Encrypt tablet

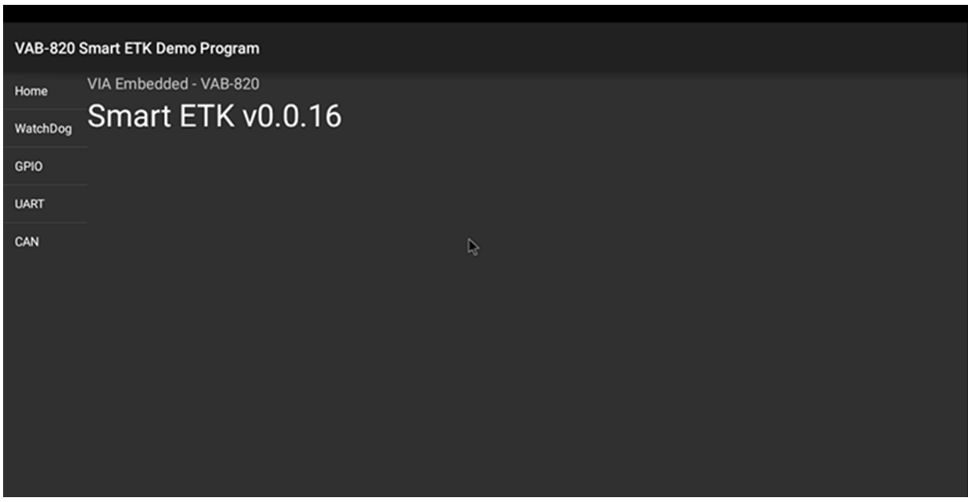
Passwords

Make passwords visible

Device administration

Device administrators
View or deactivate device administrators

Unknown sources
Allow installation of apps from unknown sources



VAB-820 Smart ETK Demo Program

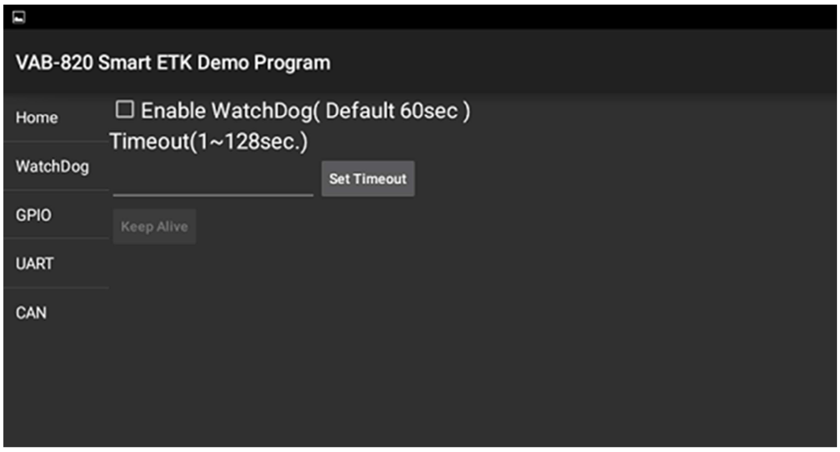
VIA Embedded - VAB-820

Smart ETK v0.0.16

Smart ETK Watchdog timer diagram

5.4.1. Testing Watchdog Timer Function

The Watchdog timer includes Enable, Set Timeout, and Keep Alive functions.



VAB-820 Smart ETK Demo Program

☐ Enable WatchDog(Default 60sec)

Timeout(1~128sec.)

Smart ETK Watchdog timer diagram

The first step is to check the checkbox of **Enable WatchDog** and set the timeout value in [ ] Seconds (1~128 Seconds).

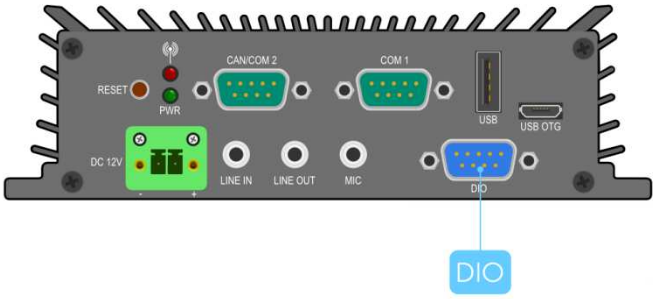
Click on the **Set Timeout** button to activate this setting.

Please note that if you only check the checkbox of **Enable WatchDog** without inputting the timeout value, the system will be reset in 60 seconds by default.

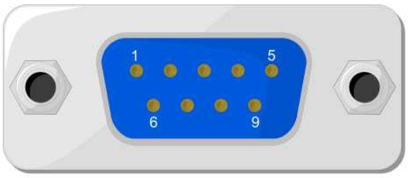
Click on the **Keep Alive** button to refresh the timeout value.

5.4.2. Testing GPIO Function

The DIO port on the AMOS-820 consists of 9 pins. The following section explains setting up these pins for input/output communication.



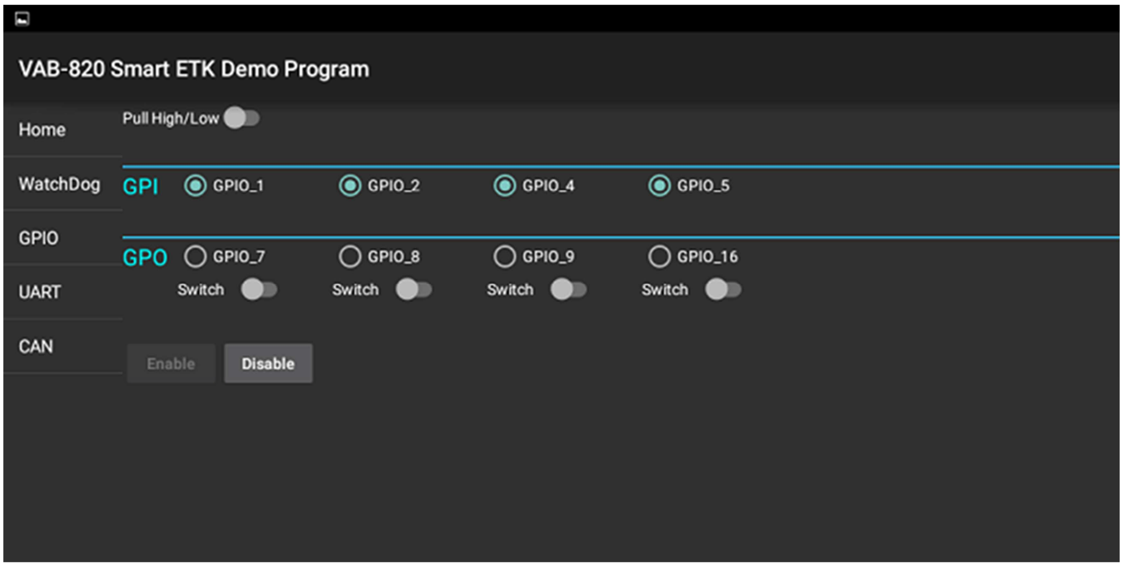
DIO port diagram



DIO port pinout diagram

Pin	Signal
1	GPIO_7
2	GPIO_1
3	GPIO_8
4	GPIO_2
5	GPIO_9
6	GPIO_4
7	GPIO_16
8	GPIO_5
9	GND

DIO pinout table



Smart ETK GPIO diagram

For demo program, the Pin 2, Pin 4, Pin 6 and Pin 8 are for GPI. The Pin 1, Pin 3, Pin 5 and Pin 7 are for GPO.

The first step is to click on the **Enable** button.

The GPI pins are low and the GPO pins are high by default.

When you switch Pull High/Low button to high, the GPO pins are high.

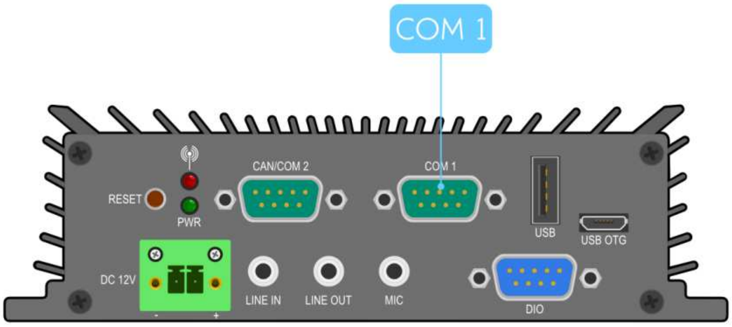
When you switch Pull High/Low button to low, the GPO pins are low.

When you click on Switch button, it changes the GPO pin status.

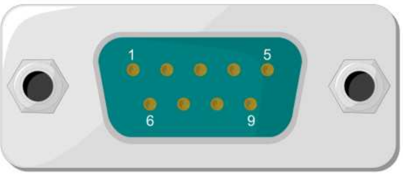
Click on the **Disable** button to disable this function.

5.4.3. Testing UART Function

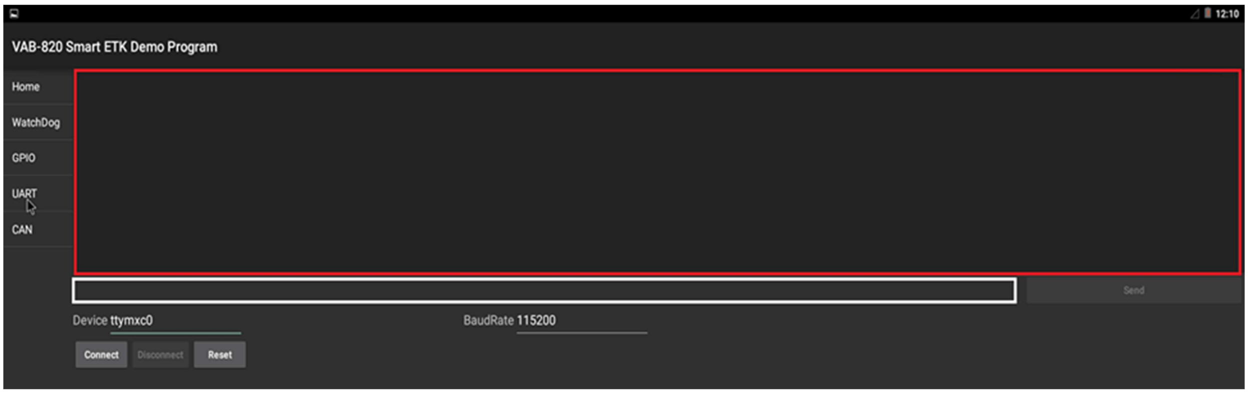
The COM 1 port of the AMOS-820 supports RS-232 standard with DTE mode. The Smart ETK UART function supports TX/RX to communicate with other device.



COM 1 port diagram



COM 1 port pinout diagram



VAB-820 Smart ETK Demo Program

Device ttymxc0

BaudRate 115200

Smart ETK UART diagram

The first step is to click on the **Connect** button.

When other devices transfer data to the AMOS-820, the data will be displayed inside the red frame.

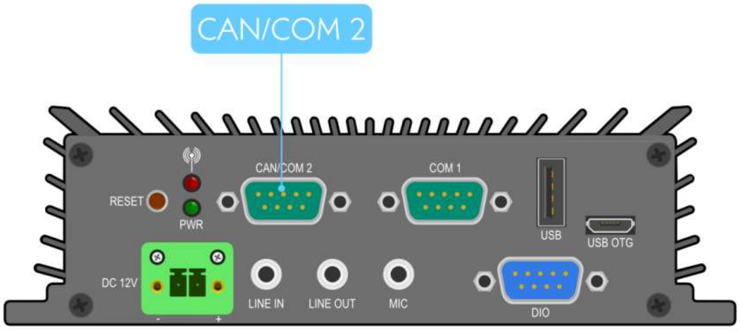
To send data to other devices, you need to type the data inside the white frame and click the **Send** button.

Click on the **Reset** button to reset this function.

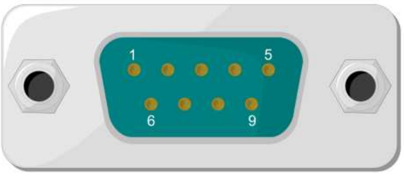
Click on the **Disconnect** button to disable this function.

5.4.4. Testing FlexCAN Function

The CAN/Debug port (COM 2) of the AMOS-820 supports debug port (COM 2) and two ports of CAN bus. The CAN bus supports CAN protocol specification Version 2.0 B while the COM 2 supports TX/RX for debugging purposes only.



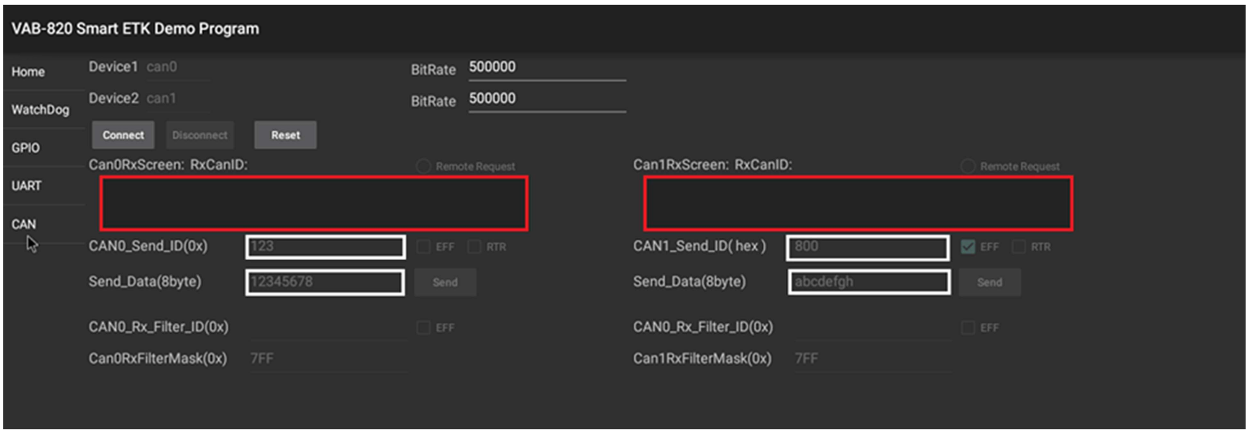
CAN/Debug port (COM 2) diagram



CAN/Debug port (COM 2) pinout diagram

Pin	Signal
1	CANH1
2	RX
3	TX
4	CANL2
5	GND
6	CANL1
7	GND
8	CANH2
9	VCC5

CAN/Debug port (COM 2) pinout table



Device1 can0 BitRate 500000

Device2 can1 BitRate 500000

Can0RxScreen: RxCanID: ○ Remote Request

CAN0_Send_ID(0x) 123 □ EFF □ RTR

Send_Data(8byte) 12345678

CAN0_Rx_Filter_ID(0x) □ EFF

Can0RxFilterMask(0x) 7FF

Can1RxScreen: RxCanID: ○ Remote Request

CAN1_Send_ID(hex) 800 ☑ EFF □ RTR

Send_Data(8byte) abcdefgh

CAN0_Rx_Filter_ID(0x) □ EFF

Can1RxFilterMask(0x) 7FF

Smart ETK FlexCAN diagram

For Smart ETK demo program, the CAN bus 1 is can0 and the CAN bus 2 is can1.

The first step is to set the BitRate value and click on the **Connect** button.

Please note that you need to disable the CAN bus first then set the CAN bus bitrate. The CAN bus bitrate should be 5000 < [BRT_value] < 1000000.

When other devices transfer data to the AMOS-820, the data will be displayed inside the red frame.

To send data to other devices, you need to type ID and data inside the white frame and click the **Send** button.

Click on the **Reset** button to reset this function.

Click on the **Disconnect** button to disable this function.

Taiwan Headquarters

1F, 531 Zhong-zheng Road,
Xindian Dist., New Taipei City 231
Taiwan

Tel: 886-2-2218-5452
Fax: 886-2-2218-9860
Email: embedded@via.com.tw

USA

940 Mission Court
Fremont, CA 94539,
USA

Tel: 1-510-687-4688
Fax: 1-510-687-4654
Email: embedded@viatech.com

Japan

3-15-7 Ebisu MT Bldg. 6F,
Higashi, Shibuya-ku
Tokyo 150-0011
Japan

Tel: 81-3-5466-1637
Fax: 81-3-5466-1638
Email: embedded@viatech.co.jp

China

Tsinghua Science Park Bldg. 7
No. 1 Zongguancun East Road,
Haidian Dist., Beijing, 100084
China

Tel: 86-10-59852288
Fax: 86-10-59852299
Email: embedded@viatech.com.cn

Europe

Email: embedded@via-tech.eu