QUICK START GUIDE

VAB-600

Android BSP v1.2.2

1.00-09072016-160200

Revision History

Version	Date	Remarks
1.00	9/07/2016	Initial release

Table of Contents

1. Introduction 1
1.1. Package Contents 1
1.1.1. BSP Folder Contents 1
1.1.2. Document Folder Contents 1
1.1.3. EVK Folder Contents 1
1.2. Version Information and Supported Features 2
2. Image Development 3
2.1. Booting from the eMMC 3
3. Build Environment Setup 4
3.1. Setting Up the Cross-Compiling Environment 5
4. Image Build 6
4.1. Building the U-Boot Binary 6
4.2. Building the Android File System 6
4.3. Making System Booting Media 7
5. Hardware Functions 8
5.1. Configuring U-Boot Display Parameters 8
5.1.1. HDMI (default setting) 8
5.1.2. LVDS (Acmepoint TP070C01 / AUO G070VW01 / AUO G104XVN01.0) 9
5.2. Smart Functions 9
5.2.1. Testing RTC Wake-up Function 10
5.2.2. Testing Watchdog Timer Function 10
5.2.3. Testing GPIO Function 11
5.2.4. Testing UART Function 14
6. SPI ROM Backup and Recovery 15
6.1. Hardware Information 15
6.2. Software Information 16
6.2.1. Backing up the SPI ROM Data 16
6.2.2. Reflashing the SPI ROM 16
6.2.3. Resetting the MAC Address 16

1. Introduction

The purpose of this document is to provide an overview of getting started with the VAB-600 board using Android.

1.1. Package Contents

There are three folders in the package as listed below.

BSP folder	Description
VAB-600_Android_source_code.zip	Source Code
Document folder	**Description**
VAB-600_Android_BSP_v1.2.2_Quick_Start_Guide_v1.00_20160907.pdf	Quick Start Guide
VAB-600_Android_EVK_v1.2.2_Image_Installation_Guide_v1.00_20160907.pdf	Image Installation Guide
EVK folder	**Description**
arm_201103_gcc4.5.2.tgz	Toolchain
VAB-600_Android_EVK_v1.2.2.zip	Precompiled Image
SmartETK_sample.apk	SmartETK demo program

VAB-600 Android BSP contents

1.1.1. BSP Folder Contents

VAB-600_Android_source_code.zip: is a complete Android BSP including kernel source code and U-Boot source code.

1.1.2. Document Folder Contents

VAB-600_Android_BSP_v1.2.2_Quick_Start_Guide_v1.00_20160907.pdf: The Quick Start Guide is to provide an overview of getting started with the VAB-600 board using Android.

VAB-600_Android_BSP_v1.2.2_Image_Installation_Guide_v1.00_20160907.pdf: The Image Installation Guide explains how to boot the Android EVK system image on the VAB-600 board in order to begin evaluating the platform.

1.1.3. EVK Folder Contents

arm_201103_gcc4.5.2.tgz: is a toolchain, which is a set of software development tools for building images for the VAB-600 board.

VAB-600_Android_EVK_v1.2.2.zip: is a precompiled Android image for evaluating the VAB-600 board.

SmartETK_sample.apk: is the demo program of the Smart ETK.

1.2. Version Information and Supported Features

- U-Boot version: 1.1.4
- Kernel version: 3.0.8
- Evaluation image: Android Ice Cream Sandwich 4.0.3
- Supports SPI with eMMC as default boot device
- Supports HDMI or LVDS single display
- Supports HDMI audio output
- Supports Acmepoint TFT-LCD resistive touch panel (through 4-wire interface)
    - Acmepoint 7" TP070C01 LVDS panel (800×480)
- Supports AUO LVDS capacitive touch panels (through USB interface)
    - AUO 7" G070VW01 LVDS panel (800×480)
    - AUO 10.4" G104XVN01.0 LVDS panel (1024×768)
- Supports COM 1 debug connector
- Supports CIR
- Supports Line-out and Mic-in
- Supports USB: keyboard, mouse, flash disk, hard disk, and UVC camera
- Supports 10/100Mbps Ethernet
- Supports EMIO-1533 USB Wi-Fi module
- Supports EMIO-2550 miniPCIe Mobile Broadband module
- Supports system bar auto hide when set FULL_SCREEN
- Supports Smart ETK v1.2.0: Watchdog, GPIO, RTC wake-up, and UART

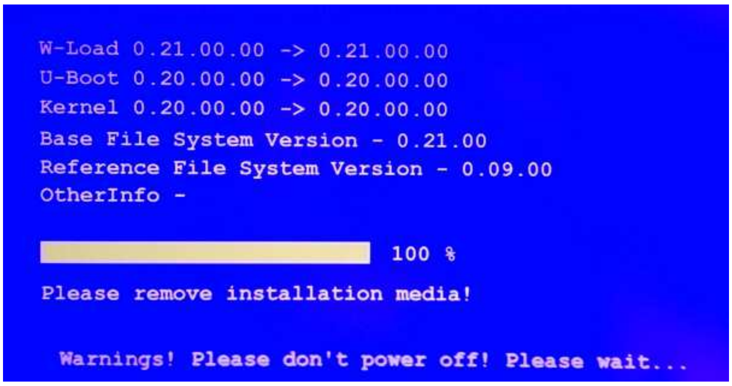
2. Image Development

This section explains the setup requirements for booting from the eMMC.

2.1. Booting from the eMMC

The first step is to insert a Micro SD card into the host machine and create a FAT formatted partition. Next, extract the **VAB-600_Android_EVK_v1.2.2.zip** and copy the **bspinst** folder and **scriptcmd** file onto the Micro SD card.

Insert the prepared Micro SD card into the VAB-600, connect an HDMI display, and power on the device to initiate the update process.



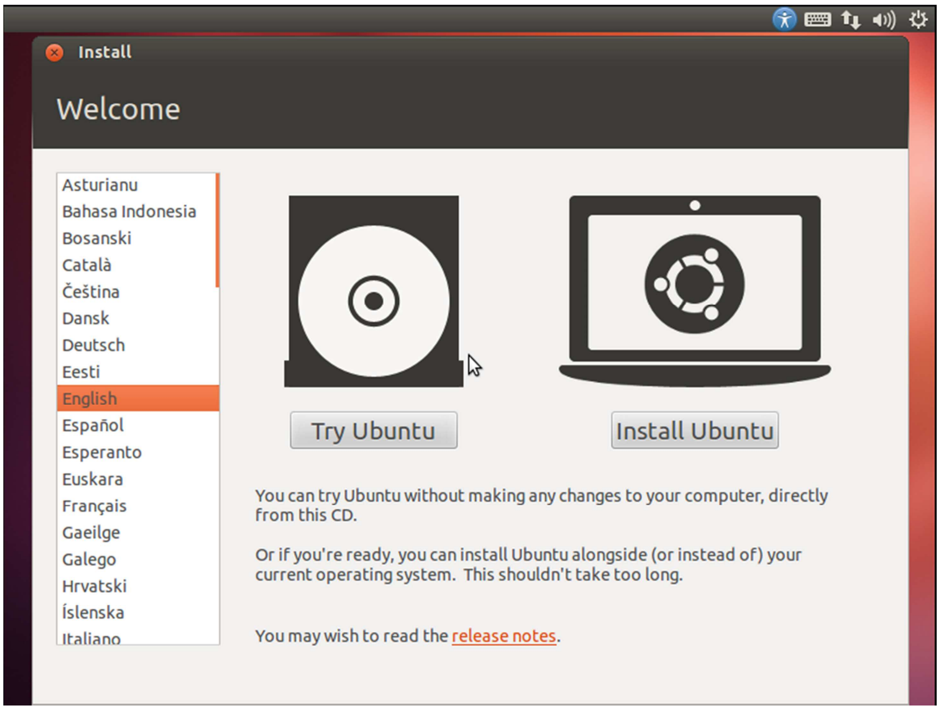
Update process screen

When the update process is completed, remove the Micro SD card. The system will automatically restart.

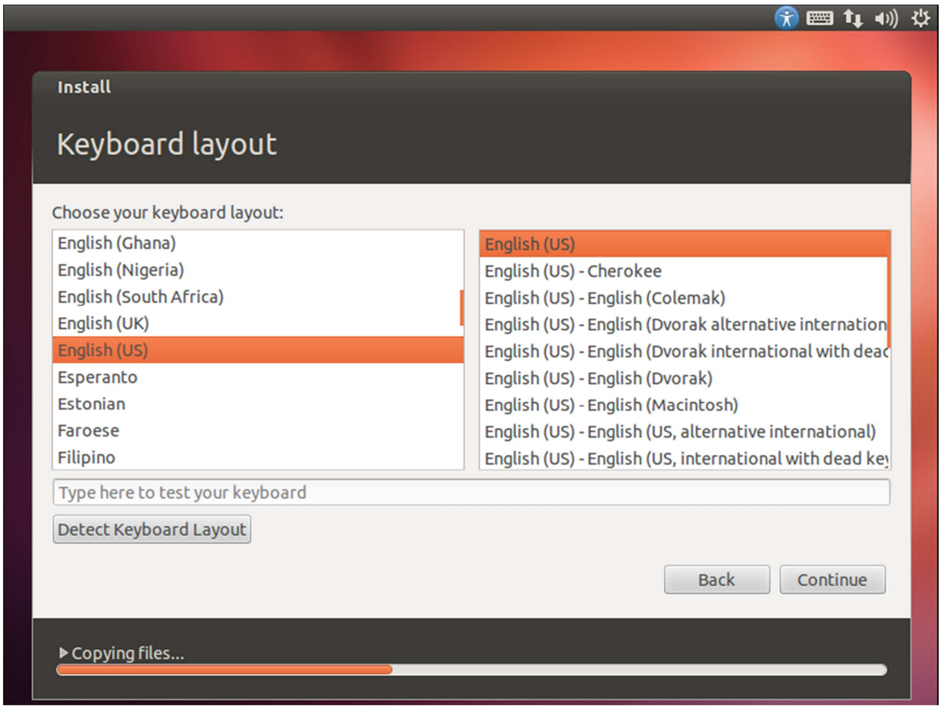
3. Build Environment Setup

This section guides you through setting up the build environment for development. All instructions are based on Ubuntu 12.04 LTS x64.

Please note that when installing the operating system, set the language setting to "English" and keyboard layout to "English (US)".



Setting language screen



Setting keyboard layout screen

3.1. Setting Up the Cross-Compiling Environment

Extract the **arm_201103_gcc4.5.2.tgz** file to /usr/local/arm/ (If this folder does not exist in the system, please create it manually).

```
$ tar –xzvf arm_201103_gcc4.5.2.tgz -C /usr/local/arm/
```

The cross compiler will then be found in the **/usr/local/arm/arm_201103_gcc4.5.2/** directory.

Download the **Java SE Development Kit 6u25** (jdk-6u25-linux-x64.bin) from the ORACLE website, and then install it using the following command:

```
$ sudo mkdir /usr/lib/jvm
$ mv jdk-6u25-linux-x64.bin /usr/lib/jvm
$ cd /usr/lib/jvm
$ sudo chmod 744 jdk-6u25-linux-x64.bin
$ ./jdk-6u25-linux-x64.bin
```

Add the toolchain path and Java path in **.bashrc** file.

```
export PATH=/usr/local/arm/arm_201103_gcc4.5.2/mybin/:$PATH
export JAVA_HOME=/usr/lib/jvm/jdk1.6.0_25
export CLASSPATH=.:$JAVA_HOME/lib
export PATH=$JAVA
```

Install the required software packages for cross-compilation.

```
$ sudo apt-get install libglapi-mesa:i386
$ sudo apt-get install git gnupg flex bison gperf build-essential \
  zip curl libc6-dev libncurses5-dev:i386 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 uboot-mkimage
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
$ sudo apt-get install xserver-xorg-lts-precise
```

Install the gcc 4.4 packages and set the system gcc version downgrade to 4.4.

```
$ sudo apt-get install gcc-4.4 g++-4.4 g++-4.4-multilib
$ sudo rm /usr/bin/gcc
$ sudo rm /usr/bin/g++
$ sudo ln -s /usr/bin/gcc-4.4 /usr/bin/gcc
$ sudo ln -s /usr/bin/g++-4.4 /usr/bin/g++
```

4. Image Build

4.1. Building the U-Boot Binary

This section describes how to build the U-Boot image from the source code.

Extract the U-Boot source.

```
$ tar -xzvf uboot.130522.tar.gz
```

Type the following command in order to use the default configurations.

```
$ cd uboot.130522
$ make wmt_config
```

Build the U-boot image with the ARM cross compiler.

```
$ make
```

When the process is completed, the u-boot.bin file will be stored in the uboot.130522 directory.

4.2. Building the Android File System

This section describes how to build the Android file system.

To begin, extract the kernel source and Android file system source.

```
$ tar -xzvf kernel_3.0.8.tar.gz
$ tar -xzvf android-4.0.3_r1.tar.gz
```

Type the following command to use the default configurations and compile the kernel.

```
$ cd kernel_3.0.8
$ make Android_defconfig
$ make clean
$ make
```

When the compilation process is completed, the resulting uzImage.bin file will be stored in the root folder of the kernel source folder.

Copy the uzImage.bin file into the Android folder for kernel update.

```
$ cp kernel_3.0.8/ANDROID_3.0.8/uzImage.bin android-4.0.3_r1/device/wmt/wmid/kernel
```

Type the following commands to build the Android file system.

```
$ cd android-4.0.3_r1
$ make clean
$ make update-api TARGET_PRODUCT=wmid TARGET_BUILD_VARIANT=user PRODUCT_BRAND=VAB
PRODUCT_MODEL=VAB-600 PRODUCT_MODEL_UNIQUE_ID=WM8950S BUILD_VERSION_TAGS=VAB-
600_BSP_v1.02.02_Evaluation
$ make -j3 TARGET_PRODUCT=wmid TARGET_BUILD_VARIANT=user PRODUCT_BRAND=VAB
PRODUCT_MODEL=VAB-600 PRODUCT_MODEL_UNIQUE_ID=WM8950S BUILD_VERSION_TAGS=VAB-
600_BSP_v1.02.02_Evaluation
```

When the process is completed, compress the Android file system using the command below:

```
$ mkdir rootfs
$ cp -a android-4.0.3_r1/out/target/product/wmid/system/* rootfs/
$ sudo tar -zcf rootfs.android4.0.3_[date].tgz -C rootfs .
```

Note: [date] is the date of the compression file, for example 20160711.

4.3. Making System Booting Media

The first step is to insert a Micro SD card into the host machine and create a FAT formatted partition. Next, extract the VAB-600_Android_EVK_v1.2.2.zip and copy the bspinst folder and scriptcmd file onto the Micro SD card.

To replace the u-boot.bin, enter the following command:

```
$ cp uboot.130522/u-boot.bin [SD]/bspinst/
```

Note: [SD] is the path of the mounted Micro SD card.

To replace the Linux kernel image, enter the following command:

```
$ cp android-4.0.3_r1/out/target/product/wmid/boot.img [SD]/bspinst/
$ cp android-4.0.3_r1/out/target/product/wmid/recovery.img [SD]/bspinst/
```

Note: [SD] is the path of the mounted Micro SD card.

To replace the Android file system, enter the following command:

```
$ rm [SD]/bspinst/packages/rootfs.android4.0_151015.1444_evk.tgz
$ cp rootfs.android4.0.3_[date].tgz [SD]/bspinst/packages
```

Notes: [date] is the date of the compression file.
[SD] is the path of the mounted Micro SD card.

5. Hardware Functions

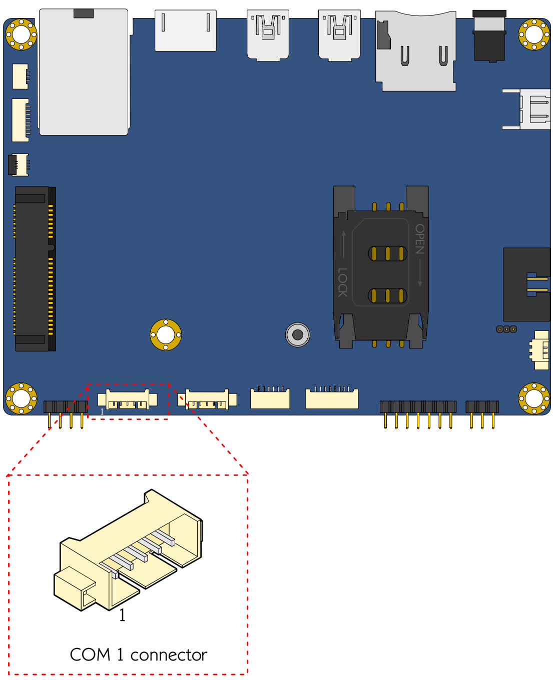
5.1. Configuring U-Boot Display Parameters

The VAB-600 Android BSP v1.2.2 supports the following display devices:

- HDMI monitor
- Acmepoint 7" TP070C01 LVDS panel (800x480)
- AUO 7" G070VW01 LVDS panel (800x480)
- AUO 10.4" G104XVN01.0 LVDS panel (1024x768)

5.1.1. HDMI (default setting)

Connection:
Connect your display to the VAB-600 board through the Mini HDMI connector, and then connect the VAB-600 board and the host machine through the debug COM 1 connector.



Debug COM 1 connector diagram

Use the following command to update the U-Boot parameters.

```
WMT # setenv wmt.display.param 4:6:1:1920:1080:60
WMT # saveenv
WMT # reset
```

1920:1080 is the resolution of the HDMI monitor used in this example.

Please note that the system will automatically detect the EDID information and set the monitor default resolution. Configuring U-Boot only affects the resolution of the U-Boot logo.

5.1.2. LVDS (Acmepoint TP070C01 / AUO G070VW01 / AUO G104XVN01.0)

Connection:
Connect the LVDS panel to the VAB-600 board through the LVDS connector on the VAB-600-D card, and then connect the VAB-600 board and the host machine through the debug COM 1 connector.

Use the following command to update the U-Boot parameters for 7" LVDS panel.

```
WMT # setenv wmt.display.param 2:0:24:800:480:60
WMT # saveenv
WMT # reset
```

800:480 is the resolution of the 7" LVDS panel (Acmepoint TP070C01 / AUO G070VW01).

Use the following command to update the U-Boot parameters for 10.4" LVDS panel.

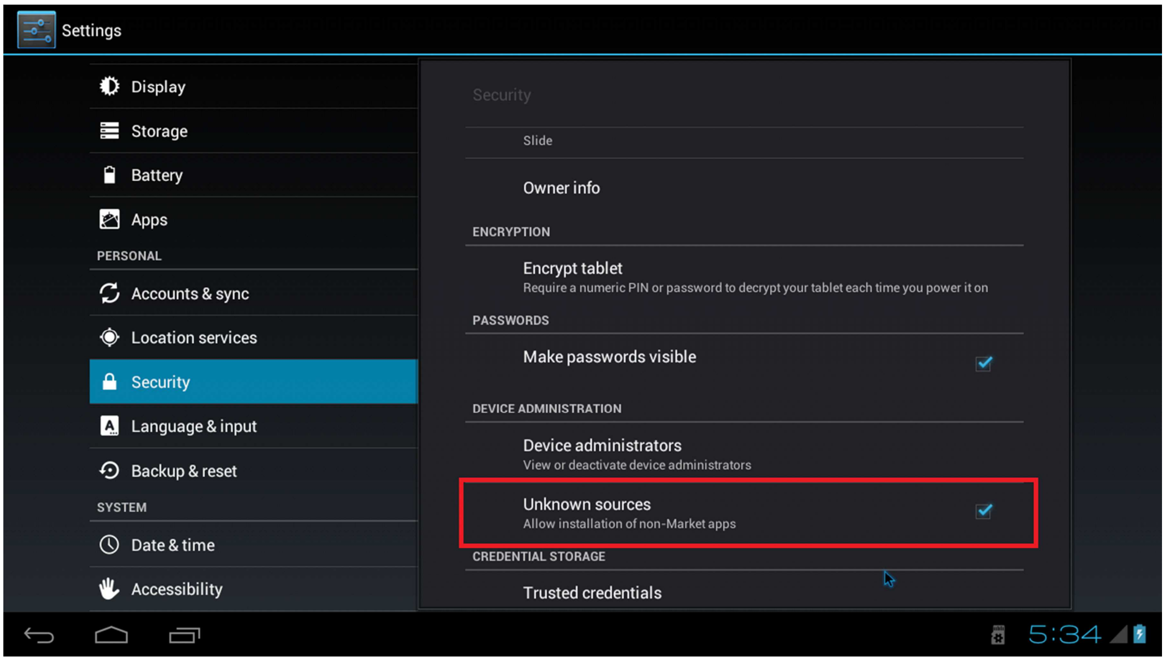
```
WMT # setenv wmt.display.param 2:0:24:1024:768:60
WMT # saveenv
WMT # reset
```

1024:768 is the resolution of the 10.4" LVDS panel (AUO G104XVN01.0).

5.2. Smart Functions

The VAB-600 Smart ETK supports RTC wake-up, Watchdog timer, GPIO and UART functions.

The first step is to copy the SmartETK_sample.apk onto a mass storage device such as a USB thumb drive. Next, from the Settings screen, click Security, and then check the checkbox of "Unknown sources".



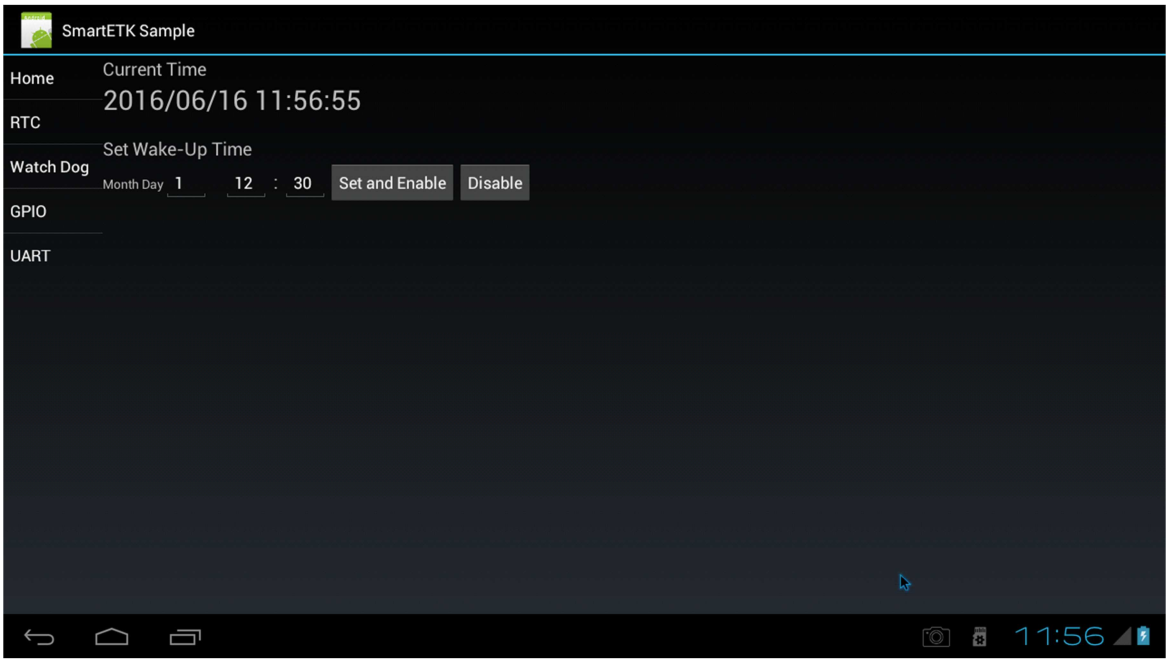
Smart ETK RTC wake-up diagram

Finally, insert the USB thumb drive into the VAB-600 and install the SmartETK_sample.apk.

When the installation process is completed, operate the SmartETK_sample.apk to test function.

5.2.1. Testing RTC Wake-up Function

RTC Wake-up allows you to set a predetermined wake-up point for a particular time.



Smart ETK RTC wake-up diagram

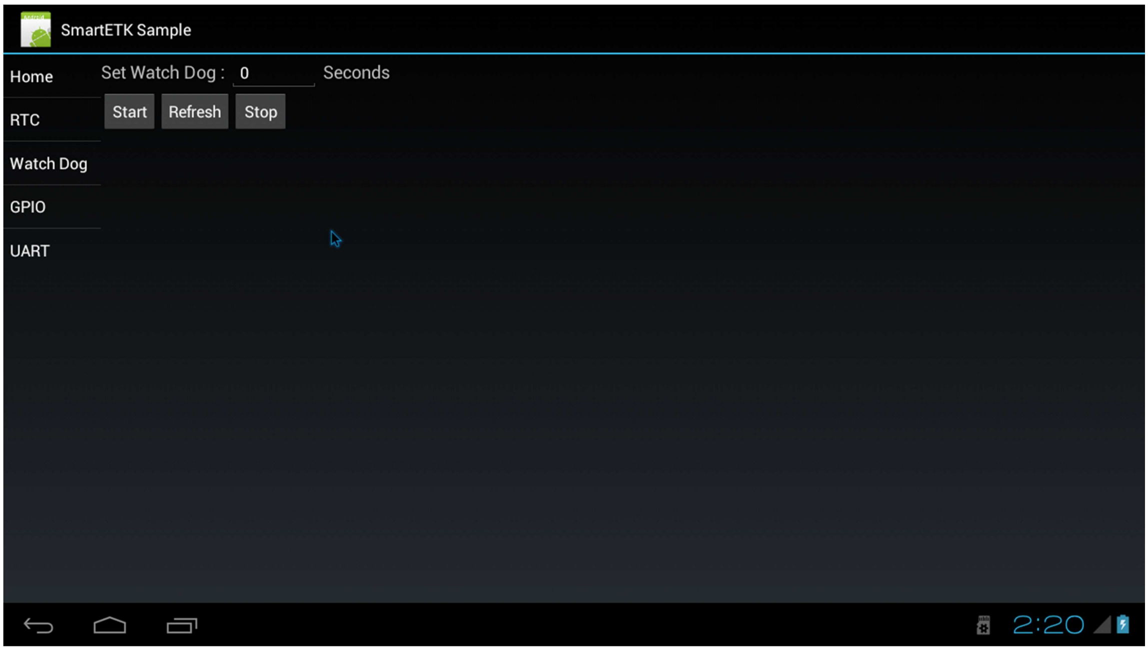
From the example above, the setting is to wake up the system at 12:30 (24hrs format) on the 1st day of each month.

Once the setting is completed, click on the **Set and Enable** button to activate.

The **Disable** button is to disable the RTC wake-up function.

5.2.2. Testing Watchdog Timer Function

The Watchdog timer includes Start, Refresh and Stop functions.



Smart ETK Watchdog timer diagram

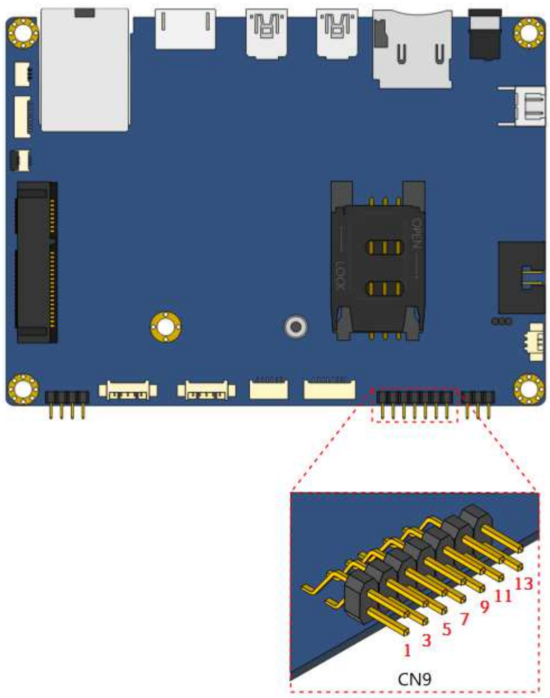
The first step is to set the countdown value in [0~1400] Seconds, and then click on the **Start** button to activate this setting.

To refresh the countdown value, click the **Refresh** button.

To disable the setting, click the **Stop** button.

5.2.3. Testing GPIO Function

Before testing the GPIO function, user must know that the GPIO uses Pin 5 to Pin 12 of CN9 pin header. The onboard pin header and pinout are shown below.

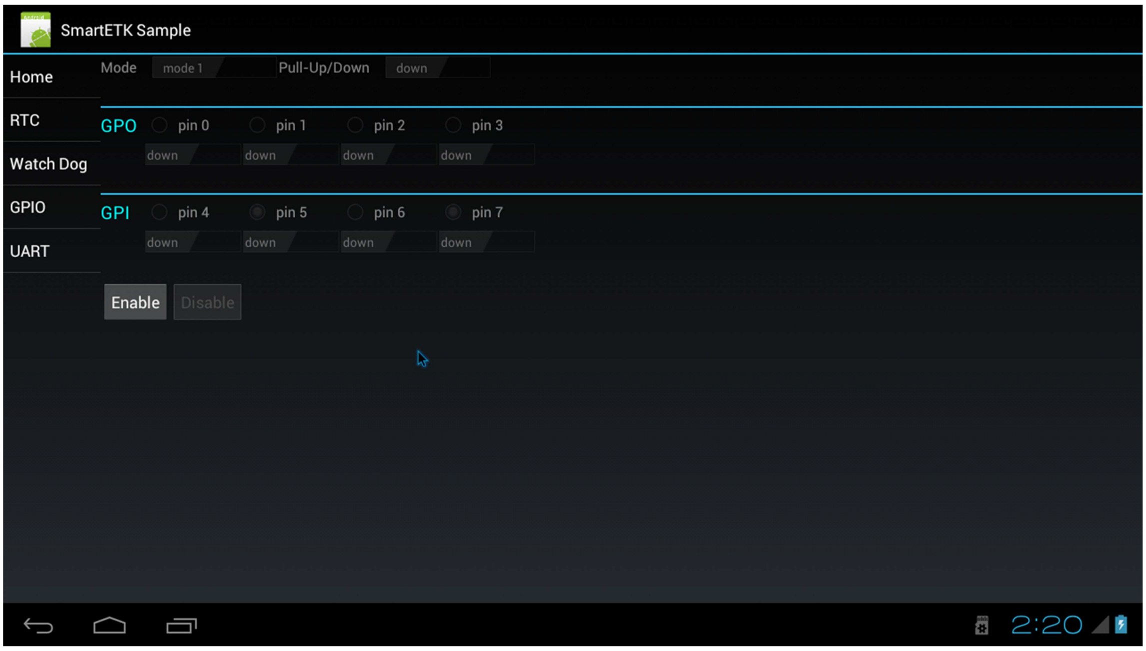


CN9 pin header diagram

Pin	Signal	Pin	Signal
1	VCC33	2	VCC_5V
3	GND	4	VSUS33
5	pin 0	6	pin 4
7	pin 1	8	pin 5
9	pin 2	10	pin 6
11	pin 3	12	pin 7
13	I2C0SDA	14	I2C0SCL

CN9 pin header pinout table

To enable the GPIO function, click the **Enable** button.



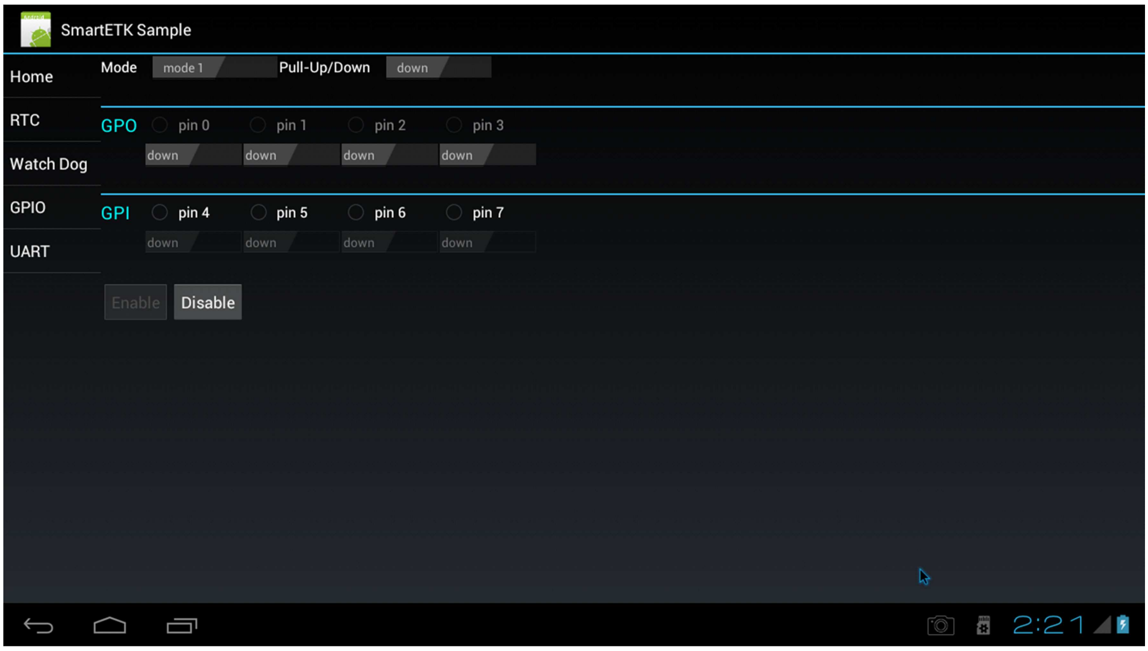
Smart ETK GPIO diagram

The GPIO function has four GPO and four GPI. The pin 0 to pin 3 are GPO and pin 4 to pin 7 are GPI. The GPO pins are "up" and the GPI pins are "down" by default.

The Mode button is used to switch the GPO and GPI pins. The default setting is "mode1".

The Pull-Up/Down button is used to change the GPI pin status. When you switch the Pull-Up/Down button to "up", the GPI pins are high. When you switch the Pull-Up/Down button to "down", the GPI pins are low.

To configure the GPO pins, select the GPO pins you want to set to "up" or "down" state.



Smart ETK GPIO enabling diagram

To switch the GPO and GPI pins, set Mode button to "mode2". Then pin 0 to pin 3 becomes the GPI, and pin 4 to pin 7 becomes the GPO.

Smart ETK GPIO switch diagram

To configure the GPI pins, switch the Pull-Up/Down button to "up".

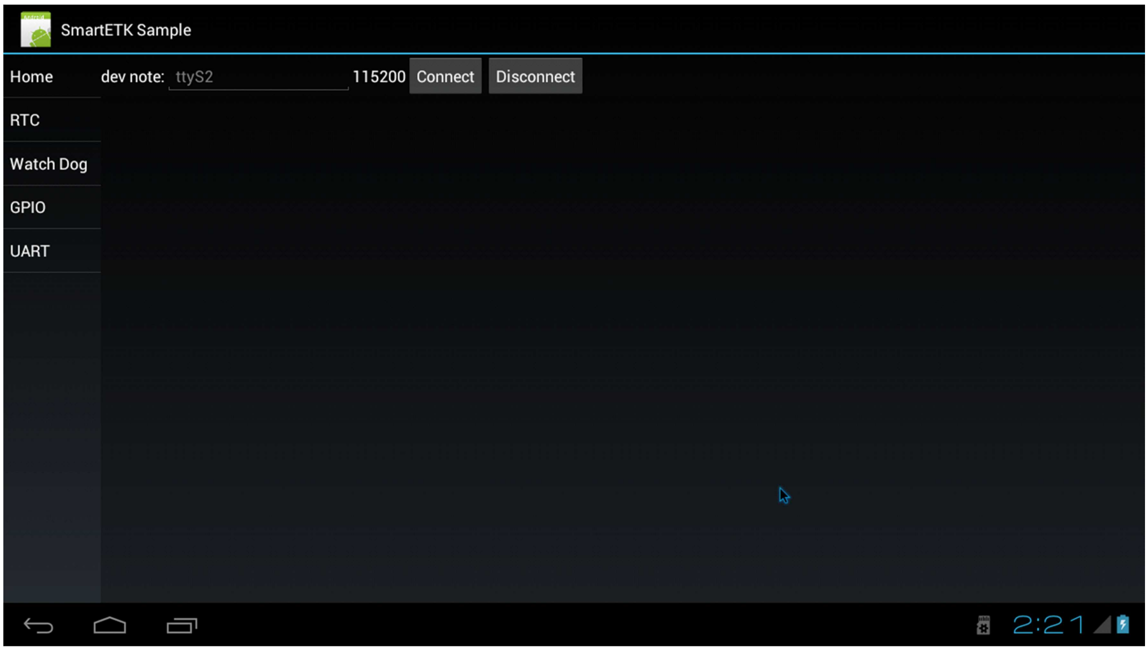
Smart ETK GPI pull-up diagram

To disable this function, click the **Disable** button.

5.2.4. Testing UART Function

Connect the host machine to COM 1 connector on the VAB-600 using the RS-232 cable.

Start a serial communication program such as Minicom or GtkTerm on the host machine with the same serial port settings using the appropriate serial device, and then click the **Connect** button on the UART page. Afterwards, the host machine and the VAB-600 board will be able to communicate with each other through the programs (for example, sending a keypress on one machine should be shown also on the other machine).



Smart ETK UART diagram

6. SPI ROM Backup and Recovery

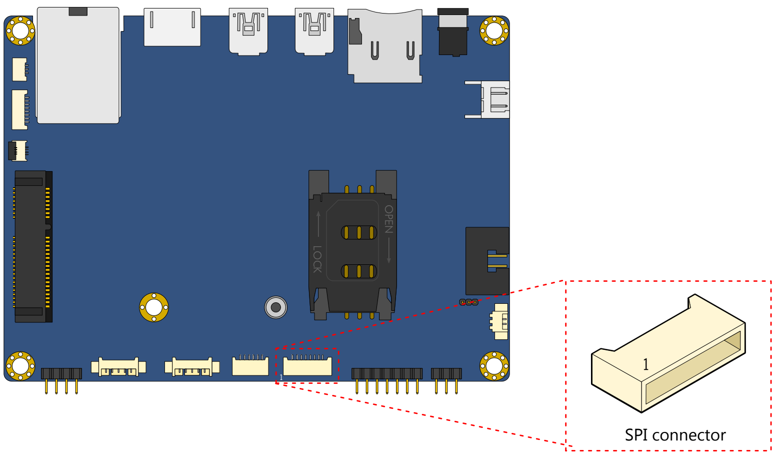
On the VAB-600 board, the SPI Flash ROM contains the U-Boot bootloader and its settings. If the ROM contents become corrupted due to a failure during a firmware update, or are flashed with a non-functional U-Boot binary during development, the board may become non-bootable. In that case, you will need to reflash the SPI ROM with a workable binary image in order to recover the board.

This section explains the SPI ROM backup and recovery procedures for the VAB-600 board.

6.1. Hardware Information

To backup or reflash the SPI ROM, a SPI Flash programmer is required. Dediprog SF100[1] is used as an example in this guide, however any other programmers can also be used as long as they are capable of handling the Mactronix MX25L4005 serial Flash chip on the VAB-600.

You will need to make your own SPI connector cable to connect the Flash programmer to the VAB-600. The SPI connector (marked as SPI1) is an ACES[2] 87213-0800G connector, which is compatible with the ACES50233-008H0H0-001 housing and 87214-series cramp-on wafer-to-board terminals. The onboard connector and pinouts are shown below.



SPI connector diagram

The pinout matching between the SF100 and the SPI connector is shown in the table below.

SF100		SPI	
Pin	Signal	Pin	Signal
8	I/O3	1	(not connected)
7	Vpp/Acc	2	SFCS1
6	MOSI	3	SFDO
5	MISO	4	SFDI
4	CLK	5	SFCLK
3	CS	6	SFCS0-
2	GND	7	GND
1	Vcc	8	VPROG_SP1(3.3V)

SFI100 and SPI connector pinout table

[1] See the Dediprog SF100 product page at http://www.dediprog.com/pd/spi-flash-solution/SF100
[2] See the ACES website at http://www.acesconn.com/

6.2. Software Information

For the backup and reflash procedures you should use the software provided by the vendor of your Flash programmer. When using the Dediprog SF100, download the "dediprog" programmer from their product page. Alternatively, flashrom[3] is a tool that supports a large number of different flash programmers.

6.2.1. Backing up the SPI ROM Data

First, make sure the VAB-600 board is powered off. Connect your Flash programmer and read out the contents of the SPI ROM into a backup file.

For the Dediprog SF100, this task can be performed by using the graphical user interface tool in Windows or the following command in Windows or Linux.

```
$ dpcmd -r SF_BACKUP.bin
```

This saves the contents of the Flash chip to the file SF_BACKUP.bin.

For other Flash programmers, refer to their documentation to correctly read the chip contents.

6.2.2. Reflashing the SPI ROM

Use either the file you just backed up (refer to the previous section), or the default SPI ROM image that is included in the VAB-600 BSP/EVK. The file name of the default image is SF_BOOTROM.bin, which can be found in the bspinst folder inside the VAB-600_Android_EVK_v1.2.2.zip file.

First, make sure the VAB-600 board is powered off. Connect your Flash programmer and perform an erase-and-write cycle.

For the Dediprog SF100, this task can be performed by using the graphical user interface tool in Windows, or the following command in Windows or Linux.

```
$ dpcmd -e -p SF_BOOTROM.bin
```

SF_BOOTROM.bin is the Flash ROM image that is used to reflash the ROM (in this case the default image).

For other Flash programmers, refer to their documentation to correctly write the chip contents.

6.2.3. Resetting the MAC Address

If the image you flashed is the default Flash ROM image from VIA, you will need to reset the MAC address in the U-Boot settings (also stored on the SPI ROM), as the default image will overwrite those settings.

Connect the VAB-600 and the host machine through the debug COM 1 connector. Update the U-Boot parameters as shown below, replacing xx:xx:xx:xx:xx:xx with the correct MAC address of your VAB-600 board.

```
WMT # setenv ethaddr xx:xx:xx:xx:xx:xx
WMT # saveenv
WMT # reset
```

[3] See the flashrom website at http://flashrom.org/Flashrom

The MAC address can be found on the sticker located on the top of the Ethernet connector.

Ethernet MAC address sticker

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