



QUICK START GUIDE

ARTiGO A600

Linux BSP v1.2.1



Copyright

Copyright © 2016 VIA Technologies Incorporated. All rights reserved.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

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 to make changes to the products described in this manual at any time without prior notice.



Revision History

Version	Date	Remarks
1.00	6/23/2016	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.....	4
4. Image Build	5
4.1. Building the U-Boot Binary	5
4.2. Building the Linux Kernel.....	5
4.3. Building the EMIO-1533 USB Wi-Fi Module Driver.....	6
5. Hardware Functions	7
5.1. Configuring the EMIO-2550 miniPCIe Mobile Broadband Module.....	7
5.1.1. Connecting to the Internet.....	7
5.1.2. Enabling GPS.....	8
5.2. Setting the Watchdog Timer	9
5.3. Programming the GPIO	9
5.4. Configuring RTC Wake-Up	12
5.5. Enabling RS-485 ports.....	13
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 ARTiGO A600 system using Debian Linux.

The ARTiGO A600 system is based on the VIA VAB-600 Pico ITX board.

1.1. Package Contents

There are three folders in the package as listed below.

BSP folder	Description
ARTiGO-A600_Linux_source_code.zip	Source Code
Document folder	Description
ARTiGO-A600_Linux_BSP_v1.2.1_Quick_Start_Guide_v1.00_20160623.pdf	Quick Start Guide
ARTiGO-A600_Linux_EVK_v1.2.1_Image_Installation_Guide_v1.00_20160623.pdf	Image Installation Guide
EVK folder	Description
arm_201103_gcc4.5.2.tgz	Toolchain
setrs485	Enable RS-485 function
ARTiGO-A600_Linux_EVK_v1.2.1.zip	Precompiled Image

ARTiGO A600 Linux BSP contents

1.1.1. BSP Folder Contents

ARTiGO-A600_Linux_source_code.zip: is a complete Debian BSP including kernel source code, U-Boot source code, and non built-in driver source code of the EMIO-1533 USB Wi-Fi module.

1.1.2. Document Folder Contents

ARTiGO-A600_Linux_BSP_v1.2.1_Quick_Start_Guide_v1.00_20160623.pdf: The Quick Start Guide is to provide an overview of getting started with the ARTiGO A600 system using Debian Linux.

ARTiGO-A600_Linux_EVK_v1.2.1_Image_Installation_Guide_v1.00_20160623.pdf: The Image Installation Guide explains how to boot the Linux EVK system image on the ARTiGO A600 system 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 ARTiGO A600 system.

Setrs485: is a file used for enabling the RS-485 function in the ARTiGO A600 system.

ARTiGO-A600_Linux_EVK_v1.2.1.zip: is a precompiled Debian image for evaluating the ARTiGO A600 system.

1.2. Version Information and Supported Features

- U-Boot version: 1.1.4
- Kernel version: 3.0.8
- Evaluation image: Debian Wheezy 7.0
- Development based on VIA WM8950_Linux_BSP_1.00.02_121212
- Supports eMMC boot
- Supports HDMI display
- Supports HDMI audio output
- Supports 7-pin COM connector (RS-232) (1 Debug console + 1 COM port TX/RX only)
- Supports 8-pin DIO connector for 8-bit GPIO (4 GPI + 4 GPO)
- Supports four 3-pole Phoenix RS-485 ports
- 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 miniPCle Mobile Broadband module
- Supports Watchdog, GPIO, and RTC wake-up

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 **ARTiGO-A600_Linux_EVK_v1.2.1.zip** and copy the **bspinst** folder and **scriptcmd** file onto the Micro SD card.

Insert the prepared Micro SD card into the ARTiGO A600, connect an HDMI display, and power on the device to initiate the update process.

```

-----
Packages List:
W-Load - 0.11.00.00 (1G RAM Support)
U-Boot - 0.20.00.00 (Display Support v0.05)
Kernel - VAB-600_Linux_BSP_v0.07
FileSystem - VAB-600_Debian_Xorg_LXDE_v0.04.tar.tgz
Packages - v1.2.0

[Progress Bar] 0 %
creating partition table

Warnings! Please don't power off! Please wait...
  
```

Update process screen

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

Power on the ARTiGO A600 again, and wait for the login prompt to appear.

The default user name is *debian* and the password is *temppwd*.

3. Build Environment Setup

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

3.1. Setting Up the Cross-Compiling Environment

Make sure that the host machine is connected to the network and run the packages update.

```
$ sudo apt-get update
```

Install the required software packages for cross-compilation.

```
$ sudo apt-get install git-core gnupg flex bison gperf build-essential \
zip curl zlib1g-dev libc6-dev lib32ncurses5-dev ia32-libs \
x11proto-core-dev libx11-dev lib32readline-gplv2-dev lib32z1-dev \
libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown \
libxml2-utils xsltprocsharutils
```

The tool to create images for use with the U-Boot boot loader, "mkimage", is provided by different packages in Ubuntu 12.04 and in newer Ubuntu releases.

On Ubuntu 12.04 install it as below.

```
$ sudo apt-get install uboot-mkimage
```

On Ubuntu 14.04 and newer versions install it as below.

```
$ sudo apt-get install u-boot-tools
```

Confirm that the GNU C library version is 2.11 or higher.

```
$ ldd-version
```

Extract the toolchain to /usr/local/arm/ (If this folder does not exist in the system, please create it manually).

```
$ tar -xzf 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.

Export the toolchain to system PATH.

```
$ export PATH=/usr/local/arm/arm_201103_gcc4.5.2/mybin/:$PATH
```

To achieve the best floating point operations performance, create an alias for the toolchain.

```
# alias arm_1103_le-as='arm_1103_le-as -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
# alias arm_1103_le-c++='arm_1103_le-c++ -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
# alias arm_1103_le-cpp='arm_1103_le-cpp -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
# alias arm_1103_le-g++='arm_1103_le-g++ -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
# alias arm_1103_le-gcc='arm_1103_le-gcc -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
# alias arm_1103_le-gcc-4.5.2='arm_1103_le-gcc-4.5.2 -mcpu=cortex-a9 -mfp=neon -mfloat-abi=softfp'
```


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 -xzf uboot.130522_Display_Support_v0.05.tgz
```

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

```
$ cd uboot.130522_Display_Support_v0.05
$ make wmt_config
```

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

```
$ make CROSS_COMPILE=/usr/local/arm/arm_201103_gcc4.5.2/mybin/arm_1103_le-
```

When the process is completed, the **u-boot.bin** file will be stored in the **uboot.130522_Display_Support_v0.05** directory. Copy this file to the **bspinst** folder on the Micro SD card to update the U-Boot.

4.2. Building the Linux Kernel

This section describes how to build the kernel image from the source code.

To begin, extract the kernel source.

```
$ tar -xzf ARTiGO-A600_Linux_BSP_Kernel_Source_v0.08.tgz
```

Type the following command to use the default configurations.

```
$ cd ARTiGO-A600_Linux_BSP_Kernel_Source_v0.08/Kernel_3.0.8
$ make ARTiGO-A600_Linux_defconfig
```

Use the kernel's menuconfig to make modifications to the default settings.

```
$ make menuconfig
```

Finally, compile the kernel.

```
$ make ubin CROSS_COMPILE=arm_1103_le- -jX
```

"X" above represents the number of parallel jobs to be run by make. A reasonable default value is the number of CPU cores of your host system. For example, use "-j4" on a quad-core system.

When the compilation process is completed, the resulting **uzlimage.bin** file will be stored in the root folder of the kernel source folder. Copy this file to the **bspinst** folder on the Micro SD card for kernel update.

4.3. Building the EMIO-1533 USB Wi-Fi Module Driver

This section describes how to build the EMIO-1533 USB Wi-Fi module driver from the source code.

Extract the EMIO-1533 driver source.

```
$ tar -xzvf vt9271-202_20151209.tgz
```

Open the `vt9271-202/build/scripts/vt9271/Makefile.vt9271` file and edit "KERNELPATH" to match your own kernel source path. The following example uses `/home/work/ARTiGO-A600_Linux_BSP_Kernel_Source_v0.08/Kernel_3.0.8` as the kernel source path.

```
$ export KERNELPATH=/home/work/ARTiGO-A600_Linux_BSP_Kernel_Source_v0.08/Kernel_3.0.8
```

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

```
$ cd vt9271-202/build
$ make
```

When the process is completed, the `vt9271.ko` kernel module will be stored in the `vt9271-202/output/` directory. Copy this file to the `/lib/modules` folder on the ARTiGO A600 system for EMIO-1533 USB Wi-Fi module support.

5. Hardware Functions

5.1. Configuring the EMIO-2550 miniPCIe Mobile Broadband Module

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

The first step to enable these functions is to insert the EMIO-2550 module and an active SIM card into the ARTiGO A600. Then connect the ARTiGO A600 to the internet through LAN port, and power it on.

5.1.1. Connecting to the Internet

Type the following command to install the modemmanager package then reboot.

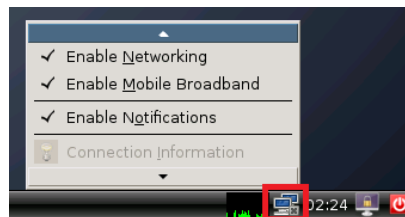
```
$ sudo apt-get install modemmanager
$ sudo reboot
```

After the system has rebooted, use the following command to confirm that the system has correctly detected the EMIO-2550 module.

```
$ lsusb
```

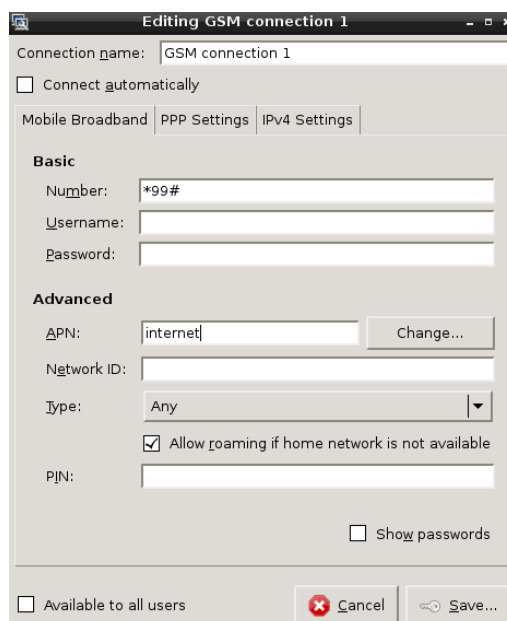
Look for "ID 1545:1102 U-Blox AG" in the ensuing list to ensure the module has been recognized.

To configure the Mobile broadband connection, click on the "Network Manager" icon located in the notification area.



Network Manger configuration

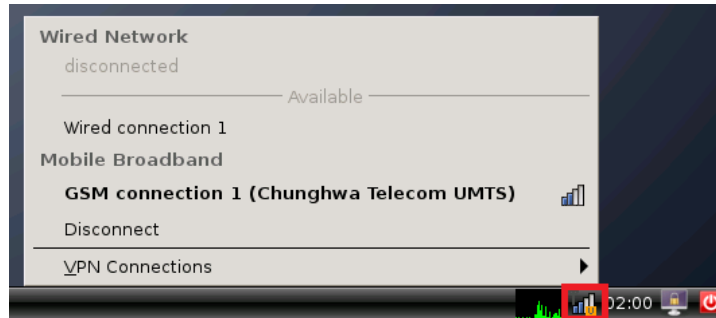
Fill in the required fields to setup your connection. If you are unsure of what the required fields and values are, check with your Mobile Broadband provider.



3G connection configuration

After the connection is created, click on the “Network Manager” icon again and select the connection you just created.

The successful connection looks like this:



Successful connection screen

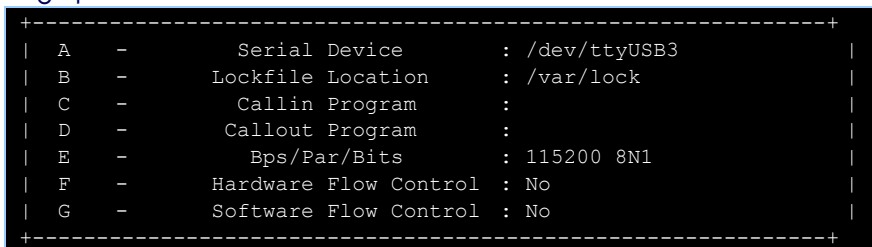
5.1.2. Enabling GPS

Use a serial port communication program such as Minicom or GtKTerm to configure the GPS function using the serial port setup configuration.

For example, install the Minicom program with the following command, and then open a Minicom terminal window.

```
$ sudo apt-get install minicom
$ sudo minicom -s
```

Select the “Serial port setup” option to configure the serial port, and then set the following settings provided in the screenshot below.



Serial port setting of GPS

When the setting is completed, type the following GPS AT commands into Minicom: AT+UGPRF=1, AT+UGRMC=1, AT+UGGSV=1, AT+UGZDA=1, and AT+UGPS=1,0

Wait for the OK message to appear before sending the next AT command.

```
AT+UGPRF=1
OK
AT+UGRMC=1
OK
AT+UGGSV=1
OK
AT+UGZDA=1
OK
AT+UGPS=1,0
OK
```

To ensure the GPS is functioning correctly, open a second Minicom terminal window and set the serial device to `/dev/ttyUSB7`. It will then output the data received from the GPS as illustrated in the screenshot below.

```
$GPRMC,14.2015,00,A,2501.93961,N,12133.66111,E,0.068,,030314,,A*73
$GPGSV,4,1,13,01,34,184,29,03,49,023,47,06,26,042,41,07,44,317,26*75
$GPGSV,4,2,13,08,13,323,,11,60,192,,13,29,242,38,16,33,071,16*79
$GPGSV,4,3,13,19,65,356,44,23,18,208,12,27,35,034,27,30,37,145,29*7B
$GPGSV,4,4,13,32,01,154,*4B
$GPZDA,142015.00,03,03,2014,00,00*62
```

GPS NMEA message

5.2. Setting the Watchdog Timer

This section describes how to use the Watchdog timer function of the ARTiGO A600.

To enable the Watchdog timer, enter the following command:

```
$ sudo /usr/bin/wdt_app-et
```

To disable the Watchdog timer, enter the following command:

```
$ sudo /usr/bin/wdt_app-dt
```

To set the Watchdog timer, enter the following command:

```
$ sudo /usr/bin/wdt_app-t<second>
```

<second>: Input the desired reboot delay time in seconds between 1 to 255.

To read the Watchdog countdown timer, enter the following command:

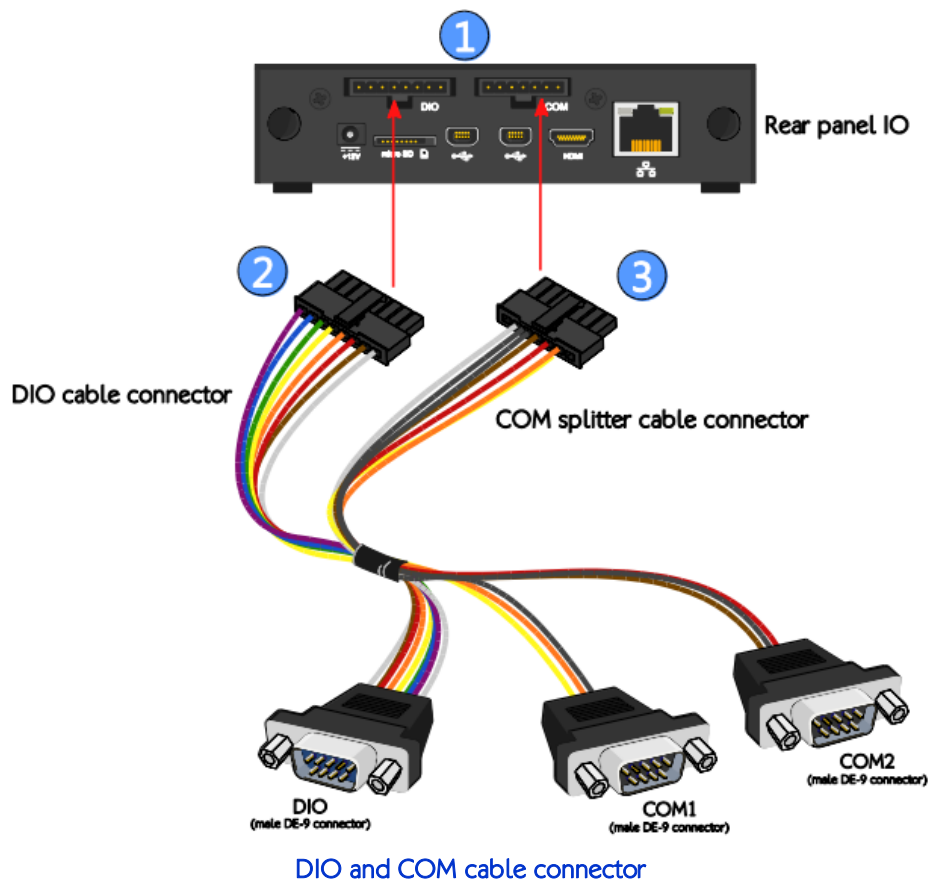
```
$ sudo /usr/bin/wdt_app-tg; echo $?
```

It will show how many seconds are left before the system reboots.

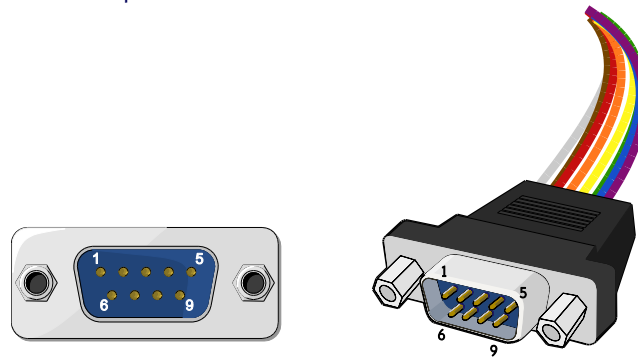
5.3. Programming the GPIO

This section describes how to use the GPIO on the ARTiGO A600.

Plug-in the DIO and COM splitter cable connectors on the ARTiGO A600 as illustrated in the figure below.



The DIO cable connector and pinout are show below.



DIO cable connector diagram

DIO pin #	Signal
1	GPIO20_CH
2	GPIO24_CH
3	GPIO21
4	GPIO25
5	GND
6	GPIO_26
7	GPIO_22
8	GPIO_27
9	GPIO_23

DIO cable connector pinout table

Use the devmem2 command to read/write the GPIO registers.

```
$ sudo devmem2 <address> <type> <data>
```

<address> : physical address of the register

<type> : data type to be read/written, [b]yte and [h]alfword(2 byte)

<data> : data to be written to the register

The corresponding addresses and BIT of GPIO pins are listed below:

Enable GPIO function: 1 for enable / 0 for disable				
Address	BIT	Attribute	Signal	Connector pin
0xd8110070	4:7	RW	GPIO[20:23]	1,3,7,9
0xd8110071	0:3	RW	GPIO[24:27]	2,4,6,8
Set GPIO mode: 1 for output mode / 0 for input mode				
Address	BIT	Attribute	Signal	Connector pin
0xd81100b0	4:7	RW	GPIO[20:23]	1,3,7,9
0xd81100b1	0:3	RW	GPIO[24:27]	2,4,6,8
GPIO output: 1 for high state / 0 for low state				
Address	BIT	Attribute	Signal	Connector pin
0xd81100f0	4:7	RW	GPIO[20:23]	1,3,7,9
0xd81100f1	0:3	RW	GPIO[24:27]	2,4,6,8
Read GPIO input: 1 for high state / 0 for low state				
Address	BIT	Attribute	Signal	Connector pin
0xd8110030	4:7	RW	GPIO[20:23]	1,3,7,9
0xd8110031	0:3	RW	GPIO[24:27]	2,4,6,8
Enable GPIO Pull-up Pull-down function: 1 for enable / 0 for disable				
Address	BIT	Attribute	Signal	Connector pin
0xd81104b0	4:7	RW	GPIO[20:23]	1,3,7,9
0xd81104b1	0:3	RW	GPIO[24:27]	2,4,6,8
Set GPIO pull up/down: 1 for up / 0 for down				
Address	BIT	Attribute	Signal	Connector pin
0xd81104f0	4:7	RW	GPIO[20:23]	1,3,7,9
0xd81104f1	0:3	RW	GPIO[24:27]	2,4,6,8

GPIO control register address table

The following example is to enable the GPIO 20 ~ GPIO 23 then set them to output mode and high state.

```
$ sudo devmem2 0xd8110070 b 0xf0
$ sudo devmem2 0xd81100b0 b 0xf0
$ sudo devmem2 0xd81100f0 b 0xf0
```

The following example is to enable and read the status of GPIO 24 ~ GPIO 27 then set them to input mode.

```
$ sudo devmem2 0xd8110071 b 0x0f
$ sudo devmem2 0xd81100b1 b 0x00
$ sudo devmem2 0xd8110031 b
```

5.4. Configuring RTC Wake-Up

RTC wake-up is enabled by the IDT1337G¹ chip to support auto power-on function. The IDT1337G chip is controlled by the I²C0 bus of ARTiGO A600.

The first step is to make sure the RTC time is consistent with the system time. To read the RTC time, use the following command:

```
$ sudo i2cdump -f -y -r 0-6 0 0x68
```

The following example value 03410101050516 means Monday, May 5th 2016 01:41:03.

```
root@arm:~# i2cdump -f -y -r 0-6 0 0x68
No size specified (using byte-data access)
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f      0123456789abcdef
00: 03 41 01 01 05 05 16                          ?A?????
```

RTC time setting sample

The following example has incorrect time value. The minutes in the RTC time is not consistent with the system time.

```
root@arm:~# date
Tue May 10 14:54:02 CDT 2016
root@arm:~# i2cdump -f -y -r 0-6 0 0x68
No size specified (using byte-data access)
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f      0123456789abcdef
00: 03 45 14 02 10 05 16                          ?E?????
```

RTC time setting with incorrect value

Use the following command to correct inconsistent values:

```
$ sudo i2cset -f -y 0 0x68 <address> <Value>
```

The corresponding addresses and values to make them consistent are listed below:

Address	Value	Description
0x0	0x\$(date +"%S")	Seconds
0x1	0x\$(date +"%M")	Minutes
0x2	0x\$(date +"%k")	Hours
0x3	0x\$(date +"%u")	day of week
0x4	0x\$(date +"%d")	date of month
0x5	0x\$(date +"%m")	Month
0x6	0x\$(date +"%g")	Year

RTC time setting table

From the example above, the minutes did not match. To correct this, enter the following command:

```
$ sudo i2cset -f -y 0 0x68 0x1 0x$(date +"%M")
```

For information of setting the RTC function and its activation time, please refer to "Timekeeper Registers" in the [IDT1337G's datasheet](#).

¹ See the IDT 1337G page at <http://www.idt.com/products/general-parts/1337g-real-time-clock-i2c-serial-interface>

5.5. Enabling RS-485 ports

Use the following command to check if the RS-485 ports correspond with their respective serial ports.

```
$ sudo cat /proc/tty/driver/usbserial
```

Look for name "Vizzini USB serial port", and then find the numbers at the beginning of those lines. In the following example, the RS-485 port 1/2/3/4 corresponds with the ttyUSB14/15/16/17 serial port.

```
root@arm:/home/debian/Desktop# cat /proc/tty/driver/usbserial
usbserinfo:1.0 driver:2.0
0: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
1: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
2: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
3: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
4: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
5: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
6: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
7: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
8: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
9: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
10: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
11: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
12: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
13: name:"GSM modem (1-port)" vendor:1546 product:1102 num_ports:1 port:1 path:usb-0000:00:04.0-1
14: name:"Vizzini USB serial port" vendor:04e2 product:1414 num_ports:1 port:1 path:usb-0000:00:04.0-2
15: name:"Vizzini USB serial port" vendor:04e2 product:1414 num_ports:1 port:1 path:usb-0000:00:04.0-2
16: name:"Vizzini USB serial port" vendor:04e2 product:1414 num_ports:1 port:1 path:usb-0000:00:04.0-2
17: name:"Vizzini USB serial port" vendor:04e2 product:1414 num_ports:1 port:1 path:usb-0000:00:04.0-2
root@arm:/home/debian/Desktop#
```

USB serial port lists of RS-485

Use a serial port communication program such as Minicom or GtKTerm to setup the RS-485 ports using the serial port setup configuration.

For example, install the Minicom program with the following command, and then open a Minicom terminal window for setting up the RS-485 port 1.

```
$ sudo apt-get install minicom
$ sudo minicom -s
```

Select the "Serial port setup" option to configure the serial port, and then set the following settings provided in the screenshot below.

```
+-----+
| A -   Serial Device       : /dev/ttyUSB14 |
| 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 RS-485

When the setting is completed, copy the "setrs485" file to ARTiGO A600 system to enable the RS-485 function. The **setrs485** file can be found in the EVK folder of the BSP.

Open a second terminal window, and then enter the following command to enable the RS-485 port 1.

```
$ sudo ./setrs485 /dev/ttyUSB14
```

Connect the host machine to RS-485 port 1 on the ARTiGO A600 using the RS-485 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. Afterwards, the host machine and the ARTiGO A600 system 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).

6. SPI ROM Backup and Recovery

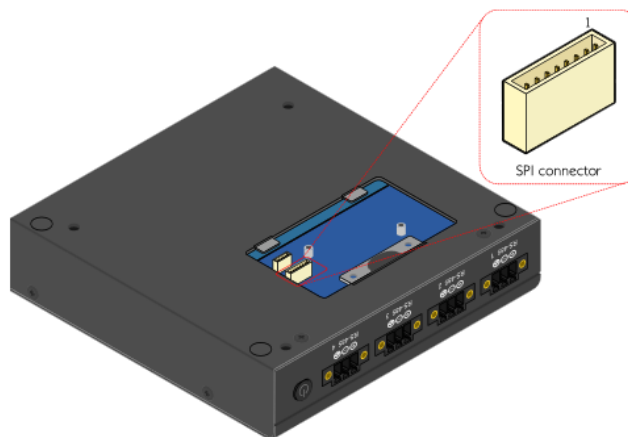
On the ARTiGO A600 system, the SPI Flash ROM contains the U-Boot bootloader and its settings. If the ROM contents become corrupted due to a failure during a firmware update, or are flashed with a non-functional U-Boot binary during development, the board may become non-bootable. In that case, you will 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 ARTiGO A600 system.

6.1. Hardware Information

To backup or reflash the SPI ROM, a SPI Flash programmer is required. Dediprog SF100² 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 ARTiGO A600.

You will need to make your own SPI connector cable to connect the Flash programmer to the ARTiGO A600. The SPI header (marked as SPI) is an ACES³ 87213-0800G connector, which is compatible with the ACES50233-008H0H0-001 housing and 87214-series cramp-on wafer-to-board terminals. The onboard SPI connector and pinouts are shown below.



SPI connector diagram

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

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

SF100 and SPI pin definition table

² See the Dediprog SF100 product page at <http://www.dediprog.com/pd/spi-flash-solution/SF100>

³ 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, flashrom⁴ 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 ARTiGO A600 system is powered off. Connect your Flash programmer and read out the contents of the SPI ROM into a backup file.

For the DediprogSF100, 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 ARTiGO A600 BSP/EVK. The file name of the default image is **SF_BOOTROM.bin**, which can be found in the **bspinst** folder inside the **ARTiGO-A600_Linux_EVK_v1.2.1.zip** file.

First, make sure the ARTiGO A600 system 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

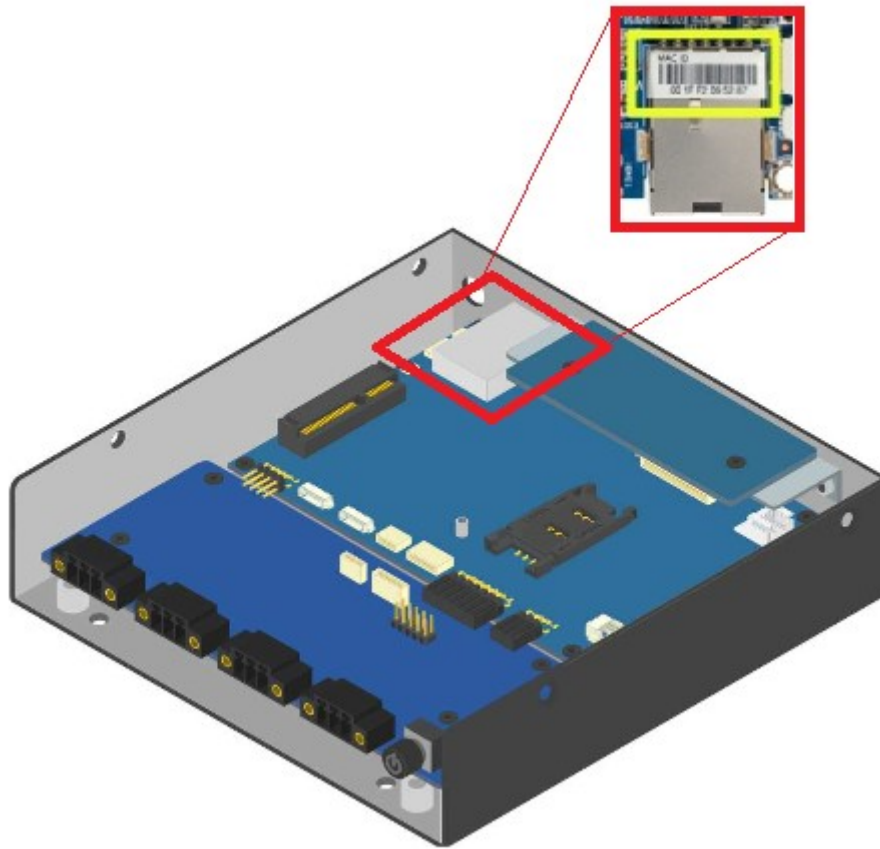
If the image you flashed is the default Flash ROM image from VIA, you will need to reset the MAC address in the U-Boot settings (also stored on the SPI ROM), as the default image will overwrite those settings.

Connect the ARTiGO A600 and the host machine through the debug COM1 cable connector. Update the U-Boot parameters as shown below, replacing xx:xx:xx:xx:xx:xx with the correct MAC address of your ARTiGO A600 system.

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

⁴ See the flashrom website at <https://www.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