Quax

Ultra-Precision Current Generator 20A

Software Reference Manual

This document describes the functionalities of the control board of the generator, based on Aria G25 SMD Module. On the board is installed an embedded Linux system, generated with Buildroot tool. It needs about 16 seconds to boot.

To prevent micro SD corruption when power off the board, the files system is mounted in read only mode.

To communicate with the board the current generator server use the WebSocket protocol (RFC 6455) on port 4444. A WebSocket library for LabView is provided with a simple example to communicate with the board over Ethernet, at link: http://www.pd.infn.it/~caste/pub/WebSockets.zip. On the board is running a Web server (lighttpd) with a simply web page to control and monitor the generator, also a Secure Shell server (sshd) is running on the system.

Getting Started

Ethernet Configuration:

By default, the Ethernet is configured with DHCP enabled, the configuration can be changed by touchscreen display or using console debug connector (mini USB connector, USB to RS232. FTDI FT230X) and edit file /etc/network/interfaces. The files system is in read only mode, use command /usr/sbin/rw to switch from read only to read write mode and /usr/sbin/ro to switch back.

WARNING the Ethernet MAC address is generated with microSD serial number.

USB device connector:

The board can be used directly connected to PC via USB device connector (USB typeB), it provides a USB Ethernet Gadget configured with static IP 10.42.0.10, configure the PC side with the same subnet address, for example 10.42.0.1. Configuration can be changed by touchscreen display or can be changed by edit file /etc/network/interfaces, logging using console debug connector. Warning the Ethernet Gadget at boot generate random MAC address, so on the linux PC (for example Ubuntu with Network Manager) the network must be reconfigured or you must create a special UDEV rule to automatically configure it. This problem is not present on Windows 7 OS (see Appendix A for installation).

Below a simple script to create udev rule for usb0 on linux PC, run the scrip only one time and with root privilege.

```
#!/bin/sh
MAC=02:11:22:33:44:55
echo "[keyfile]" >>/etc/NetworkManager/NetworkManager.conf
echo "unmanaged-devices=mac:$MAC" >>/etc/NetworkManager/NetworkManager.conf
echo "#!/bin/sh" >/etc/udev/usb0.sh
echo "/sbin/ifconfig usb0 down" >>/etc/udev/usb0.sh
echo "/sbin/ifconfig usb0 hw ether $MAC" >>/etc/udev/usb0.sh
echo "/sbin/ifconfig usb0 10.42.0.1" >>/etc/udev/usb0.sh
echo "/sbin/ifconfig usb0 up" >>/etc/udev/usb0.sh
echo 'ATTRS{idVendor}=="0525" ATTRS{idProduct}=="a4a2", RUN+="/bin/sh /etc/udev/usb0.sh"'
>>/etc/udev/rules.d/70-persistent-net.rules
```

<u>The commands accepted by server are</u>:

Status command: return the status.

Syntax: **Status?**

Returns: the status in JSON format, example:

```
{"Current":0.500057,"SetPoint":0.500057,"SlewRate":1.000000,
"Time":3848.522252,"Tpid":44.598568,"Tgen":30.713671,"Tpwr":36.000000,"Ipwr":5
.450406,"Vchg":0.024611,"Vnoise":0.004376,"Vpkpk":0.031118,"Igen":0.478216,"Ip
id":5.928622,"Vpwr":7.480000,"DAC":1620,"Ilim":7.500057,"Tbrd":33.464718}
```

Current: is the output current.
SetPoint: the ramp current set point.
SlewRate: the ramp slew rate.
Time: is the time in seconds from processor boot up.
Tpid: is the shunt temperature.
Tgen: is the temperature measured on the dissipator for the current generator power transistors .
Tpwr: is the temperature of the switching power supply.
Ipwr: is the output current of the switching power supply.
Vchg: is the voltage measured on the output of the current generator.
Vnoise: is the rms noise measured on the output of the current generator (2KSa/s 0.5Hz-1KHz bandwidth).
Vpkpk: is the peak to peak noise measured on the output of the generator (2KSa/s 0.5Hz-1KHz bandwidth).
Igen: is an estimated current (Ipwr-Ipid) of the output current.
Ipid: is the current used for shunt thermostatation.
Vpwr: is the switching power supply output voltage.
DAC: is the dac value set on the current generator.
Ilim: is the current limit set on the switching power supply.
Tbrd: is the temperature of the current generator board.

Version command: return the software version.

Syntax: **Version?**

Returns: **V1.0-13-g577351b**

Set Power command: set power supply on/off. When power off use StatusSetPoint command to determine the end of ramp down of the current.

Syntax:	**Set:Power <0/1>**
Returns:	**OK**

Set Point command: set the output current, the current is moved with a ramp of slew-rate value, use StatusSetPoint command to determine the end of ramp.

Syntax:	**Set:point <value>,<slewrate>**
Argument:	Current value from 0.000 to 20.000 resolution 1mA.
	SlewRate value from 0.01 to 1.000 A/s
Returns:	**OK**

Status Set Point command: return the status of ramp to move the current.

Syntax:	**StatusSetPoint?**
Returns:	**OK or BUSY**

Abort command: stop ramp to the current value.

Syntax:	**Set:abort**
Returns:	**OK**

Increment command: increment the current by 1,10, 100mA or 1 DAC bit. The command moves a set-point so the current reaches the set-point with the slew-rate set on the generator (see status command for slew-rate value)

Syntax:	**Set:inc <mode>**
Argument:	the mode: 0=1mA 1=10mA 2=100mA 3=1bit.
Returns:	**OK**

Decrement command: decrement the current by 1,10, 100mA or 1 DAC bit. The command moves a set-point so the current reaches the set-point with the slew-rate set on the generator (see status command for slew-rate value)

Syntax:	**Set:dec <mode>**
Argument:	the mode: 0=1mA 1=10mA 2=100mA 3=1bit.
Returns:	**OK**

Authenticate command: This command is used to obtain the information necessary for the authentication handshake. The nonce value expire after 60 seconds.

Syntax: **Authenticate?**

Returns:
```
{realm: "authorized only", nonce: "bb7a2bc19db7495606c57750f90ba775"}
```

Authorization command: This command in conjunction with Authenticate command must be used to enable the connection to accept SET commands. User and password can be added or changed using console with linux command htdigest. Example:

```
~$ htdigest /etc/wspasswd “authorized only” operator
```

/etc/wspasswd is the password file, “authorized only” is realm string obtained by **Authenticate** command and operator is the username. By default users can authenticate with user “operator” and password “quax”.

Syntax: **Authorization:<user>:<realm>:<nonce>:<response>**

Arguments:
<user> specified the username.
<realm> specified the realm string obtained by Authenticate command.
<nonce> specified the nonce string obtained by Authenticate command.
<response> must be calculate using MD5 hash by the following string:
ha1=MD5(“<user>:<realm>:<password>”);
response=MD5(“<ha1>:<nonce>”);

Returns: OK

Records command: return the history of the monitoring data.

Syntax: **Records:Range? <time>,<maxsize>**

Arguments:
<time> specified the time in seconds. First time use time=0 to obtain the first data available, then use the last time+0.1, received from a previous call, to obtain successive data.
<maxsize> is the maximum number of record to return and must be less or equals than 64

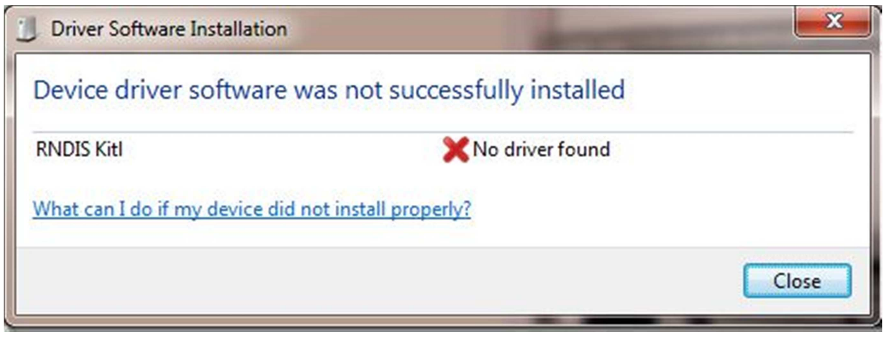
Returns: return the stored data in JSON format, example:
```
{"Tpid" : [[15.5,24.31],[16.5,24.31],[17.5,24.31],[18.5,24.31],[19.5,24.31]],
"Ipid" : [[15.5,23.94],[16.5,23.94],[17.5,23.94],[18.5,23.94],[19.5,23.94]],
"Tbrd" : [[15.5,23.94],[16.5,23.94],[17.5,24.00],[18.5,24.00],[19.5,24.00]],
"maxrecord": 1024}
```

Returns Error message Syntax: **ERROR:<number>,<message>**

Appendix A

Windows 7 RNDIS driver installation

1. After the device is connected to the PC, OS will automatically search for the RNDIS driver. After it fails to find the driver, the following message will be shown.



Driver Software Installation

Device driver software was not successfully installed

RNDIS Kitl ❌ No driver found

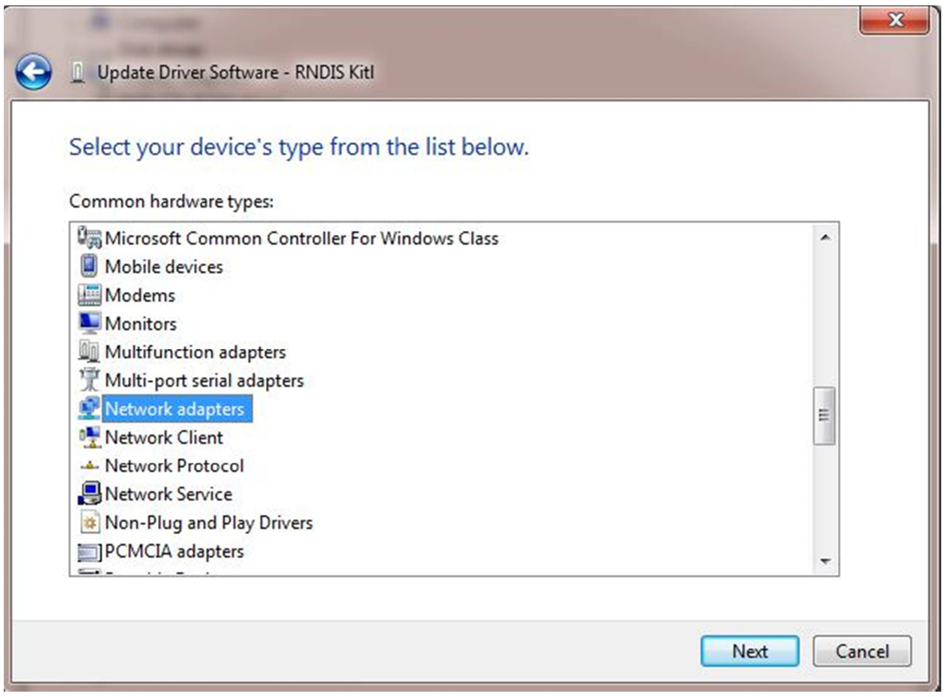
What can I do if my device did not install properly?

2. Right click on **Computer** and select **Manage**. From **System Tools**, select **Device Manager**. It will show a list of devices currently connected with the development PC. In the list, RNDIS Kitl can be seen with an exclamation mark implying that driver has not been installed.

3. Right click on it and select **Update Driver Software...** When prompted to choose how to search for device driver software, choose **Browse my computer for driver software**.

4. **Browse for driver software on your computer** will come up. Select **Let me pick from a list of device drivers on my computer.**

5. A window will come up asking to select the device type. Select Network adapters, as RNDIS emulates a network connection.



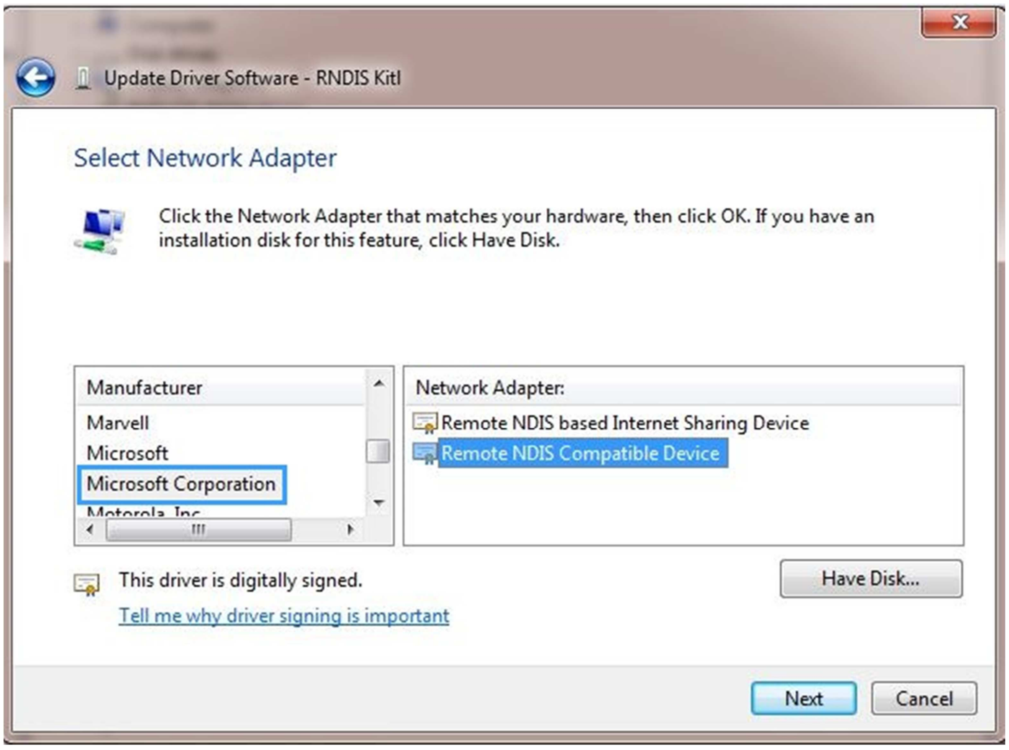
Update Driver Software - RNDIS Kitl

Select your device's type from the list below.

Common hardware types:

- Microsoft Common Controller For Windows Class
- Mobile devices
- Modems
- Monitors
- Multifunction adapters
- Multi-port serial adapters
- Network adapters
- Network Client
- Network Protocol
- Network Service
- Non-Plug and Play Drivers
- PCMCIA adapters

6. In the Select Network Adapter window, select Microsoft Corporation from the **Manufacturer** list. Under the list of **Network Adapter:**, select **Remote NDIS compatible device**.


Update Driver Software - RNDIS Kitl

Select Network Adapter

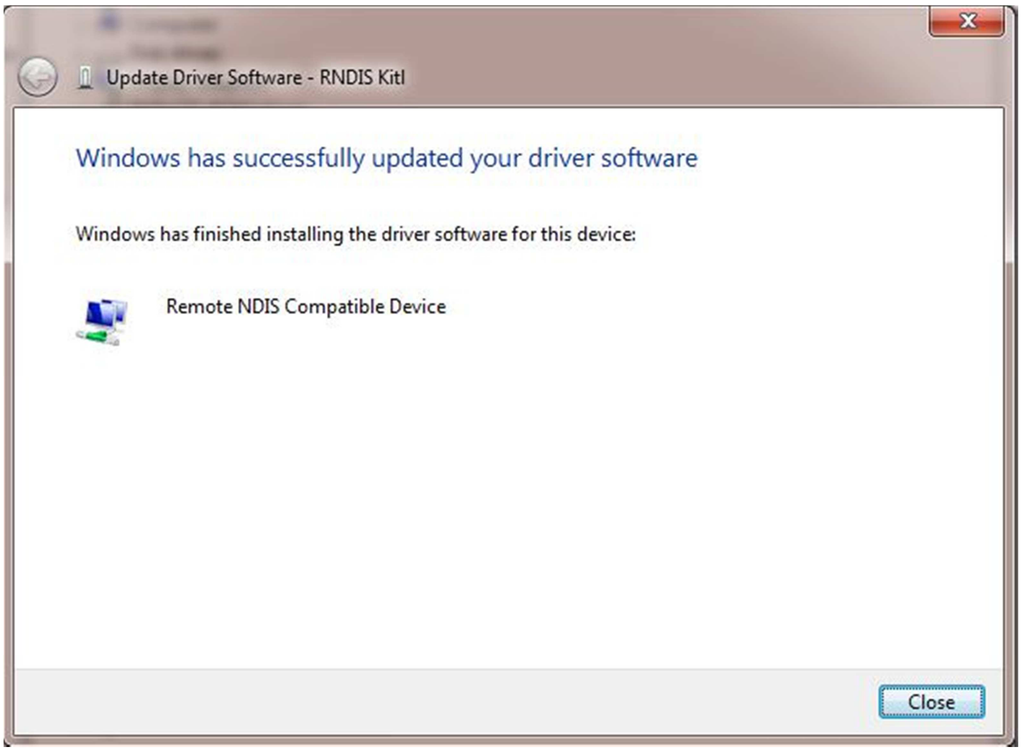
Click the Network Adapter that matches your hardware, then click OK. If you have an installation disk for this feature, click Have Disk.

Manufacturer
Marvell
Microsoft
Microsoft Corporation
Motorola, Inc.

Network Adapter:
Remote NDIS based Internet Sharing Device
Remote NDIS Compatible Device

This driver is digitally signed.
Tell me why driver signing is important

7. The RNDIS Kitl device is now installed and ready for use.


Update Driver Software - RNDIS Kitl

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Remote NDIS Compatible Device