DNI Smart Grid Data Collector

SGDC-D22 User Manual

Rev. 1.9

2017/01/19

Notice:

1. Delta Networks Inc. reserves the right to change specifications detailed in this document at any time without notice, and assumes no responsibility for any errors within this document.
2. This document contains proprietary technical information in which is the property of the Delta Networks Inc. and shall not be disclosed to others in whole or in part, reproduced, copied, or used as the basis for design, manufacturing, or sale of apparatus without written permission of Delta Networks Inc.

DNI

Table of Contents

Revision History

Version	Date	Author	Description
1.0	2012/5/9	DNI SG Team	• Initial Release.
1.1	2012/5/23	DNI SG Team	• Add Linux Kernel Upgrade & Backup section.
1.2	2012/6/4	DNI SG Team	• Add & modify sections below - Communication Settings - Daemons & Utilities - I/O Control
1.3	2012/10/12	DNI SG Team	• Modify some configurations.
1.4	2013/11/29	DNI SG Team	• Add SIM & SD card installation.
1.5	2014/4/29	DNI SG Team	• Add ZigBee feature for SGDC-D23.
1.6	2014/11/4	DNI SG Team	• Add SGDC-D24.
1.7	2016/9/2	Jacky Lai	• Update to standard format • Remove 2G(GPRS) for certification
1.8	2017/1/5	Jacky Lai	• Added FCC description in 1.4
1.9	2017/1/19	Jacky Lai	• Added RF Exposure, 2.1091description in 1.4

1. Introduction

1.1. General Description

DNI data collector SGDC-D22 is a 3G version ARM9 based embedded system with 3G for WAN communication and Ethernet port, RS-232 and RS-422/485 interfaces for LAN communication as well to collect data from devices via LAN communication and forward to data center via WAN communication. The communication capability can fulfill the requirements in smart metering and distributed energy monitoring applications as well as the sensor network and internet of things applications.

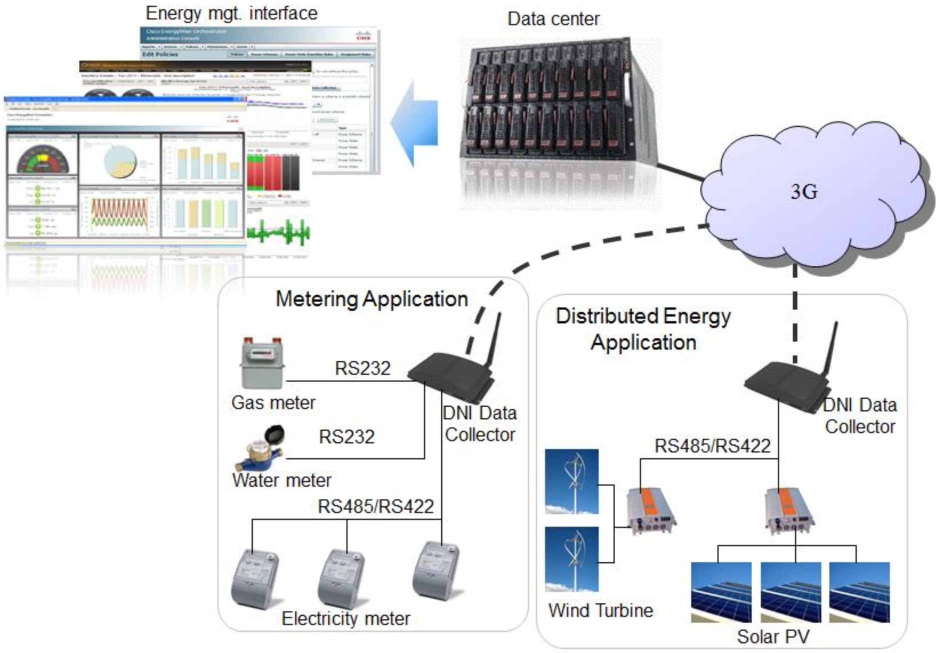


Figure 1, DNI data collector applications.

1.2. Target Applications

DNI data collector is an embedded system designed for smart grid applications as well as the IoT related applications.

1.3. Product Information

Model Name	Description
SGDC-D22	Data collector with 3G communication board.

1.4. Regulation declaration

The changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. To comply with the FCC RF exposure compliance requirements, this device and its antenna must not be co-located or operating to conjunction with any other antenna or transmitter.

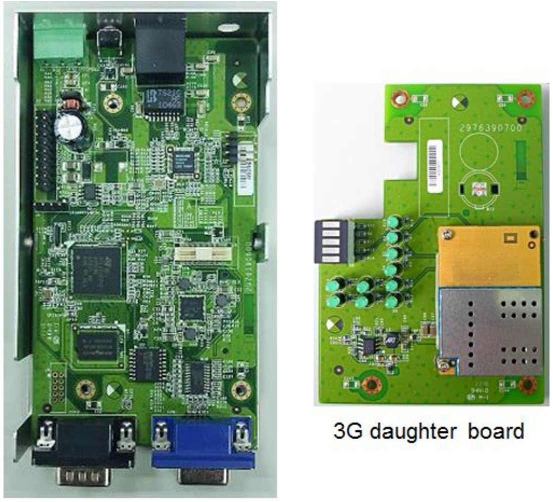
Note: This equipment has been tested and found to comply with the limits for a Class B Digital Device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction, may cause harmful interference to radio communication. However, there is no grantee that interference will not occur in a particular installation. If this equipment dose cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
--Reorient or relocate the receiving antenna.
--Increase the separation between the equipment and receiver.
--Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
--Consult the dealer or an experienced radio/TV technician for help.

This device should have at least 20 cm separation distance to persons.

2. Platform Description

2.1. Hardware Information

DNI data collector incorporates the following hardware components as shown in Figure 2.



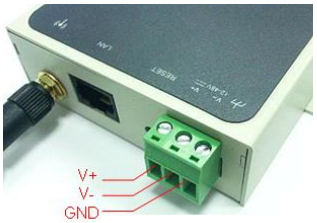
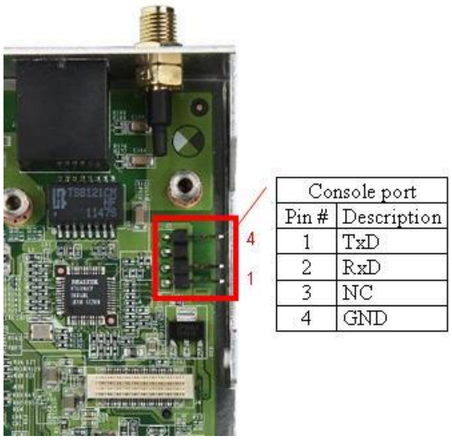
3G daughter board

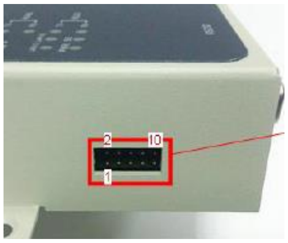
Main board

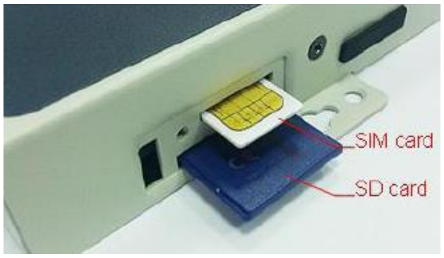
Figure 2, DNI data collector hardware components.

Feature	Description
Main Board	With CPU, memory and major interfaces of data collector.
3G Daughter Board	With 3G module and LED indicators.
CPU	ST SPEAr 320S
Flash	32MB (16MB × 2)
DDR-II	64MB
3G Module[1]	3G: (model name: SGDC-D22) • Five band: UMTS/HSPA+ 1900MHz • HSPA 3GPP Release 6, 7 • UMTS 3GPP release 4 • Output power: - Class 3 (+24dBm +1/-3dB) for UMTS 1900,WCDMA FDD BdII

The interfaces of data collector are illustrated as following table.

<table>
<tr><th>Interface</th><th>Description</th><th>Note</th></tr>
<tr><td>Power</td><td>A 3-pin terminal block to connect 12-48VDC V+, V– and GND.</td><td>
</td></tr>
<tr><td>Reset Button</td><td>Hardware reset button. Support back to factory default setting function.</td><td></td></tr>
<tr><td>Ethernet Port</td><td>RJ45 port for Ethernet connection.</td><td></td></tr>
<tr><td>Antenna</td><td>External antenna with SMA connector.</td><td></td></tr>
<tr><td>Console Port</td><td>Local port for direct connecting to concentrator (RJ45 connector).</td><td>

<table>
<tr><th colspan="2">Console port</th></tr>
<tr><th>Pin #</th><th>Description</th></tr>
<tr><td>1</td><td>TxD</td></tr>
<tr><td>2</td><td>RxD</td></tr>
<tr><td>3</td><td>NC</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table></td></tr>
<tr><td>Wall Mount</td><td>2 L shape parts for wall mount installation.</td><td></td></tr>
<tr><td>RS-232</td><td>Male DB9 connector for RS-232 connection.</td><td>PIN 2: RXD
PIN 3: TXD
PIN 5: GND</td></tr>
<tr><td>RS-422/485</td><td>Female DB9 connector for RS-422/485 connection. Support switching RS-485 or RS-422 modes by software.</td><td>Please find the pin define in separate table.</td></tr>
<tr><td>GPIO</td><td>2x5-pin digital I/O for applications.</td><td>PIN 1: GND
PIN 2: GND
PIN 3: DI3
PIN 4: DO3
PIN 5: DI2
PIN 6: DO2
PIN 7: DI1
PIN 8: DO1
PIN 9: DI0
PIN 10: DO0

</td></tr>
</table>

Interface	Description	Note
SIM Card & SD Card Slots[1]		

RS-485 pin define:

Serial Port	Device Name	Software Switch Command	Hardware PIN Definition
RS-485 (2-wire)	/dev/ttyM1	setport 1	PIN 1: GND PIN 2: DATA(A)– PIN 3: DATA(B)+
RS-422 (4-wire)	/dev/ttyM1	setport 0	PIN 1: GND PIN 2: RX– PIN 3: RX+ PIN 4: TX+ PIN 5: TX–

LED display:

Feature	Description	Note
Power	Power indicator	Turn on when power on.
Ready	Get the system ready information and turn on the ready LED when data collector connect to data center and get response from communication server.	Turn on when connect to data center.
RS-232	TxD × 1, RxD × 1	Blink when communication.
RS-485	TxD × 1, RxD × 1	Blink when communication.
3G	3G connection indicator.	Turn on when 3G connected.
3G Signal Strength[1]	5 levels for ZigBee signal strength indicators.	
LAN10 / LAN100	2 for 10/100Mbps speed indicators	Blink when communication.

2.2. Software Information

Software Package	STLinux 2.3 or above
Operation System	Linux kernel 2.6.37 or above

Utilities	Busybox 1.19.3
File System type	Jffs2
Toolchain	Gcc v4.2.4, Glibc v2.6.1, GDB v6.3

3. Tool Chain Installation

3.1. Environment

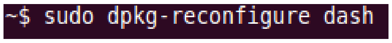
- Host OS: Ubuntu 10.04
- STLinux image file: STLinux-2.3-spear-20091209.iso
  http://ftp.stlinux.com/pub/stlinux/2.3/iso/

3.2. Installing STLinux on Ubuntu

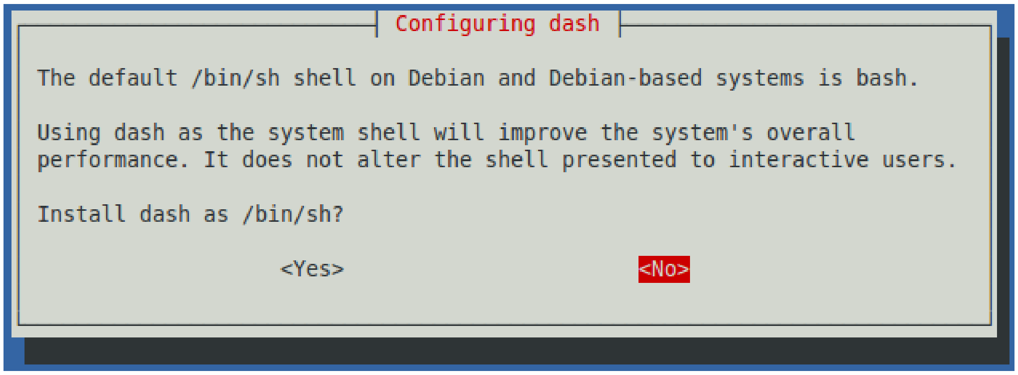
Please refer to the installation guide at http://www.stlinux.com/faq?q=node/361.

Installation steps:

- Make /bin/sh bash

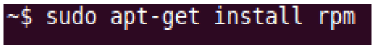


```
~$ sudo dpkg-reconfigure dash
```

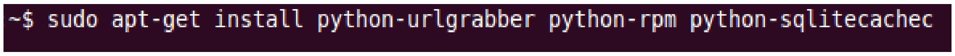


```
┤ Configuring dash ├

The default /bin/sh shell on Debian and Debian-based systems is bash.

Using dash as the system shell will improve the system's overall
performance. It does not alter the shell presented to interactive users.

Install dash as /bin/sh?

                  <Yes>                        <No>
```

- Installing RPM on Ubuntu

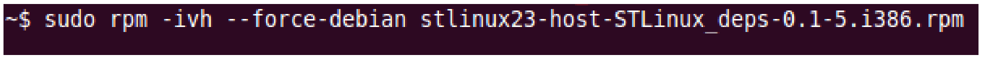


```
~$ sudo apt-get install rpm
```



```
~$ sudo apt-get install python-urlgrabber python-rpm python-sqlitecachec
```

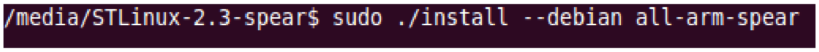
- Populating the RPM databases with "Provides"
  Download “STLinux_deps” rpm file at
  http://www.stlinux.com/sites/default/files/stlinux23-host-STLinux_deps-0.1-5.i386.rpm then install it.



```
~$ sudo rpm -ivh --force-debian stlinux23-host-STLinux_deps-0.1-5.i386.rpm
```

- Install STLinux

  Mount image file “STLinux-2.3-spear-20091209.iso” then install it.

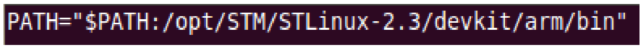


```
/media/STLinux-2.3-spear$ sudo ./install --debian all-arm-spear
```

  After installation completed, STLinux package would be place at /opt/STM/STLinux-2.3/.

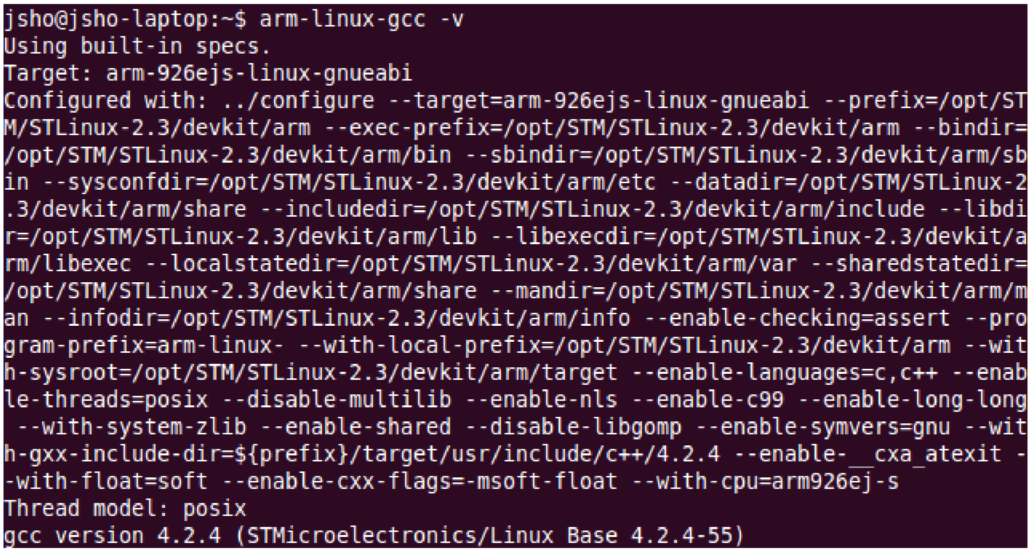
3.3. Add Tool Chain Path

- Add tool chain path at ~/.bashrc then re-login.



```
PATH="$PATH:/opt/STM/STLinux-2.3/devkit/arm/bin"
```

- Test tool chain



```
jsho@jsho-laptop:~$ arm-linux-gcc -v
Using built-in specs.
Target: arm-926ejs-linux-gnueabi
Configured with: ../configure --target=arm-926ejs-linux-gnueabi --prefix=/opt/ST
M/STLinux-2.3/devkit/arm --exec-prefix=/opt/STM/STLinux-2.3/devkit/arm --bindir=
/opt/STM/STLinux-2.3/devkit/arm/bin --sbindir=/opt/STM/STLinux-2.3/devkit/arm/sb
in --sysconfdir=/opt/STM/STLinux-2.3/devkit/arm/etc --datadir=/opt/STM/STLinux-2
.3/devkit/arm/share --includedir=/opt/STM/STLinux-2.3/devkit/arm/include --libdi
r=/opt/STM/STLinux-2.3/devkit/arm/lib --libexecdir=/opt/STM/STLinux-2.3/devkit/a
rm/libexec --localstatedir=/opt/STM/STLinux-2.3/devkit/arm/var --sharedstatedir=
/opt/STM/STLinux-2.3/devkit/arm/share --mandir=/opt/STM/STLinux-2.3/devkit/arm/m
an --infodir=/opt/STM/STLinux-2.3/devkit/arm/info --enable-checking=assert --pro
gram-prefix=arm-linux- --with-local-prefix=/opt/STM/STLinux-2.3/devkit/arm --wit
h-sysroot=/opt/STM/STLinux-2.3/devkit/arm/target --enable-languages=c,c++ --enab
le-threads=posix --disable-multilib --enable-nls --enable-c99 --enable-long-long
 --with-system-zlib --enable-shared --disable-libgomp --enable-symvers=gnu --wit
h-gxx-include-dir=${prefix}/target/usr/include/c++/4.2.4 --enable-__cxa_atexit -
-with-float=soft --enable-cxx-flags=-msoft-float --with-cpu=arm926ej-s
Thread model: posix
gcc version 4.2.4 (STMicroelectronics/Linux Base 4.2.4-55)
```

4. Getting Started

4.1. Connect to PC

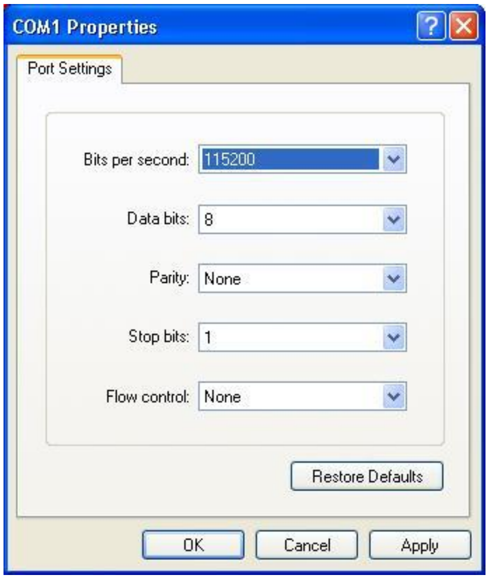
DNI data collector provides two interfaces for PC to login, configure and maintenance. Users can connect to a PC through a serial console port (RS-232) or by using SSH utility over the network connection. This section will describe how to connect DNI data collector to PC through these two interfaces.

4.1.1. Serial Console Port (RS-232)

When using the serial console port to connect, first, make sure the console cable is correctly connected between DNI data collector and a host PC.

Then open a serial port terminal emulator (e.g. Hyper Terminal or PuTTY) and fill the port settings as shown in the following table.

Baud Rate	115200 bps
Parity	None
Data Bits	8
Stop Bit	1
Flow Control	None



Bits per second: 115200

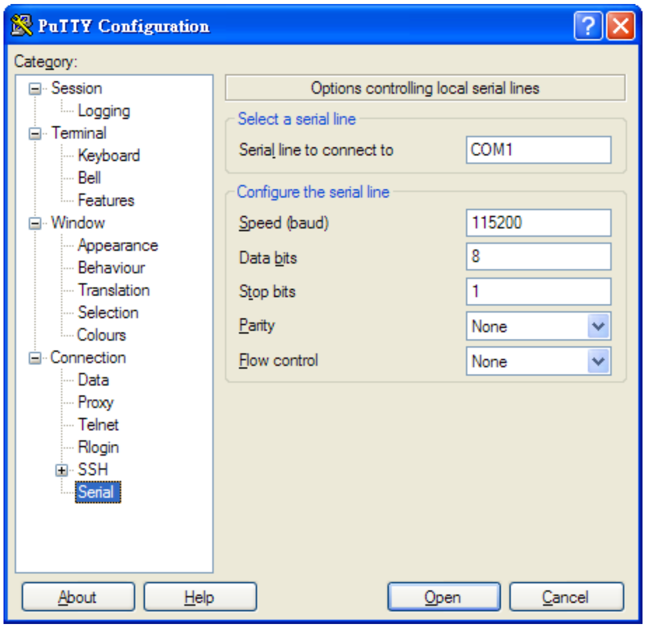
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Figure 3, Hyper Terminal Com Port Properties.



Options controlling local serial lines

Select a serial line

Serial line to connect to: COM1

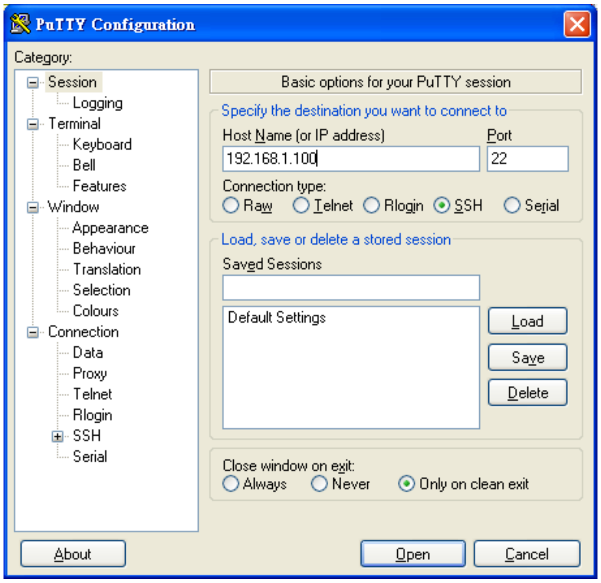
Configure the serial line

Speed (baud)	115200
Data bits	8
Stop bits	1
Parity	None
Flow control	None

Figure 4, PuTTY Configuration for Serial Connection.

4.1.2. SSH Utility (Network)

By default, DNI data collector enables the SSH service to support remote log in. The default network IP of DNI data collector is 192.168.1.100. Please also make sure that the IP used by PC is also in the 192.168.1.x subnet. The network cable can be connected directly between a PC and a DNI data collector.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.100

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
○ Always ○ Never ● Only on clean exit

Figure 5, PuTTY Configuration for SSH Connection.

4.2. SIM Card and SD Card Installation

The SIM card and SD card slots are behind the left side cover gate of the SGDC-D22. Please refer to the following picture and instructions to install SIM card and SD card.

1. Open the cover gate by rotating the screw in anti-clockwise way.
2. Insert the SIM card and SD card into the slot in correct direction. (The unfilled corner is on the right side.)
3. Close the cover gate by rotating the screw in clockwise direction, and be sure the gate is closed when the device is power-on.

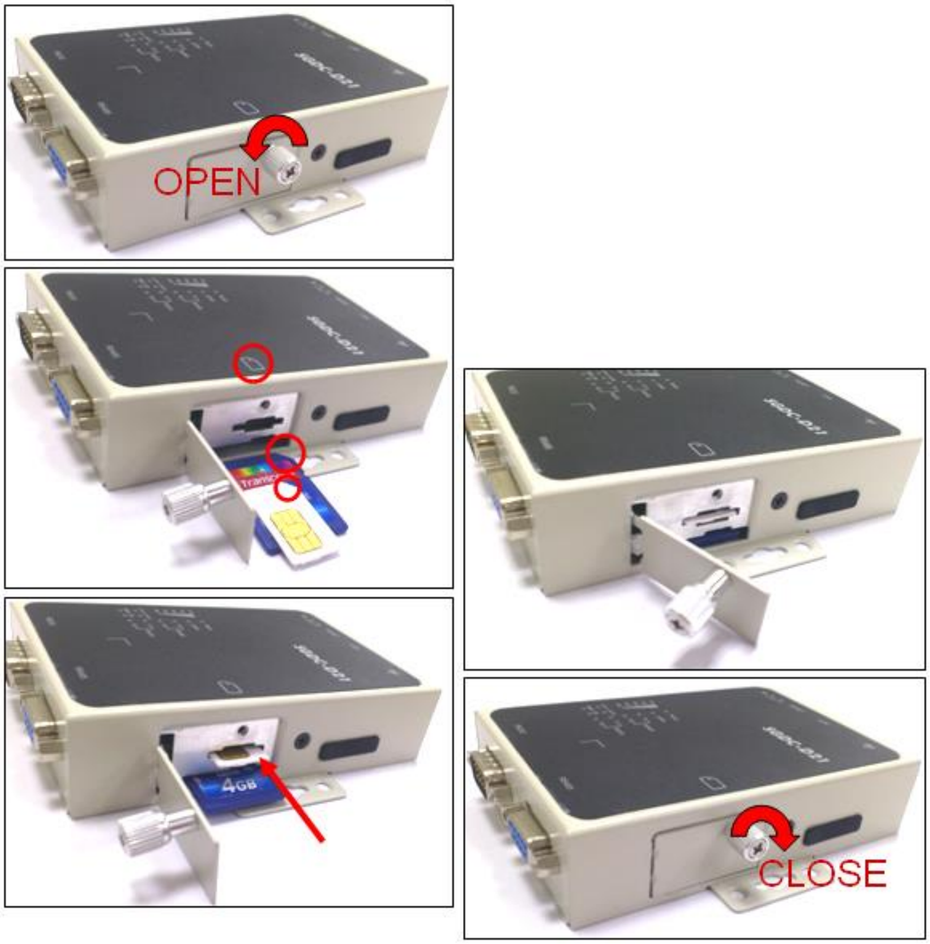
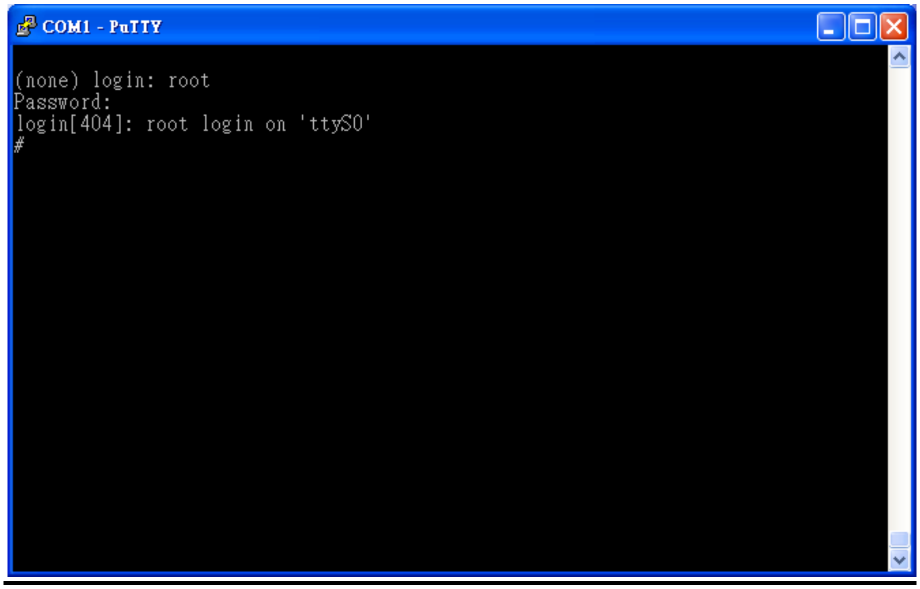


Figure 6, SIM card and SD card installation

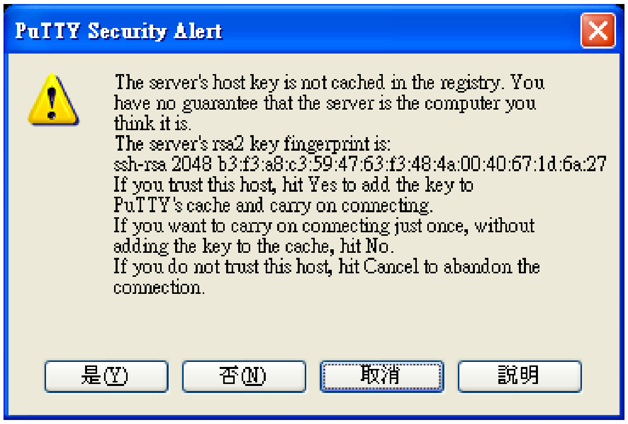
4.3. Login Data Collector

Once the connection (either serial or SSH) is established, a prompt for login will be shown in the window. The default account for log in is root. The default password for root is ‘dnidni’. First time to log in through SSH, some terminal emulator may show alerts that the host key is unknown, please accept it and continue the log in procedure.



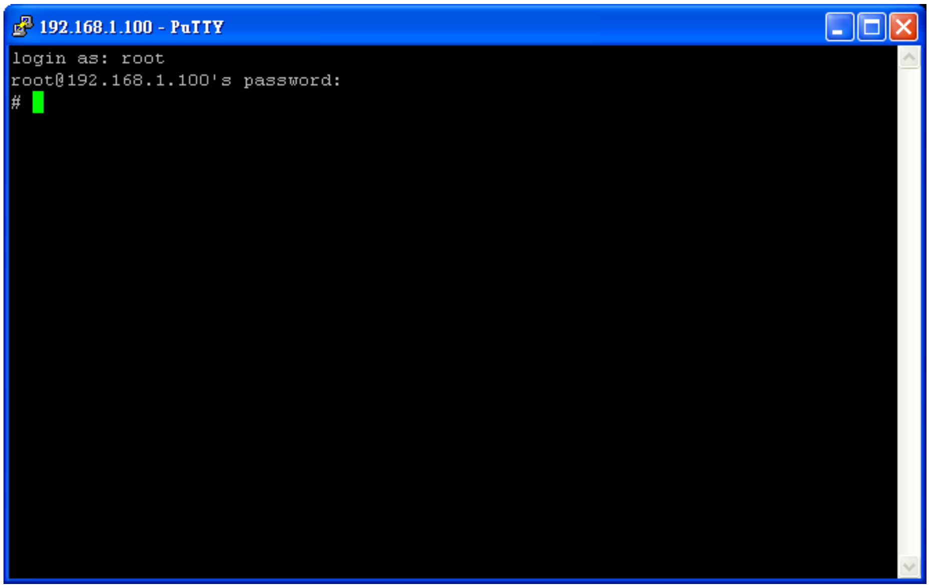
```
(none) login: root
Password:
login[404]: root login on 'ttyS0'
#
```

Figure 7, Data collector Log In Example Output (Serial).



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 b3:f3:a8:c3:59:47:63:f3:48:4a:00:40:67:1d:6a:27
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

Figure 8, Putty Security Alert.



```
login as: root
root@192.168.1.100's password:
# 
```

Figure 9, Data collector Log In Example Output (SSH).

5. The First Program

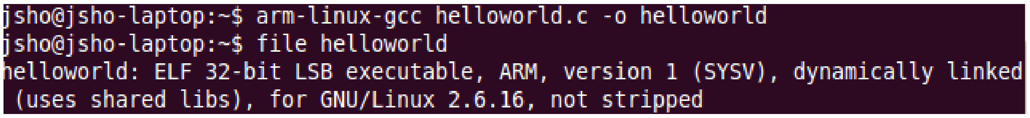
5.1. Compile helloworld.c

Use cross compiler at host to compile the program for data collector.

Use “file” command to check the program after compiling.

```
# arm-linux-gcc helloworld.c -o helloworld
# file helloworld
```

Result:



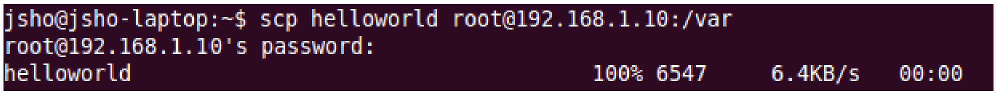
```
jsho@jsho-laptop:~$ arm-linux-gcc helloworld.c -o helloworld
jsho@jsho-laptop:~$ file helloworld
helloworld: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked
 (uses shared libs), for GNU/Linux 2.6.16, not stripped
```

5.2. Send Files to Data Collector

Use “scp” command at host to send the file to data collector.

```
# scp helloworld root@192.168.1.10:/var
```

Result:

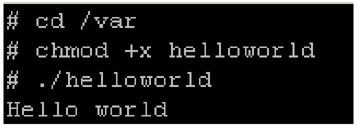


```
jsho@jsho-laptop:~$ scp helloworld root@192.168.1.10:/var
root@192.168.1.10's password:
helloworld                                    100% 6547     6.4KB/s   00:00
```

5.3. Execute Program at Data Collector

Execute the program at the data collector.

Result:



```
# cd /var
# chmod +x helloworld
# ./helloworld
Hello world
```

6. Communication Settings

6.1. Static IP on eth0

To set the static IP for DCU in eth0, please edit /etc/network/interfaces

```
# vi /etc/network/interfaces
```

```
auto eth0
iface eth0 inet static
        address 192.168.1.100
        network 192.168.1.0
        netmask 255.255.255.0
        broadcast 192.168.1.255
        gateway 192.168.1.254
```

And you can apply this configuration by executing the following command:

```
# set-static-ip
```

6.2. DHCPD and DHCP Client

DCU can act as a DHCP server to assign IP to the client in the same network.

Enable DHCPD:

```
# udhcpd
```

Edit the /etc/udhcpd.conf if needed.

```
# vi /etc/udhcpd.conf
```

```
start 192.168.1.20
end 192.168.1.99
interface eth0

opt router 192.168.1.254
option subnet 255.255.255.0
option lease 600 # 10min
opt dns 168.95.1.1 168.95.192.1

#static_lease 00:13:96:03:a7:ed 192.168.1.30
#static_lease 00:13:96:03:a5:f0 192.168.1.31
```

DCU also can act as a DHCP client to get IP configuration from DHCP server.

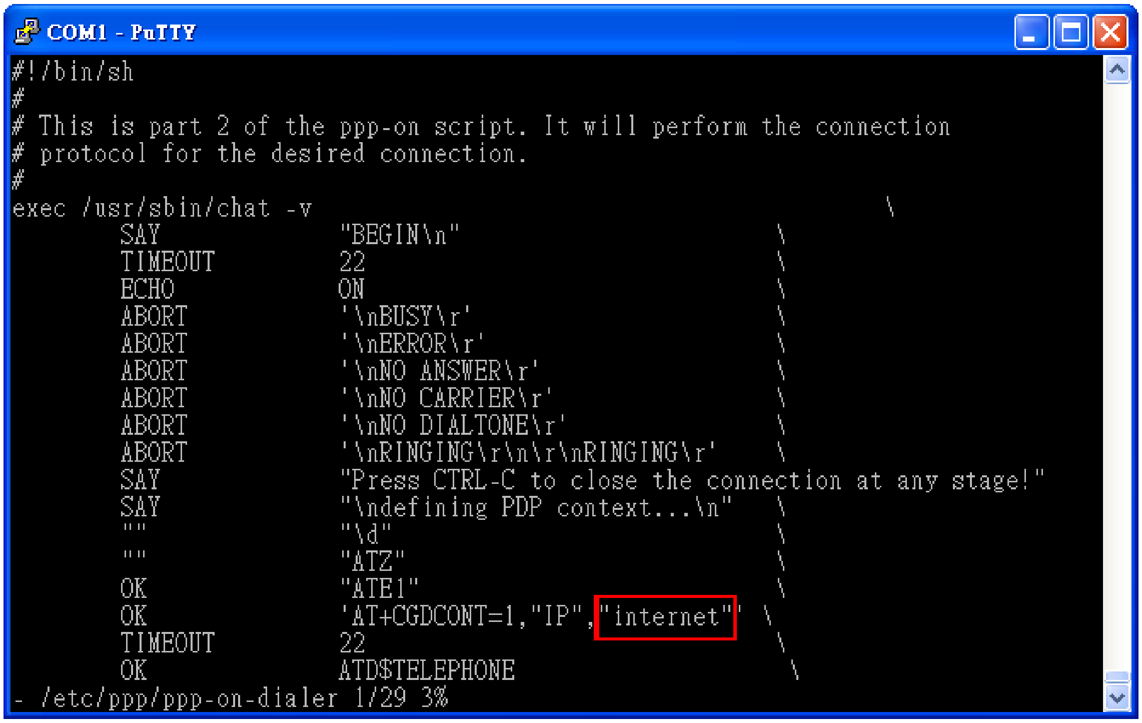
```
# dhcpip
```

To act as dhcp server or client on booting, it can be configured in /etc/rc.d/rc.local.

6.3. 3G

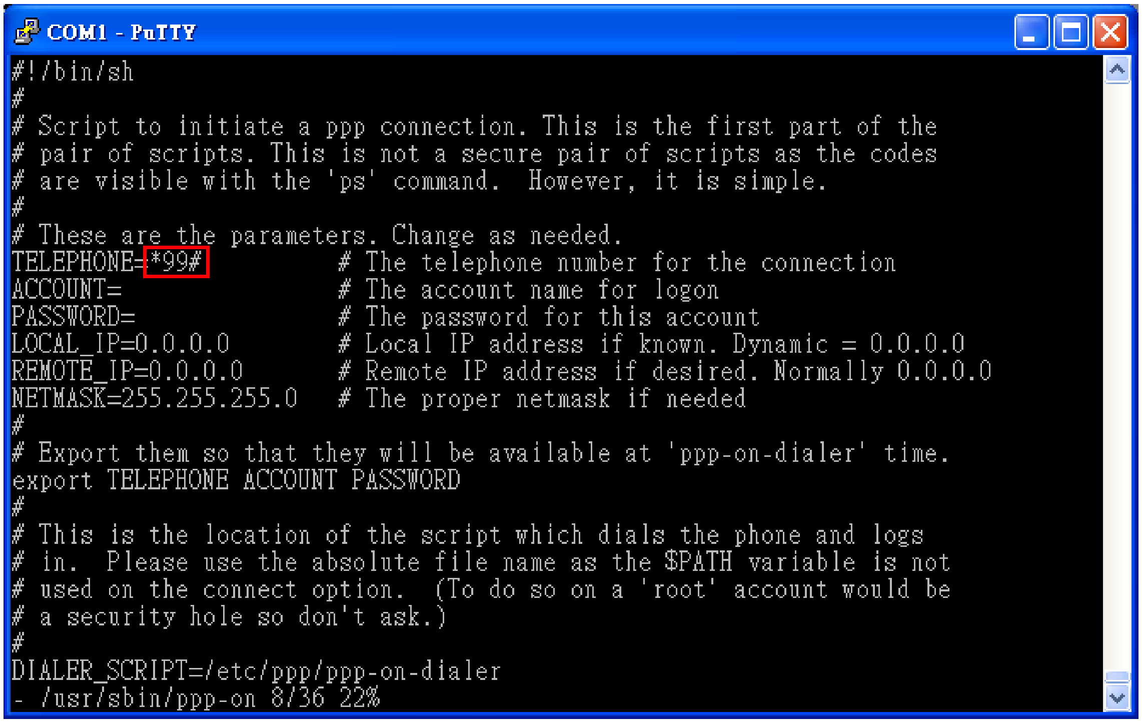
Before using 3G communication, please insert a SIM card into the slot on the 3G module. The position of the SIM slot and the direction of SIM card please refer to the section 4.2.

DNI data collector uses pppd to handle the 3G connection. 3G service providers have their own setting such as APN and dial number. Please contact the provider to get the correct values. The APN value is defined in /etc/ppp/ppp-on-dialer. And the dial number is in /usr/sbin/3g-connect.



```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v                                          \
        SAY             "BEGIN\n"                       \
        TIMEOUT         22                              \
        ECHO            ON                              \
        ABORT           '\nBUSY\r'                      \
        ABORT           '\nERROR\r'                     \
        ABORT           '\nNO ANSWER\r'                 \
        ABORT           '\nNO CARRIER\r'                \
        ABORT           '\nNO DIALTONE\r'               \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'    \
        SAY             "Press CTRL-C to close the connection at any stage!"
        SAY             "\ndefining PDP context...\n"   \
        ""              "\d"                            \
        ""              "ATZ"                           \
        OK              "ATE1"                          \
        OK              'AT+CGDCONT=1,"IP","internet"'  \
        TIMEOUT         22                              \
        OK              ATD$TELEPHONE                    \
- /etc/ppp/ppp-on-dialer 1/29 3%
```

Figure 10, Change APN.



```
#!/bin/sh
#
# Script to initiate a ppp connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
# are visible with the 'ps' command.  However, it is simple.
#
# These are the parameters. Change as needed.
TELEPHONE=*99#          # The telephone number for the connection
ACCOUNT=                # The account name for logon
PASSWORD=               # The password for this account
LOCAL_IP=0.0.0.0        # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP=0.0.0.0       # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.255.0   # The proper netmask if needed
#
# Export them so that they will be available at 'ppp-on-dialer' time.
export TELEPHONE ACCOUNT PASSWORD
#
# This is the location of the script which dials the phone and logs
# in.  Please use the absolute file name as the $PATH variable is not
# used on the connect option.  (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
- /usr/sbin/ppp-on 8/36 22%
```

Figure 11, Change Dial Number.

After confirm the APN and the dial number, use the following command to connect a 3G service.

```
# 3g-connect
```

You also can get RSSI value and reflash the signal strength led status by using the following command.

```
# get-rssi-ber
```

To get imsi code, use the command below.

```
# getimsi
```

Use the command below to disconnect the 3G service.

```
# 3g-disconnect
```

6.4. PPPoE over eth0

To access Internet via the service form your ISP, e.g. ADSL, you can use PPPoE to connect to your ISP. Before using PPPoE, you need to request your ISP to setup an ADSL modem in your house and get some information, e.g. username and password, which is needed on setting your connection.

When ADSL modem is ready to use, please use Ethernet cable to connect DCU and ADSL modem and execute the following command to connect to your ISP.

First, you need to setup the parameters for the PPPoE connection.

```
# pppoe-setup
```

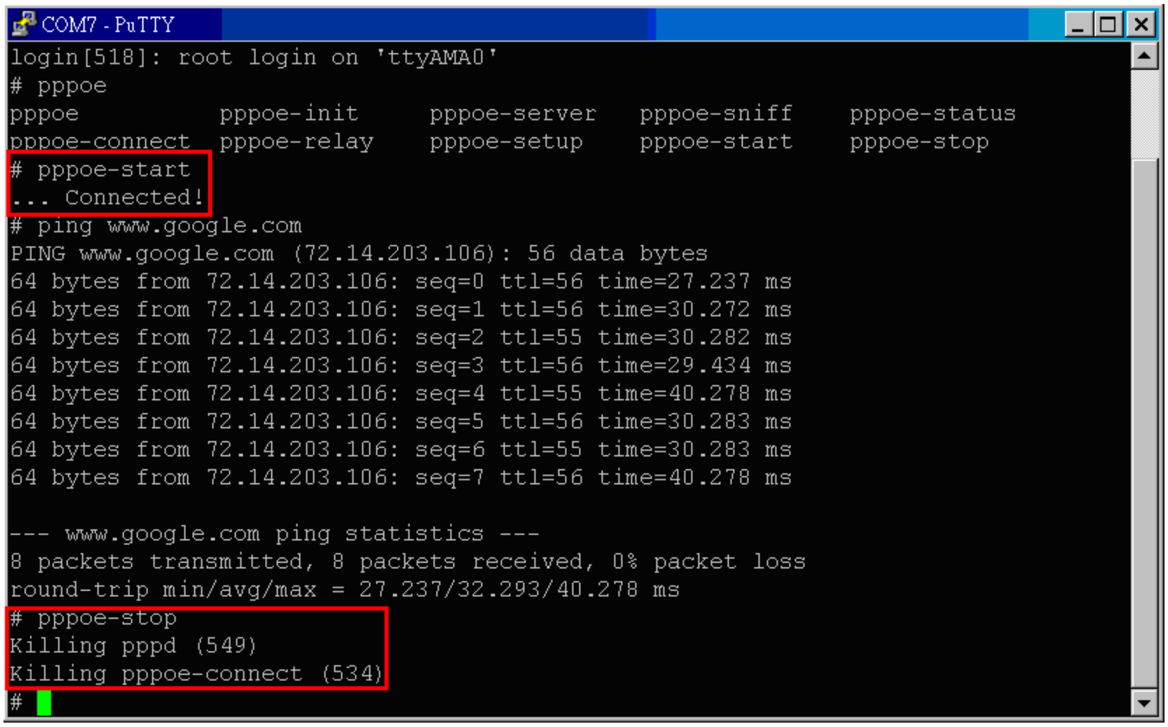
Please answer the following questions:

>>> Enter your PPPoE user name: ***your_username***
>>> Enter the Ethernet interface connected to the DSL modem: (default eth0): ***eth0***
>>> Enter the demand value (default no): ***no***
>>> Enter the DNS information here: ***server*** (Please enter a specific DNS server IP when ISP has provided it to you.)
>>> Please enter your PPPoE password: ***your_password***
>>> Choose a type of firewall (0-2): ***0***
>>> Accept these settings and adjust configuration files (y/n)? ***y***

After setting up parameters, use the following command to conect to the remote server:

```
# pppoe-start
```

When the connection is built up, there will be a message "… Connected! " on the console.



```
login[518]: root login on 'ttyAMA0'
# pppoe
pppoe          pppoe-init     pppoe-server   pppoe-sniff    pppoe-status
pppoe-connect  pppoe-relay    pppoe-setup    pppoe-start    pppoe-stop
# pppoe-start
... Connected!
# ping www.google.com
PING www.google.com (72.14.203.106): 56 data bytes
64 bytes from 72.14.203.106: seq=0 ttl=56 time=27.237 ms
64 bytes from 72.14.203.106: seq=1 ttl=56 time=30.272 ms
64 bytes from 72.14.203.106: seq=2 ttl=55 time=30.282 ms
64 bytes from 72.14.203.106: seq=3 ttl=56 time=29.434 ms
64 bytes from 72.14.203.106: seq=4 ttl=55 time=40.278 ms
64 bytes from 72.14.203.106: seq=5 ttl=56 time=30.283 ms
64 bytes from 72.14.203.106: seq=6 ttl=55 time=30.283 ms
64 bytes from 72.14.203.106: seq=7 ttl=56 time=40.278 ms

--- www.google.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 27.237/32.293/40.278 ms
# pppoe-stop
Killing pppd (549)
Killing pppoe-connect (534)
#
```

Figure 12, Connect and disconnect with PPPoE.

When you want to disconnect, use the following command to disconnect the PPPoE connection.

```
# pppoe-stop
```

7. Daemons & Utilities

7.1. Ramdisk

To mount a virtual disk on the memory, you can execute the following command to create a ramdisk.

Create a directory for ramdisk:

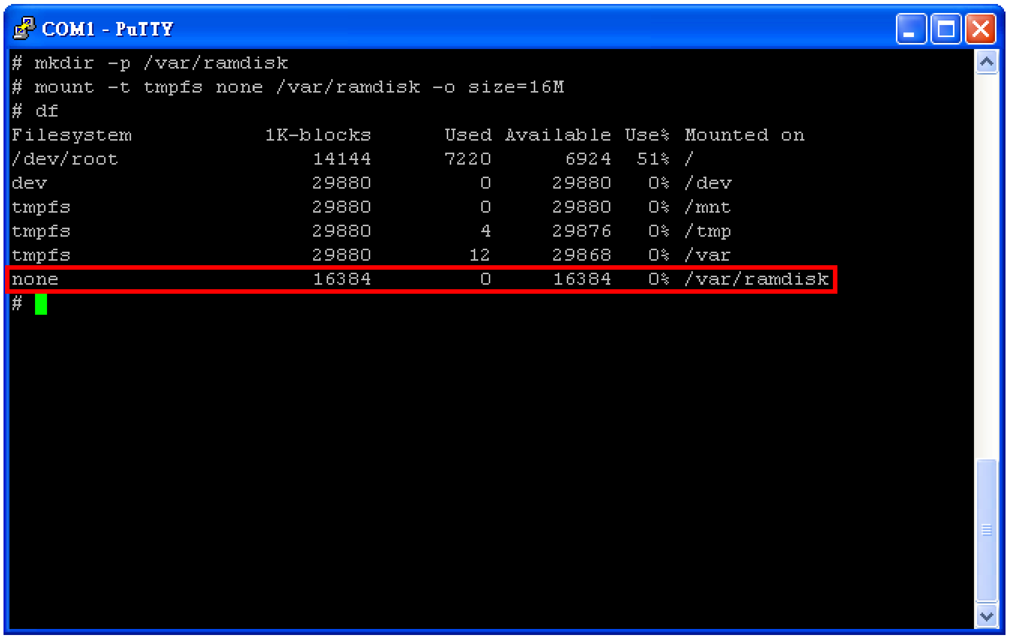
```
# mkdir –p /var/ramdisk
```

Allocate a specific memory size and mount it to the ramdisk:

```
# mount –t tmpfs none /var/ramdisk –o size=16M
```

Finally, you can use df command to check the information of the ramdisk you created.

```
# df
```



```
# mkdir -p /var/ramdisk
# mount -t tmpfs none /var/ramdisk -o size=16M
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/root                14144      7220      6924  51% /
dev                      29880         0     29880   0% /dev
tmpfs                    29880         0     29880   0% /mnt
tmpfs                    29880         4     29876   0% /tmp
tmpfs                    29880        12     29868   0% /var
none                     16384         0     16384   0% /var/ramdisk
#
```

Figure 13, Mount a ramdisk with 16MB size.

If you want to umount the ramdisk, use the command below:

```
# umount /var/ramdisk
```

7.2. NFS (Network File System)

Before using NFS, a NFS server is needed. To create a NFS server on your PC, please refer to:

Linux: http://tldp.org/HOWTO/NFS-HOWTO/server.html

Windows: http://sourceforge.net/projects/freenfs/

When your NFS server is ready, please execute the following commands to connect to your NFS server.

Create a directory for NFS:

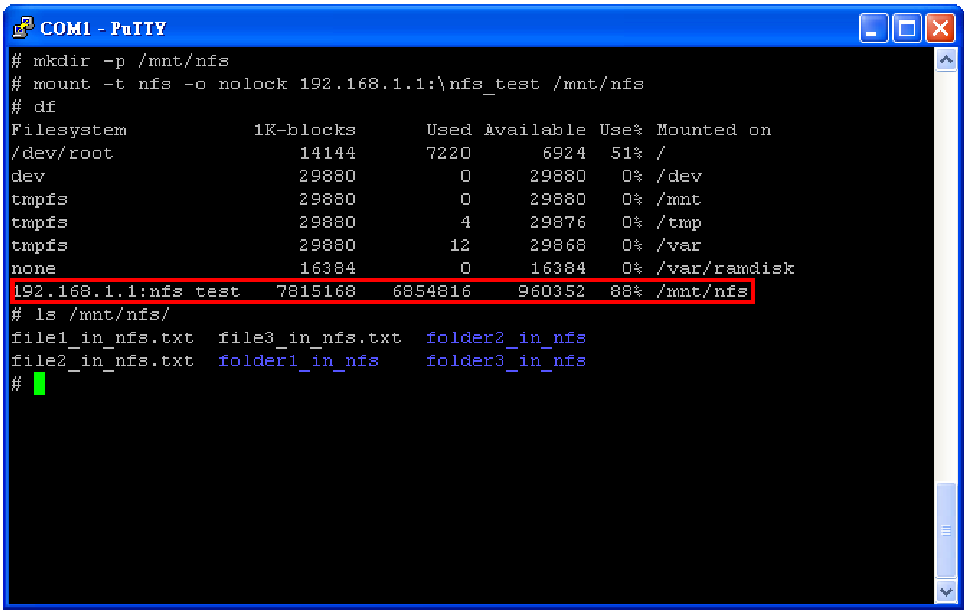
```
# mkdir -p /mnt/nfs
```

Mount NFS server directory to your local directory:

```
# mount -t nfs -o nolock server_IP:/directory /mnt/nfs
```

Finally, you can use df command to check the information of NFS directory you mounted.

```
# df
```



```
# mkdir -p /mnt/nfs
# mount -t nfs -o nolock 192.168.1.1:\nfs_test /mnt/nfs
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/root                14144      7220      6924  51% /
dev                      29880         0     29880   0% /dev
tmpfs                    29880         0     29880   0% /mnt
tmpfs                    29880         4     29876   0% /tmp
tmpfs                    29880        12     29868   0% /var
none                     16384         0     16384   0% /var/ramdisk
192.168.1.1:nfs test   7815168   6854816    960352  88% /mnt/nfs
# ls /mnt/nfs/
file1_in_nfs.txt  file3_in_nfs.txt  folder2_in_nfs
file2_in_nfs.txt  folder1_in_nfs    folder3_in_nfs
# 
```

Figure 14, Mount a remote directory to the local one via NFS.

If you want to disconnect and umount from the NFS server, use the command below:

```
# umount /mnt/nfs
```

7.3. Telnet & SSH Service

Telnet and SSH can be used to remote login to the DCU. SSH uses encryption transportation, and the transportation on Telnet is only by ACSII code.

You can turn on those services on the /etc/inetd.conf to make them as auto-run daemon after booting.

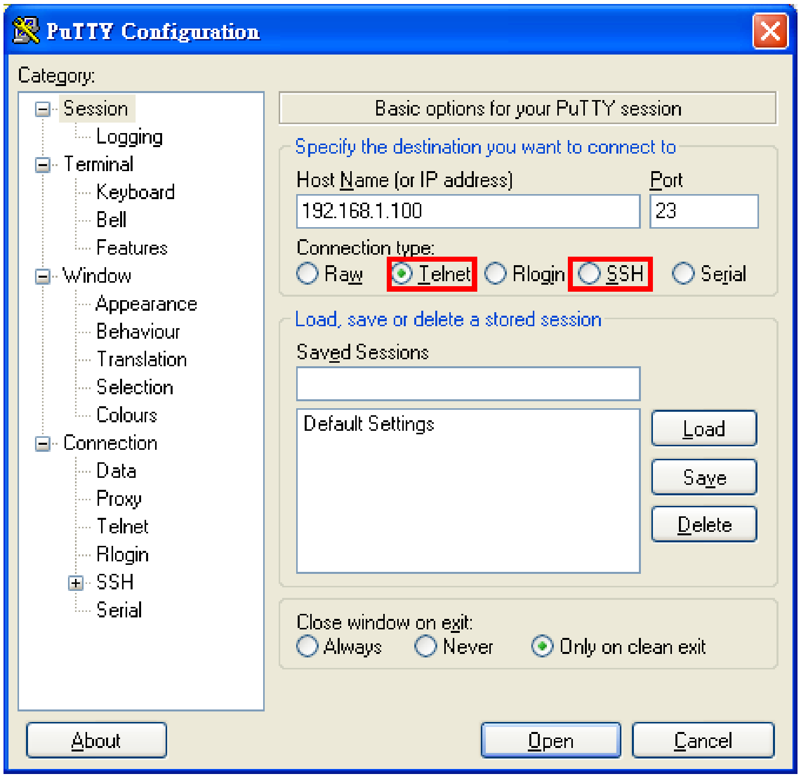
```
# vi /etc/inetd.conf
```

```
ftp stream tcp nowait root /usr/local/sbin/pure-ftpd pure-ftpd -H &
telnet stream tcp nowait root /usr/sbin/telnetd telnetd -i
ssh stream tcp nowait root /usr/bin/dropbear dropbear -i
#www stream tcp nowait root /usr/sbin/httpd httpd -i -h /home/htdocs
```

The default port of telnet service will be port 23.

The default port of SSH service will be port 22.

You can use PuTTY or other terminal programs to login DCU via those protocols.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.100

Port: 23

Connection type: ○ Raw ● Telnet ○ Rlogin ○ SSH ○ Serial

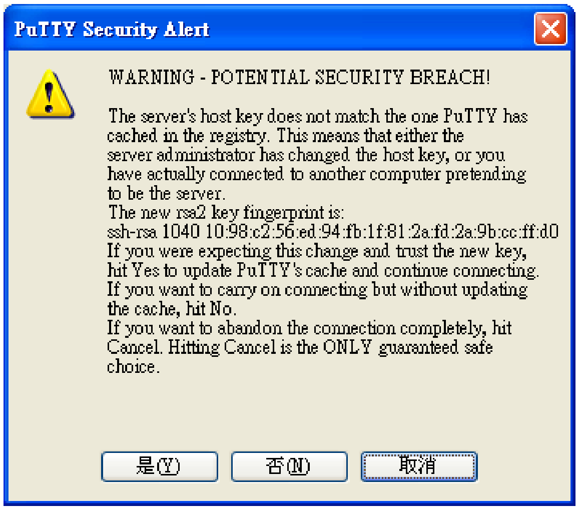
Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 15, Use Putty to create a telnet or SSH connection.

There will be a notification to user to update the key on your PC, when SSH protocol is used. Press “Yes” to allow this operation.

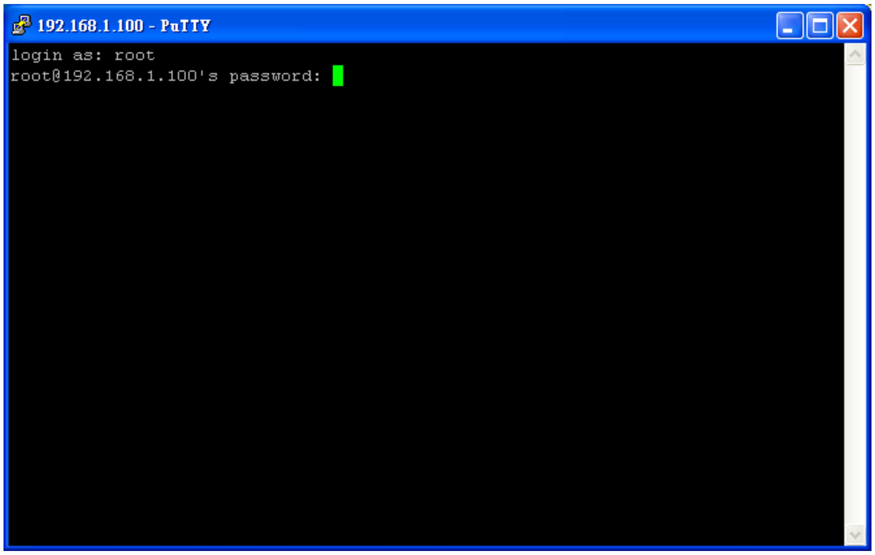


PuTTY Security Alert

WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one PuTTY has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server.
The new rsa2 key fingerprint is:
ssh-rsa 1040 10:98:c2:56:ed:94:fb:1f:81:2a:fd:2a:9b:cc:ff:d0
If you were expecting this change and trust the new key, hit Yes to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating the cache, hit No.
If you want to abandon the connection completely, hit Cancel. Hitting Cancel is the ONLY guaranteed safe choice.

Figure 16, Notification for key updating.



192.168.1.100 - PuTTY

```
login as: root
root@192.168.1.100's password:
```

Figure 17, Login to the DCU via telnet or SSH.

Default username/password: root/dnidni

After login, you can operate the DCU remotely.

7.4. FTP & SCP

FTP and SCP can be used for file transmission to the DCU. Ftp is the normal way to transmit file on the internet, and SCP is a file transmission way on the SSH protocol.

You can turn on those services on the /etc/inetd.conf to make them as auto-run daemon after booting.

```
# vi /etc/inetd.conf
```

```
ftp stream tcp nowait root /usr/local/sbin/pure-ftpd pure-ftpd -H &
telnet stream tcp nowait root /usr/sbin/telnetd telnetd -i
ssh stream tcp nowait root /usr/bin/dropbear dropbear -i
#www stream tcp nowait root /usr/sbin/httpd httpd -i -h /home/htdocs
```

The default port of ftp service will be port 21.

SCP:

SCP can be used correctly when SSHd is turned on.

FTP/SCP client on PC:

FTP: http://filezilla-project.org/download.php

SCP: http://winscp.net/

FTP/SCP client on DCU

FTP:

```
# ftpget –u username –p password HOST_IP [LOCAL_FILE] REMOTE_FILE
# ftpput –u username –p password HOST_IP    REMOTE_FILE [LOCAL_FILE]
```

SCP:

```
# scp root@HOST_IP:/REMOTE_FILE LOCAL_FILE
# scp LOCAL_FILE root@HOST_IP:/REMOTE_FILE
```

7.5. HTTP

DCU can act as a simple HTTP WEB server, and you can follow the command below to build up your own web server.

Create a folder for WEB server and create a homepage.

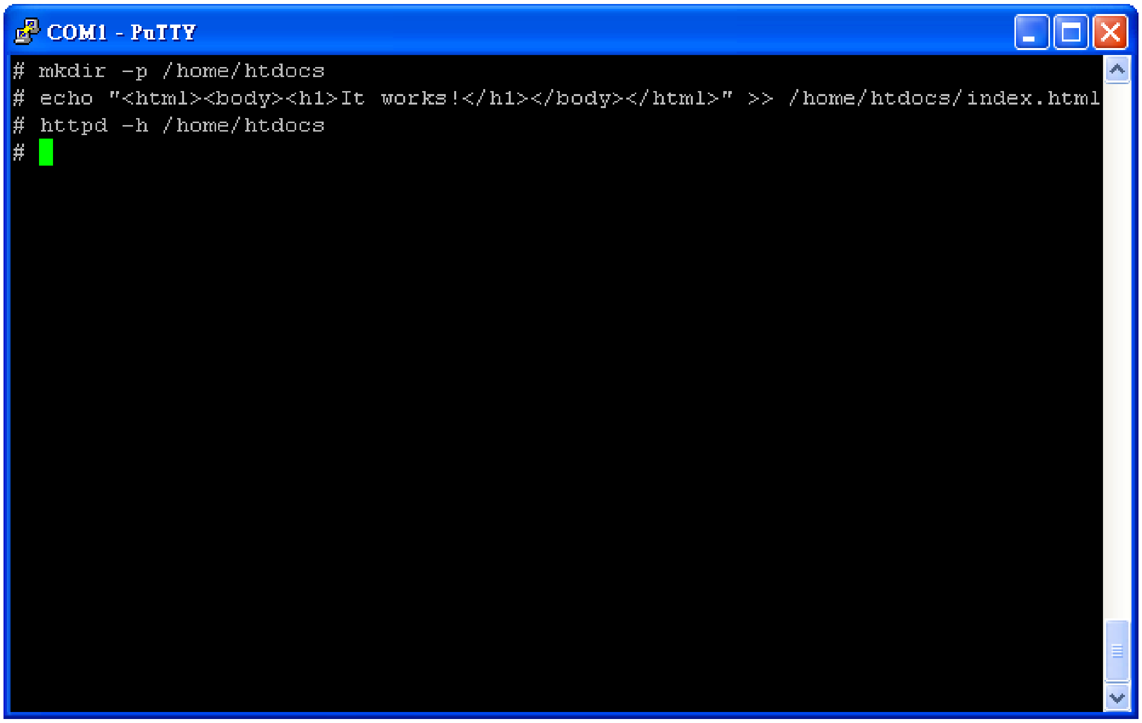
mkdir –p /home/htdocs

echo "<html><body><h1>It works!</h1></body></html>" >> /home/htdocs/index.html

You can turn on those services on the /etc/inetd.conf to make them as auto-run daemon after booting.

# vi /etc/inetd.conf

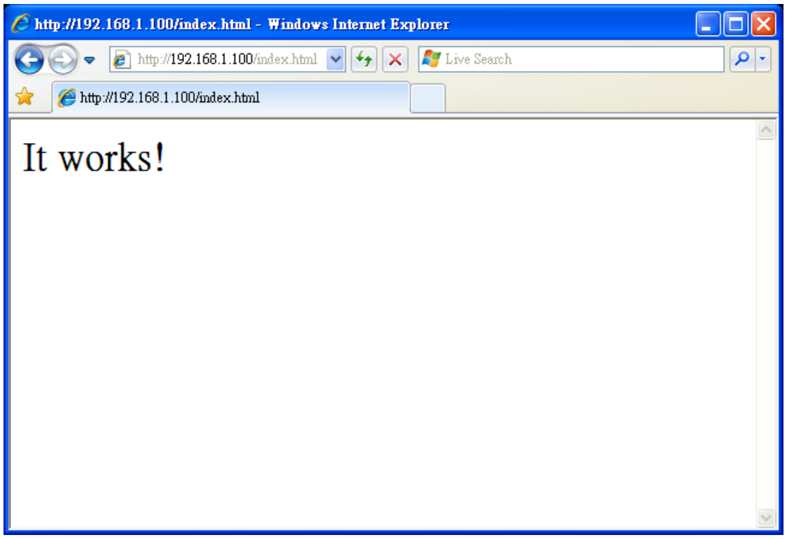
```
ftp stream tcp nowait root /usr/local/sbin/pure-ftpd pure-ftpd -H &
telnet stream tcp nowait root /usr/sbin/telnetd telnetd -i
ssh stream tcp nowait root /usr/bin/dropbear dropbear -i
www stream tcp nowait root /usr/sbin/httpd httpd -i -h /home/htdocs
```



```
# mkdir -p /home/htdocs
# echo "<html><body><h1>It works!</h1></body></html>" >> /home/htdocs/index.html
# httpd -h /home/htdocs
#
```

Figure 18, Create a simple web server.

You can use browser to show the page you created if the httpd is running well.


http://192.168.1.100/index.html - Windows Internet Explorer
http://192.168.1.100/index.html
Live Search
http://192.168.1.100/index.html

It works!

Figure 19, Web page from Http server.

7.6. NTP & RTC

NTP protocol is used to synchronize the time between DCU and the time server. You need to find a NTP server and use the following command to adjust the time on DCU.

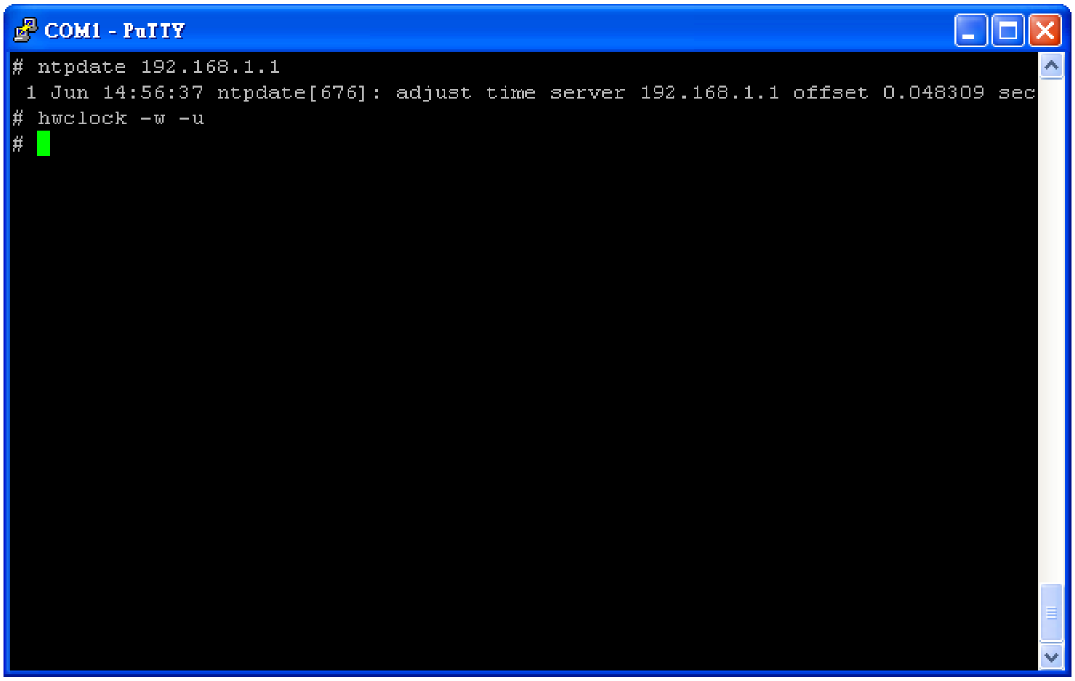
```
# ntpdate <ntpserver_ip>
```

If there is no NTP server that DCU can reach, you can also adjust Linux system time manually.

```
# date MMDDhhmmYYYY
```

After adjusting Linux system time, you need to save the correct time to the HW RTC. Please execute the following command to save time to RTC.

```
# hwclock -w –u
```



```
# ntpdate 192.168.1.1
 1 Jun 14:56:37 ntpdate[676]: adjust time server 192.168.1.1 offset 0.048309 sec
# hwclock -w -u
#
```

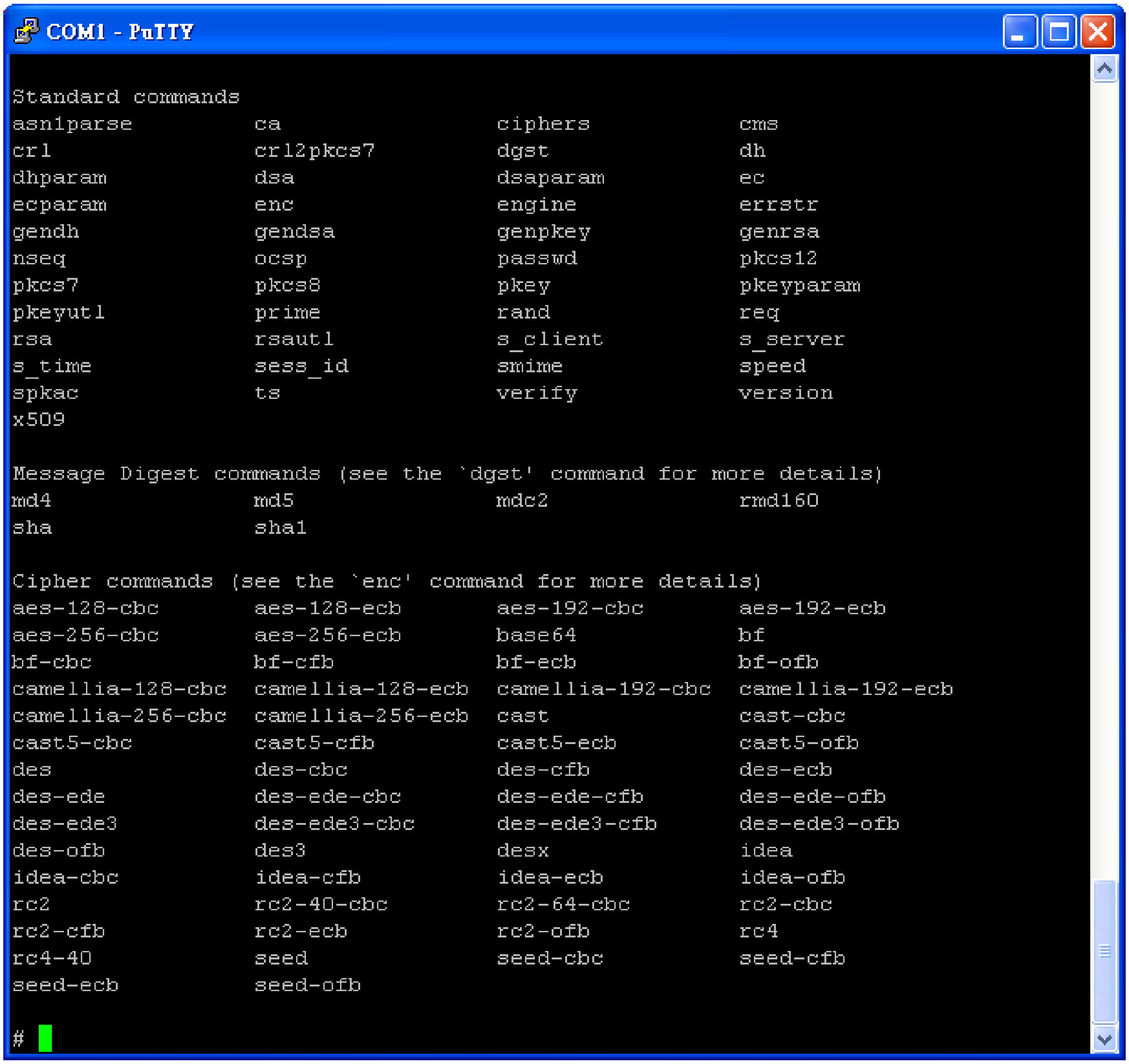
Figure 20, Time synchronization with ntp server and write back to the RTC.

7.7. SSL

OpenSSL is used to encrypt and decrypt on different algorithm for different purpose.

You can use the command to check what kind of command and algorithm are supported.

# openssl –h



```
Standard commands
asn1parse          ca                 ciphers            cms
crl                crl2pkcs7          dgst               dh
dhparam            dsa                dsaparam           ec
ecparam            enc                engine             errstr
gendh              gendsa             genpkey            genrsa
nseq               ocsp               passwd             pkcs12
pkcs7              pkcs8              pkey               pkeyparam
pkeyutl            prime              rand               req
rsa                rsautl             s_client           s_server
s_time             sess_id            smime              speed
spkac              ts                 verify             version
x509

Message Digest commands (see the `dgst' command for more details)
md4                md5                mdc2               rmd160
sha                sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc        aes-128-ecb        aes-192-cbc        aes-192-ecb
aes-256-cbc        aes-256-ecb        base64             bf
bf-cbc             bf-cfb             bf-ecb             bf-ofb
camellia-128-cbc   camellia-128-ecb   camellia-192-cbc   camellia-192-ecb
camellia-256-cbc   camellia-256-ecb   cast               cast-cbc
cast5-cbc          cast5-cfb          cast5-ecb          cast5-ofb
des                des-cbc            des-cfb            des-ecb
des-ede            des-ede-cbc        des-ede-cfb        des-ede-ofb
des-ede3           des-ede3-cbc       des-ede3-cfb       des-ede3-ofb
des-ofb            des3               desx               idea
idea-cbc           idea-cfb           idea-ecb           idea-ofb
rc2                rc2-40-cbc         rc2-64-cbc         rc2-cbc
rc2-cfb            rc2-ecb            rc2-ofb            rc4
rc4-40             seed               seed-cbc           seed-cfb
seed-ecb           seed-ofb

#
```

Figure 21, OpenSSL commands and algorithms supported.

7.8. Sendmail via SMTP

Sendmail is used to send a mail via SMTP protocol, the following command is an example to send a mail from DCU to Gmail SMTP server, and Gmail SMTP server will help you to send the mail to the receiver’s mail server.

Use vi to create an example mail:

```
# vi /var/mailtest
```

```
Subject: This is a mail test
CC: xxxx@gmail.com
To: yyyy@gmail.com
From: zzzz@gmail.com

This is a test
```

After editing the test mail content, please use the following command to send it out. (Noted: make sure that the DCU can access Internet.)

```
# sendmail -f zzzz@gmail.com -v -H 'openssl s_client -connect smtp.gmail.com:465 -quiet' -auUSERNANE
-apPASSWORD < /var/mailtest
```

8. I/O Control

8.1. Data Collector ID

To get DCU id, execute the following command.

```
# getuid
```

8.2. System Ready LED

To get turn on or turn off the ready led, execute the following command.

```
# sysrd-led-on
# sysrd-led-off
```

8.3. Buzzer

To control the buzzer, execute the following command with the unit in millisecond.

```
# buzzer <msec>
```

8.4. Reset Button

The action of the reset button is programmable. The default action is:

< 5 seconds: Reset the device.

> 5 seconds: Recovery to the default settings.

8.5. RS-232 & RS-485

RS-232 port is mapped to the Linux device /dev/ttyM0, and RS485 is /dev/ttyM1. RS-485 can support 2-wire and 4-wire mode, you need to select the correct mode before you use it. To select the RS-485 mode, execute the following command.

2-wire mode:

```
# setport 1
```

4-wire mode:

```
# setport 0
```

8.6. SD Card

SD card is designed to be mounted automatically. When a SD card is inserted into the SD card slot, there will be directory on /var/sd. The mount point will be removed when the SD card is removed.

8.7. Combine WDT Into Your Program

DNI data collector provides a watchdog timer which has a 32- bit down counter with a programmable timeout value. On timeout it generates an interrupt and reset signal. The WDT is intended to be used to generate a system reset if a software failure (or a system hang) occurs. The WDT driver provides a set of ioctls to the user. Through this interface user can configure, program and refresh the WDT. The device node of WDT is /dev/watchdog. The following code snippets demonstrate how to use the WDT.

To open the WDT interface:

```
char wdt_dev[] = "/dev/watchdog"
int fd;
fd = open(wdt_dev, O_RDWR);
if (fd < 0) {
        printf("Error in opening device\n");
}
```

To control the WDT and set the timeout as 45 seconds:

```
int ret = 0;
int timeleft=0;
struct watchdog_info ident;
int timeout = 45; /* in seconds */

/* to find out supported options in watchdog */
ret = ioctl(fd, WDIOC_GETSUPPORT, &ident);

/* to set time out */
ioctl(fd, WDIOC_SETTIMEOUT, &timeout);

/* to find out how much time is left before reset */
ret = ioctl(fd, WDIOC_GETTIMEOUT, &timeleft);

/* Refresh watchdog timer at every 10 secs to prevent reset */
while (1) {
        ioctl(fd, WDIOC,KEEPALIVE, 0);
        sleep(10);
}
```

The WDT interface IOCTL options:

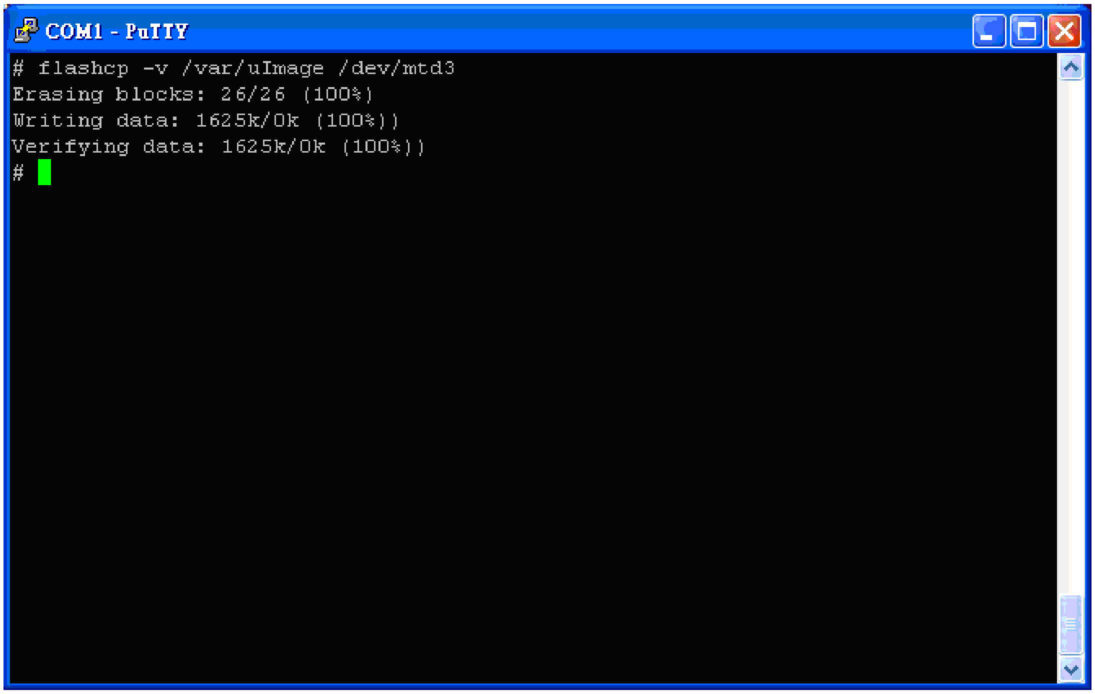
IOCTL Code	Usages
WDIOC_GETSUPPORT	The fields returned in the ident structure are: identity: A string identifying the watchdog driver firmware_version: the firmware version of the card if available. options: A flags describing what the device supports.
WDIOC_KEEPALIVE	This ioctl does exactly the same thing as a write to the watchdog device and hence refreshes the timer
WDIOC_SETTIMEOUT	Set time out in seconds, after which reset would be generated (if WDT is not refreshed)
WDIOC_GETTIMEOUT	Query the current timeout

9. Linux Kernel Upgrade & Backup

Upgrade:

Use FTP or SCP to put your kernel image into the device directory /var and execute the following command on the device to upgrade the kernel image.

```
# flashcp –v /var/<imagename> /dev/mtd3
```



```
# flashcp -v /var/uImage /dev/mtd3
Erasing blocks: 26/26 (100%)
Writing data: 1625k/0k (100%))
Verifying data: 1625k/0k (100%))
#
```

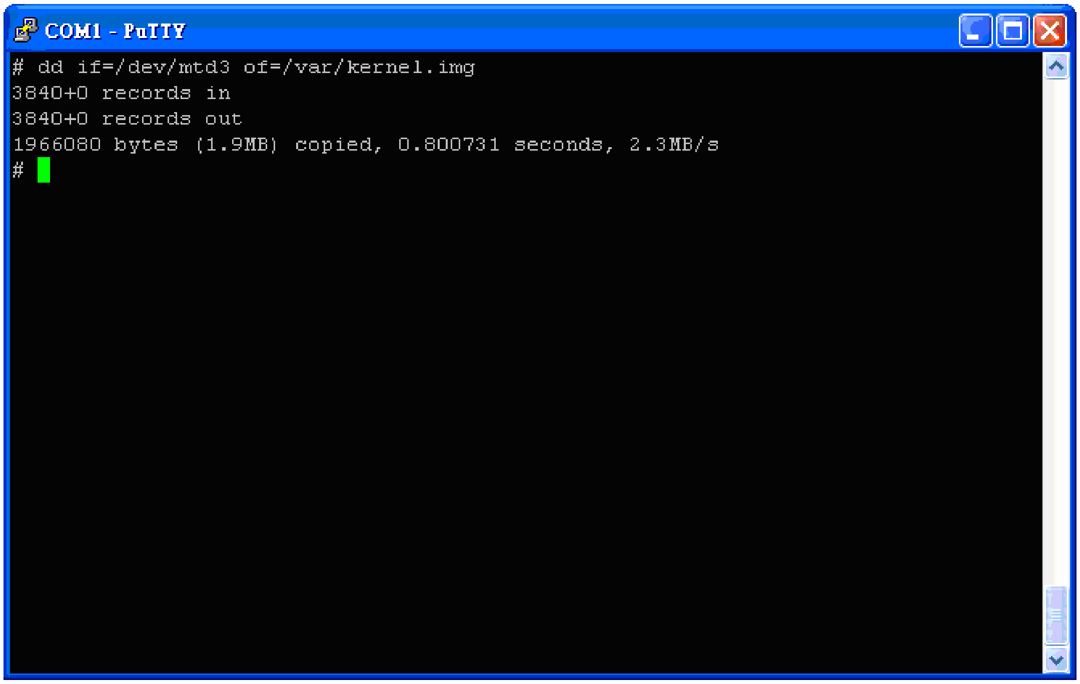
Figure 22, Upgrade Linux kernel image.

When it is done, please reboot your device to apply the kernel image you upgraded.

Backup:

Execute the following command to backup the current kernel image on the device to the /var directory.

```
# dd if=/dev/mtd3 of=/var/kernel.img
```



```
# dd if=/dev/mtd3 of=/var/kernel.img
3840+0 records in
3840+0 records out
1966080 bytes (1.9MB) copied, 0.800731 seconds, 2.3MB/s
# 
```

Figure 23, Backup Linux kernel image from DCU.

When it is done, you can restore the kernel.img via FTP or SCP to your remote computer.

10. Reference

- ST Microelectronics: http://www.st.com/internet/com/home/home.jsp
- NXP: http://www.nxp.com/#/homepage