MARK 5 MEMO #82.1

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY
WESTFORD, MASSACHUSETTS 01886

16 February 2011

Telephone: 781-981-5400
Fax: 781-981-0590

To: Mark 5 Development Group

From: Dan L. Smythe

Subject: Remote Mark 5 Chassis Power Control

The IPMI Intelligent Platform Management Interface on the S5000 Intel main boards in newer Mark 5 systems makes it possible to monitor and control chassis power over the network from a remote host.

With the Debian Etch operating system, the IPMI software is easy to install. On an Intel S5000 Mark 5, running Debian Etch with the 2.6.18 stock Etch kernel, as root:

```
# aptitude install ipmitool [ignore errors]
# /usr/share/ipmitool/ipmi.init.basic [ignore errors]
```

For Debian Lenny, replace the `ipmi.init.basic` command with these two commands:

```
modprobe ipmi_devintf
modprobe ipmi_si
```

```
# ipmitool power status
Chassis Power is on
```

If the `ipmitool` command works, and produces useful output, all you need do it make it work after the next reboot is to add the `/usr/share/ipmitool/ipmi.init.basic` command to `/etc/rc.local.`

Note that you need to be root to use the `ipmitool` command, because it can do dangerous things, such as turn off the power.

Before you can use IPMI for remote chassis power control, the IPMI chip on the main board must be configured. This configuration requires issuing a number of `ipmitool` commands to the IPMI chip on the local host. Apparently, some of these configuration commands will not work if they are entered in the wrong order. I have put most of these configuration commands in the `ipmilanconf and ipmiuser` scripts at the end of this memo. I think I got the order right.

First run the `ipmilanconf` script, as root:

```
# ipmilanconf <IP address> <gateway IP address>
```

The IP address used by the IPMI chip must be a static IP address reserved exclusively for IPMI use; that is, the IPMI chip *should not* use the same IP address as the host operating system. There are two network interfaces connected to the NIC1 connector on the S5000 main board, and another two interfaces connected to the NIC2 connector. The NIC1 connector is connected to IPMI “channel” 1, and the NIC2 connector is connected to IPMI “channel” 2. These connectors are shared with the network interfaces used by the host operating system.

An example for NIC1:

IP address for the OS: fs-pc (192.52.61.127)
IP address for IPMI: linux1-pc (192.52.61.123)
MAC address for the OS: 00:15:17:52:d6:84
MAC address for IPMI: 00:15:17:52:d6:86 (channel 1)

This main board has two additional MAC addresses for NIC2, which are not used:

MAC address for the OS: 00:15:17:52:d6:85
MAC address for IPMI: 00:15:17:52:d6:87 (channel 2)

The configuration commands in the `ipmilanconf` script configure the lan interface for IPMI channel 1, which is connected to the NIC1 connector. After running the `ipmilanconf` script, carefully check the output for errors.

Next run the `ipmiuser` script:

```
# ipmiuser 2 <username>
```

The `set password` command in this script will ask you to enter a password for `<username>` access to the IPMI chip. This password *should not* be the same as any other passwords. Again, after running the `ipmiuser` script, carefully check the output for errors.

Then enter the `ipmitool` interactive shell:

```
# ipmitool shell
ipmitool>
```

Enter `lan print 1` and see something like

```
Set in Progress         : Set in Progress
Auth Type Support       : NONE MD5 PASSWORD
Auth Type Enable        : Callback :
                        : User     :
                        : Operator :
                        : Admin    : PASSWORD
                        : OEM      :
IP Address Source       : Static Address
IP Address              : <IP address>
Subnet Mask             : 255.255.255.0
MAC Address             : XX:XX:XX:XX:XX:XX
SNMP Community String   :
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control       : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 2.0 seconds
Default Gateway IP      : <gateway IP address>
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : XXXXXXXXXXXXXXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

Enter `user list 1` and see something like

```
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv Limit
2   <username>         true    false      true       ADMINISTRATOR
```

Enter `channel info 1` and see something like

```
Channel 0x1 info:
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 0
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : enabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : enabled
    Per-message Auth    : enabled
    User Level Auth     : enabled
    Access Mode         : always available
```

If everything looks OK, then you are ready to test it. Enter `quit` to exit the `ipmitool` interactive shell.

The easiest thing to check is the `power status`. Note - you CAN NOT issue IPMI-over-lan commands from the same machine. That is, you must perform the following test from a *different* machine than the one you have just configured.

```
# ipmitool -H <IP address> -I lan -U root power status
Password:
Chassis Power is on
```

If it were `off,` you could turn it back on with

```
# ipmitool -H <IP address> -I lan -U <username> power on
```

`ipmilanconf` script:

```
#!/bin/tcsh
# ipmilanconf
# Revised:  2009 December 11, DLS
if ( ` id -u ` != 0 )  then
  echo "  You must be root to run this script.  Sorry."
  exit -1
endif
if ($2 == '') then
   echo "Usage: ipmilanconf <IP address> <gateway IP address>."
else
   ipmitool lan set 1 ipsrc static
   ipmitool lan set 1 ipaddr $1
   ipmitool lan set 1 netmask 255.255.255.0
   ipmitool lan set 1 defgw ipaddr $2
   ipmitool lan set 1 arp respond on
   ipmitool lan set 1 access on
   ipmitool channel authcap 1 4
   ipmitool lan set 1 auth admin password
endif
```

`ipmiuser` script:

```
#!/bin/tcsh
# ipmiuser
# Revised:  2009 December 11, DLS
if ( ` id -u ` != 0 )  then
  echo "  You must be root to run this script.  Sorry."
  exit -1
endif
if ($2 == '') then
   echo "Usage: ipmiuser <userid> <username>."
else
   ipmitool user set name $1 $2
   ipmitool user set password $1
   ipmitool user enable $1
   ipmitool channel setaccess 1 $1 ipmi=on callin=on privilege=4
endif
```