

\$IPMITOOLARGS is an environment variable I set to allow my utility programs to work. It contains the IP address and Most of the static param settings, which I got from you.

```
[tgorski@clover:~/src/mmc]$ echo $IPMITOOLARGS  
-H 128.104.29.84 -P "" -T 0x82 -B 0 -b 7
```

List of commands in the zip file. These are all pretty simple knock-ups, as you'll see from the .cpp files.

```
[tgorski@clover:~/src/mmc]$ ls  
%tmp_ipmitool_output alarmmask.cpp amcreset bkendpwr.cpp getpmvals gettimes.cpp junkfile nvareainfo  
setcurrent.cpp setsystime  
Makefile alarmstatus amcreset.cpp bootmode getpmvals.cpp hovride mmc_cmdlist.txt nvareainfo.cpp  
setpmvals setsystime.cpp  
alarmmask alarmstatus.cpp bkendpwr bootmode.cpp gettimes hovride.cpp mmc_ipmi_cmds.zip setcurrent  
setpmvals.cpp
```

The MMC has a 32-bit counter on it that can track unix system time, if told .

!!! Note that to these simple programs, slot 13 = MCH2. If you plug the AMC13 in MCH1 and the real MCH in MCH2, then the cards will run, but these simple programs will use the wrong IPMB address

```
[tgorski@clover:~/src/mmc]$ setsystime -s 13
```

This command returns the time stats kept by the MMC. The first is how long the MMC has been booted, the second is The length of time that the back end has been hot, and the last is the system time as the MMC knows it.

```
[tgorski@clover:~/src/mmc]$ gettimes -s 13  
MMC Up Time: 0 Days 0 Hours 6 Minutes 52 Seconds  
MMC Hot Time: 0 Days 0 Hours 6 Minutes 31 Seconds  
MMC System Time: Wed Oct 26 09:13:51 2011
```

Remote command to turn back end power off

```
[tgorski@clover:~/src/mmc]$ bkendpwr -s 13 off
```

Time query repeated. Notice the hot time is zero

```
[tgorski@clover:~/src/mmc]$ gettimes -s 13  
MMC Up Time: 0 Days 0 Hours 7 Minutes 30 Seconds  
MMC Hot Time: 0 Days 0 Hours 0 Minutes 0 Seconds  
MMC System Time: Wed Oct 26 09:14:30 2011
```

Turning back end power on

```
[tgorski@clover:~/src/mmc]$ bkendpwr -s 13 on
```

Repeated time query shows the back end hot for about 6 seconds. Note that every time the back end power comes On, the MMC automatically strobos the PROGRAM_B# line.

```
[tgorski@clover:~/src/mmc]$ gettimes -s 13  
MMC Up Time: 0 Days 0 Hours 7 Minutes 46 Seconds  
MMC Hot Time: 0 Days 0 Hours 0 Minutes 6 Seconds  
MMC System Time: Wed Oct 26 09:14:46 2011
```

This command is a remote override to cause the MMC to act as though the handle is pulled out

```
[tgorski@clover:~/src/mmc]$ hovride -s 13 out
```

With any command, the lone argument "?" will cause it to list the relevant arguments.

```
[tgorski@clover:~/src/mmc]$ hovride ?
```

command line arguments:

? - display this text

-s <slot#> - MicroTCA slot number (1-13)

-t <time> - Optional override time in seconds to max of 25 seconds

in|out|clr

Clearing the override, which means the MMC reports the handle state as it currently is

```
[tgorski@clover:~/src/mmc]$ hovride -s 13 clr
```

The cold reset causes the backend power to the V6 and S6 to go down for 3 seconds, then come up,
With a program strobe being asserted once the backend power stabilizes

```
[tgorski@clover:~/src/mmc]$ amcreset -s 13 cold
```

Performing Cold Reset of AMC back end

Write of 8 bytes of data to address 0 of the Spartan 6 config memory

```
[tgorski@clover:~/src/mmc]$ ipmitool $IPMITOOLARGS -t 0xa4 raw 0x32 0x33 0 0 0 8 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

Reading the first 10 bytes of config memory (first 8 have the written data, last two have init value of 0)

```
[tgorski@clover:~/src/mmc]$ ipmitool $IPMITOOLARGS -t 0xa4 raw 0x32 0x34 0 0 0 10
```

```
11 22 33 44 55 66 77 88 00 00
```

Writing address 5 to value 0xcc

```
[tgorski@clover:~/src/mmc]$ ipmitool $IPMITOOLARGS -t 0xa4 raw 0x32 0x33 0 5 0 1 0xcc
```

Repeat of earlier 10-byte read, with address 5 now at value 0xcc

```
[tgorski@clover:~/src/mmc]$ ipmitool $IPMITOOLARGS -t 0xa4 raw 0x32 0x34 0 0 0 10
```

```
11 22 33 44 55 cc 77 88 00 00
```