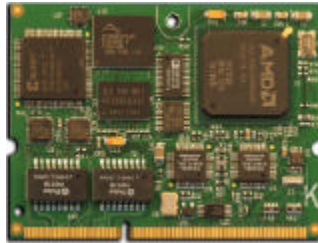


## ATCA Shelf Manager ShMM-300 V2.2.0 Firmware Update Procedure

Thank you for choosing a Schroff or Electronic Solutions ATCA chassis and Shelf Manager. The Shelf Manager in your 5, 14, or 16 slot chassis is based on Pigeon Point Systems technology. Pigeon Point Systems periodically releases new versions of the Shelf Manager firmware that provide new features, bug fixes, and support for new hardware. The details of this firmware update are in the list below.

Your shelf manager may contain either a ShMM-300 or a ShMM-500 Management Mezzanine board. The ShMM-500 and ShMM-300 use different microprocessors so it is important to install the correct firmware.

### Identifying your Shelf Management Mezzanine:



**ShMM-500**



**ShMM-300**

This procedure is for ShMM-300 based Shelf Managers. If your Shelf Manager contains a ShMM-500 Management Mezzanine please use the ShMM-500 V2.2.0 Firmware Update Procedure.

### ShMM-300 Firmware with an Ethernet Redundancy Connection

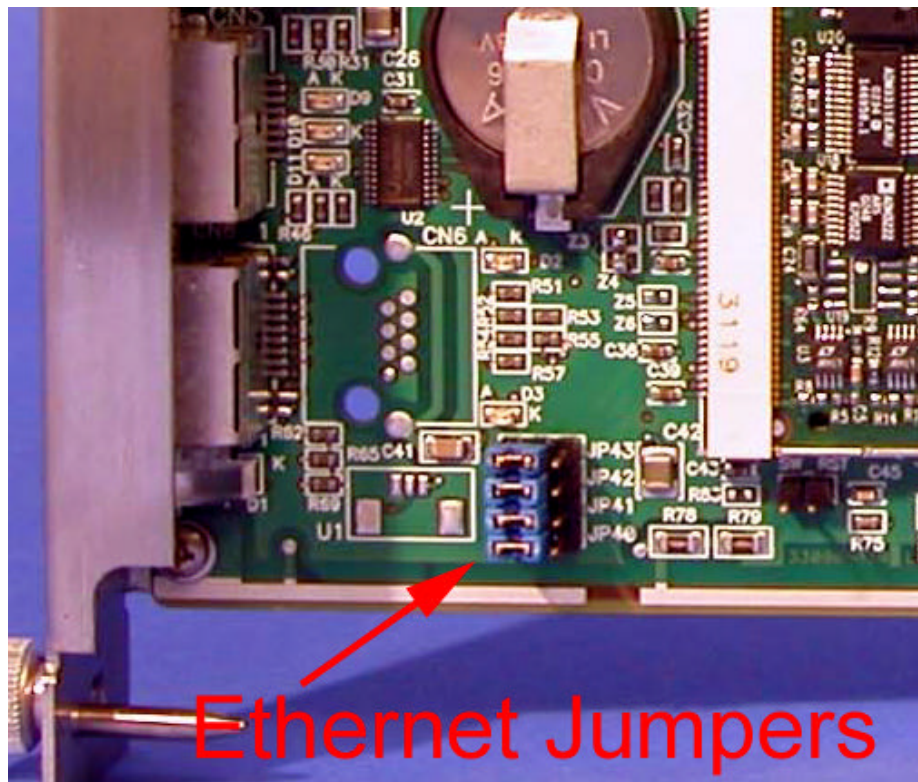
63998-05154.zip

## Updating your Shelf Manager to Release 2.2.0

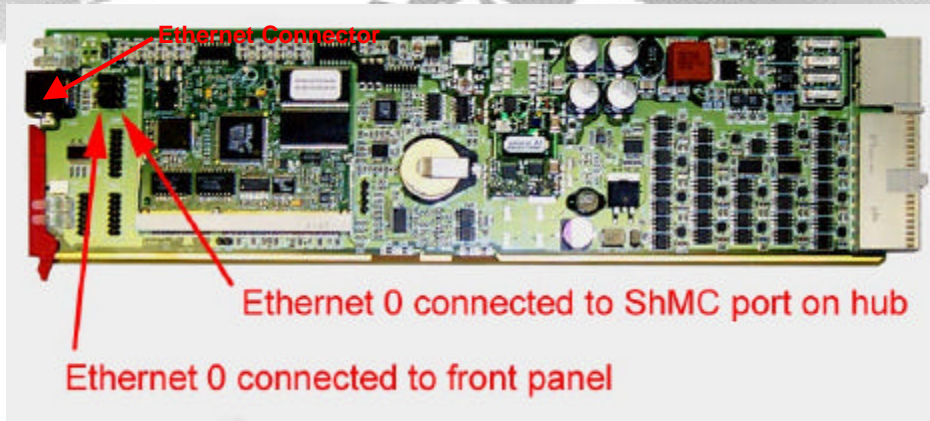
The easiest way to update the FLASH memory firmware images in your Shelf Manager is to use ARMboot to load the images into RAM from a TFTP server and then to copy the RAM images to FLASH memory.

Section 2.7.2 of the Monterey Linux User's Guide details the procedure for configuring a Linux server as a TFTP server. The Monterey Linux User's Guide is available from our ATCA WWW page at: <http://www.atca.com/atca/manuals>.

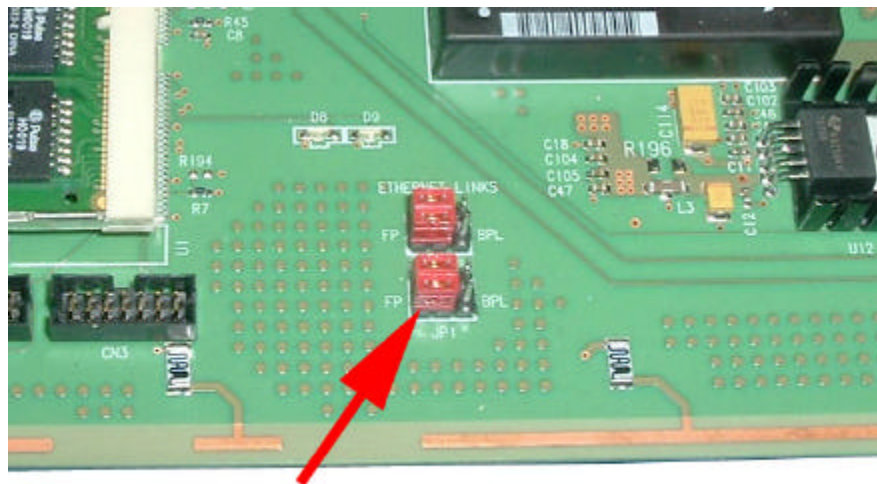
Since the shelf management application will not be running during the TFTP upgrade the Base Interface Switch will not negotiate E-keys with the shelf manager. This means that there will not be a Base Interface network connection to the shelf manager. You will need to set the jumpers on the Shelf Manager to route the primary Ethernet port to the front panel and then connect an Ethernet cable from the front panel to your network. See the ShMM-ACB-II, ShMM-ACB-III, or ShMM-ACB-FC Technical Specification for jumper details.



Ethernet jumpers on Schroff ACB-II Shelf Manager positioned to route the eth0 Ethernet port to the front panel.



Ethernet jumpers on a Schroff ACB-III Shelf Manager positioned to route the eth0 Ethernet port to the front panel.



**Ethernet Port eth0 connected to front panel**

Ethernet jumpers on an Electronic Solutions SMC-I Shelf Manager positioned to route the eth0 Ethernet port to the front panel.

You can then connect the Shelf Manager to your TFTP server through a hub or with a crossover cable.

Connect a serial console cable to from a terminal or PC running a terminal emulator (Hypercom, Minicom, etc) to the console port of the Shelf Manager. A 2-meter long console cable for Schroff Shelves is part number CBL000001, and for Electronic Solutions Shelves is part number VEN12ATCA-SC. Details of the console cable construction are at the end of this procedure. Set the terminal or emulator to 9600, N 8, 1 (9600 baud, no parity, 8 bits per character, 1 stop bit).

In the example below the commands you need to enter are colored **red**.

---

### Step 1

Download the new 2.2.0 kernel and root filesystem (rfs) images: 63998-05154.kernel, and 63998-05154.rfs. These images need to be placed on a TFTP server that is on the same network as the ShMM module.



**Step 2**

Power on your ATCA shelf and interrupt the boot-up process by pressing the space bar on the serial port console of the Shelf manager:

```
ARMboot code: 10f00000 -> 10f15a8c
DRAM Configuration:
Bank #0: 10000000 16 MB
Flash: 8 MB
Hit any key to stop autoboot:  0
ShMM #
ShMM #
```

**Step 3**

Make sure the network settings for your ShMM are set so that it can see the TFTP server on your network:

```
ShMM # printenv serverip ipaddr netmask gateway

serverip=192.168.0.7
ipaddr=192.168.0.2
netmask=255.255.0.0
gateway=192.168.0.1

ShMM#
```

If any of these IP addresses need changing, do it as follows but substitute the desired IP addresses for those shown in the example commands:

```
ShMM # setenv serverip 192.168.0.88
ShMM # setenv ipaddr 192.168.0.100
ShMM # saveenv

Un-Protected 1 sectors
Erasing sector  0 ... Erasing sector at 0x 800000
ok.
Saving Environment to Flash...done.
Protected 1 sectors
```

**Step 4**

TFTP download into RAM the 63998-05154.kernel image:

```
ShMM # tftp 10400000 63998-05154.kernel
Link: 100MBit Half Duplex.
ARP broadcast 1
eth addr: 00:06:29:b0:57:51
TFTP from server 204.62.230.65; our IP address is 97.32.101.116
Filename '63998-05154.kernel'.
Load address: 0x10400000
Loading:
#####
#####
done
Bytes transferred = 488179 (772f3 hex)
ShMM #
```

Record the hex number of bytes downloaded... this will be needed for step 6

**Note: If the TFTP download did not work DO NOT proceed to the next step.**

#### Step 5

Erase the flash area that contains the kernel image

```
ShMM # erase 1:9-24
Erase Flash Sectors 9-24 in Bank # 1:
Erasing sector 9 ... Erasing sector at 0x 20000
ok.
Erasing sector 10 ... Erasing sector at 0x 30000
ok.
...
...
Erasing sector 24 ... Erasing sector at 0x ....
ok.
Done.
ShMM #
```

#### Step 6

Program the kernel image into flash: (note that we use the # bytes transferred from tftp command as third argument):

```
ShMM # cp.b 10400000 20000 772f3
Copy to Flash... done.
ShMM #
```

#### Step 7

TFTP download into RAM the 63998-05154.rfs image, erase flash, and copy image to flash

```
ShMM # tftp 10400000 63998-05154.rfs
Link: 100MBit Half Duplex.
ARP broadcast 1
eth addr: 00:06:29:b0:57:51
TFTP from server 204.62.230.65; our IP address is 97.32.101.116
Filename '63998-05154.rfs'.
Load address: 0x10400000
Loading:
#####
#####
#####
done
Bytes transferred = 2994240 (2db040 hex)

ShMM # erase 1:25-70
Erase Flash Sectors 25-70 in Bank # 1:
Erasing sector 25 ... Erasing sector at 0x ...
ok.
...
Erasing sector 70 ... Erasing sector at 0x ...
ok.

ShMM # cp.b 10400000 120000 2db040
Copy to Flash... done.
ShMM #
```

**Step 8**

Erase the JFFS file systems:

```
ShMM # erase 2:1-70
```

```
Erasing sector 1 ... Erasing sector at 0x 802000
ok.
...
...
Erasing sector 70 ... Erasing sector at 0x bf0000
ok.
Done.
ShMM #
```

**Step 9**

Boot the new system:

```
ShMM # run bootcmd
## Booting image at 00020000 ...
  Image Name:      uClinux-2.4.17
  Image Type:      ARM Linux Kernel Image (gzip compressed)
  Data Size:       488115 Bytes = 476 kB = 0 MB
  Load Address:   10008000
  Entry Point:     10008000
  Verifying Checksum ... OK
  Uncompressing Kernel Image ... OK
## Loading Ramdisk Image at 00120000 ...
  Image Name:      63998-05154 ROMFS Ramdisk Image
  Image Type:      ARM Linux RAMDisk Image (uncompressed)
  Data Size:       2957312 Bytes = 2888 kB = 2 MB
  Load Address:   00000000
  Entry Point:     00000000
  Verifying Checksum ... OK
  Loading Ramdisk to 00120040, end 003f2040 ... OK
```

Starting kernel ...

```
Linux version 2.4.17-uc0 (root@manager.pigeonpoint.com) (gcc version
3.2) #1 Fri Jan 20 08:28:26 PST 2006
Processor: ARM/VLSI ARM 7 TDMI revision 0
Architecture: TI TMS320VC5471
On node 0 totalpages: 4096
zone(0): 0 pages.
zone(1): 4096 pages.
zone(2): 0 pages.
Kernel command line: console=ttyS0,9600 root=/dev/ram0
IP1ADDR=192.168.1.3 IP1DEVICE=eth1 IPADDR=192.168.0.2 IPDEVICE=eth0
HOSTNAME=sentry GATEWAY=192.168.0.1 RC2=/etc/rc.acbfc FLASH_RESET=n
PASSWORD_RESET=n LOGGING=ram START_RC2_DAEMONS=y RC_IFCONFIG=n
Calibrating delay loop... 6.80 BogoMIPS
Memory: 16MB = 16MB total
Memory: 14948KB available (958K code, 172K data, 48K init)
Dentry-cache hash table entries: 2048 (order: 2, 16384 bytes)
Inode-cache hash table entries: 1024 (order: 1, 8192 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 4096 (order: 2, 16384 bytes)
```

```
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
DCSS: successfully started.
Starting kswapd
JFFS2 version 2.1. (C) 2001, 2002 Red Hat, Inc., designed by Axis
Communications AB.
i2c-core.o: i2c core module
i2c-dev.o: i2c /dev entries driver module
i2c-algo-bit.o: i2c bit algorithm module, debug at 1012c5d8
i2c-c547x_bit.o: i2c C547x (bit mode) module
i2c-dev.o: Registered 'i2c of C547x (bit mode)' as minor 0
i2c_algo_c5471_init: i2c c5471 algorithm module
i2c-c5471.o: C5471 I2C adapter module
i2c-dev.o: Registered 'C5471 I2C adapter #0' as minor 1
i2c-dev.o: Registered 'C5471 I2C adapter #1' as minor 2
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with no serial options enabled
ttyS00 at 0xfffff1000 (irq = 6) is a ST16650
ttyS01 at 0xfffff0800 (irq = 7) is a ST16650
C5471 SPI driver, irq 13.
Dallas Semiconductor DS1337 Real-Time Clock driver
EEPROM driver v1.0
ledman: Copyright (C) SnapGear, 2000-2002.
ARM GPIO driver. (C) 2003 Pigeon Point Systems
WDT driver for ShMM.
block: 64 slots per queue, batch=16
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
enet: Phy @ 0x0, type 0x001378e2
eth0: C5471 ENET Version 0.2, 00:50:c2:22:52:80
SMSC LAN91C111 Driver (v2.0), (Linux Kernel 2.4 + Support for Odd Byte)
09/24/01 - by Pramod Bhardwaj (pramod.bhardwaj@smsc.com)
eth1: SMC91C11xFD (rev:1) at 0x1000300 IRQ:12 MEMSIZE:8192b NOWAIT:0
ADDR: 00:50:c2:22:52:81
VC5471: 0x00800000 at 0x00800000
Search for id:(01 22f9) interleave(1) type(2)
Found: AMD Am29LV320B
Flash on VC547x: Found 1 x16 devices at 0x0 in 16-bit mode
number of JEDEC chips: 1
VC5471: using static partition definition
Creating 3 MTD partitions on "Flash on VC547x":
0x00000000-0x00010000 : "ARMboot environment"
0x00010000-0x00100000 : "ETC"
0x00100000-0x00400000 : "VAR"
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 1024 bind 2048)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: romfs filesystem found at block 0
RAMDISK: Loading 2888 blocks [1 disk] into ram disk... |done.
VFS: Mounted root (romfs filesystem) readonly.
Freeing init memory: 48K
init started: BusyBox v0.60.5 (2006.01.19-00:10+0000) multi-call
binary
/etc/rc: Mounting filesystems...
```

```
/etc/rc: Mounted /proc
/etc/rc: Mounted /dev/pts
/etc/rc: Mounted /dev/mtdblock3 to /var
/etc/rc: Mounted ram disk to /var/log
/etc/rc: Started syslogd and klogd
/etc/rc: Mounted ram disk to /var/tmp
/etc/rc: Setting hostname sentry
/etc/rc: /dev/mtdblock2 appears to be empty ... restoring from factory
/etc...
/etc/rc: Mounted /dev/mtdblock2 to /etc
/etc/rc: Calling /etc/rc.acbfc
Board Hardware Address: 0x12
/etc/netconfig: /etc/hosts updated with sentry 192.168.0.3 entry
/etc/netconfig: Updating /etc/profile.sentry with IP settings
/etc/netconfig: Starting inetd...
/etc/rc.acbfc: Starting up DSP...
Booting:
    freq:100MHz
    size:13906 words
    addr:0x10fe0000
dsp firmware (Feb 18 2005-14:33:23)
/etc/rc.acbfc: Updating /etc/profile.sentry with specific settings
/etc/rc.acbfc: Starting snmpd...
/etc/rc.acbfc: Starting httpd...
/etc/rc.acbfc: Starting Shelf Manager ...
```

sentry login:

#### Step 10

Verify the revision level of the newly installed firmware:

```
sentry login: root
Welcome to
```



```
BusyBox v0.60.5 (2005.03.04-01:44+0000) Built-in shell (msh)
# version
Firmware Image Version 63998-05154
```

IPM Sentry Shelf Manager Command Line Interpreter

```
IPM Sentry Shelf Manager ver. 2.2.0
IPM Sentry is a trademark of Pigeon Point Systems.
Copyright (c) 2002-2006 Pigeon Point Systems
Build date/time: Jan 18 2006 16:10:01
All rights reserved
```

#### Step 11

Repeat steps 1-10 for the other shelf manager if one is installed.



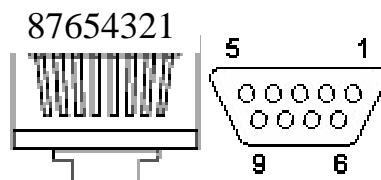
## Configuration Notes for Release 2.2.0

The ARMboot environment variable “rc2” determines which configuration script is run at boot and therefore the configuration of the shelf manager.

Chassis	Shelf Manager Carrier Board	rc2 value
Schroff 5U 5 slot	ACB-II	/etc/rc.acb2
Schroff 14U 14 slot	ACB-FC	/etc/rc.acbfc
Schroff 12U/13U 14/16 slot	ACB-III	/etc/rc.acb3
Electronic Solutions	SMC-I	/etc/rc.smc1

**Schroff Shelf Manager RJ45 to 9 Pin PC Serial Console Cable P/N CBL000001**

RJ45 Pin	RJ45 Signal Name	PC 9 pin D-Sub	Signal Name
1	DSR	4	DTR
2	CD	N/C	CD
3	DTR	6	DSR
4	Ground	5	Ground
5	RxD0	3	TX
6	TxD0	2	RX
7	CTS	7	RTS
8	RTS	8	CTS



RJ45 and 9 pin D-Sub connectors are viewed with the cable going away from you.

**Electronic Solutions RJ11 to 9 Pin PC Serial Console Cable P/N VEN12ATCA-SC**

RJ11 Pin	RJ11 Signal Name	PC 9 pin D-Sub	Signal Name
1			
2	RxD0	3	TX
3	TxD0	2	RX
4	Ground	5	Ground
		4 to 6	DTR/DSR
		4 to 1	DTR/CD
		7 to 8	RTS/CTS

## Obtaining the new firmware

A Zip file containing the firmware FLASH memory images and a firmware update instruction procedure are available from our ATCA WWW page at: <http://www.a-tca.com/atca/software>.

## Contact

If you have any questions on this notice please contact us:

### North America

[infoatca@pentair-ep.com](mailto:infoatca@pentair-ep.com)

Pentair Electronic Packaging  
170 Commerce Drive  
Warwick, RI 02886  
USA

### Europe/Asia

[infoatca@schroff.de](mailto:infoatca@schroff.de)

Schroff GmbH  
Langenalber Strasse 96-100  
75334 Straubenhardt  
Germany