



*BSP Package 2.0.2.
Tornado 1.0.x*

XTOOLS CENTER

BSP Name : VMx42/x62

BSP Release: 2.0.3

Issue 1.50

VM42/VM62/VM42LC/VM642/VM662

Board Support Package

VM42/VM62/VM42LC/VM642/VM662 Board
Support Package
User's Manual



REVISION HISTORY

Tornado BSP Package 2.0.2.				
Issue	Brief Description of Changes	BSP Release	Date of Issue	Author
1.00	Creation (Update from VxWorks 5.2)	2.0.0	13 May, 1996	BO
1.10	Add SCC driver and ER96001	2.0.1	8 July, 1996	BO
1.20	WindView Support and Ethernet Driver update (ER96003)	2.0.2	9 December, 1996	BO
1.3	Update BSP for supporting VM42 LC	2.0.2	16 April, 1997	BO
1.4	Update for BSP Package 2.0.2.	2.0.2	30 May, 1997	BO
1.5	Update BSP Package 2.0.2. / BSP Version 2.0.3	2.0.3	July 22, 1997	BO

This new version (2.0.3) of software supports :

- BSP upgrade to version 2.0.3

Next are the known problems and solutions included in this BSP release :

- The Ethernet driver of the VM42 / VM62 is delayed and this when the network traffic is very high and the VM42 / VM62 is constantly using the Ethernet driver.
- The Eeprom on the VM42 / VM62 is used to store data regarding the board, like the ethernet address and the bootline.

It happened that not all the characters stored in the bootline were correctly read, although they seemed to be correctly stored in the Eeprom.

- When you want to reset a VM42 / VM62 with the **reboot** command on the **target shell**, while an application communicates with an other board and the BNC Ethernet connector is removed, the **reboot** command is not executed.



When you reconnect the BNC Ethernet connector, the communication between the VM42 / VM62 and the other CPU board continues executing. But the **reboot** command is never executed.

When a VM42 / VM62 without BNC Ethernet connector is used, the board reboots correctly as wanted.

- In the standard BSP delivery, the BSP was configured to boot in the VxWorks 5.2 configuration; this means with the shell onboard.

The BSP was also configured to include a Wind View header file, even if Wind View was not installed (timestampDev.h).



Table of contents.

1. INTRODUCTION	6
2. MAIN FEATURES	6
3. BSP INSTALLATION	7
3.1 File Structure	8
3.1.1 Delivered files	8
3.2 Program your bootrom in the VM42 FLASHES	9
3.2.1 Build your bootprom image.	9
3.2.2 Program the bootprom image into flashes.	9
3.3 Configure your Boot Shell parameters	10
3.4 Bootline in flash eprom	11
3.5 VM42 LC	12
4. DEVICES	13
4.1 Memory map	13
4.1.1 Flash eproms	13
4.1.2 DRAM	13
4.1.3 Addresses	13
4.2 SMC Serial Port	14
4.2.1 Port definition	14
4.2.2 Number of available ports	14
4.2.3 Console definition	14
4.3 SCC Serial Ports	15
4.3.1 Port definition	15
4.3.2 Number of available ports	16
4.4 TIMERS	17
4.4.1 System Clock	17
4.4.2 Auxiliary Timer	17
4.4.3 Timestamp Timer	17
4.5 Shared memory and Backplane driver	18
4.5.1 Processor number	18
4.5.2 DUAL RAM size	18
4.5.3 DUAL RAM On board access	18
4.5.4 DUAL RAM VME Access	19
4.5.5 Mailboxes	19
4.6 WindView	20
5. THE LIBMC680X0GNUVM4262.A LIBRARY	21
5.1 E2PROM	21
5.2 Real Time Clock	22
5.3 Ethernet Driver	22
5.4 CPU Speed detection	22



6. BOOT FROM THE VME BUS	23
6.1 Build the needed bootrom image	23
6.2 Build the needed OS	23
6.3 Configure your VMEM-S2	24
6.4 Configure the VM42/VM62	24
6.5 Configure your VME BUS	25
6.6 Configure your boot parameters and boot	25



1. INTRODUCTION

The purpose of this document is to describe the specificity of the PEP VM42 / VM62 Board Support Package for Tornado 1.0 (VxWorks 5.3).

2. MAIN FEATURES

The VM42/VM62 BSP supports:

- all CPU clock frequency.(25,33 Mhz)
- the system Clock using the Periodic Interrupt Timer (PIT) of the MC68360
- the 2 SMC serial ports (RS-232) of the MC68360
- the 4 SCC serial ports (RS-232) of the MC68360
- the Ethernet controller using SCC1 of the MC68360
- the MMU and Cache of the MC68040
- the 2Kbits (256 bytes) E2PROM
- the RTC chip
- the shared memory driver using the onboard dual ported SRAM
- the VME interface and interrupt controller
- the auxiliary timer
- the timestamp driver (for Windview)
- the bootline can be stored in eprom, e2prom or at fixed place in flash eprom



3. BSP INSTALLATION

Warning

The Pep delivery of the BSP contains a standard Wind River header file. In this release Pep Modular Computers had to modify this standard header file : m68360.h.

When the BSP is installed check if the Pep delivery has updated this m68360.h header file.

*The modification history of this m68360.h file should show
"add Pep Specific structures and defines"*

Please refer to the ***Installation Note*** for the installation of this BSP.



3.1 File Structure

3.1.1 Delivered files

WIND_BASE / target	config / pvm42*	tyCoDrv.c		
		bootrom.hex		
		symTbl.c		
		pkLib.c		
		romInit.s		
		sysLib.c		
		sysALib.s		
		config.h		
		Makefile		
		pvm42.h		
		sim60.h		
	libMC68040gnuvm4262.lib			
	src / drv /	mem / e2VM42.c		
		timer / m68360Timer.c		
		serial / m360SMCPep.c		
		serial / m360SCCPep.c		
		h / drv /	timer / timerDev.h	
			timer / m360Timers.h	
	serial / tyloctIPep.h			
	serial / m360SMCPep.h			
	serial / m360SCCPep.h			
	multi / m68360.h		Modified Wind River File	
	multi / m68360Pep.h			
	multi/quicc.h			
	multi/pram.h			
	multi/registers.h			

* for the VM42 LC, the directory pvm42LC is created



3.2 Program your bootrom in the VM42 FLASHES

3.2.1 Build your bootprom image.

A standard **bootrom.hex** file can be found in the delivery in the directory **config/pvm42**.

If a new boot image file needs to be generated, run the makefile with the 'bootrom.hex' entry, so that the bootrom image for the flash eproms is generated.

```
make bootrom.hex
```

3.2.2 Program the bootprom image into flashes.

The bootrom.hex file needs to be programmed in flash eprom.

Use the **Boot Strap Loader** of the VM42/VM42LC/VM62 to do so.

(Refer to the appendix of the **Boot Strap Loader** of the VM42/VM62 hardware manual for more information regarding this issue).



3.3 Configure your Boot Shell parameters

Check the **Entering Boot Parameters** chapter of the **Tornado User's Guide** for a complete description of all the needed boot parameters.

Next you will find an example of such boot parameters.

```
[VxWorks Boot]: c
```

```
boot device:      pep
processor number: 0
host name:        mysun
file name:        /usr/tornado/target/config/pvm42/vxWorks
inet on Ethernet: 128.0.0.128
host inet:        128.0.0.1
user name:        vw
```

```
[VxWorks Boot]: @
```

And your system should boot



3.4 Bootline in flash eprom

The bootline can be stored in :

- e2prom : **standard configuration**
- at a fixed place in the flash eprom (always at the end of the first bank)

The first time the board boots there is not yet a bootline in the e2prom. At that point the BSP takes the standard bootline, defined in the config.h file and stores it in the e2prom.

From then the BSP takes the content of the e2prom as bootline.

For saving the bootline in the flash eprom add the define FLASH_BOOTLINE in the config.h file and the bootline will be saved in the flash eprom.

In the config.h file remove the */** and **/* :

```
/* Bootline in flash eprom */  
/* The bootline can be put at a fixed address */  
/* in the flash eproms */  
/* #define FLASH_BOOTLINE */
```

The address where the bootline is stored depends on the flash eproms used. The supported (tested) flash eproms are the **29F010** and **29F040**.

Since this address depends on this type of flash eproms, the BSP displays the address before retrieving and writing the bootline.



3.5 VM42 LC

The difference between a **VM42** and a **VM42 LC** board is that the **VM42 LC** has **no floating point hardware registers nor instructions** to handle them and so all this has to be done by software .

For generation of the VM42 LC BSP boot eproms and image file, the next definitions are changed in the Makefile file, compared to a normal VM42 BSP :

CPU = MC68040 into **CPU = MC68LC040**

TARGET_DIR = pvm42 into **TARGET_DIR = pvm42lc**
where **pvm42lc** is the directory where the BSP is installed in.

(/usr/vw/config/pvm42lc)

From this moment on your **make** command will compile for a MC68LC040 CPU.



4. DEVICES

4.1 Memory map

4.1.1 Flash eproms

The 32bits wide Flash eprom is available in 2 different sizes:

- 1 MB: divided in 2 banks of 512
- 4 MB: divided in 2 banks of 2 MB

Each bank can be separately protected (see Jumper B1 of the DM600 configuration in the VM42/VM62 User's Manual)

The BSP configures that area from 0x4000000 to 0x4400000.

The default configuration of the BSP is for 1 MB flash eproms.
This should be changed in the MakeSkel and config.h
ROMSIZE = 0x400000

4.1.2 DRAM

The 32bits wide DRAM is available in 2 different sizes: 4 or 16 MB.
The default configuration of the BSP is for 4 MB DRAM.

The size of this memory can be easily changed to 16MB in config.h
LOCAL_MEM_SIZE = 0x1000000

4.1.3 Addresses

See VM42/VM62 User's Manual for memory mapping.



4.2 SMC Serial Port

4.2.1 Port definition

The two SMC serial ports are available on the front panel.
The SMC1 serial port is the upper port (port0: /tyCo/0).
The SMC2 serial port is the lower port. (port1: /tyCo/1).

4.2.2 Number of available ports

The number of available port can be changed in config.h (default is 2).
It can be reduced to 1 by:

```
#undef NUM_TTY  
#define NUM_TTY 1
```

4.2.3 Console definition

The console is defined by default on the lower port (1)
This can be changed to the upper port in config.h:

```
#undef CONSOLE_TTY  
#define CONSOLE_TTY 0
```

Only the SMC Serial ports can be used as Console and not the SCC Serial ports.



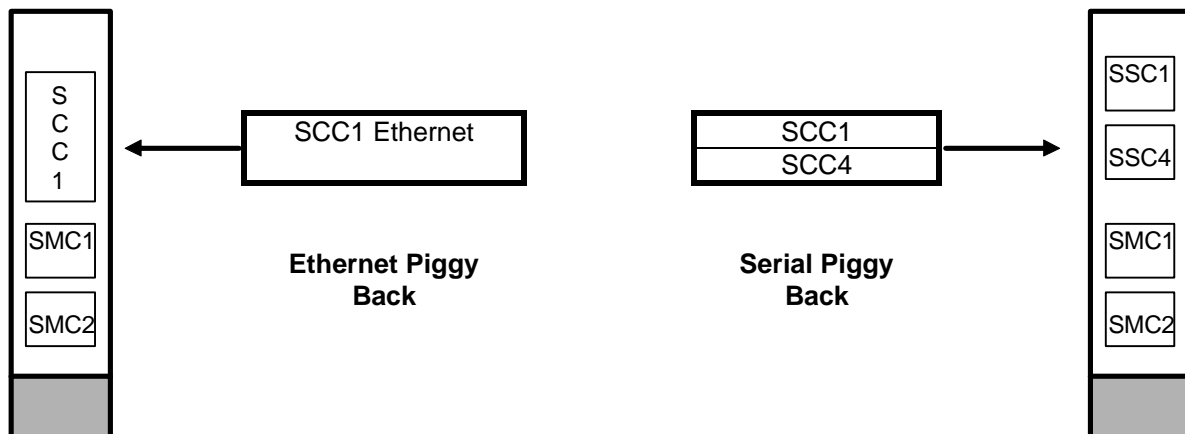
4.3 SCC Serial Ports

refer to “*SCC 68360 Driver Specification Version 1.0*”

4.3.1 Port definition

On a standard VM42 / VM62 there is only one SCC port available, this is the SCC1 port that is used for an ethernet connection with the host. Here it is not a serial port any more.

When there is no ethernet connection on the VM42 / VM62 this SCC1 port can be used as a serial channel.



There are totally 4 SCC serial ports available : SCC1 to SCC4.

The SCC1 serial port 1 : /tyScc360/1
The SCC2 serial port 2 : /tyScc360/2
The SCC3 serial port 3 : /tyScc360/3
The SCC4 serial port 4 : /tyScc360/4

On front panel
Redirected to CXC connector
Redirected to CXC connector
On front panel and
redirected on CXC connector



4.3.2 Number of available ports

The number of available port can be changed in config.h (default is 0).
It can be changed by:

```
#define NUM_SCC 1           for only use SCC 1 (when not used  
as ethernet)  
#define NUM_SCC 2         for use of SCC1 and SCC2  
#define NUM_SCC 3         for use of SCC1, SCC2 and SCC3  
#define NUM_SCC 4         for use of SCC1, SCC2, SCC3 and  
SCC4
```

Remark : When the SCC1 is used as ethernet device and the user wants to instal a serial device on SCC1, the driver will return without changing the ethernet device on SCC1.

This SCC Serial driver can not be used as a bridge, a console or as a debugging help by/for the BSP.

It can only be used in an application written by the user.

Standard the SCC Serial driver is not initialised at boot (refer to **Number of available ports** to configure the use of the SCC Serial channels).



4.4 TIMERS

4.4.1 System Clock

The timer used for the system clock is the PIT Timer of the MC68360. The default value is 60Hz. (60 ticks per second). The clock frequency can be ranged from 10 to 1000 Hz. This timer is configured to give a 0 shift to any tick rate.

Use the *sysClkRateSet* and *sysClkRateGet* function to modify or read the system clock speed.
Use *sysClkEnable* and *sysClkDisable* to enable or disable the timer interrupts.

4.4.2 Auxiliary Timer

The auxiliary timer uses the TIMERS1 and 2 of the MC68360. Its period is 320 ns (F=3,125 Mhz). Use the *sysAuxClkRateSet* and *sysAuxClkRateGet* function to modify or read the system clock speed. Use *sysAuxClkEnable* and *sysAuxClkDisable* to enable or disable the timer interrupts. Use *sysAuxClkConnect* to add a callout function (with parameter) to the auxiliary timer interrupts.

4.4.3 Timestamp Timer

The timestamp driver uses TIMERS3 and 4 of the MC68360. Its period is 320 ns (F=3.125 Mhz). Its rollover period is 1 second. The timestamp driver is only included when *INCLUDE_TIMESTAMP* is defined in *config.h*. This timer is used for Windview. Use *sysTimestampEnable* and *sysTimestampDisable* to enable and disable the timestamp timer. Use *sysTimestamp* to read the timer value. Use *sysTimestampConnect* to add a callout function (with parameter) the the timer interrupts.



4.5 Shared memory and Backplane driver

4.5.1 Processor number

Each board in the VME system has a processor number. This processor number is defined in the VxWorks boot parameters (stored in E2PROM). It can be changed from the boot shell.

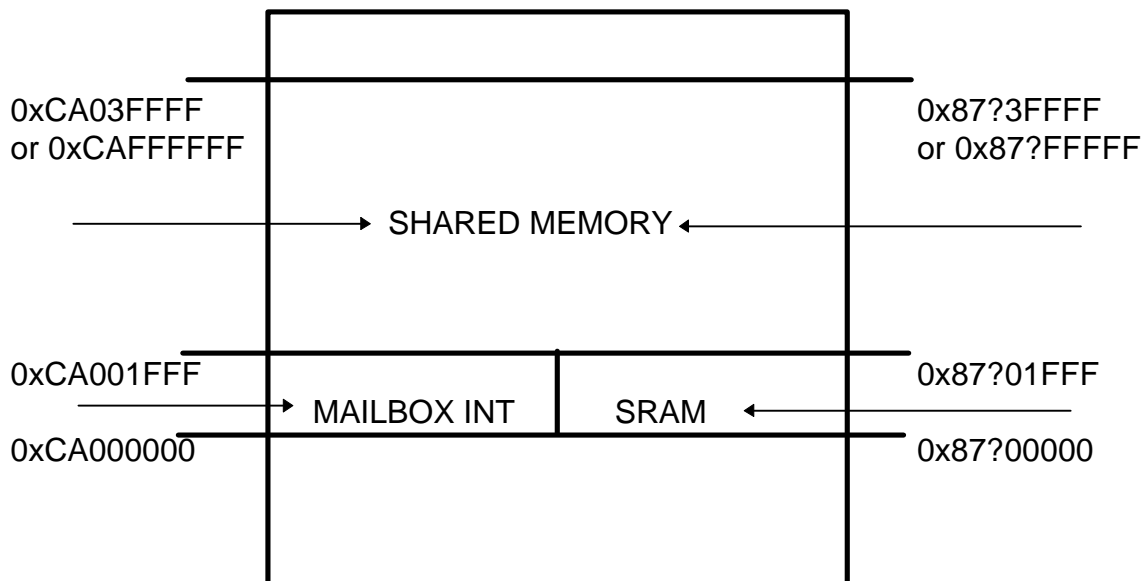
Main CPU board has processor number 0. Other CPU boards (booting from the VME bus shared memory driver) should have processor number 1,2,3...

4.5.2 DUAL RAM size

The dual ported RAM size can be either 256KB or 1MB.

This is changed in config.h

```
#define SRAM_MEM_SIZE 0x40000 /* for 256 KByte SRAM */  
#define SRAM_MEM_SIZE 0x100000 /* for 1 MByte SRAM */
```



4.5.3 DUAL RAM On board access

The onboard access can be done from 0xCA000000.

The real dual ported address range is:

from 0xCA002000 to 0xCA040000 for the 256KB version
and from 0xCA002000 to 0xCA0FFFFFF for the 1MB version.



4.5.4 DUAL RAM VME Access

Another boards can only access the dual port memory aboce 0x87?02000.
The first 8KB are reserved for the Mailbox Interrupts.

The VME base address depends on the CPU board cpu number.
CPU 0 starts at VME address 0x87100000.
CPU 1 starts at VME address 0x87200000.

...

The address range is then:

from 0x87?02000 to 0x87?3FFFF for the 256KB version
from 0x87?02000 to 0x87?FFFFFF for the 1MB version

4.5.5 Mailboxes

Mailboxes are used for the communication between VM42/VM62 cpu boards.
Access to the Local Mailbox is done through the bit7 of the board control register (0xCD000005).
Access to the VME Mailbox is done at the board base adress: 0x87?00000.



4.6 WindView

The use of WindView 1.0.1. is tested with Tornado 1.0 running under Unix. (Solaris 2.5)

To include the time stamp driver and the needed initialisation for WindView, just include the next defines in your config.h file.

```
/* Windview and TIMESTAMP driver */  
  
#if TRUE  
  #define INCLUDE_INSTRUMENTATION  
  #define INCLUDE_WINDVIEW  
  #define INCLUDE_TIMESTAMP  
#endif
```



5. THE LIBMC680X0GNUVM4262.A LIBRARY

5.1 E2PROM

This library supports the VM42/VM62 onboard E2PROM.

The first 1kbit is reserved for PEP data.

The second kbit can be used for example for the storage of the VxWorks boot parameters.

The following structure is made for the PEP data part :

part named dummy	with offset 0 and	length 18 (36 bytes)
part named Profibus	with offset 18 and	length 2 (4 bytes)
part named Ethernet	with offset 20 and	length 3 (6 bytes)
part named Reserved	with offset 23 and	length 41 (82 bytes)
part named Bootline	with offset 64 and	length 64 (128 bytes)

The functions that are available in the normal BSP's are :

e2readEther	reads the VM42/VM62 Ethernet Address
e2readBootline	reads the stored bootline
e2readDummy	reads the stored info in the dummy part
e2readProfibus	reads the Profibus info from the e2prom
e2writeBootline	writes the Bootline to the e2prom
e2writeProfibus	writes the Profibus to the e2prom

inizSPI	Initialises the SPI Controller
seReSPI	Internal function : Send/Receive Bytes over SPI
selChip	select e2prom
deSelChip	deselects e2prom
rdE2	reads e2prom location
wrE2	writes e2prom location

The functions that only are available in PEP Internal BSP's are :

e2writeDummy	writes the Dummy info to the e2prom
e2writeEther	writes the Ethernet address to the e2prom
dumpE2	dumps the whole content of the e2prom



5.2 Real Time Clock

The library supports the VM42/VM62 onboard real time clock (V3021).

STATUS rtcInit()	Initialize the RTC access
STATUS rtcSet (struct tm *t)	Set Date and Time in UNIX like structure
STATUS rtcGet (struct tm *t)	Get Date and Time in UNIX like structure

5.3 Ethernet Driver

The VM42/VM62 supports the onboard Ethernet controller on SCC1. m360attach() attaches the VM42/VM62 Ethernet device.

The Ethernet address is read from the VM42/VM62 serial E2PROM. Only one Ethernet device is supported.

5.4 CPU Speed detection

A function is used to detect the MC68360 CPU clock speed.

The use of this function might strongly disturb the auxiliary and timestamp timers.

To avoid further call to that function, a global variable *cpuClkSpeed* is assigned with the calculated CPU speed.



6. BOOT FROM THE VME BUS

In order to boot your VM42/62 board from the VME bus, you need to build a new VxWorks boot image and operating system.

6.1 Build the needed bootrom image

Modify the default startup entry of the makefile for this purpose:

```
# Values for VME boot
ROM_TEXT_ADRS    = 87000008    # ROM entry address after reset
ROM_SIZE         = 00100000    # number of bytes of ROM space
```

and remake the needed bootrom.hex image file

make bootrom.hex

6.2 Build the needed OS

Modify the default startup entry of the config.h for this purpose:

```
/*
 * Warning: The following definitions must be kept
 * synchronized with the
 * Values in MakeSkel
 *
 * To build bootprom or resident version starting at the VME
 * base address, define VME_BOOT.
 * You can know put this image in a memory board and
 * configure the memory
 * board to start at VME address 0.
 */

#define VME_BOOT

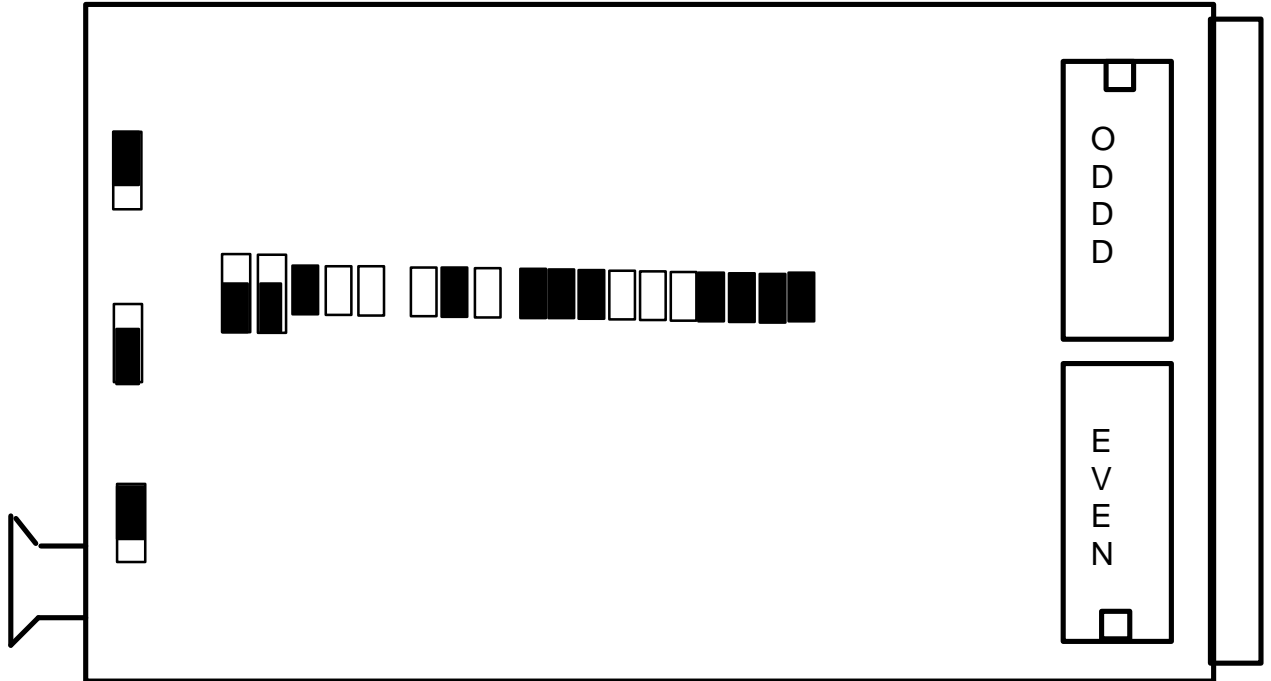
#ifdef VME_BOOT
    /* VME Base address */
    #define ROM_BASE_ADRS    0x87000000
#else
    /* base address of FLASH */
    #define ROM_BASE_ADRS    0x04000000
#endif
```

and run the **make** command again.

make



6.3 Configure your VMEM-S2



6.4 Configure the VM42/VM62

You need to set the jumper J8 in order to boot from the VME Bus.



6.5 Configure your VME BUS

Remove BG3 and IACK jumper on the first slot (for old VME buses with backplane jumpers)

Put the VM42/VM62 in the first slot.

Put the VMEM-S2 in the second slot.

Power up your system.

After a while (about 10 seconds), you will get the VxWorks Boot countdown.

Press any key to get the boot prompt.

6.6 Configure your boot parameters and boot

Use the 'c' command to configure your boot parameters

Use the '@' command to boot.