MCS8140

Network USB Processor

MCS8140 USB EVB User Manual

Revision 1.0

August, 2007

For Technical Support contact support-8140@moschip.com

Table of Contents

1. Scope

This Manual describes detailed operation and use of the MCS8140 USB Network Processor Board. This User Manual has to be used in conjunction with the MCS8140 Data Sheet, Reference Schematics & other Application Notes.

2. Introduction

This User Manual is intended for **MCS8140 USB EVB**, pictorial representation can be seen below.

MCS8140 EVB shown above can boot Linux and run automatically. This EVB can be used with USB-Server & Print Server applications. Evaluation boards from MosChip have USB Server Firmware build as default software, customers wishing to update the software or load Print Server application need to refer to Firmware Upgrade document for directions to upgrade the Firmware.

Following accessories shipped along with EVB by MosChip.

- Null Modem Serial Cable
- Ethernet Cable
- 7.5V DC / 4A Universal AC Power Adapter

3. Evaluation Board Key Features

- Based on ARM926EJ-S processor
- One Ethernet port
- Four USB 2.0 Host Controller ports
- 32 Megabytes of on-board SDRAM
- 8 Megabytes of on-board Flash
- RS-232 port, for Serial Console
- DC Socket for external wall adapter (7.5V DC/4A Adapter)
- Runs Linux Operating System
- 4 Layered board
- External Current Control circuit to limit the Current consumption per USB port under 500mA. (As per USB Spec)

4. Evaluation Board Mechanical

PCB Layout details of MCS8140-USB EVB like Allegro board file / Gerber details can be provided on demand. Write to support-8140@moschip.com if such data is needed.

5. Interfaces and Hardware Configuration

MCS8140 EVB Schematic (PDF & OrCad DSN files) can be provided on demand, write to support-8140@moschip.com.

6. Conventions

- Pin 1 of all headers and jumpers are marked with a small arrow
- All dual row headers have odd numbered pins on one side (side with pin 1 arrow) and even numbered pins on the other side
- An “X”, “#”, or an “N” at the end of a signal name signifies an active low signal

7. Default Configuration

For default operation, use jumper on JP9 & do not use jumper on JP8 and JP10. Picture shown below to indicates same

8. Connectors

Connectors used on EVB listed & briefly detailed in the table below. Unless otherwise noted all connectors follow industry standards as per naming convention

Table : Connector List

Name	Type	Description
P2	DB9	Male DB9 for Console
RJ1	RJ45	Wired Copper interface for Ethernet, MagJack e SI-40138 part used with integrated Magnetics
CN4	USB	USB A type Host port for USB Port 0
CN6	USB	USB A type Host port for USB Port 1
CN5	USB	USB A type Host port for USB Port 2
CN7	USB	USB A type Host port for USB Port 3
CN8	OTG	USB Mini AB Connector (Not Populated, its below CN4 on the EVB)
CN14	DC Power Jack	Eternal wall adapter - 7.5V DC/4A

9. Console Port

The Console Port signals are RS-232 levels on P2. P2 is a male DB9 connector with a null modem connection on it. It implements the RX and TX of the serial port signals only and is intended to be connected to a PC's serial port with a Null-MODEM cable. The Pin description of the Console Port (P2) is given below

Console Port P2 (RS-232 Levels)		
Pin	**Name**	**Description**
2	RX	Receive
3	TX	Transmit

10. Power

The Power to the Evaluation board can be supplied through DC Socket (CN14). Wall Adapter with **7.5V DC/4A** rating is needed for this. The center pin of DC Socket is positive.

11. Memory Controller

The memory controller interfaces between Flash, SDRAM.

11.1. Flash

ST Flash can contain Arm-Boot code, Linux image & Flash file system specific to application. ST Flash can be programmed in-system through Serial Console. ST Flash can be erased and reprogrammed with a new Arm Boot code and new Linux image.

11.2. SDRAM

There are two 128 Mb SDRAM chips in 16-bit format for a 32-bit SDRAM interface. The product part number for these chips is Micron 48LC8M16A2.

12. Connecting the EVB to Terminal program

The MCS8140 Evaluation Board boots Linux out-of-the-box. The board is preloaded with ARM-Boot and Linux image. A Null-MODEM serial cable should be connected between P2 of EVB and the COM Port of a Desktop PC. Use terminal emulator like Hyper Terminal (115200, 8, N, 1 with no flow control) as Serial Console. When power is applied to the board ARM-Boot code will start and then boot to Linux. Text from the ARM boot will appear first. If any key is hit, the board stops loading the code after the ARM boot code and Linux will not be loaded. See ARM boot Message below. Once Linux is booted there will be a login prompt. (See below) Type 'root' and enter on the terminal in order to login to Linux on the evaluation board. From this point tests can be run on the board.

12.1. NUPort Diagnostic Message

```
 ------------------------------------------------------------------
                            Arm-Boot
 ------------------------------------------------------------------

Build Date       : Mar 20 2007 - 16:01:45
Utility Version  : Release 3
Memory Range     : 00700000 -> 00730070
IRQ Stack        : 0085e3dc
FIQ Stack        : 0085f3dc
Eth1 address not set. Using default address: 00:50:C2:1B:7f:fe
Eth2 address not set. Using default address: 00:50:C2:1B:7f:fd
SDRAM size       : 32 MB
Flash size       : 16 MB
Environment      : Flash

Hit any key to stop autoboot:  3
```

12.2. Linux Boot

If no keys are hit, then the board will boot to Linux automatically. After Linux booting type root and then press Enter, at the command prompt of Serial Console. No password is required for this. MCS8140 EVB runs on Linux now, See screen dump below. The messages on your screen might differ slightly because of updated software.

```
01234567
------------------------------------------------------------------
                            Arm-Boot
 ------------------------------------------------------------------

Build Date       : Mar 20 2007 - 16:01:45
Utility Version  : Release 3
Memory Range     : 00700000 -> 00730070
IRQ Stack        : 0085e3dc
FIQ Stack        : 0085f3dc
Eth1 address not set. Using default address: 00:50:C2:1B:7f:fe
Eth2 address not set. Using default address: 00:50:C2:1B:7f: fd
SDRAM size       : 32 MB
Flash size       : 8 MB
Environment      : Flash

Hit any key to stop autoboot:  3

## Booting image at 1c060000 ...
## Copy image from flash 1c060000 to ram 00007fc0 ...
   Image Name:
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1415008 Bytes = 1381 kB = 1 MB
   Load Address: 00008000
   Entry Point:  00008000

   Verifying Checksum ... OK

Starting kernel ...

Uncompressing
Linux.....................................................................
.......... done, booting the kernel.
Linux version 2.6.15 (george@proliant) (gcc version 3.3.1) #20 Mon Mar 12 20:16:16 IST
2007
```

```
CPU: ARM926EJ-Sid(wb) [41069264] revision 4 (ARMv5TEJ)
Machine: Secure Communication Processor
Memory policy: ECC disabled, Data cache writeback
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
CPU0: D cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
Built 1 zonelists
Kernel command line: root=/dev/mtdblock3 rootfstype=jffs2
PID hash table entries: 256 (order: 8, 4096 bytes)
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 32MB = 32MB total
Memory: 29380KB available (2498K code, 435K data, 84K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
SCSI subsystem initialized
usbcore: registered new driver usbfs
usbcore: registered new driver hub
NetWinder Floating Point Emulator V0.97 (double precision)
NTFS driver 2.1.25 [Flags: R/O].
JFFS2 version 2.2. (C) 2001-2003 Red Hat, Inc.
Initializing Cryptographic API
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
pci_hotplug: PCI Hot Plug PCI Core version: 0.5
ttyMCS0 at I/O 0xff0dc000 (irq = 21) is a SCP serial port(16550)
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
NUPORT MAC eth0
Loaded prism54 driver, version 1.2
hostap_plx: 0.4.4-kernel (Jouni Malinen <jkmaline@cc.hut.fi>)
hostap_pci: 0.4.4-kernel (Jouni Malinen <jkmaline@cc.hut.fi>)
Flash chip probe: 800000 at 1c000000
scp_flash: Found 1 x16 devices at 0x0 in 8-bit bank
 Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
cfi_cmdset_0001: Erase suspend on write enabled
cmdlinepart partition parsing not available
Using SCP partition definition
Creating 4 MTD partitions on "scp_flash":
0x00000000-0x00040000 : "Nuport Diagnostic utility"
0x00040000-0x00060000 : "Enviroment"
0x00060000-0x001e0000 : "bZimage"
0x001e0000-0x00800000 : "UserFS"
usbmon: debugfs is not available
NET: Registered protocol family 2
IP route cache hash table entries: 512 (order: -1, 2048 bytes)
TCP established hash table entries: 2048 (order: 1, 8192 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
TCP reno registered
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
NET: Registered protocol family 15
ieee80211: 802.11 data/management/control stack, git-1.1.7
ieee80211: Copyright (C) 2004-2005 Intel Corporation <jketreno@linux.intel.com>
VFS: Mounted root (jffs2 filesystem).
Freeing init memory: 84K
2+0 records in
2+0 records out
376 inodes
4096 blocks
Firstdatazone=47 (47)
Zonesize=1024
Maxsize=268966912
Starting USB/IP Configuration  : eth0: 00:50:C2:1B:70:65
Link up  100 Mbps Full-duplex
```

```
[  OK  ]
Starting Logging Services    : [  OK  ]
Starting Mounting Misc FS    : [  OK  ]
Starting Networking          : [  OK  ]
Starting USB/IP              : [  OK  ]
Starting Embedded USB        : [  OK  ]
SIOCGIFFLAGS: No such device

SCP Flashdisk (type jffs2) assembled: Mon Mar 12 20:06:22 IST 2007

moschip login:
```

13. iBoot Program

MCS8140 has an internal boot program, which is hard coded in on-chip ROM and can't be modified. It acts as a fail-safe program if the MCS8140 flash gets accidentally erased. It has a TFTP Client program which enables downloading of files from a TFTP Server.

In order to boot from the Internal ROM (iBoot), power off the EVB, remove jumper from JP9 and short JP8 and JP10 with board jumpers.

Connect the console port of the EVB to PC Standard com port and power on the board. As soon as the EVB is powered, iBoot program appears on the Serial Console as shown below:

```
        ----------------------------------------------------------------
                        Internal Bootloader(iBOOT)
        ----------------------------------------------------------------

Build Date       : Sep  7 2006 - 15:39:48
Version          : 3.0
SDRAM size       : 32 MB
Flash size       : 8 MB

Hit any key to stop autoboot:  3
iBoot>>
```

Halt the iBoot program from the Serial Console by clicking on any key on the keyboard. (On the host PC where Serial Console is connected)

Type help in the iBoot prompt to view all the supported commands:

S No	Command	Description
1	help:	Displays the supported commands list.
2	reset	Soft reset
3	md	To display memory contents. Usage: md <address> [no.of bytes] ex: md 0x400a8100 200 This displays 200 bytes of memory starting from 0x400a8100. If number of bytes is not specified default 100 bytes will be displayed. Memory address range: Full register address space, RAM Address i.e. 0x00 to 32MB; Flash Address i.e. 10000000 to (10000000 + 8MB)
4	mw	To write into memory. Usage: mw[.b,.w,.l] <address> <value> [count] ex: mw 0x800000 0x100

		This will write 0x100 into 0x800000. If count is specified then the same value is writen into 'count' no.of bytes starting from 0x800000. Memory address range: Full register address range, RAM Address i.e 0x00 to 32MB. Not for writing into Flash.
5	go	To start application at the specified address Usage: Usage: go <address> ex: tftpboot 192.168.2.100 192.168.2.101 Nuport Diagnostic utility.bin 0x8000 This will get Nuport Diagnostic utility.bin image at 0x8000 address. Now do "go 0x8000" This command will jump to address 0x8000 and start executing the application (ex Nuport Diagnostic utility) loaded at 0x8000
6	eeinfo	To display contents of serial EEPROM.
7	eesetmac	To set the MAC address. Usage: eesetmac <eth_no> <MAC_addr> ex: eesetmac 0 11:22:33:44:55:66 This command will write eth0 mac id into the serial EEPROM.
8	erase	To erase flash. Usage: erase <start_sector> <end_sector> Ex: erase 0 1 This command will erase sectors 0,1.
9	cp.b	To write data into flash. Usage: cp.b <source_addr> <dest_addr> <no.of.bytes to write> Ex: cp.b 0x8000 0x10000000 0x40000 This command will copy 0x40000 bytes of data from 0x8000 to 0x10000000.
10	tftpboot	To boot image via network using TFTP protocol. Usage: tftpboot <ipaddr> <serverip> <filename> <load_addr> Ex: tftpboot 192.168.1.100 192.168.2.101 Nuport Diagnostic utility.bin 0x8000 This command will get the Arm-Bootimage from remote server(192.168.1.101)to address 0x8000.

13.1. Upgrading / Loading ARM Boot into On-board Flash

iBoot program allows downloading of new Boot Loader / ARM Boot on to the MCS8140 EVB. TFTP client program needs a TFTP server on the PC.

Make sure that EVB is powered off.
JP8 and JP10 need to be jumper-ed and jumper on JP9 to be removed.

Erase the Boot section of Flash by following command:

```
iBOOT>> erase 0 1

Erase Flash from 0x10000000 to 0x10020000...
Erasing flash sector  0 ... Sect  0 done.
Erasing flash sector  1 ... Sect  1 done.
Done.
```

Copy TFTP Server application into Windows XP PC (To which MCS8140 EVB is connected through Ethernet cable). Ensure that PC Ethernet Cards IP address to be 192.168.3.XXX (XXX can be 01 to 254). Default IP address of MCS8140 EVB is 192.168.3.22

Copy all required binary Firmware build files (like Boot loader, Linux Kernel image & Flash file system of concerned application) onto Windows XP PC @ the same path where TFTP Server is copied.

Open TFTP server in Windows XP PC & set the path by pointing to correct path. Refer to Firmware Upgrade document for steps to use TFTP Server on Windows PC.

Use following command to copy required build onto SDRAM :

iBOOT>> tftpboot <ipaddr> <serverip> <filename> 0x8000

Serverip = Windows PC IP Address (192.168.3.XXX)

IPaddr = MCS8140 EVB IP Address (192.168.3.22)

Filename = file name of Boot loader (Eg: Nuport.bin)

```
iBOOT>> tftpboot 192.168.3.33 192.168.3.22 Nuport.bin 0x8000

Using port: EthA
RX Buf shift state: 0x2b
Eth0 address not set. Using default address: 00:50:C0:1B:7F:02
ARP broadcast 1
eth addr: 00:e0:4c:fe:a3:65
TFTP from server 192.168.1.101; our IP address is 192.168.1.100
Filename 'nuport.bin'.
Load address: 0x8000
Loading: ########################################
done
Bytes transferred = 197432 (30338 hex)<filesize>
iBoot>>
```

Now copy the *Nuport.bin* from SDRAM to Flash, with following command :

```
iBOOT>> cp.b 0x8000 0x10000000 0x30308
Copy from 0x00008000 to 0x10000000
Copy to Flash.........................done.
iBoot>>
```

Note : 0x30308 is the file size of Nuport.bin (This is the updated ARM Boot)

Now Power off the MCS8140-USB EVB, remove jumpers from JP8 & JP10 and replace the jumper on JP9.

Power on the EVB, following would be seen in the Serial Console now:

```
------------------------------------------------------------------
                    NUPort Diagnostic Utility
------------------------------------------------------------------

Build Date        : Apr  3 2007 - 12:25:56
Utility Version   : QA Release 3 for PCI,MAC,I2S,ISO,MEM2MEM DMA,FILIERS
Memory Range      : 00700000 -> 00730338
IRQ Stack         : 0085e3dc
FIQ Stack         : 0085f3dc
SDRAM size        : 32 MB
Flash size        : 16 MB
Environment       : Flash

Hit any key to stop autoboot:  0
## Booting image at 1c060000 ...

NUPort>>
```

14. The Nuport Diagnostic Utility

14.1. Setting MAC Address

The MCS8140 Evaluation Board will come with pre-set MAC Address for Ethernet port. The MAC Address is stored in SPI EEPROM used on the EVB. MAC Address can be changed through Serial Console or by using Windows MP Utility, for MCS8140 EVB.

14.2. For Serial EEPROM

Contents of the serial EEPROM can be viewed by typing eeinfo at the Arm-Boot prompt. See below.

```
# eeinfo

Dumping EEPROM content

0x0000: 00 50 c2 19 c0 04 ff ff ff ff ff ff ff ff ff ff
0x0010: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0020: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0030: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0040: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0050: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0060: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0070: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0080: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x0090: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x00a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x00b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x00c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x00d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
0x00e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ................
```

MAC address starts at 0x100, but they will also be stored starting at 0x000.

To set MAC address use ‘eesetmac’ command at the Arm-Boot prompt. Following line sets the MAC address to 0050c219c0b8

```
Nuport>> eesetmac 0 00:50:c2:19:c0:b8
```

Type help eesetmac for more information. See below for terminal output.

```
NUPort>> help eesetmac
eesetmac eesetmac mac addr
mac  - MAC number=0
addr - mac address in format: xx:xx:xx:xx:xx:xx
example: eesetmac 0 11:22:33:44:55:66
```

14.3. For Flash memory

MAC Address can also be stored in the ST Flash (U10) by using 'setenv' command. It is stored as an environment variable, if the Serial EEPROM does not contain the MAC Address then it uses the MAC Address stored in the ST Flash. To set MAC Address in Flash type the following:

```
Nuport>> setenv ethaddr0 00:50:c2:1b:7x:xx
Nuport>> saveenv
```

15. Boot Loader Environment Variables

Boot loader environment variables are listed in the table below.

Table: Boot Loader Environment Variables

Name	Value	Description
bootparams	address	If this variable is set then, before booting linux, the boot loader builds Linux TAG list containing memory information and command line starting at that address. Do not use value 0. Optional.
bootargs	string	That string will be passed to linux kernel as command line if *bootparams* variable is set. Optional.
serverip	Ipaddress	TFTP server IP address (must be set for *tftpboot*). If serverip variable is set then *tftpboot* uses this value instead of value provided by BOOTP server.
ipaddr	Ipaddress	Board own IP address (must be set for *tftpboot*). Mandatory for *tftpboot.*
ethaddr	MACaddr	Specifies hardware address when using *tftpboot*. If not set, the board will search hardware address from serial EEPROM.
bootpaddr	MACaddr	Specifies hardware address for *bootp* communication. BOOTP server uses hardware address to select proper profile. Mandatory for *bootp*. Default is 11:11:11:11:11:11.
baudrate	integer	Selects baud rate for console. Allowed values are 9600, 19200, 38400, 57600, and 115200.
bootcmd	string	This variable contains default boot command that will be automatically executed after auto boot timeout. To enter multi command string as bootcmd use '\' (slash) before ';'.
Boot delay	integer	Auto boot delay in seconds.

15.1. Setting Environment Variables

There are a series of environment variables that can be set, viewed, and saved. The printenv command displays a list of the environment variables and their associated settings.

```
Nuport>> printenv

bootcmd=bootmf; bootp; bootm
bootdelay=3
baudrate=115200
ethaddr=11:11:11:11:11:11

Environment size: 100/131068 bytes

Nuport>>
```

The setenv command temporary changes the variables. See below.

setenv bootcmd <action> (sets the boot command action on reset)
setenv bootcmd bootmf\; bootp\; bootm

After reset the Arm-Boot will attempt to boot a Linux image from Flash (bootmf). If an image is not present it will then initiate a tftp. After an image is loaded using tftp it is then executed from memory (bootm). If a bootp server and file is not found the board will not boot.

setenv bootdelay <delay> (sets the boot delay after reset)

Note: ***This variable is initially set to 3 seconds, if this variable is changed to 0 or erased, then a the Linux booting sequence can only be interrupted with a "ctrl+c". to interrupt the Linux boot. Now, the user can change this variable back to 3 seconds or copy the new Arm-Boot back onto the ST flash.***

setenv baudrate <rate> (sets the BAUD rate of the serial port)
setenv ethaddr <MAC address> (sets the MAC address for bootp, not for Linux)

Other examples of variables:

setenv serverip <IP Address> (sets the IP address of the server, used for tftpboot)

setenv ipaddr <IP Address> (sets the IP address of the boot port, used for tftpboot)

setenv bootfile <filename>(sets the file name for tftpboot command)
This filename will override the file name used on the bootp server. If no argument (file name) is used after bootfile, then the file used will be controlled by the bootp server as described in the Installation Section of this manual.

15.2. Storing Environment Variables

The ‘saveenv’ command will store these values to the Flash so they will not be lost on power down.

16. Loading/Upgrading Firmware into Flash

This topic covered with complete details in Firmware upgrade manual. Please refer to same for detailed steps to execute this task.

17. Additional Information

Write to support-8140@moschip.com for any further queries on MCS8140 or on this document.

Important Notice

All information in this document is believed to be accurate as of the publish date.

ARM is a registered trademark of ARM Limited. All other brands or product names are the property of their respective holders.

MosChip Semiconductor products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of MosChip Semiconductor, Ltd.

MosChip Semiconductor believes the information in this document to be accurate and reliable. However, it is subject to change without notice. MosChip Semiconductor assumes no responsibility for its use, or for infringement of patent or other rights of third parties. No part of this document may be reproduced, or transmitted in any form or by any means without prior consent of MosChip Semiconductor, Ltd.