MCS8140

Network USB Processor

MCS8140 Combo EVB -User Manual

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. No responsibility is assumed by MosChip Semiconductor for its use, nor 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.

Table of Contents

SCOPE 3

1. INTRODUCTION 3

2. EVALUATION BOARD KEY FEATURES 3

3. EVALUATION BOARD MECHANICAL 4

4. INTERFACES AND HARDWARE CONFIGURATION 4

5. CONVENTIONS 4

6. DEFAULT CONFIGURATION 4

7. CONNECTORS 6

8. CONSOLE PORT 6

9. SERIAL PORT 7

10. POWER 7

11. HEADERS 7

11.1 Header Configuration Details 8
11.1.1 GPIO Header (J3) 8
11.1.2 JTAG Header (CN12) 8
11.1.3 Address Data Control Bus (CN3) 9

12. SWITCH LIST 10

13. BOOTSTRAP CONFIGURATION 11

14. MEMORY CONTROLLER 12

14.1 Flash 12

14.2 SDRAM 12

15. CONNECTING THE EVB TO TERMINAL PROGRAM 12

15.1 NUPort Diagnostic Message 13

15.2 Linux Boot 13

16 IBOOT PROGRAM 17

16.1 Upgrading / Loading ARM Boot into U10 through iBoot 18

17 THE NUPORT DIAGNOSTIC UTILTITY 20

17.1 Setting MAC Address 20
17.1.1 For Serial EEPROM 20
17.1.2 For Flash memory 21

18 BOOT LOADER ENVIRONMENT VARIABLES 21

18.1 Setting Environment Variables 21

18.2 Storing Environment Variables 22

19 LOADING / UPGRADING FIRMWARE INTO ST FLASH (U10) 23

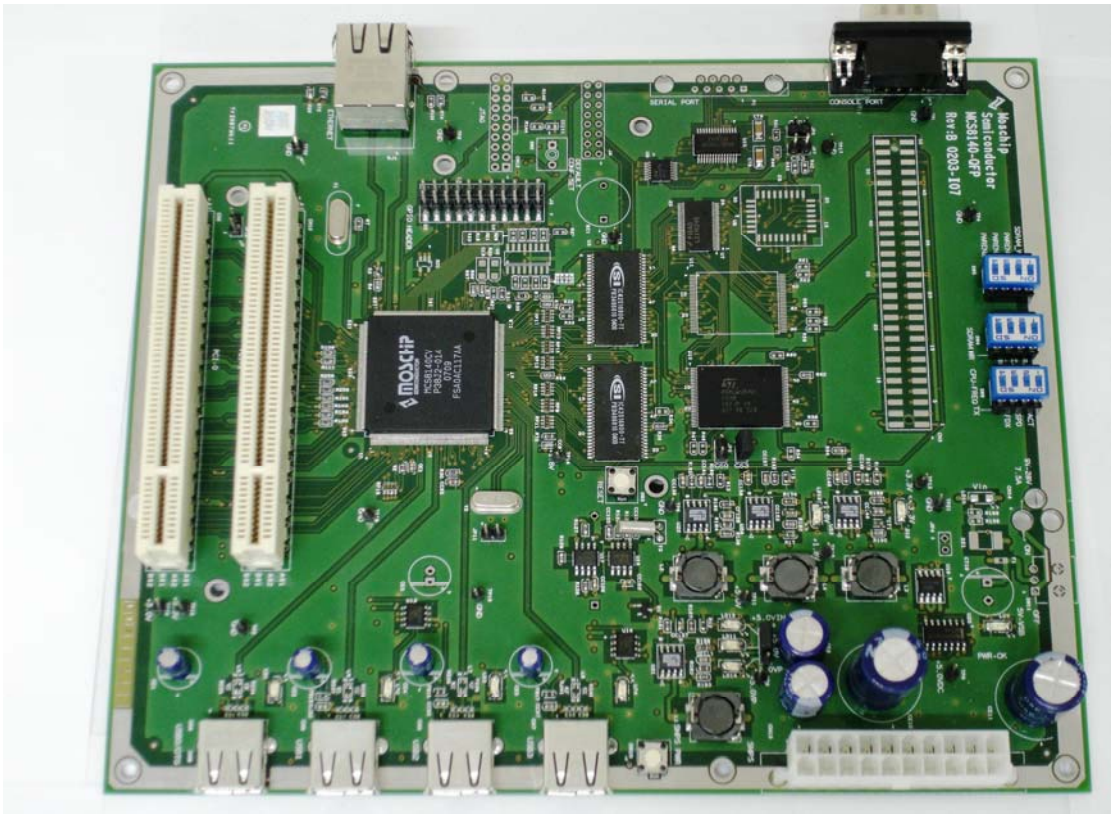
20 SUPPORT 23

Scope

This Manual describes detailed operation and use of the MCS8140 Combo EVB. This User Manual to be used in conjunction with the MCS8140 Data Sheet, Reference Schematics & other application Notes.

1. Introduction

The MCS8140 Combo EVB is shown below. It can boot Linux and run automatically. Evaluation board from MosChip are being sent out with USB-Server SW loaded. Refer to USB-Server user Manual for details on how to use USB-Server application loaded on MCS8140 EVB.



2. Evaluation Board Key Features

- Runs in standalone mode
- Based on ARM926EJ-S processor
- One Ethernet port
- Four USB 2.0 Host Controller ports
- Two 32 bit 33 MHz PCI Host Slots
- 32 Megabytes of on-board SDRAM
- 16 Megabytes of on-board Flash, 8MB x 2 Organization
- RS-232 port, for Serial Console
- On-board ATX Connector for Power Input
- Runs Linux Operating System
- Hardware-based cryptography and authentication

3. Evaluation Board Mechanical

PCB Layout details of MCS8140 EVB like Allegro board file / Gerber details can be provided on demand, write to support-8140@moschip.com

4. Interfaces and Hardware Configuration

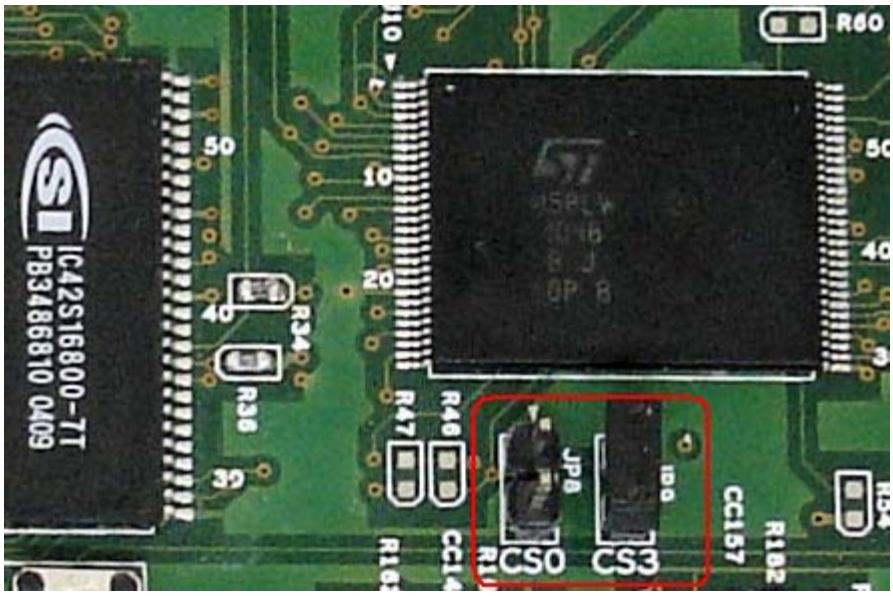
MCS8140 EVB Schematic (OrCad DSN file & PDF file) can be provided on demand, write to support-8140@moschip.com

5. Conventions

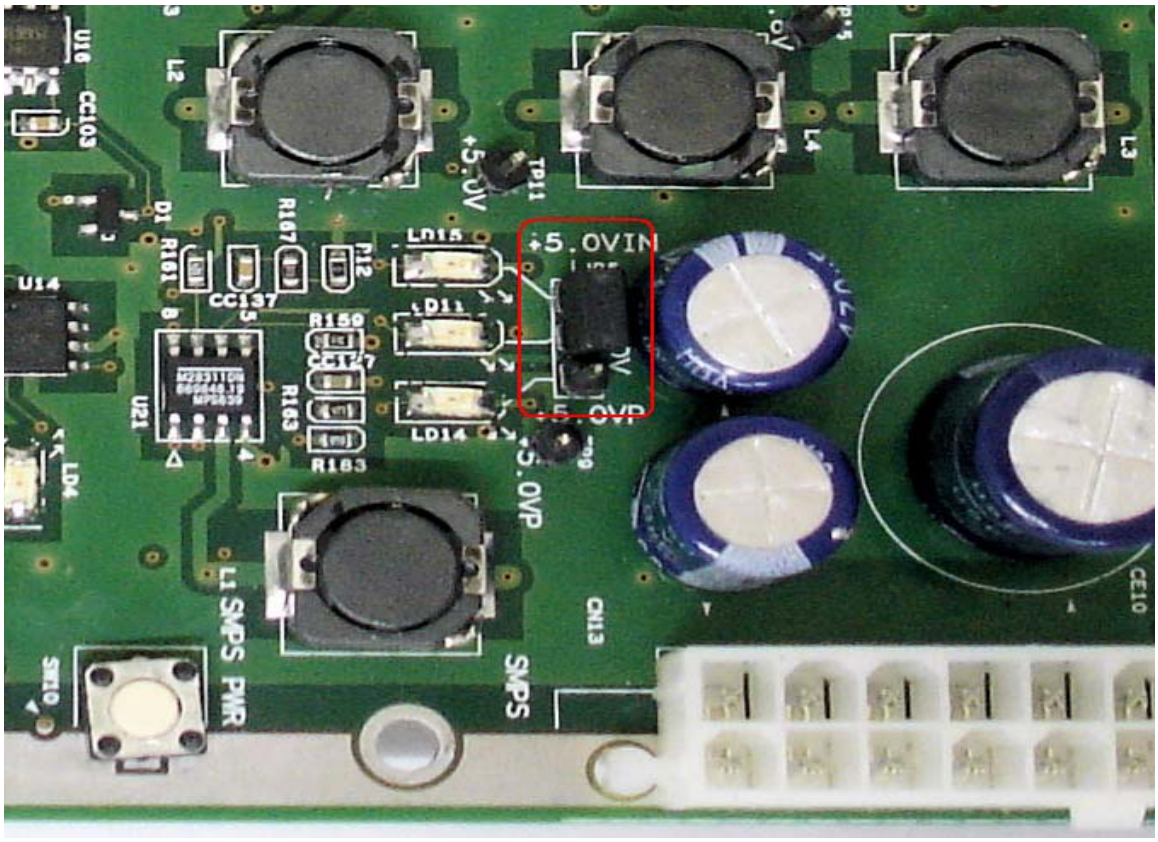
- Pin 1 of all headers and jumpers are marked with a small arrow
- All dual row headers have odd numbered pins on one side (side with pin 1 arrow) and even numbered pins on the other side
- An "X", "#", or an "N" at the end of a signal name signifies an active low signal

6. Default Configuration

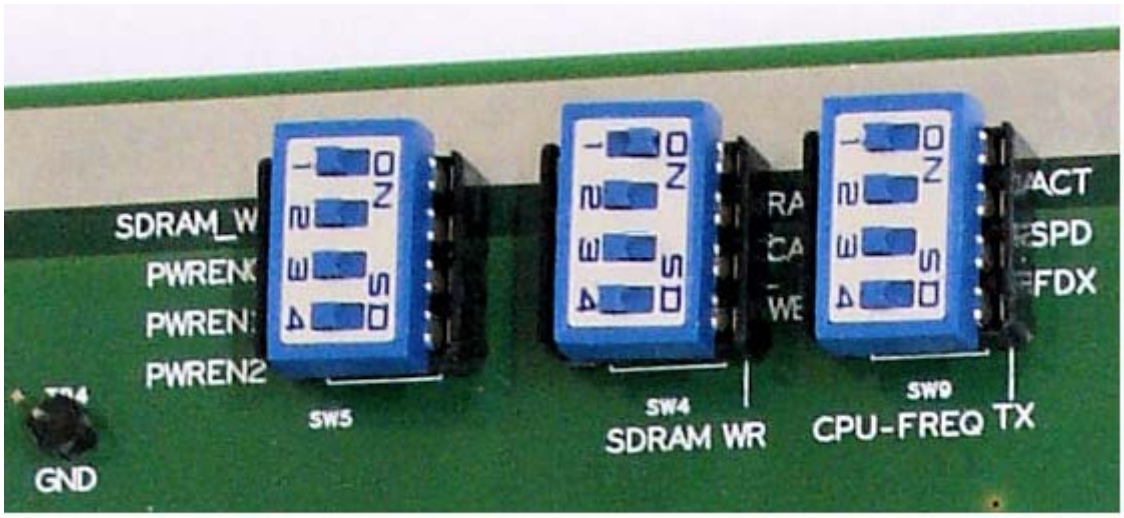
- Use Jumper on JP9 & do not use Jumper on JP8, for default operation. Picture shown below to indicate same



- JP5 pins 1 and 2 to be connected with Jumper for powering the USB & PCI from Regulated Supply



- All Boot Strap DIP switches (SW4, SW5 & SW9) into OFF condition except for the switch#1 of SW4. Switch#1 of SW4 to be set to ON position



7. 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
P1	DB9	Male DB9 for Serial port (**P1 Not Populated on EVB**)
RJ1	RJ45	Copper interface for Ethernet, MagJack make 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. To populated CN8, CN4 needs to be removed)
CN9	PCI Connector	32bit, 33Mhz PCI Host Adapter connector for PCI target devices
CN10	PCI Connector	32bit, 33Mhz PCI Host Adapter connector for PCI target devices
CN13	ATX CONN	Standard ATX connector plug-in to connect any standard ATX SMPS Supply
CN14	DC Power Jack	9V to 28V DC input (**CN14 Not populated on EVB**). Current rating of DC Power source depend on the load used on PCI slots. For EVB usage without PCI loaded, 5Amps rated Power source can be used. For PCI based applications, each PCI slot need to provide 3 Amps current as per PCI Spec. When PCI & USB are used together, use minimum of 7.5Amps rated Power source.

8. 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

9. Serial Port

The Serial Port signals are RS-232 levels on P1. P1 is a Male DB-9 connector and is intended to be connected to an external Modem. The pin description of the Serial Port (P1) is given below.

Table : Serial Port P1 (RS-232 Levels)		
Pin	**Name**	**Description**
1	CD	Data Carrier Detect
2	RX	Receive
3	TX	Transmit
4	DTR	Data Terminal Ready
5	Ground	Ground
6	DSR	Data Set Ready
7	RTS	Ready To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

Note : When Console port (P2) is selected, Serial Port(P1) is not available. Alternately when Serial Port(P1) gets selected, Console port is not available. These selections are made though Boot strap pins.

10. Power

The Power to the Evaluation board can be supplied either through DC Socket (CN14) or through the ATX connector (CN13). **Use ATX connector (CN13) to Power the EVB using any Standard PC ATX switching power supply module. (Default setting)**

Wall transformer can also be used in case EVB needs to be powered from DC Socket (CN14). Wall transformer of 9V to 28V, 7.5Amps needed for this purpose. The center pin of DC Socket is positive. DC Socket (CN14) not populated in current Evaluation boards, EVB has provision for DC Socket addition.

JP5 Jumpers to be set for power source selection, as below:

JP5 1-2: Connect Jumper for ATX Power : **Default**
JP5 2-3: Do not connect Jumper

11. Headers

The headers and the associated signals are listed in the tables below.

Table : Header List		
Name	**Type**	**Description**
CN12	Through Hole 10x2 (20 pin)	JTAG (Not Populated on EVB)
J3	SMT header 11x2 (22Pin)	GPIO
CN3	SMT header 28x2 (56 pin)	ADC Bus (Not Populated on EVB)
J4	Through Hole 8x2 (16Pin)	Reserved header (Not Populated on EVB)

11.1 Header Configuration Details

All header pins are identified on the board and on the schematic.

11.1.1 GPIO Header (J3)

Pin	Name	Description
1	3.3 V	3.3 Volts Power
2	GPIO_0	General Purpose I/O bit 0
3	GPIO_1	General Purpose I/O bit 1
4	GPIO_2	General Purpose I/O bit 2
5	GPIO_3	General Purpose I/O bit 3
6	GPIO_4	General Purpose I/O bit 4
7	GPIO_5	General Purpose I/O bit 5
8	GPIO_6	General Purpose I/O bit 6
9	GPIO_7	General Purpose I/O bit 7
10	GPIO_8	General Purpose I/O bit 8
11	GPIO_9	General Purpose I/O bit 9
12	GPIO_10	General Purpose I/O bit 10
13	GPIO_11	General Purpose I/O bit 11
14	GPIO_12	General Purpose I/O bit 12
15	GPIO_13	General Purpose I/O bit 13
16	GPIO_14	General Purpose I/O bit 14
17	GPIO_15	General Purpose I/O bit 15
18	GPIO_16	General Purpose I/O bit 16
19	GPIO_17	General Purpose I/O bit 17
20	GPIO_18	General Purpose I/O bit 18
21	GPIO_19	General Purpose I/O bit 19
22	GND	Ground

11.1.2 JTAG Header (CN12)

Pin	JTAG Pin Name	Description
1	3.3 V	3.3V
2	3.3 V	3.3 V
3	nTRST	Connected to nTRST of MCS8140
5	TDI	Connected to GPIO18 of MCS8140
7	TMS	Connected to GPIO16 of MCS8140
9	TCK	Connected to GPIO17 of MCS8140
11	RTCK	Pulled high to 3.3 V through 10K resistor
13	TD0	Connected to GPIO19 of MCS8140
15	nSRST	Connected to RST_IN of MCS8140
17	NC	
19	NC	
4,6,8,10,12,14,16,18,20		Ground

11.1.3 Address Data Control Bus (CN3)

Table: Address Data Control Bus CN3 (ADC Bus)

Pin	Name	Description
1	3.3 V	3.3 Volts
2	3.3 V	3.3 Volts
3	SDRAM_MA0	Address Bus Bit 0
4	SDRAM_MA1	Address Bus Bit 1
5	SDRAM_MA2	Address Bus Bit 2
6	SDRAM_MA3	Address Bus Bit 3
7	SDRAM_MA4	Address Bus Bit 4
8	SDRAM_MA5	Address Bus Bit 5
9	SDRAM_MA6	Address Bus Bit 6
10	SDRAM_MA7	Address Bus Bit 7
11	SDRAM_MA8	Address Bus Bit 8
12	SDRAM_MA9	Address Bus Bit 9
13	SDRAM_MA10	Address Bus Bit 10
14	SDRAM_MA11	Address Bus Bit 11
15	SDRAM_MA12	Address Bus Bit 12
16	SDRAM_MA13	Address Bus Bit 13
17	SDRAM_MA14	Address Bus Bit 14
18	SDRAM_MA15	Address Bus Bit 15
19	SDRAM_MA16	Address Bus Bit 16
20	SDRAM_MA17	Address Bus Bit 17
21	SDRAM_MA18	Address Bus Bit 18
22	SDRAM_MA19	Address Bus Bit 19
23	SDRAM_MA20	Address Bus Bit 20
24	SDRAM_MA21	Address Bus Bit 21
25	SDRAM_MA22	Address Bus Bit 22
26	SDRAM_MA23	Address Bus Bit 23
27	SDRAM_MA24	Address Bus Bit 24
28	SDRAM_MA25	Address Bus Bit 25
29	GND	Ground
30	GND	Ground
31	ADC_D0	Data Bus Bit 0
32	ADC_D8	Data Bus Bit 8
33	ADC_D1	Data Bus Bit 1
34	ADC_D9	Data Bus Bit 9
35	ADC_D2	Data Bus Bit 2

Table: Address Data Control Bus CN3 (ADC Bus)

Pin	Name	Description
36	ADC_D10	Data Bus Bit 10
37	ADC_D3	Data Bus Bit 3
38	ADC_D11	Data Bus Bit 11
39	ADC_D4	Data Bus Bit 4
40	ADC_D12	Data Bus Bit 12
41	ADC_D5	Data Bus Bit 5
42	ADC_D13	Data Bus Bit 13
43	ADC_D6	Data Bus Bit 6
44	ADC_D14	Data Bus Bit 14
45	ADC_D7	Data Bus Bit 7
46	ADC_D15	Data Bus Bit 15
47	GND	Ground
48	GND	Ground
49	ADC_CS0	Chip Select 0 (active low)
50	ADC_CS2	Chip Select 2 (active low)
51	ADC_CS1	Chip Select 1 (active low)
52	ADC_CS3	Chip Select 3 (active low)
53	ADC_CLK	Clock
54	RST_OUT	Reset generated from MCS8140 (active low)
55	ADC_RD	Read (active low)
56	ADC_WR	Write (active low)

12. Switch List

Switch List		
Name	**Type**	**Description**
SW11	Toggle Switch	ON/OFF Toggle switch for DC power Input (Not Populated on current EVB)
SW3	Push button	System Reset
SW10	Push button	ATX Power Supply ON/OFF. **This switch needs to be operated for powering the board with each iteration of power re-cycle**

13. Bootstrap Configuration

The processor detects the bootstrap levels during power-up. There are 12 bootstrap pins on the Evaluation board. The bootstrap can be pulled high or low by setting the DIP Switch buttons

Bootstrap Switch SW5				
SW Pin #	**Pin Name**	**Default switch setting**	**Internal bonding of this pin in MCS8140**	**Description**
1	SDRAM_WE	OFF	Internally pulled to High	Reserved for internal test purpose. Set the switch to "OFF" for normal operation
2	PWREN0	OFF	Internally pulled to High	Reserved for internal test purpose. Set the switch to "OFF" for normal operation
3	PWREN1	OFF	Internally pulled to High	Pin 4, Pin 3: 11 : Normal mode (Default setting) 10 : Ext MII 01 : Serial Mode 00 : Reserved
4	PWREN2	OFF	Internally pulled to High	

Bootstrap Switch SW4				
SW Pin #	**Pin Name**	**Default switch setting**	**Internal bonding of this pin in MCS8140**	**Description**
1	SDRAM-RAS	ON	Pulled High	This pin is used to Disable (Default setting) or Enable "Invert Bus Power". This settings needs to be changed based on current controller chip used (EVB using MIC2076-2BM)
2	ADC-CS2	OFF	Pulled High	This pin is used to Disable (Default setting) or Enable "iBoot"
3	SDRAM-M-CS1	OFF	Pulled High	This pin is used to Disable (Default setting) or Enable "OTG Mode"
4	ADC-RD	OFF	Pulled High	This pin is used to select SDRAM clock speed High : 100MHz SDRAM Clock (Default setting) Low : 50Mhz SDRAM Clock

Bootstrap Switch SW9				
Pin 1	**Pin 2**	**Pin 3**	**Pin 4**	**CPU Frequency**
LED-ACT	LED-SPD	LED-FDX	SP-TX	
OFF	OFF	OFF	OFF	162.5 MHz (Default setting)
OFF	OFF	ON	OFF	Reserved
OFF	OFF	OFF	ON	170 MHz
ON	OFF	OFF	OFF	150 MHz

14. Memory Controller

The memory controller interfaces between Flash, SDRAM and the ADC Bus.

14.1 Flash

Two Flash devices populated on EVB, namely U10 & U11. U8 is the third Flash on EVB, layout provision given to use (U8) 256KB Atmel Flash in a PLCC package. U8 not populated on EVB Assy.

ST Flash can contain Arm-Boot code, Linux image & Flash file system specific to application. ST Flash is the only device that can be programmed in-system. ST Flash can be erased and reprogrammed with a new *Arm Boot code* and new Linux image

14.2 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.

15. 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.

15.1 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
```

15.2 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       : 16 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
mcs8140_pci_init
pci_mcs8140_preinit
PCI Master Interface Config is 1070000
pci_mcs8140_setup
eeprom_emu is done
MCS8140# PCI core found
mcs8140 scan bus start
Ignore :PCI Slot is 0
PCI Data abort: address = 0xe8000800 fsr = 0x008 PC = 0xc0028dc8 LR =
0x00000001
PCI FATAL ERROR
Increased Program Counter
Returning From Data Abort Handler
Master Abort Interrupt 1070100
Master Abort Cleared on PCI BUS 1070100
PCI Data abort: address = 0xe8001000 fsr = 0x008 PC = 0xc0028dc8 LR =
0x00000002
PCI FATAL ERROR
Increased Program Counter
Returning From Data Abort Handler
Master Abort Interrupt 1070100
Master Abort Cleared on PCI BUS 1070100
PCI Data abort: address = 0xe8001800 fsr = 0x008 PC = 0xc0028dc8 LR =
0x00000003
PCI FATAL ERROR
Increased Program Counter
Returning From Data Abort Handler
Master Abort Interrupt 1070100
Master Abort Cleared on PCI BUS 1070100
Ignore :PCI Slot is 4
Ignore :PCI Slot is 5
Ignore :PCI Slot is 6
```

```
Ignore :PCI Slot is 7
Ignore :PCI Slot is 8
Ignore :PCI Slot is 9
Ignore :PCI Slot is a
Ignore :PCI Slot is b
Ignore :PCI Slot is c
Ignore :PCI Slot is d
Ignore :PCI Slot is e
Ignore :PCI Slot is f
Ignore :PCI Slot is 10
Ignore :PCI Slot is 11
Ignore :PCI Slot is 12
Ignore :PCI Slot is 13
Ignore :PCI Slot is 14
Ignore :PCI Slot is 15
Ignore :PCI Slot is 16
Ignore :PCI Slot is 17
Ignore :PCI Slot is 18
Ignore :PCI Slot is 19
Ignore :PCI Slot is 1a
Ignore :PCI Slot is 1b
Ignore :PCI Slot is 1c
Ignore :PCI Slot is 1d
Ignore :PCI Slot is 1e
Ignore :PCI Slot is 1f
PCI: bus0: Fast back to back transfers enabled
pci_mcs8140_postinit
mcs8140_pci_init
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>
jffs2_scan_eraseblock(): Magic bitmask 0x1985 not found at 0x00546154:
0xa85f instead
jffs2_scan_eraseblock(): Magic bitmask 0x1985 not found at 0x00546158:
0xaa0d instead
jffs2_scan_eraseblock(): Magic bitmask 0x1985 not found at 0x0054615c:
0x870a instead
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:
```

16 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), set Dip Swich 2 on SW4 to ON Position (By default it is set to OFF position). Once iBoot is selected, remove the jumper on JP9 and place the same on JP8. Above settings to be done in power off condition of EVB.

Now power the EVB by connecting Null Modem cable to Serial Console Port. 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.

16.1 Upgrading / Loading ARM Boot into U10 through iBoot

iBoot program allows downloading of new firmware images on to the MCS8140 EVB. TFTP client program needs a TFTP server running on the PC.

Make sure that External Flash is connected on CS0# region by placing jumper on JP8. Remove Jumper on JP9. (SW4 DIP switch#2 to be ON)

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 cross Over 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 Section 19, on how 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 (30308 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 : 0x30338 is the file size of Nuport.bin

Now Power of the MCS8140 EVB, set the Dip Swich 2 on SW4 to OFF position to disable iBoot. Remove Jumper on JP8 & Place 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,TSO,MEM2MEM DMA,FILTERS
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>>
```

17 The Nuport Diagnostic utiltity

17.1 Setting MAC Address

The MCS8140 Evaluation Board will come with pre-set MAC Address for Ethernet port. The MAC Address is stored in EEPROM used on the EVB. MAC Address can be changed through Serial Console only, for MCS8140 EVB.

17.1.1 For Serial EEPROM

Contents of the serial EEPROM can be viewed by typing *eeinfo* at the Arm-Bootprompt. 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
```

17.1.2 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
```

18 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 ';'.
bootdelay	integer	Auto boot delay in seconds.

18.1 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-Bootback 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 the bootp of 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.

18.2 Storing Environment Variables

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

19 Loading / Upgrading Firmware into ST Flash (U10)

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

20 Support

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