LAN93xx 10/100 ETHERNET SWITCH

ENGINEERING RELEASE NOTES OF SOFTWARE VER 2.0

1 INTRODUCTION

This document describes the software contents released in **2.0**. This release contains new features and bug fixes from the previous production release of **1.2.2**

<u>Chip-set support</u>

LAN9311	- Managed-16 bit mode with Host Bus Interface.
LAN9312	- Managed-32 bit mode with Host Bus Interface.
LAN9313	- Managed - with SPI, SMI, I2C serial interfaces.
LAN9303	- Managed - with SPI, SMI, I2C serial interfaces.
LAN89303	- Managed - with SPI, SMI, I2C serial interfaces.

<u>Platforms support</u>

- Engineering- silicon and FPGA Board
  - ➢ The Host processor is SH36417709S and the Host O.S is Linux 2.6.16
- Customer Board Rev A
  - ➢ The Host processor is SH36417709S and the Host O.S is Linux 2.6.16

2 NEW FEATURES AND BUG FIXES.

2.1 New Features

- IPv6 Multicast Listener Discovery – MLDv2 (LAN89303 Only).
- Programmatic board configuration (Engineering boards only).
- Per-port MAC Addresses for Pause Frames.
- IPv6 checksum and hop-limit validation for MLD packets.
- mld.on script to active MLD protocol (LAN89303 Only).

2.2 Bug Fixes

No.	Component	BugZilla ID	Status	Explanation
1	scripts	5804	FIXED	Build Package for LAN9303 Family does not grab LeoLite scripts
2	scripts	2952	FIXED	VLAN.op test case 4.1 fails during Part 1
3	scripts	2953	FIXED	VLAN.op test case 2.5 fails with assertion error
4	scripts	2955	FIXED	VLAN.io.1.1 test case fails with assertion failure
5	scripts	2958	FIXED	Test Case RSTP.op.1.4 fails with multiple insmod errors
6	scripts	2304	FIXED	Formatting (CR/LF), permission errors in revised config scripts
7	RSTP	3142	FIXED	LeoLite::RSTP does not fwd BPDUs in learning state, protocol ver. 0
8	SH3 Host Ethernet Driver	2836	FIXED	LeoLite::Assertion Failure when writing DiffServ table
9	RSTP	3424	FIXED	LeoLite: UNH TCN tests failing
10	IGMP-MLD	5629	FIXED	snoop.on script needs extra command
11	scripts	2957	FIXED	QOS.1.4 test case fails with assertion failure

2.3 Usage Notes of new features

- Linux shell script mld.on is used to start MLDv2 on **LAN89303 only**. Simply type "mld.on" with no arguments at the command prompt.

The following insmod arguments (smsc931x.ko) apply only to LAN9303 and LAN89303.

- The option **pauseMacAddrEnbl** is used to enable separate MAC addresses for pause frames, so they will have unique addresses on each port. Default is 0x1 (enabled).

- The option **enblDAPriority** is used to enable destination Mac address priority. Default is 0x0 (disabled).

- The option **icmpv6CksumValidate** is used to enable ICMPv6 checksum validation on MLD packets. . Default is 0x0 (disabled).

- The option **ipv6HoplimitValidate** is used to enable IPv6 Hop Limit validation for incoming MLD packets. . Default is 0x0 (disabled).

- The option **fp_lan930x_access_if** is used to indicate whether the chip is FPGA or silicon. Default is 0x0 (silicon), 0x1 indicates FPGA.

  For each of the above, a value of 0x1 enables the feature, while 0x0 disables it.

The next applies to LAN93x3 Engineering boards only (FPGA and Silicon):

- The option **strap_cfg** is used to programmatically configure the EVB for MAC or PHY mode, and to configure the management interface for I2C, SMI, or SPI.

Value	**Mode**	**Mgmt Interface**
0x01	MAC	SMI
0x02	MAC	I2C
0x03	MAC	SPI
0x04	PHY	SMI
0x05 (default)	PHY	I2C
0x06	PHY	SPI

3 RELEASE PACKAGE CONTENTS

The 2.0 binary package provides following images:

LAN9311_LAN9312_REL_B_2.0.bin	Basic Managed binary image of LAN9311, and LAN9312
LAN9311_LAN9312_REL_FM_2.0.bin	Fully Managed binary image of LAN9311, and LAN9312
LAN9313_REL_B_2.0.bin	Basic Managed binary image of LAN9313.
LAN9313_REL_FM_2.0.bin	Fully Managed binary image of LAN9313.
LAN9303_REL_B_2.0.bin	Basic Managed binary image of LAN9303/89303.
LAN9303_REL_FM_2.0.bin	Fully Managed binary image of LAN9303/89303.

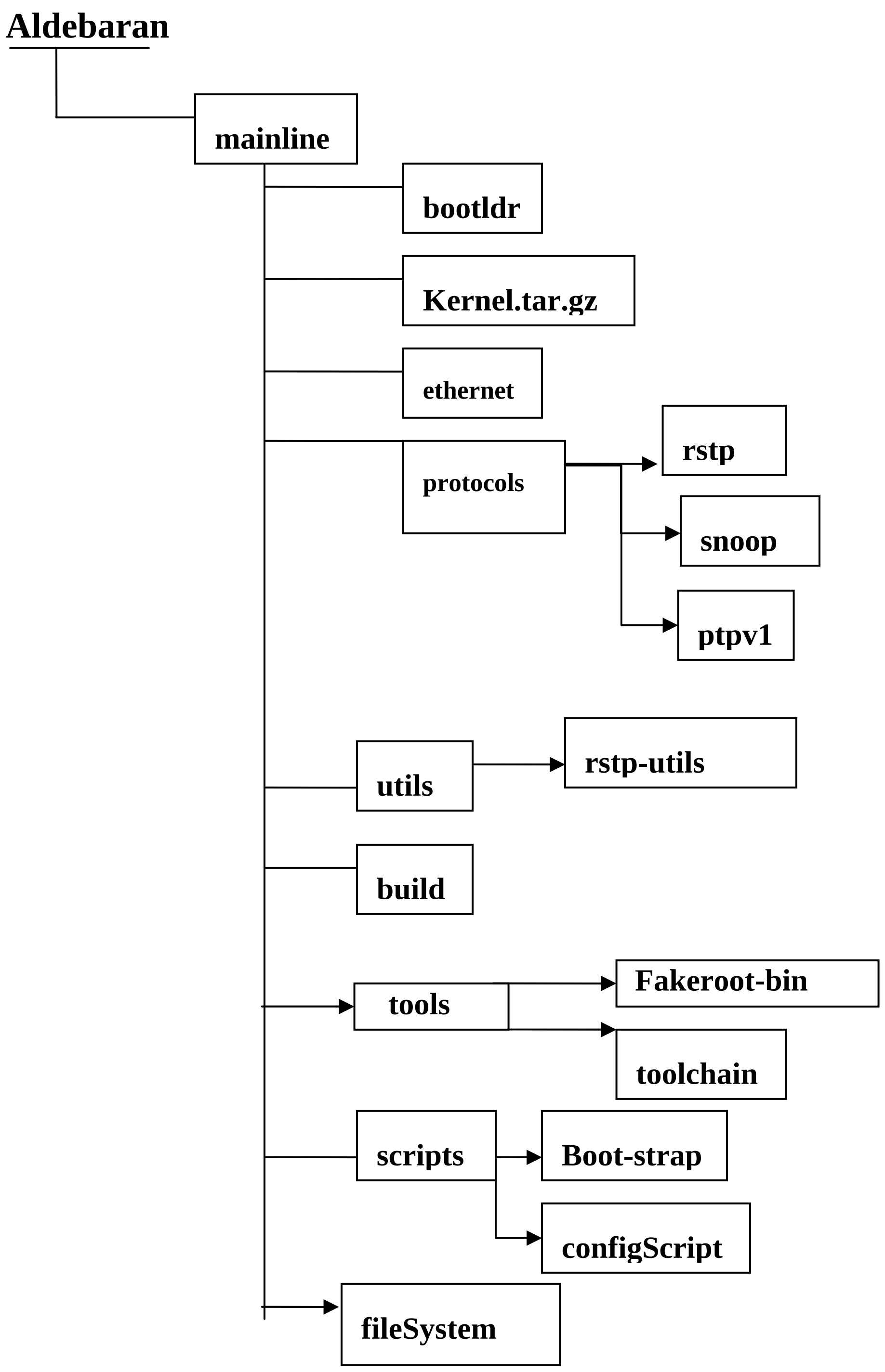
The binary package consists of the 16MB binary image. The binary image consists of boot-loader, kernel and JFFS file-system. The Basic binary package consists of all but no protocols. The Full binary package consists of basic binary image and the RSTP, PTPV1, IGMP/MLD protocol support.

The 2.0 release contains two source tar-balls. (These are common source packages for LAN9311, LAN9312, LAN9313, LAN9303 and LAN89303)

LAN93xx_PL_FULL_srcpkg_2.0.tar.gz	This platform source code package consists of boot-loader, kernel, ethernet driver, makefiles(build scripts), configScripts, jffs filesystem, and tool-set.
LAN93xx_PL_MIN_srcpkg_2.0.x.tar.gz	This platform source code package contains everything, but no kernel and tools. Instead of complete linux 2.16.16, it contains only the netlink.h headerfile.
LAN93xx_PROTO_srcpkg_2.0.x.tar.gz	This protocol source code package consists of software for RSTP, IGMP/MLD, and PTPV1.

4 SOURCE PACKAGE

By un-taring the two source balls the user will get the following source tree.



5 SOFTWARE COMPONENTS OF THE RELEASE

- Boot-loader: The Boot-loader is made for sh3 Aldebaran platform and the boot-loader comes with the smsc911x and smsc921x Ethernet driver support. The user can update the kernel and file system by tftp operation. A simple command “info” at the boot-loader terminal lists all the supported features of boot-loader.
- Kernel: The Aldebaran kernel is 2.6.16 version. The kernel is built from the tool chain and just updated the header files for the bridge and net-link protocol support. (The 2.6 kernel source code contains cyclic links of directories, the perforce doesn’t understand the cyclic link, that is the reason to keep the kernel in the zipped and tared format, the kernel is un-tared during the build process)
- Ethernet-driver. The driver is bridge aware Ethernet driver. It supports two switch interfaces. The driver is compiled for either basic or fully-managed support. If it is fully-managed it includes the components to support IGMP, RSTP and PTPV1. Ethernet driver is accompanied with the command line tool interface called **cmd**. The cmd invokes the iocl calls to interact with kernel and the ethernet driver.
- Config-scripts: The config scripts can be used to configure/enable/disable all the features sets of the switch. The configured scripts are grouped by PTPv1, QOS, STP, UTILS, and VLAN.
- File-system. It is a jffs file-system. The file system is made by the make file and the filesystem is loaded with all the pre-built binaries.
- Protocol support:
  - RSTP: Compliant with IEEE 802.1D-2004 standards. The RSTP protocol software is fully back-ward compatible i.e., it supports STP. The RSTP comes with the bridge-util function. The kernel module bridge.ko is located at /lib/modules/. The user can configure the bridge parameters by a simple command brctl. The brctl is located at /usr/bin.
  - PTPV1: The PTPV1 software is compliant with IEEE1588 2002 standard. As a switch the PTPV1 software supports boundary-clock and ordinary clock.
  - IGMP: The IGMP software is compliant with version 2 standard, and backward compatible with v1.
  - MLD: The MLD software is compliant with version 2 standard only. **It is not compatible with MLDv1.**
- Build-script: The build scripts/makefile can build the binary and source package. All of the binary packages can be built by “buildall” tool. The user can make a basic binary package and fully managed binary package. If the user wants to make a Basic or FM package for a particular platform, the user can use the following flags to make different combinations: PL_TYPE=LAN9312/LAN9313/LAN9303, PKG_TYPE=FM/B, BLD_TYPE=REL. The PL_TYPE signifies the chip-set, PKG_TYPE specifies fully-managed or basic, and the BLD_TYPE=REL is for release. The Ethernet driver remains the same, but loading the arguments are different for LAN9311/LAN9312, LAN9313, and LAN9303/89303. The ***buildpkg*** tool takes the PL_TYPE, PKG_TYPE, and BLD_TYPE as arguments and builds the binary image.

- Tool-set: The tool-set is located /tools/tool chain directory. It contains gcc, bin-utils and glibc for polaris sh3 environment. The developer can install the tool-set before building the packages.
- Misc: Fake-root bin and make-devs tool-sets are used by the makefile to make the flash-file system. The fake root fakes the root privileges for file manipulation. The user doesn't need superuser privileges to make a flash file system.

5.1 Ethernet driver and command utilities

Ethernet driver is a bridge aware driver. The ethernet driver can be built with basic or fully-managed options. In the fully-managed option, the driver peeks at all the incoming packets, and if it identifies a PTP, IGMP, MLD or STP/RSTP packet, it sends the packet to its respective module by a netlink socket interface. The Ethernet module polls the PHY interface to get the link status and provides rich ioctl utilities to probe and program the various registers/features.

The cmd utility which is located at /usr/bin can be used to get the statistics, and read and write registers. Simply typing the cmd utility in the command line terminal gives all the options that are supported.

5.2 RSTP

The rstp module can be loaded by a script called runRstp, which automatically loads the bridge modules and initiates the state machine. The user can utilize the brctl utility to configure the bridge. The list of supported commands for brctl

addbr	<bridge>	**add bridge**
delbr	<bridge>	**delete bridge**
addif	<bridge> <device>	**add interface to bridge**
delif	<bridge> <device>	**delete interface from bridge**
setageing	<bridge> <time>	**set ageing time**
setbridgeprio	<bridge> <prio>	**set bridge priority**
setfd	<bridge> <time>	**set bridge forward delay**
sethello	<bridge> <time>	**set hello time**
setmaxage	<bridge> <time>	**set max message age**
setpathcost	<bridge> <port> <cost>	**set path cost**
setportprio	<bridge> <port> <prio>	**set port priority**
showmacs	<bridge>	**show a list of mac addrs**
showstp	<bridge>	**show bridge stp info**
stp	<bridge> {on|off}	**turn stp on/off**
rstp	<bridge> {on|off}	**turn rstp on/off**
setprotver	<bridge> {version no>	**Force the protocol version number**

The setprotver bridge 0 makes the protocol machine to run in STP Mode. The showstp bridge command shows that each port variables and the state. The user can see the port state by utilizing this command.

5.3 PTPV1/IEEE1588 V1 (LAN9311/12/13 Only)

PTPV1 application consists of two binaries that are ptpd and ptpconfig.o. ptpd is a deamon process and ptpConfig.o is the user utility to set the parameters. The 2 port switch can act as a boundary clock or as a slave clock. The ptp Config supports the following options.

enable	**enable PTP**
disable	**disable PTP**
start	**start PTP daemon**
setPortMcast	**setPortMcast munilicast-address**
setAsSlaveClk	**setAsSlaveClk slavePort[0/1]**
setClockID	**PTP setClockID (ATOM/GPS/NTP/HAND/INIT/DFLT)**
setClockUUID	**setClockUUID UUID[6 characters Max]**
setClockDomaine	**setClockDomaine domaine[16 characters]**
setClockStratum	**setClockPreference 0/1**
setClockPreference	**setClockStratum 1/2/3/4/255**
setBoundaryClock	**setBoundaryClock 0/1**
setSyncINterval	**setSyncINterval time [log2(in seconds)]**
setStepsRemoved	**setStepsRemoved number [1....n]**
setPortUUID	**setPortUUID portNumber uuid[6 character s Max]**
setPortSubDomain	**setPortSubDomain subdomaine[string]**

5.4 IGMP and MLD Snooping

IGMP v2 (ipv4) and MLD v2 (ipv6) snooping are done by the module snoopMgr.o and snoopConfig.o. The snoopMgr.o is an application that gets the packet from the Ethernet bridge and maintains the registry. IGMP is backward compatible with IGMP v1; MLD is not backwards compatible.

version	<IGMP>	**IGMP Version to snoop**
add	<IGMP>	**IGMP ethernet No to snoop**
start	<IGMP>	**IGMP Start snooping**
stop	<IGMP>	IGMP stop snooping
mroute	<IGMP>	**IGMP mroute**

query	<IGMP>	**IGMP query**
Show	<IGMP>	**IGMP dump data**
Show	<IGMP>	**IGMP Timer Set**

5.5 SNMP I/F

The SNMP I/F supports ETHER-LIKE MIB (RFC 3635), IF-MIB (RFC 2863), and RMON-MIB (RFC 2819). The IF provides simple ioctl utilities to get the counter values. The user can utilize the cmd utility to get the statistics. The following commands can be used to gather the statistics.

```
cmd -c GET_MIB -s ETHER-LIKE-MIB
cmd –c GET_MIB –s RMON-MIB
cmd –c GET_MIB –s IF-MIB
```

5.6 Configuration Scripts

The configuration scripts are provided under /usr/bin/configScripts directory in the flash file system. The user can use the utilities to configure the VLAN, QOS, PTPV1, RSTP, and all the misc functionalities of the switch.

Script	Purpose	Usage	Arguments
		VLAN SCRIPTS	
pvid.read	Get current PVIDs for switch ports	pvid.read [port]	**port** is an optional argument which can be 0, 1, or 2. If not supplied PVIDs for all ports will be retrieved
pvid.set	Set PVID for a given switch port	pvid.set portnum vid [pri]	**portnum** indicates which port's PVID is to be set. Valid values are 0, 1, or 2. **vid** is the VLAN ID which will become the port's default VID (PVID). Valid range of values is 1-4094 **pri** is an optional argument. If present it sets the default VLAN priority for this port. Valid range of vlaues is 0-7.
vl.add	Add a record to the VLAN table	vl.add vid portbits [untagbits]	**vid** is the VID for this VLAN. **portbits** is a bitmap indicating which ports are to be members of this VLAN. It is a string of 1’s and/or 0’s with no spaces in between. The leftmost number maps to port 2, the middle maps to port 1, the rightmost maps to port 0. For example, **vl.add 5 011** would make ports 1 and 0 members of VID 5. **untagbits** is an optional argument. If

			present it indicates member ports for which packets should exit the switch untagged. It is a bitmap which is implemented exactly as **portbits**. Adding to the above example, **vl.add 5 011 010** would specify that packets which ingress on port 1 should have their tags removed on egress (assuming this option is enabled on the specific egress port).
vl.admall	Sets a port to admit all packets, when VLAN rules are enabled	vl.admall portnum	**portnum** indicates which port to apply this setting to. Valid values are 0, 1, or 2.
vl.admna	Set (or unset) Admit Not Active per port.	vl.admna portnum [-]	**portnum** indicates which port to apply this setting to. Valid values are 0, 1, or 2. The optional argument “-“ causes Admit Not Active to be cleared on this port.
vl.admov	Set (or unset) Admit Only VLAN per port.	vl.admov portnum [-]	**portnum**: as above. Optional argument “-” : as above.
vl.clr	Remove all ports from all VLANs in switch	vl.clr	none
vl.mbr	Enable (or disable) membership checking per port	vl.mbr portnum [-]	**portnum** and optional “-“ as above.
vl.off	Turn off VLAN rules in switch.	vl.off	none
vl.on	Enable VLAN rules in switch, and set all VLAN defaults.	vl.on	none
vl.read	Read contents of VLAN table in switch	vl.read	none
QoS SCRIPTS			
dfsrv.map	Maps a diffserv code point to a switch priority	dfsrv.map dscp pri	**dscp** is a diffserv code point, values 0-63 **pri** is a priority value, 0-7
dfsrv.pop	Populates the Diffserv table in the switch by mapping all 64 diffserv codepoints (in groups of 8) to switch prorities.	dfsrv.pop pri1 pri2 pri3 pri4 pri5 pri6 pri7 pri8	**pri1**, **pri2**, etc.: see **pri** above.
dfsrv.rd	Get switch priority value for given diffserv code point.	dfsrv.rd	**dscp**: see dfsrv.map above

egr.rt	Configure egress rate limiting in switch	egr.rt portnum q1_rate q2_rate q3_rate q4_rate egr.rt portnum -	**portnum**: see above q1_rate, q2_rate, …: Each of the four xmit queues in the switch is given an egress rate. "-" disables rate limiting
fixpri	Set (or unset) fixed priority mode in switch	fixpri [-]	"-" reverts switch to WRR priority mode
ing.rt	Enable or disable ingress rate limiting in switch	ing.rt mode ing.rt -	**mode** indicates whether rate limiting is per source port only, priority only, or port and priority. "-" disables rate limiting
ing.wcir	Sets ingress rate limit per port in Mbps	ing.wcir portnum mbps	**portnum** as above **mbps**: max ingress rate for **portnum**
pri.da	Sets (or unsets) DA_HIGHEST priority mode in switch. When set, xmit queue number is gotten from setting in ALR table for static addresses.	pri.da [-]	"-" turns off DA_HIGHEST mode.
pri.diff	Sets switch to base IP priority on Diffserv table.	pri.diff [-]	"-" unsets this mode
pri.mod	Sets port frame priority modification per vlan for all ports (egress).	pri.mod pri vid	**pri** as above, values 0-7 **vid**: see **vl.add** above
pri.tos	Sets switch to base IP priority on TOS precedence bits.	pri.tos [-]	"-" unsets this mode
pri.vl	Sets switch to use VLAN priority bits to set switch priority.	Pri.vl [-]	"-" unsets this mode
pri2que	Maps given priority to particular switch queue	pri2que pri queue	pri as above, values 0-7 que: switch xmit queue number, valid range is 0-3.
STP/RSTP SCRIPTS			
runStp	Start Spanning Tree protocol on switch	runStp	none
runRstp	Start Rapid Spanning Tree protocol on switch	runRstp	none
PTP SCRIPTS			
runptpbnd.sh	Start Precision Time Protocol (IEEE1588)	runptpbnd.sh	none

	with switch configured as Boundary Clock		
runptpslv.sh	Start Precision Time Protocol (IEEE1588) with switch configured as Ordinary Clock	runptpslv.sh	none
SNOOPING FUNCTION SCRIPTS			
igmp.on	Start IGMP Snooping daemon process and configure it as v2	igmp.on	none
mld.on	Start MLD Snooping daemon process and configure it as v2	mld.on	none
5.7 MISCELLANEOUS SCRIPTS			
addstatic	Add a static MAC address to the ALR table	addstatic MAC portnum [queue]	**MAC**: valid MAC address **portnum:** as above, 0, 1, or 2 **queue**: optionally specify which switch queue to use for xmit, values 0-3.
delstatic	Delete a static MAC from the ALR table	delstatic MAC	**MAC**: as above
fc.on	Selects manual flow control and enables pause frames for TX and RX in full duplex mode	fc.on	none
jmb.on	Enable jumbo frames in switch.	jmb.on [-]	"-": optionally disable jumbo frame capability
port.mirr	Configure port mirroring in switch	port.mirr mirr_port sniff_port [RX | TX]	**mirr_port**: number of the port to be mirrored (RX and TX by default), values 0-2 **sniff_port**: number of the port on which to monitor mirrored port, values 0-2 **RX**: optional arg for RX mirroring only **TX**: optional arg for TX mirroring only
port.pri	Set port priority by mapping all received VLAN priorities to specified priority	port.pri port pri	**port**: as above, 0-2 **pri**: as above, 0-7, OR "-" to reset to default
port.type	Set indicated egress port to specified type	port.type portnum type_code [shouldInsertTag] [changeTagBits]	**portnum:** as above, 0-2 **type_code:** 0, 1, 2 or 3 for Dumb, Access, Hybrid, or CPU port types, respectively. **shouldInsertTag:** optional arg used for Hybrid type only, either 0 or 1

			changeTagBits: optional arg for Hybrid type only. Three binary digits. From left to right: ChangeVID, ChangePriority, ChangeTag.

5.8 Flash File system:

The Flash file system comes-up with a start up script called switchStart located at /etc/init.d. The star-up script automatically loads the ethernet driver. The busy box provides all the executables and utilities support.

6 Image Handling Guide

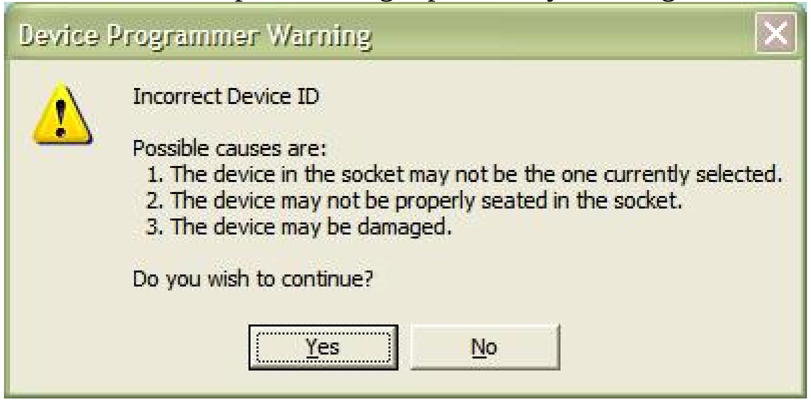
The Build scripts are provided to build the source package in various modes. The source package has all the make files in the aldebaran/mainline/build directory. The makefile can produce 6 binary outputs.

Binary Output File	Make command and options.
LAN9311_LAN9312_REL_B_2.0.bin	Make package PL_TYPE=LAN9312 PKG_TYPE= BASIC
LAN9311_LAN9312_REL_FM_2.0.bin	Make package PL_TYPE=LAN9312 PKG_TYPE=FM
LAN9313_REL_B_2.0.bin	Make package PL_TYPE=LAN9313 PKG_TYPE=BASIC
LAN9313_REL_FM_2.0.bin	Make package PL_TYPE=LAN9313 PKG_TYPE= FM
LAN9303_REL_B_2.0.bin	Make package PL_TYPE=LAN9303 PKG_TYPE=BASIC
LAN9303_REL_FM_2.0.bin	Make package PL_TYPE=LAN9303 PKG_TYPE= FM

6.1 Preparation to Program the Flash

Once the image is build, customer can burn the image on the flash with following steps:

1. Install software for EMP-30
   - o Install EPMW1_5.exe from U:\NPG\SWDEV\Software Backup\NeedhamsEmp30
2. Connect EMP-30 to Parallel Port or USB based on model
3. Put the FLASH ship in EMP-30 TSOP56A Socket
   - o if FLASH wasn't placed in right position, you will get following error.



Device Programmer Warning

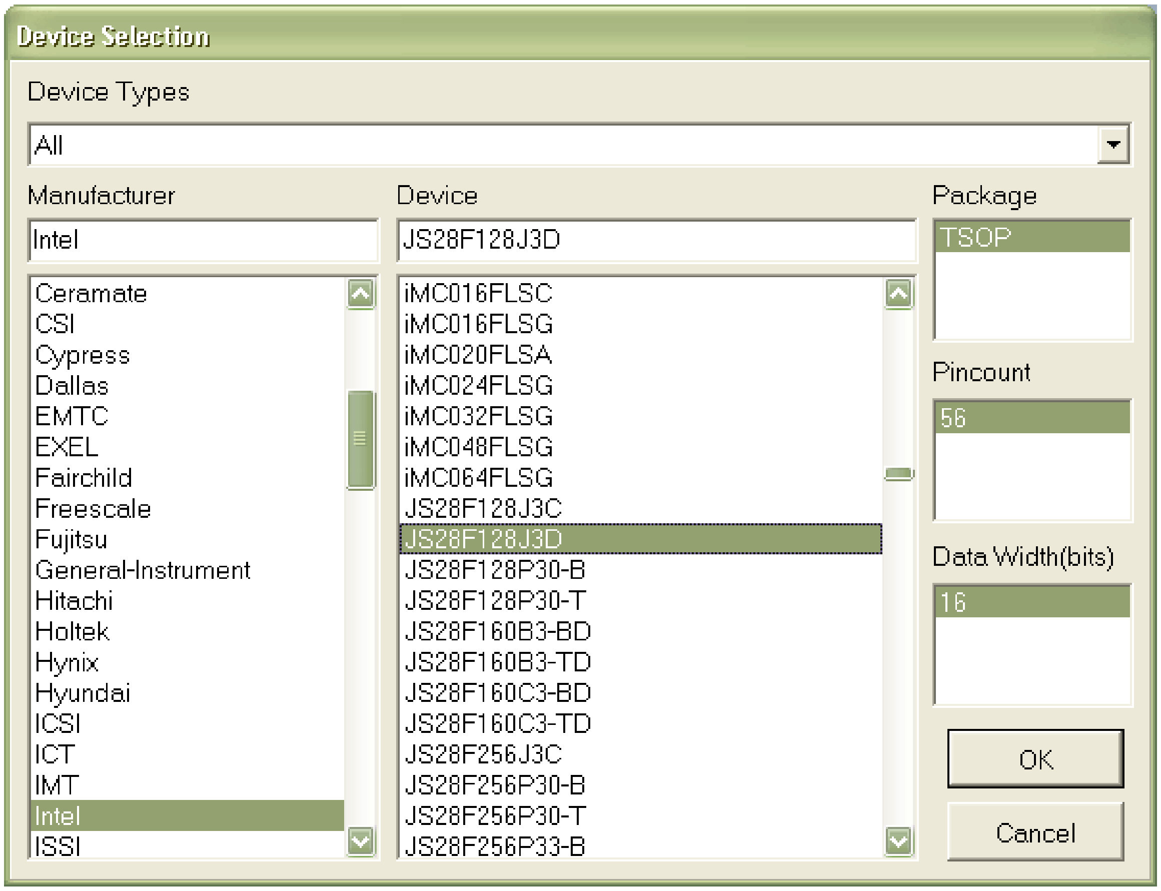
Incorrect Device ID

Possible causes are:
1. The device in the socket may not be the one currently selected.
2. The device may not be properly seated in the socket.
3. The device may be damaged.

Do you wish to continue?

4. Execute installed EMP Device Programming Software
5. After it launches, select Device if necessary.
   Device should be
   - Manufacturer : INTEL
   - Device : TE28F128J3D
   - Package : TSOP
   - Pincount : 56

- Data Width : 16



Device Types: All

Manufacturer: Intel

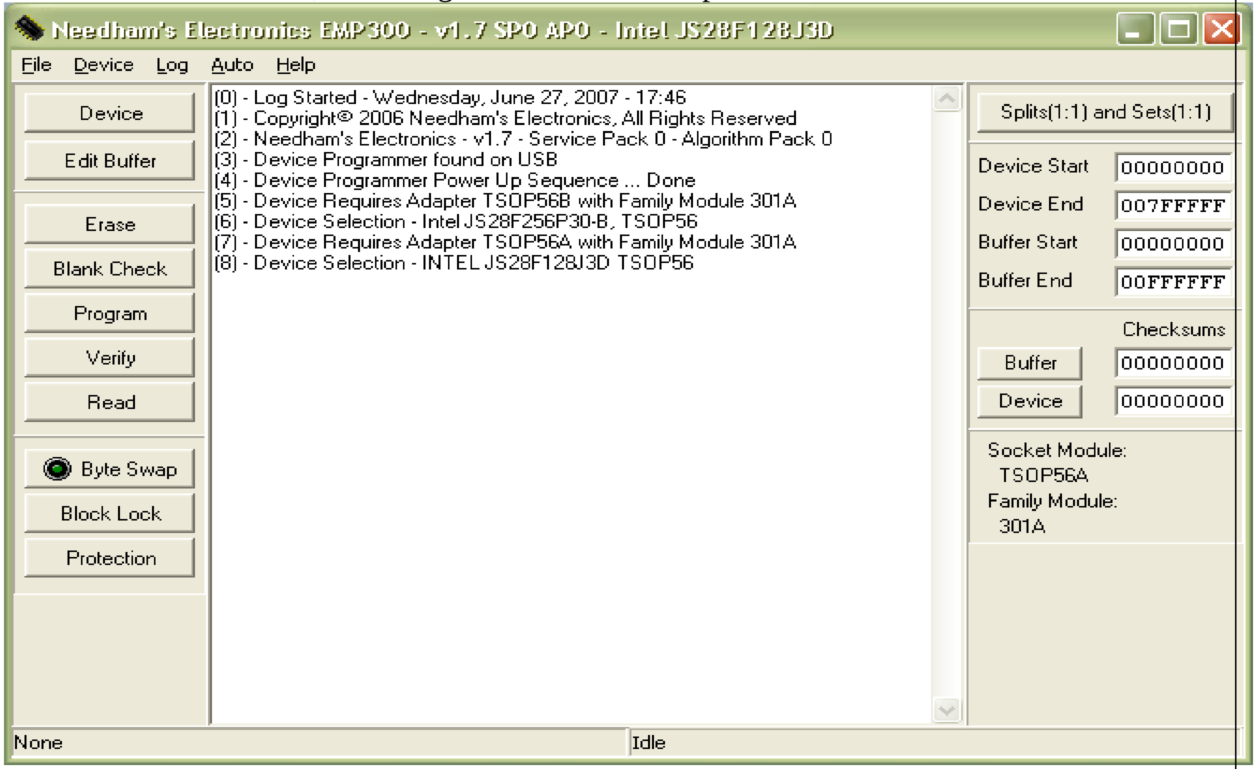
Device: JS28F128J3D

Package: TSOP

Pincount: 56

Data Width(bits): 16

6. When device is selected, following screen will come up



```
[0] - Log Started - Wednesday, June 27, 2007 - 17:46
[1] - Copyright© 2006 Needham's Electronics, All Rights Reserved
[2] - Needham's Electronics - v1.7 - Service Pack 0 - Algorithm Pack 0
[3] - Device Programmer found on USB
[4] - Device Programmer Power Up Sequence ... Done
[5] - Device Requires Adapter TSOP56B with Family Module 301A
[6] - Device Selection - Intel JS28F256P30-B, TSOP56
[7] - Device Requires Adapter TSOP56A with Family Module 301A
[8] - Device Selection - INTEL JS28F128J3D TSOP56
```

Device Start: 00000000
Device End: 007FFFFF
Buffer Start: 00000000
Buffer End: 00FFFFFF

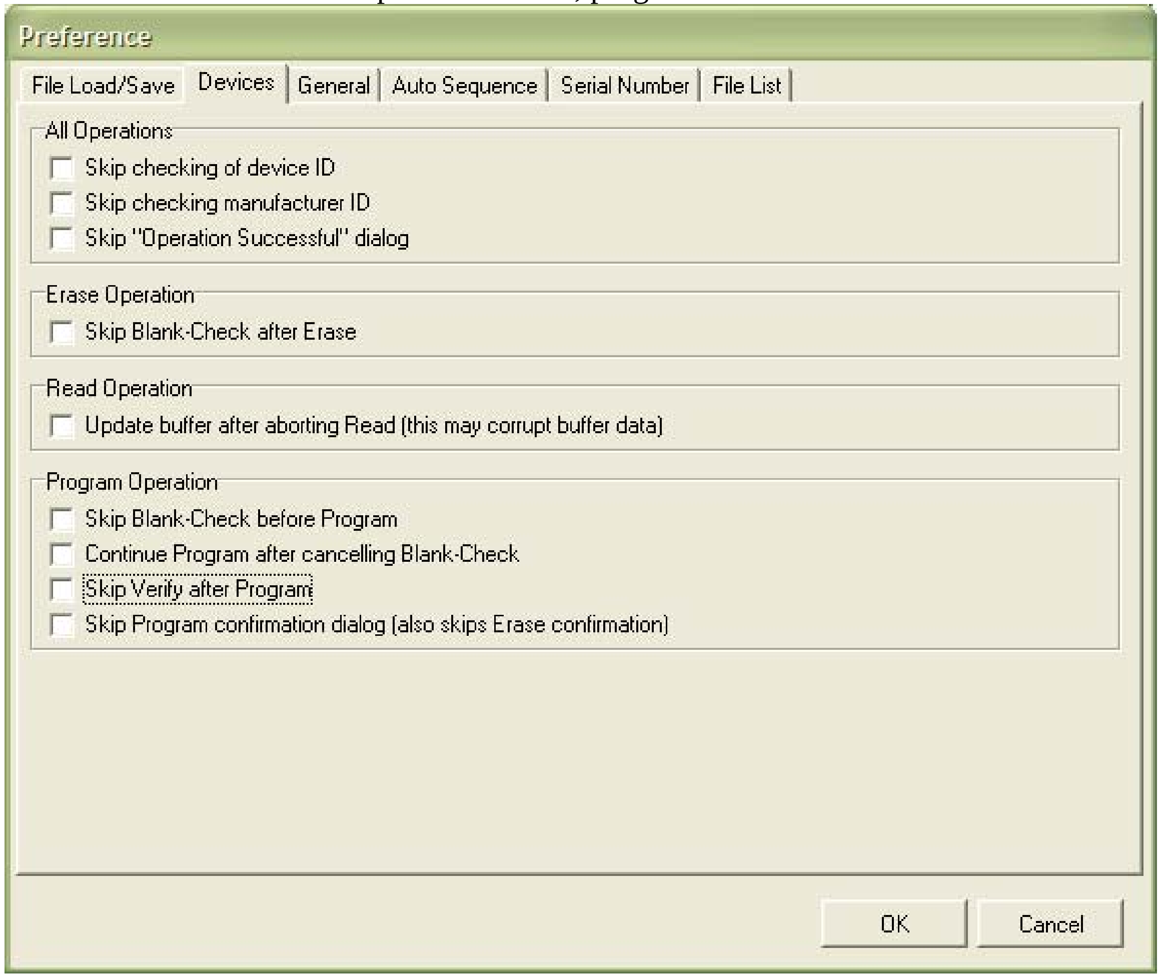
Checksums
Buffer: 00000000
Device: 00000000

Socket Module:
TSOP56A
Family Module:
301A

None | Idle

2. SET OPTIONS

1. “File -> Preference” shows options for erase, program and etc



All Operations

- [ ] Skip checking of device ID
- [ ] Skip checking manufacturer ID
- [ ] Skip "Operation Successful" dialog

Erase Operation

- [ ] Skip Blank-Check after Erase

Read Operation

- [ ] Update buffer after aborting Read (this may corrupt buffer data)

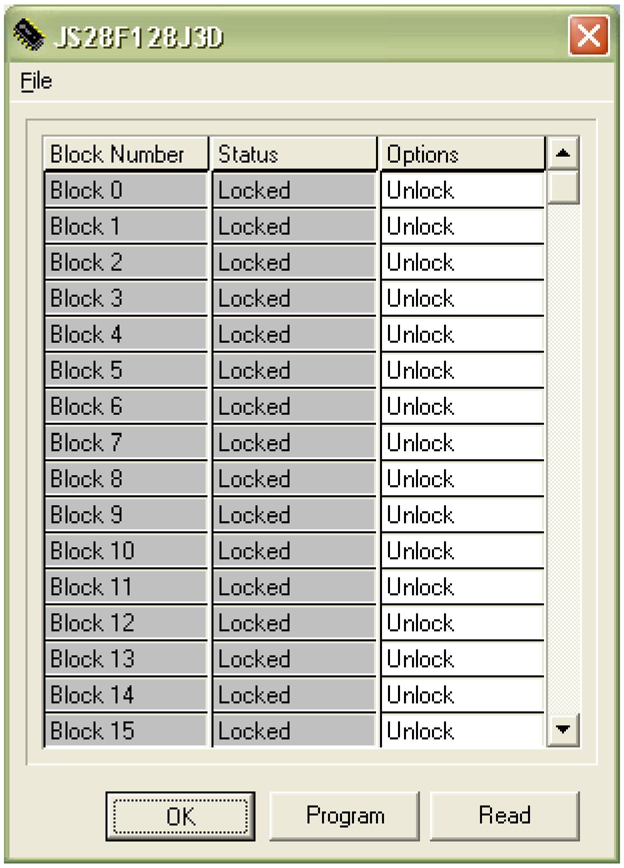
Program Operation

- [ ] Skip Blank-Check before Program
- [ ] Continue Program after cancelling Blank-Check
- [ ] Skip Verify after Program
- [ ] Skip Program confirmation dialog (also skips Erase confirmation)

To save time, you can check following options;
“Skip Blank-Check after Erase”
”Skip Blank-Check before Program”
”Skip Verify after Program”
But, make sure FLASH is ERASED before programming if you select “Skip Blank-Check before Program” option

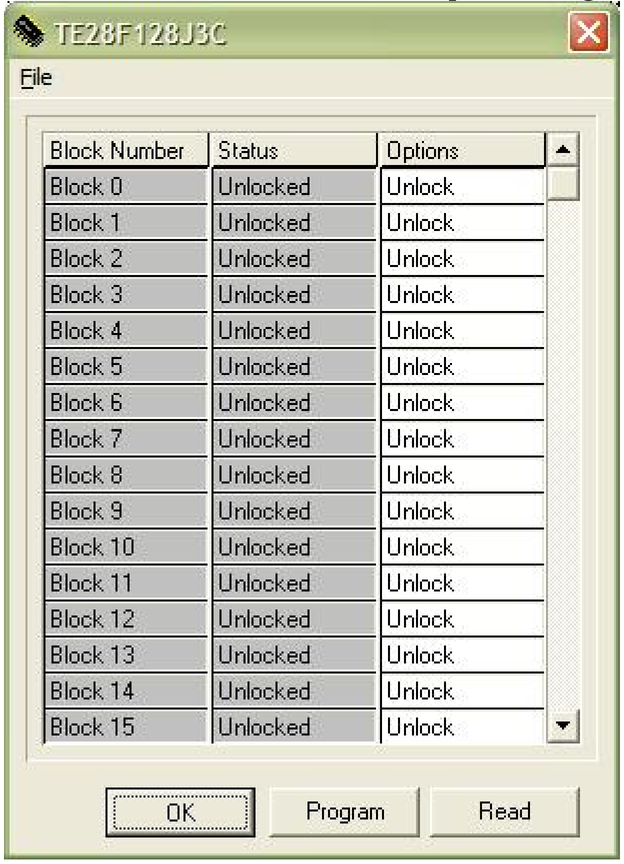
3. ERASE FLASH

1. If FLASH is new or already erased, you can skip this one
2. If FLASH is used at Foxboro (i.e, Foxboro boots with the FLASH), FLASH need to be unlocked before erase or program, because Foxboro protect certain sectors for security reason.
   If FLASH is new or used for Polaris, unlock procedure is not necessaray.
3. Press "Block Lock" button at Main window



Block Number	Status	Options
Block 0	Locked	Unlock
Block 1	Locked	Unlock
Block 2	Locked	Unlock
Block 3	Locked	Unlock
Block 4	Locked	Unlock
Block 5	Locked	Unlock
Block 6	Locked	Unlock
Block 7	Locked	Unlock
Block 8	Locked	Unlock
Block 9	Locked	Unlock
Block 10	Locked	Unlock
Block 11	Locked	Unlock
Block 12	Locked	Unlock
Block 13	Locked	Unlock
Block 14	Locked	Unlock
Block 15	Locked	Unlock

4. Some of status is set as “Locked”, press “Program” button to unlock all blocks

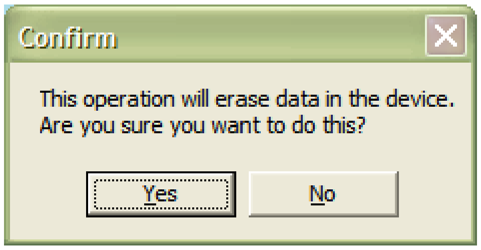


TE28F128J3C

File

Block Number	Status	Options
Block 0	Unlocked	Unlock
Block 1	Unlocked	Unlock
Block 2	Unlocked	Unlock
Block 3	Unlocked	Unlock
Block 4	Unlocked	Unlock
Block 5	Unlocked	Unlock
Block 6	Unlocked	Unlock
Block 7	Unlocked	Unlock
Block 8	Unlocked	Unlock
Block 9	Unlocked	Unlock
Block 10	Unlocked	Unlock
Block 11	Unlocked	Unlock
Block 12	Unlocked	Unlock
Block 13	Unlocked	Unlock
Block 14	Unlocked	Unlock
Block 15	Unlocked	Unlock

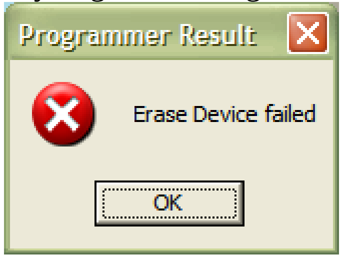
5. Erase Device
   - Press “Erase” button to erase whole FLASH



Confirm

This operation will erase data in the device.
Are you sure you want to do this?

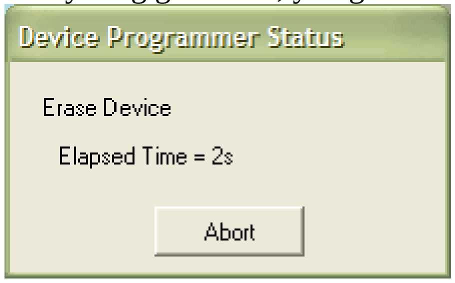
   - Confirm it by pressing “Yes”
   - It takes ~1.5min depends on model.
   - If you get following message, FLASH is either protected or mis-placed.



Programmer Result

Erase Device failed

- Everything goes well, you get


Device Programmer Status

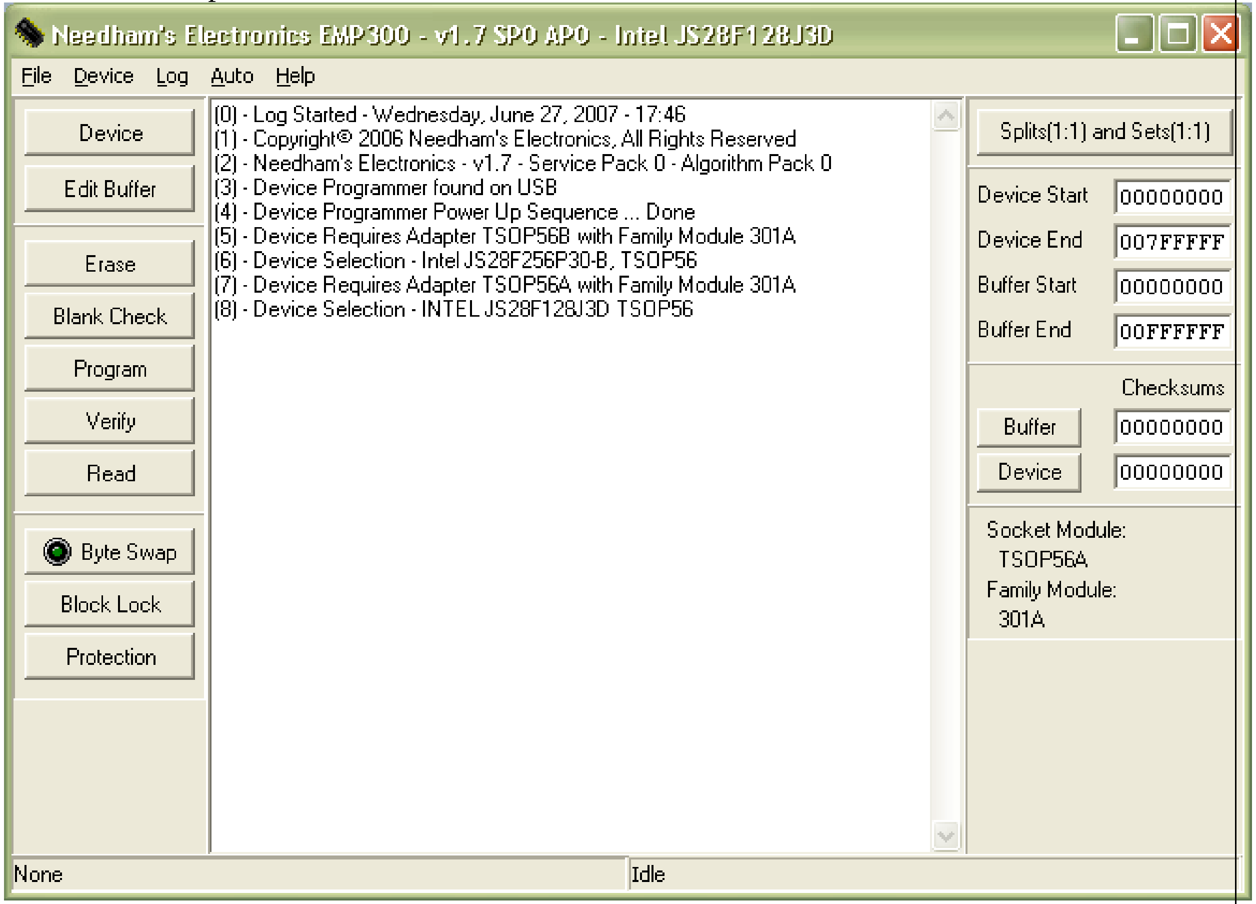
Erase Device

Elapsed Time = 2s

4. PROGRAM BINARY FILES

For an example take the **LAN9311_LAN9312_REL_B_1.0.x.bin**

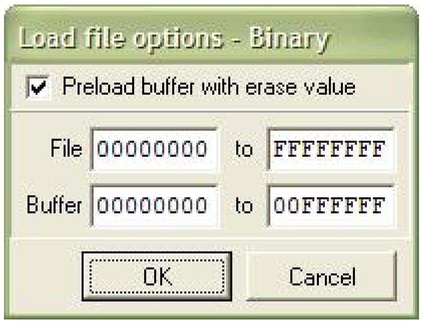
1. Load files into EMP-30 (***Order to load files is important)***
2. Select File -> Open from Menu


Needham's Electronics EMP300 - v1.7 SP0 AP0 - Intel JS28F128J3D
File Device Log Auto Help
Device
Edit Buffer
Erase
Blank Check
Program
Verify
Read
Byte Swap
Block Lock
Protection
Splits(1:1) and Sets(1:1)
Checksums
Buffer
Device

```
[0] - Log Started - Wednesday, June 27, 2007 - 17:46
[1] - Copyright© 2006 Needham's Electronics, All Rights Reserved
[2] - Needham's Electronics - v1.7 - Service Pack 0 - Algorithm Pack 0
[3] - Device Programmer found on USB
[4] - Device Programmer Power Up Sequence ... Done
[5] - Device Requires Adapter TSOP56B with Family Module 301A
[6] - Device Selection - Intel JS28F256P30-B, TSOP56
[7] - Device Requires Adapter TSOP56A with Family Module 301A
[8] - Device Selection - INTEL JS28F128J3D TSOP56
```

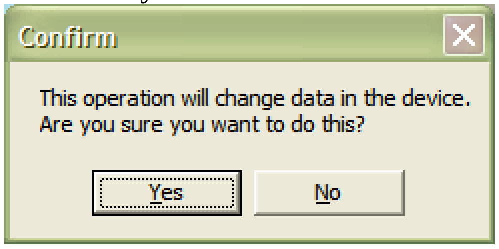
Device Start 00000000
Device End 007FFFFF
Buffer Start 00000000
Buffer End 00FFFFFF
Checksums: Buffer 00000000, Device 00000000
Socket Module: TSOP56A
Family Module: 301A
None | Idle

3. Load "**LAN9311_LAN9312_REL_B_1.0.x.bin**" into address 0x00000000



[✓] Preload buffer with erase value
File 00000000 to FFFFFFFF
Buffer 00000000 to 00FFFFFF

4. Select “Program” button
    - It will ask you to confirm the command.

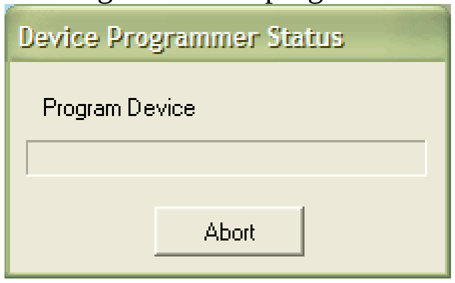


Confirm

This operation will change data in the device.
Are you sure you want to do this?

***NOTE* Depends on options, it may start to erase FLASH**

    - Pressing “Yes” will program FLASH with loaded files



Device Programmer Status

Program Device

    - It takes ~5min (depends on model) to complete