



USER GUIDE
FOR
SMT6060-TCPIP

Sundance Digital Signal Processing Inc.
4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.
Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email:
sales@sundancedsp.com
www.sundancedsp.com



APPROVAL PAGE

Name	Signature	Date
Managing Director		
Software Manager		
Design Engineer		

AUTHOR/S

Name	Signature	Date

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



DOCUMENT HISTORY

Date	Changes Made	Issue	Initials
10/30/07	Changed to new lay out and logo	0.4	NN

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



LIST OF ABBREVIATIONS

Abbreviation	Explanation
ASIC	Application Specific Integrated Circuit
BOM	Bill Of Materials
CMC	Common Mezzanine Card
Comport	Communications Port
DSP	Digital Signal Processor
FMS	Trade name for a flat-ribbon cable family from JST Mfg. Inc.
FPDP	Front Panel Data Port
FPGA	Field Programmable Gate Array
GTP	RocketIO Gigabit Transceiver Pair (Xilinx)
NA	Not Applicable
OTP	One-Time Programmable
PC	Personal Computer
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PMC	PCI Mezzanine Card
PrPMC	Processor PMC
RF	Radio Frequency
RSL	RocketIO Serial Link
SDB	Sundance Digital Bus
SDRAM	Synchronous Dynamic Random Access Memory
SHB	Sundance High-Speed Bus
SMA	Trade name for a threaded RF co-axial connector family

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



SMT	Sundance Multiprocessor Technology
TBD	To Be Determined
TI	Texas Instruments
XMC	Switched Mezzanine Card

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



TABLE OF CONTENTS

1. INTRODUCTION	8
1.1. OVERVIEW.....	8
1.2. RELATED DOCUMENTS.....	8
2. REQUIEREMENTS	9
3. CONNECTING THE HARDWARE	10
4. SYSTEM CONFIGURATION	11
4.1. CONFIGURING THE SMT363	11
4.1.1. <i>Overwriting the default IP and MAC addresses</i>	13
4.1.2. <i>Confirming the configuration</i>	15
5. DESCRIPTION	17
5.1. DIRECTORY STRUCTURE	17
5.2. THE SOCKET API LIBRARY	18
6. EXAMPLES	20
6.1. BACKGROUND.....	20
6.2. BEFORE YOU START	20
6.3. BUILDING THE EXAMPLES.....	21
6.4. RUNNING THE EXAMPLES	21
6.5. ECHOD EXAMPLE	22
6.6. FTPD EXAMPLE.....	22
6.7. RECV EXAMPLE.....	23
6.8. RECVC EXAMPLE.....	23
6.9. RECVD EXAMPLE	24
6.10. SENDC EXAMPLE.....	25
6.11. SENDD EXAMPLE	26
6.12. MULTIC EXAMPLE	27
6.13. MULTID EXAMPLE	27
6.14. SMT363 AS NETWORK INTERFACE EXAMPLE	28
6.15. CONFIGURING THE HOST.....	29
6.16. CONFIGURING THE SMT363	31
7. ACCESS TO THE SERIAL PORT OF THE NET50	32

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



TABLE OF FIGURES and TABLES

Figure 1 - SMT6060-TCPIP Hardware Connections.....	10
Figure 2 -SMT6060-TCPIP Configuring SMT363.....	11
Figure 3 - SMT363 Configuration Data	12
Figure 4 - SMT363 netboot file selection.....	12
Figure 5 - Block properties	13
Figure 6 - Block properties ID	13
Figure 7 - Change IP and MAC file example	14
Figure 8 - File selection for IP and MAC change.....	15
Figure 9 - HyperTerminal output.....	16
Figure 10 - SMT6060-TCPIP Block Diagram.....	17
Figure 11 -SMT6060-TCPIP Directory Structure	18
Figure 12 - Examle System Diagram.....	20
Figure 13 - SMT6060-TCPIP Boardinfo application	21
Figure 14 - SMT6060-TCPIP RECV Example Output	23
Figure 15 -SMT6060-TCPIP Recvc Example Output	24
Figure 16 - SMT6060-TCPIP Recvc Example Output	25
Figure 17 - SMT6060-TCPIP Sendc Example Output	26
Figure 18 -SMT6060-TCPIP Sendc Example Output	26
Figure 19 - SMT6060-TCPIP Multic Example Output	27
Figure 20 - SMT6060-TCPIP Multid Example Output	28
Figure 21 -SMT6060-TCPIP SMT363 Configuration as Network Interface	29
Figure 22 - SMT6060-TCPIP Configuring the Host	30
Figure 23 - SMT6060-TCPIP Boardinfo Host Configuration	30
Figure 24 - SMT6060-TCPIP Using the Flash Utility.....	31
Figure 25 - SMT6060-TCPIP Marking the Bootable file.....	31
Table 1 – Change IP and MAC Address.....	14

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



1. INTRODUCTION

1.1. OVERVIEW

The SMT6060 provides a network of DSP modules access to Ethernet through the Berkeley socket-API. Each of the modules in a multiprocessor DSP application is able to access the Ethernet using the socket API. The SMT363 module contains a NET50 processor and is used as a network interface.

The SMT6060 consists of a collection of software examples that simplify the development of network applications.

1.2. RELATED DOCUMENTS

TBD

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



2. REQUIEREMENTS

To develop with the SMT6060, you need

- **Code Composer Studio. v3.1 or later**
 - *Code composer is used to compile programs for the TI DSPs in the network.*
- **3L Diamond RTOS version 3.1.4 or later**
 - *Diamond is used as an RTOS that enables any task on the network to access the SMT363.*
- **SMT6001 v6.01 or later (recommended)**
 - *The SMT6001 is a utility with which you manage the contents of the flash for Sundance modules.*
- **SMT6300 v3.07 or later (recommended)**
 - Low level driver support for the host machine.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



3. CONNECTING THE HARDWARE

You should have received a cable that connects to the SMT363 module. This cable provides an Ethernet connection as well as a RS232 connection to the SMT363. Plug the RS232 port into your development PC. Use HyperTerminal, and configure the port to 9600 baud, 8-N-1.

Note: This port will later be used to output debug information from the SMT363.

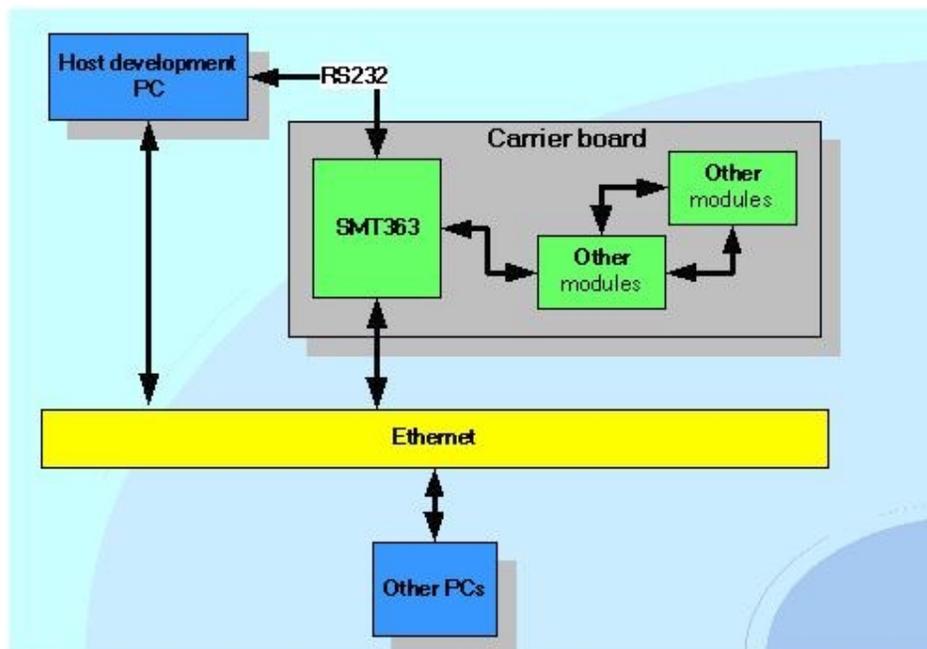


Figure 1 - SMT6060-TCPIP Hardware Connections

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



4. SYSTEM CONFIGURATION

Before you start developing, you need to ensure that you have an up to date system configuration. You need to configure the flash of the SMT363 with the correct data so that it can configure the NET50 once you power the system up.

4.1. CONFIGURING THE SMT363

Use the SMT6001 utility from Sundance to ensure that the flash of the SMT363 contains an appropriate configuration.

Select “Start->Program Files->Sundance->SMT6001->SmtFlash utility”

Ensure that you have bootloader version 5.04 or later programmed as shown below.

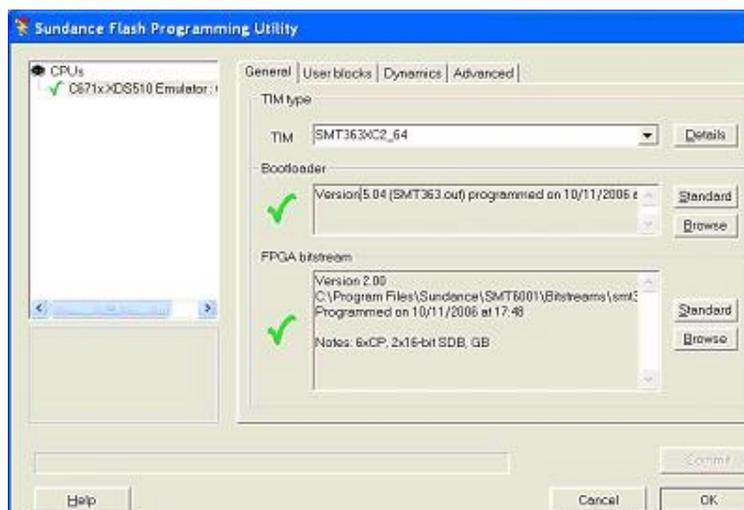


Figure 2 -SMT6060-TCPIP Configuring SMT363

Select the “User blocks” tab.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com

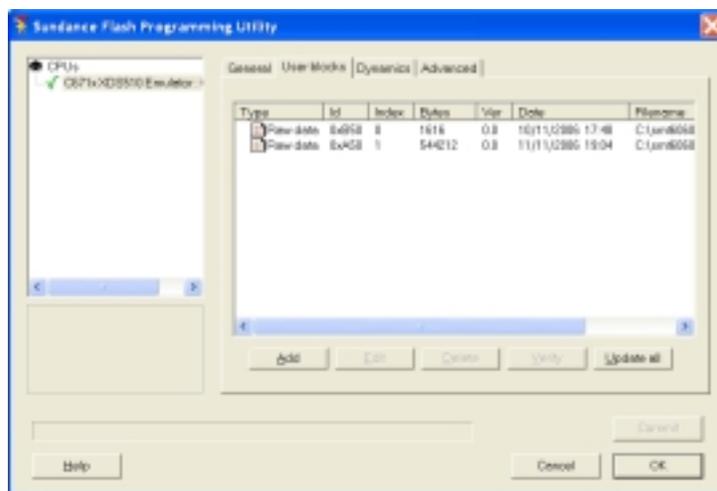


Figure 3 - SMT363 Configuration Data

There need to be two raw data sections as shown. The files used to create these data sections can be found in the `\cfg` directory of your installation.

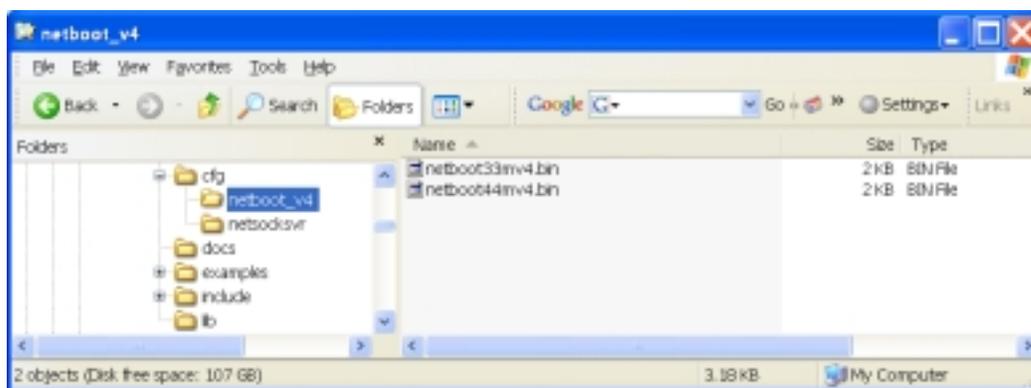


Figure 4 - SMT363 netboot file selection

The first section should have an ID value of 0xB50. (The file is `cfg/netboot_v4/netboot44mv4.bin`). (NOTE: Use `netboot33mv4.dat` for 33 MHz Net+50 boards)

If you need to add this data section, make sure that you select “big-endian” and an ID value of 0xB50.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com

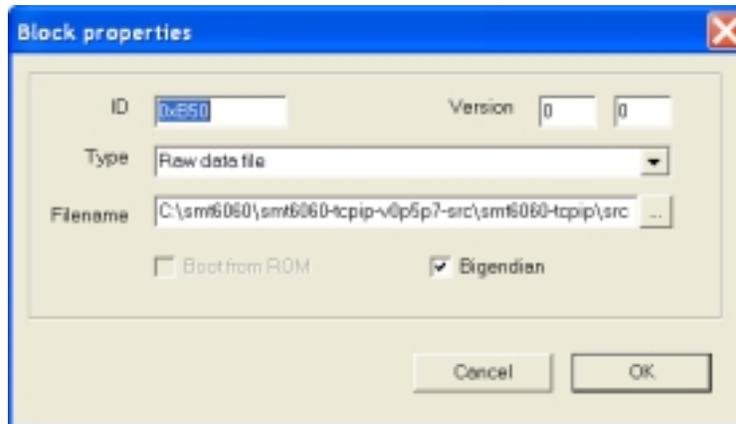


Figure 5 - Block properties

The second raw data section is *cfg/netsocksvr/release.dat* also with “big-endian” selected, and with an ID value of 0xA50. (This is the netsocksvr application that will be loaded and run on the NET50 processor.)



Figure 6 - Block properties ID

4.1.1. Overwriting the default IP and MAC addresses

The SMT363 will automatically obtain an IP address from a DHCP server. The SMT363 will have a MAC address of "00:40: AF: 00:42:06". If these values are acceptable, then you can skip this section.

If you do wish to overwrite the default values for the IP and MAC address, this can be done by specifying a data section that is programmed with the SMT6001.

Create a file that is 20 bytes long, and that has the content shown: (An example is provided with this installation package. See *sampleconfig.dat*)

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



Word	Meaning
0	IP Address (32 bits)
1	Subnet mask (32 bits)
2	Default gateway (32 bits)
3	MAC0 (32 bits)
4	MAC1 (32 bits)

Table 1 – Change IP and MAC Address

For example: The file shown below

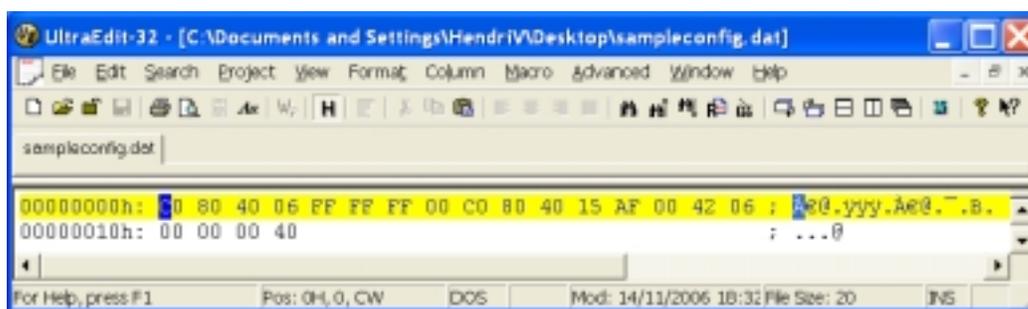


Figure 7 - Change IP and MAC file example

Defines the IP address to be 192.128.64.11 (0xC080400B)

The subnet mask to be FF.FF.FF.00

The default gateway to be 192.128.64.21 (0xC0804015)

And the MAC address to be 01.02.03.04.05.06

Now use the SMT6001 to create a raw data section (big-endian) with the ID 0xC50

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com

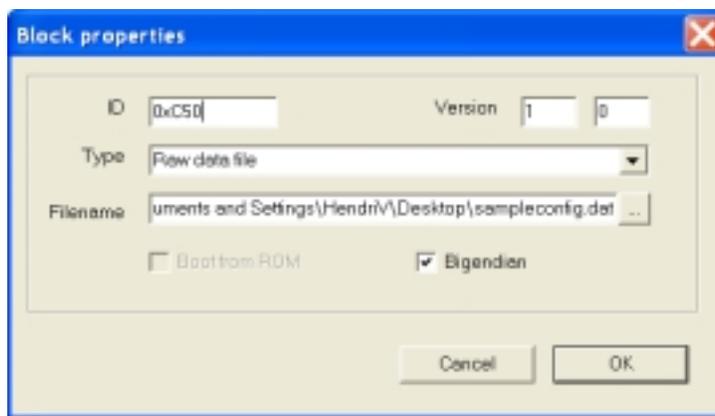


Figure 8 - File selection for IP and MAC change

Note: Specifying an IP address of 0.0.0.0 will cause the SMT363 to obtain an IP address from a DHCP server; this way you are able to specify the MAC address independently from the IP.

4.1.2. Confirming the configuration

Close the SMT6001 and Code Composer Studio. Ensure that you have HyperTerminal configured and the RS232 port connected from the SMT363 to your development PC.

Use the SMTBoardInfo application from Sundance, and issue a "Board Reset".

In the HyperTerminal window of your development PC, you should see output similar to the following:

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



```
test363 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
SMT363 NET50 Boot REV= 0003 44MHz
PHY: Intel, found at 0x000
Sdram: testing...
Sdram: tested OK
Application: loading
Application: running
:tcp: link down
ACE: Have IP address on interface eth0: 192.128.64.6
=====
Socket I/O RPC Server Application
Release 1.0.0
=====
-
Connected 00:00:32 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Figure 9 - HyperTerminal output

NOTE: your IP address will be different according to what your DHCP server assigned for the Net+50 processor (or what you have configured yourself by providing the 0xC50 data section). Take note of this IP address, as you will need to refer to it later on again.

NOTE: when using DHCP, it can take ~20sec for the Net+50 processor to get its address. The NetOS delays starting the application until the address are received.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



5. DESCRIPTION

The SMT363 provides an interface to Ethernet. It also provides an RS232 port. This port is used to output debug logging information.

You create your Diamond application as a number of tasks distributed onto the network of processors. Refer to the 3L Diamond documentation for a more detailed description of this process.

Each of the tasks that wish to access the Ethernet is linked with the socketapi.lib library. This library is your task's interface to the standard socket API.

Internally, socketapi.lib uses 3L Diamond channels to communicate with an RPCProxy task that runs on the DSP of the SMT363 module. The proxy task in turn communicates with the NET50 on the SMT363 to perform the actual network operations.

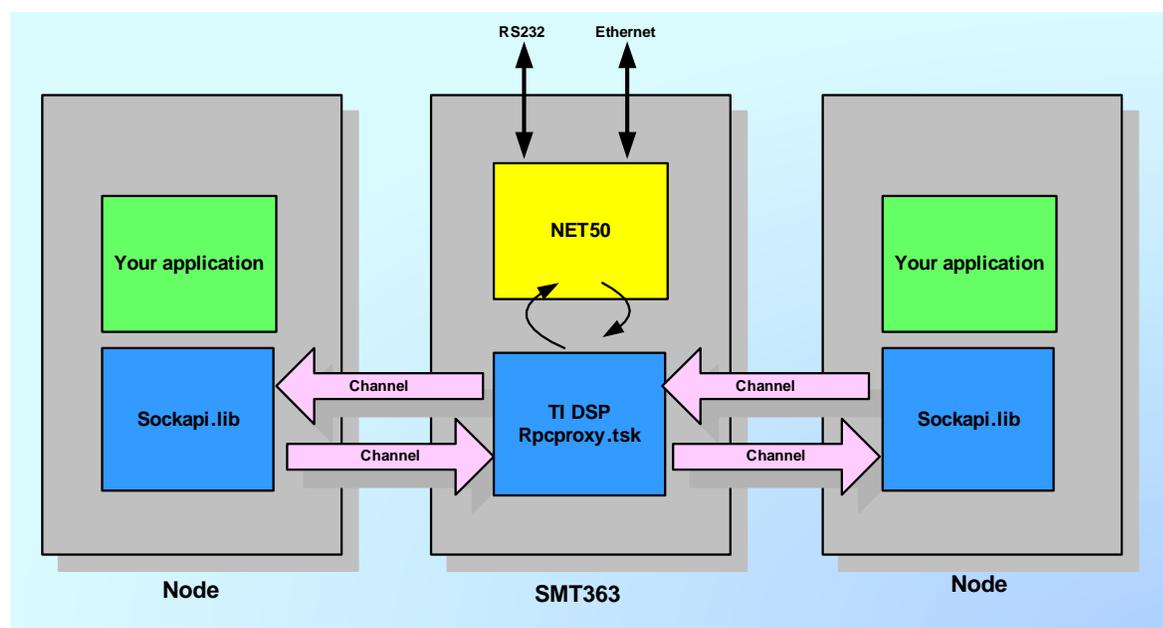


Figure 10 - SMT6060-TCPIP Block Diagram

It is important that you understand the overall system design, because you will need to ensure that the appropriate 3L Diamond configuration file is used for your application. Examples included in this package illustrate the process.

Note that you always have to place RPCProxy.tsk on the SMT363 module. Each task that you place on the network of processors will have at least 1 input and 1 output port. And each task on the network of processors should have their input and output channels connected to the proxy task as shown.

5.1. DIRECTORY STRUCTURE

The SMT6060 installs into the structure shown below. The examples directories contain a number of examples that illustrate how to use the SMT6060 libraries. The include folder contains the shared header files that are used by the examples.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



The lib folder contains the all important sockapi.lib file that you need to link your tasks to.

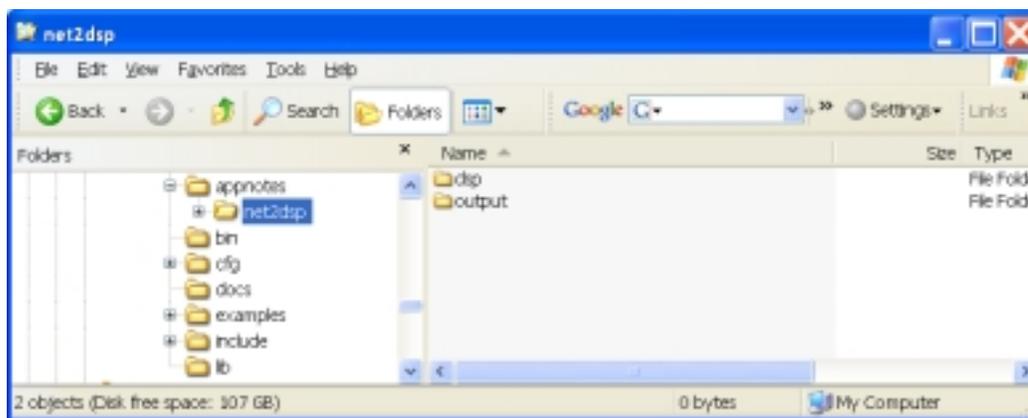


Figure 11 -SMT6060-TCPIP Directory Structure

5.2. THE SOCKET API LIBRARY

The socket API implemented for each of the node processors requires an initialization in each task before use. The `sock_init()` call is provided for this purpose. This call accepts two of the (CHAN *) arguments which were provided by the 3L loader to the main() for the task. The function below shows an extract from the DSP task's code.

```
int main(    int argc, char *argv[], char *envp[],
           CHAN * in_ports[], int ins,
           CHAN *out_ports[], int outs )
{
    /*
    ** initialize socket API
    This is a major component of the Ethernet support. This routine
    will create threads and allocate message buffers to support the
    "sockapi.h" functions.
    */
    sockapi_init = sock_init( in_ports[0], out_ports[0]);
    if( sockapi_init != 0 )    {
        printf(": sockapi init failure");
        thread_stop();
    }
    ...
}
```

An example of a 3L Diamond configuration file is shown below. This file shows the case of two Sundance modules used as the embedded system. The root processor is a SMT395_VP30 and the node is the SMT363.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



Note: This hardware configuration is used for all of the examples installed with this package.

The “option noloadcheck” is required to avoid the host from failing to load the Diamond application because the SMT363 will take some time to configure itself after a reset.

The “Wire” statement tells Diamond that there is an electrical connection between the SMT395 and SMT363 in the form of a comport. T1C1 -> T2C4

Next follows the connection and placement of the node task (echod in this case) to the RPCProxy task on the SMT363.

```
PROCESSOR root SMT395_VP30
PROCESSOR node SMT363XC2_2
option noloadcheck
```

```
WIRE ? root[CP:1] node[CP:4]
```

```
upr MAX=65408 BUFFERS=16
```

```
task rpcproxy ins=1 outs=1 stack=128K heap=?
TASK echod ins=2 outs=1 stack=128K heap=56M
```

```
connect ? rpcproxy[0] echod[0]
connect ? echod[0] rpcproxy[0]
```

```
place rpcproxy node
PLACE echod root
```

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



6. EXAMPLES

6.1. BACKGROUND

The folder *smt6060-tcpip/examples* contains a number of examples. Some of these examples run on the host machine, and some of them run on the embedded network.

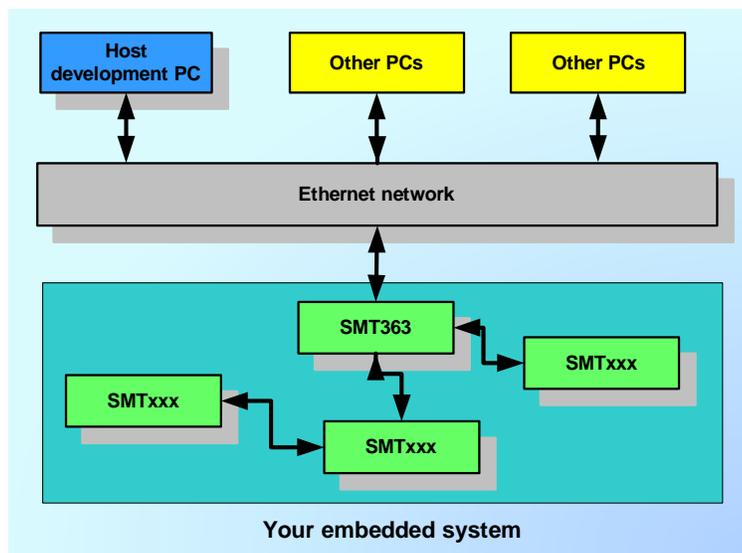


Figure 12 - Example System Diagram

6.2. BEFORE YOU START

The examples are distributed assuming 1 node module (SMT395_VP30) and the SMT363.

```
PROCESSOR root SMT395_VP30
PROCESSOR node SMT363XC2_2
```

You need to change the configuration files of the individual examples to reflect the hardware you have. Also, make sure you have the relevant wire statements and that the physical wires have been connected between processors.

The examples are using a wire connection between TIM1 comport 1 and TIM2 comport 4. This comport connection is available on the SMT310Q carrier board by selecting it with the SmtBoardInfo application.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com

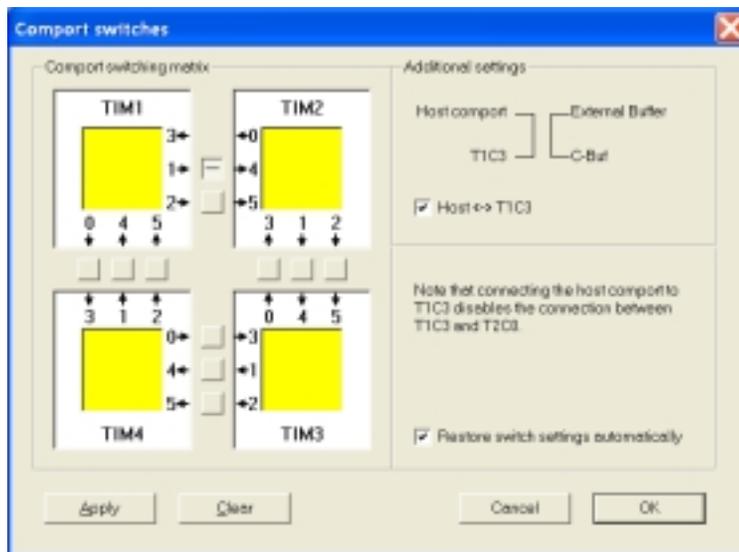


Figure 13 - SMT6060-TCPIP Boardinfo application

6.3. BUILDING THE EXAMPLES

The examples all use the standard socket API. You build and run the examples using 3L Diamond. To build an application, open the relevant *dsp* directory. For example. *smt6060-tcpip\examples\echod\dsp*

And run nmake from a command line.

The output file is produced in the output directory that is created for that purpose.

For example: *smt6060-tcpip\examples\echod\output*

Note: The examples under “clients” and “servers” are all build using make.exe and not nmake.exe.

6.4. RUNNING THE EXAMPLES

All examples are run using the 3L Diamond server application. This server application is used to display printf type messages from the running application. Note that it is possible to build “stand-alone” applications that does not communicate with the Diamond server. In those cases, you will have to remove any STDIO type functions from the example. See the 3L Diamond documentation for more information.

Note: If you do build your applications as stand-alone, then you will still be able to use HyperTerminal and the RS232 port to print debug information.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



6.5. ECHOD EXAMPLE

This example demonstrates a telnet/ECHO server application. A task on a DSP node initializes the sockets API, opens a listening socket and then proceeds to handle inbound connections in a single-threaded "select loop" style. Multiple connections are possible, and each connection will be serviced through the select-loop.

Upon a connection, the echo server will print a banner identifying the connection addresses and the current time (the DSP does not have access to the host's clock in standalone mode). Following that, every character received by the server will be echo-ed back.

Build and run echo.app.

Note the IP address of your SMT363 by looking at the output displayed in the HyperTerminal window.

On your development machine, run a telnet command and open a connection to the SMT363 module by specifying its IP address. (Also select port 1).

You should see something similar to this in your telnet session window:

```
Connected to echo server @192.128.64.3:1 from 192.128.64.3:1622
Mon Jan 1 00:01:41 1900
```

At this point, any keys you type to the telnet session will be echoed back from the SMT363 network.

```
Connected to echo server @192.128.64.3:1 from 192.128.64.3:1622
Mon Jan 1 00:01:41 1900
ffoorr eexxaamppllee
```

6.6. FTPD EXAMPLE

This example demonstrates an FTP server application. A task on a DSP node initializes the sockets API, and then initializes a daemon to listen on inbound FTP connections. Each daemon listens on a different port number, thus creating a number of independent FTP servers on various ports.

By default, the first ftpd will listen on the canonical FTP control port, 21.

The higher-numbered daemons will listen on ports separated by 1000, respectively.

Build and run ftpd.app and wait until the SMT363 has obtained an IP address.

Now open an FTP session to the IP address of the SMT363. For example:

```
ftp> open 192.128.64.6
Connected to 192.128.64.6.
220 RTEMS FTP server (Version 1.1-JWJ) ready.
User (192.128.64.6:(none)): me
230 User logged in.
ftp>
```

Log in (the user name doesn't matter).

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



Because the example doesn't actually implement a file system on the SMT363 we emulate a simple file system by providing a file "test1". To test, execute a get command like this:

```
ftp> get test1
200 PORT command successful.
150 Opening ASCII mode data connection.
```

6.7. RECV EXAMPLE

This example demonstrates a simple socket receive application. A task on a DSP node initializes the sockets API, and then proceeds to use standard socket semantics to open a port and read a quantity of bytes from a remotely networked machine.

For this example, we used an SMTP mail server's address. You need to change "smt6060-tcpip/examples/recv/dsp/socktest.c".

```
#define CONFIG_INET_ADDR           "207.22.44.49"
#define CONFIG_INET_PORT          "25"
```

to values appropriate for your network. In the above example, the test will open a socket to an SMTP port (mail server) and read the header that is offered. You can make sure that a target machine is ready, by simply telnet-ing to it.

Next, build and run recv.app. You should see output similar to this on your 3L Diamond server.

 A screenshot of a Windows-style terminal window. The title bar reads "#2 Diamond Server: C:\smt6060\smt6060-tcpip-v0p5p7-src\smt6060-tcpip\le...". The menu bar includes "File", "Go", "View", "Board", and "Help". The toolbar contains icons for file operations and execution. The main text area shows the following output:


```
: BEGIN
: sockapi initialized ok
Received: 99 bytes from 192.128.64.129
Received: 220 solar.sundance.com Microsoft ESMT MAIL Service, Versi
Ready
```

 The status bar at the bottom shows "SMT310Q [0]".

Figure 14 - SMT6060-TCPIP RECV Example Output

6.8. RECVC EXAMPLE

This example illustrates a simple receive client. You need to run a sender server on your development machine.

In the folder *smt6060-tcpip/examples/servers/sender* run the command

```
sender 1
sender:  listening on 0.0.0.0:1 ...
```

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



```
sender: connect from 192.128.64.6:1591 [8192 sndbuf]
sender: continuous send
send: No error
sender: 10010624 bytes in 18.000 sec (4.449 Mbps)
sender: stats min=8192 max=8192 avg=8192 cnt=1222
sender:  listening on 0.0.0.0:1 ...
```

This will run the sender server application on your development machine.

Next you need to build the DSP application. Before you do, change the value in recvc.c to the IP address of the development machine where the sender server is running.

```
#define CONFIG_IP_ADDR          "192.128.64.3" // address of server
```

Next, build and run the recvc.app file on the DSP.

```

: BEGIN
: sockapi initialized ok
SO_THROUGHPUT set ...
receiver: connecting to 192.128.64.3:1 ...
receiver: connected to 192.128.64.3:1 [65408 rcubuf]
receiver: 10000000 maximum bytes to recv
recv: no error
receiver: 10000000 bytes in 72.619 sec (1.102 Mbps)
receiver: stats min=892 max=8192 avg=8169 cnt=1224
: DONE

```

Figure 15 -SMT6060-TCPIP Recvc Example Output

The receiver client application will open a connection to the sender server and exchange data with it.

6.9. RECVD EXAMPLE

This example demonstrates a recv() socket server application running on the DSP network. A daemon task on a DSP node initializes the sockets API, and then listens on a socket to inbound TCP/IP connections. Each task listens on a different port number, thus creating a number of independent receiver servers on various ports.

Build and run the recvd.app application on the DSP network. Wait until the SMT363 has obtained an IP address, and the application is running.

Next, in the folder smt6060-tcpip\examples\clients\sender, run sender.exe and specify the IP address of the SMT363 as well as port 1. (The third parameter is the number of bytes to transfer during the test)

```
sender 192.128.64.6 1 100000
```

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



```

sender: connecting to 192.128.64.6:1 ...
sender: connected to 192.128.64.6:1 [8192 sndbuf]
sender: 100000 maximum bytes to send
send: No error
sender: 100000 bytes in 1.000 sec (0.800 Mbps)
sender: stats min=1696 max=8192 avg=7692 cnt=13

```

The Diamond server should be displaying an output similar to this:

 A screenshot of a terminal window titled "#2: Diamond Server: C:\smt6060\smt6060-tcpip-v0p5p7-src\smt6060-tcpip\examples\recvd\output\recvd.app". The terminal shows the following output:


```

: BEGIN
: sockapi initialized ok
SO_THROUGHPUT set ...
receiver: listening on 0.0.0.0:1 ...
SO_THROUGHPUT set ...
receiver: connect from 192.128.64.3:1143 [65408 rcvbuf]
receiver: continuous recv
recv: no error
receiver: 100000 bytes in 0.916 sec (0.873 Mbps)
receiver: stats min=892 max=8192 avg=6258 cnt=16
receiver: listening on 0.0.0.0:1 ...

```

 The terminal window has a standard Windows-style title bar and a status bar at the bottom showing "Ready", "Running", and "SMT310Q [0]".

Figure 16 - SMT6060-TCPIP Recvc Example Output

The sender client application connects to the recvd server application running on the SMT363 and exchanges data with it.

6.10. SENDC EXAMPLE

This application is a simple sender client implemented on the DSP network. To run, make sure you have the receiver server running on your development machine.

```
smt6060-tcpip\examples\servers\receiver
```

```

receiver 1
receiver: listening on 0.0.0.0:1 ...
receiver: connect from 192.128.64.6:1646 [8192 rcvbuf]
receiver: continuous recv
recv: No error
receiver: 10000000 bytes in 7.000 sec (11.429 Mbps)
receiver: stats min=292 max=4380 avg=1453 cnt=6879
receiver: listening on 0.0.0.0:1 ...

```

The client application running on the DSP will connect to a sender server to exchange data with it.

In sendc.c, change the value

```
#define CONFIG_IP_ADDR          "192.128.64.3" // address of server
```

to the address of where the receiver server is running, and rebuild the DSP application.

Now run sendc.app.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



```

#3 Diamond Server: C:\smt6060\smt6060-tcpip-v0p5p7-src\smt6060-tcpip\examples\sendc\output\sendc.app
File Edit View Board Help
[Icons]
: BEGIN
: sockapi initialized ok
SO_THROUGHPUT set ...
sender: connecting to 192.128.64.3:1 ...
sender: connected to 192.128.64.3:1 [65408 sndbuf]
sender: 10000000 maximum bytes to send
send: no error
sender: 10000000 bytes in 7.275 sec (10.997 Mbps)
sender: stats min=57984 max=65408 avg=65359 cnt=153
: DONE
Ready
SMT310Q [0]

```

Figure 17 - SMT6060-TCPIP Sendc Example Output

6.11. SENDD EXAMPLE

This example demonstrates an send() socket server application. A daemon task on a DSP node initializes the sockets API, and then listens on a socket to inbound TCP/IP connections. Each task listens on a different port number, thus creating a number of independent send servers on various ports.

Build and run the DSP application.

```

#4 Diamond Server: C:\smt6060\smt6060-tcpip-v0p5p7-src\smt6060-tcpip\examples\sendd\output\sendd.app
File Edit View Board Help
[Icons]
: BEGIN
: sockapi initialized ok
SO_THROUGHPUT set ...
sender: listening on 0.0.0.0:1 ...
SO_THROUGHPUT set ...
sender: connect from 192.128.64.3:1158 [65408 sndbuf]
sender: continuous send
send: unknown error
sender: 65408 bytes in 0.058 sec (10.465 Mbps)
sender: stats min=65408 max=65408 avg=65408 cnt=1
sender: listening on 0.0.0.0:1 ...
Ready
Running
SMT310Q [0]

```

Figure 18 -SMT6060-TCPIP Sendc Example Output

smt6060-tcpip\examples\clients\receiver

Now run a receiver client instance on your development machine, and specify the IP address of the SMT363. (The third parameter is the number of bytes to use during the test)

```

receiver          192.128.64.6          1          10000
receiver:         connecting          to          192.128.64.6:1          ...
receiver:         connected          to          192.128.64.6:1          [8192          rcvbuf]
receiver: 10000 maximum bytes to recv
recv: No error
receiver: 10000 bytes in 0.000 sec (0.000 Mbps)
receiver: stats min=1240 max=1460 avg=1428 cnt=7

```

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



During this test, the receiver client application connected with the sender server application and exchanged data with it.

6.12. MULTIC EXAMPLE

This is a multi client example. It make use of the sendc and recvc tasks, so you need to have build those before attempting to build this example. The configuration file for this example places a single instance of both the sendc and recvc tasks on the embedded network.

You need to run the sender and receiver server programs on your development machine.

```
sender 12601
sender: listening on 0.0.0.0:12601 ...
sender: connect from 192.128.64.6:2122 [8192 sndbuf]
sender: continuous send

receiver 12701
receiver: listening on 0.0.0.0:12701 ...
receiver: connect from 192.128.64.6:1865 [8192 rcvbuf]
receiver: continuous recv
```

Build and run the multic application. You should see output similar to that shown below.

 A screenshot of a terminal window titled "#3 Diamond Server: C:\sm6060\sm6060-1cpi\p-v0p5p7-src\sm6060-1cpi\p\examples\multic\output\multic2...". The terminal shows the following output:


```
: BEGIN
: BEGIN
: sockapi initialized ok
: sockapi initialized ok
SO_THROUGHPUT set ...
SO_THROUGHPUT set ...
sender: connecting to 192.128.64.3:12601 ...
receiver: connecting to 192.128.64.3:12701 ...
sender: connected to 192.128.64.3:12601 [65408 sndbuf]
receiver: connected to 192.128.64.3:12701 [65408 rcvbuf]
sender: 10000000 maximum bytes to send
receiver: 10000000 maximum bytes to recv
```

 The terminal window has a status bar at the bottom showing "Ready", "Running", and "SMT310Q [0]".

Figure 19 - SMT6060-TCPIP Multic Example Output

Note that the port values 12601 and 12701 are specified in the configuration file for the application. The technique used is to bind values to the input ports of a task. Refer to the 3L Diamond user guide for more information.

6.13. MULTID EXAMPLE

This example illustrates a multi daemon application running on the embedded system. It uses the sendd and recvd tasks, so please ensure that you have build those prior to building this application.

Build and run the multid application.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



```

#4 Diamond Server: C:\smt6060\smt6060-tcipip-v0p5p7-src\smt6060-tcipip\examples\multid\output\multid...
Go View Board Help
: BEGIN
: BEGIN
: sockapi initialized ok
: sockapi initialized ok
SO_THROUGHPUT set ...
SO_THROUGHPUT set ...
receiver: listening on 0.0.0.0:12301 ...
sender: listening on 0.0.0.0:12401 ...
SO_THROUGHPUT set ...
sender: connect from 192.128.64.3:1248 [65408 sndbuf]
sender: continuous send
SO_THROUGHPUT set ...
receiver: connect from 192.128.64.3:1250 [65408 rcvbuf]
receiver: continuous recv
Ready
SMT330Q [0]

```

Figure 20 - SMT6060-TCPIP Multid Example Output

Take note of the IP address that has been assigned to the SMT363 module, by looking at the output displayed on the HyperTerminal window.

Run an instance of the sender client, and specify the IP address of the SMT363 and port 12401 as shown.

```

sender 192.128.64.6 12401
sender: connecting to 192.128.64.6:12401 ...
sender: connected to 192.128.64.6:12401 [8192 sndbuf]
sender: continuous send

```

Likewise, run an instance of the receiver client as shown

```

receiver 192.128.64.6 12301
receiver: connecting to 192.128.64.6:12301 ...
receiver: connected to 192.128.64.6:12301 [8192 rcvbuf]
receiver: continuous recv

```

Note that the port numbers 12401 and 12301 are specified in the configuration file of this application.

6.14. SMT363 AS NETWORK INTERFACE EXAMPLE

The folder `\appnotes\net2dsp` contains an example application that configures the SMT363 as a network interface so that a programming running elsewhere on the network can remotely boot and run programs on the embedded network. The Net2Dsp application has been developed with the SMT6060.

The idea is that the SMT363 becomes a “dumb” module that does not participate in the Diamond network directly. The SMT363’s flash is programmed with the net2dsp application, and this application starts running immediately once the module has been reset. One of the comports of the SMT363 is selected to be used to talk to the rest of the embedded network. This essentially replaces the “host comport”.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com

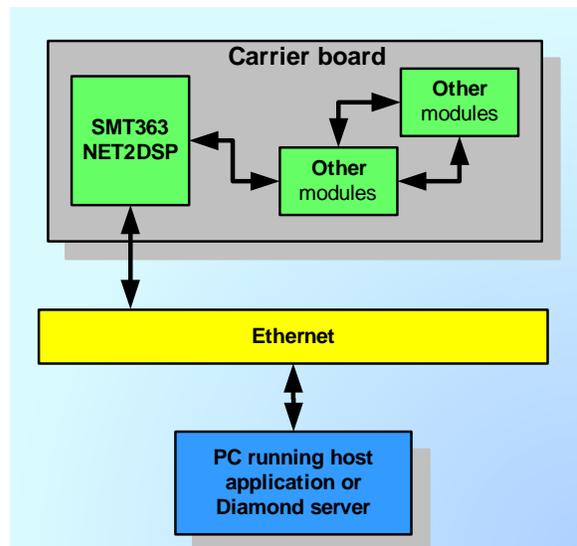


Figure 21 - SMT6060-TCPIP SMT363 Configuration as Network Interface

Note: This technique can be used to run the Diamond server on a notebook computer connected to the Ethernet of a stand-alone carrier board.

The net2dsp application implements a simple server that accepts connections from network applications. The network application then sends instructions to the server. The instructions could be:

1. Write data to a comport of the SMT363
2. Read data from a comport of the SMT363

All transactions take the form of a data structure like this:

```
typedef struct {
    int device;
    int cmd;
    int bytes;
} command;
```

“**device**” refers to which Diamond channel (and indirectly which comport) should be used for the operation.

“**cmd**” indicates read or write from device. (0 = read, and 1 = write)

“**bytes**” selects how many bytes to read or write.

6.15. CONFIGURING THE HOST

The host application is build into the Sundance host driver software. You activate the host functionality by making a number of entries into the registry of the machine where you wish to run the host software.

The key

HKEY_LOCAL_MACHINE\SOFTWARE\SmtDrv\ethernet

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



Shown below needs to contain the following values:

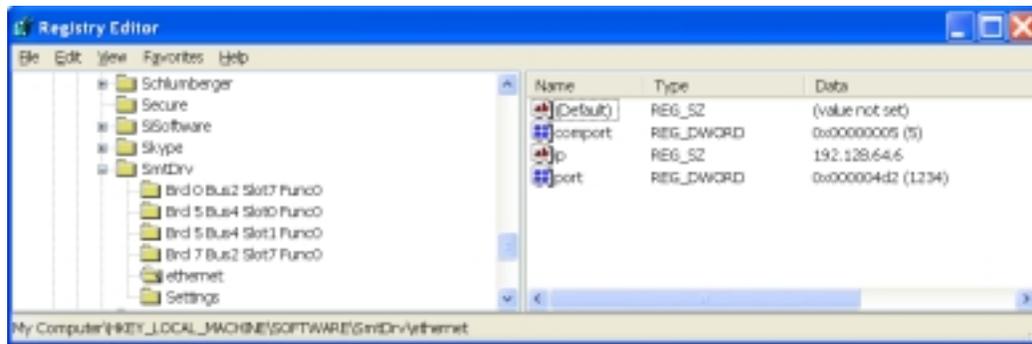


Figure 22 - SMT6060-TCPIP Configuring the Host

“comport” specifies which of the SMT363 comports you are using as a host comport. Valid values are 1,2,3 or 4.

“ip” specifies the IP address of the SMT363 module that you wish this host application to connect to.

“port” must be 1234 decimal.

When you have these values configured on the host machine, you should see a new hardware board called “Ethernet” appear. For example.

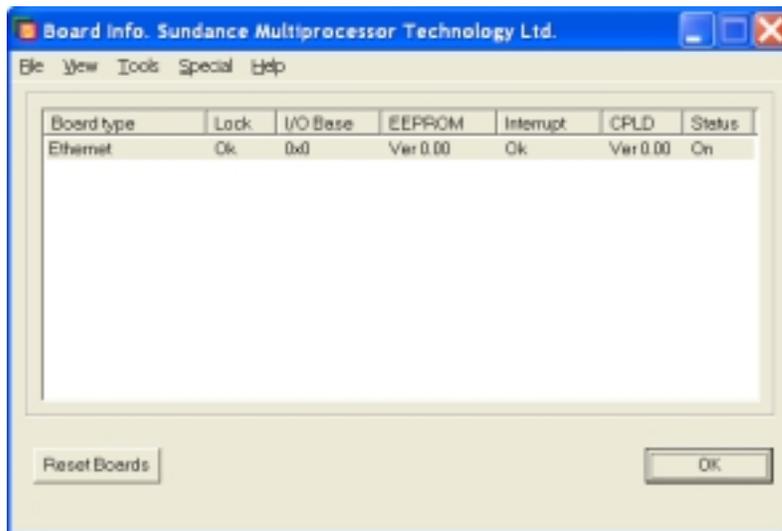


Figure 23 - SMT6060-TCPIP Boardinfo Host Configuration

Any access you now make from a host program is now going to go through the Ethernet to the SMT363 comport you selected. Note that reset is not implemented, and that you will need to ensure that you reset the embedded system each time before loading an application to it. (This is typically done with a switch on the stand alone board.)

Note. The net2dsp application can be provided to customers that do not use 3L Diamond.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



6.16. CONFIGURING THE SMT363

To enable the SMT363 to automatically run the net2dsp application when the module is reset, use the SMT6001 flash programming utility from Sundance, and add the net2dsp.app file.

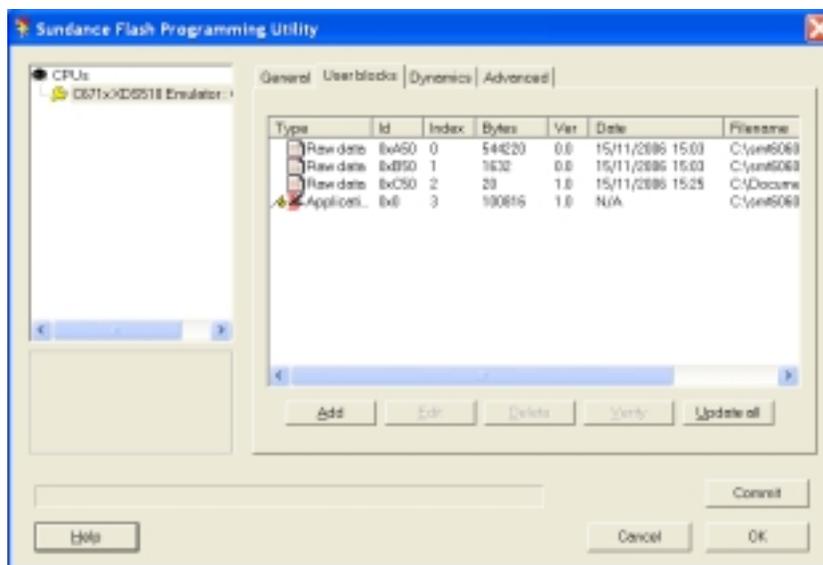


Figure 24 - SMT6060-TCPIP Using the Flash Utility

Mark the entry as bootable to ensure that it is loaded automatically when the module is reset.

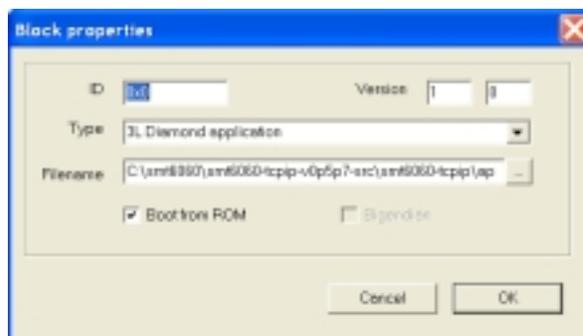


Figure 25 - SMT6060-TCPIP Marking the Bootable file

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com



7. ACCESS TO THE SERIAL PORT OF THE NET50

SockAPI.lib provides the following functions to access the serial port of the NET50.

```
int serial232_port_in(char * buf, int bytes);  
int serial232_port_printf( const char *fmt, ... );  
int serial232_port_out(const char *B, int bytes);
```

These functions can be used by any task on the network of DSPs to access the serial RS232 port of the NET50. This allows developers to output debug information on the terminal window connected to the RS232.

Sundance Digital Signal Processing Inc.

4790 Caughlin Parkway 233, Reno, NV 89519-0907, U.S.A.

Tel: +1 (775) 827-3103, Fax: +1 (775) 827-3664, email: sales@sundancedsp.com

www.sundancedsp.com