FASTCOM™ ADAPTERS

FASTCOM™: ESCC-104

High Speed Dual Channel Sync/Async Interface
for PC/104 Bus
Hardware Reference Manual

Manufactured by:

COMMTECH

COMMTECH

http://www.commtech.com.cn/

REVISION NOTES

REVISION	PAGE NUMBER	CHANGES MADE
2.1	30	Changed warranty to 2 years
2.2	30	Updated contact information
2.3	30	Changed warranty period to lifetime
2.4	23	Fixed BDF error

CONTENTS

INTRODUCTION

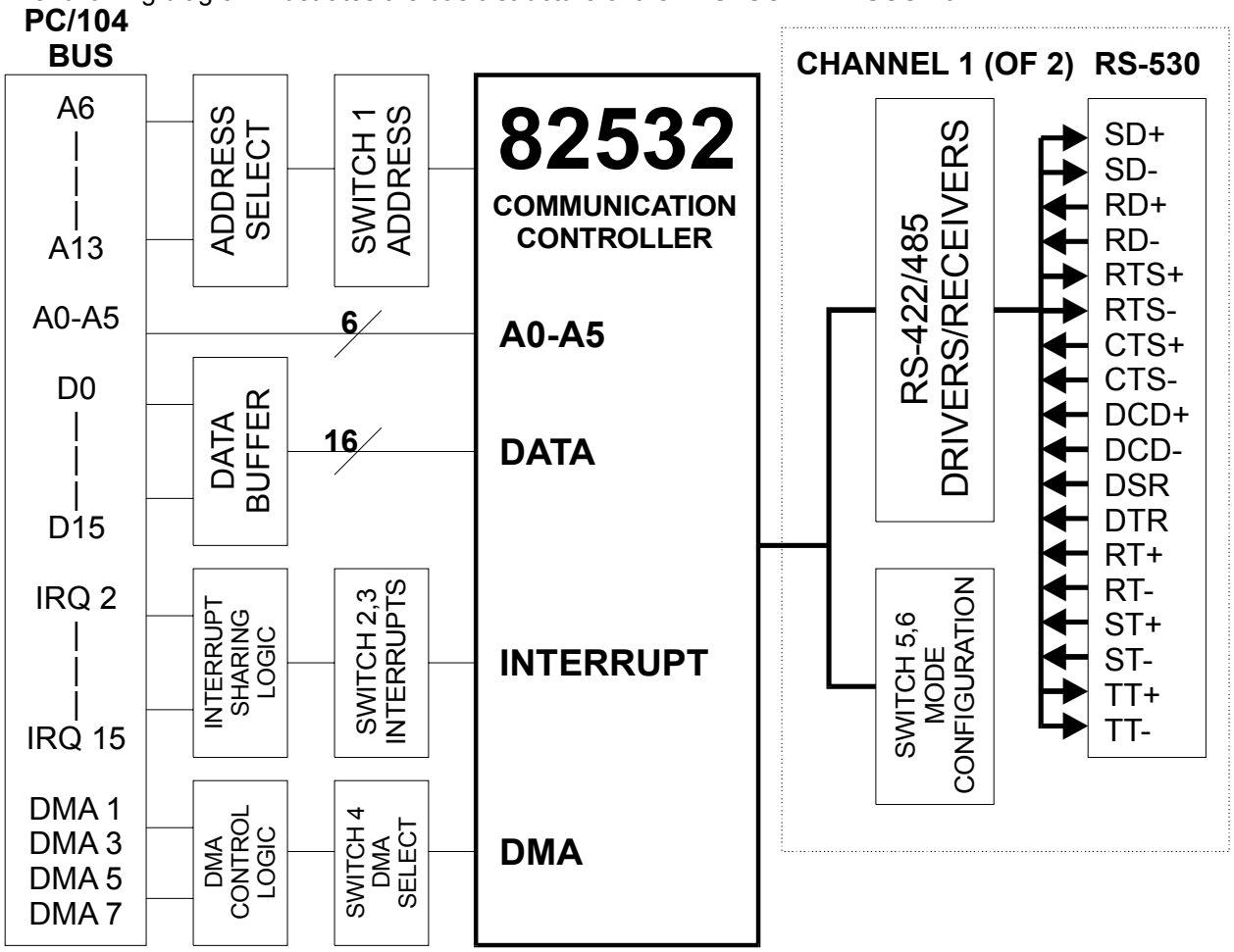
The new FASTCOM™: ESCC-104 is a very high-speed, dual channel, synchronous/asynchronous serial communications adapter based upon the Siemens 82532 Enhanced Serial Communication Controller (ESCC).

The FASTCOM™: ESCC-104 is designed to support data rates up to 10 Mbits/second (maximum data rates are affected by many factors, including computer performance, cable quality, and software overhead), and to reduce the hardware and software overhead needed for serial communications. Each sync/async channel on the FASTCOM™: ESCC-104 has its own DPLL encoder/decoder and programmable protocol support. In addition, a built-in 64 byte FIFO provides the FASTCOM: ESCC-PCI with a very high throughput as well as requiring less system CPU time than any other HDLC adapter. The FASTCOM™: ESCC-104 directly supports HDLC, X.25 LAP B, ISDN LAP D, SDLC, ASYNC, and BISYNC protocols, and features a high speed RS-422/RS-485 interface conforming to ANSI/EIA/TIA-530-A-1992 configuration (RS-530). HDLC features include choice of CRC polynomial (CRC-CCITT or CRC-32), expanded line encoding methods (FM and Manchester), and preamble transmission.

Many engineers have avoided using synchronous communication adapters because of their programming complexity. The FASTCOM™: ESCC-104 provides high-speed data communications to designers and engineers, while *greatly reducing development time* and system complexity.

The FASTCOM: ESCC-104 is also available in a PCI bus version (FASTCOM: ESCC-PCI) and an ISA bus version (FASTCOM: ESCC-ISA).

The following diagram illustrates the basic structure of the FASTCOM™: ESCC-104:



SPECIFICATIONS:

COMMUNICATION CONTROLLER:	SIEMENS 82532
DRIVERS/RECEIVERS:	RS-422/RS-485
CONNECTOR CONFIGURATION:	ANSI/EIA/TIA-530-A-1992
POWER REQUIREMENTS:	+5V @ 300mA (TYPICAL)
BUS INTERFACE:	PC/104
ENVIRONMENT:	
Operating Temperature Range:	0 to 70 C
Humidity:	0 to 90% (non-condensing)

FEATURES:

High speed, up to 10Mbits/s (dependent upon system capabilities)
Much easier to program and use than other HDLC adapters
Supports HDLC, SDLC, ISDN LAP D, and X.25 LAP B, ASYNC, BISYNC

Drivers: RS-422/RS-485 multi-drop

Excellent noise rejection, cable lengths up to 4000 feet
Use low cost "twisted pair" cable
RS-485 mode
Up to 32 FASTCOM™: ESCC-104 adapters can share the same "twisted pair"
Driver control is automatic (via the RTS line)

Serial Interface:

Internal or External Clock Source
Asynchronous, Monosync/Bisync, and HDLC/SDLC data formatting.
1X (isosynchronous) or 16X oversampling for Asynchronous format
Different modes of data encoding (NRZ,NRZI,FM0,FM1,Manchester)
CRC-CCITT or CRC-32 (for HDLC/SDLC modes)
CRC-CCITT or CRC-16 (for BISYNC mode)
Modem control lines (RTS, CTS, DTR, DCD, DSR)
Collision resolution
Programmable bit inversion
Transparent RD/SD of data bytes without HDLC framing

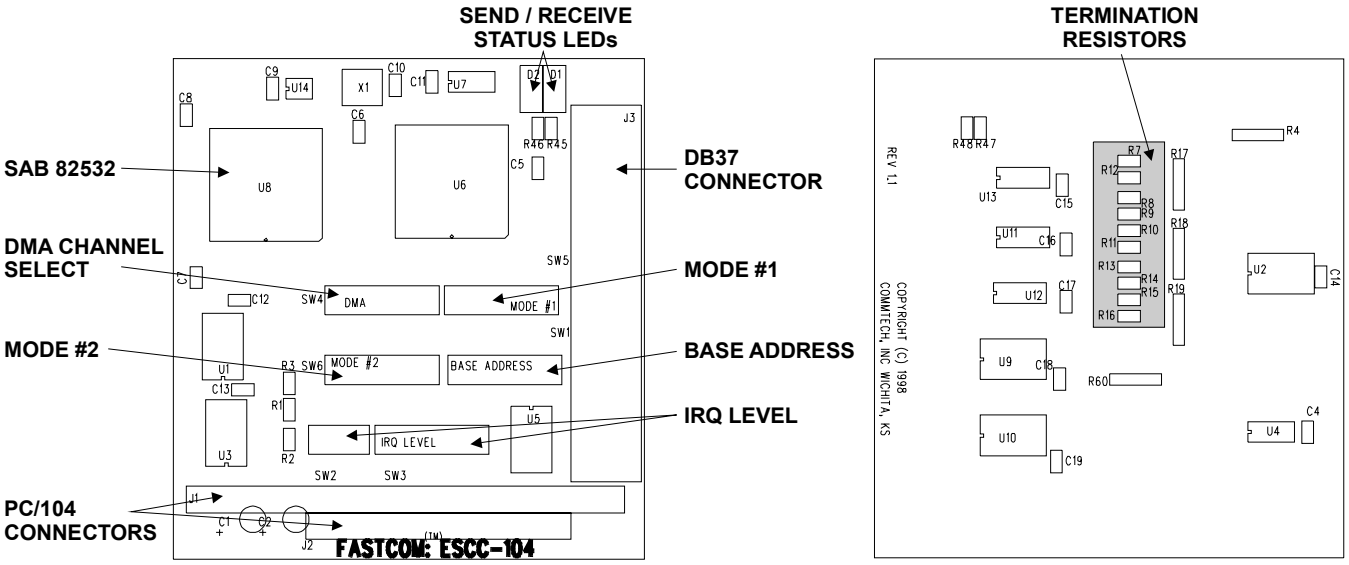
Protocol Support (HDLC/SDLC):

Types of protocol support - Automatic, Manual, Transparent
Handling of bit-oriented functions in all modes
Handling of I and S frames in Auto mode
Modulo 8 and 128 operation

64 byte FIFOs per direction
Storage of up to 17 short received frames

FASTCOM: ESCC-104

BOARD LAYOUT

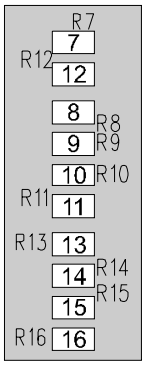


TERMINATION REFERENCE:

	SIGNAL	REF
CHANNEL 1:	RT	R12
	RD	R13
	CTS	R14
	ST	R15
	DCD	R16
CHANNEL 2:	RT	R7
	RD	R8
	CTS	R9
	ST	R10
	DCD	R11

LED INDICATORS:

RED	TRANSMIT ACTIVE	
GREEN	RECEIVE ACTIVE	

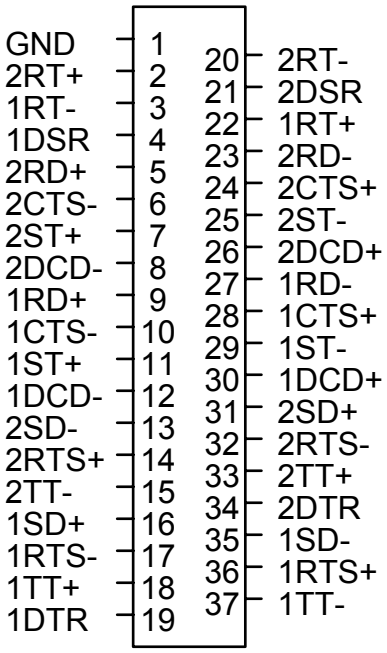


PACKING LIST:

FASTCOM: ESCC-104 CARD
CABLE ASSEMBLY
FASTCOM CD

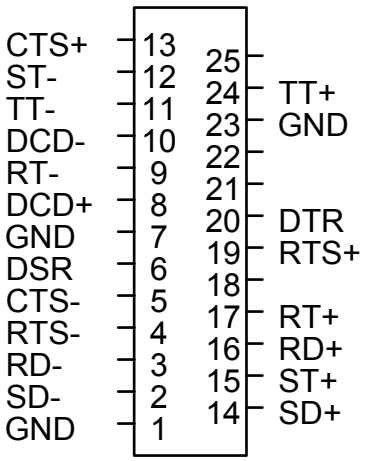
If an omission has been made, please call technical support for a replacement.

FASTCOM: ESCC-104
DB37 FEMALE CONNECTOR PIN DESCRIPTION



The cable provided splits each channel from this DB37 to individual RS-530 pin out DB25 male connectors. The DB25 pin outs are shown on the next page.

FASTCOM: ESCC-104
DB25 CABLE CONNECTOR
DESCRIPTION (RS-530 pin out)



PIN DESCRIPTIONS

PIN#	DESCRIPTION	422 TYPE	CONNECTED TO	530 CIRCUIT
1	SHIELD/GROUND		GND	
7	SIGNAL/GROUND		GND	AB
2	TRANSMIT DATA	A	SD-	BA
14	TRANSMIT DATA	B	SD+	BA
3	RECEIVE DATA	A	RD-	BB
16	RECEIVE DATA	B	RD+	BB
4	REQUEST TO SEND	A	RTS-	CA
19	REQUEST TO SEND	B	RTS+	CA
5	CLEAR TO SEND	A	CTS-	CB
13	CLEAR TO SEND	B	CTS+	CB

The RS-422 "A" side is 0 volts when a "1" is on the line (A side is the negative terminal of the 422 driver).

6	DATA SET READY	DSR	INPUT	
20	DATA TERMINAL READY	DTR	OUTPUT	

CLOCK SIGNALS

PIN#	DESCRIPTION	422 TYPE	CONNECTED TO	530 CIRCUIT
8	DATA CARRIER DETECT	A	DCD+	CF
10	DATA CARRIER DETECT	B	DCD-	CF
24	TRANSMIT CLOCK OUT	A	TT+	DA
11	TRANSMIT CLOCK OUT	B	TT-	DA
17	RECEIVE CLOCK IN	A	RT+	DD
9	RECEIVE CLOCK IN	B	RT-	DD
15	TRANSMIT CLOCK IN	A	ST+	DB
12	TRANSMIT CLOCK IN	B	ST-	DB

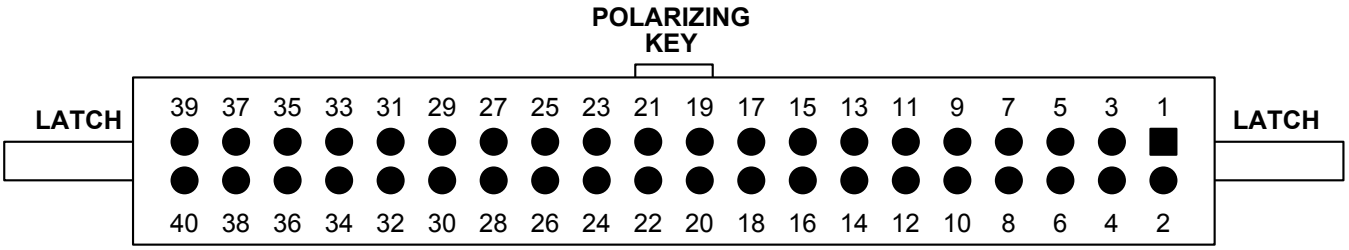
These are *opposite* of RS-422 in order to invert the signals going to the FASTCOM™: ESCC-104. The 530 specification states that data is valid on the rising edge of the clock (0->1 transition), whereas the FASTCOM™: ESCC-104 clocks on the falling edge (1->0 transition). DCD is inverted because a "1" at DCD means active to the FASTCOM™: ESCC-104, but the 530 specification requires DCD to be "0" to be active.

On the same note, if you are using Clock Mode 1 and you are not using DCD, it must be strapped Active to allow data reception.

NOTE: The DTR and DSR signals are single ended (unbalanced) as required by ANSI/EIA/TIA-530 specifications.

FASTCOM: ESCC-104-02
40 PIN LATCH-LOCK HEADER DESCRIPTION

The Fastcom: ESCC-104 is available with a 40-pin latch-lock header in place of the DB37 connector. This option allows for the use of a ribbon cable in place of the DB style shielded cable normally supplied. Following is the pin-out description of the 2x20 header.



PIN #	SIGNAL	PIN #	SIGNAL
1	GND	21	1ST-
2	2RT+	22	1DCD+
3	2RT-	23	1DCD-
4	2DSR	24	2SD+
5	1RT-	25	2SD-
6	1RT+	26	2RTS+
7	1DSR	27	2RTS-
8	2RD+	28	2TT+
9	2RD-	29	2TT-
10	2CTS+	30	2DTR
11	2CTS-	31	1SD-
12	2ST+	32	1SD+
13	2ST-	33	1RTS-
14	2DCD+	34	1RTS+
15	2DCD-	35	1TT-
16	1RD+	36	1TT+
17	1RD-	37	1DTR
18	1CTS+	38	N/C
19	1CTS-	39	N/C
20	1ST+	40	GND

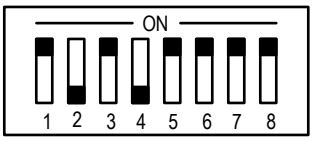
INSTALLATION

Important: Observe Electrostatic Discharge (ESD) precautions when handling the FASTCOM™: ESCC-104 board.

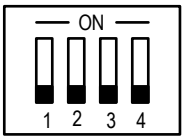
- Unpack the FASTCOM™: ESCC-104. Keep the box and static bag for warranty repair returns.
- Check the switches to be sure that they are set as illustrated below (Factory Switch Settings).

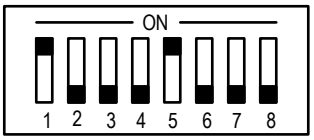
FACTORY SWITCH SETTINGS

ADDRESS SELECT (SW1) - 280H

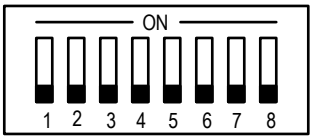


IRQ SELECT (SW2 & SW3) - 5

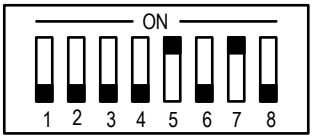




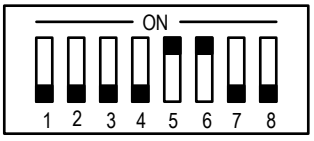
DMA SELECT (SW4) - NO DMA



MODE #1 CONFIG (SW5) - RS-422, DMA DISABLED, TXCLK IS OUTPUT



MODE #2 CONFIG (SW6) - NO LOOPBACK (485) CTS DISABLED
422 CLOCK MODE



INSTALLING THE WINDOWS NT ESCC DRIVER

The NT driver for the ESCC is the ESCCDRV.SYS file. It is a kernel mode driver. The NTINSTALL.EXE program is intended to simplify the process of adding registry information about the base address, interrupt, and DMA information about each channel.

INSTALL DRIVER: Pressing this button will copy the ESCCDRV.SYS file to your C:\winnt\system32\drivers subdirectory. It will create the following subkey in the registry:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\esccdrv

In that key it adds the following information:

DisplayName: REG_SZ:esccdrv
ErrorControl: REG_DWORD:0x01
ImagePath: REG_EXPAND_SZ:\??\C:\WINNT\System32\Drivers\esccdrv.sys
Start: REG_DWORD: 0x02
Type: REG_DWORD: 0x01

- These values are actually created by the CreateService API call.

- The C:\WINNT part of the image path will (should) reflect your installed system directory (i.e., the system drive and where your NT files are installed), and could be different depending on how you originally installed NT.

- The Start parameter is set to automatic. This will load and run the driver every time that NT is started. If you want to start and stop it manually, you can change the start type by using the Control Panel -> Devices -> Startup button after the driver is installed.

- The Type parameter is kernel mode.

ADD BOARD: Pressing this button will add the information to the registry about where a particular ESCC channel is located. It will create a subkey under the esccdrv\parameters such as esccdrv\parameters\ESCC0.

The dialog that appears when you press ADD BOARD needs to have the Base Address, Interrupt, and DMA information for the card you are installing. For example, if you use the factory defaults (do not change any of the switch settings), you would press the ADD BOARD button two times. The first time you would enter:

Base = 0x280
IRQ = 5
DMAR = 0
DMAT = 0
Channel = 0

The second time you would enter:
Base = 0x280
IRQ = 5
DMAR = 0
DMAT = 0
Channel = 1

This will create two subkeys esccdrv\parameters\ESCC0 and esccdrv\Parameters\ESCC1. The value that you enter for Base should match setting of the base address switch on the board. The value that you enter for IRQ should match the setting of the IRQ LEVEL switch on the board.

The value(s) that you enter for DMAR and DMAT will determine if the driver uses DMA or not. If they are both zero, then DMA is not used (interrupt only mode). If they are both nonzero and not equal (i.e., DMAR!=DMAT!=0), then DMA mode is used.

The example program is compiled assuming that DMA is not used (i.e., DMAR=DMAT=0). Specifically, the DMA bit in the XBCH register of the 82532 is clear, indicating that the 82532 should operate in “interrupt only” mode.

You can manually add/delete these registry entries if you wish. But earlier version(s) of the driver will crash if you do not define any boards and start/stop/start the driver.

You can manually add a value in the esccdrv\parameters\ESCCx subkey:
Buffers:REG_DWORD: 0xYY
This controls the number of receive buffers (frames) that the driver will allocate and use. The default value is 10 (0x0a) and can be any value from 2 to 100.

REMOVE BOARD: Pressing this button will delete the subkey (and its values) for the selected board (channel).

START: Pressing this button will load the driver. It is equivalent to using Control Panel -> Devices -> Start with the esccdrv selected.

STOP: Pressing this button will stop/unload the driver. It is equivalent to using Control Panel -> Devices -> Stop with the esccdrv selected.

REMOVE DRIVER: Pressing this button will attempt to stop the driver (if it is started) and will delete the esccdrv subkey and all of its subkeys. It will not delete the esccdrv.sys file from your winnt\system32\drivers directory. If you want to completely remove the driver you must delete this file manually.

The intended sequence of events is:

Press Install Driver button
Press Add Board button (2 times per board installed)
Fill in address/irq/dma information
Press Start Esccdrv button
Press OK/Exit

If you want to change/add a board later, you would need to follow the following sequence:

Press Stop Esccdrv button
Press Add Board/Remove Board as necessary
Press Start Esccdrv button
Press OK/Exit.

If you do not have any boards (channels) installed and press the start button, you must manually delete the esccdrv.sys and the esccdrv subkey before rebooting or any attempt to stop the driver could cause NT to go into its "bug check" mode (Blue Screen of Death).

The NTINSTALL.EXE program does not do any checking on the values that you enter. If you enter unreasonable values for address/irq/dma, there is a very good chance that NT will become unstable ("bug check" mode).

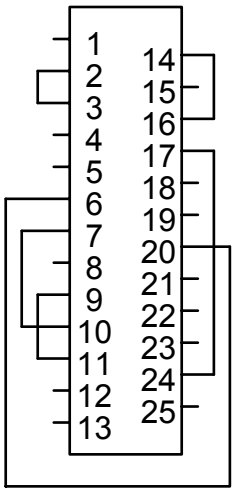
The driver only looks at the registry information on startup. If you add boards while the driver is running, the new devices will not exist until the system is restarted or you stop and restart the driver. For example if you:

Press Install Driver,
Press Add Board (enter channel 0 info),
Press Add Board (enter channel 1 info),
Press Start Esccdrv,
Press Add Board (enter channel 0 info),
Press Add Board (enter channel 1 info),
Run example program (esccmfc),

The only devices that would be available would be ESCC0 and ESCC1. If you try to open ESCC2 or ESCC3, you will get a “can’t get a handle” message. To use ESCC2 or ESCC3, you must press the STOP ESCCDRV button, then press the START ESCCDRV button (or reboot) to allow the driver to recognize the additional board.

TESTING THE INSTALLATION

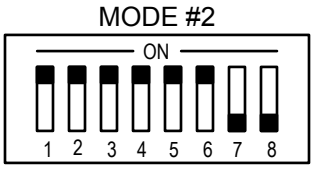
To fully test the installation of your FASTCOM™: ESCC-104, you will need to build a "loop back plug". Materials needed are a DB25 female receptacle (solder-cup style) and a few short pieces of 20 or 24 AWG stranded wire. Jumper the pins together on the DB25 as illustrated below:



SIGNALS

2	SD-
3	RD-
6	DSR
7	GND
9	RT-
10	DCD-
11	TT-
14	SD+
16	RD+
17	RT+
20	DTR
24	TT+

NOTE: You can also create a "loop back" condition on the FASTCOM™: ESCC-104 without building an adapter plug by setting the Mode #2 Switch (SW6) as follows:



Both ESCC channels are looped back

However, we recommend that you make the "loop back" plug for two reasons: one, the clock circuit can be tested and, two, you will not have to change switch settings from the factory defaults. In addition, our technical support engineers can better service your technical questions if you have made the "loop back" plug. **If you have made the loop back plug, do not change the setting of the Mode switch.**

FASTCOM: ESCC-104 WINDOWS NT TEST

1. Make sure that the driver was installed with DMAT=DMAR=0, as the test program was compiled to run in interrupt mode
2. Attach a loopback plug to the cable #1 (ESCC0, port 0)
3. From the Start button menu, select Run
   Enter: D:\fastcom_disks\escc\nt\escctest\escctest 0 h
   Click the OK button
4. You should see:

```
Created esccdrv—ESCC0
ESCC 82532 version status:82
# receive buffers ready:0
resetting
HDLC settings
SETTINGS SUCCESSFUL:168
DTR SET
DSR not SET
DTR not SET
DSR not SET
waiting for a key
read thread started
status thread started
```

(The first DSR not SET could be a DSR SET if you have a DTR -> DSR loopback wired.)

5. Press the letter "a" on the keyboard. You should see:

```
WRITEFILE esccdrv1024
returned TRUE
```

waiting for a key
STATUS, Receive Frame Start
STATUS, All Sent
received 1025 bytes:
aaaaaaaaaaaaa...(12 complete lines and one partial line of "a"s (1024 of them). The last character is not an "a".)

6. Press [esc] to exit the program.

You can test channel 1 in a similar manner by running
D:\fastcom_disks\escc\nt\escctest\escctest 1 h

Make sure that you move your loopback to cable #2 before running the test on channel 1.

You can test other operating modes by changing the last letter:

async test	D:\fastcom_disks\escc\nt\escctest\escctest 0 a
HDLC test	D:\fastcom_disks\escc\nt\escctest\escctest 0 h
Bisync test	D:\fastcom_disks\escc\nt\escctest\escctest 0 b

In async you will get a STATUS, Receive Timeout after the All sent message (and possibly at the beginning before you press a key).
The bisync test will get a STATUS, SYN detected instead of a receive frame start message.
The async test should receive 1024 bytes, displayed as 12.8 lines of the key you pressed.
The HDLC test should receive 1025 bytes, displayed as 12.8 lines of the key you pressed.
The bisync test should receive 1025 bytes, displayed as 12.8 lines of the key you pressed.

The exceptions to this are the keys t, r, i, p and h.
pressing "t" will reset the transmitter and flush the transmit queue
pressing "r" will reset the receiver and flush the receive queue
pressing "i" will start the timer (which will eventually result in a STATUS, Timer expired message. It takes about a minute in HDLC mode for the timer to timeout)
pressing "p" will stop the timer (which will prevent the STATUS, Timer expired message).
pressing "h" will issue a hunt command in bisync mode.

TEST #2

1. Press the Start Button, select the Run command
   Enter: D:\fastcom_disks\escc\nt\esccmfc\esccmfc
   Click OK

2. From the main menu select Options -> Port
   Enter 0, click OK (this selects port 0, make sure you have your loopback on cable #1)

3. From the main menu select Options -> Settings (the settings dialog will open)
   Click OK, the TXD status indicator on the screen should turn green.

4. Type a short message on the keyboard, press enter to send it.

The message you typed should appear in the lower window, and the RXD, RFS, and ALLS status indicators should turn green (if it was a short message (<32 bytes or so); a long message will likely only get a RXD status indicator).

If when running any of these tests you do not get the expected result, check your switch settings and the driver install settings to make sure that they match. If it still doesn't work, try a different address/irq combination (to remove a possible address or irq conflict).

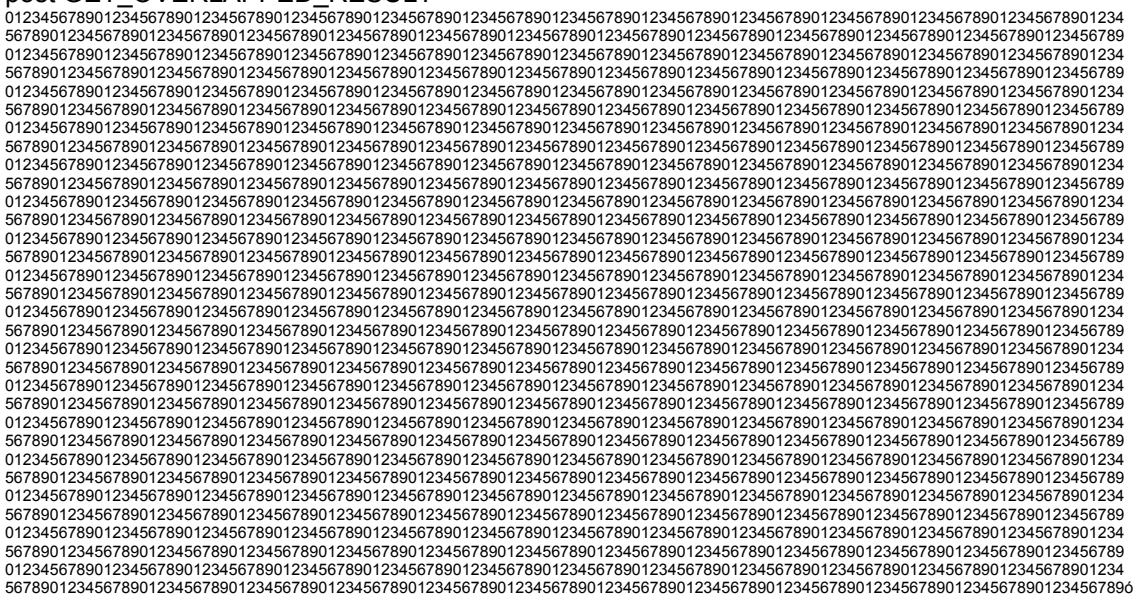
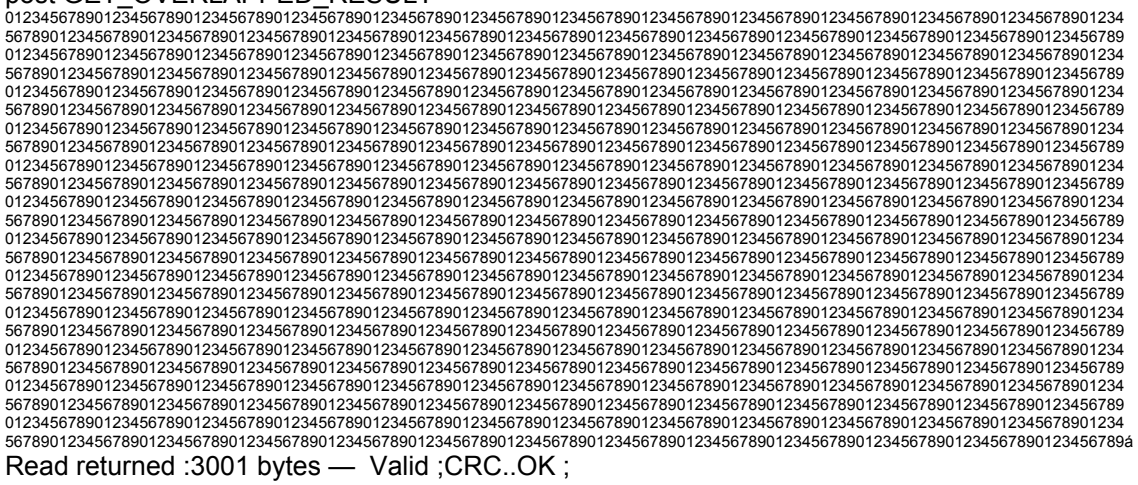
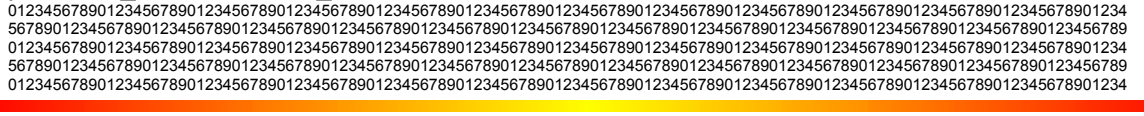
Troubleshooting tips:

1. Incorrect loopback, faulty wiring
2. Interrupt conflict
3. Address conflict

FASTCOM: ESCC-104 WINDOWS 95 TEST

1. Install loopback on cable #1 (ESCC0, port0)
2. From the Start button menu, select Run
3. Enter:
   D:\fastcom_disks\escc\w95\escctest
   Click OK
4. The display should be:

```
obtained handle to esccdrv
try to add port
port added
#ports active:2
Set ICD2053b
ICD2053B set to 16MHz
starting settings......INIT OK
Getting VSTR
VSTR contents:82
WRITE REG (CMDR->XRES)
Write Reg successful
check status
STATUS DECODE:
Transmit Done
flushing rx
RX flushed
flushing tx
TX flushed
read frame
The I/O operation is pending
write frame #1
Write successful
WAIT_OBJECT_0
post GET_OVERLAPPED_RESULT
012345678901234567890123456ó
Read returned :26 bytes —  Valid ;CRC..OK ;
write frame #1
Write #1successful
write frame #2
Write #2 successful
write frame #3 —should pend (no more tbufs)
The I/O operation is pending
write frame #4 —should fail (bufs full)
Tx buffers full; try again later.
read frame (should get 4001 bytes)
The I/O operation is pending
Read Timeout
```

WAIT_OBJECT_0
post GET_OVERLAPPED_RESULT

Read returned :4001 bytes — Valid ;CRC..OK ;
check status
STATUS DECODE:
Receive Frame Start
All Sent
read frame(should get 3001 bytes)
The I/O operation is pending
Read Timeout
WAIT_OBJECT_0
post GET_OVERLAPPED_RESULT

Read returned :3001 bytes — Valid ;CRC..OK ;
check status
STATUS DECODE:
Receive Frame Start
Transmit Done
read frame(should get 2001 bytes)
The I/O operation is pending
WAIT_OBJECT_0
post GET_OVERLAPPED_RESULT


```
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234
567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789ó
Read returned :2001 bytes —  Valid ;CRC..OK ;
check status
STATUS DECODE:
All Sent
Transmit Done
Press a key to send again, [esc] to exit
exiting prog
```

This program assumes the factory default switch settings (address 0x280, IRQ 5). If you do not have those resources available, you must recompile the example with the address and IRQ to match your board settings.

Troubleshooting tips:

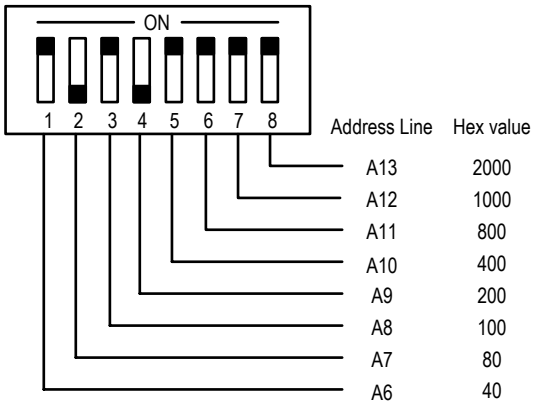
1. Incorrect loopback, faulty wiring
2. IRQ conflict with other hardware
3. Address conflict with other hardware

SWITCH DESCRIPTIONS

There are six dip switches on the FASTCOM™: ESCC-104, labeled SW1, SW2, SW3, SW4, SW5, and SW6 (See Board Layout Illustration for location). Switch 1 (labeled ADDRESS) is used to set the I/O address of the FASTCOM™: ESCC-104 board. Switches 2 and 3 (labeled IRQ SELECT) serve two functions: they select the IRQ level for the board and are used to enable/disable interrupt sharing. Switch 4 (DMA CHANNEL SELECT) selects the DMA channels to be used by each ESCC port. Switch 5 (MODE #1 CONFIG) selects either RS-422 or RS-485 mode, enables DMA and determines the direction of the TXCLK signal (if the transmit clock is received or transmitted). Switch 6 (MODE #2 CONFIG) controls the loopback function for RS-485 and the CTS disable feature.

SWITCH 1, ADDRESS

Switch 1 decodes the PC address lines as follows:



Address lines A6 through A13 are decoded by the setting of SW1 and set the base address of the FASTCOM™: ESCC-104. Address lines A0 - A5 are used on the board to select configuration and control registers on the 82532 chip.

The above diagram illustrates a base address of 280 Hex (factory default). Note that when a switch is ON it represents a "0" in the corresponding bit position (not a "1" as you might expect). Also, a switch that is OFF represents a "1" in the corresponding bit position. (If you would like to know why this is reversed, read the technical data for the address decoder chip, a 74LS682.)

So, the SW1 diagram can be decoded as follows:

A13	A12	A11	A10	A9	A8	A7	A6
0	0	0	0	1	0	1	0

You can determine the I/O address of the board by adding the Hex values for each address line that is set to a "1". In the illustration, only address lines A9 and A7 are set to "1". So, add the Hex value of A9 (200H) and A7 (80H), and the result is the I/O base address (200H + 80H = 280H). We have provided a comprehensive guide to setting the address switch in Appendix A.

Please note that not all of the I/O address space in a PC is available for your use. We have selected 280H as a default because it does not conflict with devices normally installed in a PC. However, if you wish to select another address, select an address that does not conflict with devices installed in your PC. Keep in mind that the FASTCOM™: ESCC-104 requires 34 contiguous bytes of address space.

If you want to install more than one FASTCOM™: ESCC-104 board in your computer, be sure to set each to a unique I/O address. We recommend the following addresses for a multi-board system:

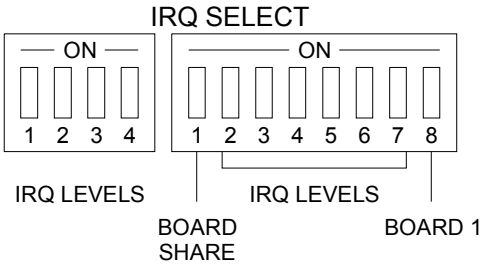
FASTCOM™: ESCC-104 BOARD 1: 280H
FASTCOM™: ESCC-104 BOARD 2: 2C0H
FASTCOM™: ESCC-104 BOARD 3: 300H

Remember that a single IRQ level can be shared between multiple FASTCOM™: ESCC-104 boards in a PC, and that DMA channels cannot be shared.

SWITCHES 2 AND 3, INTERRUPTS

Switches 2 and 3 serve two functions: they select the IRQ level for the FASTCOM™: ESCC-104 and enable/disable interrupt sharing.

The following illustrates the IRQ select switch on the FASTCOM™: ESCC-104:



Select only 1 IRQ level at a time.

	SWITCH POSITION	IRQ	PC/AT/386 Assigned
SW3	2	9	UNUSED
	3	3	COM2
	4	4	COM1
	5	5	UNUSED (LPT2)
	6	6	FLOPPY
	7	7	LPT1
SW2	1	10	UNUSED
	2	11	UNUSED
	3	12	UNUSED
	4	15	UNUSED

You can use any IRQ that is not assigned to a device installed in your PC.

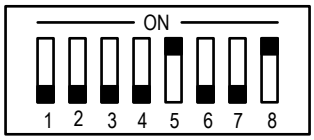
INTERRUPT SHARING

An important feature of the FASTCOM™: ESCC-104 is its ability to share one IRQ level with several FASTCOM™: ESCC-104 boards in the same computer. This is important because there are very few unassigned IRQs in the PC.

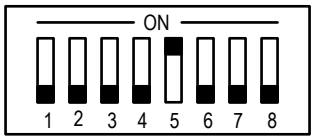
Switch 3 positions 1 and 8 control the interrupt sharing circuit on the FASTCOM™: ESCC-104. Position 1 Enables interrupt sharing in the OFF position and Disables sharing in the ON position. Position 8 is called the "Board 1" switch. In the interrupt sharing mode, this switch must be ON for the first FASTCOM™: ESCC-104 board in your system, and OFF on all other FASTCOM™: ESCC-104 boards.

Switch 3 Position 1	8	
ON	OFF	Disables IRQ sharing
OFF	ON	Enables IRQ sharing, first board
OFF	OFF	Enables IRQ sharing, second board

For example, let's assume that you have two FASTCOM™: ESCC-104 boards in your PC and want to share IRQ 5. Set Switch 3 as follows for the first board:



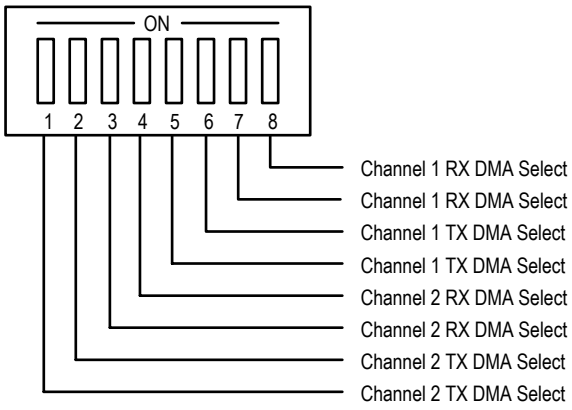
For the second board, set Switch 3 as follows:



Any additional FASTCOM™: ESCC-104 boards that share IRQ5 would be set the same as the second board.

SWITCH 4, DMA CHANNEL SELECT

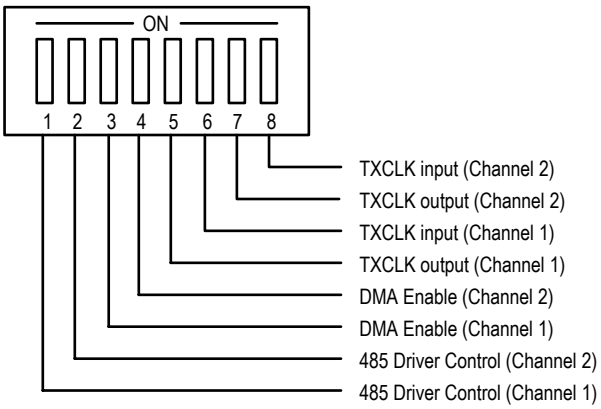
The DMA Select switch sets the DMA receive and transmit channels. **Do not set the transmit and receive channels to be the same.** It is not possible to set and use the transmit and receive channels as the same DMA channel. If the same DMA channel is selected for more than one function (TX, RX, Channel 1, Channel 2), that DMA channel will be disabled.



	DMA Channel	1	3	5	7
Transmit 1	Position 5	On	On	Off	Off
	Position 6	On	Off	On	Off
Receive 1	Position 7	On	On	Off	Off
	Position 8	On	Off	On	Off

Transmit 2	Position 1	On	On	Off	Off
	Position 2	On	Off	On	Off
Receive 2	Position 3	On	On	Off	Off
	Position 4	On	Off	On	Off

SWITCH 5, MODE #1 SWITCH



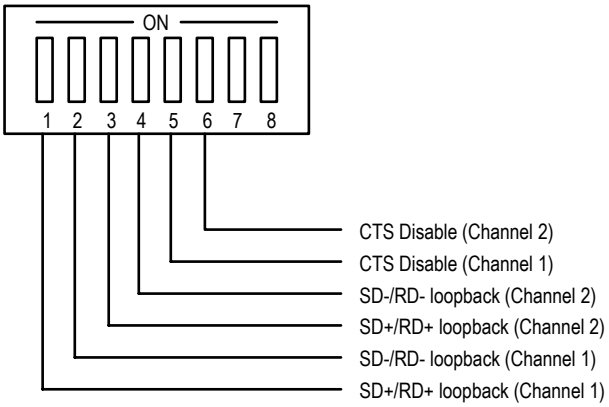
The Mode #1 switch serves three functions: it selects either RS-422 or RS-485 driver mode, enables or disables DMA, and selects the direction of the transmit clock (input or output).

The 485 driver control switch controls the transmitter enable for the SD+/- signals. In the on position, the RTS output controls the enable (RS-485 mode); in the off position the SD+/- driver is always on (RS-422 mode).

The DMA enable switch controls the connection to the ISA DMA channels. In the on position, DMA is enabled for that channel (both RX and TX DMA channel selects will be active for that channel). In the off position, DMA is disabled.

The TXCLK input/output switches connect the TXCLK pin to either the TT+/- signal lines (to output the transmit timing), or the ST+/- signal lines (to input the transmit timing). If the input selection is made with position 6 or 8 on, **do not** configure the 82532 TXCLK pin as an output!

SWITCH 6, MODE #2 SWITCH



The Mode #2 switch controls the RS-485 loopback feature, connecting the SD+/- signals to the RD+/- signals internally so you don't have to make the connection on your cable. It also controls the CTS disable feature and the output of the clock in RS-485 mode. Since the 82532 will only transmit when its CTS pin is active, this switch allows the 82532 to see the CTS input as active all the time, regardless of the signal at the connector.

The loopback switch positions are provided for RS-485 mode and should be used in pairs. Positions 1 and 2 should either be on, creating a loopback for RS-485 mode, or off, for RS-422 mode. Likewise, positions 3 and 4 should either both be on, or both off.

The CTS disable switch in the on position is used to force CTS into the active state to allow the 82532 to transmit data if the CTS signal line is not used. In the off position, the CTS signal from the connector is connected to the 82532, and the 82532 will not transmit unless the external CTS signal is active!

The 485 clock control switch selects 422 or 485 clock output mode. In the off position (default), the TXCLK is selected as an output for RS-422 (TT+/- lines are always clocking). If a gated clock is required, the on position will connect the RTS line of the channel to the TT+/- driver enable and it will become RS-485 (similar to the SD+/- lines on Switch 5). Selecting 485 clock output mode will disable the DTR line at the connector.

PROGRAMMING

Refer to the ESCC-104 Tools on the enclosed FASTCOM CD for example programs, product updates, and software for testing your installation. Refer to the Siemens SAB 82532 User's Manual for register information.

NOTES

Do not select the same channel for both DMA receive and DMA transmit.

Always set the 82532 port configuration register (PCR) to E0H.
Always set the 82532 interrupt port configuration (IPC) to 03H.
Always set the 82532 CCR1 ODS bit to 1.

Revision 3.2A of the Siemens 82532 utilizes both standard and enhanced modes of the Baud Rate Generator Register (BGR). In standard mode, the following formula is used to calculate the divisor for baud rate generation:

$$k = (N+1) * 2$$

The following hexadecimal values of N are equivalent to N equaling zero:

0x000	0x100	0x200	0x300
0x040	0x140	0x240	0x340
0x080	0x180	0x280	0x380
0x0C0	0x1C0	0x2C0	0x3C0

This is a known bug of the 82532.

RS-422 / RS-485

Most engineers have worked with RS-232 devices at least once in their career. If you have never worked with RS-422 or RS-485 devices, you will be pleased to know that working with the FASTCOM™: ESCC-104 is not much different from working with an RS-232 device.

The RS-422 standard was developed to correct some of the deficiencies of RS-232. In commercial and industrial applications, RS-232 has some significant problems. First, the cable length between RS-232 devices must be short (usually less than 50 feet at 9600 Baud). Second, many RS-232 errors are the result of cables picking up normal industrial electrical noises such as fluorescent lights, motors, transformers, and other EMF sources. Third, RS-232 data rates are functionally limited to 19.2K Baud. On the other hand, the newer RS-422 standard makes cable lengths up to 5000 feet possible and is highly immune to most industrial noises. Data rates are also improved -- the FASTCOM™: ESCC-104 features data rates up to 10 Mega Baud. These improvements were made possible by differentially driving and receiving the data as opposed to the single ended method employed by the RS-232 standard. With the RS-422 standard, the transmit signal (TX in RS-232) is a differential signal consisting of SD+ and SD-; the receive signal (RX in RS-232) consists of RD+ and RD-.

Another draw back of RS-232 is that more than two devices cannot share a single cable. This is also true of RS-422, and that's why the RS-485 standard was developed. RS-485 offers all of the benefits of RS-422 and also allows multiple units (up to 32) to share the same twisted pair. RS-485 is often referred to as a "multi-drop" or "two-wire, half duplex" network because the drivers (transmitters) and receivers share the same two lines. In fact, up to 32 stations can share the same "twisted pair". In order for an RS-485 system to work, only one driver (transmitter) can occupy the network at a time. This means that each station on the network must control the enabling/disabling of their drivers in order to avoid network conflicts. If two drivers engage the network at the same time, data from both will be corrupted. In RS-485 mode, the receivers are always enabled.

For a more detailed description of RS-422 and RS-485, we recommend the following references:

LINEAR AND INTERFACE CIRCUITS APPLICATIONS, Volume 2: Line Circuits, Display Drivers. By D.E. Pippenger and E. J. Tobaben. Published 1985 by Texas Instruments. ISBN-0-89512-185-9

Note: This book may be difficult to find in a bookstore. The best place to get it is directly from Texas Instruments or from one their component dealers. Publication # SLYA002.

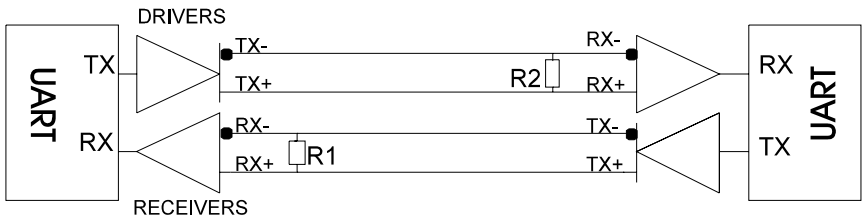
"Driver/Receiver Family Extends Data-Link Performance", ELECTRONIC PRODUCTS, January 15, 1985. By Dale Pippenger and Joe Miller

TERMINATION RESISTANCE

In both the RS-422 and the RS-485 mode, the receiver end of the cable between two stations must be terminated with a resistor equal to the characteristic impedance of the wire. This is to prevent signal reflections in the wire and to improve noise rejection. However, **you do not need to add a terminator resistor to your cables when you use the FASTCOM: ESCC-104. The termination resistance is built in.** We have installed a terminator resistor for each receiver: between each RD+ and RD- and between CTS+ and CTS- for each channel.

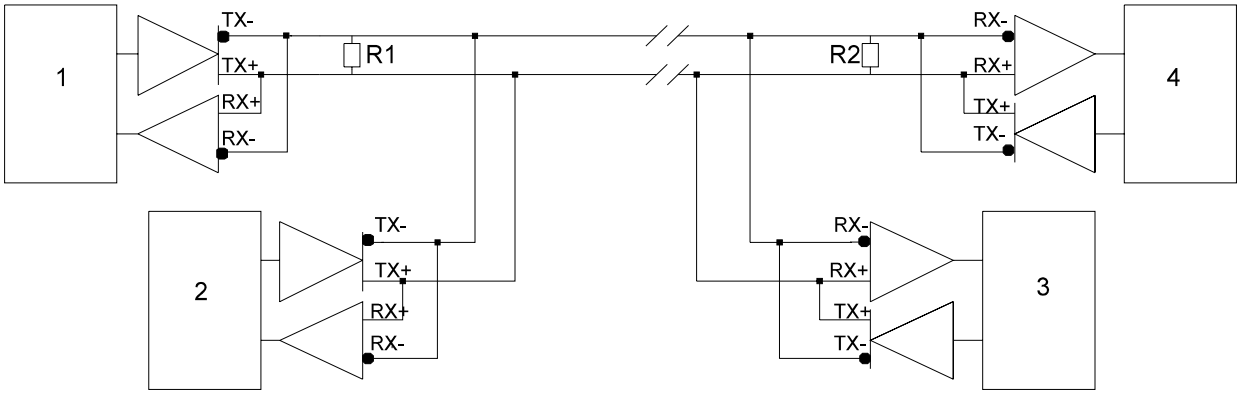
If you are using the FASTCOM™: ESCC-104 in a multi-drop network, the termination resistor should be removed from all units except the first and last (see the RS-485 illustration below). Call for technical support if you need to modify the resistor. You may also order the FASTCOM™: ESCC-104 without the termination resistor installed (it is easier to add the resistor than to remove it). Observe the resistors in the following drawings and remember that they are built into the FASTCOM™: ESCC-104:

Typical RS-422



R1 & R2 -- Line Termination (100 Ohms)

Typical RS-485



PROGRAMMABLE CLOCK GENERATOR (Cypress ICD2053B)

The FASTCOM™: ESCC-104 features the ICD2053B Programmable Clock Generator, which offers a fully user-programmable phase-locked loop in a single 8-pin package. The output may be changed "on the fly" to any desired frequency value between 391 kHz and 90 MHz (the FASTCOM™: ESCC-104 maximum is 33 MHz). The ability to dynamically change the output frequency adds a whole new degree of freedom for the designer.

Programming the ICD2053B is simple, requiring only a "bitcalc" program. This program is provided on the Fastcom CD.

The data sheet for the ICD2053B also explains the operation of this program. Open data sheet

FEATURES

- Clock outputs ranging from 391 kHz to 90 MHz (the FASTCOM™: ESCC-104 maximum is 33 MHz)
- Phase-Locked Loop oscillator input derived from external reference clock (18.432 MHz on the FASTCOM™: ESCC-104)
- Three-State output control disables output for test purposes
- Sophisticated internal loop filter requires no external components or manufacturing tweaks as commonly required with external filters
- Low power consumption makes device ideal for power- and space-critical applications
- Programmable using the FASTCOM™: ESCC-104 PVR register, bits 1 and 2 (see page 34)
- 5V operation
- High-speed CMOS technology

DETERMINING AND SELECTING BAUD RATES

Selecting the bit rate can either be very easy or quite complicated, depending on a number of factors. The best place to start is to determine the big picture (broad perspective) and narrow down the options using the various constraints that the hardware imposes.

There are four basic things that make up what the actual bitrate will be. They are:

1. Register settings of the 82532 chip. These include:
   A. Operating mode (HDLC, Bisync, Async)
      1. If async is used is it truly async (oversampled BCR=1), or
      2. isosynchronous (async format with no oversampling).
   B. Clock mode (internal or external clocks)
      1. If internal clocks, does it use clock recovery (DPLL)
      2. If BGR is used is the BDF bit 1 or 0.
2. The setting of the clock generator (ICD2053B) that feeds the OSC input to the 82532 (only a factor if an internal clock mode is used, i.e., BGR or DPLL is involved).
3. The physical switch settings of the Fastcom: ESCC-104.
   A. The switch determines the routing of the TXCLK pin as either an input or output.
   B. The switch also determines if the OSC is fed the output of the clock generator or just the output of the onboard clock (an option on the ESCC-PCI and the HSCX; the ESCC-104 does not have this option).
4. The revision of the ESCC 82532 chip (silicon)
   A. The rev 3.2 silicon incorporates an enhanced baud rate mode.

We will start with the simplest case. If you are using the Asynchronous data mode, then the most likely clock mode that you should use is 7b. It is possible to use the other clock modes; however, mode 7b is the most straightforward to work with. The bitrate will be determined by the output of the baud rate generator. The baud rate generator is clocked by the OSC input (which is set by the ICD2053B clock generator). So you have:

If you are not using oversampling (BCR = 0) the formula is:

bitrate = (input clock / BGR)

If you are using oversampling (BCR = 1 (the normal case for async)) the formula is:

bitrate = (input clock / BGR)/16

If BDF = 0	then BGR = 1
If BDF = 1	then BGR = (N+1)*2
If BDF = 1 and EBRG = 1	then BGR = (n+1)*2^m

The BCR bit is in CCR1 (bit 3)
The BDF bit is in CCR2 (bit 5)
The EGRG bit is in CCR4 (bit 6)
N = (CCR2&0xC0)<<2) + BGR, or if you prefer

	Most significant bit..... Least significant bit
n =	BGR bit 5, BGR bit 4, BGR bit 3, BGR bit 2, BGR bit 1, BGR bit 0
m =	CCR2 8, CCR2 7, BGR 7, BGR 6
N =	CCR2 8, CCR2 7, BGR 7, BGR 6, BGR 5, BGR 4, BGR 3, BGR 2, BGR 1, BGR 0

Some things to keep in mind:

The range on the ICD2053B output is 391kHz to 90MHz.
The usable range if the master clock enable bit is clear (not using master clock) is 391kHz to 33Mhz. If the master clock is set, then the range is from 391kHz to 10MHz.
Using master clock mode also places a restriction on the ratio of receive/transmit clock to the master clock frequency (as per note 2 on page 84 of the 82532 data sheet).
Fmaster/Ftransmit >= 2.5

Freceive/Fmaster <3 or 1.5 (if CCR3 bit 4 (RADD) is set and an address recognition mode is used in HDLC).

To use Master clock mode or not to use Master clock mode; that is the question.

The 82532 operating in standard (non-master clock) mode uses the transmit clock source (refer to table 5 page 84 of the data sheet) to run the internal timing of the chip. If your transmit clock source is running very slow or it is not running continuously (if external clock is supplied), then it is a good idea to switch to master clock mode. Each command issued to the 82532 (any write to the CMDR register) can take up to 2.5 clocks to complete. If the clock is very slow or stops from time to time, this can be a significant amount of time and allows for the possibility of a command being lost (written but not executed, because a previous command is not complete). If your baud rate is slow (<1MHz) or you are using a gated external clock, you should use master clock mode to allow the PC interface to the 82532 to continue to execute quickly. If this mode is used, then the OSC input must be less than 10MHz (the speed rating of the internals of the 82532). If you have a 82532 rev 3.x (silicon), then you can cause the master clock to be OSC/4 by setting CCR4 bit 7 (MCK4), thus allowing the 10MHz restriction to be lifted.

If you are using a clock mode that uses external clocks, you should respect the restriction on the ratio of receive to transmit clock frequency given in note 2 of table 5 (Freceive/Ftransmit <3 or 1.5).

If you are running in not extended baud rate mode, do not set BGR bits 5-0 to all = 0, or the chip will assume that all of the BGR bits are 0. This is a glitch in the 82532 rev 3.2 and makes the following values for N identical:

0x000,0x040,0x080,0x0C0
0x100,0x140,0x180,0x1C0
0x200,0x240,0x280,0x2C0
0x300,0x340,0x380,0x3C0

Setting the baud rate generator to any of these values will produce the same affect as setting it to 0x000.

If you use the Enhanced baud rate generator and set m = 0, the clock output will be asymmetric (non 50/50 duty cycle).

To determine the parameters that need to be sent to the ICD2053B part, you will need to obtain the bitcalc3 program from Cypress. It can be found on the web at:

http://www.cypress.com/cypress/tech_sup/web_tech/sw/clocks.html

After you have downloaded and installed it select:

Devices->programmable products->ICD2053B->CLKOUT from the menu.

Type in 18.432 MHz for the reference frequency (for the ESCC-104 and ESCC-PCI, for the HSCX type in 16.000 MHz). Enter the desired frequency (keeping in mind all of the restrictions noted above), and press the calculate button. Write down the Hex programming word (stuffed) and count the number of bits in the binary word. These are the parameters that need to be passed to the IOCTL_ESCCDRV_SET_CLOCK function to program the ICD2053B to get the desired frequency output.

An important fact about the clock generator

There is only one ICD2053B part, and only 1 OSC input to the 82532 chip. The clock generator can be programmed from either channel (ESCC0 or ESCC1), but it programs the same part. The result is that while the baud rate generators are unique on a per channel basis, the OSC input is not (i.e., the baud rate generators are independent, but the clock that feeds them is the same). If you change the clock generator output you will change the input clock to both channels. The practical thing to note about this is that if you have multiple baud rates that must be generated on multiple channels, you should select the input clock (ICD 2053B output) such that all baud rates can be derived from one clock value. *Changing the clock generator output will affect the baud rates of both channels (ESCC0, ESCC1)!!!!*

If you are using HDLC or Bisync as a data format, there is not a BCR setting (no oversampling). However, if you select a clock mode that uses the DPLL as a source, it will effectively add a divide by 16 to your function.

Selecting the appropriate clock mode is a matter of identifying what clock signals are available external to the Fastcom card, and what clock signals are required by the external device. The simplest mode is using external clocks only (mode 0a); in this mode both the receive and transmit timing are taken from the connector (RT for receive, ST for transmit). The rest of the modes are a mix of external signals and internally generated clocks/clock recovery. The DPLL modes only operate up to 2 MHz. If the bitrate is above that, you should use a non-DPLL mode.

The bitrate functions are similar to the async case:

If you are not using a clock mode that uses the DPLL, the formula is:
bitrate = (input clock / BGR)

If you are using the DPLL, the formula is:
bitrate = (input clock / BGR)/16

The input clock will depend on the clock mode. It is usually either the OSC input or the RXCLK (RT+/-) input (see table 5 page 84 of the 82532 data sheet).

If BDF = 1	BGR = 1
If BDF = 0	BGR = (N+1)*2
If BDF = 0 and EBRG = 1	BGR = (n+1)*2^m (V 3.x of the 82532 silicon only)

The BDF bit is in CCR2 (bit 5)
N = (CCR2&0xC0)<<2) + BGR, or if you prefer

	Most significant bit	Least significant bit
m =	CCR2 bit 8,CCR2 bit 7, BGR bit 7, BGR bit 6	
n =	BGR 5, BGR 4, BGR 3, BGR 2, BGR 1, BGR 0	
N =	CCR2 8,CCR2 7, BGR 7, BGR 6, BGR 5, BGR 4, BGR 3, BGR 2, BGR 1, BGR 0	

If you are using the DPLL, you should try to set its input clock to be as close to the actual bit frequency as possible. This will allow for optimal clock recovery. Also, clock recovery relies on edges in the data stream; if you transmit long segments of 0s or 1s using an encoding method that produces no edges, the results will be non-optimal. The ideal encoding for clock recovery is Manchester or a non '1' idle pattern (i.e., constant flag sequences on idle if HDLC is used, etc.).

Let's start with something easy

Let's say that you want to set up an ESCC channel to run in HDLC mode at 19200 bps, that the device in question supplies a clock with its data (receive clock), and that we need to generate (transmit) a clock that matches our transmitted data. To achieve this we should set the 82532 to clock mode 0b. Set the mode switch position 5 (or 7) to on, enabling the txclk output driver (selecting txclk as an output on TT+/-). The baudrate function bitrate = input clock / (N+1)*2 will be used. If there are no other constraints other than operating one channel at 19200 bps, then we can select both the input clock and N arbitrarily, so long as we do not violate any of the notes.

So, by selecting a value for input clock that is less than 10 MHz (since the bit rate is slow we will want to use master clock mode, which will require a 10 MHz or less clock), we can then calculate the value needed for N to get a 19200 bps output. I will pick 7.372800 MHz for the input clock. To get the programming word for this frequency, I run bitcalc and enter the reference frequency (18.432 MHz) and the desired frequency, and get the following values:
clkbits = 0x0E9920 (the stuffed (hex) programming word), and
numbits = 23 (the number of bits in the stuffed programming word (counted from the stuffed binary word)).

Calling the IOCTL_ESCCDRV_SET_CLOCK ioctl function with these parameters will get us an input clock of 7.3728 MHz (referred to as OSC in table 5).

Then solving for N we get:

19200 = 7.3728E6/((N+1)*2)
N = 191 = 0x0BF

Checking the notes to make sure we did not violate anything:

Fm/Fx = 7.3728E6/19200 = 384 > 2.5 (we are OK on this one)
Fr/Fm = rxclk input / 7.3728E6 < 3 (assuming a 19200 clock input) 19200/7.3728E6 = .0026 < 3 (we are OK on this one)
(0x0BF & 0x3f) <= 0 (checking the value of n to make sure it isn't forced to zero due to the glitch in the 82532)

Important Register Settings

MODE = 0x88

This sets the 82532 in transparent HDLC mode 0. This will use a frame structure as:
0x7E | data | CRC | CRC | 0x7E
No address recognition is used
The timer is in external mode
RTS is handled by the 82532 (active while transmitting)
Timer resolution is 32768 clocks

CCR0 = 0xC0

This sets the 82532 in power up mode
Master clock mode is enabled
NRZ is the encoding type
HDLC mode is selected

CCR1 = 0x10

This selects clock mode 0(b)
The tx pin is using a push-pull output (required)
Time fill is all '1's (idle pattern = 0xff)

CCR2 = 0x38

This selects the BGR = (N+1)*2 divisor
Selects txclk to be an output
Selects clock mode 0b (the B part)
Enables the CRC-CCITT polynomial

CCR3 = 0x00

No preamble output
CRC reset level = 0xffff
CRC is in use (both transmit and receive), not including CRC in received Data (not returned to the user)
Not using extended window for DPLL

CCR4 = 0x00

Not using master clock/4
Not using enhanced baud rate generator
FIFO threshold is 32 bytes (mandatory for NT driver in HDLC mode)

BGR = 0xBF

This sets the output clock rate to 19200 (given that the input clock was previously set to 7.3728 MHz, and the above registers are set as shown)

CCR0, CCR1, CCR4 and BGR are the most critical registers that effect the bitrate; the rest are shown for

completeness, and, depending on the system, you can easily change some parameters without affecting the bitrate (i.e., line encoding, address recognition, crc type, etc.).

And now for something a bit more difficult:

Let's say that you want to run one channel asynchronously at 38400 bps, and the second channel synchronously using HDLC at 2 Mbps. How would you go about it?

Start with the fastest bit rate and determine if there is an external clock that is received with that data or if the clock must be recovered (DPLL mode). Let's say that you want to recover the clock from the data (there are no clock lines in the system), and that the data is Manchester encoded. To get a 2 Mbps clock rate using a DPLL we will need to use the bitrate = (input clock/1)/16 function. This will require a 32 MHz input clock. To get this input clock using bitcalc as before:

clkbits = 0x5D0C60
numbits = 23

using the following register settings:

MODE = 0x88
CCR0 = 0x98
CCR1 = 0x16
CCR2 = 0x18
CCR3 = 0x00
CCR4 = 0x00

The receive source will be recovered from the data stream. The transmit source will be the BGR/16 output.

Now for the async channel. We are locked into the 32 MHz input clock, so we will try to find a value for N that gets our desired 38400 bps:

38400 = (32E6/16)/((N+1)*2)
N = 25.04 (we cannot attain non-integer values for N).

If we use N = 25 we would get:

bitrate = (32E6/16)/(25+1)*2 = 38462 bps

If we use N = 26 we would get:

bitrate = (32E6/16)/(26+1)*2 = 37037 bps

Using the closest value and setting the registers to

MODE = 0x08,
CCR0 = 0xC3,
CCR1 = 0x1F,
CCR2 = 0x38,
CCR3 = 0x00,
CCR4 = 0x80, and
BGR = 0x19

will yield an asynchronous data format (with 16X oversampling) at about 38400 bps.

If later you decide that you need to get 115200 bps on the async channel you will find:

115200 = (32E6/16)/((N+1)*2)
N = 7.68

Using N = 7

bitrate = (32E6/16)/((7+1)*2) = 125000 bps

Using N = 8

bitrate = (32E6/16)/((8+1)*2) = 111111 bps

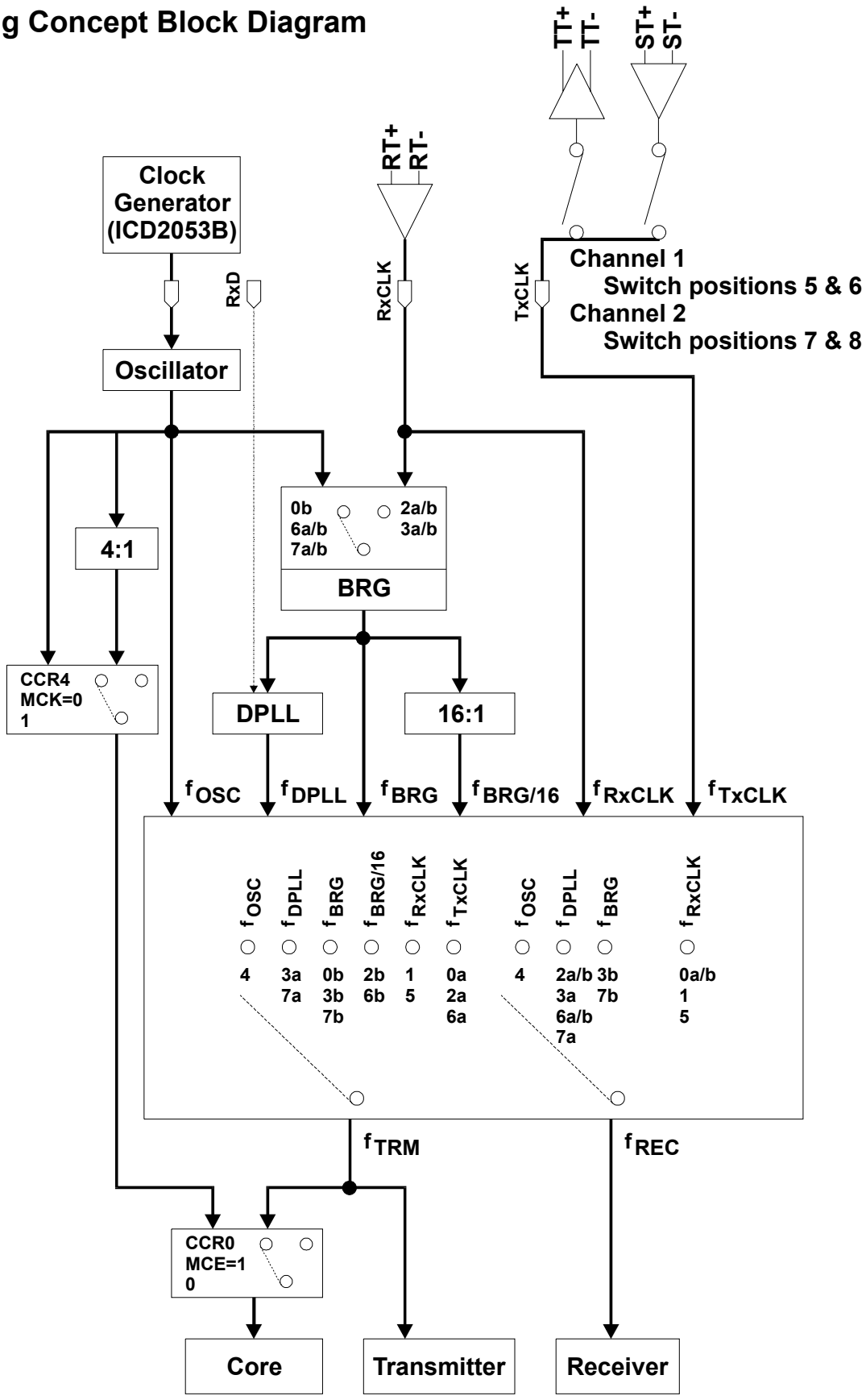
The ideal situation would be to adjust the 32 MHz clock such that the deviation between the desired and actual

rates is spread between both channels (with the DPLL recovering the clock, the actual clock that feeds it is not as critical as a clock mode that uses the clock directly).

Clock modes 0b, 3b, 4, and 7b are more sensitive to the selected rate in synchronous modes, as there is no oversampling. The rate you select is the rate you will get, whereas oversampling modes (using the DPLL or ASYNC BCR) are more tolerant to differences between the rate you set and the rate you want.

On the following page is a block diagram representation of the clocking concept.

Clocking Concept Block Diagram



TECHNICAL SUPPORT

All products manufactured by Commtech are warranted against defective materials and workmanship for the lifetime of the product. This warranty is available only to the original purchaser. Any product found to be defective will, at the option of Commtech, be repaired or replaced with no charge for labor or parts not excluded by the warranty. This warranty does not apply to any products that have been subjected to misuse, abuse, or accident or as a result of service or modification by anyone other than Commtech. In no case shall Commtech liability exceed the original product purchase price.

If any Commtech product is damaged such that it cannot be repaired, you can return it to Commtech for replacement under our *Non-Repairable Replacement* policy, regardless of the cause of damage. Commtech will replace the unit at 60% of the then-current list price.

Commtech provides extensive technical support and application suggestions. Most of the problems that occur with the FASTCOM™: ESCC-104 can be corrected by double-checking the switch positions, your cables and your program. We recommend that you build the loop back plug that is described in the Programming section of this manual. With that plug, you can quickly isolate the problem to the board, cables, or software.

If you still have unresolved questions, use the following procedure to get technical support:

1. Call our Technical Support Staff at (316) 636-1131. They are on duty from 9:00 AM to 5:00 PM Central Time.

   If you purchased your board from Kontron America, please call their 7-24 technical support line at 1-800-
   .

2. Ask for technical support for the FASTCOM: ESCC-104. Be ready to describe the problem, your computer system, your application, and your software.

3. If necessary, our staff will give you an RMA number (Return Material Authorization). Use this number on the mailing label and in all references to your board. Put the board back in its static bag and in its box. Ship the board back to us as directed.

4. If you prefer, you may FAX a description of the problem to us at (010) 6858-5001, or we can be reached on

APPENDIX A

FASTCOM: ESCC-104

REGISTER MAP

AND

I/O ADDRESS SETTINGS

FASTCOM: ESCC-104 REGISTER MAP

The following chart illustrates the register map of the FASTCOM™: ESCC-104, using the factory default address of 280H. Refer to the Siemens SAB 82532 User's Manual for additional information on the registers.

	HDLC/SDLC	ASYNC	BISYNC
Address	Read / Write	Read / Write	Read / Write
280H	STAR / CMDR	STAR / CMDR	STAR / CMDR
281H	RSTA / ----	---- / ----	---- / ----
282H	MODE / MODE	MODE / MODE	MODE / MODE
283H	TIMR / TIMR	TIMR / TIMR	TIMR / TIMR
284H	XAD1 / XAD1	---- / ----	SYNL / SYNL
285H	XAD2 / XAD2	---- / ----	SYNH / SYNH
286H	---- / RAH1	TCR / TCR	TCR
287H	---- / RAH2O	DAFO / DAFO	DAFO
288H	RAL1 / RAL1	RFC / RFC	RFC
289H	RHCR / RAL2	---- / ----	---- / ----
28AH	RBCL / XBCL	RBCL / XBCL	RBCL / XBCL
28BH	RBCH / XBCHH	RBCH / XBC	RBCH / XBCH
28CH	CCR0 / CCR0	CCR0 / CCR0	CCR/ / CCR0
28DH	CCR1 / CCR1	CCR1 / CCR1	CCR1 / CCR1
28EH	CCR2 / CCR2	CCR2 / CCR2	CCR2 / CCR2
28FH	CCR3 / CCR3	---- / ----	CCR3 / CCR3
290H	---- / TSAX	---- / ----	---- / ----
291H	---- / TSAR	---- / ----	---- / ----
292H	---- / XCCR	---- / ----	---- / ----
293H	---- / RCCR	---- / ----	---- / ----
294H	VSTR / BGR	VSTR / BGR	VSTR / BGR
295H	---- / RLCR	---- / ----	---- / ----
296H	PRE / PRE	---- / ----	PRE / PRE
297H	---- / ----	---- / ----	---- / ----
298H	GIS / IVA	GIS / IVA	GIS / IVA
299H	IPC / IPC	IPC / IPC	IPC / IPC
29AH	ISR0 / IMR0	ISR0 / IMR0	ISR0 / IMR0
29BH	ISR1 / IMR1	ISR1 / IMR1	ISR1 / IMR1
29CH	PVR / PVR	PVR / PVR	PVR / PVR
29DH	PIS / PIM	PIS / PIM	PIS / PIM
29EH	PCR / PCR	PCR / PCR	PCR / PCR
29FH	---- / ---	---- / ----	---- / ----
2A0H	FIFO / FIFO	FIFO / FIFO	FIFO / FIFO

PVR Register

The 82532 has an 8-bit I/O port (PVR) that has the following functions on the FASTCOM™: ESCC-104.

Bit:
0 - Channel select, 0 = channel 1, 1 = channel 2
1 - ICD2053B clock
2 - ICD2053B data
3 - DTR channel 1 (output)
4 - DTR channel 2 (output)
5 - DSR channel 1 (input)
6 - DSR channel 2 (input)
7 - TC from ISA bus (from PC's DMA controller) (input)

I/O ADDRESS SETTINGS

The FASTCOM™: ESCC-104 requires 34 contiguous bytes of address space. You may use any I/O address that is not used by a device installed in your system.

FASTCOM™: ESCC-104 ADDRESS SWITCH SETTINGS

ADDRESS SWITCH 1 POSITION ("1" = ON; "0" = OFF)

Hex (Decimal)	1 2 3 4 5 6 7 8
0h (0)	1 1 1 1 1 1 1 1
40h (64)	0 1 1 1 1 1 1 1
80h (128)	1 0 1 1 1 1 1 1
c0h (192)	0 0 1 1 1 1 1 1
100h (256)	1 1 0 1 1 1 1 1
140h (320)	0 1 0 1 1 1 1 1
180h (384)	1 0 0 1 1 1 1 1
1c0h (448)	0 0 0 1 1 1 1 1
200h (512)	1 1 1 0 1 1 1 1
240h (576)	0 1 1 0 1 1 1 1
280h (640)	1 0 1 0 1 1 1 1
2c0h (704)	0 0 1 0 1 1 1 1
300h (768)	1 1 0 0 1 1 1 1
340h (832)	0 1 0 0 1 1 1 1
380h (896)	1 0 0 0 1 1 1 1
3c0h (960)	0 0 0 0 1 1 1 1
400h (1024)	1 1 1 1 0 1 1 1
440h (1088)	0 1 1 1 0 1 1 1
480h (1152)	1 0 1 1 0 1 1 1
4c0h (1216)	0 0 1 1 0 1 1 1
500h (1280)	1 1 0 1 0 1 1 1
540h (1344)	0 1 0 1 0 1 1 1
580h (1408)	1 0 0 1 0 1 1 1
5c0h (1472)	0 0 0 1 0 1 1 1
600h (1536)	1 1 1 0 0 1 1 1
640h (1600)	0 1 1 0 0 1 1 1
680h (1664)	1 0 1 0 0 1 1 1
6c0h (1728)	0 0 1 0 0 1 1 1
700h (1792)	1 1 0 0 0 1 1 1
740h (1856)	0 1 0 0 0 1 1 1
780h (1920)	1 0 0 0 0 1 1 1
7c0h (1984)	0 0 0 0 0 1 1 1

Hex (Decimal)	1 2 3 4 5 6 7 8
800h (2048)	1 1 1 1 1 0 1 1
840h (2112)	0 1 1 1 1 0 1 1
880h (2176)	1 0 1 1 1 0 1 1
8c0h (2240)	0 0 1 1 1 0 1 1
900h (2304)	1 1 0 1 1 0 1 1
940h (2368)	0 1 0 1 1 0 1 1
980h (2432)	1 0 0 1 1 0 1 1
9c0h (2496)	0 0 0 1 1 0 1 1
a00h (2560)	1 1 1 0 1 0 1 1
a40h (2624)	0 1 1 0 1 0 1 1
a80h (2688)	1 0 1 0 1 0 1 1
ac0h (2752)	0 0 1 0 1 0 1 1
b00h (2816)	1 1 0 0 1 0 1 1
b40h (2880)	0 1 0 0 1 0 1 1
b80h (2944)	1 0 0 0 1 0 1 1
bc0h (3008)	0 0 0 0 1 0 1 1
c00h (3072)	1 1 1 1 0 0 1 1
c40h (3136)	0 1 1 1 0 0 1 1
c80h (3200)	1 0 1 1 0 0 1 1
cc0h (3264)	0 0 1 1 0 0 1 1
d00h (3328)	1 1 0 1 0 0 1 1
d40h (3392)	0 1 0 1 0 0 1 1
d80h (3456)	1 0 0 1 0 0 1 1
dc0h (3520)	0 0 0 1 0 0 1 1
e00h (3584)	1 1 1 0 0 0 1 1
e40h (3648)	0 1 1 0 0 0 1 1
e80h (3712)	1 0 1 0 0 0 1 1
ec0h (3776)	0 0 1 0 0 0 1 1
f00h (3840)	1 1 0 0 0 0 1 1
f40h (3904)	0 1 0 0 0 0 1 1
f80h (3968)	1 0 0 0 0 0 1 1
fc0h (4032)	0 0 0 0 0 0 1 1

Hex (Decimal)	1	2	3	4	5	6	7	8
1000h (4096)	1	1	1	1	1	1	0	1
1040h (4160)	0	1	1	1	1	1	0	1
1080h (4224)	1	0	1	1	1	1	0	1
10c0h (4288)	0	0	1	1	1	1	0	1
1100h (4352)	1	1	0	1	1	1	0	1
1140h (4416)	0	1	0	1	1	1	0	1
1180h (4480)	1	0	0	1	1	1	0	1
11c0h (4544)	0	0	0	1	1	1	0	1
1200h (4608)	1	1	1	0	1	1	0	1
1240h (4672)	0	1	1	0	1	1	0	1
1280h (4736)	1	0	1	0	1	1	0	1
12c0h (4800)	0	0	1	0	1	1	0	1
1300h (4864)	1	1	0	0	1	1	0	1
1340h (4928)	0	1	0	0	1	1	0	1
1380h (4992)	1	0	0	0	1	1	0	1
13c0h (5056)	0	0	0	0	1	1	0	1
1400h (5120)	1	1	1	1	0	1	0	1
1440h (5184)	0	1	1	1	0	1	0	1
1480h (5248)	1	0	1	1	0	1	0	1
14c0h (5312)	0	0	1	1	0	1	0	1
1500h (5376)	1	1	0	1	0	1	0	1
1540h (5440)	0	1	0	1	0	1	0	1
1580h (5504)	1	0	0	1	0	1	0	1
15c0h (5568)	0	0	0	1	0	1	0	1
1600h (5632)	1	1	1	0	0	1	0	1
1640h (5696)	0	1	1	0	0	1	0	1
1680h (5760)	1	0	1	0	0	1	0	1
16c0h (5824)	0	0	1	0	0	1	0	1
1700h (5888)	1	1	0	0	0	1	0	1
1740h (5952)	0	1	0	0	0	1	0	1
1780h (6016)	1	0	0	0	0	1	0	1
17c0h (6080)	0	0	0	0	0	1	0	1
1800h (6144)	1	1	1	1	1	0	0	1
1840h (6208)	0	1	1	1	1	0	0	1
1880h (6272)	1	0	1	1	1	0	0	1
18c0h (6336)	0	0	1	1	1	0	0	1
1900h (6400)	1	1	0	1	1	0	0	1
1940h (6464)	0	1	0	1	1	0	0	1
1980h (6528)	1	0	0	1	1	0	0	1
19c0h (6592)	0	0	0	1	1	0	0	1
1a00h (6656)	1	1	1	0	1	0	0	1
1a40h (6720)	0	1	1	0	1	0	0	1
1a80h (6784)	1	0	1	0	1	0	0	1
1ac0h (6848)	0	0	1	0	1	0	0	1
1b00h (6912)	1	1	0	0	1	0	0	1
1b40h (6976)	0	1	0	0	1	0	0	1
1b80h (7040)	1	0	0	0	1	0	0	1
1bc0h (7104)	0	0	0	0	1	0	0	1
1c00h (7168)	1	1	1	1	0	0	0	1
1c40h (7232)	0	1	1	1	0	0	0	1
1c80h (7296)	1	0	1	1	0	0	0	1
1cc0h (7360)	0	0	1	1	0	0	0	1
1d00h (7424)	1	1	0	1	0	0	0	1
1d40h (7488)	0	1	0	1	0	0	0	1
1d80h (7552)	1	0	0	1	0	0	0	1
1dc0h (7616)	0	0	0	1	0	0	0	1
1e00h (7680)	1	1	1	0	0	0	0	1
1e40h (7744)	0	1	1	0	0	0	0	1
1e80h (7808)	1	0	1	0	0	0	0	1
1ec0h (7872)	0	0	1	0	0	0	0	1
1f00h (7936)	1	1	0	0	0	0	0	1
1f40h (8000)	0	1	0	0	0	0	0	1
1f80h (8064)	1	0	0	0	0	0	0	1
1fc0h (8128)	0	0	0	0	0	0	0	1
2000h (8192)	1	1	1	1	1	1	1	0
2040h (8256)	0	1	1	1	1	1	1	0
2080h (8320)	1	0	1	1	1	1	1	0
20c0h (8384)	0	0	1	1	1	1	1	0

Hex (Decimal)	1	2	3	4	5	6	7	8
2100h (8448)	1	1	0	1	1	1	1	0
2140h (8512)	0	1	0	1	1	1	1	0
2180h (8576)	1	0	0	1	1	1	1	0
21c0h (8640)	0	0	0	1	1	1	1	0
2200h (8704)	1	1	1	0	1	1	1	0
2240h (8768)	0	1	1	0	1	1	1	0
2280h (8832)	1	0	1	0	1	1	1	0
22c0h (8896)	0	0	1	0	1	1	1	0
2300h (8960)	1	1	0	0	1	1	1	0
2340h (9024)	0	1	0	0	1	1	1	0
2380h (9088)	1	0	0	0	1	1	1	0
23c0h (9152)	0	0	0	0	1	1	1	0
2400h (9216)	1	1	1	1	0	1	1	0
2440h (9280)	0	1	1	1	0	1	1	0
2480h (9344)	1	0	1	1	0	1	1	0
24c0h (9408)	0	0	1	1	0	1	1	0
2500h (9472)	1	1	0	1	0	1	1	0
2540h (9536)	0	1	0	1	0	1	1	0
2580h (9600)	1	0	0	1	0	1	1	0
25c0h (9664)	0	0	0	1	0	1	1	0
2600h (9728)	1	1	1	0	0	1	1	0
2640h (9792)	0	1	1	0	0	1	1	0
2680h (9856)	1	0	1	0	0	1	1	0
26c0h (9920)	0	0	1	0	0	1	1	0
2700h (9984)	1	1	0	0	0	1	1	0
2740h (10048)	0	1	0	0	0	1	1	0
2780h (10112)	1	0	0	0	0	1	1	0
27c0h (10176)	0	0	0	0	0	1	1	0
2800h (10240)	1	1	1	1	1	0	1	0
2840h (10304)	0	1	1	1	1	0	1	0
2880h (10368)	1	0	1	1	1	0	1	0
28c0h (10432)	0	0	1	1	1	0	1	0
2900h (10496)	1	1	0	1	1	0	1	0
2940h (10560)	0	1	0	1	1	0	1	0

Hex (Decimal)	1	2	3	4	5	6	7	8
2980h (10624)	1	0	0	1	1	0	1	0
29c0h (10688)	0	0	0	1	1	0	1	0
2a00h (10752)	1	1	1	0	1	0	1	0
2a40h (10816)	0	1	1	0	1	0	1	0
2a80h (10880)	1	0	1	0	1	0	1	0
2ac0h (10944)	0	0	1	0	1	0	1	0
2b00h (11008)	1	1	0	0	1	0	1	0
2b40h (11072)	0	1	0	0	1	0	1	0
2b80h (11136)	1	0	0	0	1	0	1	0
2bc0h (11200)	0	0	0	0	1	0	1	0
2c00h (11264)	1	1	1	1	0	0	1	0
2c40h (11328)	0	1	1	1	0	0	1	0
2c80h (11392)	1	0	1	1	0	0	1	0
2cc0h (11456)	0	0	1	1	0	0	1	0
2d00h (11520)	1	1	0	1	0	0	1	0
2d40h (11584)	0	1	0	1	0	0	1	0
2d80h (11648)	1	0	0	1	0	0	1	0
2dc0h (11712)	0	0	0	1	0	0	1	0
2e00h (11776)	1	1	1	0	0	0	1	0
2e40h (11840)	0	1	1	0	0	0	1	0
2e80h (11904)	1	0	1	0	0	0	1	0
2ec0h (11968)	0	0	1	0	0	0	1	0
2f00h (12032)	1	1	0	0	0	0	1	0
2f40h (12096)	0	1	0	0	0	0	1	0
2f80h (12160)	1	0	0	0	0	0	1	0
2fc0h (12224)	0	0	0	0	0	0	1	0
3000h (12288)	1	1	1	1	1	1	0	0
3040h (12352)	0	1	1	1	1	1	0	0
3080h (12416)	1	0	1	1	1	1	0	0
30c0h (12480)	0	0	1	1	1	1	0	0
3100h (12544)	1	1	0	1	1	1	0	0
3140h (12608)	0	1	0	1	1	1	0	0
3180h (12672)	1	0	0	1	1	1	0	0
31c0h (12736)	0	0	0	1	1	1	0	0

Hex (Decimal)	1	2	3	4	5	6	7	8
3200h (12800)	1	1	1	0	1	1	0	0
3240h (12864)	0	1	1	0	1	1	0	0
3280h (12928)	1	0	1	0	1	1	0	0
32c0h (12992)	0	0	1	0	1	1	0	0
3300h (13056)	1	1	0	0	1	1	0	0
3340h (13120)	0	1	0	0	1	1	0	0
3380h (13184)	1	0	0	0	1	1	0	0
33c0h (13248)	0	0	0	0	1	1	0	0
3400h (13312)	1	1	1	1	0	1	0	0
3440h (13376)	0	1	1	1	0	1	0	0
3480h (13440)	1	0	1	1	0	1	0	0
34c0h (13504)	0	0	1	1	0	1	0	0
3500h (13568)	1	1	0	1	0	1	0	0
3540h (13632)	0	1	0	1	0	1	0	0
3580h (13696)	1	0	0	1	0	1	0	0
35c0h (13760)	0	0	0	1	0	1	0	0
3600h (13824)	1	1	1	0	0	1	0	0
3640h (13888)	0	1	1	0	0	1	0	0
3680h (13952)	1	0	1	0	0	1	0	0
36c0h (14016)	0	0	1	0	0	1	0	0
3700h (14080)	1	1	0	0	0	1	0	0
3740h (14144)	0	1	0	0	0	1	0	0
3780h (14208)	1	0	0	0	0	1	0	0
37c0h (14272)	0	0	0	0	0	1	0	0
3800h (14336)	1	1	1	1	1	0	0	0
3840h (14400)	0	1	1	1	1	0	0	0
3880h (14464)	1	0	1	1	1	0	0	0
38c0h (14528)	0	0	1	1	1	0	0	0

Hex (Decimal)	1	2	3	4	5	6	7	8
3900h (14592)	1	1	0	1	1	0	0	0
3940h (14656)	0	1	0	1	1	0	0	0
3980h (14720)	1	0	0	1	1	0	0	0
39c0h (14784)	0	0	0	1	1	0	0	0
3a00h (14848)	1	1	1	0	1	0	0	0
3a40h (14912)	0	1	1	0	1	0	0	0
3a80h (14976)	1	0	1	0	1	0	0	0
3ac0h (15040)	0	0	1	0	1	0	0	0
3b00h (15104)	1	1	0	0	1	0	0	0
3b40h (15168)	0	1	0	0	1	0	0	0
3b80h (15232)	1	0	0	0	1	0	0	0
3bc0h (15296)	0	0	0	0	1	0	0	0
3c00h (15360)	1	1	1	1	0	0	0	0
3c40h (15424)	0	1	1	1	0	0	0	0
3c80h (15488)	1	0	1	1	0	0	0	0
3cc0h (15552)	0	0	1	1	0	0	0	0
3d00h (15616)	1	1	0	1	0	0	0	0
3d40h (15680)	0	1	0	1	0	0	0	0
3d80h (15744)	1	0	0	1	0	0	0	0
3dc0h (15808)	0	0	0	1	0	0	0	0
3e00h (15872)	1	1	1	0	0	0	0	0
3e40h (15936)	0	1	1	0	0	0	0	0
3e80h (16000)	1	0	1	0	0	0	0	0
3ec0h (16064)	0	0	1	0	0	0	0	0
3f00h (16128)	1	1	0	0	0	0	0	0
3f40h (16192)	0	1	0	0	0	0	0	0
3f80h (16256)	1	0	0	0	0	0	0	0
3fc0h (16320)	0	0	0	0	0	0	0	0

APPENDIX B

SIEMENS

SAB 82532

DATA SHEET