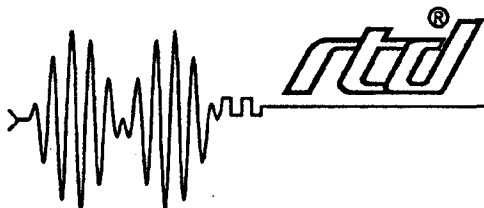


IGPS104

GPS utilityModule™

User's Manual



Real Time Devices, Inc.

"Accessing the Analog World"®

ISO9001 and AS9100 Certified

Publication No. IGPS104 9533

IGPS104
GPS utilityModule™
User's Manual



REAL TIME DEVICES, INC.

Post Office Box 906
State College, PA 16804
Phone: (814) 234-8087
FAX: (814) 234-5218
Faxback: (814) 235-1260

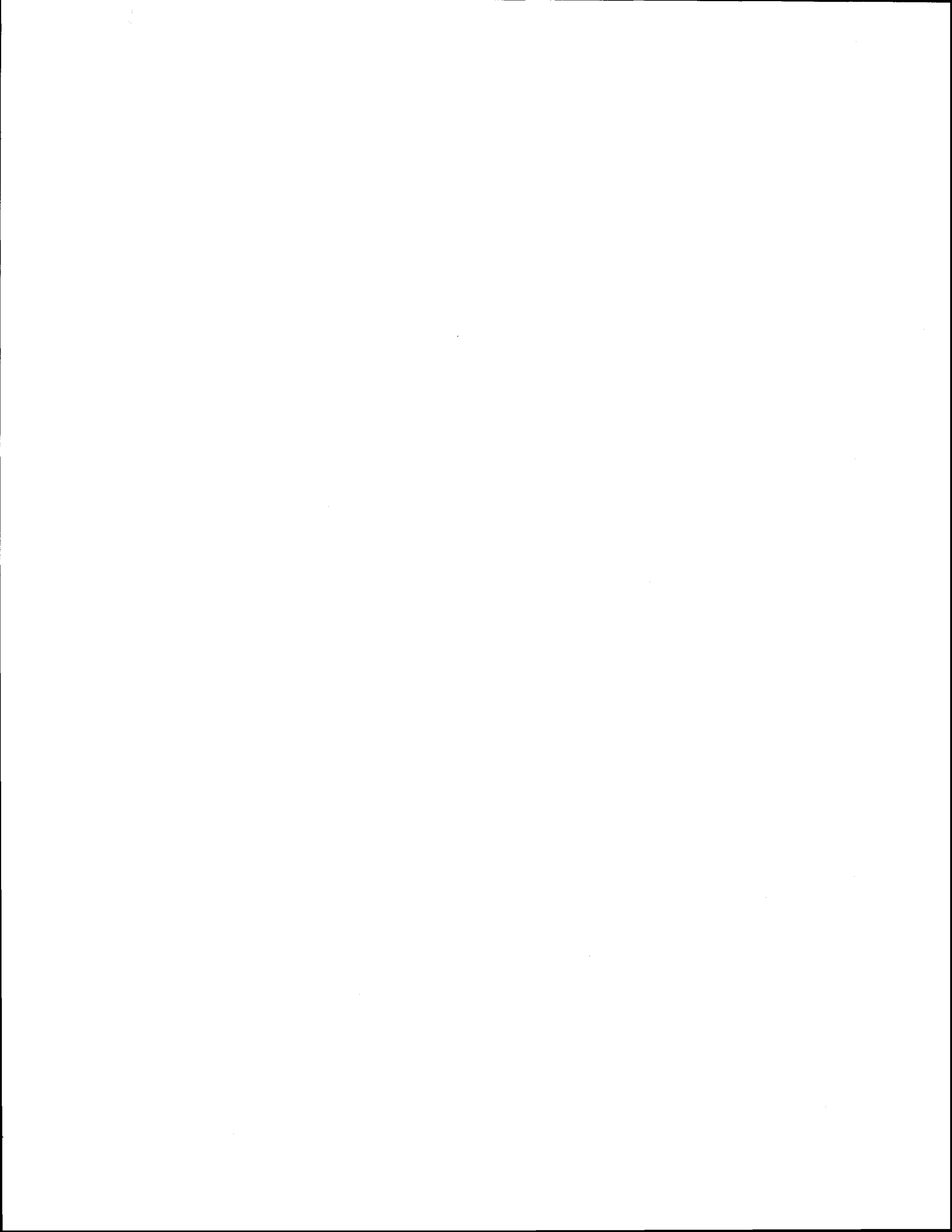
Published by
Real Time Devices, Inc.
P.O. Box 906
State College, PA 16804

Copyright 1995 by Real Time Devices, Inc.
All rights reserved
Printed in U.S.A.

PC/XT, PC/AT are registered trademarks of IBM Corporation.
MS-DOS is a registered trademark of Microsoft Corp.
PC/104 is a registered trademark of PC/104 Consortium.
The Real Time Devices Logo is a registered trademark of Real Time Devices.
The ETH Logo is a trademark of Eurotech S.R.L.
cpuModule and utilityModule are trademarks of Real Time Devices.
All other trademarks appearing in this document are the property of their respective owners.

Table of Contents

CHAPTER 1 INTRODUCTION.....	1-1
What is GPS and What Does It Do?.....	1-3
The IGPS104 Module.....	1-4
Module Specifications	1-5
CHAPTER 2 GETTING STARTED QUICKLY.....	2-1
Default Configuration	2-3
Basic Connector Locations.....	2-3
Connecting Power	2-4
Connecting an Antenna.....	2-4
Connecting to the PC/104 Bus.....	2-4
Connecting Other Inputs and Outputs.....	2-5
Running the Demonstration Program	2-5
For More Information.....	2-5
CHAPTER 3 CONNECTING AND CONFIGURING THE MODULE	3-1
Connecting the utilityModule.....	3-3
Component Locations.....	3-3
Power Connections, J4 and J5.....	3-4
Auxiliary Connector, J2.....	3-4
Multifunction I/O Connector, J3	3-5
PC/104 Bus Connector, J5	3-10
Configuring the utilityModule	3-13
Setting the Hardware Interrupt	3-13
GPS Receiver Options	3-13
CHAPTER 4 USING THE UTILITYMODULE	4-1
General Information	4-3
I/O Address Map.....	4-3
Using the Module with the Software Driver.....	4-4
Installing the Software Driver.....	4-4
Functions of the Software Driver.....	4-4
Using the Module without the Software Driver.....	4-12
Control Ports	4-12
CHAPTER 5 HARDWARE REFERENCE	5-1
Component Locations.....	5-3
Jumper Settings.....	5-3
Solder Jumper Locations	5-4
Mechanical Dimensions	5-5
APPENDIX A LIMITED WARRANTY.....	A-1



Chapter 1 INTRODUCTION

This manual is meant for engineers and programmers who wish to develop systems based on the IGPS104 utilityModule. It contains hardware, software, and interface specifications and information on software development.

The manual is organized as follows:

- Chapter 1:** INTRODUCTION
- Chapter 2:** GETTING STARTED QUICKLY
- Chapter 3:** CONNECTING AND CONFIGURING THE UTILITYMODULE
- Chapter 4:** USING THE UTILITYMODULE
- Chapter 5:** HARDWARE REFERENCE
- Appendix A:** LIMITED WARRANTY

What is GPS and What Does It Do?

The Navstar Global Positioning System (known as GPS) allows you to locate mobile or stationary objects, giving their absolute position in three-dimensional space. It does so much more easily than previous systems like DECCA, LORAN, and OMEGA. GPS is more cost-effective, accurate, and easy-to-use than any of these previous systems.

The GPS system consists of 18 satellites, placed in groups of three on six orbits approximately 60 degrees from the equatorial plane. The satellites are located to allow users to "see" at least four satellites at all times. With four satellites visible, the position of a GPS receiver in three dimensional space can be determined to at least 25 meter accuracy.

Normally, a GPS receiver needs a free cone of at least 40 degrees on the horizon to see the four required satellites. At sea or high above land this is rarely a problem, as almost nothing comes between the satellites and the receiver. On the ground, however, problems arise in inhabited regions and mountains, where hills, trees, houses, tunnels, and other objects continuously interfere with reception. Because of these problems, the GPS system alone cannot ensure accurate and continuous position measurement on the ground, especially for mobile applications. For these applications, the GPS receiver must be integrated with other sensors, enabling the receiver to cope with temporary absences of the GPS link.

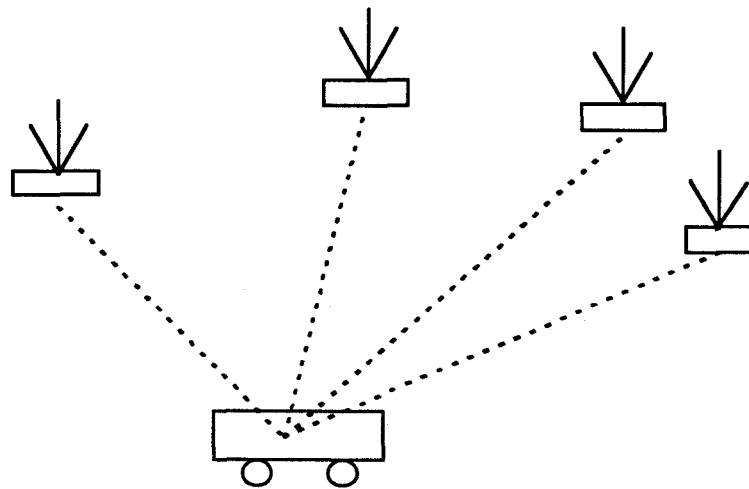


Figure 1-1

To handle occasional absence of the GPS link, a GPS receiver can be used with sensors which are less accurate than GPS over long time periods, but which provide reliable information for short and medium periods. A variety of sensors are available to meet these needs, including directional gyroscopes, rate gyroscopes, odometers, magnetometers, and others. Regardless of the sensors used, all provide either an analog or digital output signal. For that reason, a GPS receiver for mobile or ground-based use should allow acquisition of analog and digital channels in addition to GPS signals.

The IGPS104 Module

The IGPS104 module is a highly integrated device combining a GPS receiver and an intelligent analog and digital data-acquisition system. The module is based on the PC/104 standard and is therefore highly reliable and very compact. Besides integrating analog and digital I/O functions, it also provides an auxiliary RS232 serial port.

A on-board microcontroller manages the data I/O and GPS receiver, using a program stored in a Flash EPROM which is programmable from the PC/104 bus. The module's default parameters are stored in an EEPROM.

The IGPS104 module is designed for applications requiring:

- software compatibility with XT/AT computers
- hardware compatibility with PC/104 systems
- low power consumption
- small physical dimensions
- high reliability
- good noise immunity
- stand-alone use.

A low-cost option for the IGPS104 provides a *differential-ready* GPS receiver. The differential-ready receiver may be used with data from a remote reference GPS receiver, giving even greater accuracy for the most demanding users.

A block diagram of the IGPS104 module is presented below.

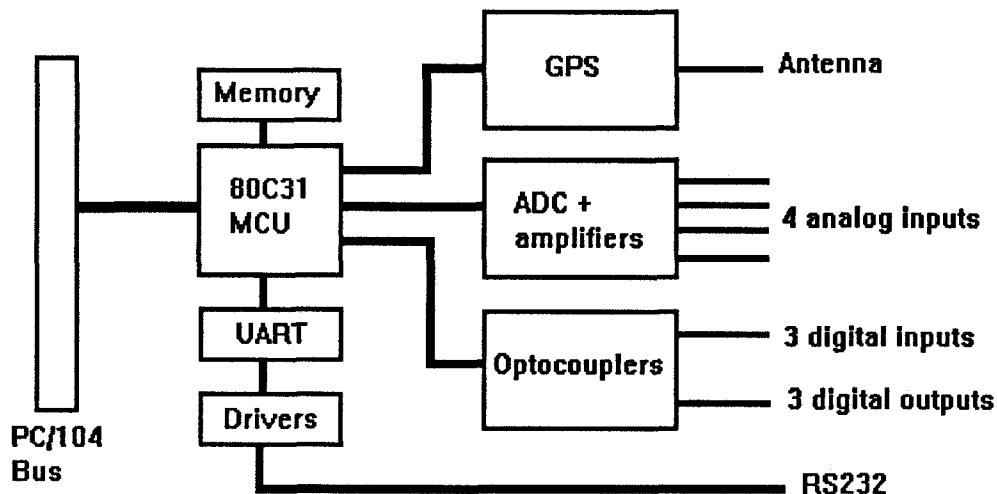


Figure 1-2 IGPS104 Block Diagram

As indicated in figure 1-2, the IGPS104 module integrates, in a 9x9 cm square:

- 8031 microprocessor
- GPS receiver
- 3 opto-isolated digital inputs

- 3 opto-isolated digital outputs
- 4 differential analog inputs
- RS232 serial port (RS422/485 optional)
- EEPROM storage for Setup parameters
- Flash EPROM for the microcontroller, reprogrammable from the PC/104 bus
- Isolated GPS receiver power supply
- Opto-isolated communication link between GPS receiver and microprocessor

The module can be used in two different ways:

- stacked with other PC/104 modules to avoid expensive backplanes and card cages
- as a stand-alone GPS and analog/digital I/O device, with a serial link to a host computer

The IGPS104 module includes software to control the on-board peripherals and communicate with the GPS receiver. If the software provided does not meet your requirements, you may modify and update it using a PC/104 cpuModule with an 8051 cross-compiler and/or assembler.

Module Specifications

CPU

- 8031 microcontroller with 24MHz clock

Memory

- 256 bytes RAM inside the microcontroller
- 64K bytes Flash EPROM program memory
- 256 bytes EEPROM for Setup parameters

GPS Receiver

- 8 received channels
- 6 active channels
- active antenna (includes 5 meter cable)
- 1 pps output
- uses TSIP communication protocol (TAIP and NMEA protocols optional)

Inputs and Outputs

- full-duplex RS232 serial port with baud rates from 50 to 38400 (RS422/485 optional)
- 3 opto-isolated digital inputs (edge counting and period measurement on 2 channels)
- 3 opto-isolated digital outputs (variable duty-cycle waveform generation on two channels)
- 4 differential analog inputs (10 bits resolution)

Connectors

- XT bus according to PC/104 specifications (J5)
- Serial port / Digital I/O / Analog Input / Serial Port connector (J3)
- Auxiliary connector (J2)

Physical Characteristics

- Dimensions: 90.2 x 95.9 mm (3.6" x 3.8"), height 16mm (0.6")
- Supply Voltage: 5V +/- 5%
- Supply Current: 900 mA, typical
- 6-layer, surface-mount PCB
- Operating environment:
 - temperature: 0 - 75 degrees C
 - relative humidity: 5 - 95%
 - altitude: 0 - 3000m
- Storage temperature: -55 to +85 degrees C

Chapter 2 GETTING STARTED QUICKLY

This chapter contains information allowing you to quickly connect and use the IGPS104 utilityModule.

For complete details on connecting, configuring, and using the IGPS104 utilityModule, please refer to the following chapters.

Default Configuration

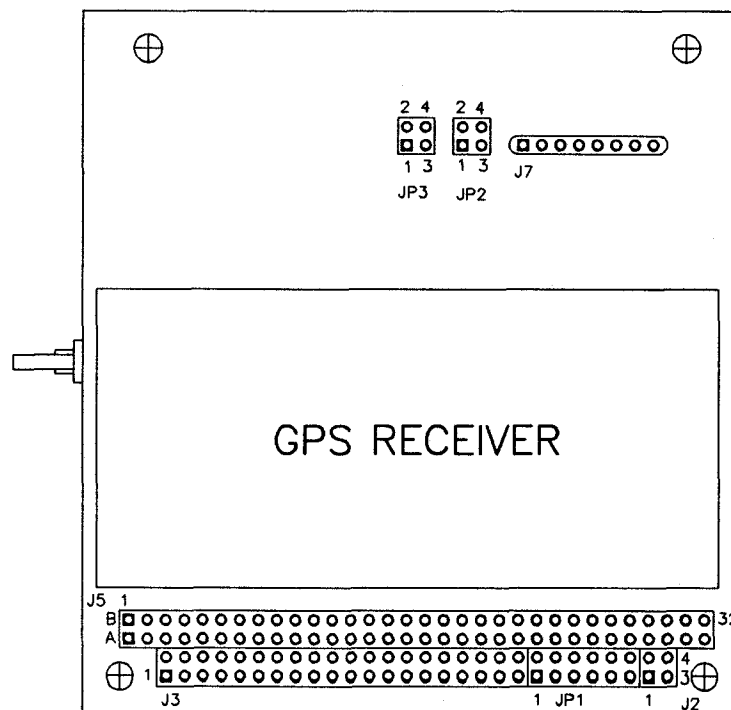
In most cases, you will not need to change the factory configuration of the utilityModule hardware. With the factory settings, you can usually connect the module to a PC/104 system and immediately use it. The factory default settings for the configuration jumpers are shown in the table below.

Default Configuration			
Jumper	Purpose	Setting	Function
JP1	Hardware Interrupt	7-8	IRQ5
JP3	Serial port	1-2,3-4	COM1 @ 3F8
JP2	Serial port interrupt	1-2	IRQ4

The only settings you may need to change before using the utilityModule are those concerning interrupts. If you can use the interrupt settings shown above, you do not need to change any jumper settings on the module. If you need to change the interrupt settings, please refer to Chapter 3 for details.

Basic Connector Locations

The following figure shows the location of the basic connectors needed to use the utilityModule. To use the diagram, position the module with the PC/104 bus connector at the six o'clock position and the component side facing up.



Connecting Power

During normal operation, the utilityModule requires only a +5VDC power supply, and power is supplied through the PC/104 bus connector, J5. If you are placing the utilityModule onto a PC/104 stack that has a +5VDC power supply, you do not need to make additional connections to supply power.

If you are using the utilityModule without a PC/104 stack or with a stack that does not include a power supply, you should refer to the following chapter for details on connecting power to the utilityModule.

Please note that the utilityModule does not require a +12VDC supply for normal operation, but +12VDC is required to re-program the Flash EPROM containing the module's software.

Connecting an Antenna

To use the utilityModule, you must connect an external antenna. Normally the utilityModule is used with the compact, magnetic-mount antenna included, which connects to the utilityModule through a five meter cable.

If you are using the utilityModule with the included antenna, simply plug the antenna cable onto the matching connector on the utilityModule. Then position the antenna so it has an unobstructed view of the sky, so that it will easily receive signals from several satellites. Once you have the utilityModule operational, you can experiment with antenna placement to optimize reception.

Try to position the antenna so that its magnetic base is parallel to the surface of the Earth. If possible, place the antenna on a magnetic surface to provide a ground plane and improve reception.

Connecting to the PC/104 Bus

Since the utilityModule uses a PC/104 stackthrough bus, you need only to stack the IGPS104 on the PC/104 stack. To do this, you will simply plug the PC/104 bus connector, J5, onto the matching connector of your cpuModule.

We recommend you follow the procedure below to ensure that stacking of the modules does not damage connectors or electronics.

- Turn off power to the PC/104 system or stack.
- Select and install standoffs to properly position the utilityModule on the PC/104 stack.
- Touch a metal part of the rack to discharge any buildup of static electricity.
- Remove the utilityModule from its anti-static bag.
- Check that any keying pins in the bus connector are properly positioned.
- Check the stacking order; make sure an XT bus card will not be placed between two AT bus cards or it will interrupt the AT bus signals.
- Hold the utilityModule by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
- Gently and evenly press the utilityModule onto the PC/104 stack.

CAUTION: Do not force the module onto the stack! Wiggling the module or applying too much pressure may damage it. If the module does not readily press into place, remove it, check for bent pins or out-of-place keying pins, and try again.

Connecting Other Inputs and Outputs

You can connect other inputs and outputs to the utilityModule through the 40-pin Multifunction I/O connector, J3. These inputs and outputs include:

- four analog inputs
- three opto-isolated digital inputs
- three opto-isolated digital outputs
- serial port
- auxiliary power inputs

You do not need to connect to any of these inputs or outputs to use the basic functions of the utilityModule. If you wish to use these inputs and outputs, please see the following chapter for information.

Running the Demonstration Program

Once you have completed the previous steps to connect the utilityModule to power, an antenna, and a cpuModule, you may run a demonstration program to verify that the IGPS104 operates correctly.

To run the demonstration program, perform the following steps:

- Boot the PC/104 system containing the IGPS104.
- Insert the IGPS102 Utility Disk into a floppy drive connected to the PC/104 cpuModule.
- Run the demonstration program by entering the command line: DEMO.BAT.

The batch file will load the software driver, then run the demo program. The demo program will display a screen showing information concerning the GPS receiver, analog inputs, and digital inputs and outputs. The analog and digital inputs will update continuously, and the digital outputs will toggle at a fixed rate, allowing you to verify operation of the inputs and outputs.

After several minutes, the GPS receiver should acquire satellite signals and display valid time and position information. Note that the time displayed is based on Universal Coordinated Time (abbreviated UTC), also known as Greenwich Mean Time (GMT). The hours displayed therefore may not match your local time.

If the GPS receiver cannot receive enough satellites to make a position fix, try relocating the antenna to improve reception.

For More Information

This chapter is intended to get the typical user up and running quickly, using the default configuration of the utilityModule. If you need more details than are provided in this chapter, please refer to the following chapters for detailed information on configuring and using your utilityModule.

Chapter 3 CONNECTING AND CONFIGURING THE MODULE

This chapter contains detailed information on connecting and configuring the IGPS104 utilityModule, including information concerning:

- Connectors
- Configuration jumpers
- Input and output specifications
- Interrupts

Connecting the utilityModule

The following section contains detailed information on connecting the IGPS104 utilityModule to external devices.

Component Locations

Figure 2-1 shows the location of the connectors, jumpers, and major components of the IGPS104 utilityModule. To use the diagram, position the module with the PC/104 bus connector at the six o'clock position and the component side facing up.

Table 2.1 lists the connectors of the module and their functions.

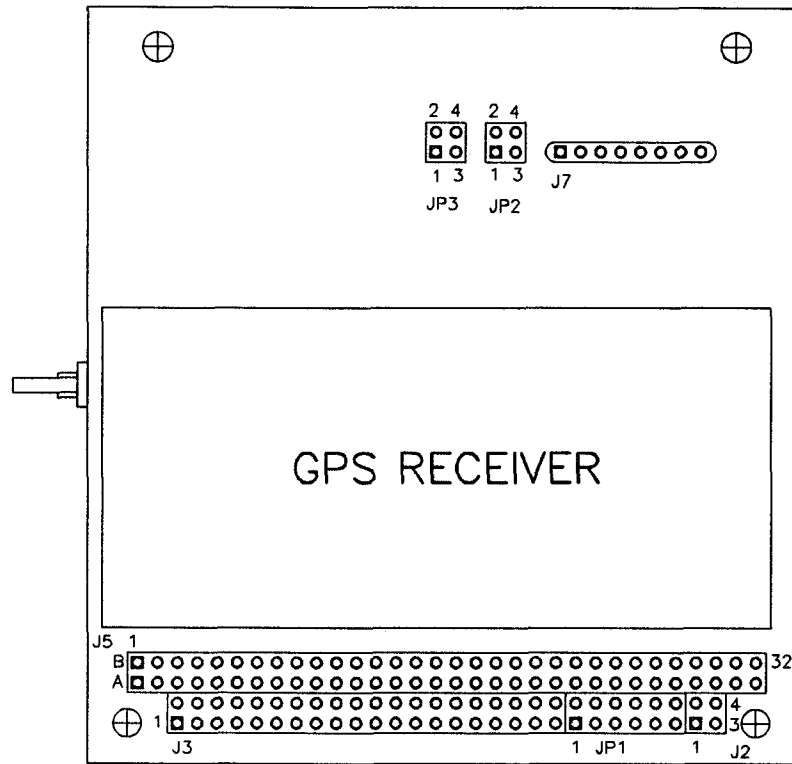


Figure 3-1 Component locations

Table 3-1 Connectors		
Connector	Function	Size
J2	Auxiliary connector	4 pin
J4	Multifunction I/O	40 pin
J5	PC/104 Bus	64 pin

NOTE: Pin 1 of connectors is indicated by a square solder pad on the circuit board.

Power Connections, J4 and J5

During normal operation, the utilityModule requires only a +5VDC power supply. You can supply power to the utilityModule through either the PC/104 bus connector, J5, or the Multifunction I/O connector, J4.

WARNING: Incorrectly connecting power will almost certainly *destroy* the module. Verify all power connections *before* applying power to the system.

Connector / Pins	Signal	Tolerance
J5 / 6,58	+5VDC	+/- 5%
J5 / 10	-5VDC	+/- 5%
J5 / 18	+12VDC	+/- 5%
J5 / 14	-12VDC	+/- 5%
J5 / 2,62,63,64	Ground	NA

The -5VDC and -12VDC supplies are not used by the module and are not generated by it: they are only conveyed on the PC/104 bus connector J5.

The +12VDC supply is not used for normal operation of the module, but is required to re-program the Flash EPROM containing the module's software.

Power Consumption

The following table gives the typical power consumption of the utilityModule using the included active antenna:

Total consumption	Module	Receiver	Active antenna
900mA	350mA	500mA	50mA

Auxiliary Connector, J2

Connector J2 of the module provides access to the following signals connected between the IGPS102 utilityModule and the GPS receiver:

- 1 pulse per second output
- Backup battery input
- GPS ground
- GPS supply voltage

The following table lists the signals of the Auxiliary Connector, J2.

Pin	Signal	Function	in/out
1	1PPS	1 pulse per second	out
2	Vbatt	GPS back-up battery (+)	in
3	GPS Gnd	GPS Ground	N/A
4	GPS Vcc	GPS Supply Voltage	out

Signal Descriptions

- **1PPS** (1 pulse per second) provides a time reference by being pulsed low for 1 μ s at a 1 Hz rate. It is an open-collector signal referred to module ground. The output voltage at low level is less than 0.8 volts when sinking 80 mA.
- **Vbatt** is the positive input for an optional backup battery. The input voltage range is +3.5 to +4.75 volts. The GPS receiver uses the backup battery at power-down to preserve information concerning the last received data. This allows quick recovery of the satellite signals when power is restored. The battery (-) should be connected to the GPS_Gnd described below.
- **GPS_Gnd** is the GPS receiver ground. It is isolated from system ground to provide good noise immunity.
- **GPS_Vcc** is the 5 volt supply for the GPS receiver. Maximum current draw is 40mA.

Multifunction I/O Connector, J3

The 40-pin Multifunction I/O connector, J3, provides the following functions:

- Analog inputs
- Opto-isolated digital inputs
- Opto-isolated digital outputs
- RS232C serial port
- Auxiliary power inputs

The following table lists the pins and functions of connector J3.

Pin	Signal	Function	in/out
1	CH0+	analog channel 0 +	in
2	CH0 -	analog channel 0 -	in
3	CH1+	analog channel 1+	in
4	CH1 -	analog channel 1 -	in
5	GND	bus PC104 ground	N/A
6	GND	bus PC104 ground	N/A
7	+12VDC	bus PC104 +12 volt	N/A
8	VCC	bus PC104 +5 volt	N/A
9	CH2+	analog channel 2 +	in
10	CH2 -	analog channel 2 -	in

11	CH3+	analog channel 3 +	in
12	CH3 -	analog channel 3 -	in
13	OUT2C	digital out 2 (collector)	out
14	OUT2E	digital out 2 (emitter)	out
15	OUT1C	digital out 1 (collector)	out
16	OUT1E	digital out 1 (emitter)	out
17	OUT0C	digital out 0 (collector)	out
18	OUT0E	digital out 0 (emitter)	out
19	IN0A	digital in 0 + (anode)	in
20	IN0C	digital in 0 - (cathode)	gnd
21	IN1A	digital in 1 + (anode)	in
22	IN1C	digital in 1 - (cathode)	gnd
23	IN2A	digital in 2 + (anode)	in
24	IN2C	digital in 2 - (cathode)	gnd
25	VCC	bus PC104 +5 volt	out
26	VCC	bus PC104 +5 volt	out
27	GND	bus PC104 ground	N/A
28	+12VDC	bus PC104 +12 volt	out
29	DCD	serial port DCD	in
30	DSR	serial port DSR	in
31	RXD	serial port RxD	in
32	RTS	serial port RTS	out
33	TXD	serial port TxD	out
34	CTS	serial port CTS	in
35	DTR	serial port DTR	out
36	RI	serial port RI	in
37	GND	bus PC104 ground	N/A
38	GND	bus PC104 ground	N/A
39	GND	bus PC104 ground	N/A
40	GND	bus PC104 ground	N/A

Analog Inputs

There are 4 analog inputs on the module. Each passes through an amplifier circuit before reaching the 10-bit analog-to-digital converter used to read the inputs. The inputs to the amplifier are differential to assure good noise immunity. An approximate schematic of the input circuit is shown below:

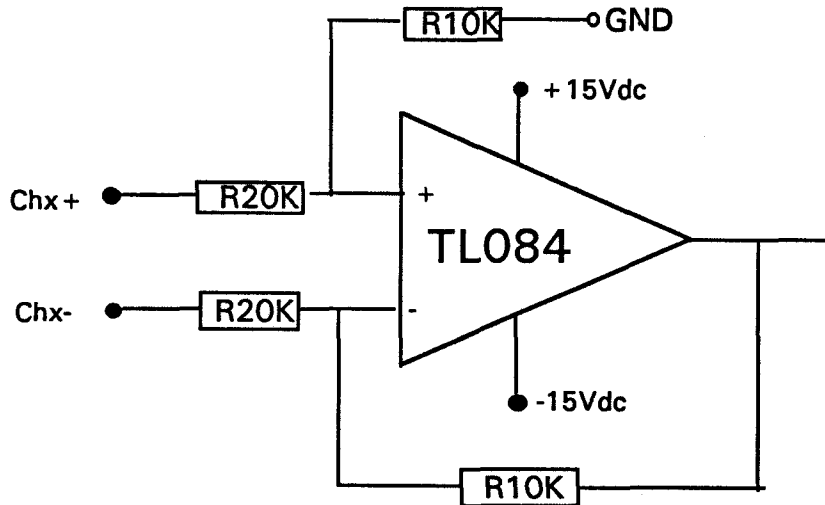


Figure 3-2

The input impedance of each analog channel amplifier is 40 Kohm. The gain of each amplifier may be changed to allow the following input voltage ranges:

- 0.0 to +4.0V
- -2.0 to +2.0V (factory default setting)
- -0.2 to +0.2V
- 0.0 to +0.4V

The gain for each input channel is set by solder jumpers on the bottom of the utilityModule, as listed in the following four tables. Please refer to the Hardware Reference section for the locations of the solder jumpers.

Table 3-6 Analog Channel 0			
RANGE	JPS11	JPS12	JPS15
-2V to +2V	OPEN	OPEN	1-2
0V to +4V	OPEN	OPEN	2-3
-0.2V to +0.2V	CLOSED	CLOSED	1-2
0V to +0.4V	CLOSED	CLOSED	2-3

Table 3-7 Analog Channel 1			
RANGE	JPS13	JPS14	JPS15
-2V to +2V	OPEN	OPEN	1-2
0V to +4V	OPEN	OPEN	2-3
-0.2V to +0.2V	CLOSED	CLOSED	1-2
0V to +0.4V	CLOSED	CLOSED	2-3

Table 3-8 Analog Channel 2			
RANGE	JPS7	JPS8	JPS15
-2V to +2V	OPEN	OPEN	1-2
0V to +4V	OPEN	OPEN	2-3
-0.2V to +0.2V	CLOSED	CLOSED	1-2
0V to +0.4V	CLOSED	CLOSED	2-3

Table 3-9 Analog Channel 3			
RANGE	JPS9	JPS10	JPS15
-2V to +2V	OPEN	OPEN	1-2
0V to +4V	OPEN	OPEN	2-3
-0.2V to +0.2V	CLOSED	CLOSED	1-2
0V to +0.4V	CLOSED	CLOSED	2-3

Digital Inputs

The module has three opto-isolated digital inputs. Figure 2-3 below illustrates the digital input circuitry.

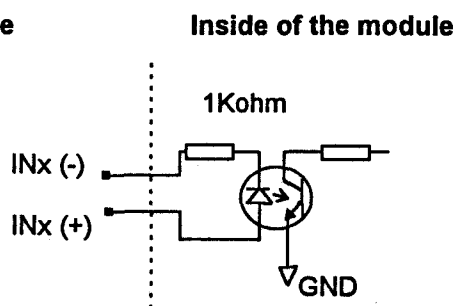


Figure 3-3 Digital Input Schematic

The minimum input current required to read a logic 1 is 4mA. The maximum input voltage to the digital inputs is 30Vdc. The minimum impulse duration which can be recognized is 20 μ s.

Digital Outputs

The module has three opto-isolated digital outputs. The maximum output sink current is 5mA and the maximum applied voltage is 5Vdc.

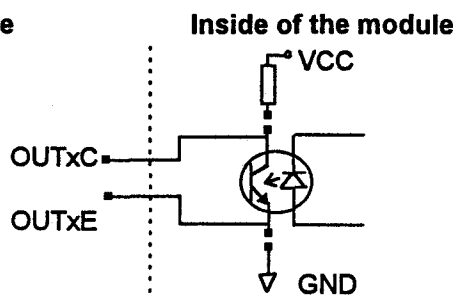


Figure 3-4 Digital Output Schematic

Configuring TTL-compatible Digital Outputs

The three output channel optocouplers can be configured to give TTL-compatible output levels by positioning solder jumpers as shown below.

- To configure channel 0 as a TTL output, close solder jumpers JPS1 and JPS2
- To configure channel 1 as a TTL output, close solder jumpers JPS3 and JPS4
- To configure channel 2 as a TTL output, close solder jumpers JPS5 and JPS6

When the outputs are configured as TTL-compatible, the output pins become: OUT0C, OUT1C, OUT2C.

Please refer to the Hardware Reference section for the locations of the solder jumpers.

RS232 Serial Port

Table 3-10 gives the pins of the Multifunction I/O connector, J4, used for the serial port, and lists the connections to give an XT or AT-compatible serial port using DB25 or DB9 connectors.

Pin	Signal	Function	in/out	DB25	DB9
29	DCD	Data Carrier Detect	in	8	1
30	DSR	Data Set Ready	in	6	6
31	RX	Receive Data	in	3	2
32	RTS	Request To Send	out	4	7
33	TX	Transmit data	out	2	3
34	CTS	Clear To Send	in	5	8
35	DTR	Data Terminal Ready	out	20	4
36	RI	Ring Indicator	in	22	9
37,38	GND	Signal Ground	--	7	5

RS422/485 Port

The module may optionally be factory configured with an RS422/485 port in place of the RS232 port. Contact factory technical support for information on this option.

Serial Port Address and Interrupt

The serial port I/O address is selected using jumper JP3, while the corresponding interrupt can be selected using jumper JP2.

The default settings for the serial port are COM1 at I/O address 3F8H, using interrupt IRQ4.

JP3 pins 1-2	JP3 pins 3-4	I/O Address	Port	Notes
inserted	inserted	3F8H	COM1	default
inserted	not inserted	2F8H	COM2	
not inserted	inserted	3E8H	COM3	
not inserted	not inserted	---	disabled	

JP2 pins 1-2	JP2 pins 3-4	Interrupt	Notes
inserted	inserted	IRQ3	
inserted	not inserted	IRQ4	default
not inserted	inserted	IRQ5	
not inserted	not inserted	---	disabled

PC/104 Bus Connector, J5

Connector J5 carries the signals of the PC/104 bus; the signals match the IEEE P966 standard. The pinout and the pin functions are listed in the following table.

Pin	J5 line A	J5 line B
1	IOCHCHK*	0V
2	SD7	RESETDRV
3	SD6	+5V
4	SD5	IRQ9
5	SD4	-5V
6	SD3	DRQ2
7	SD2	-12V
8	SD1	ENDXFR*
9	SD0	+12V
10	IOCHRDY	(KEY)
11	AEN	SMEMW
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REFRESH
20	SA11	SYSCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	TC
28	SA3	BALE
29	SA2	+5V
30	SA1	OSC
31	SA0	0V
32	0V	0V

NOTES: Signals used by the module are shown in **BOLD**.

Signals followed by a (*) are active low.

All signals on the bus can drive a maximum current of 6mA at TTL levels.

Signals on Connector J5

The following table contains brief descriptions of the various PC/104 bus lines.

Signal	I/O	Description
AEN	O	Address ENable: when this line is active (high), this means that a DMA transfer is being performed and therefore the DMA controller has control over the data bus, the address bus, and the control lines.
BALE	O	Bus Address Latch Enable, active high. When active, it indicates that the address lines SA0 to SA19 are valid.
DACK1/3*	O	DMA ACKnowledge 1-3, active low, are the answer lines to DMA requests.
DRQ1/3	I	DMA Request 1-3: these are asynchronous lines used by the peripheral devices to request DMA service. They have rising priority from DRQ1 up to DRQ3. A DMA request is performed by setting the DRQ line high and holding it until the corresponding DACK line is activated.
ENDXFR*	I/O	The only synchronous signal of the PC/104 bus, and active low. It indicates that the current bus cycle must be performed at 0 wait states. It is used only for 16-bit boards.
IOCHCHK	I	I/O Channel Check, active low, indicates an error condition that cannot be corrected.
IOCHRDY	I	I/O Channel Ready: this line, usually high (ready) is held at low level by devices which need longer bus cycles.
IOR*	O	I/O Read, active low, indicates when the devices present on the bus can send their information on the data bus.
IOW*	O	I/O Write, active low. When active, it allows the peripheral devices to read data present on the data bus.
IRQ3/9	I	Interrupt Request 2 to 7: active on rising edge. IRQ9 has top priority, the other lines have decreasing priority starting from IRQ3 down to IRQ7. The interrupt request is performed by changing the line from low to high and holding it until the microprocessor has recognised it.
KEY	N/A	Key for connector insertion.
OSC	O	OSCillator: clock with a 70 ns period and a 50% duty cycle.
REFRESH*	O	This line is active low and indicates that the current bus cycle is a DRAM refresh cycle. Refresh cycles are activated every 15 microseconds.
RESETDRV	O	This line, active high, is used to reset devices on the bus at power-on or after a reset command.
SA0/19	O	Address bits from 0 to 19: these lines are used to address the memory space and the I/O space. SA0 is the least significant bit while SA19 is the most significant bit.
SD0/7	I/O	Data bits: these are the data bus lines. SD0 is the least significant bit, SD7 is the most significant bit.
SMEMR*	O	Memory Read command, active low.

SMEMW*	O	Memory Write command, active low.
SYSCLK	O	System Clock, programmed at 8Mhz with a 50% duty cycle.
TC	O	Terminal Count: this line is active high and indicates the conclusion of a DMA transfer.

NOTE: Signals used by the utilityModule are shown in **BOLD**.

Notes on Interfacing to the PC/104 Bus

- Driving capacity of PC/104 add-on boards must be at least 6mA at low TTL level.
- Usually, no termination is required for PC/104 bus lines. If necessary, we recommend terminating each signal line of the bus with a series resistor and capacitor (40-60 ohms and 30-70 pF) between the signal and ground.
- If you need to extend the PC/104 bus to other boards, we recommend a bus extender using a 4-layer PCB (2 signal layers, 1 ground layer, and 1 power supply layer) to avoid crosstalk interference. If you use a 2-layer PCB, one layer must be entirely used as ground. Ribbon cable connections should be avoided, but if they must be used, keep them shorter than 20cm. With ribbon cable connections, the bus control signals (**IOR***, **IOW***, **MEMR***, **MEMW***, **RESET**, etc.) will be inevitably exposed to crosstalk, and possibly disturbed by address and data bus transitions.
- Some expansion boards have TTL inputs particularly sensitive to interference. Specifically, they may have problems with sensitivity of the **RESET** signal. If you notice spurious reset problems, try connecting a 200-500pF capacitor between the **RESET** signal and ground.
- If you build your own expansion boards, you must carefully select the components that will interface to the PC/104 bus. Make sure you use components having T_{rise} and T_{fall} times compatible with those of the LS and HCT logic families.

Configuring the utilityModule

The following section contains information on configuring the hardware of the utilityModule which is not covered in the previous sections.

Setting the Hardware Interrupt

The module exchanges data with the PC/104 bus using the 80C31 microprocessor on the utilityModule. The data transfers are mapped in I/O address space using a hardware interrupt, which is factory set as interrupt IRQ5.

If you need to, you can change the hardware interrupt used by the utilityModule. The possible interrupt settings are from IRQ2 through IRQ7, as shown in the following table:

position 1-2	IRQ2	
position 3-4	IRQ7	
position 5-6	IRQ6	
position 7-8	IRQ5	Default
position 9-10	IRQ4	
position 11-12	IRQ3	

NOTE: You may only insert one jumper at a time, corresponding to the selected interrupt. All other jumpers must be open.

GPS Receiver Options

The IGPS104 utilityModule is available with one of four versions of the GPS receiver:

- Module with standard GPS and TSIP interface
- Module with differential-ready GPS
- Module with standard GPS and TAIP interface (non-standard)
- Module with standard GPS and NMEA interface (non-standard)

The GPS receiver with TSIP interface is used on the standard IGPS104-GPS module. The Differential-ready GPS receiver is available on the IGPS104-DGPS module. The other receiver versions are not standard products; please contact the factory for information on their availability.

Chapter 4 USING THE UTILITYMODULE

This chapter provides information for users who wish to develop software using the IGPS104 module in one of the two following ways:

- With the IGPS102 software driver installed
- Without the software driver, using only the firmware of the GPS receiver

Please contact the factory for information on using the module as a standalone unit or in other non-standard ways.

General Information

I/O Address Map

As with all standard PC/104 boards, the Input/Output (I/O) addresses of the module are specified by 10 bits (SA0-SA9). The table below lists the I/O addresses reserved for the utilityModule.

NOTE: You *must* ensure that any other PC/104 modules or peripherals in your system *do not* use the reserved addresses listed below, or malfunctions will occur.

Address	Device
320H	data port
321H	command port
322H	command port
323H	transmit byte
324H	reserved
325H	output buffer full
326H	input buffer full
327H	reset

Using the Module with the Software Driver

The IGPS104 module comes with a software driver to simplify its use. The software driver is a terminate-and-stay-resident (TSR) program which allows you to:

- communicate with the IGPS104
- read the 4 analog inputs
- read the 3 digital inputs
- write to the 3 digital outputs

The driver routine uses two interrupts:

- INT 50H, software interrupt used for exchanging data with the application. You cannot re-assign this interrupt
- INT *i*, a hardware interrupt which you must select when installing the driver. You assign this interrupt when installing the software driver, as described below.

Installing the Software Driver

You install the TSR software driver by entering the following command line:

```
C:>GPS -i
```

where *i* is replaced with the hardware interrupt number you wish to use.

NOTE: Before you load the software driver, you must set the hardware interrupt between IRQ2 and IRQ7, by positioning jumper JP1 on the IGPS104 utilityModule. See the previous chapter for information on Setting the Hardware Interrupt.

If you wish to always have the software driver installed on your system, simply place the GPS command line into an AUTOEXEC.BAT file on your boot disk.

Once installed, the driver uses about 10Kbytes of memory. You can un-install the driver and regain that memory by entering the command line:

```
C>GPS
```

Functions of the Software Driver

The software driver allows you to perform six major operations on the IGPS104 utilityModule. These are:

- Initializing the GPS receiver
- Accessing the GPS Receiver
- Reading the Analog Inputs
- Reading the Digital Inputs
- Controlling the Digital Outputs
- Accessing the EEPROM

To perform these operations, the software driver provides the functions listed below:

Function 0H	transmit a message to GPS receiver
Function 1H	message ready from GPS receiver
Function 2H	take the message from GPS receiver
Function 3H	start reception
Function 4H	stop reception
Function 5H	reset the receive buffer
Function 6H	start module
Function 7H	read digital inputs
Function 8H	read digital inputs count
Function 9H	read digital inputs period
Function AH	select count edges
Function BH	read analog channels
Function CH	perform digital outputs
Function DH	access the EEPROM

The following sections discuss the operations you can perform on the IGPS104 and the functions needed to do so.

Initializing the Module

Before performing any other operation using the interface software, you must initialize the utilityModule by using the following interrupt routine call:

```
mov    ah, 6
int    50h
```

If you do not perform this call, the module remains in a reset state.

If you wish to use the receive and transmit functions with the utilityModule, you must follow the preceding initialization call with a call to enable the receiver:

```
mov    ah, 3
int    50h
```

Note that you do not need this second call to enable the receiver if you will only access the analog and digital inputs and outputs.

The following are the interrupt functions used to initialize the utilityModule:

Function 3 enable message reception by the IGPS module
 INPUT AH = 3

RETURN None

Function 6 Initialize the GPS module. This must be the first function called after power-on. It can be called again afterwards, to reset the module without interrupting the GPS receiver.

INPUT AH = 6

RETURN none

Note that error codes for these routines are identical to those listed in the following section.

Accessing the GPS Receiver

There are five interrupt functions provided to access the GPS receiver.

The functions for communicating with the GPS receiver use a message protocol based on the TSIP standard (version 1.16). The software driver does not check the content of messages, but does assure correct syntax while transmitting and receiving.

When transmitting, the software driver:

- adds the correct header to the transmitted message
- performs "stuffing" operations on the message
- transforms integers, floats, and long integers from DOS format to TSIP format
- adds the correct trailer to the transmitted message

When receiving, the software driver:

- removes the header from the received message
- performs "unstuffing" operations on the message
- checks that the number of received bytes is coherent with the message
- transforms integers, floats, and long integers from TSIP format to DOS format
- removes the trailer from the received message
- provides a 2Kbyte circular receive buffer.

You can communicate with the GPS receiver using the following interrupt functions:

Function 0 transmit a message to the GPS receiver

- INPUT
- AH = 0
 - AL = msg number
 - BL = value indicating either *request* or *set* function (see notes below).
 - DS:SI = pointer to the information part of the message

RETURN AH = error code

NOTES The two registers DS:SI point to the information part of the message to be transmitted. The information part must *not* contain any of the following parts, as these are added by the software driver:

- initial DLE
- message identifier
- final DLE
- final ETX

The message format must agree with the TSIP (version 1.16) specifications. If integers, floats, or doubles are part of the message, they must be in DOS format.

As messages have different lengths, the message type must be specified in register AL, using the following codes:

Message:

AL = 1DH: BL = 0 clear oscillator
BL = 1 set oscillator offset

AL = 35H: BL = 0 request serial I/O options
BL = 1 set serial I/O options

AL = 38H: BL = 0 request satellite system data
BL = 1 load satellite system data

AL = 3DH: BL = 0 request data channel A configuration
BL = 1 set data channel A configuration

AL = 71H: BL = 0 request position filter
BL = 1 set position filter

AL = 73H: BL = 0 request altitude filter
BL = 1 set altitude filter

AL = 75H: BL = 0 request high 8
BL = 1 set high 8

NOTES The length of the buffer pointed to by DS:SI depends on the selection made. If the message to be transmitted has no information part and simply consists of a message identifier, the pointer (DS:SI) need not be passed.

Function 1 Check whether messages from the GPS are present in the receive buffer

INPUT AH = 1

RETURN AH = 0, if there are messages in the receive buffer.
AH = 1, if the buffer is full. In this case a buffer flush is performed.
AH = 0FFH, if there are no messages in the receive buffer.

Function 2 Read a message from the receive buffer. If no valid message is present, wait for one to arrive.

INPUT AH = 2

ES:DI points to a buffer which is to receive the message

RETURN If AH = 0,
AL = message identifier
CX = message length
ES:DI points to a buffer with the information part of the message
If AH <> 0, then AH = error code

NOTES In Receive mode, only messages conforming to the TSIP 1.16 protocol are handled.
The buffer pointed to by ES:DI receives only the information part of the message. If integers, float or long integers are included in the message, the values are returned in DOS format.

Function 4 Disable message reception by the GPS receiver
 INPUT AH = 4

RETURN none

Function 5 Flush the receive buffer
 INPUT AH = 5

RETURN none

The following are the error codes which may be returned by any function call.

Error Codes 01H transmission buffer full
 02H transmission request of message with ID < 1DH and ID > 75
 03H transmission request of message with invalid ID
 04H subfunction required in invalid transmission
 05H received message with invalid length
 06H received message with invalid ID
 07H receive-data timeout from the GPS module
 08H invalid command
 09H erase operation attempted in an EEPROM reserved location
 0AH invalid EEPROM address
 0BH address outside the EEPROM address space

Reading the Analog Inputs

Four analog inputs are available on the module. Each input can be read individually and is sampled by a 10-bit ADC every 400 microseconds. The analog inputs are filtered by a one-pole, low-pass digital filter having a 56Hz cut-off frequency.

You can read the values of the analog inputs using the following interrupt functions:

Function 0BH reads the value of the analog channel.
 INPUT • AH = 0BH
 • if AL = 0, reads analog channel 0
 • if AL = 1, reads analog channel 1
 • if AL = 2, reads analog channel 2
 • if AL = 3, reads analog channel 3
 • ES:DI points to the buffer which receives the word holding the channel value

RETURN if AH = 0, then no errors occurred
 if AH <> 0, then AH = error code

Reading the Digital Inputs

The module provides three digital inputs. For each input, software can perform the following operations:

- return input state
- select number-of-transitions or period-measurement functions
- return number-of-counts measured on digital input (only for 2 channels).
- return period-measurement for digital input (only for 2 channels)

You can read the values of the digital inputs using the following functions:

Function 07H	Reads the state of the digital inputs.
INPUT	<ul style="list-style-type: none"> • AH = 07H • if AL bit 0 = 1, reads digital input 0 • if AL bit 1 = 1, reads digital input 1 • if AL bit 2 = 1, reads digital input 2
RETURN	<p>If AH = 0, then:</p> <ul style="list-style-type: none"> • AL bit 0 = state of digital input 0 (if selected) • AL bit 1 = state of digital input 1 (if selected) • AL bit 2 = state of digital input 2 (if selected) <p>If AH \neq 0, then AH = error code</p>
Function 08H	Reads number of counts recorded on the digital input channels. Counts are recorded on the edge transition selected by function 0AH.
INPUT	<ul style="list-style-type: none"> • AH = 0AH • if AL bit 0 = 1, reads count of digital input 0 • if AL bit 1 = 1, reads count of digital input 1 • ES:DI points to the buffer which receives the count word.
RETURN	<p>if AH = 0, then no errors occurred</p> <p>if AH \neq 0, then AH = error code</p>
NOTES	To obtain accurate counts, do not apply signals having a frequency over 1KHz to each channel.
Function 09H	Returns the period of the signal on the digital input, using the edge transition selected by the 0AH function.
INPUT	<p>AH = 09H</p> <ul style="list-style-type: none"> • if AL bit 0 = 1, reads period of digital input 0 • if AL bit 1 = 1, reads period of digital input 1 <p>ES:DI points to the buffer which receives a double word representing the period expressed in hundreds of microseconds.</p>
RETURN	<p>if AH = 0, then no errors occurred</p> <p>if AH \neq 0, then AH = error code</p>
NOTES	To obtain accurate period measurements, do not apply input signals with frequency over 1kHz. Note that period is measured with an accuracy of +/- 100 microseconds.

Function 0AH Selects the edge transition of the digital inputs on which counting or period measurement functions are performed.

INPUT AH = 0AH
 if AL bit 0 = 0, uses falling edge of digital input 0
 bit 0 = 1, uses rising edge of digital input 0
 if AL bit 1 = 0, uses falling edge of digital input 1
 bit 1 = 1, uses rising edge of digital input 1

RETURN if AH = 0, then no errors occurred
 if AH > 0, then AH = error code

Controlling the Digital Outputs

The module provides three digital outputs which can be driven one of two ways:

- fixed state (high or low)
- variable duty cycle signal

Note that variable duty cycle outputs are only available on digital outputs 0 and 1.

You may control the digital outputs using the functions below:

Function 0CH Control digital output.

INPUT AH = 0CH

 For digital outputs 0 and 1:
 AL = 0, control of digital output 0
 = 1, control of digital output 1

- BL = low-time of signal, expressed in hundreds of microseconds
- (if BL = 0, the output signal is always high)
- BH = high-time of signal, expressed in hundreds of microseconds
- (If BH = 0, the output signal is always low)

 For digital output 2:
 AL = 2, control of digital output 2

- BL = 0, make output low
- BL = 1, make output high

RETURN if AH = 0, then no errors occurred
 if AH > 0, then AH = error code

Accessing the EEPROM

The EEPROM on the module has 1024 memory bits organized in 64 words, each 16 bits long. Of these, seven words are used by the firmware to correct offsets of the analog-to-digital converter. The remainder are available to you as non-volatile memory locations.

Note that to correctly write to the EEPROM, you must perform a complete write cycle, including the following steps:

- enable the EEPROM
- erase the relevant locations
- write the relevant locations
- disable the EEPROM

To only read the EEPROM, you do not need to enable the EEPROM and can simply use the read routine.

You can access the EEPROM using the following functions:

Function 0DH	EEPROM control.
INPUT	<ul style="list-style-type: none"> • AH = 0DH • AL = 0 enables EEPROM writing and erasing. Remains enabled until a disable command is given. • AL = 1 disables writing and erasing the EEPROM. This must always be the last operation performed on the EEPROM to avoid undesired access to the device. • AL = 2 erases words, starting from the address specified in the CH register, for the length specified in the CL register. Note that CH must contain a word address greater than 7, and that the specified length in CL must be compatible with the size of the EEPROM. • AL = 3 reads the word specified by the address contained in the CH register, and places it in the buffer pointed to by registers ES:DI • AL = 4 writes into the EEPROM, at the address specified in the CH register, the value contained in the BX register. Note that CH must contain a word address greater than 7 and less than or equal to 40H.
RETURN	<p>if AH = 0, then no errors occurred if AH > 0, then AH = error code</p>

Using the Module without the Software Driver

As mentioned earlier, you can use the IGPS104 utilityModule without using the TSR software driver. To do so, you must communicate directly with the 80C31 microcontroller on the utilityModule. You do that by reading and writing a set of control "ports" or registers, as described in the next section.

Control Ports

You can communicate directly with the GPS module using a set of control ports, which are mapped at I/O addresses 320H through 327H.

The following sections explain the functions of the control ports for the GPS module and give examples illustrating their use.

326H Port

This port is used to synchronize the exchange of information. When bit 0 of this port reads as 0, the IGPS module is ready to accept data.

Writing to this port will reset the IGPS module (see port 327H).

An example is given below:

```

;scope: wait for GPS module ready
waitIBF      proc    near
              push   ax
              push   dx
              push   cx
              mov    cx,0                ;timeout
lpwaitibf:
              mov    dx,326h            ;wait input buf full
              in     al,dx
              test   al,1
              jz    waitibfexit        ;GPS module ready to accept command
              loop   lpwaitibf         ;GPS module not ready
              stc
              jmp   wrestst
waitibfexit:  clc
wrestst:
              pop    cx
              pop    dx
              pop    ax
              ret
waitIBF      endp

```

325H Port

This port is also used to synchronize the exchange of information. When bit 1 of this port is set to 1, the IGPS module is ready to transmit data to the PC/104 bus. Bit 1 is reset by a read operation performed on port 320H. An example is presented below:

```

waitOBF      proc    near
              push   ax
              push   dx
              push   cx
              mov    cx,0                ;timeout
lpwaitobf:
              mov    dx,325h            ;some data from micro ?
              in     al,dx
              test   al,1                ;yes
              jnz   waitobfexit

```

```

        loop    lpwaitobf
        stc
        jmp    errobf

waitobfexit:  clc
errobf:
        pop    cx
        pop    dx
        pop    ax
        ret
waitOBF     endp

```

320H Port

This port is used to read data from the GPS module. The following is an example:

```

;scope: get byte from GPS module
getbyte  proc    near
        call   waitOBF          ;wait for answer from module
        mov    dx,320h          ;get byte from module
        in     al,dx
        ret
getbyte  endp

```

321H Port

Every time it receives a byte from the GPS receiver, the IGPS104 module generates an interrupt on the PC/104 bus. The cause of the interrupt is indicated on port 321H, according to the following code:

- code 01H receive interrupt
- code 02H parity error
- code 03H framing error

When the interrupt occurs, regardless of its cause, you must read port 320H, which contains the received data. The following is an example of an interrupt routine:

```

;interrupt routine
.....
.....
mov     dx,321h          ;get interrupt cause
mov     ah,0
in      al,dx
cmp     al,intablen     ;is for me
ja      intexit
mov     bx,offset inttable ;do interrupt service
add     ax,ax
add     bx,ax
call    cs:[bx]
.....
.....
inttable  dw     notused          ;0 table
          dw     rxint           ;read request
          dw     parityerr       ;parity error occurs
          dw     framing         ;framing error
intablen  equ    ($-inttable)/2
.....
.....
;process parity error
parityerr proc    near
          call   getbyte         ;get byte from microcontroller

```

```

inc      byte ptr cs:rxerror ;increment error count
ret
parityerr endp
.....
.....
    
```

322H Port

This is the port where the IGPS module accepts commands, with replies returned on port 320H. The commands accepted at port 322H are listed below:

Code Dec.	Meaning	Data Returned at 320H Port
00	read the state of the digital channels	1 byte of which bits 0,1,2 give the state of the digital inputs (0,1,2)
01	read count of digital channel 0	2 bytes (first LSB, then MSB) with the number of counts performed for channel 0
02	read count of digital channel 1	2 bytes (first LSB, then MSB) with the number of counts performed for channel 1
03	read count of digital channels 0 and 1	4 bytes, of which the first two concern digital input 0, while the other two concern digital input 1
04	read period of digital channel 0	4 bytes transmitted from the less significant to the most significant, containing the period of the signal measured in 100us multiples.
05	read period of digital channel 1	4 bytes transmitted from the less significant to the most significant, containing the period of the signal measured in 100us multiples.
06	read period of digital channels 0 and 1	8 bytes, of which the first four are the signal period on digital input 0 and the other four are the period on digital input 1
07	read analog channel 0	2 bytes (first LSB and then MSB) with the filtered reading of the input present on analog channel 0
08	read analog channel 1	2 bytes (first LSB and then MSB) with the filtered reading of the input present on analog channel 1
09	read analog channel 2	2 bytes (first LSB and then MSB) with the filtered reading of the input present on analog channel 2
10	read analog channel 3	2 bytes (first LSB and then MSB) with the filtered reading of the input present on analog channel 3
11	reserved	
12	Receive start	none
13	Receive stop	none
14	trigger of digital channel 0 on rising edge	none
15	trigger of digital channel 0 on falling edge	none
16	trigger of digital channel 1 on rising edge	none

17	trigger of digital channel 1 on falling edge	none
18	set digital output channel 0	note 1
19	set digital output channel 1	note 1
20	set digital output channel 2	note 2
21	EEPROM enable	none
22	EEPROM erase	note 3
23	read a word from the EEPROM	note 4
24	write a word into the EEPROM	note 5
25	disable the EEPROM	none

NOTES:

- 1 Commands 18 and 19 do not wait for an answer but must be followed by two other writes to port 322H. The first byte must contain the time to keep the digital output low, expressed in 100us multiples. The second byte must contain the time to keep the digital output high.
- 2 Command 20 does not wait for an answer but must be followed by another write to port 322H, of a command byte specifying the state for the output. If this byte is zero, the port will be set to a low level, otherwise the digital output will be set high.
- 3 Command 22 does not wait for an answer but must be followed by two other writes to port 322H. The first is a command byte holding the starting address of the erase operation. The second specifies the number of words to be erased.
- 4 Command 23 waits for another command to port 322H before giving an answer. The second command specifies the address where the read operation must be performed. When this byte arrives, the word read is sent back (least significant byte first, then the most significant byte).
- 5 Command 24 does not wait for an answer but must be followed by other three writes to port 322H. The first two bytes following the command must contain the value you want written to the EEPROM (first the LSB byte, then the MSB byte), while the third byte must contain the EEPROM address where you wish the value to be written.

The following example shows how to read an analog channel:

```

;read analog channel 0
analog0      proc    near
              call   waitIBF          ;GPS controller ready?
              jc     an0exit
              mov    al,#07           ;out cmd
              mov    dx,322h
              out    dx,al
              call   waitIBF          ;command understood?
              jc     an0exit
              call   readcount        ;get word answer
              jc     an0exit
              mov    cl,6
              shr    ax,cl
              mov    word ptr es:[di],ax
              clc
an0exit:
              ret
analog0      endp                    ;scope: read word from GPS controller
readcount    proc    near
              call   waitOBF          ;wait for answer
              jc     readcounterxit   ;timeout?
              mov    dx,320h          ;no

```

```

        in     al,dx                ;read info
        mov   ah,al
        call  waitOBF
        jc   readcountexit
        mov   dx,320h              ;get answer
        in   al,dx                 ;read msb info
        xchg  al,ah
readcountexit:
        ret
readcount  endp
    
```

323H Port

This is the port where bytes to be transmitted are written. The following is an example of its use:

```

;scope:      tx one byte
;input       AL=byte to be transmitted
puttxbyte    proc  near
              push  ax
              push  cx
              push  dx
              push  bx
              call  waitIBF          ;microcontroller ready?
              jc   putexit          ;no
doputcmd:    mov   dx,323h          ;yes transmit byte
              out   dx,al
putexit:     pop   bx
              pop   dx
              pop   cx
              pop   ax
              ret
puttxbyte    endp
    
```

327H Port

This port allows you to substitute an EPROM bank for that normally used by the microcontroller. This may be done only once before a reset command. The module startup procedure is shown below; it must be strictly followed for proper operation.

```

resetmodule  proc  near
              mov   dx,327h          ;set Eprom bank on GPS module
              mov   al,1
              out   dx,al
              mov   dx,326h          ;clear IBF
              mov   al,0
              out   dx,al
              mov   cx,0
              loop  $
              mov   dx,326h          ;start microcontroller
              mov   al,1
              out   dx,al
              loop  $
              mov   dx,320h          ;clear OBF
              in   al,dx
              ret
resetmodule  endp
    
```

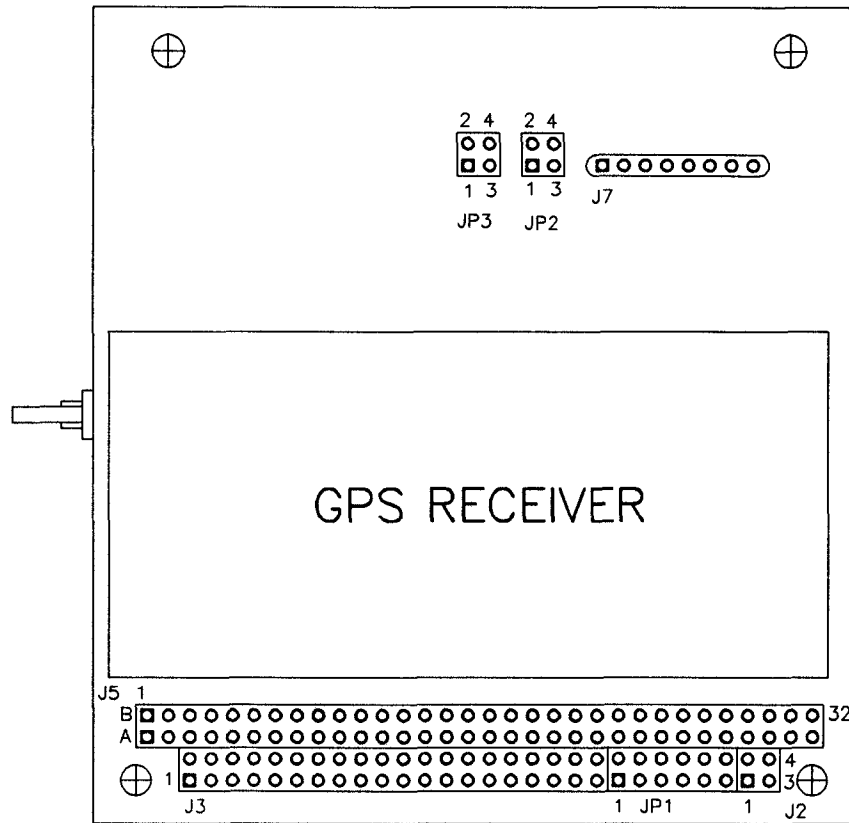
Chapter 5 **HARDWARE REFERENCE**

This chapter gives information on the utilityModule hardware, including:

- component locations
- jumper locations and settings
- solder jumper locations and settings
- mechanical dimensions

Component Locations

The following figure shows the locations of the connectors and major components of the IGPS104 utilityModule. To use the diagram, position the module with the PC/104 bus connector at the six o'clock position and the component side facing up.



Jumper Settings

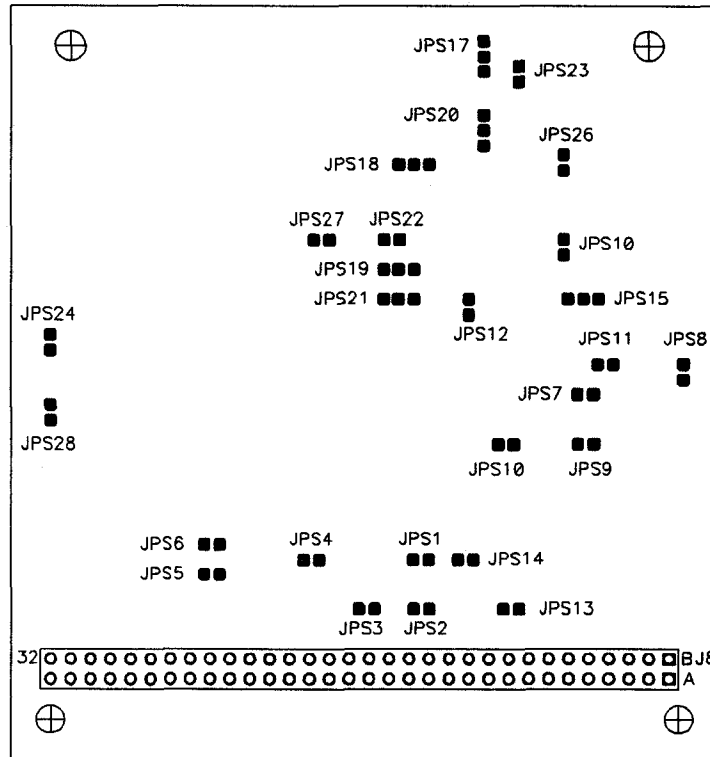
The following tables summarize the settings for the configuration jumpers. For complete information on the jumpers, please refer to the chapter on Configuring the Hardware.

Locations of these jumpers are shown in the preceding figure.

Table 5-1 Jumper Summary		
Jumper	Purpose	Default
JP1	selects hardware interrupt for module	pins 7-8, IRQ5
JP2	selects serial port interrupt	pins 1-2, IRQ4
JP3	selects serial port I/O address and port	pins 1-2, 3-4, COM1 at 3F8

Solder Jumper Locations

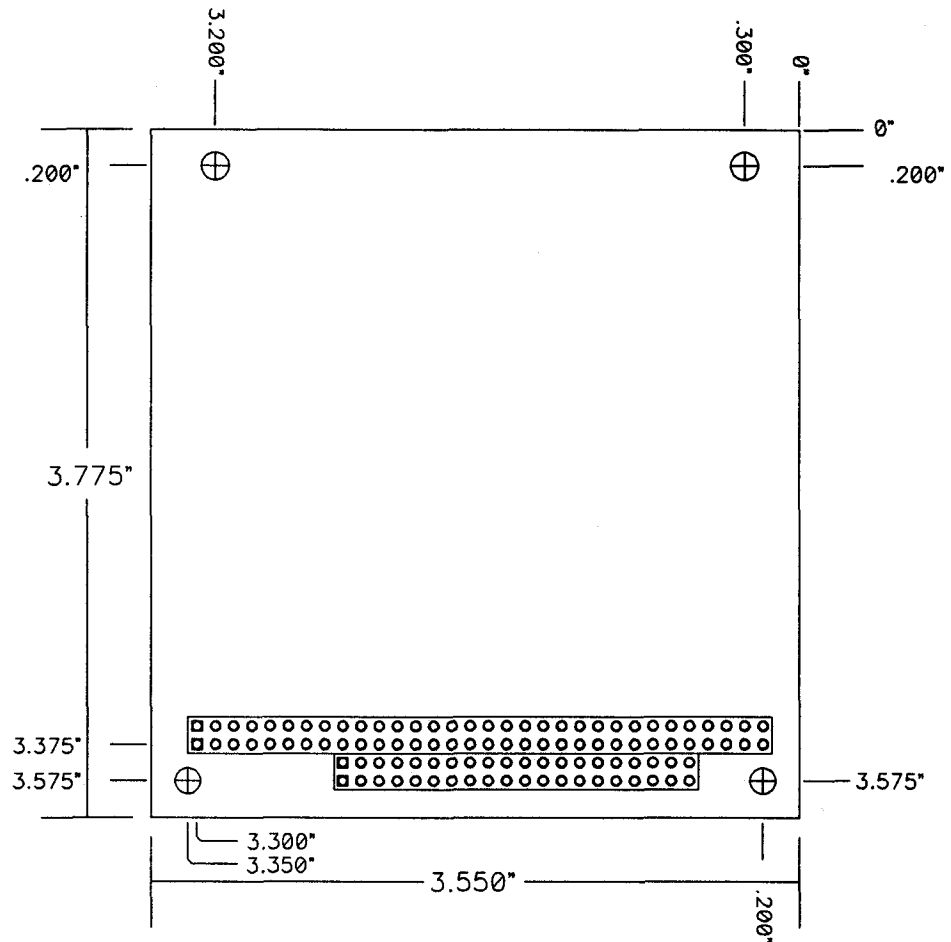
To reconfigure the analog input channel amplifiers or digital outputs, you may need to change solder jumpers on the utilityModule. All solder jumpers are located on the module's solder side. The figure below shows the approximate locations of the solder jumpers. To use the diagram, position the module with the PC/104 bus connector at the six o'clock position and the solder side facing up.



Solder Jumper	Purpose	Default
JPS1	pulls-up collector of output 0 optocoupler	closed
JPS2	grounds emitter of output 0 optocoupler	closed
JPS3	pulls-up collector of output 1 optocoupler	closed
JPS4	grounds emitter of output 1 optocoupler	closed
JPS5	pulls-up collector of output 2 optocoupler	closed
JPS6	grounds emitter of output 2 optocoupler	closed
JPS7, 8	select range of analog input 2	
JPS9, 10	select range of analog input 3	
JPS11, 12	select range of analog input 0	
JPS13, 14	select range of analog input 1	
JPS15	selects bipolar or unipolar A-D converter operation	pins 1-2, bipolar

Mechanical Dimensions

The following figure shows the mechanical dimensions of the utilityModule. All dimensions are in inches (+/- 0.005").



Appendix A Limited Warranty

Real Time Devices, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from REAL TIME DEVICES. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, REAL TIME DEVICES will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to REAL TIME DEVICES. All replaced parts and products become the property of REAL TIME DEVICES. Before returning any product for repair, customers are required to contact the factory for an RMA number.

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED AS A RESULT OF ACCIDENT, MISUSE, ABUSE (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by REAL TIME DEVICES, "acts of God" or other contingencies beyond the control of REAL TIME DEVICES), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN REAL TIME DEVICES. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES ARE EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND REAL TIME DEVICES EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE DURATION OF THIS WARRANTY. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, THE PURCHASER'S SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL REAL TIME DEVICES BE LIABLE TO THE PURCHASER OR ANY USER FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

