GPRS6055ER / GPRS6055RER
User's Manual

GSM//GPRS Modem PC/104 Module

(Real Time Devices)
RTD Embedded Technologies, Inc.
"Accessing the Analog World"®

BDM-610020058
Rev. A

GPRS6055ER / GPRS6055RER

User's Manual

RTD EMBEDDED TECHNOLOGIES, INC.
103 Innovation Blvd
State College, PA 16803-0906

Phone: +1-814-234-8087
FAX: +1-814-234-5218

E-mail
sales@rtd.com
techsupport@rtd.com

Web Site
http://www.rtd.com

Manual Revision History

Rev A	New manual

Published by:

RTD Embedded Technologies, Inc.
103 Innovation Boulevard
State College, PA 16803

Table of Contents

Introduction

Product Overview

The GPRS6055 / GPRS6055R are designed to provide tri-band GSM and GPRS for PC/104 based systems. Included on the GPRS6055 is a Siemens MC55 tri-band GSM cellular modem and on the GPRS6055R is a Triorail TRM:2 tri-band GSM cellular modem for railroad applications.

The GPRS6055 has a UART chip that permits communication with the serial port on the GSM over the PC/104 bus without using other serial ports in the PC/104 system. The GSM module supports an enhanced AT command set.

Board Features

GPRS6055ER Features

- Direct connections to the Siemens MC55/TRM:2 GSM module
- Tri-band GSM module
    - GSM 900/1800/1900 MHz (MC55)
    - GSM GSM-R/900/1800 MHz (TRM:2)
    - GPRS Multislot Class 10
    - AT command set
    - SMS
    - Fax
- 16 digital I/O
- PC/104 compliant

GSM Receiver

The GPRS6055 wireless GPRS/GSM modem unit provides a direct and reliable GPRS connection to GPRS/GSM 900/1800/1900MHz mobile fields around the world. GPRS/GSM connectivity is achieved using the Siemens MC55. This unit works in the 900/1800/1900MHz bands.

Connect any standard quad-band GSM antenna directly to the MCX connector of the GPRS6055. The antenna should be connected to the MC55 using a flexible 50-Ohm cable. In IDAN installations the antenna connection is brought to the front side of the IDAN frame.

A SIM card socket is located on the solder side of the module. The SIM card can only be removed when the MC55 has been placed in shutdown mode.

I/O Interfaces

The GPRS6055ER can be controlled and monitored by software through the a dedicated serial port of the module.

16C550 Compatible UART

The GSM receiver module communicates through a dedicated UART channel allowing other serial ports in the system to be free for the user.

Connector Description

The GSM antenna interfaces are female MCX type miniature coaxial connectors. Connect your antenna directly to the GPRS6055ER antenna connector, or use a short cable inside your enclosure to connect to a feed through connector to allow connection of the antenna to the wall of your enclosure.

All other I/O connections to the GPRS6055ER use 0.1” header type terminals.

Available Options

The GPRS6055ER is available as a starter kit, bundled with an active antenna. It may also be purchased as an IDAN module for integration into an RTD IDAN system.

The following is a summary of the different GPRS6055ER configurations:

Part Number	**Description**
GPRS6055ER	GPRS6055ER with Siemens MC55 GSM module
SK-GPRS6055ER	GPRS6055ER with GSM antenna
IDAN-GPRS6055ERS	GPRS6055ER mounted in an IDAN frame
IDAN-SK-GPRS6055ERS	GPRS6055ER mounted in an IDAN frame with GSM antenna
GPRS6055RER	GPRS6055RER with Triorail TRM:2 GSM module
SK-GPRS6055RER	GPRS6055RER with GSM antenna
IDAN-GPRS6055RERS	GPRS6055RER mounted in an IDAN frame
IDAN-SK-GPRS6055RERS	GPRS6055RER mounted in an IDAN frame with GSM antenna

For antenna specifications, please refer to the “Additional Information” chapter of this manual.

Getting Technical Support

If you are having problems with your system, please try the following troubleshooting steps:

- **Simplify the System** – Remove modules one at a time from your system to see if there is a specific module that is causing a problem.
- **Swap Components** – Try replacing parts in the system one-at-a-time with similar parts to determine if a part is faulty or if a type of part is configured incorrectly.

If problems persist, or you have questions about configuring this product, obtain the PCI BIOS listing information of the GPRS6055ER and other modules in the system. After you have this information, contact RTD Embedded Technologies via the following methods:

Phone: +1-814-234-8087

E-Mail: techsupport@rtd.com

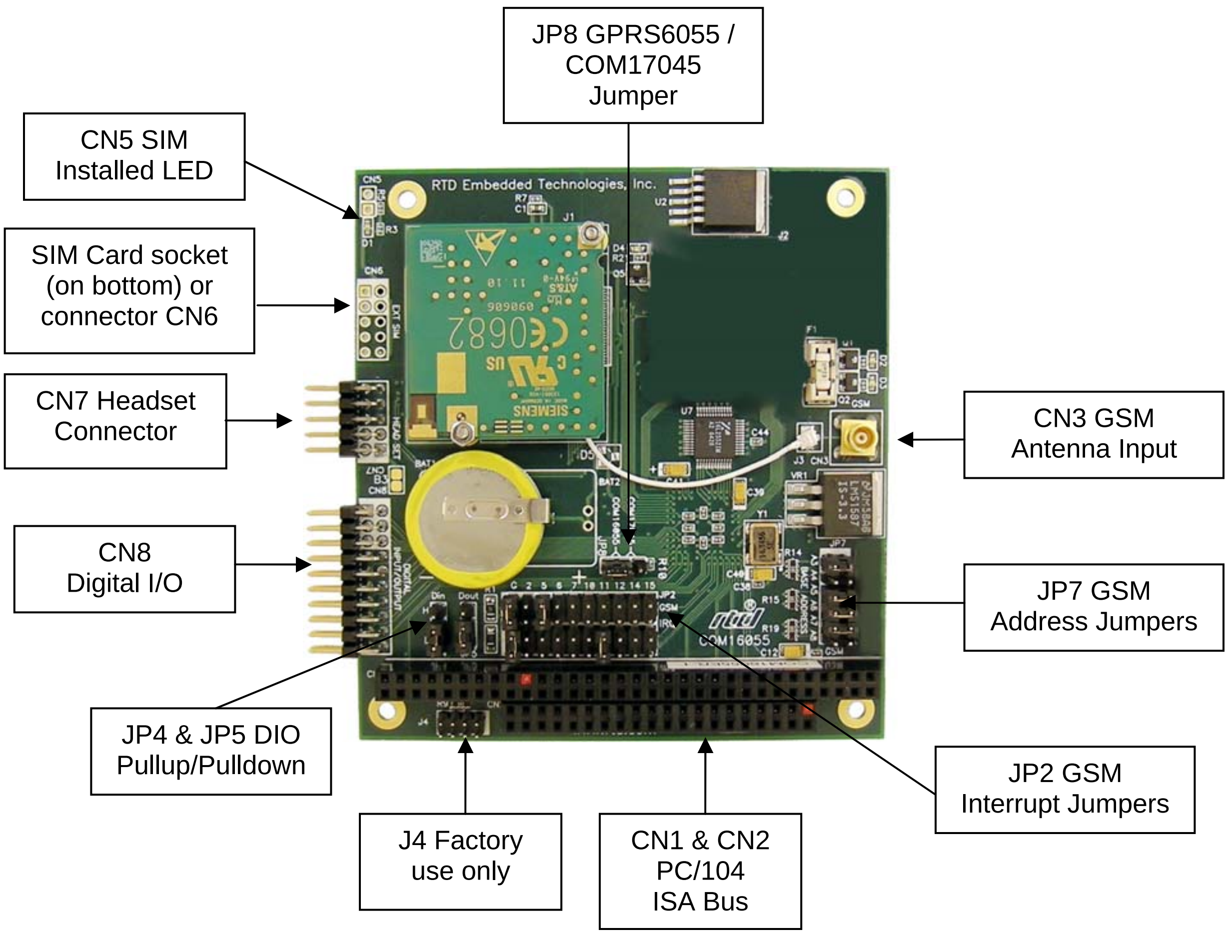
Be sure to check the RTD web site (http://www.rtd.com) frequently for product updates, including newer versions of the board manual and application software.

Board Connections

Connector and Jumper Locations

The following diagram shows the location of all connectors and jumpers on the GPRS6055ER. Future revisions of the GPRS6055ER may have cosmetic differences. For a description of each jumper and connector, refer to the following sections.

GPRS6055ER Connector and Jumper Locations

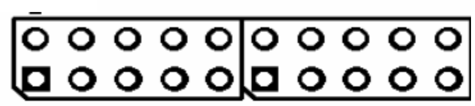


External I/O Connections

The following sections describe the external I/O connections of the GPRS6055ER.

CN7 – Digital Input/Output Connector

The GPRS6055 offers 16 bit-programmable digital I/O lines. These can be pulled high or low through 10K Ohm resistors using JP4 to control bits 0 – 7 and JP5 to control bits 8 - 15.

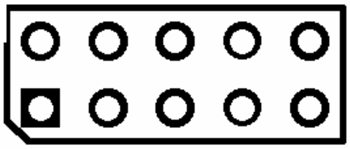


Pin	Name	CN7 Description GPRS6055 Mode
1	GND	Ground
2	DIO0	Digital Input/Output Bit 0
3	DIO1	Digital Input/Output Bit 1
4	DIO2	Digital Input/Output Bit 2
5	DIO3	Digital Input/Output Bit 3
6	DIO4	Digital Input/Output Bit 4
7	DIO5	Digital Input/Output Bit 5
8	DIO6	Digital Input/Output Bit 6
9	DIO7	Digital Input/Output Bit 7
10	+5 VDC	+5 Volts DC
11	GND	Ground
12	DIO8	Digital Input/Output Bit 8
13	DIO9	Digital Input/Output Bit 9
14	DIO10	Digital Input/Output Bit 10
15	DIO11	Digital Input/Output Bit 11
16	DIO12	Digital Input/Output Bit 12
17	DIO13	Digital Input/Output Bit 13
18	DIO14	Digital Input/Output Bit 14
19	DIO15	Digital Input/Output Bit 15
20	+5 VDC	+5 Volts DC

Pin	Name	CN7 Description COM17045 Mode
1	GND	Ground
2	DIO0	Digital Output Bit 0
3	DIO1	Digital Output Bit 1
4	DIO2	Digital Output Bit 2
5	DIO3	Digital Output Bit 3
6	DIO4	Digital Output Bit 4
7	DIO5	Digital Output Bit 5
8	DIO6	Digital Output Bit 6
9	DIO7	Digital Output Bit 7
10	+5 VDC	+5 Volts DC
11	GND	Ground
12	DIO8	Digital Input Bit 0
13	DIO9	Digital Input Bit 1
14	DIO10	Digital Input Bit 2
15	DIO11	Digital Input Bit 3
16	DIO12	Digital Input Bit 4
17	DIO13	Digital Input Bit 5
18	DIO14	Digital Input Bit 6
19	DIO15	Digital Input Bit 7
20	+5 VDC	+5 Volts DC

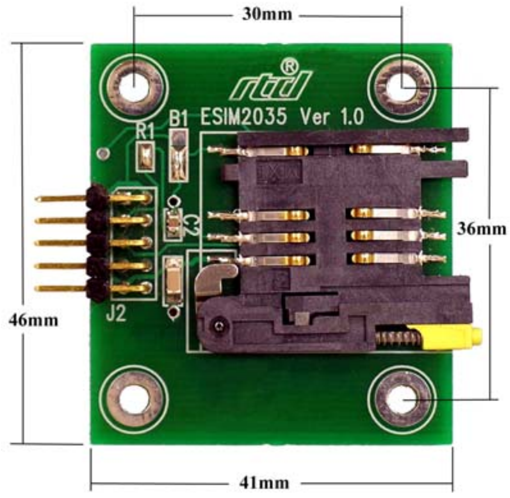
CN6 – SIM Module/Connector

The GPRS6055ER can use either an onboard SIM module or an external SIM connected through a cable. The GPRS6055ER-1 uses an on-board SIM module and will not have CN6 installed. The GPRS6055ER-2 uses an external SIM module. The pin out of the external connector CN6 is shown below.



Pin	Name	CN6 Description
1	GND	Ground
2	Vcc	SIM Power
3	RST	SIM Reset
4	GND	Ground
5	IO	SIM I/O Data
6	GND	Ground
7	CLK	SIM Clock
8	IN	SIM Card Detect
9	Vcc	SIM Power
10	GND	Ground

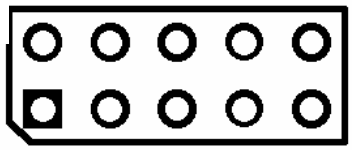
The figure below shows a picture of the external SIM card interface board.



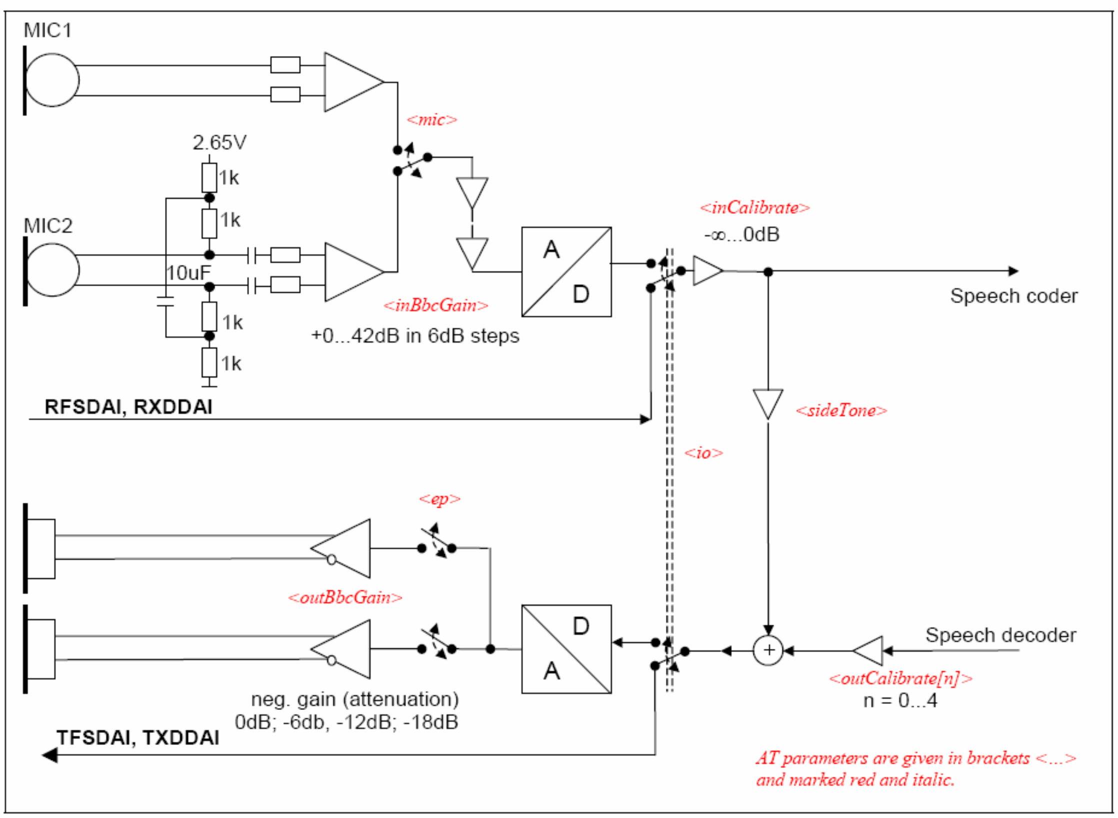
ESIM2035 board

CN7 – GSM Headset Connector

The GPRS6055ER Headset connector is used to connect a headset to the GSM module for voice operation. The pin out of the external connector CN7 is shown below.



Pin	Name	CN7 Description
1	GND	Ground
2	MICP1	Microphone 1+
3	MICN1	Microphone 1-
4	EPP1	Earphone 1+
5	EPN1	Earphone 1-
6	MICP2	Microphone 2+
7	MICN2	Microphone 2-
8	EPP2	Earphone 2+
9	EPN2	Earphone 2-
10	GND	Ground



Microphone interface 1 is high impedance (~50k Ohm) and shall be used preferably if an opamp or a CODEC is connected or additional microphone feeding is needed.

Microphone interface 2 is high impedance (~1k Ohm) and shall be used preferably if internal microphone feeding can be used, e.g. for an internal microphone.

Both EP1 and EP2 are the same.

Jumpers

The following sections describe the jumper configuration options available on the GPRS6055ER. For a reference that shows the location of each set of jumpers, refer to the diagram of the GPRS6055ER at the beginning of this chapter. The default factory jumper settings are listed in the following table:

Jumper	Description	Default Factory Setting
JP1	Reserved	Reserved
JP2	GSM Interrupt Jumper	Default Interrupt 5 and G
JP3	Reserved	Reserved
JP4	Pull-up or Pull-down for DIO0-7	1-2 – Pull-up 2-3 – Pull-down (default) No connect – Neither
JP5	Pull-up or Pull-down for DIO8-15	1-2 – Pull-up 2-3 – Pull-down (default) No connect – Neither
JP6	Reserved	Reserved
JP7	GSM Base Address Jumper (A8 – A3, A9 is high)	Default 2E8h
JP8	GPRS6055/COM17045 Mode	1-2 – COM17045 Compatible 2-3 – GPRS6055 (default)

LED Indicators

D1 – On when the SIM card is enabled.

D2 – Reserved

D3 – Reserved

D4 – On when GSM is transmitting

Note: The GSM AT^SSYNC command serves to configure the SYNC pin of the application interface. The pin can either be used to indicate the current consumption in a transmit burst (default setting) or to drive a status LED connected to the pin. See the AT^SSYNC command for details.

Board Installation

Installing the Hardware

The GPRS6055ER can be installed into a PC/104. It can be located almost anywhere in the stack, above or below the CPU as long as all PC/104 bus constraints are met.

Static Precautions

Keep your board in its antistatic bag until you are ready to install it into your system! When removing it from the bag, hold the board at the edges, and do not touch the components or connectors. Handle the board in an antistatic environment, and use a grounded workbench for testing and handling of your hardware.

Steps for Installing

1. Shut down the PC/104 system and unplug the power cord.
2. Ground yourself with an anti-static strap.
3. Line up the pins of the GPRS6055ER's PC/104 connector with the PC/104 bus of the stack and gently press the board onto the stack. The board should slide into the matching PC/104 connector easily. Do not attempt to force the board, as this can lead to bent/broken pins.
4. Attach the external antenna to the MCX connector.
5. If any boards are to be stacked above the GPRS6055ER, install them.
6. Attach any necessary cables to the PC/104 stack.
7. Re-connect the power cord and apply power to the stack.
8. Apply power to the system, and verify that all of the hardware is working properly. Once power is applied, the GSM module will automatically initialize.

Configuring Software

Refer the software user's manual for software configuration and operation.

Hardware Description

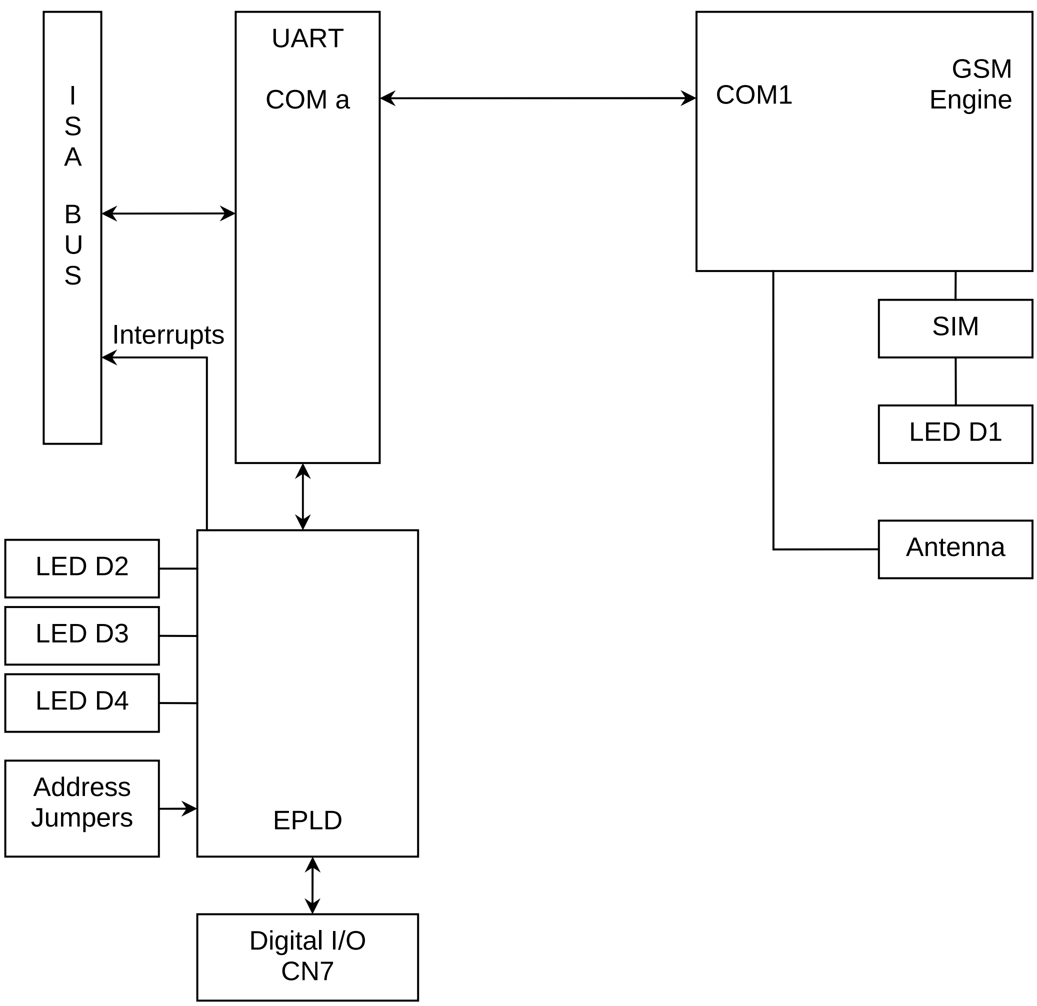
Overview

This chapter describes the major hardware building blocks of the GPRS6055ER. The components discussed in this chapter include:

- Siemens MC55 or Triorail TRM:2 GSM Module
- Antenna

Block Diagram

Below is a block diagram of the GPRS6055ER.



The GPRS GSM wireless modem module

The GPRS6055 wireless E-GPRS/GSM modem is built around the Siemens MC55 tri-band 900/1800/1900 MHz GPRS cellular engine. It is designed both for handling complex industrial applications such as telemetry, telematics or communication, and for integration in stationary or mobile fields all over the world. General information on this product is available at www.siemens.com.

The GPRS6055 is capable of powerful communication using GSM data interfacing. GPRS data rates can reach up to 85.6 kbit/s max downlink depending on the network capacity and the network load. Engineers must take into account that GPRS data rates will vary dynamically depending on network conditions. This must be considered when designing software and system specifications. Roaming agreements and GPRS data support may not in all cases be available when moving from country to country. Check with your local network provider for GPRS coverage.

The GPRS6055ER is capable of FAX and standard SMS text messages. The data terminal rate is 9600 baud for all host commands (AT commands). The GPRS6055ER modem module antenna interface connector uses an MCX connector. The mating antenna connectors and cables are supplied by RTD.

GPRS/GSM wireless modem module interfaces

The GPRS6055 GPRS/GSM modem is connected to the host computer through a dedicated ISA serial port. The default configuration for the GSM serial ports is:

GSM Interface

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

GSM Antenna considerations

Typically standard GSM antennas use a female FME connector. This connector needs an adapter unit before it can be connected to the GPRS6055ER.

RTD recommends the use of high quality antennas with the GPRS6055ER. We have tested successfully with antennas from Hirschmann Rheinmetall Elektronik.

Visit http://www.hirschmann.de/ for information on GSM antennae.

A very useful AT command that shows quality of the signal reception is: AT+CSQ. The format of the response is AT+CSQ: received signal strength, bit error rate. The received signal strength shows the quality of the network signal and ranges from 0 to 31 as shown in the table below. A value of greater than 10 should give an acceptable connection. The bit error rate number will range between 0 and 7.

Received Signal Strength Values (99 = undetectable signal)							
0	- 113 dBm	8	- 97 dBm	16	- 81 dBm	24	- 65 dBm
1	- 111 dBm	9	- 95 dBm	17	- 79 dBm	25	- 63 dBm

2	- 109 dBm	10	- 93 dBm	18	- 77 dBm	26	- 61 dBm
3	- 107 dBm	11	- 91 dBm	19	- 75 dBm	27	- 59 dBm
4	- 105 dBm	21	- 89 dBm	20	- 73 dBm	28	- 57 dBm
5	- 103 dBm	13	- 87 dBm	21	- 71 dBm	29	- 55 dBm
6	- 101 dBm	14	- 85 dBm	22	- 69 dBm	30	- 53 dBm
7	- 99 dBm	15	- 83 dBm	23	- 67 dBm	31	=> - 51 dBm

SIM-card reader

Standard 3V and dual voltage SIM-cards can be used with the GPRS6055. Older 5V SIM cards will not work, though they may operate in standard GSM cellular phones. The SIM-card holder has a card detection circuit that will in theory allow hot insertion and removal of the card. This is **NOT** recommended, since the SIM card contents can become corrupted if it is removed while the MC55/56 GSM modem is writing to it.

LED D1 will turn on when a SIM card is enabled.

A very useful AT command that shows detection of the SIM card is: **AT^SCID**. The SIM card identifier is given as a reply **^SCID: value** shows the ID of the SIM card. If no ID is detected the MC55/56 can not read the SIM card and can not connect to the GSM service provider network.

To add an entry to your SIM card you may use the AT+CPBW command. In this example we add the RTD phone number +1-814-234-8087 to the SIM card memory location "1" with the following AT command set:

AT+CPBW=1, 18142348087, 145, RTD

AT+CREG? Will indicate if the GPRS6055 is logged into the network. If the reply for example is **+CREG: 0, 1** it means that connection to the home network is valid. A complete AT-instruction set documentation is included in the MC55/56 user's manual.

GPRS6055ER Digital I/O

The GPRS6055ER has 16 bit-programmable digital I/O bits. RTD's driver software exports functions to use the digital I/O and the operation is covered in the Drivers Users Manual. JP4 controls a 10K Ohm pull up/down on DIO bits 1-8 and JP5 controls a 10K Ohm pull up/down on DIO bits 9 -16. For programming information see GPRS6055 Module Programming section.

Interrupts

Interrupts are used to notify the host CPU that an event happened on a particular device. In general, interrupts are more efficient than a polling technique, where the CPU must query the device status at regular intervals. Devices that use interrupts have a special connection to the CPU, called an *interrupt request line* (IRQ). When the device needs the CPUs attention, it asserts the IRQ line. Once the interrupt has been processed, the IRQ line is de-asserted.

The GPRS6055ER uses one ISA interrupt for the GSM. However, it will not actually generate interrupts unless the Interrupt Enable register has been properly programmed.

Since the GPRS6055ER has 16C550 UARTs, it supports all of the standard serial port interrupt events. These events include:

- Received data available
- Transmit buffer empty
- Line Status Register change
- Modem Status Register change

A detailed explanation of serial port interrupts is beyond the scope of this manual. For more information, consult a serial port programming reference. The chapter titled “Additional Information” lists some resources to help the user.

Note: When the UART clock is running at a higher frequency, transmit/receive interrupts will happen more frequently. Many operating systems can not process interrupts quickly enough to handle this load. When developing your software, be sure to consider the operating system’s limitations.

GPRS6055 Module Programming

This chapter shows you how to program and use your GPRS6055. It provides a general description of the I/O map. Detailed serial port programming tips are not within the scope of this manual.

Defining the Memory Map

The memory map of the GPRS6055 occupies a group of eight bytes of host PC I/O space. This window is freely selectable by the user by jumpers JP7 for the GSM BASE. After setting the base address you have access to the internal resources of the GPRS6055 control logic. The board also has 16 digital I/O lines that can be accessed several ways.

Digital I/O

The 16 digital I/O bits have different functionality in COM17045 and GPRS6055 modes. COM17045 mode is designed to be software compatible with the RTD COM17045 GSM/GPS module which is 8 outputs and 8 inputs. GPRS6055 mode, which is the default, allows bit programmable direction for all bits. Note the digital I/O registers can be accessed at 400h and 404h – 407h above the GSM.

COM17045 Mode (JP8 = COM17045)			
ADDR (hex)	**REGISTER**	**DIR**	**COMMENTS**
GSM BASE + 400h	Digital I/O	I/O	Digital I/O Port

GPRS6055 Mode (JP8 = GPRS6055 -- Default)			
ADDR (hex)	**REGISTER**	**DIR**	**COMMENTS**
GSM BASE + 400h	Digital I/O	I/O	Digital I/O Port
GSM BASE + 404h	Digital I/O	I/O	Digital I/O bits 0 - 7
GSM BASE + 405h	Digital I/O	I/O	Digital I/O bits 8 – 15
GSM BASE + 406h	Digital I/O Dir	I/O	Digital I/O direction bits 0 - 7
GSM BASE + 407h	Digital I/O Dir	I/O	Digital I/O direction bits 0 - 7

Digital I/O – R/W at BASE + 400h

This register is intended for COM17045 compatibility. If you are using the board in GPRS6055 mode, please use registers at BASE + 404h – 407h.

This address is used to interface to the digital I/O port of the GPRS6055, writing to this address will transfer the data out of the output port, while reading from this address will return the data from the digital inputs. The digital I/O is accessed at GSM BASE + 400h. This register is compatible with the COM17045 module, but works in either COM17045 or GPRS6055 mode. Note the default direction of the 16 digital I/O bits is that bits 0 - 7 are outputs and 8 - 15 are inputs. If you are in GPRS6055 mode and you change the default direction registers, then this register will not operate properly.

Digital I/O – Write GSM BASE + 400h GPRS6055 or COM17045 Modes		
Bit 0	Output 0	CN8 Pin 2
Bit 1	Output 1	CN8 Pin 3
Bit 2	Output 2	CN8 Pin 4
Bit 3	Output 3	CN8 Pin 5
Bit 4	Output 4	CN8 Pin 6
Bit 5	Output 5	CN8 Pin 7
Bit 6	Output 6	CN8 Pin 8
Bit 7	Output 7	CN8 Pin 9

Digital I/O – Read GSM BASE + 400h GPRS6055 or COM17045 Modes		
Bit 0	Input 0	CN8 Pin 12
Bit 1	Input 1	CN8 Pin 13
Bit 2	Input 2	CN8 Pin 14
Bit 3	Input 3	CN8 Pin 15
Bit 4	Input 4	CN8 Pin 16
Bit 5	Input 5	CN8 Pin 17
Bit 6	Input 6	CN8 Pin 18
Bit 7	Input 7	CN8 Pin 19

Advanced Digital I/O – R/W at GSM BASE + 400h

These addresses are used to interface to the digital I/O port in GPRS6055 mode. These are the same digital I/O as above, but with enhanced capability. The 16 bits each have a direction bit. If the direction bit is set to output, a value written to the data bit is provided on the connector. A read will result in the value on the connector pin (i.e. the output value). If the direction is set to input, a value written to the data bit is ignored and a read will result in the value on the connector pin.

Digital I/O Data – Write/Read GSM BASE + 404h GPRS6055 Mode Only (Reset = 00h)		
Bit 0	I/O 0	CN8 Pin 2
Bit 1	I/O 1	CN8 Pin 3
Bit 2	I/O 2	CN8 Pin 4
Bit 3	I/O 3	CN8 Pin 5
Bit 4	I/O 4	CN8 Pin 6
Bit 5	I/O 5	CN8 Pin 7
Bit 6	I/O 6	CN8 Pin 8
Bit 7	I/O 7	CN8 Pin 9

Digital I/O Data – Write/Read GSM BASE + 405h GPRS6055 Mode Only (Reset = 00h)		
Bit 0	I/O 8	CN8 Pin 12
Bit 1	I/O 9	CN8 Pin 13
Bit 2	I/O 10	CN8 Pin 14
Bit 3	I/O 11	CN8 Pin 15
Bit 4	I/O 12	CN8 Pin 16
Bit 5	I/O 13	CN8 Pin 17
Bit 6	I/O 14	CN8 Pin 18
Bit 7	I/O 15	CN8 Pin 19

Digital I/O Direction – Write/Read GSM BASE + 406h GPRS6055 Mode Only (Reset = 00h)		
Bit 0	I/O 0	0 = CN8 Pin 2 is an input, 1 = output
Bit 1	I/O 1	0 = CN8 Pin 3 is an input, 1 = output
Bit 2	I/O 2	0 = CN8 Pin 4 is an input, 1 = output
Bit 3	I/O 3	0 = CN8 Pin 5 is an input, 1 = output
Bit 4	I/O 4	0 = CN8 Pin 6 is an input, 1 = output
Bit 5	I/O 5	0 = CN8 Pin 7 is an input, 1 = output
Bit 6	I/O 6	0 = CN8 Pin 8 is an input, 1 = output
Bit 7	I/O 7	0 = CN8 Pin 9 is an input, 1 = output

Digital I/O Direction – Write/Read GSM BASE + 407h GPRS6055 Mode Only (Reset = FFh)		
Bit 0	I/O 0	0 = CN8 Pin 12 is an input, 1 = output
Bit 1	I/O 1	0 = CN8 Pin 13 is an input, 1 = output
Bit 2	I/O 2	0 = CN8 Pin 14 is an input, 1 = output
Bit 3	I/O 3	0 = CN8 Pin 15 is an input, 1 = output
Bit 4	I/O 4	0 = CN8 Pin 16 is an input, 1 = output
Bit 5	I/O 5	0 = CN8 Pin 17 is an input, 1 = output
Bit 6	I/O 6	0 = CN8 Pin 18 is an input, 1 = output
Bit 7	I/O 7	0 = CN8 Pin 19 is an input, 1 = output

GSM I/O Space

ADDR (hex)	REGISTER	DIR	COMMENTS
GSM BASE + 0	TXD	O	Standard 16C550 UART Registers
	RXD	I	
	BAUD div. Low	I/O	
GSM BASE + 1	BAUD div. High	I/O	
	IRQ enable	I/O	
GSM BASE + 2	IRQ ID	I/O	
GSM BASE + 3	Line control	I/O	
GSM BASE + 4	Modem control	I/O	
GSM BASE + 5	Line status	I/O	
GSM BASE + 6	Modem status	I/O	
GSM BASE + 400h	Digital I/O	I/O	Digital I/O Port
GSM BASE + 401h	Reserved		Reserved
GSM BASE + 402h	GSM status	I/O	Configuration registers
GSM BASE + 403h	GSM control	I/O	Power control
GSM BASE + 404h	Digital I/O	I/O	Digital I/O bits 0 - 7
GSM BASE + 405h	Digital I/O	I/O	Digital I/O bits 8 – 15
GSM BASE + 406h	Digital I/O Dir	I/O	Digital I/O direction bits 0 - 7
GSM BASE + 407h	Digital I/O Dir	I/O	Digital I/O direction bits 0 - 7

I/O map of the GPRS6055 GSM UART

GSM I/O UART (GSM BASE + 0 to GSM BASE + 7)

This is the UART registers for the GSM module. These resources are not described in detail, since they are mapped as a standard PC serial port. For more details on the EXAR 16C550 UART chip programming please download the component specific data from the website: http://www.exar.com.

GSM Status – R/W at GSM BASE + 402h (00h after reset)

GSM Status – Write GSM BASE + 402h		
Bit 0	RESERVED	
Bit 1	/EN_INT	UART interrupt enabled; 1 - disabled
Bit 2	RESERVED	
Bit 3	RESERVED	
Bit 4	RESERVED	
Bit 5	RESERVED	
Bit 6	RESERVED	
Bit 7	RESERVED	

GSM Status – Read GSM BASE + 402h		
Bit 0	/EN_RST	1 – Reset GSM in reset, 0 – GSM active
Bit 1	/EN_INT	GSM interrupt enabled; 1 – disabled
Bit 2	TEMP_LOW	1 - Board temperature below -20C
Bit 3	TEMP_HIGH	1 - Board temperature over +70C
Bit 4	RESERVED	
Bit 5	RESERVED	
Bit 6	RESERVED	
Bit 7	RESERVED	

GSM Control – R/W at GSM BASE + 403h (00h after reset)

GSM Control – Write GSM BASE + 403h		
Bit 0	Ignition signal	1 – > 0 – results in IGN signal
Bit 1	Powerdown MC55	1 – Power down; 0 – Power on
Bit 2	RESERVED	
Bit 3	RESERVED	
Bit 4	RESERVED	
Bit 5	RESERVED	
Bit 6	RESERVED	
Bit 7	RESERVED	

GSM Control – Read GSM BASE + 403h		
Bit 0	Ignition signal	State of Ignition Bit
Bit 1	Powerdown MC55	1 – Power down; 0 – Power on
Bit 2	RESERVED	
Bit 3	RESERVED	
Bit 4	RESERVED	
Bit 5	RESERVED	
Bit 6	RESERVED	
Bit 7	RESERVED	

RTD ISA ID

ADDR (hex)	REGISTER	DIR	COMMENTS
BASE + 800h	RTD ID Data	Read	Read next RTD ID Character
BASE + 801h	RTD ID Data	Read	Read next RTD ID Character
BASE + 802h	Reset RTD ID	Read	Reset RTD ID counter

I/O map of the GPRS6055 RTD ISA ID

BA + 800h/BA + 801h RTD ID Data (read only, 8-bit or 16-bit)

RTD ID is a method to identify a board on the ISA bus. There are two 8-bit registers mapped at **BA + 800h** and **BA + 801h**. The registers can be read as two 8-bit or one 16-bit. An internal pointer is auto-incremented with every read to either address so the data read will step through each index as indicated below. The pointer is set to zero at reset and can be reset to zero by a read to **BA + 802h**.

BA + 402h RTD ID Reset Pointer (read only, 8-bit only)

A read to **BA + 802h** will set the internal pointer to zero. The pointer is set to zero at reset.

RTD ID Data Read Indexes

Index	Data	8-Bit Read	16-Bit Read
0	Device ID	56h	6056h
1	Device ID	60h	—
2	RTD Vendor ID	35h	1435h
3	RTD Vendor ID	14h	—
4	EPLD Revision	Revision LSD	Revision
5	EPLD Revision	Revision MSD	—
6–9	Reserved	Ignore	Ignore
10	Board Name String	G	GP
11	Board Name String	P	—
12	Board Name String	R	RS
13	Board Name String	S	—
14	Board Name String	6	60
15	Board Name String	0	—
16		5	55
17		5	—
18	Board Name String	<nul>	<nul><nul>
19	Board Name String	<nul>	—
20–255	Unused	FFh	FFFFh

Starting up and logging into the GSM network

With no power applied insert your +3V or dual voltage SIM into the card- holder on the solder side of the board. Connect the antenna cable to the MC55 antenna connector and power up your PC/104 system. The GPRS6055 will by initialize with the system. After this the status LED will blink for a while until the MC55 is logged into the network. If you have the PIN code enabled, the GPRS6055 status LED will continue to blink until the PIN code is given through the terminal mode with AT command **AT+CPIN"XXXX",** unless **AT^SFLC** (facility lock for PIN code) has been set. Once the GPRS6055 is logged onto the network the LED will be lit continuously.

INTERRUPTS

What is an interrupt?

An interrupt is an event that causes the processor in your computer to temporarily halt its current process and execute another routine. Upon completion of the new routine, control is returned to the original routine at the point where its execution was interrupted.

Interrupts are a very flexible way of dealing with asynchronous events. Keyboard activity is a good example; your computer cannot predict when you might press a key and it would be a waste of processor time to do nothing whilst waiting for a keystroke to occur. Thus the interrupt scheme is used and the processor proceeds with other tasks. When a keystroke finally occurs, the keyboard then 'interrupts' the processor so that it can get the keyboard data. It then places it into the memory, and then returns to what it was doing before the interrupt occurred. Other common devices that use interrupts are A/D boards, network boards, other used serial ports etc.

Interrupt request lines

To allow different peripheral devices to generate interrupts on the same computer, the PC AT bus has interrupt request channels (IRQ's). A rising edge transition on one of these lines will be latched into the interrupt controller. The interrupt controller checks to see if the interrupts are to be acknowledged from that IRQ and, if another interrupt is being processed, it decides if the new request should supersede the one in progress or if it has to wait until the one in progress has been completed. The priority level of the interrupt is determined by the number of the IRQ as follows; IRQ0 has the highest priority whilst IRQ15 has the lowest. Many of the IRQ's are already used by the standard system resources, IRQ0 is dedicated to the internal timer, IRQ1 is dedicated to the keyboard input, IRQ3 for the serial port COM2, and IRQ4 for the serial port COM1. Often interrupts 2,5,7,10,11 and 15 are free for the user.

8259 Programmable Interrupt Controller

The chip responsible for handling interrupt requests in a PC is the 8259 Interrupt Controller. To use interrupts you will need to know how to read and set the 8259's internal interrupt mask register (IMR) and how to send the end-of-interrupt (EOI) command to acknowledge the 8259 interrupt controller.

Interrupt Mask Register (IMR)

Each bit in the interrupt mask register (IMR) contains the mask status of the interrupt line. If a bit is set (equal to 1), then the corresponding IRQ is masked, and it will not generate an interrupt. If a bit is cleared (equal to 0), then the corresponding IRQ is not masked, and it can then generate an interrupt. The interrupt mask register is programmed through **port 21h**.

End-of-Interrupt (EOI) Command

After an interrupt service routine is complete, the 8259 Interrupt Controller must be acknowledged by writing the value 20h to port 20h.

What exactly happens when an interrupt occurs?

Understanding the sequence of events when an interrupt is triggered is necessary to correctly write interrupt handlers. When an interrupt request line is driven high by a peripheral device (such as the GPRS6055), the interrupt controller checks to see if interrupts are enabled for that IRQ. It then checks to see if other interrupts are active or requested and determines which interrupt has priority. The interrupt controller then interrupts the processor. The current code segment (CS), instruction pointer (IP), and flags are pushed onto the system stack, and a new set if CS and IP are loaded from the lowest 1024 bytes of memory.

This table is referred to as the interrupt vector table and each entry to this table is called an interrupt vector. Once the new CS and IP are loaded from the interrupt vector table, the processor starts to execute code from the new Code Segment (CS) and from the new Instruction Pointer (IP). When the interrupt routine is completed, the old CS and IP are popped from the system stack and the program execution continues from the point where interruption occurred.

Using Interrupts in your Program

Adding interrupt support to your program is not as difficult as it may seem especially when programming under DOS. The following discussion will cover programming under DOS. Note that even the smallest mistake in your interrupt program may cause the computer to hang up and will only restart after a reboot. This can be frustrating and time-consuming.

Writing an Interrupt Service Routine (ISR)

The first step in adding interrupts to your software is to write an interrupt service routine (ISR). This is the routine that will be executed automatically each time an interrupt request occurs for the specified IRQ. An ISR is different from other sub-routines or procedures. First on entrance the processor registers must be pushed onto the stack_before anything else! Second, just before exiting the routine, you must clear the interrupt on the GPRS6055 by writing to the Status register, and write the EOI command to the interrupt controller. Finally, when exiting the interrupt routine the processor registers must be popped from the system stack and you must execute the IRET assembly instruction. This instruction pops the CS, IP and processor flags from the system stack. These were pushed onto the stack when entering the ISR.

Most compilers allow you to identify a function as an interrupt type and will automatically add these instructions to your ISR with one exception: most compilers do not automatically add the EOI command to the function, you must do it yourself. Other than this and a few exceptions discussed below, you can write your ISR as any code routine. It can call other functions and procedures in your program and it can access global data. If you are writing your first ISR, we recommend you stick to the basics; just something that enables you to verify you have entered the ISR and executed it successfully. For example: set a flag in your ISR and in your main program check for the flag.

Note: If you choose to write your ISR in in-line Assembly, you must push and pop registers correctly and exit the routine with the IRET instruction instead of the RET instruction.

There are a few precautions you must consider when writing ISR's. The most important is, **do not use any DOS functions or functions that call DOS functions from an interrupt routine.** DOS is not re-entrant; that is, a DOS function cannot call itself. In typical programming, this will not happen because of the way DOS is written. But what about using interrupts? Consider then the following situation in your program: If DOS function X is being executed when an interrupt occurs and the interrupt routine makes a call to the same DOS function X, then function X is essentially being called while active. Such cases will cause the computer to crash. DOS does not support such operations. The general rule is do not call any functions that use the screen, read keyboard input or any file I/O routines, these should not be used in ISR's.

The same problem of reentrancy also exists for many floating-point emulators. This effectively means that you should also avoid floating point mathematical operations in your ISR.

Note that the problem of reentrancy exists, no matter what programming language you use. Even, if you are writing your ISR in Assembly language, DOS and many floating point emulators are not re-entrant. Of course there are ways to avoid this problem, such as those which activate when your ISR is called. Such solutions are, however, beyond the scope of this manual.

The second major concern when writing ISR's is to make them as short as possible in term of execution time. Spending long times in interrupt service routines may mean that other important interrupts are not serviced. Also, if you spend too long in your ISR, it may be called again before you have exited. This will lead to your computer hanging up and will require a reboot.

Your ISR should have the following structure:

Push any processor registers used in your ISR.

Put the body of your routine here

Clear the interrupt bit by reading GPRS6055 RXD register
Issue the EOI command to the 8259 by writing 20h to 20h
Pop all registers. Most C compilers do this automatically

The following C example shows what the shell of your ISR should be like:

```
/*-----------------------------------------------------------------------------
| Function:     new_IRQ_handler
| Inputs:       Nothing
| Returns:      Nothing
|-----------------------------------------------------------------------------*/
void interrupt far new_IRQ_handler(void)
{
        IRQ_flag = 1;           // Indicate to process interrupt has occurred
        {
                // Your program code to read UART
                // read to a data buffer for example:
                Guc_buffer[Gi_bufpos++] = inp(gi_SERIAL_DATA);
        }
        outp(0x20, 0x20);       // Acknowledge the interrupt controller
}
```

Saving the Startup Interrupt Mask Register (IMR) and interrupt vector

The next step after writing the ISR is to save the startup-state of the interrupt mask register, (IMR) and the original interrupt vector you are using. The IMR is located in address 21h. The interrupt vector you will be using is located in the interrupt vector table which is an array of pointers (addresses) and it is locate din the first 1024 bytes of the memory (Segment 0 offset 0). You can read this value directly, but it is better practice to use DOS function 35h (get interrupt vector) to do this. Most C compilers have a special function available for doing this. The vectors for the hardware interrupts on the XT - bus are vectors 8-15, where IRQ0 uses vector 8 and IRQ7 uses vector 15. Thus if your GPRS6055 is using IRQ5 it corresponds to vector number 13.

Before you install your ISR, temporarily mask out the IRQ you will be using. This prevents the IRQ from requesting an interrupt while you are installing and initializing your ISR. To mask the IRQ, read the current IMR at I/O port 21h, and set the bit that corresponds to the IRQ. The IMR is arranged so that bit 0 is for IRQ0 and bit 7 is for IRQ7. See the paragraph entitled *Interrupt Mask Register (IMR*) earlier in this discussion for help in determining your IRQ's bit. After setting the bit, write the new value to I/O port 21h.

With the startup IMR saved and the interrupts temporarily disabled, you can assign the interrupt vector to point to your ISR. Again you can overwrite the appropriate entry in the vector table with a direct memory write, but this is not recommended. Instead use the DOS function 25h (Set Interrupt Vector) or, if your compiler provides it, the library routine for setting up interrupt vectors. Remember that interrupt vector 8 corresponds to IRQ0, vector 9 for IRQ1 etc.

If you need to program the source of your interrupts, do that next. For example, if you are using transmitted or received messages as an interrupt source program it to do that. Finally, clear the mask bit for your IRQ in the IMR. This will enable your IRQ.

Common Interrupt mistakes

Remember hardware interrupts are from 8-15, XT IRQ's are numbered 0-7. Do not forget to clear the IRQ mask bit in the IMR Forgetting to send the EOI command after ISR code. Disables further interrupts.

Example on Interrupt vector table setup in C-code:

```
void far _interrupt new_IRQ1_handler(void );            /* ISR function */
#define IRQ1_VECTOR 3                                   /* Name for IRQ */
void (interrupt far *old_IRQ1_dispatcher)
        (es,ds,di,si,bp,sp,bx,dx,cx,ax,ip,cs,flags);    /* Variable to store old IRQ_Vector */
void far _interrupt new_IRQ1_handler(void );

/*-------------------------------------------------------------------
| Function:     init_irq_handlers
| Purpose:      Set the pointers in the interrupt table to point to
|               our functions i.e. setup for ISR's.
|-------------------------------------------------------------------*/
void init_irq_handlers(void)
{
        _disable();
        old_IRQ1_handler = _dos_getvect(IRQ1_VECTOR + 8);
        _dos_setvect(IRQ1_VECTOR + 8, new_IRQ1_handler);
        Gi_old_mask = inp(0x21);
        outp(0x21,Gi_old_mask & ~(1 << IRQ1_VECTOR));
        _enable();
}

|/*-------------------------------------------------------------------
| Function:     restore, do this before exiting program
| Purpose:      Restore the interrupt vector table.
|-------------------------------------------------------------------*/
void restore(void)
{
        /* Restore the old vectors */
        _disable();
        _dos_setvect(IRQ1_VECTOR + 8, old_IRQ1_handler);
        outp(0x21,Gi_old_mask);
        _enable();
}
```

GPRS6055ER Specifications

GPRS6055ER Specifications

- PC/104 interface
  - 8-bit, 8.25 MHz (typical)
  - Individual ISA Interrupt for GSM
- UART
  - 1 channel 16C550 with 16 byte FIFOs
  - Oscillator frequency 14.7456 MHz
- Digital I/O
  - 16 bit-programmable in GPRS6055 mode (COM17045 mode is 8 in and 8 out)
  - Jumper selected 10K pull up/down in 8-bit blocks
- Size: 3.6”L x 3.8”W x 0.6”H (90mm L x 96mm W x 15mm H)
- Weight: 0.24bs (0.10 Kg)
- Power Consumption: 2W @ 5 VDC Typical

Siemens MC55 GSM Modem Specifications

General Features:

- Siemens MC55 Tri-Band GSM
  - 900/1800/1900 MHz
- Triorail TRM:2 Tri-Band GSM
  - 900/1800/GSM-R MHz
- Compliant to GSM phase 2/2+
- Output power:
  - class 4 (2 W) for EGSM900
  - class 1 (1 W) for GSM1800
  - class 1 (1 W) for GSM1900
- AT commands Hayes GSM 07.05 and GSM 07.07
- AT commands for RIL compatibility (RIL/NDIS)
- TCP/IP stack access via AT commands
- (SAT Release 99)
- Ambient temperature: –20º C … +70º C
- Restricted Operation: –25º C … +75º C
- Auto switch-off at +75 ºC

Specification for fax:

- Group 3, class 1

Specifications for data (GPRS):

- Multislot class 10: max 85.6 kbit/s (downlink)
- Modulation and coding scheme MCS 1– 4
- Mobile station class B

Specifications for SMS:

- Point-to-point MO and MT
- Text and PDU mode
- SMS cell broadcast

Specifications for voice:

- Basic hands free operation:
  - Echo cancellation
  - Noise reduction

SIM card reader:

- 3.3V cards
- SIM card detection

Antenna Interface

- 50 Ohms Impedance
- MCX straight jack receptacle connector

GPRS6055ER Operating Conditions

Cooling	Convection
Operating temperature	-20º to +70º C
Restricted Operation (Emergency voice calls only)	-25º to +75º C
Humidity	RH up to 95% non-condensing
Storage temperature range	-40^{o} C to $+85^{o}$ C

Limited Warranty

RTD Embedded Technologies, 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 RTD EMBEDDED TECHNOLOGIES, INC. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD EMBEDDED TECHNOLOGIES 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 RTD EMBEDDED TECHNOLOGIES. All replaced parts and products become the property of RTD EMBEDDED TECHNOLOGIES. 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 RTD EMBEDDED TECHNOLOGIES, "acts of God" or other contingencies beyond the control of RTD EMBEDDED TECHNOLOGIES), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN RTD EMBEDDED TECHNOLOGIES. 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 RTD EMBEDDED TECHNOLOGIES EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MECHANTABILITY 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 RTD EMBEDDED TECHNOLOGIES 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.