Contents

1 Manual

1.1 Disclaimer

All trademarks and registered trademarks used in this manual are either trademarks or registered trademarks of the relevant trademark owners. LabView®, LabWindows®, NI-488.2 are either trademarks or registered trademarks of National Instruments Corp. Microsoft® MS-DOS®, Windows®, Windows NT®, Windows XP®, Visual Basic® and Visual C++® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Our continous product improvement makes specifications subject to change without notice.

1.2 Overview of the Manual

This manual covers all topics starting with the installation up to the usage of applications and programming the ines IEEE-488 interface.

Installation

The chapters concerning the installation should be read carefully since they contain useful information about the requirements and the installation preparation.

Using the interface

After the installation has been finished, there are different choices how to continue: One may start the favorite application like HP-VEE to check out the communication. Some useful hints will be found in the application interface section. An introduction into developing applications that control the ines GPIB bus will be found under the Programming Language Interface topic.

Cable length for IEEE-488 bus systems

To fulfill the IEEE-488 standard the maximum length of cable that shall be used to connect together a group of devices within one bus system is (1) 2 m times the number of devices (2) Or 20 m, whichever is less The maximum interconnection distance in a system depends on the transmission speed. If all devices are connected within a 2 m distance and 48 mA line drivers with open collectors are used, the bus works with an maximum transmission speed of 250 kB/s for a distance up to 20 m. Under the same conditions the maximum transmission speed can be increased to 500 kB/s with Tri-State drivers. If all devices are connected within a 1 m distance and 48 mA with Tri-State drivers used, the bus works with an maximum transmission speed of 1 MB/s for a distance up to 10 m.

For two interfaces (one controller and one device) the cable length should not exceed 4m. You can use a cable of greater length and it might work properly but it is not within the IEEE-488 standard and you must check the functionality and the signal edges very carefully.

1.3 Software Overview

The INES GPIB for Windows allows you to connect to the GPIB in many ways. You can choose from a set of programming language interface or use commercial available application development environments. In addition interfaces compatible to other vendors' GPIB interface solutions are available.

1.3.1 New Features of Version 6

While every attempt has been made to ensure compatibility to previous versions of INES GPIB for Windows the following major changes has been made.

Industry Standard Interface

The industry-standard GPIB32.DLL (ibrd, ibwrt,...) style interface is now fully supported on the Microsoft® Windows® 2000 and Windows XP® operating systems. Existing applications run directly without modifications.

Enhanced Device Operation

The software now supports standard conformant implementations of IEEE-488.2 compatible devices (instruments) via a new C++ interface class library.

Up to eight interfaces supported

The count of concurrently operating interface cards per computer has been increased to 8 (eight).

New C++ API

A native and easy to use C++ API has been added to the set of programming languages supported. The native C++ implementation of a GPIB API directly support the object oriented programming paradigm. The API provides classes for GPIB interfaces, devices, transfer parameters etc.

GNU Compiler Collection supported

In addition to the Microsoft Visual C++ compiler, the popular GCC compiler (see www.mingw.org) is supported.

1.3.2 Features continued in Version 6

Programming Language Interfaces

32 bit for C/C++ (Borland C 4.5, Borland C 5.0, Microsoft Visual C++ 2.0, Microsoft Visual C++ 4.0, Microsoft Visual C++ 6.0) Visual Basic (4-6), Visual C++, Delphi(2-5), HT-Basic

Application Development Environments

Agilent VEE, LabView 6+

1.3.3 Features obsoleted by Version 6

Version 5.1 is the last version supporting the 16-bit interface libraries and Windows 3.11, 95, 98, ME, NT4. Also, the WALI interface program is no more supported. The Online Manual media has been changed to HTML and PDF.

1.4 Installation

Installation background

The installation process of INES GPIB for Windows constists of two parts. First, the device drivers for the interface card(s) must be installed. This process is guided by the *Hardware Wizard*. The Hardware Wizard is started whenever a new, unknown devices is mounted, inserted or attached to the computer. After the device driver software has been installed successfully, the generic software components are installed by *setup.exe*.

1.4.1 Installation of Plug&Play cards

Minimum requirements

In order to install INES GPIB for Windows successfully you need

- A Microsoft Windows compatible Personal Computer
- one or more ines GPIB interface card(s) of any of the following type(s)
  - GPIB-PCI (12.002.00)
  - GPIB-PCMCIA
  - GPIB-PCI-XL

– GPIB-PCMCIA-XL (32 bit Cardbus)

Removal of previous installations

If a previous version of INES GPIB for Windows is installed remove it prior to installation by using <distribution-dir-or-drive>\tools\igclreg.exe

Installation procedure

Follow these steps to install a ines Plug&Play card:

1. Always follow the instructions of the computer manufacturer for the installation of add-on cards.
2. Insert the card into a free slot, turn on the computer and start Windows. If you have more than one card to install, insert all cards.
3. Windows recognizes the new hardware. Let the Hardware Wizard search for a driver and choose CD-ROM as source.
4. After the Hardware Wizard has finished, execute *setup.exe* from the installation CD. Follow the instructions of the installation program.

1.4.2 Installation of ISA cards

ISA cards are different to Plug&Play cards. Because the presense of hardware cannot be detected automatically, the card(s) must be added explicitely to the system.

Minimum requirements

In order to install INES GPIB for Windows successfully you need

- A Microsoft Windows compatible Personal Computer with ISA or PC/104 expansion bus
- one or more ines GPIB interface card(s) of any of the following type(s)
  – GPIB-PC104
  – GPIB-PC (NEC 7210 C installed)
  – GPIB-PC-HS (iGPIB 7210 1.1 installed)

Removal of previous installations

If a previous version of INES GPIB for Windows is installed remove it prior to installation by using <distribution-dir-or-drive>\tools\igclreg.exe

Installation procedure

Follow these steps to install a ines ISA card:

1. Carefully configure the IRQ and IO settings of your card to use non-conflicting resources. Unresolved resource conflicts could damage your hardware. Disable all DMA channels. NOTE your settings!
2. Install (all) INES GPIB ISA card(s)
3. Power on again your computer.
4. Select ⇒Windows ⇒Settings ⇒Control Panel ⇒Add/Remove Hardware . The Add/Remove Hardware Wizard appears.
5. Select ⇒Add/Troubleshoot a device, ⇒Next . Windows does search for new Plug&Play hardware.
6. Select ⇒Add a new device ⇒Next.
7. Select ⇒No, I want to select hardware from a list ⇒Next .
8. Select ⇒Other devices .
9. Select ⇒Have Disk , direct Windows to your distribution disk or directory.

10. Select the type of your INES GPIB card
11. Enter the resources assigned in step 1. If you cannot assign a resource because Windows reports it as allocated: Quit the *Add/Remove Hardware Wizard* and see *Troubleshooting Conflicting Resources* below.
12. Select ⇒Yes
13. Select ⇒ Start Hardware Installation
14. Repeat starting with step 5 for each card not yet installed. Otherwise allow the *Add/Remove Hardware Wizard* to reboot your computer.
15. Use the Windows *Device Manager* to verify the installation. Each device installed must appear as properly functioning device.
16. After the Hardware Wizard has finished, execute setup.exe from the installation CD-ROM (or the directory containing the distribution content) and follow the instructions of the installation program. This adds the generic software components to your computer.

Troubleshooting Conflicting Resources

If Windows reports resources as being allocated:

- there might be a real resource conflict with existing hardware. Reconfigure the cards in question and start again.
- there might be remains from previously installation attempts, *icglreg.exe* was not able to remove automatically. In this case open the *Add/Remove Hardware Wizard*, Choose ⇒Deinstall Device and deinstall the devices wrongly reported as conflicting.

1.5 Diagnostics and Support

The INES GPIB for Windows provides a facility to log the interface calls and bus communication of your application. In order to create (or re-initialize) a log file, you must run the *iediag.exe* command line application from the \tools subdirectory of the INES GPIB installation directory. The iediag.exe application creates a log file named *iediag.log* in the root directory of your disk drive *c:* and gathers system information which is written to the first lines of the file.

Whenever a GPIB function library (DLL) is used the first time by your application program it checks for a file named c:\iediag.log to be present. If it is present, interface calls and bus communication are logged by appending lines to that file.

You can use a text editor (wordpad) to view the information in the log file. No further tools are required.

Please note that logging requires computing and IO resources and makes applications using the GPIB interface run slower than they do run with logging disabled. For that reason make sure to remove the log file (or rename it to a name different to iediag.log) when logging is no more required.

When you are contacting technical support, please provide the log file with your request by following the prodedure below.

1. Before starting your application program please run the iediag.exe tool. It is located in the \tools subdirectory of the INES GPIB installation directory.
2. Then, run your application until the problem occurs.
3. After finishing your application program, the file C:\iediag.log should contain logging data. Please attach this file to your request.

1.6 Using Applications

The INES GPIB for Windows allows you to run applications designed for the GPIB-32.DLL interface, and several commercially available application development environments. This sections describes how to use this applications with the INES GPIB for Windows

1.6.1 LabView Version 6

You need to ensure that the system has access to the INES GPIB-32.DLL. Note that this DLL has the same name as other vendor's version. If you have installed such a GPIB library, you must remove or rename the other vendor's GPIB-32.DLL and make sure the INES version of GPIB-32.DLL is in the \Windows\System directory. Access to the INES GPIB-32.DLL will allow you to use the low level Labview GPIB I/O instructions.

Enabling the VISA drivers to recognise the INES GPIB hardware in Labview 6 requires the following procedure. Labview 6.0i uses a configuration program called "Measurement and Automation Explorer 2.0" or "MAX 2.0". You must use this program to install a NON Plug'N'Play type GPIB interface card from the list of National Instrument GPIB cards.

Select the "AT-GPIB/TNT" model. When you choose the AT-GPIB/TNT card you will be brought into a configuration procedure for the card setup. Accept the default choices.

When done, you will see a GPIB interface installed in the configuration tree of the MAX utility. If you highlight this item and then click "Scan for Devices" the program should detect the devices connected to your GPIB bus and they can now be addressed using the VISA instrument drivers.

1.6.2 Agilent/HP VEE

The INES GPIB-32.DLL library can be used with Agilent/HP VEE by following these steps:

1. Ensure your system has access to GPIB-32.DLL - either via the path or by copying the DLL to C:\WINDOWS\SYSTEM\GPIB-32.DLL.
2. Activate Agilent-VEE
3. Using the Agilent-VEE instrument manager, set your device addresses in VEE to a number of the form 14xx where xx is the GPIB address. So, for example, if your analyzer is at GPIB address 5 then set the address to 1405.

1.6.3 Agilent IO Libraries

The INES GPIB-32.DLL library can be used with the Agilent IO Libraries (later than M.01.01) by following these steps:

1. Ensure your system has access to GPIB-32.DLL - either via the path or by copying the DLL to the system directory (e.g. C:\windows\system32\GPIB-32.DLL).
2. In the system directory (e.g. c:\windows\system32) the file iegpibn32.dll must be present. In order to use the INES GPIB-32.DLL with Agilent IO Libraries rename iegpibn32.dll to gpibn32.dll by erasing the first two characters.
3. If not already done, install the Agilent IO Libraries.
4. Start Agilent IOCONFIG
5. From the list Available Interface Types select GPIB using NI-488.2. Then click Configure. Accept or modify the settings. Then click OK.
6. The INES GPIB interface(s) can now be operated using the Agilent IO Libraries.

1.6.4 HTBasic

INES GPIB cards may be used with HTBasic version 7. Our GPIB-32.DLL is interchangeable with the like element from other vendors - and so, for GPIB support with your INES GPIB interface, you can simply use the NI driver that comes with the HT Basic package. Follow these instructions:

1 Install your GPIB card and the GPIB software.
2 Select the installation option which copies GPIB-32.DLL to your system directory.
3 Activate HT Basic.
4 Load the National Instruments GPIB driver (with which the INES driver is compatible) via the command: LOAD BIN "GPIBNI". Alternatively, add line LOAD BIN "GPIBNI" to your AUTOST file so that the GPIB driver is automatically loaded when you start HTBasic.
5 Load ("GET") and run your program.

1.6.5 Applications using the GPIB-32.DLL

You need to ensure that the system has access to the INES GPIB-32.DLL. Note that this DLL has the same name as other vendor's versions. If you have installed an other vendor's GPIB library, you must remove or rename the other vendors GPIB-32.DLL and make sure the INES version of GPIB-32.DLL is in the \Windows\System directory. Access to the INES GPIB-32.DLL will allow you to operate these applications with the INES GPIB interfaces.

1.7 The INES driver library

In order to write programs that control the GPIB bus, there must be an interface to the GPIB hardware. There are drivers for many programming languages so that they can use the ines driver and function library.

Using the ines driver library

The ines IEEE488 interface system supplies the developer with libraries to create windows application which may control the GPIB bus. The interface to applications is designed as dynamic link libraries (DLLs), which is the typical method for Windows. These DLLs contain the GPIB functions like IeEnter(), IeOutput() or IeInpt() as described in the function reference. These functions are independent from the used hardware interface and programming language. The GPIB functions are calling an underlying interface, the device driver. The driver and its implementation depends on the used interface hardware and the Windows Version.

The following description demonstrates the general usage of the GPIB functions while the IEEE-488 commands covers a detailed description of every function. Since the implementation of the GPIB functions is the usual Windows method by using DLLs, it is possible to control the GPIB bus via any application or programming language which allows to call external DLLs. Special care has to be taken for multithreaded application. The GPIB functions don't protect itself from being called concurrently by separate threads in such a environment. The application must ensure that only one GPIB function may be called at a time.

The samples files may be used for a quick introduction into accessing the GPIB bus. A device is programmed to acquire a buffer of sample data and inform the computer via SRQ when the buffer has been filled up. Then the data is read into the computers data buffer and displayed. The sample applications require a Keithley Instrument K195A DMM or compatible at primary GPIB address 9.

1.7.1 Programming in C/C++

The ines IEEE488 interface for C and C++ supplies the developer with import libraries and include files to access the DLL functions. The GPIB functions are simple C functions. For C++, the declaration is nested in an extern "C" declaration automatically in the include file. In order to use the correct data types and values, the file IEEE488.H must be included. This file includes other necessary files like types488.h itself and contains

the necessary function declarations. It is strongly recommended to use the types shown in the command reference to write portable applications. Before including the ines IEEE488.H header file, the WINDOWS.H include file must be included and WINDOWS has to be defined:

All functions return an error code. This error code should be evaluated for non zero, which indicates the occurrence of an error.

On occurrence of an error a messagebox appears with the error message. If the application handles this error by itself, this effect is not desired most times. In this case the function IeMode() may be called to turn off the error messages:

The C/C++ interface is supplied with import library files which must be linked to the application. The library names are IE488F3W.LIB and IE488H3W.LIB for 32-bit applications. The library IE488F3W.LIB is the import library for the GPIB functions and IE488H3W.LIB contain the error messages and some error related function references. For C/C++ compilers that cannot access the library functions via the supplied import libraries this library may be recreated from the DLLs by using a tool (like 'implib') which is supplied with the compiler.

1.7.2 Programming in in Visual Basic

The ines function DLL and help DLL may be accessed from Visual Basic directly. The necessary declarations are found in the IEEE488W.BAS (IEEE488.TXT for Visual Basic 1.0) file. This file has to be included in the Visual Basic project or in the GLOBAL.BAS file. GPIB functions may be called in the usual manner:

All functions return an error code as an integer value. Any nonzero value means the an error has occurred. On occurrence of an error a messagebox appears with the error message. If the application handles the error itself, this effect is not desired most times. In this case the function IeMode() may be called to turn off the error messages:

1.7.3 Programming in Borland Delphi

The ines function and help DLL may be accessed by Delphi directly. The file IEEE488W.PAS has to be included in the project. This file contains the unit 'ieee488w' where all necessary declarations are found to write applications that are controlling the GPIB bus. Delphi includes the ieee488w unit if the corresponding file 'IEEE488W.PAS' if specified in the project. Common data types are numeric values, which must be declared as Word and pointer to data areas, which have to be declared as array [0..nn] of Byte.

If Pascal strings are used, the StrPCopy should be used. The Send function which is part of the Delphi sample uses this conversion:

All functions return an error code as an integer value. Any nonzero value means the an error has occurred. On occurrence of an error a messagebox appears with the error message. If the application handles the error itself, the messagebox is not desired most times. In this case the function IeMode() may be called to turn off the error messages:

1.7.4 Programming in HT Basic

HT Basic is considered an GPIB application because it does not use the INES function library. Please see the HT Basic subsection of the Application Interfaces section.

1.7.5 Function reference

1.7.5.38 UNT

2 Appendix

2.1 Overview of the IEEE-488 GPIB

Introduction

The IEEE-488 or IEC 625 interface (GPIB) is a powerful tool for communication and control of devices used in the fields of measurement, automation, and data analysis. The acceptance of the interface as a worldwide standard had its start when Hewlett Packard created the HP-IB (Hewlett Packard Interface Bus). It allowed the company to equip completely different instruments with one standard interface.

In the following years, the HP-IB has become accepted worldwide without major changes in the form of the standards IEEE-488 and IEC 625. The only difference between IEEE-488 and IEC 625 is in the type of connector used.

The IEEE-488 uses a 24 pin amphenol connector, while the IEC 625 uses a 25 pin connector (called the MIN-D-SUB). The connection to different types can be made easily with the use of an adapter cable.

Fifteen devices can be operated on the IEEE-488 bus in a parallel manner. The management of the bus is handled by the Controller. By using a three line handshake protocol, it is assured that the slowest connected and addressed device determines the data transfer rate of the overall system. The benefit to this three line handshake is extremely reliable data transfer without the use of a separate timing parameter or clock.

The IEEE-488 Bus

The IEEE-488 bus is a communication system of up to 15 devices with a single system controller (usually a PC or similar computer). The interfaces are connected with a 24 pin cable to one another.

Eight of the 24 lines provide byte oriented, paralleled data transfer. The other sixteen lines provide signals for communication control as well as the necessary grounding (GND). Each device that is connected to the bus, as well as the system controller, is identified by a unique address. An address is a primary address only, or a primary address together with a secondary address. Secondary addressing is commonly used to address subdevices of a device. A primary address ranges from 0 to 30 and a secondary address ranges from 0 to 31.

The controller

A device that is capable of sending interface messages is called a controller. There may be only one active controller in a system. The controller which functions as the active controller once the system has been started is called the system controller. Interface messages are defined by the IEEE-488 standard and they are common to every device conforming to IEEE-488.1 or IEEE-488.2. Interface messages are used to setup the GPIB data transfer direction by specifiing a talker and one or more listener(s), reset the interface of a device and a device itself, query for status information and configure an instrument for remote control. A device receives these interface messages, which are identified by the ATN line being active. Interface messages (sometimes called commands) that affect every device connected to the IEEE-488 bus belong to the universal command group and interface messages the affect only the addressed device are called addressed command group.

Talker listener devices

A device which can send device specific messages to the bus after it has been addressed is called a talker. A device which can receive device specific data from the bus is called a listener. Most devices do combine both of these characteristics. To transfer data there must be exactly one talker and at least one listener. Since device specific messages are not specified by the IEEE-488.1 specification these messages vary between the different instruments. The more recent IEEE-488.2 specification uses the SCPI (Standard Commands for Programmable Instruments) language to control instruments. These messages bases on the IEEE-488.1 hardware protocol. For this reason it is possible to control a IEEE-488.2 device by a controller that just conforms to IEEE-488.1.

Transfering data

At the beginning of a communication sequence, the controller sends a talker and a listener address. Then, the ATN (attention) line is set to 'false' and the addressed talker starts with its data transfer. Each of the data bytes is transfered by using the three line handshake which enshures that the talker does not send another data byte before the listener has accepted the actual byte. The data transfer is terminated by a special character the EOS (End Of Sequence; usually a linefeed, ASCII code 10) or by a special control line, the EOI line (End Or Identify). Using the EOI line is commonly used for binary data transfer. After the data has been transfered, the controller sets ATN to 'true' and may initiate the next sequence. The controller can also address itself as a talker/listener which is the most usual case.

2.2 Notes on Specific Hardware

2.2.1 GPIB-PC/104

I/O address configuration

The DIP switch SW 1-2 on the card is used to select the base I/O address of the card. Position 6 is ignored. Set position 1 through 5 according to the following table. Choose an unallocated region of I/O addresses.

GPIB-PC104 address selection

HEX	DEC	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
100	256	on	on	on	off	on
120	288	off	on	on	off	on
140	320	on	off	on	off	on
160	352	off	off	on	off	on
180	384	on	on	off	off	on
1A0	416	off	on	off	off	on
1C0	448	on	off	off	off	on
1E0	480	off	off	off	off	on
200	512	on	on	on	on	off
220	544	off	on	on	on	off
240	576	on	off	on	on	off
260	608	off	off	on	on	off
280	640	on	on	off	on	off
2A0	672	off	on	off	on	off
2C0	704	on	off	off	on	off
2E0	736	off	off	off	on	off
300	768	on	on	on	off	off
320	800	off	on	on	off	off
340	832	on	off	on	off	off
360	864	off	off	on	off	off
380	896	on	on	off	off	off
3A0	928	off	on	off	off	off

HEX	DEC	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5
3C0	960	on	off	off	off	off
3E0	992	off	off	off	off	off

IRQ address configuration

IRQ 5 is preselected. If conflicting with another card change it to an unallocated IRQ channel.

2.2.2 GPIB-PCW, GPIB-HS-NT+, GPIB-AT

I/O address configuration

The I/O address DIP switch on the card is used to select the base I/O address of the card. Position 8 must always be ON. Set position 1 through 7 according to the following table. Choose an unallated region if I/O addresses.

GPIB-PC, GPIB-HS-NT+, GPIB-AT address selection

HEX	DEC	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6	Pos 7
100	256	ON	ON	ON	ON	ON	OFF	ON
108	264	OFF	ON	ON	ON	ON	OFF	ON
110	272	ON	OFF	ON	ON	ON	OFF	ON
118	280	OFF	OFF	ON	ON	ON	OFF	ON
120	288	ON	ON	OFF	ON	ON	OFF	ON
128	296	OFF	ON	OFF	ON	ON	OFF	ON
130	304	ON	OFF	OFF	ON	ON	OFF	ON
138	312	OFF	OFF	OFF	ON	ON	OFF	ON
140	320	ON	ON	ON	OFF	ON	OFF	ON
148	328	OFF	ON	ON	OFF	ON	OFF	ON
150	336	ON	OFF	ON	OFF	ON	OFF	ON
158	344	OFF	OFF	ON	OFF	ON	OFF	ON
160	352	ON	ON	OFF	OFF	ON	OFF	ON
168	360	OFF	ON	OFF	OFF	ON	OFF	ON
170	368	ON	OFF	OFF	OFF	ON	OFF	ON
178	376	OFF	OFF	OFF	OFF	ON	OFF	ON
180	384	ON	ON	ON	ON	OFF	OFF	ON
188	392	OFF	ON	ON	ON	OFF	OFF	ON
190	400	ON	OFF	ON	ON	OFF	OFF	ON
198	408	OFF	OFF	ON	ON	OFF	OFF	ON
1A0	416	ON	ON	OFF	ON	OFF	OFF	ON
1A8	424	OFF	ON	OFF	ON	OFF	OFF	ON
1B0	432	ON	OFF	OFF	ON	OFF	OFF	ON
1B8	440	OFF	OFF	OFF	ON	OFF	OFF	ON

HEX	DEC	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6	Pos 7
1C0	448	ON	ON	ON	OFF	OFF	OFF	ON
1C8	456	OFF	ON	ON	OFF	OFF	OFF	ON
1D0	464	ON	OFF	ON	OFF	OFF	OFF	ON
1D8	472	OFF	OFF	ON	OFF	OFF	OFF	ON
1E0	480	ON	ON	OFF	OFF	OFF	OFF	ON
1E8	488	OFF	ON	OFF	OFF	OFF	OFF	ON
1F0	496	ON	OFF	OFF	OFF	OFF	OFF	ON
1F8	504	OFF	OFF	OFF	OFF	OFF	OFF	ON
200	512	ON	ON	ON	ON	ON	ON	OFF
208	520	OFF	ON	ON	ON	ON	ON	OFF
210	528	ON	OFF	ON	ON	ON	ON	OFF
218	536	OFF	OFF	ON	ON	ON	ON	OFF
220	544	ON	ON	OFF	ON	ON	ON	OFF
228	552	OFF	ON	OFF	ON	ON	ON	OFF
230	560	ON	OFF	OFF	ON	ON	ON	OFF
238	568	OFF	OFF	OFF	ON	ON	ON	OFF
240	576	ON	ON	ON	OF	ON	ON	OFF
248	584	OFF	ON	ON	OFF	ON	ON	OFF
250	592	ON	OFF	ON	OFF	ON	ON	OFF
258	600	OFF	OFF	ON	OFF	ON	ON	OFF
260	608	ON	ON	OFF	OFF	ON	ON	OFF
268	616	OFF	ON	OFF	OFF	ON	ON	OFF
270	624	ON	OFF	OFF	OFF	ON	ON	OFF
278	632	OFF	OFF	OFF	OFF	ON	ON	OFF
280	640	ON	ON	ON	ON	OFF	ON	OFF
288	648	OFF	ON	ON	ON	OFF	ON	OFF
290	656	ON	OFF	ON	ON	OFF	ON	OFF
298	664	OFF	OFF	ON	ON	OFF	ON	OFF
2A0	672	ON	ON	OFF	ON	OFF	ON	OFF
2A8	680	OFF	ON	OFF	ON	OFF	ON	OFF
2B0	688	ON	OFF	OFF	ON	OFF	ON	OFF
2B8	696	OFF	OFF	OFF	ON	OFF	ON	OFF
2C0	704	ON	ON	ON	OFF	OFF	ON	OFF
2C8	712	OFF	ON	ON	OFF	OFF	ON	OFF
2D0	720	ON	OFF	ON	OFF	OFF	ON	OFF
2D8	728	OFF	OFF	ON	OFF	OFF	ON	OFF

HEX	DEC	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6	Pos 7
2E0	736	ON	ON	OFF	OFF	OFF	ON	OFF
2E8	744	ON	ON	OFF	OFF	OFF	ON	OFF
2F0	752	ON	OFF	OFF	OFF	OFF	ON	OFF
2F8	760	OFF	OFF	OFF	OFF	OFF	ON	OFF
300	768	ON	ON	ON	ON	ON	OFF	OFF
308	776	OFF	ON	ON	ON	ON	OFF	OFF
310	784	ON	OFF	ON	ON	ON	OFF	OFF
318	792	OFF	OFF	ON	ON	ON	OFF	OFF
320	800	ON	ON	OFF	ON	ON	OFF	OFF
328	808	OFF	ON	OFF	ON	ON	OFF	OFF
330	816	ON	OFF	OFF	ON	ON	OFF	OFF
338	824	OFF	OFF	OFF	ON	ON	OFF	OFF
340	832	ON	ON	ON	OFF	ON	OFF	OFF
348	840	OFF	ON	ON	OFF	ON	OFF	OFF
350	848	ON	OFF	ON	OFF	ON	OFF	OFF
358	856	OFF	OFF	ON	OFF	ON	OFF	OFF
360	864	ON	ON	OFF	OFF	ON	OFF	OFF
368	872	OFF	ON	OFF	OFF	ON	OFF	OFF
370	880	ON	OFF	OFF	OFF	ON	OFF	OFF
378	888	OFF	OFF	OFF	OFF	ON	OFF	OFF
380	896	ON	ON	ON	ON	OFF	OFF	OFF
388	904	OFF	ON	ON	ON	OFF	OFF	OFF
390	912	ON	OF	ON	ON	OFF	OFF	OFF
398	920	OFF	OFF	ON	ON	OFF	OFF	OFF
3A0	928	ON	ON	OFF	ON	OFF	OFF	OFF
3A8	936	OFF	ON	OFF	ON	OFF	OFF	OFF
3B0	944	ON	OFF	OFF	ON	OFF	OFF	OFF
3B8	952	OFF	OFF	OFF	ON	OFF	OFF	OFF
3C0	960	ON	ON	ON	OFF	OFF	OFF	OFF
3C8	968	OFF	ON	ON	OFF	OFF	OFF	OFF
3D0	976	ON	OFF	ON	OFF	OFF	OFF	OFF
3D8	984	OFF	OFF	ON	OFF	OFF	OFF	OFF
3E0	992	ON	ON	OFF	OFF	OFF	OFF	OFF
3E8	1000	OFF	ON	OFF	OFF	OFF	OFF	OFF
3F0	1008	ON	OFF	OFF	OFF	OFF	OFF	OFF
3F8	1016	OFF	OFF	OFF	OFF	OFF	OFF	OFF

IRQ address configuration

IRQ 5 is preselected. If conflicting with another card change it to an unallocated IRQ channel.