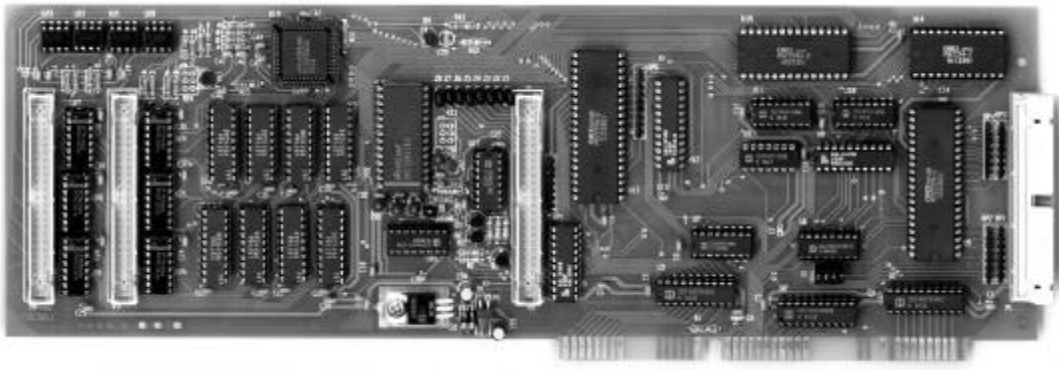
Servo To Go, Inc.

ISA Bus Servo I/O Card
Hardware Manual



Motion Control for 2, 4, 6, or 8 Servo Motors,
Plus 32 Bits of Digital I/O, 8 Channels of Analog Input and a Timer/Interrupt Generator

Servo To Go, Inc.

Phone: 317-257-1655

Fax: 317-251-3958

E-mail: servotogo@msn.com

Internet world wide web URL: http://www.servotogo.com

8117 Groton Ln. Indianapolis, IN 46260-2821, USA

Revision V, December 1996

Table of Contents

1.0 Summary

The hardware described in this manual is a low cost, general purpose, motion control input/output board which can control up to eight motors simultaneously from an ISA-bus based computer such as an IBM compatible PC. The following is a summary of the hardware functionality:

- **Encoder Input**
  - Up to 8 channels of encoder input
  - A, B, and I (sometimes called 'marker') input
  - 24 bit counters
  - Single-ended or differential (RS422 compatible) input signals
  - The index pulse input can be configured for normally high or normally low.
  - Up to 10 MHz input rate
- **Analog Output**
  - Up to 8 channels of analog output
  - + 10 V to - 10 V span.
  - 13 bit resolution
  - Sign bit digital output for each channel (Opto-22 compatible)
- **Digital Input and Output**
  - 32 bits, configurable in various input and output combinations
  - Opto-22 compatible
- **Analog Input**
  - 8 channels of analog input
  - 13 bit resolution
  - Configurable as +/-10V or +/-5V spans.
- **Interval Timers**
  - Capable of interrupting the PC
  - Timer interval is programmable to 10 minutes in 25 microsecond increments
- **Battery Backup Input**
  - Used to maintain encoder counting capability in event of a power failure.
- **Board Address Detection with IRQ software selectable**
  - Used to determine the board base address automatically without a configuration file.
  - IRQ number is software selectable - no board jumper required.
- **Watchdog Timer**

Sample applications include:

- Robotics
- Machine tools
- Motion picture camera control
- Specialty machine control
- Controls design education
- Automated test equipment
- Medical instrumentation
- Virtual reality “rides”

Although the board is typically used to perform servo motor control, it can also be used for specialized I/O. For example, in encoder position monitoring or any other application where encoder input as well as analog and digital I/O are required.

The board is simply and efficiently accessed by the use of a set of registers located in the I/O space of the PC. Connection to the outside is accomplished through four 50-pin connectors.

2.0 Setup and Installation

This section describes the configuration of the board via a small set of user selectable jumpers. This section describes each of the five jumpers, as well as the battery back up connector, which are listed below.

Feature Name	Description
Jumper J1	Base address selection
Jumper J2	ADC input range selection
Jumper J3	DAC latch on interrupt or on direct write selection
Jumper J4	Watchdog timer output selection
Jumper J5	Watchdog timer enable selection
Connector P5	Optional battery backup connection

Please see Appendix C - Component Locations, for the position of the jumpers on the board.

2.1 Base Address - Jumper J1

Insert jumpers to select the corresponding base address of the board. Inserting a jumper selects match on low, removing a jumper selects match on high. For example, if the SA5 jumper is “in”, then A5 will have to be low in order to have a “match” and select the card.

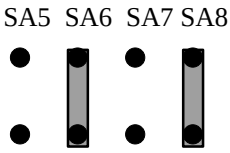
SA8	SA7	SA6	SA5	Addresses Selected		
IN	IN	IN	IN	200-1F	&	600-1F
IN	IN	IN	OUT	220-3F	&	620-3F
IN	IN	OUT	IN	240-5F	&	640-5F
IN	IN	OUT	OUT	260-7F	&	660-7F
IN	OUT	IN	IN	280-9F	&	680-9F
IN	OUT	IN	OUT	2A0-BF	&	6A0-BF
IN	OUT	OUT	IN	2C0-DF	&	6C0-DF
IN	OUT	OUT	OUT	2E0-FF	&	6E0-FF
OUT	IN	IN	IN	300-1F	&	700-1F
OUT	IN	IN	OUT	320-3F	&	720-3F
OUT	IN	OUT	IN	340-5F	&	740-5F
OUT	IN	OUT	OUT	360-7F	&	760-7F
OUT	OUT	IN	IN	380-9F	&	780-9F
OUT	OUT	IN	OUT	3A0-BF	&	7A0-BF
OUT	OUT	OUT	IN	3C0-DF	&	7C0-DF
OUT	OUT	OUT	OUT	3E0-FF	&	7E0-FF

The J1 jumper configuration is software accessible and will appear in the “BRDTST” register.

Established I/O address for many common devices appear in appendix B.

Base address selection example:

SA8	SA7	SA6	SA5	Address Selected
OUT	IN	OUT	IN	340-5F & 740-5F



2.2 Analog Input Range - Jumper J2

Insert a single appropriate jumper, marked on the board, to select the input range of the analog-to-digital converter. J2 is a three pin jumper, the jumper should be on pins 1 and 2 or pins 2 and 3. The possible input voltage ranges are -10 to +10V, and -5 to +5V. The input resolution of the ADC will be 13 bits for these ranges. Obviously, the resolution would be 12 bits for 0-5V and 0-10V spans.

2.3 DAC Latch on interrupt - Jumper J3

The board can be configured to generate periodic interrupts to the PC. This is necessary when the board is used in a control algorithm application. Each sample period, the control algorithm needs to read the encoders and write to the DACs. The encoders are automatically latched when the periodic interrupt occur. Using this jumper, the user can select weather or not the DACs are also simultaneously latched at the instant of the period interrupt. If both the encoders and DACs are latched by the hardware, the control algorithm can be performed at anytime during the window. With no change in performance from one sample period to the next. If the jumper is on the right-hand side, the DACs will not be latched by the interrupt, but will be latched when they are written to. If the jumper is on the left, the DACs’ output will change when the periodic interrupt occurs (when timer counter 0 times out, please see section 5.3.3 for more information).

2.4 Watchdog timer output selection - Jumper J4

The board contains a timer so that a "watchdog" will be constantly watching writes to the card. If no writes occur within a specified time, the watchdog will time-out and "bark", thus signifying that something has gone wrong, for example, the processor may not be responding to interrupts as it should. When the watchdog times-out the output of the DACs are set to zero. If this jumper is in, pin 17 of connector P2 (also connected to bit DIO.31) will be high until the board is again written to. This event can be used to trip a relay in order to take some external action such as applying breaks to the machine that is being controlled. Since the watchdog output is also connected to DIO.31, the I/O port DIO.24-31 must be configured as input port in order to read-in the state of the watchdog and not interfere with the operation of pin 17 of connector P2. If the jumper is out, no additional event will occur (besides zeroing of the DACs) and port DIO24-31 can be configured as either an input or as an output port and pin 17 of P2 can be used normally.

2.5 Watchdog timer enable - Jumper J5

This jumper determines if the watchdog is enabled or not. If the jumper is on the right hand position, the watchdog timer is enabled and will be reset by any write to any DAC. If a write does not occur before the watchdog times out, then a "bark" event occurs. If the jumper is on the left, the watchdog will be constantly reset by an oscillator and will consequently never time-out, effectively disabling its usefulness, therefore, it will never zero the DACs. Also, if J4 is out, DIO31 can be used by the user.

The following table lists the watchdog's time-out times:

Minimum	Typical	Maximum
62.5 milliseconds	150 milliseconds	250 milliseconds

If J5 is on the right, and at least one DAC is written to at least every 62mS, the watchdog will not time out. If J5 is on the right, and writes to the DAC stop, the watchdog will time out in at most 250mS, and typically in less than 150mS. If J5 is on the left, the watchdog will never time out.

2.6 Battery Backup Connection - Connector P5

The board contains an optional battery connection on connector P5. In the event that power should be lost to the card, the encoder counters will not loose their position count and will continue to operate if a battery is connected to this connector. With battery backed-up encoder counters, the controlled machine would not need to be re-calibrated. The supply voltage must be between 7 and 15 volts. The ground connection is on the right side of the connector, with the positive voltage on the left. The battery backed-up portion of the board requires 150 milliamps of current. Of course, a battery or some other uninterruptable power supply should also supply the encoders in order for the counter circuit to have valid quadrature input signals.

3.0 Connections

Connection to the outside world is accomplished through four 50-pin connectors and one 2-pin connector. The following tables lists the pin-out for each of the 50-pin connectors. Connection can be made with standard .1 inch spacing, 50-pin, ribbon cable headers.

Connector Name	Location*	Pin Count	Description
P1	Furthest right, has a bracket.	50	24 bits of digital I/O (Ports A, B, and C) which are Opto-22 compatible.
P2	2nd from right	50	8 bits of user I/O (Port D), 8 motor direction bits, for a total of 16 Opto-22 compatible bits, and 8 channels of analog input.
P3	3rd from right	50	Encoder input and analog output for Axis 0-3
P4	Left-most	50	Encoder input and analog output for Axis 4-7
P5	near bottom edge	2	Battery input, right pin is ground, left pin is +V.

* Note: These locations are from the perspective of having the board's component side up, with the ISA bus connector at the lower right. Please refer to Appendix C. Pin 1 of each of the 50-pin connectors is in the upper left corner of the connectors.

8 Channel Encoder Input and Analog Output Connectors

Connector P3, Motion I/O Axis 0-3			
Pin	**Name**	**Pin**	**Name**
1	Analog Gnd	**2**	DAC 0
3	Analog Gnd	**4**	Analog Gnd
5	DAC 2	**6**	Analog Gnd
7	Analog Gnd	**8**	DAC 1
9	Analog Gnd	**10**	Analog Gnd
11	DAC 3	**12**	Analog Gnd
13	Gnd	**14**	A 0 +
15	A 0 -	**16**	Gnd
17	B 0 +	**18**	B 0 -
19	Gnd	**20**	I 0 +
21	I 0 -	**22**	Gnd
23	A 1 +	**24**	A 1 -
25	Gnd	**26**	B 1 +
27	B 1 -	**28**	Gnd
29	I 1 +	**30**	I 1 -
31	Gnd	**32**	A 2 +
33	A 2 -	**34**	Gnd
35	B 2 +	**36**	B 2 -
37	Gnd	**38**	I 2 +
39	I 2 -	**40**	Gnd
41	A 3 +	**42**	A 3 -
43	Gnd	**44**	B 3 +
45	B 3 -	**46**	Gnd
47	I 3 +	**48**	I 3 -
49	NC	**50**	NC

Connector P4, Motion I/O Axis 4-7			
Pin	**Name**	**Pin**	**Name**
1	Analog Gnd	**2**	DAC 4
3	Analog Gnd	**4**	Analog Gnd
5	DAC 6	**6**	Analog Gnd
7	Analog Gnd	**8**	DAC 5
9	Analog Gnd	**10**	Analog Gnd
11	DAC 7	**12**	Analog Gnd
13	Gnd	**14**	A 4 +
15	A 4 -	**16**	Gnd
17	B 4 +	**18**	B 4 -
19	Gnd	**20**	I 4 +
21	I 4 -	**22**	Gnd
23	A 5 +	**24**	A 5 -
25	Gnd	**26**	B 5 +
27	B 5 -	**28**	Gnd
29	I 5 +	**30**	I 5 -
31	Gnd	**32**	A 6 +
33	A 6 -	**34**	Gnd
35	B 6 +	**36**	B 6 -
37	Gnd	**38**	I 6 +
39	I 6 -	**40**	Gnd
41	A 7 +	**42**	A 7 -
43	Gnd	**44**	B 7 +
45	B 7 -	**46**	Gnd
47	I 7 +	**48**	I 7 -
49	NC	**50**	NC

32 Bit Digital I/O, 8 Channel Analog Input and Sign Bit Output Connectors

Connector P1, Digital I/O			
Pin	**Name**	**Pin**	**Name**
1	Opto-23, C7	**2**	Gnd
3	Opto-22, C6	**4**	Gnd
5	Opto-21, C5	**6**	Gnd
7	Opto-20, C4	**8**	Gnd
9	Opto-19, C3	**10**	Gnd
11	Opto-18, C2	**12**	Gnd
13	Opto-17, C1	**14**	Gnd
15	Opto-16, C0	**16**	Gnd
17	Opto-15, B7	**18**	Gnd
19	Opto-14, B6	**20**	Gnd
21	Opto-13, B5	**22**	Gnd
23	Opto-12, B4	**24**	Gnd
25	Opto-11, B3	**26**	Gnd
27	Opto-10, B2	**28**	Gnd
29	Opto-9, B1	**30**	Gnd
31	Opto-8, B0	**32**	Gnd
33	Opto-7, A7	**34**	Gnd
35	Opto-6, A6	**36**	Gnd
37	Opto-5, A5	**38**	Gnd
39	Opto-4, A4	**40**	Gnd
41	Opto-3 A3	**42**	Gnd
43	Opto-2, A2	**44**	Gnd
45	Opto-1, A1	**46**	Gnd
47	Opto-0, A0	**48**	Gnd
49	+5V	**50**	Gnd

Connector P2, Analog & Digital I/O			
Pin	**Name**	**Pin**	**Name**
1	ADC Chan 0	**2**	Analog Gnd
3	ADC Chan 1	**4**	Analog Gnd
5	ADC Chan 2	**6**	Analog Gnd
7	ADC Chan 3	**8**	Analog Gnd
9	ADC Chan 4	**10**	Analog Gnd
11	ADC Chan 5	**12**	Analog Gnd
13	ADC Chan 6	**14**	Analog Gnd
15	ADC Chan 7	**16**	Analog Gnd
17	Opto-15, D7	**18**	Gnd
19	Opto-14, D6	**20**	Gnd
21	Opto-13, D5	**22**	Gnd
23	Opto-12, D4	**24**	Gnd
25	Opto-11, D3	**26**	Gnd
27	Opto-10, D2	**28**	Gnd
29	Opto-9, D1	**30**	Gnd
31	Opto-8, D0	**32**	Gnd
33	Opto-7, S7	**34**	Gnd
35	Opto-6, S6	**36**	Gnd
37	Opto-5, S5	**38**	Gnd
39	Opto-4, S4	**40**	Gnd
41	Opto-3, S3	**42**	Gnd
43	Opto-2, S2	**44**	Gnd
45	Opto-1, S1	**46**	Gnd
47	Opto-0, S0	**48**	Gnd
49	+5V	**50**	Gnd

Where, for example:

Opto-23, C7	is	the I/O Port C Bit 7.
DAC 1	is	the digital-to-analog converter channel 1.
A 1 +	is	the encoder “A” signal for channel 1. For differential input it is the more positive “A” signal, and for single-ended input it is the only “A” signal.
A 1 -	is	the more negative differential input for encoder channel “A”. For single-ended mode signals, this pin must be left unconnected.
ADC Chan 7	is	the analog-to-digital converter channel 7
Analog Gnd	is	analog ground
Opto-0, S0	is	a digital output corresponding to the sign bit of the analog output.
Opto-9, D1	is	a digital input or output for Port D bit 1.
Gnd	is	digital ground.
NC	is	not connected.
+5	is	5 volt power.

Notes on using +5: Keep your current usage to less than 500 ma. total for the card. If your current requirements exceed this, use a separate supply. Often, you will be using this power for one side of optical isolation; don't defeat the purpose by connecting grounds together.

4.0 Register Address Definitions

The functionality of the board can be accessed through the set of registers shown in the table on the next page. Note that some of the registers have different meanings depending on whether a read or a write operation is being performed.

The addresses are divided into two groups, a low 32 byte page when the address bit 10 is not set, and a high 32 byte page when the A10 bit is set. A large address set allows faster processing of data to and from the card because extra write operations are not required to specify which data is to be accessed. With a small address space an indirect-access scheme would need to be used; with the larger space, data is accessed directly.

The following integrated circuits, or their equivalents, are used to implement some of the functionality of the board:

Function	IC Designation	Manufacturer	Description
Digital I/O	82C55	NEC	Programmable peripheral interface
Timers	82C54	Intel	Programmable interval timer
Interrupt Control	82C59	Intel	Programmable interrupt controller
Analog input	ADC1241 or 12441	National	13-bit ADC
Analog output	Max547	Maxim	Octal, 13-bit DAC
Encoder Input	LS7166, 26LS32	LSI Logic, TI	24 bit, 1.2 MHz counter, RS-422 receiver

Register Address Offset Definition Table

Offset	A10 = 0 Read	A10 = 0 Write	A10 = 1 Read	A10 = 1 Write
0x00	CNT0.D	CNT0.D	DIO.0-7	DIO.0-7
0x01	CNT1.D	CNT1.D	DIO.24-31	DIO.24-31
0x02	CNT0.C	CNT0.C	DIO.8-15	DIO.8-15
0x03	CNT1.C	CNT1.C	BRDTST	---
0x04	CNT2.D	CNT2.D	DIO.16-23	DIO.16-23
0x05	CNT3.D	CNT3.D	INT.C	INT.C
0x06	CNT2.C	CNT2.C	---	MIO.0-23
0x07	CNT3.C	CNT3.C	ODDRST	MIO.24-31
0x08	CNT4.D	CNT4.D	TIMER.0	TIMER.0
0x09	CNT5.D	CNT5.D	IRR, ISR	OCW2-3, ICW1
0x0A	CNT4.C	CNT4.C	TIMER.1	TIMER.1
0x0B	CNT5.C	CNT5.C	IMR	OCW1, ICW2
0x0C	CNT6.D	CNT6.D	TIMER.2	TIMER.2
0x0D	CNT7.D	CNT7.D	IRR, ISR	OCW2-3, ICW1
0x0E	CNT6.C	CNT6.C	---	TMRCMD
0x0F	CNT7.C	CNT7.C	IMR	OCW1, ICW2
0x10	---	DAC0.L*	ADC.L & LTCH0*	START CONV & LTCH0*
0x11	---	DAC0.H*	ADC.H & LTCH0*	START CONV & LTCH0*
0x12	---	DAC1.L*	ADC.L & LTCH1*	START CONV & LTCH1*
0x13	---	DAC1.H*	ADC.H & LTCH1*	START CONV & LTCH1*
0x14	---	DAC2.L*	ADC.L & LTCH2*	START CONV & LTCH2*
0x15	---	DAC2.H*	ADC.H & LTCH2*	START CONV & LTCH2*
0x16	---	DAC3.L*	ADC.L & LTCH3*	START CONV & LTCH3*
0x17	---	DAC3.H*	ADC.H & LTCH3*	START CONV & LTCH3*
0x18	---	DAC4.L*	ADC.L & LTCH4*	START CONV & LTCH4*
0x19	---	DAC4.H*	ADC.H & LTCH4*	START CONV & LTCH4*
0x1A	---	DAC5.L*	ADC.L & LTCH5*	START CONV & LTCH5*
0x1B	---	DAC5.H*	ADC.H & LTCH5*	START CONV & LTCH5*
0x1C	---	DAC6.L*	ADC.L & LTCH6*	START CONV & LTCH6*
0x1D	---	DAC6.H*	ADC.H & LTCH6*	START CONV & LTCH6*
0x1E	---	DAC7.L*	ADC.L & LTCH7*	START CONV & LTCH7*
0x1F	---	DAC7.H*	ADC.H & LTCH7*	START CONV & LTCH7*

* Note: these registers (registers at offset 0x10 to 0x1F) must be accessed as full 16 bit words, not as bytes.

The base address of the board is added to the register offset, shown above, to form the entire address of the register. Remember that the registers are in two groups: the low page and the high page. For example, if the base address of the board is 0x200, the CNT0.D register will appear at 0x200, while the DIO.0-7 register will appear at 0x600. These registers appear in the I/O space of the computer.

5.0 Operation and Register Bit Definitions

This section describes the operation and usage of each of the registers defined in section 4.0.

5.1 Digital I/O Section

The digital I/O section provides 32 input or output signals. The signal levels and connectors are Opto-22 compatible for direct connection to industry standard isolation I/O modules. The signals come from an 82C55 integrated circuit, which will sink or source 5 ma. Each line is pulled up by a 10K resistor to +5 volts; so inputs will default to a known level if a line is broken or not connected. During initialization, the control register should be set by the software to select which lines are inputs and which are outputs, then the output bits should be initialized by writing to the port registers.

5.1.1 Register DIO.0-7 Digital I/O Port A

This port can be configured as either 8 inputs or 8 outputs by using the mode control register. It is accessed from offset 0x00 in the high page for reads or writes. The input data can be read from this location if this port is setup as an input port, and written to this location if the port is setup as an output port.

5.1.2 Register DIO.8-15 Digital I/O Port B

Exactly the same as register DIO.0-7, this port can also be configured as either 8 input bit or 8 output bits. It is also a read/write register. It is accessed from offset 0x02 in the high page.

5.1.3 Register DIO.16-23 Digital I/O Port C

This port can be configured in groups of 4 bits, as either 4 inputs or 4 outputs. For example, the low order 4 bits can be set as inputs and the high order 4 bits as outputs. It is accessed from offset 0x04 in the high page for reads and writes. When set as outputs, bits can be individually controlled by writing to the mode control register (please see section 5.1.5B below for more information).

5.1.4 Register DIO.24-31 Digital I/O Port D

The same as register DIO.0-7, this port can also be configured as either 8 input bits or 8 output bits. It is accessed from offset 0x01, also in the high page.

5.1.5 Register MIO.0-23 Digital I/O Mode Control Register 1

This register is accessed from offset 0x06 in the high page for writes only.

A. Mode Control Command

The bit configuration shown here sets the general I/O mode. Other modes are available, but would normally not be used for isolated I/O. For more information refer to the 82C55 data sheet.

| 1 | 0 | 0 | A | CH | 0 | B | CL |

where:

A = For port A, B = For port B, CH = Upper 4 bits of port C, CL = Lower 4 bits of port C.

writing a 1 = Input, and a 0 = Output

For example, if the word: 10010001 were written to this address, then port A would be configured as an input port, port B as an output port, the upper 4 bits of port C would be outputs, and the lower 4 bits of port C would be inputs.

B. Bit Set/Reset Command

The following bit configuration, when written to this address, can be used to set individual bits in port C if those bits are configured as outputs.

| 0 | 0 | 0 | 0 | B2 | B1 | B0 | Set |

where:

B2, B1, B0 = binary code of bit number in port C to modify

Set = 1 to set or 0 to reset

This command may be used as a convenience to set bits in port C.

5.1.6 Register MIO.24-31 Digital I/O Mode Control Register 2

This write-only register is at offset 0x07 in the high page. Similar to MIO.0-23, this register is used primarily to configure DIO.24-31 (Port D) as either 8 input bits or 8 output bits. Note: reading this address will clear the odd numbered index pulse latch (see register ODDRST, section 5.1.8).

A. Mode Control Command

Write the byte, shown below, to set the direction of Port D. This byte actually sets the direction of three ports of an 82C55 IC. Besides Port D, the chip implements the BRDTST and INT.C registers. BRDTST is an input, which is the power-on default for the 82C55, so BRDTST works even before you write this byte. INT.C is an output, so you have to write this byte before INT.C will work.

| 1 | 0 | 0 | D | 0 | 0 | 1 | 0 |

where:

D = Direction of DIO.24-31 (port D), 1 = Input, and 0 = Output.

B. Bit Set/Reset Command

Individual bits in the INT.C register (see section 5.1.9) can be set by writing the following byte to this address.

| 0 | 0 | 0 | 0 | B2 | B1 | B0 | Set |

where:

B2, B1, B0 = binary code of bit number in interrupt control port to modify

Set = 1 to set or 0 to reset

Note: An 82C55 is a parallel output IC with three 8-bit ports. This board uses two 82C55s. One of these is used for Ports A, B and C (MIO.0-23); it is described in section 5.1.5. The second, described in this section, is used for Port D (MIO.24-31), the INT.C register, and the BRDTST register. This 82C55 must be initialized before you use the INT.C register, therefore, the direction for port D must be set before you use INT.C. If you change the direction of port D later, INT.C will be cleared -- instantly redirecting your interrupts. So, if you want to change the direction of Port D after you've set up INT.C and started using interrupts, you should mask interrupts and restore INT.C. Here is how to do it in C:

```
bySaveIntc = _inp(wBaseAddress + INTC);      // INTC needs to be saved, because
                                             // MIO_2 reinitializes the 8255 which
                                             // implements the INTC register.
bySaveIMR = _inp(wBaseAddress + IMR);        // get the current interrupt mask
_outp(wBaseAddress + OCW1, 0xff);            // mask off all interrupts
_outp(wBaseAddress + MIO_2, 0x82);           // set port D to output
_outp(wBaseAddress + INTC, bySaveIntc);      // restore interrupt control reg.
_outp(wBaseAddress + OCW1, bySaveIMR);       // restore interrupt mask
```

5.1.7 Register BRDTST Board Test and Address Select Register

This register can be used to determine the base address of the board through software. It is a read only register and is located at offset 0x03. The following is the BRDTST register bit definition:

| SER | Q2 |Q1 | Q0 | SA8 | SA7 | SA6 | SA5 |

Where SER = Serial data sequence corresponding to Q2-Q0 according to the following table:

When Q2, Q1, Q0 is:	7	6	5	4	3	2	1	0
then SER is:	0	1	1	1	0	1	0	1

where Q2, Q1, Q0 increment by one on each successive read of this register,

and where SA8, SA7, SA6, SA5 = The J1 jumper selection.

Through the use of this register and some software, a program does not need to be informed of the base address of the board by the user. The user needs only to set the base address which does not conflict with another device and then plug the card in. The software can then read the most significant four bits of all the possible locations where this register might be located in the I/O space until the above bit sequence is detected. Once the bit sequence is detected, the base address of the board is then known.

For example, if a possible BRDTST register is read, and its Q2, Q1, Q0 bits equal 6, then the SER bit should be a 1. If it is not, then this possible BRDTST location is not a valid one. If the SER bit is equal to one, continue to read the location. On each read, Q2, Q1, Q0 should increment by one and the above serial data sequence should appear in bit 7 of the register. If it does, then the location is valid and the base address is known (board base address equals location tested minus 0x03). As a confirmation, the J1 jumper configuration (base address selection jumper set) will appear in the lower 4 bits.

Reading this register clears the even numbered index pulse latch of the pair selected by bits IXS1 and IXS0 in the INT.C register.

5.1.8 Register ODDRST Reset Odd Index Pulse Latch

Reading this register clears the odd numbered index pulse latch of the pair selected by bits IXS1 and IXS0 in the INT.C register. To clear the even numbered index pulse latch, read the BRDTST register (see section 5.1.7).

5.1.9 Register INT.C Interrupt Control Register

This register is located at offset 0x05. It is implemented in an 82C55 IC, which it shares with a couple other registers, notably Port D. The 82C55 must be initialized before INT.C is usable. It is initialized by the MIO.24-31 register (section 5.1.6). INT.C can also be accessed on a bit by bit basis by using the MIO.24-31 register. It is a read/write register with the following bit definition:

| AZ | IXLVL |IXS1 | IXS0 | USRINT | IA2 | IA1 | IA0 |

Where:

AZ	During conversions, the hardware will perform an auto-zero of the ADC before performing a start conversion, if this bit is zero. This functionality is useful in correcting errors due to temperature changes. An auto-zero will add 26 clock periods to conversion time.
IXLVL	Used to select the active level for the index pulse input. A high (1) will select an active high input (from the RS-422 converter). A low (0) will select an active low input. The active level will cause the index pulse interrupt request to be latched to a high level causing the interrupt controller to generate an interrupt if it is enabled (see IRR register description in 5.2).
IXS1, IXS0	Used to select the two index pulse signals to watch and transfer to the latches as follows: 00=Index 0&1, 01=Index 2&3, 02=Index 4&5, 03=Index 6&7. Note: for interrupts to be enabled for a particular channel, a write to the interrupt controller also needs to occur.
USRINT	User controllable interrupt. Interrupt occurs on a low to high transition of this bit if enabled in the interrupt controller (see section 5.2 for more information).
IA2, IA1, IA0	Selects the interrupt request number to use. The following table shows the correspondence between IRQ number and the number encoded with bits IA0-IA2.

If IA2, IA1, and IA0 is:	0	1	2	3	4	5	6	7
then the IRQ number selected is:	IRQ3	IRQ15	IRQ7	IRQ12	IRQ5	IRQ10	IRQ9	IRQ11

5.2 Interrupt Controller Section

The interrupt controller section provides a real-time interrupt, index pulse interrupts, and an end-of-conversion interrupt. The board uses an 82C59 interrupt controller. Since only a few of the features of this chip are used, you only need to be concerned with two things: 1) initializing the chip, and 2) masking and unmasking the interrupts.

To initialize the 82C59 first send a 0x1A to the ICW1 register. This sets the chip for single chip, level triggered mode. Then send 0x00 to ICW2; we're not using this register, but we need to write it in sequence. The IRQ should be selected before initialization, which requires that INT.C is initialized. Here are the steps you would go through to start timer interrupts:

```
_outp(wBaseAddress + MIO_2, 0x92);  // initialize INTC as output reg.
                                    // sets port D to input since we have
                                    // to set it to something.
_outp(wBaseAddress + INTC, 0x04);   // selects IRQ 5
_outp(wBaseAddress + ICW1, 0x1a);   // initialize 82C59 as single chip,
                                    // level triggered
_outp(wBaseAddress + ICW2, 0x00);   // ICW2 - not used, must write
_outp(wBaseAddress + OCW1, 0xff);   // mask off all interrupt sources (the
                                    // interrupt on the motherboard isn't
                                    // enabled yet, you do that when you install
                                    // your interrupt handler.).

       // install interrupt handler
       // start timer

byMotherMask = _inp(0x21);           // get motherboard interrupt mask
byMotherMask &= ~0x20;               // unmask bit for IRQ 5
_outp(0x21, byMotherMask);           // enable interrupt on motherboard

       ...

_outp(wBaseAddress + OCW1, ~0x04);   // unmask timer interrupt
```

Undesired interrupt sources need to be masked, and desired sources, unmasked. Write the mask to the OCW1 register. The mask can be read back from the same address, but the 82C59 documentation refers to it as the IMR register when reading. Here is the interrupt source that each bit of the register refers to:

| IXEVN | IXODD | LIXEVN |LIXODD | EOC | TP0 | UI | TP2 |

Where:

IXEVN, IXODD	The index pulses corresponding to the pair selected by IXS1 and IXS0 will appear here. These are only useful to see the actual level of the index pulse input. They are not qualified by IXLVL.
LIXEVN, LIXODD	The latched-high-when-active output of the index pulse pair selected by IXS1 and IXS0 will appear here. The active level is set by IXLVL signal in register INT.C (see section 5.1.8 for more information). To reset the even bit, read BRDTST; to reset the odd bit, read ODDRST. The bits are level triggered, and can not be reset if they are active.
EOC	This is the end-of-conversion bit. The analog input section can be configured to generate an interrupt when the ADC is finished with a conversion. The hardware also sets the interrupt request flag at the end of a conversion, weather or not interrupts are enabled. It is cleared by writing a zero here.
UI	A user controlled interrupt input (see INT.C register for more information). This may be used during software development to debug an interrupt service routine without using the timer generated interrupt.
TP0 and TP2	These bits refer to timer pulse 1, 2, or 3. TP0 is the main timer which can be used to generate periodic interrupts during which time servo calculations are performed. TP1 feeds TP0. TP2 is running off a faster clock and is user definable. TP1 can not generate an interrupt. These bits are not latched when the timer terminal event occurs.

Notes:

The above bit definition applies not only to the IMR (Interrupt Mask Register), but also to the ISR (Interrupt Status Register), and the IRR (Interrupt Request Register). Write a 1 to disable the interrupt and a 0 to enable the interrupt.

An EOI is not required to be sent to the 82C59 on the servo board, because that section of the chip is not used. An EOI must, however, be sent to the 82C59 (or equivalent) on the mother board, if you are using interrupts.

If polling, the IRR register needs to be read. Send a 0x0A to OCW3 which selects the IRR for the next read. Then read from IRR (which is the same address, but the manufacturer's data sheet refers to it by a different name in this circumstance). Even if the interrupts are masked (disabled), the hardware will still set bits in the IRR.

There are only two address for the 82C59, yet 10 different register names are used, depending on circumstance, in the manufacturer's documentation. Here is a summary of the addresses and names:

Offset	Read	Write
0x09	(a preceding OCW3 selects) IRR (if OCW3 = 0x0a) ISR (if OCW3 = 0x0b)	ICW1 (xxx1xxxx) OCW2 (xxx00xxx) OCW3 (xxx01xxx)
0x0B	IMR	ICW2 (if following ICW1) ICW3 (may follow ICW2) ICW4 (may follow ICW2 or ICW3) OCW1 (all other times)

We set up our 82C59 as level triggered. The interrupts on the motherboard are edge triggered.

The register at offset 0x09 also appears at 0x0D, and the register at offset 0x0B also appears at 0x0F.

The 82C59 registers ICW3 and ICW4 are not used. ICW1 and ICW2 are only used during initialization (ICW = Initialization Command Word), all other writes should be to OCW1, 2, or 3 (OCW = Operation Command Word) . Refer to a 82C59 data sheet for further information.

5.3 Timer Section

This section provides a periodic real-time interrupt. The input for this section is a 7.15909 MHz clock, divided down from the 14.31818 MHz bus oscillator. This is fed into two timers - timers 1 and 2. The output of timer 1 is then fed into timer 0. Although the timers can be setup in many different modes, the following modes are recommended for normal operation. Other modes can be found in an 82C54 data sheet.

5.3.1 Register TMRCMD Timer Command Register

This write-only register is used to set the mode and other parameters defined further in section 5.3.2 and section 5.3.3. It is located at offset 0x0E in the high page. The timer counter values can be read from or loaded to the associated timer register, after first writing a control word here to specify the appropriate function.

The bit definitions are as follows:

| SC1 | SC0 | RL1 | RL0 | M2 | M1 | M0 | BCD |

where:

SC1	SC0	Selects
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Read Back

RL1	RL0	Selects
0	0	Counter Latch
0	1	Read/Load LSB
1	0	Read/Load MSB
1	1	Read/Load LSB followed by MSB

M2	M1	M0	Selects
0	0	0	Mode 0 Interrupt On Terminal Count
0	0	1	Mode 1 Programmable One-Shot
X	1	0	Mode 2 Real-time Interrupt
X	1	1	Mode 3 Square Wave Generator
1	0	0	Mode 4 Software triggered Strobe
1	0	1	Mode 5 Hardware Triggered Strobe

BCD	Selects
0	Binary - 16 Bits
1	BCD - 4 Decades

5.3.2 Register TIMER.2 Timer Counter Value

This timer is user definable and can be used for general purpose timer/interrupt functions. It is located at offset 0x0C in the high page.

5.3.3 Registers TIMER.0 and TIMER.1 Timer Counter 0 and 1

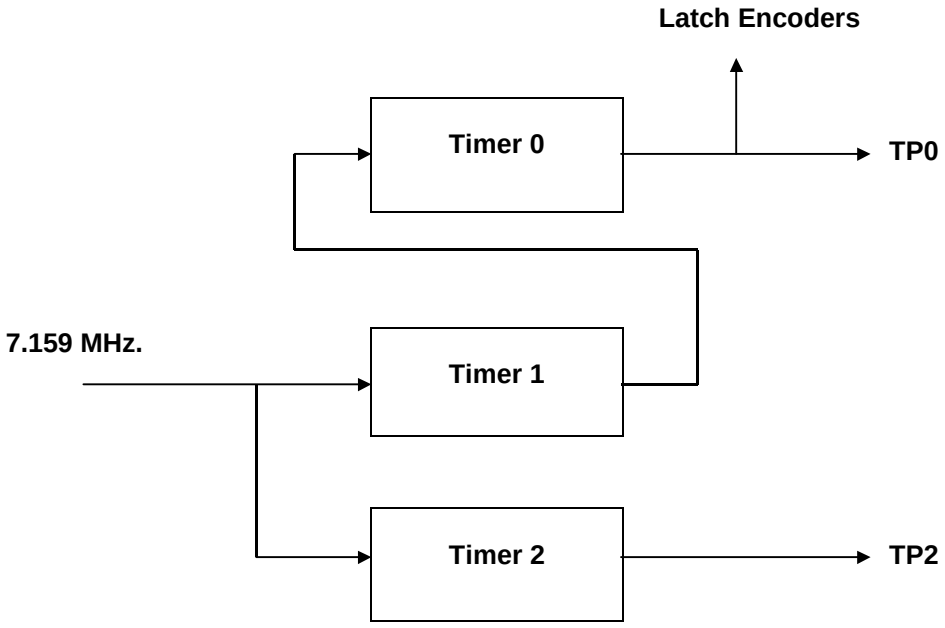
These timers are used to generate the real-time interrupt request from several microseconds up to 10 minutes. Register TIMER1 is located at offset 0x0A and register TIMER0 is located at offset 0x08. Timer 1 should be set for the square wave mode (Mode 3) with a value of 180 decimal (range: 2-65535). This will provide about a 25 microsecond time interval for the input of timer 0. Timer 0 should be set for real-time interrupts (Mode 2). Its value can then be set (range: 2-65535) to obtain an interval of 50 microseconds to 1.638 seconds in 25 microsecond increments. For example, if the interval for timer 1 is set to 180, then timer 0 should be set to 40 to obtain a 1mS interrupt interval.

The hardware will also latch the encoder counts into their respective latch registers on the rising edge of TIMER0's output. This saves 8 extra write operations when reading the encoders in the interrupt service routine.

The resolution of timers one and two is 139.68 nanoseconds.

The program sequence is to set the control word (TMRCMD register) then set the count value (TIMER.X register). On the next clock, loading is performed and then counting starts.

Note: these timers can be configured to generate interrupts at a rate faster than the computer can possibly service the interrupt. For example, setting the interrupt rate at 10 microseconds would probably result in the computer constantly servicing the interrupt, with no time left over for any other processing, thereby "locking up" the computer.



5.4 Encoder Input Section

These counters accept the quadrature encoder inputs “A” and “B”. From this input stream they generate a 24 bit position count. The hardware can be configured so that the count is multiplied by 1, 2, or 4. As a convenience, the count value is automatically latched to all counters for reading by the real-time interrupt.

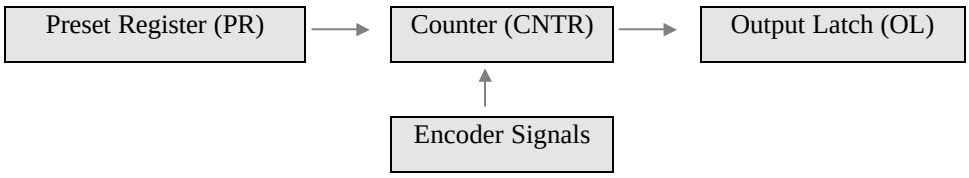
If one of the encoder inputs becomes disconnected, the counter will increment then decrement on each edge of the other pulse. This may be used to determine if the inputs are valid by detecting a count change of at least two. The maximum encoder input rate is 1.2 MHz.

The actual inputs are buffered using a RS-422 compatible receiver which accepts either differential or single ended inputs. For differential inputs, both inputs need to be connected for each channel. For single ended operation only the positive input (for example A+) is to be connected.

The counter specific registers are arranged in memory so that two registers for two axis can be read or written simultaneously by performing word wide read and write operations. Reading two encoders at a time saves a considerable amount of processing time when reading all eight encoders in the interrupt service routine. They can also be accessed individually, if desired, by performing byte wide operations.

Another time-saving measure is that the hardware is configured so that the timer-terminal-count event automatically latches all of the counters to their respective output latches. This occurs even if the timer interrupt is disabled. This saves a write operation, during the interrupt service routine, for each encoder input. If the timer is setup in mode 5, it will be effectively disabled and the timer terminal count event will not occur. If the timer itself is disabled (as opposed to the timer *interrupt*), the encoder counters can still be latched to their output latches through the use of a register command (see below).

The following diagram illustrates the operation of the counters.



5.4.1 Register CNTX.D Counter Data Register for Channel X

These registers are used to access the count portion of the counters, they are located in the low page. On reads, the counter register is referred to as the Output Latch, and on writes it is referred to as the Preset Register. The 24 bit data in an individual internal counter data register is accessed by a series of three 8 bit reads to the Output Latch. Two encoder output latches appear next to each other in the I/O space, therefore, with three 16 bit reads, the 24 bit counter data for each of two axis can be obtained. A pointer, internal to the counter hardware, points to one of the three bytes which is next in line to be accessed. This pointer can be reset using the RADR bit in the Master Control Register (see 5.4.2b).

A. Output Latch (OL)

This read-only register returns the byte of the latched count pointed to by the internal counter address pointer. The 3 byte sequence is from least significant byte to the most significant byte. As each byte is read, the internal address pointer is incremented to the next byte. The count register is copied (latched) to this register when the real-time interrupt occurs. The count register can also be latched to this register by software (see the counter master control register).

B. Preset Register (PR)

This write-only register is used to set the count to a specified, or preset, value. Write the byte of the preset value pointed to by the address pointer. The three byte sequence is from least significant byte to the most significant byte. This value is then used to load the counter and for comparison to the counter value.

5.4.2 Register CNTX.C Counter Command Register for Channel X

This register, is used to access the command portion of the counter. On reads, it is referred to as the Output Status register. On writes it is referred to as one of the following:

Name	D7	D6
Master Control register	0	0
Input Control register	0	1
Output Control register	1	0
Quadrature register	1	1

A. Output Status Register

This read-only register may be used to return the status information about the currently selected register.

| X | X | X | UP | SIGN | CMP | CRY | BRW |

where:

Bit Name	Bit Description
UP	Reading a 1 means the counter is counting up, reading a 0 means it is counting down.
SIGN	Reading a 1 means the counter has overflowed and a 0 means it has underflowed.
CMP	This bit toggles every time the counter preset (PR) equals the count (CNTR).
CRY	This bit toggles every time the counter overflows.
BRW	This bit toggles every time the counter underflows
X	not used

B. Master Control Register

This write-only register is written to the counter command register with bits 7 and 6 set to 0. It is used to clear or transfer the count value and associated flags.

| 0 | 0 | MRST | RCMP | TPR | RCNT | TOL | RADR |

where:

Bit Name	Bit Description
MRST	Master Reset - Reset all control registers, flags, and the internal address pointer. Also, set SIGN and set Preset Register (PR) to 0xFFFFFF.
RCMP	Resets the Output Status Register's CMP bit.
TPR	Transfer Preset Register (PR) to Counter (CNTR), a 24 Bit operation.
RCNT	Reset CNTR, BRW, and CRY. Set SIGN.
TOL	Transfer Counter (CNTR) to the Output Latch (OL), a 24 Bit operation.
RADR	Reset the address pointer.

C. Input Control Register

This write-only register is written to the counter command register with bit 7 set to 0 and bit 6 set to 1. It is used to set the counter operating mode (recommended setting is 0x68, in bold below).

| 0 | 1 | P3 | P4 | ENA/B | DCR | INC | MDE |

where:

Bit Name	Bit Description
P3	Pin 3 Function, **1 = OL Load**, 0 = CNTR Load
P4	Pin 4 Function, 1 = Gate, **0 = Reset Counter**
ENA/B	Enable A/B, **1 = Enable Inputs A and B**, 0 = Disable
DCR	Decrement, 1 = Decrement counter once, **0 = No operation**
INC	Increment, 1 = Increment counter once, **0 = No operation**
MDE	1 = A is clock and B is direction (0=up, 1=down), **0 = A is up count and B is down count** (overridden in quadrature mode)

D. Output Control Register

This read-only register is written to the counter command register with bit 7 set to 1 and bit 6 set to 0. It is used to define the count mode and the function of output pins which are not used. The recommended setting for this register is 0x80.

E. Quadrature Register

This register is written to the counter command register with bit 7 and bit 6 set to 1. It is used to select and enable the quadrature mode. It has the following options (recommended setting, 0xC3, is in bold below):

0xC0 - Disable quadrature mode

0xC1 - Enable times 1 (X1) quadrature mode

0xC2 - Enable times 2 (X2) quadrature mode

0xC3 - Enable times 4 (X4) quadrature mode

5.5 Analog Output Section

These write-only registers provide access to the digital-to-analog converters (DACs) as 16 bit words. Accessing the registers as bytes is invalid. The registers are located at even offsets beginning at location 0x10 and ending at 0x1E. The digital-to-analog converters have a 13 bit resolution. The output voltage has a -10V to +10V range. Additionally, Opto-22 compatible digital outputs corresponding to the sign of the analog signal are automatically generated by the hardware. These outputs are located on the P2 connector. Upon a hardware reset, and at power on, the DAC output is set to 0V by the hardware.

Below are some example DAC values:

Value Written	Output Voltage
0x0000	-10V
0x1000	0V
0x1FFF	+10V

These values are inverted, compared to the datasheet for the MAX547B because an inverting amplifier buffers the output of the MAX547B.

Two optional potentiometers may be installed to adjust the reference voltage (zero offset) for the analog input as well as the analog output circuits. Also, the board can be configured so that the DACs are all latched simultaneously by the hardware at the time of the periodic interrupt. Refer to Section 2.3 – Jumper J3, for more information.

5.6 Analog Input Section

The analog-to-digital converters (ADCs) are controlled by a set of registers located at even offsets beginning at location 0x10 and ending at 0x1E in the high bank of I/O addresses. They must be accessed as 16 bit words. To read an analog channel:

1. Read from the ADC. This is a dummy read which only sets the input multiplexer to the proper channel. Ignore the data, just the act of reading sets the multiplexer. For example, to get the voltage on channel 2, read 16 bits from 0x14 in the high bank of I/O addresses.

2. Write to the STARTCONV & LATCHn register to start the conversion on a particular channel. For example, to get the voltage on channel 2, write 0x0000 to offset 0x14 in the high bank of the I/O addresses.

3. The software must wait (see table below) for the conversion to finish. Other necessary operations can be performed by the software during this time. Refer to the ADC12441 data sheet to calculate the exact time (note: the clock is 1.7978 MHz). The End-Of-Conversion bit can be configured to cause an interrupt. Alternatively, the EOC bit always sets its interrupt request flag, even if the EOC interrupt is disabled, so the software may poll this bit to determine if the conversion has finished.

4. Read 16 bits from the ADC location that you read from in step 1. The data read will be the newly converted channel input.

Status of the Auto-zero (AZ) Bit in Register INT.C	Polling Delay	Not Polling Delay
0 = Auto-zero	19 µS	34 µS
1 = Don't Auto-zero	4 µS	19 µS

Note: It will take the "polling delay" amount of time for the EOC bit to go to zero after the start conversion command is issued. When the bit transitions from a zero to a one, the conversion is finished. Polling without auto-zero is probably the preferred method, therefore, after waiting 4 µS, check EOC, if it is a one then the conversion is done.

The value read from the ADC is a normal 2's compliment value. Below are some examples which assume the input range (set by jumper J2) is set at -10 to +10 Volts:

Value Read	Input Voltage
0x0FFF	9.9975V
0x0000	0.0000V
0x1FFF	-0.0024V
0x1000	-10.0000V

Please see section 2.2 for information on how to set the input range.

6.0 Appendix A - IBM PC IRQs

IBM PC Hardware Interrupt Table (in order of priority)

IRQ#	Interrupt	Function
IRQ0	0x08	timer (55ms intervals, 18.2 per second)
IRQ1	0x09	keyboard service required
IRQ2	0x0A	slave 8259 or EGA/VGA vertical retrace
IRQ8	0x70	real time clock (AT,XT286,PS50+)
IRQ9	0x 71	software redirected to IRQ2 (AT,XT286,PS50+)
IRQ10	0x 72	reserved (AT,XT286,PS50+)
IRQ11	0x 73	reserved (AT,XT286,PS50+)
IRQ12	0x 74	mouse interrupt (PS50+)
IRQ13	0x 75	numeric coprocessor error (AT,XT286,PS50+)
IRQ14	0x 76	fixed disk controller (AT,XT286,PS50+)
IRQ15	0x 77	reserved (AT,XT286,PS50+)
IRQ3	0x0B	COM2 or COM4 service required, (COM3-COM8 on MCA PS/2)
IRQ4	0x0C	COM1 or COM3 service required
IRQ5	0x0D	fixed disk or data request from LPT2
IRQ6	0x0E	floppy disk service required
IRQ7	0x0F	data request from LPT1 (unreliable on IBM mono)

7.0 Appendix B - Common I/O Port Addresses

The following is a table of common I/O port address definitions in the IBM PC.

Address	Description of Device
000-0FF	Reserved for DMA, PIC, PIT, PPI, Keyboard, etc.
0F0-0FF	Math coprocessor (AT, PS/2)
100-10F	POS Programmable Option Select (PS/2)
110-1EF	System I/O channel
170-17F	Fixed disk 1 (AT)
1F0-1FF	Fixed disk 0 (AT)
200-20F	Game Adapter
210-217	Expansion Card Ports (XT)
220-26F	Reserved for I/O channel
270-27F	Third parallel port
280-2AF	Reserved for I/O channel
2A2-2A3	MSM58321RS clock
2B0-2DF	Alternate EGA, or 3270 PC video (XT, AT)
2E2-2E3	Data acquisition adapter (AT)
2E8-2EF	COM4 non PS/2 UART (Reserved by IBM)
2F0-2F7	Reserved
2F8-2FF	COM2 Second Asynchronous Adapter
300-31F	Prototype Experimentation Card (except PCjr)
320-32F	Hard Disk Controller (XT)
330-33F	Reserved for XT/370
340-35F	Reserved for I/O channel
360-36F	PC Network
370-377	Floppy disk controller (except PCjr)
378-37F	Second Parallel Printer
380-38F	Secondary Binary Synchronous Data Link Control (SDLC) adapter
390-39F	Cluster Adapter
3A0-3AF	Primary Binary Synchronous Data Link Control (SDLC) adapter
3B0-3BF	Monochrome Display Adapter (write only)
3C0-3CF	EGA/VGA
3D0-3DF	Color Graphics Monitor Adapter (ports 3D0-3DB are write only)
3E8-3EF	COM3 non PS/2 UART (Reserved by IBM)
3F0-3F7	Floppy disk controller (except PCjr)
3F8-3FF	COM1 Primary Asynchronous Adapter

Note:

The I/O space at addresses 0x200, 0x300, and 0x340 are each large enough for this card and are also usually not used on most systems.

Port addresses are not always constant across PC, AT and PS/2

Many cards designed for the ISA BUS use only the lower 10 bits of the port address but some ISA adapters use addresses beyond 0x3FF. Any address that matches in the lower 10 bits will decode to the same card. It is up to the adapters to resolve or ignore the high bits of the port addresses. An example would be the Cluster adapter that has a port address of 390h. The second cluster adapter has a port address of 790h which resolves to the same port address with the cards determining which one actually gets the data.

8.0 Appendix C - Component Locations