AOP-12D

Multi-Function
Analogue Output Card

Blue Chip Technology

User Manual

AOP-12d

User Manual

Document Part N°	0127-0146
Document Reference	AOP-12d\..\0127-0146.doc
Document Issue Level	2.0
Manual covers PCBs identified	KFA12o Rev. B

Information offered in this manual is correct at the time of printing. Blue Chip Technology accepts no responsibility for any inaccuracies. This information is subject to change without notice.

Blue Chip Technology Ltd.
Chowley Oak, Tattenhall,
Chester, Cheshire, CH35 9EX.
Telephone : (01829) 772000   Facsimile : (01829) 772001.

Amendment History

Issue Level	Issue Date	Author	Amendment Details
1.0	9/8/95	SH	First approved issue, new front sheet.
2.0	20/12/95	EGW	Addition of EMC information to Technical Section. Errors corrected. References to current outputs removed. Added layout diagram. Earlier part no. was 127-038. Filename was ...\Userg.doc

OUTLINE DESCRIPTION

The AOP-12d is a PC-compatible short card which provides digital inputs and outputs, and analogue outputs.

There are 24 TTL compatible programmable digital input/outputs available externally. There are also three programmable timers. One of the timer outputs is available externally to the user.

There are 12 analogue outputs available as ±10 Volts. Output resolution is 12 bits.

DMA data transfer is available on the analogue output channels.

SPECIFICATION

Analogue Outputs

Analogue Outputs	12 channels
Resolution	12 Bit Monotonic
Voltage Outputs	±10 Volts @ 10mA maximum one O/P, or 5mA each from all O/Ps
Output Error Volts	0.5% of Span
Output Settling Time	3µS to ±1 LSB
Data Transfer	I/O Port or DMA
DMA Channels Supported	1,2 and 3
Fastest DMA Transfer Rate	12µS per Transfer
Channel Selection	Any or all channels may be selected to be updated
DMA Transfer Initialisation	Software Start Signal
Maximum Time Skew	
Channel 1 To Channel 12	144µS
Between Channels	12µS
DMA Timing Source	On-board Programmable Timer

Digital Input/Outputs

Number of Channels	24

Digital Inputs

High Level Input	2.2 Volts minimum
Current	10µA sink
Low Level Input	0.8 Volts maximum
Current	10µA source

Digital Outputs

Logic High Voltage	3.5 Volt minimum
Current	400µA source
Logic Low Voltage	0.4 Volt
Current	2.5mA sink

Timers

Number Of Timer Channels	3
Timer Usage	
Timer 0	Pre-Scalar for timer 1
Timer 1	Timer for DMA transfer
Timer 2	Uncommitted (output available)
Timer 0	
Resolution	1µS
Minimum Time	2µS
Maximum Time	130mS
Timer 1	
Resolution	timer 0 output value
Minimum Time	4µS
Maximum Time	2.3 hours
Timer 2	
Resolution	1µS
Minimum Time	2µS
Maximum Time	130mS

Board Connectors

PC ISA 8-bit card	
Analogue Signals	50 Way Male `D' Type
Digital Signals	50 Way IDC Male Box Header

ELECTROMAGNETIC COMPATIBILITY (EMC)

This product meets the requirements of the European EMC Directive (89/336/EEC) and is eligible to bear the CE mark.

It has been assessed operating in a Blue Chip Technology Icon industrial PC. However, because the board can be installed in a variety of computers, certain conditions have to be applied to ensure that the compatibility is maintained. It meets the requirements for an industrial environment (Class A product) subject to those conditions.

- The board must be installed in a computer system which provides screening suitable for the industrial environment.

- Any recommendations made by the computer system manufacturer/supplier must be complied with regarding earthing and the installation of boards.

- The board must be installed with the backplate securely screwed to the chassis of the computer to ensure good metal-to-metal (i.e. earth) contact.

- Most EMC problems are caused by the external cabling to boards. With analogue boards particular attention must be paid to this aspect. It is imperative that any external cabling to the board is totally screened, and that the screen of the cable connects to the metal end bracket of the board and hence to earth. It is recommended that round screened cables with a braided wire screen are used in preference to those with a foil screen and drain wire. Use metal connector shells which connect around the full circumference of the screen; they are far superior to those which earth the screen by a simple “pig-tail”. Standard ribbon cable will not be adequate unless it is contained wholly within the cabinetry housing the industrial PC.

- If difficulty with interference is experienced the cable should also be fitted with a ferrite clamp as close possible to the connector. The preferred type is the Chomerics clip-on style, type H8FE-1004-AS.

- It is recommended that cables are kept as short as possible, particularly when dealing with low level signals.

- Ensure that the screen of the external cable is bonded to a good RF earth at the remote end of the cable.

Failure to observe these recommendations may invalidate the EMC compliance.

> Warning
> This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

EMC Specification

A Blue Chip Technology Icon industrial PC fitted with this card meets the following specification:

Emissions	EN 55022:1995	
	Radiated	Class A
	Conducted	Class A & B
Immunity	EN 50082-1:1992 incorporating:	
	Electrostatic Discharge	IEC 801-2:1984
	Performance Criteria B	
	Radio Frequency Susceptibility	IEC 801-3:1984
	Performance Criteria A	
	Fast Burst Transients	IEC 801-4:1988
	Performance Criteria B	

QUICK INSTALLATION

Before installing the card into your computer system, there are a number of links which must be set.

The settings of these links will depend upon the computer system into which the card is being fitted. Users unfamiliar with the settings of links should refer to the section “Detailed Card Installation”. For those unfamiliar with Binary and Hexadecimal numbers, a brief explanation is included in the Appendices.

Base Address

Select an unused I/O address range for the card. The card requires a block of 16 contiguous addresses.

The base address is set on jumper block JP3. Fitting a link is equivalent to a logic “0”. Leaving the link open is equivalent to a logic “1”. The card is shipped with the default address setting of 300 $_{\text{Hex.}}$ This is suitable for most small installations.

Interrupts

The DAC generates interrupts at the end of a DMA transfer. Interrupts IRQ2 to IRQ7 are provided. The PIO and the Timer cannot generate interrupts.

The interrupt setting is selected by a link on jumper block JP1. If interrupt operation is not required leave the link off.

DMA Settings

The card can transfer data from memory to the analogue outputs using DMA.

The settings are controlled by links on jumper blocks JP2 and JP4. DMA channels 1 and 3 are provided.

JP4 controls the setting of the DMA Request channels for the DAC. JP2 controls the setting of the DMA Acknowledge channels. The settings must be the same on both JP2 and JP4. If DMA operation is not required, the links may left open to allow the unused channels to be used by other cards.

The appendix contains a section explaining the use of DMA

Fitting the Card

Once all the links have been set, the card can be installed into the host computer.

Observe all safety precautions and anti-static precautions. If possible try and locate the card away from 'noisy' cards such as hard disc controllers, network cards and processor cards.

USING THE CARD

External Input/Output Connections

The AOP-12d has two connectors for external circuitry.

The analogue output signals are available at a standard 50 pin D-type connector which protrudes through the end bracket of the printed circuit board.

The digital input output signals are presented on a 50 way IDC header at the inner end of the printed circuit board. These signals may be brought to a connector on a second bracket the rear cover of the PC using a 50 way ribbon extension cable. Filtered connectors are recommended for EMC.

Analogue Connections

The following table shows the pin out of the D-type analogue connector CON1. The pins are arranged in three rows.

PIN	USAGE	PIN	USAGE	PIN	USAGE
1	V output 1	18	No connect	34	V output 12
2	No connect	19	V output 7	35	No connect
3	No connect	20	No connect	36	No connect
4	V output 2	21	No connect	37	No connect
5	No connect	22	V output 8	38	No connect
6	No connect	23	No connect	39	No connect
7	V output 3	24	No connect	40	No connect
8	No connect	25	V output 9	41	No connect
9	No connect	26	No connect	42	No connect
10	V output 4	27	No connect	43	No connect
11	No connect	28	V output 10	44	No connect
12	No connect	29	No connect	45	No connect
13	V output 5	30	No connect	46	No connect
14	No connect	31	V output 11	47	No connect
15	No connect	32	No connect	48	No connect
16	V output 6	33	No connect	49	Analogue ground
17	No connect			50	No connect

Analogue Voltage Outputs

The analogue output signals from the AOP-12d are available as voltages only, with the ability to supply a limited current.

Each of the 12 output signals (Voutput-1 to Voutput-12) has a corresponding analogue ground or 0 Volt connection. The voltage outputs are referenced to these connections. Measuring output voltages with reference to other ground points (particularly the digital ground) will give electrically noisy results.

The voltage output has a span from +10 Volts to -10 Volts with an output drive of 10mA maximum for any single output and 5mA maximum each for all outputs simultaneously.

If large capacitive loads are to be connected to the voltage output, it is recommended that a series resistance of approximately 100 ohms is placed in series with the output voltage to avoid oscillations occurring at the output.

Digital Connections

The following table shows the pin out of the IDC digital signal connector CON2. The pins are arranged in two rows.

PIN	SIGNAL	PIN	SIGNAL
1	DIO port A, bit 0	2	DIO port A, bit 1
3	DIO port A, bit 2	4	DIO port A, bit 3
5	DIO port A, bit 4	6	DIO port A, bit 5
7	DIO port A, bit 6	8	DIO port A, bit 7
9	DIO port B, bit 0	10	DIO port B, bit 1
11	DIO port B, bit 2	12	DIO port B, bit 3
13	DIO port B, bit 4	14	DIO port B, bit 5
15	DIO port B, bit 6	16	DIO port B, bit 7
17	DIO port C, bit 0	18	DIO port C, bit 1
19	DIO port C, bit 2	20	DIO port C, bit 3
21	DIO port C, bit 4	22	DIO port C, bit 5
23	DIO port C, bit 6	24	DIO port C, bit 7
25	Digital ground	26	Digital ground
27	Digital ground	28	Digital ground
29	Digital ground	30	Digital ground
31	Timer2 Output	32	Timer 0 Output
33	Digital ground	34	Digital ground
35	Digital ground	36	Digital ground
37	Digital ground	38	Digital ground
39	Digital ground	40	Digital ground
41	Digital ground	42	Digital ground
43	Digital ground	44	Digital ground
45	Digital ground	46	Digital ground
47	Digital ground	48	Digital ground
49	Digital ground	50	Digital ground

DIO - Digital input/output.

OPERATION OF THE CARD

Programmable Digital Input/Outputs

The AOP-12d includes an NEC µPD71055 device which is equivalent to an Intel 8255 PIO.

This device provides 24 programmable digital I/O channels. It is suitable for sensing the presence of, or driving TTL connections only. These connections should be kept as short as possible, less than 2 metres is recommended.

The digital I/O appears to the PC as four ports. The first three can be set as input or output by writing suitable codes to the fourth Control Port.

These four ports are mapped into the AOP-12d address map as follows:

BASE	
+ 4	Programmable Digital I/O Port A (read/write);
+ 5	Programmable Digital I/O Port B (read/write);
+ 6	Programmable Digital I/O Port C (read/write);
+ 7	Control Port (write only).

A summary of the codes required to change the operation of the ports are given later. A typical sequence of events to use this feature would be :

- Decide on the input/output mix and write the appropriate code to BASE + 7.

- Read from the selected output port or write to the selected output port.

Control Codes

The µPD71055 can operate in one of 3 modes.

The first (Mode 0) provides for simple inputs and outputs for three, 8 bit ports. Data is written to or read from a specified port (A, B, or C) without the use of handshaking.

The following table gives a summary of the most commonly used ‘Control Words’ which must be written to the control port to configure the µPD71055 I/O ports in Mode 0.

CONTROL WORD (hex)	CONTROL WORD (decimal)	SET ALL of PORT A as	SET ALL of PORT B as	SET HIGH 4 BITS of C as	SET LOW 4 BITS of C as
80	128	Output	Output	Output	Output
81	129	Output	Output	Output	Input
82	130	Output	Input	Output	Output
83	131	Output	Input	Output	Input
88	136	Output	Output	Input	Output
89	137	Output	Output	Input	Input
8A	138	Output	Input	Input	Output
8B	139	Output	Input	Input	Input
90	144	Input	Output	Output	Output
91	145	Input	Output	Output	Input
92	146	Input	Input	Output	Output
93	147	Input	Input	Output	Input
98	152	Input	Output	Input	Output
99	153	Input	Output	Input	Input
9A	154	Input	Input	Input	Output
9B	155	Input	Input	Input	Input

Mode 1 enables the transfer of data to or from a specified 8 bit port (A or B) in conjunction with strobes or handshaking signals on port C.

In Mode 2, data is transferred via one bi-directional 8 bit port (A) with handshaking (port C).

Refer to the µPD71055 or i8255 data sheet for full details of the settings and use of Modes 1 and 2.

Timers

The AOP-12d includes an NEC µPD71054 timer chip which is equivalent to an Intel 8254.

The timer chip contains three independent 16 bit counters which may be operated in different modes. There are five basic modes of operation with each mode providing a different output signal from the three output pins of the device.

IMPORTANT

> Timers 0 and 1 are crucial to the operation of the board. The DAC section is controlled by the output of these timers so for all operating modes of the DAC, these timers must be configured to run. See the section on TIMER INITIALISATION for code examples to configure these timers.

The reference clock for timers 0 and 2 is 1Mhz. Timer 1 is in series with the output of timer 0.

Timer 0 is committed as the first divider for DMA in the AOP-12d and its output is also available on the external connector.

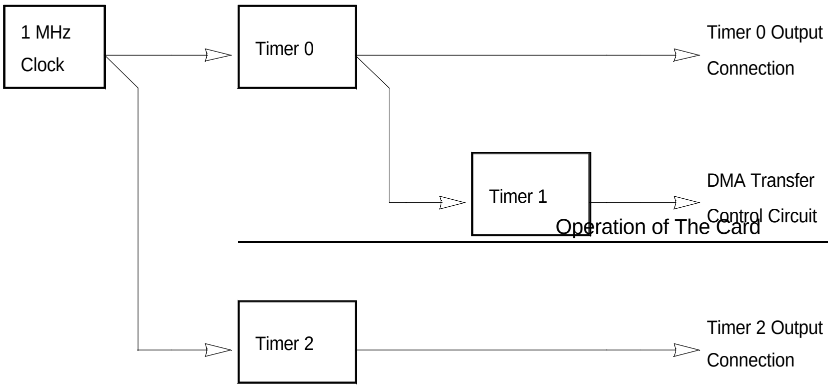
Timer 2 is uncommitted in the AOP-12d and its output is available on the external connector.

In DMA mode, timers 0 and 1 set the rate at which data is DMA'd. To set a particular DMA rate, use the following equation:

Note that the 1 MHz clock is divided by 8 by the circuitry giving
a reference of 125 KHz

DMA RATE (in Hz) = 1250000 / divider

where "divider" = (65536 * TIMER 1 value) + (TIMER 0 value)



Block Diagram of Timer Connections

The timer circuit appears to the PC as four ports. These four ports are mapped into the AOP-12d address map as follows:

BASE	
+ 8	Timer/Counter 0 (read/write).
+ 9	Timer/Counter 1 (read/write).
+ 10	Timer/Counter 2 (read/write).
+ 11	Control register, (write only).

Bits 6 and 7 in the control register enable and disable Timer 0 and 1, and Timer 2.

Timer Initialisation

Before the DAC section can be operated IN ANY MODE Timers 0 and 1 must be configured. To do this, follow this sequence of writing to the timer registers

Output hex 0 to control register (base + 0). This disables timers 0 and 1.

Read base + 0. This resets the internal state machine logic.

Output hex 34 to the timer control register (base + 11). This sets timer 0 into mode 2.

Output 3 to timer 0 count register (base + 8). This writes 3 into the LOW 8 bits of the 16 bit counter register.

Output 0 to timer 0 count register (base + 8). This writes 0 into the HIGH 8 bits of the 16 bit counter register.

Output hex 78 to the timer control register (base + 11). This sets timer 1 into mode 4.

Output hex 1 to timer 1 count register (base + 9). This writes 1 into the LOW 8 bits of the 16 bit counter register.

Output hex 0 to time 1 count register (base + 9).

Output hex 40 to control register 1 (base + 0).

In BASIC the initialisation sequence would look like this:

```
10 out (BASEADD + 0,&h0)
20 dummy% = inp(BASEADD)
30 out (BASEADD + 11,&h34)
40 out (BASEAAD + 8,&h03)
50 out (BASEAAD + 8,&h00)
60 out (BASEADD + 11,&h78)
70 out (BASEADD + 9,&h03)
80 out (BASEADD + 9,&h00)
90 out(BASEADD,&h40)
```

DAC Section

The Digital to Analogue converter is accessed as 4 ports. These ports are mapped into the PC at the following addresses.

BASE	
+ 0	DAC Control register
+ 1	Channel enable register 1
+ 2	Channel enable register 3
+ 3	DAC output register

The DAC is 12 bits wide, therefore two write operations are required to the DAC output register to load the required value. The low 8 bits are sent first followed by the high 4 bits.

Output Value	Output Voltage
0	-10 V
2048	0 V
4095	+10 V

The DAC section operates in one of two basic modes, I/O or DMA.

To send data to the DAC in I/O mode use the following sequence:

Initialise the timer (see Timer Initialisation above).

Write 0 to BASE + 1 and BASE + 2, this enables all the channels.

Read BASE + 0 , this resets the internal state machine.

Write the low 8 bits of the DAC data into BASE + 3.

Write the high 4 bits of the DAC data into BASE + 3.

Write the channel number to update (1 to 12) to BASE + 0.

Write 0 to BASE + 0.

To operate in DMA mode updating a single channel use the following sequence:

Initialise the timer, as above.

Program timers 0 and 1 for the required output rate.

Write 0 to BASE + 1 and BASE + 2, this enables all the channels.

Program the DMA controller.

Write the channel number (1 to 12) + 240 to BASE + 0.

Start the DMA operation by reading BASE + 0.

Auto Channel Scanning

In DMA mode it is possible to output to all 12 channels by setting the AUTO CHANNEL SCANNING bit in channel enable register 2.

When this bit is set, after each DMA transfer the channel number will be incremented so the next DMA transfer will take place on the next channel. When the channel reaches 12, it will continue to increment as if channels 13-16 were present, before resetting to channel 1.

To use this mode, the software must set-up a multi-dimensional array to store the data for 16 channels (12 real + 4 phantom channels).

In BASIC:	DIM dataarray%(numsamples,16)
in C:	int dataarray[numsamples][16]

Software example 4 demonstrates the use of channel scanning.

MAPS AND REGISTERS

Card Address Map

BASE + n	R/W	SECTION	FUNCTION
0	W	DAC	DAC Control register
0	R	DAC	Initiate DMA transfer / reset state machine logic
1	W	DAC	Channel Enable register 1
2	W	DAC	Channel Enable register 2
3	W	DAC	DAC output data
4	R/W	PIO	PIO PORT A data
5	R/W	PIO	PIO PORT B data
6	R/W	PIO	PIO PORT C data
7	R/W	PIO	PIO Control register
8	R/W	Timer	Timer 0 count register
9	R/W	Timer	Timer 1 count register
10	R/W	Timer	Timer 2 count register
11	R/W	Timer	Timer control register

DAC Control Register

(Base + 0)

DATA BIT	FUNCTION
0	DAC Channel Select / update Bit 0 *1
1	DAC Channel Select / update Bit 1 *1
2	DAC Channel Select / update Bit 2 *1
3	DAC Channel Select / update Bit 3 *1
4	0 = Disable DMA Request 1 = Enable DMA Request
5	0 = I/O Mode 1 = DMA Mode
6	0 = Disable DAC control timers (timers 0 and 1) 1 = Enable DAC control timers (timers 0 and 1)
7	0 = Disable uncommitted timer 2 1 = Enable uncommitted timer 2

*1 writing 0 into all four bits updates the previously selected DAC output

Channel Enable Registers

Register 1 (Base + 1)

DATA BIT	FUNCTION
0	0 = Enable channel 1 output 1 = Disable channel 1 output
1	0 = Enable channel 2 output 1 = Disable channel 2 output
2	0 = Enable channel 3 output 1 = Disable channel 3 output
3	0 = Enable channel 4 output 1 = Disable channel 4 output
4	0 = Enable channel 5 output 1 = Disable channel 5 output
5	0 = Enable channel 6 output 1 = Disable channel 6 output
6	0 = Enable channel 7 output 1 = Disable channel 7 output
7	0 = Enable channel 8 output 1 = Disable channel 8 output

Register 2 (Base + 2)

DATA BIT	FUNCTION
0	0 = Enable channel 9 output 1 = Disable channel 9 output
1	0 = Enable channel 10 output 1 = Disable channel 10 output
2	0 = Enable channel 11 output 1 = Disable channel 11 output
3	0 = Enable channel 12 output 1 = Disable channel 12 output
4	0 = Disable AUTOMATIC channel scanning *1 1 = Enable AUTOMATIC channel scanning
5	0 = Enable DMA Cycle mode *2 1 = Disable DMA Cycle mode
6	Not Used, always write 0
7	Not Used, always write 0

*1 When this bit is 1 the selected DAC channel will increment after each DAC update (DMA MODE ONLY)

*2 When this bit is 1 the AUTO-INITIALISE function of the DMA control is disabled (See USING DMA for a description of AUTO-INITIALISE)

SAMPLE PROGRAM DESCRIPTIONS

The disk supplied with the card contains several example programs to demonstrate the various operating modes.

QBASIC and C Examples

EXAMPLE1.BAS
EXAMPLE1.C

These demonstrate the simple I/O mode to output a single value to one analogue output channel.

EXAMPLE2.BAS
EXAMPLE2.C

These demonstrate reading and writing to the digital I/O.

EXAMPLE3.BAS
EXAMPLE3.C

These demonstrate the DMA mode to output a sine wave to a single analogue output channel

EXAMPLE4.BAS
EXAMPLE4.C

These demonstrate the DMA mode to output a sine wave to all twelve analogue output channels.

DETAILED CARD INSTALLATION

Before installing the card into your computer system, there are a number of links which must be set. The settings of these links will depend upon the computer system into which the card is being fitted.

The positions of these links are shown on the card layout diagram towards the end of this manual

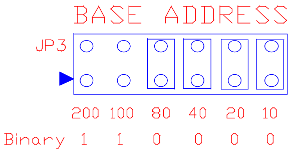
Base Address

The card may be located in any 62 pin slot in the PC motherboard but must be set up to appear at a specified address in the I/O port map. Available positions are shown in the IBM-PC Technical Reference Guide. However, for those who do not possess a copy of this document, a good place is the location normally allocated to the prototyping card as supplied by IBM. This address is 300 $_{\text{Hex}}$ which is the factory default setting.

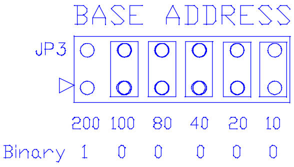
However, no two devices should be used while set to the same address since contention will occur and neither card will work. If your machine contains a card with a conflicting address then another reasonably safe address to use is 200 to 21F $_{\text{Hex}}$.

A set of links are provided on the card to set the base address within the IBM-PC I/O port map. The address is in binary with the presence of a link representing a 0 and the absence of a link representing a 1.

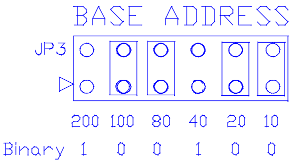
To set the base address to 300 $_{\text{Hex}}$, locate the jumper block JP3 labelled Base Address. Set the following pattern on the links as indicated below with Connector CON2 on right hand side and the gold fingers to the lower edge.:-



Other examples are:



Address hex 200



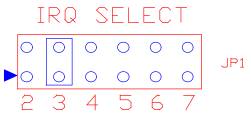
Address hex 240

Interrupts

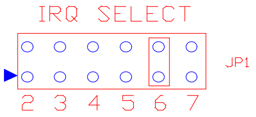
The DAC generates an interrupt signal at the end of a DMA transfer.

The interrupt is selected on jumper block JP1.

The diagram below shows the DAC set to produce an interrupt request on IRQ3.



The second example shows the setting if the DAC generates interrupt request IRQ6.

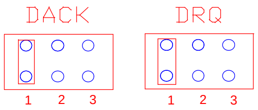


DMA Settings

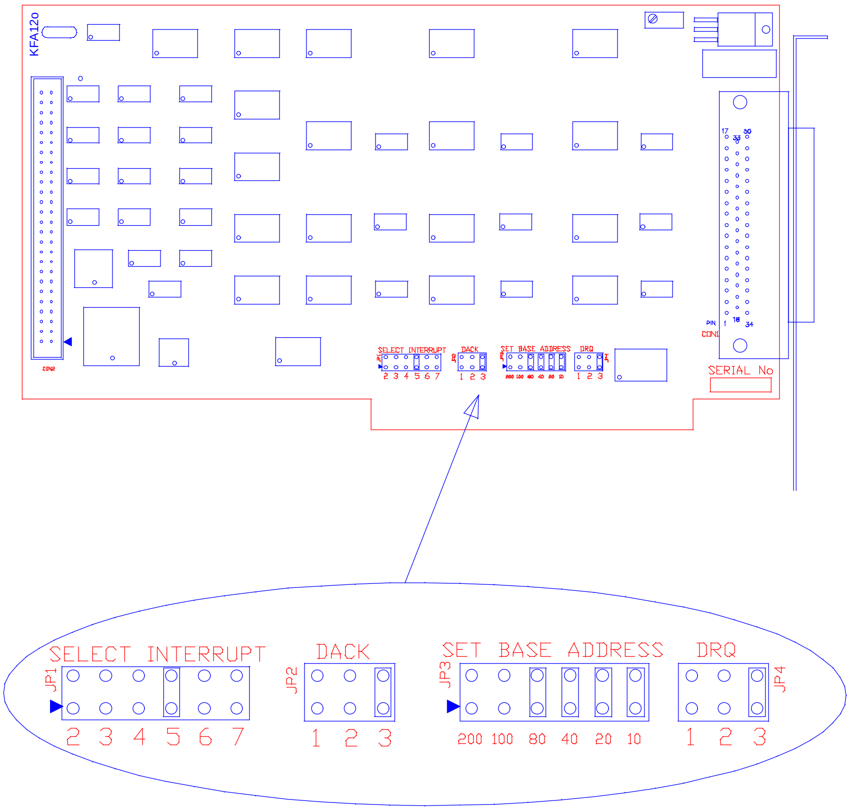
The DMA selection is set on two sets of jumper blocks, JP2 and JP4.

JP4 controls which channel the DAC uses to request Direct Memory Access. Only channels 1,2 and 3 are available. Jumper JP2 sets the channel on which the DMA controller acknowledges the request. It is essential that the pattern of links on the two jumper blocks correspond.

The example below shows the link settings for the DAC to generate DRQ1 and receive DACK1.



Card Layout Diagram



Card Layout showing Selector Link Positions

APPENDIX A - NUMBERING SYSTEMS

Binary and Hexadecimal Numbers

The normal numbering system is termed DECIMAL because there are ten possible digits (0 to 9) in any single column of numbers. Decimal numbers are also referred to as numbers having a Base 10. When counting, the numbers increment in the units column from 0 up to 9. The next increment resets the units column to 0 and carries over 1 into the next column. This 1 indicates that there has been a full ten (the base number) counts in the units column. The second column is therefore termed the “tens” column.

It is more convenient when programming to use a number system that provides a clearer picture of the hardware at an operational or register level. The two most common number systems used are BINARY and HEXADECIMAL. These two systems provide an alternative representation to decimal numbers.

For a binary number there are only 2 possible values (0 or 1) and as a result binary numbering is often known as Base 2. When counting in binary numbers, the number increments the units column from 0 to 1. At the next increment the units column is reset to 0 and 1 is carried over to the next column. This column indicates that a full two counts have occurred in the units column. Now the second column is termed the “twos” column.

Hexadecimal numbers may have 16 values (0 to 9 followed by the letters A to F). It is also known as a system with the Base 16. With this counting system the units increment from 0 to 9 as with the decimal system, but at the next count the units column increments from 9 to A and then B, C and so on up to F. After F the units column resets to 0 and the next column increments from 0 to 1. This 1 indicates that sixteen counts have occurred in the units column. The second column is termed the “sixteen’s” column.

The following table shows how the three systems indicate successive numbers

Decimal
Base 10

0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
1	0
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
2	0

Binary
Base 2

0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0

Hexadecimal
Base 16

0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
0	A
0	B
0	C
0	D
0	E
0	F
1	0
1	1
1	2
1	3
1	4

Notice how the next higher column does not increment until the lesser one to its right has overflowed.

Binary representation is ideally suited where a visual representation of a computer register or data is needed. Each column is termed a BIT (from **B**inary dig**IT**). Only five Bits are shown in the above table. With larger numbers, more Bits are required. Normally Bits are arranged in groups of eight termed BYTES. By definition there are 8 BITS per BYTE. Each Bit (or column) has a value. In the binary table above the rightmost or least significant column each digit has a value of 1. Each digit in the next column has a value of 2, the next 4, then 8 and so on.

The following diagram illustrates this.

BIT No	7	6	5	4	3	2	1	0
DECIMAL VALUE	128	64	32	16	8	4	2	1

To determine the decimal value of a binary pattern, add up the decimal number of each column containing a binary “1”.

BIT No	7	6	5	4	3	2	1	0
DECIMAL VALUE	128	64	32	16	8	4	2	1
BINARY NUMBER	1	1	0	0	0	1	1	0

The above example shows the binary pattern that is equivalent to 198 $_{\text{Decimal}}$.

The binary string defining a Byte can be unwieldy. To make it less error prone, the 8 bits forming a byte are divided into two groups of 4 bits, known as NIBBLES. With four bits there are 16 possible numeric combinations (including zero). A convenient method of representing each nibble is to use the hexadecimal base 16 system.

When converting binary to hex, the byte is divided into nibbles each represented by a single hex digit. This technique is applied to the selection of the base address for the circuit board. The following diagram illustrates the construction of a hex number.

BIT No	7	6	5	4	3	2	1	0
NIBBLE VALUE	8	4	2	1	8	4	2	1
BINARY NUMBER	1	1	0	0	0	1	1	0

ÀÄÄÄ-ÄÂÄÄÄÄÄÙÀÄÄÄÄÂÄÄÄÄÙ

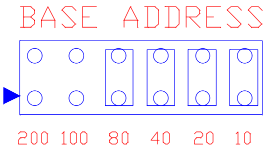
HEXADECIMAL: C 6

Hexadecimal upper nibble = (1 x 8) + (1 x 4) + (0 x 2) + (0 x 1) = 12
lower nibble = (0 x 8) + (1 x 4) + (1 x 2) + (0 x 1) = 6

The resulting value is C6 $_{\text{Hex}}$, since 12 $_{\text{Decimal}}$ is C $_{\text{Hex}}$.

Base Address Selection

Each column can be physically represented on the board by a pair of pins. In practice, the boards cover a range of addresses (usually 16 $_{\text{Decimal}}$). Therefore the low order four bits are not included, but two higher order bits are added. This gives an address range of 0 to 3F0 $_{\text{Hex}}$. The following diagram shows a typical set of pins.



Here a link is fitted to denote a binary or logic "0", or left open to indicate a binary or logic "1". The example shows a base address setting of 300 $_{\text{Hex}}$.

APPENDIX B - PC MAPS

PC XT/AT I/O Address Map

Address	Allocated to:
000-01F	DMA Controller 1 (8237A-5)
020-03F	Interrupt Controller 1 (8259A)
040-05F	Timer (8254)
060-06F	Keyboard Controller (8742) Control Port B
070-07F	RTC and CMOS RAM, NMI Mask (Write)
080-09F	DMA Page Register (Memory Mapper)
0A0-0BF	Interrupt Controller 2 (8259)
0F0	Clear NPX (80287) Busy
0F1	Reset NPX (80287)
0F8-0FF	Numeric Processor Extension (80287)
1F0-1F8	Hard Disk Drive Controller
200-207	Reserved
278-27F	Reserved for Parallel Printer Port 2
2F8-2FF	Reserved for Serial Port 2
300-31F	Reserved
360-36F	Reserved
378-37F	Parallel Printer Port 1
380-38F	Reserved for SDLC Communications, Bisync 2
3A0-3AF	Reserved for Bisync 1
3B0-3BF	Reserved
3C0-3CF	Reserved
3D0-3DF	Display Controller
3F0-3F7	Diskette Drive Controller
3F8-3FF	Serial Port 1

PC XT Interrupt Map

Number	Allocated to:
NMI	Parity
0	Timer
1	Keyboard
2	Reserved
3	Asynchronous Communications (Secondary) SDLC Communications
4	Asynchronous Communications (Primary) SDLC Communications
5	Fixed Disk
6	Diskette
7	Parallel Printer

PC AT Interrupt Map

Level		Allocated to:
CPU NMI		Parity or I/O Channel Check
Ctlr 1	Ctlr 2	(Interrupt Controllers)
IRQ 0		Timer Output 0
IRQ 1		Keyboard (Output Buffer Full)
IRQ 2		Interrupt from CTLR 2
	IRQ 8	Real-time Clock Interrupt
	IRQ 9	S/w Redirected to INT 0AH (IRQ 2)
	IRQ 10	Reserved
	IRQ 11	Reserved
	IRQ 12	Reserved
	IRQ 13	Co.-processor
	IRQ 14	Fixed Disk Controller
	IRQ 15	Reserved
IRQ 3		Serial Port 2
IRQ 4		Serial Port 1
IRQ 5		Parallel Port 2
IRQ 6		Diskette Controller
IRQ 7		Parallel Port 1

DMA Channels

0	Memory Refresh
1	Spare
2	Floppy Disk Drive
3	Spare

APPENDIX C - USING DMA

Direct Memory Access or DMA is a process by which data can be transferred directly from the memory of the PC into an I/O card or directly from the I/O card into the PC memory with no intervention from the processor. This can greatly increase the throughput of data and at the same time, reduce the overhead of processor time.

The DMA Controller

DMA is controlled by the PC using one of two DMA Controllers. The DMA Controllers are INTEL 8237 or compatible devices, each containing four channels. The first one is used for byte transfers in the bottom 1 MB of system memory, the second can transfer words into the bottom 16 MB.

Blue Chip Technology boards only allow DMA channels 1 or 3 on the first controller to be used. Normally channel 0 is reserved for memory refresh control and channel 2 is used by the floppy disk drives.

In order to begin a DMA transfer, first the I/O board must be configured to enable DMA operation - consult the relevant section of the manual on how to do this. Secondly, the DMA controller must be programmed to begin the transfer. The DMA controller is programmed by writing to I/O ports in much the same way as the card is programmed.

These port locations are fixed for all PCs as follows:-

I/O ADDRESS	READ/ WRITE	DESCRIPTION
0000H	R/W	DMA Channel 0 Current Address
0001H	R/W	DMA Channel 0 Current Word Count
0002H	R/W	DMA Channel 1 Current Address
0003H	R/W	DMA Channel 1 Current Word Count
0004H	R/W	DMA Channel 2 Current Address
0005H	R/W	DMA Channel 2 Current Word Count
0006H	R/W	DMA Channel 3 Current Address
0007H	R/W	DMA Channel 3 Current Word Count
0008H	R/W	Command/Status Register
0009H	R/W	DMA Request Register
000AH	R/W	DMA Single Bit Mask Register
000BH	R/W	DMA Mode Register
000CH	R/W	DMA Clear Byte Pointer
000DH	R/W	DMA Master Clear
000EH	R/W	Clear Mask Register
000FH	R/W	DMA Write All Mask Register Bit
0081H	R/W	Page Register DMA Channel 2
0082H	R/W	Page Register DMA Channel 3
0083H	R/W	Page Register DMA Channel 1

The DMA Controller Registers

In order to begin a DMA transfer there are several registers within the DMA controller which need to be configured. The relevant registers are described below:-

Mode Register Port 0BH

7	6	5	4	3	2	1	0
MODE Bit 1	MODE Bit 0	AUTO INC/ DEC	AUTO INIT.	TRANS MODE Bit 1	TRANS MODE Bit 0	CHAN SEL Bit 1	CHAN SEL Bit 0

Mode Bits

Bit1	Bit 0	FUNCTION
0	0	Demand Mode
0	1	Single Mode
1	0	Block Mode
1	1	Not Used

The mode bits set the particular mode for the channel, normally this will be set for SINGLE mode.

Auto INC/DEC

0	Increment Address
1	Decrement Address

When a DMA transfer takes place, the transfer address can either be incremented or decremented after each transfer selected by this bit.

Auto Initialise

0	Disabled
1	Enabled

If set to AUTO INITIALISE, when the DMA transfer reaches the end of a block, the DMA controller will reload all its initial values and repeat the transfer. This is useful on ANALOGUE OUT boards for outputting continuous wave forms.

Transfer Mode

Bit 1	Bit 0	FUNCTION
0	0	Verify Transfer
0	1	Write Transfer
1	0	Read Transfer
1	1	Not Used

Use WRITE TRANSFER if the I/O board is generating a value to be written into memory (Analogue in), use read transfer when values are written from memory into the board (Analogue out).

Channel Select

Bit 1	Bit0	FUNCTION
0	0	DMA Channel 0 select
0	1	DMA Channel 1 select
1	0	DMA Channel 2 select
1	1	DMA Channel 3 select

Mask Register Port 0AH

7 NOT USED	6 NOT USED	5 NOT USED	4 NOT USED	3 NOT USED	2 CLR/SET MASK	1 MASK Bit 1	0 MASK Bit 0

CLR/SET Mask

0	Clear Mask Bit
1	Set Mask Bit

Mask

Bit 1	Bit 0	FUNCTION
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Setting a mask bit for a particular channel disables the DMA operation on that channel.

Status Register Port 0BH READ

7 CHAN 3 DMA RQ	6 CHAN 2 DMA RQ	5 CHAN 1 DMA RQ	4 CHAN 0 DMA RQ	3 CHAN 3 at TC	2 CHAN 2 at TC	1 CHAN 1 at TC	0 CHAN 0 at TC

DMA RQ	1 = DMA Request made on channel N
AT TC	1 = Channel N has reached Terminal Count (TC)

The status register indicates which channels have made a DMA request and which channels have reached Terminal Count. Terminal count is set when the number of bytes specified by the TRANSFER LENGTH register have been transferred.

Clear Byte Pointer Flip Flop Port 0CH

Any write to this port resets the byte pointer flip flop. This ensures that the first write to the Start Address or Transfer Length registers will go into the LSB of that register.

DMA Transfer Start Address Ports 00H, 02H, 04H, 06H

Sets the 16 bit start address for the DMA transfer, send LS Byte first.

Transfer Length Ports 01H, 03H, 05H, 07H

Sets the 16 bit length of the DMA transfer, send LS Byte first.

Page Register Ports 83H, 81H, 82H

Sets the upper eight bits of the physical memory address of the DMA transfer.

Addressing

In order for DMA to operate correctly the page register and start address register should be set-up to inform the DMA controller where in memory the transfer is to take place.

The INTEL x86 family of microprocessors address memory using 2 address pointers called SEGMENT and OFFSET, this is called a LOGICAL address. The microprocessor combines the SEGMENT and OFFSET to produce a PHYSICAL address which it uses to access the memory.

If we consider a PC which has 1Mbyte of memory, each memory location will have a PHYSICAL address of 0 to 1048575 (or 0 to $\text{FFFFF}_{\text{Hex}}$).

It has a LOGICAL range of 0000:0000 to F000:FFFF. The colon is commonly used to separate the SEGMENT address from the OFFSET address (Segment:Offset)

To convert from a LOGICAL address to a PHYSICAL address use the following formula:

PHYSICAL ADDRESS = (SEGMENT * 16) + OFFSET

Most programming languages allow access to the SEGMENT and OFFSET addresses of variables in memory. For example, in QUICK BASIC the following commands can be used:

```
DIM dat%(1000)
seg = VARSEG(dat%)
offs = VARPTR(dat%)
```

In this example, the variables “seg” and “offs” would contain the SEGMENT and OFFSET addresses of the array “dat%”. In order to pass this address to the DMA controller the segment and offset values need to be converted into DMAPAGE and DMAOFFSET addresses as follows:

```
PhyAdd = (seg * 16) + offs
DMAPAGE = (PhyAdd / 65536) AND 15
DMAOFFSET = PhyAdd AND 65535
```

DMA Limitations

The DMA controller is only capable of incrementing or decrementing the DMAOFFSET address, the DMAPAGE value is fixed throughout the transfer. This means that a maximum of 64K bytes can be transferred in one operation.

Programming Example

To set-up a DMA transfer the following program sequence is required:

> Define an area of memory for the transfer
>
> Set-up the I/O board for DMA operation
>
> Disable the DMA channel being used.
>
> Load the start address into PAGE register. This is the START address register for the DMA channel being used.
>
> Load the length count into the TRANSFER LENGTH register. Note that the DMA controller only transfers 8 bits at a time, each value written to an ANALOGUE OUT board or read from an ANALOGUE IN board is 2 bytes long so the transfer length will be twice the number of samples to be taken.
>
> Load the mode for the selected DMA channel.
>
> Enable the DMA channel.

The following extract from a QUICK BASIC program demonstrates how to program the DMA controller for a WRITE transfer.

```
2000 REM PROGRAM THE DMA CONTROLLER
2005 REM FIRST EXTRACT THE SEGMENT AND OFFSET
      ADDRESS OF OUR DATA
2010 seg = VARSEG(DAT%(0))
2020 offs = VARPTR(DAT%(0))
2025 REM Transfer the Logical SEGMENT:OFFSET address
2027 REM into a physical PAGE:OFFSET address
2030 PAGE& = seg1% AND &HF000
2040 PAGE& = PAGE& / 4096
2050 PAGE& = PAGE& AND 15
2060 OFFSET& = seg
2070 OFFSET& = OFFSET& * 16
2080 OFFSET& = OFFSET& + offs
2090 OFFSET& = OFFSET& AND 65535
2100 REM Setup the DMA registers.
2105 OUT (&HA),7: REM DISABLE DMA CHANNEL 3
2110 OUT (&HC), 0: REM RESET BYTE SELECT FF
2120 OUT (&HB), &H47: REM AUTO INC ON CH3,WRITE
      TRANSFER, SINGLE MODE
2130 OUT (&H82), PAGE&: REM SET PAGE FOR DMA CH3
2150 OUT (&H6), (OFFSET& AND 255): REM SET UP START
      FOR LS 8 BITS
2160 OUT (&H6), (OFFSET& AND &HFF00) / 256: REM SET
      UP START FOR MS 8 BITS
2170 OUT (&H7), length% AND 255: REM BYTE COUNT LS 8
      BITS
2180 OUT (&H7), length% / 256: REM BYTE COUNT MS
      BITS
2190 OUT (&HA), 3: REM ENABLE DMA TXFER
2200 RETURN
```