

# Qontrol-II for the QL

Qontrol-II is an extremely useful QL interface board for control applications. On board there is a powerful 6522 Versatile Interface Adapter (VIA) and space for an 8-channel fully-automatic fast A/D device.

The 6522 VIA supplies you with 16 programmable data lines, configurable handshaking, two versatile 16-bit timers and shift register operations. The VIA may be programmed for use in either polling or interrupt modes.

SuperBASIC extensions are available within the board ROM to permit control programs to be written in a high level language. Those people who like to program in assembler need have no fears however. The ROM also contains a set of link vectors to provide just as much capability.

The A/D chip that is an option for the board is the Analog Devices 7581-JN. Any of the 8 available channels may be independently read at more than 1,500 samples/second. The channels are internally sampled in rotation and therefore, by tying an input signal to all 8 channels, a maximum sample rate of 12,500 samples/second can be achieved.

Within the SuperBASIC extensions there are a number of routines dedicated to the access of this A/D device.

The ease of programming control applications using the ROM extensions can be seen by a few examples.

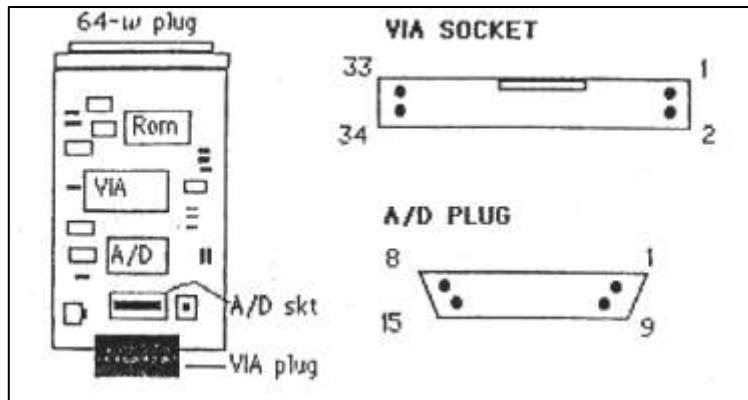
For instance, you can sample up to 8 digital input lines at 1000 samples/sec. and display the readings graphically (either as a once off or continuously) using just **ONE** command - LOGICANAL

DIGISCOPE is the analogue equivalent and can display up to 4 channels at a time. You can test and set individual bits in a port (using TSTBIT and SETBIT), collect and send whole bytes (RDVIA, WRVIA), count a set number of pulses or a variable number of pulses under bit control (COUNT, WCOUNT), create programmable mark-space ratio waveforms for testing/controlling other equipment and so on.

If you create a good display of some experimental data, you can of course obtain a hardcopy printout of the screen (in either resolution) using the extension PRTDUMP.

All in all, QONTROL-II is not just a control board. It is the basis for as good an experimental or control application program as you care to write!

The board comes with a small manual detailing the extensions and machine code link vectors available and the hardware of the board. Details are given on how to add an external board to QONTROL-II to turn it into a BBC compatible User-port and Parallel printer port. The A/D channels can also be switched to emulate the BBC Analogue input connector.



## **FACILITIES**

### **SuperBASIC Functions**

x=AD\_BASE  
 x=PIA\_BASE  
 x=RDVIA (port)  
 x=TSTBIT (port,bit)  
 x=WCOUNT(control,level)

### **SuperBASIC Procedures**

QHELP [#n]  
 ADGRAB chan,samples,buffer  
 DIGISCOPE waves,chans,source,delay  
 PORTSET dataport,iobyte  
 GRAB dataport,samples,buffer  
 WRVIA port,byte  
 SETBIT port,bit,level  
 COUNT pulses  
 FREERUN delay,latch,go  
 LOGICANAL waves,chans,source,delay  
 PRTDUMP #n,resol,lflag

### **ADDITIONAL DEVICES**

ATODn            [n=0 to 7]  
 PORTn-w        [n=0(B),1(A) w=iobyte]  
 PAR-wl         [w=0 to 255 (width) l=LF flag]

The extensions provided permit easy and powerful programming within the control environment. The ROM also contains machine code links.

**PRICE: £89.95 +VAT**

CARE Electronics, 800 St. Albans Rd., Garston, Watford, Herts. WD2 6NL  
 Tel (0923) 672102

## **QONTROL-II INTERFACE BOARD**

Thank you for buying the Qontrol-II interface board - the first QL expansion board dedicated for use in control applications. We are sure you will find it immensely useful, whatever your application may be.

Apart from the facilities described in this small document there will be support for the Qontrol-II board through information in QL WORLD and from CARE Electronics. The PCB mentioned later, for turning your Qontrol-II ports into BBC micro compatible connections will soon be available and a FAX/RTTY board and software package is already being designed and written. So remember to watch the adverts and read the magazine. Of course if you have a particularly popular application design of your own, why not tell QL WORLD about it and get your name in print?

### **INTRODUCTION**

The board contains three major chips. A ROM containing all the support software, a 6522 VIA (Versatile Interface Adapter) for all digital I/O work and space for a 7581-JN A/D Convertor for all your analogue input applications. The latter is a fairly expensive (though powerful) chip and as such is not supplied as standard. This keeps the cost down for those just wishing to perform digital i/o. You may fit your own 7581 A/D chip at a later date or ask Care Electronics for the cost of fitting.

The ROM contains sixteen extensions to SuperBASIC to enable all you BASIC programmers to write your control applications. A set of vectors also exist for machine code programmers to obtain the same facilities. In addition to the extra commands, there are three new devices that can be used when manipulating QDOS channels.

The 6522 VIA supports two programmable 8-bit data ports, 4 communication control lines, an 8-bit shift register and two 16-bit counter-timers. It is not the aim of this document to fully describe this chip as it would take up too much space! For those of you who are willing and able to program this type of chip the information is readily available in datasheets or books on 6502/68000 programming and interfacing. Everyone else will be more than supported by the facilities within the Qontrol-II ROM.

The 7581 A/D chip contains 8 channels that are sampled automatically in strict rotation. Each channel can sample a signal at a rate of over 1,500 samples/second. If you were to tie an analogue signal to all 8 input lines you could sample at a maximum of 12,500 samples - /second.

### **1. ROM COMMANDS**

There are 16 additional SuperBASIC procedures and functions and 3 additional devices.

## 1.1 FUNCTIONS

### **x=AD\_BASE**

Returns the base address of the A/D chip. Each of the eight channels is then an offset (0..7) from this address. To read a channel you could use the command `val=PEEK(ADBASE+chan)`

### **x=PIA\_BASE**

Returns the base address of the 6522 VIA chip (sometimes called a PIA chip). The chip has sixteen internal registers that can be read or written to:

Each of these registers is at an offset address (0..15) from the base address. You can read and write to any of these registers using the commands

```
val=PEEK(PIA_BASE+reg) and  
val=POKE PIA_BASE+reg,val
```

However, for simple I/O operations it is recommended that you use the `PORTSET`, `RDVIA`, `WRVIA`, `TSTBIT` and `SETBIT` language extensions instead.

### **x=RDVIA(port)**

This requires a VIA port address in the range 0..15. It will return the byte value of the contents of that register. For simple I/O operations the only ports (registers) you need to know about are 0 (data port B) and 1 (data port A). Any data port bits that have been assigned as outputs (see `PORTSET`) will simply return the last bit value written to that port.

### **X=TSTBIT (port,bit)**

This requires a port (register) address in the range 0..15 and a bit value in the range 0..7. It will test the state of that bit and return the value 0 if it is reset or 1 if it is set. For simple I/O operations the only ports you need to know about are 0 (data port B) and 1 (data port A).

### **z=WCOUNT(contro],level)**

This will count pulses (actually high to low transitions) coming in on data port B, bit 7. The count will start/stop depending on the level (0 or 1) of the control bit (0..6). For example, if you give the command `x=WCOUNT(4,1)` the count will not start until bit 4 of port B is at logical level 1. (high). Once started the count will stop when bit 4 becomes reset (low).

## 1.2 PROCEDURES

### **QHELP {#n}**

Provides a quick help list of the facilities available within the Control-II ROM. Invaluable! The specified channel number is optional, and channel #1 will be used by default.

**ADGRAB chan,samples,buffer**

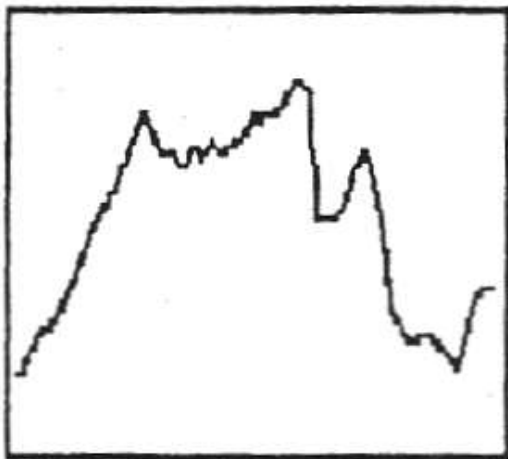
Sampling an A/D channel within SuperBASIC using the PEEK(AD\_BASE+chan) method may not be quick enough for some applications. ADGRAB provides a fast (1000 samples/sec) method for grabbing samples. The chan number must be between 0 and 7. The routine will grab the specified number of samples, placing the results in the specified buffer. Warning: it is the programmer's responsibility (eg. through respr) to ensure that the buffer space is both reserved and big enough. For example, to collect 2000 samples (two seconds worth) of a signal on A/D port 2 the following could be used:

```
buf=RESPR(2000)
ADGRAB 2,2000,buf
```

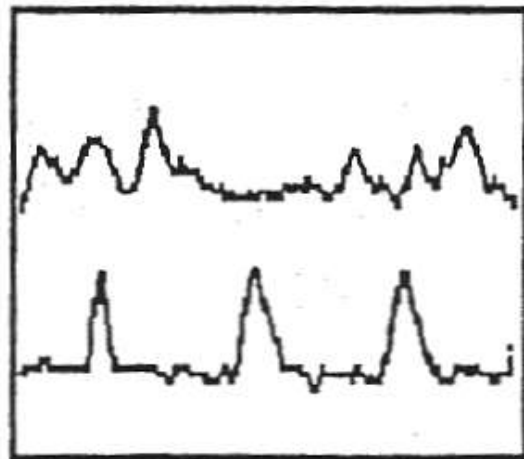
The Qontrol-II ROM extension DIGISCOPE could then be used to display the readings in blocks of 400 samples.

**DIGISCOPE waves,chans,source,delay**

Up to 4 analogue waveforms can be displayed using this command. The display is adjusted and waveforms scaled depending on the number of channels requested. For example:



**Single channel**



**Two channel**

In multi-channel displays the channels requested are displayed with the lowest channel number at the top and in order of increasing channel number. The channel numbers need not be contiguous though of course the displays will be adjacent. Channel numbers are not displayed automatically and must therefore be added by your own program if this is required. Scaling and sample timing is also not included. This gives maximum versatility to the applications software. Maximum and minimum delimiter lines are displayed for each channel.

**Waves** specifies the number of channels required (1..4). **Chans** is the channel definition byte (0..255). Each bit in the byte represents its corresponding channel. If the bit is set, the channel will be

selected for display. For example to monitor 3 channels (0, 3 and 6) the definition byte would be  $2^0 + 2^3 + 2^6 = 1+8+64 = 73$ . It is an error if the number of bits set in **chans** does not correspond to the number of specified **waves**. **Source** should be set to 0 if you want the A/D channels to be sampled as requested. A higher value will force DIGISCOPE to display the contents of the buffer at address source. In the latter case the number of waves will be reduced to one automatically, irrespective of your setting. Delay is the number of 1/50 second that you want to delay between successive samples. The delay specified will be ignored if a buffer is being displayed. At the end of each scan you can press **ENTER** to perform another scan. **SPACE** to repeat scanning until another keypress, or **ESC** to exit from **DIGISCOPE**.

#### **PORTSET dataport, iobyte**

This command can be used to set up either of the two 8-bit data ports (A or B). Each bit in the port byte may be configured as an input or an output. The dataport must be either 0 (port B) or 1 (port A). The iobyte (0..255) should have the corresponding bit set (1) if the line is to be an input, or reset (0) if an output. For example to set port A with bits 0..3 as inputs and 4..7 as outputs, the iobyte would be given by:

```
bit 7 6 5 4 3 2 1 0
      0 0 0 0 1 1 1 1 = 15
```

and the command would be **PORTSET (1,15)**.

#### **GRAB dataport, samples,buffer**

Sampling a data port within SuperBASIC using the **PEEK(PIA\_BASE+chan)** method may not be quick enough for some applications. **GRAB** provides a fast (1000 samples/sec) method for grabbing samples. The dataport number must be either 0 (port B) or 1 (port A). The routine will grab the specified number of samples, placing the byte results in the specified buffer. Warning: it is the programmers responsibility (eq. through respr) to ensure that the buffer space is both reserved and big enough. For example, to collect 2000 samples (two seconds worth) of signals on port B the following could be used:

```
but =RESPR(2000)
GRAB 0,2000,buf
```

#### **WRVIA portbyte**

This will write the specified byte value to the appropriate VIA port (address offset 0..15). If a data port is written to pins designated as input lines will not be affected.

#### **SETBIT port,bit,level**

This will set or reset any bit in any VIA register without affecting any others in the byte. The port address must be between 0..15, the bit number between 0..7 and the level either 0 to reset the bit or 1 to set it.

#### **COUNT pulses**

This will wait until a specified number of pulses have been read on bit 7 of data port B. (In practice it is the number of high-low transitions that is counted).

### **FREERUN delay,latch,go**

This is a rather special command that enables you to create internal free running time delays with an option to output a digital waveform with a corresponding mark:space ratio. Some experience of the workings of the 6522 VIA would be helpful to fully understand what is going on, but the following information will allow you to use the facility.

The 6522 VIA 16-bit Timer-1 is used in its free-running mode. The count-down and time-out flags are affected depending on the delay specified (in microseconds). If go is 0, no output on bit 7 of data port B occurs. If go is 1, the output level of this bit 7 toggles (changes state) on each time-out. If latch is 1, the timer will free-run and each time-out will be delay microseconds long. If go is 1, the pulse output will have an even mark:space ratio. If we use FREERUN with latch equal to 0, the delay value will be entered into the timer latches but no effect will occur on the current timeout. This means that the delay can be altered between successive timeouts. If go is 1, the pulse output could then have a variable mark:space ratio. Consider two examples.

To obtain a continuous pulse stream output on PB7 with a frequency of 50Hz a toggle must occur every 10ms (10,000 usec). The command: FREERUN 10000,1,1

would therefore do the job. To set up a pulse train at say, 25Hz. with a mark:space ratio of 1:4 we would need to monitor the timeout flag for timer-1 in the VIA interrupt flag register (bit 6 of port address 13). This bit becomes set on every timeout and is reset when read. Provided bit 7 of port B was set high prior to entry, the following program would perform the task:

```
10 FREERUN 8000,1,1           ; Set/run mark
20 REPEAT i
30   FREERUN 32000,0,1       ; Set space
40   IF TSTBIT(13,6)<>1 : GOTO 40 ; Wait
50 FREERUN 8000,0,1         ; Set mark
60 IF TSTBIT(13,6)<>1 : GOTO 60   ; Wait
70 END REPEAT i
```

The SuperBASIC version will be relatively limited in terms of speed in this latter example because of its interpretation speed. However, in assembler much faster signals can be programmed.

### **LOGICANAL waves,chans,source,delay**

This is the digital version of DIGISCOPE and the channels requested are displayed with the lowest channel number at the top and in order of increasing channel number. The major difference is that the display will always be arranged for the maximum number of waves possible (eight), though not all wave levels need be displayed. The channel numbers selected (0..7) need not be contiguous and of course the displays may not be adjacent. Channel numbers are not displayed automatically and must therefore be added by your own program if this is required. Sample timing is also not included. This gives maximum versatility to the applications software. Maximum and minimum delimiter lines are displayed for each channel.

Waves specifies the number of channels required (1..8). Chans is the channel definition byte (0...255). Each bit in the byte represents its corresponding port bit. If the bit is set, the port bit will be selected for display. For example to monitor 3 bits (0, 3 and 6) the definition byte would be  $2^0 + 2^3 + 2^6 = 1+8+64 = 73$ .

It is an error if the number of bits set in chans does not correspond to the number of specified waves. Source should be set to 0 if you want the port B byte to be sampled as requested, or 1 if you want the port A byte. A higher value will force LOGICANAL to display the contents of the buffer at address source. In the latter case the number of waves will be still be set by waves and chans. Delay is the number of 1/100 second that you want to delay between successive samples. The delay specified will be ignored if a buffer is being displayed. At the end of each scan the key ESC can be pressed to exit from logicanal, ENTER to perform another scan, or SPACE to repeat scanning until another key is pressed.

#### **PRTDUMP #n,resol,flag**

This procedure will dump a copy of the screen to any EPSON compatible graphics printer. It is the programmer's responsibility to ensure that the specified channel is in fact a printer channel. Resol should be set to 0 if the screen is in low-resolution (8 colour mode) or 1 if in high-resolution mode (4 colour mode). lflag should be set to 0 if your printer does not need a LF character after a CR, or set to 1 if a LF character is required.

### **1.3 ADDITIONAL DEVICES**

#### **ATODn**

Any of the eight A/D ports may be opened as an individual channel with the name ATOD. The channel number 'n' must be between 0..7. The channel can only be opened for input.

#### **PORTn\_w**

Either of the two VIA data ports may be opened as a channel. The channel number 'n' must be 0 for port B or 1 for port A. The value 'w' is the iobyte value for setting the bit input/output designation (see PQRSET for more information).

#### **PAR\_w1**

This permits the opening of a parallel printer port attached to VIA data port A. The BBC micro compatible connection board (or an equivalent just for the printer part) would have to be attached to make this parallel printer port driver work properly (see later).

## **2. ASSEMBLER LINK VECTORS AND CALLS**

A number of link vectors exist within the ROM to enable assembly language programmers to access the facilities described above. These routines will clearly be of use to make control applications easier to write at assembly level.



The table starts at offset \$30 from the start of the ROM. For a board inserted into expansion slot 0 (see QL World Connexion series or QL Technical manual), the absolute table address is therefore \$C0030.

## 2.1 VECTOR ACCESSING

The method of accessing these vectors is to set up any registers as required and perform a JSR vector. On exit from the routine D0.L will contain any error code (standard QDOS codes). For example, to find the base address of the VIA chip, the following could be used:

```

move.l #$C0000,a4 ;board base
jsr    $34(a4)
                ;address in D1.L

```

All registers except **A7** may be affected by the called routine.

<u>Vector offset</u>	<u>Use</u>
\$30	Returns AD_BASE
\$34	Returns PIA_BASE
\$38	Returns RDVIA byte
\$3C	Returns TSTBIT result
\$40	Returns WCOUNT total
\$44	Performs ADGRAB
\$48	Performs DIGISCOPE
\$4C	Performs PORTSET
\$50	Performs GRAB
\$54	Performs WRVIA
\$58	Performs SETBIT
\$5C	Performs COUNT
\$60	Performs FREERUN
\$64	Performs LOGICANAL
\$68	Performs PRTDUMP

## 2.2 PARAMETER PASSING

The parameters passed to and from the individual vectored calls are a detailed below. It is the programmer's responsibility to ensure that these are correct. No checking is performed other than on port values etc. In certain cases the QL will complain bitterly if you fail to obey the rules !

### FUNCTIONS

```

adbase ($30).
  Entry: none           Exit: A2.L base address

piabase ($34)
  Entry: none           Exit: A2.L base address

rdvia ($38)
  Entry: D1.B port offset  Exit: DIB port value

```

tstbit (\$3C)  
Entry: **A1.L** ptr to table      Exit: D1.B bit level  
      A6.L zero (0)

Table pointed to by A1.L has two word entries:  
      0(A1) - port offset  
      2(A1) - bit number

wcount (\$40)  
Entry: A1.L ptr to table      Exit: DIV count total  
      A6.L zero (0)  
Table pointed to be ALL has two word entries:  
      **0(M)** - control bit number  
      **2(M)** control bit level

### PROCEDURES

#### **adgrab (\$44)**

Entry: A2.L absolute addr. Exit: none  
      of A/D chan.  
      A4.L buffer address  
      D4.L sample count

The absolute address of an A/D channel can be obtained by using  
adbase and adding the offset 0..7.

#### **digiscope (\$48)**

Entry: A1.L ptr to table      Exit: none  
      A6.L zero (0)  
Table pointed to by ALL has four long-word entries:  
      \$0(A1) - number of waves  
      **\$4(A1)** - channel selection byte  
      \$8(A1) - source  
      **\$C(A1)** - delay in 1/50 second

#### **portset (\$4C)**

Entry: D1.W port offset      Exit: none  
      D2.W iobyte

#### **grab (\$50)**

Entry: A2.L absolute addr. Exit: none  
      of VIA chan.  
      A4.L buffer address  
      D4.L sample count

The absolute address of a VIA data port can be obtained by using  
piabase and adding the offset 0..1. Remember that 0 is port B and 1  
is port A.

#### **vrvia (\$54)**

Entry: D1.L port offset      Exit: none  
      D2.B byte to write

#### **setbit (\$58)**

Entry: A1.L ptr to table      Exit: none

A6.L zero (0)

Table pointed to by A1.L has three word entries:

0(A1) - port offset  
2(A1) - bit number  
4(A1) - level

#### **count (\$5C)**

Entry: A1.L ptr to table      Exit: none

A6.L zero (0)

Table pointed to by A1.L has one word entry:

0(A1) - number of pulses

#### **freerun (\$60)**

Entry: A1.L ptr to table      Exit: none

A6.L zero (0)

Table pointed to by A1.L has three word entries:

0(A1) - delay in microseconds  
2(A1) - latch flag  
4(A1) - PB7 **o/p** enable flag

#### **logicanal (\$64)**

Entry: A1.L ptr to table      Exit: none

A6.L zero (0)

Table pointed to by A1.L has four long-word entries:

\$0(A1) - number of waves  
\$4(A1) - channel selection byte  
\$8(A1) - source  
\$C(A1) - delay in 1/100 second

#### **prtdump (\$68)**

Entry: A1.L channel ID      Exit: none

A6.L ptr to block

The block pointed to by A6.L must be an area in RAM and be of the following form:

\$00(A6) - (long) reserved  
\$04(A6) - (long) reserved  
\$08(A6) - (long) reserved  
\$0C(A6) - (long) reserved  
\$10(A6) - (byte) resolution flag  
\$11(A6) - (byte) line-feed flag

### **2.3 DEVICE ACCESS**

The three devices are automatically set up when the QL is reset. As such they become recognised as valid QDOS devices. To, for example, open one of these devices you would still use the TRAP#2 IO.OPEN (\$01) call to QDOS.

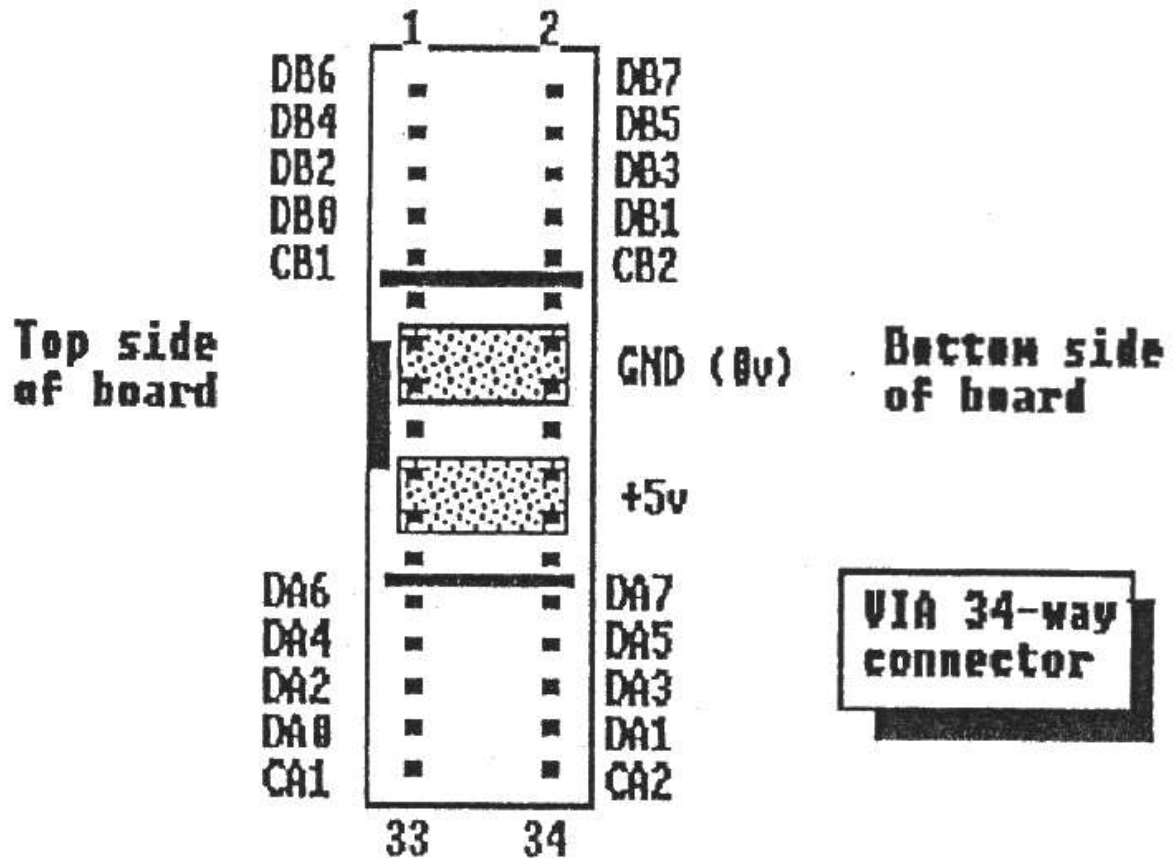
### **3. BOARD CONNECTORS AND SWITCHES**

There is one 34-way IDC plug on the end of QONTROL-11 which carries the connections for the digital VIA device. Just behind that connector there is a 15-way vertically mounted D-type socket which carries the connections for the analogue input channels. Beside the analogue connector there is a switch that can be used to select

between normal eight channel mode and BBC micro compatible 4-channel plus 2-button mode.

### 3.1 Digital VIA Connector

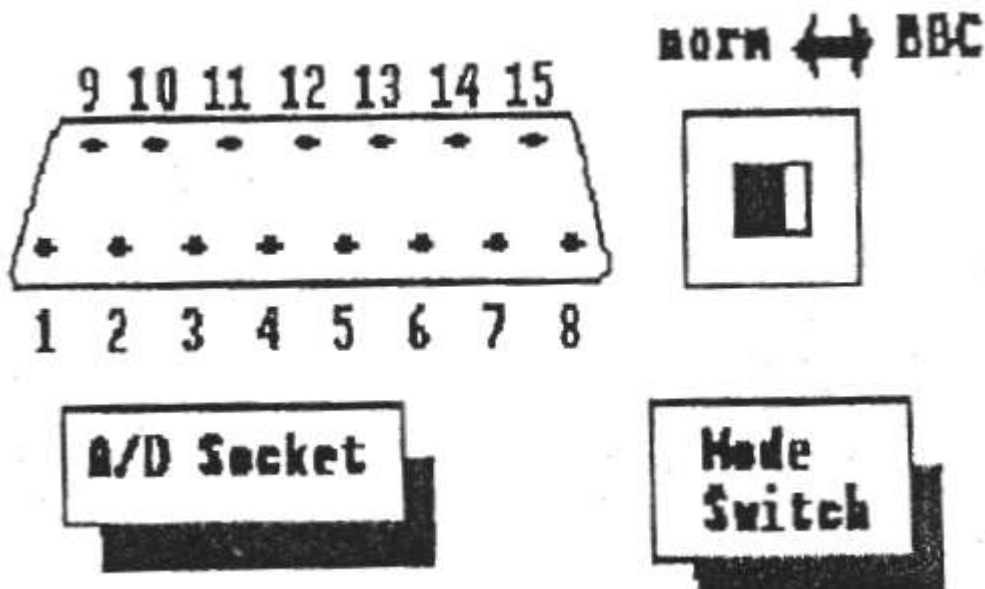
The VIA 34-way connector has the connections shown in the figure below. No more than 150mA should be drawn from the +5v rail unless the QL computer has an upgraded power supply. In any event the maximum that may be drawn is 600mA.



Note that the DAx and DBx lines are the 8-bit port data lines and that CAx and CBx are the communication control lines. All of these data and control lines are TTL compatible.

### 3.2 ANALOGUE SOCKET AND SWITCH

The socket is a 15-way D-type vertically mounted socket with the connections shown below. Note that there are two sets of pin usage depending on whether the adjacent switch is in the normal position or in the BBC micro Compatible position.



NORMAL MODE:

1	+5v	9	ACS
2	AC4	10	AC7
3	GND (0v)	11	Reference (+10v)
4	AC3	12	AC2
5	GND (0v)	13	AC6
6	GND (0v)	14	Reference (+10v)
7	AC1	15	AC0
B	GND (0v)		

BBC MICRO COMPATIBLE MODE:

1	+5v	9	Light pen strobe
2	GND (0v)	10	Button 1
3	GND (0v)	11	Reference (+10v)
4	AC3	12	AC2
5	GND (0v)	13	Button 0
6	GND (0v)	14	Reference (+10v)
7	AC1	15	AC0
8	GND (0v)		

In BBC micro mode pin 9 (light pen) is still connected directly to A/D channel 5. The button inputs are normally high (giving a reading of around 190-195)-and are driven low (a reading of below 50) when the BBC joystick buttons are pressed. Note that this can be monitored on A/D channels 6 and 7.

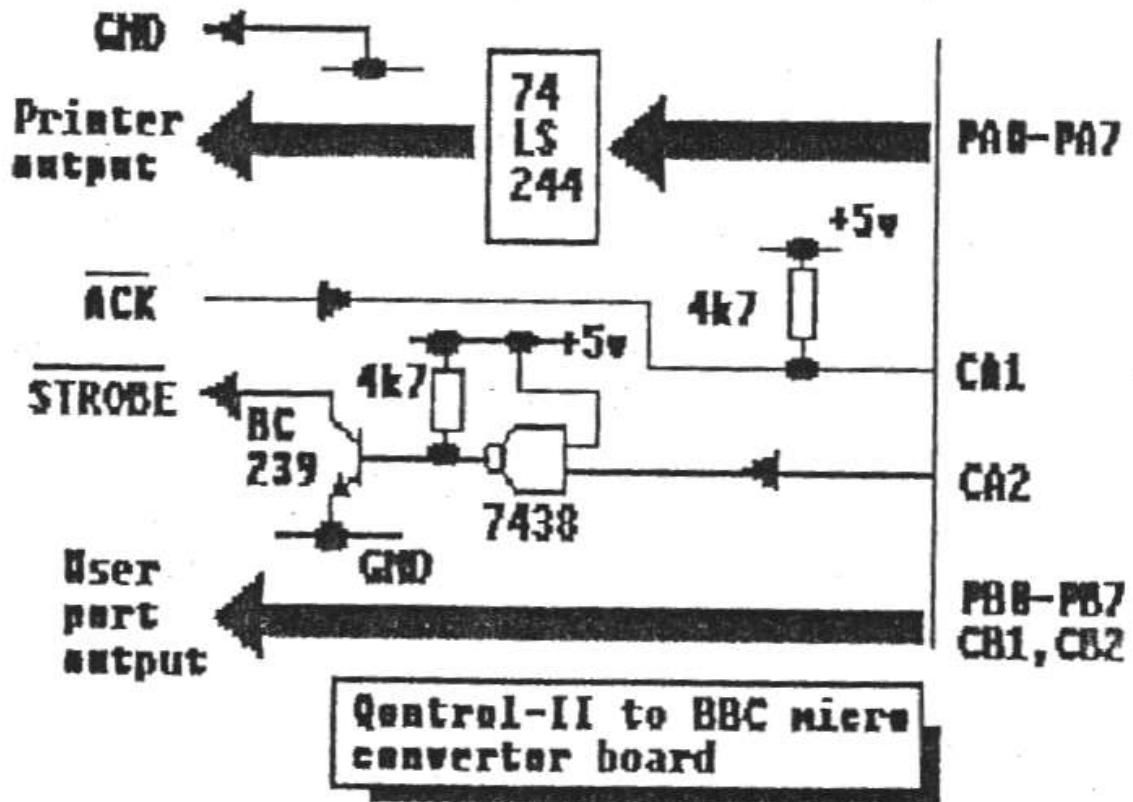
The Qontrol-II reference voltage is 10 volts and not 1.8 volts as on the BBC micro. In practice this should not cause any operational difficulties. The A/D channels always return an 8-bit result (0..255). The full scale deflection reading of 255 being obtained when the input is equal to the reference voltage (ie. 10 volts). Each degree of resolution therefore corresponds to approximately 39

millivolts. The technical performance characteristics of the 7581-JN AID convertor are:

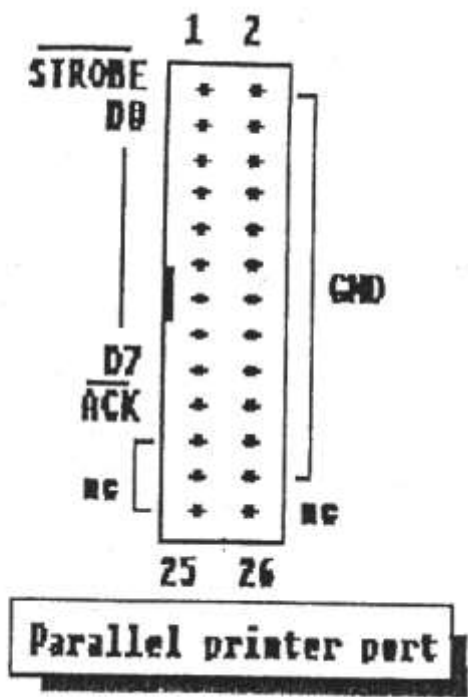
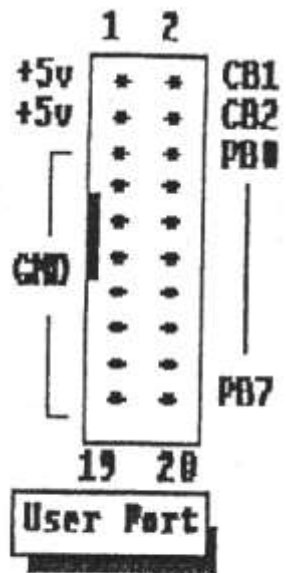
Relative accuracy: 1.875 LSB  
 Differential Nonlinearity: 1.875 LSB  
 Maximum offset error: 200 mV  
 Typical input resistance: 20 kohms  
 Absolute maximum AIN: +/-17 V

#### 4. BBC COMPATIBILITY BOARD

The Qontrol-II parallel printer device PAR can only be sensibly used when some additional electronics are linked to the outside of VIA data port A. You are of course welcome to perform this task yourself and leave VIA port B doing whatever is required of it. However a board is being made available which will convert the Qontrol-II output socket into BBC micro compatible parallel printer port and user port connectors. The circuit for this board is shown below.



The two connectors required are a 20-way IDC plug for the User port and a 26-way IDC plug for the Parallel printer port (Centronics compatible). The pin-outs are:



Using the Qontrol-II driver PAR will be sufficient to provide the Centronics compatible printer driver. To use the equivalent User port you will need to set up the port as for your BBC applications by using PORTSET.

cno/cno