Instruction manual for Morley Analogue & User Interface

MORLEY
E·L·E·C·T·R·O·N·I·C·S · L·T·D

Copyright

Morley Electronics acknowledges the support given by Acorn Computers Ltd. All references to Acorns' trademarks are acknowledged. Many thanks to Acorn for allowing us to use their software on board the interfaces.

We would be grateful to receive any constructive comments you may have with respect to this manual or the hardware so that we can better serve our customers. Any comments should be made in writing and returned to the address given below. This and any other correspondence should be forwarded to :

Morley Electronics Limited.
Morley House,
West Chirton,
North Shields,
Tyne and Wear.
NE29 7TY

Tel (091) 257 6355
Fax (091) 257 6373

N.B. Technical support is available from 4:00 pm to 5:30 pm, Monday to Friday. Please make sure you have read the appropriate parts of the manual before seeking assistance.

First published: 1994
Published by Morley Electronics Limited
Control number: MEL-TEM-009
Issue: 1
Page 1 of 19

Contents

Introduction

The Morley Electronics User/Analogue expansion card is designed in two formats; the 8 bit, for the Acorn A30x0 and A4000 series of computers, and the 16 bit designed for use with the Acorn A300, A400, A540 series and the A5000 computers.

This User/Analogue expansion card fits inside your Acorn computer and expands its capability by providing a User Port and an Analogue to Digital Converter.

The User Port is compatible with the User Port on the Archimedes 1/0 expansion card, and largely compatible with the User Port interface on the BBC Model B and Master 128 microcomputers. This enables you to connect your A3000 computer to a wide range of peripheral equipment already available for these machines.

The Analogue to Digital Converter is based on a 10-bit integrating converter, but in this implementation the accuracy can only be relied on to 8 bits. In practice it should prove to be 100% compatible with the ADC fitted to the BBC Model B and Master 128 computers when using documented system calls.

A3000 Installation

The 8 bit User/Analogue expansion cards should be fitted by an Acorn Authorised Dealer. Take your Acorn computer (in its original packaging) to an Acorn dealer who will install it for you. The dealer may make a charge for this service.

A3010 & A3020 Installation

Refer to Appenix E of the Welcome Guide (called Inside the computer). This will tell you how to remove the computer case and shows where the internal expansion card socket is. The expansion card is fitted to this socket. Make sure all the pins of the interface are alligned with the sockets when inserting the card and then secure the interface with the two screws provided. Reassembly is described in Appendix E of the Welcome Guide.

A4000 Installation

Refer to Appenix E of the Welcome Guide (called Inside the computer). This will tell you how to remove the computer case.

The inside of the computer will look similar to figure 1 below.

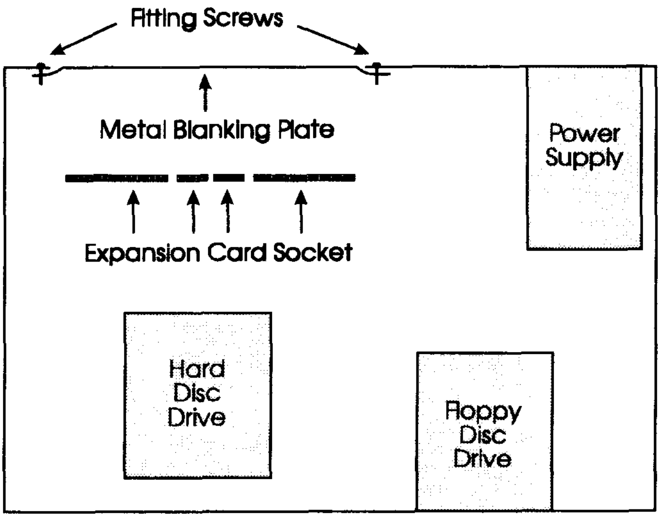


Figure 1. Inside of A4000

Remove the metal blanking plate and store it in a safe place. Insert the User/Analogue interface into the expansion card socket. Make sure all the pins of the interface are alligned with the sockets when inserting the card and then secure the interface with the two screws removed from the metal blanking plate.

Reassemble the computer as described in Appendix E of the Welcome Guide.

Warning: Before installing your interface, ensure that your machine is both switched off and unplugged from the mains to avoid the risk of electric shock. If you have any doubts about performing this part of the upgrade, please have it done by an authorised Acorn dealer as any damage caused will not be covered by your warranty.

Remove the monitor and disconnect the keyboard and all cables. Remove the six cover fixing screws on the underside of the A5000 (figure 2)

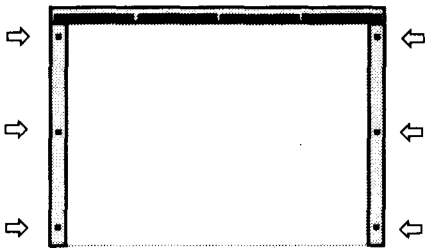


Figure 2. A5000 top cover fixing screws

Slide the top cover towards the rear of the machine and remove.

Some models of the A5000 have an EMC case shield (a metal plate) which fits over the case. To remove this, unscrew the five fixing screws on the righthand side o f the computer, as seen from the rear. The shield has interleaved tabs which fit into the edges of the case. Slide the shield off the computer carefully; there is a hole in the top to help you by the insertion of a screwdriver.

If no other interfaces are to be fitted, remove the uppermost blanking plate by removing the two fixing screws (figure 3)

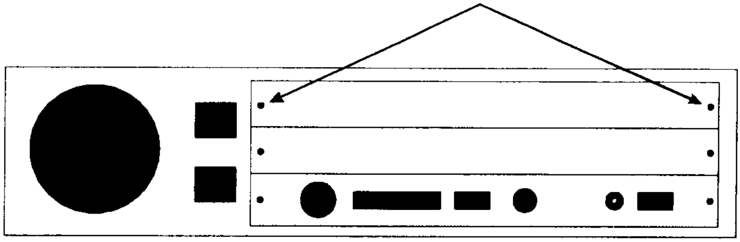


Figure 3. Rear of A5000

Once the blanking plate is removed, it is no longer required and should be stored in a safe place. Then take the half slot blanking plate supplied in the kit and using the T-piece supplied, screw the Revolution component and the half slot spacer together as in figure 4.

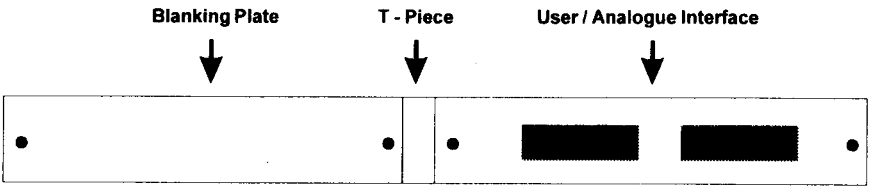


Figure 4. User/Analogue interface attached to blanking plate.

Taking the full assembly, install in the machine, making sure that the interface connector meets with the backplane correctly and secure the assembly by replacing the two screws shown in figure 3.

Refit the EMC case if this was originally present and replace the five screws. Replace the outer case and replace the screws shown in figure 2.

A300/400/540 Installation.

Remove the five screws that hold the lid in place as shown in figure 5.

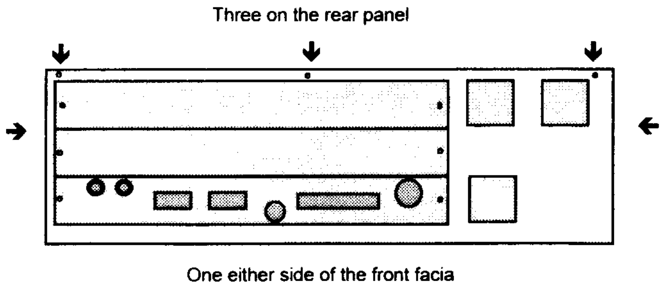


Figure 5. rear of A300/A400 series

Then slide the lid towards the rear of the machine. If the lid is stiff, pull the lid alternatively at either side until it becomes free. Then follow the instructions for fitting as for the A5000 detailed on the previous page.

User Port

The User Port consists of 8 data lines and two control lines from half of a 65C22 Versatile Interface Adapter chip (VIA). The VIA contains 16 internal registers, and these are mapped into memory. On the BBC Microcomputer and on Archimedes computers, legal access to these registers is made by using the two OSBYTE calls which read and write to SHEILA, numbers 150 and 151.

The signals available on the connector are the 8 data lines PBO to PB7 on pins 6, 8, 10, 12, 14, 16, 18 and 20 respectively, and the two interrupt/handshake/shift register control lines CB I and CB2 on pins 2 and 4 respectively.

When used for data transfer using handshaking, the CB2 signal is a 'data ready' output to the peripheral, and the CBI signal is a 'data taken' input from the peripheral.

When used in interrupt mode, CB I and CB2 cause the IRQ line of the VIA to be set low. However, interrupts from the VIA are not normally supported. See below.

Serial data can be shifted into or out of the CB2 pin under control of e ither an internal timer or from an external clock applied to CB 1.

The Archimedes implementation

The User Port is implemented using half of a 65C22 VIA chip. As on the BBC Microcomputer, the VIA registers are memory mapped, and control is exercised in the same way through OSBYTE calls 150 and 151, which read from and write to the I/O page SHEILA. It cannot be assumed that SHEILA is mapped into memory at a specific location, so direct access to the User Port through writing to or reading from specific addresses does not work.

Incompatibilities with the BBC User Port

The VIA chip on the expansion card is running at 2 MHz instead of the BBC Microcomputer's 1 MHz device. This means that the internal timers of the VIA are running twice as quickly as expected. If the shift registers are being used under control of the internal timer, then these too run twice as fast.

Power which may be taken from the User Port **must not exceed 500 mA.**

The interrupt signal from the VIA is supported, but a suitable interrupt handler for the Archimedes computer must be written.

Using the interface

The interface must be used through the legal BASIC and RISC OS commands. Any software that tries to access specific memory locations in the earlier BBC Microcomputer I/O space will not work. Also, OSBYTE calls 150 and 151 use the 6502 registers on the BBC Microcomputer and so are implemented slightly differently on the Archimedes computer. In general, parameters passed in A on the BBC Microcomputer are passed in the least significant byte of RO on the Archimedes computer. Those passed in X are now passed in the LSB of RI, and those passed in Y are now passed in the LSB of R2.

*FX commands still work as on the BBC Microcomputer, the parameters being passed in the correct registers automatically. (For details on *FX commands, see the Acorn User Guide supplied with your computer.)

The legal commands are:

OSBYTE 150 Read a byte from SHEILA
OSBYTE 151 Write a byte to SHEILA

The 16 VIA registers which are memory mapped to the SHEILA I/O space have offsets' &60 to &6F hex (96 to 111 decimal).

On entry: R0 contains the OSBYTE number.
RI contains the offset in SHEILA.
R2 contains the byte to be written (for the write command).
On exit: R2 contains the byte which was read (for the read command).

An example

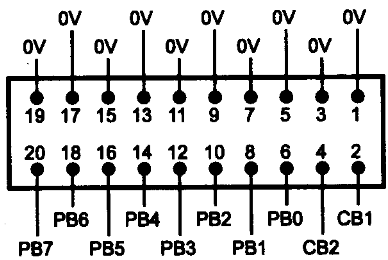
The User Port is controlled via the 16 registers of the VIA chip that are mapped into the I/O space SHEILA at offsets &60 to &6F. Listing 1 (see page 14), for example, shows how to write &FF to DDRB (data direction register B).The same example in assembly language is shown in listing 2 (see page 14).

User Port address allocation

There is a SWI instruction which returns the absolute address location of the User Port/ ADC upgrade in the memory map. The SWI can be called either with its name (I/O_Podule_Hardware) or its number (&40500). On exit R1 contains the base address of the upgrade hardware. All other registers are preserved. The User Port VIA is &2000 above this base address, and the VIA registers are four bytes apart.

User Port technical specification

Min/Max operating voltage	4.5V / 5.5V
Max supply current to expansion card	100mA (+current supplied from User Port)
Max output drive capability	1 TTL i/p
Max input load	1 TTL i/p
Max output current (+5V)	500 mA



User Port pin outs viewed from rear.

Analogue to Digital Converter

The Analogue to Digital Converter IC is a 10 bit integrating converter, but in this implementation the accuracy can be relied upon to only 8 bits. Its output can be between 0 and 65520, but only 8 bits are significant so accuracy is to the nearest multiple of 256. Its two voltage references are 0V and Vref. 0V corresponds to 0 and Vref corresponds to 65520, so any applied voltage between 0V and Vref (typically 1.8V) will generate a number in direct proportion.

On the BBC Microcomputer and the Archimedes, legal access is made to the ADC either by using the BASIC keyword ADVAL, or using the OSBYTE calls 16, 17, 128, 188, 189 and 190. Access to the registers can also be gained using the two OSBYTE calls which read and write to SHEILA, numbers 150 and 151.

ADVAL

ADVAL is a BASIC function which takes a single parameter, the channel number (0 to 4). If the parameter is 0, ADVAL returns a 2-byte number. The low byte will give the status of the two 'fire buttons' as follows:

Button	Status
0	No buttons pressed
1	Left fire button pressed
2	Right fire button pressed
3	Both fire buttons pressed

If the parameter is between 1 and 4, ADVAL returns a 2-byte number which is the value of that ADC channel. This value is in the range 0 to 65520 in steps of 16 (in 12-bit mode) and steps of 256 (in 8-bit mode). However, accuracy is only to the nearest multiple of 256 in either mode, because in this implementation only the high byte is guaranteed accurate.

OSBYTE 16 Select ADC channels which are to be sampled
On entry: R0 contains 16
R1 contains the number of channels to be sampled (0 to 4)
If RI contains 0 then sampling is disabled

OSBYTE 17 Force ADC conversion
On entry: R0 contains 17
RI contains the channel number to be forced (0 to 4)
If RI contains 0 then no conversion is forced.

OSBYTE 128 Read ADC channel value and fire button status
On entry: R0 contains 128
R1 contains channel number to be read (0 to 4)
On exit: If RI contained 0 on entry then the two lowest bits (bits 0 and 1) of RI indicate the status of the 'fire buttons', and R2 contains the number of the channel which was last used for ADC conversion, or 0 if no conversion has been completed. If RI contained 1 to 4 on entry then RI (low) and R2 (high) contain the 16-bit value for that channel.

OSBYTE 188 Read current ADC channel
On entry: R0 contains 188
On exit: RI contains the current ADC channel number

ADVAL contd.

OSBYTE 189 Read maximum ADC channel number

On entry: R0 contains 189

On exit: RI contains the maximum channel number to be used (0 to 4)

This maximum number is set by OSBYTE 16

OSBYTE 190 Read whether 12-bit or 8-bit conversion

On entry: R0 contains 190

On exit: If R1 contains 0 or 12 then the conversion is 12 bit

If RI contains 8 then the conversion is 8-bit

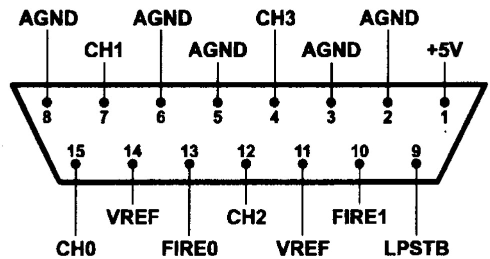
Note that conversion is guaranteed only to 8 bits due to the implementation.

An example

An example is given in listing 3 (see page 15). Listing 4 (see page 15) is the same example performed using OSBYTE call 128 and the BASIC SYS command.

ADC technical specification

Input Voltage Range	0-Vref (Vref typically 1.8V)
Accuracy	8 bits
Max conversion time	Isms
Input Impedance	1000 Mohms



ADC connector pin-outs, viewed from rear

Listing 1

```
10 osbyte%=6                          :REM SYS 6 is equivalent to OSBYTE
20 writebyte%=151                     :REM OSBYTE number for write byte
30 offset%=&62                        :REM offset in SHEILA of DDRB
40 byte%=&FF                          :REM byte to put in DDRB
50 SYS osbyte%,writebyte%,offset%,byte%
```

Listing 2

```
10                                    :REM write &FF to User Port DDRB
20 osbyte%=6                          :REM SWI 6 is equivalent to OSBYTE
30 writebyte%=151                     :REM OSBYTE number for write byte
40 offset%=&62                        :REM offset in SHEILA of DDRB
50 byte%=&FF                          :REM byte to put in DDRB
60 DIM code% 100
70 P%=code%
80 [
90 STMFD R13 !,{R0-R12,R14}           \ save registers on stack
100 MOV R0,#writebyte%                \ put OSBYTE number in R0
110 MOV R2,#offset%                   \ put offset in RI
120 MOV R2,#byte%                     \ put byte to be written in R2
130 SWI osbyte%                       \ execute OSBYTE call
140 LDMFD R13 !,{R0-R12,PC}           \put registers from stack & return
150 ]
160 CALL code%
```

The BASIC keyword ADVAL takes a parameter which is the ADC channel number. The ADVAL function performs an OSBYTE 128 call, reading the value on the specified channel.

Listing 3

```
10                                :REM read value of ADC channel 2
20                                :REM read fire button status & last
25                                :REM channel used
30 AtoD%=ADVAL(2)                 :REM convert on channel 2 & read result
40 firebutton%=ADVAL(0)           :REM firebuttons and channel status
50 PRINT AtoD%                    :REM print channel 2 ADC value
60 PRINT firebutton% AND 3        :REM print firebutton status
70 channel%=firebutton%DIV256     :REM shift right channel status
80 PRINT channel%                 :REM and print it
```

In the above example,
line 30 reads the value of ADC channel 2 into the BASIC variable AtoD%. Line 40 reads the fire button status and the ADC channel last used. Lines 50 to 80 print out the three pieces of information: the ADC channel 2 value, the fire button status, and the last channel to perform a conversion.

Listing 4

```
10                                :REM read value of ADC channel 2
20                                :REM read fire button status &last
25                                :REM channel using OSBYTE calls
30 osbyte%=6                      :REM SYS 6 is equivalent to OSBYTE
40 readADC%=128                   :REM OSBYTE no. for read ADC value
50 channel%=2                     :REM ADC channel required
60 status%=0                      :REM parameter 0 to read status
70 SYS osbyte%,readADC%,channel%TO ,low%,high%
80 SYS osbyte%,readADC%,status%TO ,firebutton%,channel%
90 AtoD%=low%+(high%<<8)
100 PRINT AtoD%
110 PRINT firebutton%
120 PRINT status%
```

In the above example,
line 70 reads ADC channel 2
line 80 reads the status of the two fire buttons and the last channel used.

Warranty Notice

The User/Analogue expansion card is warranted free from defects in materials and workmanship for a period of 12 months from the date of purchase. During this time it will be replaced or repaired free of charge at Morley Electronics discretion. This warranty will not apply if the unit has been tampered with or modified in any way.

Morley Electronics will not be held liable for any injury, loss or damage, direct or consequential, arising out of use, or the inability to use, this product.