

The μ -I/O expansion board gives users the ability to add analog inputs and outputs, digital inputs and isolated high current outputs to the μ FlashTCP. The board installs easily to the back of the μ FlashTCP with a short ribbon cable. If more I/O is required, several boards can be added to the system. A library of C and QuickBasic functions is supplied to eliminate the hassle of programming devices on a serial bus.

Features:

Analog to Digital:

- 12 bit resolution
- 4.096V reference (1mV/count)
- Single-ended or differential modes

Digital Inputs:

- TTL Logic levels
- Diode protected and current limited

Digital to Analog:

- 12 Bit
- Voltage output (0-4.095 V)
- Current output (0-20 mA)

Drivers:

- 1 Amp transistors
- 60V max Vce
- 2500 V optical isolation

System requirements:

- μ FlashTCP SBC
- Interface Cable
- Stacking standoffs

Configuration:

Two jumpers allow configuration of each μ -I/O board. Jumpers configure the direction of signals on BUS1 and BUS2 (J1 and J2).

Shunts should be installed on JP1 between pins 1-3 and 2-4 or 3-5 and 4-6. BUS2 (J2) will transmit to a second I/O board if JP1 has 1-3 and 2-4 shorted. BUS1 (J1) will transmit data when JP1 has 3-5 and 4-6 shorted. Signals will be received on the remaining BUS connector.

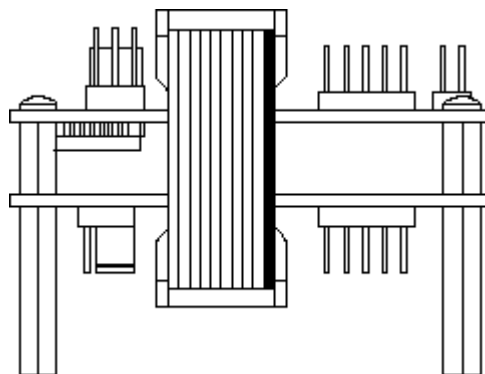
Installation:

The μ -I/O is connected to the μ FlashTCP using a 10-pin ribbon cable. The cable connects between J1 (BUS1) on the μ -I/O and J5 on the μ FlashTCP. The jumpers on JP1 should short pins 1-3 and 2-4 (default position). The IO board is attached to the bottom of the μ FlashTCP with the stacking standoffs. Additional boards are attached in a similar manner using the BUS2 connectors and changing JP1 accordingly.

Connectors:

Interface connectors use Molex KK style headers and housings.

Connector	Description	Molex Pn	Jk micro P/N
J3,4	3 pin housing	22-01-3037	28-0012
J5,6,9	4 pin housing	22-01-3047	28-0035
J7,8	8 pin housing	22-01-3087	28-0037
	crimp terminal	08-50-0114	28-0013



Connector Pinouts:

J5 / AD1	
Vref	1
AD0	2
AD1	3
GND	4

J6 / AD2	
Vref	1
AD2	2
AD3	3
GND	4

J3 / DA1	
Vout1	1
Iout1	2
GND	3

J4 / DA2	
Vout2	1
Iout2	2
GND	3

J7 / OUT1	
Out1+	1
Out1-	2
Out2+	3
Out2-	4
Out3+	5
Out3-	6
Out4+	7
Out4-	8

J8 / IN1	
IN1	1
IN2	2
IN3	3
IN4	4
IN5	5
IN6	6
IN7	7
IN8	8

J9 / PWR/GND	
Vcc	1
Vcc	2
GND	3
GND	4

J1 and J2 (BUS1 and BUS2)			
Data	1	2	CLK
Reset/	3	4	CS IN (OUT)
Vcc	5	6	GND
IRQ5	7	8	IRQ6
Vcc	9	10	GND

Software:

uDRIVER.OBJ contains routines callable from C or QuickBASIC. This library allows easy access to the various functions of the μ-I/O board. Although the C and BASIC routines have common names, the C routines are functions (returning values when required) and the BASIC routines are subroutines (results are returned in variables passed to the subroutine). Functionally they are identical. The assembly source code (UIO.ASM) and C header file (UIO.H) are also included. The object file may be regenerated using TASM (Turbo Assembler).

All values passed and returned from the library functions are 16 bits wide. Many functions do not use all 16 bits, but the complete word is required. When the complete word is not used, data fills the low order bits and the high order bits are unused and ignored. In the following descriptions, C function syntax is shown first and BASIC syntax, where different, is shown in brackets ([]).

int GetVersion() [GetVersion(*data*)]

Returns the version number of the driver library.
Least significant digit is minor revision.
Most significant digits are major revision.

void PutA2DChannel(*address*)

Sets the channel to get data from on the next call to GetA2DChannel().
address refers to a device (or channel). Address values are sequential, starting at 0, for each type of device.

int GetA2D() [GetA2D(*data*)]

Returns the conversion results from previously selected A/D

void PutD2AChannel(*address*)

Selects a D/A channel.
address refers to a device (or channel). Address values are sequential, starting at 0, for each type of device.

void PutD2A(*data*)

Writes output value to the selected D/A channel.
data is a value read from an A/D or written to a D/A. Only the low 12 bits are used.

void PutDriverChannel(*address*)

Selects driver.
address refers to a device (or channel). Address values are sequential, starting at 0, for each type of device. Each driver bit is addressed individually.

void PutDriver(*state*)

Turns the selected driver ON or OFF.
state is the value written to a driver or read from an input: 0=OFF, non-zero = ON.

void PutDIChannel(*address*)

Selects digital input.
address refers to a device (or channel). Address values are sequential, starting at 0, for each type of device. Each input bit is addressed individually.

int GetDI() [GetDI(*state*)]

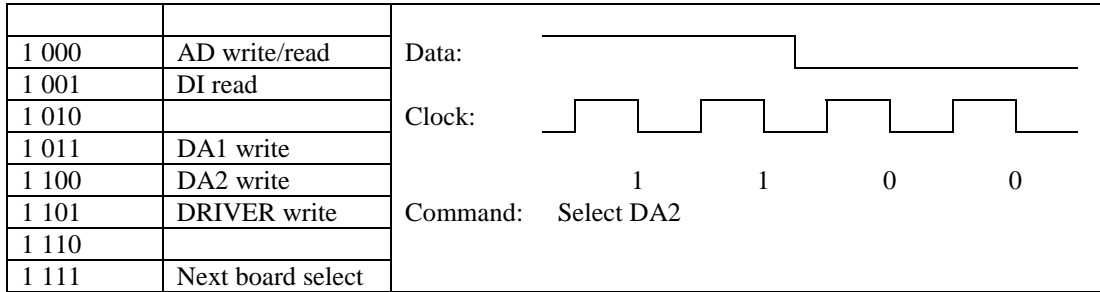
Returns the conversion results from previously selected Digital Input
state is the value written to a driver or read from an input: 0=OFF, non-zero = ON.

Hardware Notes:

The μ-I/O board operates on a serial bus implemented using processor I/O lines for clock, data, reset and chip select signals. The data signal is bi-directional, reset and CS are active low, and data must be valid on the falling edge of the clock signal.

To reset the board(s), assert the reset line. A reset must be issued in order to change the selected channel.

To select a device, a start bit and a 3 bit address are clocked to the board. There are 6 selectable devices on each board: AD read/write, DI read, DA1 write, DA2 write, DRIVER write, and Next Board Select. Data is clocked MSB first when selecting a board.



After a device has been selected, information is written to (or read from) the device as per the device specification.

The Driver circuit consists of a shift register and a latch. Data is clocked in MSB first and latched on the falling edge of chip select.

When using the Digital to Analog converters in voltage output mode, it may be necessary to install a resistor (approximately 10k ohm) between pins 2 and 3 of the output. This will reduce noise on the voltage output.

Revision Notes:

991001A First Issue.

010501A Fix error in jumper descriptions, clarify meaning of jumpers. Fix address.