MEMS MS2/MS3 Optical Switch Module

Operation Manual

No Liability for Errors

DiCon reserves the right to correct technical and typographical errors in this manual at any time, without prior notice. In no event shall DiCon be liable for errors in this manual or for any damages arising out of, or relating to, this manual.

Product Warranty/Limitation of Remedies

DiCon warrants, to the original Buyer, all of its products to be free from defects in both workmanship and material for a period of one year from the date of shipment. This warranty extends to all products which have proved defective through normal use, but excludes products that have been damaged, mishandled, disassembled, modified, or misused by Buyer or any other person. This warranty is in lieu of all other warranties, and DiCon disclaims all other warranties express or implied, including any warranty of merchantability, fitness for a particular purpose, or arising from the course of dealing between the parties or usage of trade. DiCon does not extend any warranty of any kind whatsoever to any purchaser of the products from Buyer or to any end-user of the products. DiCon, at its sole choosing, will replace or repair to proper working condition any products under warranty that are returned. Products repaired or replaced under warranty are only warranted for the remaining unexpired period of time of the original warranty. DiCon reserves the right to issue a credit memo for any defective product as an alternative to product replacement or repair. DiCon will not accept Buyer generated debit memos. Buyer may not set off or withhold payment because any product is defective. In no event shall DiCon’s liability under this warranty and this contract exceed the purchase price of the products. In no event shall DiCon be liable under this warranty or this contract for consequential, incidental or special damages.

Return Material Authorization Terms

DiCon will only accept a return of products for which a DiCon Return Material Authorization (“RMA”) Number has been issued to Buyer prior to the shipment of the return products to DiCon. This RMA Number must be displayed on all return shipment documents. DiCon will refuse all returns that are not accompanied by an RMA Number. All risks of any such refused shipment are the sole responsibility of Buyer.

For warranty returns, DiCon will only accept return products accompanied by a statement of defects. DiCon will not evaluate returns not including this information, and such returns will be returned to Buyer at Buyer’s expense.

Warranty returns proved defective through damage, mishandling, disassembly, modification, or misuse by Buyer or any other person, and warranty returns found non-defective, will be subject to evaluation and processing fees, and repair costs if applicable. Non-warranty returns will be evaluated and will be subject to evaluation and processing fees. If non-warranty repair work is necessary, Buyer will be notified of repair costs before a repair work order is initiated. Confirming POs are required for non-warranty repair work.

For warranty returns, Buyer is responsible for one-way freight costs to DiCon, including any duty and taxes. DiCon will cover freight costs for return shipment to Buyer. Shipment charges billed to DiCon without prior approval from DiCon will be re-invoiced to Buyer.

For non-warranty returns, Buyer is responsible for two-way freight costs, including any duty and taxes. If shipment consists of returns that are both warranty and non-warranty, the shipment will be considered as non-warranty.

DiCon will not accept Buyer generated debit memos.

All international return shipments to DiCon, including packaging and airway bill, must be marked "Goods made in the United States; enter as American Goods Returned ('AGR')" and state the reason for the return to the United States. DiCon will refuse all returns that are not properly documented. All risks of any such refused shipment are the sole responsibility of Buyer.

International returns should be sent via Federal Express, UPS, or DHL. International returns may be processed using DiCon's brokerage: EWI Inc. 305 Harbor Way, South San Francisco, CA 94080. Contact Harvey Louis at TEL: (650) 794-1388, FAX: (650) 794-1389. If one of these carriers or DiCon's broker is not used, DiCon may invoice Buyer for any additional costs including duty and taxes.

Reverse Engineering / Confidentiality

Buyer shall not reverse engineer, decompile, disassemble, modify, reproduce or copy any products or any software within any products. Buyer shall not analyze or identify the chemical composition or the physical characteristics of any products. Buyer shall not furnish DiCon specifications to any other person.

Software License

DiCon does not transfer ownership of software contained in any products. DiCon grants to Buyer a perpetual non-exclusive license to use software in the operation of the product in which it is contained. This license is transferable only with the transfer of ownership of the product.

Contents

1. Product Overview

This manual is intended for use with part numbers beginning with the following:

- MEMS 1xN Switches: MS2-1xN or MS3-1xN
- MEMS 2x2 Switches: MS2-2x2
- MEMS 2x2 Add Drop Switches: MS2-2x2AD
- MEMS 2x2 Blocking Switches: MS2-2x2BK

1.1 MEMS 1xN Optical Switch

DiCon's MEMS 1xN Optical Switch is based on a micro-mechanical system (MEMS) chip. The MEMS chip consists of an electrically movable mirror on a silicon support. The 1xN MEMS chip has two axes of rotation. Voltages applied to the MEMS chip cause the mirror to tilt along one or both axes, which changes the coupling of light between a common fiber and N input/output fibers.

The MEMS 1xN Optical Switch is a non-latching device. When the electrical power is removed, the switch will return to the default state.

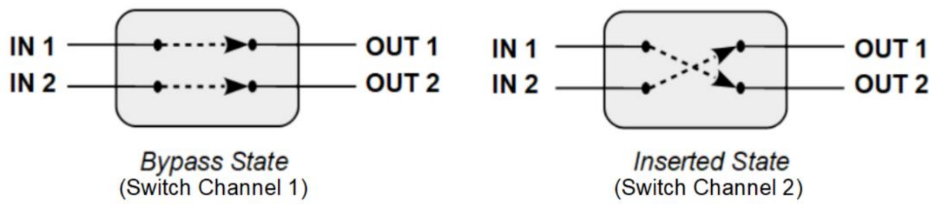
The MEMS 1xN Optical Switch provides channel selection between sets of single input fibers and sets of N output fibers. The module allows up to five MEMS switch components to be co-packaged with the option of switching synchronously. The switch is bi-directional and can be used as either a 1xN or as an Nx1 switch. In a 1 to N application, the common fiber is used as the input and the N channels are used as output fibers. When the switch is operated as an N to 1, the N channels are the N inputs and the common fiber is the output.

1.2 MEMS 2x2 Optical Switch

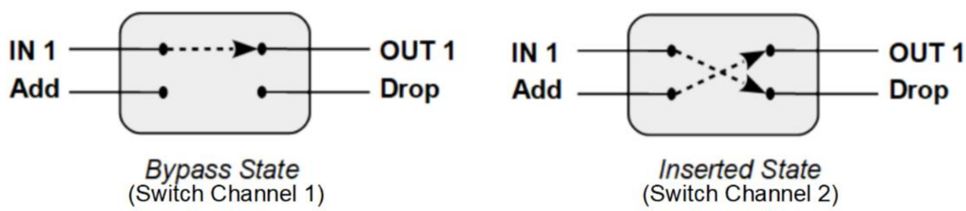
DiCon's MEMS 2x2 Optical Switch is based on a micro-mechanical system (MEMS) chip. The MEMS chip consists of an electrically movable mirror on a silicon support. The 2x2 MEMS chip has two axes of rotation. Voltages applied to the MEMS chip cause the mirror to tilt along one or both axes, which changes the coupling of light between two input fibers and two output fibers.

There are three configurations of 2x2 switches:

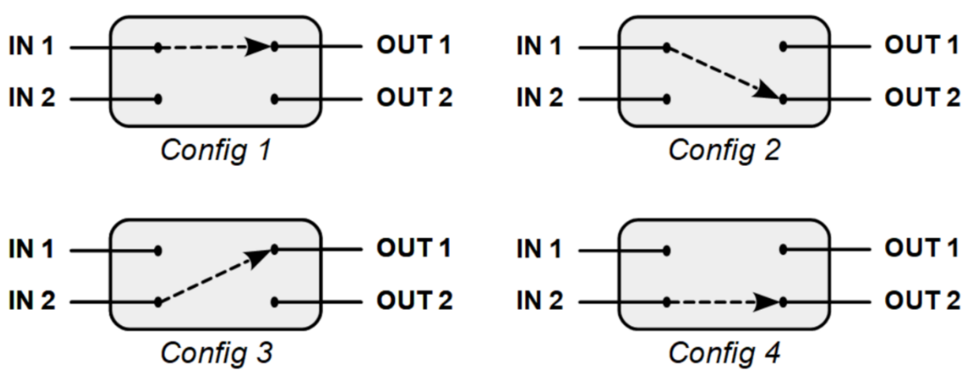
- **MEMS 2x2 Switch** (standard configuration), 2 switch states



- **MEMS 2x2 Add Drop Switch**, 2 switch states



- **MEMS 2x2 Blocking Switch**, 4 switch states



2. Switch Operation

2.1 Pin Assignments

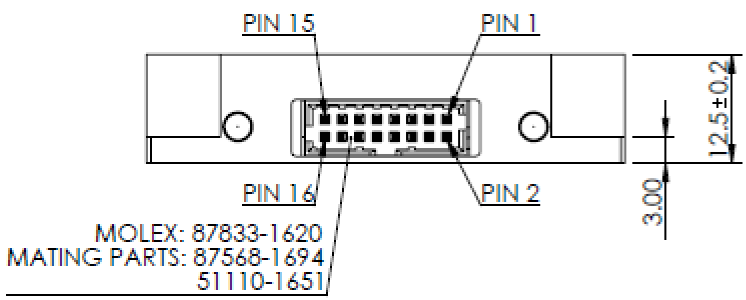
The MEMS Optical Switch Module (Size 2 and Size 3) operates through a 16-pin connector. The pin assignments for RS-232, I^2C, and TTL control interfaces are listed in tables 1, 2, and 3 respectively. The electrical connector is a Molex 87833-1620 male connector, which mates with the female connector 87568-1694 or 51110-1651.

Warning!

Failure to ensure that the electrical connections are made properly can damage the module. Beware that if the electrical jumper has the same type of connector on both ends, special care must be taken to ensure that the correct end is plugged into the module. If the electrical jumper is reversed, damage will occur to the switch module because this will connect power to pins on the module that will become damaged if a voltage is applied.

Do not apply voltages to any pin labeled 'NC'. Any voltage applied to these pins can cause immediate and catastrophic damage to the switch. Applying a voltage greater than the maximum rating or any voltage to a pin labeled 'NC' will void the switch warranty.

Figure 1. DiCon Defined Electrical Pin-out for MEMS Switch Module (Size 2 and Size 3)



(Units in mm)

Molex Pin Assignment:

Please note that Molex's pin assignment for the mating Molex connector, 87568-1694, is reversed compared to DiCon's pin assignment.

Warning! Please refer to the warning on page 7.

Table 1. RS-232 Pin Assignment (DiCon Defined Pin-Out)

DiCon PIN #	Name	Description	Direction	Specification	Unit
1	NC	No Connection			
2	NC	No Connection			
3	Vcc	Power Supply	IN	+12	VDC
4	Vcc	Power Supply	IN	+12	VDC
5	GND	Signal & Power Ground			
6	GND	Signal & Power Ground			
7	NC	No Connection			
8	NC	No Connection			
9	232TX	RS232 TX	OUT	-15 to +15	VDC
10	232RX	RS232 RX	IN	-15 to +15	VDC
11	NC	No Connection			
12	NC	No Connection			
13	/BUSY	Normally pulled high. While a module is busy, it will be pulled low.	OUT	LVTTL	VDC
14	/ALARM	Normally pulled high. While a module has logged alarms, it will be pulled low.	OUT	LVTTL	VDC
15	NC	No Connection			
16	/RESET	Low level active for hardware reset.	IN	LVTTL	VDC

Table 2. I^2C Pin Assignment (DiCon Defined Pin-Out)

DiCon PIN #	Name	Description	Direction	Specification	Unit
1	NC	No Connection			
2	SDA	I^2C serial data	IN/OUT	LVTTL	VDC
3	Vcc	Power Supply	IN	+12	VDC
4	Vcc	Power Supply	IN	+12	VDC
5	GND	Signal & Power Ground			
6	GND	Signal & Power Ground			
7	SCL	I^2C Serial Clock	IN	LVTTL	VDC
8	NC	No Connection			
9	NC	No Connection			
10	NC	No Connection			
11	NC	No Connection			
12	NC	No Connection			
13	/BUSY	Normally pulled high. While a module is busy, it will be pulled low.	OUT	LVTTL	VDC
14	/ALARM	Normally pulled high. While a module has logged alarms, it will be pulled low.	OUT	LVTTL	VDC
15	NC	No Connection			
16	/RESET	Low level active for hardware reset.	IN	LVTTL	VDC

Warning! Please refer to the warning on page 7

Table 3. TTL Pin Assignment (DiCon Defined Pin-Out)

DiCon Pin #	Name	Description	Direction	Specification	Unit
1	D0	Data 0 Input	IN	LVTTL	VDC
2	D5	Data 5 Input	IN	LVTTL	VDC
3	Vcc	Power Supply	IN	+5	VDC
4	Vcc	Power Supply	IN	+5	VDC
5	GND	Signal & Power Ground			
6	GND	Signal & Power Ground			
7	D4	Data 4 Input	IN	LVTTL	VDC
8	D1	Data 1 Input	IN	LVTTL	VDC
9	NC	No Connection			
10	NC	No Connection			
11	D2	Data 2 Input	IN	LVTTL	VDC
12	D3	Data 3 Input	IN	LVTTL	VDC
13	/BUSY	Normally pulled low. While a module is busy, it will be pulled high.	OUT	LVTTL	VDC
14	/ALARM	Normally pulled low. While a module has logged alarms, it will be pulled high.	OUT	LVTTL	VDC
15	/STROBE	Falling edge active to synchronize command execution.	IN	LVTTL	VDC
16	/RESET	Low level active for hardware reset.	IN	LVTTL	VDC

2.2 Power Pins (Pins 3 & 4)

The power pins 3 & 4, named VIN in the pin assignment tables above, are the power supply pins to the MEMS optical switch module. It is recommended that both of these pins should be connected to the supply voltage.

2.3 Ground Pins (Pins 5 & 6)

The signal & power ground pins 5 & 6, named GND in the pin assignment tables above, are tied together electrically inside the module and share both pins. It is recommended that both pins are connected to ground and not left floating.

Please note that case ground is floating and is not connected to the ground pins. Also, it is not necessary to ground the case.

2.4 Reset Pin (Pin 16)

The reset pin is a LVTTL input. It is an optional pin and it is not required to be used, in order to operate the switch. If it is not desired to use this pin, then this pin can be left floating. If the reset pin is to be used, then this pin should be left in the logic high state for normal switch operation. If the reset pin is set to logic low, then the switch module will be reset.

2.5 Electrical Specifications

Table 4. Electrical Specifications

Parameter		Logic Low	Logic High	Damage Threshold	Unit
Latching Type		Non-latching			
Input	I^2C Interface[1]	<0.4	3.0 to 5.0	-0.3 // +7.0	VDC
	RS232 Interface	<0.5	+5.0	-30 // +30	VDC
	LVTTL Interface[2]	<0.4	2.4 to 3.3	-0.5 // +3.8	VDC
Output	I^2C Interface[1]	<0.3	2.4 to 5.0	-0.3 // +5.5	VDC
	RS232 Interface	-5	+5.0	-15 // +15	VDC
	LVTTL Interface[2]	<0.4	2.9 to 3.3	-0.5 // +4.6[2]	VDC
		Minimum	**Typical**	**Maximum**	
Vcc Power Supply Voltage	RS232 or I^2C type	10.8	12.0	13.2	VDC
	TTL type	4.75	5.0	5.25	VDC
Power Consumption	RS232 or I^2C type		1.0	1.3	W
	TTL type		0.4	0.7	W

1. Pullup to Vin or Vout on customer equipment.
2. If driving the input or output with 5V TTL logic, install a 220 – 1000 ohm resistor in series to limit input current. The damage threshold is 6 VDC with this drive configuration.

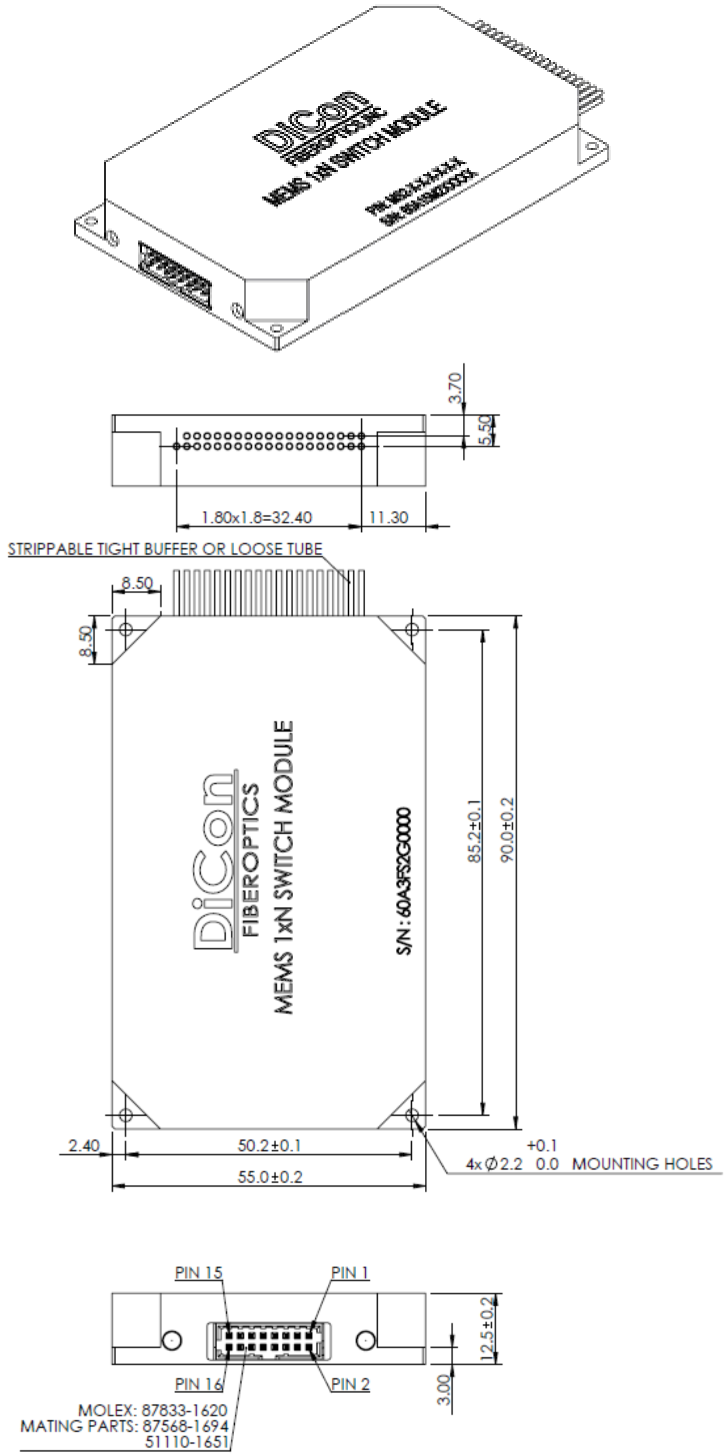
2.6 Environmental Specifications

Table 5. Environmental Specifications

Parameter	Specification	Unit
Operating Temperature	-5 to 70	^oC
Storage Temperature	-40 to 85	^oC

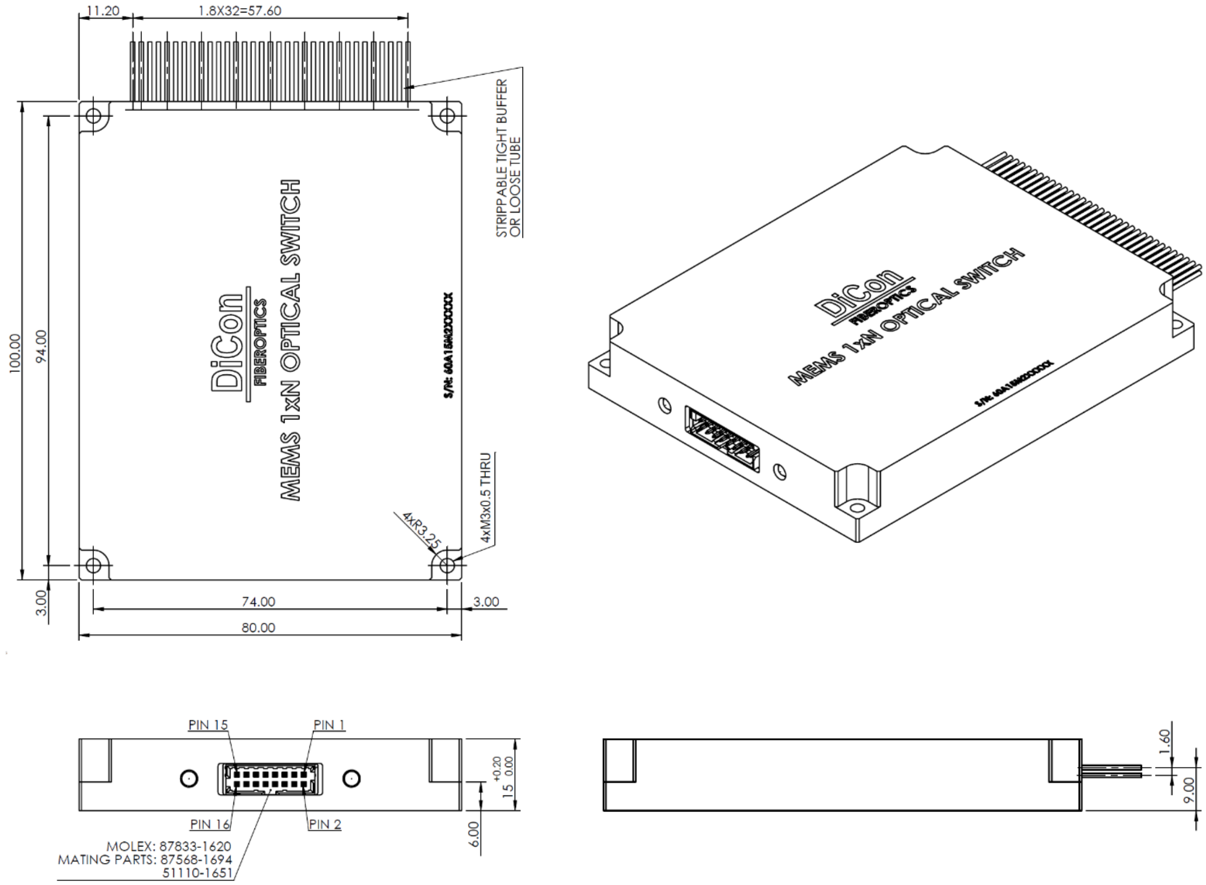
3. Mechanical Dimensions

Figure 2. Size 2 Mechanical Dimensions



(Units in mm)

Figure 3. Size 3 Mechanical Dimensions



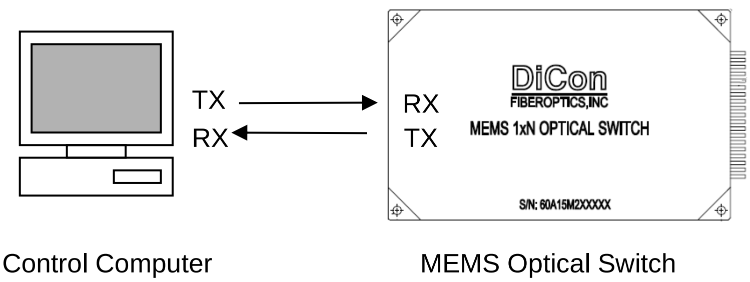
(Units in mm)

4. RS232 Interface

4.1 RS232 Control Line Connection

To control the switch module with RS232 control, the TX port from the control computer needs to be connected to the RX port on the RS232 module. Similarly, the RX port on the computer needs to be connected to the TX port on the switch module, as shown below in figure 4.

Figure 4. RS232 TX and RX control line connection diagram



4.2 RS232 Parameters

The RS232 baud rate is 115,200bps with 8 data bits, 1 stop bit and no parity. All RS232 ASCII commands use <CR> as the terminator character. And the RS232 ASCII responses use <LF> and <CR><LF>> as the terminator character. Table 6 lists the conventions used in this manual for RS232 control.

Table 6. Conventions

Convention	Meaning
(…)	Enclosure for a variable. The '(' and ')' characters are not part of the data.
[…|…|…]	Have one or none
{…|…|…}	Must have one
'and'	'and' is a comment
<SP>	Separator that is a space character
<CR>	Carriage return as a terminator
<LF>	Line feed

4.3 RS232 Command Set

Table 7. RS232 Serial Port (ASCII) Command Set

Command	Description
ID?	Queries the switch's identification string
CF?	Queries the input/output channel dimensions of the switch
EO	Sets the echo option
ER?	Queries the system status/error
I1	Sets the state of the optical switch to the output channel N
I1?	Queries the output channel
PK	Sets the optical switch to parking state

ID?

Description	Queries the switch's identification string.
Parameters	None
Reply	Four string values 1. Device manufacturer name 2. Device model name 3. Device firmware number and version 4. Device serial number
Example	`(Send):ID?<CR>` `(Receive):<LF>DiCon Fiberoptics Inc,MS1x36,FW97198 Rev.C4, 60A0EM2D0001<CR><LF>>`

CF?

Description	Queries the input/output channel dimensions of the switch.
Parameters	None
Reply	Two numerical values 1. Maximum input channels 2. Maximum output channels
Example	`(Send):CF?<CR>` `(Receive): <LF>1,32<CR><LF>>`

EO

Description	Sets the echo configuration, and returns the current echo flag. 0 indicates that echo is off 1 indicates that echo is on When echo is on, the device will transmit every character that it receives from the RS232 interface. By default, echo is off. The echo setting is volatile. The default value is restored at startup.
Parameters	Echo setting (numeric)
Reply	One numerical value 0 indicates that echo is off 1 indicates that echo is on
Example 1	`(Send):EO<SP>1<CR>` *Turns echo on.* `(Receive): <LF>1<CR><LF>>` *Indicates echo is on.*
Example 2	`(Send):EO<SP>0<CR>` *Turns echo off.* `(Receive): <LF>0<CR><LF>>` *Indicates echo is off.*

ER?

Description	Queries the system status/error.
Parameters	None
Reply	Error code. Refer to **Table 8** for possible return codes.
Example	`(Send):ER?<CR>` `(Receive): <LF>ERR0001<CR><LF>>` *Invalid command.*

I1

Description	Sets the state of the switch to the output channel number ***n***

Parameters	Two numerical values. The first number is the input channel number and the second number is the requested output channel.
	1. Input channel number (***1*** for 1xn)
	2. Output channel number 0 to ***n***
	
	The commanded output channel should be an integer from 0 to n, where ***n*** is the number of channels in the switch (ex. For a 1x12 switch, ***n*** is 12). Commanding the switch to position 0 will set the switch to the parking position.
	
	Note for 2x2 Switches – Standard & Add Drop:
	Bypass State, set output channel number = 1
	`(Send):I1<SP>1<CR>`
	Inserted State, set output channel number = 2
	`(Send):I1<SP>2<CR>`
	
	Note for 2x2 Switches – Blocking:
	Config 1, set output channel number = 1
	`(Send):I1<SP>1<CR>`
	Config 2, set output channel number = 2
	`(Send):I1<SP>2<CR>`
	Config 3, set output channel number = 3
	`(Send):I1<SP>3<CR>`
	Config 4, set output channel number = 4
	`(Send):I1<SP>4<CR>`

Reply	None
Syntax	**I1<SP>(output channel number n)<CR>** The output channel number is from 0 to ***n***.
Example 1	`(Send):I1<SP>12<CR>` *Sets Switch to channel 12*
Example 2*	`(Send):I1<SP>0<CR>` *Sets Switch to the parking state*

* Command “I1 0” is supported starting from firmware 97198 Rev.C3.

I1?

Description	Queries the state of the switch
Parameters	None
Reply	A numerical value for the output channel number ***n*** will be returned. A return value of 0 indicates that the switch is in the off state since power up or is in the parking state (see Example 2).
Example 1	`(Send):I1?<CR>` `(Receive): <LF>12<CR><LF>>`
Example 2	`(Send):I1?<CR>` `(Receive): <LF>0<CR><LF>>`

PK

Description	Sets the switch to parking state
Parameters	None
Reply	None
Example	`(Send):PK<CR>` `(Receive):`

The return codes for various error conditions are shown below in Table 8.

Table 8. MEMS 1xN Switch Module Return Codes for RS232 Control

Return Code	Description
+0	Successful
ERR0001	Invalid Command
ERR0002	Value Out of Range
ERR0003	Command Fail

5. I²C Interface

This section defines the MEMS 1xN Switch Module I²C command set, which implements communication with the microcontroller (MCU) that is incorporated inside of the MEMS 1xN Switch Module. The I²C interface itself conforms to the Philips I²C specification.

Communication between a controlling PC, or other control electronics, and the MEMS 1xN Switch Module's microcontroller is conducted in Master-Slave fashion, with the microcontroller acting as the SLAVE device, and the PC acting as the MASTER device.

The MEMS 1xN Switch Module cannot initiate communications. In addition, if there are multiple SLAVE devices in the system, then there cannot be communications between the SLAVE devices.

For detailed information on this I²C implementation, refer to the NXP I²C User Manual: www.nxp.com/documents/user_manual/UM10204.pdf

5.1 I²C Address

The MEMS 1xN Switch Module is provided with a default I²C address of 0x73 (decimal 115). The address is a 7-bit address, and it occupies the seven most-significant bits of the address byte. At customer request, a different default address can be stored in the EEPROM, at time of manufacture. Starting from firmware 97198 Rev.C4, customers can change the MEMS 1xN Switch Module's I²C address using command 0x37.

5.2 Physical and Electrical Interface

As shown in Figure 1 and Table 2 the I²C interface uses the following signals.

Table 9. I²C Signals

Signal Name	Description
SDA	I²C Data (pin 2)
SCL	I²C Clock (pin 7)
/BUSY	Busy (pin 13)
/ALARM	Alarm (pin 14)
/RESET	Hardware Reset (pin 16) Logic Low Active

SCL is a standard I²C clock, with a rate of 100 kHz.

5.3 I²C Command Format

An I²C command consists of the slave address, a command byte, and optionally one or more data bytes, and CRC bytes.

■ Write Command

STA	COMMAND CODE	DATA	CRC16	P
Byte1	Byte2	Byte 3~(N-2)	Byte N-1, N	
address*2	command code	[Data length]	[Data Block]		

■ Read

STA	COMMAND CODE	DATA	CRC16	P
Byte1	Byte2	Byte 3~(N-2)	Byte N-1, N	
address*2+1	command code	[Data length]	[Data Block]		

■ Error Response

STA	COMMAND CODE	EXCEPTION CODE	CRC16	P
Byte1	Byte2	Byte3	Byte 4, 5	
address*2+1	0x80 + command code	1 to 127		

STA = I2C start with address and R/W bit
P = I2C stop
CRC16 = ModBus CRC16 (include address with R/W bit)

5.3.2 I²C Master-to-Slave Communication

To use the I²C interface for transmitting data (Master-to-Slave):

1. The Master sends a START condition, the address byte, one or more data bytes, and finally terminates the operation with the STOP condition.
2. The address byte for a WRITE operation is the 7-bit slave address followed by the READ/WRITE bit set to 0. Therefore, the effective write address for a MEMS 1xN Optical Switch with default address 115 is 115 x 2 = 230 (0xE6).
3. During transmission the Slave must acknowledge all bytes using a low-going ACK (acknowledge) pulse (SDA low). Upon acknowledging receipt of the byte, the Slave leaves the SDA high so that the Master can generate the STOP condition if desired.
4. If the ACK pulse (SDA low) is not received, the Master must abort the transfer.

The figure below illustrates the I²C write operation for the MEMS 1xN Optical Switch:

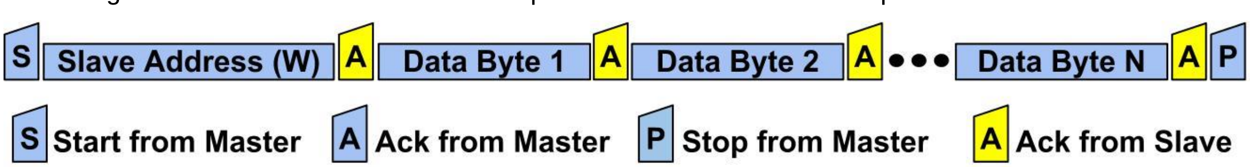


Figure 5. I²C Write Operation

5.3.3 I^{2}C Slave-to-Master Communication

To use the I^{2}C interface for receiving data (Slave-to-Master):

1. The Master sends the START condition and the address byte.
2. The address byte for a READ operation is the 7-bit slave address with the READ/-WRITE bit set to 1. Therefore, the effective read address for a MEMS 1xN Optical Switch with the default address is (115*2) + 1 = 231 (0xE7).
3. After acknowledging its READ address, the Slave sends bytes to the Master. The Master acknowledges all bytes except the last one by using a low-going ACK (acknowledge) pulse (SDA low).
4. Upon acknowledging receipt of the byte, the Master leaves the SDA high.

Note that typically a read operation is preceded by a write operation for a query command.

The figure below illustrates the I^{2}C read operation for the MEMS 1xN Optical Switch:

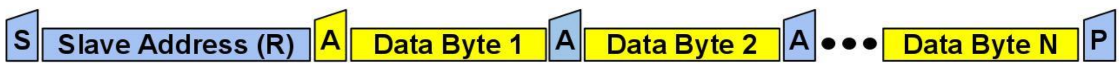


Figure 6. I^{2}C Read Operation

5.3.4 Device Response

Every command will generate a reply from the device. The reply acknowledges that the command was completed successfully, or indicates an error occurred by including the bit 0x80 in the command code byte. When an error occurs, the reply will include a single data byte that is the error code. See error codes in **Table 11**.

5.3.5 I^{2}C Command Sets

Table 10. I^{2}C Command Codes and Description*

Code	Command Name	Description
0x30	Polling Status	Gets the system status/error
0x31	Get Device Info	Gets the switch's identification string
0x32	Get Firmware Version	Gets the switch's firmware version
0x33	Get Serial Number	Gets the switch's serial number
0x35	Get Firmware Part Number	Gets the switch's firmware part number
0x36	Get Hardware Part Number	Gets the switch's hardware part number
0x37	Set I^{2}C Address	Sets the switch's I^{2}C address
0x38	Reset	Resets the switch
0x70	Get Device Dimension	Gets the input/output channel dimensions of the switch
0x78	Set Output Channel	Sets the state of the optical switch to the output channel N
0x79	Get Output Channel Number	Gets the output channel number of the switch

* *Commands 0x32, 0x33, 0x35, 0x36, 0x37 and 0x38 are supported starting from firmware 97198 Rev.C4.*

0x30 Polling Status

Description	Gets the system status/error
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Fixed length, 1 data byte
Reply Data	Byte 3: Status
Example	`(Tx): STA, 0x30, CRC16` `(Rx): STA, 0x30, 0x00, CRC16`

0x31 Get Device Info

Description	Gets the switch's identification string. The identification string is comprised of four comma separated strings: 1. Device manufacturer name 2. Device model name 3. Device firmware number and version 4. Device serial number
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Variable length
Reply Data	Byte3: Length of reply string Byte4 ~: Device identification string
Example	`(Tx): STA, 0x31, CRC16` `(Rx): STA, 0x31, 0x3A, 'DiCon Fiberoptics Inc, MS1x36, FW97198 Rev.C4,60A0EM2G0001', CRC16`

0x32 Get Firmware Version

Description	Gets the switch's firmware version
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Fix length, 7 data byte
Reply Data	String of firmware version
Example	`(Tx): STA, 0x32, CRC16` `(Rx): STA, 0x32, '3.4.0.5', CRC16`

* Command 0x32 is supported starting from firmware 97198 Rev.C4.

0x33 Get Serial Number

Description	Gets the switch's serial number. One string: 1. Device's serial number
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Variable length
Reply Data	Byte3: Length of reply string Byte4 ~: Device's serial number
Example	`(Tx): STA, 0x33, CRC16` `(Rx): STA, 0x33, 0x0C, '60A3FS2G0001', CRC16`

* Command 0x33 is supported starting from firmware 97198 Rev.C4.

0x35 Get Firmware Part Number

Description	Gets the switch's firmware part number. One string: 1. Device's firmware part number
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Variable length
Reply Data	Byte3: Length of reply string Byte4 ~: Device's firmware part number
Example	`(Tx): STA, 0x35, CRC16` `(Rx): STA, 0x35, 0x07, '97198C4', CRC16`

* Command 0x35 is supported starting from firmware 97198 Rev.C4.

0x36 Get Hardware Part Number

Description	Gets the switch's hardware part number. One string: 1. Device's hardware part number
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Variable length
Reply Data	Byte3: Length of reply string Byte4 ~: Device's hardware part number
Example	`(Tx): STA, 0x36, CRC16` `(Rx): STA, 0x36, 0x07, '32781B2', CRC16`

* Command 0x36 is supported starting from firmware 97198 Rev.C4.

0x37 Set I^2C Address*

Description	Sets the I^2C address
Command Packet Type	Fixed length, 1 data byte
Command Parameters	Byte 3: I^2C address (I^2C address can be set to any address between 0 to 127 in decimal.)
Reply Packet Type	None
Reply Data	None

Example	`(Tx): STA, 0x37, 0x74, CRC16`
	`(Rx):`
	
	The switch's default I^2C address is 115 in decimal (0x73 in hex). This example sets I^2C address to 116 in decimal (0x74 in hex).
	Power cycle is needed after setting the I^2C address.

* Command 0x37 is supported starting from firmware 97198 Rev.C4.

0x38 Reset

Description	Soft reboot by restarting the microprocessor
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	None
Reply Data	None
Example	`(Tx): STA, 0x38, CRC16` `(Rx):`

* Command 0x38 is supported starting from firmware 97198 Rev.C4.

0x70 Get Device Dimensions

Description	Gets the input/output channel dimensions of the switch
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Fixed length, 2 data bytes
Reply Data	Byte3: Maximum input channels Byte4: Maximum output channels

Example 1	`(Tx): STA, 0x70, CRC16`
	`(Rx): STA, 0x70, 0x01, 0x0C, CRC16`
	
	The reply indicates that this is a 1x12 switch.

Example 2	`(Tx): STA, 0x70, CRC16`
	`(Rx): STA, 0x70, 0x01, 0x20, CRC16`
	
	The reply indicates that this is a 1x32 switch.

0x78 Set Output Channel

Description	Sets the state of the optical switch to the output channel N
Command Packet Type	Fixed length, 1 data byte
Command Parameters	Byte 3: Output channel number ***n*** (***n*** or 0) The commanded output channel should be an integer from 0 to n, where ***n*** is the number of channels in the switch (ex. For a 1x12 switch, ***n*** is 12). Commanding the switch to position 0 will set the switch to the parking position. Note for 2x2 Switches – Standard & Add Drop: Bypass State, set output channel number = 1 Inserted State, set output channel number = 2 Note for 2x2 Switches – Blocking: Config 1, set output channel number = 1 Config 2, set output channel number = 2 Config 3, set output channel number = 3 Config 4, set output channel number = 4
Reply Packet Type	Fixed length, 1 data byte
Reply Data	Byte 3: Status

Example 1	`(Tx): STA, 0x78, 0x04, CRC16`
	`(Rx): STA, 0x78, 0x00, CRC16`
	
	This example sets switch to channel 4.

Example 2	`(Tx): STA, 0x78, 0x00, CRC16`
	`(Rx): STA, 0x78, 0x00, CRC16`
	
	This example sets switch to parking position.

Example 3 (2x2 Switch)	`(Tx): STA, 0x78, 0x01, CRC16`
	`(Rx): STA, 0x78, 0x00, CRC16`
	
	This example sets 2x2 switch to Bypass state.

0x79 Get Output Channel Number

Description	Gets the output channel number of the switch
Command Packet Type	Fixed length, 0 data byte
Command Parameters	None
Reply Packet Type	Fixed length, 2 data byte
Reply Data	Byte 3: Status Byte 4: Current Channel Number

Example	`(Tx): STA, 0x79, CRC16`
	`(Rx): STA, 0x79, 0x00, 0x0B, CRC16`
	
	The switch's output channel is currently set to 11.

Table 11. MEMS 1xN Switch Module Return Codes for I²C Control

Return Code	Description
0	Successful
1	Invalid Command
2	Value Out of Range
3	Command Fail

5.4 Channel in Hex

Table 12. Channel in Hex (up to 56 channels)

Channel	I^2C	Channel	I^2C
Channel 1	0x01	Channel 29	0x1D
Channel 2	0x02	Channel 30	0x1E
Channel 3	0x03	Channel 31	0x1F
Channel 4	0x04	Channel 32	0x20
Channel 5	0x05	Channel 33	0x21
Channel 6	0x06	Channel 34	0x22
Channel 7	0x07	Channel 35	0x23
Channel 8	0x08	Channel 36	0x24
Channel 9	0x09	Channel 37	0x25
Channel 10	0x0A	Channel 38	0x26
Channel 11	0x0B	Channel 39	0x27
Channel 12	0x0C	Channel 40	0x28
Channel 13	0x0D	Channel 41	0x29
Channel 14	0x0E	Channel 42	0x2A
Channel 15	0x0F	Channel 43	0x2B
Channel 16	0x10	Channel 44	0x2C
Channel 17	0x11	Channel 45	0x2D
Channel 18	0x12	Channel 46	0x2E
Channel 19	0x13	Channel 47	0x2F
Channel 20	0x14	Channel 48	0x30
Channel 21	0x15	Channel 49	0x31
Channel 22	0x16	Channel 50	0x32
Channel 23	0x17	Channel 51	0x33
Channel 24	0x18	Channel 52	0x34
Channel 25	0x19	Channel 53	0x35
Channel 26	0x1A	Channel 54	0x36
Channel 27	0x1B	Channel 55	0x37
Channel 28	0x1C	Channel 56	0x38

5.5 CRC Example

An example of a C language function performing CRC generation is shown on the following pages. All of the possible CRC values are preloaded into two arrays, which are simply indexed as the function increments through the message buffer. One array contains all of the 256 possible CRC values for the high byte of the 16–bit CRC field, and the other array contains all of the values for the low byte.

Indexing the CRC in this way provides faster execution than would be achieved by calculating a new CRC value with each new character from the message buffer.

Note: This function performs the swapping of the high/low CRC bytes internally. The bytes are already swapped in the CRC value that is returned from the function. Therefore the CRC value returned from the function can be directly placed into the message for transmission.

The function takes two arguments:

unsigned char *puchMsg; A pointer to the message buffer containing binary data to be used for generating the CRC unsigned short usDataLen; The quantity of bytes in the message buffer.

CRC Generation Function

```
/* The function returns the CRC as an unsigned short type */

unsigned short CRC16 ( puchMsg, usDataLen)
unsigned char *puchMsg ; /* message to calculate CRC upon */
unsigned short usDataLen ; /* quantity of bytes in message */
{
unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
unsigned uIndex ; /* will index into CRC lookup table */
while (usDataLen--) /* pass through message buffer */
{
uIndex = uchCRCLo ^ *puchMsgg++ ; /* calculate the CRC */
uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex} ;
uchCRCHi = auchCRCLo[uIndex] ;
}
return (uchCRCHi << 8 | uchCRCLo) ;
}

/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
};


/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};
```

6. TTL Interface

Warning!

All digital lines are LVTTL. The typical LVTTL voltage for the HIGH state is 3.3 V, and the damage threshold is 3.6 V. Do not apply a voltage higher than 3.6 V to any of the data pins or this will damage the internal PCB and repair will not be covered under warranty.

To clarify, the digital lines are defined by the DiCon pin assignment in table 3 on page 9, and consist of all data inputs D0 – D5 (pins 1, 2, 7, 8, 11 and 12), the busy pin (pin 13), the alarm pin (pin 14), the strobe pin (pin 15), and the reset pin (pin 16).

6.1 Data Inputs D0 – D5 (Pins 1, 2, 7, 8, 11 and 12)

The data inputs D0 – D5 are LVTTL inputs and are used for channel selection. The channel number is defined in the logic table presented in section 6.5 below.

Please note that any unused data inputs must be tied to ground, and not left floating. A floating state on an unused data input could be mistaken as a high state and set the switch to an incorrect switch state. To assure accurate control of the switch, connect all unused data inputs to ground. For example, a 1x4 switch would utilize data inputs D0 and D1, but would not use D2 through D5. In this case, D2 through D5 should be connected to ground.

6.2 Busy (Pin 13)

The busy pin is a LVTTL output that indicates whether the switch is busy or not. A high state indicates that the switch is busy conducting a switch, and commands should not be sent at this time. Please note that use of the busy pin is optional and is not needed in order to operate the switch. It can be helpful however to monitor and assure that the switch is not busy prior to sending a new switch command. If the busy pin is not going to be used, this pin can be left unconnected.

6.3 Alarm (Pin 14)

The alarm pin is a LVTTL output that indicates whether there is an error with the switch. A high state indicates that there is an internal processing or commanding error. Please note that the alarm pin is optional, and does not need to be used in order to operate the switch. It can be helpful to monitor though, to assure that no errors occur. If the alarm pin is not going to be used, then this pin can be left unconnected.

6.4 Strobe (Pin 15)

The strobe pin is a LVTTL input and acts like a ‘Go’ pin. This pin should be set to a high state when the switch module is not changing state. When a switch is desired, the strobe pin should be pulsed low. Upon the falling edge of the strobe pin, the switch module will read the data inputs D0-D5 and then change to the new switch state.

6.5 Parallel Digital I/O Logic Table

Active Channel	D5	D4	D3	D2	D1	D0
CH 01	0	0	0	0	0	0
CH 02	0	0	0	0	0	1
CH 03	0	0	0	0	1	0
CH 04	0	0	0	0	1	1
CH 05	0	0	0	1	0	0
CH 06	0	0	0	1	0	1
CH 07	0	0	0	1	1	0
CH 08	0	0	0	1	1	1
CH 09	0	0	1	0	0	0
CH 10	0	0	1	0	0	1
CH 11	0	0	1	0	1	0
CH 12	0	0	1	0	1	1
CH 13	0	0	1	1	0	0
CH 14	0	0	1	1	0	1
CH 15	0	0	1	1	1	0
CH 16	0	0	1	1	1	1
CH 17	0	1	0	0	0	0
CH 18	0	1	0	0	0	1
CH 19	0	1	0	0	1	0
CH 20	0	1	0	0	1	1
CH 21	0	1	0	1	0	0
CH 22	0	1	0	1	0	1
CH 23	0	1	0	1	1	0
CH 24	0	1	0	1	1	1
CH 25	0	1	1	0	0	0
CH 26	0	1	1	0	0	1
CH 27	0	1	1	0	1	0
CH 28	0	1	1	0	1	1
CH 29	0	1	1	1	0	0
CH 30	0	1	1	1	0	1
CH 31	0	1	1	1	1	0
CH 32	0	1	1	1	1	1
CH 33	1	0	0	0	0	0
CH 34	1	0	0	0	0	1
CH 35	1	0	0	0	1	0
CH 36	1	0	0	0	1	1
CH 37	1	0	0	1	0	0
CH 38	1	0	0	1	0	1
CH 39	1	0	0	1	1	0
CH 40	1	0	0	1	1	1
CH 41	1	0	1	0	0	0
CH 42	1	0	1	0	0	1
CH 43	1	0	1	0	1	0
CH 44	1	0	1	0	1	1
CH 45	1	0	1	1	0	0
CH 46	1	0	1	1	0	1
CH 47	1	0	1	0	1	0
CH 48	1	0	1	0	1	1
CH 49	1	0	1	1	0	0
CH 50	1	0	1	1	0	1
CH 51	1	0	1	1	1	0
CH 52	1	0	1	1	1	1
CH 53	1	1	0	0	0	0
CH 54	1	1	0	0	0	1
CH 55	1	1	0	0	1	0
CH 56	1	1	0	0	1	1

6.6 TTL Control Procedure

The procedure to change the switch state via TTL control is as follows. Please note that all timing requirements in section 6.7 must be followed in order to assure a proper switch occurs:

1) Set the Strobe pin to high, and leave it high until a switch is desired.
2) Set the Data Input pins to the requested switch state.
3) Before commanding a switch, check the busy and alarm pins, if desired.
4) When a switch is desired, pulse the strobe pin low. On the falling edge of the strobe, the MEMS switch will move to the newly requested switch state.

6.7 Parallel Digital I/O Timing Diagram

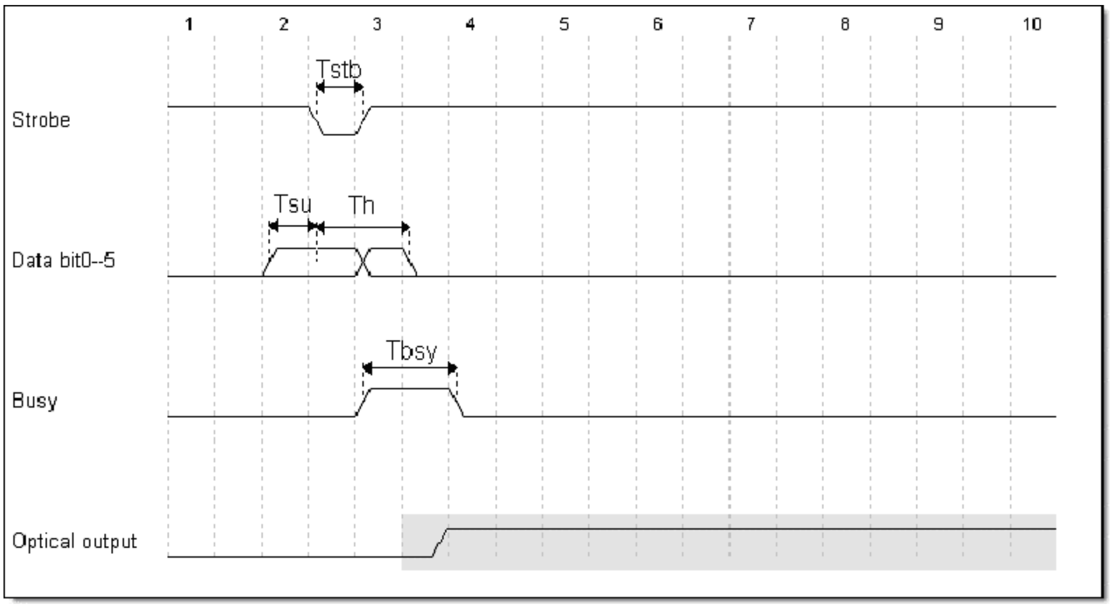


Figure 7. Timing Diagram

Notes:
1. T_{su} is the minimum required data set-up time, relative to the falling edge of Strobe. The channel address <D5:D0> must remain stable preceding the falling edge of Strobe.
2. T_h is the minimum required data hold time, relative to the falling edge of Strobe. The channel address <D5:D0> must remain stable preceding the falling edge of Strobe.
3. T_{stb} is the minimum required pulse width of Strobe

Parameter	Description	Min	Max	Units
T_{su}	Setup time. The channel address (<D5:D0>) must remain stable preceding the falling edge of Strobe.	100	-	µs
T_h	Hold time. The channel address (<D5:D0>) must remain stable following the falling edge of Strobe.	100	-	µs
T_{stb}	Strobe pulse width	1	-	ms
T_{bsy}	Switching time. During this period there may be invalid optical transmission on all channel.	-	30	ms

7. Handling Fiberoptic Components and Cables

Fiber optic components require special handling. Follow these guidelines when handling the cables and connectors.

7.1 Handling Fiber Optic Cables

To avoid cable damage and to minimize optical loss, follow these guidelines when handling fiber optic cables.

- Handle the fiber pigtail outputs carefully.
- The minimum bend radius for most optical cables is 35mm. Never bend an optical cable more sharply than this specification. Optical performance will degrade, and the cable might break.
- Avoid bending the optical cable near a cable strain relief boot. Bending an optical cable near a strain relief boot is one of the easiest ways to permanently damage the optical fiber.
- Avoid bending the optical cable over a sharp edge.
- Avoid using cable tie wraps to hold optical cable. Tie wraps when tightened can create micro-bends or break an optical cable. Microbends can cause a dramatic reduction in optical performance.
- Do not pull on the bare fiber as this can break the fiber inside the component.
- Avoid using soldering irons near optical cables. Accidental damage can easily occur when a soldering iron is used near an optical cable. In addition, solder splatter can contaminate and permanently damage optical fiber connectors.
- To assure the most stable, repeatable optical performance after the optical cables have been connected, immobilize the cables using wide pieces of tape or another form of mechanical cushion.

7.2 Storing Optical Connectors

All switches that include optical connectors are shipped with dust caps covering those optical connectors. Optical connectors should remain covered at all times when the instrument is not in use.

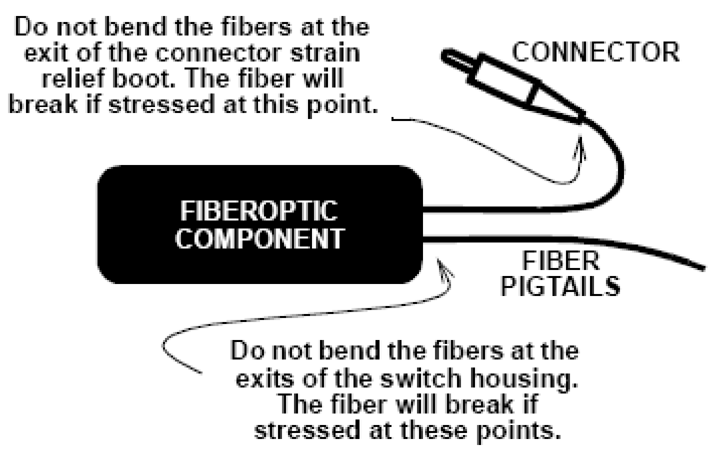


Figure 8. Fiber optic component, connectors, and fiber pigtails

7.3 Cleaning Optical Connectors

Clean any exposed connector using a cleaning kit supplied by the connector manufacturer or high-grade isopropyl alcohol and a cotton swab. To clean with alcohol and a swab, dab the tip of a cotton swab in alcohol and then shake off any excess alcohol. The tip should be moist, not dripping wet. Stroke the swab tip gently across the surface of the connector and around the connector ferrule. Either allow the connector a minute to dry, or blow-dry the connector using compressed air. Be careful when using compressed air: improper use may deposit a spray residue on the connector.

7.4 Mating Optical Connectors

Follow these instructions when mating optical connectors.

- Clean both connectors prior to mating. Any small particles trapped during the mating process can permanently damage the connector.
- Smoothly insert the appropriate connector ferrule into the adapter. Do not allow the fiber tip to contact any surface. If the tip accidentally contacts a surface before mating, stop. Re-clean the connector and try again.
- Tighten the connector until it is finger tight or to the torque specified by the connector manufacturer. Do not over-tighten the connector as this can lead to optical loss and connector damage.
- Check the optical insertion loss. If the loss is unacceptable, remove the connector, re-clean both ends of the mate, and reconnect them. You may have to repeat this process several times before a low-loss connection is made.
- After you make the connection, monitor the stability of the optical throughput for a few minutes. Optical power trending (slowly increasing or decreasing) is caused by the slow evaporation of alcohol trapped in the connector. Continue to monitor optical power until it stabilizes. If the loss is unacceptable, re-clean the connectors and start again.

DiCon
FIBEROPTICS, INC

DiCon Fiberoptics, Inc.
1689 Regatta Boulevard
Richmond, CA 94804
USA

Phone: (510) 620-5200
Fax: (510) 620-4100
Email: info@diconfiberoptics.com
Web: www.diconfiberoptics.com