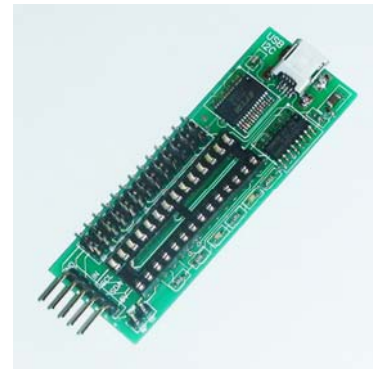


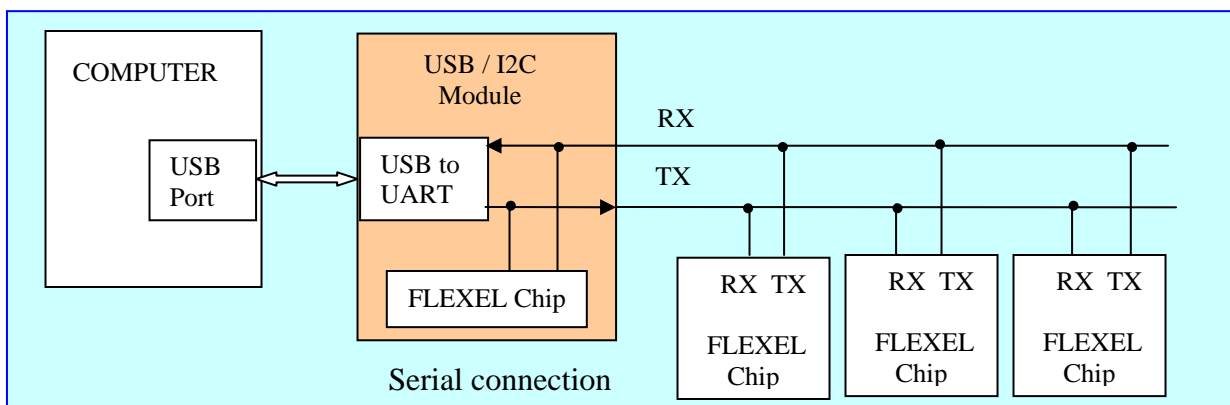
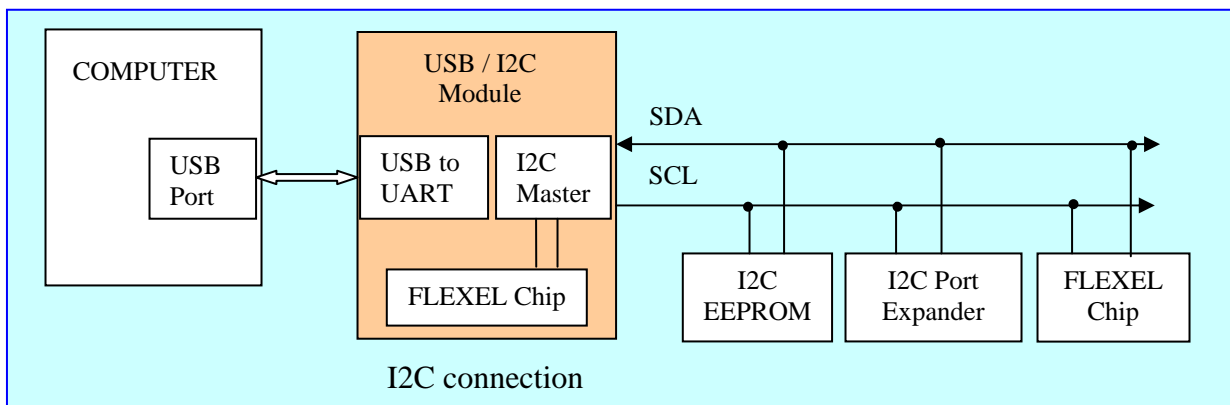
USB to I2C / Serial Communication Module

The USB-I2C module provides a complete interface between PC and the I2C bus. The module can also be used as USB to Serial port converter to connect your peripherals to PC. Its compact size and convenient features make it an ideal tool for your projects.



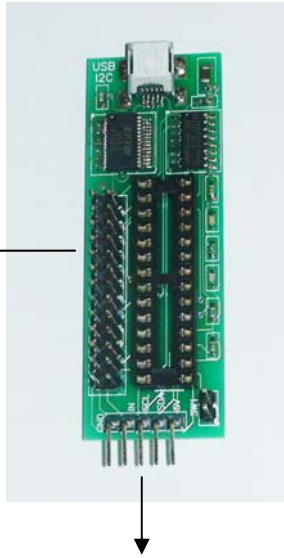
Key Features

- Mini USB connector and Full speed 12 Mbps USB-to-Serial converter.
- Support the dual interface: standard 100Kbps I2C interface and 9600bps Serial interface.
- The module is a master for I2C bus.
- On-board 28 pin socket for FLEXEL chip connection.
- 26 pin connector for FLEXEL inputs/outputs.
- 5 pin connector for I2C/Serial bus.
- On-board the pull up resistors for I2C bus.
- 2 LEDs to indicate USB traffic.
- The module is self powered from the USB and can supply up to 50mA at 5V for external circuitry.
- Small form factor (2.1" x 0.9").



The diagram below shows the module connections.

Pin Name	Pin No.	Pin No.	Pin Name
P0.0	1	2	P2.0
P0.1	3	4	P2.1
P0.2	5	6	P2.2
P0.3	7	8	P2.3
P0.4	9	10	P2.4
P0.5	11	12	P2.5
P0.6	13	14	P2.6
P0.7	15	16	P2.7
SDA	17	18	P1.4
SCL	19	20	P1.5
SEL1	21	22	P1.6
SEL2	23	24	P1.7
GND	25	26	VDD



Pin No.	5	4	3	2	1
Pin Name	GND	IN	SCL	SDA	VDD

Table 1 shows the pin assignment of I2C/Serial connector.

Table 1

Pin No.	Pin Name	Description
1	+5V	The +5V supply from USB. The module can supply up to 50mA to external devices. Leave this pin unconnected if your external device requires more than 50mA, or has its own supply. Do not apply your own 5V supply to this pin.
2	SDA / RX	I2C SDA signal (RX – for serial interface). This pin should be connected directly to the SDA pin on your I2C device.
3	SCL / TX	I2C SCL signal (TX – for serial interface). This pin should be connected directly to the SCL pin on your I2C device.
4	INT	Interface select pin. Leave this pin unconnected if you use I2C interface and connect to ground for Serial interface. Note: You can also use jumper SW1 on board to select the communication interface.
5	GND	Ground pin must be connected to the 0V (Ground) on your external device.

Table 2 shows the pin assignments of FLEXEL connector.

Table 2

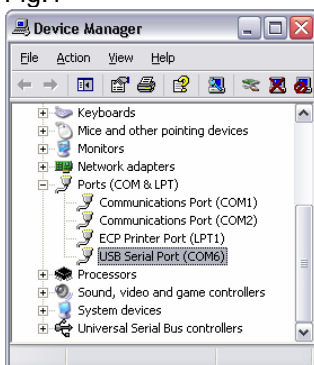
Pin No.	Pin Name	Description
1	P0.0	Port 0, Bit 0
2	P2.0	Port 2, Bit 0

3	P0.1	Port 0, Bit 1
4	P2.1	Port 2, Bit 1
5	P0.2	Port 0, Bit 2
6	P2.2	Port 2, Bit 2
7	P0.3	Port 0, Bit 3
8	P2.3	Port 2, Bit 3
9	P0.4	Port 0, Bit 4
10	P2.4	Port 2, Bit 4
11	P0.5	Port 0, Bit 5
12	P2.5	Port 2, Bit 5
13	P0.6	Port 0, Bit 6
14	P2.6	Port 2, Bit 6
15	P0.7	Port 0, Bit 7
16	P2.7	Port 2, Bit 7
17	SDA	I2C SDA signal (RX – for serial interface)
18	P1.4	Port 1, Bit 4
19	SCL	I2C SCL signal (TX – for serial interface)
20	P1.5	Port 1, Bit 5
21	SEL1	Select the communication protocol , Bit 1
22	P1.6	Port 1, Bit 6
23	SEL2	Select the communication protocol , Bit 2
24	P1.7	Port 1, Bit 7
25	GND	Ground connection
26	VDD	The +5V supply from USB. The module can supply up to 50mA to external devices. Leave this pin unconnected if your external device requires more than 50mA, or has its own supply. Do not apply your own 5V supply to this pin.

Software and Hardware Installation:

1. Install the Virtual Com Port Drivers. The module uses the FTDI FT232R USB chip to communicate with USB. The drivers are available directly from FTDI website (www.ftdichip.com). The drivers appear to the PC as an extra Com Port. Application software accesses the USB device in the same way as it would access a standard Windows Com Port. The serial data format is: one start bit, 8 data bits, and one stop bit. The baud rate is 9600 bit/s.
2. Connect the module to computer USB port. To see which COM port has been assigned to module, right click on “My Computer” desktop icon and select the “Device Manager” tab. Now scroll down and open the “Ports (COM & LPT)” tab. You should see the USB serial port string with port number (Fig.1).

Fig.1



Command Description.

The computer initiates an I2C-bus data transfer through a series of ASCII commands. The commands start from S character (0x53) and terminated with a P character (0x50). Unrecognized commands are ignored by the USB/I2C module.

To prevent the computer from handing the module due to an unfinished command sequence, the USB/I2C module has a time-out feature. The delay between any two bytes of data coming from the computer should be less than 255 ms. If this condition is not met, the module will time-out and clear the receive buffer. The module then starts to wait for the next command from the computer.

The module supports:

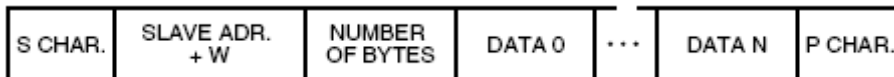
1. Read/Write single byte for non-registered devices (as I/O Expanders).
2. Read multiple bytes without setting new address (as the pressure sensors, ADC, etc).
3. Read/Write single or multiple bytes for 1 byte addressed devices (as EEPROM with 1 byte address, etc).
4. Read/Write single or multiple bytes for 2 byte addressed devices (as EEPROM with 2 byte address).
5. The full range of commands for FLEXEL chips.

Write N bytes to I2C device.

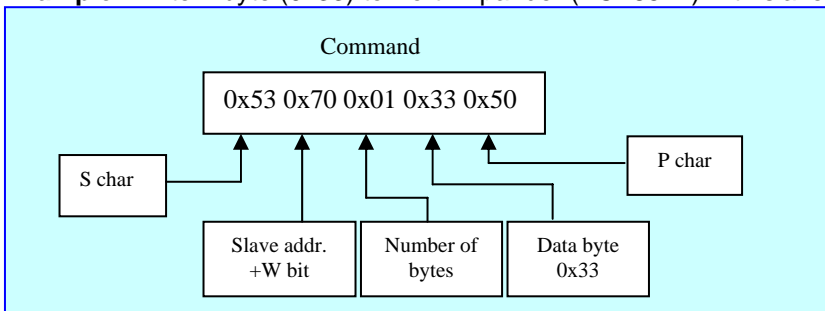
The computer issues the write command by sending an S character followed by an I2C-bus slave device address, the total number of bytes to be sent, and I2C-bus data which begins with the first byte (DATA 0) and ends with the last byte (DATA N). The frame is then terminated with a P character. Once the computer issues this command, the module will access the I2C-bus slave device and start sending the I2C-bus data bytes.

Note that the second byte sent is the I2C-bus device slave address. The least significant bit (W) of this byte must be set to 0 to indicate this is an I2C-bus write command.

The computer sends



Example: Write 1 byte (0x33) to Port Expander (PCF8574) with slave address 0x38

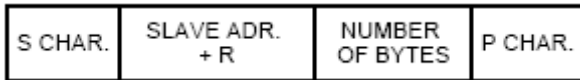


Read N bytes from I2C device.

The computer issues the read command by sending an S character followed by an I2C-bus slave device address, and the total number of bytes to be read from the addressed I2C-bus slave. The frame is then terminated with a P character. Once the computer issues this command, the USB/I2C module will access the I2C-bus slave device, get the correct number of bytes from the addressed I2C-bus slave, and then return the data to the computer.

Note that the second byte sent is the I2C-bus device slave address. The least significant bit (R) of this byte must be set to 1 to indicate this is an I2C-bus read command.

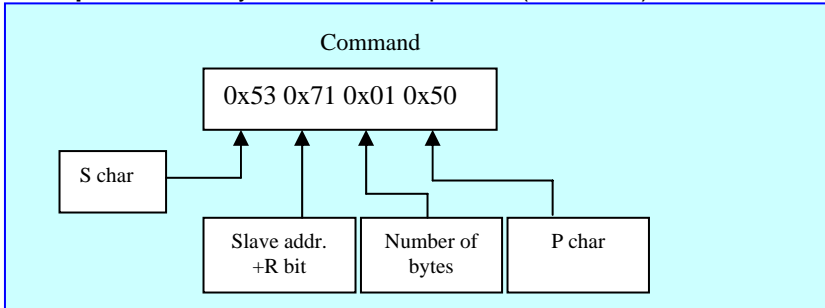
The computer sends



Module responds



Example: Read 1 byte from Port Expander (PCF8574) with slave address 0x38.

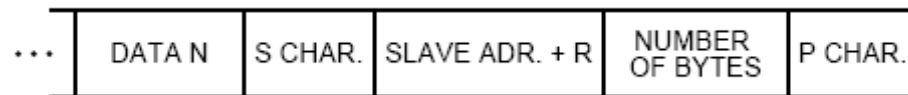
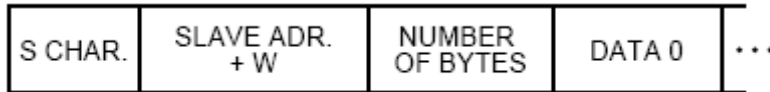


Repeated START: read after write.

The USB/I2C module also supports 'read after write' command as specified in the I2C-bus specification. This allows a read command to be sent after a write command without having to issue a STOP condition between the two commands.

The computer issues a write command as normal, then immediately issues a read command without sending a STOP (P) character after the write command.

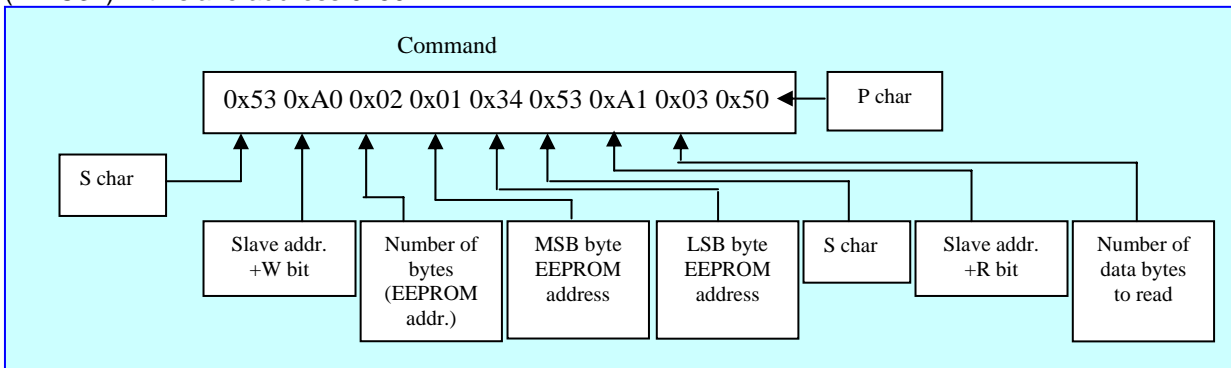
The computer sends



Module responds



Example: Read 3 bytes (start from address 0x0134) from EEPROM with 2 byte address (24LC32) with slave address 0x50.

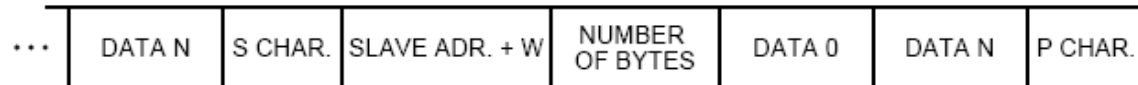
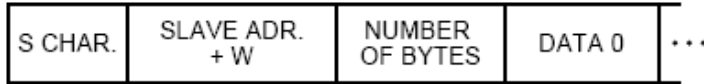


Repeated START: write after write.

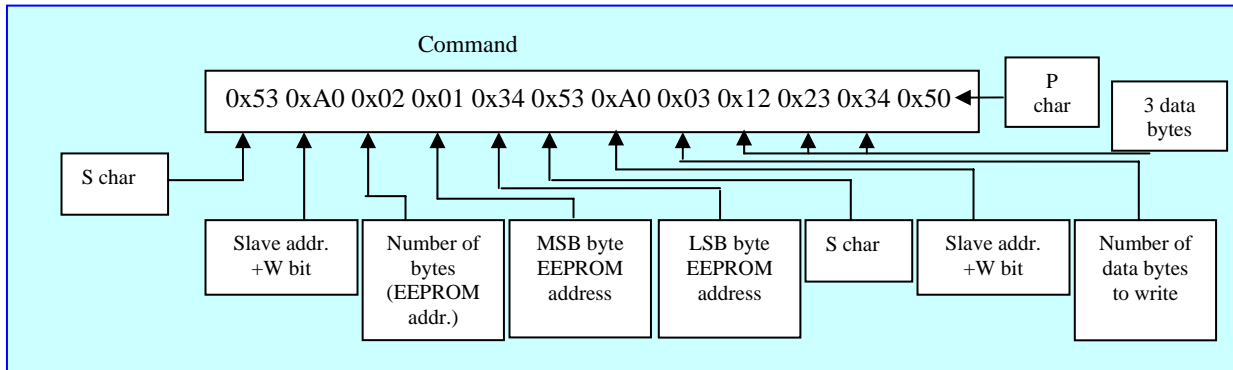
The USB/I2C module also supports 'write after write' command as specified in the I2C-bus specification. This allows a write command to be sent after a write command without having to issue a STOP condition between the two commands.

The computer issues a write command as normal, then immediately issues a second write command without sending a STOP (P) character after the first write command.

The computer sends



Example: Write 3 bytes (start from address 0x0134) to EEPROM with 2 byte address (24LC32) with slave address 0x50.



Communication with FLEXEL Chips.

The computer initiates an I2C-bus data transfer through a series of ASCII commands. The commands start from '/' character (0x2F) and terminated with a CR character (0x0D - carriage return).

See the FLEXEL CHIP USER MANUAL for FLEXEL command description.

Copyright 2008 IMS

Liability Disclaimer

IMS, LLC is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs of recovering, reprogramming, or reproducing any data stored in or used with IMS products.