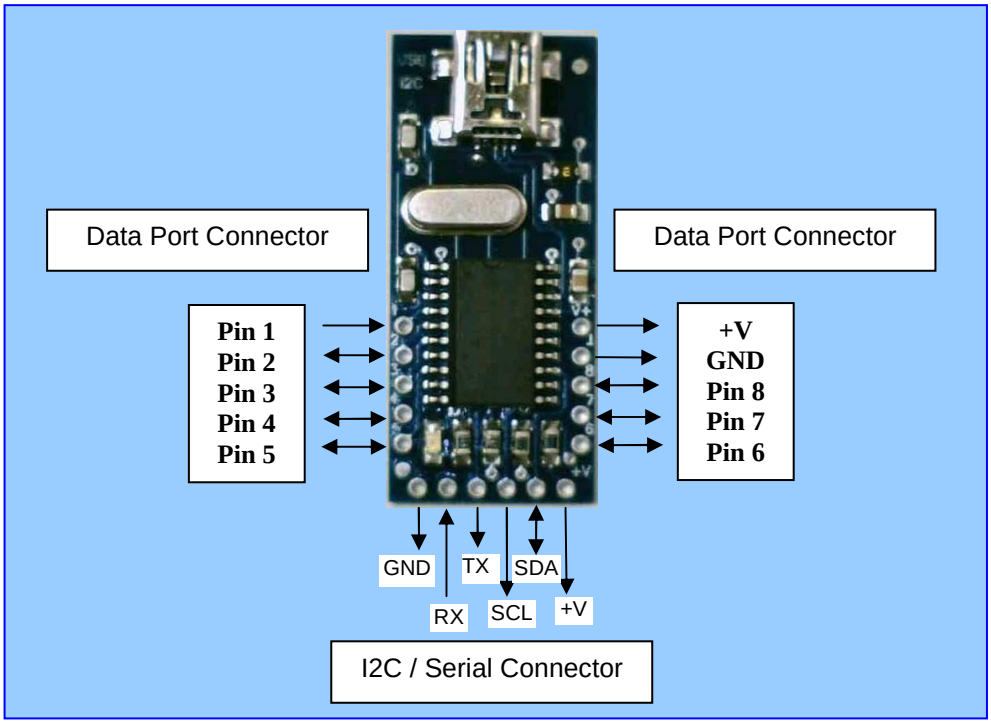
USB to I2C / Serial Communication Module

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 converter.
- Support the dual interface: I2C interface and Serial interface.
- The module is a master for I2C bus.
- Standard 100Kbps and Fast 400Kbps I2C interface.
- 8 Bits Data Port.
- Data Port pins can be set as digital input / output, analog input or PWM output.
- 10 Bit ADC, up to 5 analog inputs
- On-board the pull up resisters for I2C bus.
- LED to indicate USB connection.
- The module is self powered from the USB and can supply up to 50mA at 5V for external circuitry.
- Small form factor (1.6” x 0.7”).



I2C / Serial Connector

Pin No.	6	5	4	3	2	1
Pin Name	GND	RX	TX	SCL	SDA	+V

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

Pin No.	Pin Name	Description
1	+V	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	I2C SDA signal. This pin should be connected directly to the SDA pin on your I2C device.
3	SCL	I2C SCL signal. This pin should be connected directly to the SCL pin on your I2C device.
4	TX	Serial TX signal (output). This pin should be connected directly to the RX pin on your device.
5	RX	Serial RX signal (input). This pin should be connected directly to the TX pin on your device.
6	GND	Ground pin must be connected to the 0V (Ground) on your external device.

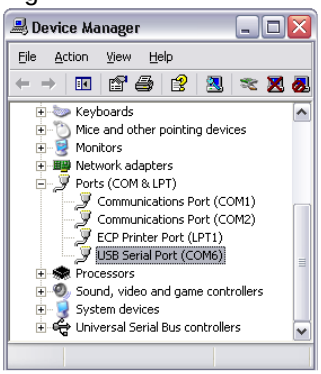
Table 2 shows the pin assignments of **Data Port** connector.
Table 2

Pin No.	Pin Function
1	Digital Input
2	Digital Input, Digital Output, PWM Output
3	Digital Input, Digital Output
4	Digital Input, Digital Output
5	Digital Input, Digital Output, Analog Input
6	Digital Input, Digital Output, Analog Input
7	Digital Input, Digital Output, Analog Input
8	Digital Input, Digital Output, Analog Input
9	GND Ground pin must be connected to the 0V (Ground) on your external device.
10	+V 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 drivers are available directly from website. 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.
2. Connect the module to computer USB port.
3. Once the module has been connected to the PC USB port, Windows “New Hardware Found Wizard” will prompt the user for additional driver information. At this point, the user need only direct Windows to the directory containing the appropriate **mchpcdc*.inf*** file.
4. 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).

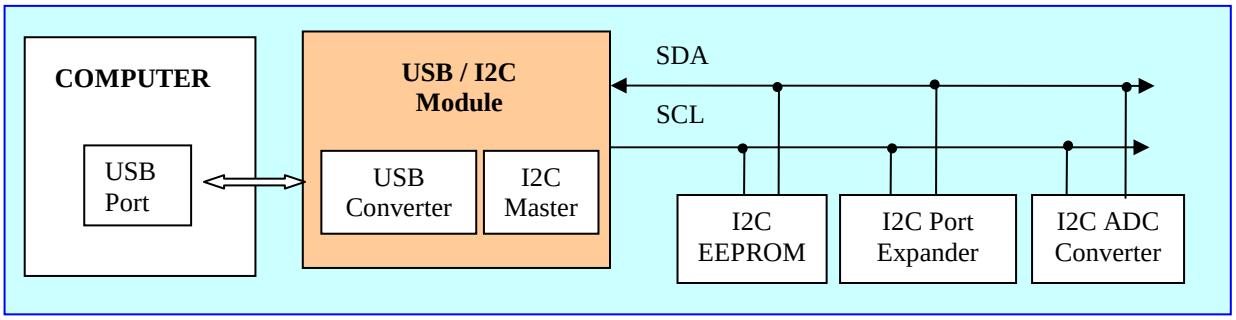
Fig.1



I2C Interface.

The I2C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device.

The diagram below shows the module connections.



Command Description.

The computer initiates an I2C-bus data transfer through a series of commands. The command set includes the commands for module mode setup, I2C communication and interfacing with data port.
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.

Module mode setup commands.

Mode setup command.

The computer issues the write command by sending a command character followed by data byte.

Command	Data
0x96	Data byte

Data byte defines the module mode:

0x01 – I2C 100KBits mode
0x02 – I2C 400KBits mode
0x03 – Serial mode 9600 baud rate

After power up the module default in I2C 100 KBits mode.
The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

Example: Computer sends 2 bytes to set I2C 400 KBits mode: **0x96 0x02**
Note: All commands and data in HEX format.

Read software version.

The computer issues the read software version command by sending a command byte: **0x97**
Module returns data byte with software version number.

I2C communication commands.

The module supports:

1. Read/Write single byte for non-registered devices (as I/O Expanders PCF8574).
2. Read/Write multiple bytes for non-registered devices (as the pressure sensors, ADC).
3. Read/Write single or multiple bytes for 1 byte addressed devices.
4. Read/Write single or multiple bytes for 2 byte addressed devices.
5. Read/Write single or multiple bytes for 1 byte addressed EEPROM.
6. Read/Write single or multiple bytes for 2 byte addressed EEPROM.

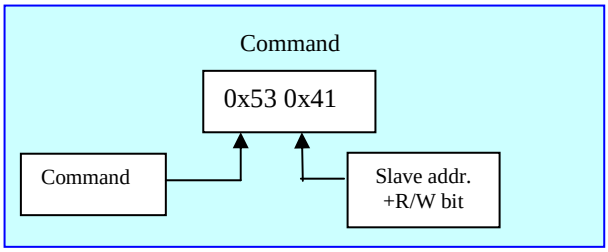
Read/Write single byte for non-registered devices to I2C device.

The computer issues the command by sending a command character **0x53** followed by an I2C-bus slave device address for read command and a command character **0x53** followed by an I2C-bus slave device address and data byte for write command. The computer sends command byte, slave address byte and data byte as one frame. The delay between any two bytes of data coming from the computer should be less than 255 ms.
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 (R/W) of this byte must be set to 1 to indicate this is an I2C-bus read command and 0 to indicate the I2C write command.

Read single byte from non-registered devices

Command	Slave Address+R/W bit
0x53	Address byte

Example: Read data byte from I/O Expander PCF8574 with slave address 0x40.

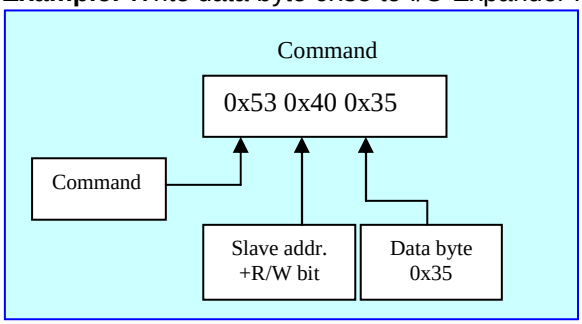


Write single byte to non-registered devices

Command	Slave Address+R/W bit	Data
0x53	Address byte	Data byte

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

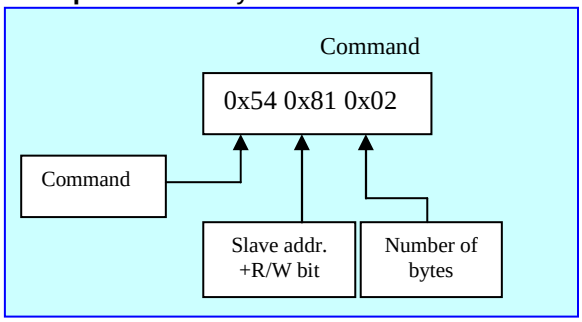
Example: Write data byte 0x35 to I/O Expander PCF8574 with slave address 0x40.



Read multiple bytes from non-registered devices

Command	Slave Address+R/W bit	Number of bytes
0x54	Address byte	Data byte

Example: Read 2 bytes from device with slave address 0x80.

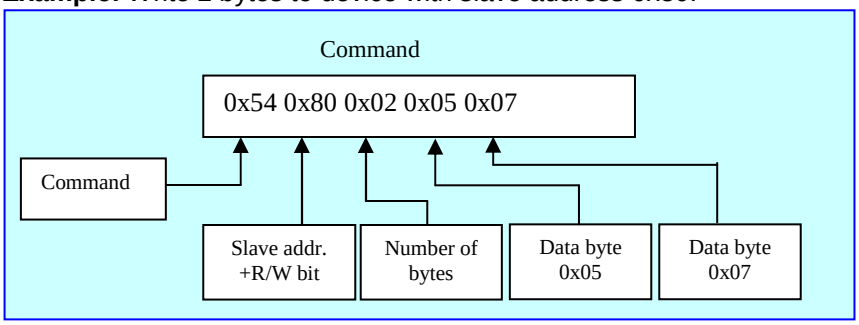


Write multiple bytes to non-registered devices

Command	Slave Address+R/W bit	Number of bytes	Data
0x54	Address byte	Number of bytes	Data bytes

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

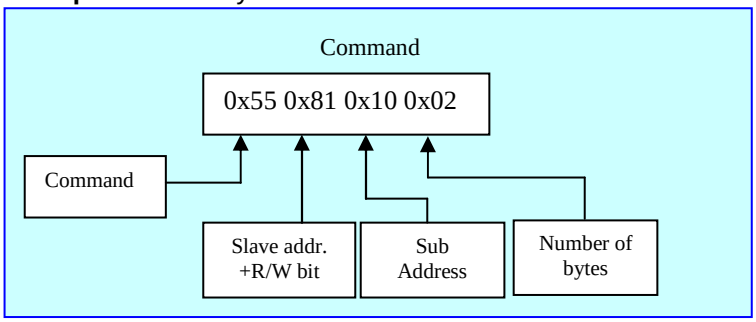
Example: Write 2 bytes to device with slave address 0x80.



Read multiple bytes from 1 byte addressed devices

Command	Slave Address+R/W bit	Sub Address	Number of bytes
0x55	Address byte	Address byte	Data byte

Example: Read 2 bytes from device with slave address 0x80 and sub address 0x10.

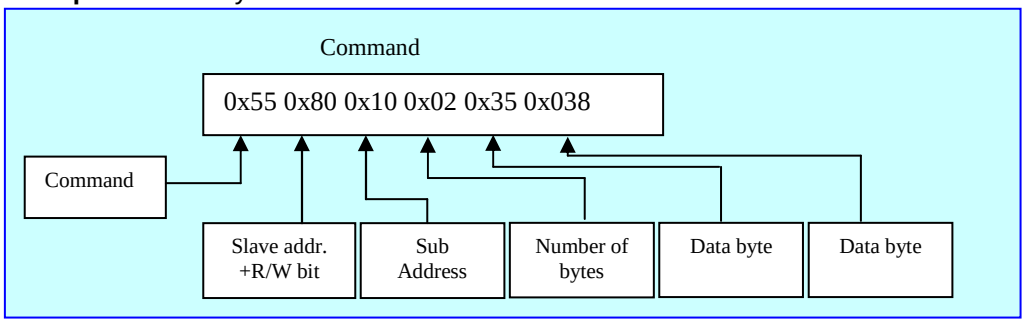


Write multiple bytes to 1 byte addressed devices

Command	Slave Address+R/W bit	Sub Address	Number of bytes	Data
0x55	Address byte	Address byte	Data byte	Data bytes

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

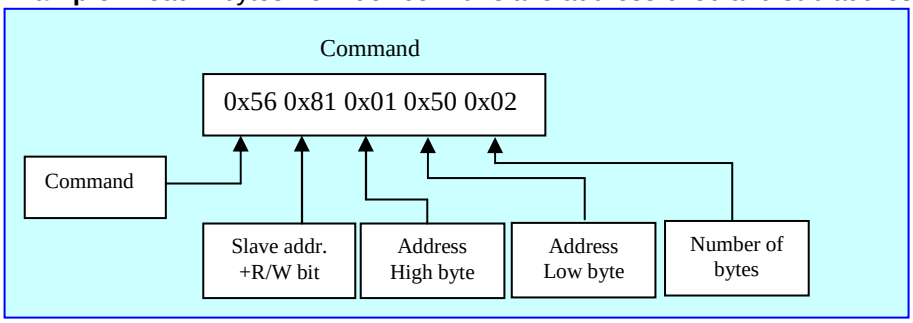
Example: Write 2 bytes to device with slave address 0x80 and sub address 0x10.



Read multiple bytes from 2 byte addressed devices

Command	Slave Address+R/W bit	Sub Address High byte	Sub Address Low byte	Number of bytes
0x56	Address byte	Data byte	Data byte	Data byte

Example: Read 2 bytes from device with slave address 0x80 and sub address 0x0150.

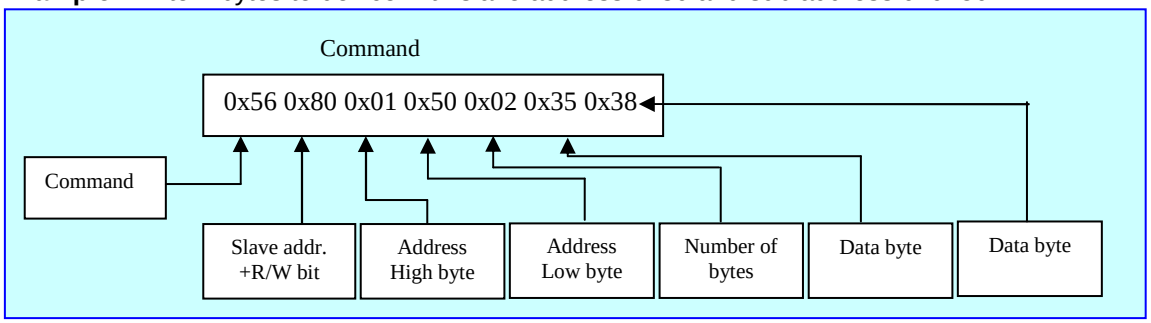


Write multiple bytes to 2 byte addressed devices

Command	Slave Address+R/W bit	Address High byte	Address Low byte	Number of bytes	Data
0x56	Address byte	Address byte	Address byte	Data byte	Data bytes

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

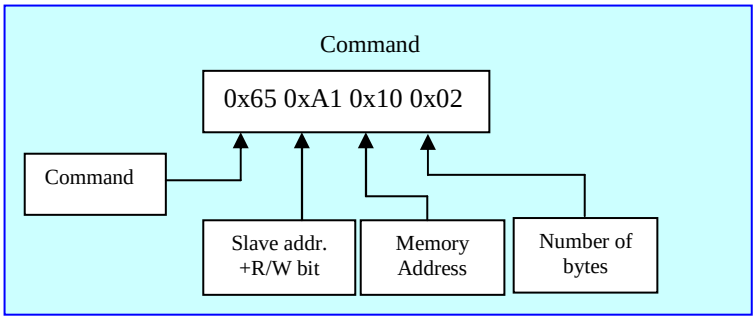
Example: Write 2 bytes to device with slave address 0x80 and sub address 0x0150.



Read multiple bytes from 1 byte addressed EEPROM

Command	Slave Address+R/W bit	EEPROM Address	Number of bytes
0x65	Address byte	Address byte	Data byte

Example: Read 2 bytes from EEPROM with slave address 0xA0 and memory address 0x10.

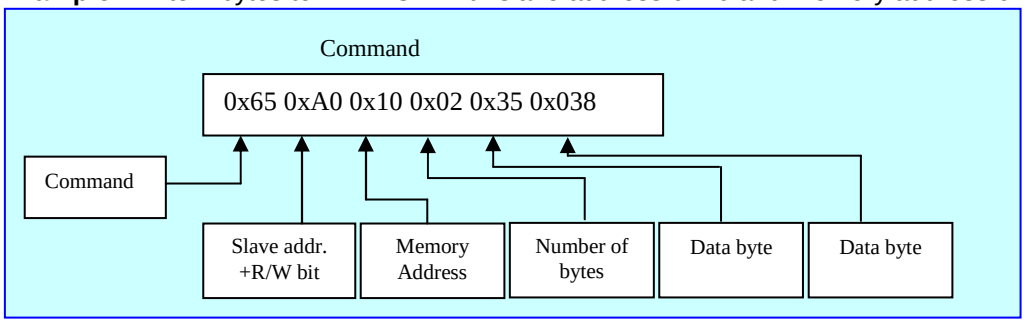


Write multiple bytes to 1 byte addressed EEPROM

Command	Slave Address+R/W bit	EEPROM Address	Number of bytes	Data
0x65	Address byte	Address byte	Data byte	Data bytes

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

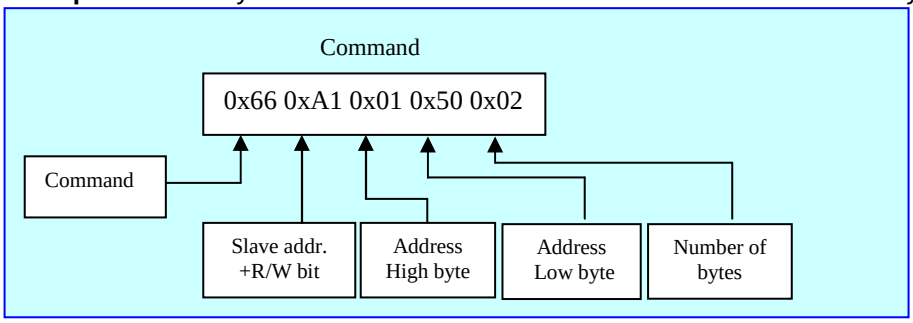
Example: Write 2 bytes to EEPROM with slave address 0xA0 and memory address 0x10.



Read multiple bytes from 2 byte addressed EEPROM

Command	Slave Address+R/W bit	EEPROM Address High byte	EEPROM Address Low byte	Number of bytes
0x66	Address byte	Address byte	Address byte	Data byte

Example: Read 2 bytes from EEPROM with slave address 0xA0 and memory address 0x0150.

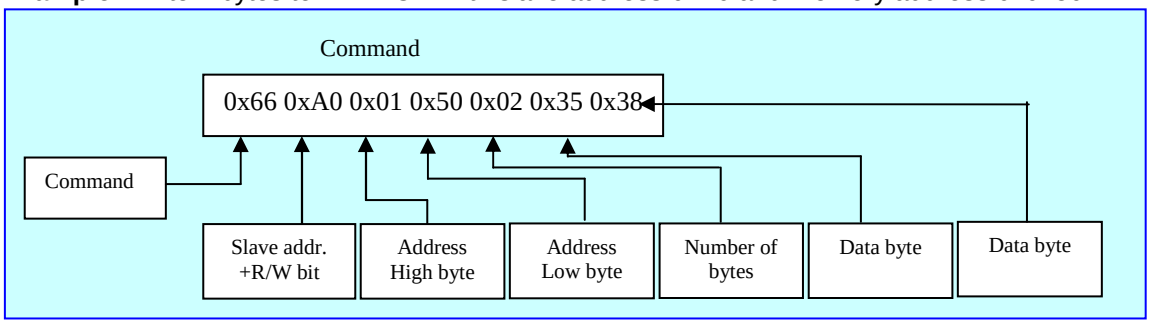


Write multiple bytes to 2 byte addressed EEPROM

Command	Slave Address+R/W bit	EEPROM Address High byte	EEPROM Address Low byte	Number of bytes	Data
0x66	Address byte	Address byte	Address byte	Data byte	Data bytes

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).

Example: Write 2 bytes to EEPROM with slave address 0xA0 and memory address 0x0150.



Data Port Commands.

Set Pin

Command	Pin Number
0x90	Pin2 … Pin8

The command set up data port pin as digital output and sets high logical level (“1”).
The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).
Example: Computer sends 2 bytes to set Pin2 to “1”: **0x90 0x02**

Clear Pin

Command	Pin Number
0x91	Pin2 … Pin8

The command set up data port pin as digital output and sets low logical level (“0”).
The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).
Example: Computer sends 2 bytes to set Pin2 to “0”: **0x91 0x02**

Get Pin

Command	Pin Number
0x92	Pin1 … Pin8

The command set up data port pin as digital input and returns a byte with pin logical level.
Example: Computer sends 2 bytes to read Pin2: **0x92 0x02**

Get Analog

Command	Pin Number
0x93	Pin4 … Pin8

The command set up data port pin as analog input and returns 2 bytes with 10-bits ADC value (MSB - first, LSB - second).
Example: Computer sends 2 bytes to take Pin4 analog value: **0x93 0x04**

Set PWM Value

Command	High byte PWM	Low byte PWM
0x94	Data byte	Data byte

The command set up PWM duty cycle (0…1023).
The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).
Example: Computer sends 3 bytes to set PWM value to 0x0150: **0x94 0x01 0x50**

Set PWM Frequency

Command	Frequency
0x95	0, 1, 2

The command set up data port pin 2 as PWM output and set up PWM frequency.
0 – 3 KHz
1 – 12 KHz
2 – 48 KHz

The module returns 0x00 (zero) byte if the write command succeeded and non-zero if the write failed. The computer should wait for this byte before send the next command (time out after 250mS).
Example: PC sends 2 bytes to set up Pin2 as PWM output with frequency 12 KHz: **0x95 0x01**
Note: Send this command before the Set PWM Value command.

Command Summary.

Command Code	Description
0x96	Module mode setup
0x97	Read software version
0x53	Read/Write single byte for non-registered devices
0x54	Read/Write multiple bytes for non-registered devices
0x55	Read/Write single or multiple bytes for 1 byte addressed devices
0x56	Read/Write single or multiple bytes for 2 byte addressed devices
0x65	Read/Write single or multiple bytes for 1 byte addressed EEPROM
0x66	Read/Write single or multiple bytes for 2 byte addressed EEPROM
0x90	Set data port pin
0x91	Clear data port pin
0x92	Get data port pin value
0x93	Get analog input value
0x94	Set PWM value
0x95	Set PWM frequency