I2C Commands:

# I2C Command Format

### I2C Request Command Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STA** | **COMMAND CODE** | **DATA** | **CRC16** | **P** |
| Byte1 | Byte2 | Byte 3~(N-2) | Byte N-1, N |  |
| address×2  (1~127)×2 | 65 to 110 for private  1 to 64 and 111 to 127 for public | [Data Block] |  |  |

### I2C Response Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STA** | **COMMAND CODE** | **DATA** | **CRC16** | **P** |
| Byte1 | Byte2 | Byte 3~(N-2) | Byte M-1, M |  |
| address×2+1  (1~127)×2+1 | command code | [Data Block] |  |  |

## I2C Error Response Format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STA** | **COMMAND CODE** | **EXCEPTION CODE** | **CRC16** | **P** |
| Byte1 | Byte2 | Byte3 | Byte 4, 5 |  |
| address×2+1  (1~127)×2+1 | 0x80 + command code | 1 to 17 |  |  |

*STA = I2C start with address and R/W bit*

P = I2C stop

***CRC16 = ModBus CRC16 (start from address with R/W bit)***

CRC16 count from STA+Command Code+ Data

**The default address for I2C is 0xXX.**

**(All data are formatted with Big-Endian bit-ordering)**

**Example: Assume the IIC device is 0x73**

**Get Device firmware version  
0xE6,0x31,0x04,0x8A**

***Explanation:***

**0xE6: 0x73×2 (IIC write command)**

**0x31: Get Device firmware version command code**

**0x048A: CRC16**

## Error Code for I2C

|  |  |
| --- | --- |
| **Return Code** | **Description** |
| 0 | Successful |
| 1 | Invalid Command |
| 2 | Value Out of Range |
| 3 | Command Fail |
| 4 | Checksum Fail |
| 16 | Busy signal from devices |
| 17 | Alarm signal from devices |

**Example: Assume the IIC device is 0x73**

## Public Command Code, Parameters and Data Response

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Command | Code | | | Parameters | Data Response | |
| **Get Board Temperature** | | **0x7A** | None | | | Byte3: Temperature (8bit signed integer) (Celsius) |
| (T): STA,0x7A,CRC16  (R): STA,0x7A,0x25,CRC16 | | | | | | |
| Get Product ID | **0x31** | | | None | Byte3: data length  Byte4~: Manufacturer and Device name (ASCII) | |
| (T): STA,0x31,CRC16  (R): STA,0x31,0x19,’ DiCon Fiberoptics Inc,PROJECTNAME’,CRC16 | | | | | | |
| Get Device Firmware version | **0x32** | | | None | Byte3~6: firmware version, format “(8bit unsigned int). (8bit unsigned int). (8bit unsigned int). (8bit unsigned int)” | |
| (T): STA,0x32,CRC16  (R):  **0xE7, 0x32, 0x01, 0x05, 0x06, 0x05, 0x56, 0xFC (ie. firmware version: 1.5.6.5)** | | | | | | |
| Get Device Serial Number | **0x33** | | | None | Byte3: data length (8bit unsigned int)  Byte4~: serial number (ASCII) | |
| (T): STA,0x33,CRC16  (R): STA,0x33,0x0C,’15A18EG00009’,CRC16  **Exp: 0xE7, 0x33, 0x0C, 0x31, 0x35, 0x41, 0x31, 0x38, 0x45, 0x47, 0x30, 0x30, 0x30, 0x30, 0x39, 0xDD, 0x1E** | | | | | | |
| Get Firmware Part Number | **0x35** | | | None | Byte3: data length (8bit unsigned int)  Byte4~: FW part number (ASCII) | |
| (T): STA,0x35,CRC16  (R): STA,0x35,0x06,’ 97305C’,CRC16 | | | | | | |
| Get Hardware Part Number | **0x36** | | | None | Byte3: data length (8bit unsigned int)  Byte4~: HW part number (ASCII) | |
| (T): STA,0x36,CRC16  (R): STA,0x36,0x06,’ 33251C’,CRC16 | | | | | | |
| Set I2C address | **0x37** | | | Byte3: address (from 1 to 126 ) (8bit unsigned int) | Byte3: Error code | |
| (T): STA,0x37,0x73,CRC16  (R): STA,0x37,0x00,CRC16 (Successful return, note that the address will be active on next power on)  (R): Could be “STA,0xB7,0x02,CRC16” (Failed case, 0x02: Value Out of Range) | | | | | | |
| Reset Device | **0x38** | | | None | None | |
| (T): STA,0x38,CRC16 (Set the command to the Device, which will reset the device) | | | | | | |
| **Factory Reset** | | **0x3B** | None | | | None |
| (T): STA,0x3B,CRC16 | | | | | | |
| Get Device Dimension | **0x70** | | | None | Byte3: Number of channels (8 bit unsigned int) | |
| (T): STA,0x70,CRC16  (R): STA,0x70,0x08,CRC16 | | | | | | |