

**IPC5000 Controller  
Communications User Manual  
57-77-25-20**

---

# Notice

Copyright 2004 by Honeywell

Release 0 March 2004

## Warranty / Remedy

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Contact your local sales office for warranty information. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace without charge those items it finds defective. The foregoing is Buyer's sole remedy and is **in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose**. Specifications may be changed without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in their application.

## Industrial Measurement and Control

Honeywell Pte LTD

Honeywell Building

17 Changi Business Park Central 1

Singapore

---

---

<b>1. INTRODUCTION</b> .....	<b>1</b>
<a href="#">1.1 Overview</a> .....	1
<a href="#">1.2 Modbus/TCP Interface</a> .....	1
<a href="#">1.3 Modbus RTU RS422/485 Communication Ports</a> .....	4
<b>2. IEEE 32-BIT FLOATING POINT REGISTER INFORMATION</b> .....	<b>9</b>
<a href="#">2.1 IEEE Floating Point Data Format</a> .....	9
<b>3. MODBUS/TCP &amp; MODBUS RTU FUNCTION CODES</b> .....	<b>14</b>
<a href="#">3.1 Function Code 01 – Read Digital Output Status</a> .....	16
<a href="#">3.2 Function Code 02 – Read Digital Input Status</a> .....	18
<a href="#">3.3 Function Code 03/04 – Read Input Registers</a> .....	19
<a href="#">3.4 Function Code 05 - Force Single Digital Output</a> .....	21
<a href="#">3.5 Function Codes 06 - Preset Single Register</a> .....	22
<a href="#">3.6 Function Code 08 - Loopback Message</a> .....	23
<a href="#">3.7 Function Codes 16(10h) – Preset Multiple Registers</a> .....	24
<a href="#">3.8 Function Code 17(11h) – Report Device ID</a> .....	25
<b>4. MODBUS RTU EXCEPTION CODES</b> .....	<b>28</b>
<a href="#">4.1 Introduction</a> .....	28
<b>5. REGISTER MAP FOR PROCESS AND OPERATION TYPE VARIABLES</b> .....	<b>30</b>
<a href="#">5.1 Register Map Overview</a> .....	30
<a href="#">5.2 Loop Value Register Map</a> .....	32
<a href="#">5.3 Example for queries using Function Codes 3, 6,16</a> .....	34

---

---

<a href="#">5.4 Analog Input Value Register Map – Function Code 03,04</a>	35
<a href="#">5.5 Time Register Map</a>	36
<a href="#">5.6 Set Point Program Register Maps</a>	37
<a href="#">5.7 Set Point Programmer Value Register Map</a>	40
<a href="#">5.8 Set Point Programmer Register Map for IPC5000</a>	41
<a href="#">5.9 Set Point Programmer Segment Register Map</a>	43
<a href="#">5.10 Segment Register Map</a>	45
<a href="#">5.11 Example for determining a Segment Register</a>	45
<a href="#">5.12 Segment Set Point Event Register Map</a>	46
<a href="#">5.13 Example for determining a Segment Event Register</a>	48
<a href="#">5.14 Program Link Register Map</a>	49
<a href="#">5.15 PID group register Map</a>	51
<a href="#">5.16 Controller group register Map</a>	53
<a href="#">5.17 Program group register Map</a>	54
<a href="#">5.18 Setup Group Register Map</a>	56
<a href="#">5.19 Operation group Register Map</a>	59
<a href="#">5.20 No Loop group Register Map</a>	60
<a href="#">5.21 Fix PV Event group Register Map</a>	61
<b><a href="#">6. CRC-16 CALCULATION FOR MODBUS RTU</a></b>	<b>63</b>
<b><a href="#">7. RS-232C COMMUNICATION</a></b>	<b>64</b>
<a href="#">7.1 Overview</a>	64
<a href="#">7.2 Communication Configuration</a>	64

---

---

<b><a href="#">7.3 Communication Function Code</a></b> .....	<b>65</b>
<b><a href="#">7.4 Function Code 20(14h) - Read Configuration Reference Data</a></b> .....	<b>66</b>
<a href="#">7.4.1 Modbus Protocol Read Configuration Example</a> .....	69
<b><a href="#">7.5 Function Code 21(15h) - Write Configuration Reference Data</a></b> .....	<b>74</b>
<a href="#">7.5.1 Modbus Protocol Write Configuration Example</a> .....	76
<b><a href="#">7.6 Read, Write and Override Parameter register addresses</a></b> .....	<b>79</b>
<a href="#">7.6.1 SP/PV block Register address</a> .....	79
<a href="#">7.6.2 SP block Register Map</a> .....	80
<a href="#">7.6.3 PID Constants Register Map</a> .....	86
<a href="#">7.6.4 PID Reference Register address</a> .....	87
<a href="#">7.6.5 Compensate set Register Address</a> .....	88
<a href="#">7.6.6 Fix PV Event Register Address</a> .....	90
<a href="#">7.6.7 Operation button action Register address</a> .....	92
<a href="#">7.6.8 Program Data Register address</a> .....	93
<a href="#">7.6.9 Link Set Register address</a> .....	100
<a href="#">7.6.10 Analog input group Register address(5000H ~ 5013H)</a> .....	102
<a href="#">7.6.11 Analog output group Register address(5020H ~ 503BH)</a> .....	103
<a href="#">7.6.12 Analog PWM group Register address(5040H ~ 504BH)</a> .....	104
<a href="#">7.6.13 Digital output group Register address(5050H ~ 5057H)</a> .....	104
<a href="#">7.6.14 Mode Event group Register address(4300H ~ 434FH)</a> .....	105
<a href="#">7.6.16 Alarm Event group Register address</a> .....	106
<a href="#">7.7 Connection of RS-232C communication</a> .....	108

---



---

# 1. Introduction

## 1.1 Overview

The IPC5000 controller provides Modbus communication support on three communication interfaces.

- Network port: Modbus TCP on an Ethernet connection(Optional)
- RS232 port: Modbus RTU Configuration Interface(Basic)
- RS485 port: Modbus RTU(Optional)

## 1.2 Modbus/TCP Interface

### Introduction

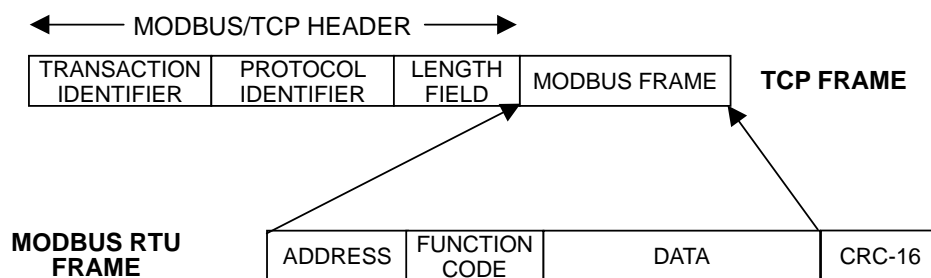
IPC5000 controllers support the Modbus/TCP(also called Modbus TCP/IP or Modbus Ethernet) protocol for communications with third part HMI and SCADA software via a direct Ethernet TCP/IP connection.

The controller's Ethernet 10Base-T Host port is used. Ethernet TCP allows multiple concurrent connections to hosts for data interchange. The IPC5000 supports 5 concurrent host connections using Modbus/TCP protocol message via this port.

### Modbus/TCP Protocol

Modbus/TCP protocol, developed by Groupe Schneider's Modicon Division, is a popular, open standard for data interchange over Ethernet TCP/'IP networks using a Modbus RTU command structure.

It is simply an encapsulation of Modicon's Modbus RTU protocol within a TCP/IP frame as shown below which includes header information and the Modbus frame.



**Figure 1-2.1 Modbus RTU Protocol within a TCP/IP Frame**

The open Modbus/TCP Specification is followed with respect to the physical data link, and network layers. The message structure within the Modbus frame uses standard Modbus RTU function codes.

The Address part of the Modbus frame is not used (set to 00) since there is no sub-addressing intended or required. The

---

controller IP address is the identifying address, set independently at the controller.

The error checking is supported by TCP/IP network protocols and not part of the Modbus frame.

The Transaction Identifiers and Protocol Identifiers in the header are normally all 0's (4 bytes total) while the Length field identifies the number of bytes in the Modbus frame. The controller will transmit the correct number of bytes for the remainder of the frame. However, the controller does not check this field for messages received.

The standard IEEE 32-bit floating point and 16-bit integer formats are used.

## Parameter Addressing

The definition in Table 6-1 is the register map overview listing starting and ending addresses.

Greater detail for parameter addressing relating to a particular function class, e.g, loops, setpoint programmer, etc. is in referenced sub-sections. Function Codes 1, 2, 3, 4, 5, 6, 8, 16 (10h), and 17 (11h) are supported (see Table 4-1 Modbus/TCP and Modbus RTU Function Codes Definitions).

Examples for read or write access to parameters supported by the various function codes are provided in Sections 4.1 through 4.8.

## Reference

The Open Modbus/TCP Specification can be obtained at the Modicon website:

<http://www.modicon.com/openmbus/standards/openmbus.htm>

## IPC5000 Ethernet Communication Setup

See Figure 1-2.2 for setting up the following network parameters;

IP Address, Subnet Mask (optional), Default Gateway IP Address (optional)

1. Be sure the PC, HMI panel, or other Host device has a Network Interface Card (NIC) with an IP address (fixed or DHCP served) that allows access to controllers on the same or other subnet. Consult your IT department or network administrator for allocating IP addresses to the controllers as required.
2. You will need to set each controller's IP address prior to network connection since every IPC5000 controller is shipped with the default IP address of 192.168.0.2. Placing multiple controllers on the same network before they have been given unique IP addresses will cause problems.

Subnet Mask (if other than default, 255.255.255.0), and Default Gateway IP address (if required, otherwise leave at default 192.168.0.2). See your IT network administrator for proper entries.

**Note: After the new network parameters have been downloaded, Please turn off and on the power again.**

The fixed IP address of each controller shall be set independently prior to placing on the network. See your IT systems administrator for allocating IP addresses, subnet masks, or default gateway IP address as necessary (network address filtering and routing may be necessary if the controller network access will not to be confined locally within the plant environment).

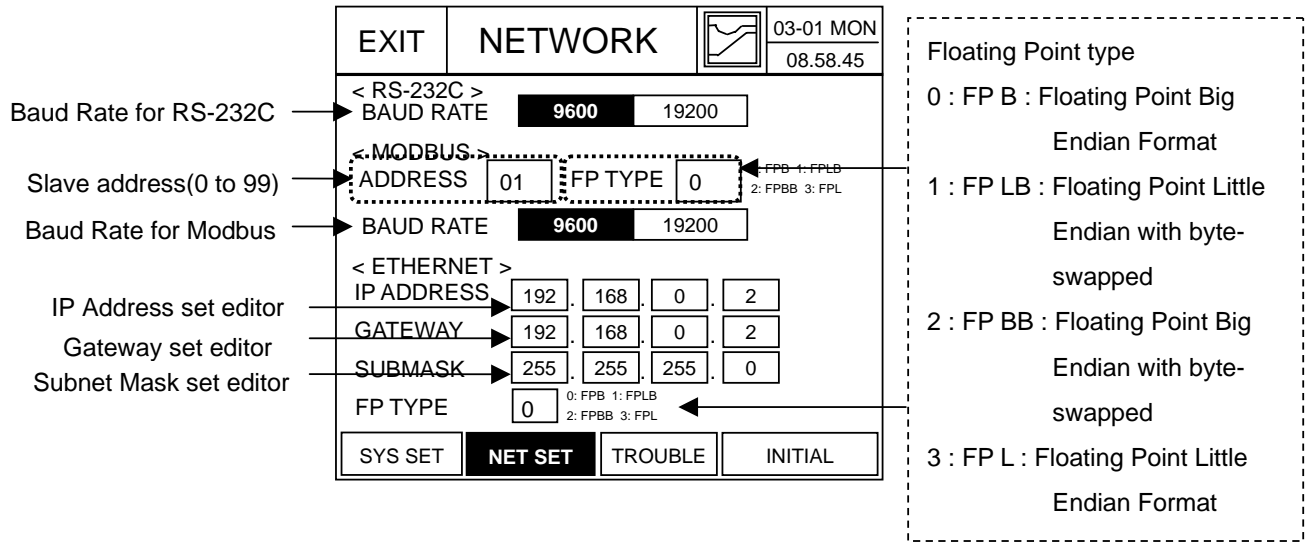


Figure 1-2.2 Communication Setup

### Ethernet 10Base-T Network Connections

Ethernet 10Base-T networks operating at 10MB/sec are supported. A typical network arrangement is as shown below.

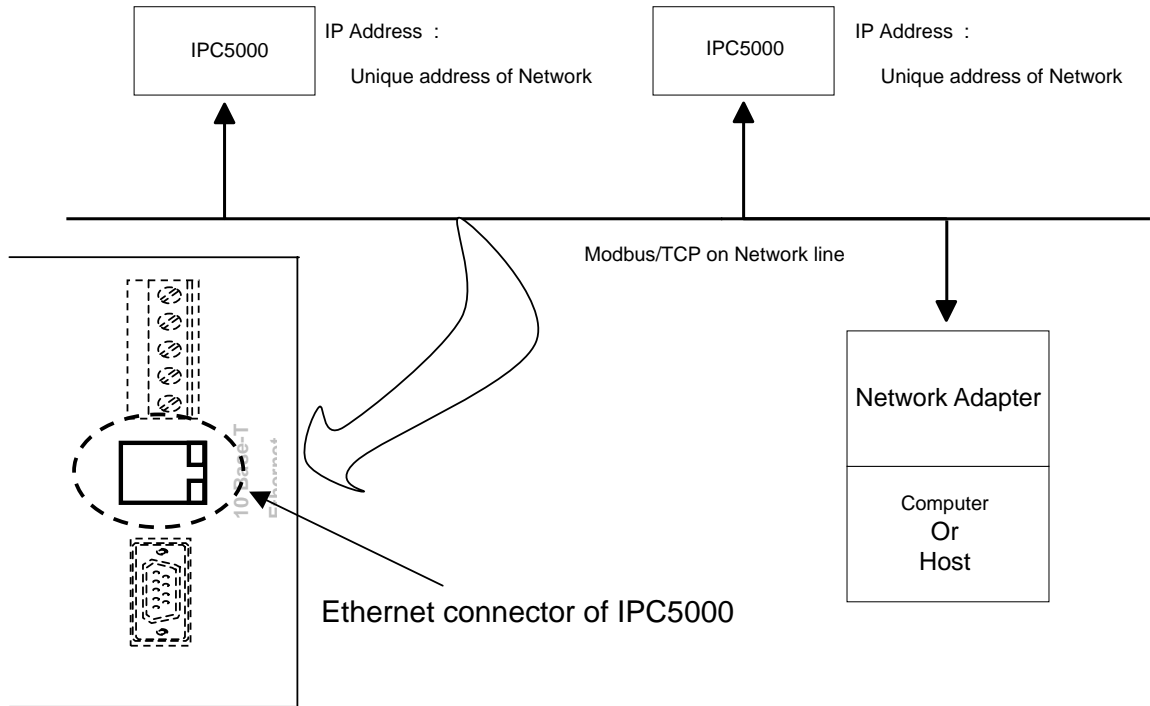


Figure 1-2.3 Ethernet 10Base-T Network Connections

---

## 1.3 Modbus RTU RS422/485 Communication Ports

### Modbus RTU Implementation

This implementation is designed to provide a popular data exchange format connecting the IPC5000 Slave Port (MODBUS) to both Honeywell and foreign master devices. The Modbus RTU allows the instrument to be a citizen on a data link shared with other devices which subscribe to the Modbus RTU RS-485 specification.

These instruments DO NOT emulate any MODICON type device. The Modbus RTU specification is respected in the physical and data link layers. The message structure of the Modbus RTU function codes are employed and standard IEEE 32-bit floating point and integer formats are used. Data register mapping is unique to these instruments. The definition in Table 1-6 is the register mapping for the IPC5000 and the corresponding parameter value.

### Modbus RTU Configuration Interface

This user manual does not include the configuration interfaces for the instruments supporting the Modbus RTU Protocol.

---

## Wiring

The option model has the RS422/485 communication function provides Modbus RTU protocol support.

The LON/MODBUS port allows the IPC5000 controller to network with up to 31 other slave IPC5000 controllers and devices on a Modbus RTU link.

Figure 1-3.1 and Figure 1-3.2 show the LON/MODBUS connector wiring when using either a shielded twisted pair or 4-wire shielded cable.

**(Note) LON means LonWorks communication and IPC5000 now does not support LonWorks.**

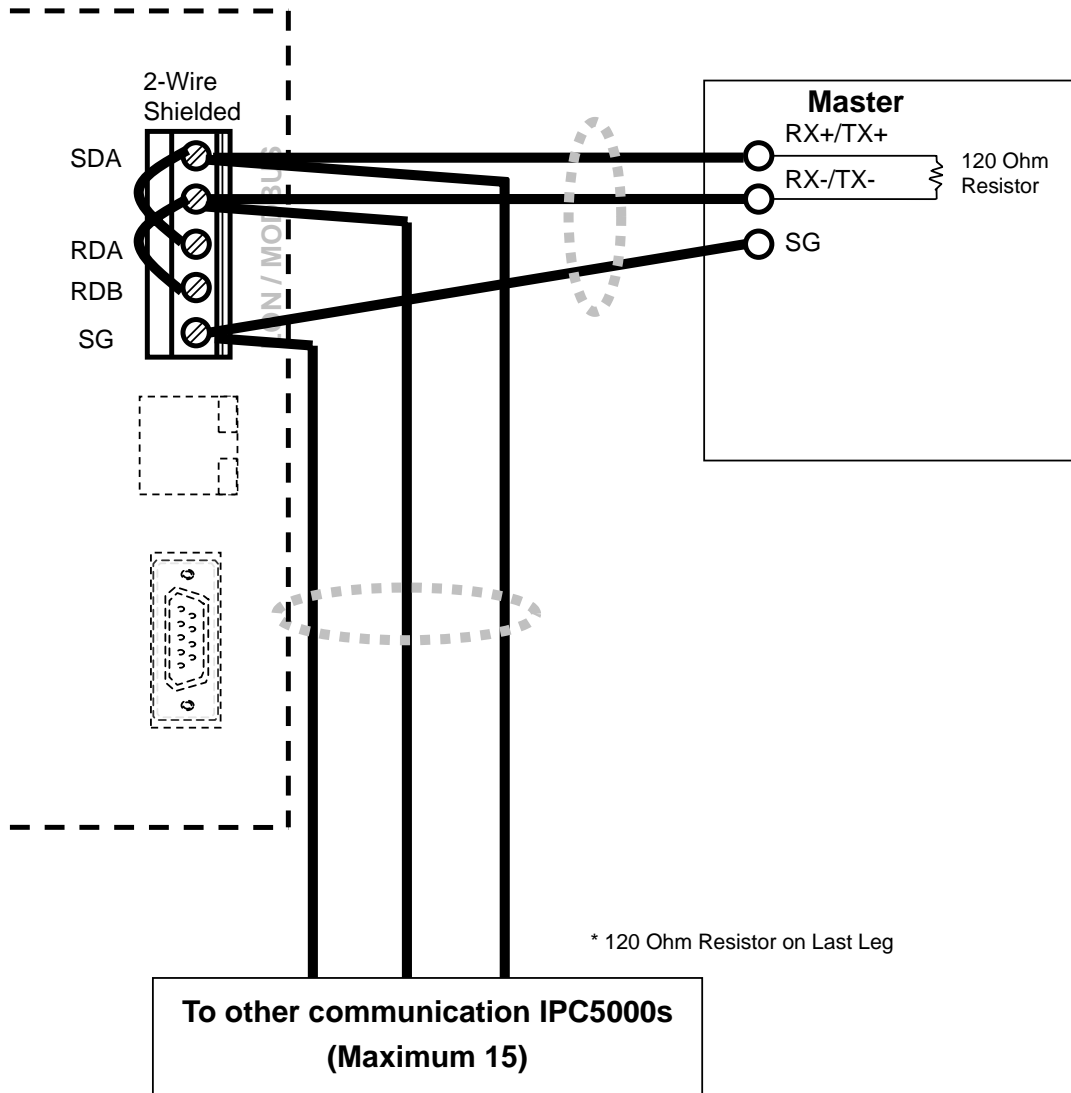


Figure 1-3.1 RS422/485 communication Option Connec(2-Wire shield)

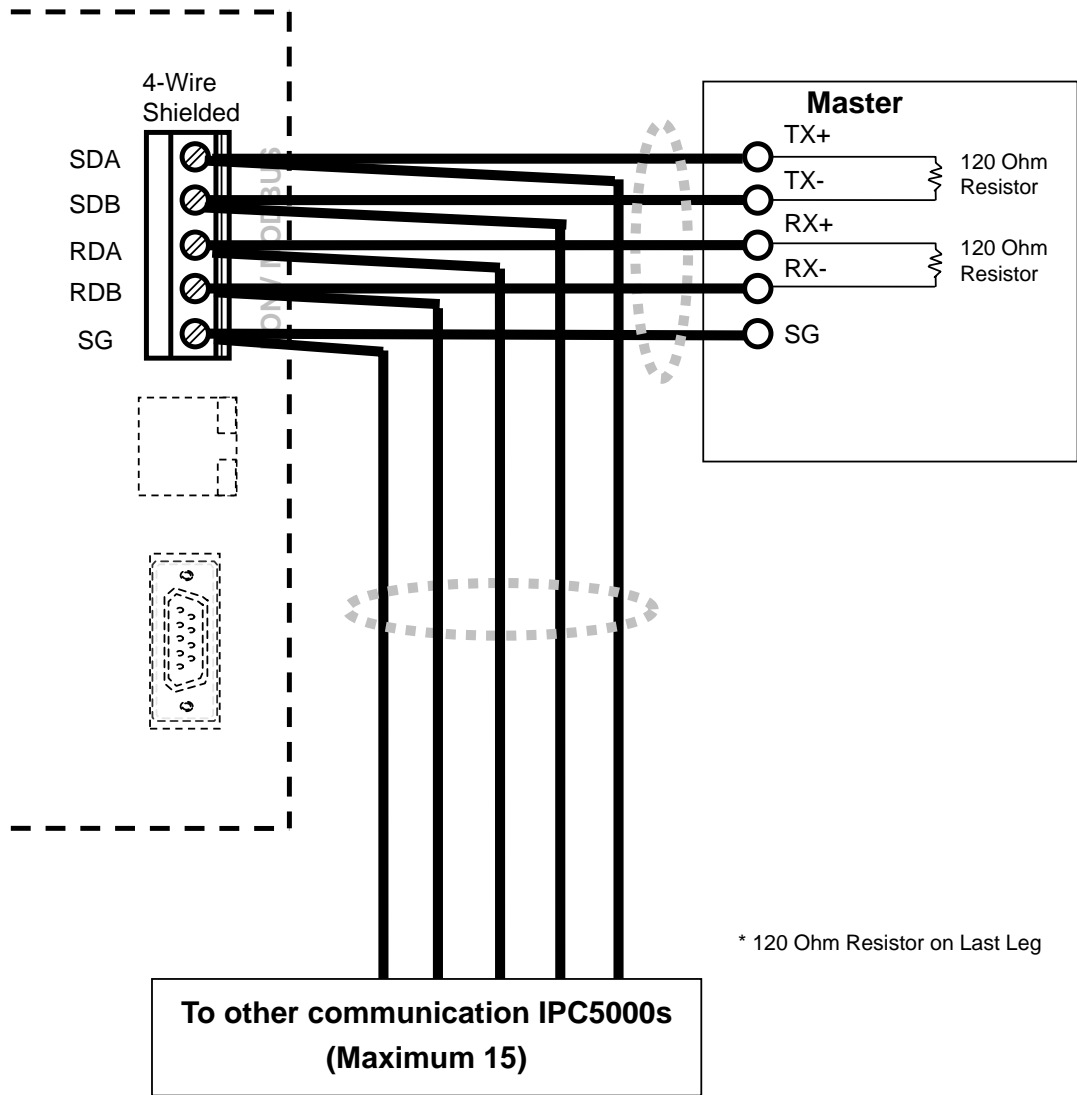


Figure 1-3.2 RS422/485 Communications Option Connect(4-Wire shield)

---

## Setup

### Slave address

Like Figure 1-2.2 of page 22, Slave address can be set from 0 to 99 on NETWORK screen of IPC5000.

### Baud Rate

Like Figure 1-2.2 of page 22, Baud Rate can be set between 9600 and 19200.

## Modbus RTU Message Format

**Table 1-3.1 Modbus RTU Message Formats**

<b>Coding System</b>	8 bit binary
<b>Number of data bits per character</b>	10 Bits start bits – 1 data bits – 8 parity bits – 0 stop bits – 1
<b>Parity</b>	Not used
<b>Bit transfer rate</b>	9600, 19200 Selectable(See Figure 1-2.2 of page 22)
<b>Duplex</b>	Half duplex Transceiver
<b>Error checking</b>	CRC(cyclic redundancy check)
<b>Polynomial</b>	(CRC-16 10100000000001)
<b>Bit transfer order</b>	LSB first
<b>End of message</b>	Idle line for 3.5 or more characters(>1.82 msec for 19200)

## Modbus RTU Link Layer

The link layer includes the following properties/behaviors:

- Slave address recognition,
- Start / End of Frame detection,
- CRC-16 generation / checking.
- Transmit / receive message time-out,
- Buffer overflow detection,
- Framing error detection,
- Idle line detection.

Errors detected by physical layer in message received by the slave are ignored and the physical layer automatically restarts by initiating a new receive on the next idle line detection.

---

## General Modbus RTU message format

### Query message format

[Slave Address, Function Code, Function code dependent data, CRC 16]

### Response message format

[Slave Address, Function Code, Function code dependent data, CRC 16]

\* If an error is detected in a valid message the response function code is modified by adding 80(hex) and the function code dependent data is replaced by an exception response code as described in Section 1.5 – Modbus RTU Exception Codes

Between messages, the RS-485 link is in a high impedance state. During this time receiving devices are more susceptible to noise generated false start of messages. Although noise-generated messages are rejected due to address, framing, and CRC checking, they can cause the loss of a good message when they are included in the message stream. In the slave the transmitting device enables its transmitter line driver and forces an idle line state onto the link for three character time slots prior to transmitting. This forces termination of any noise generated messages and improves message frame synchronization.

## Modbus RTU Data Layer

The data layer includes:

- Diagnostic loopback,
- Function code recognition/rejection,
- Busy/repoll,
- Data error code generation

Errors detected by the data layer are rejected and the slave responds to the polling device with a Modbus type status exception error. A summary of the Modbus status exception codes is listed in Section 1.5 – Modbus RTU Exception Codes.

## 2. IEEE 32-bit Floating Point Register Information

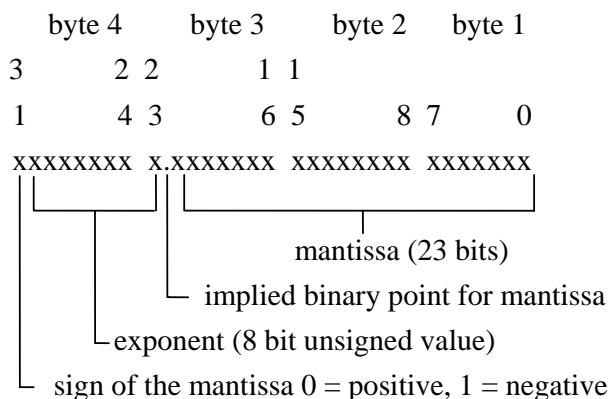
The Modbus applications support IEEE 32-bit floating point information for several of the function codes.

### 2.1 IEEE Floating Point Data Format

The formula for calculating the floating point number is:

$$\text{mantissa} \times 2^{(\text{exponent} - 127)}$$

(23 bit signed binary with 8 bit biased binary exponent)



<Figure 3-1> IEEE Floating Point Data format

#### Mantissa and Sign

The mantissa is defined by a sign bit (31) and a 23-bit binary fraction. This binary fraction is combined with an “implied” value of 1 to create a mantissa value, which is greater than or equal to 1.0 and less than 2.0.

The mantissa is positive if the sign bit is zero (reset), and negative if the sign bit is one (set). For example:

DECIMAL	HEXADECIMAL	BINARY
100	42C80000	01000010 11001000 00000000 00000000

The sign bit (31) is zero, indicating a positive mantissa. Removing the sign bits and exponent bits, the mantissa becomes:

HEXADECIMAL	BINARY
480000	xxxxxxx x1001000 00000000 00000000

Add an “implied” value of one to the left of the binary point:

BINARY
1.1001000 00000000 00000000

Using positioned notation, this binary number is equal to:

$$1.0 + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4}) = 1.0 + 0.5 + 0.0 + 0.0 + 0.0625 = 1.5625$$

---

## Exponent

The exponent is defined by an unsigned 8-bit binary value (bits 23 through 30). The value of the exponent is derived by performing a signed subtraction of 127 (decimal) from the 8-bit exponent value.

DECIMAL	HEXADECIMAL	BINARY
100	42C80000	01000010 11001000 00000000 00000000

Removing the sign and mantissa bits, the exponent becomes:

DECIMAL	HEXADECIMAL	BINARY
133	85	x1000010 1xxxxxxx xxxxxxxx xxxxxxxx

or:

$$1X2^7 + 0X2^6 + 0X2^5 + 0X2^4 + 0X2^3 + 1X2^2 + 0X2^1 + 1X2^0 =$$

Subtract a bias of 127 (decimal) from the exponent to determine its value:  $133 - 127 = 6$ .

## Mantissa and Exponent Combination

Combining the mantissa and exponent from the two previous examples:

$$\text{float number} = \text{mantissa} \times 2^{\text{exponent}}$$

$$\text{float number} = 1.5625 \times 2^6 = 1.5625 \times 64 = 100.0$$

Below is a list of sample float values in IEEE format:

DECIMAL	HEXADECIMAL
100.0	42C0000
-100.0	C2C80000
0.5	3F000000
-1.75	BFE00000
0.0625	3D800000
1.0	3F800000
0.0	00000000
2.0	40000000
55.32	425047AE

---

## Reserved Operands

Per the Standard certain exceptional forms of floating-point operands are excluded from the numbering system. These are as follows:

EXCEPTION	EXPONENT	MANTISSA
+/- Infinity	All 1's	All 0's
Not-a-Number(NAN)	All 1's	Other than 0's
Denormalized Number	All 0's	Other than 0's
Zero	All 0's	All 0's

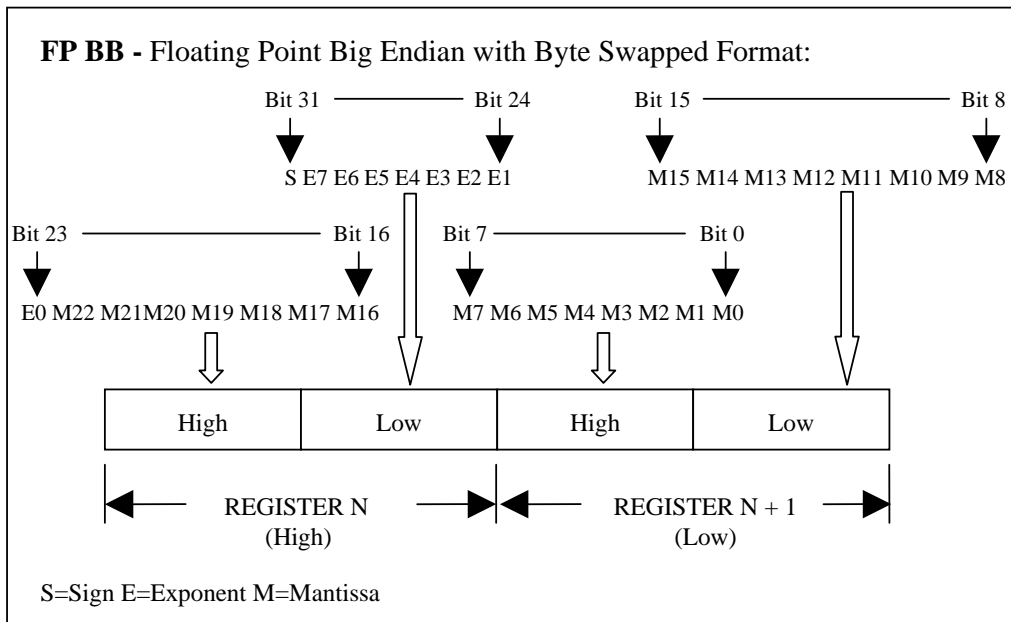
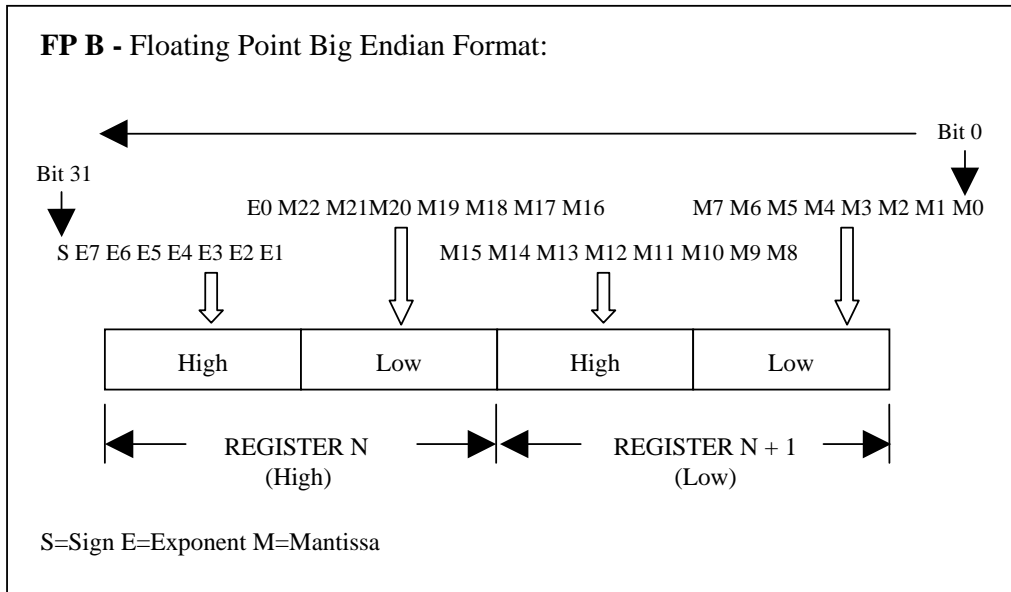
## Modbus Double Register Format

Each IEEE 32-bit floating point number requires two consecutive registers(four bytes) starting with the register defined as the starting register for the information. The stuffing order of the bytes into the two registers differs among Modbus hosts. The selections are : (See Figure 1-2.2 of Page 22 for selecting Floating Point type)

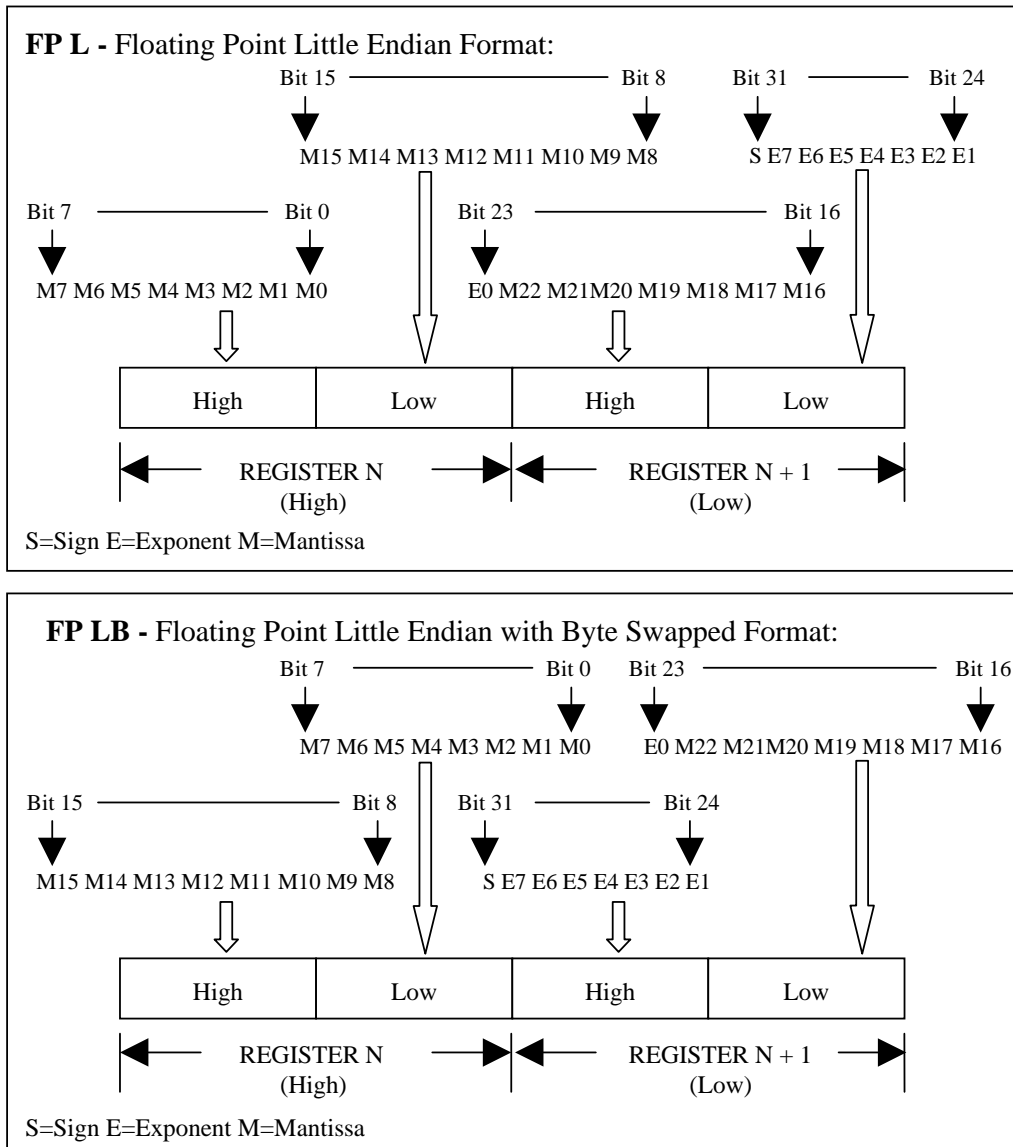
Selection	EXPONENT	Byte order (See Figure 2-1.1)	Notes
FP B	Floating Point Big Endian Format	4, 3, 2, 1	IPC5000 default
FP BB	Floating Point Big Endian with byte-swapped	3, 4, 1, 2	
FP L	Floating Point Little Endian Format	1, 2, 3, 4	
FP LB	Floating Point Little Endian with byte-swapped	2, 1, 4, 3	Modicon and Wonderware standard

See IEEE Formats starting on next Page.

## IEEE Floating Point Formats



Continued next page



**Figure 2-1.1 IEEE Floating Point Formats**

**Table 2-1.1 IEEE Floating Point Number Example in FP B Format**

Value (decimal)	IEEE FP B MSB    LSB	Register N		Register N + 1	
		high	low	high	low
100.0	42C80000h	42h	C8h	00h	00h
55.32	425D47AEh	42h	5Dh	47h	AEh
2.0	40000000h	40h	00h	00h	00h
1.0	3F800000h	3Fh	80h	00h	00h
-1.0	BF800000h	BFh	80h	00h	00h

---

## 3. Modbus/TCP & Modbus RTU Function Codes

The IPC5000 Modbus protocol uses a subset of the standard Modbus function codes to provide access to process-related information. These standard function codes provide basic support for IEEE 32-bit floating point numbers and 16-bit integer register representation of instrument's process data.

Table 3-1 and Table 3-2 list the Function Code definitions and the maximum number of Object Addresses allowed per request.

Repolling of data is not supported by these instruments.

**Table 3-1 Modbus RTU Function Codes Definitions**

<b>Function Code</b>	<b>Name</b>	<b>Usage</b>
01	Read Coil Status	Read the State of a Digital Output
02	Read Input Status	Read the State of a Digital Input
03 04	Read Holding Register / Read Input Registers	Read Data in 16 bit Register Format(high/low). Used to read integer or floating point process data. Registers are consecutive and are imaged from the instrument to the host
05	Force Single Coil	Write data to force Digital Output ON/OFF Values of FF 00 forces DO ON Values of 00 00 forces DO OFF Values of FF FF releases the force of the DO All other values are illegal and will not effect the DO IPC5000 doesn't not support this function code.
06	Preset Single Register	Write Data in 16-bit integer Format(high/low) ONLY.
08	Loopback Test	Used for diagnostic testing of the communication port
16(10h)	Preset Multiple Registers	Write Data in 16-bit Format(high/low). Used to write integer and floating point override data. Registers are consecutive and are imaged from the host to the instrument.
17(11h)	Report Device ID	Read instrument ID and connection information, ROM version, etc.
20(14h)	Read General Reference	Used to Read or upload the instrument's configuration into the host device.
21(15h)	Write General Reference	Used to Write or download an instrument's configuration into the instrument from a host device.

---

**Table 3-2 Maximum Number of Object Address for Each Instrument Type**

<b>Object Name</b>	<b>Number of Object Addresses</b>
Analog Inputs	02
Analog Output	04
Discrete Input	12
Discrete Output/Coil	12
Loop	2
Segments per Set Point Programmer	2
Setpoint programmer for IPC5000	2
Segment Event Groups	2
Link Groups	2
PID Constant Group	2
Controller Group	2
Program Group	2
Setup Group	2
Operation Group	2
Fix PV Event Groups	2
No Loop Group	1

---

## 3.1 Function Code 01 – Read Digital Output Status

### Description

Function code 01(0X reference) is used to read a Digital Output's (DO) ON/OFF status of the slave device in a binary data format. All binary data transferred using function code 01 is mapped into bytes.

The specific number of Dos available in an instrument or available via one Function Code 01 message is instrument-model specific.

Broadcast is not supported.

### Query

The query message specifies the starting DO and the quantity of coils to read. Dos are addressed starting at zero: **DO 1 through 16 are addressed as 0 through 15 respectively.**

**Query message format for function code 01**

Slave address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number DO High	Number DO Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	-----------------------	----------------------	----------------	---------------	-----------	-----------

Example : Read DOs number 1 to 7 from slave at address 02.

RTU example : 02 01 00 00 00 07 CRC CRC

TCP example : 00 01 00 00 00 07

### Response

The DO status in the response message is packed as one DO per bit of the data field. Status is indicated as : 1 = ON: 0 = OFF. The LSB of the first data byte contains the DO addressed in the query. The other Dos follow toward the high order end of this byte, and from low order to high order in subsequent bytes.

If the returned DO quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zero(toward the high order end of the byte). The byte count field specifies the quantity of data bytes returned.

**Response message format for function code 01**

Slave address (00 for TCP)	Function Code	Byte Count	Data	Data	...	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	------------	------	------	-----	-----------	-----------

Example : DOs number 2 and 7 are on, all others are off.

RTU example : 02 01 01 42 CRC CRC

TCP example : 00 01 01 42

In the response the status of DOs 1-7 is shown as the byte value 42 hex, or 0100 0010 binary. DO 8 is the MSB of this byte, and DO 1 is the LSB. Left to right, the status of DO 7 through 1 is: ON-OFF-OFF-OFF-OFF-ON-OFF. DO #8 was not requested and so bit #7 or the MSB was padded with a 0.

---

**Table 3-3 IPC5000 Digital Output Mapping**

Coil Number	Instrument Function
1	Open Collector #1
2	Open Collector #2
3	Open Collector #3
4	Open Collector #4
5	Open Collector #5
6	Open Collector #6
7	Open Collector #7
8	Open Collector #8
9	Open Collector #9
10	Open Collector #10
11	Open Collector #11
12	Open Collector #12

## 3.2 Function Code 02 – Read Digital Input Status

### Description

Function code 02(1X reference) is used to read a Digital Input's(DI) ON/OFF status of the slave device in a binary data format. All binary data transferred using function code 01 is mapped into bytes.

The specific number of inputs available in an instrument or available via one Function Code 02 message is instrument-model specific.

Broadcast is not supported

### Query

The query message specifies the starting input and the quantity of inputs to read. Inputs are addressed starting at zero:

**Input 1 through 16 are addressed as 0 through 15 respectively.**

**Query message format for function code 02**

Slave address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number DI High	Number DI Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--------------------------	-------------------------	-------------------	------------------	--------------	--------------

Example: Read inputs number 1 to 7 from slave at address 02

RTU example : 02 02 00 00 00 07 CRC CRC

TCP example : 00 02 00 00 00 07

### Response

The input status in the response message is packed as one input per bit of the data field. Status is indicated as : 1 = ON; 0 = OFF. The LSB of the first data byte contains the input addressed in the query. The other inputs follow the high order end of this byte, and from low order to high order in subsequent bytes.

If the returned input quantity is not multiple of eight, the remaining bits in the final data byte will be padded with zero(toward the high order the byte). The byte count field specifies the quantity of data bytes returned.

Response message format for function code 02

**Response message format for function code 02**

Slave address (00 for TCP)	Function Code	Byte Count	Data	Data	...	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	------------	------	------	-----	--------------	--------------

Example: Inputs number 2 and 7 are on, all others are off.

RTU example : 02 02 01 42 CRC CRC

TCP example : 00 02 01 42

In the response the status of inputs 1 – 7 is shown as the byte value 42 hex, or 0100 0010 binary. Input 8 is the MSB of this byte, and input 1 is the LSB. Left to right, the status of input 7 through 1 is: ON-OFF-OFF-OFF-OFF-ON-OFF.

Input #8 was not requested and so bit #7 or the MSB was padded with a 0.

---

## 3.3 Function Code 03/04 – Read Input Registers

### Description

Function code 03(4X reference) or Function code 04(3X reference) is used to read the binary contents of input registers in the slave referenced in Section 6(Register Map for Process and Operation Type Variables). Function codes 3 and 4 are not restricted to inputs. They may transmit alarm status, control parameters, etc.

The specific supported registers available in an instrument or available via one Function Code 03/04 message is instrument-model specific. When a master station requests a register that is not supported by the specific device the slave will respond with zeros for that register.

If a request is made to an address that does not exist in the map in Section 6, the instrument is to honor that request and return zeros. This behavior will greatly enhance the bandwidth on the link vs. making several different requests for non-contiguous data elements.(i.e. Consider a device that contains AI #1 and AI #3 and for some reason AI #2 is an invalid request.) The contiguous method would allow the read of AI #1 through AI #3 and the data location for AI #2 would be zeros.

Broadcast is not supported.

### Query

The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero: registers 1-16 are addressed as 0-15.

**Query message format for function code 03/04**

Slave address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number Address High	Number Address Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	-----------------------	----------------------	---------------------	--------------------	-----------	-----------

Example: Read analog inputs #1 and #2 in addresses 1800-1803 as floating point values from a slave at address 02 RTU example : 02 04 18 00 00 04 CRC CRC TCP example : 00 04 18 00 00 04
--

---

## Response

The register data in the response message are packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

The floating point values require two consecutive registers. A request for a single floating point value must be for two registers. The first 16 bits of the response contain the IEEE MSB of the float value. The second 16 bits of the response contain the IEEE LSB of the float value, If the master station request only one register at an address of floating point value, the slave may respond with an exception with illegal data address code.

The Modbus RTU protocol has a single byte count for function codes 03 and 04, therefore the Modbus RTU protocol can only process up to 64 floating point and 127 integer values in a single request.

### Response message format for function codes 03/04

Slave address (00 for TCP)	Function Code	Byte Count	Data	Data	...	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	------------	------	------	-----	-----------	-----------

Example: Analog inputs #1 and #2 as floating point values where AI #1 = 100.0 and AI #2 = 55.32

RTU example : 02 04 08 42 C8 00 00 47 AE 42 5D CRC CRC

TCP example : 00 04 08 42 C8 00 00 47 AE 42 5D

---

## 3.4 Function Code 05 - Force Single Digital Output

### Description

Force a single digital output (OX reference) to either ON or OFF. These are the same digital outputs (DO) used in Function Code 01.

When broadcast, the same function forces the same DO in all attached slave devices.

**(NOTE) IPC5000 does not support this function code.**

### Query

The query message specifies the DO to be forced. Registers are addressed starting at zero:

DO 1 is address 0.

The requested ON/OFF state is specified by a constant in the query data field. A value of FF 00 hex requests it to be ON.

A value of 00 00 hex requests it to be OFF. A value of FF FF releases the force.

**Query message format for function code 05**

Slave address (00 for TCP)	Function Code	DO Address High	DO Address Low	Force Data High	Force Data Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	-----------------	----------------	-----------------	----------------	-----------	-----------

Example : Force DO 06 in a slave at address 02.

RTU example : 02 05 00 06 FF 00 CRC CRC

TCP example : 00 05 00 06 FF 00

### Response

The normal response is an echo of the query, returned after the DO state has been forced.

**Response message format for function code 05**

Slave address (00 for TCP)	Function Code	DO Address High	DO Address Low	Force Data High	Force Data Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	-----------------	----------------	-----------------	----------------	-----------	-----------

Example : Force DO 6 ON in a slave at address 02.

RTU example : 02 05 00 06 FF 00 CRC CRC

TCP example : 00 05 00 06 FF 00

---

## 3.5 Function Codes 06 - Preset Single Register

### Description

Presets integer value into a single register (4X references).

The IPC5000 does not support Broadcast.

The registers that are specified in Section 5 with an access type “W” and integer and bit packed data types, can be written to via Function Code 06.

### Query

The query message specifies the register references to be preset. Registers are addressed starting at zero:

Register 1 is addressed as 0.

Query message format for function code 06

Slave address (00 for TCP)	Function Code	Address High	Address Low	Preset Data High	Preset Data Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--------------	-------------	------------------	-----------------	-----------	-----------

Example : Set Loop #1 to Auto(address 00Fah) to a slave at address 02.

RTU example : 02 06 00 FA 00 01 CRC CRC

TCP example : 00 06 00 FA 00 01

### Response

The normal response is an echo of the query returned after the register contents have been preset.

Slave address (00 for TCP)	Function Code	Address High	Address Low	Preset Data High	Preset Data Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--------------	-------------	------------------	-----------------	-----------	-----------

Example : Set Loop #1 to Auto(address 00Fah) to a slave at address 02.

RTU example : 02 06 00 FA 00 01 CRC CRC

TCP example : 00 06 00 FA 00 01

---

## 3.6 Function Code 08 - Loopback Message

### Description

Echoes received query message.

### Query

Message can be any length up to half the length of the data buffer minus 8 bytes.

**Query message format for function code 08**

Slave address (00 for TCP)	Function Code	Any data, length limited to approximately half the length of the data buffer	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--	-----------	-----------

Example :

RTU example : 02 08 01 02 03 04 CRC CRC

TCP example : 00 08 01 02 03 04

### Response

**Response message format for function code 08**

Slave address (00 for TCP)	Function Code	Data bytes received	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	---------------------	-----------	-----------

Example:

RTU example : 02 08 01 02 03 04 CRC CRC

TCP example : 00 08 01 02 03 04

---

## 3.7 Function Codes 16(10h) – Preset Multiple Registers

### Description

Presets values into a sequence of holding registers(4X references).

The IPC5000 does not support Broadcast.

The register assignments specified in Section 5 with an access type “W”, can be written to via Function Code 16(10h).

### Query

The query message specifies the register references to be preset. Registers are addressed starting at zero:

Register 1 is addressed as 0.

#### Query message format for function code 16(10h)

Slave Address (00 for TCP)	Function Code	Starting Address High	Start Address Low	Number Address High	Number Address Low	Byte Count	Data	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--------------------------	----------------------	------------------------	-----------------------	------------	------	-----------	-----------

Example : Preset Final SP Ramp Value of Loop 1(address BC08h) for Fix control to 100.0 from a slave at address 02.

RTU example : 02 10 BC 08 00 02 04 42 C8 00 00 CRC CRC

TCP example : 00 10 BC 08 00 02 04 42 C8 00 00

### Response

The normal response returns the slave address, function code, start address and the quantity of registers preset.

The floating-point values require two consecutive addresses. A request to preset a single floating point value must be for two addresses. The byte order of the floating-point number is determined by the setting of the byte swap configuration value. In this example the byte swap order is FP B. Refer to subsection 2.1. The first 16 bits of the response contain the IEEE MSB of the float value. The second 16 bits of the response contain the IEEE LSB of the float value. The Byte order is configurable See Subsection 2.1. if the master station requests only one address at an address of a floating point value the slave will response with an illegal data address exception(See Section 4) code.

#### Response message format for function code 16(10h)

Slave address (00 for TCP)	Function Code	Starting Address High	Start Address Low	Number Address High	Number Address Low	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	--------------------------	----------------------	------------------------	-----------------------	-----------	-----------

Example : Response from preset Final SP Ramp value #1(address BC08h) to 100.0 from a slave at address 02.

RTU example : 02 10 BC 08 00 02 CRC CRC

TCP example : 00 10 BC 08 00 02

## 3.8 Function Code 17(11h) – Report Device ID

### Description

Function code 17(11h) is used to report the Device Information that includes information like: Slave ID, device description, and firmware version.

### Query

The query message specifies the function code only.

#### Query message format for function code 17(11h)

Slave address (00 for TCP)	Function Code	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	-----------	-----------

Example : Read Device ID from a slave at address 02.

RTU example : 02 11 CRC CRC

TCP example : 00 11

### Response

The response is a record format describing the instrument.

#### Response message format for function code 17(11h)

Slave address (00 for TCP)	Function Code	Byte Count	Slave ID	RUN Indicator Status	Device Specific Data	CRC (RTU)	CRC (RTU)
-------------------------------	---------------	------------	----------	----------------------	----------------------	-----------	-----------

**Slave ID** – The Slave ID number for the IPC5000 is 50(Hex)

(one byte)(byte 3)

**Run Indicator Status** : (one byte)(byte 4)

00 = OFF; FF = ON

#### Device Specific Data :

Device Description	Model ID	Device Class ID	Device Mapping
--------------------	----------	-----------------	----------------

**Device Description** : (bytes 5-20)

16 Character ASCII Message with the following format:

'I'	'P'	'C'	'5'	'0'	'0'	'0'	' '	Up to 9 character version number in floating point notation	Zeros are appended for the remaining bytes
-----	-----	-----	-----	-----	-----	-----	-----	---	--

---

**For example :**

A IPC5000 with version 1.0 would have the following device description :

'I'	'P'	'C'	'5'	'0'	'0'	'0'	' '	'1'	'.'	'0'	' '	0	0	0	0	0
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	---	---	---	---

**Model ID :**

00(one byte)(byte 21)

**Device Class ID :**

The Device Classification. (one byte)(byte 22)

Class ID	Class
00	Generic Class(Fixed Address Mapable)
01-FF	Future

**Generic Class(00) Device Mapping :**

Describes the I/O and feature mapping.

Number of Records	Record #1	Record #2	Record ...	Record #n
-------------------	-----------	-----------	------------	-----------

**Number of Records :**

1 Byte unsigned value 00-FFh(byte 23)

**Record Description :**

Byte	Description
00	Type of Data Element (See Data Element Values Table Below)
01	Starting Address of Data Element Record(High)
02	Starting Address of Data Element Record(Low)
03	Number of Data Element(High)
04	Number of Data Element(Low)

---

**Data Element Values Table :**

Byte	Description
00	Analog Inputs
01	Analog Outputs
02	Discrete Inputs
03	Discrete Outputs
04	Control Loops
05	Set Point Programmers
24	Setpoint programmer for IPC5000
25	Segment Event Group
27	Link Group
28	PID Constant Group
29	Controller Group
30	Program Group
31	Setup Group
32	Operation Group
33	Fix PV Event Group
34	No Loop Group

---

## 4. Modbus RTU Exception Codes

### 4.1 Introduction

When a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- *Slave device receives the query without a communication error and can handle the query normally.*  
It returns a normal response.
- *Slave does not receive the query due to a communication error.*  
No response is returned. The master program will eventually process a time-out condition for the query.
- *Slave receives the query but detects a communication error (parity, LRC or CRC).*  
No response is returned. The master program will eventually process a time-out condition for the query.
- *Slave receives the query without a communication error but cannot handle it (i.e., request is to a nonexistent coil or register).*  
The slave will return with an exception response informing the master of the nature of the error (Illegal Data Address.)

The exception response message has two fields that differentiate it from a normal response:

#### **Function Code Field:**

In a normal response, the slave echoes the function code of the original query in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are below 80 hex). In an exception response, the slave sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hex higher than the value would be for a normal response.

With the function code's MSB set, the master's application program can recognize the exception response and can examine the data field for the exception code.

#### **Data Field:**

In a normal response, the slave may return data or statistics in the data field. In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

#### **Query**

Example : Internal slave error reading 2 registers starting at address 1820h from the controller.

```
00 03 18 20 00 02
```

#### **Response**

Example : Return MSB in Function Code byte set with Slave Device Failure(04) in the data field.

```
00 83 04
```

---

**Table 4-1 Modbus RTU Data Layer Status Exception Codes**

Exception Code	Definition	Description
01	Illegal Function	The message received is not an allowable action for the addressed device.
02	Illegal Data Address	The address referenced in the function-dependent data section of the message is not valid in the addressed device.
03	Illegal Data Value	The value referenced at the addressed device location is no within range.
04	Slave Device Failure	The addressed device has not been able to process a valid message due to a bad device state.
06	Slave Device Busy	The addressed device has ejected a message due to a busy state. Retry later.
07	NAK, Negative Acknowledge	The addressed device cannot process the current message. Issue a PROGRAM POLL to obtain device – dependent error data.
09	Buffer Overflow	The data to be returned for the requested number of registers is greater than the available buffer space. <i>Function Code 20 only.</i>

# 5. Register Map for Process and Operation Type Variables

## What's in this section ?

This section describes all parameters accessible by Function Code 03, 04, 06 and 10h. Section 5.1 gives a global overview of each function and its addresses/registers. Sections 5.2 through 5.20 contain the details on each function and each of its parameters.

Your particular controller may not contain all parameters shown. If you see a function that is not on your controller, either it is not available for that controller model or it is not in your configuration.

## 5.1 Register Map Overview

Table 5-1 describes the global register map for Function Code 03, 04, 06 and 10h. Details on each address are in sections 5.2 through 5.20.

### *Conversion of Address (Hex) Number to Register (Decimal) Number*

To convert the address number to the register number, convert the address from hexadecimal to decimal and add 1. Registers are addressed starting at zero: registers 1-16 are addressed as 0-15.

To convert the register number to the address number, subtract 1 from the register and convert to hex.

**Table 5-1 Global Register Map**

Start Address (hex)	End Address (hex)	Description	Notes
0000	< 0040	Miscellaneous Parameters	It is not available for IPC5000
0040	00FF	Loop #1 (floating point & bit packed)	
0140	01FF	Loop #2 (floating point & bit packed)	
1800	187F	Analog Input Value(#1-#2)	
1DF0	1DF6	Time	
1E00	1E0F	Set Point Programmer #1	
1E10	1E1F	Set Point Programmer #2	
1F00	1F3F	Set Point Programmer #1 Additional Values	
1F40	1F7F	Set Point Programmer #2 Additional Values	
8000	844B	Set Point programmer #1 segment (1 – 100)	Only for IPC5000
844C	8897	Set Point programmer #2 segment (1 – 100)	Only for IPC5000
8898	88B0	Set Point Programmer #1	Only for IPC5000
88B1	88C9	Set Point Programmer #2	Only for IPC5000
88CA	A999	Segment Set Point Event #1(1 – 100)	Only for IPC5000
A99A	CA69	Segment Set Point Event #2(1 – 100)	Only for IPC5000

---

Start Address (hex)	End Address (hex)	Description	Notes
CB00	CB3B	Link Group #1 (1 – 10)	
CB3C	CB77	Link Group #2 (1 – 10)	
CB78	CC46	PID Group #1 (1 – 8)	
CC48	CD16	PID Group #2 (1 – 8)	
CD17	CD2B	Controller Group #1	
CD2A	CD3E	Controller Group #2	
CD3F	CD4C	Program Group #1	
CD4D	CD5A	Program Group #2	
CD5B	CD72	Setup Group #1	
CD73	CD8A	Setup Group #2	
CD8B	CD9B	Operation Group #1	
CD9C	CDAC	Operation Group #2	
CDAD	CDF4	Fix PV Event Group	
CDF5	CDFE	No Loop Group	

## 5.2 Loop Value Register Map

This table contains addresses of Loop #1; see Table 5-1 on Page 35 for addresses of Loop #2

**Table 5-2 Loop Value Register Map Address**

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
0040	0065	PV	R	Floating Point in Engineering Units
0042	0067	Remote Set Point; SP2	R	Floating Point in Engineering Units It is not available for IPC5000
0044	0069	Working Set Point	R	Floating Point in Engineering Units
0046	0071	Output	R/W	Floating Point in Engineering Units In case of Manual Mode, it is only writable.
0048	0073	PV	R	Floating Point in Engineering Units
004A	0075	Input #1	R	Floating Point in Engineering Units It is not available for IPC5000
004C	0077	Gain #1(Prop Band #1 if active)	R	Floating Point in Engineering Units (Gain or Proportional Band) <b>Note 1</b>
004E	0079	Direction	R	Floating Point <b>Note 2</b>
0050	0081	Reset #1	R	Floating Point in Engineering Units Minutes/Repeat <b>Note 1</b>
0052	0083	Rate #1	R	Floating Point in Minutes <b>Note 1</b>
0054	0085	Cycle Time #1	R	Floating Point in Seconds
0056	0087	PV Low Range	R	Floating Point in Engineering Units
0058	0089	PV High Range	R	Floating Point in Engineering Units
005A	0091	Alarm #1 SP #1	R	Floating Point in Engineering Units It is not available for IPC5000
005C	0093	Alarm #1 SP #2	R	Floating Point in Engineering Units It is not available for IPC5000
0060	0097	Gain #2(Pro Band #2 if active)	R	Floating Point in Engineering Units (Gain or Proportional Band) <b>Note 1</b>
0062	0099	Deadband	R	Floating Point <b>Note 1</b>
0064	0101	Reset #2	R	Floating Point in Engineering Units Minutes/Repeat <b>Note 1</b>
0066	0103	Rate #2	R	Floating Point in Minutes <b>Note 1</b>
0068	0105	Cycle Time #2	R	Floating Point in Seconds

**Note 1** : In order to write data in these registers, after knowing a current PID zone or group by using CD95(hex) for Loop 1 or CDA6(hex) for Loop 2, the data has to be written in the PID constant area.

The Gain #1, Reset #1, Rate #1 is PID constants for heating of duplex control or normal PID control and the Gain #2, Reset #2, Rate #2 is PID constants for cooling of duplex control.

**Note 2** : **0.0 = Reverse; 1.0 = Direct**

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
006A	0107	LSP #1	R/W	Floating Point in Engineering Units
006C	0109	LSP #2	R/W	Floating Point in Engineering Units It is not available for IPC5000
006E	0111	Alarm #2 SP #1	R	Floating Point in Engineering Units It is not available for IPC5000
0070	0113	Alarm #2 SP #2	R	Floating Point in Engineering Units It is not available for IPC5000
0074	0117	SP Low Limit	R/W	Floating Point in Engineering Units Operator Limit
0076	0119	SP High Limit	R/W	Floating Point in Engineering Units Operator Limit
0078	0121	Working Set Point	R	Floating Point in Engineering Units
007A	0123	Output Low Limit	R/W	Floating Point in Engineering Units <b>Note 1</b>
007C	0125	Output High Limit	R/W	Floating Point in Engineering Units <b>Note 1</b>
007E	0127	Output Working Value	R/W	Floating Point in Engineering Units In case of Manual Mode, it is only writable.
0088	0137	Bias	R/W	Floating Point in Engineering Units
008A	0139	Deviation	R	Floating Point in Engineering Units (SP – PV)
008E	0143	Manual Reset	R	Floating Point in Engineering Units <b>Note 1</b>
00F7	0248	Enable/Disable Fuzzy	R/W	Bit Packed Bit 0 : 0 : Disable; 1: Enable Bit 1-15 : Unused
00F8	0249	Demand Tune Request	R/W	Bit Packed(one shot action, activates auto tuning until auto tuning completed) Bit 0 : 0 : Disable; 1: Enable Bit 1-15 : Unused It is not available at Manual or STOP mode.
00FA	0251	Auto/Manual State	R/W	Bit Packed Bit 0 : 0 : Manual; 1: Auto Bit 1-15 : Unused It is not available at TUNE mode.
00FD	0254	Tune Set State	R/W	Bit Packed(Select Heat/Cool tuning constant set) Bit 0 : 0 : Heat Tune Set 1 : Cool Tune Set Bit 1-15 : Unused It is available for Duplex control

**Note 1** : In order to write data in these registers, after knowing a current PID zone or group by using CB78(hex) for Loop 1 or CC48(hex) for Loop 2, the data has to be written in the PID constant area.

---

## 5.3 Example for queries using Function Codes 3, 6,16

### Example 1 :

**Query :** Read PV, Remote SP, Working SP, Output for Loop 1 from IPC5000 using Function Code 3(hex codes).

This will be accomplished by accessing contiguous registers.

00	03	00	40	00	08	CRC	CRC
----	----	----	----	----	----	-----	-----

**Response :** Where PV = 1000.0, Remote SP = 0.0(Fixed ), Working SP = 1000.0, Output = 50.0

00	03	10	44	7A	00	00	00	00	00	00	44	7A	00	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

42	48	00	00	CRC	CRC
----	----	----	----	-----	-----

### Example 2 :

**Query :** Write a Local Setpoint,(address 006A) to 100.0 for Loop 1 at IPC5000 using Function Code 16(10 hex).

Function code 16 is used for presetting multiple registers. Registers are addressed starting at zero : Register 1 is addressed as 0

00	10	00	6A	00	02	04	42	C8	00	00	CRC	CRC
----	----	----	----	----	----	----	----	----	----	----	-----	-----

**Response :** from preset of LSP #1, address 006A to 100.0

00	10	00	6A	00	02	CRC	CRC
----	----	----	----	----	----	-----	-----

### Example 3 :

**Query :** Set Loop #1 to Auto(register 00FA hex) at controller. This is a bit-packed single register data type. Registers are addressed starting at 0 : Register 1 is addressed as 0.

00	06	00	FA	00	01	CRC	CRC
----	----	----	----	----	----	-----	-----

**Response :** The control mode of the controller at address 0 is changed to AUTO

00	06	00	FA	00	01	CRC	CRC
----	----	----	----	----	----	-----	-----

---

## 5.4 Analog Input Value Register Map – Function Code 03,04

**Table 5-3 Analog Input Value Register Map Address- Function Code 03,04**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1800	6145	Analog Input #1	R	Floating Point in Engineering Units
1802	6147	Analog Input #2	R	

### Example

Read Analog Inputs 1 and 2 from IPC5000 using Function Code 03

00	03	18	00	00	04	CRC	CRC
----	----	----	----	----	----	-----	-----

Response from IPC5000 where AI 1 = 100.0 and AI 2 = 55.32

00	03	08	42	C8	00	00	42	5D	47	AE	CRC	CRC
----	----	----	----	----	----	----	----	----	----	----	-----	-----

## 5.5 Time Register Map

**Table 5-5 Time Register Map Addresses**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1BE0	7137	Hours	R/W	0 to 23
1BE1	7138	Minutes	R/W	0 to 59
1BE2	7139	Seconds	R/W	0 to 59
1BE3	7140	Month	R/W	1 to 12
1BE4	7141	Day	R/W	1 to 31
1BE5	7142	Year	R/W	Read : 1999 to 2099 Write : 00 to 99 represents 2000 to 2099.
1BE6	7143	Week Day	R/W	0 : Sunday 1 : Monday 2 : Tuesday 3 : Wednesday 4 : Thursday 5 : Friday 6 : Saturday

**(Note) In order to set Time and Date, use Function Code 6. Function Code 16 is not available for setting them.**

---

## 5.6 Set Point Program Register Maps

### Summary

The Set Point programmers are listed according to category related to program status and interaction, other programmer parameters and program segment mapping ,editing and style. A section is also provided to aid in configuring a SP programmer and recipe interface for third party software.

### Function Code Support :

Read – Function Code 3,4

Write – Function Code 6,16(10 Hex)

### Considerations and Methods for Downloading, Operating , and Reading Status of SP Programs :

A SP programmers interface can be developed(and recipe containing a SP profile can be created) in third party software using the parameters listed in the following table.

In creating a SP Programmer interface showing a number of segments, a graphic display might include a table referencing the maximum number of ramp/soak segments that you will be using for your process. The parameters to be referenced for each segment are listed in Table 5-10 – Register Map

### Programmer Numbers

The parameters that follow refer to SP Programmer 1. Table 5-6 lists the Starting and Ending addresses for all of the SP programmers.

**Table 5-6 SP programmer Addresses**

Programmer Number	Starting Address (Hex)	Starting Address (Decimal)	Ending Address (Hex)	Ending Address (Decimal)	Notes
SP Programmer 1	1E00	7681	1E0F	7696	
SP Programmer 2	1E10	7697	1E1F	7712	
SP Programmer 1	8898	34969	88B0	34992	It is used for only IPC5000
SP Programmer 2	88B1	34993	88C9	35017	It is used for only IPC5000

---

## Controlling the Program

For controlling the program, the following parameters should be accessed :

Programmer	Address (Hex)	Address (Decimal)	Notes
Current Segment Number	1E02	7683	See Table 5-7
Current Time Remaining	1E08	7689	See Table 5-7
Status	1E0B	7692	See Table 5-7
Start(write only)	1E0C	7693	See Table 5-7
Hold(write only)	1E0D	7694	See Table 5-7
Advance(write only)	1E0E	7695	See Table 5-7
Reset(write only)	1E0F	7696	See Table 5-7
Segment Edit	88AF	34991	See Table 5-8

## Parameters for the Profile

You will also need to define the following for the parameters for profile :

Programmer	Address (Hex)	Address (Decimal)	Notes
Start Setpoint Loop 1	8898	34968	See Table 5-8
Start Setpoint Loop 2	889A	34970	See Table 5-8
Start Mode	889C	34972	See Table 5-8
Priority	889D	34793	See Table 5-8
Saving Program number	88B0	34992	See Table 5-8

## Procedure for Downloading Setpoint Programs

These steps are for programmer 1. For programmer 2, adjust the register address accordingly by adding offset for starting addresses.

Table 5-6.1 is for downloading using Function Codes 3, 4, 6, 16.

**Table 5-6.1 Steps to Download a Setpoint Program using Modbus Function Codes 3, 4, 6, 16.**

Step	Action
1	Set the programmer to RESET by writing any number to 7696(1E0F). This can be done either with function code 6 or 16.
2	Set saving program number by writing any number to 34992(88B0). The program number for downloading can be set from 0 to 31. This can be done either with function code 6 or 16.
3	Write the header information for parameters relevant to the program – registers 34968(8898) – 34990(88AE) except 34974(889E) because the register is read only. Registers 8898 and 889A are floats and must be written using function code 16. Registers 889C – 88B0 is signed 16 bit integer and can be written with either function code 6 or 16.
4	Write the information for each segment required in the profile – registers 8000 – 800A for segment 0 except read only register. 800B – 8015 for segment 1 etc. The first 3 register are floats and must be written using function code 16. The last registers are signed 16 bit integer and can be written with either function code 6 or 16.
5	Write the information for each segment event group required in the profile – registers 88CA – 891D for segment 0 except read only register. 891E – 8971 for segment 1 etc. The first 2 registers are floats and must be written using function code 16. The last registers are signed 16 bit integer and can be written with either function code 6 or 16. One segment event group for one segment can have up to 12 events.

## Procedure for Uploading Setpoint Programs

**Table 5-6.2 Steps to Upload a Setpoint Program using Modbus Function 3, 4, 6, 16**

Step	Action
1	Set loading program number by writing any number to 34992(88B0). The program number for uploading can be set from 0 to 31. This can be done either with function code 6 or 16.
2	Read the header information desired-register 34968(8898) – 34990(88AE) using function code 3 or 4. Registered segment count can be known by reading register 34974(889E).
3	Read the information for each segment desired using function code 3 or 4 – registers 8000-800A for segment 0. 800B – 8015 for segment 1 etc. The information for each segment can be read by the registered segment count.
4	Read the information for each segment events desired using function code 3 or 4 – registers 88CA-891D for segment 0. 891E-8971 for segment 1 etc. The information for each segment events can be read by registered segment event count in one segment.

## 5.7 Set Point Programmer Value Register Map

**Table 5-7 SetPoint Programmer #1 Value Register Map Address**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1E00	7681	Set Point Output	R	Floating Point in Engineering Units. It is not available for IPC5000.
1E02	7683	Current Segment Number	R	Floating Point; # 0 to # 99 Current running segment
1E08	7689	Segment Time Remaining	R	Floating Point in Seconds or Time units
1E0B	7692	Status	R	Bit Packed Bit 0 : 1 = Ready Bit 1 : 1 = Run Bit 2 : 1 = Hold Bit 3 : 1 = End If Bit 4 = 1 and Bit 5 = 1, then Time Unit is Minutes and Seconds If Bit 5 = 1 and Bit 6 = 1, then Time Unit is Hours and Minutes Bit 7 : Ramp Units 0 : Time 1 : Rate Bit 8 and 9 : Unused Bit 10 : 0 : Current segment is a soak 1 : Current segment is a ramp Bit 11 to 15 : Unused
1E0C	7693	Start(Run)	W	Unsigned 16 bit integer 1 : Run Enable 0 : Data ignored <i>Note 1, Note 2</i>
1E0D	7694	Hold	W	Unsigned 16 bit integer 1 : Hold Enable 0 : Hold Disable <i>Note 1, Note 2</i>
1E0E	7695	Advance	W	Unsigned 16 bit integer 1 : Advance Enable 0 : Data ignored <i>Note 1,3</i>
1E0F	7696	Reset(Stop)	W	Unsigned 16 bit integer 1 : Stop Enable 0 : Data ignored <i>Note 1, Note 2</i>

*Note 1* : These registers are available for Programmer mode as well as Controller mode(Fix control = Constant control)

*Note 2* :

Current Status	Command		
	Run	Hold	Stop
Run	Set error	Hold	Stop
Hold	Run	Set error	Stop
Stop	Run	Set error	Set error

*Note 3* : Current segment move to next segment per one command.

## 5.8 Set Point Programmer Register Map for IPC5000

These parameters can be available or not according to Synchronous or Asynchronous mode of controller.

**Table 5-8 Set Point Programmer Register Map Addresses**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
8898	34968	Start Setpoint Loop 1(SSP1)	R/W	Floating Point Set Range : SP Low/High Limit SSP2 is not available at Async. mode.
889A	34970	Start Setpoint Loop 2(SSP2)	R/W	
889C	34972	Start Mode	R/W	Signed 16 bit integer 0: SSP; 1: SPV; 2:TIME
889D	34973	Priority	R/W	Signed 16 bit integer 0: PV1; 1: PV2 It is not available at Async. mode.
889E	34974	Segment Count	R	Signed 16 bit integer Segment number registered to a program One program can have 100 segments.
889F	34795	All Repeat Count	R/W	Signed 16 bit integer Set Range : 1 to 999 It is the number to repeat one program.
88A0	34976	Repeat End Segment 1	R/W	Signed 16 bit integer Set Range : Within Segment Count End Segment for repeat have to be larger than Start Segment.
88A1	34977	Repeat Start Segment 1	R/W	
88A2	34978	Number of Repeats 1	R/W	Signed 16 bit integer Set Range : 1 to 999
88A3	34979	Repeat End Segment 2	R/W	Signed 16 bit integer Set Range : Within Segment Count End Segment for repeat have to be larger than Start Segment.
88A4	34980	Repeat Start Segment 2	R/W	
88A5	34981	Number of Repeats 2	R/W	Signed 16 bit integer Set Range : 1 to 999
88A6	34982	Repeat End Segment 3	R/W	Signed 16 bit integer Set Range : Within Segment Count End Segment for repeat have to be larger than Start Segment.
88A7	34983	Repeat Start Segment 3	R/W	
88A8	34984	Number of Repeats 3	R/W	Signed 16 bit integer Set Range : 1 to 999
88A9	34985	Repeat End Segment 4	R/W	Signed 16 bit integer Set Range : Within Segment Count End Segment for repeat have to be larger than Start Segment.
88AA	34986	Repeat Start Segment 4	R/W	
88AB	34987	Number of Repeats 4	R/W	Signed 16 bit integer Set Range : 1 to 999
88AC	34988	Repeat End Segment 5	R/W	Signed 16 bit integer Set Range : Within Segment Count End Segment for repeat have to be larger than Start Segment.
88AD	34989	Repeat Start Segment 5	R/W	
88AE	34990	Number of Repeats 5	R/W	Signed 16 bit integer Set Range : 1 to 999

**Continued Next page**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
88AF	33991	Segment edit	W	Signed 16 bit integer It is a edition command for deleting or inserting a segment into a program. Bit 0 – 7 : Segment number to be inserted or deleted. Bit 8 : 0 : Insert; 1 : Delete The segment number of Bit 0 – 7 has to be within segment count. By Insert command, the values to be initialized to zero are saved.
88B0	34992	Program Save Request	R/W	Signed 16 bit integer Program number to be saved or read. Async mode : 0 ~ 15 for Loop 1, 16 ~ 31 for Loop 2 Sync mode : 0 ~ 31, no related to loop.

### Example

This is a example to insert initial data to segment 1 of program 3.

Segment count registered to program 3 have to be larger than 1.

Set program number to be edited as 3.

00	06	88	B0	00	03	CRC	CRC
----	----	----	----	----	----	-----	-----

Response :

00	06	88	B0	00	03	CRC	CRC
----	----	----	----	----	----	-----	-----

Insert initial data to segment 2

00	06	88	AF	00	01	CRC	CRC
----	----	----	----	----	----	-----	-----

Response : The data of Program 3 is shown like the right picture of Figure 5-8

00	06	88	AF	00	01	CRC	CRC
----	----	----	----	----	----	-----	-----

PROGRAM 3 SEGMENTS 2

SEG	SP	PID	TIME	JC GS	EVENT SET
00	50.0	6	00:20	1	M M M P T P
				0	A T T 0 0 0
01	50.0	3	01:00	1	M M M P P P
				2	A P T 0 0 0

PROGRAM 3 SEGMENTS 3

SEG	SP	PID	TIME	JC GS	EVENT SET
00	50.0	6	00:20	1	M M M P T P
				0	A T T 0 0 0
01	50.0	6	00:00	1	M M M 0 0 0
				0	A 0 0 0 0 0
02	50.0	3	01:00	1	M M M P P P
				2	A P T 0 0 0

<Figure 5-8> Insert a initial segment to a program

## 5.9 Set Point Programmer Segment Register Map

A profile contains up to 100 segments. Each segment is made up of 10 registers. This table contains Segment Map Address of SP programmer #1 and #2. See the Global Register Map – Table 5-1 for starting and ending addresses(hex). Each segment has is made up of 12 events like Table 5-9-1.

**Table 5-9 Set Point Programmer Segment Map Addresses**

Start Address (Hex)	Start Address (Decimal)	End Address (Hex)	End Address (Decimal)	Description
8000	32769	800A	32778	Set Point Programmer #1 Segment 1
800B	32779	8015	32789	Set Point Programmer #1 Segment 2
8016	32790	8020	32800	Set Point Programmer #1 Segment 3
8021	32801	802B	32811	Set Point Programmer #1 Segment 4
802C	32812	8036	32822	Set Point Programmer #1 Segment 5
8037	32823	8041	32833	Set Point Programmer #1 Segment 6
:	:	:	:	:
8441	33857	844B	33867	Set Point Programmer #1 Segment 100
844C	33868	8456	33878	Set Point Programmer #2Segment 1
8457	33879	8461	33889	Set Point Programmer #2Segment 2
8462	33890	846C	33900	Set Point Programmer #2Segment 3
846D	33901	8477	33911	Set Point Programmer #2Segment 4
8478	33912	8482	33922	Set Point Programmer #2Segment 5
8483	33923	848D	33933	Set Point Programmer #2Segment 6
:	:	:	:	:
888D	34757	8897	34967	Set Point Programmer #2Segment 100

**Table 5-9-1 Set Point Programmer Segment Event Map Address**

Start Address (Hex)	Start Address (Decimal)	End Address (Hex)	End Address (Decimal)	Description
88CA	35018	891D	35101	Events Group of Set Point Programmer #1 Segment 1
891E	35102	8971	35185	Events Group of Set Point Programmer #1 Segment 2
8972	35186	89C5	35269	Events Group of Set Point Programmer #1 Segment 3
89C6	35270	8A19	35353	Events Group of Set Point Programmer #1 Segment 4
8A1A	35354	8A6D	35437	Events Group of Set Point Programmer #1 Segment 5
8A6E	35438	8AC1	35521	Events Group of Set Point Programmer #1 Segment 6
:	:	:	:	:
A946	43334	A999	43417	Events Group of Set Point Programmer #1 Segment 100
A99A	43418	A9ED	43501	Events Group of Set Point Programmer #2 Segment 1
A9EE	43502	AA41	43585	Events Group of Set Point Programmer #2 Segment 2
AA42	43586	AA95	43669	Events Group of Set Point Programmer #2 Segment 3
AA96	43670	AAE9	43753	Events Group of Set Point Programmer #2 Segment 4
AAEA	43754	AB3D	43837	Events Group of Set Point Programmer #2 Segment 5
AB3E	43838	AB91	43921	Events Group of Set Point Programmer #2 Segment 6
:	:	:	:	:
CA16	51734	CA69	51817	Events Group of Set Point Programmer #2 Segment 100

**Table 5-9-2 Each Event Group Map Address of Set Point Programmer #1 Segment 1**

Start Address (Hex)	Start Address (Decimal)	End Address (Hex)	End Address (Decimal)	Description
88CA	35018	88D0	35024	Event #1 of Set Point Programmer #1 Segment 1
88D1	35025	88D7	35031	Event #2 of Set Point Programmer #1 Segment 1
88D8	35032	88DE	35038	Event #3 of Set Point Programmer #1 Segment 1
88DF	35039	88E5	35045	Event #4 of Set Point Programmer #1 Segment 1
88E6	35046	88EC	35052	Event #5 of Set Point Programmer #1 Segment 1
:	:	:	:	:
8917	35095	891D	35101	Event #12 of Set Point Programmer #1 Segment 1

## 5.10 Segment Register Map

The table below describes the registers that are part of a setpoint programmer segment. To determine the actual register address for a parameter within a segment, add the register offset to the start address of the segment.

Writing to these registers is permissible in the stop mode or running program number is not same with setting program number.

**Table 5-10 Segment Register Map Addresses**

Register Offset within Segment(hex)	Parameter Name	Access	Notes
0(0)	Setpoint Value 1	R/W	Floating Point Set Range : SP Low/High range
2(2)	Setpoint Value 2	R/W	Floating Point At Sync mode, it is Target Setpoint of Loop 2 and not available for Async Set Range : SP Low/High range
4(4)	Segment time	R/W	Floating Point Set Range : 00.00 to 99.59
6(6)	PID Group number 1	R/W	Signed 16 bit Integer Set Range : 1 to 8; It is only available for Segment PID.
7(7)	PID Group number 2	R/W	Signed 16 bit Integer 오류! 연결이 잘못되었습니다.
8(8)	Program End	R/W	Signed 16 bit Integer 0 : Continuous Shifting ; 1 : HOLD Shifting; 2 : Fix Control Shifting
9(9)	Segment G.Soak	R/W	Signed 16 bit Integer 0 : NONE; 1 : FRONT; 2 : REAR; 3 : ALL
10(A)	Time or Rate	R	Signed 16 bit Integer 0 : Time; 1: Rate

## 5.11 Example for determining a Segment Register

To change the segment time in segment #8 of setpoint programmer #2 for Loop 2, the register address is determined as follows.

Step 1 : Use Table 5-1 to determine the start address for setpoint programmer #2 profile. The value is 844C hex

Step 2 : Calculate the offset address for segment 8 in a profile. This is calculated as :

$$\begin{aligned}
 \text{Segment \#8 offset address} &= (\text{segment number} - 1) * 11 \\
 &= (8-1) * 11 \\
 &= 77 \text{ or } 4D \text{ Hex}
 \end{aligned}$$

Step 3 : Use the table above to determine the register offset for the segment time. The value is 4.

Step 4 : Calculate the address by adding the results of steps 1, 2 and 3 to determine the register address.

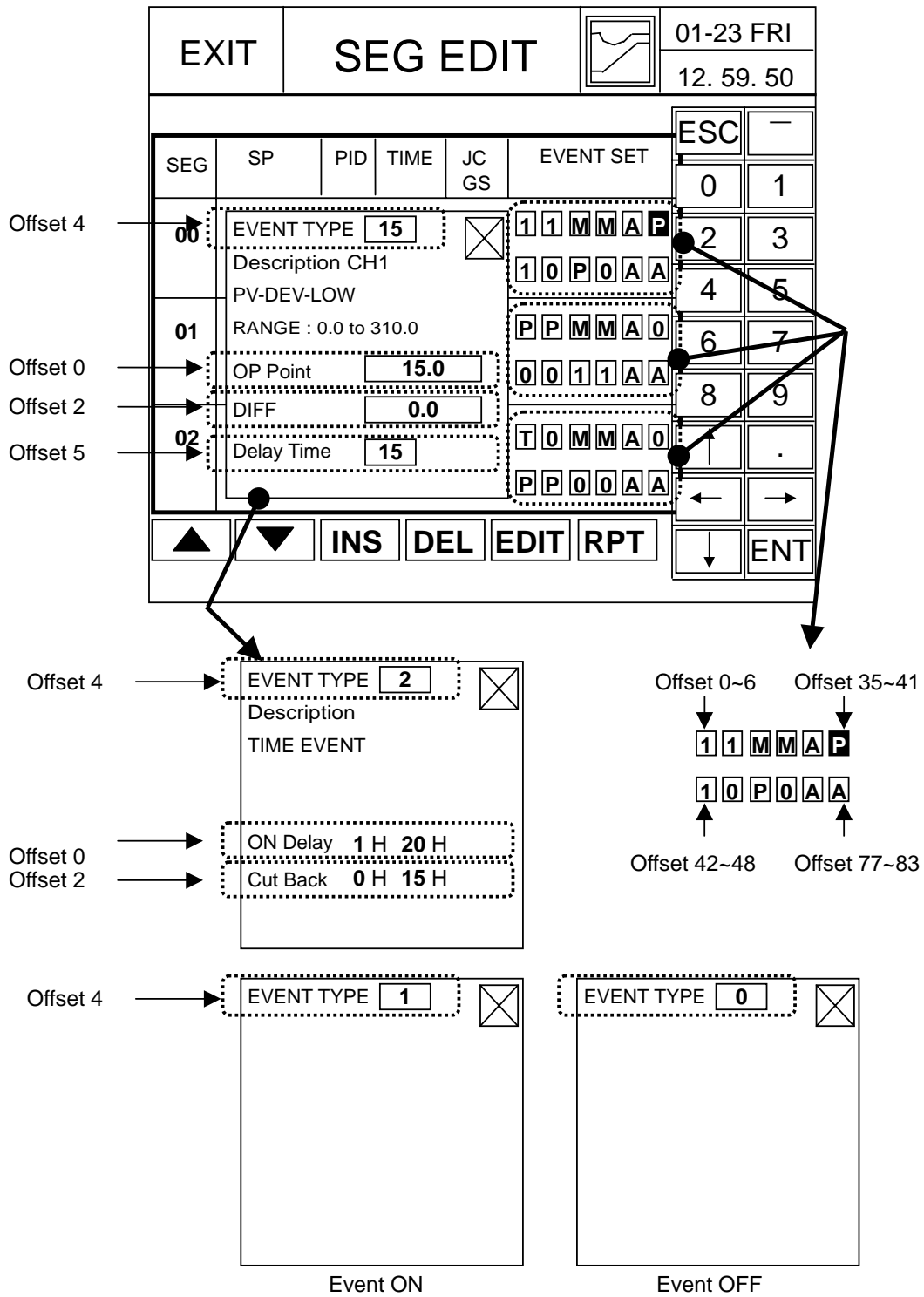
$$\begin{aligned}
 \text{Register address} &= \text{Setpoint program \#2 profile base address} + \text{Segment 8 offset address} + \text{Segment time} \\
 &\quad \text{register offset} \\
 &= 844C + 4D + 4 \\
 &= 849D \text{ Hex}
 \end{aligned}$$

## 5.12 Segment Set Point Event Register Map

It is the event area of segment data like Figure 5.12. Event count is same with segment number registered to a program and event number registered to a segment is up to 12. Writing to these registers is permissible in the stop mode or running program number is not same with setting program number. According to invalid event type, writing is enable or disable. The Event type number of Table 5-12 is not available when the event number is assigned for Mode or Alarm event.

**Table 5-12 Segment Set Point Event Register Map Address**

Register Offset within Segment(hex)	Parameter Name	Access	Notes
0	Max Value or OP point	R/W	Floating Point OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
2	Min Value or Differential	R/W	Floating Point Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
4	Event type number	R/W	Signed 16 bit Integer 0: OFF 1: ON(Time Event) 2: ON Delay and Cut Back setup(Time Event) 11: PV-ABS-LOW for Channel 1(PV Event) 12: PV-ABS-LOW for Channel 2(PV Event) 13: PV-ABS-HIGH for Channel 1 (PV Event) 14: PV-ABS-HIGH for Channel 2 (PV Event) 15: PV-DEV-LOW for Channel 1 (PV Event) 16: PV- DEV-LOW for Channel 2 (PV Event) 17: PV-DEV-HIGH for Channel 1 (PV Event) 18: PV-DEV-HIGH for Channel 2 (PV Event) 21: SP-ABS-LOW for Channel 1 (PV Event) 22: SP-ABS-LOW for Channel 2 (PV Event) 23: SP-ABS-HIGH for Channel 1 (PV Event) 24: SP-ABS-HIGH for Channel 2 (PV Event) 25: SP-DEV-OFF for Channel 1 (PV Event) 26: SP-DEV-OFF for Channel 2 (PV Event) 27: SP-DEV-ON for Channel 1 (PV Event) 28: SP-DEV-ON for Channel 2 (PV Event) 31: DV-DEV-OFF for Channel 1 (PV Event) 32: DV-DEV-OFF for Channel 2 (PV Event) 33: DV-DEV-ON for Channel 1 (PV Event) 34: DV-DEV-ON for Channel 2 (PV Event) 35: MV-ABS-LOW for Channel 1 (PV Event) 36: MV-ABS-LOW for Channel 2 (PV Event) 37: MV-ABS-HIGH for Channel 1 (PV Event) 38: MV-ABS-HIGH for Channel 2 (PV Event)
5	Time Delay	R/W	Signed 16 bit Integer Set Range: 0 ~ 99
6	Select	R	Signed 16 bit Integer



< Figure 5-12> Segment Event set screen

---

## 5.13 Example for determining a Segment Event Register

To change the event type number of the digital output #3 in segment #8 of setpoint programmer #2 for Loop 1, the register address is determined as follows.

Step 1 : Use Table 5-1 to determine the start address for setpoint programmer #2 profile. The value is 88CA hex

Step 2 : Calculate the offset address for segment 8 in a profile. This is calculated as :

$$\begin{aligned}\text{Segment \#8 offset address} &= (\text{segment number} - 1) * 84 \\ &= (8-1) * 84 \\ &= 558 \text{ or } 24C \text{ Hex}\end{aligned}$$

Step 3 : Use the table above to determine the register offset for the event type number. The value is 18(12Hex).

Step 4 : Calculate the address by adding the results of steps 1, 2 and 3 to determine the register address.

$$\begin{aligned}\text{Register address} &= \text{Setpoint program \#2 profile base address} + \text{Segment 8 offset address} + \text{Event type} \\ &\quad \text{number offset} \\ &= 88CA + 24C + 12 \\ &= 8B28 \text{ Hex}\end{aligned}$$

## 5.14 Program Link Register Map

A profile contains up to 10 links. Each link is made up of 6 registers. This table contains Link Map Address of Link group #1 and #2.

**Table 5-14.1 Each Link group register Map Addresses**

Start Address (Hex)	Start Address (Decimal)	End Address (Hex)	End Address (Decimal)	Description
CB00	51968	CB05	51973	Link group #1 of Loop1
CB06	51974	CB0B	51979	Link group #2 of Loop1
CB0C	51980	CB11	51985	Link group #3 of Loop1
CB12	51986	CB17	51991	Link group #4 of Loop1
:	:	:	:	:
CB36	52022	CB3B	52027	Link group #10 of Loop1
CB3C	52028	CB41	52033	Link group #1 of Loop2
CB42	52034	CB47	52039	Link group #2 of Loop2
CB48	52040	CB4D	52045	Link group #3 of Loop2
CB4E	52046	CB53	52051	Link group #4 of Loop2
:	:	:	:	:
CB72	52082	CB77	52087	Link group #10 of Loop2

The table below describes the registers that are part of a Link group Register. To determine the actual register address for a parameter within a Link, add the register offset to the start address of the Link.

See Figure 5-14 for mapping each register position on the Link Set screen.

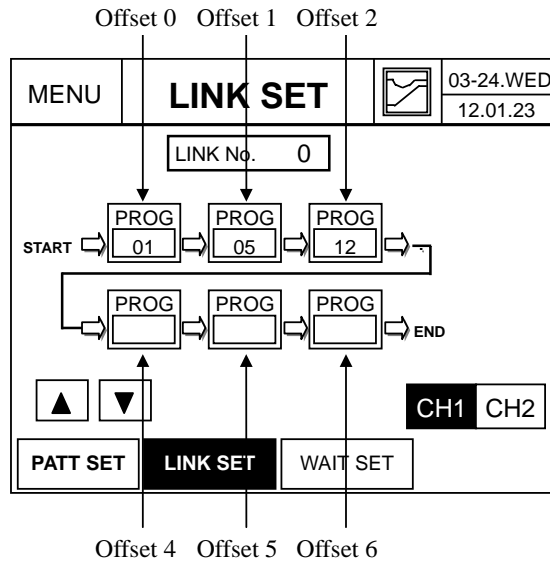
**Table 5-14.2 Link register Map Addresses**

Register Offset within Link	Parameter Name	Access	Notes
0	Link Program number 1	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
1	Link Program number 2	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
2	Link Program number 3	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
3	Link Program number 4	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
4	Link Program number 5	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
5	Link Program number 6	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>

**Note 1 :** Set Range : 0 to 15 for Loop 1 and 16 to 31 for Loop 2 at Async mode

0 to 31 for Loop1/2 at Sync mode

**Note 2 :** Write 255(FF hex) to the register address to delete the registered program number.



< Figure 5-14 > LINK SET

**How to find the Register address of program number registered to Link group of Loop 1**

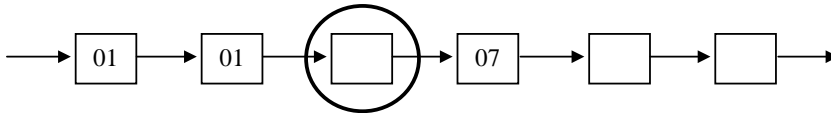
Register address = Start address of Link group #1 of Loop 1+ (Link group number - 1) \* 6 + Offset value of the parameter from Table 5-14.2.

**Example :** Find the register address of the third Link program number of Link group #4

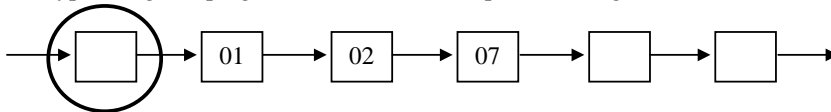
The Register address = CB00 hex + (4-1) \* 6 + 02 = CB14 hex

< **Caution** > Program number has to be registered in sequence.

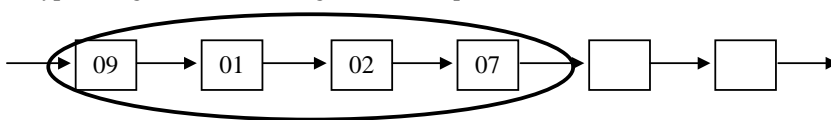
1) Incorrect type to register program number : The third part is not registered.



2) Incorrect type to register program number : The first part is not registered.



3) Correct type : Program number is registered in sequence.

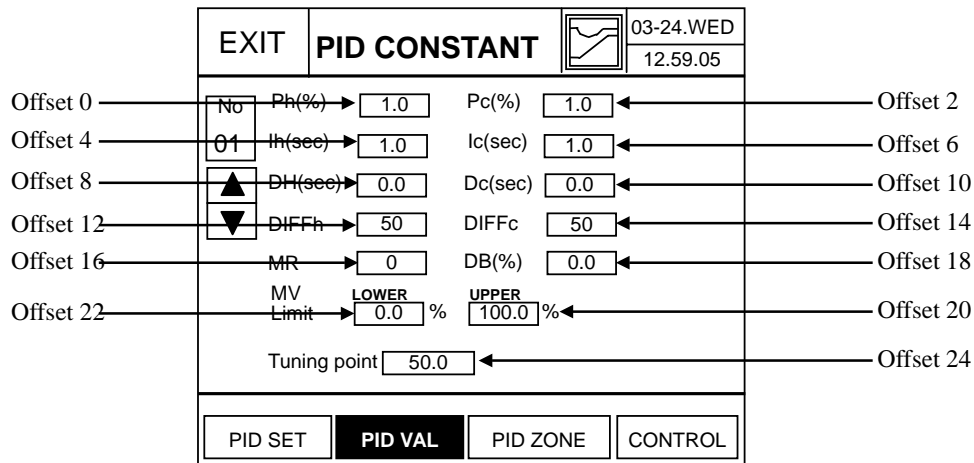


## 5.15 PID group register Map

A profile contains up to 8 PID groups. Each PID group is made up of 26 registers. This table contains PID group Map Address of Loop 1 and 2.

**Table 5-15.1 Each PID group register Map Addresses**

Start Address (Hex)	Start Address (Decimal)	End Address (Hex)	End Address (Decimal)	Description
CB78	52088	CB90	52112	PID group #1 of Loop1
CB92	52114	CBAA	52138	PID group #2 of Loop1
CBAC	52140	CBC4	52164	PID group #3 of Loop1
CBC6	52166	CBDE	52190	PID group #4 of Loop1
:	:	:	:	:
CC2E	52270	CC46	52294	PID group #8 of Loop1
CC48	52296	CC60	52320	PID group #1 of Loop2
CC62	52322	CC7A	52346	PID group #2 of Loop2
CC7C	52348	CC94	52372	PID group #3 of Loop2
CC96	52374	CCAE	52398	PID group #4 of Loop2
:	:	:	:	:
CCFE	52478	CD16	52502	PID group #8 of Loop2



**< Figure 5-15 > PID set screen**

The table below describes the registers that are part of a PID group Register. To determine the actual register address for a parameter within a PID group, add the register offset to the start address of the PID group.

See Figure 5-15 for mapping each register position on the PID set screen.

**Table 5-15.2 PID group register Map Addresses**

Register Offset within PID Group	Parameter Name	Access	Notes
0	Gain or PB (HEAT)	R/W	Floating Point Set Range : 0.0 to 9999.0
2	Gain or PB (COOL)	R/W	Floating Point Set Range : 0.0 to 9999.0
4	Reset (HEAT)	R/W	Floating Point Set Range : 0.02 to 50.0
6	Reset (COOL)	R/W	Floating Point Set Range : 0.02 to 50.0
8	Rate (HEAT)	R/W	Floating Point Set Range : 0.0 to 10.0
10	Rate (COOL)	R/W	Floating Point Set Range : 0.0 to 10.0
12	Differential (HEAT)	R/W	Floating Point Set Range : 0.0 to 100.0
14	Differential (COOL)	R/W	Floating Point Set Range : 0.0 to 100.0
16	Manual Reset	R/W	Floating Point Set Range : -100.0 to 100.0
18	Dead band	R/W	Floating Point Set Range : -5.0 to 25.0
20	Output high limit	R/W	Floating Point Set Range : -5.0 to 105.0 and Output High Limit > Output Low Limit
22	Output low limit	R/W	
24	Auto tuning setpoint	R/W	Floating Point Set Range : SP Low/High Limit

**How to find the Register address of each parameter registered to PID group of Loop 1 :**

Register address = Start address of PID group #1 of Loop 1+ (PID group number – 1) \* 26 + Offset value of the parameter from Table 5-15.2.

**Example :** Find the register address of Differential(HEAT) of PID group #6

The Register address = CB78 hex + (6-1) \* 26 + 12 = CC06 hex

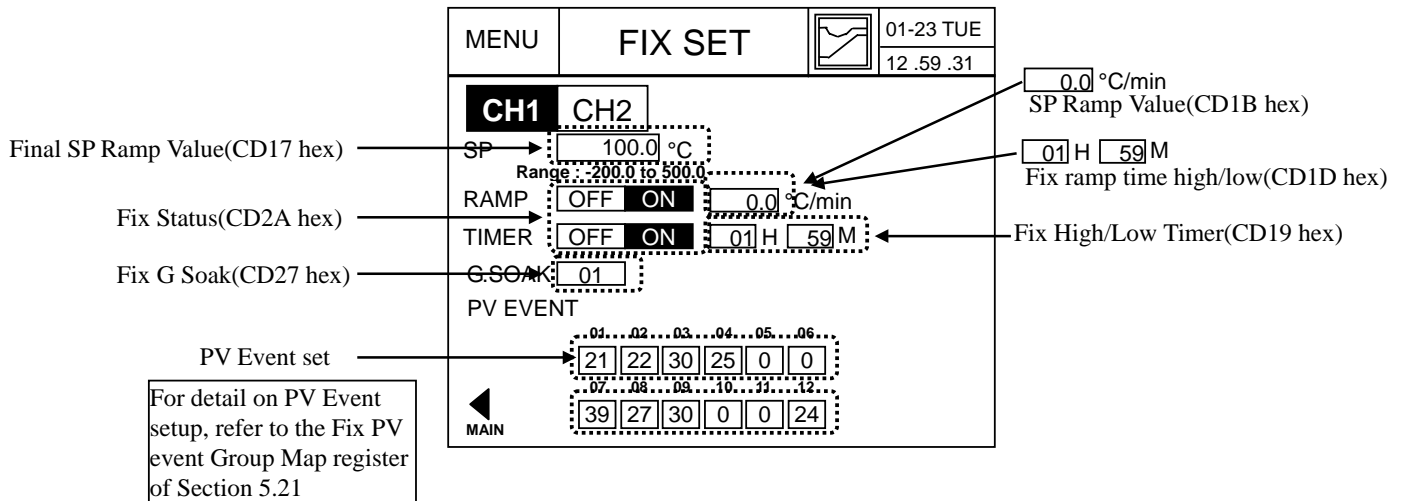
## 5.16 Controller group register Map

This Controller group is made to be needed for Constant(=Fix) Control and there are two groups for Loop 1(CD17 to CD2A) and Loop 2(CD2B to CD3E).

**Table 5-16 Controller group register Map addresses**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
CD17	52503	Final SP Ramp Value	R/W	Floating Point Set Range : SP Low/High Limit
CD19	52505	Fix High/Low Timer	R/W	Floating Point Set Range : 00.00 to 99.59
CD1B	52507	SP Ramp Value	R/W	Floating Point Set Range : 0.0 to 9.9
CD1D	52509	Fix ramp time high/low	R/W	Floating Point Set Range : 00.00 to 99.59
CD27	52519	Fix G Soak	R/W	Signed 16 bit Integer 0: NONE; 1:FRONT; 2:REAR; 3:ALL
CD2A	52522	Fix Status	R/W	Bit Packed Bit 0: SP Ramp 0: OFF; 1:ON Bit 1: Timer 0:OFF; 1:ON

See Figure 5-16 for mapping each register position on the FIX set screen.



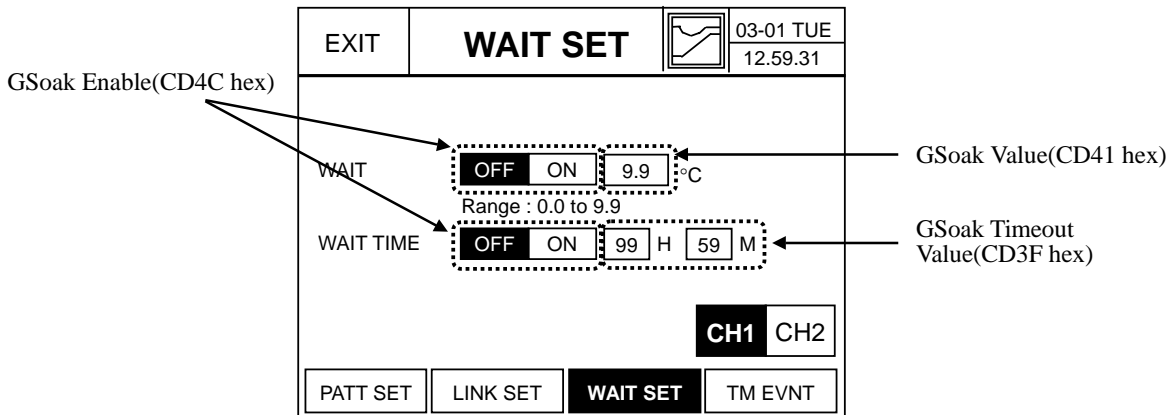
**< Figure 5-16 > Fix Set screen**

## 5.17 Program group register Map

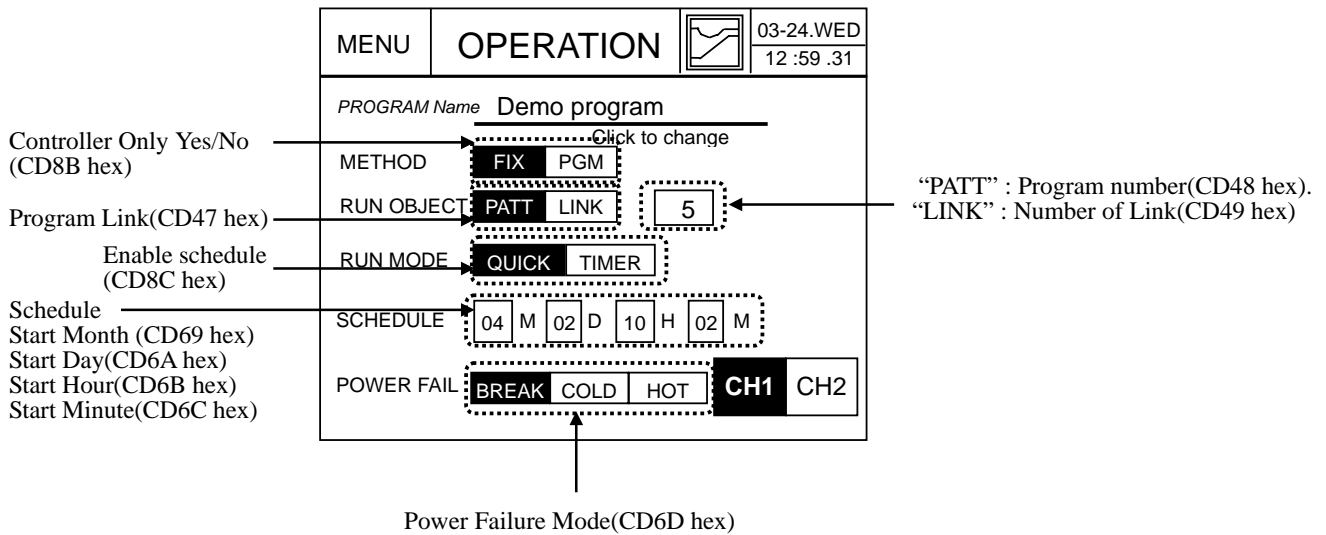
This Program group is made to be needed for Programmer Control and there are two groups for Loop 1(CD3F to CD4C) and Loop 2(CD4D to CD5A).

**Table 5-17 Program group register Map addresses**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
CD3F	52543	GSoak Timeout Value	R/W	Floating Point Set Range : 00.00 to 99.59 See Figure 5-17.1 for mapping this register position on the Wait set screen.
CD41	52545	GSoak Value	R/W	Floating Point Set Range : 0.0 to 9.9 See Figure 5-17.1 for mapping this register position on the Wait set screen.
CD47	52551	Program Link	R/W	Signed 16 bit Integer 0: PATT; 1: LINK See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD48	52552	Program Number	R/W	Signed 16 bit Integer Set Range : 0 to 15 for Loop 1 and 16 to 31 for Loop 2 at Async mode 0 to 31 at Sync mode See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD49	52553	Number of Links	R/W	Signed 16 bit Integer Set Range : 0 to 9 See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD4C	52556	GSoak Enable	R/W	Bit Packed Bit 0: Value Enable 0: OFF; 1:ON Bit 1: Timeout Enable 0:OFF; 1:ON See Figure 5-17.1 for mapping this register position on the Wait set screen.



**< Figure 5-17.1 > WAIT Setup screen**



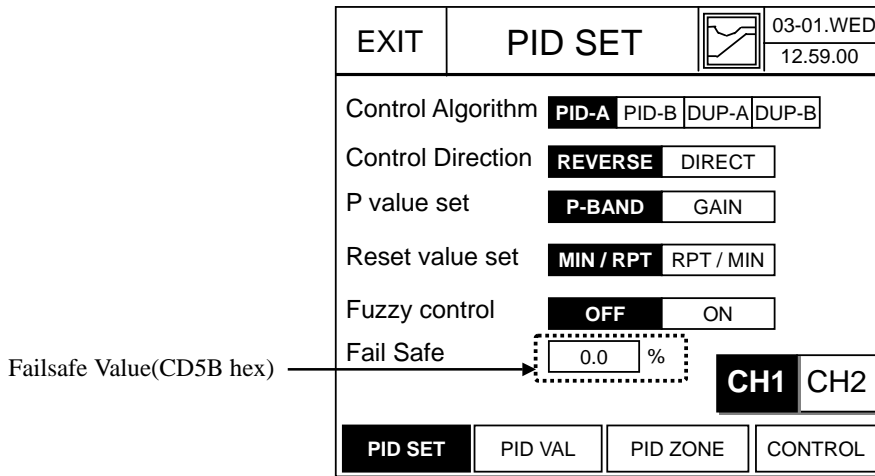
< Figure 5-17.2 > Operation Setup screen

## 5.18 Setup Group Register Map

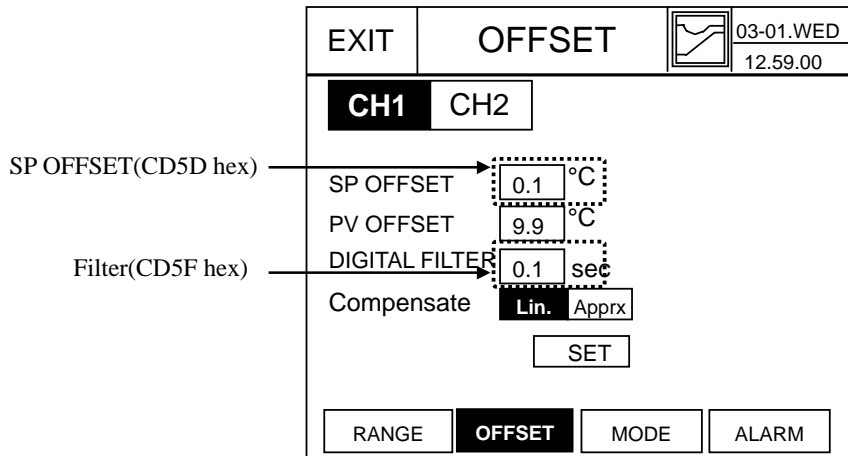
This Setup group is made to be needed to setup a system and there are two groups for Loop 1(CD5B to CD72) and Loop 2(CD73 to CD8A).

**Table 5.18 Setup group register Map addresses**

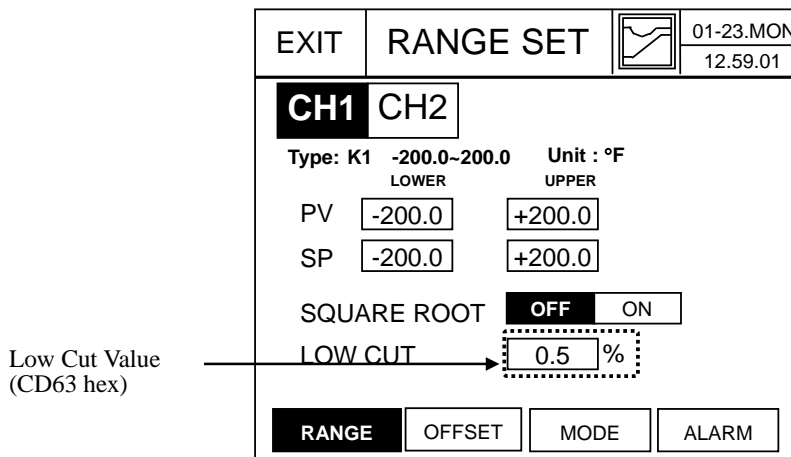
Address (hex)	Register (decimal)	Channel Number	Access	Notes
CD5B	52571	Failsafe Value	R/W	Floating Point Set Range : -5.0 to 105.0 See Figure 5-18.1 for mapping this register position on the PID set screen.
CD5D	52573	SP OFFSET	R/W	Floating Point Set Range : -99.9 to 99.9 See Figure 5-18.2 for mapping this register position on the OFFSET screen.
CD5F	52575	Filter	R/W	Floating Point Set Range : 0.0 to 120.0 See Figure 5-18.2 for mapping this register position on the OFFSET screen.
CD61	52577	Preset Manual Value	R/W	Floating Point Set Range : -5.0 to 105.0 See Figure 5-18.4 for mapping this register position on the CONTROL set screen.
CD63	52579	Low Cut Value	R/W	Floating Point Set Range : 0.0 to 5.0 See Figure 5-18.3 for mapping this register position on the RANGE set screen.
CD69	52585	Schedule Start Month	R/W	Signed 16 bit Integer Set Range : 1 to 12 See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD6A	52586	Schedule Start Day	R/W	Signed 16 bit Integer Set Range : 1 to 31 See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD6B	52587	Schedule Start Hour	R/W	Signed 16 bit Integer Set Range : 0 to 23 See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD6C	52588	Schedule Start Minute	R/W	Signed 16 bit Integer Set Range : 0 to 59 See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD6D	52589	Power Failure Mode	R/W	Signed 16 bit Integer 0: BREAK; 1:COLD; 2: HOT See Figure 5-17.2 for mapping this register position on the Operation set screen.



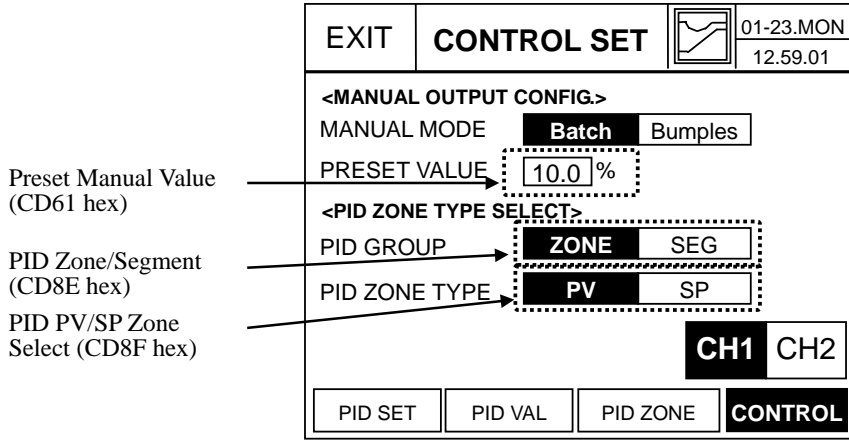
< Figure 5-18.1 > PID Set screen



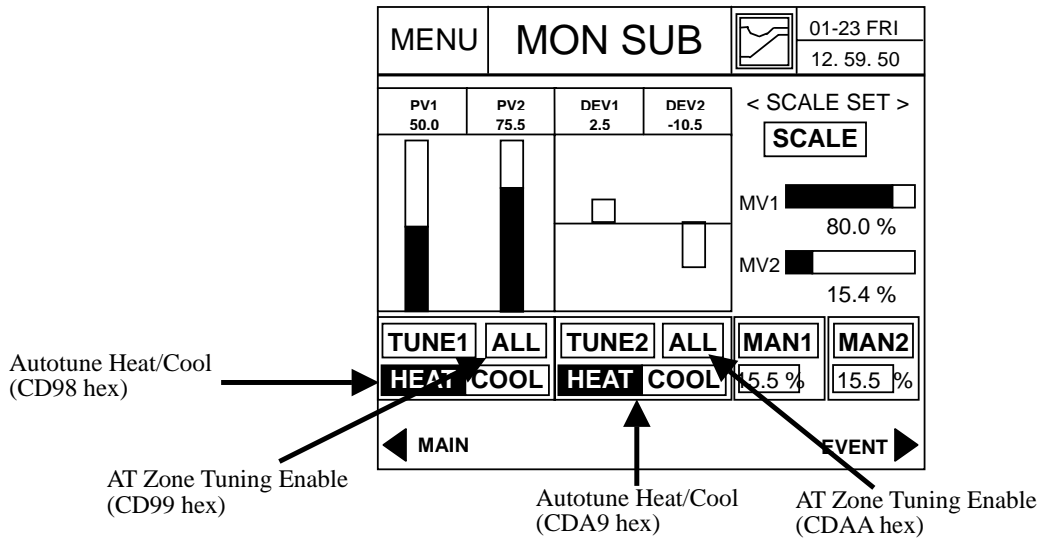
< Figure 5-18.2 > OFFSET screen



< Figure 5-18.3 > RANGE set screen



< Figure 5-18.4 > CONTROL SET screen



< Figure 5-18.5 > MON SUB screen

## 5.19 Operation group Register Map

This Operation group is made to be needed to check and monitor a system configuration and there are two groups for Loop 1(CD8B to CD9B) and Loop 2(CD9C to CDAC).

**Table 5.19 Operation group register Map addresses**

Address (hex)	Register (decimal)	Channel Number	Access	Notes
CD8B	52619	Controller Only Yes/No	R/W	Signed 16 Integer 0: Controller mode; 1: Programmer mode See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD8C	52620	Enable Schedule	R/W	Signed 16 Integer 0:QUICK; 1: TIMER See Figure 5-17.2 for mapping this register position on the Operation set screen.
CD8D	52621	AT Enable	R	Signed 16 Integer 0: AT OFF; 1: AT ON
CD8E	52622	PID Zone/Segment	R/W	Signed 16 Integer 0:ZONE PID; 1: SEGMENT PID See Figure 5-18.4 for mapping this register position on the CONTROL set screen.
CD8F	52623	PID PV/SP Zone Select	R/W	Signed 16 Integer 0: PV; 1: SP See Figure 5-18.4 for mapping this register position on the CONTROL set screen.
CD90	52624	Current Part Repeat count	R	Signed 16 Integer
CD91	52625	Current All Repeat count	R	Signed 16 Integer
CD92	52626	Total time(Hour)	R	Signed 16 Integer
CD93	52627	Total time(Min)	R	Signed 16 Integer
CD94	52628	Current Link number	R	Signed 16 Integer
CD95	52629	Current PID group number	R	Signed 16 Integer
CD97	52631	Analog Input Status	R	0: Normal; 1: High over; 2: Low over; 3: IN Fail(Burn out)
CD98	52632	Autotune Heat/Cool	R/W	Signed 16 Integer 0: Heat tuning; 1: Cool tuning See Figure 5-18.5 for mapping this register position on the MON SUB screen.
CD99	52633	AT Zone Tuning Enable	R/W	Signed 16 Integer 0: Disable Automatic tuning 1: Enable Automatic tuning See Figure 5-18.5 for mapping this register position on the MON SUB screen.

## 5.20 No Loop group Register Map

**Table 5.20 No Loop group register Map addresses**

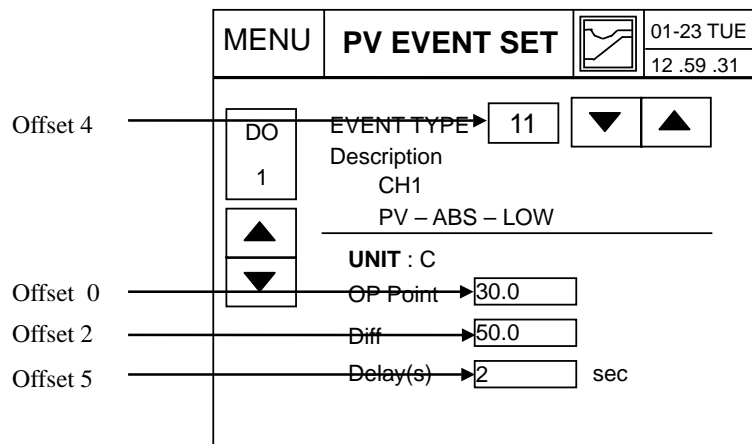
Address (hex)	Register (decimal)	Channel Number	Access	Notes
CDF5	52575	DOSET Status	R	Signed 16 bit Integer If Mode or Alarm event was assigned to Digital outputs, the bit of the digital output is set to 0. Bit 0 : 0: Set disable; 1: Set Enable; Digital output 1 Bit 1 : 0: Set disable; 1: Set Enable; Digital output 2 Bit 2 : 0: Set disable; 1: Set Enable; Digital output 3 Bit 3 : 0: Set disable; 1: Set Enable; Digital output 4 Bit 4 : 0: Set disable; 1: Set Enable; Digital output 5 Bit 5 : 0: Set disable; 1: Set Enable; Digital output 6 Bit 6 : 0: Set disable; 1: Set Enable; Digital output 7 Bit 7 : 0: Set disable; 1: Set Enable; Digital output 8 Bit 8 : 0: Set disable; 1: Set Enable; Digital output 9 Bit 9 : 0: Set disable; 1: Set Enable; Digital output 10 Bit 10 : 0: Set disable; 1: Set Enable; Digital output 11 Bit 11 : 0: Set disable; 1: Set Enable; Digital output 12
CDF6	52576	Time Unit	R/W	Signed 16 bit Integer 0:HH:MM; 1:MM:SS
CDF7	52577	Key Lock(Auto tuning)	R/W	Signed 16 bit Integer 0: Unlock; 1: Lock

## 5.21 Fix PV Event group Register Map

One Fix PV Event group can be set up to 12 digital output numbers and each group is same with digital output number. There is six registers in one group and the parameter is described in Table 5-21. Refer to figure 5.21 for screen position of each parameter. The start address of Fix PV Event group is CDAD Hex and The End address is CDF4 Hex.

**Table 5.20 Fix PV Event group register Map addresses**

Register Offset within Fix PV Event Group	Parameter Name	Access	Notes
0	OP point or Max Value	R/W	Floating Point OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
2	Min Value or Differential	R/W	Floating Point Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
4	Event type number	R/W	Signed 16 bit Integer 0: OFF 11: PV-ABS-LOW for Channel 1(PV Event) 12: PV-ABS-LOW for Channel 2(PV Event) 13: PV-ABS-HIGH for Channel 1 (PV Event) 14: PV-ABS-HIGH for Channel 2 (PV Event) 15: PV-DEV-LOW for Channel 1 (PV Event) 16: PV- DEV-LOW for Channel 2 (PV Event) 17: PV-DEV-HIGH for Channel 1 (PV Event) 18: PV-DEV-HIGH for Channel 2 (PV Event) 21: SP-ABS-LOW for Channel 1 (PV Event) 22: SP-ABS-LOW for Channel 2 (PV Event) 23: SP-ABS-HIGH for Channel 1 (PV Event) 24: SP-ABS-HIGH for Channel 2 (PV Event) 25: SP-DEV-OFF for Channel 1 (PV Event) 26: SP-DEV-OFF for Channel 2 (PV Event) 27: SP-DEV-ON for Channel 1 (PV Event) 28: SP-DEV-ON for Channel 2 (PV Event) 31: DV-DEV-OFF for Channel 1 (PV Event) 32: DV-DEV-OFF for Channel 2 (PV Event) 33: DV-DEV-ON for Channel 1 (PV Event) 34: DV-DEV-ON for Channel 2 (PV Event) 35: MV-ABS-LOW for Channel 1 (PV Event) 36: MV-ABS-LOW for Channel 2 (PV Event) 37: MV-ABS-HIGH for Channel 1 (PV Event) 38: MV-ABS-HIGH for Channel 2 (PV Event)
5	Time Delay	R/W	Signed 16 bit Integer Set Range: 0 ~ 99



< Figure 5-21 > Fix PV Event set screen

**How to find the Register address of each parameter registered to PID group of Loop 1 :**

Register address = Start address of Fix PV Event group #1 of Loop 1+ (Digital output number – 1) \* 6 + Offset value of the parameter from Table 5.21.

**Example :** Find the register address of Event type number of digital output number #6

The Register address = CDAD hex + (6-1) \* 6 + 04 hex = CDCF hex

---

## 6. CRC-16 Calculation for Modbus RTU

See following function:

```
extern void calculate_CRC(unsigned char *message, int length, unsigned char *CRC)
{
    unsigned char CRCHi, CRCLo, TempHi, TempLo;
    static const unsigned char table[512] = {
        0x00, 0x00, 0xC0, 0xC1, 0xC1, 0x81, 0x01, 0x40, 0xC3, 0x01, 0x03, 0xC0, 0x02, 0x80, 0xC2, 0x41,
        0xC6, 0x01, 0x06, 0xC0, 0x07, 0x80, 0xC7, 0x41, 0x05, 0x00, 0xC5, 0xC1, 0xC4, 0x81, 0x04, 0x40,
        0xCC, 0x01, 0x0C, 0xC0, 0x0D, 0x80, 0xCD, 0x41, 0x0F, 0x00, 0xCF, 0xC1, 0xCE, 0x81, 0x0E, 0x40,
        0x0A, 0x00, 0xCA, 0xC1, 0xCB, 0x81, 0x0B, 0x40, 0xC9, 0x01, 0x09, 0xC0, 0x08, 0x80, 0xC8, 0x41,
        0xD8, 0x01, 0x18, 0xC0, 0x19, 0x80, 0xD9, 0x41, 0x1B, 0x00, 0xDB, 0xC1, 0xDA, 0x81, 0x1A, 0x40,
        0x1E, 0x00, 0xDE, 0xC1, 0xDF, 0x81, 0x1F, 0x40, 0xDD, 0x01, 0x1D, 0xC0, 0x1C, 0x80, 0xDC, 0x41,
        0x14, 0x00, 0xD4, 0xC1, 0xD5, 0x81, 0x15, 0x40, 0xD7, 0x01, 0x17, 0xC0, 0x16, 0x80, 0xD6, 0x41,
        0xD2, 0x01, 0x12, 0xC0, 0x13, 0x80, 0xD3, 0x41, 0x11, 0x00, 0xD1, 0xC1, 0xD0, 0x81, 0x10, 0x40,
        0xF0, 0x01, 0x30, 0xC0, 0x31, 0x80, 0xF1, 0x41, 0x33, 0x00, 0xF3, 0xC1, 0xF2, 0x81, 0x32, 0x40,
        0x36, 0x00, 0xF6, 0xC1, 0xF7, 0x81, 0x37, 0x40, 0xF5, 0x01, 0x35, 0xC0, 0x34, 0x80, 0xF4, 0x41,
        0x3C, 0x00, 0xFC, 0xC1, 0xFD, 0x81, 0x3D, 0x40, 0xFF, 0x01, 0x3F, 0xC0, 0x3E, 0x80, 0xFE, 0x41,
        0xFA, 0x01, 0x3A, 0xC0, 0x3B, 0x80, 0xFB, 0x41, 0x39, 0x00, 0xF9, 0xC1, 0xF8, 0x81, 0x38, 0x40,
        0x28, 0x00, 0xE8, 0xC1, 0xE9, 0x81, 0x29, 0x40, 0xEB, 0x01, 0x2B, 0xC0, 0x2A, 0x80, 0xEA, 0x41,
        0xEE, 0x01, 0x2E, 0xC0, 0x2F, 0x80, 0xEF, 0x41, 0x2D, 0x00, 0xED, 0xC1, 0xEC, 0x81, 0x2C, 0x40,
        0xE4, 0x01, 0x24, 0xC0, 0x25, 0x80, 0xE5, 0x41, 0x27, 0x00, 0xE7, 0xC1, 0xE6, 0x81, 0x26, 0x40,
        0x22, 0x00, 0xE2, 0xC1, 0xE3, 0x81, 0x23, 0x40, 0xE1, 0x01, 0x21, 0xC0, 0x20, 0x80, 0xE0, 0x41,
        0xA0, 0x01, 0x60, 0xC0, 0x61, 0x80, 0xA1, 0x41, 0x63, 0x00, 0xA3, 0xC1, 0xA2, 0x81, 0x62, 0x40,
        0x66, 0x00, 0xA6, 0xC1, 0xA7, 0x81, 0x67, 0x40, 0xA5, 0x01, 0x65, 0xC0, 0x64, 0x80, 0xA4, 0x41,
        0x6C, 0x00, 0xAC, 0xC1, 0xAD, 0x81, 0x6D, 0x40, 0xAF, 0x01, 0x6F, 0xC0, 0x6E, 0x80, 0xAE, 0x41,
        0xAA, 0x01, 0x6A, 0xC0, 0x6B, 0x80, 0xAB, 0x41, 0x69, 0x00, 0xA9, 0xC1, 0xA8, 0x81, 0x68, 0x40,
        0x78, 0x00, 0xB8, 0xC1, 0xB9, 0x81, 0x79, 0x40, 0xBB, 0x01, 0x7B, 0xC0, 0x7A, 0x80, 0xBA, 0x41,
        0xBE, 0x01, 0x7E, 0xC0, 0x7F, 0x80, 0xBF, 0x41, 0x7D, 0x00, 0xBD, 0xC1, 0xBC, 0x81, 0x7C, 0x40,
        0xB4, 0x01, 0x74, 0xC0, 0x75, 0x80, 0xB5, 0x41, 0x77, 0x00, 0xB7, 0xC1, 0xB6, 0x81, 0x76, 0x40,
        0x72, 0x00, 0xB2, 0xC1, 0xB3, 0x81, 0x73, 0x40, 0xB1, 0x01, 0x71, 0xC0, 0x70, 0x80, 0xB0, 0x41,
        0x50, 0x00, 0x90, 0xC1, 0x91, 0x81, 0x51, 0x40, 0x93, 0x01, 0x53, 0xC0, 0x52, 0x80, 0x92, 0x41,
        0x96, 0x01, 0x56, 0xC0, 0x57, 0x80, 0x97, 0x41, 0x55, 0x00, 0x95, 0xC1, 0x94, 0x81, 0x54, 0x40,
        0x9C, 0x01, 0x5C, 0xC0, 0x5D, 0x80, 0x9D, 0x41, 0x5F, 0x00, 0x9F, 0xC1, 0x9E, 0x81, 0x5E, 0x40,
        0x5A, 0x00, 0x9A, 0xC1, 0x9B, 0x81, 0x5B, 0x40, 0x99, 0x01, 0x59, 0xC0, 0x58, 0x80, 0x98, 0x41,
        0x88, 0x01, 0x48, 0xC0, 0x49, 0x80, 0x89, 0x41, 0x4B, 0x00, 0x8B, 0xC1, 0x8A, 0x81, 0x4A, 0x40,
        0x4E, 0x00, 0x8E, 0xC1, 0x8F, 0x81, 0x4F, 0x40, 0x8D, 0x01, 0x4D, 0xC0, 0x4C, 0x80, 0x8C, 0x41,
        0x44, 0x00, 0x84, 0xC1, 0x85, 0x81, 0x45, 0x40, 0x87, 0x01, 0x47, 0xC0, 0x46, 0x80, 0x86, 0x41,
        0x82, 0x01, 0x42, 0xC0, 0x43, 0x80, 0x83, 0x41, 0x41, 0x00, 0x81, 0xC1, 0x80, 0x81, 0x40, 0x40,
    };
    CRCHi = 0xff;
    CRCLo = 0xff;
    while(length)
    {
        TempHi = CRCHi;
        TempLo = CRCLo;
        CRCHi = table[2 * (*message ^ TempLo)];
        CRCLo = TempHi ^ table[(2 * (*message ^ TempLo)) + 1];
        message++;
        length--;
    };
    CRC [0] = CRCLo;
    CRC [1] = CRCHi;
    return;
}
```

---

## 7. RS-232C communication

### 7.1 Overview

IPC5000 has RS-232C communication function basically. This manual shows Communication Specification , Command, Response and system about communication of a Host computer and IPC5000 through RS-232C.

### 7.2 Communication Configuration

#### Methods

- NETWORK : 1 (Master) : 1 (Slave) SINGLE-DROP
- Mode for serial communication : Asynchronous mode

#### SPEED :

9600 or 19200 BPS( See Figure 1-2.2 of page 22)

#### Parity Check :

NONE

#### Bit Length :

8 bits

#### Stop Bits :

1 bit

#### X Control :

Ignoring

#### RS/CS Control :

Ignoring

---

## 7.3 Communication Function Code

### Introduction

Configuration Modbus function code 20 and 21 are used and package data is ASCII type.

### Register address structure

**Table 7-3.1 Integer Parameter Type**

Register Numbers	Name	Access	Notes
3	Value(16 bit integer)	Read/Write	16 bit Unsigned integer
4	Program data(16 bit integer)	Read/Write	

**Table 7-3.2 Floating Point Parameter Type**

Register Numbers	Name	Access	Notes
3	Value	Read/Write	IEEE Floating Point
4	Program data	Read/Write	

### Register Count

The register count depends on the data format of the registers being read or written.

Integer data is represented in sixteen bits and is transferred high byte first.

Floating point data is transferred in IEEE 32-bit format.

The register count definitions are :

0001 = Integer Data

0002 = Floating Point Data

## 7.4 Function Code 20(14h) - Read Configuration Reference Data

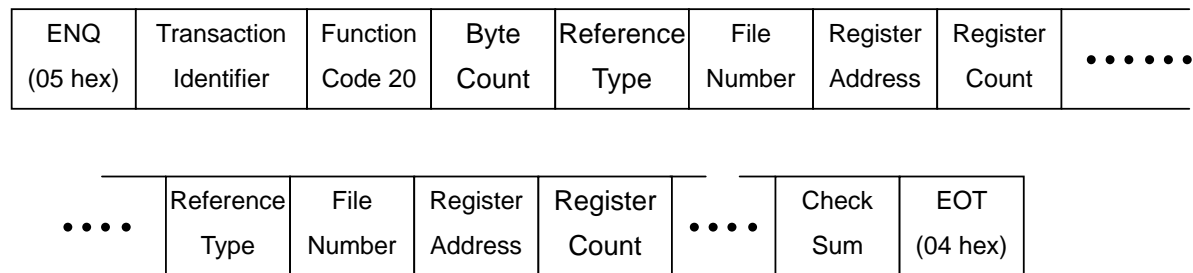
### Description

Function code 20(14 Hex) is used in the IPC5000 to read information stored in the IPC5000 configuration database. Each IPC5000 configuration item is explicitly addressed by a file number and register address. IEEE 32-bit floating point and 16-bit integer formats are supported.

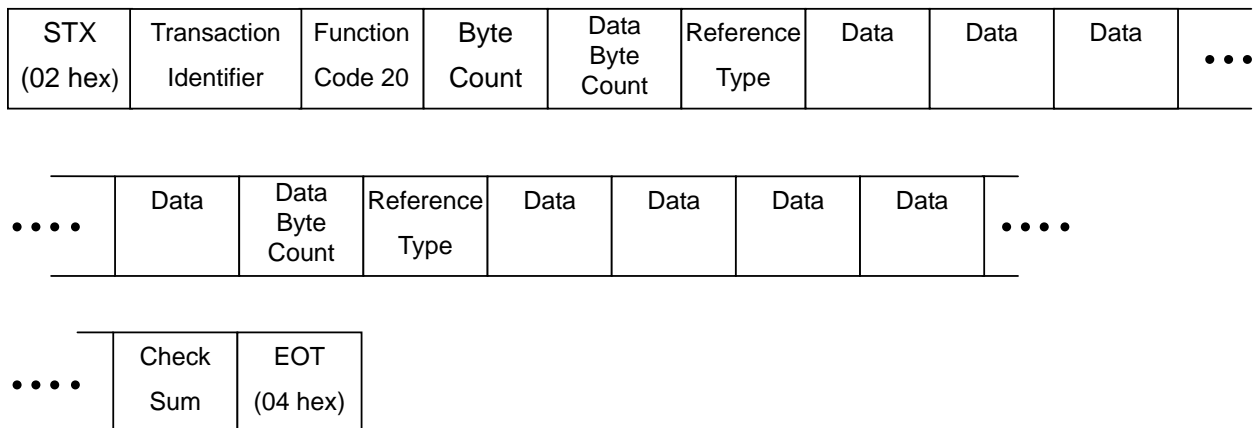
### Request and Response Formats

The Request and Response formats for Function code 20(14 Hex) are shown below. Detail for each block reference follow.

#### Request Message Format



#### Response Message Format



#### Transaction Identifier(8 bytes)

Transaction Identifier means transaction number of message and a Host sends the same number into response message. Transaction Identifier is used for classifying transaction messages.

---

## Byte Count

The Byte Count equals the number of bytes transmitted in either the request or response message and will be the minimum number required to transmit all requested data.

## Data Byte Count

The Data Byte Count is the number of data bytes of the *sub response* including the Reference Type but not including itself. A floating point sub response has four bytes of data and one byte representing the reference type making the data byte count equal to five.

## Reference Type Definitions

The Reference Type definition is always 06.

*See examples in Subsection 7.4.1*

## File Number

The file number word contains the register number from the register address structure tables on page 3.

**Table 7-4.1 File Number and Register address structure**

File Number	Register address structure
0003 (30, 30, 30, 33 hex)	SP/PV block memory, floating, integer value
0004 (30, 30, 30, 34 hex)	Program data memory(SP1,SP2, Segment time ...)

## Register Address

The register address word represents each data of IPC5000 for the parameter(s) being accessed.

The register address word is made up of two bytes—Register address map structure as follow Table 7-4.2 and 7-4.3

See *Sub Section 7.6* for the detail information.

**Table 7-4.2 Register address range of File number 0003**

Title	Sub Title	Register Map Range(Hex)	Type
SP Block		0000 to 1F00	INT and FP
PV Block		2000 to 3F00	INT and FP
Operation button action	RUN, STOP, HOLD, ADV, TUNE, MANUAL	4000 to 401F	INT
Link List Group		4110 to 4178	INT
Fix PV Event Group	Event type number, Time Delay	4200 to 4216	INT
	OP or Min Value and Diff or Max Value	9000 to 9017	FP
Mode Event Group		4300 to 434F	INT
Alarm Event	Mode, Assign, Action, Direction	4400 to 449F	INT
	Min and Max Value	905C to 9083	FP
Analog Input Group		5000 to 501E	INT
Analog Output Group		5020 to 503C	INT
Analog PWM Group		5040 to 504C	INT
Digital Output Group		5050 to 5058	INT
PID Constant Group		8000 to 80D0	FP
PID Reference Group		80DD to 80EC	FP
CP and Bias Group		8100 to 8127	FP
IP Address		FFF4 to FFF5	INT
Gateway Address		FFF8 to FFF9	INT
Subnet Mask Address		FFFC to FFFD	INT

**Table 7-4.3 Register address range of File number 0004**

Title	Sub Title	Register Map Range(Hex)	Type
Program Data	Start mode, Priority, Segment count, All Repeat Count, Start and End Segment, Partial Repeat count(#1 to #5), etc.	0000 to 0025	INT
Segment Data	PID Group 1 and 2, Program End(JC), G Soak, Time/Rate	0026 to 040D	INT
Segment Event Data	Event type number, Time Delay, Select	040E to 202D	INT
Program Data	Start SP 1 and 2	3300 to 3303	FP
Segment Data	SP1, SP2 and Segment time	3000 to 3257	FP
Segment Event Data	OP point or Max value, Differential or Min value	3400 to 46BF	FP

**Note**

- 1) INT : Signed 16 bit Integer
- 2) FP : Floating Point

## 7.4.1 Modbus Protocol Read Configuration Example

### Example 1

The following is an example of a request to read the Target SP of channel 1 on FIX SET screen using Function code 20.

Request Message (Read Target SP of Loop 1) = 0800 hex

ENQ00000000140706000308000002(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
00000000	Transaction Identifier(30H,30H, 30H,30H, 30H,30H, 30H,30H)
14	Function Code 20(31H, 34H)
07	Byte Count(30H, 37H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
0800	Register Address : Target SP of Loop 1(30H, 38H, 30H, 30H)
0002	Register Count : Floating Point Data(30H, 30H, 30H, 32H)
Check Sum	Exclusive OR Value from Function code to Register Count(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message

STX000000001406050642480000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
00000000	Transaction Identifier(30H,30H, 30H,30H, 30H,30H, 30H,30H)
14	Function Code 20(31H, 34H)
06	Byte Count(30H, 36H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
42480000	50.0 : Target SP Value of Loop 1(34H,32H,34H,38H,30H,30H,30H,30H)
Check sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

## Example 2

The following is an example of a request to read the Process Value of Loop 1 and 2 using Function code 20.

Request Message (Read Process Value of Loop 1 and 2)  
 ENQ000000A0140E0600033000000206000330010002(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
000000A0	Transaction Identifier(30H,30H,30H,30H,30H,30H,41H,30H)
14	Function Code 20(31H, 34H)
0E	Byte Count(30H, 45H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
3000	Register Address : Process Value of Loop 1(33H, 30H, 30H, 30H)
0002	Register Count : Floating Point Data(30H, 30H, 30H, 32H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
3001	Register Address : Process Value of Loop 1(33H, 30H, 30H, 31H)
0002	Register Count : Floating Point Data(30H, 30H, 30H, 32H)
Check Sum	Exclusive OR Value from Function code to Register Count(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message  
 STX000000A0140C050643C80000050644600000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
000000A0	Transaction Identifier(30H,30H, 30H,30H, 30H,30H, 41H,30H)
14	Function Code 20(31H, 34H)
0C	Byte Count(30H, 43H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
43C80000	400.0 : Process Value of Loop 1(34H,33H,43H,38H,30H,30H,30H,30H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
44600000	896.0 : Process Value of Loop 2(34H,34H,36H,30H,30H,30H,30H,30H)
Check sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

### Example 3

The following is an example of a request to read the Ramp Value and OFF/ON status of Loop 1 on FIX SET screen using Function code 20.

Request Message (Read Ramp status and value of Loop 1 for Fix control)  
 ENQ000001A1140E0600030100000106000308020002(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
000001A1	Transaction Identifier(30H,30H,30H,30H,30H,31H,41H,31H)
14	Function Code 20(31H, 34H)
0E	Byte Count(30H, 45H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
0100	Register Address : OFF/ON status of Loop 1(30H, 31H, 30H, 30H)
0001	Register Count : Integer Data(30H, 30H, 30H, 31H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
0802	Register Address : Ramp Value of Loop 1(30H, 38H, 30H, 32H)
0002	Register Count : Floating Point Data(30H, 30H, 30H, 32H)
Check Sum	Exclusive OR Value from Function code to Register Count(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message  
 STX000001A1140A03060000050640000000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
000001A1	Transaction Identifier(30H,30H, 30H,30H, 30H,31H, 41H,31H)
14	Function Code 20(31H, 34H)
0A	Byte Count(30H, 41H)
03	Sub Message Length(30H, 33H)
06	Reference Type : IEEE Floating Point(30H, 36H)
0000	The Ramp status of Loop 1 is OFF(30H,30H,30H,30H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
40000000	2.0 : Ramp Value of Loop 1(34H,30H,30H,30H,30H,30H,30H,30H)
Check sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

#### Example 4

The following is an example of a request to read the Working Set Point of Loop 1 and 2 using Function code 20.

Request Message (Read the Working Set Point of Loop 1 and 2)  
 ENQ0000000140E0600032A0000020600032A010002(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
14	Function Code 20(31H, 34H)
0E	Byte Count(30H, 45H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
2A00	Register Address : Working Set Point of Loop 1(32H, 41H, 30H, 30H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
2A01	Register Address : Working Set Point of Loop 2(32H, 41H, 30H, 31H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
Check Sum	Exclusive OR Value from Function code to Register Count(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message  
 STX00000000140C050642480000050642A00000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
00000000	Transaction Identifier(30H,30H, 30H,30H, 30H,30H, 30H,30H)
14	Function Code 20(31H, 34H)
0C	Byte Count(30H, 43H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
42480000	50.0 : Working Set Point of Loop 1(34H,32H,34H,38H,30H,30H,30H,30H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
42A00000	80.0 : Working Set Point of Loop 2(34H,32H,41H,30H,30H,30H,30H,30H)
Check sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

## Example 5

The following is an example of a request to read the Target SP 1 and Segment time of Segment #0 registered to Program #0 using Function code 20.(Before reading, The program #0 has to be set to the register 290C Hex for program edition)

Request Message (Read the Target SP 1 and time of Segment #0 registered to Program #0)

ENQ00000000140E06000430000002060004300020002(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
14	Function Code 20(31H, 34H)
0E	Byte Count(30H, 45H)
06	Reference Type(30H, 36H)
0004	File number : Access Data Value(30H, 30H, 30H, 34H)
3000	Register Address : Target SP 1(33H, 30H, 30H, 30H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
06	Reference Type(30H, 36H)
0004	File number : Access Data Value(30H, 30H, 30H, 34H)
3002	Register Address : Segment time(33H, 30H, 30H, 32H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
Check Sum	Exclusive OR Value from Function code to Register Count(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message

STX00000000140E05064316000005063F800000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
00000000	Transaction Identifier(30H,30H, 30H,30H, 30H,30H, 30H,30H)
14	Function Code 20(31H, 34H)
0C	Byte Count(30H, 43H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
43160000	150.0 : Target SP1(34H,33H,31H,36H,30H,30H,30H,30H)
05	Sub Message Length(30H, 35H)
06	Reference Type : IEEE Floating Point(30H, 36H)
3FA66666	1.30 : Segment time(33H,46H,41H,36H,36H,36H,36H,36H)
Check sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

---

## 7.5 Function Code 21(15h) - Write Configuration Reference Data

### Introduction

Function Code 21(15 Hex) is used in the IPC5000 to allow writes of integer and floating point value to the IPC5000 configuration database and override values.

The configuration database of the IPC5000 is located in Flash Memory. The override values are stored in RAM.

Integer format is usually used to write to "Digital" configuration items. Floating Point format is mostly used to write to "Analog" configuration items.

### Request and Response Formats

The Request and Response formats for Function 21(15 Hex) are shown below. Detail for each block reference follow.

#### Request Message Format

ENQ (05 Hex)	Transaction Identifier	Function Code 21	Byte Count	Reference Type	File Number	Register Address	Register Count	.....
...	Data	Data	Data	Data	Reference Type	File Number	....	Check Sum EOT (04 Hex)

#### Response Message Format(echo back of request)

STX (02 Hex)	Transaction Identifier	Function Code 21	Byte Count	Reference Type	File Number	Register Address	Register Count	.....
...	Data	Data	Data	Data	Reference Type	File Number	....	Check Sum EOT (04 Hex)

#### Transaction Identifier(8 bytes)

Transaction Identifier means transaction number of message and a Host sends the same number into response message. Transaction Identifier is used for classifying transaction messages.

#### Reference Type Definitions

The Reference Type definition is always 06.

See examples in Subsection 7.5.1

---

## **File Number**

The file number word contains the register number from the register address structure table 7-4.2 and 7-4.3. File number 3 and 4 are currently supported for each general register and Program register.

## **Register Address**

The register address is used to designate the access type,

## **Restrictions on Parameter Numbers in One Message**

The Maximum number of writeable parameters per write request is 4.

## 7.5.1 Modbus Protocol Write Configuration Example

### Example 1

The following is an example of a request to write Gain value for Heating of PID group #1 of Loop 1 using Function code 21(15Hex)

Request Message (Write the Heating Gain value( = 1.5) of PID group #1 of Loop 1)

ENQ00000000150B060003800000023FC00000(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
15	Function Code 21(31H, 35H)
0B	Byte Count(30H, 42H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
8000	Register Address : Gain Value for Heating (38H, 30H, 30H, 30H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
3FC00000	1.5 : Data(33H,46H,43H,30H,30H,30H,30H,30H)
Check Sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message

STX00000000150B060003800000023FC00000(Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
15	Function Code 21(31H, 35H)
0B	Byte Count(30H, 42H)
06	Reference Type(30H, 36H)
0003	File number : Access Data Value(30H, 30H, 30H, 33H)
8000	Register Address : Gain Value for Heating (38H, 30H, 30H, 30H)
0002	Register Count : Floating Point data(30H, 30H, 30H, 32H)
3FC00000	1.5 : Data(33H,46H,43H,30H,30H,30H,30H,30H)
Check Sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

## Example 2

The following is an example of a request to write the Target SP and time of segment #0 registered to program #0 using Function code 21(15 Hex)(Before writing, The program #0 has to be set to the register 290C Hex for program edition)

Request Message (Write Target SP = 50.0 and time = 00.30 of segment #0)

ENQ00000000215160600043000000242480000060004300200023E99999A(Check sum)EOT

Where :

Protocol	Remarks
ENQ	Protocol Header(05H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
15	Function Code 21(31H, 35H)
16	Byte Count(31H,36H)
06	Reference Type(30H,36H)
0004	File number : Access Data Value(30H,30H,30H,34H)
3000	Register Address : Target SP (33H,30H,30H,30H)
0002	Register Count : Floating Point data(30H,30H,30H,32H)
42480000	50.0 : Data(34H,32H,34H,38H,30H,30H,30H,30H)
06	Reference Type(30H,36H)
0004	File number : Access Data Value(30H,30H,30H,34H)
3002	Register Address : Segment time (33H,30H,30H,32H)
0002	Register Count : Floating Point data(30H,30H,30H,32H)
3E99999A	00.30 : Data(33H,45H,39H,39H,39H,39H,39H,41H)
Check Sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

This is the response to the above request.

Response Message

STX0000000015160600043000000242480000060004300200023E99999A (Check sum)EOT

Where :

Protocol	Remarks
STX	Protocol Header(02H)
00000000	Transaction Identifier(30H,30H,30H,30H,30H,30H,30H,30H)
15	Function Code 21(31H, 35H)
16	Byte Count(31H,36H)
06	Reference Type(30H,36H)
0004	File number : Access Data Value(30H,30H,30H,34H)
3000	Register Address : Target SP (33H,30H,30H,30H)
0002	Register Count : Floating Point data(30H,30H,30H,32H)

Continued next page

---

Protocol	Remarks
42480000	50.0 : Data(34H,32H,34H,38H,30H,30H,30H,30H)
06	Reference Type(30H,36H)
0004	File number : Access Data Value(30H,30H,30H,34H)
3002	Register Address : Segment time (33H,30H,30H,32H)
0002	Register Count : Floating Point data(30H,30H,30H,32H)
3E99999A	00.30 : Data(33H,45H,39H,39H,39H,39H,39H,41H)
Check Sum	Exclusive OR Value from Function code to Data(2 bytes).
EOT	Protocol Tail(04H)

---

## 7.6 Read, Write and Override Parameter register addresses

### 7.6.1 SP/PV block Register address

SP/PV block is main data structure of IPC5000. The SP blocks are set data type, PV blocks are current status or value and most of PV blocks are read only variables. See Table 7-6.1 for maximum index number per block.

**Table 7-6.1 Maximum index number per block**

SP or PV	Block No. (Decimal)	Block No (Hex)	Maximum index number
SP	01	01	32
	02	02	32
	03	03	32
	04	04	32
	05	05	32
	06	06	16
	08	08	8
	09	09	8
	10	0A	8
	11	0B	8
	12	0C	8
	31	1F	32
PV	40	28	16
	41	29	16
	42	2A	8
	43	2B	16
	44	2C	16
	48	30	8
	49	31	8
	50	32	8

## 7.6.2 SP block Register Map

Table 7-6.2 SP block register map address of 0100 to 011F hex range

SP Block Address (Hex)	Loop	Information	Data type	R/W	Remarks
0100H	1	Fix Ramp Enable	INT	R/W	0: OFF; 1: ON
0101H	2				
0102H	1	Fix Timer Enable	INT	R/W	0: OFF; 1: ON
0103H	2				
0106H	1	Fix Timer minutes	INT	R/W	Fix Control Timer Set Range : 0 to 59
0107H	2				
0108H	1	Autotune Heat/Cool	INT	R/W	0: Heat; 1: Cool
0109H	2				
010CH	1	AT Zone Tuning Enable	INT	R/W	0: Disable Automatic tuning 1: Enable Automatic tuning
010DH	2				
010FH	2				
0112H	1	Controller Only Yes/No	INT	R/W	0: Controller mode; 1: Programmer mode
0113H	2				
0114H	1	Program Link	INT	R/W	0: PATT; 1: LINK
0115H	2				
0116H	1	Program Number	INT	R/W	Set Range : 0 to 15 for Loop 1 and 16 to 31 for Loop2 at Async mode ; 0 to 31 at Sync mode
0117H	2				
0118H	1	Number of Links	INT	R/W	Set Range : 0 to 9
0119H	2				
011AH	1	Enable Schedule	INT	R/W	0: QUICK; 1: TIMER
011BH	2				
011CH	1	Schedule Start Month	INT	R/W	Set Range : 1 to 12
011DH	2				
011EH	1	Schedule Start Month	INT	R/W	Set Range : 1 to 31
011FH	2				

**Table 7-6.3 SP block register map address of 0200 to 031F hex range**

SP Block Address	Loop	Information	Data type	R/W	Remarks
0201H	1	Schedule Start Hour	INT	R/W	Set Range : 0 to 23
0202H	2				
0203H	1	Schedule Start Minute	INT	R/W	Set Range : 0 to 59
0204H	2				
0205H	1	Power Failure Mode	INT	R/W	0: BREAK; 1: COLD; 2: HOT
0206H	2				
020BH	Common	System RUN Lock	INT	R/W	0: UNLOCK; 1: LOCK
020FH	1	Fix G Soak	INT	R/W	0: NONE; 1: FRONT; 2: REAR; 3: ALL
0210H	2				
0211H	1	GSoak Enable for Value	INT	R/W	0: OFF; 1: ON
0212H	2				
0213H	1	GSoak Enable for Timeout	INT	R/W	0: OFF; 1: ON
0214H	2				
0215H	1	GSoak Timeout Upper Value	INT	R/W	Set Range : 00 to 99
0216H	2				
0217H	1	GSoak Timeout Lower Value	INT	R/W	Set Range : 00 to 59
0218H	2				
0219H	1	Manual mode status	INT	R	0: Manual OFF; 1: Manual ON
021AH	2				
021BH	1	Autotune status	INT	R	0: Autotune OFF; 1: Autotune ON
021CH	2				
0304H	Common	System time unit	INT	R/W	0: Hour and Minute; 1: Minute and Second
0306H		System Ramp mode	INT	R/W	0: Time Ramp; 1: Slope Ramp
0308H	1	PID Zone/Segment	INT	R/W	0: ZONE PID; 1: SEGMENT PID
0309H	2				
030AH	1	PID PV/SP Zone Select	INT	R/W	0: PV; 1: SP
030BH	2				
030EH	1	FUZZY Control Enable	INT	R/W	0: OFF; 1: ON
030FH	2				
0310H	1	Control Algorithm Select	INT	R/W	0: PID-A; 1:PID-B; 2: DUP-A; 3:DUP-B
0311H	2				
0312H	1	P Value Select	INT	R/W	0: PB; 1: GAIN
0313H	2				
0314H	1	Reset Value Select	INT	R/W	0: MIN; 1:RPT
0315H	2				
031BH	Common	System Control mode	INT	R/W	0: Asynchronous; 1: Synchronous
031CH	Common	SP Tracking	INT	R/W	0: OFF; 1: ON
031DH	Common	DO Channel Assign	INT	R/W	Set Range : 1 to 12 It is only available for Async mode.

**Table 7-6.4 SP block register map address of 0400 to 060F hex range**

SP Block Address	Loop	Information	Data type	R/W	Remarks
<b>040CH</b>	Common	Autotune Lock	INT	R/W	0: Unlock; 1: Lock
<b>0414H</b>	1	Square Root Enable	INT	R/W	0: OFF; 1: ON
<b>0415H</b>	2				
<b>041DH</b>	1	Compensate Select	INT	R/W	0: Linear; 1: Approx.
<b>041EH</b>	2				
<b>0500H</b>	1	Control Direction	INT	R/W	0: Reverse; 1: Direct
<b>0501H</b>	2				
<b>0502H</b>	1	Manual Output type	INT	R/W	0: Batch; 1: Bumpless
<b>0503H</b>	2				
<b>050AH</b>	Common	Trouble #1 Channel Assign	INT	R/W	0: Channel 1 1: Channel 2
<b>050BH</b>	Common	Trouble #2 Channel Assign	INT	R/W	0: Channel 1 1: Channel 2
<b>0510H</b>	Common	Event 1 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0511H</b>	Common	Event 2 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0512H</b>	Common	Event 3 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0513H</b>	Common	Event 4 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0514H</b>	Common	Event 5 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0515H</b>	Common	Event 6 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0516H</b>	Common	Event 7 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0517H</b>	Common	Event 8 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0518H</b>	Common	Event 9 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0519H</b>	Common	Event 10 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>051AH</b>	Common	Event 11 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>051BH</b>	Common	Event 12 Type	INT	R/W	Set Range : 21 to 40(= PV Event number)
<b>0600H</b>	1	Fix Timer hour	INT	R/W	Set Range : 00 to 99
<b>0601H</b>	2				
<b>0602H</b>	1	Fix Ramp Time high	INT	R/W	Set Range : 00 to 99
<b>0603H</b>	2				
<b>0604H</b>	1	Fix Ramp Time low	INT	R/W	Set Range : 00 to 59
<b>0605H</b>	2				
<b>0606H</b>	Common	Trouble message delay	INT	R/W	Set Range : 1 to 99 second
<b>060CH</b>	1	Failsafe	INT	R/W	Set Range : -5.0 to 105.0
<b>060DH</b>	2				
<b>060EH</b>	1	Manual Output Value	INT	R/W	Set Range : -5.0 to 105.0
<b>060FH</b>	2				

**Table 7-6.5 SP block register map address of 0800 to 0C07 and 1F0A hex range**

SP Block Address	Loop	Information	Data type	R/W	Remarks
<b>0800H</b>	1	Final SP Ramp Value	FP	R/W	Set Range : SP Low/High Limit
<b>0801H</b>	2				
<b>0802H</b>	1	Fix SP Ramp Value	FP	R/W	Set Range : 0.0 to 9.9
<b>0803H</b>	2				
<b>0902H</b>	1	GSoak Value	FP	R/W	Set Range : 0.0 to 9.9
<b>0903H</b>	2				
<b>0905H</b>	1	PV Low Range	FP	R/W	Set Range : PV Low < SP Low and PV Low > AI type low Range
<b>0906H</b>	2				
<b>0A00H</b>	1	PV High Range	FP	R/W	Set Range : PV High > SP High and PV High < AI type High Range
<b>0A01H</b>	2				
<b>0A03H</b>	1	SP Low Range	FP	R/W	Set Range : SP Low > PV Low and SP Low < SP High
<b>0A04H</b>	2				
<b>0A06H</b>	1	SP High Range	FP	R/W	Set Range : SP High < PV High and SP High > SP Low
<b>0A07H</b>	2				
<b>0B01H</b>	1	SP Offset	FP	R/W	Set Range : -99.9 to 99.9
<b>0B02H</b>	2				
<b>0B03H</b>	1	PV Offset	FP	R/W	Set Range : -99.9 to 99.9
<b>0B04H</b>	2				
<b>0B05H</b>	1	Digital Filter	FP	R/W	Set Range : 0.0 to 120.0 second
<b>0B06H</b>	2				
<b>0C04H</b>	1	Manual Output Preset Value	FP	R/W	Set Range : -5.0 to 105.0
<b>0C05H</b>	2				
<b>0C06H</b>	1	Low Cut	FP	R/W	Set Range : 0.1 to 5.0
<b>0C07H</b>	2				
<b>1F0AH</b>	Common	System Language Display	INT	R/W	0: English; 1: Korean; 2: Chinese

**Table 7-6.6 PV block register map address of 2800 to 2C03 hex range**

SP Block Address	Loop	Information	Data type	R/W	Remarks
2800H	1	Current Part Repeat Count	INT	R	
2801H	2				
2802H	1	Current All Repeat Count	INT	R	
2803H	2				
2804H	1	Total time Hour	INT	R	
2805H	2				
2806H	1	Total time Minute	INT	R	
2807H	2				
2808H	1	Current Segment remained time hour	INT	R	
2809H	2				
280AH	1	Current Segment remained time minute	INT	R	
280BH	2				
280CH	1	Current Link number	INT	R	
280DH	2				
2900H	1	Current running Program number	INT	R	
2901H	2				
2902H	Common	Single flag data I	INT	R	Bit 0~1: 00: Async mode; 01: Sync mode Bit 3 : 0: HH/MM; 1: MM/SS Bit 4 : 0: Time Ramp; 1: Slope Ramp Bit 5 ~ 15 : Don't care.
2903H	1	Single flag data II	INT	R	Bit 0~3 : Operation Mode(STOP, RUN..) <b>Note 1</b> Bit 4 : 0: Fix; 1: Program Bit 5 : 0: Pattern; 1: Link Bit 6 : 0: Auto; 1: Manual Bit 7~8 : Decimal Point Bit 9~10 : 00: Normal; 01: High over; 10: Low over; 11: Burn out Bit 11 : 0: Fix timer off; 1: Fix timer on Bit 12 : 0: Heat tune; 1: Cool tune Bit 13 : 0: Automatic Tune OFF; 1: Automatic Tune ON Bit 14 : 0: PID-A or B; 1: DUP-A or B
2904H	2		INT	R	
2905H	Common	Digital output status	INT	R	Bit 0: Digital output 1 Bit 1: Digital output 2 Bit 2: Digital output 3 Bit 3: Digital output 4 Bit 4: Digital output 5 Bit 5: Digital output 6 Bit 6: Digital output 7 Bit 7: Digital output 8 Bit 8: Digital output 9 Bit 9: Digital output 10 Bit 10: Digital output 11 Bit 11: Digital output 12 Bit 12~15: Don't care
2906H	1	Display MV output value	INT	R	
2907H	2				

SP Block Address	Loop	Information	Data type	R/W	Remarks
<b>290CH</b>	1	Edited Program number for Communication	INT	R/W	Set Range : 0 to 15 for Loop 1 and 16 to 31 for Loop2 at Async mode ; 0 to 31 at Sync mode
<b>290DH</b>	2				
<b>2A00H</b>	1	Working Set Point	FP	R	
<b>2A01H</b>	2				
<b>2A04H</b>	1	Deviation	FP	R	
<b>2A05H</b>	2				
<b>2B00H</b>	1	Processing Segment No.	INT	R	
<b>2B01H</b>	2				
<b>2B02H</b>	1	Next Segment No	INT	R	
<b>2B03H</b>	2				
<b>2B04H</b>	1	Previous Segment No	INT	R	
<b>2B05H</b>	2				
<b>2B06H</b>	1	Current PID group or zone number	INT	R	
<b>2B07H</b>	2				
<b>2B08H</b>	1	Operation status	INT	R	<i>Note 1</i>
<b>2B09H</b>	2				
<b>2B0AH</b>	1	Analog input status	INT	R	0: Normal; 1: High over; 2: Low over; 3: Input fail(Burn Out)
<b>2B0BH</b>	2				
<b>2B0CH</b>	1	Status for display	INT	R	<i>Note 2</i>
<b>2B0DH</b>	2				
<b>2C00H</b>	1	Heating Cycle time	INT	R	
<b>2C01H</b>	2				
<b>2C02H</b>	1	Cooling Cycle time	INT	R	
<b>2C03H</b>	2				
<b>290AH</b>	Common	IPC5000 Software version	INT	R	Example : version 1.5 -> 16

*Note 1* : 0: STOP; 1: RUN; 2:READY; 3:END; 4: BREAK; 5: TROUBLE; 6: HOLD; 7: WAIT; 8: TUNE

*Note 2* : 0: STOP; 1: RUN; 2:READY; 3:END; 4: BREAK; 5: TROUBLE; 6: HOLD; 7: WAIT; 8: TUNE; 9: COLD; 10: HOT;

**Table 7-6.7 PV block register map address of 3000 to 3205 hex range**

SP Block Address	Loop	Information	Data type	R/W	Remarks
<b>3000H</b>	1	Process Value(PV)	FP	R	
<b>3001H</b>	2				
<b>3002H</b>	AUX	Process Value(PV)	FP	R	
<b>3100H</b>	1	Next Segment Time	FP	R	
<b>3101H</b>	2				
<b>3102H</b>	1	Previous Segment Time	FP	R	
<b>3103H</b>	2				
<b>3200H</b>	1	Processing Segment Set Point	FP	R	
<b>3201H</b>	2				
<b>3202H</b>	1	Next Segment Set Point	FP	R	
<b>3203H</b>	2				
<b>3204H</b>	1	Previous Segment Set Point	FP	R	
<b>3205H</b>	2				

### 7.6.3 PID Constants Register Map

The parameters of this register can be seen in PID Constant set screen. The parameter numbers are 13. The register address range is 8000 to 8067 hex for Loop 1 and 8068 to 80CF for Loop 2. The group or zone number of the PID Constants is from 1 to 8.

**Table 7-6.8 PID Constant register map addresses**

Parameter Description	OFFSET	Data type	Access	Data range or Enumerated selection
	Hex			
Gain or PB (HEAT)	00	FP	R/W	Set Range : 0.01 to 1000 Gain 0.1 to 9999 PB
Gain or PB (COOL)	01	FP	R/W	
Reset (HEAT)	02	FP	R/W	Set Range : 0.02 to 50.00
Reset (COOL)	03	FP	R/W	
Rate (HEAT)	04	FP	R/W	Set Range : 0.00 to 10.00
Rate (COOL)	05	FP	R/W	
Differential (HEAT)	06	FP	R/W	Differential value Set Range : 0.0 to 100.0 % of span
Differential (COOL)	07	FP	R/W	
Manual Reset	08	FP	R/W	Set Range : -100 to 100
Dead band	09	FP	R/W	Set Range : -5.0 to 25.0 %
Output high limit	0A	FP	R/W	Set Range : -5.0 to 100.0 %
Output low limit	0B	FP	R/W	Set Range : -5.0 to 100.0 %
Auto tuning setpoint	0C	FP	R/W	Set Range : Dependant on SP limit range

#### How to find the Register address of each parameter registered to PID Constants of Loop 1 and 2:

Loop1 Register Address = 8000H + (PID group or Zone No. - 1) \* 13 + Offset Value of the parameter from Table 7-6.8

Loop2 Register Address = 8068H + (PID group or Zone No. - 1) \* 13 + Offset Value of the parameter from Table 7-6.8

## 7.6.4 PID Reference Register address

The parameters of this register can be seen in PID Zone set screen.

**Table 7-6.9 PID Reference register map addresses for Loop 1**

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
Reference value #1	80DD	FP	R/W	PID zone reference value Set Range : Value #1 > PV low limit
Reference value #2	80DE	FP	R/W	PID zone reference value Set Range : Value #1 < Value #2 < Value #3
Reference value #3	80DF	FP	R/W	PID zone reference value Set Range : Value #2 < Value #3 < Value #4
Reference value #4	80E0	FP	R/W	PID zone reference value Set Range : Value #3 < Value #4 < Value #5
Reference value #5	80E1	FP	R/W	PID zone reference value Set Range : Value #4 < Value #5 < Value #6
Reference value #6	80E2	FP	R/W	PID zone reference value Set Range : Value #5 < Value #6 < Value #7
Reference value #7	80E3	FP	R/W	PID zone reference value Set Range : Value #6 < Value #7 <= PV High limit

**Table 7-6.10 PID Reference register map addresses for Loop 2**

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
Reference value #1	80E4	FP	R/W	PID zone reference value Set Range : Value #1 > PV low limit
Reference value #2	80E5	FP	R/W	PID zone reference value Set Range : Value #1 < Value #2 < Value #3
Reference value #3	80E6	FP	R/W	PID zone reference value Set Range : Value #2 < Value #3 < Value #4
Reference value #4	80E7	FP	R/W	PID zone reference value Set Range : Value #3 < Value #4 < Value #5
Reference value #5	80E8	FP	R/W	PID zone reference value Set Range : Value #4 < Value #5 < Value #6
Reference value #6	80E9	FP	R/W	PID zone reference value Set Range : Value #5 < Value #6 < Value #7
Reference value #7	80EA	FP	R/W	PID zone reference value Set Range : Value #6 < Value #7 <= PV High limit

## 7.6.5 Compensate set Register Address

The parameters of this register can be seen in Compensate set screen.

**Table 7-6.11 Compensate set register map addresses for Loop 1**

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
CP set value #1	8100	FP	R/W	Compensation 1st value Set Range : Depend on PV limit
Bias value #1	8101	FP	R/W	Bias 1st value Set Range : Depend on PV limit
CP set value #2	8102	FP	R/W	Compensation 2nd value Set Range : Depend on PV limit
Bias value #2	8103	FP	R/W	Bias 2nd value Set Range : Depend on PV limit
CP set value #3	8104	FP	R/W	Compensation 3rd value Set Range : Depend on PV limit
Bias value #3	8105	FP	R/W	Bias 3rd value Set Range : Depend on PV limit
CP set value #4	8106	FP	R/W	Compensation 4th value Set Range : Depend on PV limit
Bias value #4	8107	FP	R/W	Bias 4th value Set Range : Depend on PV limit
CP set value #5	8108	FP	R/W	Compensation 5th value Set Range : Depend on PV limit
Bias value #5	8109	FP	R/W	Bias 5th value Set Range : Depend on PV limit
CP set value #6	810A	FP	R/W	Compensation 6th value Set Range : Depend on PV limit
Bias value #6	810B	FP	R/W	Bias 6th value Set Range : Depend on PV limit
CP set value #7	810C	FP	R/W	Compensation 7th value Set Range : Depend on PV limit
Bias value #7	810D	FP	R/W	Bias 7th value Set Range : Depend on PV limit
CP set value #8	810E	FP	R/W	Compensation 8th value Set Range : Depend on PV limit
Bias value #8	810F	FP	R/W	Bias 8th value Set Range : Depend on PV limit
CP set value #9	8110	FP	R/W	Compensation 9th value Set Range : Depend on PV limit
Bias value #9	8111	FP	R/W	Bias 9th value Set Range : Depend on PV limit
CP set value #10	8112	FP	R/W	Compensation 10th value Set Range : Depend on PV limit
Bias value #10	8113	FP	R/W	Bias 10th value Set Range : Depend on PV limit

**Table 7-6.12 Compensate set register map addresses for Loop 2**

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
CP set value #1	8114	FP	R/W	Compensation 1st value Set Range : Depend on PV limit
Bias value #1	8115	FP	R/W	Bias 1st value Set Range : Depend on PV limit
CP set value #2	8116	FP	R/W	Compensation 2nd value Set Range : Depend on PV limit
Bias value #2	8117	FP	R/W	Bias 2nd value Set Range : Depend on PV limit
CP set value #3	8118	FP	R/W	Compensation 3rd value Set Range : Depend on PV limit
Bias value #3	8119	FP	R/W	Bias 3rd value Set Range : Depend on PV limit
CP set value #4	811A	FP	R/W	Compensation 4th value Set Range : Depend on PV limit
Bias value #4	811B	FP	R/W	Bias 4th value Set Range : Depend on PV limit
CP set value #5	811C	FP	R/W	Compensation 5th value Set Range : Depend on PV limit
Bias value #5	811D	FP	R/W	Bias 5th value Set Range : Depend on PV limit
CP set value #6	811E	FP	R/W	Compensation 6th value Set Range : Depend on PV limit
Bias value #6	811F	FP	R/W	Bias 6th value Set Range : Depend on PV limit
CP set value #7	8120	FP	R/W	Compensation 7th value Set Range : Depend on PV limit
Bias value #7	8121	FP	R/W	Bias 7th value Set Range : Depend on PV limit
CP set value #8	8122	FP	R/W	Compensation 8th value Set Range : Depend on PV limit
Bias value #8	8123	FP	R/W	Bias 8th value Set Range : Depend on PV limit
CP set value #9	8124	FP	R/W	Compensation 9th value Set Range : Depend on PV limit
Bias value #9	8125	FP	R/W	Bias 9th value Set Range : Depend on PV limit
CP set value #10	8126	FP	R/W	Compensation 10th value Set Range : Depend on PV limit
Bias value #10	8127	FP	R/W	Bias 10th value Set Range : Depend on PV limit

## 7.6.6 Fix PV Event Register Address

The parameters of these registers can be seen in Fix Set screen. The address range for Signed 16 bit integer is from 4200 Hex to 4216 Hex and For Floating Point is from 9000 Hex to 9017 Hex.

Table 7-6.13 Fix PV Event Register Map addresses

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
Event type #1	4200	INT	R/W	<b>Note 1</b> (Page 96)
Time Delay #1	4201	INT	R/W	0 ~ 99 Second
Max value or OP Point #1	9000	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #1	9001	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #2	4202	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #2	4203	INT	R/W	0 ~ 99 Second
Max value or OP Point #2	9002	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #2	9003	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #3	4204	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #3	4205	INT	R/W	0 ~ 99 Second
Max value or OP Point #3	9004	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #3	9005	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #4	4206	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #4	4207	INT	R/W	0 ~ 99 Second
Max value or OP Point #4	9006	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #4	9007	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #5	4208	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #5	4209	INT	R/W	0 ~ 99 Second
Max value or OP Point #5	9008	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #5	9009	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #6	420A	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #6	420B	INT	R/W	0 ~ 99 Second
Max value or OP Point #6	900A	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #6	900B	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #7	420C	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #7	420D	INT	R/W	0 ~ 99 Second
Max value or OP Point #7	900C	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #7	900D	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0

Parameter Description	Register Address	Data type	Access	Data range or Enumerated selection
	Hex			
Event type #8	420E	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #8	420F	INT	R/W	0 ~ 99 Second
Max value or OP Point #8	900E	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #8	900F	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #9	4210	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #9	4211	INT	R/W	0 ~ 99 Second
Max value or OP Point #9	9010	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #9	9011	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #10	4212	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #10	4213	INT	R/W	0 ~ 99 Second
Max value or OP Point #10	9012	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #10	9013	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #11	4214	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #11	4215	INT	R/W	0 ~ 99 Second
Max value or OP Point #11	9014	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #11	9015	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0
Event type #12	4216	INT	R/W	<b>Note 1</b> (Page 96)
Delay Time #12	4217	INT	R/W	0 ~ 99 Second
Max value or OP Point #12	9016	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential #12	9017	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0

**NOTE 1**

- 11: PV-ABS-LOW for Channel 1    12: PV-ABS-LOW for Channel 2  
13: PV-ABS-HIGH for Channel 1    14: PV-ABS-HIGH for Channel 2  
15: PV-DEV-LOW for Channel 1    16: PV- DEV-LOW for Channel 2  
17: PV-DEV-HIGH for Channel 1    18: PV-DEV-HIGH for Channel 2  
21: SP-ABS-LOW for Channel 1    22: SP-ABS-LOW for Channel 2  
23: SP-ABS-HIGH for Channel 1    24: SP-ABS-HIGH for Channel 2  
25: SP-DEV-OFF for Channel 1    26: SP-DEV-OFF for Channel 2  
27: SP-DEV-ON for Channel 1    28: SP-DEV-ON for Channel 2  
31: DV-DEV-OFF for Channel 1    32: DV-DEV-OFF for Channel 2  
33: DV-DEV-ON for Channel 1    34: DV-DEV-ON for Channel 2  
35: MV-ABS-LOW for Channel 1    36: MV-ABS-LOW for Channel 2  
37: MV-ABS-HIGH for Channel 1    38: MV-ABS-HIGH for Channel 2

## 7.6.7 Operation button action Register address

The operation buttons of these parameters can be seen in Monitor and MON SUB screen.

**Table 7-6.14 Operation button action Register Map addresses**

Parameter Description	Register Address	Data type	Access	Loop	Data range or Enumerated selection
	Hex				
RUN action #1	4000	INT	W	1	1 : Run Enable 0 : Data ignored <i>Note 1, Note 2</i>
RUN action #2	4001	INT	W	2	
STOP action #1	4002	INT	W	1	1 : Stop Enable 0 : Data ignored <i>Note 1, Note 2</i>
STOP action #2	4003	INT	W	2	
HOLD action #1	4004	INT	W	1	1 : Hold Enable 0 : Hold Disable <i>Note 1, Note 2</i>
HOLD action #2	4005	INT	W	2	
ADV action #1	4006	INT	W	1	1 : Advance Enable 0 : Data ignored <i>Note 1, Note 3</i>
ADV action #2	4007	INT	W	2	
Manual action #1	4008	INT	W	1	1: Manual Enable 0: Manual Disable
Manual action #2	4009	INT	W	2	
AT action #1	400A	INT	W	1	1: Autotune Enable 2: Autotune Disable
AT action #2	400B	INT	W	2	
Segment Edit #1	400C	INT	W	1	0 to 7 Bit is Segment number If 8 to 15 Bit is zero, inserting the segment. If 8 to 15 bit is one, deleting the segment. Before Segment edition, write edited program number to register 290C hex in order to insert or delete the segments registered to the program.
Segment Edit #2	400D	INT	W	2	
Power fail message clear #1	400E	INT	W	1	Returns current BREAK, COLD or HOT display to normal operation(RUN,STOP,..) display.
Power fail message Clear #2	400F	INT	W	2	

**Note 1 :** These registers are available for Programmer mode as well as Controller mode(Fix control = Constant control)

**Note 2 :**

Current Status	Command		
	Run	Hold	Stop
Run	Set error	Hold	Stop
Hold	Run	Set error	Stop
Stop	Run	Set error	Set error

**Note 3 :** Current segment move to next segment per one command.

## 7.6.8 Program Data Register address

The parameters of this register can be seen in Program and Segment edit screen and the file number of this register is 4.

These register addresses are used for reading and writing a program data or segment data register to a program.

Please set file number as 4 to read and write some data of these registers. The below table 7-6.15, 7-6.16 and 7-6.17 show the start and end address of each program of Integer and floating point type. The table 7-6.18, 7-6.19 and 7-6.20 show the detail information and Offset value of each data type and the register address of a parameter can be found from the tables.

**Table 7-6.15 Program register address map(Program data) of integer type**

Channel No.	Start address (Hex)	Start address (Decimal)	End address (Hex)	End address (Decimal)
1	0000	0000	0012	0018
2	0013	0019	0025	0037

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
Start Mode	0000	INT	R/W	0: SSP; 1: SPV; 2:TIME
Priority	0001	INT	R/W	0: PV1; 1: PV2. It is not available at Async. mode.
Segment Count	0002	INT	R	Segment number registered to a program One program can have 100 segments.
All Repeat count	0003	INT	R/W	Set Range : 1 to 999. It is the number to repeat one program.
#1 End Segment	0004	INT	R/W	Set Range : Within Segment Count
#1 Start Segment	0005	INT	R/W	End Segment for repeat have to be larger than Start Segment.
#1 Repeat count	0006	INT	R/W	Repeat count set Range : 1 to 999
#2 End Segment	0007	INT	R/W	
#2 Start Segment	0008	INT	R/W	
#2 Repeat count	0009	INT	R/W	
#3 End Segment	000A	INT	R/W	
#3 Start Segment	000B	INT	R/W	
#3 Repeat count	000C	INT	R/W	
#4 End Segment	000D	INT	R/W	
#4 Start Segment	000E	INT	R/W	
#4 Repeat count	000F	INT	R/W	
#5 End Segment	0010	INT	R/W	
#5 Start Segment	0011	INT	R/W	
#5 Repeat count	0012	INT	R/W	

**Table 7-6.16 Program register address map(Segment data) of Integer type**

Channel number	Segment number	Start address (Hex)	Start address (Decimal)	End address (Hex)	End address (Decimal)
1	0	0026	38	002A	42
	1	002B	43	002F	47
	2	0030	48	0034	52
	3	0035	53	0039	57
	4	003A	58	003E	62
	5	003F	63	0043	67
	:	:	:	:	:
	99	0215	533	0219	537
2	0	021A	538	021E	542
	1	021F	543	0223	547
	2	0224	548	0228	552
	3	0229	553	022D	557
	4	022E	558	0232	562
	5	0233	563	0237	567
	:	:	:	:	:
	99	0409	1033	040D	1037

**Table 7-6.17 Program register address map(Segment Event data) of Integer type**

Channel number	Segment number	Start address (Hex)	Start address (Decimal)	End address (Hex)	End address (Decimal)
1	0	040E	1038	0431	1073
	1	0432	1074	0455	1109
	2	0456	1110	0479	1145
	3	047A	1146	049D	1181
	4	049E	1182	04C1	1217
	5	04C2	1218	04E5	1253
	:	:	:	:	:
	99	11FA	4602	121D	4637
2	0	121E	4638	1241	4673
	1	1242	4674	1265	4709
	3	1266	4710	1289	4745
	4	128A	4746	12AD	4781
	5	12AE	4782	12D1	4817
	6	12D2	4818	12F5	4853
	:	:	:	:	:
	99	200A	8202	202D	8237

**Table 7-6.18 Program register address map of floating point for SP1&2, Segment time**

Channel number	Segment number	Start address (Hex)	Start address (Decimal)	End address (Hex)	End address (Decimal)
1	0	3000	12288	3002	12290
	1	3003	12291	3005	12293
	2	3006	12294	3008	12296
	3	3009	12297	300B	12299
	4	300C	12300	300E	12302
	5	300F	12303	3011	12305
	:	:	:	:	:
	99	3129	12585	312B	12587
2	0	312C	12588	12590	312E
	1	312F	12591	12593	3131
	2	3132	12594	12596	3134
	3	3135	12597	12599	3137
	4	3138	12600	12602	313A
	5	313B	12603	12605	313D
	:	:	:	:	:
	99	3255	12885	3257	12887

**Table 7-6.19 Program register address map of floating point for Start SP1&2**

Channel number	Segment number	Start address (Hex)	Start address (Decimal)	End address (Hex)
0	3300	13056	3301	13057
1	3302	13058	3303	13059

**Table 7-6.20 Program register address map of floating point for Max/OP point and Min/Diff**

Channel number	Segment number	Start address (Hex)	Start address (Decimal)	End address (Hex)	Channel number
1	0	3400	13312	13335	3417
	1	3418	13336	13359	342F
	2	3430	13360	13383	3447
	3	3448	13384	13407	345F
	4	3460	13408	13431	3477
	5	3478	13432	13455	348F
	:	:	:	:	:
	99	3D48	15688	3D5F	15711
2	0	3D60	15712	3D77	15735
	1	3D78	15736	3D8F	15759
	2	3D90	15760	3DA7	15783
	3	3DA8	15784	3DBF	15807
	4	3DC0	15808	3DD7	15831
	5	3DD8	15832	3DEF	15855
	:	:	:	:	:
	99	46A8	18088	46BF	18111

**Table 7-6.21 Segment data register offset of Integer type**

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
PID Group number 1	<b>0000</b>	INT	R/W	Set Range : 1 to 8; It is only available for Segment PID.
PID Group number 2	<b>0001</b>	INT	R/W	
Program End	<b>0002</b>	INT	R/W	0 : Continuous Shifting ; 1 : HOLD Shifting; 2 : Fix Control Shifting
Segment G.Soak	<b>0003</b>	INT	R/W	0 : NONE; 1 : FRONT; 2 : REAR; 3 : ALL
Time or Rate	<b>0004</b>	INT	R	0 : Time; 1: Rate

### How to find the Segment Register address registered to each program

**Example :**

Find the register address of Program End of Segment #4 registered to Program #3.

(Note : Edited program register(290C Hex) have to be set as 3)

Step 1 : Use Table 7-6.16 to determine the start address of segment register. The address is 0026H.

One segment register count is 05Hex from Table 7-6.21.

Step 2 : Register address = start address of segment register + (Segment number \* 05H)

+ Program End offset from Table 7-6.21

$$= 0026H + 4 * 5 + 2 = 003CH$$

**\* One program can contain up to 100 segments but actually the number of segments registered to the program can be known to segment count**

**Table 7-6.22 Segment Event data register offset of Integer type**

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
Event type #1	0000	INT	R/W	<b>Note 1</b>
Delay Time #1	0001	INT	R/W	0 ~ 99 second
Selector #1	0002	INT	R	

**NOTE 1**

0 : OFF 1 : ON(Time Event) 2 : On Delay and Cut Back setup(Time Event)

- 11: PV-ABS-LOW for Channel 1(PV Event)    12: PV-ABS-LOW for Channel 2(PV Event)
- 13: PV-ABS-HIGH for Channel 1(PV Event)    14: PV-ABS-HIGH for Channel 2(PV Event)
- 15: PV-DEV-LOW for Channel 1(PV Event)    16: PV- DEV-LOW for Channel 2(PV Event)
- 17: PV-DEV-HIGH for Channel 1(PV Event)    18: PV-DEV-HIGH for Channel 2(PV Event)
- 21: SP-ABS-LOW for Channel 1(PV Event)    22: SP-ABS-LOW for Channel 2(PV Event)
- 23: SP-ABS-HIGH for Channel 1(PV Event)    24: SP-ABS-HIGH for Channel 2(PV Event)
- 25: SP-DEV-OFF for Channel 1(PV Event)    26: SP-DEV-OFF for Channel 2(PV Event)
- 27: SP-DEV-ON for Channel 1(PV Event)    28: SP-DEV-ON for Channel 2(PV Event)
- 31: DV-DEV-OFF for Channel 1(PV Event)    32: DV-DEV-OFF for Channel 2(PV Event)
- 33: DV-DEV-ON for Channel 1(PV Event)    34: DV-DEV-ON for Channel 2(PV Event)
- 35: MV-ABS-LOW for Channel 1(PV Event)    36: MV-ABS-LOW for Channel 2(PV Event)
- 37: MV-ABS-HIGH for Channel 1 (PV Event)    38: MV-ABS-HIGH for Channel 2(PV Event)

**How to find the Event register address of segment registered to each program**

**Example :**

Find the register address of Event type and Delay time set for Digital output #3 of Segment #4 registered to Program #3 of Channel 1.(Note : Edited program register(290C Hex) have to be set as 3)

Step 1 : Use Table 7-6.17 to determine the start address of segment Event register. The address is 040EH.

One segment Event register count is 0CHex

Step 2 : Register address = start address of segment Event register + (Segment number \* (0CH\*03H)) + (Digital output number – 1) \* 3H + offset of Event type from Table 7-6.22.

$$= 040EH + 4 * 24H + 2*3H + 0 = 04A4H$$

Register address = start address of segment Event register + (Segment number \* (0CH\*03H)) + (Digital output number – 1) \* 3H + offset of Delay time from Table 7-6.22.

$$= 040EH + 4H * 24H + 2*3H + 1 = 04A5H$$

**Table 7-6.23 Program register address of floating point(SSP1&2)**

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
#1 Start Setpoint 1(SSP1)	3300	FP	R/W	Set Range : SP Low/High Limit
#1 Start Setpoint 2(SSP2)	3301	FP	R/W	SSP2 is not available at Async. mode
#2 Start Setpoint 1(SSP1)	3302	FP	R/W	Set Range : SP Low/High Limit
#2 Start Setpoint 2(SSP2)	3303	FP	R/W	SSP2 is not available at Async. mode

**Table 7-6.24 Segment register offset of floating point(SP1&2 and Time)**

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
Target Set Point 1	0000	FP	R/W	Set Range : SP Low/High range
Target Set Point 2	0001	FP	R/W	At Sync mode, it is Target Setpoint of Loop 2 and not available for Async Set Range : SP Low/High range
Segment time	0002	FP	R/W	Set Range : 00.00 to 99.59

## How to find the SP1, SP2 and Time address of segment registered to each program

### Example :

Find the register address of SP1, SP2 and Segment time of Segment #4 registered to Program #3 of Channel 1.

(Note : Edited program register(290C Hex) have to be set as 3)

Step 1 : Use Table 7-6.18 to determine the start address of the segment register. The address is 300CH.

$$\begin{aligned} \text{Start address of Segment register \#4} &= \text{Segment number} * 3\text{H} + 3000\text{H} \\ &= 4 * 3\text{H} + 3000\text{H} = 300\text{CH} \end{aligned}$$

Step 2 :

$$\begin{aligned} \text{SP1 register address} &= \text{start address of segment number} + 0\text{H} \\ &= 300\text{CH} + 0\text{H} = 300\text{CH} \end{aligned}$$

$$\begin{aligned} \text{SP2 register address} &= \text{start address of segment number} + 1\text{H} \\ &= 300\text{CH} + 1\text{H} = 300\text{DH} \end{aligned}$$

$$\begin{aligned} \text{Segment time register address} &= \text{start address of segment number} + 2\text{H} \\ &= 300\text{CH} + 2\text{H} = 300\text{EH} \end{aligned}$$

**Table 7-6.25 Segment event register offset of floating point(Max/OP point or Min/Differential)**

Parameter Description	Offset	Data Type	Access	Data range or Enumerated selection
	Hex			
Max value or OP Point	0000	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential	0001	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0

**How to find the Max/Op point or Min/Diff address of segment registered to each program**

**Example :**

Find the register address of Max/OP Point and Min/Differential for Digital output #2 of Segment #4 registered to Program #3 of Channel 1.

(Note : Edited program register(290C Hex) have to be set as 3)

Step 1 : Use Table 7-6.20 to determine the start address of the segment register. The address is 3460H.

$$\begin{aligned} \text{Start address of Segment register \#4} &= \text{Segment number} * 18\text{H} + 3400\text{H} \\ &= 4 * 18\text{H} + 3400\text{H} = 3460\text{H} \end{aligned}$$

Step 2 :

$$\begin{aligned} \text{Max/OP Point register address} &= \text{start address of segment number} + (\text{Digital output number} - 1) * 2\text{H} + 0\text{H} \\ &\text{from Table 7-6.25} \\ &= 3460\text{H} + (2-1) * 2\text{H} + 0\text{H} = 3462\text{H} \end{aligned}$$

$$\begin{aligned} \text{Min/Differential register address} &= \text{start address of segment number} + (\text{Digital output number} - 1) * 2\text{H} + 1\text{H} \\ &\text{from Table 7-6.25} \\ &= 3460\text{H} + (2-1) * 2\text{H} + 1\text{H} = 3463\text{H} \end{aligned}$$

## 7.6.9 Link Set Register address

A profile contains up to 10 links. Each link is made up of 6 registers. This table contains Link Map Address of Link group #1 and #2.

**Table 7-6.26 Each Link group set register Map Addresses**

Start Address (Hex)	End Address (Hex)	Description
4110	4115	Link group #1 of Loop1
4116	411B	Link group #2 of Loop1
411C	4121	Link group #3 of Loop1
4122	4127	Link group #4 of Loop1
:	:	:
4146	414B	Link group #10 of Loop1
414C	4151	Link group #1 of Loop2
4152	4157	Link group #2 of Loop2
4158	415D	Link group #3 of Loop2
415E	4163	Link group #4 of Loop2
:	:	:
4183	4188	Link group #10 of Loop2

The table below describes the registers that are part of a Link group Register. To determine the actual register address for a parameter within a Link, add the register offset to the start address of the Link.

**Table 7-6.27 Link register Map Addresses**

Register Offset within Link group	Parameter Name	Access	Notes
0	Link Program number 1	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
1	Link Program number 2	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
2	Link Program number 3	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
3	Link Program number 4	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
4	Link Program number 5	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>
5	Link Program number 6	R/W	Signed 16 bit integer, <i>Note 1, Note 2</i>

**Note 1 :** Set Range : 0 to 15 for Loop 1 and 16 to 31 for Loop 2 at Async mode

0 to 31 for Loop1/2 at Sync mode

**Note 2 :** Write 255(FF hex) to the register address to delete the registered program number.

---

**How to find the Register address of program number registered to Link group of Loop 1**

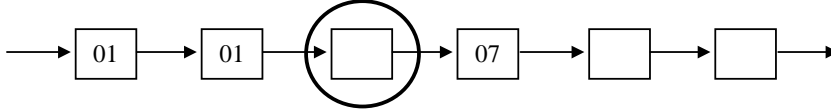
Register address = Start address of Link group #1 of Loop 1+ (Link group number - 1) \* 6 + Offset value of the parameter from Table 7-6.27.

**Example :** Find the register address of the third Link program number of Link group #4

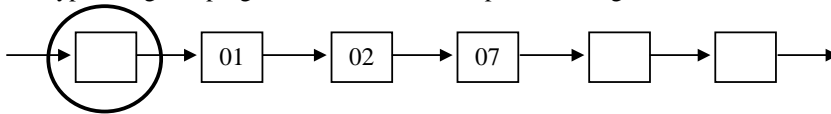
The Register address = 4110 hex + (4-1) \* 6 + 02 = 4124 hex

**< Attention >** Program number has to be registered in sequence.

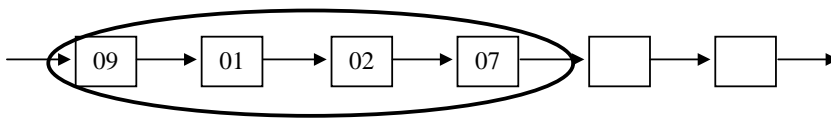
1) Incorrect type to register program number : The third part is not registered.



2) Incorrect type to register program number : The first part is not registered.



3) Correct type : Program number is registered in sequence.



## 7.6.10 Analog input group Register address(5000H ~ 5013H)

Table 7-6.28 Analog input group register address Map

Parameter Description	Register Address	Data Type	Access	Data range or Enumerated selection
	Hex			
# 1 Analog input type	5000	INT	R/W	<i>Note 1</i>
# 1 Analog input unit	5001	INT	R/W	0: Degree C; 1: Degree F; 2: None
Not Used	5002	INT		
#1 Analog input low scale	5003	INT	R/W	Floating point is dependent on decimal point <i>Note 2</i>
#1 Analog input high scale	5004	INT	R/W	Floating point is dependent on decimal point <i>Note 2</i>
Not Used	5005	INT		
Not Used	5006	INT		
# 1 Analog input decimal point	5007	INT	R/W	0: None; 1: First; 2: Second; 3: Third Decimal 3 is only available for Linear input type.
Not Used	5008	INT		
Not Used	5009	INT		
# 2 Analog input type	500A	INT	R/W	<i>Note 1</i>
# 2 Analog input unit	500B	INT	R/W	
Not Used	500C	INT		
# 2 Analog input low scale	500D	INT		Floating point is dependent on decimal point <i>Note 2</i>
# 2 Analog input high scale	500E	INT		Floating point is dependent on decimal point <i>Note 2</i>
Not Used	500F	INT		
Not Used	5010	INT		
# 2 Analog input decimal point	5011	INT	R/W	
Not Used	5012	INT		
Not Used	5013	INT		

**Note 1** : 0 : Volt 0 ~ 10V; 1 : Volt 0 ~ 5V; 2 : Volt 1 ~ 5V; 3 : AMP 0 ~ 20mA; 4 : AMP 4 ~ 20mA; 5 : Pt1 -200 ~ 500;  
6 : Pt2 -200 ~ 200; 7 : Pt3 -100 ~ 150; 8 : Pt4 -50 ~ 200; 9 : Pt5 -40 ~ 60; 10 : Pt6 0 ~ 100; 11 : Pt7 0 ~ 300;  
12 : Pt8 0 ~ 500; 13 : JPt1 -200 ~ 500; 14 : JPt2 -200 ~ 200; 15 : JPt3 -100 ~ 150; 16 : JPt4 -50 ~ 200;  
17 : JPt5 -40 ~ 60; 18 : JPt6 0 ~ 100; 19 : JPt7 0 ~ 300; 20 : JPt8 0 ~ 500; 21 : K1 -200 ~ 200; 22 : K2 0 ~ 1200;  
23 : K3 0 ~ 800; 24 : K4 0 ~ 400; 25 : J(IC) 0 ~ 800; 26 : R 0 ~ 1600; 27 : S 0 ~ 1600; 28 : B 0 ~ 1800;  
29 : E 0 ~ 800; 30 : T -200 ~ 300; 31 : N 0 ~ 1300;

**Note 2**

In case that decimal point is second, Analog input scale is integer. For example, the expression of 100.00 is 10000(2710H).

## 7.6.11 Analog output group Register address(5020H ~ 503BH)

Table 7-6.29 Analog output group register address Map

Parameter Description	Register Address	Data Type	Access	Data range or Enumerated selection
	Hex			
# 1 Analog output type	5020	INT	R/W	0: 4 ~ 20mA; 1: PWM
# 1 Analog output source	5021	INT	R/W	When type is 4 ~ 20mA output 0: MV1; 1: MV2; 2: MV3; 3: MV4; 4: PV1; 5: PV2; 6: SP1; 7: SP2; 8: DV1; 9:DV2; When type is PWM output 10: PW1; 11: PW2; 12: PW3; 13: PW4
# 1 Analog output source low range	5022	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
# 1 Analog output source high range	5023	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
Not Used	5024	INT		
Not Used	5025	INT		
Not Used	5026	INT		
# 2 Analog output type	5027	INT	R/W	0: 4 ~ 20mA; 1: PWM
# 2 Analog output source	5028	INT	R/W	When type is 4 ~ 20mA output 0: MV1; 1: MV2; 2: MV3; 3: MV4; 4: PV1; 5: PV2; 6: SP1; 7: SP2; 8: DV1; 9:DV2; When type is PWM output 10: PW1; 11: PW2; 12: PW3; 13: PW4
# 2 Analog output source low range	5029	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
# 2 Analog output source high range	502A	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
Not Used	502B	INT		
Not Used	502C	INT		
Not Used	502D	INT		
# 3 Analog output type	502E		R/W	0: 4 ~ 20mA fixed. Don't try to write
# 3 Analog output source	502F	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4; 4: PV1; 5: PV2; 6: SP1; 7: SP2; 8: DV1; 9:DV2;
# 3 Analog output source low range	5030	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
# 3 Analog output source high range	5031	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
Not Used	5032	INT		
Not Used	5033	INT		
Not Used	5034	INT		
# 4 Analog output type	5035	INT	R/W	0: 4 ~ 20mA fixed. Don't try to write
# 4 Analog output source	5036	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4; 4: PV1; 5: PV2; 6: SP1; 7: SP2; 8: DV1; 9:DV2;
# 4 Analog output source low range	5037	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
# 4 Analog output source high range	5038	INT	R/W	First decimal point is fixed 100.0 -> 1000(03E8H)
Not Used	5039	INT		
Not Used	503A	INT		
Not Used	503B	INT		

**Note** : #3 and #4 Analog output is option. In only case of option model that include #3 and #4 analog output, it is available.

## 7.6.12 Analog PWM group Register address(5040H ~ 504BH)

Table 7-6.30 Analog PWM group register address Map

Parameter Description	Register Address	Data Type	Access	Data range or Enumerated selection
	Hex			
Not Used	5040	INT		
# 1 Analog PWM source	5041	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4
# 1 Analog Cycle time	5042	INT	R/W	Set Range : 1 to 99
Not Used	5043	INT		
# 2 Analog PWM source	5044	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4
# 2 Analog Cycle time	5045	INT	R/W	Set Range : 1 to 99
Not Used	5046	INT		
# 3 Analog PWM source	5047	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4
# 3 Analog Cycle time	5048	INT	R/W	Set Range : 1 to 99
Not Used	5049	INT		
# 4 Analog PWM source	504A	INT	R/W	0: MV1; 1: MV2; 2: MV3; 3: MV4
# 4 Analog Cycle time	504B	INT	R/W	Set Range : 1 to 99

## 7.6.13 Digital output group Register address(5050H ~ 5057H)

Table 7-6.31 Digital output group register address Map

Parameter Description	Register Address	Data Type	Access	Data range or Enumerated selection
	Hex			
Digital output 1 type	5050	INT	R/W	Open Collector output
Digital output 1 PWN number	5051	INT	R/W	Output type : 0: Normal; 1: PWM PWM number : 0: PW1; 1: PW2; 2: PW3; 3: PW4
Digital output 2 type	5052	INT	R/W	Open Collector output
Digital output 2 PWN number	5053	INT	R/W	Output type : 0: Normal; 1: PWM PWM number : 0: PW1; 1: PW2; 2: PW3; 3: PW4
Digital output 13 type	5054	INT	R/W	Relay output
Digital output 13 PWN number	5055	INT	R/W	Output type : 0: Normal; 1: PWM PWM number : 0: PW1; 1: PW2; 2: PW3; 3: PW4
Digital output 14 type	5056	INT	R/W	Relay output
Digital output 14 PWN number	5057	INT	R/W	Output type : 0: Normal; 1: PWM PWM number : 0: PW1; 1: PW2; 2: PW3; 3: PW4

## 7.6.14 Mode Event group Register address(4300H ~ 434FH)

The address range of this register is from 4300H to 434FH.

**Table 7-6.32 Mode Event group register offset**

Parameter Description	OFFSET	Data Type	Access	Data range or Enumerated selection	
	Hex				
Mode type	00	INT	R/W	Mode	Value
				STOP	0
				RUN	1
				READY	2
				END	3
				TROUBLE	5
				HOLD	6
				WAIT	7
				TUNE	8
				MAN	10
				FIX	11
				DOWN	12
				UP	13
Assign for channel	01	INT	R/W	CH1(0) / CH2(1) / BOTH(2)	
Condition	02	INT	R/W	OR(0) / AND(1)	
Registration number (Digital output number)	03	INT	R/W	0 to 12, 0: No registration	

### How to find the Register address according to Mode Event number.

Register Address = 4300H + (Event number - 41) \* 04H + OFFSET value from Table 7-6.32

#### Example 1 :

Find the register address of condition parameter of Mode Event #44.

Register Address = 4300H + (44-41) \* 04H + 02H = 430EH

#### Example 2:

Find the register address of registration number of Mode Event #58.

Register Address = 4300H + (58-41) \* 04H + 03H = 4347H

## 7.6.16 Alarm Event group Register address

The address range for Integer type is from 4400H to 449FH and for Floating point is from 905CH to 9083H.

**Table 7-6.33 Alarm Event group register offset**

Parameter Description	OFFSET	Data Type	Access	Data range or Enumerated selection
	Hex			
Not Used	00	INT		
Alarm Type	01	INT	R/W	0: INNER; 1: Diagnos; 2: FAIL
Alarm Action	02	INT	R/W	0: RUN; 1: ALL;
Alarm Assign	03	INT	R/W	0: CH1; 1: CH2
Not Used	04	INT		
Alarm registration number (Digital output number)	05	INT	R/W	0 to 12, 0 : No registration
Event type number	06	INT	R/W	<b>Note 1</b>
Time Delay	07	INT	R/W	0 ~ 99 second
Max value or OP Point	00	FP	R/W	OP point or Max Value : -19999.0 ~ 20000.0 The OP point is -5.0 to 105.0 when event type number is 35 to 38.
Min value or Differential	01	FP	R/W	Differential : 0 ~ 100.0 Min Value : -19999.0 to 20000.0

### NOTE 1

0 : OFF

11: PV-ABS-LOW for Channel 1(PV Event)    12: PV-ABS-LOW for Channel 2(PV Event)  
 13: PV-ABS-HIGH for Channel 1(PV Event)    14: PV-ABS-HIGH for Channel 2(PV Event)  
 15: PV-DEV-LOW for Channel 1(PV Event)    16: PV- DEV-LOW for Channel 2(PV Event)  
 17: PV-DEV-HIGH for Channel 1(PV Event)    18: PV-DEV-HIGH for Channel 2(PV Event)  
 21: SP-ABS-LOW for Channel 1(PV Event)    22: SP-ABS-LOW for Channel 2(PV Event)  
 23: SP-ABS-HIGH for Channel 1(PV Event)    24: SP-ABS-HIGH for Channel 2(PV Event)  
 25: SP-DEV-OFF for Channel 1(PV Event)    26: SP-DEV-OFF for Channel 2(PV Event)  
 27: SP-DEV-ON for Channel 1(PV Event)    28: SP-DEV-ON for Channel 2(PV Event)  
 31: DV-DEV-OFF for Channel 1(PV Event)    32: DV-DEV-OFF for Channel 2(PV Event)  
 33: DV-DEV-ON for Channel 1(PV Event)    34: DV-DEV-ON for Channel 2(PV Event)  
 35: MV-ABS-LOW for Channel 1(PV Event)    36: MV-ABS-LOW for Channel 2(PV Event)  
 37: MV-ABS-HIGH for Channel 1 (PV Event)    38: MV-ABS-HIGH for Channel 2(PV Event)

---

**How to find the Register address according to Alarm Event number.**

(1) Integer type data(Alarm Type, Alarm Action, Alarm Assign, Alarm registration number, Event type number, Time Delay)

Register Address = 4400H + (Event number – 61) \* 08H + OFFSET value from Table 7-6.33

**Example 1 :**

Find the register address of Alarm assign parameter of Alarm Event #62.

Register Address = 4400H + (65-61) \* 08H + 03H = 4423H

**Example 2:**

Find the register address of Event type number parameter of Alarm Event #78

Register Address = 4400H + (78-61) \* 08H + 06H = 448EH

(2) Floating Point type data(OP Point/Max and Differential/Min value)

Register Address = 905CH + (Event number – 61) \* 2H + OFFSET value from Table 7-6.33

**Example 1 :**

Find the register address of OP Point/Max value parameter of Alarm Event #65

Register Address = 905CH + (65 – 61) \* 2H + 00H = 9064H

**Example 2 :**

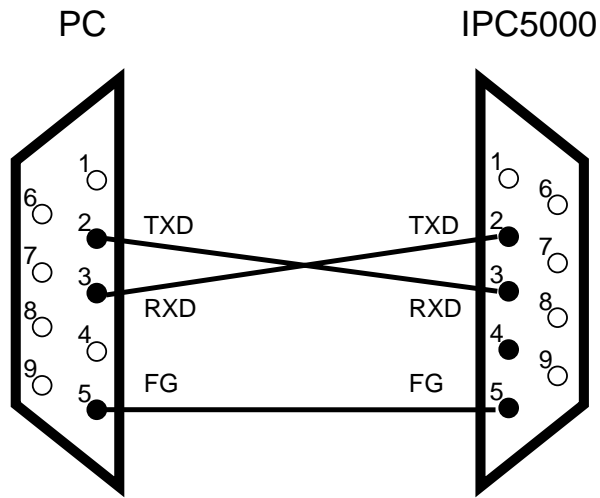
Find the register address of Differential/Min value parameter of Alarm Event #78

Register Address = 905CH + (78 – 61) \* 2H + 01H = 9047H

---

## 7.7 Connection of RS-232C communication

- 9 pin to 9 pin

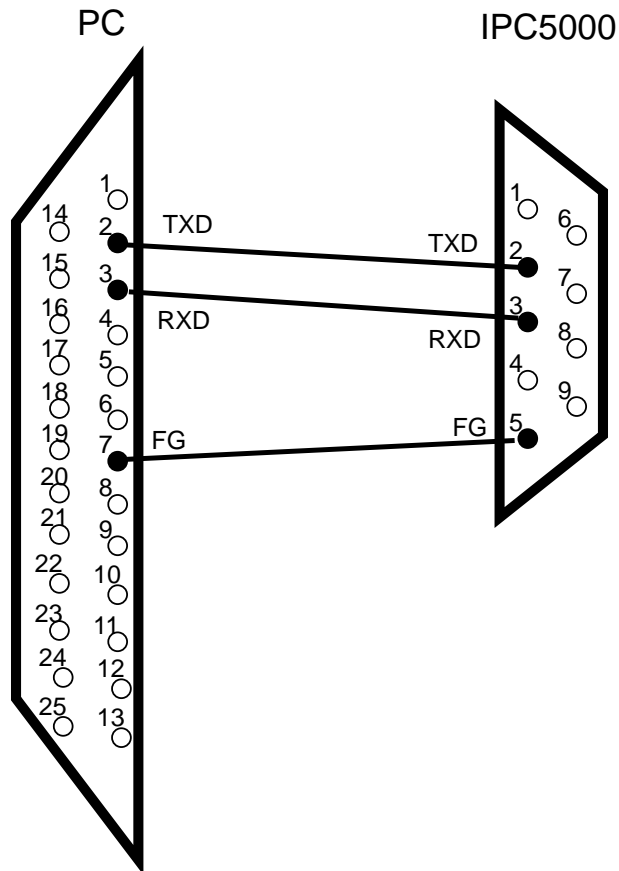


< Figure > 7-7.1 RS-232C communication 9 pin Connection

In this 3-wire cable, **pin 2 and 3** should be **crossed** and pin 5 should be directly wired. The length of the communication line between PC and IPC5000 should be 15m(49.2 ft.) or less.

---

- 25 pin to 9 pin



< Figure 7-7.2 > RS-232C communication 25 to 9 pin Connection

In this 3-wire cable **pin 2 and 3** should be connected directly and pin 5 should be directly wired to pin 7. The length of the communication line between PC and IPC5000 should be 15m(49.2 ft.) or less.

---

## HONEYWELL ASIA PACIFIC AFFILIATES

### Australia

Honeywell Limited  
5 Thomas Holt Drive  
North Ryde NSW 2113  
Phone: (61) 2-9370-4500  
Fax: (61) 2-9370-4525  
Toll Free: 1300-36-39-36  
Toll Free Fax: 1300-36-04-70  
e-mail: [salomon.ayach@honeywell.com](mailto:salomon.ayach@honeywell.com)  
Web: [www.honeywell.com.au](http://www.honeywell.com.au)

### China – PRC – Beijing

Honeywell China Inc.  
15F Han Wei Plaza, East Tower  
No.7 Guang Hua Road  
Choyang District  
Beijing 100020, P.R.C.  
Phone: (86) 10 6561-0208 Ext. 205  
Fax: (86) 10 6561 0618  
e-mail: [jin.gui.wang@honeywell.com](mailto:jin.gui.wang@honeywell.com)

### China – PRC – Shanghai

Honeywell (Tianjin) Ltd.  
23F Tower B City Center,  
100 Zun Yi Road,  
Shanghai 200051, P.R.C.  
Phone: (86) 21 6237-0237 Ext. 305  
Fax: (86) 21 6236 1237  
e-mail: [jing.sheng.qiao@honeywell.com](mailto:jing.sheng.qiao@honeywell.com)

### China – Hong Kong S.A.R

Honeywell Ltd.  
25F Honeywell Tower Olympia Plaza  
255 King's Road  
North Point, HongKong  
Phone: (852) 2331-9133  
Fax: (852) 2331-9998  
e-mail: [wilson.chow@honeywell.com](mailto:wilson.chow@honeywell.com)

### China – PRC – Shenzhen

Honeywell China Inc.  
Units 04-07, 32F Shenzhen Kerry Center  
Renminnan Road, Luo Hu District  
Shenzhen 518001, P.R.C.  
Phone: (86) 755-518-1226  
Fax: (86) 755-518-1221  
e-mail: [robin.tu@honeywell.com](mailto:robin.tu@honeywell.com)

### Indonesia

Honeywell Indonesia Pte Ltd.  
Wisma Budi, #405 4<sup>th</sup> Floor  
H.R. Rasuna Said Kav C-6  
Jakarta 12940, Indonesia  
Phone: (6221) 521-3330  
Fax: (6221) 521-3735  
e-mail: [ariantowibowo@honeywell.com](mailto:ariantowibowo@honeywell.com)

### India

TATA Honeywell Ltd.  
55A 8&9  
Hadapsar Industrial Estate  
Pune 411013, India  
Phone: (91) 20 6875-532  
Fax: (91) 20 6875-535  
e-mail: [ashutoshd@tatahoneywell.co.in](mailto:ashutoshd@tatahoneywell.co.in)

### Japan

Honeywell Inc. Sensing&Control  
TF B/D 14-6 Shibaura 1-Chome  
Minato Ku Tokyo 105-0023 Japan  
Phone: (81) 3 5440-1425  
Fax: (81) 3 5440-1368  
e-mail: [tetsuo.shinno@honeywell.com](mailto:tetsuo.shinno@honeywell.com)

### South Korea

Honeywell Korea Co. Ltd.  
18F KukJe Center B/D  
191 HanGangRo-2Ga  
YongSan-Gu, Seoul, 140-702, Korea  
Phone: (82) 2 799-6176  
Fax: (82) 2 792-9013  
e-mail: [byeongdeok.choi@honeywell.com](mailto:byeongdeok.choi@honeywell.com)  
Web: [www.honeywell.co.kr](http://www.honeywell.co.kr)

### Malaysia

Honeywell Engineering Sdn Bhd  
2F Wisma CSA  
No.4 Jalan Bersatu 13/4  
46200 Petaling Jaya  
Selangor Darul Ehsan  
Phone: (603) 7950 4759  
Fax: (603) 7958 8922  
e-mail: [ks.yong@honeywell.com](mailto:ks.yong@honeywell.com)

### New Zealand

Honeywell Limited  
264 Mt. Eden Road  
Mt. Eden Auckland  
New Zealand  
Phone: (64) 9 623-5050  
Fax: (64) 9 623-5060  
Toll Free: 0800 202-088  
e-mail: [ProductCustomerService@honeywell.com.au](mailto:ProductCustomerService@honeywell.com.au)

### Philippines

Honeywell Systems (Philippines) Inc.  
E-1507A, 15F Tektite Tower 1  
Exchange Road, Ortigas Center  
Pasig City 1605, Philippines  
Phone: (63) 2 636 1649  
Fax: (63) 2 636 1650  
e-mail: [melchor.nicolas@honeywell.com](mailto:melchor.nicolas@honeywell.com)

### Sensing&Control

**Asia Pacific Headquarters**  
Honeywell Building  
17 Changi Business Park, Central 1  
Singapore 486073  
Phone: (65) 355-2828  
Fax: (65) 445-3033  
Web: [www.honeywell.com/sensing](http://www.honeywell.com/sensing)  
e-mail: [info.sc@honeywell.com](mailto:info.sc@honeywell.com)

### Singapore

Honeywell South East Asia  
Honeywell Private Limited  
Honeywell B/D  
17 Changi Business Park, Central 1  
Singapore 486073  
Phone: (65) 355-2828  
Fax: (65) 445-3033  
Web: [www.honeywell.com/sensing](http://www.honeywell.com/sensing)  
e-mail: [info.sc@honeywell.com](mailto:info.sc@honeywell.com)

### Thailand

Honeywell Systems (Thailand) Ltd.  
252/121  
25F Muang Thai-Phatra Office Tower II  
Ratchadapis Road, Huay Khwang  
Bangkok 10320, Thailand  
Phone: (662) 693-3099  
Fax: (662) 693-3085  
e-mail: [somboon.rungjaraspan@honeywell.com](mailto:somboon.rungjaraspan@honeywell.com)

### Taiwan R.O.C.

Honeywell Taiwan Ltd.  
10F Honeywell B/D  
168-1 Lien Chen Road, Chung Ho City  
Taipei Hsien, Taiwan R.O.C.  
Phone: (886) 2 2245-1000  
Fax: (886) 2 2245-3242  
e-mail: [steven.chi@honeywell.com](mailto:steven.chi@honeywell.com)

### For Countries (SEAsia) Listed below,

See Honeywell SEAsia Regional Office  
Bangladesh  
Cambodia  
Guam  
Laos  
Myanmar  
Nepal  
Pakistan  
Sri Lanka  
Vietnam  
East Timor

---

## Industrial Measurement & Control

Honeywell Pte LTD  
Honeywell Building  
17 Changi Business Park Central 1  
Singapore

<http://www.honeywell.com>

# Honeywell