Technical Bulletin, Communicating with Honeywell™ TDC3000 Systems

OMNI FLOW COMPUTERS, INC.
12620 West Airport Boulevard, Suite 100
Sugar Land, Texas 77478 United States of America
Phone-281.240.6161 Fax: 281.240.6162
www.omniflow.com

52-0002-0003/Rev B

NOTE: User Manual Reference - This Technical Bulletin complements the information contained in the User Manual, and is applicable to all firmware revisions Versions 71+.

Communication Options with Honeywell TDC3000 Systems – The OMNI Flow Computer can communicate with Honeywell TDC3000. Systems via SIO modules in combination with APM or HPM modules. PLCG or CLM modules communicate directly with the OMNI.

MVIP Testing – The OMNI Flow Computer has been tested by Honeywell Phoenix as part of their MVIP certification program.

Contact Honeywell at: www.honeywell.com

Table of Contents

Scope

All firmware revisions Version 71+ of OMNI 6000/OMNI 3000 Flow Computers have the capability of communicating with Honeywell™ TDC3000 Systems. This is a feature that requires specific communication modules.

Abstract

This Technical Bulletin addresses the various serial communication options that can be used to transfer data between OMNI Flow Computers and Honeywell TDC3000 systems. The hardware equipment used and the limitations of each method are also discussed.

Three (3) types of serial communication modules are available.

1. Serial I/O (SIO) module in combination with either an Advanced Process Manager (APM) or High Performance Process Manager (HPM) module.
2. Programmable Logic Controller Gateway (PLCG)
3. Communication Link Module (CLM)

MVIP testing was performed using an OMNI 6000 and Honeywell module types one (1) and two (2). Due to the unavailability of equipment and time constraints, tests were not performed using the CLM module. After MVIP testing, it was the opinion of the Honeywell Engineer that communications with the more powerful and flexible CLM module would pose no problem to the OMNI. The nature of the types of tasks performed by the CLM module usually mean that a certain amount of custom I/O driver programming is the norm. This being the case, the CLM is the most flexible but also the most expensive connectivity option.

Communication Method 1: APM / HPM - SIO

Honeywell Engineers state that with regard to serial communication there are no differences between the APM-SIO connection and the HPM-SIO connection. This document will target the APM system but all discussion will also apply to the HPM system.

The APM is an I/O rack system used to get I/O signals into the DCS system. It is comprised of a plug in APM processor module and various other Serial I/O, Analog I/O, and Digital I/O plug in modules. The APM rack system can be expanded by adding one (1) or more additional racks. Assuming open slots are available, up to 16 (sixteen) SIO modules can be connected to each APM system. Each SIO module is connected to the target equipment via a Field Termination Assembly (FTA). Each FTA has two (2) serial ports with each port individually configurable as either an RS232 port or two (2) -wire RS485 port. Port characteristics are as follows:

- Modicon compatible Modbus RTU protocol
- Maximum baud rate of 19200 kbps
- Data bits 8
- Stop bits and parity are selectable

FTA Array Points

Each FTA has a maximum amount of memory space allocated by the APM. This memory is organized in sixteen (16) blocks called Array Points. In addition, each HPM or APM is limited to eighty (80) Array points in total that must be shared between all the SIO modules in its rack system. Each Array Point can therefore hold 512 bits of data and can hold one (1) type of data variable.

Each Array Point can therefore be configured as one (1) of the following:

512	Coils or Status points
32	16 bit Short Integer registers
16	IEEE Floating point variables
16	32 bit Long Integer variables

With a maximum of sixteen (16) array points available per FTA it can be seen that data consolidation and grouping becomes very important. Typical TDC3000-OMNI systems will require a mixture of data types to be exchanged and this further complicates the configuration process. The user must take care not to waste valuable memory space by partially filling array points. Try to minimize the types of variables (e.g.: if you only need to read a few short integers consider converting them to long integers within the flow computer using variable statements). The limited number of array points also impacts how many OMNI Flow Computers can be connected (multi-dropped) to each FTA for example: Most applications require long integer totalizers, IEEE floating point values, and alarm statuses. This means that at least three (3) array points will be needed per OMNI and assumes that sixteen (16) IEEE floats, sixteen (16) totalizers and 512 alarms will be sufficient to transfer all the data needed by the TDC3000 system (extremely unlikely, as there could be up to four (4) meter runs configured).

32-Bit Long Integer Variables

Long integer types are not supported directly by the TDC3000 system. They can be read as two (2) concatenated 16-bit short integers and combined within the TDC3000 system. The Honeywell cannot write to OMNI long integer types because the Honeywell SIO Modbus protocol does not support Modbus function code sixteen (16) (write multiple registers) for integer registers. The protocol does however support writing to IEEE Floating point variables. OMNI's experience has shown that there are very few instances where the TDC3000 system needs to write long integers within the flow computer. Typical long integer data, that there has been a need to write in the past, has been duplicated in IEEE floats as shown.

	Long Integer	IEEE Float
Meter #1 - Current MF in Use	5113	7796
Meter #2 - Current MF in Use	5213	7797
Meter #3 - Current MF in Use	5313	7798
Meter #4 - Current MF in Use	5413	7799
Station Running Batch Size	5819	7787
Station Next Batch Size	5820	7783

	Long Integer	IEEE Float
Meter #1 - Next Batch Size	5820	7783
Meter #2 - Running Batch Size	5825	7788
Meter #2 - Next Batch Size	5826	7784
Meter #3 - Running Batch Size	5831	7789
Meter #3 - Next Batch Size	5832	7785
Meter #4 - Running Batch Size	5837	7790
Meter #4 - Next Batch Size	5838	7786

Configuring the OMNI Flow Computer

Setup the flow computer serial port settings to match the Honeywell FTA settings and make sure to select 'Modicon Compatible'.

In view of the Honeywell array point limitation it is important to group the data as efficiently as possible within the OMNI Flow Computer. Two (2) options are available;

1) Custom data packet arrays

2) Move data to flow computer scratchpad variables using Variable Statements

Method one (1) must be used if it will be necessary to both read and write into the variables. Method two (2) can only be used when it is only necessary to read data.

Data Grouping Option (a) Custom Data Packet Setup

The OMNI Flow Computer has three (3) custom data packet areas where data can be grouped. These three (3) data areas are addressed starting at Modbus addresses 001, 201 and 401. Configure these data areas by completing the custom packet setup menus in the flow computer.

When the OMNI serial port is set as being 'Modicon Compatible' the custom packet data is read/write accessible by the TDC3000 system. Unlike the FTA arrays, the OMNI does allow mixed data types within a custom data packet/array. This means that multiple FTA array points can be associated with one (1) custom packet.

Modbus Function Codes Used to Access Custom Packet Data within the OMNI

The OMNI supports the following Modbus function codes to access custom packet data:

- Read Multiple Registers 03
- Write Multiple Registers 16
- Write Single Register 06

From the Example, it may be seen that Boolean variables must be handled differently when grouped within a custom array. They cannot be accessed using the normal Modbus function codes 01, 05 and 15. They can be read and written as byte-packed bits within Registers not as Coils and Status bits. For this reason it is recommended that writes to Boolean coils be accomplished by using the normal Modbus function code 05 and writing directly to the database Boolean point address.

CAUTION: ⚠ Because Boolean data is byte packed the user must ensure that the number of Booleans included in the custom packet are grouped in such a way as to ensure that the packet always contains an even number of bytes (i.e. the function codes we are using expect to be dealing with 'registers' and you can't have half a register).

Here is an example showing a typical setup using the custom packet located at address 001.

			ADDRESS	FTA ARRAY # USED	
Packet #01 Point #	……. …	710 1	0001 - 0016	1	
# of Points	……. …	8			Total 16 Floats
Packet #02 Point #	……. …	720 1	0017 - 0032	1	
# of Points	……. …	8			
Packet #03 Point #	……. …	730 1	0033 - 0048	2	
# of Points	……. …	8			Total 16 Floats
Packet #04 Point #	……. …	740 1	0049 - 0064	2	
# of Points	……. …	8			
Packet #05 Point #	……. …	510 1	0065 - 0072	3	
# of Points	……. …	4			
Packet #06 Point #	……. …	520 1	0073 - 0080	3	

# of Points	4			Total 16 Long Int.
Packet #07 Point #	5301	0081 - 0088	3	
# of Points	4			
Packet #08 Point #	5401	0089 - 0096	3	
# of Points	4			
Packet #09 Point #	3101	0097 - 0100	4	
# of Points	4			
Packet #10 Point #	3201	0101 - 0104	4	
# of Points	4			Total 16 Short Int.
Packet #11 Point #	3301	0105 - 0108	4	
# of Points	4			
Packet #12 Point #	3401	0109 - 0112	4	
# of Points	4			
Packet #13 Point #	1105	0113 - 0115	5	
# of Points	48			
Packet #14 Point #	1205	0116 - 0118	5	Total 24 Packed
# of Points	48			Bytes
Packet #15 Point #	1305	0119 - 0121	5	
# of Points	48			
Packet #16 Point #	1405	0122 - 0124	5	
# of Points	48			
Packet #17 Point #	0			
# of Points	0			
Packet #18 Point #	0			
# of Points	0	*These packets are available but*		

	…	
Packet #19 Point #	……. 0	*are not used in this example.*
	…	
# of Points	……. 0	
	…	
Packet #20 Point #	……. 0	
	…	
# of Points	……. 0	
	…	

The Example shows a total of thirty-two (32) floating points, sixteen (16) long integers, sixteen (16) short integers and 192 Boolean status bits packed in twenty-four (24) bytes being mapped in one (1) custom data packet and five (5) FTA arrays.

Data Grouping Option (b) Variable Statement Moves to Scratchpad Variables

Option (b) is limited to when data needs to be read but not written to. Non contiguous data is moved into the flow computer scratchpad variables located at:

Boolean Scratchpad Variables	1501 through 1699
Integer Scratchpad Variables	3501 through 3599
String Scratchpad Variables	4501 through 4599
Long Integer Scratchpad Variables	5501 through 5599
Floating Point Scratchpad Variables	7501 through 7599

User Boolean statements are used to group Boolean bits as follows:

Example

1025:	1501=1105:1169	Move 64 bits to 1501 through 1564
1026:	1565=1205:1269	Move 64 bits to 1565 through 1628

User variable statements are used to move all of the remaining data types as follows:

Example

7025:	7501=7101:7103	Move 3 floats to 7501 through 7503
7026:	7504=7201:7203	Move 3 floats to 7504 through 7506

Communication Method 2: Programmable Logic Gateway (PLCG)

The PLCG is meant to receive 'register' data from PLCs representing unscaled analog values and 16-bit counters. Functionality is built into the PLCG which allows the user to easily scale analog inputs of 0-9999 or 0-4095 into engineering units. Alarm points can also be entered and monitored. This philosophy is at odds with the OMNI Flow Computer as the vast majority of the variables within the flow computer are in engineering units requiring no scaling or alarm checking in the PLCG. In addition most of the data is contained in IEEE floating point format or 32-bit long integer values.

The Modbus protocol supported by the PLCG unlike the APM-SIO module does not support reads or writes of IEEE floating point data. The protocol also does not support multiple register writes which would be required to write data to a flow computer long integer type.

The PLCG can however be configured to scale other nominal ranges such as 0-999 of which there are some variables of this type within the flow computer are as follows:

	Mtr#1	Mtr#2	Mtr#3	Mtr#4	Station
Current Gross Flow Rates	3142	3242	3342	3442	3804
Current Net Flow Rates	3140	3240	3340	3440	3802
Current Mass Flow Rates	3144	3244	3344	3444	3806
Current S&W Corrected Flow Rates	3149	3249	3349	3449	

Current Temperature	3147	3247	3347	3447	3809
Current Pressure	3146	3246	3346	3446	3808
Current Analog Density	3148	3248	3348	3448	3810

Counter inputs ranging from 0-65535 are treated more generically requiring no scaling and are usually used for display purposes or are passed to an Application Module (AM) for processing.

There are two (2) options to monitor totalizing within the OMNI Flow Computer:

1) Read long integer totalizers as two (2) consecutive counter inputs and combine in the Application Module (AM) as follows:
   - Totalizer = (high register * 65536) + low register
2) Read specially provided 16-bit integer non-resetable totalizers that roll at 65536 within the OMNI data base as follows:

	Mtr#1	**Mtr#2**	**Mtr#3**	**Mtr#4**	**Station**
Gross Totalizer	3143	3243	3343	3443	3805
Net Totalizer	3141	3241	3341	3441	3803
Mass Totalizer	3145	3245	3345	3445	3807
S&W Corrected Net Totalizer	3150	3250	3350	3450	

The advantage of option one (1) is that any of the internal totalizers of the flow computer can be read in this manner and the results displayed by the TDC3000 system will match the flow computer displayed values. Option tow (2) is limited to one (1) set of non-resetable totals which are not normally displayed at the flow computer and are of limited use.

Using '**Variable Statements**' within the OMNI Flow Computer makes it easy to convert just about any variable within the flow computer's database into a 16-bit register that can be 'read' by the PLCG as either a counter or an analog (assuming the data will fit): the only problem being the availability of enough variable statements (64 are provided).

Example 1: *Variable read as counter for display only*

7025:	3501=7105*#10	3501 contains M #1 temperature in tenths of degrees

Example 2: *Variable read as unscaled analog 0-4095 representing 50 to 150°F*

7026:	7105-#50	Adjust for 50 degree zero point
7027:	3502=7026*#40.95	100 degree span = 4095, move to scratch integer 3502

Note that in Example 2, no attempt was made to limit the impact of over or under range values passed to the PLCG. It is the authors understanding that inputs outside of the expected range cause 'bad process value' alarms in the PLCG.

Selection of Communication Method

Analysis of the various methods available shows that communications via the APM-SIO or HPM-SIO are most likely to provide the best solution, providing reasonable access to the flow computer's database and requiring no custom driver programming in the TDC3000 system. Because of the awkward philosophical fit between the PLCG and flow computer type devices, many of the built in features of the PLCG (such as scaling and alarming) cannot be used. For this reason the use of a PLCG is not recommended except for instances where one already exists in a system and has an open port and an APM or HPM is not available. The CLM module is potentially the most flexible solution but the cost impact of any custom software driver development must be determined. OMNI does not know whether a compatible protocol driver exists at this time, please contact Honeywell for more information in this regard.

DOCUMENT REVISION HISTORY

DOCUMENT INITIAL RELEASE DATE.....03-May-2003

REVISION	DATE	PURPOSE / CHANGE REQUEST
A	03-May-2003	Maintained on the web – initial release
B	12-March-2009	DCR 090077