Technical Bulletin, Understanding OMNI Flow Computer Time Synchronization

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

NOTE: User Manual Reference - This Technical Bulletin complements the information contained in the OMNI User Manuals

Table of Contents

Figures

Scope

This Technical Bulletin applies to OMNI 3000 and 6000 liquid and gas flow computer metering systems beginning with the release of firmware versions/revisions 20.73/20.74+ through 27.73/27.74+. It is not applicable to some of the earlier checksum releases of 20.73/20.74+ through 27.73/27.74+ firmware.

Before implementing any of the recommendations in this Technical Bulletin, thoroughly test the firmware in your particular OMNI Flow Computer to determine if its firmware version supports the use of Variable Statements to synchronize the date/time.

Abstract

The most direct method to synchronize time in an OMNI Flow Computer or multiple flow computers is to use a Modbus 'broadcast' WRITE message. This is the most used method of synchronizing time among pipeline and process plant SCADA/DCS systems.

NOTE: Review Note relating to Ethernet broadcasts.

When there are significant and/or unpredictable transmission delays from the host system, it may be desirable to synchronize time via a digital I/O point instead of a Modbus broadcast write via an Ethernet or serial port link.

It is recommended that time synchronization of the OMNI 3000/6000 Flow Computer be performed a maximum of one (1) time in a twenty-four (24)-hour period.

OMNI provides a "few" special cases for trapping certain changes made by Variable statements. The Time and Date 16-bit registers are one of these special cases. Variable statements within the flow computer are used to make changes to the clock registers based on a change in state of a digital input channel.

There are special considerations that must be taken into account when attempting to synchronize the time via a digital input channel and Variable statements.

Ethernet-to-Serial Port Terminal Servers

NOTE: It is not normally possible to broadcast the time and date to multiple IP Addresses (i.e., multiple devices) via Ethernet from the same supervisory system unless special software is written in the supervisory system to allow it to do so.

However, a single Ethernet Modbus TCP or encapsulated Modbus broadcast message to set the Date/Time can be utilized to synchronize both the flow computer with the SE-Module installed and all other devices connected to the repeater port of that SE-Module.

Broadcast message sent through an Ethernet connection are always passed through the RS-485 two-wire repeater serial port of the SE-Module to other OMNI's or other Modbus devices connected to the flow computer's ES-Module repeater port.

The use of multi-channel Terminal Servers, which act as a single connection to multiple serial channel converters, can inhibit a fast turnaround in Ethernet communications because they must service each of the serial port devices individually in a "round robin" fashion using only one Ethernet port. This can result in significantly long periods between communications with the same serial device.

Background

Some registers contain values that are checked at certain parts of the program execution (like an alarm limit, for example), and can be changed at will at any time. Others, such as the Time and Date, or serial port baud rate settings, require an exception routine to run any time they are changed. The exception routine might be needed to set up program pointers, check the validity of the setting, or manage interrupts while a hardware chip (e.g., the Real-Time Clock) is written to. When registers are written to via Modbus, an OMNI Flow Computer automatically checks for and runs any exception routines as needed.

Register changes made via OMNI Variable statements act like the old 'Peek' and 'Poke' statements in BASIC. Because they are created by the user, it is impossible to anticipate what the user is likely to do and duplicate the exception processing of the serial I/O routines.

It is possible to modify the Time and Date via the registers and also have the exception routine execute to WRITE to the hardware registers of the RTC (Real Time Clock) chip.

CAUTION: ⚠ OMNI advises against doing multiple WRITES to the RTC's hardware registers as this could cause shortened, missing, or extended calculation cycles. For this very reason, OMNI also advises against using the peer-to-peer feature of the OMNI Flow Computer in an attempt to keep time synchronized between multiple flow computers.

The Time (Hours, Minutes, Seconds) and Date (Month, Day, Year) can be modified by writing to the registers located at Modbus addresses 3867 through 3869 and 3870 through 3872 respectively. These are the only Time and Date registers that can do this via Variable statements (e.g., one cannot modify Time and Date using the Time and Date ASCII registers or Long Integer registers, only the 16-bit registers).

NOTE: When writing the Time or Date in the OMNI via Modbus link, you may write to either the ASCII registers or the 16-bit registers.

To minimize the number of Variable statements required, you may want to consider synchronizing only the minutes and seconds, or possibly only the seconds.

Example

OMNI has provided the digital I/O hardwired signal to synchronize the hardware clock registers as an "alternative" method. The negative of this approach is that the user must send the digital pulse at a certain time of day. To avoid the possibility of jumping the OMNI past a time trigger event, such as an hourly report time, a user should execute the synchronization at an unusual time, like 03:17:30 A.M., for example (i.e., distinctly apart from an hour, half or quarter hour time line, at which time critical totalization and averaging functions are occurring inside the flow computer).

NOTE: Synchronizing the time in the flow computer at non-critical hours applies for any method of time synchronization - whether writing the time via Modbus (including Ethernet) or using the digital input with variable statements.

In the example that follows in Figure 1, it is assumed that digital I/O point 5 will be used to synchronize the clock at 03:17:30 A.M.

Digital I/O Assignment

Figure 1 illustrates the use of the OMNICOM for Windows PC-based program to configure a Digital I/O as an input to accept the signal that will trigger the time synchronization.

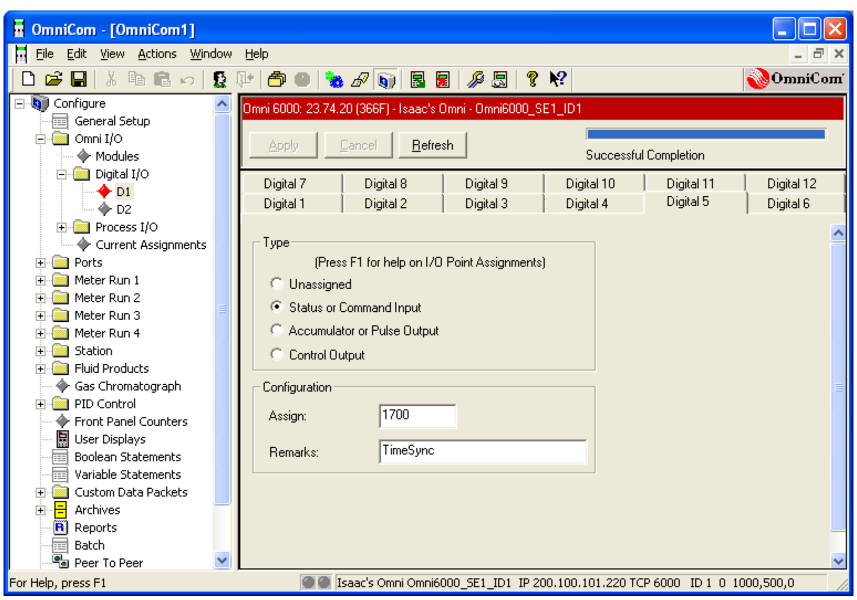

Figure 1. Digital I/O Configured as Input

Boolean Statements

Figure 2 illustrates the Boolean Statements used to 1) transfer the status of the digital input to a 2-second momentary scratchpad register to ensure detection, and 2) detect a transition of Lo to Hi of the digital input.

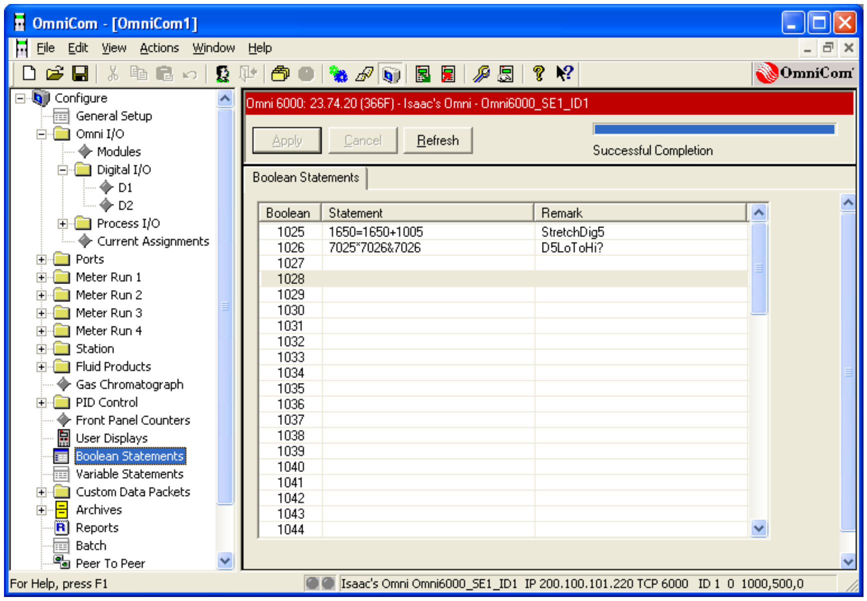

Figure 2. Boolean Statements

1025: 1650=1650+1005	(Ensures that digital I/O 5 is greater than 500 ms by passing its status to a 2-second momentary register at address 1650)
1026: 7025*7026&7026	(Detect rising edge of 7026 [actually Digital I/O 5 low to high transition])

Variable Statements

Figure 3 illustrates the Variable Statements used to 1) save the status of the digital input for comparison against the next cycle for purposes of determining if a change has occurred in the digital input status, 2) determine if the change in the digital input signal was specifically a Low to High transition, and 3) setting the Hours, Minutes, and Seconds when synchronizing the time.

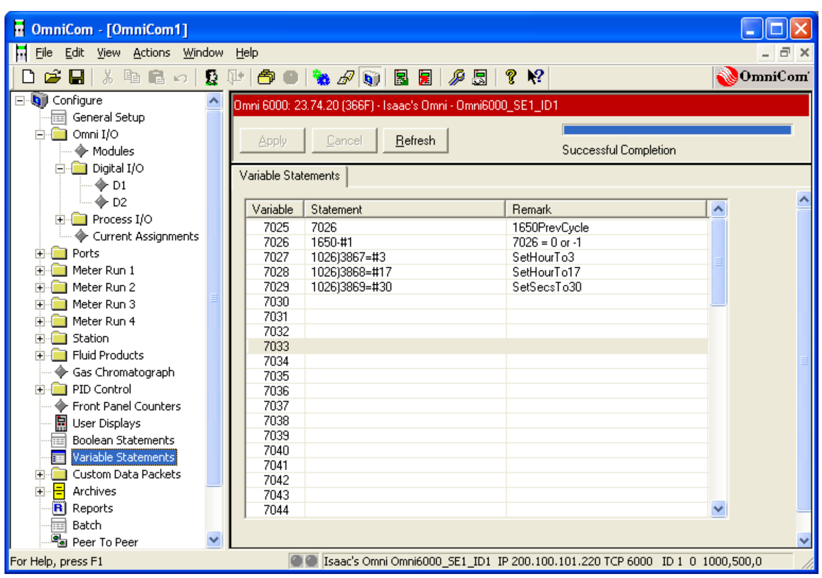

Figure 3. Variable Statements

7025: 7026	(state of 1650 previous 500 ms cycle)
7026: 1650-#1	(7026= -1 means 1650 false, 7026= +0 means 1650 true. When a Boolean statement calls the status of a Variable statement, a value of 0 or greater in the variable statement is considered true and only a negative value is considered false)
7027: 1026)3867=#3	(if Boolean statement 1026 is true, Set Hours to 3)
7028: 1026)3868=#17	(if Boolean statement 1026 true, Set Minutes to 17)
7029: 1026)3869=#30	(if Boolean statement 1026 true, Set Seconds to 30)

Although the flow computer updates the Time registers in three (3) separate Variable statements, the actual values are written to the hardware chip all at once at the end of the Variable statements execution.

Programmable Statements Analysis

Digital I/O point 5 must be activated for at least 101 ms to ensure that it is captured by the flow computer. The amount of time it remains active after that is not important. Boolean statement 1025 is used to "OR" the digital I/O state into a 2 second "one-shot" Boolean scratchpad register (1650) for the purpose of stretching the digital input signal to greater than 500 ms. This step can be eliminated if the digital input signal fed into the OMNI digital I/O point 5 can be guaranteed to remain active high for more than 500 ms.

The OMNI then ensures that the "one-shot" action is synchronized to the 500 ms calculation cycle by monitoring its value for this calculation cycle and the previous cycle using variable statements 7025 and 7026. A 500 ms flag (Boolean statement 1026) is then created which goes TRUE for 1 calculation cycle (500 ms) on the rising edge (i.e., Low to High transition) of digital I/O 5 (status passed to registers 1650 and 1026). This 500 ms flag is used by statements 7027, 7028, and 7029 to move constants representing 03, 17, and 30 into the Time registers at addresses 3867, 3868, and 3869 respectively.

A user can, if preferred, transmit predetermined values for Hours, Minutes, and Seconds, into three (3) 16-bit Modbus "Scratch Pad" registers located at register range 3501 through 3599 and move these values into 3867 through 3869 when digital input 5 (or Boolean scratchpad 1650) goes high. The example does not do this, it uses predefined constants instead.

DOCUMENT REVISION HISTORY

DOCUMENT INITIAL RELEASE DATE.. 14-August-2006

REVISION	DATE	PURPOSE / CHANGE REQUEST
A	14-August-2006	Maintained on the Web - Initial release
B	06-March-2009	DCR 090053
C	22-September-2009	DCR 090265