Tel +1 (717) 767-6511
Fax +1 (717) 764-0839
www.redlion.net

Omni Flow Computer Master Driver v1.x
Omni Flow Computer Master Modicon Compatible Driver 1.x

Information Sheet for Crimson v3.0+

Compatible Devices

Omni Flow Computer devices equipped with Ethernet or serial capabilities.

Verified Device

Omni 6000

Overview

Please review the following information to achieve successful communications. Ensure that the correct driver is selected according to the Modicon compatible protocol selections in the Omni Flow Computer.

If an encapsulated Ethernet driver is required, a virtual port should be used. A new virtual port can be created in Crimson 3.0 by selecting "Network" in the Communications tree, click on the "Add Ports" tab and click "Create New Virtual Port". The new virtual port will appear in the Communications tree. The Driver Selection picker will appear once the virtual port is selected.

Accessing Data

Numeric Data

Numeric tags in combination with data types available in this driver, provide access to signed and unsigned data up to 32-bits.

64-bit Doubles

Access to each 64-bit double value is possible by using 2 elements of a 32-bit data array in conjunction with user functions provided in Crimson 3.0+. Create a numeric tag, select "DHR – 64-bit Holding Register" and enter the register number in the "Element" field. On the "Data" tab, for the "Extent" select "Array" and enter twice as many consecutive registers you want to read or write. Set the "Treat as" to "Integer".

Then use the following user functions to get and set double values, respectively.

> cstring AsTextR64WithFormat(Format, Data)
> Where Format is a string containing the desired width and Data is the first element in the array of the double value that will be passed as a string.
>
> void TextToR64(Input, Output)
> Where Input is a string representing a double value and Output is the first element in the array of the double value to be set.
>
> Example: To display a double value, create a string tag, select "General" for "Source" and enter "AsTextR64WithFormat("17", MyDoubleArray[0])". Use this string tag on a display page. To write an entered value, create a string tag, place on a display page, right click and select "Properties", on the "Entry" tab in the "On Entry Complete" select "General" and enter "TextToR64(MyStringTag, MyDoubleArray[0])". Repeat for every element of the array required incrementing the array index by 2 each time.
>
> Note: Other 64-bit math functions are also available. Please review the Crimson 3.0 Reference Manual.

String Data

When accessing items with a data type of "String", the item should be assigned to a string tag and the tags Packing should be set to ASCII Big-Endian.

Exception Codes

An Exception Code register is available in the Address Selection box for the Omni Flow in Crimson which when displayed will provide useful information of the latest exception code received.

This register is encoded as follow when considered as an 8 digit hexadecimal number: FFOOOOEE, where FF is the function code, OOOO is the address and EE is the exception code.

The user can trigger on all or any part of this value and perform a variety of functions via trigger actions, user functions and programs.

Custom Packets / Archives

Several options are available in working with Custom Packet Data.

Custom Packet Data (Auto) will receive the indicated custom data on demand. In the event that the custom packet contains the same type of data, that specific data type should be used. Otherwise, it is the programmer's responsibility to parse the data consistent with the Omni Flow configuration. User programs and other mechanisms are provided in Crimson 3.0 to achieve this goal.

The following data items are available for manual Custom Packet and Archive access:

1) Custom/Archive Data Point – Set this register to the point number of the desired data.
2) Custom/Archive Record Count – Set this register to the record count of the desired data.
3) Send Manual Custom/Archive Request – Set this bit register to initiate the request to read the desired data.
4) Manual Custom Packet/Archive Data – Provided that the above request is successful, data will be copied to the byte array that is mapped to this data item.

Read and Write ASCII Text Buffers

The following data items are provided for access to ASCII text buffers.

1) ASCII Text Buffer Read Point – Set this register to the point number that should be specified in the Read ASCII Text Buffer request.
2) Send ASCII Text Buffer Read Request – Set this bit register to initiate the request to read the ASCII text buffer as specified in step 1.
3) Read ASCII Text Buffer Data – Provided that the read request is successful, text received will be copied to the string tag array that is mapped to this data item. The tag array should be set to a packing of ASCII Big-Endian. Each response will begin at element 0. User functions and programs can be used to process this data before the next Read ASCII Text Buffer request is sent.

Likewise, provisions have been provided to send an ASCII Text Buffer from the Red Lion to the Omni device as follows:

1) ASCII Text Buffer Write Point – Set this register to the point number that should be specified in the Write ASCII Text Buffer request.
2) ASCII Text Buffer Write Quantity – Set this register to the quantity of bytes that will be included in the buffer write operation.
3) Write ASCII Text Buffer Data – Set each element beginning at element 0 with the text to be sent. This data item should be mapped to a string tag array that should be set to a packing of ASCII Big-Endian.
4) Send ASCII Text Buffer Write Data – Set this bit register to initiate the write to the Write ASCII Text Buffer as indicated in the steps above.

User Function DevCtrl:

Functionality via the following user function is included with this driver:

int DevCtrl (Device, Function, Data)
Where Device is the index of the device to be controlled, Function is the function to be executed and Data is a string containing any parameter necessary for that function.

The charts below details functions available.

Serial:

Function	Description	Data String Contents
1	Set device drop number	Desired drop number
4	Get device drop number	None – empty

Ethernet:

Function	Description	Data String Contents
1	Set device IP address	Desired IP address
2	Set device port	Desired port
3	Set device drop number	Desired drop number
4	Get device IP address	None – empty
5	Get device port	None – empty
6	Get device drop number	None – empty

Cable Information

Serial:

G3 RS232 Port	Omni 6000
2 - Rx	TB-3 Pos 7 - Tx
5 - Tx	TB-3 Pos 9 - Rx
3,4 - Common	TB-3 Pos 10 - Return

Ethernet: Standard 10-Base-T Ethernet Cable

Revision History

02/18/13 – Created
04/18/13 – Revised Custom Packet information to reflect behavior change.
04/18/13 – Revised 64-bit double access notes.
04/26/13 – Add encapsulated driver information in Overview.
04/26/13 – Revised 64-bit double access notes.
04/29/13 – Updated exception code notes.
05/01/13 – Updated Custom Packet / Archive notes.
05/15/13 – Revised 64-bit Doubles info to reference new function sTextR64WithFormat.