



DeltaV / KTron Data Connection

Introduction

The K-Tron (KCM) allows class1 implicit, class3, and unconnected (UCMM) explicit connections via Ethernet/IP from the VIM (Virtual I/O Module) in a DeltaV IO rack. This TechNote details the procedure for creating a Class1 connection. For class1, the K-Tron acts as a server, with the DeltaV controller receiving from and writing data to locations in the K-Tron Assembly tables (up to 5 tables). These assembly areas may be mapped in the K-Tron to data tags that are used in the K-Tron logic. These assembly areas are continuously updated between the K-Tron and the VIM. Figure 1 shows an example of the mapping as configured in the K-Tron KCM module. This K-Tron mapping may be modified depending on the configuration in the K-Tron and on the values that are necessary for transport between the two systems. MYNAH supplies two connection definition files (vdf files) that may be imported and used as is, or as a start for configuring the actual connection definition. Once these have been defined, they may be added to the configuration of any VimNet Explorer viol definition.

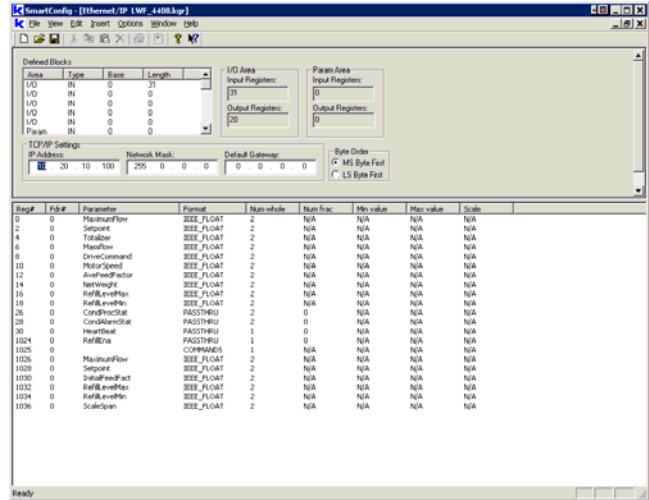


Figure 1

The size of the assembly area array that needs to be included in the communication connection message buffer is dependent on the highest address (and the size of the data structure assigned to this address). It is not necessary to transfer the complete data table. Using this method, the communications is initialized. The size and timing of the communications messages is determined by the VIM.

Since this is a mix of data types, the connection to DeltaV is "mapped" to allow the use of several DeltaV datasets with different data types. These data sets hold values "mapped" to different portions of the data buffers of the connection. These are dependent on the type of data expected in the IO buffer. The mapped connections may be used to transfer complex data structures between the two systems.

VIM Configuration

This section helps users in configuring K-Tron connections to the VIM. It assumes the user is familiar with configuring the VIM using the VimNet Explore utility.

Open the VimNet Explorer. If the appropriate configuration is not already loaded, create a configuration containing the VIM and K-Tron Device. Right click on the "EthernetIP Connection Library" leaf in the navigation panel on the left of the display. Select "Add Connection Definition" or "Import Connection Definition" to load one of our default vdf files.

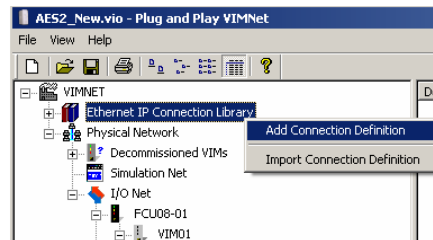


Figure 2



This opens the "VIM_EtipBufferMapping" Utility shown in Figure 3. This is the utility that allows the configuration of the connection between the VIM and the K-Tron. This figure shows the definition of the K-Tron_LWF.vdf module MYNAH supplies as a starting point.

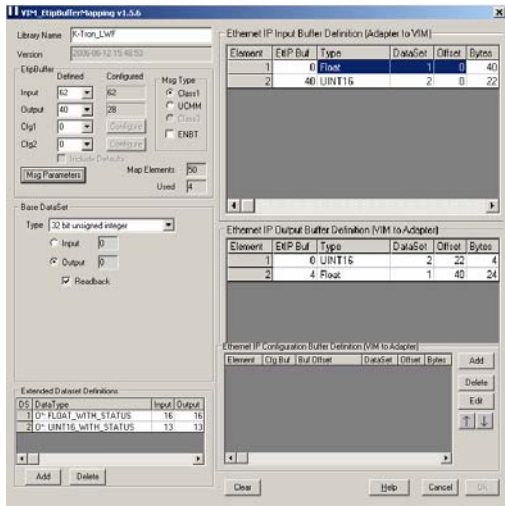


Figure 3

If you wish, you may enter a new name in the "Library Name" field. Then, edit the Etip Buffer section below. Select the message type radio button (Class1) and make sure the ENBT field is unchecked.

In the "Input" and "Output" fields, select the number of bytes that will be included in the communications message buffers. The Input specified the size of the K-Tron to VIM (target to originator or TO), which is the assembly data in the K-Tron that will be transmitted to the VIM, while the Output specifies the bytes transmitted from the VIM to the K-Tron (Originator to target or OT).

Next, select the "MsgParameters" button to open the "SpecialData" utility (Figure 4). This utility allows you to enter or modify the necessary parameters to configure the actual Ethernet/IP Class1 connection. First enter the Connection Points the connection. The connection point is related to but not the same as the K-Tron Address specified in the K-Tron configuration utility. Specify parameters for Originator to Target (OT) and Target to Originator (TO) connections.

Table with 3 columns: IO Area Table, Input Assembly Instance, Output Assembly Instance. Rows include IO Area 1-5 and N/A.

Table 1

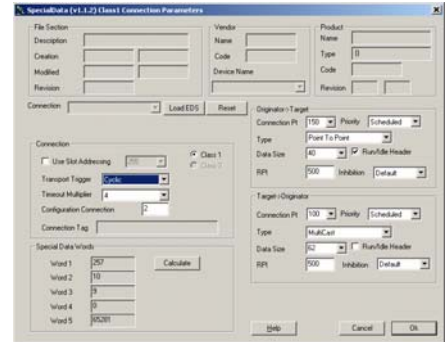


Figure 4

In Table 1, set the Originator (VIM) to Target (K-Tron) to 150, to specify the IO area 1. The values may be from 0x96 (150) to 0x9A (154). Next, set the Target (K-Tron) to Originator (VIM) at 100, to specify IO area 1. The mapping for all the I/O area tables are shown in Table 1. If the connection is one-way, then specify a connection point of 255 for the direction that will not carry data.

Both OT and TO connections should be set to "Scheduled" and Inhibition "Default". The OT connection should be "Point to Point" while TO should be "MultiCast". Make sure the OT connection has the "Run/Idle Header" checkbox checked and the TO does not. Now, set the RPI (requested packet interval) to the required interval. This is the expected time for the K-Tron (TO) and VIM (OT) to send data messages. The Data size for both of these was set using the "VIM_EtipBufferMapping" utility.



Finally, you need to set the overall connection parameters. These are the “Use Slot Addressing” (unchecked), Transport Trigger, “Cyclic”, and Timeout Multiplier (4). Leave the Configuration Connection at 0.

The last step in the SpecialData words is to select “OK” to save these parameters into the current connection definition in the VIM_EtipBufferMapping” utility.

In the VIM_EtipBufferMapping” utility, select the datasets to map these connections to. If there are both input and output connections, this will require two datasets (unless the data values in both assembly tables are intended to be the same). In Figure 5, the first dataset is configured as a 32-bit unsigned integer output (with read-back). The second dataset is configured as a floating point output dataset with read-back. The second dataset is then configured as 16-bit unsigned integer output with read-back. This allows the connection to write one set of values to the K-Tron and read the second set back from it, and track the variables in the same dataset. The first dataset is for backward compatibility. The third dataset could be deleted, and its associated mapping elements could be mapped to the first dataset instead.

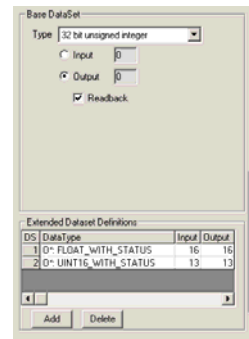


Figure 5

Next, you will set the specific registers to transfer. Select the “Add” button in the EthernetIP Input Buffer Definition (Adapter to VIM) to add the Input Definition. This dialog allows the user to select the dataset (2: FLOAT_WITH_STATUS) for the inputs, set the offset in the dataset (1) and number of bytes (40). Both “Update Status” and “Unmapped Bytes” should be unchecked. As this uses mixed data types, then you will need to enter a second element in the input (1: UINT16_WITH_STATUS), set the offset (0) and bytes (22). Repeat this for the output buffer (Set 2, Offset 22, bytes 4) and Set 1, Offset 40, bytes 24). Each element will point to a specific range of bytes in the buffer and map these to the specific dataset registers for that type.

Selecting the “Fields” button will open a dialog (Ethernet/IP Buffer Fields) shown in Figure 7 that will allow you to enter a series of field definitions. Each field may specify a K-Tron tag definition assigned to the assembly table(s) for the connection.

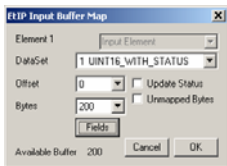


Figure 6

These have no effect on the communications between the K-Tron and DeltaV; however, the field definitions allow you to document the format of the transfer buffer, and will show the specific DeltaV dataset registers associated with the specified fields.

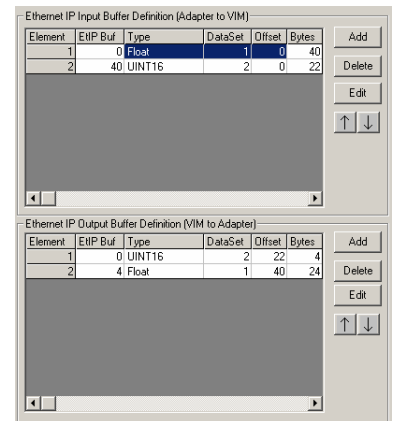


Figure 7



Each field is added by selecting the “Add” button, then filling out the parameters (including the data type of the field), the starting byte in the field (and bit if a bit field). Also, the size (elements) may be set if the field is an array of the data type.

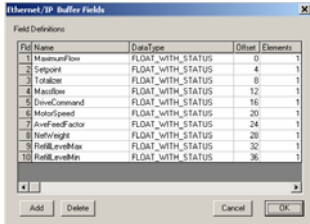
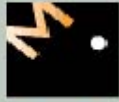


Figure 8

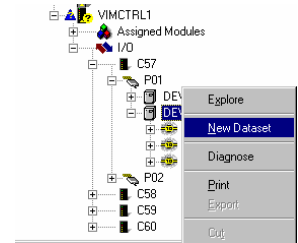
Finally, you select “OK” on all of the dialog boxes to accept the configured values. You are returned to the VimNet Explorer “Ethernet IP Connection Library” with the new definition added. To add this to the VIM configuration select the Node, the specific VIM, card, port and device for the connection, then right click on the device and select “Add Connection” In the dialog box add a description and select the specific connection definition from the “Ethernet Device Definition” combo box. The connection is added to the list and the next available dataset(s) in DeltaV are assigned. You can open this connection by selecting the plus (+) in front of it, and examine the definition of each assigned dataset here. When configuring the actual DeltaV datasets (DeltaV Explorer), you may use these definitions as a reference.



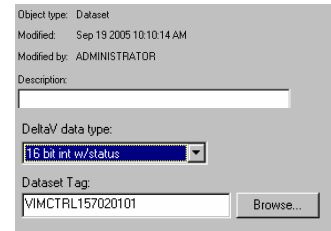
DeltaV Configuration

The actual connection between the K-Tron and DeltaV is configured in VimNet Explorer. The DeltaV dataset(s) assigned in DeltaV Explorer may be on any card (57-60) / port(1-2) and device, but they must match the definition as specified in VimNet Explorer. One or more datasets may be assigned to a connection as necessary. The first dataset defines the connection; the remaining datasets for the connection (if required) must immediately follow the first.

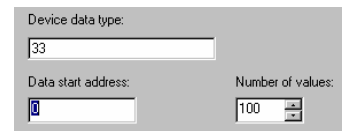
To start, using DeltaV Explorer, select the device, right click, and select "New Dataset" to open the Dataset properties dialog. Select the data direction of the connection data for this dataset. If you are assigning an output assembly instance (AO) to the data this would be "Output." If the assembly instance was input (AI), then use "Input". Refer to the VimNet Explorer definition of the dataset for actual requirements.



The DeltaV tab of the dataset property box is used to select the data type of the data that will be presented. This should be appropriate for the data that will be accessed in the K-Tron. If the data in the assembly table is floating point, only a floating point dataset in DeltaV is appropriate as noted above. Mapping of datasets in the connection will allow this to be assigned separately from integer and bit registers.

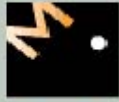


On the PLC tab, the "Device data type" may be specified, but is not required. If specified, the first dataset associate with the connection is specified with a device data type of 39 (mapped EthernetIP Class1 client) on the PLC tab remaining dataset are assigned 36 (EthernetIP extension data). In addition, the number of values on this tab will determine the size of the connection data read/written from this dataset. The number of registers in the associated datasets mapping elements determines the total data in both directions, by the data direction, type, and size in each dataset.



All words on the special data tab should be 0 (any values here are ignored).





From the K-Tron SmartConfig Utility:

I/O Input Blocks (Sent from K-Tron to VIM)

Class 1 connected I/O data (implicit messaging)
 Assembly Instances 0x64 (100) through 0x68 (104)
Class 3 connected explicit messaging
Unconnected explicit messaging
 Assembly Object 0x04
 Instances 0x64 (100) through 0x68 (104)
 Attribute 3
 Service 0x0E (Get_Attribute_Single)

Vendor Object 0xA0

 Instance 1
 Attributes 1 through 5
 Service 0x0E (Get_Attribute_Single)

I/O Output Blocks (Sent from VIM to K-Tron)

 Class 1 connected I/O Data (implicit messaging)

 Assembly Object 0x04

 Instances 0x96 (150) through 0x9A (154)

 Class 3 connected explicit messaging

 Unconnected explicit messaging

 Assembly Object 0x04

 Instances 0x96 (150) through 0x9A (154)

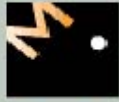
 Attribute 3

 Service 0x10 (Set_Attribute_Single)

Vendor Object 0xA1

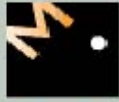
 Instance 1

 Attributes 1 through 5



M Y N A H

Service 0x10 (Set_Attribute_Single)



DeltaV Input Field Xreferences

These are the fields read from the K-Tron based on the definition shown in Figure 1. These are sent by the K-Tron at the interval specified in the RPI (Request Packet Interval) for the connection. The values are updated to the DeltaV datasets on any change.

Data	Buffer Offset (Size)	DeltaV Register	Element(s)
MaximumFlow	0(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R1.CV'	1 Registers
Setpoint	4(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R2.CV'	1 Registers
Totalizer	8(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R3.CV'	1 Registers
Massflow	12(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R4.CV'	1 Registers
DriveCommand	16(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R5.CV'	1 Registers
MotorSpeed	20(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R6.CV'	1 Registers
AveFeedFactor	24(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R7.CV'	1 Registers
NetWeight	28(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R8.CV'	1 Registers
RefillLevelMax	32(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R9.CV'	1 Registers
RefillLevelMin	36(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R10.CV'	1 Registers
CondProcStat	52(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS03/R7.CV'	2 Registers
CondAlarmStat	56(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS03/R9.CV'	2 Registers
HeartBeat	60(2)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS03/R11.CV'	1 Registers

DeltaV Output Field Xreferences

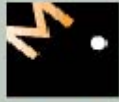
These are the fields written to the K-Tron based on the definition shown in Figure 1. Writing to any of these tags will be written immediately.

Data	Buffer Offset (Size)	DeltaV Register	Element(s)
RefillEna	0(2)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS03/R12.CV'	1 Registers
Command	2(2)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS03/R13.CV'	1 Registers
MaximumFlow	4(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R11.CV'	1 Registers
Setpoint	8(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R12.CV'	1 Registers
InitialFeedFact	12(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R13.CV'	1 Registers
RefillLevelMax	16(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R14.CV'	1 Registers
RefillLevelMin	20(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R15.CV'	1 Registers
ScaleSpan	24(4)	'//VIMCTRL1/IO1/C57/P01/DEV01/DS02/R16.CV'	1 Registers

Please contact us for any questions about these TechNotes at:

MYNAH Technologies

504 Trade Center Boulevard
Chesterfield, MO 63005 USA
1 888 506 9624 (North America)
1 636 681 1555 (International)
1 636 681 1660 (fax)
email: support@mynah.com



M Y N A H

TechNote VIM 4.0.1-2
Module: Generic Ethernet/IP VIM
Version: 4.0.1
Date: July 14, 2006

©MYNAH Technologies 2006. All rights reserved.

Designs are marks of MYNAH Technologies, Emerson Process Management, DeltaV, and the DeltaV design are marks of one of the Emerson Process Management of companies. All other marks are property of their respective owners. The contents of this publication are presented for informational purposes only, and while every effort has been made to ensure their accuracy, they are not to be construed as warranties or guarantees, expressed or implied, regarding the products or services described herein or their use or applicability. All sales are governed by our terms and conditions, which are available on request. We reserve the right to modify or improve the design or specification of such products at any time without notice.

While this information is presented in good faith and believed to be accurate, Mynah Technologies does not guarantee satisfactory results from reliance upon such information. Nothing contained herein is to be construed as a warranty or guarantee, express or implied, regarding the performance, merchantability, fitness or any other matter with respect to the products, nor as a recommendation to use any product or process in conflict with any patent. Mynah Technologies reserves the right, without notice, to alter or improve the designs or specifications of the products described herein.