1. RTU Point Map for Multiplexing ASI VAV Controller Points

1.1 To Define an ASI Board Type

Currently the VAV controllers use the board map defined for Allen Bradley PLCs. A PLC is defined as a “board” of type “mux” (multiplexer) so that an HSQ RTU can communicate with a PLC. HSQ software treats a PLC or a VAV as a mux, but the communication protocols are different. The PLCs and the VAV controllers are two different pieces of hardware and any version of software should only include one type of mux. To define ASI board points, define a board of type PLC. There are (3) point types supported: AI, DI and SP, i.e., analog input, digital input and set point.

1.2 Creating the Point Map

1. Open an MS DOS window and select the RTUDIAG.EXE program.

2. Select **RTU....** from the Command Menu.

3. From the RTU... menu, select **RTU Hardwr Cnfg**. Then, from the RTU Hardwr Cnfg menu, select **Point Map**.

4. From the **Point Map** dialog box, select **Load Default Table**.

5. Move the cursor to a line in the Point Map and press ENTER to edit the line. In the resulting edit window, move the cursor to "BDTYPE", press ENTER, and type "PLC". The edit window will now contain six items, as shown below:

```
         PT TYPE | FIRST PT| LAST PT| BD TYPE|MUX TABL
QUIT          SP       61      120      PLC   PLC TABL
```

   Select QUIT and observe that the line in the point map itself now has only four items: SP, 61, 120 and PLC.

6. Repeat this process for additional SPs (up to 60 per line in the Point Map) and for AIs and DIs as required.

Example 1: DI Point Type Edit Window for a PLC "Board"

```
         PT TYPE | FIRST PT| LAST PT| BD TYPE|MUX TABL
QUIT          DI        1       48      PLC   PLC TABL
```

Example 2: AI Point Type Edit Window for a PLC "Board"

```
         PT TYPE | FIRST PT| LAST PT| BD TYPE|MUX TABL
QUIT          AI        1       16      PLC   PLC TABL
```

1.3 Building a PLC Table

In the Edit Window (illustrated below), move the cursor to the PLC TABL column and press ENTER. The PLC Table will open. Press INSERT to add the first line to the table and then press ENTER to edit the line.

	PT TYPE	FIRST PT	LAST PT	BD TYPE	MUX TABL
QUIT	AI	1	16	PLC	PLC TABL

Each line in a PLC Table specifies consecutive VAV registers that correspond to RTU points covered by that line. If the FIRST PT is 97 and the LAST PT is 100, then four VAV registers have been related to four RTU points. If there are breaks in the VAV register specification**, then a new line in the PLC Table is required.**

NOTE: The range of the PLC table points, first-to-last, must be within the range of the point specification on any given line, otherwise an error message, as shown below, will be displayed.

```
Indicated Point out of range - Illegal value selected
```

Example 3: PLC Table (suitable for Analog Inputs)

PLC TYPE	FIRST PT	LAST PT	PLC ID	FILE NUM	REG NUM	MASK
5	1	8	34	9	47	65535
5	9	16	35	9	47	65535

- **PLC TYPE** - this value is never accessed and is irrelevant in this context
- **FIRST PT** - is the first point defined by this PLC MUX table.
- **LAST PT** - is the last point defined by this PLC table.
- **PLC ID** - ID for the ASI controller with which the RTU communicates.
- **FILE NUM** - this field is used to denote the ASI point table number, normally in the range from 1 to 255. For standard analog or digital inputs use table 9. See the section on ASI point maps below.
- **REG NUM** - this is the entry number in the table defined by the file above, for standard analog or digital inputs use entry 47. *See the section on point maps below*.
- **MASK** - not used by ASI software

1.3.1 ASI Point Maps

ASI data, measurements, setpoints, and configurations are stored in a set of point tables. All communication between the RTU and VAV controllers is in the form of table messages, either writing to or reading from the point tables. A complete description of the communication protocol and point tables is contained in the ASI literature (see ASIC/1-8055 User's Manual, ASI Controls.) There are roughly twenty tables with fifty or so points per table, each entry containing one byte. Sensor readings are two bytes long and take up two entries each. The ASI interface software thus reads two bytes for each AI or DI, while SPs read only one byte. Sensor values are available in a number of different formats, converted or raw. Table 9, entry 47 is recommended as a starting point to read converted sensor values.

1.3.2 Setpoints

Setpoints can be read in the same manner as analog inputs. The important difference is that only 1 byte per point is read, rather than two for AIs. Setpoints can also be written to, using the write SP command.

1.3.2.1 Special Considerations for Set Points (SPs)

ASI VAV Controllers (VAVs) use "tables" (not to be confused with tables in an RTU Point Map) stored in an EEPROM and in RAM to read and write to locations that can be directly related to "points" in an HSQ RTU. An RTU can read and write these points by using SPs defined in a Point Map. The results of such writing can be temporary or permanent, depending upon whether a particular point is in the EEPROM or in RAM, and, if in RAM, whether the VAV firmware also writes to this point.

Each VAV "Table" has several "Entries", each consisting of one byte. In an RTU's "PLC Table", the VAV's "table" number is placed under "FILE NUM" and the VAV's "Entry" number is placed under "REG NUM". The "Entry" number corresponds to the first point stated on a particular line in a PLC Table. For example, if the "FIRST PT" on a line is 151 and the "LAST PT" is 152, and the "FILE NUM" is 2 and the "REG NUM" is 10, then point 151 corresponds to entry 10 and point 152 corresponds to entry 11, both in VAV Table 2. Here, a request from RTUDIAG to "Read SP 152" would show the value of Entry 11 in VAV Table 2.

A line in a PLC Table to read both the state (occupied, unoccupied, night setback, etc.) and the mode (heating, cooling, or deadband) for VAV 7 is shown in the illustration below:

PLC TYPE	FIRST PT	LAST PT	PLC ID	FILE NUM	REG NUM	MASK
5	193	193	7	10	5	65535

The request is "Read SP 193". This reads Entries 5 and 6 (each consisting of only four bits, for a total of one byte) in VAV Table 10. The four LSBs (Entry 5) of this byte reflect the state, while the four MSBs (Entry 6) reflect the mode. (Note: It is very unusual for an "entry" to consist of only four bits, but this is one such case.)

Commands to the VAV controllers are performed by writing values to SPs. The VAV communication task writes the SPs to VAV tables, as specified in the RTU point map. The VAV communication task will not write values to a VAV and read values from that VAV in the same scan. The VAV comm task can detect any writing to a SP (from the Host, from RTUDIAG, or by a Control Block) even if the value does not change. If any SP has been written, then the VAV comm task writes values to the VAV tables on that scan and does not read any values.

Repeated writing to the VAV SPs causes the VAV communication task always to send values to the VAV and never read values from the VAV, so SP values cannot be read by the Host (or by RTUDIAG). Therefore, any method of writing SPs (whether by control blocks or otherwise) must do so only when they do not have the correct (usually the latest) values.

One example of a permanent point value change is that of forcing a damper open or closed. This is done by overriding the VAV hardware outputs. One VAV table entry must have bits set to turn on the 'open damper' DO and turn off the 'close damper' DO, or the opposite. Another VAV table entry must have bits set to override these DO's and force them to the specified values. The damper will stay open or closed until the override bits are set to zero.

A line in a PLC Table to force the damper for VAV 7 is shown in the illustration below:

PLC TYPE	FIRST PT	LAST PT	PLC ID	FILE NUM	REG NUM	MASK
5	225	226	7	10	40	65535

The command is "Write SP 40, C, 3". This writes a decimal 3 to Entry 40 in VAV Table 10, thereby placing a 1 in each of the two LSBs (bit 0 and bit 1) in that byte. To then force the damper open, the command is "Write SP 41, C, 1". This writes a decimal 1 to Entry 41, also in VAV Table 10.

An example of a temporary point value change is that of forcing occupied state. Every 15 minutes the VAV sets the state according to its internal schedule, and does so by writing to its RAM. This overrides whatever command was issued by the RTU. When this happens, a control block used to 'force occupied state' sees that the state is not correct and repeats the command to force occupied state.

A line in a PLC Table to force occupied state for VAV 7 is shown in the illustration below:

PLC TYPE	FIRST PT	LAST PT	PLC ID	FILE NUM	REG NUM	MASK
5	206	206	7	10	30	65535

To force occupied state, the command is "Write SP 206, C, 65". This writes a decimal 65 to Entry 30 in VAV Table 10, thereby placing a 1 in bit 0 (LSB) and a 1 in bit 6 of that byte. It is the 1 in bit 0 that forces the state; the other 1 merely forces the lights in the area to be on, a sensible result.

1.3.3 Serial Configuration

The following serial configuration should be downloaded to the RTU for communication with ASI controllers on COM 2. ASI protocol requires 2 stop bits that are currently hard coded into the RTU software for COM 2, as stop bits are not configurable using **Rtudiag**. Please note that different values for back porch time are required for operation at 1200 and 9600 baud.

	1200 baud	**9600 baud**
protocol	ASI	ASI
front porch (ms)	60	60
back porch (ms)	20	5
transmission timeout (ms)	120	120
frame timeout (ms)	20	20
reply delay (ms)	10	10
silence bytes	0	0

Other baud rates have not been tested but should work with an intermediate value for back porch time.

1.3.4 Digital Inputs

Digital inputs read exactly as analog inputs, but their value is converted to a zero, if the analog value is below or equal to 2048; and to a one, if greater than 2048.

1.3.5 RTU Scanning Operation

When scanning the controllers all input data will be requested at every scan interval, including DP values.

1.3.6 ASIC/1-8055 VAV Controller Definitions

To program the controllers, refer to the programming manuals published by ASI Controls. The interface software was designed using the ASIC/1-8055 User's Manual, copyright October 1997, ASI Controls.

1.3.7 Error Messages

If the RTU receives an error message from a VAV controller it will send a MUX DOWN COS message to the Host.

If a transmission error occurs while communicating with a VAV, the RTU will try to repoll the unit two more times before reporting a MUX DOWN COS to the host. If on the previous scan a given VAV was reported as down, it will be polled only once per scan; the RTU will not bother repolling the unit until the next scan. If a VAV marked as down comes back up, a MUX UP COS will be transmitted.