Overview

This document describes the structure of the Modbus register address map, which is used to configure, control, and monitor the System Control Panel (SCP). Use this document with the Modbus Converter Owner’s Guide (975-0685-01-01 Revision A or later) or the Conext ComBox Owner’s Guide (975-0679-01-01 Revision A or later).

The information in this document is intended for use only by qualified persons who have a detailed technical understanding of the Modbus protocol.

The Modbus map is divided into rows of Modbus registers. Each row indicates the Modbus register address, its name, data type, access type, units, scale, offset, and applicable notes as required. External Modbus Master devices, such as the Schneider Electric M340 PLC, can read and write the Modbus registers to configure, control, or monitor the device remotely.

Document Applicability

The SCP Device Modbus map applies to the following products, as listed in Table 1.

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Product Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>865-1050</td>
<td>Conext System Control Panel for XW/ SW</td>
</tr>
</tbody>
</table>
Supported Modbus Data Types

Table 2 lists the supported data types.

Table 2  Modbus Data Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint16</td>
<td>unsigned 16-bit integer [0,65535]</td>
</tr>
<tr>
<td>uint32</td>
<td>unsigned 32-bit integer [0,4294967295]</td>
</tr>
<tr>
<td>str&lt;nn&gt;</td>
<td>packed 8-bit character string, where &lt;nn&gt; is the length of characters in the string. Two characters are packed into each Modbus register. Example: str20 = 20-character string (packed into 10 Modbus registers) str16 = 16-character string (packed into 8 Modbus registers)</td>
</tr>
</tbody>
</table>

Converting Data to Units of Measurement

Data from a Modbus register is converted to units of measurement using the following algorithm:

\[
\text{result} = \left[\left(\text{data @ Modbus Register}\right) \times \text{scale}\right] + \text{offset}
\]

**Example: Read the Battery Temperature**

The following example shows a conversion of the battery temperature located at Modbus Address 0x0056.

Modbus Address = 0x0056
Scale = 0.01
Offset = -273.0
Data type = uint16 (one Modbus register)
units: deg C

Reading one Modbus Register from address 0x0056 yields 0x7440

Apply the offset and scale as follows:

\[
\text{result} = (0x7440 \times 0.01) + (-273.0)
\]

\[
= (29760 \times 0.01) + (-273.0)
\]

\[
= 297.60 + (-273.0)
\]

\[
= 24.60
\]

The conversion yields a battery temperature of 24.60 degrees Celsius.

Writing Modbus Registers

Modbus does not provide an error response when data written to a Modbus Register is out of range or invalid. To confirm that a Modbus Register is correctly written, you should read it back and compare it with the expected value.

For descriptions of settings and their valid values, refer to the product’s user manual (975-0298-01-01).
## SCP Device Modbus Map

### Table 3 Configuration and Status Registers

<table>
<thead>
<tr>
<th>Modbus Address</th>
<th>Name</th>
<th>Type</th>
<th>read/write (r/w)</th>
<th>Units</th>
<th>Scale</th>
<th>Offset</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>Device Name</td>
<td>str16</td>
<td>r/w</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x000A</td>
<td>FGA Number</td>
<td>str20</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0014</td>
<td>Unique ID Number</td>
<td>str20</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x001E</td>
<td>Firmware Version</td>
<td>str20</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0028</td>
<td>Modbus Address</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0029</td>
<td>Device Number</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x002A</td>
<td>System Instance</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x002B</td>
<td>Hardware Serial Number</td>
<td>str20</td>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0040</td>
<td>Device State</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>See “Operating State”</td>
</tr>
<tr>
<td>0x0041</td>
<td>Device Present</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=Inactive (all data invalid) 1=Active (data valid)</td>
</tr>
<tr>
<td>0x0042</td>
<td>Active Faults Flag</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=No Active Faults 1=Has Active Faults</td>
</tr>
<tr>
<td>0x0043</td>
<td>Active Warnings Flag</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=No Warnings 1=Unit has Active Warnings</td>
</tr>
<tr>
<td>0x0044</td>
<td>Fault Bitmap 0</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>See “Fault Bits”</td>
</tr>
<tr>
<td>0x0045</td>
<td>Warning Bitmap 0</td>
<td>uint16</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>See “Warning Bits”</td>
</tr>
<tr>
<td>0x0046</td>
<td>Configuration Errors</td>
<td>uint32</td>
<td>r</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0048</td>
<td>Reset</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=Reboot 2=Reset to Factory</td>
</tr>
<tr>
<td>0x0049</td>
<td>Operating Mode</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>2=Standby 3=Operating</td>
</tr>
<tr>
<td>0x004A</td>
<td>Clear</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>See “Clear Command”</td>
</tr>
<tr>
<td>0x004B</td>
<td>Display Brightness</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x004C</td>
<td>Display Contrast</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x004D</td>
<td>Backlight Timeout</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x004E</td>
<td>Button Beep</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=Disabled 1=Enabled</td>
</tr>
<tr>
<td>0x004F</td>
<td>Fault Alarm</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=Disabled 1=Enabled</td>
</tr>
<tr>
<td>0x0050</td>
<td>Set Degrees</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=Fahrenheit 1=Celsius</td>
</tr>
<tr>
<td>0x0051</td>
<td>Time Format</td>
<td>uint16</td>
<td>r/w</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0=AM/PM 1=24-Hour</td>
</tr>
</tbody>
</table>
Operating State

Device State can report one of the following values:

- 0=Hibernate
- 1=Power Save
- 2=Safe Mode
- 3=Operating
- 4=Diagnostic Mode
- 5=Remote Power Off
- 255=Data Not Available

Fault Bits

Fault Bitmap 0 can report one or more of the following values:

- bit0=F250 System clock not set
- bit1=F251 A device is lost
- bit2=F252 New device detected
- bit3=F500 Internal failure
- bit4=F501 Memory failure
- bit5=F503 Internal reset
- bit6=F504 Lost network connection
- bit7=F505 Internal failure

Warning Bits

Warning Bitmap 0 can report one or more of the following values:

- bit0=W250 The value failed to change
- bit1=W251 User confirm prompt: Equalization process
- bit2=W252 User confirm prompt: Restore default settings
- bit3=W254 The value failed to change
- bit4=W255 System clock not set
- bit5=W256 A device is lost
- bit6=W257 New device detected
- bit7=W258 Device instance is duplicated
- bit8=W298 Caution: changing frequency can damage loads
- bit9=W501 SCP has fixed a memory problem - restored defaults

Clear Command

The Clear command clears the fault, warning, event, and communication logs. Configure Clear using one of the following values:

- 1=Fault Log
- 2=Active Faults
- 4=Warning Log
- 8=Active Warnings
- 16=State Event Log
- 32=Communication Statistics
- 64=Statistics
- 128=User Statistics
- 255=All