Control Solutions, Inc.
Six Pak™
Generic LonWorks® I/O Node
User Guide

CONTROL SOLUTIONS, INC.
2179 FOURTH STREET, SUITE 2-D ● PO BOX 10789
WHITE BEAR LAKE, MN 55110-0789

www.csimn.com

Control Solutions, Inc.

Six Pak

Generic LonWorks® I/O Node

User Guide

Table of Contents

Control Solutions, Inc.

Six Pak™

Generic LonWorks® I/O Node

User Guide

Rev. 1.0 • July 2007

This manual corresponds with firmware version 1.2.

IMPORTANT SAFETY CONSIDERATIONS:

Proper system design is required for reliable and safe operation of distributed control systems incorporating LonWorks I/O modules and other such devices. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using any Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as well as any consequence for improper system design.

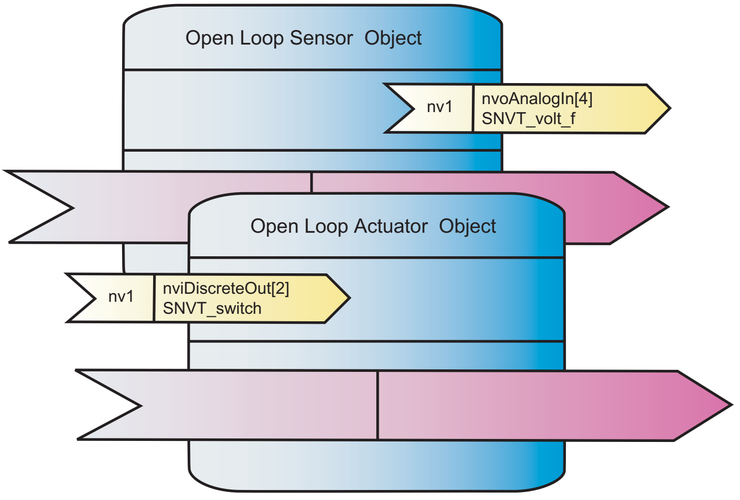
1. Introduction

Six Pak is 6-point LonWorks I/O node designed for general purpose use in building automation, facility management, and commercial control applications. A single Six Pak provides LonWorks network access to 4 analog sensors and 2 discrete actuators. These I/O points are accessed via LonMark open loop sensor and open loop actuator objects or Functional Profiles.

---

2. LonMark Functional Profiles

An overview of the set of functional profiles that make up the Six Pak I/O devices is shown above. There is also a node object in addition to the sensor and actuator objects. Details of each type of object are given in the following sections.



The following objects are defined in Six Pak:

Obj #	Description
0	Node Object
1-4	Open Loop Sensor - Analog Input
5-6	Open Loop Actuator - Discrete Output

Six Pak LonMark certification is pending. All network variables and configuration properties are standard LonMark types of scope 0. No additional resource files are necessary for configuration and operation of Six Pak.

---

3. Node Object

Object Request `nviRequest`

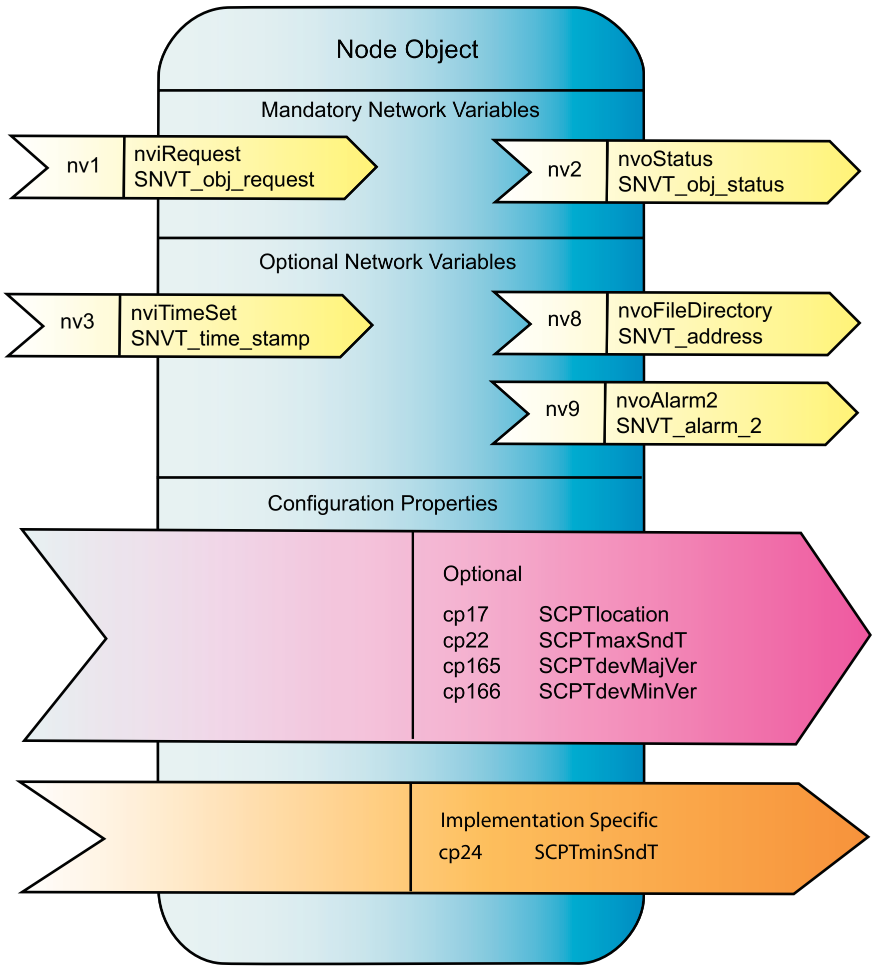
This network variable is used to request operations to be performed or modes to be specified for functional blocks within the device. The following requests are processed by all sensor and actuator objects:

RQ_NORMAL
RQ_DISABLED
RQ_UPDATE_STATUS
RQ_REPORT_MASK
RQ_OVERRIDE
RQ_ENABLE
RQ_RMV_OVERRIDE

The following additional requests are processed by analog input sensor objects:

RQ_UPDATE_ALARM
RQ_CLEAR_ALARM
RQ_ALARM_NOTIFY_DISABLED
RQ_ALARM_NOTIFY_ENABLED

Functional blocks must be enabled before they will produce data. The “enabled” status will be retained through power cycles. Override status will also be retained through power cycles. The initial default state is enabled for all objects.



Object Status `nvoStatus`

This output network variable reports the status for any functional block within the device. The following bit fields, as defined for SNVT_obj_status, are supported by Six Pak:

invalid_id
invalid_request
disabled
locked_out
in_alarm
in_override
report_mask

Real Time Update `nviTimeSet`

This SNVT_time_stamp input network variable sets the internal real time clock with a time stamp to be used for subsequent alarm reports. The internal clock is maintained by software, and will count up from zero if not set. Time is maintained as a 32-bit count of seconds since 2000-01-01T00:00:00Z.

File Directory `nvoFileDirectory`

SCPT values are accessed via the direct memory read/write file transfer method. The file directory is implemented as specified by LonMark standards, and is used primarily by network management tools.

Alarm Report `nvoAlarm2`

Any enabled alarms will report alarm instances via this output network variable which uses the SNVT_alarm_2 type. The updating of this variable is throttled by SCPTminSndT. New alarms will be reported as they occur. In addition, any existing alarm conditions will be reported as a result of issuing RQ_ALARM_UPDATE to the sensor object desired.

Node Location `SCPTlocation`

This configuration property may be used by the system integrator to store physical location information as desired.

Maximum Send Time `SCPTmaxSndT`

This configuration property sets the maximum period of time that can expire before the functional block automatically updates the nvoStatus output network variable. This configuration property also applies to the nvoAlarm2 output, setting the maximum period of time that can expire before the alarm output variable is updated even if updated with only AL_NO_CONDITION.

Device Major Version `SCPTdevMajVer`

This configuration property is read-only, and provides software version information for this device as defined by LonMark standards.

Device Minor Version `SCPTdevMinVer`

This configuration property is read-only, and provides software version information for this device as defined by LonMark standards.

Alarm Minimum Send Time `SCPTminSndT`

This configuration property sets the minimum time that must expire before the nvoAlarm2 output will be updated again after the previous update. This rate throttle applies primarily in the case of multiple simultaneous alarm conditions.

---

4. Analog Input - Open Loop Sensor Object

Sensor Value `nvoAnalogIn[16]`

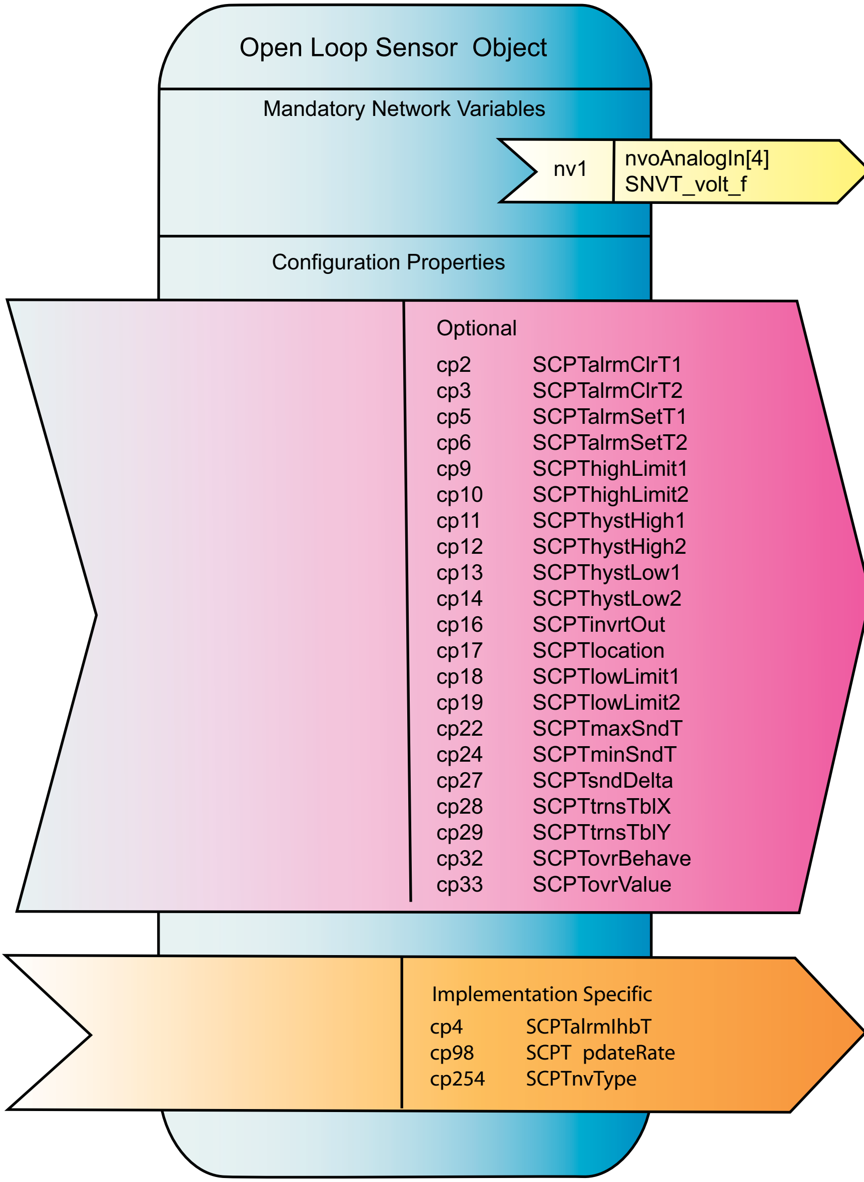
This output network variable transmits the value from the sensor after conversion to correct units. The units are conveyed in the type specified by the SCPTnvType configuration property. The scaled data is maintained internally as a floating point value that is derived by applying the translation table entries to the raw sensor data. This floating point value is then further translated, as applicable, to network variable type data. The default network variable type is SNVT_volt_f but may be changed to any scalar type, or to SNVT_switch, SNVT_lev_disc, or SNVT_state. An attempt to change the type to an invalid type will result in the Node Object nvoStatus producing an invalid_request status for the object being changed.

Alarm 1 Clear Time `SCPTalrmClrT1`

This is the amount of time that must pass without the Alarm 1 condition present before it will be considered a valid clear alarm. If the alarm condition returns before this time has expired, the time count starts over and the alarm remains set.

Alarm 2 Clear Time `SCPTalrmClrT2`

This is the amount of time that must pass without the Alarm 2 condition present before it will be considered a valid clear alarm. If the alarm condition returns before this time has expired, the time count starts over and the alarm remains set.



Alarm 1 Set Time `SCPTalrmSetT1`

This is the amount of time that must pass with the Alarm 1 condition present before it will be considered a valid alarm. If the alarm condition clears before this time has expired, the time count starts over and the alarm remains clear.

Alarm 2 Set Time `SCPTalrmSetT2`

This is the amount of time that must pass with the Alarm 2 condition present before it will be considered a valid alarm. If the alarm condition clears before this time has expired, the time count starts over and the alarm remains clear.

Alarm 1 High Limit `SCPThighLimit1`

This configuration property sets the alarm high limit against which the sensor's output network variable is tested for the Alarm 1 condition. A high alarm condition will exist when the network variable's value is above this level. This property inherits the same NV type (SNVT) assigned to the respective output network variable. No limitations are imposed on the value of the limit.

Alarm 2 High Limit `SCPThighLimit2`

This configuration property sets the alarm high limit against which the sensor's output network variable is tested for the Alarm 2 condition. A high alarm condition will exist when the network variable's value is above this level. This property inherits the same NV type (SNVT) assigned to the respective output network variable. No limitations are imposed on the value of the limit.

Alarm 1 High Limit Hysteresis `SCPThystHigh1`

This configuration property sets the hysteresis that applies to the Alarm 1 high limit. Once the network variable's value exceeds the high limit, it must drop below the high limit by this amount before the alarm will be considered clear. After meeting the hysteresis criteria, the alarm clear time begins counting before the alarm status will become indicated as clear.

Alarm 2 High Limit Hysteresis `SCPThystHigh2`

This configuration property sets the hysteresis that applies to the Alarm 2 high limit. Once the network variable's value exceeds the high limit, it must drop below the high limit by this amount before the alarm will be considered clear. After meeting the hysteresis criteria, the alarm clear time begins counting before the alarm status will become indicated as clear.

Alarm 1 Low Limit Hysteresis `SCPThystLow1`

This configuration property sets the hysteresis that applies to the Alarm 1 low limit. Once the network variable's value exceeds the low limit, it must rise above the low limit by this amount before the alarm will be considered clear. After meeting the hysteresis criteria, the alarm clear time begins counting before the alarm status will become indicated as clear.

Alarm 2 Low Limit Hysteresis `SCPThystLow2`

This configuration property sets the hysteresis that applies to the Alarm 2 low limit. Once the network variable's value exceeds the low limit, it must rise above the low limit by this amount before the alarm will be considered clear. After meeting the hysteresis criteria, the alarm clear time begins counting before the alarm status will become indicated as clear.

NOTE: The alarm set and clear times and hysteresis values do not take effect until the next system reset, or the next issuing of RQ_NORMAL, RQ_ENABLE, or RQ_ALARM_NOTIFY_ENABLED to the sensor object.

Object Location `SCPTlocation`

The location string found in the node object should be used to identify the location of the device. The primary purpose of this location string (one per analog sensor) is to identify the source of alarms as reported by the nvoAlarm2 of the node object. The first 22 characters of this location string will be included in the alarm message as the alarm description.

Alarm 1 Low Limit `SCPTlowLimit1`

This configuration property sets the alarm low limit against which the sensor's output network variable is tested for the Alarm 1 condition. A low alarm condition will exist when the network variable's value is below this level. This property inherits the same NV type (SNVT) assigned to the respective output network variable. No limitations are imposed on the value of the limit.

Alarm 2 Low Limit `SCPTlowLimit2`

This configuration property sets the alarm low limit against which the sensor's output network variable is tested for the Alarm 2 condition. A low alarm condition will exist when the network variable's value is below this level. This property inherits the same NV type (SNVT) assigned to the respective output network variable. No limitations are imposed on the value of the limit.

Maximum Send Time `SCPTmaxSndT`

This "Send Heartbeat" configuration property sets the maximum period of time that can expire before the functional block will automatically update the respective sensor's output network variable. The default is zero, which disables the heartbeat.

Minimum Send Time `SCPTminSndT`

This "output throttling" configuration property sets the minimum period of time that must expire before the functional block will allow an update of the respective sensor's output network variable. A zero value does not disable this timer, but rather means there is no throttle on network traffic generated by this object.

Send on Delta `SCPTsndDelta`

This is the minimum change in value that must occur before the output network variable will be updated, subject to minimum send time. A zero means any change in data will result in network traffic. This value inherits the NV type of the sensor's output network variable.

Translation Table X `SCPTtrnsTblX`

This is a table of floating point values that is used in conjunction with the Y table to create a linearization table. It is used to translate 4-20mA signals to a 0-100% range, and also used to linearize thermistors and convert raw readings to temperature.

Translation Table Y `SCPTtrnsTblY`

This is a table of floating point values used in conjunction with the X table to create a linearization table. The points must be in matched X-Y pairs to be effective, and values must be in monotonically increasing order. Values between points will be extrapolated.

Override Behavior `SCPTovrBehave`

This configuration property specifies the override behavior for this functional block (or object). It may be set to “retain” the previous value, go to a default value of zero, or go to the value specified by the Override Value that follows.

Override Value `SCPTovrValue`

This is the value that will be assigned to the network variable output in place of any value derived from hardware while the functional block is in override. The value assumes the NV type of the network variable overridden. The override will go into effect when the override command is issued via nviRequest in the node object.

Alarm Inhibit Time `SCPTalrmIhbT`

This is the time period for which alarms are inhibited after enabling, node reset, or node is put on line.

Object Update Rate `SCPTupdateRate`

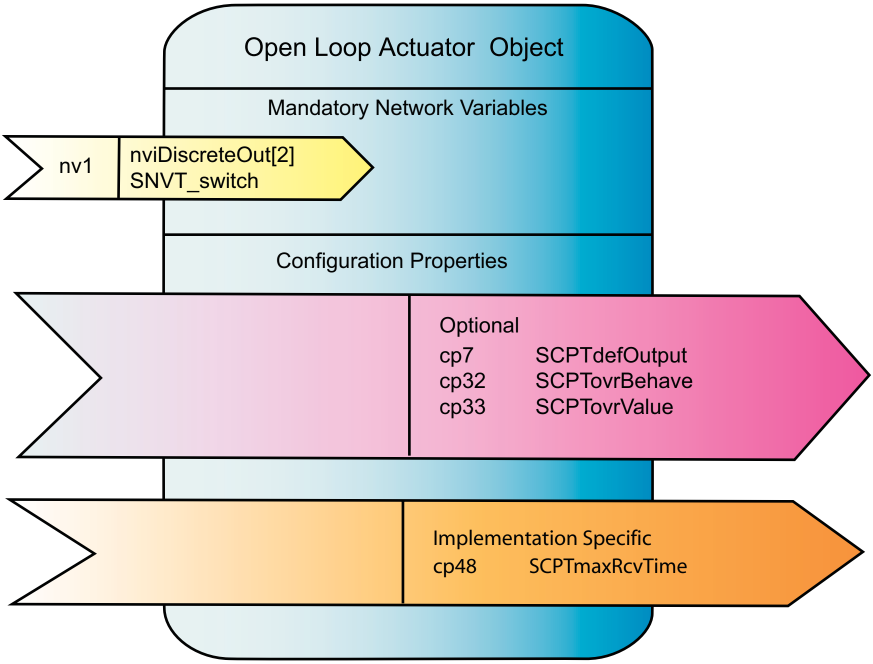
This is the rate, in tenths of seconds, at which the sensor functional block will be updated. An update consists of evaluating data from the hardware to check for alarms, send on delta, etc. Minimum and maximum send times are also updated in this increment.

Network Variable Type `SCPTnvType`

This configuration property assigns the network variable type (SNVT) to the respective output network variable. Any scalar type is supported, both 2-byte and 4-byte sizes including floating point. In addition, SNVT_switch, SNVT_lev_disc, and SNVT_state are available. Other enumerations or structures are not supported.

Invert Output `SCPTinvrtOut`

This configuration property inverts the interpretation of the input when used as a discrete input. When set, the input will be pulled up to +5V through 6K ohms in anticipation of contact closure to ground as the switched input.



5. Discrete Output - Open Loop Actuator Object

Actuator Value `nviDiscreteOut[8]`

This input network variable receives the state at which to set the discrete output hardware. This network variable is fixed at type SNVT_switch. A state of 1 and non-zero value will turn on the output relay. Either a state of 0 or zero value will turn off the relay.

Default Output Level `SCPTdefOutput`

This SCPTdefltOut configuration property sets the default output level that is assumed when the object is disabled, or when overridden to the default value. This property is of type SNVT_switch.

Override Behavior `SCPTovrBehave`

This configuration property specifies the override behavior for this functional block (or object). It may be set to “retain” the previous value, go to a default value defined by the default output configuration property, or go to the value specified by the Override Value that follows.

Override Value `SCPTovrValue`

This is the SNVT_switch value that will be assigned to the hardware output in place of any value received from the network while the functional block is in override. The override will go into effect when the override command is issued via nviRequest in the node object.

Maximum Receive Time `SCPTmaxRcvTime`

This "receive heartbeat" configuration property sets the maximum time that can elapse after an update to the network input variable before the hardware will assume the default output value. A value of zero (0) disables the receive failure detect mechanism.

---

6. Getting Started

There are a large number of configuration properties required to fully configure Six Pak. To simplify initial use of Six Pak, all of these properties default to known states that can be used to provide minimum functionality. Alarms are disabled, minimum send times default to 15 seconds, update rates default to 5 seconds, maximum send times and send on delta default to zero. I/O points default to object enabled, with the following configurations:

• Analog Input: 0-10V input voltage, data range 0 to 10.000 in SNVT_volt_f
• Discrete Output: SNVT_switch

Translation tables are initialized to all zeroes (no points defined). If the tables contain no translation points, data is passed through without conversion of raw data. Raw data for an analog input is a floating point number in the range of 0 to 10.000. Raw data for a discrete output is SNVT_switch, but no translation is available for discrete outputs. Refer to the following section on Setting Up Translation Tables to establish I/O data ranges other than the defaults.

7. Setting Up Translation Tables

Translation tables are sets of X-Y data points used to scale and optionally linearize data. This translation is used to do simple things like translate a 0-100% of scale to a 4-20mA signal with the proper offset to obtain 4mA output at a 0% data value. More advanced translation such as converting highly nonlinear raw thermistor readings to a linear temperature scale is also done using these tables.

The translation table operates as a lookup table with data between points interpolated between those points. Most of the time only two points will be entered in the table, but the table allows for up to 7 points. The data in the table is floating point values. All values in the X table must be in increasing order. Values in the Y table must correlate with the data at the same position in the X table. Some example translations follow.

Translation of 4-20mA input to data scale of 0-100 would use the following:

X table entry	Y table entry
1.97647	0
9.88235	100

Replace 0, 100 with any data range that corresponds to 0-100% of full scale.

Translate an A/UI input to SNVT_switch as follows:

X table entry	Y table entry
1.0	0
10.0	100

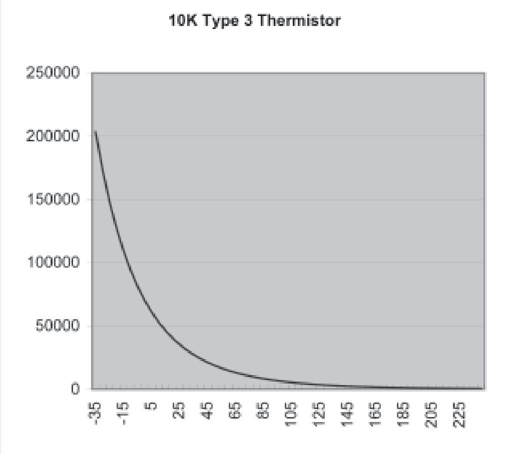
This will produce a state of "off" (SNVT_switch state=0, value=0) when the input is at or below 1 volt, and state=1 with value 0..100% for the input range of 1 to 10V.

The sense of the input may be inverted for contact closure to ground. The input jumper is set in the thermistor position to read a contact closure to ground. In this configuration, the input senses full scale voltage when the contact is open, and (near) zero voltage when the contact is closed. If one wants to produce a SNVT_switch "on" when the contact is closed, the following translation would be used:

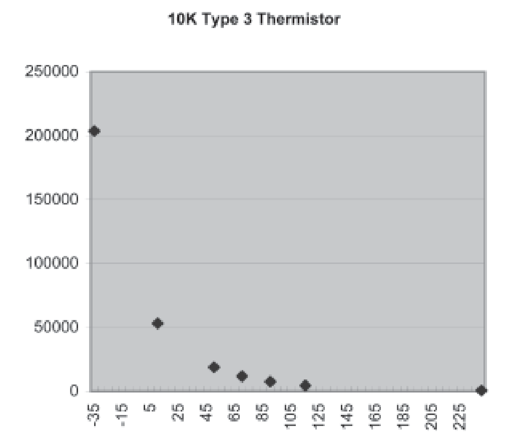
X table entry	Y table entry
1.0	100
8.0	0

Using the above translation table, any voltage below 1V will result in SNVT_switch state=1 and value=100% while any voltage over 8V will result in SNVT_

switch state=0 and value=0.
Thermistors may be linearized using the translation table. The following chart illustrates the characteristic curve of a 10K type 3 thermistor.



The area of prime interest in most building automation applications is the region near room temperature. The chart shown below illustrates a set of 7 points chosen to linearize the 10K type 3 thermistor with emphasis on best accuracy around room temperature.



The linearization table provides up to 7 points (X-Y pairs). The general formula for calculating any thermistor translation table is:

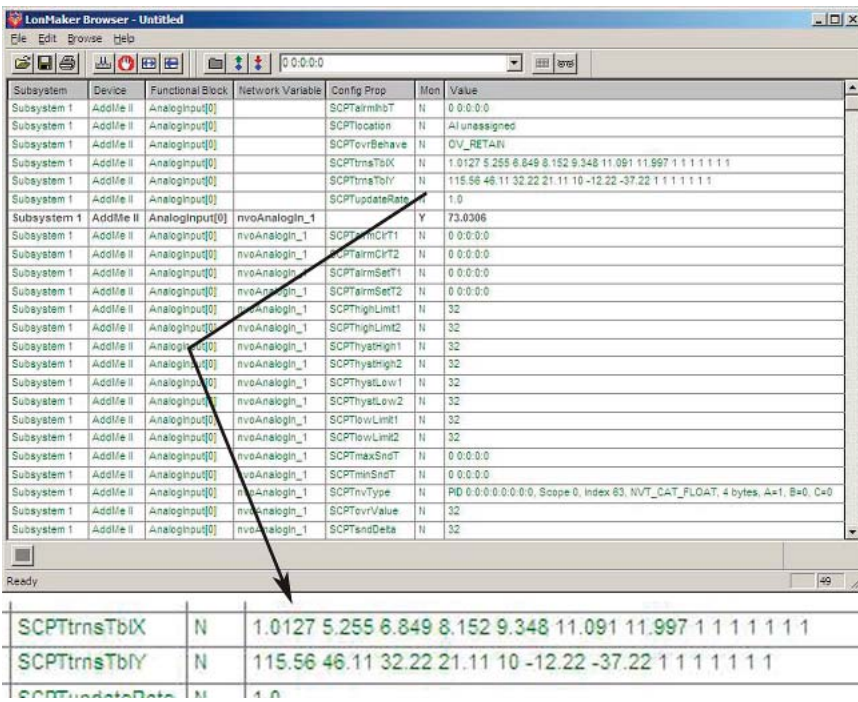
$$X = 12.3288 * R / (R + 6040)$$

where R is the resistance at temperature Y

The translation table for converting 10K type 3 thermistor input to degrees is as follows:

Temp ºF (ºC)	10K III resistance	X table entry	Y table entry
240 (115)	539.4	0.83044	240 (115.56)
115 (46)	4472	5.2449	115 (46.11)
90 (32)	7516	6.8356	90 (32.22)
70 (21)	11720	8.1359	70 (21.11)
50 (10)	18790	9.3298	50 (10.00)
10 (-12)	53070	11.069	10 (-12.22)
-35 (-37)	203600	11.97359	-35 (-37.22)

Entering the translation table for a 10K type 3 thermistor for an analog input via LonMaker would look like the following. Note that if the NV type is SNVT_temp (or temp_p, temp_f), the translation table must be entered in degrees C since the SNVT is transmitted as degrees C, even though LonMaker displays degrees F. If an uninterpreted scalar such as SNVT_count_f is used, temperature may be represented in degrees F directly.



(above example for illustration only, refer to tables for actual numbers.)

Note that there are two sets of numbers for each set of points. The first set of 7 floating point numbers are the data values. The second set of numbers, which may only be 1 or 0, are a bit field with bits set to 1 to indicate that the corresponding position in the translation table should be used as a valid point (even if the value is zero). In the case of a thermistor, all data points in the table are filled. The translation table entries for a 4-20mA input scaled to 0-100% would look like the following.

SCPTtrnsTblX	N	1.97647 9.88235 0 0 0 0 0 1 1 0 0 0 0 0
SCPTtrnsTblY	N	0 100 0 0 0 0 0 1 1 0 0 0 0 0

The following table is used to translate raw input to temperature for a 10K type 2 thermistor:

Temp ºF (ºC)	10K II resistance	X table entry	Y table entry
240 (115)	438.3	0.83412	240 (115.56)
115 (46)	4182	5.04393	115 (46.11)
90 (32)	7331	6.7596	90 (32.22)
70 (21)	11880	8.17333	70 (21.11)
50 (10)	19900	9.4581	50 (10.00)
10 (-12)	62480	11.2420	10 (-12.22)
-35 (-37)	280100	12.0686	-35 (-37.22)

The following table is used to translate raw input to temperature for a 20K type 4 thermistor:

Temp ºF (ºC)	10K II resistance	X table entry	Y table entry
230 (110)	438.3	1.46832	230 (110.0)
115 (46)	4182	6.9792	115 (46.11)
90 (32)	7331	8.6803	90 (32.22)
70 (21)	11880	9.85155	70 (21.11)
50 (10)	19900	10.7591	50 (10.00)
10 (-12)	62480	11.8143	10 (-12.22)
-10 (-23)	280100	12.0596	-10 (-23.33)

Translation of 4-20mA output from a data scale of 0-100 would use the following table. The full scale raw hardware output is a binary value of 255. The output range is actually 0 to 20mA, so to make a data range of 0 to 100% correspond to 4 to 20mA, the offset of binary 51 at 0% is needed (results in 4mA at 0%).

X table entry	Y table entry
0	51
100	255

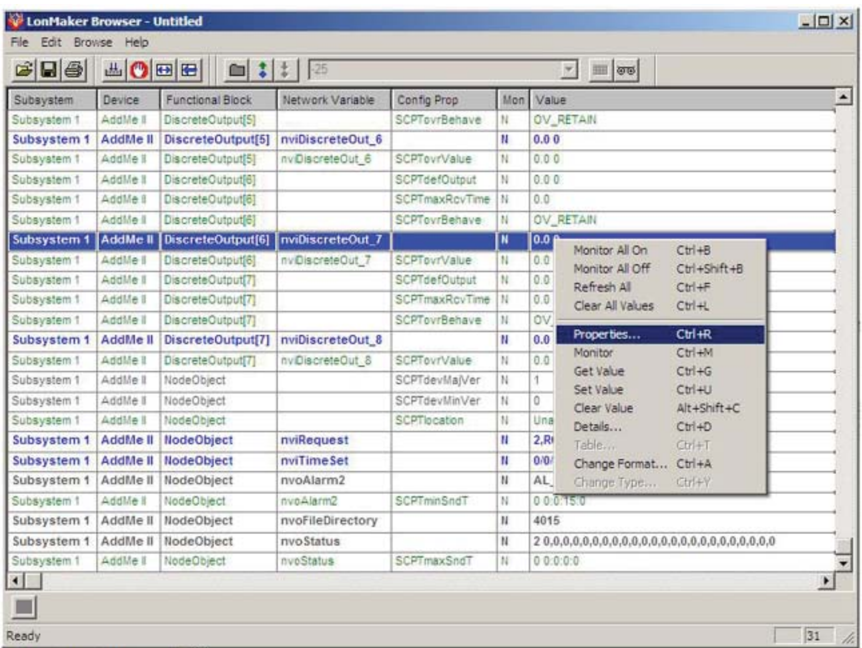
Replace 0, 100 with any data range that corresponds to 0-100% of full scale.

8. Functional Block (Object) Status

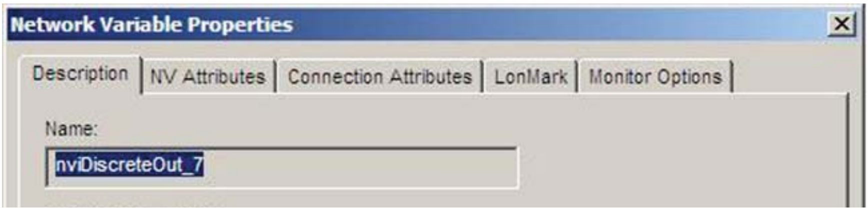
Each of the Six Pak sensor and actuator objects maintains its own set of status bits. These are accessed via the node object's nviRequest and nvoStatus as defined by LonMark standards. Object numbers (Object ID) used to access individual object status are as follows:

0	Node Object
1-4	Open Loop Sensor - Analog Input
5-6	Open Loop Actuator - Discrete Output

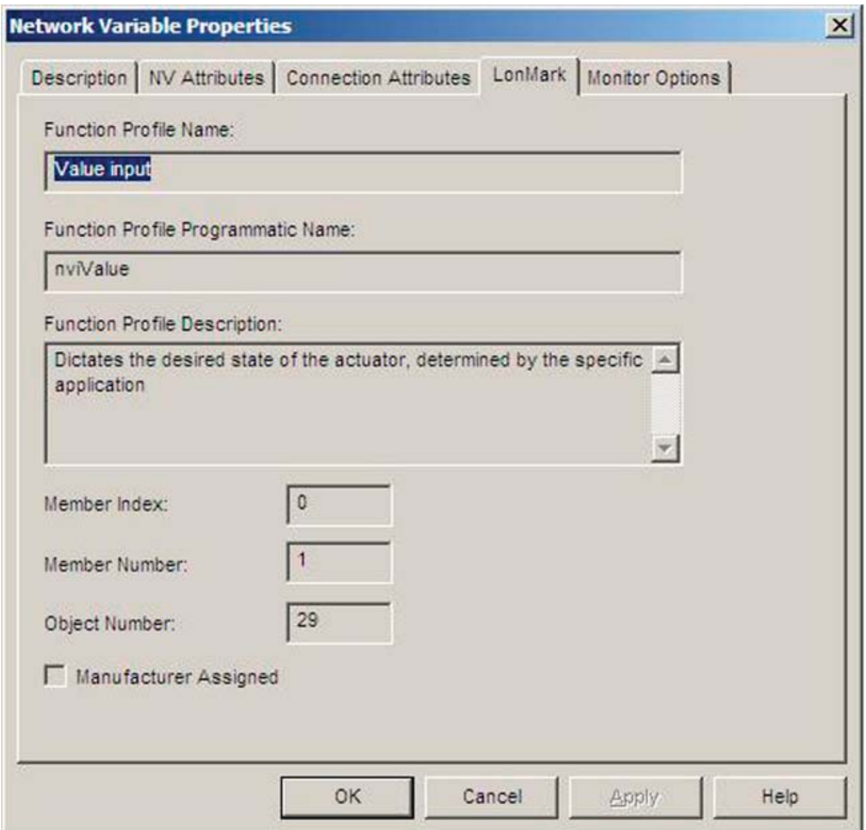
To find the Object ID associated with any particular network variable, you can use the LonMaker browser to look it up. Right click on the desired variable, and select "Properties…" from the popup menu.



Select the "LonMark" tab on the property sheet that appears next.



The number shown near the bottom of the page indicated as "Object Number" is the number to make a note of for use in making object requests.



Once you have determined which object number to issue a request to, enter the object number followed by the request in the edit box at the top of the browser window, and click on the button to the left (red arrow pointing down). For example, enter 2,RQ_ENABLE to enable object #2.

Right click on nvoStatus, and select "Monitor" from the popup menu to enable monitoring of status so that it is updated following each request that is made. The

"Mon" column will indicate "Y" when monitoring is enabled, "N" if not.

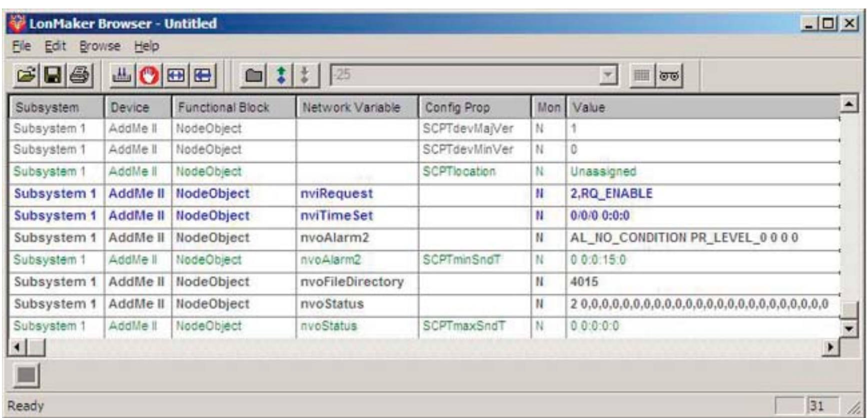
The following requests are processed by all sensor and actuator objects:

RQ_NORMAL
RQ_DISABLED
RQ_UPDATE_STATUS
RQ_REPORT_MASK
RQ_OVERRIDE
RQ_ENABLE
RQ_RMV_OVERRIDE

The following additional requests are processed by analog input sensor objects:

RQ_UPDATE_ALARM
RQ_CLEAR_ALARM
RQ_ALARM_NOTIFY_DISABLED
RQ_ALARM_NOTIFY_ENABLED

The "Normal" request is the same as issuing both "Enable" and "Remove Override" (RQ_RMV_OVERRIDE) requests at the same time. All objects default to enabled.



The status is indicated as:

2 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Bits which are utilized by Six Pak are indicated as follows:

2 a,b,c,0,0,0,0,0,0,0,0,0,0,0,0,0,0,d,e,f,0,0,g,0

where '0' will always be returned as zero, and bit positions labeled will be set to 1 (one) when the following conditions are true:

a – invalid_id: Object ID was invalid, e.g., object #70 does not exist
b – invalid_request: Unsupported SNVT type change requested
c – disabled: Object is disabled

d – in_alarm: Alarm condition is set
e – in_override: Object is in override state
f – report_mask: Set to indicate this value is a report mask
g – alarm_notify_disabled: Set if analog input alarms turned off

When the network variable type of an object is changed using the LonMaker browser, the object will be disabled first, then re-enabled after the change. Upon receiving the object enable request, the functional block will check its network variable type. If it has been changed to an invalid (unsupported) type, it will be restored to its previous type, and the invalid_request bit will be set in the object status.

To check the object status without making any changes, simply issue an update request by entering 2,RQ_UPDATE_STATUS in nviRequest. (Object 2 used as an example here, you would use whatever object number you wish to check.)

---

9. Analog Input Alarm Processing

There are a total of 4 alarms per analog input sensor object, 2 high alarm limits and 2 low alarm limits. Hysteresis and clear/set times apply to each. The instances of each alarm are reported via the nvoAlarm2 variable found in the node object. Alarm limits inherit the network variable type of the object's output network variable. Alarm limit variable types are therefore changed (automatically) right along with any change made to the output network variable type.

Alarm limits are not forced within any particular bounds. Although usually impractical to do so, if one wishes to, high alarms may be set lower than low alarms. There is no interaction among alarm limits.

HIGH ALARMS

A high alarm condition is considered "set" when the output network variable's value is at or above the high alarm limit. High alarm #1 would generally be set lower than high alarm #2, but does not have to be. However, alarm #2 will always be assigned a higher priority level than alarm #1. Alarm #1 is given priority level 1 and alarm #2 is given priority level 2 (the higher level).

The high alarm is considered "clear" when the network variable's value falls below the high limit by a minimum threshold set by the respective high alarm hysteresis property.

LOW ALARMS

A low alarm condition is considered "set" when the output network variable's value is at or below the low alarm limit. Low alarm #1 would generally be set higher than low alarm #2, but does not have to be. Alarm #1 is given alarm priority level 1, and alarm #2 is given alarm priority level 2.

The low alarm is considered "clear" when the network variable's value rises above the low limit by a minimum threshold set by the respective low alarm hysteresis property.

HYSTERESIS

Each high and low alarm has a hysteresis value associated with it. This creates an optional "dead band" such that the alarm, once set, will not return to the clear state until the network variable's value returns to normal by the margin established by this property.

CLEAR & SET TIMES

The set and clear times establish time delays such that brief transitions past alarm limits will not be observed while longer term transitions past alarm limits do result in an alarm responses. An alarm condition that would otherwise be considered set will not be processed as set until the set time has expired without the corresponding alarm becoming clear during that time. An alarm condition that would otherwise be considered clear will not be processed as clear until the clear time has expired without the corresponding alarm becoming set during that time. If the set or clear time is configured as zero, there is no delay in processing the alarm. The clear time begins counting after accounting for any applicable hysteresis.

ALARM NOTIFY ENABLE

Alarm notification is disabled by default. It is enabled by issuing a RQ_ALARM_NOTIFY_ENABLE to the object via the object request variable in the node object. The notify enable state is retained through reset and/or power loss. The alarm_notify_disabled bit will be set in the object's status bit mask if alarms are disabled. This bit is only valid for analog input sensor objects.

ALARM REPORTING

The "in_alarm" bit will be set in the object's status bit mask if alarm notify is enabled and any of the alarm conditions are set. Once set, this status bit will remain set until a RQ_CLEAR_ALARM request is issued to the object. In addition, the node object's alarm reporting variable nvoAlarm2 will be used to report alarms in more detail.

Alarm reports will be generated automatically via nvoAlarm2 as they occur. In addition, alarm status may be requested by issuing a RQ_UPDATE_ALARM request to the object for which status is desired. The reporting variable nvoAlarm2 will produce a single update for each occurrence of an alarm as they occur. The update request may result in multiple messages (nvoAlarm2 updates) if multiple alarm conditions exist. When there are multiple alarms to report, the first message (nvoAlarm2 update) will contain AL_HEADER as the alarm type, followed by the alarms, followed by a final message with AL_FOOTER as the alarm type.

The RQ_UPDATE_ALARM request will result in one message with AL_NO_CONDITION as the alarm type if there are no alarms to report. Each object maintains a history of sorts such that the most recent instance of an alarm transition is "remembered". If an alarm is set, it will produce an alarm set message. If an alarm is clear, it will produce an alarm clear message that includes the timestamp recorded at the time the alarm cleared. One message will be produced for each remembered alarm transition if one has occurred since the last RQ_CLEAR_ALARM. The history is cleared by the RQ_CLEAR_ALARM and a subsequent RQ_UPDATE_ALARM will return AL_NO_CONDIITON again until the next alarm transition(s).

The alarm notification message produced by nvoAlarm2 includes the alarm type, alarm priority level, time stamp, sequence number, and description as defined for LonMark SNVT_alarm_2. The time stamp is maintained only in seconds; the milliseconds field will be zero. The internal time is maintained only in software, and is set using the node object's nviTimeSet. The internal time resets to zero at power up or system reset, and increments upward from zero if not otherwise set. Therefore relative or elapsed time can be reported even if real time is not.

The description field of the alarm report is taken from the SCPTlocation property of the sensor object. Since this is the only means of linking an alarm message with its source, this ASCII string should be set to something meaningful that indicates the source of the alarm. It is not necessary to indicate high or low alarm in the location string as this is automatically generated in nvoAlarm2. Since SNVT_alarm_2 allows 22 characters for the description, only the first 22 characters of the 31-character SCPTlocation are used in the alarm message.

---

10. Changing Network Variable Type

Changeable type network variables for sensor and actuator objects default to SNVT_count_f, but can be changed using LonMaker (or appropriate network management tool). Using the LonMaker browser, right click on the network variable to be changed. Select "Change Type…" from the menu.

The check box labeled "Standard Network Variable Type" must be checked. The check box labeled "Display types of same size only" does not need to be checked. Select the desired network variable type, and click Apply or OK.

Once the network variable type has been changed, check the object's status. If the selected type is not supported by Six Pak, the invalid_request bit will be set, and the object will be disabled.

SNVT types that are supported by Six Pak include all scalar types such as signed or unsigned integer (8-bit, 16-bit, 32-bit), or floating point. In addition, SNVT_switch, SNVT_lev_disc, and SNVT_state are provided with special case translation. Enumerations and structures other than state and switch are not supported.

SNVT_switch is translated as follows: When converting to SNVT_switch, the value is bounded to 0..100%, and state set to 1 if value is nonzero. When converting from SNVT_switch, the intermediate value is set to zero if the state is zero, otherwise the intermediate value is set to the SNVT_switch value field.

SNVT_lev_disc is translated as follows: When converting from SNVT_lev_disc, the value is set to the following: ST_OFF=0, ST_LOW=25, ST_MED=50, ST_HIGH=75, ST_ON=100. When converting to SNVT_lev_disc, the value is bounded at 100, then divided by 25 and truncated to an integer in the range of 0 to 4 (ST_OFF to ST_ON).

SNVT_state is translated as follows: When converting to SNVT_state, the value is converted to unsigned 16-bit binary, and placed in SNVT_state. Note that since SNVT_state is defined as a structure, but is converted as a 16-bit binary number, the structure's bit0 is the most significant bit (MSB) of the binary number and the structure's bit15 is the least significant bit (LSB). The process is simply reversed when converting from SNVT_state.

Scalar types are converted according to the scale factors set in the SCPTnvType configuration property. These factors will be set automatically if the change is made via the LonMaker browser. The scale factors are assumed to be 1,1,0 for floating point types, but vary for other integer types that are more application specific. In conjunction with applying the scale factors to non-floating point types, the type category is used to make data size conversions on scalar types.

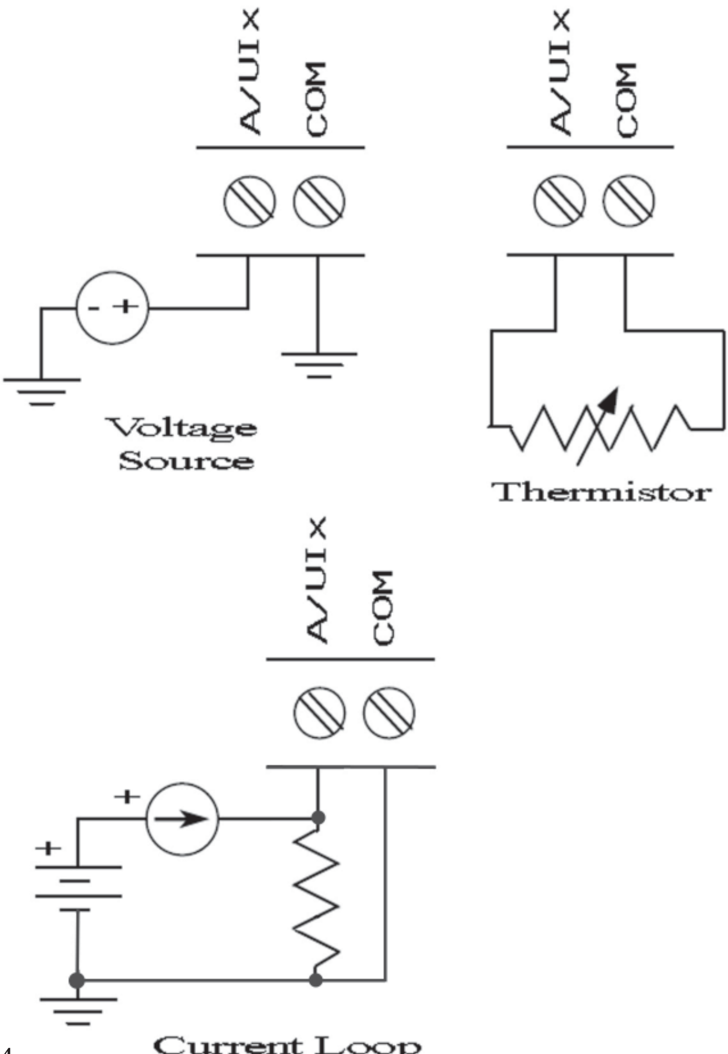
---

11. Analog Input Hardware & Sensor Field Wiring

The analog to digital (A/D) converter is a fast successive approximation convterter with 10 bits of resolution. Inputs are referenced to the internally regulated 5VDC power supply. This means of referencing is optimized for thermistors, which get their excitation from the same supply. Any variation in 5V regulation will be cancelled out completely, which is most critical near room temperature for thermistors.

There are no configuration jumpers inside the Six Pak. Software controlled FETs do the necessary switching to select either voltage or thermistor input types. A current loop signal (4-20mA) may be read by putting a 249 ohm 1% or 499 ohm 1% resistor across the input.

The following diagram shows the various field wiring configurations for Six Pak analog inputs. The thermistor jumper position is also used for sensing dry contact closures to ground.



Voltage input is the default configuration for the AI's. Thermistor input is automatically switched when the network variable type is changed to SNVT_temp, SNVT_temp_f, or SNVT_temp_p. The thermistor setting supplies 5VDC to the A/UI terminal via a 6.04K 1% resistor. The "thermistor setting" is also used for dry contact input. When the contact is closed to ground, the input will swing from 5V to ground. The SCPTinvrtOut may be set to ST_ON to force selection of the thermistor hardware setting if a temperature SNVT type is not used.

To create an X-Y translation table for thermistors, calculate X = 12.3288R / (R+6040) and set the corresponding Y to the temperature corresponding to resistance R. This translation applies to the normalized 10V readings. Thus for a 10K type III thermistor, which has a resistance of 11,720 ohms at 70°F the normalized reading will be 8.136.

---

12. Power Supply & Network Wiring

Six Pak includes a TP/FT-10 (FTT-10) free topology 78k baud transceiver. Network wiring should be done per LonWorks guidelines for the TP/FT-10 channel.

Six Pak will operate from 24VAC or 10-30VDC. Connect DC+ to POWER, and DC- to GND, or connect AC across POWER and GND. **IMPORTANT: DO NOT POWER SIX PAK from the same source used to power AddMe II or Babel Buster gateways that are documented as requiring ungrounded AC.** The older products have full wave bridge rectifiers which require isolated AC. The Six Pak uses a half wave rectifier that allows grounded AC to be used.

---

13. Front Panel Operation

The standard Six Pak front panel is illustrated below.

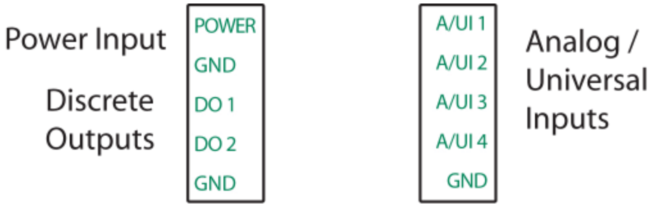
The button labeled "Service" is the service button used in standard LonWorks fashion. There are two LEDs on the front panel. The service LED (labeled "SVC") behaves in standard LonWorks fashion. It should flash once at power up, and remain off if the node is configured and online. The wink LED (labeled "WNK") flashes about twice a second with a duty cycle of mostly on in response to the wink command, then continues to blink briefly once every 5 seconds indefinitely if the Six Pak application is running. At power up, the wink LED will exhibit wink command behavior once the Neuron Chip has booted up and completed initialization.

Power on can be verified by pressing the service button at any time. It is hard wired to the SVC LED which will come on when the button is pressed if power is present. There is also a power LED inside the unit just above DO2.

---

14. Connector Terminals

The screw terminals for wiring Six Pak are labeled on the enclosure. An illustration of the labels and notation of their signal grouping is shown below. The terminal blocks are screw clamps accepting 22-14AWG wire. The terminal blocks may be unplugged from the base unit.



15. Opening the Case

Open the case by gently sliding a screw driver under the edge of the cover at the locking tab. Lift up over the tab.

When closing the Six Pak enclosure, note that the locking tabs at the ends of the base unit are not the same size. The cover will only fit on one way.

---

16. Mounting on DIN Rail

Six Pak is designed to be mounted on DIN rail. To attach to the rail, hook Six Pak on the top edge, and push onto the bottom edge. The enclosure should snap into place.

To remove, insert a small screw driver in the release tab at the bottom center of the enclosure. Gently pry downward no more than 1/8” (3 mm). Pull the bottom of Six Pak away from the rail and lift upward.

17. Trouble Shooting

Node does not seem to be responding.

(a) Verify that the LEDs are providing indications as described under Front Panel Operation. If Power LED is not lighted, verify power source. If service LED remains on or continues flashing, verify node installation with network management tool.

(b) Test a “wink” command. The Wink LED should flash rapidly for a short time each time a wink command is received by the node. If there is no response, check network connection.

Node responds to network, but some I/O points do not seem to be responding.

(a) Is object disabled? Check object’s status by issuing RQ_UPDATE_STATUS to the object via nviRequest in the node object. Enable the object if disabled.

(b) Is the object in override? Check object’s status using RQ_UPDATE_STATUS, issue RQ_RMV_OVERRIDE if in override.

(c) Check minimum send time and update rate for sensor objects. If set too long, sensors may appear to not respond. Test minimum send time of zero and update time of 2 seconds.

(d) Check maximum receive time for actuator objects. If non-zero, output will assume default value if network updates not received often enough. Test object with maximum receive time set at zero.

I/O point responds, but not correctly.

(a) Check to see that the network variable type is as expected both as configured in Six Pak and as expected by nodes communicating with Six Pak. Verify that NV type changes were accepted by checking to see that invalid_request was not set as a result of making the change request.

(b) Check to see that internal jumpers are set in the correct position if it is an analog input that is not responding correctly.

(c) Check to see that the translation table is set up correctly. To test the I/O point with raw data, set all values in the translation table to zero. This includes both the data points and the bit masks. See the section on Setting Up Translation Tables.

18. Specifications

- 4 Analog/universal inputs
    - 0-10VDC, Thermistor, dry contact
    - 10-bit resolution
- 2 Discrete outputs
    - Open drain FET, 1A @ 24VDC
- 3150 Neuron® Chip, 48K Flash memory, 10K RAM
- FTT-10A transceiver (TP/FT-10)
- Powered by 10-30VDC or 24VAC 50/60 Hz
- Power Consumption:
    0.2A @ 24VDC
- DIN rail mounting, 100mm H x 36mm W x 60mm D
- Operating environment: -40°C to +85°C, 5% to 90% humidity

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

WARRANTY

Control Solutions products are warranted against defects in materials and workmanship for a period of 1 (one) year from date of shipment from factory. Defective units will be repaired or replaced, at manufacturer's discretion, at no cost to user except when negligence or improper use has resulted in damage. The express warranty stated herein is in lieu of all other warranties, express or implied, including without limitation any warranties of merchantability or fitness for a particular purpose and all other warranties are hereby disclaimed and excluded by Control Solutions, Inc.

Quality LonWorks®
Product Development
Since 1995.

CONTROL SOLUTIONS, INC.™
2179 FOURTH STREET, SUITE 2-D • PO BOX 10789
WHITE BEAR LAKE, MN 55110-0789

www.csimn.com