SECTION 25 15 16
SOFTWARE FOR PROGRAMMING LOCAL CONTROL UNIT

PART 1 - GENERAL

1.1 SUMMARY

A. Section includes software requirements for Local Control Units and programming logic documentation.

B. All Local Control Units shall be free programmable controllers. This ensures advanced control and alarm strategies can be implemented. The use of application specific controllers limits programming and diagnostic capabilities. The requirement to add annotation to control logic facilitates troubleshooting.

C. Related Sections
   1. 25 00 00 Integrated Automation
   2. 25 13 00 Control and Monitoring Network
   3. 25 12 23 Client-Server Information/Database Integration
   4. 25 14 00 Local Control Units
   5. 25 15 00 Building Controls System Server Software
   6. Stanford Network Architecture diagrams (MC-01)

1.2 REFERENCES

A. Refer to 25 00 00 Integrated Automation

1.3 DEFINITIONS

A. Refer to 25 06 11 Integrated Automation Definitions

1.4 SYSTEM DESCRIPTION

A. Software for Direct Digital Control system installed. Software includes programming tools used to meet the design requirements of controllers, specified sequences and design intent applications.

B. Programming tools shall be graphical based. Software requiring line code is not acceptable.

C. Programming and configurations shall utilize current point naming standards.

D. Licenses
   1. Provide product licenses for systems and third party software used in system development, including documentation for all applications, databases, browsers, communications software etc.
   2. Owner shall be the named license holder of all software.

1.5 SUBMITTALS

A. Provide all product licenses for software used on project. List Stanford University as the Owner, including but limited to:
   1. Programming Tools
   2. Third party software installed

B. Submit Programming documentation for local control units. Organize with table of contents and include:
1. All programming wire sheets (provide one typical for terminal devices)
   a. Any two interconnected blocks that are shown on one wire sheet shall be shown with an interconnecting line, with limited use of references.
   b. Any two interconnected blocks that are shown on separate wire sheets shall include references to the connected block, specifically defining the controller, point, point type and wire sheet where the connected block is located.

2. Annotate programming to correlate to specific sequence requirements.

3. Include final (as-installed) programs/configurations to include:
   a. All tuning parameters such as set points, gains, throttling ranges, resets, limits, etc.
   b. Calibration values
   c. File name based on equipment schedule
   d. Results of testing procedures such as diagnostics, functional testing, loop tuning, etc.

C. Control Logic Documentation:
   1. Acceptance of the completeness of Control Logic Documentation shall be determined by Owner’s Representative.
   2. Control logic shall be graphical and annotated to describe how it accomplishes the sequence of operation. Annotations shall be sufficient to allow Owner’s Representative to relate each program component block to corresponding portions of the specified Sequence of Operation.
   3. Include control response, settings, setpoints, throttling ranges, gains, reset schedules, adjustable parameters and limits.
   4. Each wire sheet shall have a title indicating the controller’s identifying designations and the HVAC system controlled.
      a. Any two interconnected blocks that are shown on one wire sheet shall be shown with an interconnecting line, with limited use of references.
      b. Any two interconnected blocks that are shown on separate wire sheets shall include references to the connected block, specifically defining the controller, point, point type and wire sheet where the connected block is located.

1.6 QUALITY ASSURANCE

A. Provide most current revision of software.

B. Provide updates to software and firmware during warranty period

C. Local Control Units shall provide sufficient internal memory for the specified control sequences to be programmed and have at least 25 percent of the memory available for future use.

D. Local Control Units shall provide sufficient spare programming blocks and software points for the control sequences to be programmed and have at least the following capacity available for future use before programming shall be accepted.
   1. 20 % spare PID Loops
   2. 20% Spare Network Variable Outputs (Lon NVO’s or BACnet NVs)
   3. 20% Spare Network Variable Inputs (Lon NVI’s or BACnet NVs)

E. Where the same system is installed in multiple locations, one system must be tuned and the same tuning parameters may be used in like applications

F. Actuator movement shall not occur before the effects of previous movement have had sufficient time (minimum one time constant) to have affected the sensor.
PART 2 - PRODUCTS

2.1 PROGRAMMING INTERFACE TOOL

A. The Programming Interface Tool used to program the free programmable controllers shall be a block programming interface. Users shall drag and drop block objects from a library and connect them to assemble a customized program on a wiresheet.

B. Any Programming Interface Tool that uses line code programming is not acceptable.

C. The program source code, produced by the Programming Interface Tool, shall display all values in US units. LON Variables (SNVT’s, NVI’s, NVO’s, SCPT’s, UCPT’s, etc.) that are natively metric (SI) units may be displayed in metric units in a SINGLE INSTANCE of each variable and may only appear on a wiresheet labeled NVI or NVO. Beyond the allowable single instance, variables shall be converted from metric units to US units and the US units shall be used for all block logic on all other wiresheets.

D. Program source code, produced by the Programming Interface Tool that has variables predominantly displayed in metric units shall not be acceptable, even if some variables have conversion blocks to display in US units.

E. Program source code produced by the Programming Interface Tool that has multiple instances of the same variable in metric units shall not be acceptable.

F. The Programming Interface Tool shall be a plug-in for Tridium Niagara Workbench. A user shall be able to upload and download controllers, by navigating the network from Workbench, selecting a controller, and launching the Programming Interface Tool plug-in.

G. The Programming Interface Tool must be fully compatible with Vykon brand Supervisor Station. Stanford University updates the Vykon-brand supervisor stations the same day that Tridium releases new Niagara software builds. OEM channel partner branded Niagara software builds typically lag behind Tridium’s release of Vykon branded Niagara software. Contractor shall guarantee and demonstrate that the Programming Interface Tool has the functional ability to upload, modify programming and download controllers in a scenario where the Vykon brand Supervisor Station is updated to a new Niagara software revision that is not yet supported by the OEM channel partner of the Programming Interface Tool. Any Programming Interface Tool that is not able to upload, modify programming and download controllers when the Vykon brand Supervisor Station is updated to a new Niagara software revision that is not yet supported by the OEM channel partner of the Programming Interface Tool, shall not be acceptable.

H. A user shall be able to synchronize the on-line and off-line database for each controller, by navigating the network from Workbench, selecting a controller, and launching the Programming Interface Tool plug-in.

I. Any controller on a network, that cannot be uploaded, downloaded, or on-line database synchronized with off-line database, or off-line database synchronized with on-line database, by navigating the network trunk from Workbench, selecting a controller, and launching the Programming Interface Tool plug-in, shall not be acceptable.

J. The program source code, produced by the Programming Interface Tool shall become the property of Stanford University. Owner shall be able to modify, maintain, re-use or furnish code to future consultants or Contractors, without restriction.

K. Provide real-time debugging showing input and output values as the code is executed.

L. Include basic loop control including but not limited to:
   1. Two-position control
   2. Proportional, integral control

M. Applications to include but not limited to:
1. Air Handlers
2. Chilled Water System
3. Hot Water/Steam Systems
4. Energy Recovery Systems
5. Air Terminals
6. Fan Coil Units
7. Dx units
8. Exhaust Systems

PART 3 - EXECUTION

3.1 PROGRAMABLE CONTROLLER SOFTWARE

A. Provide software for programming of controllers.

B. Point names in the controller shall exactly match the point names proxied into Niagara.
   1. The allowable exception is the standard of dropping Lon Network Variable prepends i.e. NVI, NVO, NCI, etc. for example “nviSetpoint” in controller programming would be proxied into Niagara as “Setpoint”.

C. All point reference tags names on wiresheets shall exactly match the point names that they reference.

D. The first sheet shall be titled SOO and include revision history and sequence of operations. Revision History is a Summary of major revisions to code and authors:
   1. Created by: Name of Programmer
   2. Company: Name of Controls Company
   3. For: Name of original Stanford Project this code was created for.
   4. Created on: Date of original Stanford Project this code was created for.
   5. Based Upon: Manufacturer Source Code i.e. Distech ECLAVAV_SD_PROPHTG_SI_2.0.3
   6. Edited by: Name of Programmer(s) that made modifications to the original code
   7. Company: Controls Company that made modifications to the original code
   8. For: Name of current Stanford Project this code was modified for.
   9. Edited on: Date of current Stanford Project this code was modified for.
   10. Summary of Major revisions
   11. Sequence of Operations

E. Wiresheet I/O tabs shall list points into and out of controller with no wiresheet logic on the I/O sheet.
   1. Example of I/O sheet
F. Control logic shall be graphical and annotated to describe how it accomplishes the sequence of operation. Annotations shall be sufficient to allow Owner’s Representative to relate each program component block to corresponding portions of the specified Sequence of Operation.

1. Example of level of detail for block logic annotation

G. Provide all final job specific software configuration documentation, data files, application level files and code utilized on project.

H. Include all labor to properly install, tune, debug, checkout and commission the installed system.
I. Tune all control loops to obtain the fastest stable response without hunting, offset or overshoot. Tuning constants shall be set so that continuous oscillation of actuators does not occur. A steady state shall be achieved.

J. Annotate, wire sheets, blocks, etc. to facilitate future maintenance and modifications.

K. Program all controllers for stand-alone control capability and to meet sequences specified.

L. Stand-alone control capability. Upon loss of communication to JACE or Supervisor, Local Control Unit shall:
   1. Command equipment to Occupied Mode.
   2. Control to Occupied Mode Setpoints.

M. Provide software as utilized for complete operating system.
   1. Include software for engineering and application modifications.
   2. Include software data/programming files for each separate controller, programming code and documentation.

END OF SECTION