Mapping IEC 61131-3 Directly Represented Variables to CANopen
This paper

- Maps CANopen variables hierarchy to IEC-61131-3 “Directly Represented Variables”
- Could help in defining some uncovered “manufacturer specific” aspects of DS-405
- Proposes some algorithms for implementation
About IEC 61131-3

- IEC 61131 rules PLC programming art

- Languages:
  - 3 graphical
    - SFC
    - FBD
    - LD
  - 2 textual
    - ST
    - IL
About IEC 61131-3

- PLC Programmers organize their work through “Program Organization Units” (POU)

- POUs can be:
  - Functions (no instance, idempotents)
  - Function Blocks (instantiated, internal state)
  - Programs (instantiated, internal state, I/O access)
About IEC 61131-3

- Usually, function blocks are instantiated in Programs in order to be “connected” to I/Os.
Direct Representation

- “Directly represented variables” represent I/Os
- Locations are integers separated by dots
IEC 61131-3 and CANopen

- CANopen PLC users have to define an intermediate nomenclature between Programs interface and CANopen.

<table>
<thead>
<tr>
<th>Variable Panel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>IN1</td>
</tr>
<tr>
<td>2</td>
<td>IN2</td>
</tr>
<tr>
<td>3</td>
<td>IN3</td>
</tr>
<tr>
<td>4</td>
<td>IN4</td>
</tr>
<tr>
<td>5</td>
<td>IN5</td>
</tr>
<tr>
<td>6</td>
<td>SRL</td>
</tr>
</tbody>
</table>

%IX0.1

- NodeID: 0x10
- Index: 0x2000
- SubIndex: 0x01
IEC 61131-3 and CANopen

- File format for this nomenclature is not specified

<table>
<thead>
<tr>
<th>Variable Panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

%IX0.1
NodeID: 0x10
Index: 0x2000
SubIndex: 0x01

PLC Program

Network description & configuration

DS-306 EDS & DCF

DS-405 nodelist.cpj
Proposal

- Express nomenclature in variables locations

<table>
<thead>
<tr>
<th>Variable Panel</th>
<th>#</th>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IN1</td>
<td>Local</td>
<td>BOOL</td>
<td>%IX0.1</td>
<td>NodeID: 0x10</td>
</tr>
<tr>
<td>2</td>
<td>IN2</td>
<td>Local</td>
<td>BOOL</td>
<td>%IX0.2</td>
<td>Index: 0x2000</td>
</tr>
<tr>
<td>3</td>
<td>IN3</td>
<td>Local</td>
<td>BOOL</td>
<td>%IX0.3</td>
<td>SubIndex: 0x01</td>
</tr>
<tr>
<td>4</td>
<td>IN4</td>
<td>Local</td>
<td>BOOL</td>
<td>%IX0.4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IN5</td>
<td>Local</td>
<td>BOOL</td>
<td>%IX0.5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SR1</td>
<td>Local</td>
<td>SR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Panel</th>
<th>#</th>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DigitalOut</td>
<td>Local</td>
<td>BYTE</td>
<td>%Q8503254.0.16.250088</td>
<td>CANopen</td>
</tr>
<tr>
<td>2</td>
<td>DigitalIn</td>
<td>Local</td>
<td>BYTE</td>
<td>%Q8503254.0.32.245761</td>
<td>CANopen</td>
</tr>
<tr>
<td>3</td>
<td>AnalogOut1</td>
<td>Local</td>
<td>MYTYPE</td>
<td>%QW503254.0.16.2551</td>
<td>CANopen</td>
</tr>
<tr>
<td>4</td>
<td>AnalogOut2</td>
<td>Local</td>
<td>MYTYPE</td>
<td>%QW503254.0.16.2551</td>
<td>CANopen</td>
</tr>
<tr>
<td>5</td>
<td>AnalogOut3</td>
<td>Local</td>
<td>INT</td>
<td>%QW503254.0.16.2551</td>
<td>CANopen</td>
</tr>
<tr>
<td>6</td>
<td>AnalogIn1</td>
<td>Local</td>
<td>INT</td>
<td>%DW503254.0.32.25601</td>
<td>CANopen</td>
</tr>
<tr>
<td>7</td>
<td>AnalogIn2</td>
<td>Local</td>
<td>INT</td>
<td>%DW503254.32.25601</td>
<td>CANopen</td>
</tr>
</tbody>
</table>

Directly represented variables
Numbering scheme

- Variable location explicitly represent the path in the CANopen hierarchy to access any object.

- Example: Remote First DS-401 Read Input Byte

<table>
<thead>
<tr>
<th>Direction</th>
<th>Data Size</th>
<th>Protocol</th>
<th>CanOpen</th>
<th>Bus-ID</th>
<th>Node-ID</th>
<th>Transmit Type</th>
<th>Index</th>
<th>SubIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>Byte</td>
<td></td>
<td></td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>6000h</td>
<td>1</td>
</tr>
</tbody>
</table>

%IB503254.0.2.1.24576.1
Numbering scheme

- Depending on count of integers, location may represent remote or local CANopen variables
- Example: Read access to local Error Register

<table>
<thead>
<tr>
<th>Direction</th>
<th>Data Size</th>
<th>Byte Protocol</th>
<th>CANopen Bus-ID</th>
<th>Index</th>
<th>SubIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>In</td>
<td>Byte</td>
<td>CanOpen</td>
<td>0</td>
<td>1001h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
%IB503254.0.4097.0
Numbering scheme

- Local representations can also be used to implicitly declare arbitrary new OD entry
- Example: New Manufacturer Specific entry

---

**Direction:** In  
**Data Size:** Byte  
**Protocol:** CanOpen  
**Bus-ID:** 0  
**Index:** 2001h  
**SubIndex:** 0

%IB503254.0.8193.0
Where it helps

- Avoids network reconfiguration steps when revamping CANopen machines
- Eases portage of PLC programs through different CANopen controller brands
- Provides long-term re-usability, independently of PLC solutions life time.
Implementation

- CANopen configuration tool
  - Change default variables location calculation
  - Generate PLC node configuration from
    - declared variable locations
    - network description files
    - optional additional DCF
Implementation

- PLC workbench
  - Extract variable locations at compile time
  - Pass locations list to CANopen configuration tool
PDO mapping

• Remote variables access are mapped to PDO
• Establishing arbitrary PDO mapping implies to:
  – be a master node
    • slaves can only access/declare variables local OD
  – define new PDO mapping in each accessed node
    • may be stored in master's concise DCF
  – declare corresponding mapped PLC's OD entries
  – define corresponding PDO mapping in PLC's OD
PDO mapping

- More than one mapping is possible for one set of locations
  - Some side effect may appear on some implementations
- Some algorithm may be defined to fix:
  - Mapped variable layout in node's PDOs
  - Pre-configured PDO remapping politic
  - Mapped variables order in master OD
Ergonomic thoughts

- Compensate hard to read representation
  - Hexadecimal display option
    - %IB503254.0.16.1.24576.1 => %IB503254.0.10h.1.6000h.01h
  - Tooltip texts showing CANopen OD entry name
  - Point and click variable to jump to configuration tool

- Smart manipulation of locations
  - Variables substitution across programs
  - Programs and EDS import and re-mapping
Questions