IEC 61131-3 Variable Usage and Initialization in the RTAC

Chris Bontje

INTRODUCTION

The SEL-3530 Real-Time Automation Controller (RTAC) offers a powerful programming library that follows IEC 61131-3 and allows for the following languages to be used:

- Continuous function chart (CFC).
- Ladder logic diagram (LD).
- Structured text (ST).

Each of these programming languages allows the user to enter customized logic functions in the RTAC to perform a variety of automated tasks. The programming languages themselves can use different data signals for inputs and outputs, including the following:

- RTAC tag data (system tags, client/server tags, and virtual tags).
- Local variables (defined within the user program space).
- Global variables (defined in a global variable list [GVL], and possibly retained in memory).

PROBLEM

With so many different possibilities for program inputs and outputs, it can be difficult for an RTAC programmer to determine which type of tag or variable to use and where to use it. This application note discusses basic guidelines for the creation and use of the different tag types that are encountered by a programmer and the initialization of each tag type.

While local variables are only usable within specific user programs, RTAC tags and global variables can be shared between multiple programs and can be used to perform complex data exchanges. To control program flow at a default state, it is often desirable to have a default startup value to which tags and variables are initialized when the RTAC starts. This ensures that no invalid program paths are followed due to incorrect input values.

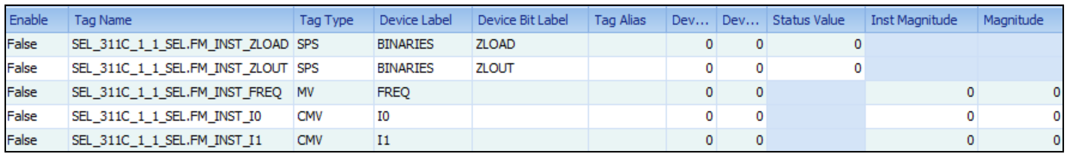
RTAC TAG USAGE AND INITIALIZATION

RTAC tags used in an IEC 61131-3 program are usually owned by either a client or server device or are a member of a virtual tag list. All tags belonging to these categories are created using semistandardized IEC 61850 data types that typically include the raw data value (Boolean or analog), along with auxiliary information such as time stamps and quality components.

The following are some examples of these tags (full documentation is available in Appendix B of the ACSELERATOR RTAC® SEL-5033 Software Instruction Manual):

- SPS – single point status (Boolean on/off data).
- MV – measured variable (analog and magnitude only).
- CMV – complex measured value (analog, magnitude, and angle).
- DNPC – DNP controllable single point (DNP3 control, includes operSPC data types for each DNP3 control code type, such as trip and close).
- RBC – SEL remote bit control (set, clear, and pulse operSPC types included).
- INS – signed integer.

These tag types are typically assigned a default startup value of 0 (for numeric-style data) or FALSE (for Boolean-style data), unless otherwise specified in the tag configuration properties (as shown in Figure 1).



Enable	Tag Name	Tag Type	Device Label	Device Bit Label	Tag Alias	Dev...	Dev...	Status Value	Inst Magnitude	Magnitude
False	SEL_311C_1_1_SEL.FM_INST_ZLOAD	SPS	BINARIES	ZLOAD		0	0	0		
False	SEL_311C_1_1_SEL.FM_INST_ZLOUT	SPS	BINARIES	ZLOUT		0	0	0		
False	SEL_311C_1_1_SEL.FM_INST_FREQ	MV	FREQ			0	0		0	0
False	SEL_311C_1_1_SEL.FM_INST_I0	CMV	I0			0	0		0	0
False	SEL_311C_1_1_SEL.FM_INST_I1	CMV	I1			0	0		0	0

Figure 1 Tag Configuration Properties

Note that the Boolean-style tags are programmed with a non-zero default **Status Value**, whereas the numeric-style tags can be set up with both default **Instantaneous Magnitude** (pre-dead-band processing) and **Magnitude** (post-dead-band processing) values. Upon establishing communication with an end device or tag processor operations, RTAC tags are assigned real-world field values and are no longer considered at a default state. RTAC tags of this type are exposed to the RTAC HMI environment and count toward the total tag limit associated with a particular RTAC device type. As of December 2017, this is 5,000 for the SEL-3505, 100,000 for the SEL-3555, and 25,000 for all other RTAC types.

LOCAL AND GLOBAL VARIABLE USAGE AND INITIALIZATION

Local and global variables are typically not associated directly with any one client or server device, but rather are associated to user-entered IEC 61131-3 programming that is entirely customizable, depending on user programming preferences and style. Local and global variables are typically assigned one of the IEC 61131-3 fundamental data types, but can also use the tag data types mentioned previously. Examples of fundamental IEC 61131-3 data types include the following (full documentation is available in the ACSELERATOR RTAC Instruction Manual):

- BOOL – Boolean data (false or true).
- REAL – 32-bit real or floating-point data (numeric, with decimal point precision).
- INT – 16-bit signed integer data (numeric, with a range of –32,768 to +32,767).

Local and global variables are initialized to default values according to the following three principles (in order of priority as defined by IEC 61131-3 standard):

- If the RETAIN option is enabled, the previous value of a variable is stored in nonvolatile RAM (NVRAM) and retained through power cycles and settings changes.
- Initialization value, if present in variable declaration.
- Default value derived from variable data type.

The definition of local program variables is typically similar to the following pattern.

```
VAR
        <VARIABLE_NAME>:<VARIABLE_DATA_TYPE><:=OPTIONAL_INITIALIZATION_VALUE>;
END_VAR
```

For example, a program may have a local variable definition section similar to the following.

```
VAR
        EVENT_TYPE:MV;
        TEMP_CALC:REAL;
        OPERATOR_RESET:BOOL:=TRUE;
END_VAR
```

EVENT_TYPE and **TEMP_CALC** are initialized to default data type values of 0 (because MV and REAL data types are considered to be numeric style), and **OPERATOR_RESET** is preloaded with an initialization value of TRUE.

Global variables are defined similarly in a GVL, except the **VAR_GLOBAL** keyword is used rather than **VAR** in the variable definition header field.

```
VAR_GLOBAL
        <VARIABLE_NAME>:<VARIABLE_DATA_TYPE><:=OPTIONAL_INITIALIZATION_VALUE>;
END_VAR
```

A GVL is added to the RTAC program to define global variables. It may be formatted as follows.

```
VAR_GLOBAL
        SYSTEM_STATE:REAL;
        FIRST_RUN:BOOL:=FALSE;
END_VAR
```

These variables are used in programs by referring to their GVL name (e.g., MyGVL) and the variable name as defined in the GVL. For example, MyGVL.SYSTEM_STATE refers to the REAL variable defined previously (initialized to 0) and MyGVL.FIRST_RUN refers to the BOOL variable defined previously (which was force-initialized to FALSE).

When creating an analog tag (MV, CMV, INS, and so on) associated with a client device, server device, or virtual tag list, there are several additional parameters that are set to default values as part of the system initialization. These include values such as dead-band and zero-dead-band limits, minimum and maximum analog limits, and alarm limits (LOLO, LO, HI, and HIHI).

Note that if MV or CMV data types are defined in custom program code as local or global variables, then these additional parameters must be initialized if the tag will eventually be used as part of a tag processor source expression that will automatically check against the parameters when processing it. If the parameters are not initialized, it is possible for several quality flags to be set in the corresponding destination tag, indicating issues such as OUT OF RANGE.

GLOBAL RETAINED VARIABLE INITIALIZATION

Global retained variables can be used by a programmer to preserve variable states through RTAC power cycles or even after revised settings changes. One important note concerning global retained variables is that due to the IEC 61131-3 variable initialization priorities defined previously, variables that are programmed with the RETAIN tag in place are not necessarily initialized to default values (determined by data type) on the first run, but rather are assigned the values residing in the memory location their NVRAM address aligns with. This can create problems if that memory location was previously used on a past execution for some other purpose and contains random leftover data values. Therefore, the first time settings using retained variables are sent to the RTAC, they must be sent with an optional flag to initialize these retained variables to safe values.

To define a set of global retained variables, follow a definition format similar to regular global variables, but use the RETAIN qualifier in the variable declaration as follows.

```
VAR_GLOBAL RETAIN
        <VARIABLE_NAME>:<VARIABLE_DATA_TYPE>:=<REQUIRED INITIALIZATION VALUE>;
END_VAR
```

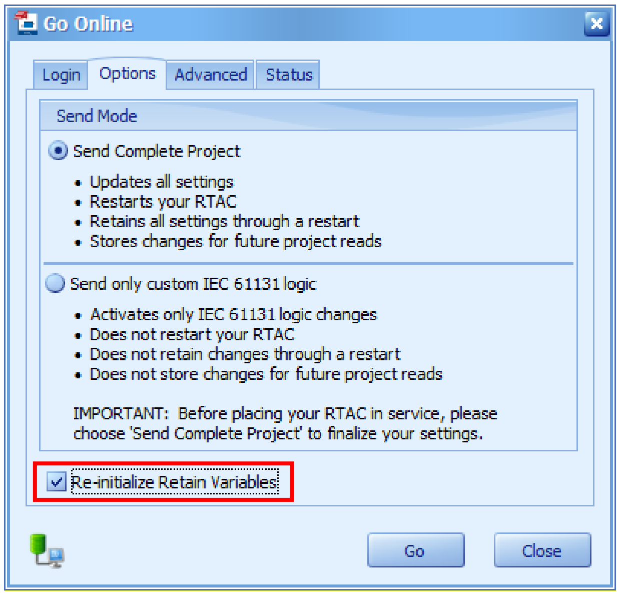
A global retained variable list (named MyGVL in this example) is added to the RTAC program, and may be formatted as shown in the following example.

```
VAR_GLOBAL RETAIN
        SYSTEM_STATE:REAL:=123.45;
        FIRST_RUN:BOOL:=TRUE;
END_VAR
```

To initialize system state and first run values to safe defaults (123.45 and TRUE in the above example), use the **Go Online** feature within ACSELERATOR RTAC to connect to the RTAC. Before issuing settings, visit the **Options** tab. On the **Options** tab, as shown in Figure 2, check the **Re-initialize Retain Variables** box and then click **Go** to send the settings to the RTAC.

After sending the settings to the RTAC, confirm the initialized variable contents by clicking on the retained global variable list and reviewing the values while online with the RTAC.

NOTE: Any time the global retained variable list is manipulated (variables are added, the order of variables is changed, and so on), use the **Re-initialize Retain Variables** option to set all variables to safe defaults. Otherwise, the contents of the variables will be unpredictable due to shifting memory locations.



(●) Send Complete Project

- Updates all settings
- Restarts your RTAC
- Retains all settings through a restart
- Stores changes for future project reads

( ) Send only custom IEC 61131 logic

- Activates only IEC 61131 logic changes
- Does not restart your RTAC
- Does not retain changes through a restart
- Does not store changes for future project reads

IMPORTANT: Before placing your RTAC in service, please choose 'Send Complete Project' to finalize your settings.

[✓] Re-initialize Retain Variables

Figure 2 Re-initialize Retain Variables

SCHWEITZER ENGINEERING LABORATORIES, INC.
2350 NE Hopkins Court • Pullman, WA 99163-5603 USA
Tel: +1.509.332.1890 • Fax: +1.509.332.7990
www.selinc.com • info@selinc.com