SIEMENS

SIMATIC
S7-200 SMART
Update to the S7-200 SMART System Manual, edition 10/2013

Product Information

Overview to documentation update S7-200 SMART

In spite of efforts to ensure the accuracy and clarity in the product documentation, some of the pages in the *S7-200 SMART System Manual* contain information that is incomplete, incorrect or misleading.

This document contains the following updates

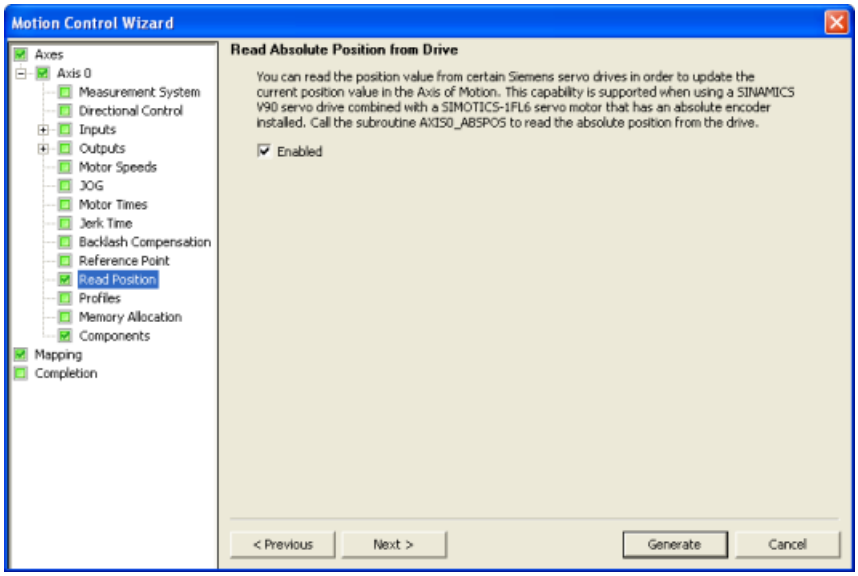
Using the AXISx_ABSPOS subroutine to read the absolute position from a SINAMICS V90 servo drive

The following sections provide additional information about the how to use the AXISx_ABSPOS subroutine in your project to read the absolute position from a SINAMICS V90 servo drive:

A5E32938409-AB, 01/2015

AXISx_ABSPOS subroutine

The STEP 7-Micro/WIN SMART's Motion wizard creates the AXISx_ABSPOS subroutine when you select the "Enabled" check box in the "Read Absolute Position from Drive" dialog and click the "Generate" button to construct the wizard-generated subroutines:



The subroutine interface and parameter description for the AXISx_ABSPOS subroutine is shown below:

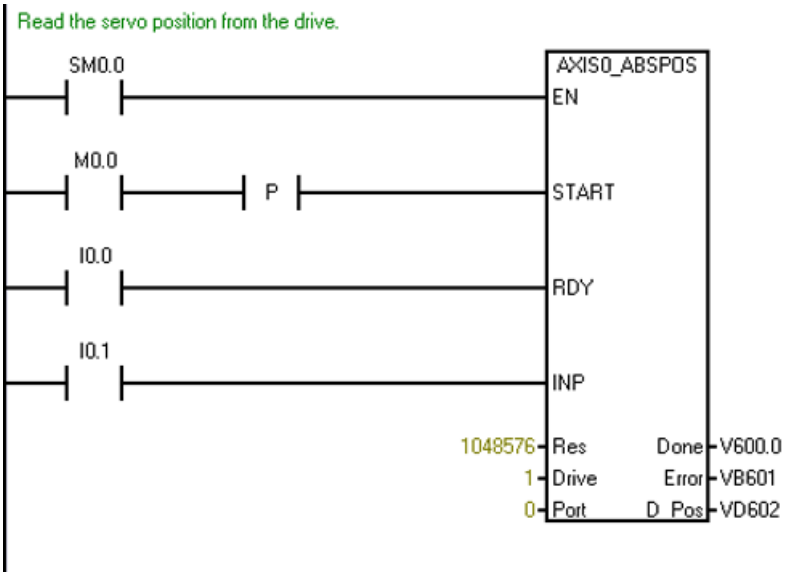
LAD / FBD	STL
AXIS0_ABSPOS EN START RDY INP Res Done Drive Error Port D_Pos	`CALL AXISx_ABSPOS, START, RDY, INP, Res, Drive, Port, Done, Error, D_Pos`

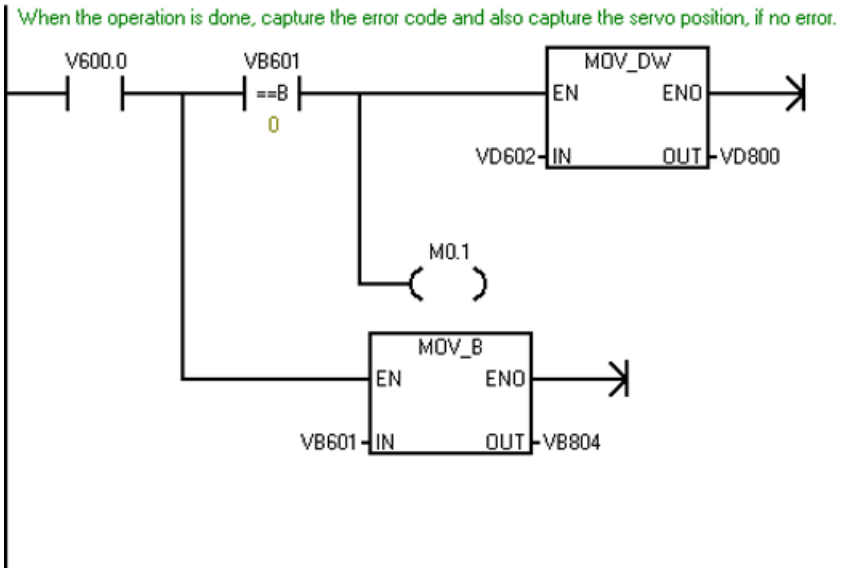
Inputs / outputs	Data type	Description
EN	BOOL	Turn on the EN input bit to enable the subroutine. Ensure that the EN bit stays on until the DONE bit signals that the execution of the subroutine has completed.
START	BOOL	Turn on the START input parameter to obtain the current absolute position from the specified servo drive. To ensure that only one operation to read the current position is performed at a time, use an edge detection element to pulse the START parameter on.
RDY	BOOL	The RDY input parameter indicates the readiness state of the servo drive, which is typically provided by a digital output signal from the drive. This subroutine will read the absolute position from the drive only if this parameter is on.
INP	BOOL	The INP input parameter indicates the standstill state of the motor, which is typically provided by a digital output signal from the drive. This subroutine will read the absolute position from the drive only if this parameter is on.

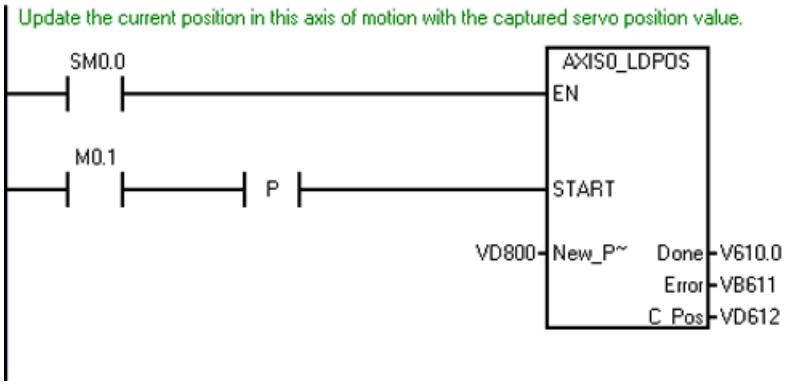
Inputs / outputs	Data type	Description
Res	DINT	The Res input parameter must be set to the resolution of the absolute encoder connected to your servo motor. For example, the single turn resolution of a SIMOTICS S-1FL6 servo motor with absolute encoder is 20 bits, so you would enter a value of 1048576 for this motor/encoder combination.
Drive	BYTE	Set the Drive input parameter to match the RS485 address of the servo drive to be accessed by this subroutine. The valid drive address range is 0 to 31.
Port	BYTE	Set the Port input parameter to designate the CPU port to be used to communicate with the servo drive: • 0: Onboard RS485 port (Port 0) • 1: Optional RS485/RS232 signal board (Port 1)
Done	BOOL	The subroutine turns on the Done output parameter when the subroutine's work is complete.
Error	BYTE	The subroutine writes the result code generated from this execution of the subroutine into the Error output parameter. Possible error codes are: • 0: no error • 128: subroutine busy or START input not set • 135: parameter error • 136: execution error (in other words, unable to communicate with the drive)
D_Pos	REAL, DINT	The subroutine writes the current absolute position returned by the servo drive into the D_Pos output parameter. • This parameter is valid only when the Done parameter = ON and the Error parameter = "no error". • The position value is REAL if the configured motion axis measurement system = "engineering units" and is DINT if the configured motion axis measurement system = "relative pulses".

AXISx_ABSPOS and AXISx_LDPOS subroutines usage examples

The absolute position is valid only after successful completion of the AXISx_ABSPOS subroutine (Done parameter = ON and Error parameter = "no error") when executed with the START parameter on. Since the Error and D_Pos parameters revert to default values when the subroutine is executed with the START input off, you must include instructions in your program to capture the valid absolute position value after completion of the subroutine.







Interconnections

Digital I/O

Refer to the section "Connection examples with PLCs" in the *SINAMICS V90 / SIMOTICS S-1FL6 Operating Instructions* document to find wiring diagrams for connection of the suggested digital control signals between an S7-200 SMART CPU and a V90 servo drive.

Communications

The AXISx_ABSPOS subroutine obtains the position data from the drive using serial communications on the RS485 link between the two devices. Therefore, connect a cable between the RS485 port on the S7-200 SMART CPU (or optionally the S7-200 SMART CM01 signal board) and the RS485 port on the V90 servo drive.

Refer to the appropriate sections of the *S7-200 SMART System Manual* and the *SINAMICS V90 / SIMOTICS S-1FL6 Operating Instructions* documents for descriptions of the RS485 ports on the S7-200 SMART CPU and V90 servo drive.

Commissioning

Control mode

"PTI" mode is the drive control mode setting that allows movement speed and distance to be controlled from an external pulse train. The default control mode in the V90 servo drive is basic "PTI" mode, but you can check the mode setting by reading the value of parameter "p29003" and verifying that the value = "0". It is possible to use compound control modes (PTI/S and PTI/T) with the pulse train output from the S7-200 SMART CPU . These are advanced features and are not within the scope of this document. For assistance with these features, refer to the *SINAMICS V90 / SIMOTICS S-1FL6 Operating Instructions* document.

Setpoint pulse input channel

For correct operation with the digital outputs of the S7-200 SMART CPU, you must select the "24V single end pulse train input" setting for the setpoint pulse input channel parameter (parameter "p29014" = 1) in the V90 servo drive.

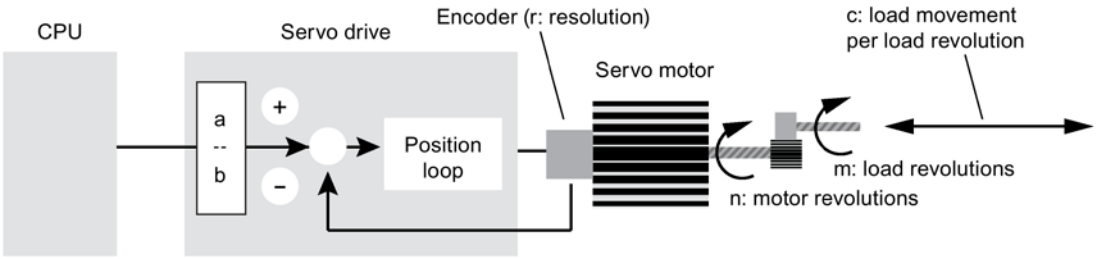
Setpoint pulse train input format

Ensure that the CPU's axis of motion output phasing and polarity settings (established in the "Directional Control" dialog of the STEP 7-Micro/WIN SMART Motion wizard) are consistent with the V90 servo drive's setpoint pulse train input format setting (parameter "p29010").

Common engineering units basis

When using a motion axis on the S7-200 SMART CPU to control the movement speed and distance of a servo motor, you must establish a common definition of the engineering units between the axis of motion (CPU) and the drive.

The following diagram shows the elements of a motion system:



To establish a common engineering unit definition between the CPU and the servo drive, you must consider the following motion system variables when commissioning your system:

- **Electronic gearing**: In the V90 servo drive, the "a" and "b" values determine the drive's electronic gear ratio, a feature that allows a frequency conversion on the pulse train issued from the CPU. Since the maximum pulse frequency issued from an axis of motion in the S7-200 SMART CPU is 100 kHz, while the encoder resolution of SIMOTICS S-1FL6 servo motors installed with absolute encoders is 2^20 pulses per revolution, use of the drive's electronic gear feature is likely, in many applications, to achieve higher motor speeds. For example, to achieve a 10x increase in the setpoint pulse frequency within the servo drive compared to the frequency of the CPU pulse train supplied to the drive, then you must set the electronic gear ratio to "10:1".

  In the V90 servo drive, setting parameter "p29012[0]" establishes the numerator of the electronic gearing ratio ("a"), while setting parameter "p29013" establishes the denominator of the ratio ("b"). Also, when using electronic gearing, set the parameter "p29011" value to "0". The valid range for the electronic gear ratio (a / b) in the V90 servo drive is between "0.02" and "200".

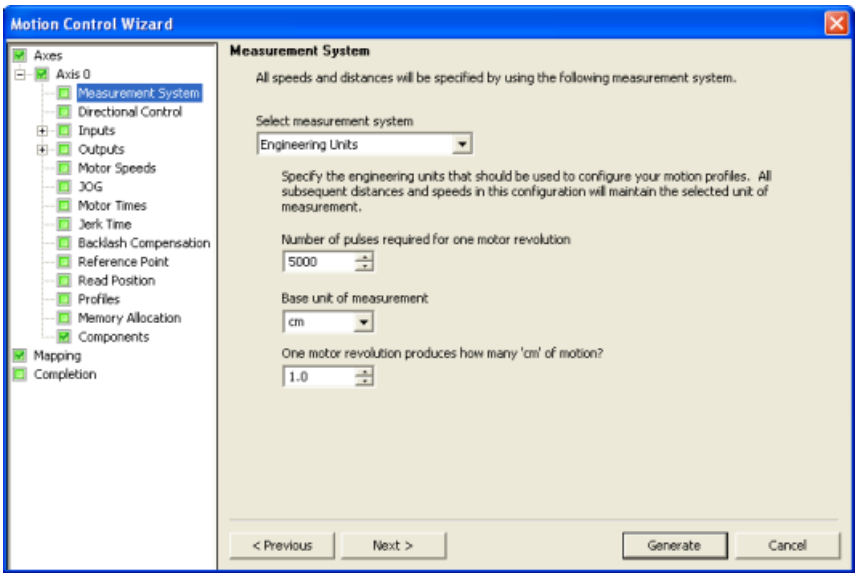
  Refer to the "Electronic Gear Ratio" section of the *SINAMICS V90 / SIMOTICS S-1FL6 Operating Instructions* document for more information.

- **Mechanical factors**: The "m" and "n" values establish the mechanical relationship between a load revolution and a motor revolution, applicable when a gearing mechanism is used. When the V90 servo drive is in "PTI" control mode, its internal mechanical gearing ratio parameters are fixed at 1:1, but the physical "m" and "n" values are important in establishing the correct engineering unit conversion factors for the axis of motion, as shown below.

  The "c' value establishes the relationship between load movement, in the specified engineering unit, and load revolutions. "20 cm of load movement per load revolution" and "360 degrees of load movement per load revolution" are examples of this conversion factor.

- **Encoder resolution**: The "r" value is the resolution of the absolute encoder in your servo motor. As stated above, the encoder resolution of SIMOTICS S-1FL6 servo motors installed with absolute encoders is 2^20 pulses per revolution or "1048576". When the V90 servo drive is paired with a motor containing an absolute encoder, the drive automatically detects the encoder type and obtains its resolution. However, in your program, you must specify this resolution value in the AXISx_ABSPOS subroutine's "Res" input parameter and also in one of the engineering unit conversion factor calculations shown below.

- **Measurement system settings in the Motion wizard**: When using the STEP 7-Micro/WIN SMART Motion wizard to configure the measurement system for a CPU axis of motion, there are three conversion settings that must be specified. The first setting relates CPU pulses to motor revolutions, the second setting establishes the base engineering unit name, and the third setting relates motor revolutions to load movement:



  - **Setting #1**: **"Number of pulses required for one motor revolution"**: This setting defines the relationship between CPU pulses and motor revolutions. The relevant equation that yields the correct value for this setting is:

    (1) Number of pulses required for one motor revolution = r * (b / a)

    where, "r" = encoder resolution, expressed as encoder pulses per motor revolution,
    "a" and "b" = electronic gearing (E-gear) ratio parameters ("a" = value of V90 parameter "p29012[0]" and "b" = value of V90 parameter "p29013")

    For example, if the desired E-gear ratio is 128:1 and the motor's absolute encoder resolution is 2^20 or 1048576, then:

    "Number of pulses required for one motor revolution" = **1048576 * (1 / 128) = 8192**

  - **Setting #2**: **"Base unit of measurement"**: This setting establishes the base engineering unit name for speed and distance settings throughout the Motion wizard. To avoid confusion, the selection should match the engineering unit relevant at the load. For example, if load movement and speed is to be expressed in "cm" and "cm / second", then the "cm" selection should be chosen for this setting.

  - **Setting #3: "One motor revolution produces how many "xxx" of motion?"**: This setting defines the relationship between motor revolutions and load movement in the defined engineering unit (for example. cm and degrees). The relevant equation that yields the correct value for this setting is:

    (2) One motor revolution produces how many "xxx" of motion = c * (m / n)

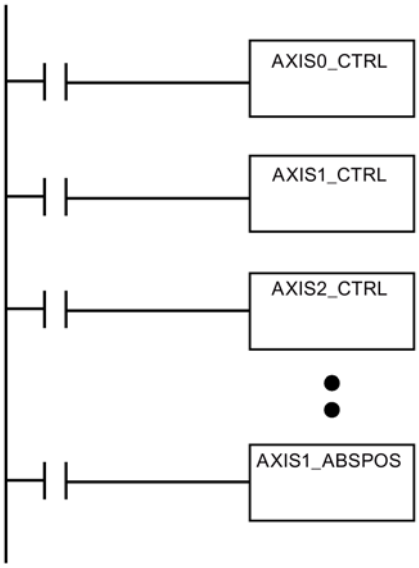
    where, "c" = load movement (in the defined engineering unit) per load revolution,
    "m/n" = external gearing ratio expressed as load revolutions per motor revolution

    For example, if the mechanical gear ratio is 1:2 and the load movement per load revolution is 10 cm, then:

    "One motor revolution produces how many cm of motion" = **10 * (1 / 2) = 5**

Important facts to know

- Do not call the AXISx_ABSPOS subroutine from within an interrupt routine or from a subroutine called within an interrupt routine.
- If you have configured multiple axes of motion in your CPU project, ensure that the AXISx_CTRL subroutines for all axes are executed prior to executing the first AXISx_ABSPOS subroutine for any axis. The AXISx_CTRL subroutine contains code to initialize the V memory area used commonly by all instances of the AXISx_ABSPOS subroutine in your program to manage the communications with the servo drive.



- If you configure your motion axis measurement system to the "relative pulses" setting instead of the "engineering units" setting, you can still use the AXISx_ABSPOS subroutine to return position information from the V90 servo drive. Note, however, that the position value returned in the "D_pos" parameter of the subroutine will then be of type DINT and is the actual position value reported by the servo drive (there are no engineering unit conversions performed on the value).

Using USS communications with the Siemens V90 servo drive

Siemens designed the USS communications library for use with Siemens general purpose drives such as the Siemens Micromaster series. Siemens does not intend for the USS communications library to be used with special purpose drives such as the V90 servo drive. The control interface of the V90 servo drive is different from that of a general purpose drive. For this reason, do not use the USS communications library with the V90 servo drive.

Characteristics of I0.6 and I0.7 in ST20 and ST30 CPUs

In the ST20 and ST30 DC/DC/DC CPU models, I0.6 and I0.7 are fast, wide-ranging inputs similar to I0.0 to I0.3. They are NOT Sink/Source (IEC Type 1 sink) as erroneously stated in the manual.

In addition, the filter times are individually selectable in the ST30, SR30, ST40, SR40, CR40, ST60, SR60 and CR60 CPU models for points I1.6 and greater, not I0.6 and greater as erroneously stated in the S7-200 SMART V2.0 System Manual.

Additional information about the GET/PUT wizard

The Get/Put wizard supports communication between multiple remote CPUs. The following information supplements the information that the S7-200 SMART system manual and the STEP 7-Micro/WIN SMART online help provide.

Get/Put wizard memory usage

When you have set a non-zero timeout value, the Get/Put wizard allocates bits at V0.1, V0.2, and V0.3 regardless of where you configure the memory range.

As a workaround, you can configure VB0 to be the base memory address range in the wizard or set the timeout value to 0.

Timeout parameter of the NET_EXE subroutine

The Get/Put wizard generates a NET_EXE subroutine that you can call from your main program to enable network communication. The NET_EXE subroutine has a Timeout parameter. If you set the timeout parameter to 0, the instruction executes and enables network communications with no time delay. A positive integer timeout value indicates a time delay in seconds before the instruction executes. If you set the Timeout value to be lower than the number of seconds than the default timeout of an Ethernet connection (about 30 seconds), then the NET_EXE instruction will never execute if a device is

offline. The timeout value will expire before the CPU determines that a device is offline. For this reason, you should either use a Timeout value of 0 (no timer) or a value greater than 40 seconds.

Number of supported remote CPU IP addresses

The Get/Put wizard supports 16 operations. The STEP 7-Micro/WIN SMART online help incorrectly states 24 operations. You can configure each of the 16 operations to be to a unique IP address; therefore, you can use Get/Put to communicate to up to 16 CPU IP addresses. (You can also configure multiple operations to the same IP address.)

Effect of communication timeout between partners on other partners

If you have configured two or more remote partner connections with the GET/PUT wizard, a communication failure between one pair causes a delay in communication with the other pair or pairs. The S7-200 SMART CPU provides a 30-second timer between communication partners. The thirty seconds must expire before the CPU determines that a communication loss has occurred and proceeds to attempt communication with the next set of partners. The S7-200 SMART CPU processes communication between multiple pairs of remote communication partners sequentially.

String copy error in protected block

If your program calls STR_COPY from a password-protected POU, and passes a string literal such as 'ABC' in the IN parameter, the instruction does not output the string correctly at the OUT parameter.

You can avoid the problem by using one of the following methods:

- Remove the password protection from the POU that calls STR_CPY. You can then call this POU from another password-protected POU if you need to protect the block.
- Assign the string literal to a memory address and pass that addressed location to the IN parameter of STR_CPY.

Jogging the axis of motion from the motion control panel sometimes does not terminate

You configure a jog interval for an Axis of Motion in the motion control wizard of STEP 7-Micro/WIN SMART and you can initiate a Jog+ or Jog- motion from the motion control panel.

In rare instances, clicking the Jog+ or Jog- button starts a movement of the Axis of Motion, but the motion does not terminate at the end of the configured jog interval. The motion control panel of STEP 7-Micro/WIN SMART occasionally does not send the communication message that terminates the jog interval. The axis continues to move until some other event stops it.

If this problem occurs, click the STOP button on the Motion Control Panel to terminate the jog for the motion axis.

Note

This problem does not affect jogging an Axis of Motion through the user program in RUN mode. The problem exists only with the behavior of the motion control panel when the CPU is in STOP mode.

Guidelines for backlash compensation for an axis of motion

If your axis of motion requires backlash compensation, set the backlash compensation to be a value greater than 1. A value of 1 is not sufficient for the firmware to apply as a backlash compensation value.

Problems with motion profile direction of travel

A motion profile specifies direction of travel for up to three axes of motion. The S7-200 SMART CPU incorrectly reports a direction change error when execution of a motion profile starts, and all of the following conditions are true:

- The motion profile contains more than one step.
- The selected profile operation mode is Absolute Position.
- The direction of travel for the profile is negative.
- The current axis position at the time that the profile execution starts is equal to the confiured ending position of the first step.

To avoid this error, ensure that the ending position of the previous axis movement does not coincide with the ending position of the first step of the next executing profile if the configured direction of travel for the next profile is negative.

Memory overlap errors with memory allocations in wizards

The STEP 7-Micro/WIN SMART wizards allocate memory in blocks for the wizard-generated data storage. Sometimes, however, errors occur with memory addresses at the boundaries of the memory addresses that you specify. Sometimes a memory overlap occurs between one wizard memory allocation and another. Memory overlaps happen with the Motion wizard with multiple axes, the Text Display wizard, and with the PID wizard with multiple loops.

You can use the cross-reference table to identify whether any memory overlaps exist. If so, edit the memory addresses of overlapping objects to eliminate the overlap.

Analog limit values

The S7-200 SMART CPU reports "lower limit exceeded" and "upper limit exceeded" errors for analog values. The thresholds for these errors are as follows:

- Lower limit exceeded: value is less than -32512
- Upper limit exceeded: value is greater than +32511

Also note the over and under range regions:

- Under range: value is between -32512 and -27648
- Over range: value is between +27649 and +32511

EM AT04 support for B type thermocouples

The EM AT04 (6ES7 288-3AT04-0AA0) supports Type B thermocouples as well as the other types listed in the S7-200 SMART System Manual.

TD400C display problem

Due to an error, values greater than 32767 display as 32767 when the data format of the TD400C value display is unsigned. You also cannot enter a value greater than 32767 for a value with an unsigned display type.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG

Update to the S7-200 SMART System Manual, edition 10/2013
A5E32938409-AB, 01/2015