

ELIMAR SOFT (www.elimar.com)

Siemens Step 5 AS-511 (Programming Port) DDE server. SDDE 1.10

1. Introduction.

SDDE is a Windows NT/95/98 DDE server for the Siemens Step 5 range of PLCs. The actual version supports S5-100, S5-90/95U and S5-115U series of PLCs. This version does not support 135U & 155U series. The driver uses AS-511 protocol in the programming port of the PLCs.

Using SDDE is quite simple and does not require special knowledge of Scada programming or PLC configuration. Data can be accessed directly from an Spreadsheet any Scada package supporting DDE or any other program that supports the DDE protocol.

2. Hardware Installation.

To connect your PC serial port to the PLC you need a TTY interface converter. You can use the interface cable supplied by Siemens for PLC programming or use another PC board supporting the TTY (current loop interface). Please contact support@elimar.com or fax USA 810-279-1708 if you require additional help.

3. Software Installation.

SDDE comes as a single executable setup file . The code has been kept small to reduce memory requirements. An additional SDDE.INI file is placed in the Windows directory for port configuration. The format of the INI file is:

```
; INI File for SDDE Server
[SDDE00]
Port = "COM1"
[SDDE01]
Port = "COM2"
[SDDE02]
Port = "COM3"
...
[SDDEXX]
Port = "COMX"
```

SDDE can have several instances named SDDE00, SDDE01, SDDE02...SDDEXX each instance must have a configured port in the INI file. This is; if you execute SDDE 3 times for instance you can monitor the values of 3 PLCs connected at the same time to 3 COM ports in your PC. Virtually there is no limit in the number of PLCs that can be connected other than the power of your computer and the availability of serial ports. SDDE uses standard Windows serial port programming so multi-port boards with standard Windows drivers can be used for multiple PLC support.

4. PLC support.

SDDE V 1.02 supports S100, S90/95U and S5-115U series of PLCs the programs reads and writes DB variables directly from the PLC memory and does not require any modification in the PLC programming. Additional reading of PLC inputs, outputs and flags are also included.

DB Syntax		
Data Type	Format	Example
Bit Values	DBnDw.b	DB10D4.6
String Values	DBSw.c	DB8S8.10
Left Byte	DBnDLw	DB10DL5
Right Byte	DBnDRw	DB10DR5
Word Values	DBnDWw	DB10DW5
Double Word Values	DBnDDw	DB10DD5

Where:		
n	DB Number	Values 1..255
w	Word number in DB	Values 0..255
b	Bit 0..15	Values 0 ..15
c	Number of characters	Values 1..255

Input Syntax			
Data Type	English Format	German Format	Example
Bit Values	Iw.b	Ew.b	E5.6
Byte Values	IBw	EBw	EB10
Word Values	IWw	EWw	EW20
Double Word Values	IDw	EDw	ED56

Output Syntax			
Data Type	English Format	German Format	Example
Bit Values	Qw.b	Aw.b	E5.6
Byte Values	QBw	ABw	EB10
Word Values	QWw	AWw	EW20
Double Word Values	QDw	ADw	ED56

Flag/Mark Syntax			
Data Type	English Format	German Format	Example
Bit Values	Fw.b	Mw.b	E5.6
Byte Values	FBw	MBw	EB10
Word Values	FWw	MWw	EW20
Double Word Values	FDw	MDw	ED56

Status Item:

A Status variable is included to report the Status of the communication with the PLC.
This DDE Item return a value of 1 when the communication link is OK and 0 if not.

5. Data Types.

The actual version of SDDE supports the following formats for byte, word and double word requests:

KF: is the integer STEP5 Format and is the default format.

KH: is the hexadecimal STEP5 Format and can be configured in the SDDE.INI file.

KBCD: is the BCD STEP5 Format and can be configured in the SDDE.INI file.

To specify a format in the Item name append the Format indicator in the Item string.

Example:

If DB10DW0 contains 928 (3A0H)

DB10DW0KF will be the same as DB10DW0 and will return the string 928

DB10DW0KH will return the string 3A0

DB10DW0BCD will return the string 3A0

The format for Hexadecimal and BCD values can be configured in the SDDE.INI file, the configuration string uses "printf" C format.

For Byte or Word the defaults are:

HexForm = "0x%04X" in our example will be 0x03A0.

BCDForm = "%04X" in our example will be 03A0.

For Long values use:

LHexForm = "0x%08IX" in our example will be 0x000003A0.

LBCDForm = "%08IX" in our example will be 000003A0.

The Hex formats will use a "0x" prefix this can be changed to for instance "\$" with "\$%04X".

The 04 indicates 4 digits with filling with zeros to the left the complete width, "\$%4X" will print without leading zeroes in the left.

The BCD numbers are evaluated in the same manner as the Hexadecimal ones. There is no test of BCD value so a wrong BCD value in the PLC will show as a Hexadecimal value with letters.

5. DDE Support.

As a DDE server SDDE can serve many Windows applications with ease. The mechanism of DDE in automatic mode is very simple; the client application advises or pre-requests some items to the Server. The Server at a fixed interval sends updated information to the client of the requested Items. The Server can organize the data to be read in a way to group the different requests in Serial messages. The requests are controlled and are treated synchronously. Asynchronous requests are not accepted to avoid uncontrolled demand of the serial link. At the moment we introduce an Item in a Spreadsheet or a Scada it will request Asynchronously the Item value. The DDE server will respond with a Not Available Message to indicate that the value has not yet been evaluated. At the next refreshing period the value will be updated.

Sending of values to the PLC from the client is done by the use of the Poke command, each time the server gets up to refresh the values of the PLC tests before if there are any values pending to be modified in the PLC and proceeds to send them.

The Not Available Message can be configured in the SDDE.INI file with the following syntax.

NotAvailableMessage = "???" Means that not available requests will receive the "???" string.

6. Optimization.

The S5 programmer's port is slow with tremendous overhead, so communications must be optimized to get the highest speed possible. The SDDE starts a communication for each DB that has been demanded to read. If the words that are to be read are in consecutive positions in the DB the communication is shorter and therefore faster. Try to join as much as possible DB to be read. The writing is done byte by byte so the data separation has no effect.

7. Use with Microsoft Excel 97 & Excel 2000.

To read values in an Excel cell use the following format:

```
=SDDE|COMport!ItemName
```

For instance to access DB10 DW0 in a PLC connected to COM1 use the following syntax:

```
=SDDE|COM1!DB10DW0
```

To send a value we must use VBA (Macros in Visual Basic for Applications).

To start a conversation we use:

DDEInitiate to start the conversation.

DDEPoke to send the data.

DDETerminate to close the conversation.

In this example we Poke the Item in A7 with the value in A8.

```
Public Sub DDEStart()
```

```
    Dim Conv1 As Long, Item1 As String, Value1 As Variant
```

```
    Item1 = Sheets("Page1").Range("A7").Text
```

```
    Set Value1 = Sheets("Page1").Range("B7")
```

```
    Conv1 = DDEInitiate("SDDE", "COM1")
```

```
    DDEPoke Conv1, Item1, Value1
```

```
    Worksheets("Page1").Cells(1, 1).Value = Answer
```

```
    DDETerminate (Conv1)
```

```
End Sub
```

8. Use with InTouch Version 7.

The SCADA programs used for industrial control can use SDDE to communicate with STEP5 CPUs. Some old drivers that use still old DOS TSR routines do not work with Windows 95/98 in the fastest CPUs, old drivers don't support Windows NT. SDDE is designed with the standard Windows serial drivers so can use any serial port with a standard driver.

To scan a variable add a new Tag and set the Item name to the corresponding PLC variable i.e. DB10DW6. Add a new Access Name with Application Name = "SDDE" and Topic Name = "COM1" (or the serial port you are using). You are ready to use this Tag in the Window Maker.

9. Use with Lotus 123 from Lotus Version 9 (Smartsuite Milenium)

To get values from a Cell use the function @DATALINK("SDDE","COMx","Item")

Example:

To get DW5DB6 from comport 1 type @DATALINK("SDDE","COM1","DW5DB6")

Using scripting:

```
Dim A As Application
Dim C As Conversation
Dim V As Advise
```

Sub DDEStart

```
Dim ItemName As String

Set A = CurrentApplication

ItemName = [A:B5].CellValue 'A:B5 contains the item name.

Set C = A.NewConversation ("SDDE", "COM1")
Set V = C.NewAdvise (ItemName, $TextFormat, [A:C5], )
```

End Sub

Sub DDEEnd

```
V.Clear
C.Clear
```

End Sub

Sub Poke

```
C.Poke [A:B5].CellValue , $TextFormat , [A:D5]
```

End Sub

9. DISCLAIMER

This software is provided "as is" and is without warranty of any kind. The author of this software does not warrant, guarantee or make any representations regarding the use or results of use of

this software in terms of reliability, accuracy or fitness for purpose. You assume the entire risk of direct or indirect, consequential or inconsequential results from the correct or incorrect usage of this software even if the author has been informed of the possibilities of such damage. Neither the author nor anybody connected to this software in any way can assume any responsibility.