

# SP7 LabVIEW Toolkit

## Quick Reference Guide V 2019.6



DATA AHEAD **SP7** is a toolkit that allows communication with Siemens PLCs from LabVIEW. Use **SP7** to read and write different data blocks, memory blocks, digital input/output modules as well as analog input modules. **SP7** can also be used on NI LabVIEW Real-Time systems.

<b>SP7 LabVIEW Toolkit Quick Reference Guide V 2019.6</b>	<b>1</b>
What is SP7?	2
Supported functionality	2
Supported controllers	3
Usage with S7-1200/1500	3
Support Siemens LOGO! 8 and S7-200	4
Deployment	6
Requirements	6
Building an Executable Application	6
Warning	6
Installing SP7	7
License explanations	9
Developer license	9
Runtime license	9
Activating SP7	10
Deactivating SP7	14
<b>Getting Started</b>	<b>16</b>
Read digital inputs with an implemented example	16
<b>SP7 API</b>	<b>18</b>
Connection	18
Read operations	18
Write operations	19
Applications / Examples	19
<b>Error Codes</b>	<b>20</b>
<b>Support and Feedback</b>	<b>21</b>
<b>VI Overview (LabVIEW)</b>	<b>22</b>

## What is SP7?

**SP7** is an API (“Application Programming Interface”) for NI LabVIEW™ that offers read and write functionality for Siemens SIMATIC S7 PLCs („Programmable Logic Controllers”) over Ethernet. It can be used in typical test & measurement applications, to quickly transfer data between a computer and a PLC. **SP7** provides a selection of functions to transfer data in different data types supported by the PLC.



Although **SP7** uses a well-established protocol to transfer data, DATA AHEAD cannot guarantee complete compatibility with all PLCs. When developing programs, careful use of **SP7** and regular backing up of user data is advised, to avoid any possible loss of data. In general, DATA AHEAD cannot be held liable for any loss of data that occurred whilst using **SP7**.

## Supported functionality

- Open Connection
- Close Connection
- Read/Write DB
  - o (data type: Array of BOOL, BYTE, INT, DINT, REAL, LREAL, STRING, WORD, DWORD, DATE, TIME, TIME of DAY, S5TIME, single Bit)
- Read/Write Memory Storage
  - o (data type: Array of BIT, BYTE, WORD, INT, DINT and REAL, single Bit)
- Read Digital Input
  - o (data type: Array of BIT or single bit)
- Read/Write Digital Output
  - o (data type: Array of BIT, write single bit)
- Read/Write Analog Input/Output
  - o (data type: Array of WORD)

## Supported controllers

Siemens SIMATIC S7 controllers with an Ethernet interface, such as

- S7-200
- S7-300
- S7-400
- S7-1200
- S7-1500

and Siemens LOGO! 8 logic module.

## Usage with S7-1200/1500

- Permit PUT/GET communication in Siemens project settings

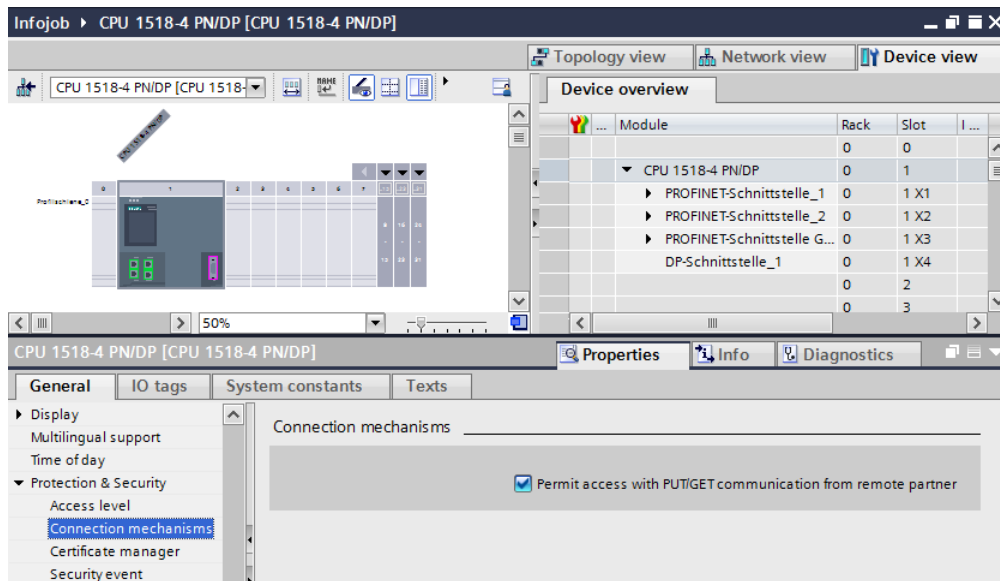


Image 1: Permit access with PUT/GET communication

- Accessibility of the PLC has to be “full access/no protection”

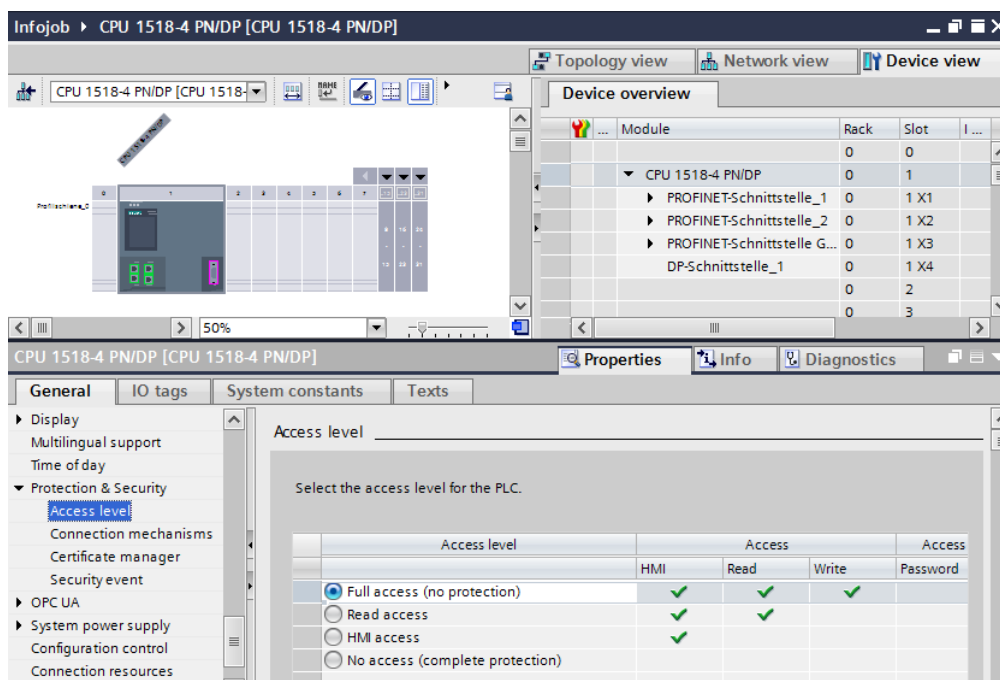


Image 2: Set full access

DB's have to be global accessible and the offset has to be static ("Optimize db lock access" unchecked)

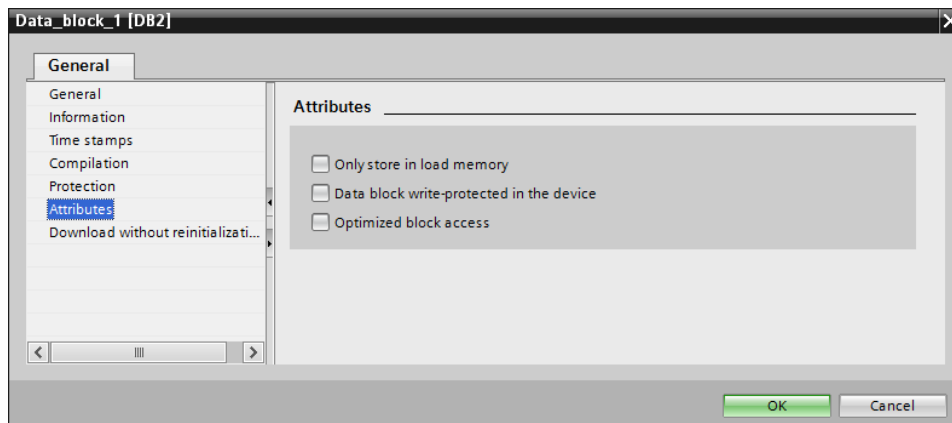


Image 3: Turn off optimized block access

## Support Siemens LOGO! 8 and S7-200

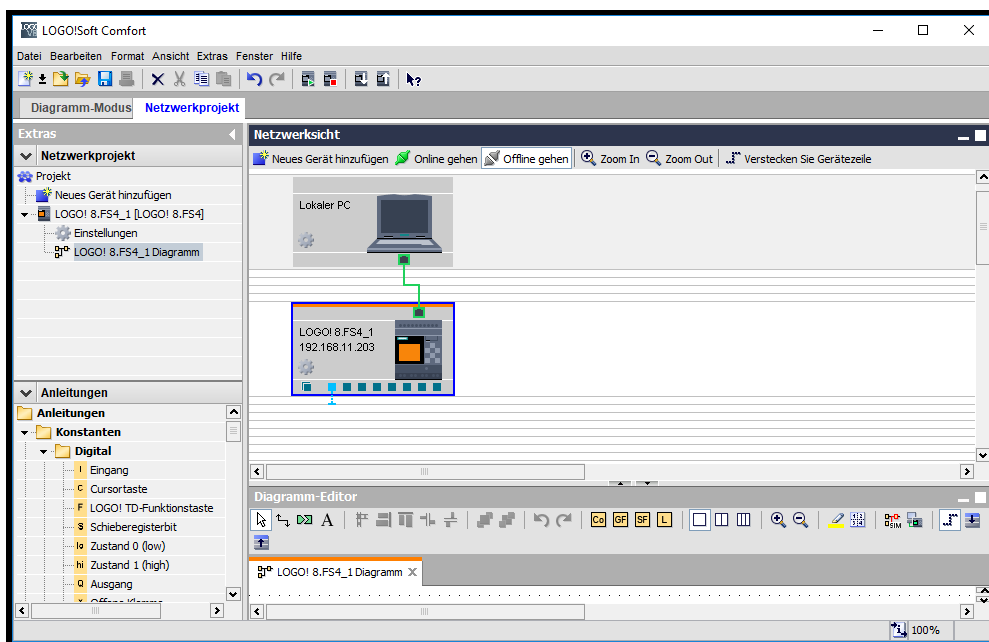


Image 4: Configuration LOGO!Soft Comfort

In order to connect to a LOGO! 8 logic module or a S7-200 you need to allow a **server connection** on your LOGO! 8 /S7-200. Configure this with the LOGO!Soft comfort software resp. Step7 for S7-200: from the **Tools** menu, choose **Ethernet Connections**. Right click on Ethernet Connections and select "Add connections" to add a new connection.



Image 5: Configuration of LOGO! connection settings. The same applies to a S7-200.

Now configure this by double-clicking the new connection (Image 5). Under **Remote Properties (Client)** set the TSAP to 03.00. Under **Local Properties (Server)** activate Connect with an Operator Panel (OP) to set the local TSAP to 02.00. It is also useful to activate “Accept all connection requests”, otherwise specify the IP-address from the client from which you wish to connect to your LOGO! 8.

Apply the same settings when using a S7-200. Choose a local TSAP of 02.00 and a Remote TSAP of 03.00.

SP7 can then access the **VM** memory area of the LOGO module via read/write **DB 1**. All LOGO resources are mapped into that memory area. Refer to your [LOGO documentation](#) for further details.

Block Type	VM Address (From)	VM Address (To)	Range
I	1024	1031	8 bytes
AI	1032	1063	32 bytes
Q	1064	1071	8 bytes
AQ	1072	1103	32 bytes
M	1104	1117	14 bytes
AM	1118	1245	128 bytes
NI	1246	1261	16 bytes
NAI	1262	1389	128 bytes
NQ	1390	1405	16 bytes
NAQ	1406	1469	64 bytes

Image 6: Mapping between LOGO! 8 I/O and VM addresses

**Remark:** Although not tested it should be able to connect to a LOGO! 7 too by applying the same network settings as mentioned above. However the mappings of the VM memory area differ from the ones mentioned in Image 6. See the documentation for you LOGO! Module for further details.

## Deployment

### Requirements

**SP7** does not require any additional software to be deployed.

### Building an Executable Application

**SP7** does not require any additional software to be used in an executable. It relies on LabVIEW functions only; therefore an executable will run on any target that LabVIEW support including all NI Real-Time target.

There are no specific OS requirements for an executable that makes use of the **SP7** API other than the requirements that come with the specific LabVIEW version you are using.

### Warning



While the functionality of **SP7** has been tested extensively, DATA AHEAD cannot guarantee flawless communication with specific PLCs. Taking precautions, such as implementing buffering in your application, is highly recommended. Incorrect usage of the **SP7** API can potentially lead to a loss of data.



PLCs often control industrial automation equipment. Manipulation and incorrect usage of PLCs can lead to serious hazards and possibly have harmful consequences. Only authorized and qualified personnel may implement interfaces to a PLC that controls and automates machines and processes.

DATA AHEAD GmbH cannot be held liable for errors, for damage to property or pecuniary damages caused during the usage of the **SP7** API.

## Installing SP7

LabVIEW 2011 or newer is required to install **SP7**.

The easiest way to install **SP7** is to use the National Instruments Tools Network or the JKI VI Package Manager (Image 1). Search for “SP7” and select „Install & Upgrade Packages” to install SP7 to your LabVIEW installation.

**SP7** requires the JKI VI Package Manager 2014.0.0 or newer.

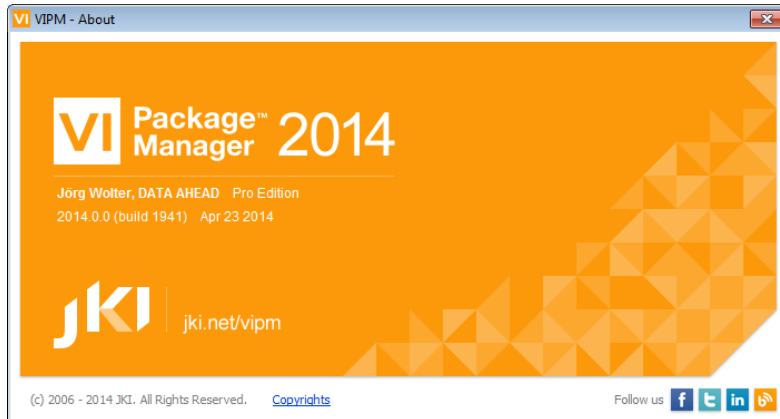


Image 7: JKI Package Manager to find and install SP7

Upon successful installation, a new palette will be installed in LabVIEW that can be found in the “data communication” function palette of the block diagram (Image 8).

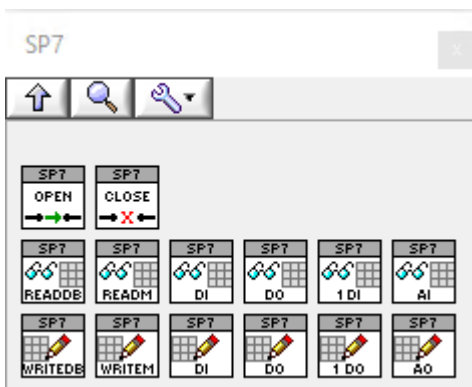


Image 8: SP7 palette in "data communication"

Furthermore, examples will be added to the LabVIEW Example Finder. To search for examples open the help tab and choose „Example Finder“ (Image 9). In the Example Finder, search for „**SP7**“

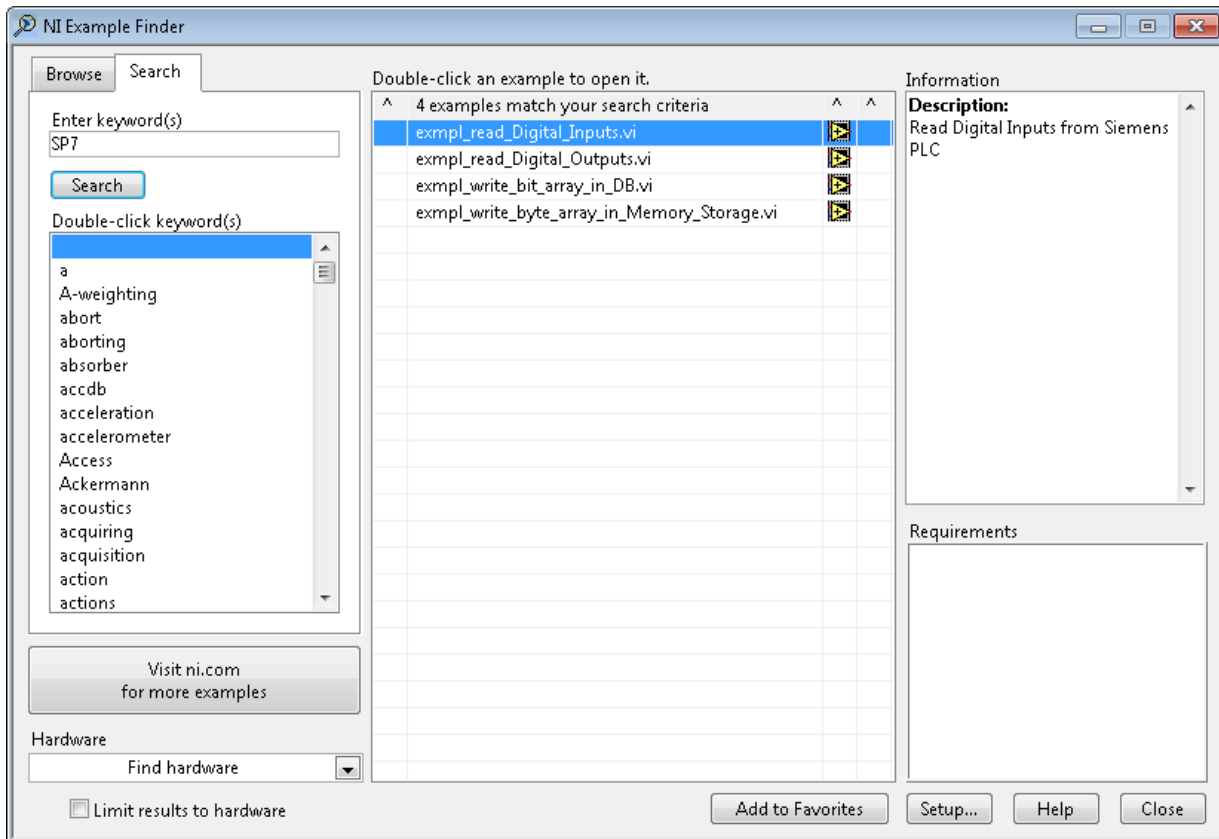


Image 9: SP7 examples in the NI Example Finder



## License explanations

### Developer license

The developer license is needed in order to use the SP7 API. Upon purchasing a license from NI, you will receive an email titled “*Activation Information for your ni.com purchase: SP7 - DATA AHEAD AG*” that contains your **License ID** and **Password** for the developer suite. With this information you can activate SP7 which is described in “Activating SP7”.

Extract from the Eula:

The license is a single End-User PC License. You may install and use one copy of the software on a single computer. The SOFTWARE PRODUCT is used on only one computer at any one time. The provided SOFTWARE PRODUCT is not used on more computers than the number of licensed users at any one time.

### Runtime license

The runtime license is needed when your program or executable makes use of the SP7 API. It is only needed once in a life time of the program and it doesn't need to be activated in any way. The runtime licensing system is based on honesty between customer and developer.

## Activating SP7

You can evaluate **SP7** for 30 days. After this period you have to activate the **SP7** add-on in LabVIEW. Upon purchasing a license from NI, you will receive an email titled “Activation Information for your ni.com purchase: **SP7** - DATA AHEAD AG” that contains your License ID and Password for the developer suite. In order to activate **SP7**, your target system needs to be connected to the Internet. If there is no Internet connectivity available on the target computer, you can also activate **SP7** through a web browser on a different computer (see step 3 – alternative option).

### Attention:

If you want to deactivate your license (e.g. since you want to transfer your single seat license to another system), you have to activate **SP7** in LabVIEW 2014 or newer AND automatically through an Internet connection, not through a web browser on another system (step 3 – alternative option). You also have to install v2018.4 or higher of **SP7**.

**Step 1:** Open LabVIEW. Select “Help” in the menu bar and then on the menu point “Activate Add-ons” to carry out the function.

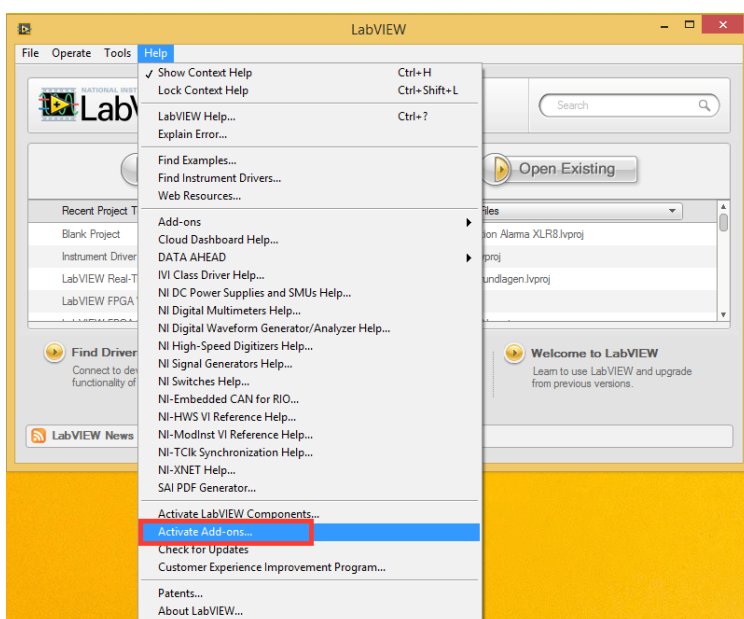
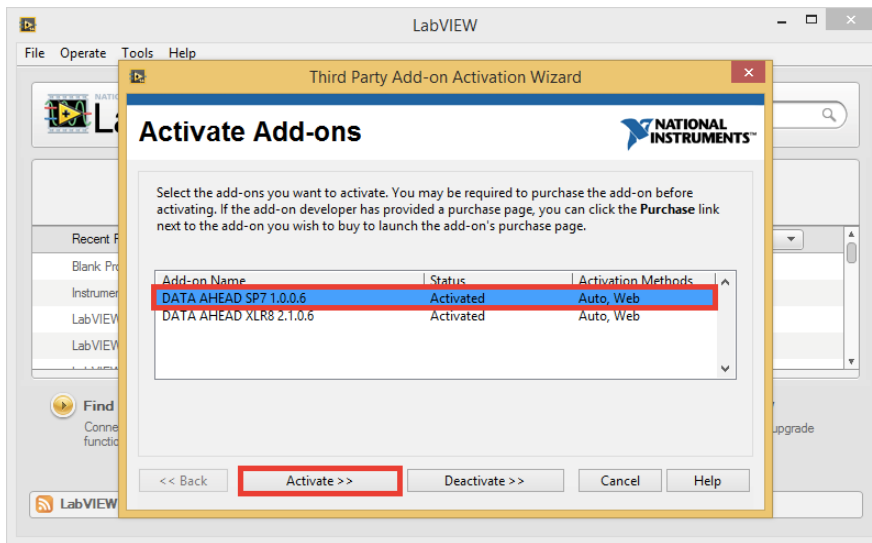


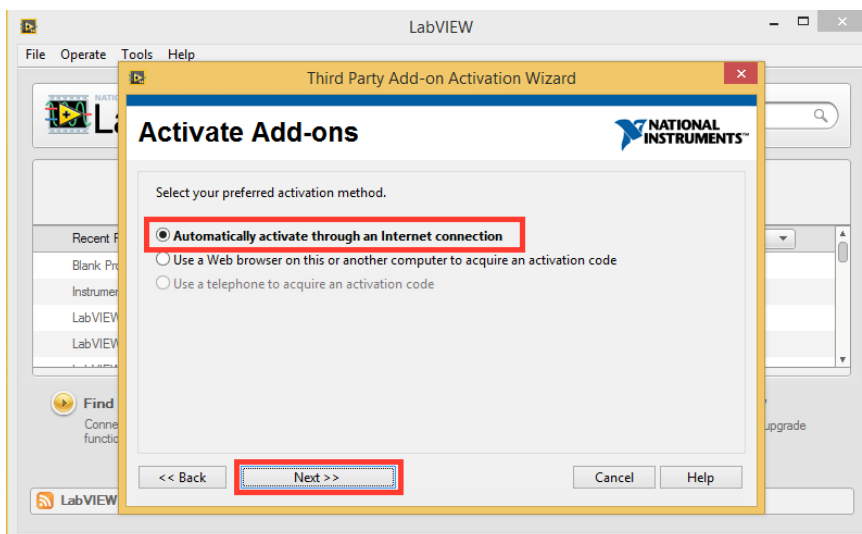
Image 10: Activation Step 1

**Step 2:** The following pop-up window will appear and display your **SP7** installation and activation status. Select **SP7** “DATA AHEAD **SP7** x.x.x.xx” and click “next” or “Activate”.



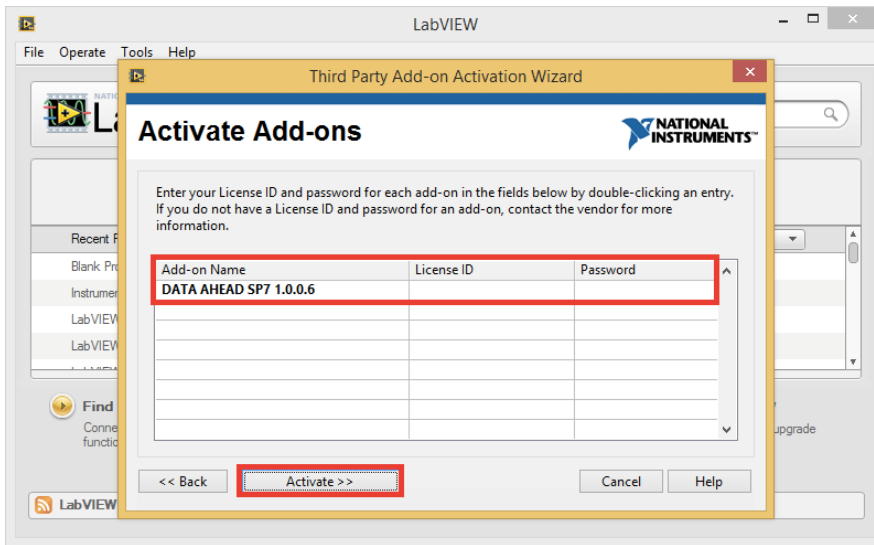
**Image 11: Activation Step 2**

**Step 3:** Now select the method of activation. If the computer is connected to the Internet select the first option. If you do not have Internet access, see the alternative option below. To proceed, click “Next”.



**Image 12: Activation Step 3**

**Step 4:** To complete the activation, enter the provided License ID and Password that you received in the email from NI. Please mind capital letters. After entering the License ID and Password click “Activate”.



**Image 13: Activation Step 4**

**Step 3 (alternative):** If no Internet connection is available, selecting the second option will direct you to National Instruments activation website where you will have to enter the user codes displayed in the dialog as well as your License ID and Password. This option allows you to activate the toolkit from another computer that is connected to the Internet. But be aware: you will not be able to properly deactivate the license.

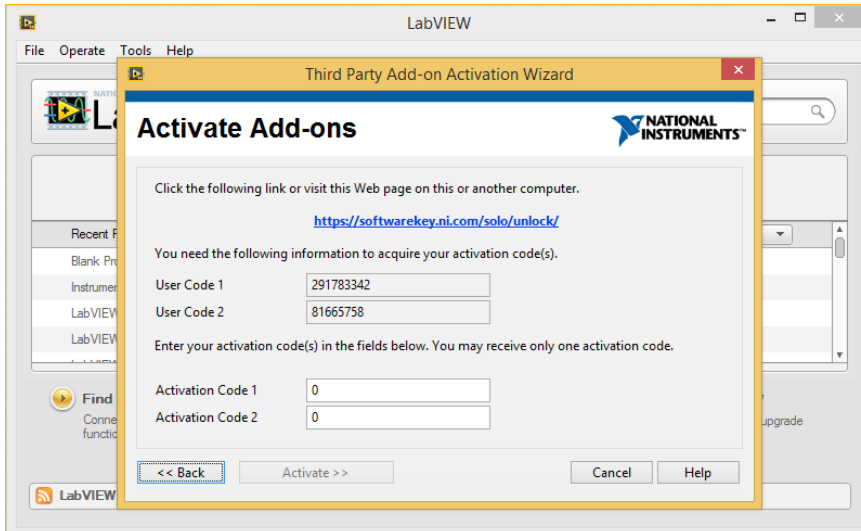


Image 14: Activation Step 3, option 2: activate through a web browser – LabVIEW dialog

**Step 4 (alternative):** Open the website [https://softwarekey.ni.com/solo/unlock/](\"https://softwarekey.ni.com/solo/unlock/\") on a computer with Internet connection. Enter the User Code 1 and User Code 2 on the website, as displayed in the dialog on the target computer. Enter License ID and Password from the activation e-Mail. Click "Continue". Activation Code 1 and 2 will now be generated that you can enter in the dialog on the target computer. Finish the dialog by clicking "Activate".

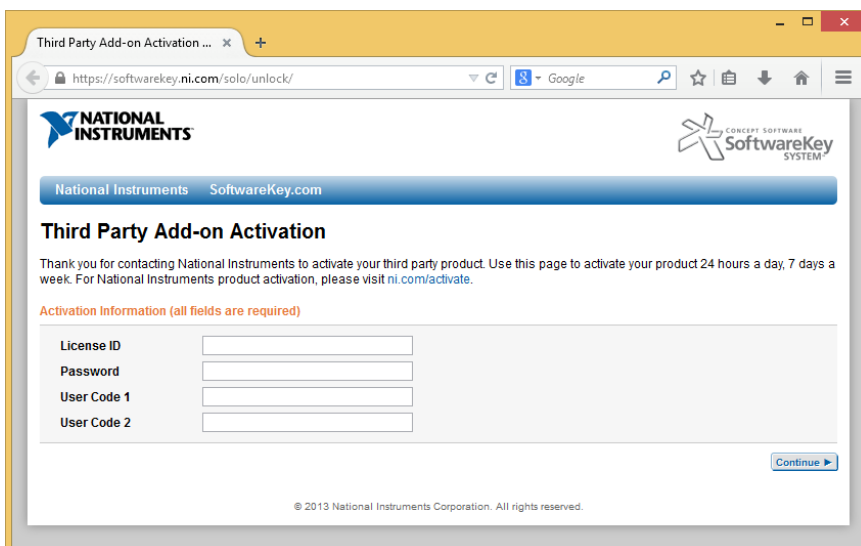


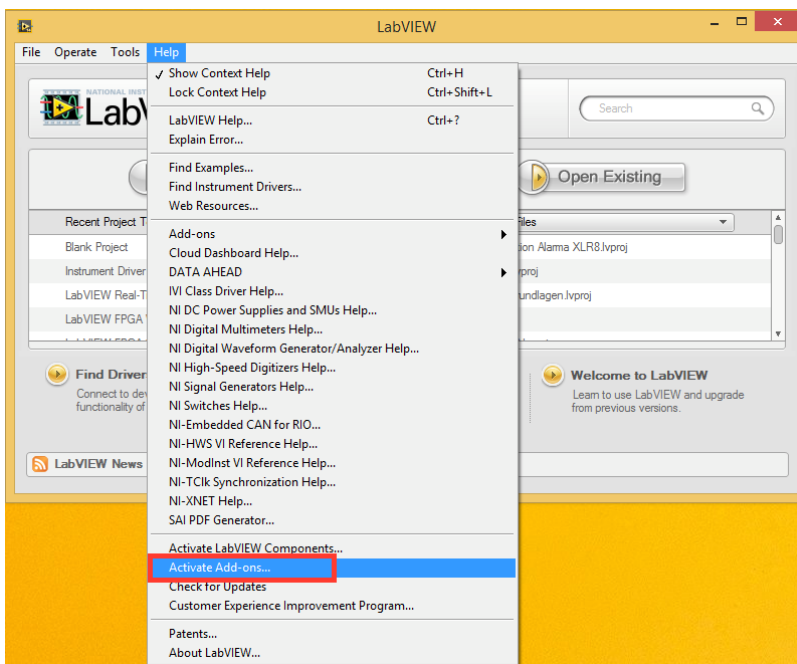
Image 15: Activation Step 3, option 2: activate through a web browser – activation website

## Deactivating SP7

As already stated on page 10 there are some important requirements for the deactivation of a SP7 license. You need to:

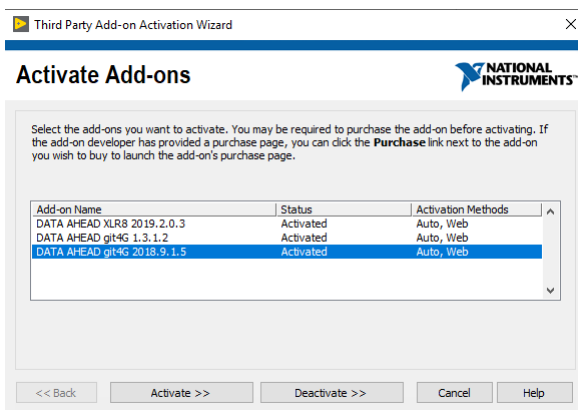
- install and activate v2018.4 or higher of SP7
- automatically activate through an internet connection (Step 3 of activation procedure)
- install and activate using LabVIEW 2014 or higher

**Step 1:** Start LabVIEW 2014 or higher and select Help -> Activate Add-ons...



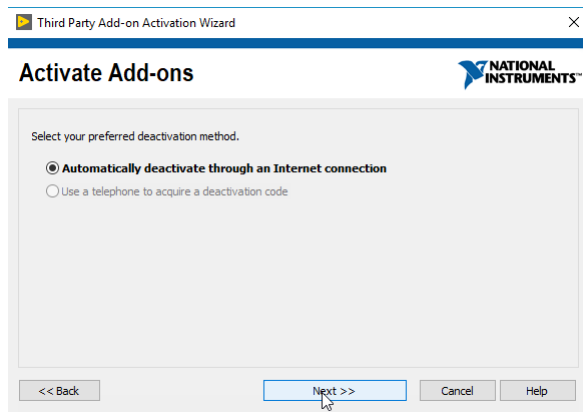
**Image 16: Step 1 deactivation**

**Step 2:** Select your SP7 license and press the button “Deactivate >>”. Make sure that you have an internet connection for this to work.



**Image 17: Step 2 deactivation**

**Step 3:** Your only option is to “Automatically deactivate through an Internet connection”  
Confirm with „Next >>“



**Image 18: Step 3 deactivation**

**Step 4:** On success confirm the final dialog with “Finish >>” or  
“Restart LabVIEW >>”.

Feel free to contact us by e-mail if you experience any problems with this procedure.

## Getting Started

To get started, please locate the examples that are shipped with the **SP7** package. All of the examples include a short description in their respective VI documentation and explain the appropriate usage of the API VIs. For instance, open the example “exmpl\_read\_Digital\_Inputs.vi”.

To accomplish a successful connection to the plc, ensure that there are enough connection resources set in the plc. The number of allowed connections to the plc can be set in the plc develop environment under cpu settings.

### Read digital inputs with an implemented example

To read digital inputs please open example „exmpl\_read\_Digital\_Inputs.vi” from the LabVIEW example finder.

LabVIEW: Help-> Find Examples... – search for „**SP7**”.

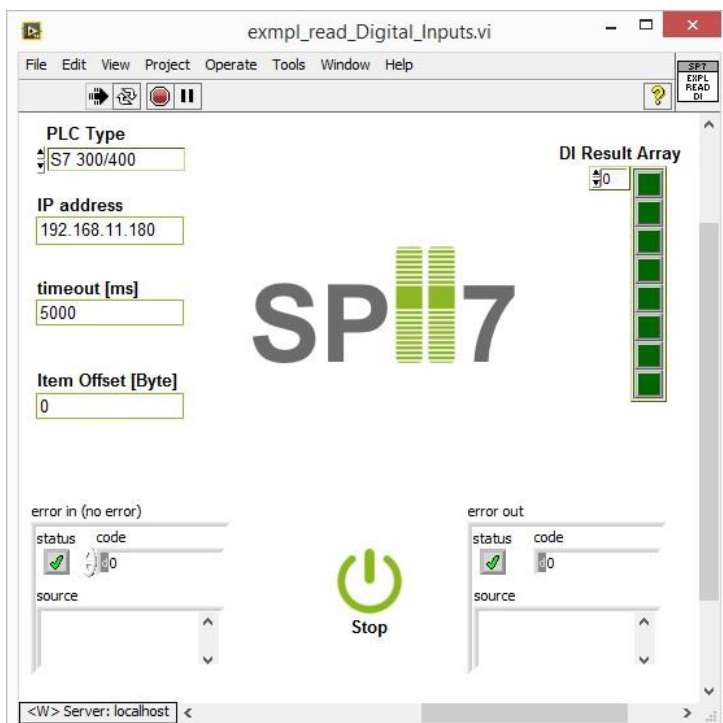


Image 19: front panel of the example „exmpl\_read\_Digital\_Inputs.vi”

You will see three input parameters (Image 19) and an output of a Boolean array.

Please enter the IP address of the PLC in the “IP address” control.

A physical Ethernet connection to the PLC and its IP address are required for **SP7**.

Note: You can use network or IP scanning programs to find an unknown IP address.

The second parameter specifies the timeout in milliseconds.

Setting a higher value here can be useful in unstable or slow networks. At the same time, a high timeout value might cause your programs to respond more slowly.



Geräteübersicht						
	Baugruppe	Baugr...	Steck...	E-Adresse	A-Adres...	Typ
	PLC_1	0	2			IM 151-8 PN
	PME DC24V_1	0	4	2043*		PME DC24V
	4DI x DC24V ST_1	0	5	10.0...1...		4DI x DC24V
	4DI x DC24V ST_2	0	6	2.0...2.3		4DI x DC24V
	4DO x DC24V / 0,5A ST_1	0	7		3.0...3.3	4DO x DC24
	2AI x U ST_1	0	8	320...323		2AI x U ST

Image 20: device overview in Siemens' TIA Portal software

The third parameter is the item offset in byte. These offsets are identical to the addresses specified in the PLC device overview of your PLC configuration software.

Image 20 highlights addresses "2.0...2.3" of a "4DI x DC24V ST\_2" digital input module. The address "2.x" here corresponds to an **SP7** item offset of 2.

Depending on your PLC configuration and the modules installed, the addresses available to you may differ.

When the VI is running, a query of the digital input module takes place every 50 ms. The values are visualized in the "DI Result Array".

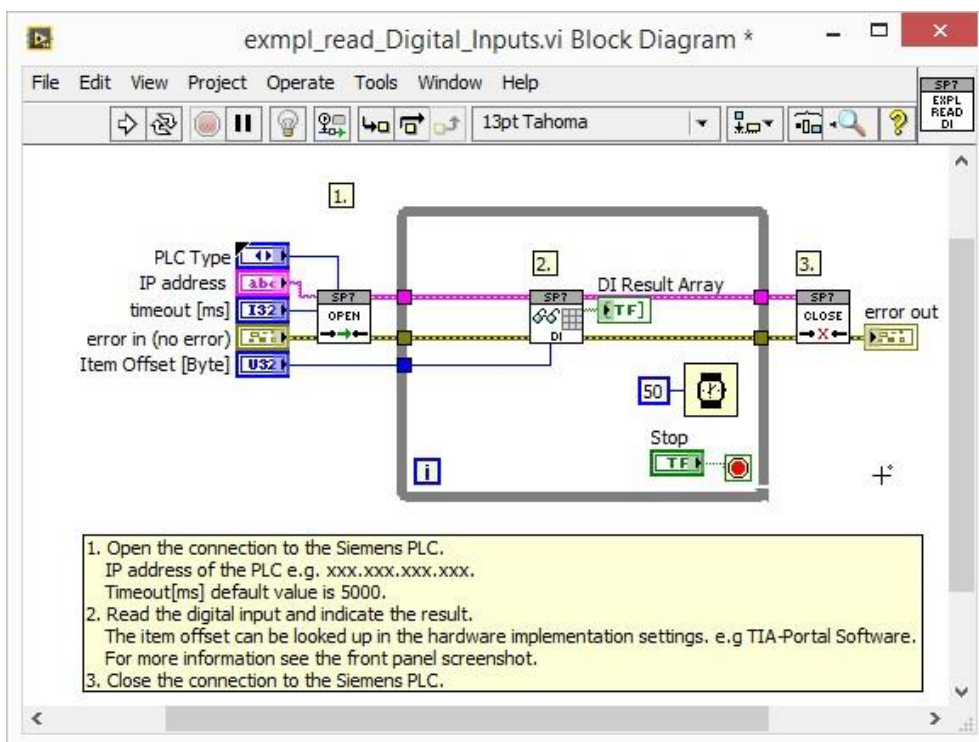


Image 21: example of block diagram „exmpl\_read\_Digital\_Inputs.vi“

The corresponding block diagram is shown in Image 21. This image depicts the three **SP7** functions used here: opening a connection, then continuously reading the digital input module until the "Stop" button is pressed. Eventually, the connection is closed.

## SP7 API

All VIs can be found in the **SP7** sub-palette of the „Data Communication“ palette. A detailed help is included in the description of each VI (-> Ctrl+H).

### Connection

These VIs establish and close a connection to a PLC.



VI	Name and description (more information in the context help)
	<b>connect.vi</b> Opens an Ethernet connection to a given PLC
	<b>disconnect.vi</b> Closes the connection.

Chart 1: functions to establish a connection

### Read operations

These VIs can be used to read data from a PLC. The item offset value must be entered in bytes. (For example: M10.0 -> enter "10").







	<b>Read DB.vi</b> (Polymorph: BOOL, BYTE, INT, DINT, REAL, LREAL, STRING, WORD, DWORD, DATE, TIME, TIME of DAY, S5TIME, single BOOL) Reads data from a specified data block.
	<b>Read M.vi</b> (Polymorphic: BIT, BYTE, INT16, INT32, REAL, WORD, single BOOL) Reads data from a specified memory block
	<b>Read DI.vi</b> Reads the specified digital input as a BIT array
	<b>Read DO.vi</b> Reads the specified digital output as a BIT array
	<b>Read DI single.vi</b> Reads a single specified digital input
	<b>Read AI.vi</b> Reads the specified analog input value as WORD.

Chart 2: read operations

## Write operations

Use the following VIs to write data to a PLC. Depending on the required data type, polymorphic VIs are available.







	Write DB.vi (Polymorphic: BOOL, BYTE, INT, DINT, REAL, LREAL, STRING, WORD, DWORD, DATE, TIME, TIME of DAY, S5TIME, single BOOL) Writes data to a specified data block.
	Write M.vi (Polymorphic: BIT, BYTE, INT16, INT32, REAL, WORD, single BOOL) Writes data to a specified memory block.
	Write DI.vi Writes a BIT array to a digital input. (simulated DI only)
	Write DO.vi Writes a BIT array to a digital output.
	Write DO single.vi Writes a single BIT to one digital output.
	Write AO.vi Writes a Word-Array to an analogue Output (not scaled). The scale values are different to any output module therefore you have to get the factors from the analogue module description.

Chart 3: write operations

## Applications / Examples

SP7 ships with a number of example programs that can be found using the LabVIEW example finder: Help -> Find Examples – Search for „SP7“





	Exmpl_read_Digital_Inputs.vi This VI demonstrates how to access the digital inputs.
	Exmpl_read_Digital_Outpus.vi This VI demonstrates how to read digital outputs' values.
	Exmpl_write_bit_array.vi This VI demonstrates how to write a BIT array in a data block
	Exmpl_write_byte_array_in_Memory_Storage.vi This VI demonstrates how to write a BYTE array in the memory storage

Chart 4: applications and examples

## Error Codes

SP7 will create only a few specific error codes. Besides these error codes typical TCP/IP error codes like 56 or 66 can occur (cf. LabVIEW help for further explanation).

Error Code	Description	Possible Cause(s)
6002	Could not verify the connection. Please check the ip address and the timeout.	Error thrown during "connect.vi"
6010	Data not valid! Value out of range	<ul style="list-style-type: none"> <li>- Put/Get protocol not active</li> <li>- DB still optimized</li> <li>- write/read on address range that does not exist</li> <li>- other reasons</li> </ul>
6011	Data not valid! there is no peripheral at given address	
6012	No answer from PLC to request	Connection has been lost or closed by the peer
6013	Data not valid! A piece of data is not available, e.g. when trying to read a bit block (with a length other than 1) or a non existing DB (200 family)	
6015	Data not valid! The data address exceeds the address range	
6016	Data not valid! Cannot read a bit block with a length other than 1	
6017	Data not valid! Write data size does not fit item size	
6020	Data not valid! A piece of data is not available, e.g. when trying to read a non existing DB	
6042	S5TimeValue is out of range (above 9999999) and will be set to zero	Wrong value for "write DB S5 Time array"
6043	DateAndTime year is out of range	Wrong value for "write DB Date Time Array", year below 1990 or above 2168
6044	Date Time Stamp year is out of range	Wrong value for "write DB Date Array", year below 1990 or above 2168
6045	You entered an item number below 1.	Try to read/write 0 elements of hardware (i.e. DI, DO, AI, AO)

## Support and Feedback

Please contact us at [produkte@dataahead.de](mailto:produkte@dataahead.de)

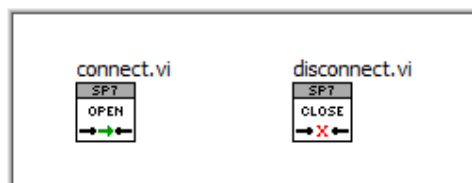
For latest news and support on our toolkits, go to  
<https://decibel.ni.com/content/groups/data-ahead-toolkit-support>

DATA AHEAD AG  
Prinzregentenufer 3  
90489 Nürnberg  
Germany

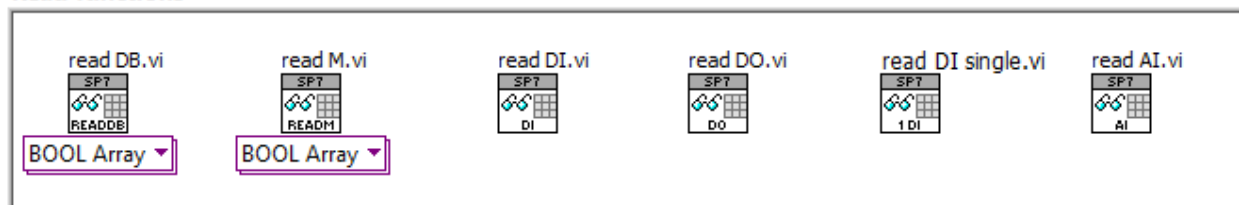
© Copyright 2019, DATA AHEAD

## VI Overview (LabVIEW)

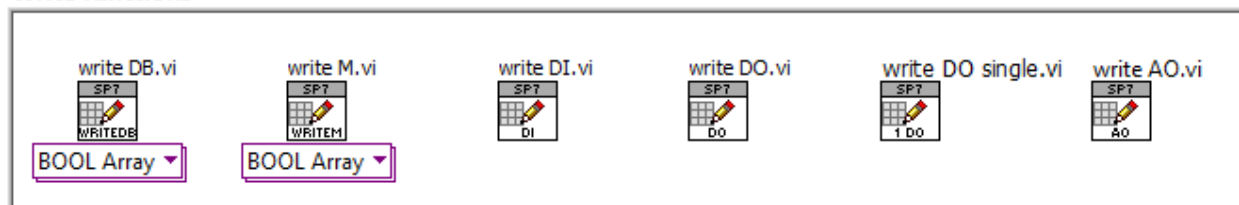
### Connection



### Read functions



### Write functions



### Examples

