Home

Library description

This library allows the programmer to create an array while the PLC is running (also known as *runtime*), without knowing beforehand its size or datatype.
An analogy can be made with the C default library functions "malloc" and "free".

Make sure to check the License agreement.

A list of current features and the library changelog can also be found in Library features and changelog.

> ⚠ **Warning**
>
> This library is still a work in progress. Therefore, expect many bugs, as well as interface and functionality changes.
> It is not yet meant to be used in a production environment, but to act as a proof-of-concept.

> ⚡ **Danger**
>
> This library will run for 2 hours for testing purposes. After the time is elapsed, the PLC will go to **STOP**.

Guides and examples

- Example project.

Function blocks overview

- LRA_CreateRuntimeArray.
- LRA_ReadRuntimeArray.
- LRA_WriteRuntimeArray.
- LRA_DeleteRuntimeArray.

License agreement

Attribution-NonCommercial-ShareAlike 4.0 International

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

1. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
2. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
3. BY-NC-SA Compatible License means a license listed at creativecommons.org/compatiblelicenses , approved by Creative Commons as essentially the equivalent of this Public License.
4. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
5. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
6. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

7. License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution, NonCommercial, and ShareAlike.
8. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
9. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
10. Licensor means the individual(s) or entity(ies) granting rights under this Public License.
11. NonCommercial means not primarily intended for or directed towards commercial advantage or monetary compensation. For purposes of this Public License, the exchange of the Licensed Material for other material subject to Copyright and Similar Rights by digital file-sharing or similar means is NonCommercial provided there is no payment of monetary compensation in connection with the exchange.
12. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
13. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
14. You means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

1. **License grant** .
    1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
        1. reproduce and Share the Licensed Material, in whole or in part, for NonCommercial purposes only; and
        2. produce, reproduce, and Share Adapted Material for NonCommercial purposes only.
    2. **Exceptions and Limitations** . For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
    3. **Term** . The term of this Public License is specified in Section 6(a) .
    4. **Media and formats; technical modifications allowed** . The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or

hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients .
    1. Offer from the Licensor – Licensed Material . Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
    2. Additional offer from the Licensor – Adapted Material . Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.
    3. No downstream restrictions . You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
6. No endorsement . Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i) .

2. **Other rights** .
    1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
    2. Patent and trademark rights are not licensed under this Public License.
    3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties, including when the Licensed Material is used other than for NonCommercial purposes.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

1. **Attribution** .
    1. If You Share the Licensed Material (including in modified form), You must:

1. retain the following if it is supplied by the Licensor with the Licensed Material:
    1. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
    2. a copyright notice;
    3. a notice that refers to this Public License;
    4. a notice that refers to the disclaimer of warranties;
    5. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
2. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
3. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

2. **ShareAlike** .

In addition to the conditions in Section 3(a) , if You Share Adapted Material You produce, the following conditions also apply.

1. The Adapter’s License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-NC-SA Compatible License.
2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.
3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

1. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database for NonCommercial purposes only;
2. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui

Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b) ; and

3. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

1. **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**
2. **To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.**
3. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

1. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
2. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

    1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
    2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

3. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
4. Sections 1 , 5 , 6 , 7 , and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

1. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
2. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

1. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
2. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.
3. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
4. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Library features and changelog

Current implemented features:

- Create, read, write and delete arrays with size supplied during runtime.
- Define array type during runtime.
- Type checking during runtime.
- Types supported: Byte, Word, DWord and LWord.
- Error codes for functions.

Changelog:

- 0.1.0 - Initial Release.

Example project

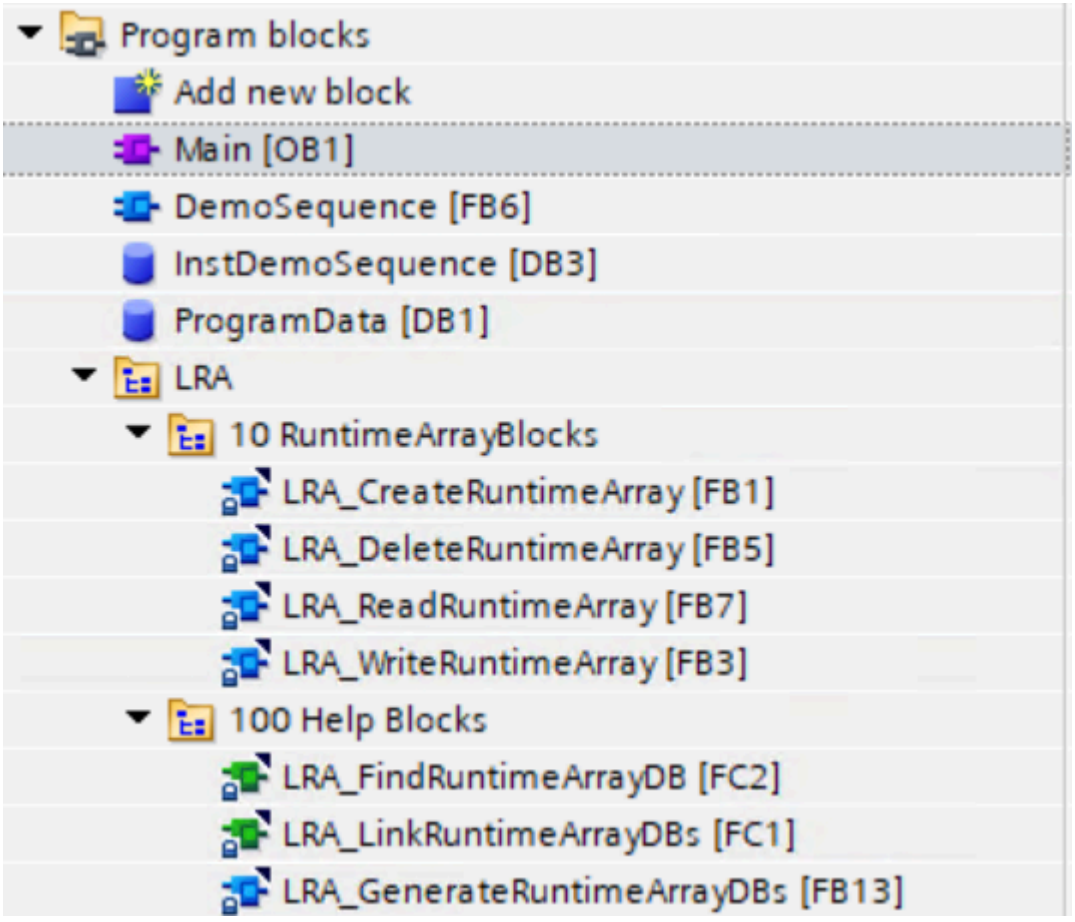
Current documentation version: 0.0.1

0.0.1 - Initial Release.

Introduction

A basic project showcasing the blocks of this library, as well as a possible workflow, where arrays can be created, read, written and deleted during runtime by the user.
The parameters present at the "ProgramData" DB allows the user to modify the demo sequence, defining the number of arrays to be created, as well as their data types and size.
Although the example is written using a state machine to call the library blocks and variants/references to transfer data around, this is not strictly necessary and it can be modified to better suit other scenarios.
All variables used to control the example can also be accessed using OPC UA, an HMI project or the Web API, none present in the current example project version.

Project organization



The "Main" OB is responsible for calling the "DemoSequence" FB, where the example itself resides.
Everything under the LRA folder belongs to the library itself, but the help blocks are not supposed to be called directly by the user. They are, instead, called by library functions themselves.
The "DemoSequence" FB contains the state machine where the arrays are created, written, read and then (optionally) deleted.
All example parameters can be modified offline or online in the "ProgramData" DB.

Demo sequence parameters: inputs and outputs

The following inputs and outputs for the example project are present:

ProgramData

	Name	Data type	Start value
1	Static		
2	demoSequenceInterface	"typeDemoSequenceInterface"	
3	inputs	"typeDemoSequenceInputs"	
4	startDemo	Bool	false
5	arraysUpperBound	UDInt	10
6	numberOfArrays	DInt	3
7	typeArray	String	'Byte'
8	deleteAtEnd	Bool	FALSE
9	stepMode	Bool	FALSE
10	nextStep	Bool	TRUE
11	outputs	"typeDemoSequenceOutputs"	
12	valueRead	LWord	16#0
13	currentStep	Int	0
14	currentError	Word	16#0
15	functionRuntime	Time	T#0ms
16	numberOfCycles	UInt	0

- Inputs (can be modified by user):
  - startDemo: Set to TRUE to start the demo sequence. Will be set again to FALSE when the sequence is over.
  - arraysUpperBound: define the upper bound of the arrays that will be created during the demo sequence[1]. The arrays always start at zero.
  - numberOfArrays: number of arrays of size "arraysUpperBound" to be created by the sequence.
  - typeArray: data type of created array. The following data types are currently supported: "Byte", "Word", "DWord", "LWord".
  - deleteAtEnd: if TRUE, the created arrays will be deleted before the sequence is over.

    - stepMode: if TRUE, it will be necessary to set the "nextStep" parameters to TRUE to let the sequence run for one more cycle. Note that it might be necessary multiple cycles to finish the sequence. This is used mostly to understand better how the example works.
    - nextStep: if TRUE, and stepMode is also TRUE, will let the sequence run for one more cycle.
- Outputs (modified by the software):
    - valueRead: last value read from the created array by the "LRA_ReadRuntimeArray" function.
    - currentStep: current step of the state machine responsible for the demo sequence.
    - currentError: last error output by any of the library functions. If more than one error is present, only the last one is shown.
    - functionRuntime: time spent by the PLC to complete the sequence. Can be used to benchmark different scenarios.
    - numberOfCycles: cycles run by the PLC to complete the sequence. Can be used to benchmark different scenarios.

Demo sequence steps overview

1. SM_IDLE: values used by the function are reset. Wait for "startDemo" to be TRUE to advance to next step.
2. SM_CREATE_ARRAYS: create all the runtime arrays. Number of arrays, upper bound and data type are taken from "ProgramData" DB input parameters.
3. SM_WRITE_ARRAYS: write to all positions of all created arrays. Value written is based on PLC time tick and is used simply as an example.
4. SM_READ_ARRAYS: read all positions of all created arrays.
5. SM_DELETE_ARRAYS: if "deleteAtEnd" parameter is TRUE, delete all created arrays.

Step-by-step instruction

1. Download the software either to PLCSIM, PLCSIM Advanced or to a real PLC from the S7-1500 family, modifying the hardware in the project if necessary.
2. Open the "ProgramData" DB, modify the "inputs" parameters.
3. Set "startDemo" to TRUE, so that the sequence can run.
4. Observe the effects through the "outputs" fields and the created DBs in the online PLC software.
5. Repeat from step 2, with new parameters if desired.

---

1. The created arrays can have different sizes, but the sequence would have to be modified for that or called multiple times with the desired size. ↩

LRA_CreateRuntimeArray

Current documentation version: 0.0.1

0.0.1 - Initial Release.

Description

The "LDA_CreateRuntimeArray" is used to create an array with size and datatype chosen during runtime in the work memory of the PLC. When created, a number will be output by the function and used for further operations with the *runtime array*.

Values can be read and written using the "LRA_ReadRuntimeArray" and "LRA_WriteRuntimeArray".

The created *runtime array* can be deleted with the "LRA_DeleteRuntimeArray" so that the work memory is made available again.

> ⚠ **Warning**
>
> The following limitations apply:
>
> - Max *runtime array* size is limited by available PLC work memory during its creation.
> - The created *runtime arrays* and the user created DBs share the same resource pool. Therefore, the max combined number of these blocks that might exist is 1000[1].
> - Array speed creation depends on the PLC and program load. Bigger *runtime arrays* take more calls to be created.
> - Created arrays are maintained when the PLC goes to STOP, but **deleted** in case of a POWER OFF. Retentive arrays are not yet supported.
> - All other specificities and limitations of the default function "CREATE_DB" also apply.

> ⚠ **Warning**
>
> In case there is an error during the array creation, the partially created array is not automatically deleted. The programmer is responsible for the detection and deletion trigger.

Functional description

This function works asynchronously. Therefore, depending on the size and array datatype, it might require multiple calls to complete its task.
If the done bit is set, the creation was successful and the *runtime array* number is available. A new trigger on "start" is needed if a new array creation is desired.
If the error bit is set, the creation failed and a new trigger on "start" is needed to try to create the *runtime array* again. It is recommended to delete the array if this happens.

> **Note**
>
> Always check the done and error bits to correctly evaluate if the job has been completed or failed.

Parameters

Parameter	*Declaration*	*Data type*	*Description*
start	Input	Bool	Positive edge triggers start of array creation.
arrayType	Input	String	Type of array to be created. Only the following datatypes are supported: "Byte", "Word", "DWord", "LWord".
arrayUpperBound[2]	Input	UDInt	Size of array defined by its upper bound. The array always start at 0.
runtimeArrayNumber	InOut	DB_DYN	Variable to store array identification number.
busy	Output	Bool	TRUE while array creation is ongoing.
error	Output	Bool	TRUE for one cycle when there was an error in the array creation.
errorId	Output	Word	Stores error code.
done	Output	Bool	TRUE for one cycle when the array creation was successful.

Parameter errorId

Error code (W#16#...)	**Description**
0000	No error.

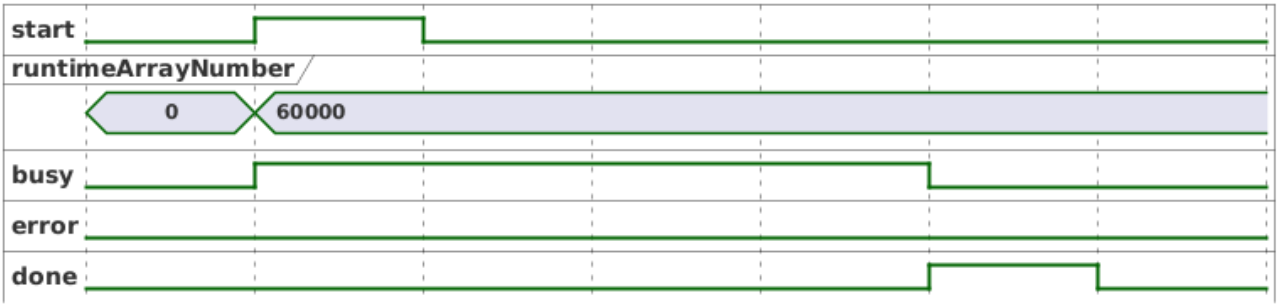
Error code (W#16#...)	Description
8001	User requested an array type that is not supported.
8002	Internal error.
8A92	Max number of DB blocks for CPU has been reached.
8AB1	No free DB number available.
8AB2	Insufficient free memory.
8ABB	Insufficient load memory.
8AC3	Max number of "CREATE_DB" instances was reached. In this case, the error bit is **not** set and a retry is done in the next block call.

Troubleshooting

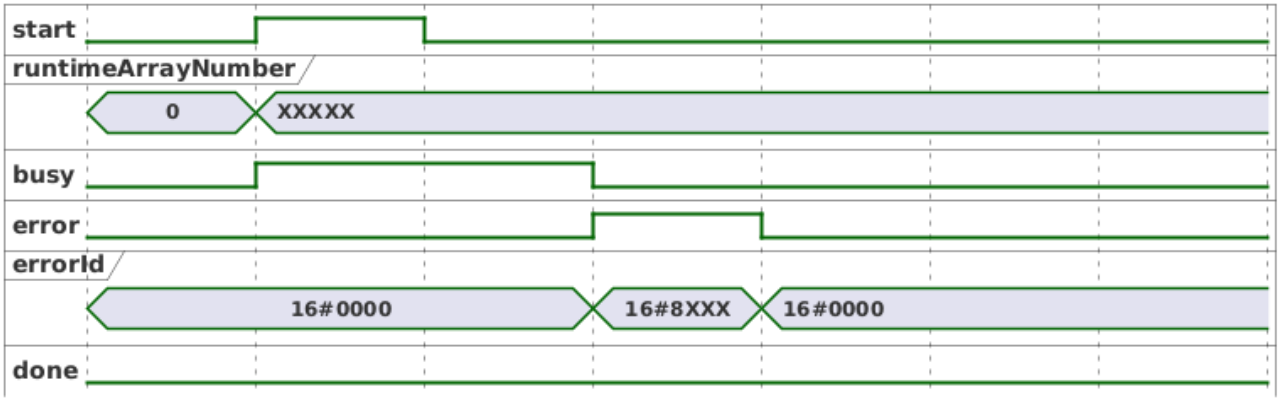
Error code (W#16#...)	Description
0000	No troubleshooting necessary.
8001	Check supported datatypes list and monitor function input during runtime.
8002	Inform library developer.
8A92	Reduce number of created blocks.
8AB1	Delete an user created DB or *runtime array*.
8AB2	Reduce work memory use or try again with a smaller *runtime array* size.
8ABB	Reduce load memory use.
8AC3	Wait for next function call.

Digital timing diagram

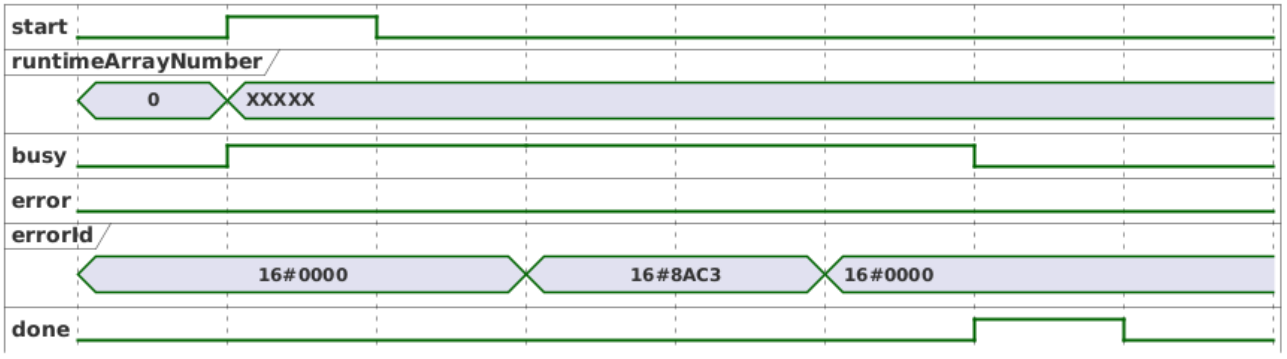
Array created successfully:



Error while creating array:



Max number of "create_db" instances reached:



See also

- LRA_ReadRuntimeArray
- LRA_WriteRuntimeArray.
- LRA_DeleteRuntimeArray.

---

1. Some PLCs might support an even smaller number. Check the hardware datasheet. ↩
2. It is recommended to use even array upper bounds, as odd sizes must be rounded up due to a limitation in the underlying function. ↩

LRA_ReadRuntimeArray

Current documentation version: 0.0.1

0.0.1 - Initial Release.

Description

The "LRA_ReadRuntimeArray" is used to read an index position of a *runtime array*.

Runtime arrays can be create with "LRA_CreateRuntimeArray" and written with "LRA_WriteRuntimeArray".

Functional description

This function works synchronously. Therefore, the operation is done in one block call.

Parameters

Parameter	***Declaration***	***Data type***	***Description***
start	Input	Bool	Continuous read of runtime array index while TRUE.
runtimeArrayNumber	Input	DB_DYN	Identifier of array as a number.
runtimeArrayIndex	Input	DInt	Array index to be read.
indexReadValue	InOut	Variant	Value read from runtime array index.
error	Output	Bool	TRUE for one cycle if there was a read error.
errorId	Output	Word	Stores error code.
done	Output	Bool	TRUE for one cycle if the array index was read without errors.

Parameter errorId

Error code (W#16#...)	**Description**
0000	No error.
8001	User requested an array type that is not supported.

Error code (W#16#...)	Description
8007	Array not found.
8008	Read index out of bounds.
8009	No array type found.
800A	Datatype mismatch.

Troubleshooting

Error code (W#16#...)	Description
0000	No troubleshooting necessary.
8001	Check supported datatypes list and monitor function input during runtime.
8007	Check function input and if the read DB exists.
8008	Check index value and array size.
8009	Make sure the array type identifier did not get modified by the user. If not, this is an internal error and the developer must be contacted.
800A	Check provided "indexReadValue" and array datatype.

See also

- LRA_CreateRuntimeArray.
- LRA_WriteRuntimeArray.
- LRA_DeleteRuntimeArray.

LRA_WriteRuntimeArray

Current documentation version: 0.0.1

0.0.1 - Initial Release.

Description

The "LRA_WriteRuntimeArray" is used to write to an index position of a *runtime array*.

Runtime arrays can be create with "LRA_CreateRuntimeArray" and read with "LRA_ReadRuntimeArray".

Functional description

This function works synchronously. Therefore, the operation is done in one block call.

Parameters

Parameter	***Declaration***	***Data type***	***Description***
start	Input	Bool	Continuous write of runtime array index while TRUE.
runtimeArrayNumber	Input	DB_DYN	Identifier of array as a number.
runtimeArrayIndex	Input	DInt	Array index to be written.
indexWriteValue	InOut	Variant	Value to be written to runtime array index.
error	Output	Bool	TRUE for one cycle if there was a write error.
errorId	Output	Word	Stores error code.
done	Output	Bool	TRUE for one cycle if the array index was written without errors.

Parameter errorId

Error code (W#16#...)	**Description**
0000	No error.

Error code (W#16#...)	Description
8001	User requested an array type that is not supported.
8007	Array not found.
8008	Read index out of bounds.
8009	No array type found.
800A	Data type mismatch.

Troubleshooting

Error code (W#16#...)	Description
0000	No troubleshooting necessary.
8001	Check supported data types list and monitor function input during runtime.
8007	Check function input and if the read DB exists.
8008	Check index value and array size.
8009	Make sure the array type identifier did not get modified by the user. If not, this is an internal error and the developer must be contacted.
800A	Check provided "indexWriteValue" and array data type.

See also

- LRA_CreateRuntimeArray.
- LRA_ReadRuntimeArray.
- LRA_DeleteRuntimeArray

LRA_DeleteRuntimeArray

Current documentation version: 0.0.1

0.0.1 - Initial Release.

Description

The "LDA_DeleteRuntimeArray" is used to delete a *runtime array* created by "LRA_CreateRuntimeArray". The memory is then freed for further use.

Values can be read and written using the "LRA_ReadRuntimeArray" and "LRA_WriteRuntimeArray".

> ⚠ **Warning**
>
> The following limitations apply:
>
> - The minimum number of necessary function calls for deletion is $n + 1$, where n is the number of DBs that must be deleted.
> - All other specificities and limitations of the default function "DELETE_DB" also apply.

Functional description

This function works asynchronously. Therefore, depending on the size and array datatype, it might require multiple calls to complete its task.
If the done bit is set, the deletion was successful.
If the error bit is set, the deletion failed. Check the error code for more details.

> 🖉 **Note**
>
> Always check the done and error bits to correctly evaluate if the job has been completed or failed.

Parameters

Parameter	*Declaration*	*Data type*	*Description*
start	Input	Bool	Trigger start of array deletion.
runtimeArrayNumber	Input	DB_DYN	ID of array to be deleted.
busy	Output	Bool	TRUE while array deletion is ongoing.
error	Output	Bool	TRUE for one cycle when there was an error in the array deletion.
errorId	Output	Word	Stores error code.
done	Output	Bool	TRUE for one cycle when the array deletion was successful.

Parameter errorId

Error code (W#16#...)	Description
0000	No error.
8007	Array was not found.
800B	Linked array not found.
8B0C	Internal error.
8BC3	Max number of "DELETE_DB" instances was reached. In this case, the error bit is **not** set and a retry is done in the next block call.

Troubleshooting

Error code (W#16#...)	Description
0000	No troubleshooting necessary.
8007	Check supported datatypes list and monitor function input during runtime.
800B	Check if the DB was deleted somewhere else in the software.
8B0C	Contact library developer.
8BC3	Wait for next function call.

Digital timing diagram

Array deleted successfully:

Error while deleting array:

See also

- LRA_CreateRuntimeArray.
- LRA_ReadRuntimeArray.
- LRA_WriteRuntimeArray.