Communication in the ARION network – table definition

Abstract

Implementing a communication network of extension modules defined in a table.

Author: Jiří Palát, Zbyněk Říha
File: ap0025_en_03.pdf

Attachments

File content: ap0025_en_02.zip

arion_p1_en_02.dso	Example No. 1 – Programme operation **DM-AI12**
arion_p2_en_02.dso	Example No. 2 – Programme operation **DM-AO8x**
arion_p3_en_02.dso	Example No. 3 – Programme operation **DM-DI24**
arion_p4_en_02.dso	Example No. 4 – Programme operation **DM-DO18**
arion_p5_en_02.dso	Example No. 5 – Programme operation **DM-PDO6NI6**
arion_p6_en_02.dso	Example No. 6 – Programme operation **DM-RDO12**
arion_p10_en_01.dso	Example No. 10 – Programme operation **DM-UI8DO8**
arion_p11_en_01.dso	Example No. 11 – Programme operation **DM-UI8RDO8**
arion_p12_en_01.dso	Example No. 12 – Programme operation **DM-UI8AO8U**
arion_p13_en_01.dso	Example No. 13 – Programme operation **AMR-OP7x (AMR-OP6x, AMR-OP3xA, AMR-OP3x)**

Contents

Revision history

Version	Date	Changes
001	10. 10. 2008	New document
002	02. 08. 2013	Document updated according to the behaviour of DetStudio 1.7.4, chapters 2.3., 2.4., 2.5., 3.2.1, 3.3.3 and 4. updated; existing examples updated, obsolete examples removed, new examples added.
003	26. 01. 2015	Information about new products (**AMR-OP60, AMR-OP3xA**) added. Chapters 2.3.2 and 4.2 amended.

Related documentation

1. Help tab for the PseDet section of the DetStudio development environment
   file: Psedet_en.chm
2. Data sheets for **DM-xxx modules**
   files: dm-xxx_d_en_xxx.pdf
3. Operating manual for **AMR-OP70/xx**
   file: amr-op70xx_g_en_xxx.pdf
4. Operating manual for **AMR-OP60/xx**
   file: amr-op60xx_g_en_xxx.pdf
5. Operating manual for **AMR-OP3x**
   file: amr-op3x_g_en_xxx.pdf
6. Operating manual for **AMR-OP3xA**
   file: amr-op3xa_g_en_xxx.pdf
7. Technical guide for the ARION protocol
   file: arion_ms_en_xxx.pdf
8. Application note AP0002 – Communication in MP-Bus network
   file: ap0002_en_xx.pdf
9. Application note AP0016 – Principles of RS485 interface usage
   file: ap0016_en_xx.pdf
10. Application note AP0017 – Counter inputs, measuring rotations and impulses
    file: ap0017_en_xx.pdf
11. Application note AP0028 – OpenTherm device in ARION network
    file: ap0028_en_xx.pdf
12. Application note AP0038 – Use of digital outputs as frequency or impulse outputs
    file: ap0038_en_xx.pdf

1 Definitions of terms

DM-xxx Modules
Modules allowing extended number of control system inputs and outputs by means of the ARION communication network.

Programmable on-wall controllers AMR-OPxx
Programmable on-wall controllers **AMR-OP7x**, **AMR-OP6x**, **AMR-OP3x and AMR-OP3xA** communicating in the ARION network that serve to measure temperature, set a temperature correction request, regimes and other parameters (depending on types of controllers and their firmware).

DetStudio
A development environment by the AMiT company serving for control systems parametrization. This environment is available at amitomation.com.

RS485
It is a half-duplex serial bus allowing for multiple units to communicate at a single signal pair. More information is available in the document AP0016 – Principles of RS485 interface usage.

Network
An aggregate designation of technical means that realize connection and data transfer between connected devices.

Signal
Digital/analogue input or output.

Data type
Defines the type of a signal group. The ARION network is able to transfer various data types: analogue inputs AI, analogue outputs AO, digital inputs DI, digital outputs DO and special data type Spec.

Node
Defines signals of a single data type of an extension module in the ARION network. Each extension module in the ARION network may be defined by one or several nodes. Therefore, we distinguish single-node and multiple-node modules.

1.1 Software requirements

The table definition of the ARION, described in this Application Note, is enabled in the DetStudio development environment, version 1.0.76 or higher, provided the operating system is NOS version 3.50 or higher.

2 ARION protocol

ARION is a communication protocol for communication between AMiT control systems with extension I/O modules. These extension I/O modules allow increasing the number of inputs/outputs (both digital and analogue) of the control system. The maximum number of extension modules connected into a single communication network is 63 which is able to achieve extension to the control system e.g. by up to 1512 digital (63× **DM-DI24**) or by up to 756 analogue inputs (63× **DM-AI12**). Modules can be combined freely.

ARION protocol is a serial half-duplex protocol, which also brings certain limitations. In the number of devices connected increases, time requirements for data transfer from/to modules increases as well; prolonged period of possible communication with modules connected corresponds to it as well. Find below the calculation of a minimum period with which communication with modules connected is possible.

2.1 Communication mode

There are two modes of communication with extension module in the ARION network:

- **Periodic mode** the NOS does it automatically; inputs/outputs are accessed by means of SW modules similarly as with inputs/outputs directly in the system. The communication period is set in the definition table.
- **Event mode** actual transfer is invoked in an application programme upon request.

2.2 Time ratios

After the control system restart, the communication network initializes. After initialization, the control system sends information on communication modes to individual modules and awaits their responses. If a module provides no response, the system marks it as unconnected. After initialization, actual communication with modules in the ARION network is carried out.

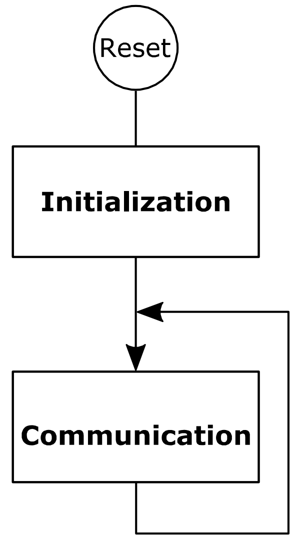


Fig. 1 – The sequence of events in the ARION network after the control system restart

2.3 Calculating the minimum period of communication with modules

Due to the fact that ARION is a serial communication protocol, communication with modules takes more time than operation of inputs/outputs directly in the control system. The minimum time for operation of inputs/outputs in extension modules we need to consider in our calculation when implementing a communication network is stated below.

Extension I/O modules have one or more types of inputs/outputs; the communication with individual modules depends on these types. Find detailed information on individual modules in the Help tab for the PseDet section of the DetStudio development environment.

Note:
*When using the module **`ARI_Select`**, it is suitable to consider only those modules in ARION network that are selected for the given configuration into the calculation. These calculations have to be made for all possible configurations (according to the use of the module **`ARI_Select`**). It is then possible to put the longest calculated time as the minimum communication period. If possible, we recommend putting at least double value.*

2.3.1 Single-node modules

Single-node modules are modules that have only one type of inputs/outputs, e.g. **DM-DO18**, **DM-AI12**, etc.). The minimum communication period with single-node modules recommended is stated in the following table:

Minimum communication period for single-node modules

Data transfer rate	**Minimum communication period [ms]**
9,600 bps	T_{DM} = 50 × DIG + 150 × ANL
19,200 bps	T_{DM} = 25 × DIG + 80 × ANL
38,400 bps	T_{DM} = 15 × DIG + 45 × ANL
57,600 bps	T_{DM} = 12 × DIG + 35 × ANL

Where DIG is the number of extension modules processing the digital signal and ANL is the number of extension modules processing the analogue signal.

The value calculated in this manner is the minimum possible communication period (parameter **`PeriodXX`** in the Arion definition table), in which a network of connected modules is operated. If this period is reduced, there is no guarantee the connected extension modules will work properly. In the optimal situation, the communication period applies is longer than this calculated value.

Default values pre-set in DetStudio are set as follows:

- DI 2,000 ms
- DO 2,000 ms
- AI 2,000 ms
- AO 2,000 ms

If other values are required, this default setting can be changed, or it is possible to change values for specific extension modules directly in the definition table. The communication period defined in the table may vary for each module.

Note:
*The calculation is performed for all types of nodes of extension modules defined that are to be transferred, i.e. they have non-zero value of the parameter **`PeriodXX`**. The minimum time of communication period calculated must be kept in all nodes defined; otherwise a loss of requests and incorrect activities may occur during communication.*

2.3.2 Multiple-node modules

Multiple-node modules are modules with multiple types of inputs/outputs. These include:

- Modules type **DM-xxx** with multiple inputs/outputs (e.g. **DMPDO6NI6**, **DM-UI8DO8**, etc.).
- Programmable controllers **AMR-xxx** (e.g. **AMR-OPxx**, **AMR-RTVxx**, etc.).

In multiple-node modules, we can choose one of the following communication modes:

- All channels communicate with the same period.
- Analogue channels have communication periods different from digital channels.

A special chapter then involves communication with products type **AMR-OPxx** (which have AMiT firmware, or a sample application that works with so called registers in the ARION network), or general communication with controllers **AMR-xxx** which work with so called registers.

Calculation of time ratios in identical communication periods

Minimum communication period recommended for multiple-node modules is determined in the same way as in single-node modules, but all types of inputs/outputs must be calculated for each module.

E.g. **DM-PDO6NI6** has digital outputs and analogue inputs. Digital outputs can be used as classic DO (calculated in the formula as DIG parameter) or as PDO (calculated in the formula as ANL parameter). When using digital outputs as classic DO, the minimum communication period with this module for speed of 19,200 bps will be as follows:

$T_{DM} = 25 \times 1 + 80 \times 1 = 105$ ms

Communication with individual input/output types does not have to be the same, e.g. Digital outputs can be written with period 500 ms and analogue inputs read with period 2 s.

Calculation of time ratios in different communication periods

In practice, it may often be beneficial to divide periodic communications into two groups with various requirements on data transfer frequency. This is directly recommended for controllers **AMR-xxx** which use so called registers, or controllers **AMR-OPxx** with AMiT firmware – see the following chapter. In such cases, time ratios are determined in the following way:

1. All periodic transfers in the network are divided into two groups. One group (usually digital signals) is transferred with a shorter period T_1 (i.e. faster), the other group (usually analogue signals, but not necessarily all of them) is transferred with a longer period T_2 (i.e. more slowly).
2. A desired ratio between the periods T_1 and T_2 is determined (e.g. the first group communicates five times more often than the other group). For the purpose of further calculation, this ratio is designated with the letter P to make:

   $T_2 = P \times T_1$

   The value of ratio P may be e.g. 2, 3, 5 or 10 – the higher the value, the shorter the minimum period for signals from the first group compared to the minimum period of signals from the second group.
3. A minimum communication period is determined independently for each group that would apply in a fictitious case when the network would only communicate with the specific group separately. These periods are designated as T_{S1} and T_{S2}. They are determined by the procedure specified in the previous chapter, but the calculation of T_{S1} only concerns nodes and signals from the first group (the other group is omitted), and the calculation of T_{S2} only includes nodes and signals from the second group.

4. The following applies for minimum periods T_1 and T_2 in a real situation when both groups of signals transmit together in a single bus:

   T1 = TS1 + TS2 / P

   T2 = TS1 × P + TS2

   The resulting value T_1 is the minimum period for the first group of nodes (or signals, more specifically) on multi-node modules; the value T_2 (P-times longer) is the minimum period for nodes (or signals, more specifically) from the second group. After a suitable round-up, we may put these periods into the tables to nodes of the the first and second group.

On-wall controllers AMR-OPxx

On-wall controllers **AMR-OPxx** with AMiT firmware or a sample project working with so called registers in ARION network are multiple-node modules defined by digital inputs (DI) and input-output registers (REG). The communication mode with digital inputs is different from the one used in standard multi-node modules and the communication time is therefore shorter. The minimum communication period recommended with **AMR-OPxx** controllers is stated in the following table:

Minimum period for AMR-OPxx controllers

Data Transfer Rate	Minimum communication period [ms]
9,600 bps	T_{DI} = 30 × AMR
19,200 bps	T_{DI} = 16 × AMR
38,400 bps	T_{DI} = 10 × AMR
57,600 bps	T_{DI} = 8 × AMR

Where AMR is the number of **AMR-OPxx** controllers. The minimum period is determined by the sum of minimum period values for DI.

AMR-xxx controllers

All products of the line **AMR-xxx** are freely programmable. Determining the minimum communication period therefore depends on the programmed method of entering data into the ARION network. Regulators are able to enter data into the ARION network:

- by means of channels AI, AO, DI, DO,
- by means of so called registers in combination with DI channel,
- only by means of so called registers.

Use of AI, AO, DI, DO

In this data entry method, it is necessary to perform the calculation according to the chapters above where communication of channel AI, AO, DI and DO is described for modules type **DM-xxx**.

Combination of registers with channel DI

This method of communication is identical with the procedure stated for on-wall controllers **AMR-OPxx**. The controller informs about its changes in provided registers into the ARION network by means of the DI channel. All registers are then read based on the information about the change. To make an actual calculation, use the table stated in chapter “On-wall controllers **AMR-OPxx**”. The controller can define maximum 9 registers.

Using independent registers

In this data entry method, it is necessary to consider the number of controllers in the ARION network as well as the number of registers (REG) defined in the controller. The controller can define maximum 9 registers. The resulting communication table with one controller is determined by the following table.

Minimum communication period with a single controller

Data Transfer Rate	Minimum communication period [ms]
9,600 bps	T_{REG} = 50 + (25 × REG)
19,200 bps	T_{REG} = 40 + (15 × REG)
38,400 bps	T_{REG} = 20 + (10 × REG)
57,600 bps	T_{REG} = 15 + (8 × REG)

Where REG is the number of registers communicated from the controller in the ARION network.

Therefore, we can determine the duration of communication with a single controller that contains the number of registers (REG) used in the calculation, according to the aforementioned table.

2.4 Connection loss detection (GuardTime)

The value of parameter **`GuardTime`** represents the amount of time after the communication failure the output modules set all outputs to a secure state If no valid communication frame arrives into the extension module after the defined period expires, the extension model detects a communication failure. If it is an output module, it sets all its outputs in a secure state. The secure state is set firmly and cannot be changed by the user. After the communication is renewed, it is again possible to set output module values required.

Secure state for various output types

Output type	Secure state
Digital outputs	0 V
Relay outputs	Open
Analogue Outputs	0 V

The connection loss detection period always needs to be set (parameter **`GuardTime`**) with respect to the communication network load and to requirements of the technology controlled. In a period of communication with modules shorter than 10 s **the value `GuardTime` should be at least 2.5 times of this period**. In a period of communication with modules 10 s or longer the value **`GuardTime`** should be set in order to timely secure the controlled technology in case of a communication failure. The value 0 is set to disable the connection loss detection.

2.4.1 GuardTime in application of AMR-OPxx controllers

If we use the module **`AmrOp3x7xA`** for communication with **AMR-OPxx** controllers, the direct consequence will be the necessity to **increase the parameter `GuardTime` value in all ARION modules** by times stated in the following table:

Increasing the `GuardTime` value when using AMR-OPxx controllers

Data Transfer Rate	Prolonging the GuardTime period [ms]
9,600 bps	T_{GD+} = 300 × AMR
19,200 bps	T_{GD+} = 160 × AMR
38,400 bps	T_{GD+} = 80 × AMR
57,600 bps	T_{GD+} = 60 × AMR

Where AMR is the number of **AMR-OPxx** controllers.

2.5 Example of time ratio calculation

The ARION network is defined according to the following picture.

Arion

Options

Network parameters: Port= 0 Baud= 19200 Id=0

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	1	0	-	-	-	2000
2	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	2	0	2000	-	-	-
3	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	3	0	-	2000	-	-
4	DM-RDO12	12x switching relays 250V/6A	4	0	-	2000	-	-
5	DM-PDO6NI6	6x Ni1000, 6x digital OUT 24V DC, 1 A, galvanic isolation	5	0	-	2000	2000	0
6	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	7	0	2000	-	-	-
7	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	8	0	2000	-	-	-
8	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	9	0	2000	-	-	-
9	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	10	0	2000	-	-	-

Fig. 2 – Defined ARION network

For now, the table still has default communication period values. Communication with **DM-xxx** modules will only take place periodically. Transfer of values type DI in **AMR-OPxx** controllers will also take place periodically and in values type AI, AO and DO (in **AMR-OPxx** controllers) will be written according to events (i.e. these values are not counted in the calculations). In the module **DM-PDO6NI6** the analogue inputs will be periodically read and digital outputs written that will be used as classic DO (not PDO).

2.5.1 Calculation of time ratios in identical communication periods

We assume that all node types of defined ARION modules are transferred in the same period. For the speed 38,400 bps, we use the following formula based on the previous tables:

$T_{DM} = 15 \times DIG + 45 \times ANL$,

The values for the defined ARION network are:

DIG = 8 (**DM-DI24**, **DM-DO18**, **DM-RDO12**, **DM-PDO6NI6,** 4 × **AMR-OPxx**),

ANL = 2 (**DM-AO8U**, **DM-PDO6NI6**),

We put in the values and we get:

$T = T_{DM} = 15 \times 8 + 45 \times 2 = 120 + 90 = 210$ ms

The minimum period for all nodes is 210 ms.

If possible, **use at least double period**, or operate critical modules with shorter period and less important modules with longer period – see the following calculation. Using longer communication periods guarantees higher communication reliability (as it does not get influenced by other communications and application programme).

2.5.2 Calculation of time ratios in different communication periods

Since it is recommended to transfer various types of nodes with different periods, we shall calculate the minimum period for this case. The periodic communication will take place with various periods as follows:

Longer period: for AI (**DM-PDO6NI6**).

Shorter period: for DI (**DM-DI24** a **AMR-OPxx)**, AO (**DM-AO8U**) and DO (**DM-DO18**, **DM-RDO12** and **DM-PDO6NI6)**.

We establish 5 as the ratio of the longer period T_2 and the shorter period T_1.

Minimum communication period is determined according to the formulas:

$T_1 = T_{S1} + T_{S2} / P$

$T_2 = T_{S1} \times P + T_{S2}$

where P = 5.

First, we establish fictitious minimum periods for a group of nodes communicating in shorter period. For single-node and multiple-node modules, we use the formula from the table stated in chapter 2.3 for the speed 38,400 bps:

$T_{DM} = 15 \times DIG + 45 \times ANL$

We put in the values:

DIG = 8 (**DM-DI24**, **DM-DO18**, **DM-RDO12**, **DM-PDO6NI6,** 4 × **AMR-OPxx**)

ANL = 1 (**DM-AO8U**)

we get

$T_{S1} = T_{DM} = 15 \times 8 + 45 \times 1 = 120 + 45 = 165$ ms

Similarly, we establish fictitious minimum periods for a group of nodes communicating in longer period.

We put in the values:

DIG = 0

ANL = 1 (**DM-PDO6NI6**)

we get

$T_{S2} = T_{DM} = 15 \times 0 + 45 \times 1 = 0 + 45 = 45$ ms

We put the calculated values into the formulas and we get:

$T_1 = T_{S1} + T_{S2} / P$

$T_1 = 165 + 45 / 5 = 174$ ms ≈ <u>180 ms</u>

$T_2 = T_{S1} \times P + T_{S2}$

$T_2 = 165 \times 5 + 45 = 870$ ms ≈ <u>870 ms</u>

The minimum period for nodes communicating in shorter periods is 180 ms.

The minimum period for nodes communicating in longer periods is 870 ms.

If possible, **use at least double period**. Using longer communication periods guarantees higher communication reliability (as it does not get influenced by other communications and application programme).

The values calculated show that when using various communication period, it is possible to communicate with the given node types more often than when using a single common communication period.

In order to apply the stated ARION network, you may leave the settings from the aforementioned picture.

2.5.3 Determining GuardTime parameter value

When using communiation methods set in the picture “Fig. 2 – Defined ARION network”, the default value for calculation of `GuardTime` (T_{GD}) parameter value is given by the following formula:

$T_{GD} = 2.5 \times T$

Where T is the selected communication period.

We put in the values and we get:

T_{GD} = 2.5 × 2,000 = 5,000 ms.

Since there are **AMR-OPxx** controllers in the ARION network, it is necessary to modify the value calculated according to the table in the chapter 2.4.1 GuardTime in application of **AMR-OPxx** controllers.

For the communication speed selected, 38,400 bps, the resulting **`GuardTime`** for all modules is given by the formula:

T_{GD} = T_{GD} + 80 × AMR

We put in the values and we get:

T_{GD} = 5,000 + 80 × 4 = 5,320 ms ≈ <u>5,500 ms</u>

2.6 Checking the communication period calculation correctness

After we define the network in DetStudio (see chapters below), we can use a direct DetStudio function to check the correctness of our calculations (to calculate time ratios in identical communication periods). The function is available in the toolbar of the ARION bookmark (menu "Options/Calculate minimum period").

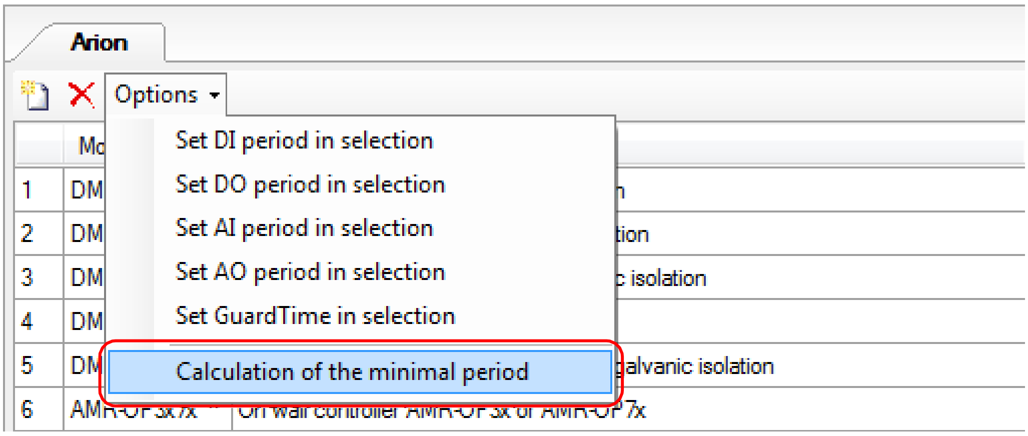


Fig. 3 – Calculating the minimum period of communication using DetStudio

3 ARION network implementation

In order for the entire ARION network to work correctly, it is necessary to correctly design, connect and configure individual network modules and program communication.

When connecting the ARION network, it is necessary to follow recommendations stated in AP0016 – Principles of RS485 interface usage and pay attention to quality realisation of cabling that significantly influences correct functioning. Apart from cabling, network is also affected by quality power supply to individual devices. We recommend using stabilized sources and also overvoltage protections in an an environment with heavy interference.

3.1 Connecting the communication network

Communication by means of the ARION protocol is MASTER-SLAVE (Multi SLAVE) and takes place on the line RS485. Extension modules can be connected to the control system directly in the RS485 interface or in the RS232 interface using a converter (e.g. **DM-232TO485**). The following picture shows both options.

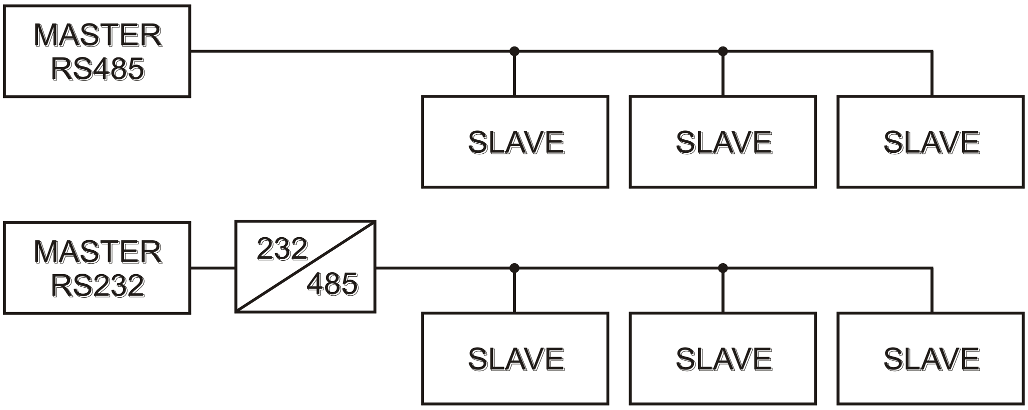


Fig. 4 – Connecting extension modules to the control system

Note:
*The converter **DM-232TO485** connected to RS232 of an AMiT control system is set as controlled by RTS signal.*

3.2 HW configuration of extension modules

Each device connected needs to have an address and communication speed set. The address must be unique for each extension module in the network. All extension modules have to have the same communication speed, according to the communication speed set in the application. Both address and the communication speed are set using the switches on each extension module.

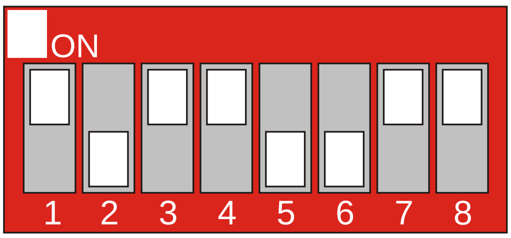


Fig. 5 – Setting communication parameters for extension modules

Switches 1 to 6 serve to set the module address in the ARION network. The address values may range from 1 to 63. Address 0 is not permitted (it is reserved for MASTER).

Note:
Some extension modules may take up multiple addresses (e.g. ***DM-PDO6NI6****). This information is stated in the datasheet of each extension module.*

Switches 7 and 8 (7 = BAUD0, 8 = BAUD1) serve for communication speed setting. Significance of individual address switches as well as communication speed setting options are stated in the following table:

Switch values

ADR0	Value 1
ADR1	Value 2
ADR2	Value 4
ADR3	Value 8
ADR4	Value 16
ADR5	Value 32

BAUD0	**BAUD1**	**Communication speed**
OFF	OFF	9,600 bps
ON	OFF	19,200 bps
OFF	ON	38,400 bps
ON	ON	57,600 bps

The picture above shows the module address 13 and communication speed 57,600 bps.

Note
All changes in individual switch positions manifest only after the extension module restart (unplugging and replugging the power supply).

In case the device in ARION network is not equipped with switches, the following communication parameters must be set on it:

- Using the service menu (e.g. **AMR-OP7x**),
- By means of an application within the user programme (e.g. **AMR-DI2RDO2**).

3.2.1 LED descriptions

All extension modules are equipped with indication LED lights that allow for visual check-ups of their activities. The following tables describe LED functions for various modules.

System LED

Module	**LED**	**Function**
All modules	PWR	Lights in case supply voltage is connected.
	RUN	Blinks in intervals ca 2 s (1:1) – indicates that the module is functional.

Communication LED

Module	**LED**	**Function**
All modules	RxD	Lights when data is received from ARION network.
	TxD	Lights when data is transmitted into ARION network.
DM-MPBUS	RxMP	Lights when data is received from MP-Bus network.
	TxMP	Lights when data is transmitted into MP-Bus network.
DM-OT	RxOT	Lights when data is received from OpenTherm network.
	TxOT	Lights when data is transmitted into OpenTherm network.

LED indicating input/output status

Module	LED	Function
DM-DI24	DI0 to DI23	Lights if the value log is set on the corresponding input. 1.
DM-DO18	DO0 to DO17	Lights if the value log is set on the corresponding output. 1.
DM-RDO12	RL0 to RL11	Lights if the corresponding relay output is switched on.
DM-AI12	AI0 to AI11	Lights in case the value set to input is within the AD converter range.(1)
DM-AO8U(I)	AO0 to AO7	Lights if the required output value is higher than ca 0 V (0 mA) or according to the selected settings.(2)
DM-PDO6NI6	NI0 to NI5	Lights if the sensor Ni1000 is connected.
	DO0 to DO5	Lights if the output value is in log. 1.
DM-UI8DO8	UI0 to UI7	Lights in case the value set to input is within the AD converter range.(1)
	DO0 to DO7	Lights if the output value is in log. 1.
DM-UI8RDO8	UI0 to UI7	Lights in case the value set to input is within the AD converter range.(1)
	RL0 to RL7	Lights if the corresponding relay output is switched on.
DM-UI8AO8U	UI0 to UI7	Lights in case the value set to input is within the AD converter range.(1)
	AO0 to AO7	Lights if the required output value is higher than ca 0 V or according to the selected settings.(2)

In some cases, behaviour of LED indicator lights can help visually interpret the cause of the problem occurred.

Note

(1) *Exact value is in the interval of 0.7 % of the range to 99.7 % of the range.*

(2) *In modules, behaviour of LED lights corresponding to individual analogue outputs can be determined by means of ARION. We can set this behaviour by means of two fictitious outputs placed on positions 8 and 9 in the module. The LED behaviour is as follows:*
The value on outputs is lower than the value on output on position 8 – LED is off.
The value on outputs is higher than the value on output on position 8 and lower than the value on output on position 9 – LED is on.
The value on outputs is higher than the value on output on position 9 – LED is blinking.
It is a fact that 0 to 32767 corresponds to 0 % to 100 % of the analogue output range.
If no values are wirtten into outputs 8 and 9 (i.e. the values are zero), LED is on for values on outputs higher than zero.

3.3 SW configuration – table definition

DetStudio development environment from version 1.0.76 on allows us to define the network of extension I/O ARION modules using a table. In order to define the ARION table, we need to program the link between database variables and ARION network extension modules. When using periodic communication in the ARION network, the application code is similar as in the use of local inputs/outputs directly in the control system, but instead of modules **`DigIn`**, **`AnOut`** etc. we use modules **`ARI_DigIn`**, **`ARI_AnOut`** etc. In the table definition, a dynamic change of the ARION network assembly, see chapter 3.3.4. If the user wants to test the status of network nodes and data transfer, they can do so by means of special modules **`ARI_State`** that serve exactly this purpose. We strongly recommend you use them.

Implementing communication with extension modules in the ARION network consists of three steps:

1. Setting the communication line parameters.
2. Making the definition table.
3. Programming the links between ARION modules and database variables.

3.3.1 Setting ARION network communication parameters

ARION network parameters are displayed in the properties window accessible by clicking the item “Arion” located in the folder “Communication” in the project window.

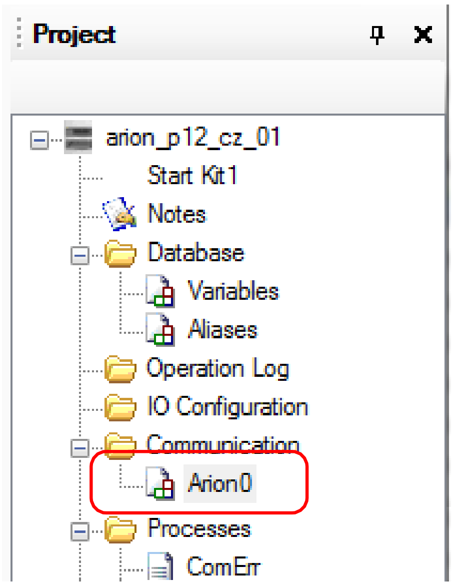


Fig. 6 – Item “Arion” in the folder “Communication”

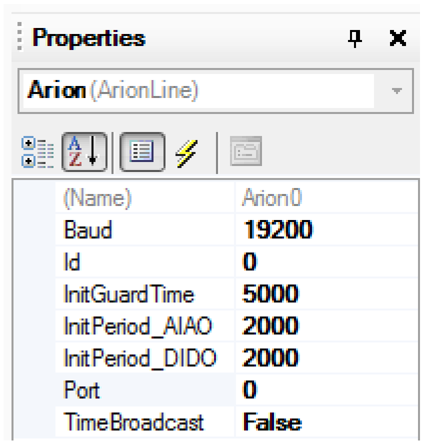


Fig. 7 – Editing ARION communication parameters

The parameters signify the following:

Baud – ARION network communication speed in bps. Acceptable values are 9,600, 19,200, 38,400 and 57,600 bps.

Id – ARION network identifier. Serves for dynamic configuration described in more detail in chapter 3.3.4.

InitGuardTime – Default value of **`GuardTime`** parameter used upon entry of a new module of the given ARION network (does not influence lines already entered).

InitPeriod_AIAO – Default value of the communication period with analogue-type nodes. The value is inserted into parameters **`PeriodAI`** a **`PeriodAO`** upon entering a new module of the given ARION network where applicable (does not affect lines already entered).

InitPeriod_DIDO – Default value of the communication period with digital-type nodes. The value is inserted into parameters **`PeriodDI`** a **`PeriodDO`** upon entering a new line where applicable (does not affect lines already entered).

Port – number of a control system communication port on which communication through ARION protocol is defined.

TimeBroadcast – enables/disables periodic time mark distribution (control system time) to all network nodes able to receive time marks (e.g. **AMR-OP7x**).

3.3.2 Table definition of ARION network extension modules

The list of defined ARION network modules is accessible by double-clicking the item **Communication/Arion** in the project window.

Arion

Options ▾ Network parameters: Port= 1 Baud= 38400 Id=0

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	0	-	-	4000	-
2	DM-AO8I	8x analogue OUT 0-20mA, 12 bit resolution	2	0	-	-	-	2000
3	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	3	0	-	-	-	2000
4	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	4	0	1000	-	-	-
5	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	5	0	-	3000	-	-
6	DM-RDO12	12x switching relays 250V/6A	6	0	-	2000	-	-
7	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	7	0	3000	-	-	-
8	DM-PDO6NI6	6x Ni1000, 6x digital OUT 24V DC, 1 A, galvanic isolation	8	0	-	1000	3000	0
9	DM-UI8DO8	8 combined inputs, 8 digital outputs	10	0	1000	1000	3000	-
10	DM-UI8AO8U	8 combined inputs, 8 analog voltage outputs	11	0	1000	-	3000	2000

Fig. 8 – Table with defined ARION network modules

We can add ARION modules into the table by dragging them from the toolbox by using the button in the upper left corner of the table, by pressing the key **Insert** from the context menu.

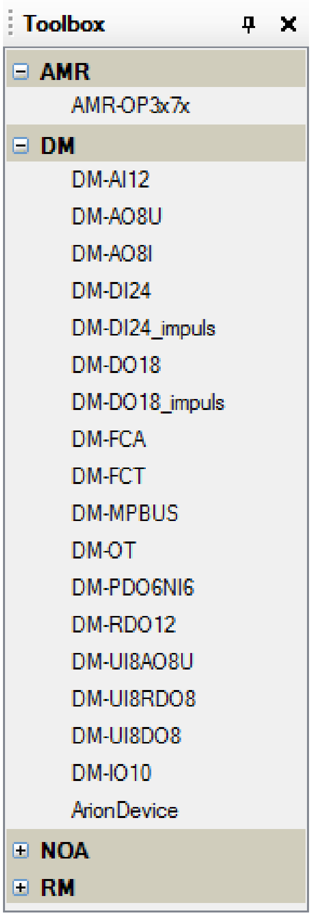


Fig. 9 – Selecting extension modules from the toolbox window

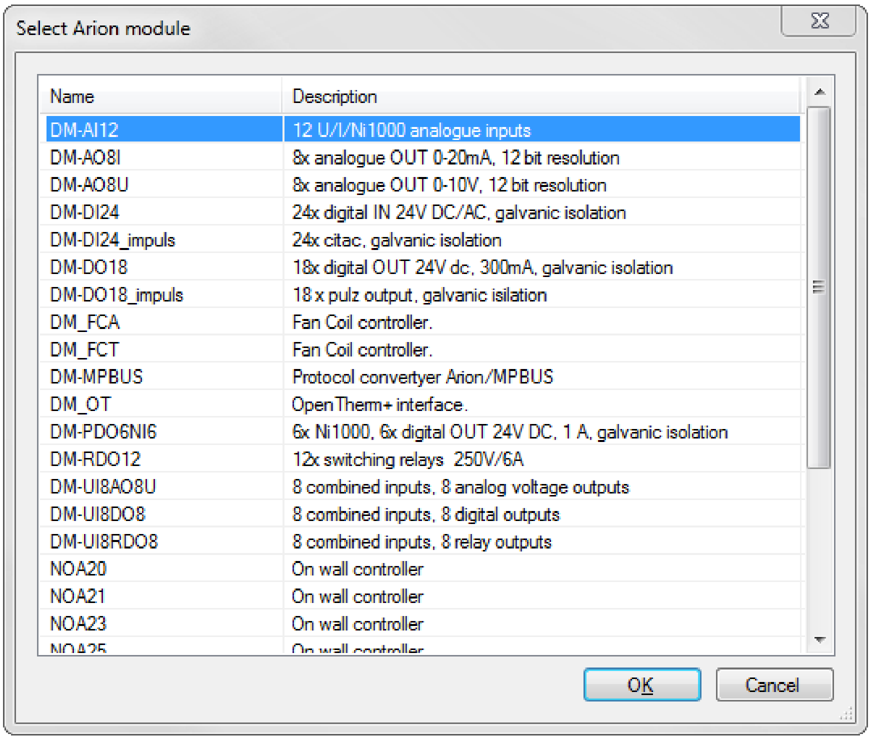
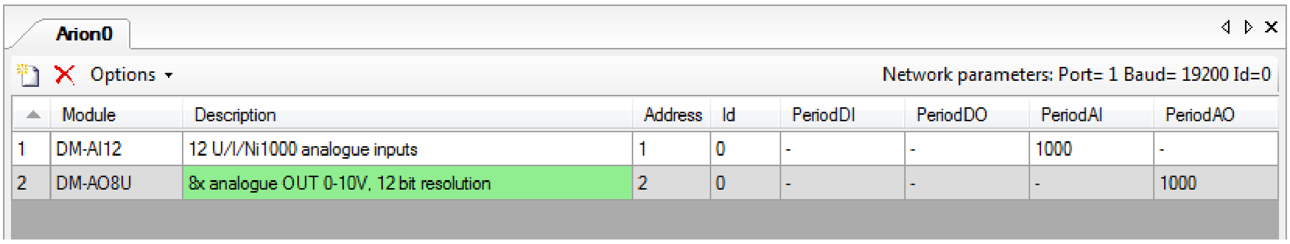


Fig. 10 – Window with extension modules selection

Each extension module corresponds to one row of the **Arion** table. We can delete modules by using the key **Delete,** by means of the button or from the context menu.



	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	0	-	-	1000	-
2	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	2	0	-	-	-	1000

Fig. 11 – Table with defined extension modules

Parameters of the selected ARION communication line are located on the right side of the toolbar. There are three buttons on the left side of the toolbar – for adding/removing a module (or a group of modules) ARION, for batch changes in parameters **`PeriodXX`** and **`GuardTime`** and for calculation of the minimum recommended communication period.

ARION extension modules do not have accessible all parameters **`PeriodXX`**, but only those **`PeriodXX`** parameters that make sense for the given extension module. Non-editable parameters are marked with a hyphen instead of their value.

When entering a new extension module, DetStudio automatically sets its address to the lowest possible number (it checks for collisions of addresses set).

Note
*If you want to mark multiple extension modules, hold the **Shift** key and click the end of the desired block or hold the **Ctrl** key and click the desired modules.*

*One ARION network is able to connect maximum 63 modules. Some modules may take up multiple addresses (e.g. **DM-PDO6NI6** takes up 2 addresses), which decreases the maximum number of ARION modules by this number (e.g. maximum 31 modules **DM-PDO6NI6** on one ARION network).*

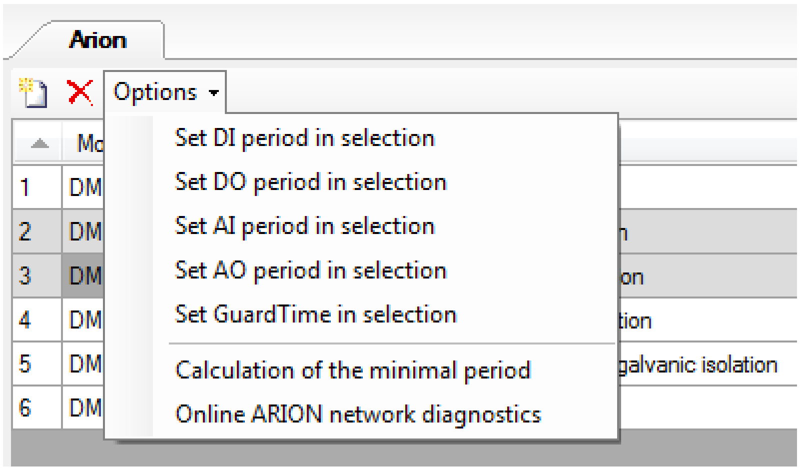


Fig. 12 – Toolbar – options for setting ARION module parameters

Active row in the table of ARION modules is light-grey. An active editable cell in the table is green, non-editable cell is dark-grey. Parameters from the active line of ARION modules table are also displayed in the properties window, see the following picture.

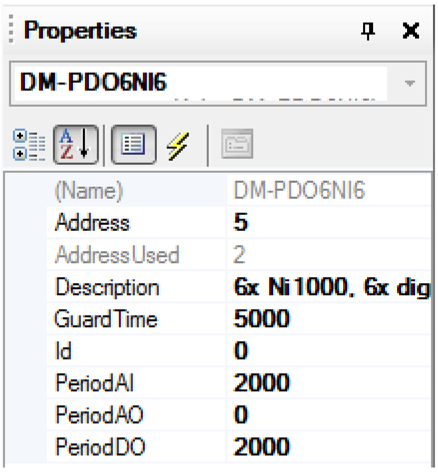


Fig. 13 – Properties window for the module **DM-PDO6NI6**

ARION module parameters have the following descriptions:

No description (not stated in the properties window) **–** module order in the table.

Module (in the properties window **Name**) – ARION module name.

Description (in the properties window **Description**) – text with module description, the default text can be changed deliberately.

Address (in the properties window **Address**) – module address in the ARION network.

Id – ARION module identifier. Used in dynamic assembly configuration, see chapter 3.3.4 Dynamic configuration.

PeriodDI – digital inputs communication period in miliseconds.

PeriodDO –digital outputs communication period in miliseconds.

PeriodAI – analogue inputs communication period in miliseconds.

PeriodAO – analogue outputs communication period in miliseconds.

Compared to the ARION table, the properties window also states the following parameters:

AdressUsed – the number of addresses the given module takes up.

GuardTime – time for connection loss detection in miliseconds.

Setting the correct operation period for individual extension modules (parameter **`PeriodXX`**) significantly influence correct functioning of the entire ARION network. The calculation of the minimum period is stated above. We recommend you use the value calculated in this method as the minimum period of the **`PeriodXX`** parameter. The value calculated in this method should not shorten, otherwise there is no guarantee that the entire network functions properly. Recommendations for **`GuardTime`** parameter settings are stated above.

3.3.3 Program operation of extension modules

Communication with extension ARION modules is based on periods or events. The actual physical transfer between the ARION network buffer and remote network nodes is then invoked according to the the type (periods or events). The following modules serve to read/write data from/to the ARION network buffer:

- **`ARI_AnIn`** – reading analogue value from ARION buffer and conversion.
- **`ARI_NumAI`** – reading numeric value of the analogue input (A/D converter value).

- **`ARI_AnOut`** – converting and writing the analogue value into the ARION network buffer.
- **`ARI_NumAO`** – writing numeric value (A/D converter value) of the analogue output.
- **`ARI_DigIn`** – reading states of digital inputs from ARION network buffer.
- **`ARI_DigOut`** – writing states of digital outputs into ARION network buffer.
- **`ARI_RegIn`** – reading register values from ARION network buffer register.
- **`ARI_RegIn`** – writing register values into ARION network buffer register.

Selected AMiT products that provide multiple information in the ARION network (e.g. **AMR-OPxx**) can be operated in two methods:

- By combination of the aforementioned modules.
- By an autonomous module designated for the given HW type.

In case of the **AMR-OPxx** product mentioned, the module **`AmrOp3x7xA`** can be used – operation of on-wall controllers **AMR-OPxx** in the ARION network.

Note
The detailed description of the modules is available in the Help tab for the PseDet section of the DetStudio development environment.

Periodic communication

In case of periodic communication, the appropriate communication periods (**`PeriodXX`** parameters) are set to a non-zero value in the ARION network definition table. According to this parameter value, the physical transfer is invoked automatically between the ARION network buffer and extension I/O modules.

Attention
*Periodic reading/writing of valid data from/to the corresponding channel of a remote node is performed always in the time specified by the communication period with the remote node. It is therefore futile to use the **`ARI_xxx`** module to read/write data from/to the control system internal buffer more often than in a double period of communication with the remote node. The period of communication with a remote node should be at least half compared to the period of reading/writing data from/to the control system internal buffer.*

Event communication

In case of non-periodic communication (parameters **`PeriodXX`** have zero value), or if a request to invoke physical data transfer outside the set communication period is issued, we use the module **`ARI_Trig`**. This module invokes data transfer between the ARION network buffer and the network remote node, defined by its address and data type (AI/AO/DI/DO). The module does not wait for the communication to finish. If the application wants to respond to the communication finish, it has to test the status of modules **`ARI_State`**. Before the communication finishes, the modules **`ARI_DigIn`**, **`ARI_AnIn`** etc. return the values acquired from the last communication finished, because the ARION network buffer contents has not been changed yet.

Example of an event communication:

```
//Event communication - writing on digital outputs
ARI_DigOut 1, 0, 7, States[0,0], 0x0000  //change in DO buffer
If WriteDO.0
     ARI_Trig 1, 3                       //invoking actual transfer
     Let WriteDO = 0
EndIf
```

Attention
The event of data reading/writing by module **`ARI_Trig`** *from/to a specific channel of a node in the ARION network is only performed once within one process. It is therefore futile to use mutiple* **`ARI_Trig`** *modules in a single process to read/write from/to one channel. Data will only be read/written by means of the first* **`ARI_Trig`** *module in the event.*

Transfer status

If the application requires to know the status of the node connected, we must use module **`ARI_State`**. This module returns the current node-connection status (parameter **`State`**) and status of transfer of the selected data type (parameter **`Transfer`**) for the node of the selected type and address.

Attention
If the communication fails, the last known values read from the ARION network remain in the control system's internal buffer. We recommend you always use the **`ARI_State`** *module at least to discover the communication failure with given nodes in the ARION network.*

Example of communication-failure check:

```
//Learning the node status in the ARION network
     ARI_State 13, ARN_State, 5, ARN_Transf

     If ARN_State.0      //If the initialization went well, we are communicating
          ARI_DigOut 13, 0, 5, DataDO, 0x0000
     Else
          //Code for the required action in case of communication error
     EndIf
//Node initialization failed
     Let COM_Error.0 = not(ARN_State.0)
     ErrSig COM_Error, 0x0001, COM_Alarm_K, 0x0001, COM_Alarm.0, Com_alarm_I.0, 5, 20,
20000, 0, 0, 0
//Communication failed
     Let COM_Error.1 = ARN_State.1
     ErrSig COM_Error, 0x0002, COM_Alarm_K, 0x0002, COM_Alarm.1, Com_alarm_I.1, 5, 20,
20000, 1, 0, 0
```

3.3.4 Dynamic configuration

The table allows us to define multiple ARION network configuration variants that the user can activate in the course of the application operation. In order to activate/deactivate any configuration variant, we need to restart the control system. Therefore, it is possible to create e.g. universal applications for multiple solutions where we can select the specific configuration variation from the menu.

The parameter **`Id`** (see the following picture) and the module **`ARI_Select`** serve to identify the individual configuration variants.

Arion

Options — Network parameters: Port= 1 Baud= 19200 Id=0

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	1	-	-	4000	-
2	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	2	2	1000	-	-	-
3	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	3	3	-	3000	-	-
4	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	4	1	2000	-	-	-
5	AMR-OP3x7x	On wall controller AMR-OP3x or AMR-OP7x	5	3	2000	-	-	-

Fig. 14 – Parameter **Id** for ARION network and extension modules

We state all network configuration variants available in the operation for further selection into the table. We assign a numeric identifier to the individual table items the module **ARI_Select** will link to.

We use the module **ARI_Select** to select the ARION network configuration variation, i.e. we can choose which network extension modules should be active (initialized). We usually place the module **ARI_Select** into a process type INIT. ARION network modification (i.e. setting the corresponding variable influencing the performance of module **ARI_Select**) can be set during application operation in the control system, but the change will be carried out only at the moment of system initialization.

If the module **ARI_Select** is not invoked, the entire definition is valid as it was created in the definition table.

When using the module **ARI_Select**, it is necessary to use the identifier with the value 0 in it (even if no extension module corresponds to the given **Id**). Inducing another **ARI_Select** module only activates those items of the definition table with identifier (parameter **Id**) corresponding to the module parameter. Other items in the table are irrelevant. In this way, you may gradually activate several groups of items from the definition table. If the **ARI_Select** module activates only groups involving extension modules with **Id** different than 0, communication in the network would not be functional.

Note
*We can also use the module **ARI_Select** to enable/disable not only selected groups of ARION modules, but also the entire ARION network.*

Example of dynamic configuration

We have a defined network of ARION extension modules according to the following picture.

Arion

Options — Network parameters: Port= 1 Baud= 19200 Id=2

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	0	-	-	2000	-
2	DM-AO8I	8x analogue OUT 0-20mA, 12 bit resolution	2	1	-	-	-	2000
3	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	3	1	-	-	-	2000
4	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	4	0	2000	-	-	-
5	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	5	2	-	2000	-	-
6	DM-RDO12	12x switching relays 250V/6A	6	2	-	2000	-	-

Fig. 15 – Defined network of ARION extension modules

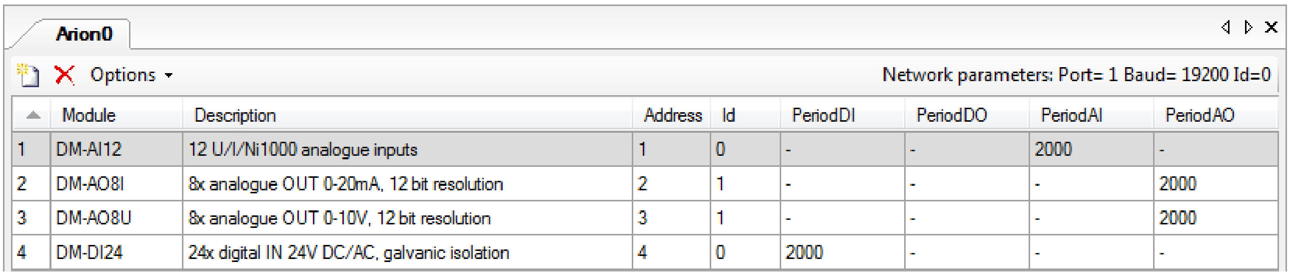
The request will be to initialize input modules (**DM-AI12** and **DM-DI24**) every time and further to make an option to choose between initializing modules of analogue outputs (**DM-AO8I** and **DM-AO8U**) and digital (relay) outputs (**DMDO18** and **DM-RDO12**).

The **`Id`** parameter helps distinguish ARION modules into three groups. The group with **`Id`** = 0 which will be initialized every time, includes extension input modules and the actual definition of the ARION network on the communication port 1 (marked in a red frame in the picture). Another group with **`Id`** = 1 involves modules of analogue outputs (marked in a green frame in the picture). The third group with **`Id`** = 2 involves modules of digital (relay) outputs (marked in a blue frame in the picture). Initialization of the second or third group will be chosen according to the value of 0. bit of the variable **`AriConfig`**.

We achieve the desired function by entering the following code into the process type INIT:

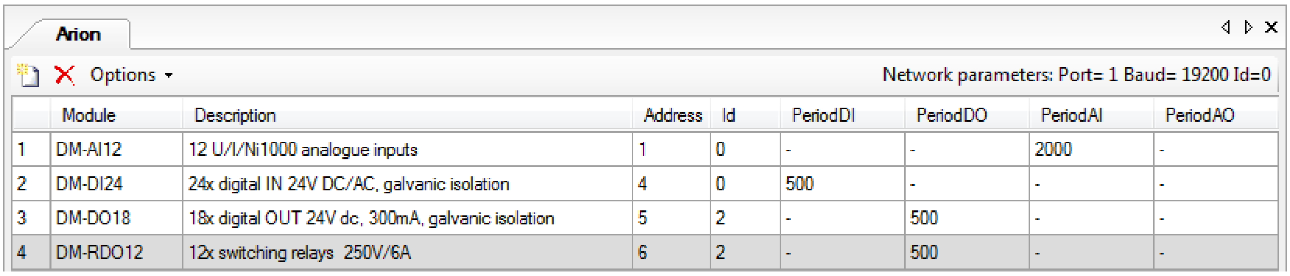
```
ARI_Select 0                //Arion + DM-AI12 + DM-DI24
If AriConfig.0
     ARI_Select 1           //DM-AO8I + DM-AO8U
Else
     ARI_Select 2           //DM-DO18 + DM-PDO12
Endif
```

When meeting the stated condition, the ARION network will be initialized in the programme according to Fig. 16; when the condition is not met, the ARION network will be initialized according to Fig. 17.


Arion0
Options
Network parameters: Port= 1 Baud= 19200 Id=0

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	0	-	-	2000	-
2	DM-AO8I	8x analogue OUT 0-20mA, 12 bit resolution	2	1	-	-	-	2000
3	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	3	1	-	-	-	2000
4	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	4	0	2000	-	-	-

Fig. 16 – ARION network configuration – groups 0 and 1 initialized


Arion
Options
Network parameters: Port= 1 Baud= 19200 Id=0

	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AI12	12 U/I/Ni1000 analogue inputs	1	0	-	-	2000	-
2	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	4	0	500	-	-	-
3	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	5	2	-	500	-	-
4	DM-RDO12	12x switching relays 250V/6A	6	2	-	500	-	-

Fig. 17 – ARION network configuration – groups 0 and 2 initialized

Note
*If the module **`ARI_Select`** activates only groups with **`Id`** = 1 and **`Id`** = 2 (without activating groups with **`Id`** = 0), the communication in the ARION network will not work.*

4 Sample projects

The attachment ap0025_en_xx.zip also features the following applications created for the control system **StartKit**, in DetStudio development environment. Applications can be changed for any control system equipped with a serial communication line, by means of a DetStudio menu "Tools/Changing station type...".

The following table states the overview of AMiT products available that communicate by means of ARION Protocol and can be parametrized in DetStudio development environment.

AMiT products available for ARION

Module name	Module type	I/O number
DM-DI24	Digital Inputs	24
DM-DO18	Digital Outputs	18
DM-RDO12	Relay Outputs	12
DM-AI12	Analogue Inputs	12
DM-AO8I	Analogue Outputs	8
DM-AO8U	Analogue Outputs	8
DM-PDO6Ni6	Multiple-node module	6× Ni1000 6× PDO
DM-UI8AO8U	Multiple-node module	8× DI / AI / Ni1000 8× AO
DM-UI8PDO8	Multiple-node module	8× DI / AI / Ni1000 8× PDO
DM-UI8RDO8	Multiple-node module	8× DI / AI / Ni1000 8× RDO
DM-OT	ARION/OpenTherm converter	–
DM-MPBUS	ARION/MP-Bus converter	–
AMR-xxx	Programmable controllers	Depending on controller type

Note
The list of modules is up to date with the latest modification of this application note. Find an updated overview of all modules available at amitomation.com.

4.1 Example No. 1 – Communication with DM-AI12

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-AI12** (address 1).

Module **DM-AI12** has inputs configured as follows:

Input	Type
AI0, AI1	Ni1000
AI2	0 V to 5 V
AI3	0 V to 10 V
AI4 to AI11	0 mA to 20 mA

ARION network with an extension module **DM-AI12** is defined according to the following picture.

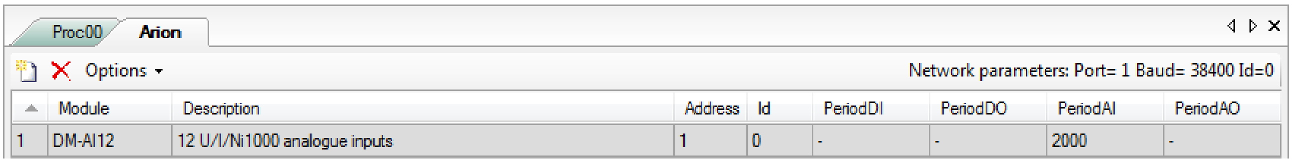


Fig. 18 – Definition of ARION network with **DM-AI12** extension module

Reading values from all analogue inputs:

```
ARI_AnIn 1, 0, 12, Data_all[0,0], param[0,0], 10.000, 0.000, 10.000, 0.000, 10.000
//Converting AI0 input voltage to temperature
Ni1000U2T Data_all[0,0], AI_temperat[0,0], 6180, 15.000, 3920.000
//Converting AI1 input voltage to temperature
Ni1000U2T Data_all[1,0], AI_temperat[1,0], 6180, 15.000, 3920.000
```

A single **`ARI_AnIn`** module is used to read values of all inputs. Since individual signals use various constants for conversion into physical quantity, the matrix of conversion constant of 12 lines is specified in the parameter **`Conversion`** (each line for one signal, see the following table). The module **`Ni1000U2T`** converts the value of voltage in inputs AI0 and AI1 into temperature.

Values of param matrix set

Row	Range	El. min	El. max	Phys. min	Phys. max
0	5	0	5	0	5
1	5	0	5	0	5
2	5	0	5	0	100
3	10	0	10	0	100
4	20	4	20	0	100
…					
11	20	4	20	0	100

Reading the value of one analogue input (AI2):

```
ARI_AnIn 1, 2, 1, AI2, NONE[0,0], 10.000, 0.000, 10.000, 0.000, 100.000
```

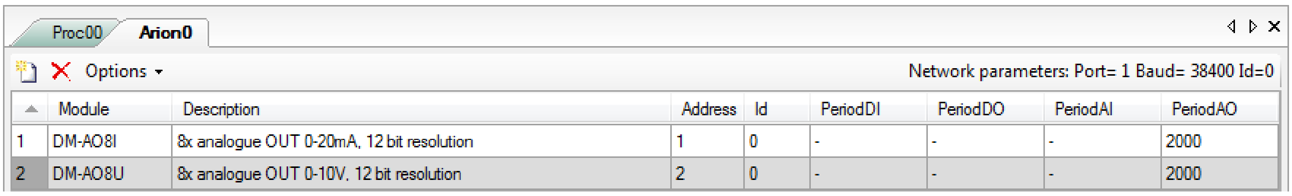
This example is included in the attachment ap0025_en_xx.zip under the name arion_p1_en_xx.dso.

Attention
*ARION network buffer values are zero after installing the application into the control system! If you use module **`Filtr1R`** to filter values measured, we recommend you process is only after establishing communication with module **DM-AI12** successfully (see chapter Transfer status).*

4.2 Example No. 2 – Communication with DM-AO8x

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-AI12** (address 1) and **DM-AO8U** (address 2).

ARION network with extension modules **DM-AO8I** and **DM-AO8U** is defined according to the following picture.



	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-AO8I	8x analogue OUT 0-20mA, 12 bit resolution	1	0	-	-	-	2000
2	DM-AO8U	8x analogue OUT 0-10V, 12 bit resolution	2	0	-	-	-	2000

Fig. 19 – Definition of ARION network with extension modules **DM-AO8I** and **DM-AO8U**

Reading values from all analogue outputs:

```
//Writing into all DM-AO8I outputs
ARI_AnOut 1, 0, 8, Data_all_I[0,0], NONE[0,0], 20.00, 0.00, 20.00, 0.00, 100.00
//Writing into all DM-AO8U outputs
ARI_AnOut 2, 0, 8, Data_all_U[0,0], NONE[0,0], 10.00, 0.00, 10.00, 0.00, 100.00
```

Individual signals use the same constants to convert into physical quantities, that is why it is not necessary to specify conversion constant matrices.

Writing values into one analogue output (AO7):

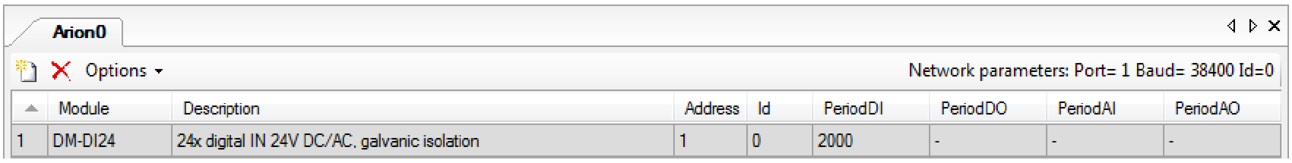
```
//Writing into one output(AO7) DM-AO8I
ARI_AnOut 1, 7, 1, Data_1_I, NONE[0,0], 20.000, 0.000, 20.000, 0.000, 100.000
//Writing into one output(AO7) DM-AO8U
ARI_AnOut 2, 7, 1, Data_1_U, NONE[0,0], 10.000, 0.000, 10.000, 0.000, 100.000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p2_en_xx.dso.

4.3 Example No. 3 – Communication with DM-DI24

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-DI24** (address 1).

ARION network with an extension module **DM-DI24** is defined according to the following picture.



	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-DI24	24x digital IN 24V DC/AC, galvanic isolation	1	0	2000	-	-	-

Fig. 20 – Definition of ARION network with **DM-DI24** extension module

The module **DM-DI24** can be used in the mode of digital inputs or in the mode of counter inputs (the module "DM-DI24_impuls" in the ARION table is used to detect incoming impulses; more detailed description of this mode is available in Application note AP0017 – Counter inputs, measuring rotations and impulses). In this example, we work with the module **DM-DI24** inputs only as with standard DI.

Reading values from all digital inputs:

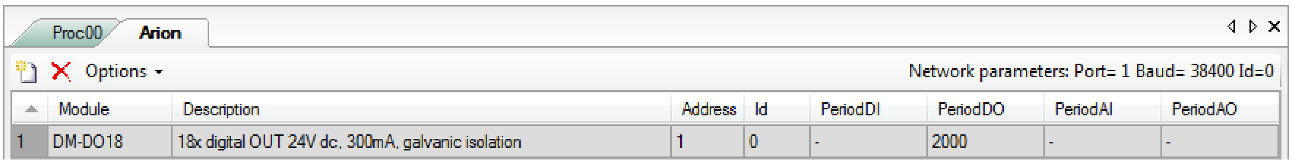
```
ARI_DigIn 1, 0, DataDI_all, 0x0000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p3_en_xx.dso.

4.4 Example No. 4 – Communication with DM-DO18

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-DO18** (address 1).

ARION network with an extension module **DM-DO18** is defined according to the following picture.



	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-DO18	18x digital OUT 24V dc, 300mA, galvanic isolation	1	0	-	2000	-	-

Fig. 21 – Definition of ARION network with **DM-DO18** extension module

We can use the module **DM-DO18** in the mode of standard digital outputs or in the mode of impulse/PWM outputs (the module "DM-DO18_impuls" is used for the generation of impulses/PWM in the ARION table; more detailed description of this mode is available in Application note AP0038 – Use of digital outputs as frequency or impulse outputs). In this example, we work with the module **DM-DO18** outputs only as with standard DO.

Writing values onto all digital outputs:

```
ARI_DigOut 1, 0, 18, dataDO, 0x0000
```

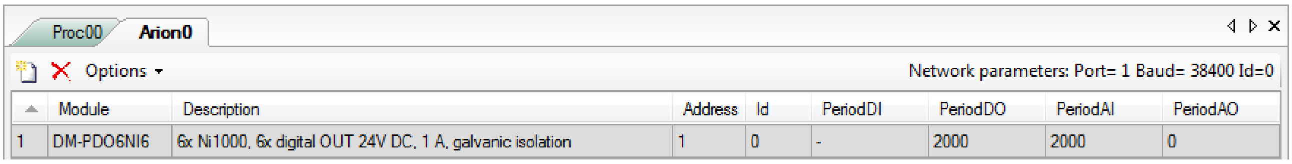
This example is included in the attachment ap0025_en_xx.zip under the name arion_p4_en_xx.dso.

4.5 Example No. 5 – Communication with DM-PDO6NI6

Module **DM-PDO6NI6** takes up 2 addresses on the network (it is therefore possible to only connect 31 such modules into the ARION network). The first of the addresses is assigned to the node DO (PDO) (set on the module switches). The node Ni1000 gets an address higher by 1. Statuses of connections between the control system and the module can be tested only at the node with the first address. The module is defined as one row in the ARION table.

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-PDO6NI6** (addresses 1 and 2).

ARION network with an extension module **DM-PDO6NI6** is defined according to the following picture.



	Module	Description	Address	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-PDO6NI6	6x Ni1000, 6x digital OUT 24V DC, 1 A, galvanic isolation	1	0	-	2000	2000	0

Fig. 22 – Definition of ARION network with **DM-PDO6NI6** extension module

The module takes up 2 addresses (X and X+1, where X is set on the **DM-PDO6NI6** switch). For the stated example, there are digital outputs at the address 1 and Ni1000 inputs at the address 2.

Outputs of the module **DM-PDO6NI6** can be used in the standard digital output mode or in the PWM output mode (more detailed description of PWM mode is available in Application note AP0038 – Use of digital outputs as frequency or impulse outputs). In this example, we work with the module **DM-PDO6NI6** outputs only as with standard DO.

Reading values from all Ni1000 inputs:

```
ARI_AnIn 2, 0, 6, AI_voltage[0,0], NONE[0,0], 5.000, 0.000, 5.000, 0.000, 5.000
//Converting voltage into temperature
Ni1000U2T AI_voltage[0,0], AI_temperat[0,0], 6180, 15.000, 3920.000
Ni1000U2T AI_voltage[1,0], AI_temperat[1,0], 6180, 15.000, 3920.000
Ni1000U2T AI_voltage[2,0], AI_temperat[2,0], 6180, 15.000, 3920.000
Ni1000U2T AI_voltage[3,0], AI_temperat[3,0], 6180, 15.000, 3920.000
Ni1000U2T AI_voltage[4,0], AI_temperat[4,0], 6180, 15.000, 3920.000
Ni1000U2T AI_voltage[5,0], AI_temperat[5,0], 6180, 15.000, 3920.000
```

Attention
ARION network buffer values are zero after installing the application into the control system! If you use module ***`Filtr1R`*** *to filter values measured, we recommend you process is only after establishing communication with module* ***DM-PDO6NI6*** *successfully (see chapter Transfer status).*

Writing values onto all digital outputs:

```
ARI_DigOut 1, 0, 6, DO_data, 0x0000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p5_en_xx.dso.

4.6 Example No. 6 – Communication with DM-RDO12

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-RDO12** (address 1).

ARION network with an extension module **DM-RDO12** is defined according to the following picture.

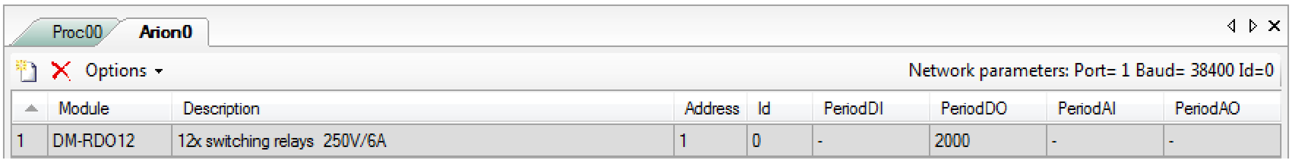


Fig. 23 – Definition of ARION network with **DM-RDO12** extension module

Writing values into all relay outputs:

```
ARI_DigOut 1, 0, 12, dataRDO, 0x0000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p6_en_xx.dso.

4.7 Example No. 10 – Communication with DM-UI8DO8

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-UI8DO8** (address 1) where the first four universal inputs are configured as analogue and the other four configured as digital. In terms of HW, the first four universal inputs are configured in the analogue mode according to the following table.

Universal input	Type
UI0	Ni1000
UI1	0 V to 5 V
UI2	0 V to 10 V
UI3	0 mA to 20 mA

In this case, the ARION network with an extension module **DM-UI8DO8** is defined according to the following picture.

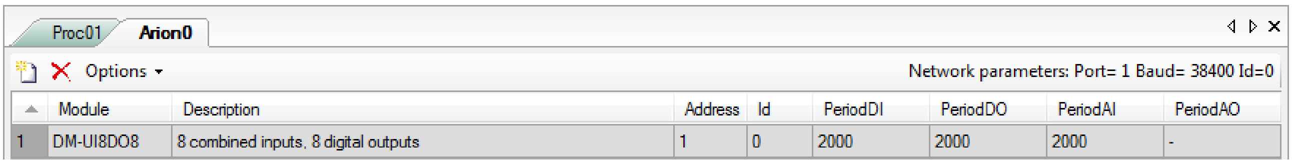


Fig. 24 – Definition of ARION network with **DM-UI8DO8** extension module

Writing values onto all digital outputs:

```
ARI_DigOut 1, 0, 8, dataDO, 0x00000000
```

Reading the first four universal inputs in analogue mode

```
ARI_AnIn 1, 0, 4, dataAI[0,0], paramUI[0,0], 10.000, 0.000, 10.000, 0.000, 10.000
```

```
//Converting measured voltage into temperature
Ni1000U2T dataAI[0,0], UI0_temperat, 6180, 15.000, 3920.000
```

A single **ARI_AnIn** module is used to read all four analogue inputs. Since individual signals use various constants for conversion into physical quantity, the matrix of conversion constant of 4 lines is specified in the parameter **Conversion** (each line for one universal input in analogue mode, see the following table). The module **Ni1000U2T** converts the value of AI0 input into temperature. The remaining values measured are converted to the range 0 to 100.

Values of paramUI matrix set

Row	Range	El. min	El. max	Phys. min	Phys. max
0	5	0	5	0	5
1	5	0	5	0	100
2	10	0	10	0	100
3	20	4	20	0	100

Attention
*ARION network buffer values are zero after installing the application into the control system! If you use module **Filtr1R** to filter values measured, we recommend you process is only after establishing communication with module **DM-UI8DO8** successfully (see chapter Transfer status).*

Reading the other four universal inputs in digital mode

```
ARI_DigIn 1, 4, dataDI[0,0], 0x00000000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p10_en_xx.dso.

4.8 Example No. 11 – Communication with DM-UI8RDO8

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-UI8RDO8** (address 1) where the first four universal inputs are configured as analogue and the other four configured as digital. In terms of HW, the first four universal inputs are configured in the analogue mode according to the following table.

Universal input	Type
UI0	Ni1000
UI1	0 V to 5 V
UI2	0 V to 10 V
UI3	0 mA to 20 mA

In this case, the ARION network with an extension module **DM-UI8RDO8** is defined according to the following picture.

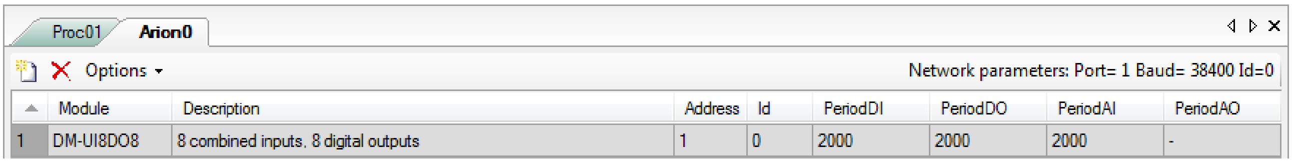


Fig. 25 – Definition of ARION network with **DM-UI8RDO8 extension module**

Writing values into all relay outputs:

```
ARI_DigOut 1, 0, 8, dataRDO, 0x00000000
```

Reading the first four universal inputs in analogue mode

```
ARI_AnIn 1, 0, 4, dataAI[0,0], paramUI[0,0], 10.000, 0.000, 10.000, 0.000, 10.000
//Converting measured voltage into temperature
Ni1000U2T dataAI[0,0], UI0_temperat, 6180, 15.000, 3920.000
```

A single **`ARI_AnIn`** module is used to read all four analogue inputs. Since individual signals use various constants for conversion into physical quantity, the matrix of conversion constant of 4 lines is specified in the parameter **`Conversion`** (each line for one universal input in analogue mode, see the following table). The module **`Ni1000U2T`** converts the value of AI0 input into temperature. The remaining values measured are converted to the range 0 to 100.

Values of paramUI matrix set

Row	Range	El. min	El. max	Phys. min	Phys. max
0	5	0	5	0	5
1	5	0	5	0	100
2	10	0	10	0	100
3	20	4	20	0	100

Attention
*ARION network buffer values are zero after installing the application into the control system! If you use module **`Filtr1R`** to filter values measured, we recommend you process is only after establishing communication with module **DM-UI8RDO8** successfully (see chapter Transfer status).*

Reading the other four universal inputs in digital mode

```
ARI_DigIn 1, 4, dataDI[0,0], 0x00000000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p11_en_xx.dso.

4.9 Example No. 12 – Communication with DM-UI8AO8U

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **DM-UI8AO8U** (address 1) where the first four universal inputs are configured as analogue and the other four configured as digital. In terms of HW, the first four universal inputs are configured in the analogue mode according to the following table.

Universal input	Type
UI0	Ni1000
UI1	0 V to 5 V
UI2	0 V to 10 V
UI3	0 mA to 20 mA

In this case, the ARION network with an extension module **DM-UI8AO8U** is defined according to the following picture.

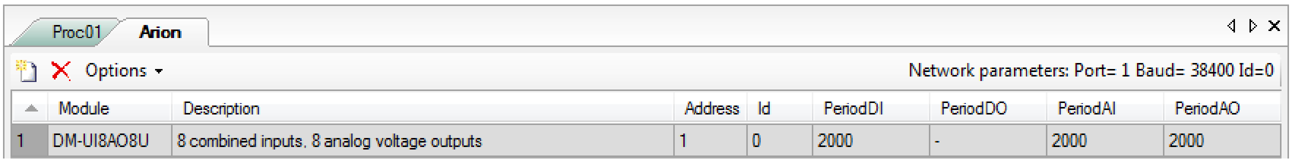


Fig. 26 – Definition of ARION network with **DM-UI8AO8U** extension module

Writing values into all analogue outputs:

```
ARI_AnOut 1, 0, 8, dataAO[0,0], NONE[0,0], 10.000, 0.000, 10.000, 0.000, 100.000
```

Individual signals use the same constants to convert into physical quantities, that is why it is not necessary to specify conversion constant matrices.

Reading the first four universal inputs in analogue mode

```
ARI_AnIn 1, 0, 4, dataAI[0,0], paramUI[0,0], 10.000, 0.000, 10.000, 0.000, 10.000
//Converting measured voltage into temperature
Ni1000U2T dataAI[0,0], UI0_temperat, 6180, 15.000, 3920.000
```

A single **ARI_AnIn** module is used to read all four analogue inputs. Since individual signals use various constants for conversion into physical quantity, the matrix of conversion constant of 4 lines is specified in the parameter **Conversion** (each line for one universal input in analogue mode, see the following table). The module **Ni1000U2T** converts the value of AI0 input into temperature. The remaining values measured are converted to the range 0 to 100.

Values of paramUI matrix set

Row	Range	El. min	El. max	Phys. min	Phys. max
0	5	0	5	0	5
1	5	0	5	0	100
2	10	0	10	0	100
3	20	4	20	0	100

Attention
*ARION network buffer values are zero after installing the application into the control system! If you use module **Filtr1R** to filter values measured, we recommend you process is only after establishing communication with module **DM-UI8AO8U** successfully (see chapter Transfer status).*

Reading the other four universal inputs in digital mode

```
ARI_DigIn 1, 4, dataDI[0,0], 0x00000000
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p12_en_xx.dso.

4.10 Programmable on-wall controllers AMR-OP7x/AMR-OP6x

These controllers come with firmware that allows measurement of ambient temperature, setting a request for temperature adjustment, learning/setting modes Auto / Comfort / Energy saving, or other values, depending on the firmware variation selected.

Unlike in modules of remote inputs and outputs, communication parameters (address, communication speed) are set directly on the display of **AMR-OP7x** or **AMR-OP6x** (see Operation manual.

4.10.1 Example 13 – Communication with AMR-OP7x/AMR-OP6x

Realizing communication with ARION Protocol (port 1, 38,400 bps) with **AMR-OP7x** controller (or **AMR-OP6x** controller) at address 1 which comes with firmware produced by AMiT.

The ARION network with **AMR-OP7x** controller or **AMR-OP6x** controller is defined according to the following picture.

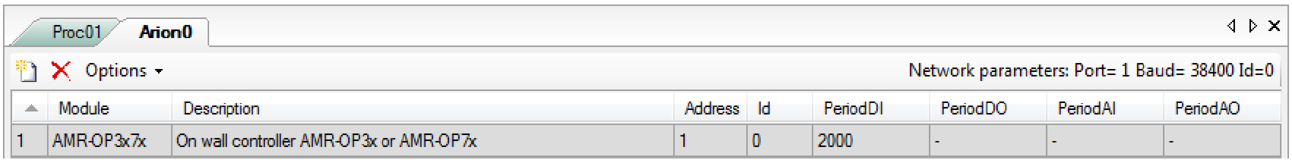


Fig. 27 – Definition of ARION network with **AMR-OP7x/AMR-OP6x**

Module **`AmrOp3x7xA`** enables work with all controller values. You may both read and write data using this module (see the module description in the Help tab for the PseDet section, DetStudio environment).

Reading/writing all values required

```
AmrOp3x7xA 1, OP_State, 0, Ts, Tscor, RoomMode, FanMode, NONE.0, Ti, NONE
```

This example is included in the attachment ap0025_en_xx.zip under the name arion_p13_en_xx.dso.

4.11 Programmable on-wall controllers AMR-OP3x(A)

These controllers come without firmware to facilitate communication with ARION network. However, there are sample projects available at amitomation.com in the section “Products/Sample projects/AMREG – sample projects” that allow reading/writing selected data from **AMR-OP3x(A)**. After implementing them into **AMR-OP3x(A)**, we can communicate with the controllers in the same way as described in the previous chapter on **AMR-OP7x**.

5 Most frequent problems

5.1 Communication keeps failing

Are all modules connected to power supply?
If so, LED PWR is on on all modules.

Are all modules fully functional?
If so, LED RUN blinks on all modules in intervals ca 2 s (1:1).

Is the communication speed and address of each module set correctly?
If so, LED RxD i TxD are on or blink (with various periods and intervals) on appropriate modules. If only LED RxD blinks, it means that although the network is connected correctly, the communication speed or address on the given extension module on in software are not set correctly.

5.2 Communication has been established, but it is not reliable

It is possible the communication network is overloaded.
If so, LED TxD on the control system is on, or turns off for very brief periods of time.

- It is possible that the communication network has been overloaded by requests for connection loss detection (parameters `GuardTime` in the definition table) – we must extend their periods.
- This situation may occur with low communication period with modules connected, low communication speed and higher number of modules – we must increase the communication period or increase communication speed.

Strong interference may occur near the communication lines.

We must lay the cabling of line RS485 in accordance with recommendations in the Application note AP0016 – Principles of RS485 interface usage and use suitable overvoltage protection.

6 Technical support

All information on communication in an Ethernet network will be provided by the technical support department of the company AMiT. Do not hesitate to contact the technical support via e-mail using the following address: **support@amit.cz**.

7 Warning