

20.4 Configuring data endpoints

To use WCC Lite as a programmable logic controller, it needs to be configured in a particular way. The PLC functionality of the WCC Lite only allows for the use of data that has been configured in the Excel configuration spreadsheet. This has been done for security purposes and to preserve transmission medium only for data that is available. Only topics defined in the configuration can post or get data. If a certain data entry exists but it has not been linked to a PLC program, all calls from PLC runtime application to redis database will be ignored. Therefore it is highly advised to prepare and upload the Excel configuration before using this signal in the PLC application.

Some parameters are mandatory for PLC usage. These parameters are shown in two tables below (one for Devices, one for Signals tab). Please note that other parameters can be used as well, but are not covered because they aren't specific to PLC functionality.

Table. Mandatory parameters for Devices tab

Parameter	Type	Description
name	string	User-friendly device name
device_alias	string	Device alias to used in configuration
enable	boolean	Enabling/disabling of a device
protocol	string	Selection of protocol (IEC 61499)

Table. Mandatory parameters for Signals tab

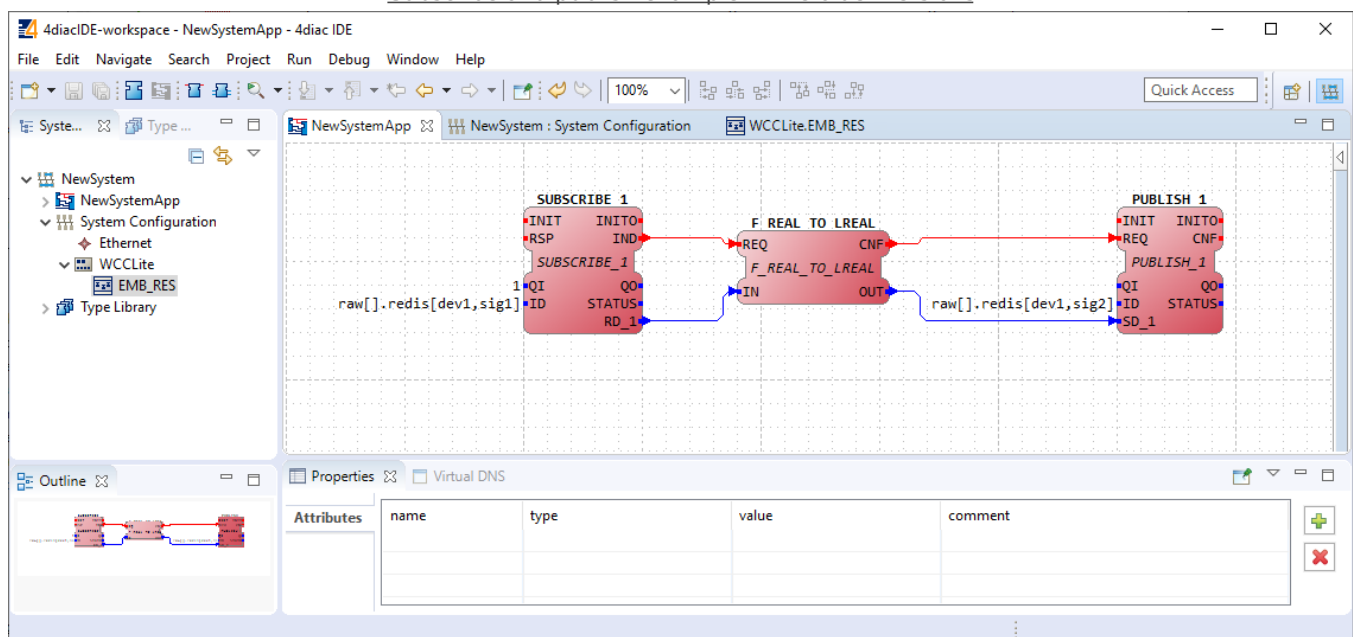
Parameter	Type	Description
signal_name	string	User-friendly signal name
device_alias	string	Device alias from a Devices tab
signal_alias	string	Unique signal name to be used
source_device_alias	string	device_alias of a source device
source_signal_alias	string	signal_alias of a source signal

If an upload consisting of configuration for IEC 61499 has been successful, one should be able to access a configuration stored in /etc/iec61499.json file where protocol-specific parameters are shown in a JSON format. If the file is missing, make sure you have a correct firmware version installed and haven't made any typing errors.

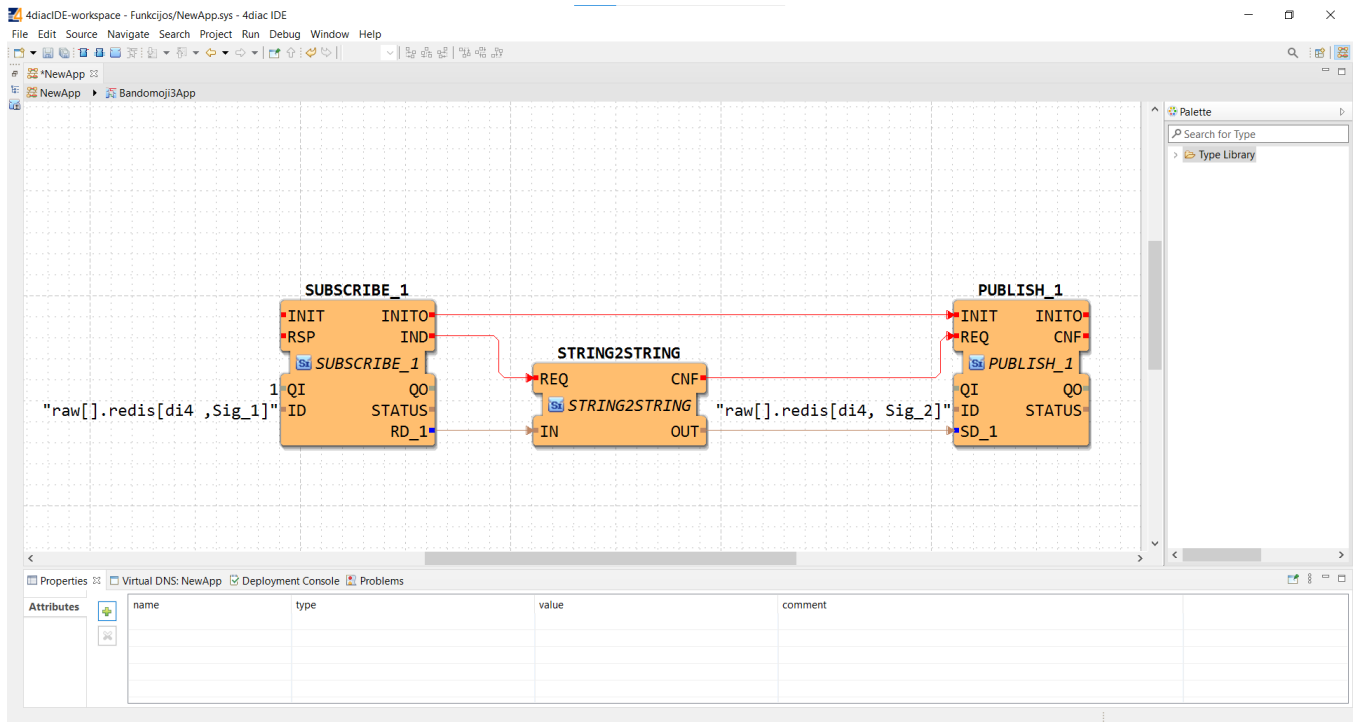
Parameters mentioned earlier, namely device_alias and signal_alias, are the only parameters one needs to fill to bind Excel configuration to 4Diac framework. Two types of blocks are used for data transmission - PUBLISH blocks to write data to REDIS database and SUBSCRIBE blocks to acquire data from database as soon as it changes its value. Both of them have an ID connection. To connect a block to a data point, one should set this pin as raw[].redis[device_alias,signal_alias], e.g. raw[].redis[example_plc_device,example_plc_signal_alias]. As of version 2.0.0 the pin values must be surrounded by double commas.

Examples with SUBSCRIBE and PUBLISH function blocks are shown in the images below.

Subscribe and publish example in the older versions



Subscribe and publish example in the newer versions (from 2.0.0)



⚠ Outputs of variable type ANY cannot be directly wired to inputs of the same type and therefore need to be explicitly typed using transitional function blocks.

⚠ No more than 20 tags should be published over a period of 5 seconds, as this may overfill the queue. A "publish only on change" policy is advised.

⚠ Currently only **PUBLISH_1** and **SUBSCRIBE_1** function blocks are supported.

⚠ The functionality of the subscribe function is dependent on the presence of the "F_STRING_TO_" function block, for example (F_STRING_TO_REAL). Likewise, the publish function's proper operation relies on the availability of the "F_TO_TO_STRING" FB, for example (F_REAL_TO_STRING).

If every step until now has been successful, the user can now proceed to debug the PLC application.

🔄Revision #1

★Created 11 January 2024 09:21:45 by Gabriele

🔧Updated 11 January 2024 09:38:18 by Gabriele