Y-ASI4U-V5-DB Software

This document explains how to configure and establish minimalist master-slave communication between the Y-ASI4U-V5-DB and an ASi-5 master using the EK-RA2A1 as a platform for complex slave evaluation. The Y-ASI4U-V5-DB is included in the ASI4U-V5-EVB-V1P0 ASI4U-V5 Starter Kit.

The hardware and software ecosystem provides a quick start and easy understanding of ASi-5 technology.

AS-international provides an online course covering the specifics of the ASi-5 protocol:

https://www.as-interface-academy.net/course/view.php.

Contents

Figures

Tables

1. Terms and Definitions

AMC	Acyclic management channel
DSMBE	Device specific multi-byte entries
DSSBE	Device specific single-byte entries
FSP	Flexible software package
GPIO	General purpose input output
GUI	Graphical user interface
IDE	Integrated development environment
IP	Internet protocol
LED	Light emitting diode
MCU	Microcontroller unit
ORDS	On request data service
PC	Personal computer
PLC	Programmable logic controller
PMOD	Peripheral module interface
PSMBE	Profile specific multi-byte entries
PSSBE	Profile specific single-byte entries
PWR	Power
SPI	Serial peripheral interface
SREC	S-record
USB	Universal serial bus

2. Hardware Setup

The following hardware tools are required to evaluate the ASI4U-V5_ASSP:

1. Y-ASI4U-V5-DB
2. EK-RA2A1
3. E2 lite debug probe
4. ASi-5 Master Gateway

Contact Renesas representatives or visit the Renesas website for information on obtaining these devices and development boards.

2.1 Y-ASI4U-V5-DB

The Y-ASI4U-V5-DB is a Development Board for the ASI4U-V5 ASSP, a Renesas silicon solution implementing an Actuator-Sensor-Interface version 5 (ASi-5) fieldbus transceiver. It minimizes the integration required for an ASi-5 interface.

For detailed hardware information, refer to the Y-ASI4U-V5-DB Hardware User Manual document.

2.2 EK-RA2A1

The EK-RA2A1 is an evaluation kit for the Renesas RA2A1 microcontroller, enabling quick evaluation of several sensors. It includes a dedicated interface for easy connection to the Y-ASI4U-V5-DB.

For detailed kit information, visit the Renesas website.

2.3 E2 Lite Debug Probe

The ASI4U-V5_ASSP integrates a Renesas MCU: RL78/G14 (used as an intelligent on-chip flash memory device), which requires a Renesas debug probe for programming.

Renesas E2 lite is the recommended probe for this project. For information on supported MCUs, visit the Renesas website.

2.4 ASi-5 Master Gateway + ASi Power Supply + Splitter.

These components are available from third party vendors, depending on the Ethernet-based network protocol used. The test setup described in this document uses the BWU3847 ASi-5 Gateway from Bihl and Wiedemann. Additionally needed are:

- A dedicated ASi power supply is required to power the gateway.
- A splitter is used to connect the Y-ASI4U-V5-DB via CN23 to the ASI network.

3. File Package Structure

A file package is distributed along with the ASI4U-V5-EVB-V1P0 and can be downloaded from:

https://www.renesas.com/en/design-resources/boards-kits/asi4u-v5-evb-v1p0

The package is structured as the following (individual files are referenced by name in this document):

- ASI4U-V5-EVB-V1P0
  - CODESYS
  - PROFINET_Example.project
    - EK_RA2A1
  - ASI5_Complex_Slave_EK_RA2A1.zip
  - ASi-5_Complex_Slave_API_Documentation.pdf
    - Y-ASI4U-V5-DB
  - Dataflash_Generator

Appendix A ASi4U-V5_Bin_Release_3.00.zip

Appendix B ASI4U-V5_Dataflash_Generator_User_Guideline_3.00.pdf

- User_Factory_Page

Appendix C Complex_Slave

- 4Bytes_CS_4242_0200000000.srec
- 32Bytes_CS_4242_0100000000.srec

Appendix D Simple_Slave

- 8I8O_4242_0100000000.srec
- 16I_4242_0200000000.srec
- 16O_4242_0300000000.srec

4. ASi-5 Slave Configuration

4.1 Operation Mode Configuration

The Y-ASI4U-V5-DB supports two ASi slave modes:

- Simple Slave
- Complex Slave

In Simple Slave mode (see section 4.1.1), the Y-ASI4U-V5-DB operates as standalone ASi device. In Complex Slave mode (see section 4.1.2), the EK-RA2A1 transmits data to the Y-ASI4U-V5-DB via SPI.

Figure 1 illustrates both modes in a block diagram.

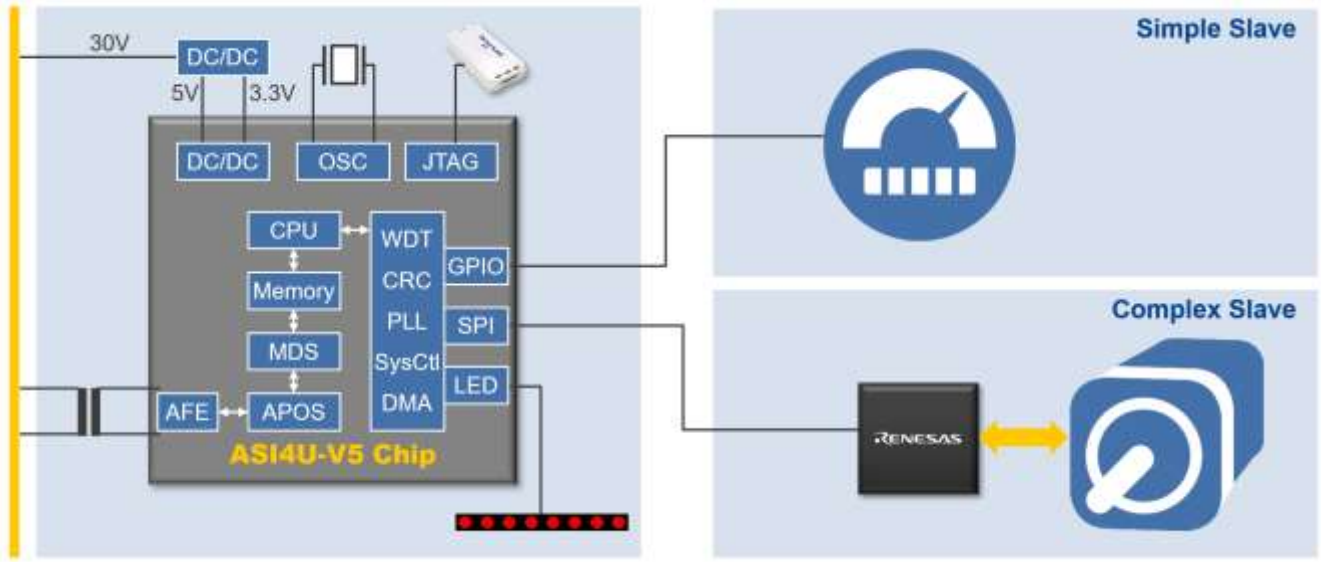

Figure 1. Simple/Complex Slave Block Diagram

4.1.1 Simple Slave Mode

In Simple Slave mode, sensor digital data is input through the ASI4U-V5_ASSP GPIOs. Up to 22 digital I/Os can be used for cyclic communication. Those pins can be configured as Input, Output, or Event pins. Note that restrictions apply regarding pin assignment to specific functionalities.

The software file package includes the following user/factory pages examples for Simple Slave mode:

- 8I8O_4242_0100000000.srec
  8 Digital Inputs and 8 Digital Outputs, 1Transport Channel, 1Subcyle, I/O-Code 0x000
- 16I_4242_0200000000.srec
  16 Digital Inputs, 1Transport Channel, 1Subcyle, I/O-Code 0x004
- 16O_4242_0300000000.srec
  16 Digital Outputs, 1Transport Channel, 1Subcyle, I/O-Code 0x004

For configuration details, refer to the ASi-5 Device Profiles document.

GPIOs P0_0 to P0_15 toggle depending on input or output state. These GPIOs are accessible via CN14 on the Y-ASI4U-V5-DB. GPIO states P0_0 to P0_7 are indicated by LED3 to LED10, see Table 1Table 3 for corresponding GPIO mapping.

Table 1. Simple Slave GPIO Configuration

GPIO	Pin CN14	LED	8I8O	16I	16O
P0_0	5	LED3	Input	Input	Output
P0_1	6	LED4	Input	Input	Output
P0_2	7	LED5	Input	Input	Output
P0_3	8	LED6	Input	Input	Output
P0_4	9	LED7	Input	Input	Output
P0_5	10	LED8	Input	Input	Output
P0_6	11	LED9	Input	Input	Output
P0_7	12	LED10	Input	Input	Output
P0_8	13	n/a	Output	Input	Output
P0_9	14	n/a	Output	Input	Output
P0_10	15	n/a	Output	Input	Output
P0_11	16	n/a	Output	Input	Output
P0_12	17	n/a	Output	Input	Output
P0_13	18	n/a	Output	Input	Output
P0_14	19	n/a	Output	Input	Output
P0_15	20	n/a	Output	Input	Output

GPIOs configured as inputs can be toggled manually by applying 3.3 V using a jumper cable, as shown in Figure 2 by red.

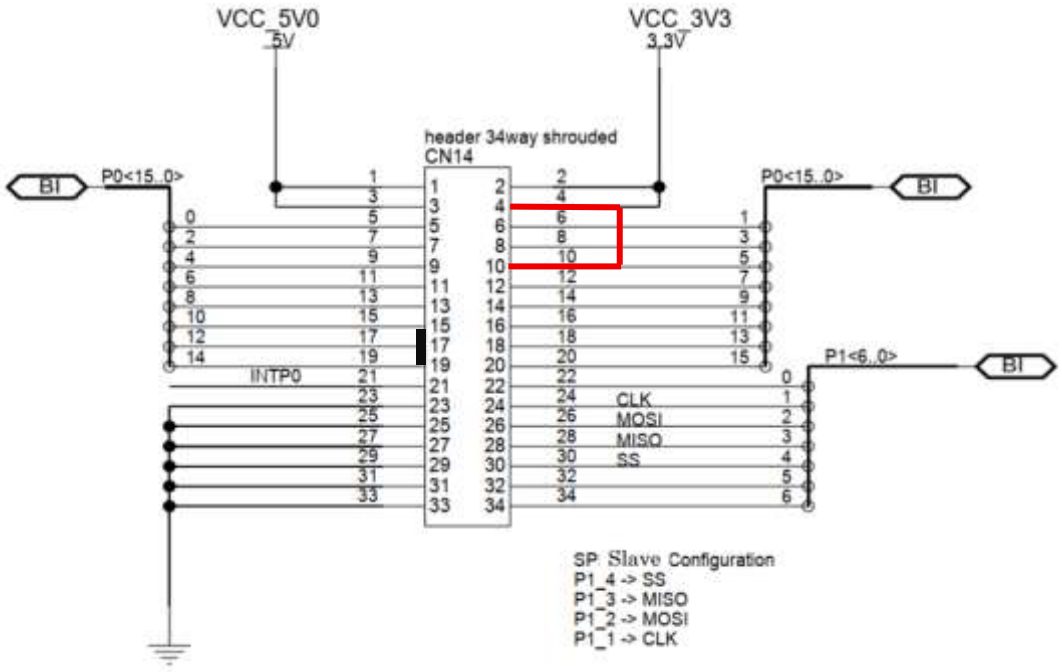

Figure 2. Example: Toggling GPIO P0_5 High

4.1.2 Complex Slave Mode

In Complex Slave mode, the ASI4U-V5_ASSP communicates with an external MCU through SPI. This mode supports up to 32-bytes of cyclic process data on the ASi network and enables the AMC/ORDS channel for additional ASi-5 functionalities.

The software file package includes the following user/factory pages examples for Complex Slave mode:

- 32Bytes_CS_4242_0100000000.srec
  Demonstrates 32-Bytes of processed data per cycle using channel bundling and multiplexing
- 4Bytes_CS_4242_0200000000.srec
  Alternative 4 -Bytes process data profile.

4.2 User/Factory Page Parameter Explanation

This section provides an overview of the dataset configuration, Table 2 shows the user/factory page memory map.

Table 2. RL78 NVRAM User/Factory Page Blocks

	Address in NVRAM	Description
User Page	0x000 … 0x01F	NVRAM Header block 0
	0x020 … 0x3FF	User Page block 0
	0x400 …0x41F	NVRAM Header block 1
	0x420 … 0x7FF	User Page block 1
Factory page	0x800 … 0x81F	NVRAM Header block 2/3
	0x820 … 0xBFF	Factory Page block 2
	0xC00 … 0xFFF	Factory Page block 3

The user/factory pages are stored in the NVRAM memory of the ASI4U-V5_ASSP and contains all device parameters. The user page block is backed up to maintain consistency in case of power fail during a read operation by the ASi-5 Master. The parameters are defined as the following:

- Factory page parameters (see Table 3): defined during the ASi-5 slave manufacturing process
- User page parameters (see Table 4): set during ASi-5 slave installation in the ASi network.

Factory page parameters are organized into four categories:

- DSMBE: Device Specific Multi-Byte Entries
- PSMBE: Profile Specific Multi-Byte Entries
- DSSBE: Device Specific Single-Byte Entries
- PSSBE: Profile Specific Single-Byte Entries.

The "Length" field determines the payload size in multi-byte-entry format, while single-byte entries have no length fields (see Figure 3).

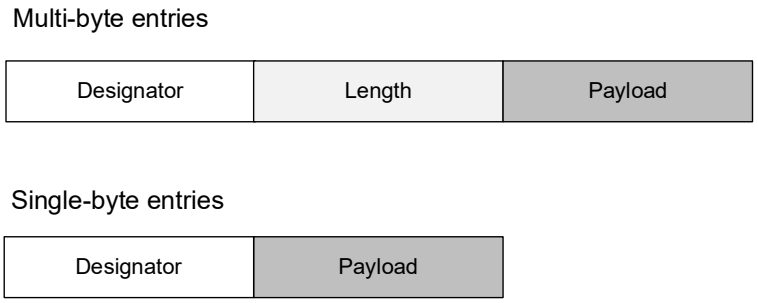

Figure 3. Factory Page Parameter Fields

Table 3. Factory Page Parameters

Index	Name	Description
DSMBE		
0x00	ASIID	Unique identification number containing vendor ID and a product ID. Displayed on ASi Master HMI during addressing. Vendor ID is assigned by the ASi international association to every ASi-5 Slave manufacturer, product ID is a vendor specific unique parameter used to determine an ASi-5 Slave product. For testing, use Renesas vendor ID: 4242.
0x01	Manufacturer Device Name	Identifies the product manufacturer. Vendor specific parameter It consists of a 32-bit hexadecimal value read as an ASCII string by the master. Example: 3849384F(hex) = 8I8O (ASCII).
0x02	Manufacturer Device Family	Identifies product family (ASCII string), vendor specific.
0x03	Hardware Revision	Identifies product hardware revision, vendor specific.
0x04	Manufacturer Device Version	Identifies device version, vendor specific.
DSSBE		
0x80	ASIC Configuration	Enables the Simple or Complex Slave mode.
0x81	Number of Profiles	Declares the number of profiles contained in the factory page. Refer to the *Device Implementation Guide* document for multiple profile edition.
0x83	SPI Process Data Mode	Selects the format of the SPI frame in the case of a Complex Slave implementation. Process Data Image is used for Complex Slave implementation. Raw Transport Image is used for Safety Slave implementation.
0x84	SPI Process Data Size	Sets processed data size to transfer between the external MCU and the ASI4U-V5_ASSP for Complex Slave mode.
0x85	SPI ODC Size	Sets the size of the ODC (On Demand Communication) in case of a Complex Slave implementation. ODC is used to establish an acyclic communication between the ASI4U-V5_ASSP and the external MCU to request diagnostic information to the ASI4U-V5_ASSP. The total SPI frame must be a multiple of four.
0x87	Write Protection	Activates the write protection of the factory page block stored in the flash and NVRAM (user page is not concerned).
0x88	SPI AsiTime	Used as a time stamp regarding the diagnostic messages sent to the ASi-5 Master. It is made of the 2 lower bytes of the ASi Master's system clock.
PSMBE		
0x40	Profile ID	Declares the used profile ID. The profile IDs can either be generic or custom. Refer to the *ASi-5 Device Profiles* document.
0x41	Substitute Values	Sets the substitutes values. If the process data watchdog is triggered due to communication loss, the ASi-5 Slave replaces the output values by substitute values. Refer to the *ASi-5 Specification* document.
0x42	PDT (Process Data Type)	Sets the length of the data to be sent to the master. The setting is explained in the *ASi-5 Specification* and *ASi-5 Device Profiles* documents.

Index	Name	Description
0x43	TLSC (Transport Layer Slave Configuration)	Sets the number of transport channels and the number of sub-cycles used by the ASi-5 Slave to transfer its data to the master. Refer to *ASi-5 Device Profiles* document for describing the TLSC values for generic profiles.
0x44	Input Invert Mask	Inverts the bit value of the inputs.
0x45	Output Invert Mask	Inverts the bit value of the outputs.
0x46	Diagnostic Events	Used to get diagnostic information from a potential controller attached to a specified index pin. Refer to the *Device Implementation Guide* document.
0x47	Index Events	Sets the events allocated to the Index pins.
0x48	Default Parameter Image	Parameter Image used if a profile change or a factory reset has been performed.
0x49	Arbitration Mask	Used for safety salves. Refer to *Safety Monitor Specification 1.2* document.
PSSBE		
0xC0	ODC Timeout	Sets the timeout delay within which an ODC response from the external MCU should occur.
0xC1	Port Count	Sets the number of ports.
0xC2	Input Pin Filter	Enables an "input stability counter" acting as signal cleaner to return a filtered output to the ASI4U-V5_ASSP. This functionality is typically enabled for actuators like push buttons. Also used to set the minimal hold time to detect a diagnostic event. Refer to the *Device Implementation Guide* document.
0xC3	Diagnostic Pin recovery Time	Sets the minimal waiting time between 2 diagnostic events.
0xC4	Diagnostic Location Configuration	Defines the range for port and channel number for all pins. Refer to the *ASi-5 Specification* document.
0xC6	Pin Multiplexing	Enables the possibility to multiplex the pin status. It periodically activates the pins as inputs and outputs according to the data strobe signal status (DSR).
0xC7	Process Data Input Pins	Sets the number of input pins to activate. The input pins are coming first according to the BIT list. *Device Implementation Guide* document_*implementation_guide.pdf*.
0xC8	Process Data output Pins	Sets the number of output pins to activate.
0xC9	Diagnostic Pins	Sets the number of pins used to trigger a diagnostic. Maximum of three pins can be used.
0xCA	Index Input Pins	Maps GPIOs as Index inputs.

Index	Name	Description
0xCB	Index Output Pins	Maps GPIOs as Index outputs.

Table 4. User Page Parameters

Size (Bytes)	Name	Description
1	Configuration Flags	Configures the ASi-5 Slave as a standard or a priority slave. The priority slave requires a transport channel for all sub-cycles whereas the standard slave only requires a transport channel for a single sub-cycle.
1	Process Data Watchdog	Configures a timeout for the PDT. If the watchdog is triggered, it replaces the output values by substitutes values if communication is lost.
2	Logical Address	Address to identify the ASi-5 Slave in the ASi network. The address must be unique and is typically allocated by the ASi-5 Master during the addressing procedure (can also be allocated by the ASi-5 Addresser).
4	Actual Profile	Stores the current used profile ID (multiple profiles can be declared in the factory page).
32	Device Designator	Contains the device designator (ASCII string).
1	Energy Saving Group	Allows to manage the power consumption of automation equipment through the ASi network.
128	Parameter Image	Used to recover the ASi-5 Slave parameters if a slave must be replaced (Also stored in the Asi-5 Master).
4	sPici (Parameter Image Change Indicator)	Counter is incremented each time a change in the parameter image is performed.
7	Last Master ASIID	Stores the Vendor and Product ID of the last ASi-5 Master which communicated with the ASi-5 Slave.
1	Factory Reset Count	Counter is incremented each time a factory reset is performed.

4.3 User/Factory Page Example

Figure 4 shows an extract of an NVRAM header block 2/3 and factory page block 2 example, demonstrating the factory page structure in the SREC file (refer Table 3).

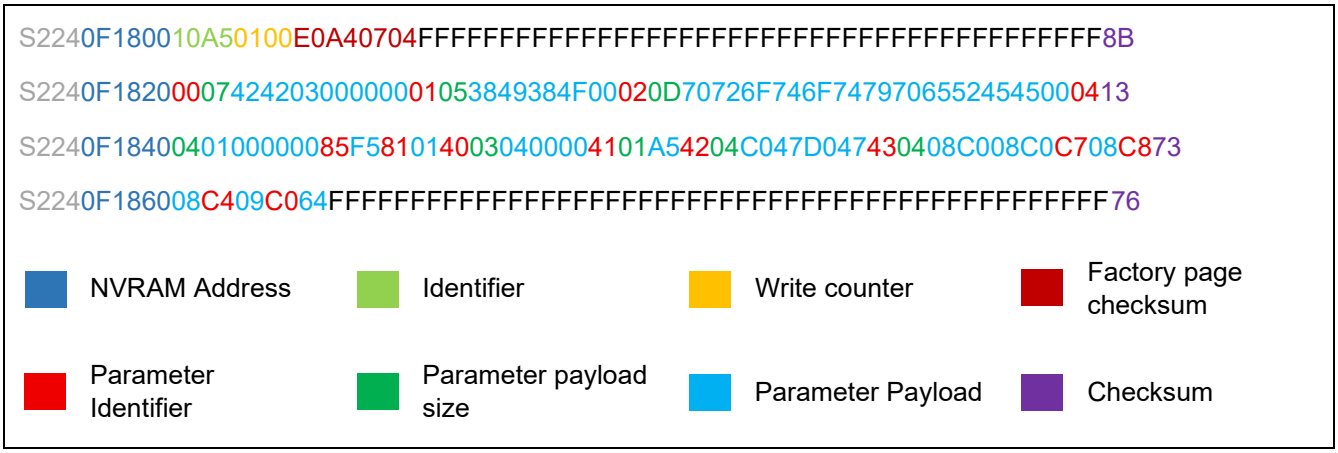

Figure 4. Factory Page Example

Figure 5 shows an extract from an NVRAM header block 0 + User Page block 0 example.

```
S2240F100010A5010090AE030004FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD8
S2240F1020001E0000040000000000000000000000000000000000000000007A
S2240F1040000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF92
S2240F1060FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7C
S2240F1080FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5C
S2240F10A0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3C
S2240F10C0FFFFFFFFFFFFFFFFFFFF00000020FFFFFFFFFFFFFFFFFFFFFFFFFFF8
```

NVRAM Address | Identifier | Write counter | User page checksum

Page version | Checksum

Figure 5. User Page Example

NOTE
The user page organization differs from the factory page. In the factory page, all parameters are stored sequentially.

4.4 ASI4U-V5 Dataflash Generator

Renesas developed the ASI4U-V5 Dataflash Generator to simplify custom user/factory page creation. The application translates custom parameter sets into the binary representation of the respective user/factory page.

The application is located in the software file package. To start the application, open *ASI4U-V5 Dataflash Generator.exe*. The GUI appears as shown in Figure 6.

Figure 6. ASI4U-V5 Dataflash Generator

For complete documentation and tutorial, refer to the *ASI4U-V5 Dataflash Generator User Guideline* document.

5. Firmware Setup

5.1 Tool Installation

Follow these steps to install the firmware:

1. Download and install Renesas Flash Programmer V3 from:
   https://www.renesas.com/en/software-tool/renesas-flash-programmer-programming-gui
2. Connect the E2 Lite probe to the user PC through USB.
3. Connect the Y-ASI4U-V5-DB to the E2 Lite probe through connector CN30.
4. Connect the Y-ASI4U-V5-DB to the ASi network to power the ASi-5 Slave (refer to Figure 7).
   Note: Ensure the connector orientation matches Figure 8 (red wire on the left).

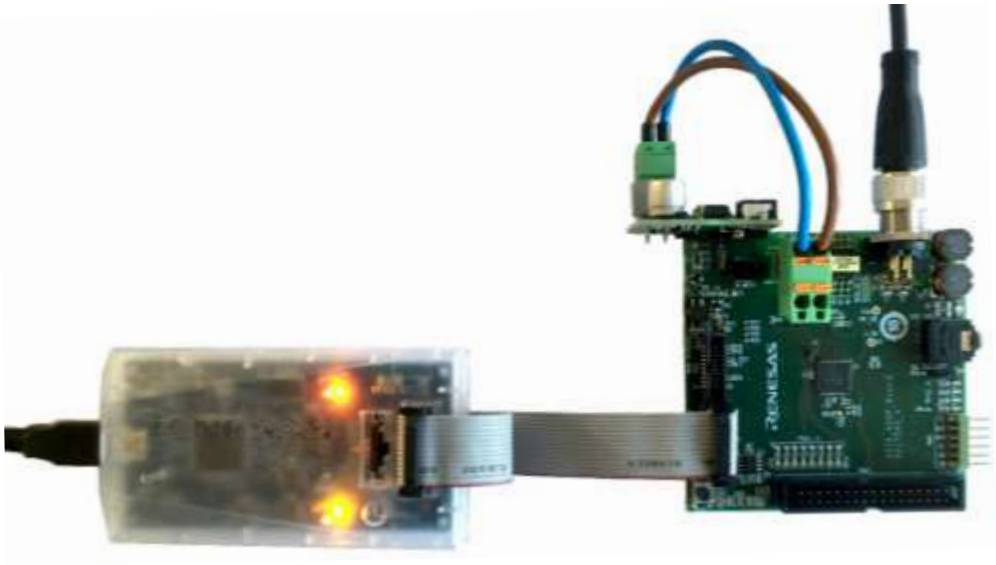

Figure 7. E2 Lite Probe Connection To Y-ASI4U-V5-DB

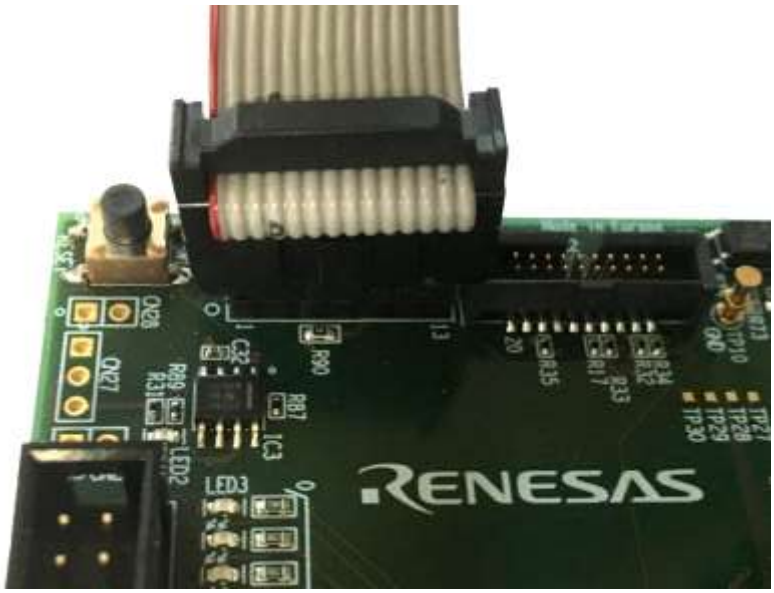

Figure 8. E2 Lite Probe Connection Close-Up

5. Start Renesas Flash Programmer V3.
6. Create a new project by the following steps:

a. Select **File** > **New Project**….
b. Select **RL78** as **Microcontroller**.
c. Choose a project name and a project folder.
d. Select **E2 emulator Lite** as **Tool**.
e. Click on **Connect**.

The console returns **Operation completed** as in Figure 9.

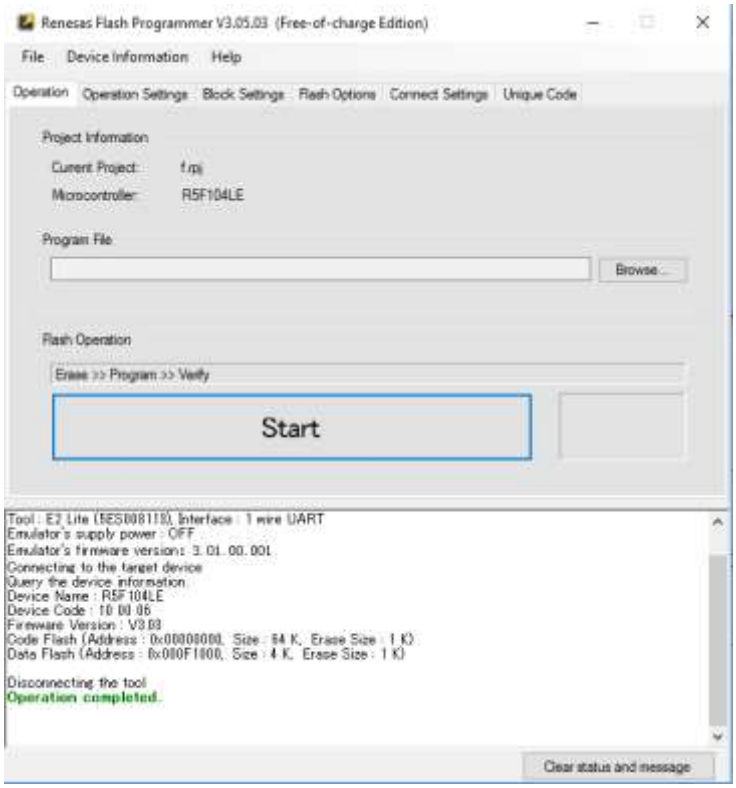

Figure 9. Renesas Flash Programmer GUI

5.2 Firmware Update

Two flash files must be written to the ASI4U-V5 ASSP NVRAM as the following:

- ASi-5 Slave application (merged with RL78 Bootloader)
  Download the latest ASI4U-V5 firmware (ASi5_FW_5.0_BL.srec file from:
  https://www.renesas.com/eu/en/document/swr/asi5-firmware-v50
- The user/factory page dataset containing the device parameters
  This file is created using the Dataflash Generator application to meet custom requirements. Sample configuration datasets are provided in the software file package.
  The datasets are formatted according to Motorola SREC.

The ASi-5 Slave application and user/factory pages must be flashed separately to the RL78 NVRAM as described in section 5.2.1.

5.2.1 ASi-5 Firmware Update

Update the firmware by following these steps:

1. In Renesas Flash Programmer, navigate to **Block Settings** tab.
2. Enable flashing only in the **Code Flash** section as shown in Figure 10.

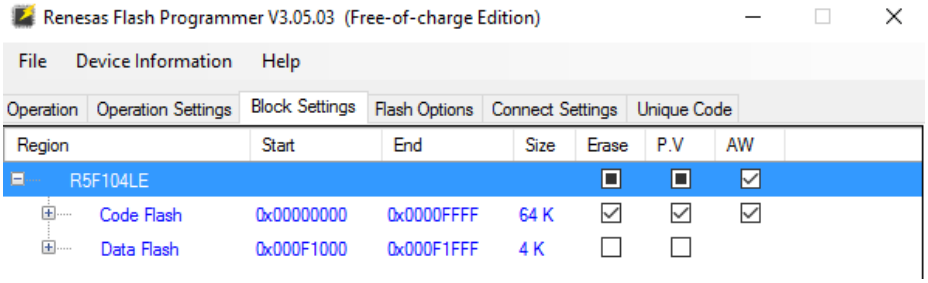

Figure 10. Memory Block Setting For ASi5 Firmware Update

3. Return to the **Operation** tab and click on **Browse**.
4. Select the **ASi-5 Firmware: ASi5_FW_5.0_BL.srec**.
5. Click on **Start**.
   During the FW update, the LED RP4<0> switches on.

After successful programming, the console displays **Operation completed** as shown in Figure 11.

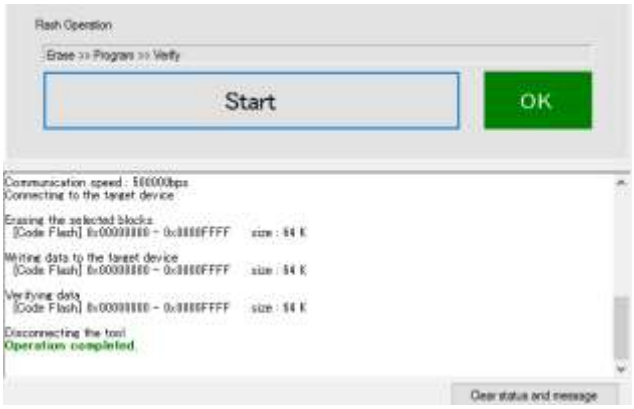

Figure 11. Renesas Flash Programmer Console

5.2.2 User/Factory Page Update

Update the user and factory pages by following these steps:

1. Go to **Block Settings** tab.
2. Enable flashing only in the data flash section as shown in Figure 12.

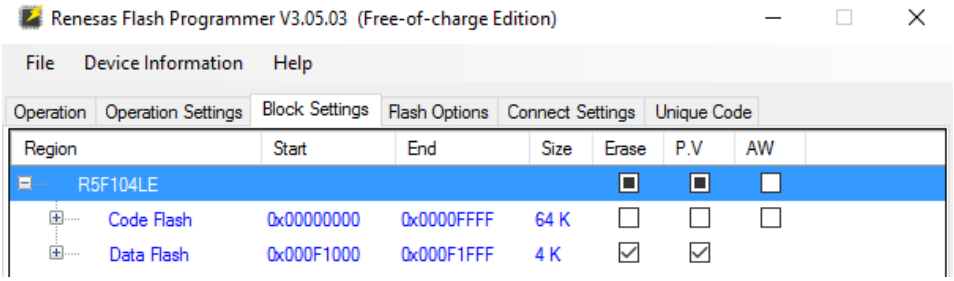

Figure 12. Memory Block Setting User/Factory Page Update

3. Return to the **Operation** tab and click on **Browse**.
4. Select the flash file corresponding to your application to be flashed on the target (user/factory page).
5. Click on **Start**.
   During the FW update, the LED RP4<0> briefly switches on.

After successful programming, the console displays **operation completed** as shown in Figure 13.

Figure 13. Renesas Flash Programmer Console

6. Disconnect the debug probe and press the reset button SW1 on the board.
   LEDs RP7_0 (green) blinks, and RP7_1 (red) is permanently on.

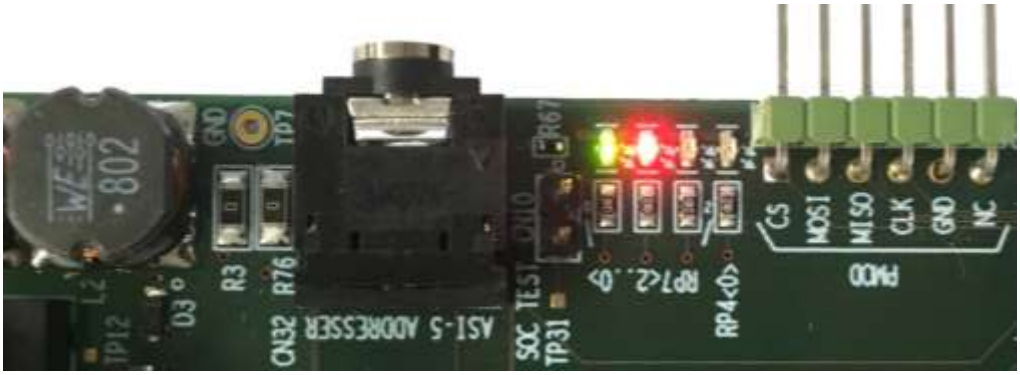

Figure 14. Status LEDs Y-ASI4U-V5-DB

6. EK-RA2A1

6.1 Board/Platform Presentation

The ASI4U Kit includes a firmware package for the EK-RA2A1 to facilitate quick evaluation of the Complex Slave configuration. This board features the Renesas RA2A1 MCU and utilizes the Renesas Flexible Software Package (FSP).

The boards connect through PMOD connectors, enabling the EK-RA2A1 to input data through the Y-ASI4U-V5-DB GPIOs or use the SPI interface. An additional cable connects AN000-P500 to P1_6 (refer to Figure 15).

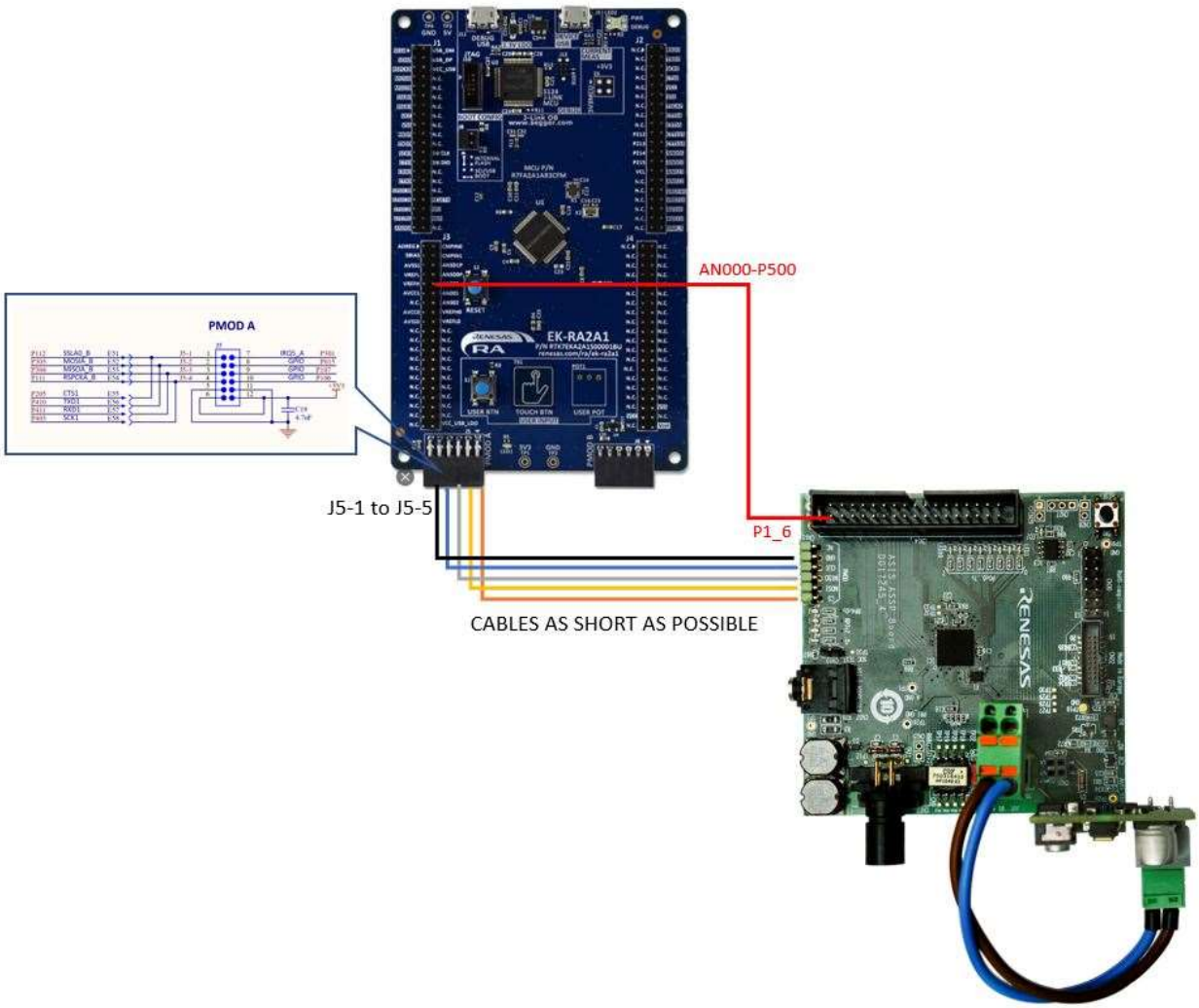

Figure 15. Connection Between Y-ASI4U-V5-DB and EK-RA2A1

For API documentation, refer to the *ASi-5 Complex Slave API Documentation* in the Software File Package.

NOTE
Use cables as short as possible for the inter-board connection. Alternatively, connect the boards directly by flipping one board and plugging the PMODs together.

6.2 e2studio IDE Setup

Setup the e2studio IDE by following these steps:

1. Install the latest Renesas eclipse based e2studio IDE version 2025-10 (25.10.0) and the Flexible Software Package (FSP) for RA version 6.2.0 or higher from the following link:
   https://github.com/renesas/fsp/releases

2. Launch e2studio and select a workspace path (see Figure 16).

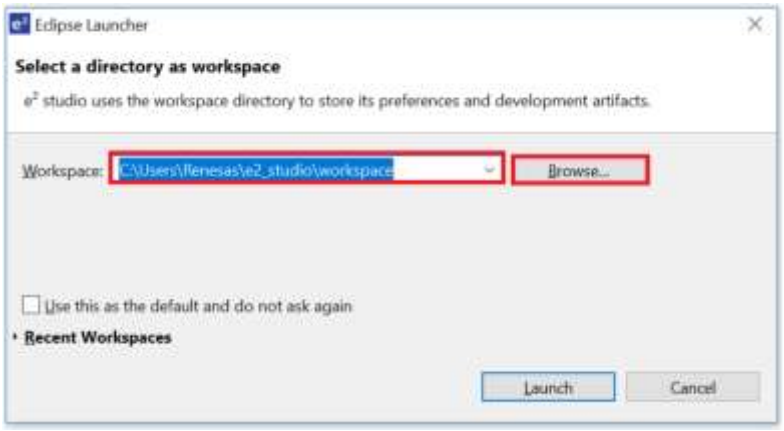

Figure 16. e2studio Workspace Selection

3. Click **Launch**.
4. Click **Import** from the **File** drop-down menu.
5. In the Import dialog box, select **General**, and then select **Existing Projects into Workspace**.
6. Click **Next**.
7. Click **Select archive file:** and click **Browse**.
8. Select folder **ASI5_Complex_Slave_EK_RA2A1.zip**.
9. In **Import** dialog window, select **ASI5_Complex_Slave_EK_RA2A1** project and click **Finish**.

6.3 Project Organization

The e2studio project demonstrates both Simple and Complex Slave configurations. The toolchain is GCC, which is installed by default with FSP.

Figure 17 illustrates the project structure.

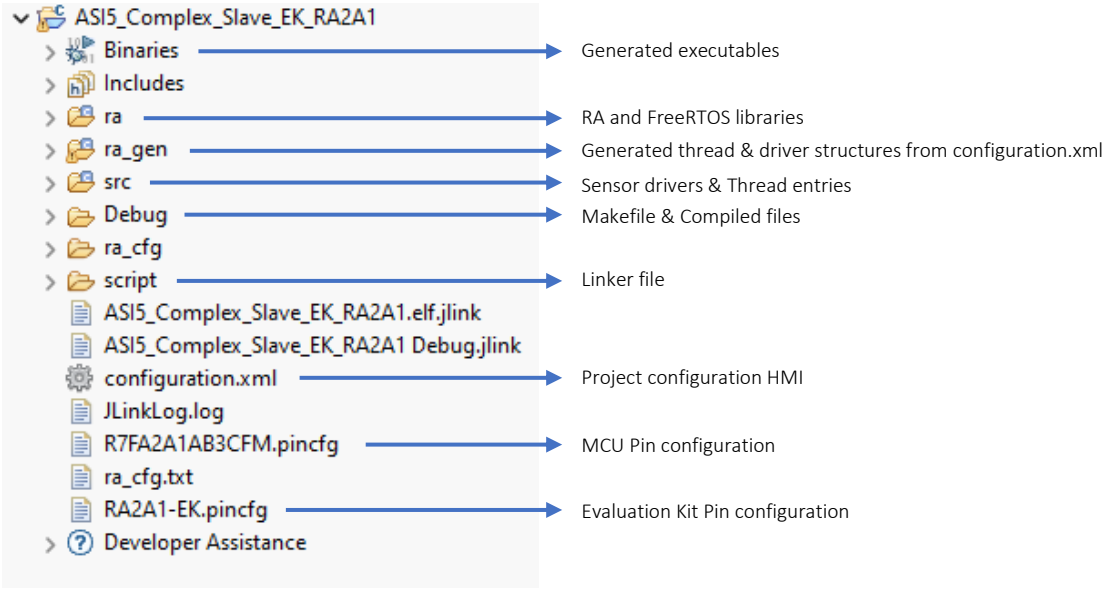

Figure 17. e2studio Project Structure

6.4 Project Contents Compilation and Debugging

After importing the project, double-click the **configuration.xml** file in the project structure to open the project configurator. This interface provides easy configuration of MCU peripherals, pins, clocks, and other settings by following these steps:

1. Click **Generate project contents** to update the threads and drivers project configuration structures as shown in Figure 18.

Figure 18. Project Configurator

In the **Stacks** tab, view the list of drivers and threads used in the project. Driver properties can be viewed by opening the **Property** (see Figure 19).

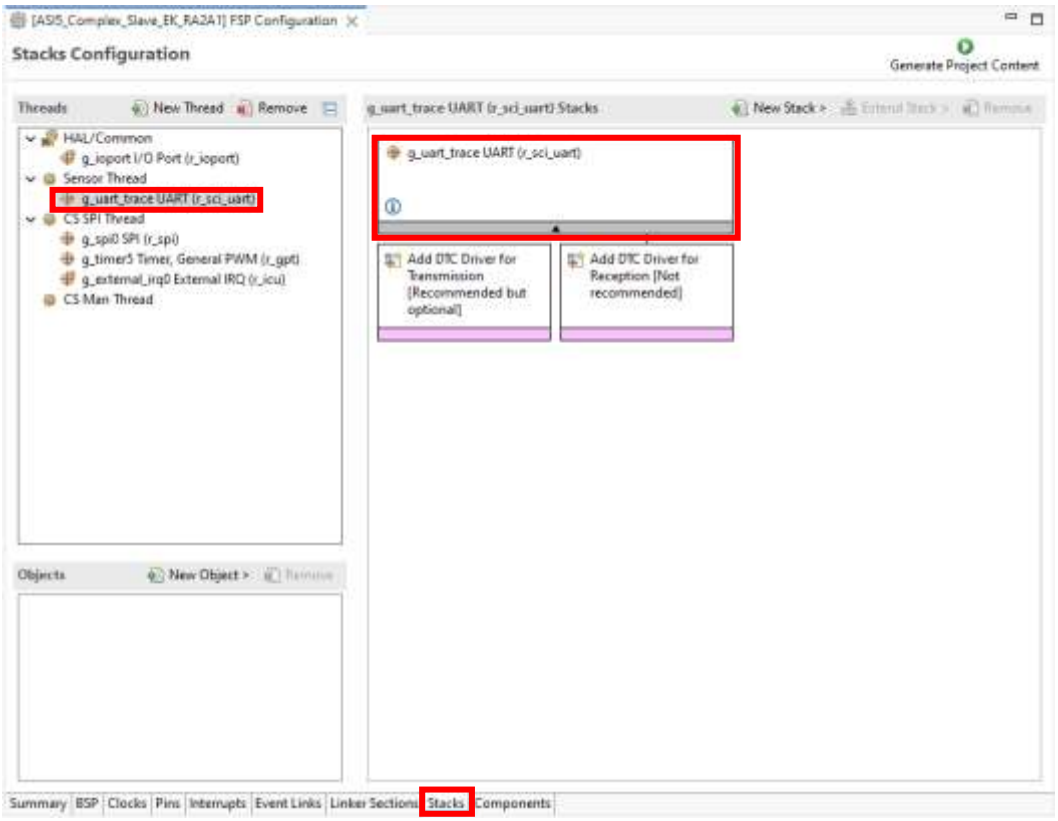

Figure 19. Configuration.xml Stacks Thread/Driver Visualization

2. Build the project by clicking the **Build** button (see Figure 20).

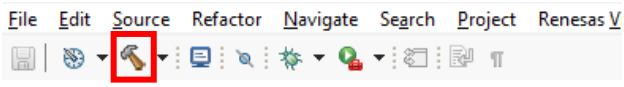

Figure 20. Build Button

Once complete, the console returns the information shown in Figure 21.

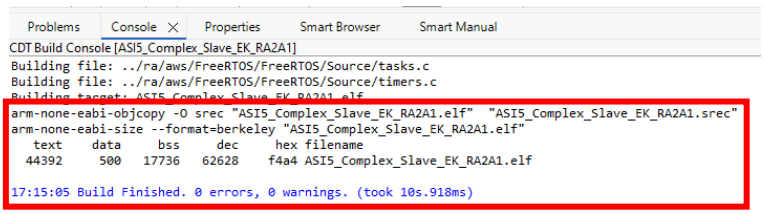

Figure 21: e2studio Console Build Information

3. Connect the EK-RA2A1 via J11 (Debug USB) to the PC.
4. Create a debug session by the following steps:
   a. Go to **Run** > **Debug configurations**.
   b. Double-click on **Renesas GDB Hardware Debugging**.
   A local debug configuration is created.
   c. Ensure that the **Main** settings (Figure 22) and **Debugger** settings (Figure 23) are set correctly.

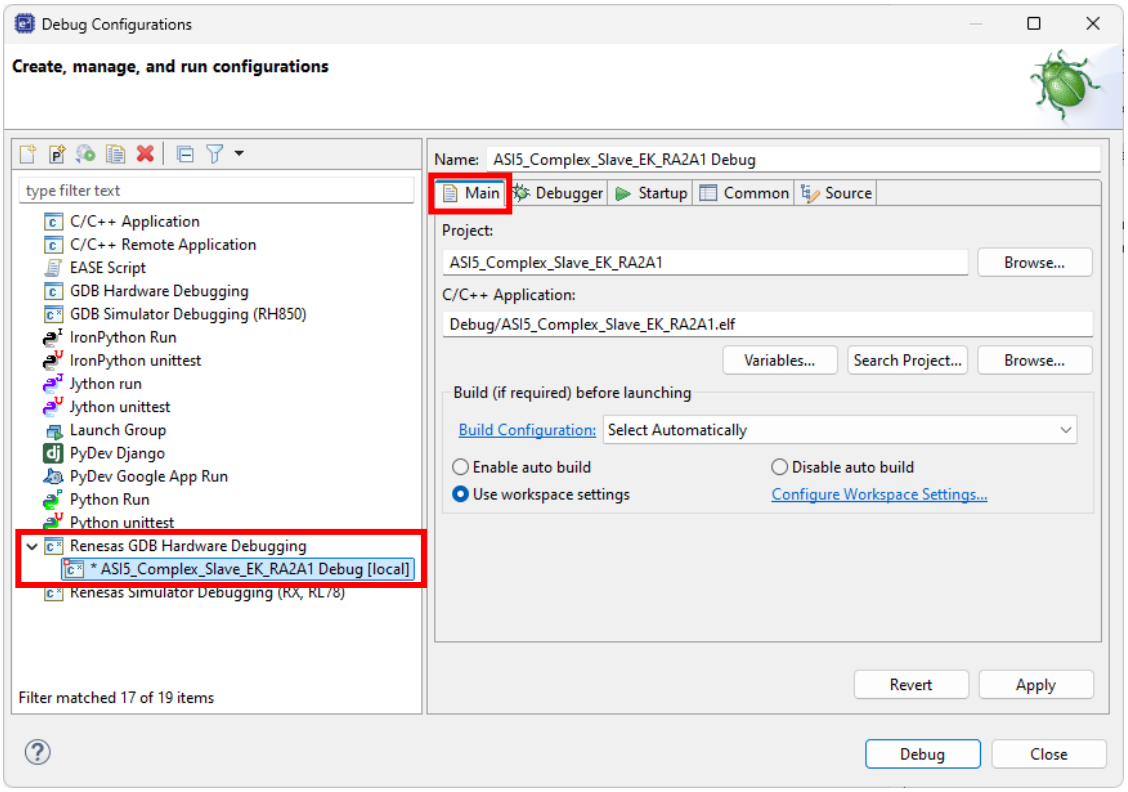

Figure 22. Debug Configuration

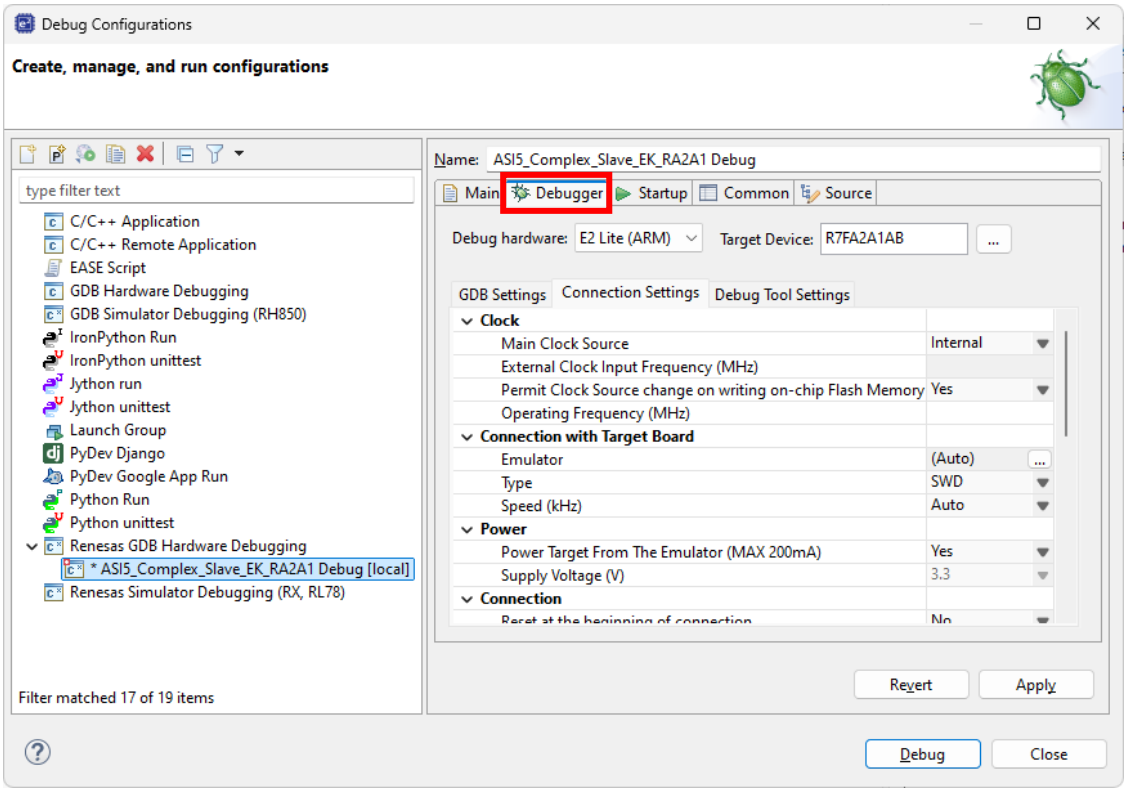

Figure 23. Debug Configuration Debugger Settings

d. Click on **Apply**.

e. Click on **Debug** to launch the debug session.

f. Click on **Resume** button twice in the Debug view configuration,
Pause the code execution via the **Suspend** button or terminate the session via **Terminate** button (see Figure 24).

Figure 24. Debug Session Control Elements

7. Example PROFINET / ASi-5 Test Setup with Complex Slave

This example integrates the ASi-5 Complex Slave into an industrial network to display data in a virtual PLC environment on a PC.

The network (Figure 25) is composed of the following:

- A PROFINET network
- An ASi-5 fieldbus network.

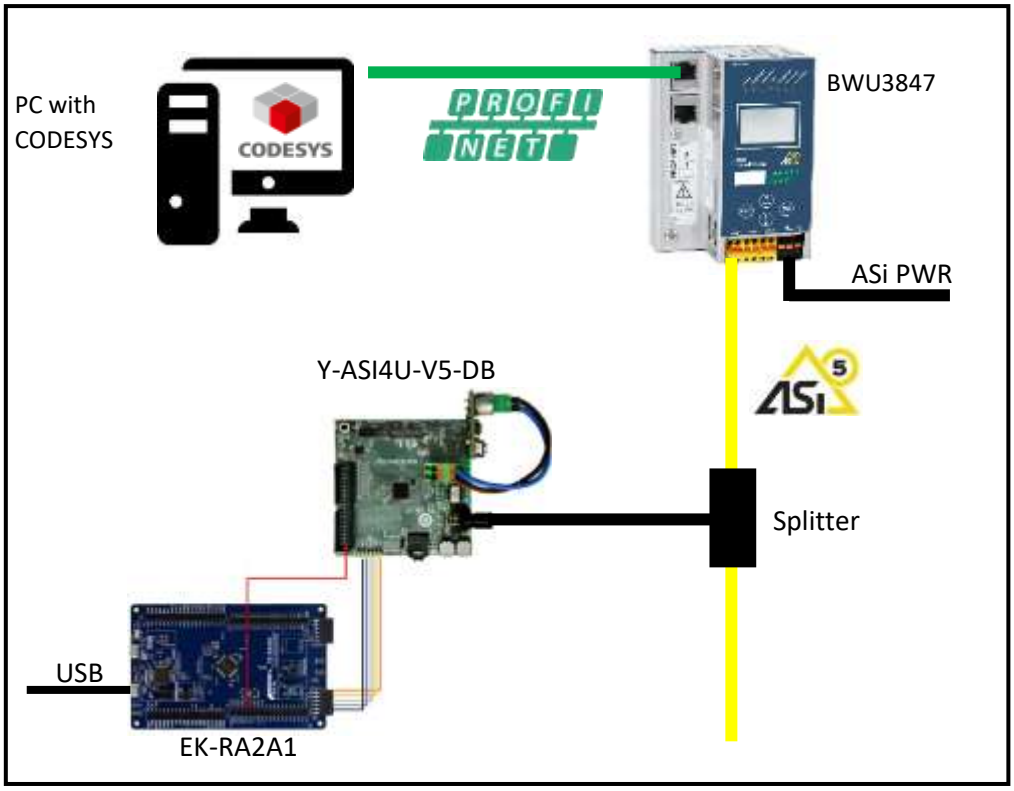

Figure 25. PROFINET/ASi-5 Network Structure

This example uses a BWU3847, an ASi-5/3 to PROFINET gateway. CODESYS® software running on a PC serves as the PLC implementing a PROFINET Controller.
It is possible to implement this solution with EtherCAT, EtherNet-IP, or other industrial network gateways.

7.1 BWU3847 Configuration

Follow these steps to configure the BWU3847 gateway correctly:

1. Set up the network as shown in Figure 25. Power the EK-RA2A1 via USB.
2. Power the BWU3847 with an ASi Power Supply (red and yellow wires). Connect the BWU3847 to the user computer using an Ethernet cable on PROFINET Port 2 (upper port), refer to Figure 26.

Figure 26. BWU3854 Cabling

3. Click on **OK** on the BWU3847 to enter the main configuration menu. Use the push buttons of the gateway to navigate through the menu as the following:
   a. Navigate to **PROFINET** > **TCP/IP** > **CONFIGURATION** and set:
      IP address: 192.168.42.2
      Netmask: 255.255.255.0
      Gateway: 0.0.0.0.
   b. Return to the main menu and navigate to **Slave Addr Tool** > **ASI-5 Circuit 1**.
   c. Select **Switch mode** if **Only available in prj mode** message is displayed.
   d. Select **Add new slave** and confirm **next slavenumber "001"**.
   e. Select the shown slave number **xxxxxxxxxxxxxx** and click **set to 001**.
      The BWU3847 confirms by displaying **OK**.
      Note: If the ASi Slave does not appear, press the reset switch on the EK-RA2A1 and the reset switch on the Y-ASI4U-V5-DB.
   f. Return to the main menu and navigate to **Quick Setup**.
   g. Confirm the warning message by clicking **OK**, then select **Store + run**.
      The BWU3847 confirms by displaying **OK** and returning to main display.
4. Configure the IP Address of the user PC's Ethernet interface as shown in Figure 27.

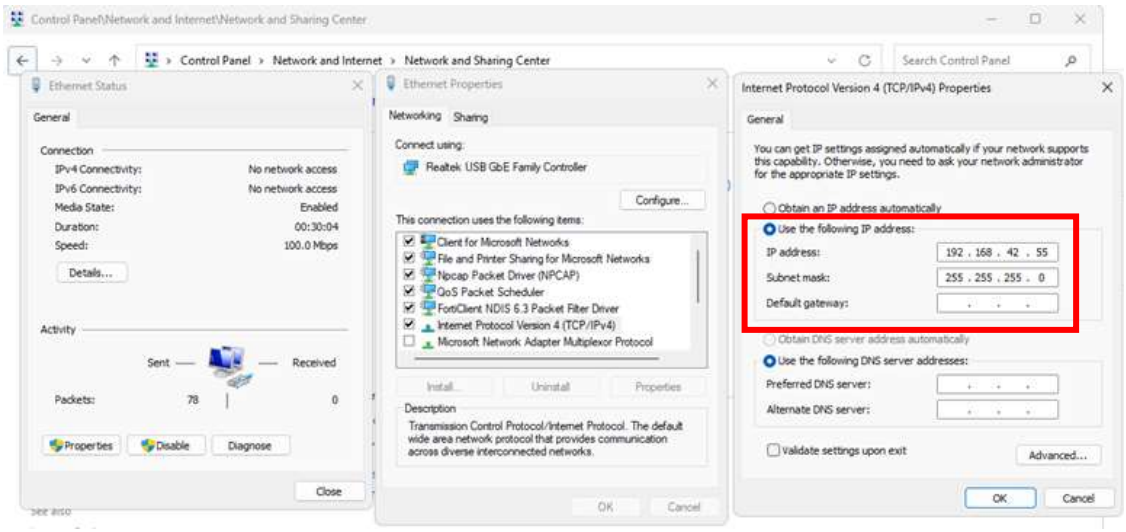

Figure 27. PC IP settings

5. Open a web browser on the user PC and access the webserver of BWU3847 through https://192.168.42.2/html/startreact.html#/
   a. On **Home** tab, the ASi Slave is listed in ASi Circuit 1, see Figure 28.
   b. Click on ASi Slave **001**.

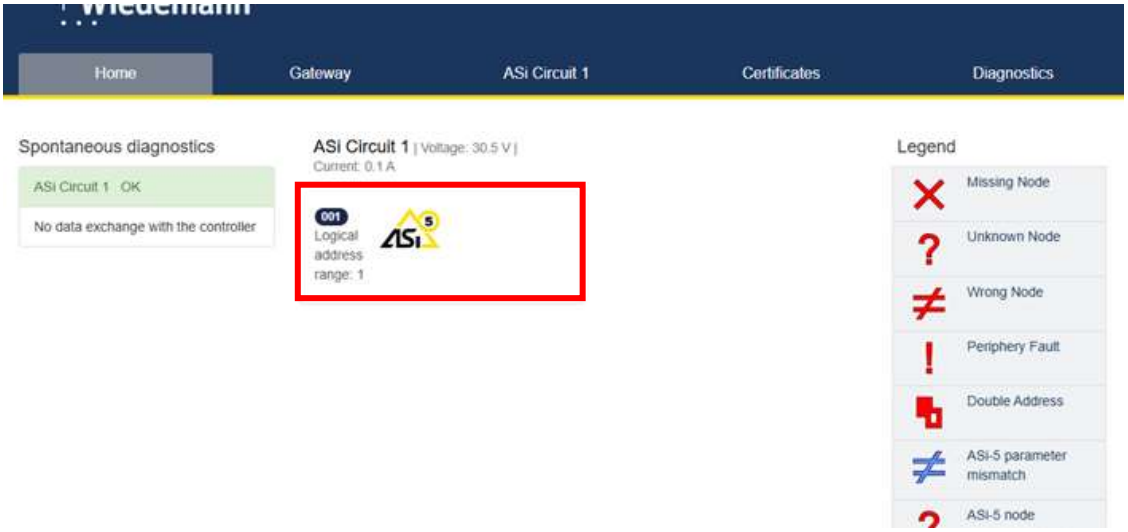

Figure 28. BWU3847 Webserver Home Tab

The Webserver shows details of the Slave, including Input and Output Bytes. The first two Address Bytes show the 16-bit counter values, increasing by 10 per second, see Figure 29.

Figure 29. BWU3847 Webserver Slave Details

By clicking the  icon for locator function, **RP7_0** and **RP7_1** on the Y-ASI4U-V5-DB blink simultaneously.

7.2 CODESYS Installation and Configuration

This section explains how to create a CODESYS project and integrate the Bihl+Wiedemann BWU3847 as a PROFINET Device.

7.2.1 Prerequisites

- Download and install CODESYS Development System V3 64 Bit from: https://store.codesys.com/en/codesys.html#product.attributes.wrapper
- BWU3847 is powered and reachable over Ethernet.

If using the PROFINET_Example.project from the file package, open it through **File** > **Open Project**… then proceed with step 4 in section 7.2.3.

7.2.2 Installation

Follow this procedure to create a project that integrates the BWU3847 into a PROFINET network:

1. Create a new project:
   a. Select **File** > **New Project**…
   b. In **Categories**, select **Projects**.
   c. In **Templates**, select **Standard Project**.
   d. Enter a project name and click **OK** (see Figure 30).

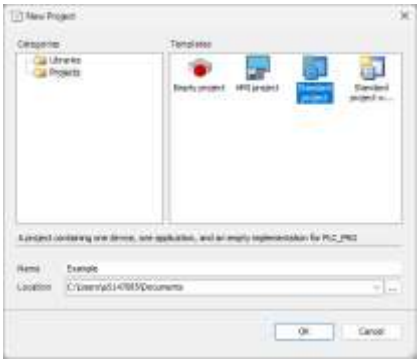

Figure 30. New Project Pop-Up

   e. In the dialog (see Figure 31), select:

i. Device: **CODESYS Control Win V3 (CODESYS)**.
ii. Programming language: **Structured Text (ST)**.

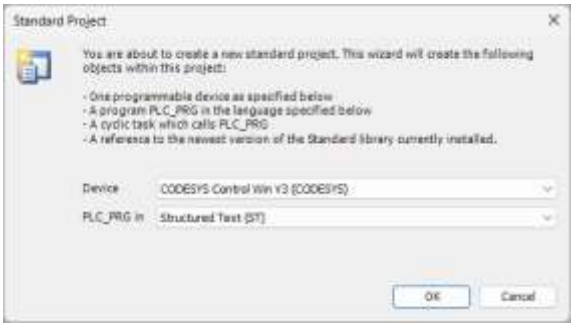

Figure 31. Standard Project Settings Pop-Up

f. Click **OK**.

2. Install the BWU3847 Device Description (GSDML):
   a. Download the BWU3847 GSD package 3847_GSD.zip from: https://www.bihl-wiedemann.de/eu/products/as-interface-mastergateways/gateways/product-selector-gateways/s/BWU3847
   b. In CODESYS, install the GSDML file:
      i. Select **Tools** > **Device Repository**…
      ii. Click **Install**…
      iii. Select the **GSDML-V2.41-Bihl und Wiedemann-AS-i Gateway BWU3847-20250610.xml file**.
      iv. Click **Open**, then close the **Device Repository** window after installation completes.

3. Add PROFINET Devices to the CODESYS Device Tree
   Add the Ethernet adapter, PROFINET controller, BWU3847 gateway, and the I/O module in the order below.
   a. Add the PC Ethernet adapter:
      i. In the **Devices** tree, right-click **Device (CODESYS Control Win V3)**.
      ii. Select **Add Device**…
      iii. In the dialog, select: **Fieldbuses** > **Ethernet Adapter** > **Ethernet**.
      iv. Click **Add Device** (see Figure 32).

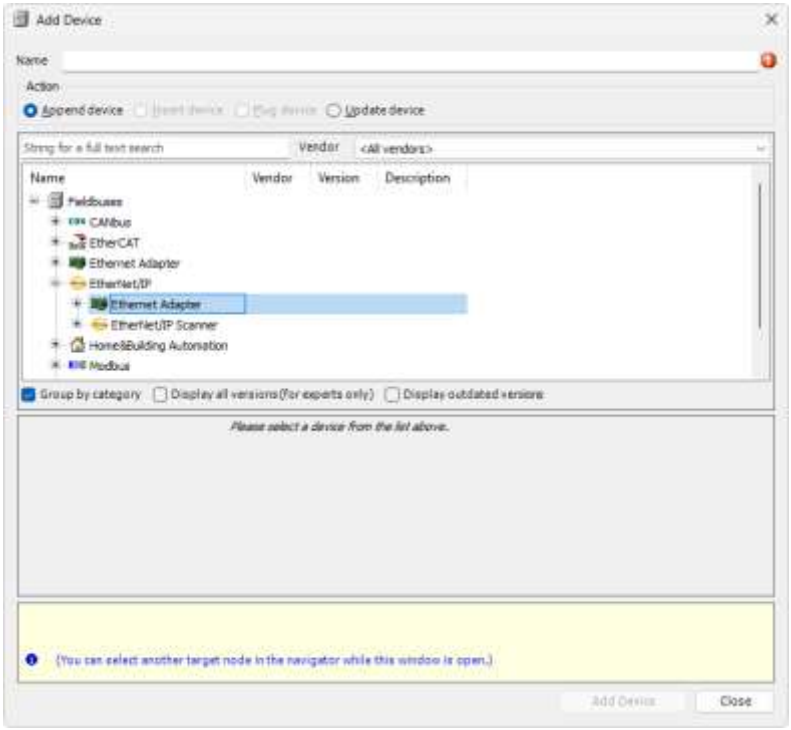

Figure 32. Add Device Pop-Up

b. Add the PROFINET controller:
   i. In the **Devices** tree, select **Ethernet (Ethernet)**.
   ii. Choose: **Fieldbuses** > **PROFINET** > **PROFINET Controller** > **PN-Controller**.
   iii. Click **Add Device**.

c. Add the BWU3847 as a PROFINET device:
   i. In the **Devices** tree, select **PN_Controller (PN-Controller)**.
   ii. Choose **Fieldbuses** > **PROFINET** > **PROFINET Device** > **Gateway** > **AS-i** > **B+W BWU3847 AS-i PROFINET Gateway** > **BWU3847 DAP V2.41 NAP S2**.
   iii. Click **Add Device**.

d. Add the I/O module (Digital I/O Data):
   i. In the **Devices** tree, select **BWU3847_DAP_V2_41_NAP_S2**.
   ii. Choose **Fieldbuses** > **PROFINET** > **PROFINET Module** > **Digital I/O Data** > **AS-i** > **B+W BWU3847 AS-i PROFINET Gateway** > **BWU3847 DAP V2.41 NAP S2**.
   iii. Click **Add Device**.
   iv. Click **Close**.

After these steps, the device tree should match Figure 33.

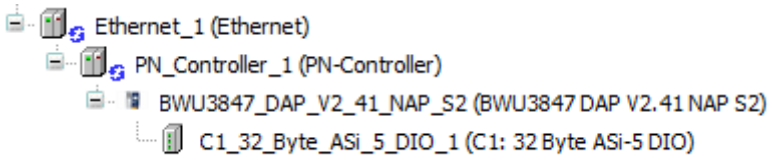

Figure 33. Ethernet / PROFINET Device Tree

7.2.3 Configuration

Follow this procedure to configure the added devices:

1. Configure the Ethernet Adapter:
   a. Double-click **Ethernet (Ethernet)**.
   b. Open the **General** tab.
   c. Click **Browse…** and select the correct **PC Ethernet interface** (as identified in section 6.1).
   d. Configure the adapter IP settings according to Figure 34.

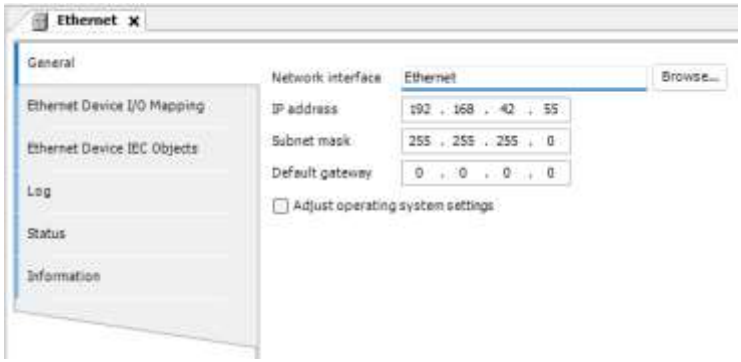

Figure 34. General Configuration Ethernet Adapter

2. Configure the PROFINET Controller:
   a. Double-click **PN_Controller (PN-Controller)**.
   b. Open the **General** tab.
   c. Set the adapter parameters as shown in Figure 35.

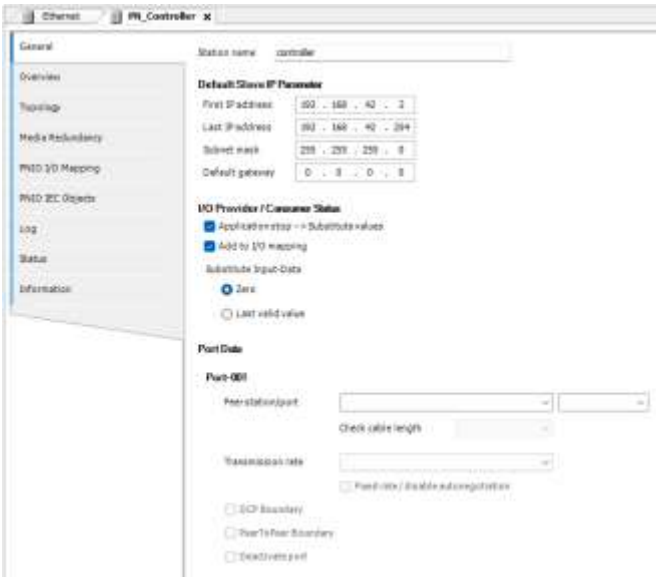

Figure 35. General Configuration PROFINET Controller

3. Configure the BWU3847 PROFINET Device:
   a. Double-click **BWU3847_DAP_V2_41_NAP_S2**.
   b. Open the **General** tab.

c. Set the device parameters according to Figure 36.

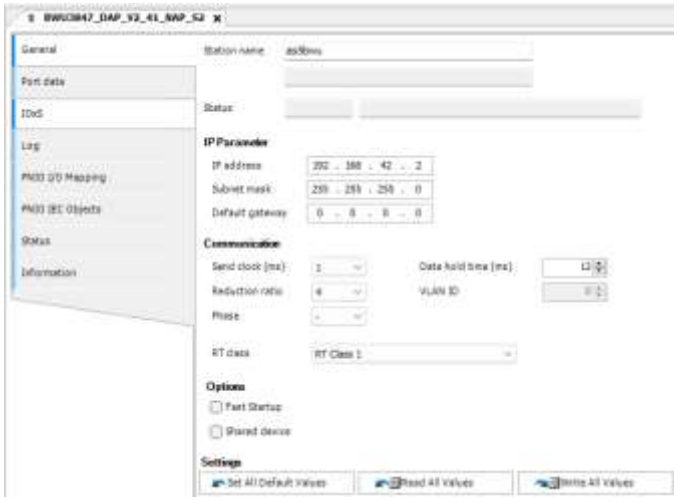

Figure 36. General Configuration PROFINET Gateway

4. View/Map the I/O Data (Sensor Data):
   a. Double-click **C1_32_Byte_ASi_5_DIO (C1: 32 Byte ASi-5 DIO)**.
   b. Open the **PNIO Module I/O Mapping** tab.
   c. Expand **Sensor Data variable** (see Figure 37).

When the PLC is running, this tab is used to visualize the sensor data of the connected AS-i complex slave(s).

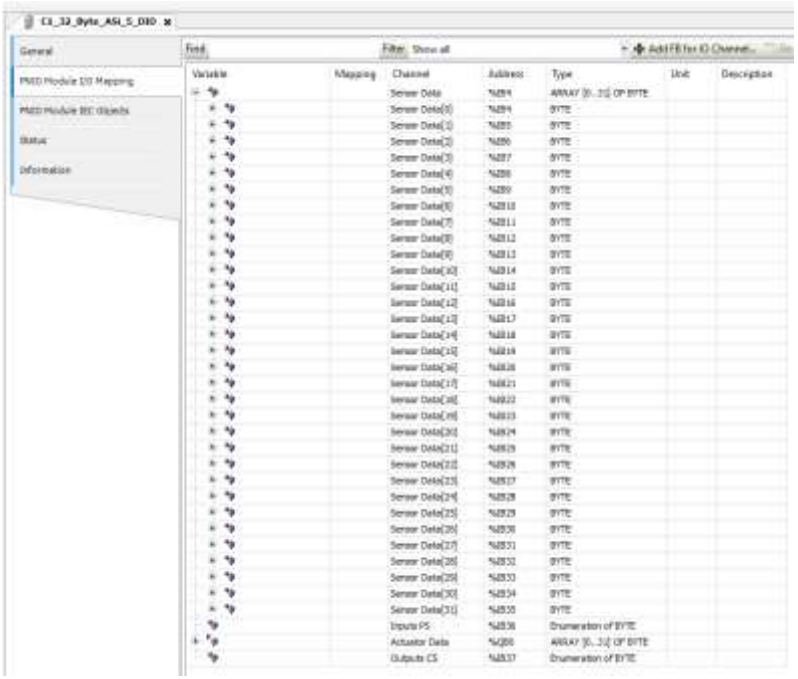

Figure 37. PNIO Module I/O Mapping

7.3 Network Activation & Data Visualization

Execute the project from section 7.2 on a virtual PLC within CODESYS to enable data visualization by the following steps:

1. In the Windows taskbar notification area, right-click the **CODESYS Control Win SysTray Icon** and select **Start PLC** (see Figure 38).
   The icon turns red, indicating the PLC is running.

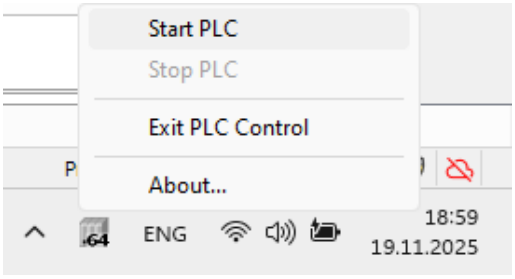

Figure 38. CODESYS Control Win SysTray

2. In CODESYS, select **Build** > **Generate Code**.
3. Click **Login** in the toolbar (see Figure 39) and enter credentials for virtual PLC login.
   CODESYS switches to execution mode.

Figure 39. Symbol Bar Login Icon

4. Click **Start** next to **Login** to begin code execution.
   All devices marked with green icons are online (see Figure 40). Figure 41 shows the successful PROFINET connection to PLC for BWU3847.

Figure 40. Device Tree Execution Mode

Figure 41. BWU3847 Display

In the **PNIO Module I/O Mapping** tab of **C1_32_Byte_ASi_5_DIO (C1:32 Byte ASi-5 DIO)**, the 16-bit counter value of the ASi Complex Slave appears in Sensor Data[0] and Sensor Data[1] Bytes (see Figure 42).

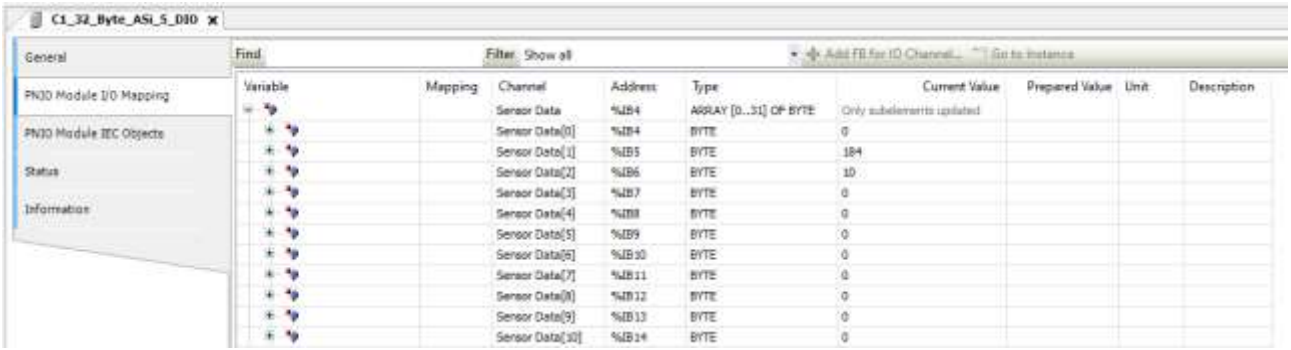

Figure 42. Sensor Data of ASi Complex Slave

8. Revision History

Revision	Date	Description
01.41	Apr 10, 2026	▪ Converted to new template ▪ Updated Figure 1 ▪ Added Table 2 ▪ Updated Figure 4 and Figure 5
01.40	Nov 24, 2025	▪ Dataflash generator update ▪ RA2A1 platform shift ▪ Network example update ▪ General rework and restructuring
01.30	Jan 14, 2022	▪ Add explanation about MP function
01.20	Nov 29, 2021	▪ Add reference ASi academy + MP profile generation
01.10	July 03, 2020	▪ Add reference to Demo2
01.00	May 05, 2020	▪ First version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

ROHS COMPLIANCE

Renesas Electronics' suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/