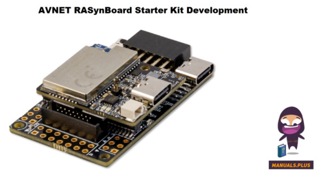
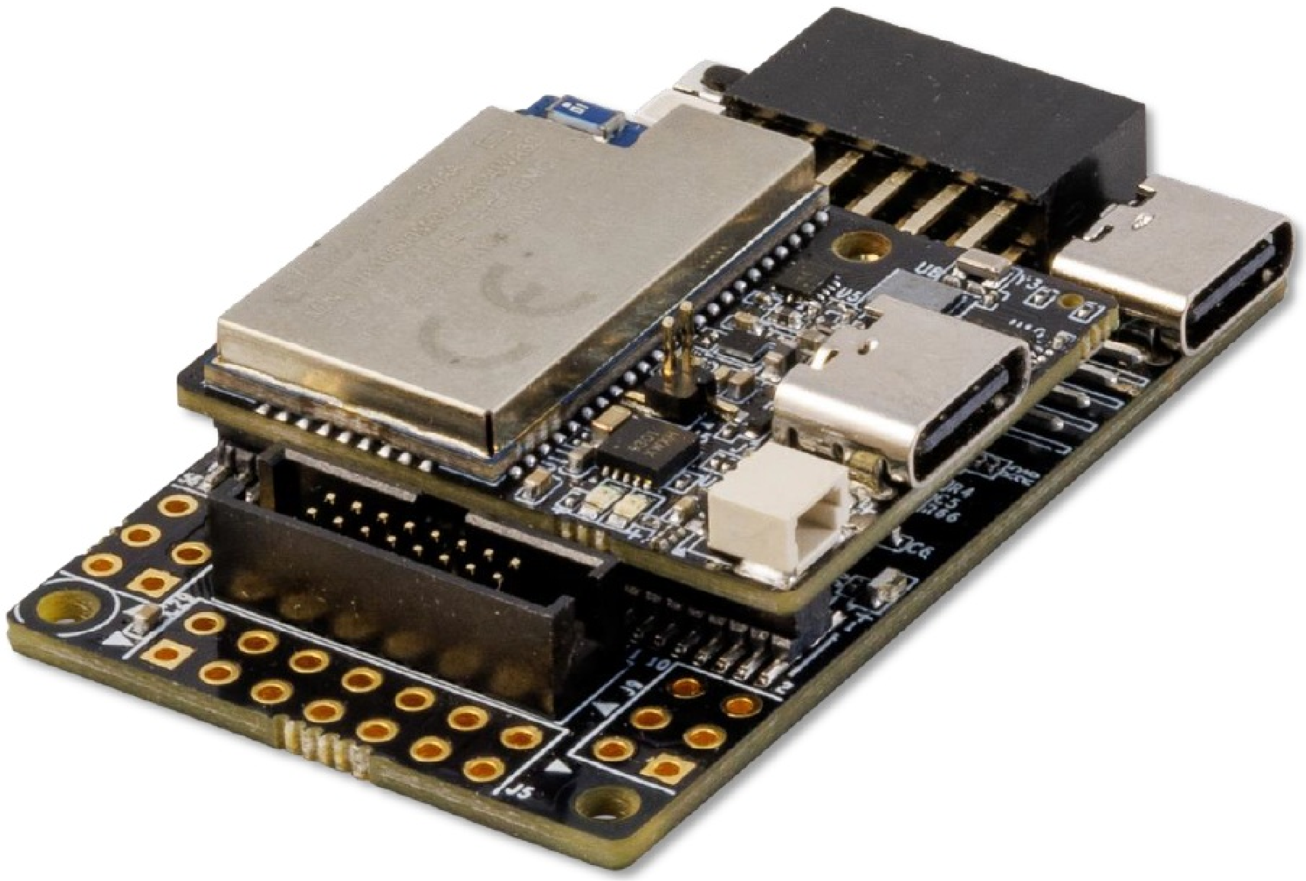


AVNET RASynBoard Starter Kit Development User Guide

Contents

AVNET

AVNET RASynBoard Starter Kit Development



Product Information

Specifications:

- Product Name: RASynBoard Starter Kit
- Document Version: 4.2
- Document Date: June 20, 2023
- Author: Peter Fenn
- Classification: Public

Product Usage Instructions

Hardware Setup for Application Development

1. Connect the 5V power input to power the RASynBoard Starter Kit.
2. If using a LiPo battery, connect it for input and charging.
3. Implement Debug UART for debugging purposes.

Software Installation

1. Utilize an SD Card for NDP120 firmware files and data recording.
2. Install NDP120 firmware files on a microSD Card.
3. Program the RA6M4 MCU using Renesas Flash Programmer (RFP).

Run the Pre-Built Alexa Demo Application

1. Perform keyword inference using the Syntiant Alexa model.

Build & Debug an Application using Renesas e2-studio IDE

1. Use the Renesas e2-studio IDE to build and debug applications.
2. Refer to notes regarding source code in the application.

I/O Board Expansion Connectors

1. Connect devices to the left-side connectors as needed.
2. Utilize the right-side connectors for additional expansion.

Hardware Requirements

- **RASynBoard EVK (Starter Kit)**: Avnet p/n: AES-RASYNB-120-SK-G
- **2mm Jumper**: Handle version preferred for 2mm-pitch header
- **PL2303TA based USB-to-Serial Debug Console Cable**
- **USB 2.0 micro SD Card Reader/Programmer**
- **USB-C to USB-A 3ft power+data cable**

FAQ

- Q: Where can I find additional technical support for the RASynBoard Starter Kit?
  A: For additional technical support, please visit the official website or contact the manufacturer directly.

Document Control

- Document Version: 4.2
- Document Date: 06/20/2023
- Document Author: Peter Fenn
- Document Classification: Public
- Document Distribution: Public

Version History

Version	Date	Comment
4.2	06/20/2023	Public release for production version (v3) RASynBoard PCBs

Hardware Requirements

Listing of hardware items used during development (typical pricing shown)

#	Image	Hardware Details	Notes
1		RASynBoard EVK (Starter Kit) Avnet p/n: AES-RASYNB-120-SK-G URL: https://avnet.me/rzboard	*RASynBoard Starter Kit item*
2		2mm Jumper (handle version preferred) for 2mm-pitch header *(Supplied in Kit. Disables E2OB debugger MCU when not in use)* https://www.amazon.com/CQRobot-Standard-Connector-Raspberry-Motherboard/dp/B0B5LFMPRB/	*RASynBoard Starter Kit item ($0.08 ea.)*
3		PL2303TA based USB-to-Serial Debug Console Cable https://www.amazon.com/JANSANE-PL2303TA-Serial-Console-Raspberry/dp/B07D9R5JFK	*$3.66 ea. (in 3-pack)*
4		USB 2.0 micro SDCard Reader/Programmer https://www.amazon.com/IOGEAR-MicroSD-Reader-Writer-GFR204SD/dp/B0046TJG1U/	*$4.99 ea.*
5		USB-C to USB-A 3ft power+data cable. eg. https://www.amazon.com/gp/product/B09XVCQSR5/	*$2.40 ea. (in 5-pack)*

Software Requirements

1		**Renesas Flash Programmer** *(version **3.11.01**, 5 Jan 2023)* https://www.renesas.com/rfp Download link for Windows version: https://www.renesas.com/us/en/document/swe/renesas-flash-programmer-v31101-windows
2		**Renesas e2Studio IDE** *(version **2022-10** or later)* https://www.renesas.com/e2studio Download link for Windows version for RA family: https://www.renesas.com/us/en/software-tool/e2studio-information-ra-family
3		**Tera Term terminal application** https://osdn.net/dl/ttssh2/teraterm-4.106.exe/
4		**PL2303 software driver** for Windows 10 prolific_usb_serial_3.8.28.0(station-drivers).zip

Overview

RASynBoard Core Board is a tiny (25mm x 30mm), ultra-low power, edge AI/ML board, based on Syntiant NDP120 Neural Decision Processor, a Renesas RA6M4 host MCU plus a power-efficient DA16600 Wi-Fi/BT combo module. The NDP120 subsystem with on-board digital microphone, IMU motion sensor and SPI Flash memory, achieves highly efficient processing of acoustic- and motion events. Battery and USB-C device connectors facilitate stand-alone use, while a compact under-board connector enables integration with custom OEM boards and additional sensors.

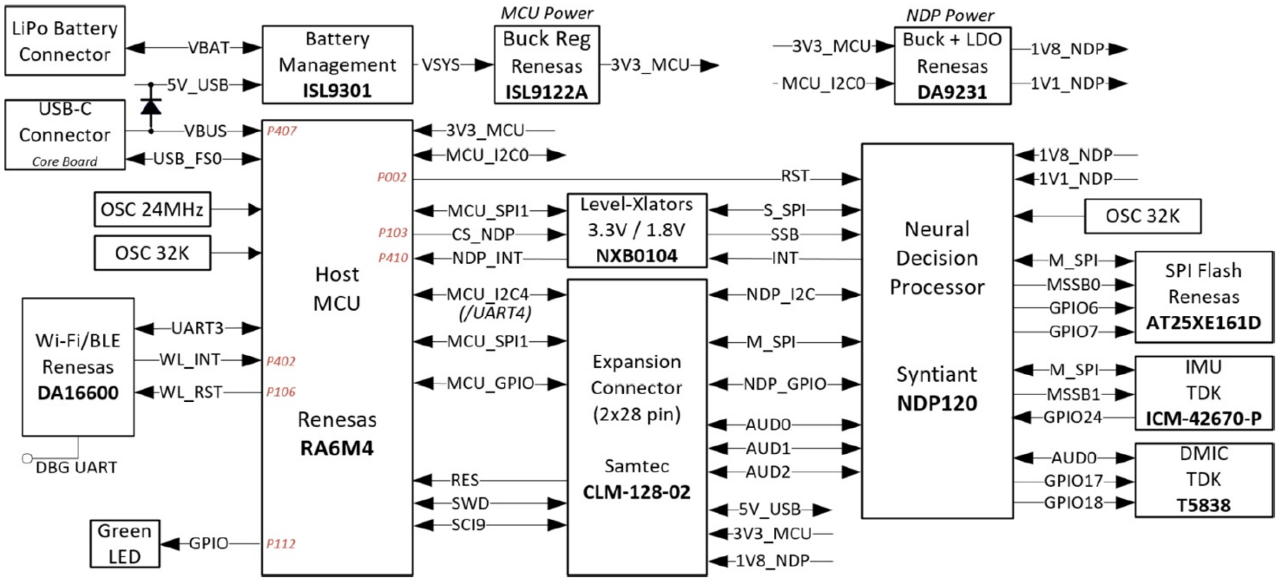
Renesas RA6M4 MCU application software development and debug is supported via Renesas e2 Studio IDE application, interfaced via the E2Lite debug interface implemented on RASynBoard's IO carrier board. With a

power-efficient 200 MHz Arm Cortex- M33 core, versatile set of peripheral interfaces and advanced security, the RA6M4 as a host microcontroller has much to offer, plus Wi-Fi and BLE connectivity via a companion power-efficient DA16600 wireless module, integrated onto the core board.
NDP120 application software and AI/ML models for popular use-cases (pre-engineered by Syntiant and others) are loaded from local microSD card or SPI Flash, for efficient execution on the ultra-low power NDP120 neural accelerator device, which is particularly well suited for Always-On Speech and Sensor-Fusion applications.

Core Board Features

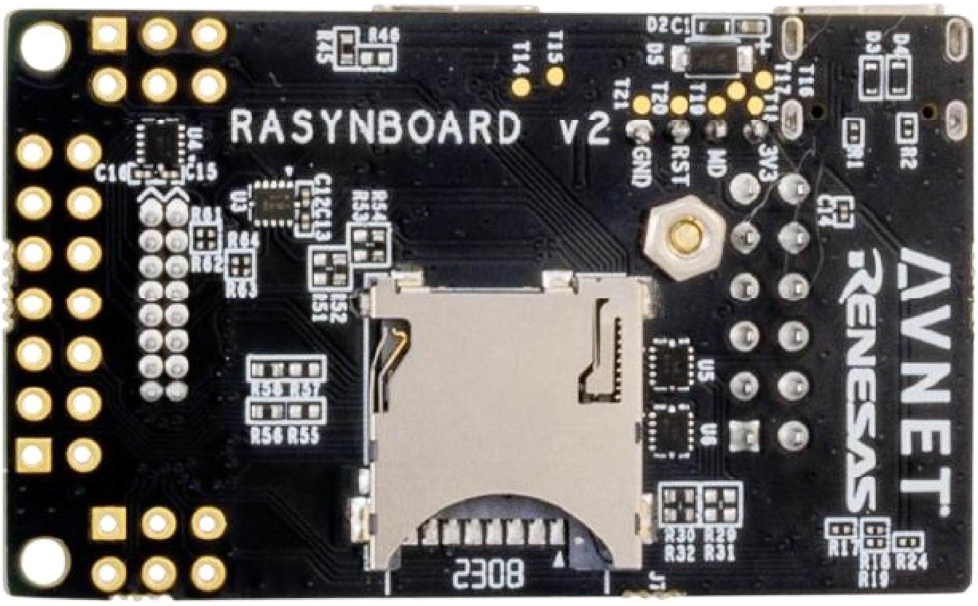
- Syntiant NDP120 Neural Engine
  - Syntiant Core 2 Deep Neural Network
  - Arm Cortex M0 and HiFi 3 DSP
- Renesas RA6M4 Microcontroller
  - 1x Arm Cortex M33 (200 MHz)
  - 1 MB flash memory, 256 KB SRAM
  - USB 2.0 device interface
- Renesas DA16600 Wi-Fi/BT Combo Module
  - 802.11bgn 1×1 2.4 GHz Wi-Fi and BT 5.1
- Additional Onboard Memory
  - 2 MB SPI NOR Flash
- Battery Management
  - LiPo battery management and connector
- Onboard Sensors
  - IMU 6-axis motion sensor (ICM42670-P)
  - PDM digital microphone (MMICT5838)
- Expansion Connector & Dimensions
  - 2×28 pin board-to-board connector
  - 25 mm x 30 mm

Core Board Block Diagram



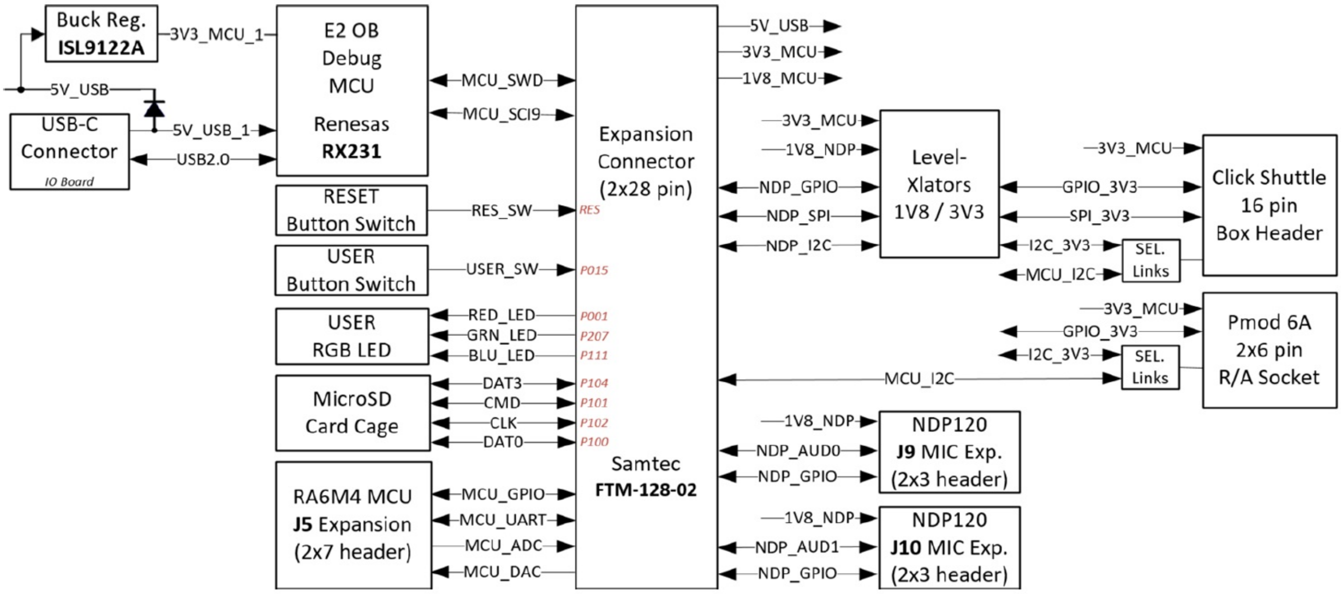
RASynBoard Starter Kit adds an IO Board (50mm x 30mm), for a versatile, compact, two-board evaluation kit assembly. This pins-out a subset of the NDP120 and RA6M4 I/Os to popular Pmod, Click header and expansion header footprints, enabling connection with additional external microphones and sensor options. An onboard debugger MCU (SWD and UART interfaces), button switches, RGB LED and removable MicroSD storage, further maximize the prototyping versatility and utility.

IO Board Features

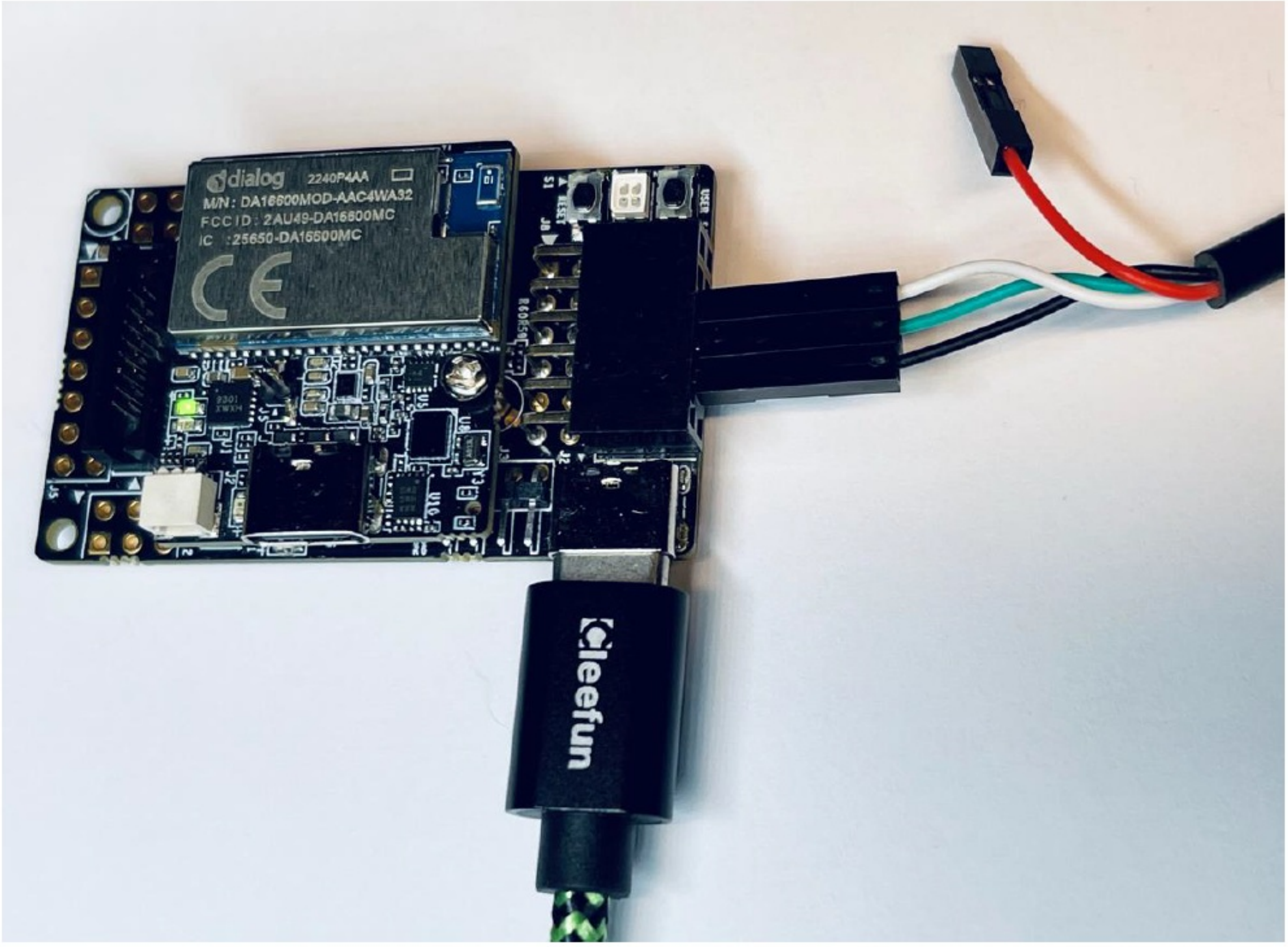


- Onboard Debugger and USB Serial interface
  - Renesas E2 OB debugger MCU (USB C to SWD and serial interf ace)
  - 3.3V buck regulator for debugger circuits Expansion Interfaces and Storage
  - 2×28 pin board to board connector
  - 2×8 pin MikroE Click shuttle box header
  - 2×6 pin Pmod type 6A (I2C) socket
  - 2×7 pin MCU e xpansion header
  - 2×3 pin DMIC expansion headers (two) 3.3V level translated expansion interfaces
  - Micro SD card cage for removable storage
- User Interfaces
  - 2x Button Switches (Reset and User)
  - 1x User RGB LED
- Dimensions
  - 50 mm x 30 mm

I/O Board Block Diagram



Hardware Setup for Application Development



5V Power Input

One of three connectors can be used to power RASynBoard Which power source to use, depends on the use case

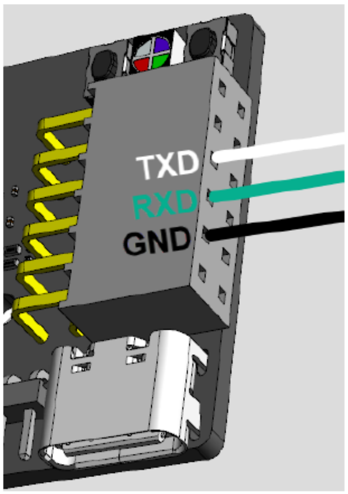
Connector	Use Case	Comments
IO Board USB-C connector	For RA6M4 flash programming, de bug and development runtime	Power & development interface **during software development**
Core Board USB-C connector	If Core Board used standalone	Power & flash programming if Core Bo ard is used **standalone**
Core Board battery connector	For LiPo battery operation	**Battery-powered operation**

While a 5V power source applied to either of the USB-C connectors can be used to power both boards, during application software development and testing for the RA6M4 MCU, simply power both boards from the development computer via the E2OB onboard-debugger USB-C connector on the

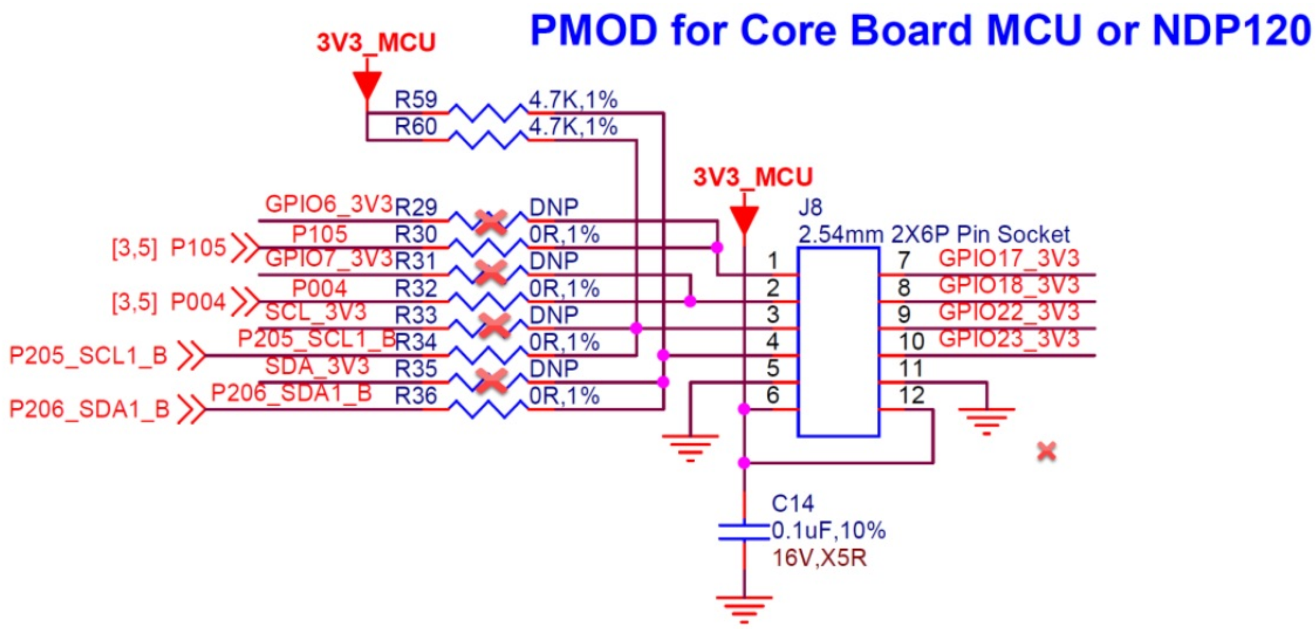
IO Board
LiPo Battery Input and Charging
The RASynBoard module is designed to also support operation from a Lithium Polymer battery On the Core Board two status LEDs only operate when external 5V power is connected:

LED Name	LED Color	Comments
PPR – Power Presence indicator	Green	
CHG – Charging indicator	Red	

Debug UART Implementation
During Starter Kit-based development and debugging of the RA6M4 application, it is convenient to use SCI4 assigned to P206 and P205 signals. To implement a debug UART, access these signals using fly-leads from the USB-UART cable, connected to the 3 pins of the J8 Pmod connector highlighted below. This UART is then accessed via the COM port that gets enumerated in Windows, using eg. Tera Term console application



#	Pmod Signal	NDP120 Pin / RA6M4 Pin	#	Pmod Signal	NDP120 Pin
1	INT	~~GPIO6_3V3~~ P105	7	GPIO	GPIO17_3V3
2	GPIO	~~GPIO7_3V3~~ P004	8	GPIO	GPIO18_3V3
3	SCL	~~SCL_3V3~~ P205 (TXD4)	9	GPIO	GPIO22_3V3
4	SDA	~~SDA_3V3~~ P206 (RXD4)	10	GPIO	GPIO23_3V3
5	GND	GND	11	GND	GND
6	3V3	3V3_MCU	12	3V3	3V3_MCU



Notes:

- The Tera Term console application should be configured for 115200 8N1 operation
- Pmod and Click connectors are factory-configured with I2C-related signals connected to I2C/UART pins of the RA6M4 microcontroller. Other signals are connected to the NDP120. (For optimum low-power performance, at a future date the sensors will be serviced directly by the NDP120)

Software Installation

1. Download and install the Renesas e2Studio IDE (version 2022-10 or later)
   https://www.renesas.com/e2studio
   Download link for Windows version for RA family:
   https://www.renesas.com/us/en/software-tool/e2studio-information-ra-family
2. Download and install Renesas Flash Programmer (version 3.11.01, 5 Jan 2023) **https://www.renesas.com/rfp**
   Download link for Windows version:
   https://www.renesas.com/us/en/document/swe/renesas-flash-programmer-v31101-windows
3. Download and install Tera Term (or equivalent) serial console application **https://osdn.net/dl/ttssh2/teraterm-4.106.exe/**
4. Download and install the PL2303 software driver for Windows 10 (for USB-Serial cable)
   prolific_usb_serial_3.8.28.0(station-drivers).zip

SD Card Use for NDP120 Firmware Files and Data Recording
The microSD card on underside of the IO Board serves multiple purposes:

- It provides a convenient way to load .synpkg firmware files into the NDP120 on startup
- It provides a way to copy .synpkg and .ini files into SPI flash on the Core board (ie for later standalone use of RASynBoard, where NDP120 firmware then gets loaded from SPI flash)
- It facilitates the recording of training data (eg. from onboard microphone or IMU) for training of the DNN model in 3rd party tools (eg. Edge Impulse Studio)

How to install NDP120 firmware files on microSD Card

On reset of RASynBoard, the RA6M4 application will attempt to load three .synpkg firmware files into the RAM memory of the NDP120 device.
By default, these .synpkg files load from uSDcard mass storage, based on settings in a config.ini file located on this uSDcard. What is to be loaded, is specified in the following format in this config file:

- MCU=mcu_fw_120.synpkg
- DSP=dsp_firmware_noaec_ff.synpkg
- DNN=menu_demo_512_rasyn_newph.synpkg

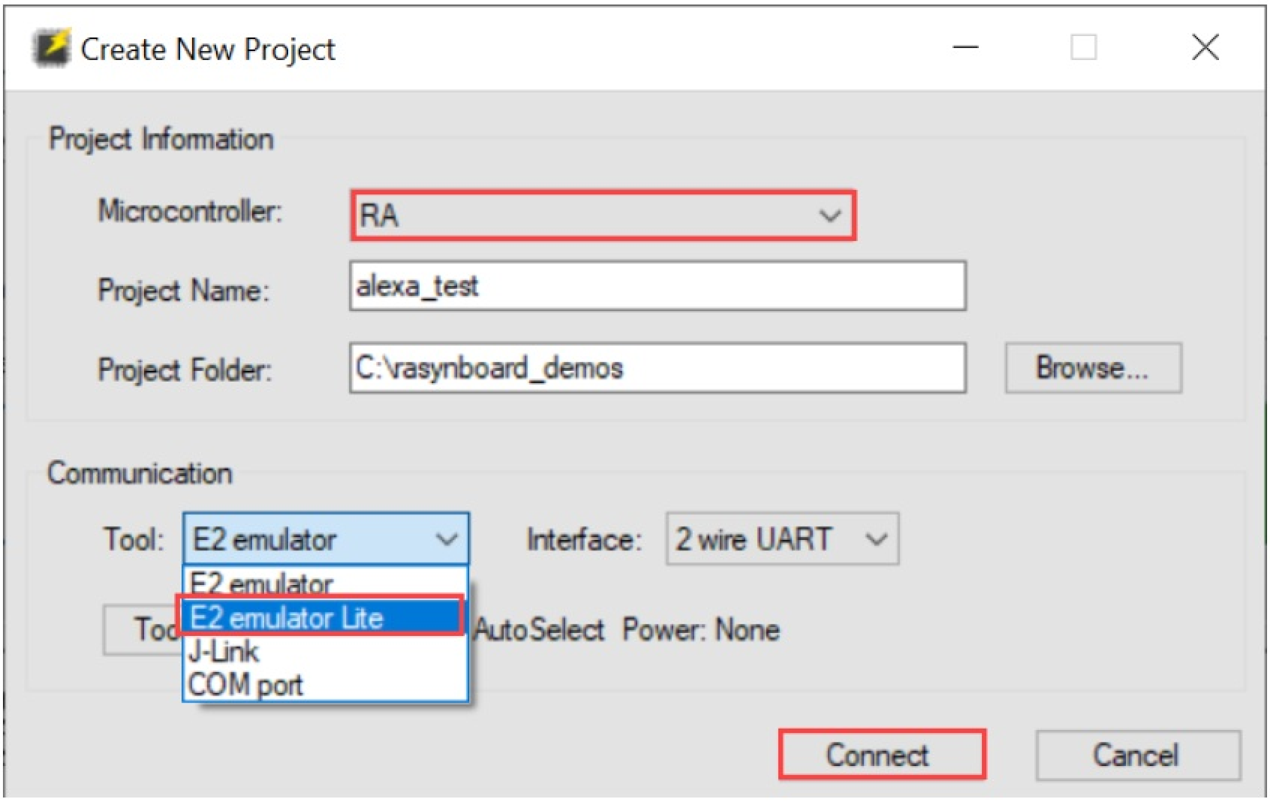
For backward compatibility, if this config.ini file is missing (or an older version of the demo application is used), then .synpkg files with the following default filenames must be present on the uSD card:

- mcu_fw_120.synpkg
- dsp_firmware.synpkg
- ei_model.synpkg

At this time, writing a new set of .synpkg files to uSDcard requires removal of uSDcard from the IO board, writing the files (from the computer) using a suitable SDcard adapter, then returning the uSDcard to the IO board.
Explanation of the different sections in this config.ini file is provided in the Appendix of this User Guide

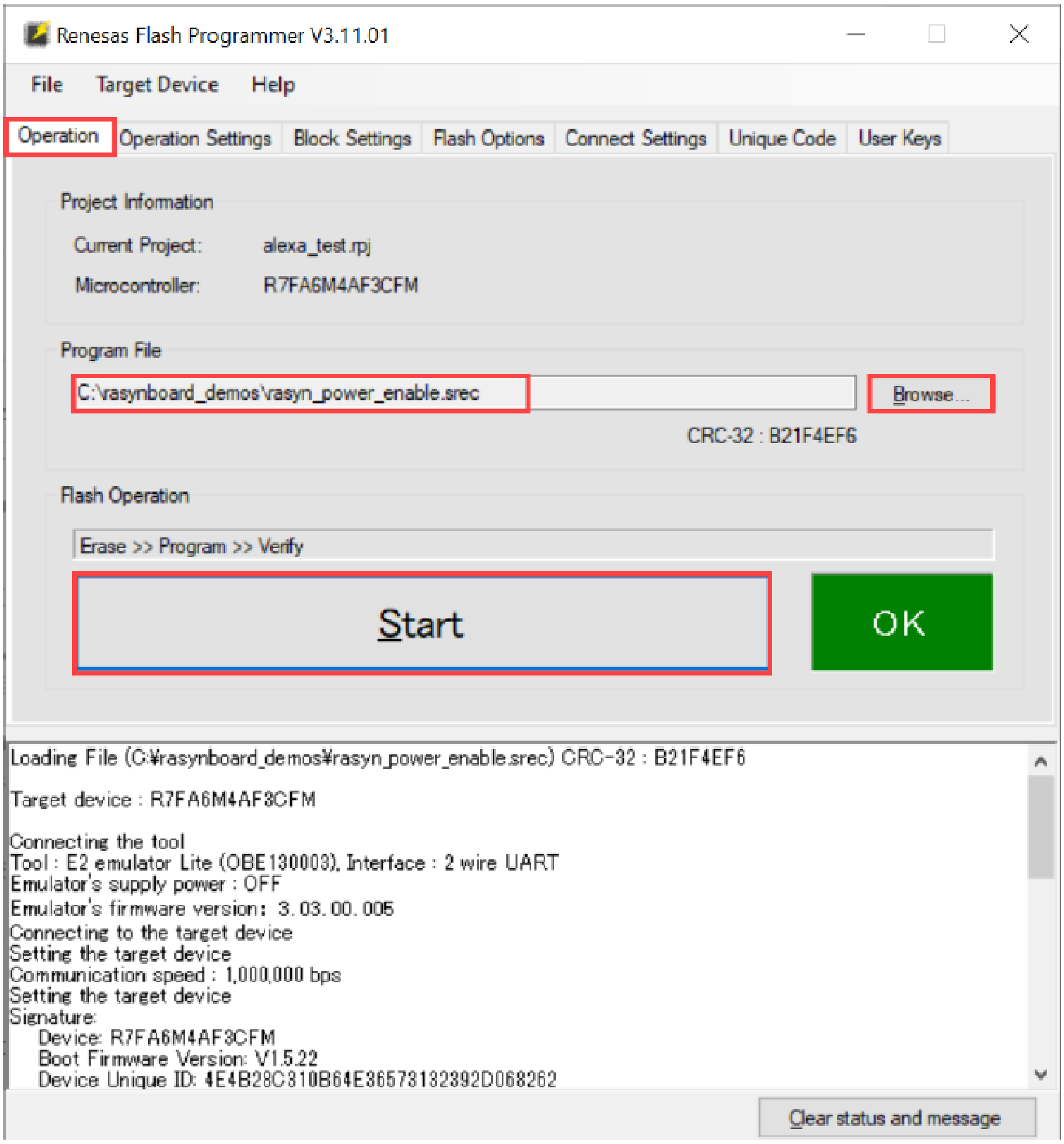
How to use Renesas Flash Programmer (RFP) to program RA6M4 MCU
If wanting to use a prebuilt RA6M4 binary file (supplied in .srec format), then use the following procedure:

1. Using the photos on page 7 prepare your hardware setup (do not power up RASynBoard yet)
2. To allow programming (and SWD debugging) of the RA6M4 MCU via the E2OB debugger on the IO Board:
    1. Remove the 2mm shorting jumper from J3 on the IO Board
    2. Remove the 2mm shorting jumper from J5 on the Core Board (if there is one fitted)
    3. Connect USB cable from IO Board USB-C connector to a USB-A port on the development PC
3. Open Renesas Flash Programmer (version 3.11.01)
4. From the menu bar, select File → New Project → complete the Create New Project form as shown in the screenshot below (for Tool, select E2 Emulator Lite) then click the Connect button…



(**Note**: For the Interface type, either 2 wire UART (the default) or SWD can be used)

5. Select the Operation tab, then for Program File click Browse, then select the pre-built binary file (.SREC)
6. Click the large Start button to program the selected .srec file into the RA6M4 MCU flash memory…



7. Once successfully programmed, power-down RASynBoard by disconnecting the USB cable from the development computer.
8. Fit the 2mm shorting jumper across pins 1 & 2 of J3 on the IO Board (to strap E2OB in Reset mode)
9. Reconnect USB cable from USB-C on IO Board to the development computer or other +5V source
10. Provided the uSD card has the expected three .synpkg files, the RGB-Blue LED should illuminate for about 3 seconds while these files are read from the SD card, and loaded into the RAM of the NDP120

Run the Pre-Built Alexa Demo Application
Keyword inference using the Syntiant Alexa model

1. After reset of the RA6M4 MCU, it will read the MCU, DSP and DNN .synpkg firmware files from the file-system on the FAT32 formatted uSDcard and load these into the NDP120 RAM memory.
2. Wait until firmware has loaded (RGB LED turns-off), then speak the “Alexa” keyword If this is recognized by the NDP120, two forms of confirmation will be seen:
    1. the RGB Green LED will briefly illuminate
    2. a message is also output to the Tera Term console each time the keyword is recognized

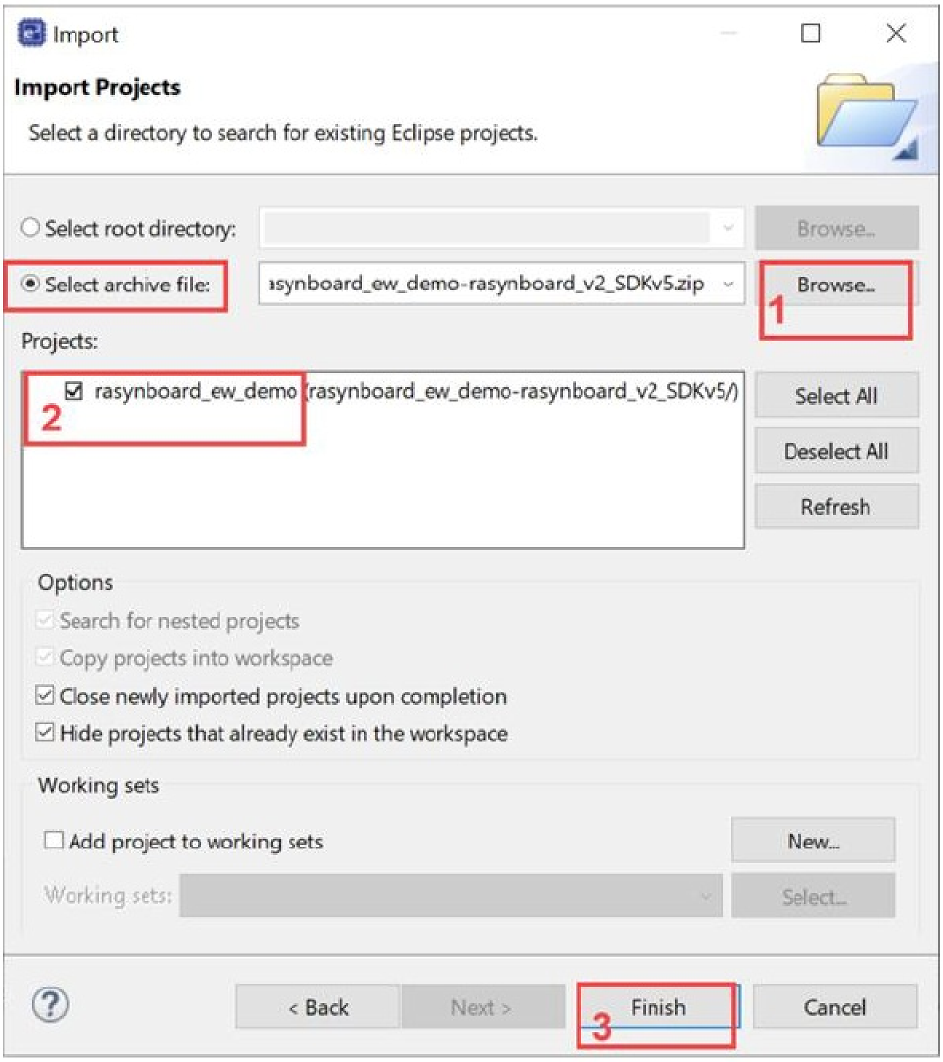
Build & Debug an application using Renesas e2-studio IDE
The rasynboard_ew_demo project is the FreeRTOS-based RA6M4 application that is factory-programmed into

RA6M4 MCU flash memory. It exercises multiple features of the RASynBoard Starter Kit. Originally developed for voice UI based demos at Embedded World, this application is useful to developers as it has evolved to support a wide range of RASynBoard functions, eg:

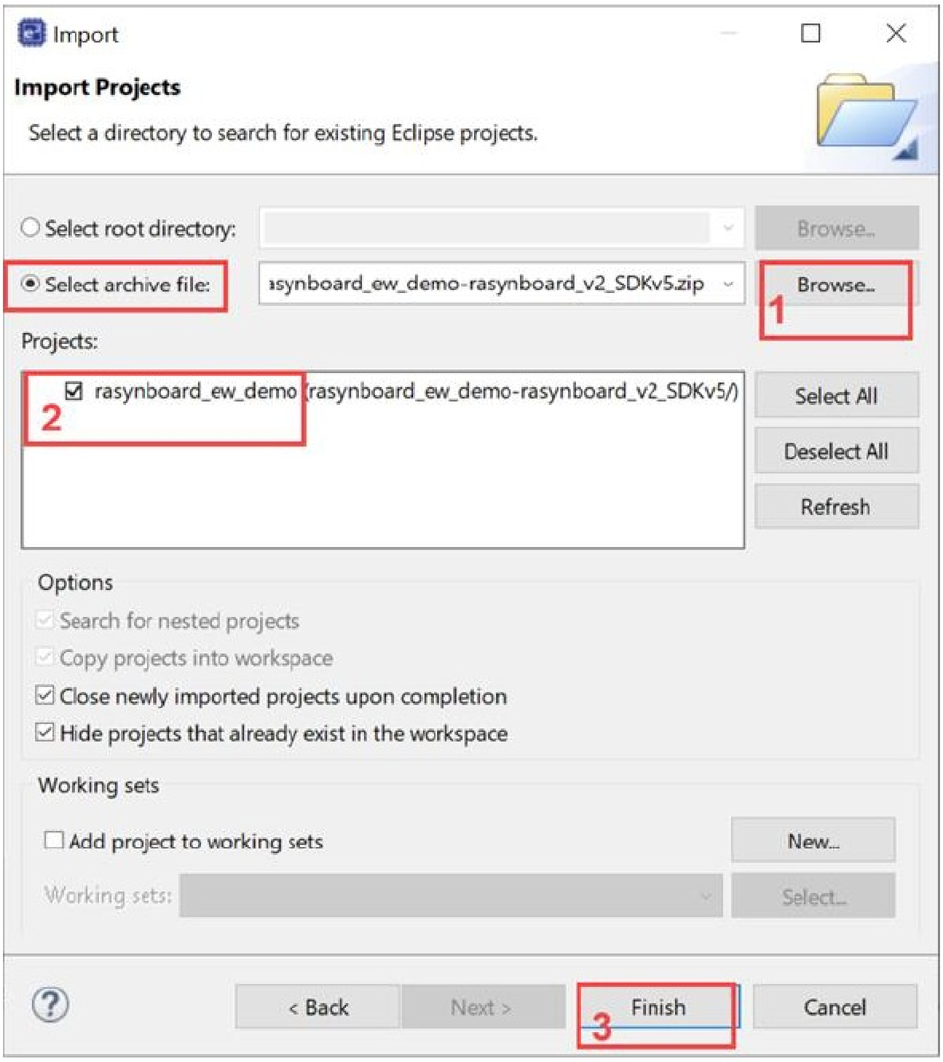
1. Load NDP120 firmware files from uSDcard (MCU, DSP, DNN .synpkg files)
2. Write recorded microphone audio to uSD card (.wav files)
3. Write NDP120 firmware to SPI Flash
4. Load NDP120 firmware from SPI Flash
5. Receive new voice-command inference results from the NDP120
6. Provide console debug output via UART (SCI4)
7. Service the local Status Green LED on the Core Board
8. Service the RGB_LED output on the IOBoard
9. Respond to button-press events on the IO Board
10. Establish wireless BLE comms with remote Raspberry Pi board that drives GUI display
11. Send inference results via BLE to remote Raspberry Pi board that drives GUI display
12. Implement Sleep Mode entry and exit (where only the NDP120 remains active)
13. etc…

To import and build the project rasynboard_ew_demo use the following steps:

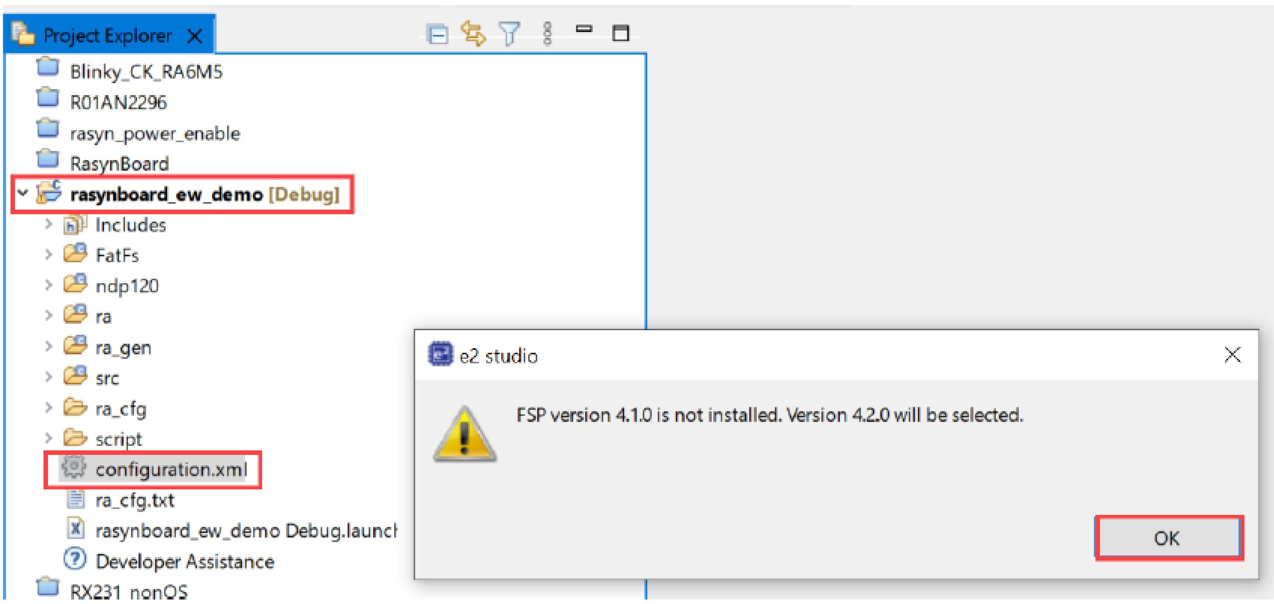
1. On development computer, open e2-studio IDE, then select “File -> Import” to open the import dialog



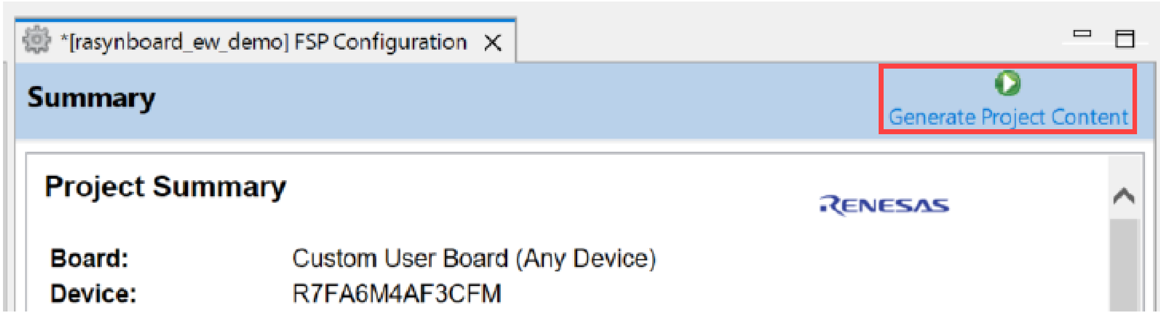
2. Select the rasynboard_ew_demo directory in the subsequent Import dialog



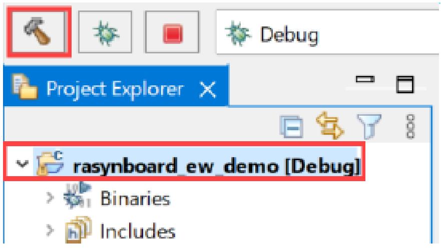
3. After importing an existing project into your workspace, when using this for first time, you must open the (FSP) configuration and click on Generate Project Content, before attempting to build the project!



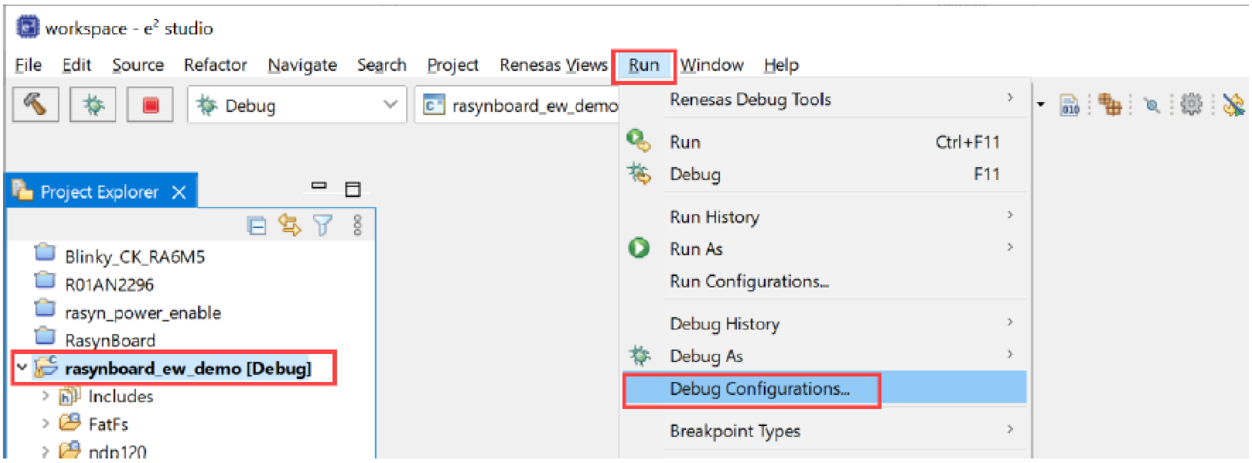
4. Click OK on the dialog warning that a different FSP version will be used, after the FSP panel opens-up, click on Generate Project Content (in top right-hand corner)



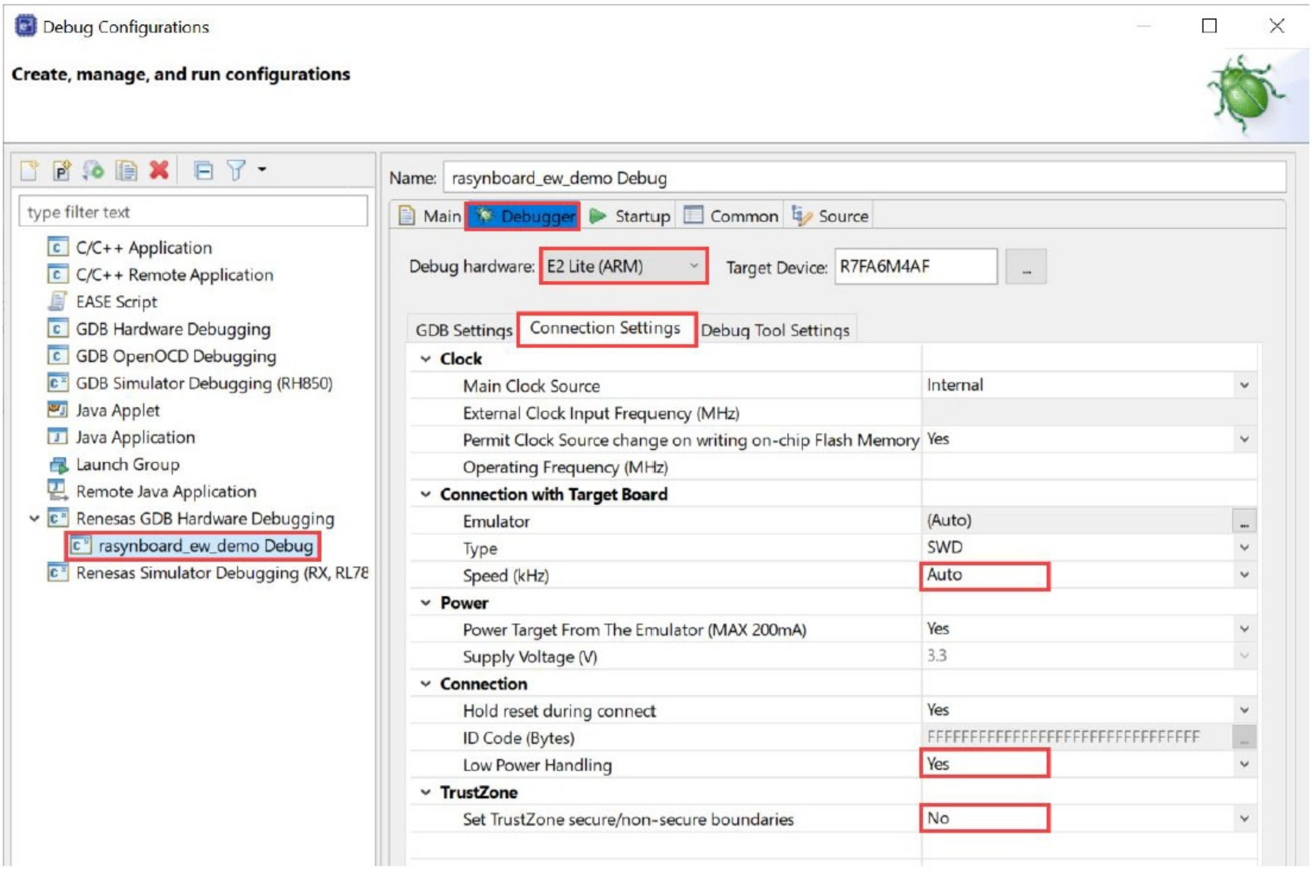
5. Now with the project name highlighted in Project Explorer left sidebar, click on the Build (hammer) icon on the toolbar to compile the project



6. An additional one-time setup step is needed before starting a debug session: With the project name selected, open the Run menu, then select Debug Configurations…



7. With the rasynboard_ew_demo Debug configuration selected in the left panel, open the Debugger tab, set the Debug hardware to E2 Lite (Arm), then select Connection Settings and make sure all settings match what is shown below…

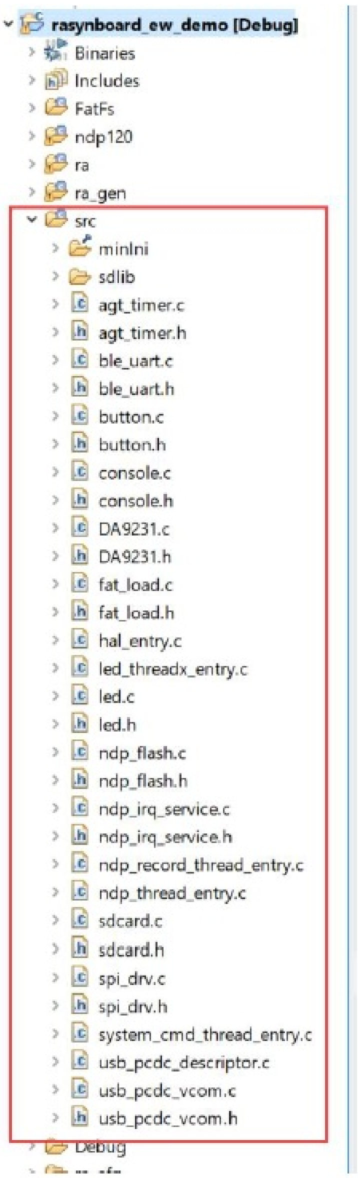


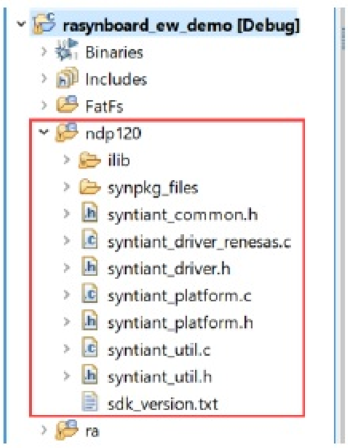
8. Once the Debug Configuration is correctly set, apply these settings, then launch Debug (Debug can be launched from within this form, or by using the Debug icon at top left of the IDE)

Notes regarding Source Code in this Application

The src folder contains the user source files for this custom application
The ndp120 folder contains Syntiant NDP120 Tiny iLib SDK library files with APIs accessed in this project
The ndp120\syn_pkg_files folder contains the MCU, DSP, DNN .synpkg files and a sample config.ini file Note!

1. Edit the config.ini file to match the specific configuration that you want to exercise
2. Make sure to copy all files in this folder to your uSDCard before running the application!!!





For this FreeRTOS based implementation, four threads are serviced (located in files of matching names)

- led_threadx_entry(); // Manages Core Board Green Status LED
- ndp_thread_entry(); // NDP120 configuration and service functions
- system_cmd_thread_entry(); // Manages IO Board RGB LED and Sleep mode entry/exit
- ndp_record_thread_entry(); // Manages audio .wav file recording to uSDcard

Syntiant SDK configuration APIs are called from the ndp_thread_entry() function:

```
ndp_irq_init();
init_fatfs();
button_init();
ble_uart_init();

/* Delay 100 ms */
R_BSP_SoftwareDelay(100, BSP_DELAY_UNITS_MILLISECONDS);
/* read config info of ndp firmwares */
get_synpkg_config_info();
/* Choose the appropriate debug print console */
if (get_print_console_type() == CONSOLE_USB_CDC)
{
    start_usb_pcdc_thread();
    console_deinit();
}
/* Start NDP120 program */
ret = ndp_core2_platform_tiny_start(0, 1);
if(ret == 0) {
    printf("ndp_core2_platform_tiny_start done\r\n");
    xSemaphoreGive(g_binary_semaphore);
} else {
    printf("ndp_core2_platform_tiny_start failed %d\r\n", ret);
}

ret = ndp_core2_platform_tiny_feature_set(SYNTIANT_NDP_FEATURE_PDM);
if (ret){
    printf("ndp_core2_platform_tiny_feature_set set 0x%x failed %d\r\n",
                SYNTIANT_NDP_FEATURE_PDM, ret);
}

ndp_info_display();

if (!is_motion_mode()) {
    ret = ndp_core2_platform_tiny_sensor_ctl(0, 0);
    if (!ret){
        printf("disable sensor[0] functionality\n");
    }
}
```

A Syntiant SDK API called in response to NDP interrupt, returns an index for the recognized voice keyword and processes this via a case statement in a while(1) loop later in same ndp_thread_entry() function

```c
while (1)
{
    /* Wait until NDP recognized voice keywords */
    evbits = xEventGroupWaitBits(g_ndp_event_group, EVENT_BIT_VOICE | EVENT_BIT_FLASH, pdTRUE, pdFALSE , portMAX_DELAY
    if( evbits & EVENT_BIT_VOICE )
    {
        xSemaphoreTake(g_ndp_mutex,portMAX_DELAY);
        ndp_core2_platform_tiny_poll(&notifications, 1);
        ndp_core2_platform_tiny_match_process(&ndp_nn_idx, &ndp_class_idx, &sec_val, NULL);
        printf("get ndp match with nn_id=%d, class_idx=%d %s sec-val\n\n",
               ndp_nn_idx, ndp_class_idx, (sec_val>0)?"with":"without");
        xSemaphoreGive(g_ndp_mutex);

        switch (ndp_class_idx) {
            case 0:
                /* Voice: OK-Syntiant; light Amber Led */
                current_stat.led = LED_EVENT_NONE;
                q_event = led_event_color[ndp_class_idx];
                xQueueSend(g_led_queue, (void *)&q_event, 0U );
                ble_send(ble_at_sting[V_WAKEUP], strlen(ble_at_sting[V_WAKEUP]));
                break;
            case 1:
                /* Voice: Up; light Cyan Led */
                current_stat.led = LED_EVENT_NONE;
                q_event = led_event_color[ndp_class_idx];
                xQueueSend(g_led_queue, (void *)&q_event, 0U );
                ble_send(ble_at_sting[V_UP], strlen(ble_at_sting[V_UP]));
                break;
            case 2:
                /* Voice: Down; light Magenta Led */
                current_stat.led = LED_COLOR_MAGENTA;
                current_stat.timestamp = xTaskGetTickCount();

                if (last_stat.led != LED_COLOR_MAGENTA)
                {
                    /* first receive 'Down'  keyword */
                    q_event = led_event_color[ndp_class_idx];
                    xQueueSend(g_led_queue, (void *)&q_event, 0U );
                    ble_send(ble_at_sting[V_DOWN], strlen(ble_at_sting[V_DOWN]));
                }
                else
                {
                    /*Judging the received 'Down""Down' keyword*/
                    TickType_t duration = current_stat.timestamp - last_stat.timestamp;
                    printf("duration time =%d \n", duration);
                    if ( duration < pdMS_TO_TICKS(3600UL) )
                    {
                        /* valid, send led blink envent */
                        q_event =  LED_BLINK_DOUBLE_BLUE;
                        xQueueSend(g_led_queue, (void *)&q_event, 0U );
                        /* Send 'idle' and 'advstop' to bluetooth */
                        ble_send(ble_at_sting[V_IDLE], strlen(ble_at_sting[V_IDLE]));
                        ble_send(ble_at_sting[V_STOP], strlen(ble_at_sting[V_STOP]));
                        /* clear led state */
                        current_stat.led = LED_EVENT_NONE;
                    }
                    else
                    {
                       /* invalid time */
                       q_event = led_event_color[ndp_class_idx];
                       xQueueSend(g_led_queue, (void *)&q_event, 0U );
                       ble_send(ble_at_sting[V_DOWN], strlen(ble_at_sting[V_DOWN]));
                   }
               }
               break;
           case 3:
               /* Voice: Back; light Red Led */
               current_stat.led = LED_EVENT_NONE;
               q_event = led_event_color[ndp_class_idx];
               xQueueSend(g_led_queue, (void *)&q_event, 0U );
               ble_send(ble_at_sting[V_BACK], strlen(ble_at_sting[V_BACK]));
               break;
           case 4:
               /* Voice: Next; light Green Led */
               current_stat.led = LED_EVENT_NONE;
```

I/O Board Expansion Connectors

The IO Board provides multiple expansion interfaces, these are tabled here across two pages

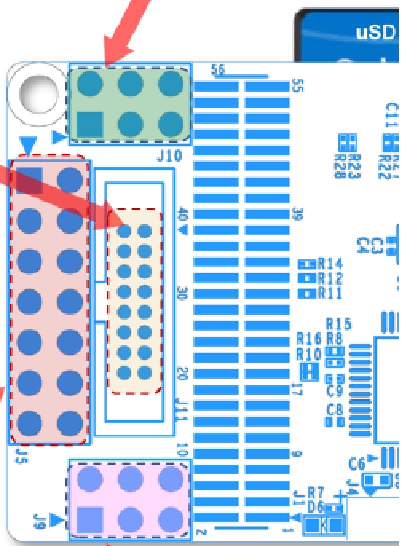
Left-Side Connectors

J10: AUD1 NDP120

NPD120 Pin	NDP120 Pin
GND	AUD1_PDAT
1V8_NDP	AUD1_PCLK0
AUD2_SDO	AUD1_PCLK1

J11: Click Shuttle Connector

#	Click Signal	NDP120 Pin / RA6M4 Pin	#	Click Signal	NDP120 Pin / RA6M4 Pin
1	AN	GPIO17_3V3	2	PWM	GPIO23_3V3
3	RST	~~GPIO18_3V3~~ P500	4	INT	~~GPIO24_3V3~~ P304
5	CS	MSSB1_3V3	6	RXD	NDP_RXD_3V3
7	SCK	MSCK_3V3	8	TXD	NDP_TXD_3V3
9	MISO	MMOSI_3V3	10	SCL	~~SCL_3V3~~ P205_SCL1_B
11	MOSI	MMISO_3V3	12	SDA	~~SDA_3V3~~ P206_SDA1_B
13	3V3	3V3_MCU	14	5V	VSYS
15	GND	GND	16	GND	GND



J5: Core Board MCU I/O

#	RA6M4 Pin	#	RA6M4 Pin
1	P301	2	3V3_MCU_1
3	P302	4	P003
5	P303	6	P304
7	P105	8	P500
9	P206_SDA1_B (*or* RXD4)	10	P205_SCL1_B (*or* TXD4)
11	P014	12	P004
13	GND	14	GND

J9: AUD0 NDP120

NDP120 Pin	NDP120 Pin
GND	AUD0_PDAT
1V8_NDP	
MSSB2_1V8	AUD0_PCLK1

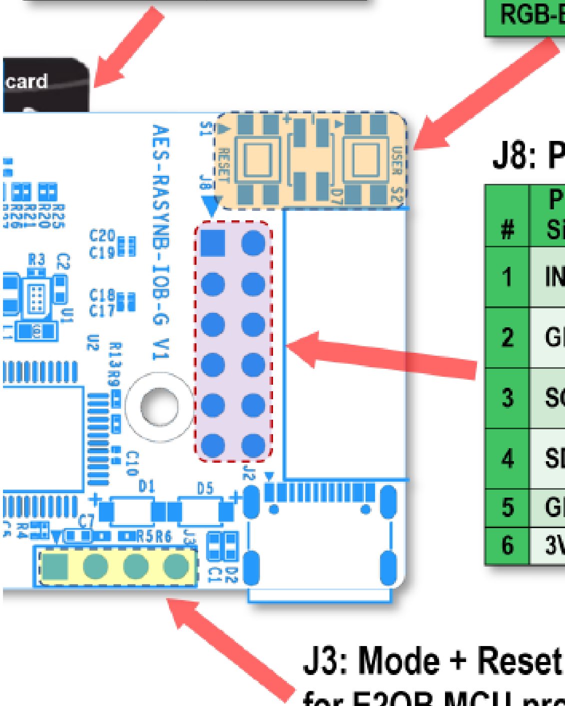
Right-Side Connectors

J7: MicroSD Card

Function	RA6M4 Pin
DAT3 (CS)	P104
CMD (DI)	P101
CLK (SCLK)	P102
DAT0 (DO)	P100

P/B Switches and RGB LED

Label	Board Function	RA6M4 Pin
RES	RESET Switch	RES
USER	USER Switch	P015
RGB-R	RGB RED LED	P001
RGB-G	RGB GRN LED	P207
RGB-B	RGB BLU LED	P111



J8: Pmod NDP120 I/O

#	Pmod Signal	NDP120 Pin / RA6M4 Pin	#	Pmod Signal	NDP120 Pin
1	INT	~~GPIO6_3V3~~ P105	7	GPIO	GPIO17_3V3
2	GPIO	~~GPIO7_3V3~~ P004	8	GPIO	GPIO18_3V3
3	SCL	~~SCL_3V3~~ P205_SCL1_B	9	GPIO	GPIO22_3V3
4	SDA	~~SDA_3V3~~ P206_SDA1_B	10	GPIO	GPIO23_3V3
5	GND	GND	11	GND	GND
6	3V3	3V3_MCU	12	3V3	3V3_MCU

J3: Mode + Reset strapping for E2OB MCU programming

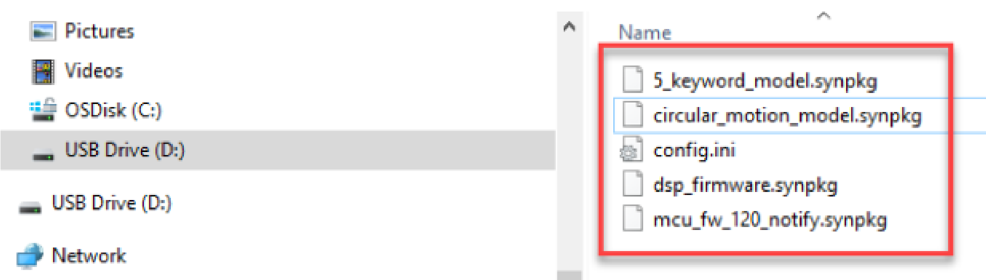
J3 Signal Name	E2OB MCU Pin
3V3_MCU_1	VCC, VREFH
E_MD	MD/FINED input
E_RES#	RESET# input
GND	GND

Appendix – Development Notes

SDcard Config.ini File Settings

The config.ini file facilitates rapid reconfiguration of how RASynBoard is to be used (without rebuild of the application). The RA6M4 application reads settings for how it should operate, from this text file on startup. Typical config.ini settings shown below are for applications using onboard microphone(s) or IMU sensors

```
[NDP Firmware]
Mode=1        # select microphone mode:  1 >single mic  2 >dual mic 3 >circle motion

[Single Mic]
MCU=mcu_fw_120_notify.synpkg
DSP=dsp_firmware.synpkg
DNN=menu_demo_512_general_newph_v100_rasyn_pdm0_ext_icm.synpkg

[Dual Mic]
MCU=mcu_fw_120_notify.synpkg
DSP=dsp_firmware_noaec_ff.synpkg
DNN=menu_demo_512_noaec_newph_v100_rasyn_icm.synpkg

[Circle Motion]
MCU=mcu_fw_120_notify.synpkg
DSP=dsp_firmware.synpkg
DNN=circular_motion_NDP120B0_icm42670.synpkg

[Led]
# set led response color for each voice command, choose from
"red","green","blue","yellow","cyan" and "magenta".
IDX0=yellow   # ok-syntiant
IDX1=cyan     # up
IDX2=magenta  # down
IDX3=red      # back
IDX4=green    # next

[Debug Print]
Port=1          # select debug port:  1 >by UART;  2 >by USB-VCOM
```

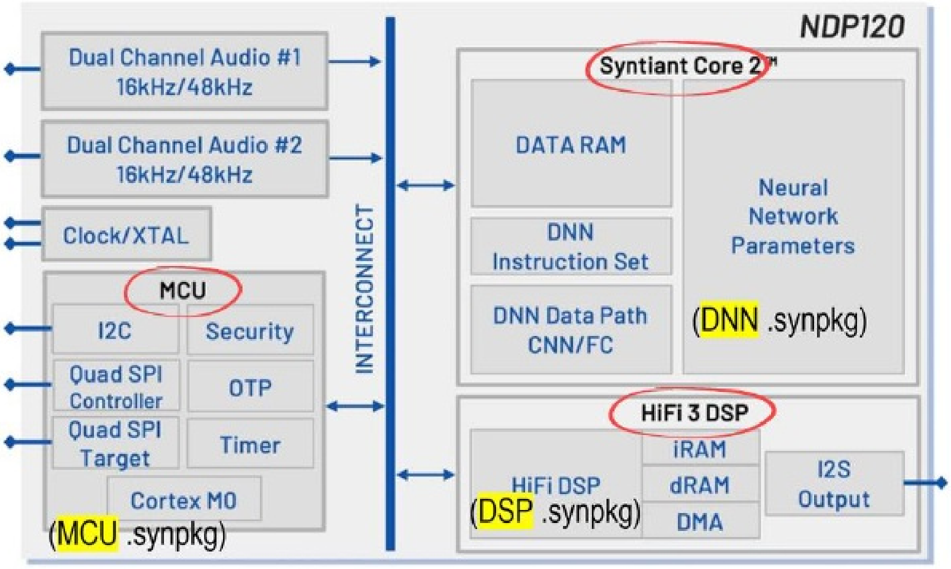


Explanation of Fields within Config.ini File

- **[NDP Firmware]**
  Defines which set of firmware binary images are to be loaded into the NDP120 on startup. Three application use case modes are presently defined, using onboard microphone(s) or IMU sensors:
  - Single Microphone mode (eg. voice commands or audio events)
  - Dual Microphone mode (eg. voice commands or audio events)
  - IMU Sensor mode (eg. motion/vibration / motor-anomalies / hand gestures)
- **[Single Mic]**
  Defines the specific three firmware binary images to load into the NDP120 for single mic operation.
- **Dual Mic]**
  Defines the specific three firmware binary images to load into the NDP120 for dual mic operation. Note: An optional add-on microphone hardware accessory is needed to use this option
- **[Circle Motion]**
  Defines the specific three firmware binary images to load into the NDP120 for IMU motion detection (Note: At

the time of RASynBoard Starter Kit release, only the circular motion gesture is detected)

- **[Led]**
  Defines for each supported inference index, the color which the RGB LED on IO board should illuminate when that specific inference gets identified by the NDP120
- **[Debug Print]**
  Defines to which UART the debug printf console output should be routed. Two options can presently be specified (see next section in the Appendix for more detail):
  - by UART
  - by USB-VCOM
- The three firmware binaries that load into the NDP120 on startup are as shown below:



NDP120 Core	Default Filenames	Comments
MCU	mcu_fw_120_notify.synpkg	Arm Cortex-M0 firmware image for NDP120 peripherals, etc management
DSP	dsp_firmware.synpkg	Tensilica HiFi 3 DSP firmware image (eg. for audio prep rocessing)
DNN	(depends on application)	Deep Neural Network firmware image (ie. the trained AI model)

Options for Implementing Debug UART Console Output

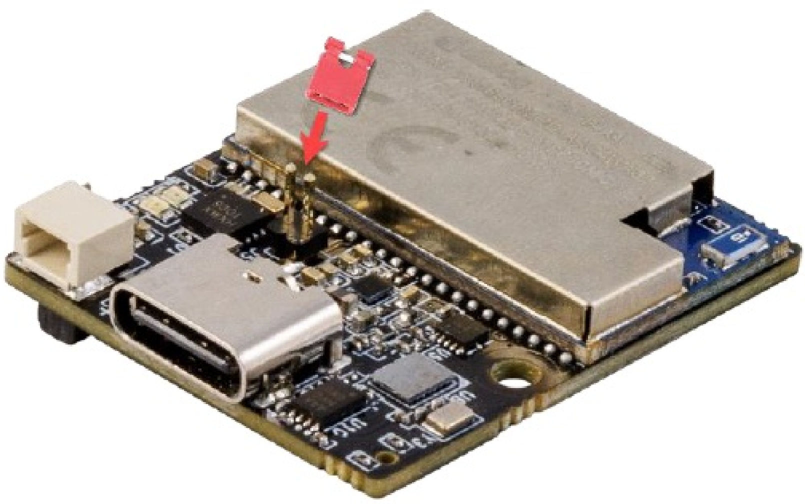
- Use of an external USB-to-Serial adapter cable (default method) A debug UART from the RA6M4 MCU is implemented in the application using SCI4 assigned to the P206 and P205 port signals. This UART however will no longer be available if P206 and P205 are needed to implement an I2C bus to the Pmod and/or Click connectors. In that case, alternative methods such as the following can be used

- Implement USB device CDC VCOM port via USB-C connector on Core board If console output is always needed when running the application, then within the application you can implement a USB VCOM port, over which the debug UART output is then transmitted
  Disadvantages of USB solution: – An additional USB connection between RASynBoard and the development PC is required – Power measurement will be more challenging: due to use of USB peripheral interface and because there will now be two power connections to the board during development
- Implement SEMIHOST for terminal output within e2 Studio IDE Simplest is to have printf trace output (ie. stdio) sent to a different UART (preferably one not utilized on the board), then use the Arm Cortex-M “semihost” feature, to output printf debug info from this UART to the terminal view within E2Studio IDE. How to implement this with the RA family is documented in the e2 studio IDE User Guide (Goto help then search for ”semihost”)
  Disadvantages of SEMIHOST solution: – Semihost is only accessible during development / while using the e2 studio IDE

Standalone Core Board Operation

It is possible to program and run RASynBoard Core Board “standalone” (without IO Board), note however:

- RA6M4 flash programming is only possible using the RFP application (with 1.27mm jumper on J5)
- SWD based debugging is not supported when Core Board is used standalone
- NDP120 firmware must load from SPI Flash (since microSD card storage not available)
- RGB LED, button switches, microSD card, Pmod and Click interfaces, etc are not available
  Note: The only time the smaller 1.27mm pitch jumper (taped to inside lid of the Starter Kit box) needs to be fitted to Core Board J5, is to set standalone (SCI boot mode) based USB-UART programming of the RA6M4 MCU, via the Core Board USB-C interface (- this not typically needed)

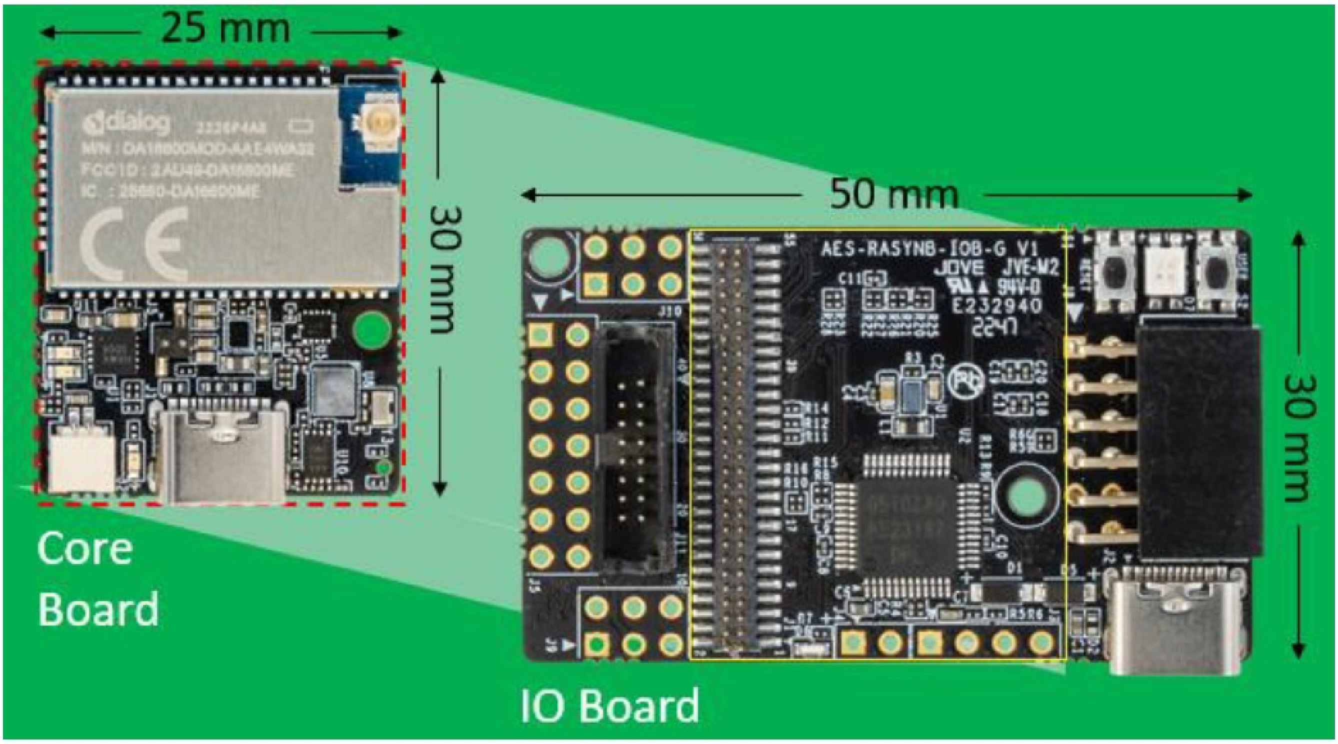


J5 Jumper	RA6M4 Boot Mode
Removed	**Self-boot/Run mode** Note: This jumper must also be removed when using IO Board
Bridge **J5** pin 1 & 2	**SCI boot mode** for MCU flash programming via Core Board's USB-C interface

How to Transfer .synpkg files to SPI Flash (to direct boot NDP120 from SPI Flash)

From the running application, hold-down the USER button-switch for at least 3 seconds to copy the .synpkg firmware files (for MCU, DSP and DNN cores) and .ini settings from the uSDcard to the SPI flash
Direct boot of the NDP120, using these files on the SPI Flash, will be initiated if the uSDcard is not inserted

Mechanical Considerations



Technical Support

Online technical support is accessible via the RASynBoard product page

Use the link provided in the sidebar at https://avnet.me/rasynboard

Documentation for key components can be found at:

- Syntiant NDP120 AI/ML page www.syntiant.com/ndp120
- Renesas RA6M4 MCU page www.renesas.com/RA6M4
- Renesas DA16600MOD page www.renesas.com/DA16600MOD

Ordering Part Numbers

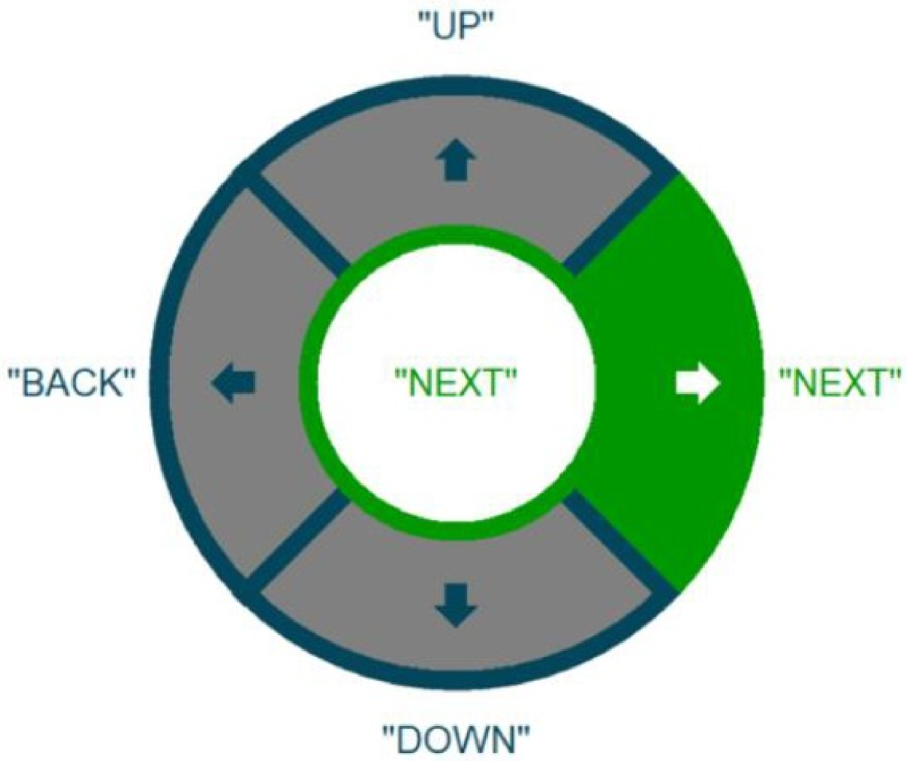
Part Number	**Description**	**Price and Availability**
AES-RASYNB-120B-SK-G	RASynBoard NDP120 Evaluation Kit	https://avnet.me/rasynboard
AES-RASYNB-120B-G	RASynBoard NDP120 Core Board only	Pricing on Request

Out-of-Box Application (rasynboard_ew_demo)
RASynBoard Starter Kit – Uses NDP120 for key-word spotting AI model, that has been trained with five command words – On recognition of a valid voice command, it confirms by illuminating onboard RGB LED for 1 sec (The RGB LED color identifies which command word has been recognized) – Sends inference results via BLE wireless to remote SBC board that generates a GUI response

Voice Command	RGB LED Color
OK SYNTANT	AMBER
NEXT	GREEN
BACK	RED
UP	CYAN
DOWN	MAGENTA
DOWN (x2 for LP mode)	BLUE (x2 rapid blinks)

Remote Single Board Computer (Raspberry Pi400)

- Receives inference results from RASynBoard via BLE wireless
- Drives the Voice-UI GUI on a large HDMI display screen



RASynBoard Power Consumption
Total power consumption for the RASynBoard 2-board assembly is monitored in two modes, using an inline USB power meter

- NDP120-only (listening for wake-word) – RA6M4 MCU is in SLEEP mode – BLE is OFF
- Wake Word recognized (all cores now up) – RA6M4 MCU is awake – BLE is ON!

Entry / Exit of Low-Power Sleep Mode
NDP120-only low-power mode is entered by repeating the DOWN voice command This low-power mode is exited by subsequent use of using any of the 5 supported voice commands

---

Documents / Resources

	AVNET RASynBoard Starter Kit Development [pdf] User Guide RASynBoard Starter Kit Development, Starter Kit Development, Kit Development, Development

References

- DA16600MOD - Ultra-Low Power Wi-Fi + BluetoothÂ® Low Energy Combo Modules for Battery Powered IoT Devices | Renesas
- RA6M4 - 200MHz ArmÂ® CortexÂ®-M33 TrustZoneÂ®, High Integration with Ethernet and OctaSPI |

Renesas

- Hardware — Syntiant
- RASynBoard | Avnet Boards
- RZBoard V2L | Avnet Boards
- CQRobot 100 Pieces Black Standard Computer Jumper Caps with Handle Pin Shunt Short Circuit 2-Pin Connector 2.0mm. for Arduino Raspberry Pi PCB PC DVD HDD Motherboard Shorting and Other Project.
- CLEEFUN USB C Cable [3ft, 5-Pack], USB A to Type C Cable Fast Charging Charger Cord Braided for iPhone 15 Pro Max/Pro/Plus, for Samsung Galaxy S24 S23 S22 S21 S20 Ultra S10, Moto, Pixel : Electronics
- IOGEAR USB 2.0 SD Portable Card Reader - Dual Slot - Rate Up To 480Mbps - USB Powered - SDXC/SDHC/SD/Micro SDXC/Micro SD/Micro SDHC/M2/MS/CF/UHS-I - Mac/Win/Chrome - GFR204SD : Electronics
- EVISWIY PL2303TA USB to TTL Serial Cable Debug Console Cable for Raspberry Pi 3 Pack : Electronics
- DA16600MOD - Ultra-Low Power Wi-Fi + BluetoothÂ® Low Energy Combo Modules for Battery Powered IoT Devices | Renesas
- eÂ² studio | Renesas
- RA6M4 - 200MHz ArmÂ® CortexÂ®-M33 TrustZoneÂ®, High Integration with Ethernet and OctaSPI | Renesas
- Renesas Flash Programmer (Programming GUI) | Renesas
- Log In | Renesas Electronics Corporation
- eÂ² studio -information for RA Family | Renesas
- Hardware — Syntiant
- User Manual