

AS019PMODEXP-POCZ

Software User Manual

Introduction

This document describes and explains the user manual for AS019PMODEXP-POCZ, which is Qt based automation system. The RZ/G2L platform is used to host the Linux OS. The graphical user interface (GUI) is designed and developed using Qt framework. The RZ/G2L platform is used as sensor data collector and to control/prevent the hazardous event in home/closed environment. The HS3001 sensor is used to collect the temperature and humidity data, which is interface over PMOD connector. The relays and camera are used to indicate the alarm/error event detection

It involves the system overview of the POC, connection set up, how to set host environment to build the BSP, flashing and application usage details.

Target Device

- RZ/G2L SMARC EVK
 - ARM CPU Dual Cortex® A55 running at 1.2Ghz clock frequency.
 - Main Memory DDR4 SDRAM: 2GBytes
 - QSPI NOR FLASH 64Mbyte
 - eMMC 64GByte
- HS3001
 - Highly accurate, fully calibrated relative humidity and temperature sensor

References

- [RZ/G2L-EVKIT - RZ/G2L Evaluation Board Kit | Renesas](#)
- [US082-HS3001EVZ - Relative Humidity Sensor Pmod™ Board \(Renesas Quick-Connect IoT\) | Renesas](#)
- [HS300x Datasheet \(renesas.com\)](#)
- <https://www.renesas.com/us/en/document/mah/rzg2l-group-rzg2lc-group-users-manual-hardware-0?r=1467981>

Acronyms

Name	Description
BSP	Board Support Package
DTB	Device Tree Blob
EVK	Evaluation Kit
eMMC	Embedded Multimedia
FAT	File Allocation Table
GPIO	General Purpose Input Output
GUI	Graphical User Interface

HDMI	High-Definition Multimedia Interface
I2C	Inter-Integrated Circuit
LCD	Liquid-Crystal Display
OS	Operating System
PC	Personal Computer
PMOD	Peripheral Module
POC	Proof Of Concept
SCIF	Serial Communication Interface with FIFO
SD	Secure Digital
SDK	Software Development Kit
SMARC	Smart Mobility ARChitecture
SoC	System on Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

Contents

1	Overview.....	5
2	Operating Environment	5
3	AS019 Details	6
3.1	Hardware Details	6
3.2	Connection Setup	8
3.3	Configuration Switch Settings	9
3.3.1	SW1 Switch Settings.....	9
3.3.2	SCIF Download Mode.....	9
3.3.3	QSPI Boot Mode	10
3.3.4	Input Power Supply Voltage Selection	10
3.3.5	PMOD1 Type-6A Selection	10
4	Software Build.....	12
4.1	Qt6 BSP Build	12
5	Flashing/Programing	14
5.1	Flashing IPL.....	14
5.2	Qt6 System Binaries	18
5.2.1	SD Card partition	18
5.2.2	SD Card Format.....	19
5.2.3	SD Card Load/Program	19
6	Boot	21
7	Qt Application	22
7.1	Qt Application Build.....	22
7.1.1	Qt6 SDK	22
7.1.2	Qt Creator.....	22
7.1.3	Open the project and build the application.	24
7.2	UI Screens.....	27
7.2.1	Home Screen.....	27
7.2.2	Settings Screen	31
7.2.3	Graphical Screen	32
7.2.4	Alarm/Alert Screen.....	33
8	Limitation & Know Issues	35

List Of Tables

Table 1. Operating Environment	5
Table 2. Mode and Storage Selection	9
Table 3. SW12 Settings - SCIF Download Mode	10
Table 4. SW12 Settings – QSPI Boot Mode	10
Table 5. SW11-4 Settings – Power Supply Selection	10
Table 6. Loaders Load & Program Address	17

List Of Figures

Figure 1. AS019 Hardware Details	6
Figure 2. SW1 Switch.....	9
Figure 3. SCIF Download Mode	9
Figure 4. PMOD1 Type-6A Selection	11
Figure 5. System Binaries	14
Figure 6. IPL Programming Setup	15
Figure 7. Serial Configuration Settings	15
Figure 8. SCIF Download Mode Screen	16
Figure 9. Flash Writer Tool Transfer.....	17
Figure 10. U-Boot Env Settings.....	21
Figure 10. Qt Creator – Qt Version.....	22
Figure 12. Qt Creator – Compiler	23
Figure 13. Qt Creator – Kits	24
Figure 14. Qt Creator – Open Project.....	25
Figure 15. Qt Creator – Build	26
Figure 16. Home Screen	28
Figure 17. AS048 Block Diagram	29
Figure 18. AS019 Block Diagram	30
Figure 19. Settings Screen.....	31
Figure 20. Graphical View Screen.....	32
Figure 21. Alert Screen – Normal Condition	33
Figure 22. Alert Screen – Alert Condition	34

1 Overview

The RZ/G2L SMARC EVK and US082-HS3001EVZ is used in smart home automation safety system AS019PMODEXP-POCZ. The PMOD adaptor board is designed to facilitate the PMOD sensor module and relay module interfacing. The relays are controlled based on error/ alarm event occurrence for safety control. The relays eventually control the motors connected to it.

The home automation application is designed and developed using Qt GUI framework. The Qt creator tool is used to implement the following.

- UI Screens
- Peripheral communication – I2C PMOD sensor
- Application and/Business logic

This document briefs the Qt6 BSP build, flashing and application usage.

2 Operating Environment

The operation of this software project has been confirmed with the following environment.

Item	Description
RZ/G2L SMARC EVK	RZ/G2L-EVKIT - RZ/G2L Evaluation Board Kit Renesas
Microprocessor	RZ/G2L MPU
BSP	Qt6 BSP
OS	Linux kernel
Build	Yocto
Boot Media	SPI Flash – Bootloaders (Kernel & FS from uSD)
Host Build PC	Ubuntu – 20.04
Host PC – Flashing	Windows 10 - Tera term

Table 1. Operating Environment

3 AS019PMODEXP-POCZ Details

This chapter briefs the hardware modules, existing POCs, HDMI display, Power supply and other accessories used in this development. PMOD adaptor is developed to interface multiple sensors and their interconnection schemas options.

3.1 Hardware Details

This section explains the details of each component and/or sub hardware modules/POC used in development of this POC, with their part number wherever applicable.

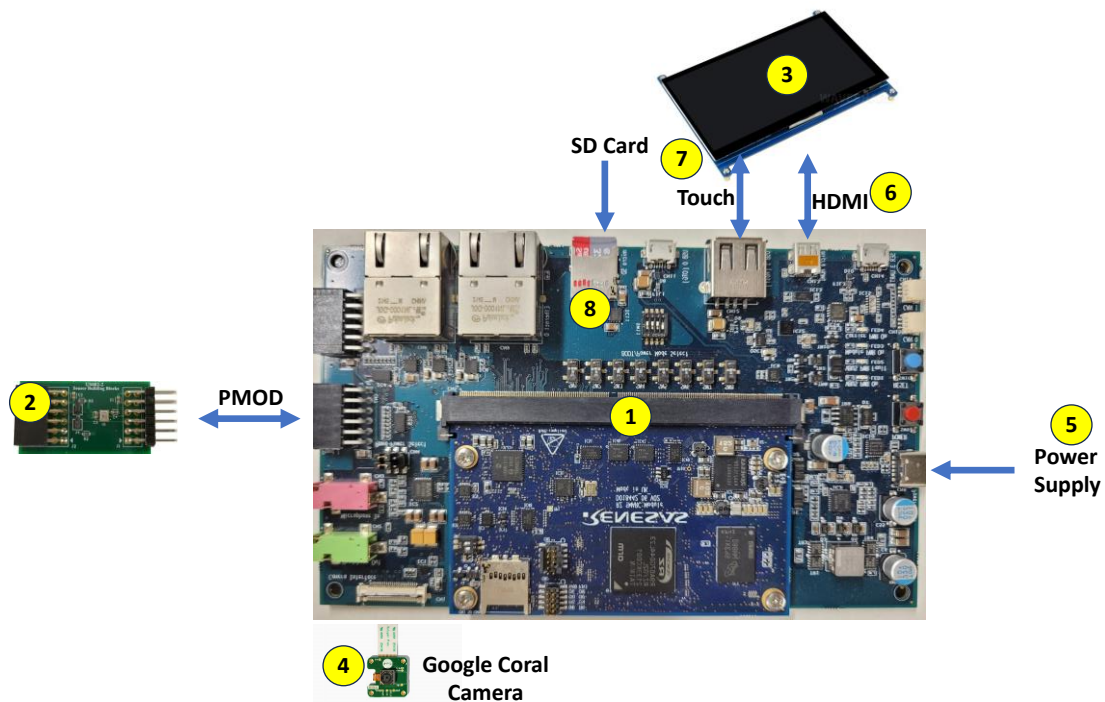


Figure 1. AS019PMODEXP-POCZ Hardware Details

1 RZ/G2L SMARC EVK: It consists of 2 part numbers RTK9744L23C01000BE and RTK97X4XXB00000BE. The RTK9744L23C01000BE is SMARC SOM module with RZ/G2L MPU (R9A07G044L23GBG), PMIC, DDR4 memory, SPI Flash memory and EMMC memory. The RTK97X4XXB00000BE SMARC carrier board embedded with camera, display, Micro SD, USB, PMOD (2A, 3A/6A) and others. For more details refer [RZ/G2L, RZ/V2L SMARC Module Board User's Manual: Hardware \(renesas.com\)](#).

2

US082-HS3001EVZ: This has HS3001, a high-performance relative humidity and temperature sensor. It is featured with a standard Pmod™ Type 6A (Extended I2C) connection for the on-board sensor to plug into any desired MCU/MPU evaluation kit.

3

6

7

7 Inch HDMI LCD display with USB capacitive touch screen having 1024x600 pixel. The item # 6 is HDMI cable and item #7 is the USB cable for touch screen.

4

Coral camera, which has 5-MegaPixel OmniVision sensor and dual lane MIPI output interface

5

USB Type-C Power Source, Default 5V~15V @3A: (min 15W requested) with Option 9V @3A: 27W

8

Micro SD Card 32GB: The kernel image and Device Tree File DTB is programmed in windows partition. The rootfs file system is programmed in Linux partition.

3.2 Connection Setup

Please refer figure 1: AS019PMODEXP-POCZ Hardware Details for connection and follow below steps to setup the system

- Connect US082-HS3001EVZ temperature and humidity sensor module to PMOD1 (CN7) connector of SMARC EVK platform.
- Connect the display port of HDMI Display to Micro HDMI port (CN13) of SMARC EVK platform. Micro HDMI convertor should be used for HDMI data. For touch, USB cable is connected between USB1 Host port (CN12) of SMARC EVK to touch port of HDMI display panel.
- Connect coral camera to MIPI CSI connector CN1 of SAMRC EVK platform
- Program micro SD card with kernel image, DTB file and rootfs file system, before inserting into the SD slot (CN10) of SMARC EVK platform
- Plug the USB type C cable to power on the system

3.3 Configuration Switch Settings

3.3.1 SW1 Switch Settings



Figure 2. SW1 Switch

/

SW1-1	DEBUGEN	SW1-2	microSD/eMMC selection
OFF	JTAG debug mode	OFF	Select eMMC on RTK9744L23C01000BE
ON	Normal operation	ON	Select microSD slot on RTK9744L23C01000BE

Table 2. Mode and Storage Selection

Please set the SW1 switch settings for Normal operation and Micro SD card/slot selection.

3.3.2 SCIF Download Mode

Change switch DP SW12 to set SCIF Download mode and QSPI Boot Mode

SCIF Download Mode(*1)		QSPI Boot (1.8V) Mode	
			
SW11-1	OFF	SW11-1	OFF
SW11-2	ON	SW11-2	OFF
SW11-3	OFF	SW11-3	OFF
SW11-4	ON	SW11-4	ON

Figure 3. SCIF Download Mode

Switch Number	Pin6	Pin5	Pin4	Pin3	Pin2	Pin1
SW12	ON	ON	OFF	OFF	OFF	OFF

Table 3. SW12 Settings - SCIF Download Mode

3.3.3 QSPI Boot Mode

Change switch DP 'SW12' switch settings to below for QSPI Boot Mode

Switch Number	Pin6	Pin5	Pin4	Pin3	Pin2	Pin1
SW12	ON	ON	OFF	ON	ON	ON

Table 4. SW12 Settings – QSPI Boot Mode

3.3.4 Input Power Supply Voltage Selection

Based on the supply voltage connected, set the SW11-4 DP switch accordingly

Please select input voltage setting as below.

SW11-4	Input voltage selection
OFF	Input 9V
ON	Input 5V

Table 5. SW11-4 Settings – Power Supply Selection

3.3.5 PMOD1 Type-6A Selection

Modify the SW3 and SW4 switch settings on RZ/G2L SMARC carrier board to make PMOD1 as Type-6A only for I2C Clock and I2C Data.

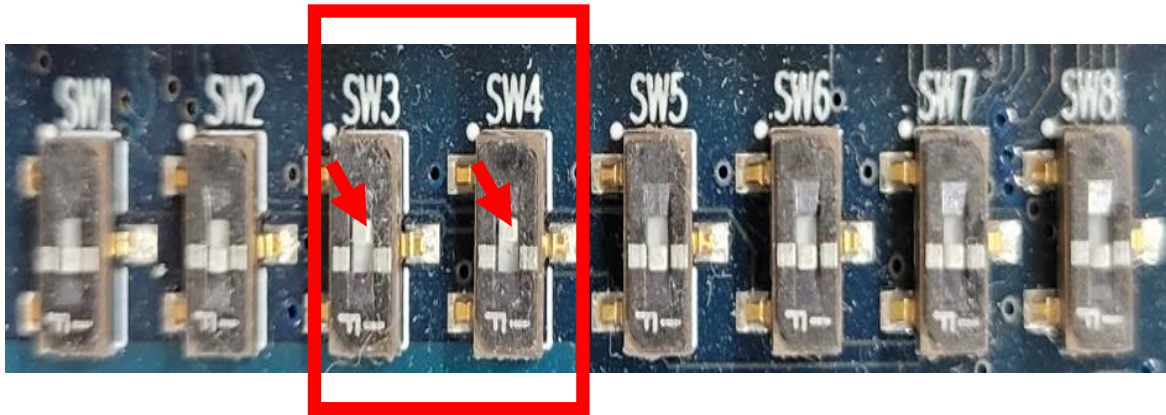


Figure 4. PMOD1 Type-6A Selection

4 Software Build

This chapter describes the details and procedure of building the Qt6 BSP for RZ/G2LSMARC EVK platform. The host build PC requirement is captured in chapter 2: Operating Environment. This chapter also describes the build host PC packages dependencies, fetch the meta layers for Qt6, set up the build environment and configure the build for the RZ/G2L platform.

4.1 Qt6 BSP Build

Download the **porting-qt-6.2.0-to-rzg2l-master.zip** from POC release folder. The **porting-qt-6.2.0-to-rzg2l-master.zip** includes following

- rzg_bsp_v300.tar.gz: Yocto recipe packages,
- RTK0EF0045Z13001ZJ-v1.0_JP.zip: for enabling GPU,
- RTK0EF0045Z15001ZJ-v0.56_JP.zip: for enabling video codec.

Unzip the **porting-qt-6.2.0-to-rzg2l-master.zip** folder and follow the below steps to build Qt6 system binaries/images.

- Update the system with all host dependent and required packages
`$ sudo apt-get update`
`$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat cpio python python3 python3-pip python3-pexpect xz-utils debianutils iputils-ping libssl1.2-dev xterm p7zip-full python-is-python3`
- Configure the git for user name and eMail
`$ git config --global user.email "you@example.com"`
`$ git config --global user.name "Your Name"`
- Download the below meta sources from the repos using git command
 - meta-qt6
`$ git clone -b v6.2.0 git://code.qt.io/yocto/meta-qt6.git`
 - meta-python2
`$ git clone -b dunfell https://git.openembedded.org/meta-python2`
 - meta-boot2qt
`$ git clone -b v6.2.0 git://code.qt.io/yocto/meta-boot2qt.git`
`$ git am patch/0001-meta-boot2qt-patch.patch`
 - meta-mingw
`$ git clone -b yocto-3.3.4 git://code.qt.io/yocto/meta-mingw.git`
`$ git am patch/0001-meta-mingw-yocto-3.3.4.patch`
- Set up the build environment
`$ source poky/oe-init-build-env`
- Update the configuration file for RZ/G2L SMARC EVK machine
`$ cp ../meta-renesas/docs/template/conf/smarc-rzg2l/*.conf ./conf/`
- Modify build/conf/bblayers.conf file to use meta-qt6 layer.
`- `QT_LAYER = "${@os.path.isdir("${TOPDIR}/../meta-qt6")}"``

```
- `${@}${TOPDIR}/../meta-qt6' if '${QT_LAYER}' == 'True' else "}"`  
g. Now build the Qt image  
$ export DISTRO=poky  
$ bitbake core-image-qt
```

The build artifacts are available in the *build/tmp/deploy/images/smarc-rzg2l* folder location. For future reference in this document this folder location path is mentioned as {BUILD_DIR_OUTPUT}

- h. To build and populate software development kit, use below command
\$ bitbake core-image-qt -c populate_sdk

5 Flashing/Programing

The flashing procedure can be initiated with system binaries generated from chapter 4: 'Software Build' or use the binary folder (Pre-Built Images) available in POC release folder. An example of system binary folder snapshot is given below for the reference.

```

drwxrwxr-x 3 nadhusudhan nadhusudhan 4.0K Oct 28 2022 .
lrwxrwxrwx 2 nadhusudhan nadhusudhan 54 Oct 31 2022 core-image-qt-smarc-rzg2l-testdata.json -> core-image-qt-smarc-rzg2l-20221031120812.testdata.json
-rw-r--r-- 2 nadhusudhan nadhusudhan 331K Oct 31 2022 core-image-qt-smarc-rzg2l-20221031120812.testdata.json
lrwxrwxrwx 2 nadhusudhan nadhusudhan 56 Oct 31 2022 core-image-qt-smarc-rzg2l.manifest -> core-image-qt-smarc-rzg2l-20221031120812.rootfs.manifest
-rw-r--r-- 2 nadhusudhan nadhusudhan 51K Oct 31 2022 core-image-qt-smarc-rzg2l-20221031120812.rootfs.manifest
-rw-r--r-- 2 nadhusudhan nadhusudhan 7.6K Oct 31 2022 core-image-qt-env
-rw-r--r-- 2 nadhusudhan nadhusudhan 2.2G Oct 31 2022 core-image-qt-smarc-rzg2l-20221031120812.rootfs.ext4
lrwxrwxrwx 2 nadhusudhan nadhusudhan 52 Oct 31 2022 core-image-qt-smarc-rzg2l.ext4 -> core-image-qt-smarc-rzg2l-20221031120812.rootfs.ext4
-rw-r--r-- 2 nadhusudhan nadhusudhan 418M Oct 31 2022 core-image-qt-smarc-rzg2l-20221031120812.rootfs.tar.bz2
lrwxrwxrwx 2 nadhusudhan nadhusudhan 55 Oct 31 2022 core-image-qt-smarc-rzg2l.tar.bz2 -> core-image-qt-smarc-rzg2l-20221031120812.rootfs.tar.bz2
-rw-r--r-- 2 nadhusudhan nadhusudhan 272M Oct 31 2022 core-image-qt-smarc-rzg2l-20221031120812.rootfs.wic.xz
lrwxrwxrwx 2 nadhusudhan nadhusudhan 54 Oct 31 2022 core-image-qt-smarc-rzg2l.wic.xz -> core-image-qt-smarc-rzg2l-20221031120812.rootfs.wic.xz
lrwxrwxrwx 2 nadhusudhan nadhusudhan 75 Nov 2 2022 image-smarc-rzg2l.bin -> image--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 17M Nov 2 2022 image--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.bin
lrwxrwxrwx 2 nadhusudhan nadhusudhan 75 Nov 2 2022 image -> image--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 216K Nov 2 2022 modules--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.tgz
lrwxrwxrwx 2 nadhusudhan nadhusudhan 77 Nov 2 2022 modules-smarc-rzg2l.tgz -> modules--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.tgz
lrwxrwxrwx 2 nadhusudhan nadhusudhan 87 Nov 2 2022 r9a07g044l2-smarc-smarc-rzg2l.dtb -> r9a07g044l2-smarc--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.dtb
lrwxrwxrwx 2 nadhusudhan nadhusudhan 87 Nov 2 2022 r9a07g044l2-smarc.dtb -> r9a07g044l2-smarc--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.dtb
-rw-r--r-- 2 nadhusudhan nadhusudhan 37K Nov 2 2022 r9a07g044l2-smarc--5.10.83-clp1+gitAUTOINC+e52825e45b-r1-smarc-rzg2l-20221102065625.dtb
-rw-r--r-- 2 nadhusudhan nadhusudhan 678K Nov 2 2022 u-boot-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.bin
lrwxrwxrwx 2 nadhusudhan nadhusudhan 56 Nov 2 2022 u-boot-smarc-rzg2l.bin-smarc-rzg2l -> u-boot-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.bin
lrwxrwxrwx 2 nadhusudhan nadhusudhan 56 Nov 2 2022 u-boot-smarc-rzg2l.bin -> u-boot-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.bin
lrwxrwxrwx 2 nadhusudhan nadhusudhan 56 Nov 2 2022 u-boot.bin-smarc-rzg2l -> u-boot-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.bin
lrwxrwxrwx 2 nadhusudhan nadhusudhan 56 Nov 2 2022 u-boot.bin -> u-boot-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 2.0M Nov 2 2022 u-boot-elf-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.srec
lrwxrwxrwx 2 nadhusudhan nadhusudhan 61 Nov 2 2022 u-boot-elf-smarc-rzg2l.srec -> u-boot-elf-smarc-rzg2l-v2021.10+gitAUTOINC+c08f788c27-r0.srec
-rw-r--r-- 2 nadhusudhan nadhusudhan 75K Nov 2 2022 bl2-smarc-rzg2l.elf
-rw-r--r-- 2 nadhusudhan nadhusudhan 39K Nov 2 2022 bl2-smarc-rzg2l.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 122K Nov 2 2022 bl31-smarc-rzg2l.elf
-rw-r--r-- 2 nadhusudhan nadhusudhan 25K Nov 2 2022 bl31-smarc-rzg2l.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 39K Nov 2 2022 bl2-smarc-rzg2l_pmtc.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 25K Nov 2 2022 bl31-smarc-rzg2l_pmtc.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 268K Nov 2 2022 Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
-rw-r--r-- 2 nadhusudhan nadhusudhan 268K Nov 2 2022 Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB.mot
-rw-r--r-- 2 nadhusudhan nadhusudhan 116K Nov 2 2022 bl2_bp-smarc-rzg2l.srec
-rw-r--r-- 2 nadhusudhan nadhusudhan 39K Nov 2 2022 bl2_bp-smarc-rzg2l.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 702K Nov 2 2022 fip-smarc-rzg2l.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 2.1M Nov 2 2022 fip-smarc-rzg2l.srec
-rw-r--r-- 2 nadhusudhan nadhusudhan 702K Nov 2 2022 fip-smarc-rzg2l_pmtc.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 116K Nov 2 2022 bl2_bp-smarc-rzg2l_pmtc.srec
-rw-r--r-- 2 nadhusudhan nadhusudhan 39K Nov 2 2022 bl2_bp-smarc-rzg2l_pmtc.bin
-rw-r--r-- 2 nadhusudhan nadhusudhan 2.1M Nov 2 2022 fip-smarc-rzg2l_pmtc.srec

```

Figure 5. System Binaries

Flashing the system binaries with RZ/G2L SMARC EVK involves following two steps

- IPL
- SD Card image.

The programming of SD Card Image includes following

- Windows Partition: Kernel Image and DTB
- Linux Partition: Filesystem (Rootfs)

5.1 Flashing IPL

- Set the SW12 DP switch to put the RZ/G2L MPU in serial download mode. Please refer the section 3.2.1: SCIF Download Mode for SW12 switch settings
- Connect the debug serial port of RZ/G2L SMARC EVK to windows host PC, running tera term terminal application.

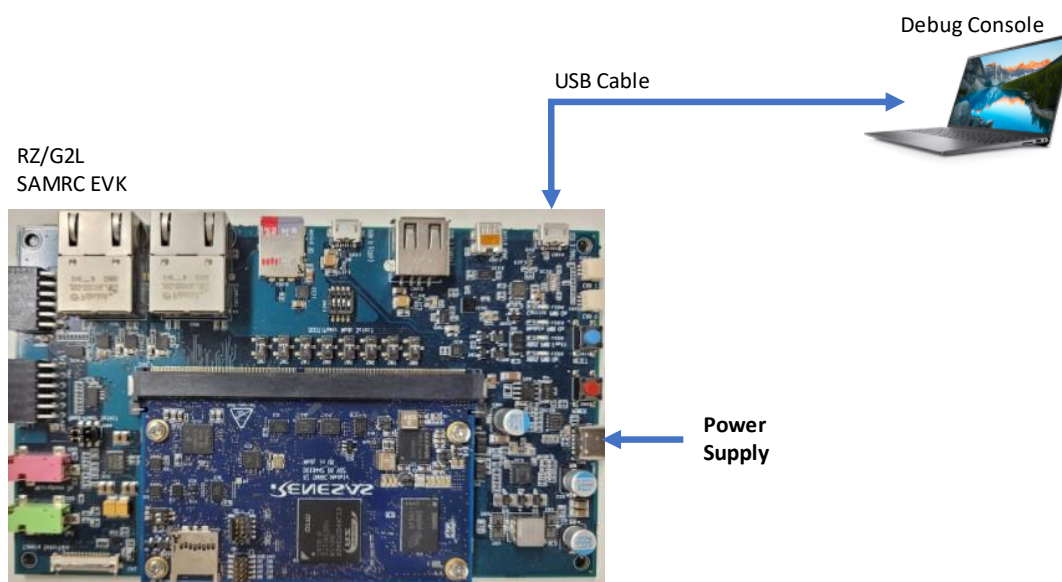


Figure 6. IPL Programming Setup

- Open the Tera term application and configure the serial settings as below

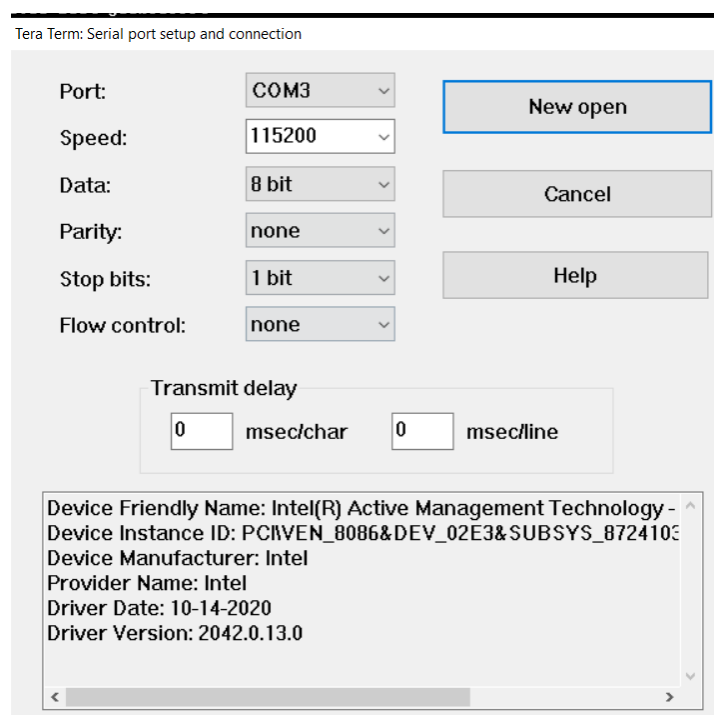


Figure 7. Serial Configuration Settings

- Power on the system with USB Type-C power adapter. Press “POWER” button followed “RESET” button. The following screen will be displayed on serial console terminal confirming that MPU has gone to serial download mode

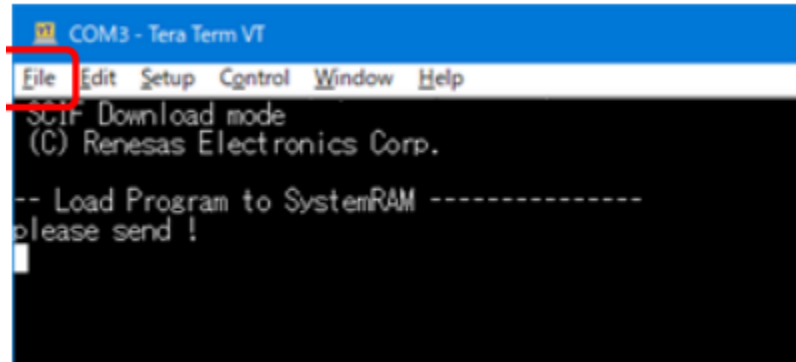
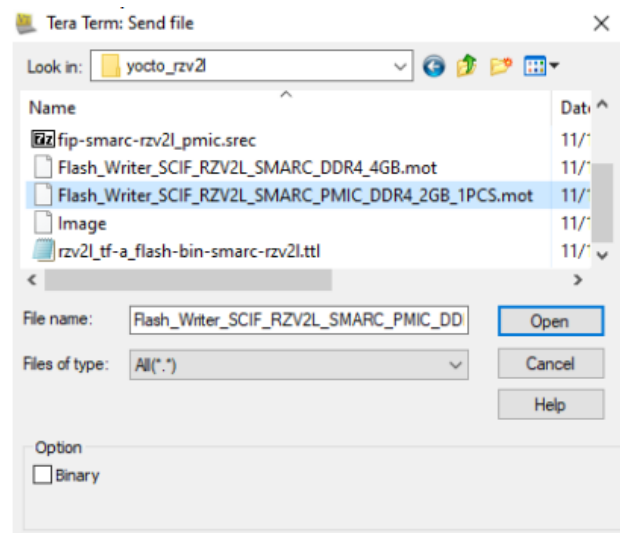
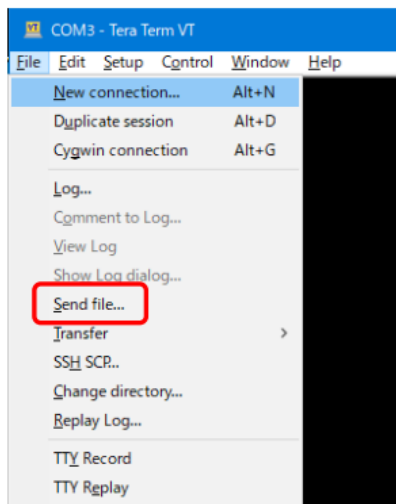


Figure 8. SCIF Download Mode Screen

- Transfer the Flash writer tool as shown below.

(File --> Send file --> Flash_Writer_SCIF_RZV2L_SMARC_PMIC_DDR4_2GB_1PCS.mot --> Open)



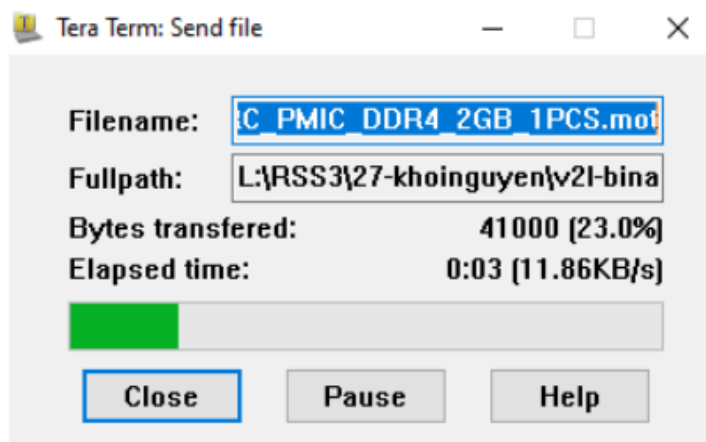


Figure 9. Flash Writer Tool Transfer

- Program the Bootloaders through Flash Writer Tool to SPI Flash. Use ***XL S2*** command to download the bootloaders to respective Flash address and RAM address as shown below

File name	Address to load to RAM	Address to save to ROM
bl2_bp-smarc-rzg2l_pmic.srec	11E00	00000
fip-smarc-rzg2l_pmic.srec	00000	1D200

Table 6. Loaders Load & Program Address

5.2 Qt6 System Binaries

5.2.1 SD Card partition

Create two partitions on Micro SD card. Use **fdisk** command on Linux/Ubuntu OS environment for partitioning

- First Partition: FAT32
- Second Partition : Linux

An instance of creating partition log in Micro SD card is shown below for the reference

```
$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): o

Created a new DOS disklabel with disk identifier 0x6b6aac6e.

Command (m for help): n
Partition type
   p primary (0 primary, 0 extended, 4 free)
   e extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-62333951, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62333951, default
62333951): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: y

The signature will be removed by a write command.

Command (m for help): n
Partition type
   p primary (1 primary, 0 extended, 3 free)
   e extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): (Push the enter key)
First sector (1026048-62333951, default 1026048): (Push the enter key)
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1026048-62333951,
default 62333951): (Push the enter key)

Created a new partition 2 of type 'Linux' and of size 29.2 GiB.

Command (m for help): p
Disk /dev/sdb: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Disk model: Transcend
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```

Disklabel type: dos
Disk identifier: 0x6b6aac6e

Device Boot Start End Sectors Size Id Type
/dev/sdb1 2048 1026047 1024000 500M 83 Linux
/dev/sdb2 1026048 62333951 61307904 29.2G 83 Linux

Filesystem/RAID signature on partition 1 will be wiped.

Command (m for help): t
Partition number (1,2, default 2): 1
Hex code (type L to list all codes): b
Changed type of partition 'Linux' to 'W95 FAT32'.
Command (m for help): w
The partition table has been altered.
Syncing disks.

```

5.2.2 SD Card Format

- Format the first partition with VFAT filesystem format. The **mkfs.vfat** command can be used for this formatting. An instance of using this command is given below for the reference

```
$ sudo mkfs.vfat -v -c -F 32 -n RZ_WIN /dev/sdb1
```

- Format the second partition with ext4 filesystem format. The **mkfs.ext4** command can be used for this formatting. An instance of using this command is given below for the reference

```
$ sudo mkfs.ext4 -L RZ_EXT /dev/sdb2
```

5.2.3 SD Card Load/Program

- Load or Program the kernel Image file and Device Tree Blob DTB file to window partition. An instance of copying the file is given below for the reference

```

$cd ${BUILD_DIR_OUTPUT}
$ cp -arL Image r9a07g044l2-smarc.dtb /media/user/RZ_WIN/
$ sync

```

- Load or Program the kernel Image file and Device Tree Blob DTB file to window partition. An instance of copying the file is given below for the reference

```

$cd /media/user/RZ_EXT
$ sudo tar -xvzf ${BUILD_DIR_OUTPUT}/core-image-qt-smarc-rzg2l.tar.bz2
$ sync

```

- Copy the Qt Application (**SHD_03**) in /home/root folder location. Please refer section 7.1: Qt Application, for building the Qt application.
- Service for auto launch on power up (Need super user - permission)
 - Create a file **home_auto.service** in **/etc/systemd/system/** folder location of SD card with below file content. (Note: Change file attributes **chmod a+x home_auto.service**)

*[Unit]**Description=Service to launch Home Automation QT Application**After=weston@root.service multi-user.target**[Service]**WorkingDirectory=/home/root/**Environment=QT_WAYLAND_DISABLE_WINDOWDECORATION=1**Environment=QT_GSTREAMER_CAMERABIN_VIDEOSRC="rzg2l_csi2"**Environment=DISPLAY=":0"**Environment=QT_GSTREAMER_CAMERABIN_VIDEOSRC_DEVICE="/dev/video0"**Environment=QT_QPA_PLATFORM=wayland**ExecStart=/home/root/launch_qt.sh**Restart=always**RestartSec=3**[Install]**WantedBy=multi-user.target*

- Create **launch_qt.sh** script file **/home/root/** folder location. (Note: Change file attributes **chmod a+x launch_qt.sh**)

```
#!/bin/bash
QT_WAYLAND_DISABLE_WINDOWDECORATION=1
media-ctl -d /dev/media0 -r
media-ctl -d /dev/media0 -l "'rzg2l_csi2 10830400.csi2':1 -> 'CRU output':0 [1]"
media-ctl -d /dev/media0 -V "'rzg2l_csi2 10830400.csi2':1 [fmt:UYVY8_2X8/1920x1080 field:none]"
media-ctl -d /dev/media0 -V "'ov5645 0-003c':0 [fmt:UYVY8_2X8/1920x1080 field:none]"
export QT_GSTREAMER_CAMERABIN_VIDEOSRC="rzg2l_csi2"
export DISPLAY=":0"
export QT_GSTREAMER_CAMERABIN_VIDEOSRC_DEVICE="/dev/video0"

export XDG_RUNTIME_DIR=/run/user/0
export QT_QPA_PLATFORM=wayland
```

./SHD_03

- Remove weston tool bar in the main panel
Add following highlighted lines in **weston.ini** file located in **/etc/xdg/weston/** folder location of SD card

```
[core]
idle-time=0
require-input=false
repaint-window=17
```

*[shell]**panel-position=none*

- Un-mount the SD card using below command

```
$ sync
$ umount /media/user/RZ_*
```

- To enable the home_auto.service service to run on every power on/reset and launch QT Application, use below command. (Execute the this command only once on first time execution. For subsequent power on cycle this is not required)

```
$systemctl enable home_auto
```

6 Boot

- Insert Micro SD Card with system binaries loaded to SD slot in carrier board. (Please refer chapter 5, for Flashing/Programming)
- Connect the accessories to RZ/G2L SMARC EVK. Accessories comprise of HDMI display, Touch, Camera, Temperature Sensor and Power Supply. Please refer section 3.2, for connection Setup.
- Configure the switch setting to QSPI Boot Mode. Please refer section 3.3.3 for details
- Please refer section 3.3.5 for PMOD1 Type-6A selection
- Please refer to Figure 7: Serial port configuration settings of Tera term application
- Power on the system using USB Type-3 adaptor.
- Please interrupt the booting process in bootloader to U-Boot prompt and avoid further booting process. Please make sure the environmental variable is set as follows to load from micro SD.

```
=> setenv sd_boot1 'mmc dev 1 ; fatload mmc 1:1 0x48080000 Image ; fatload mmc 1:1  
0x48000000 /Image-r9a07g044l2-smarc.dtb'  
  
=> setenv sd_boot2 'setenv bootargs 'root=/dev/mmcblk1p2 rootwait' ; booti 0x48080000  
- 0x48000000'  
  
=> setenv bootcmd 'run sd_boot1 sd_boot2'  
  
=> saveenv  
  
Saving Environment to MMC... Writing to MMC(0)....OK
```

Figure 10. U-Boot Env Settings

- Please make sure the MMC device is set to Micro SD and not eMMC. Also kernel image and DTB file names are intact with that of SD Card storage device
- Power On/Off or Reset the system to execute Qt home automation application automatically.

7 Qt Application

The Qt application is developed using Qt creator tool. The application is designed using Qt widgets. This chapter describes the UI screens, configuration settings, graphical plots, peripheral sensor data collection and application logic implemented. The Qt application is auto launched on system power-on. The method and procedure for auto launch is explained in chapter 6, Boot.

7.1 Qt Application Build

7.1.1 Qt6 SDK

- a. Extract/Install the Software Development Kit -SDK to the preferred location

\$./poky-glibc-x86_64-core-image-qt-aarch64-smarc-rzg2l-toolchain-3.1.14.sh

For instance, **/opt/poky/latestqt6** can be considered

7.1.2 Qt Creator

Setup and configure Qt Creator to build for RZ/G2L platform

- a. Tools -> Kits -> Qt Versions -> Add

- Add Qt included in the SDK

(e.g. **/opt/poky/latestqt6/sysroots/x86_64-pokysdk-linux/usr/bin/qmake**)

An instance is shown in below figure for the reference

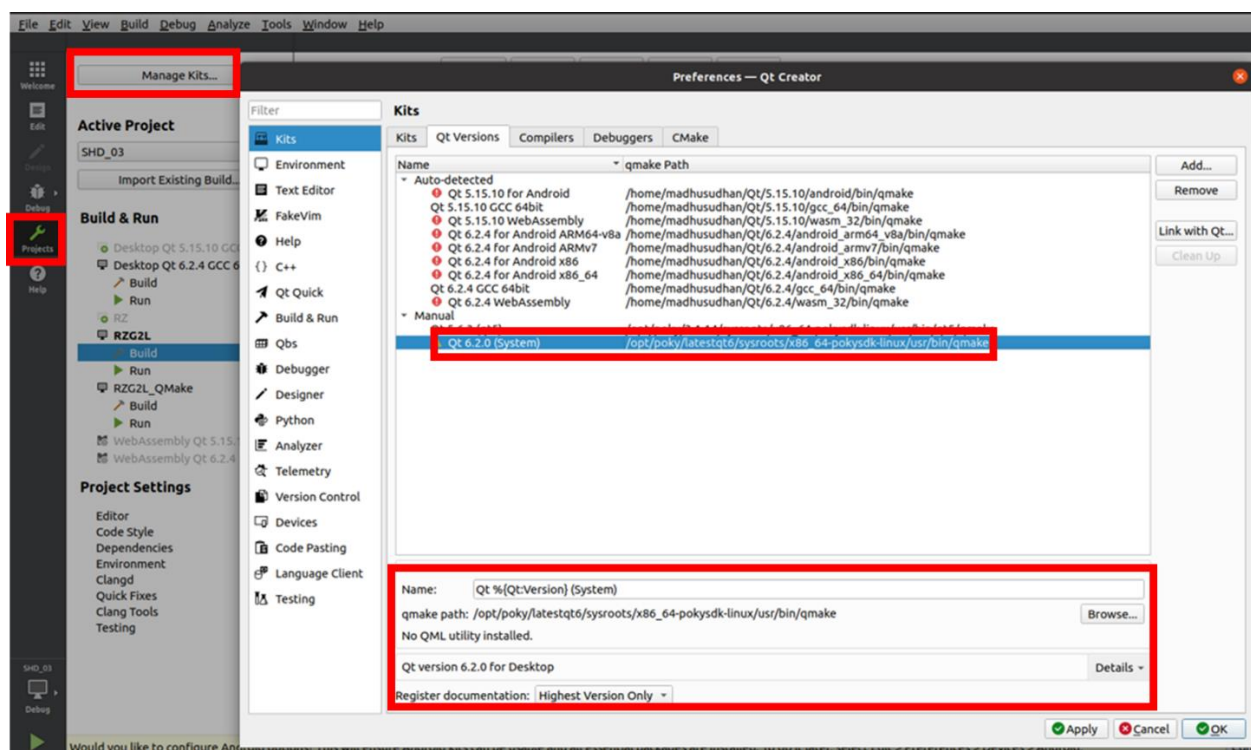


Figure 11. Qt Creator – Qt Version

b. Tools -> Kits -> Compilers -> Add

- Add gcc & g++ in the SDK
 - `**gcc**`: `/pot/poky/latestqt6/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-gcc`
 - `**g++**`: `/pot/poky/latestqt6/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-g++`

An instance is shown in below figure for the reference

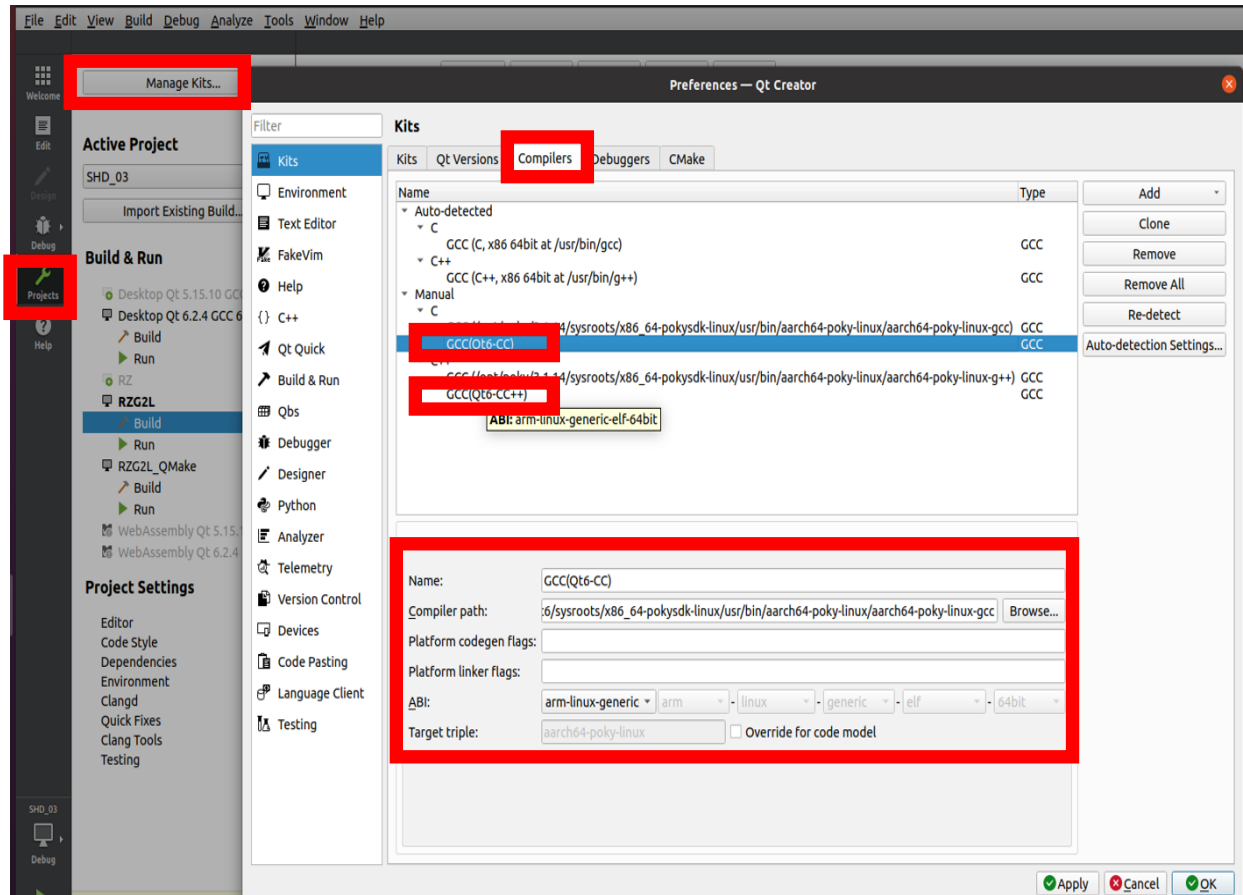


Figure 12. Qt Creator – Compiler

c. Tools -> Kits -> Add

- Device type: Remote Linux Device
- Sysroot: `/opt/poky/latestqt6/sysroots`
- Compiler: The added compilers in step b
- Qt version: The added Qt in step b

An instance is shown in below figure for the reference

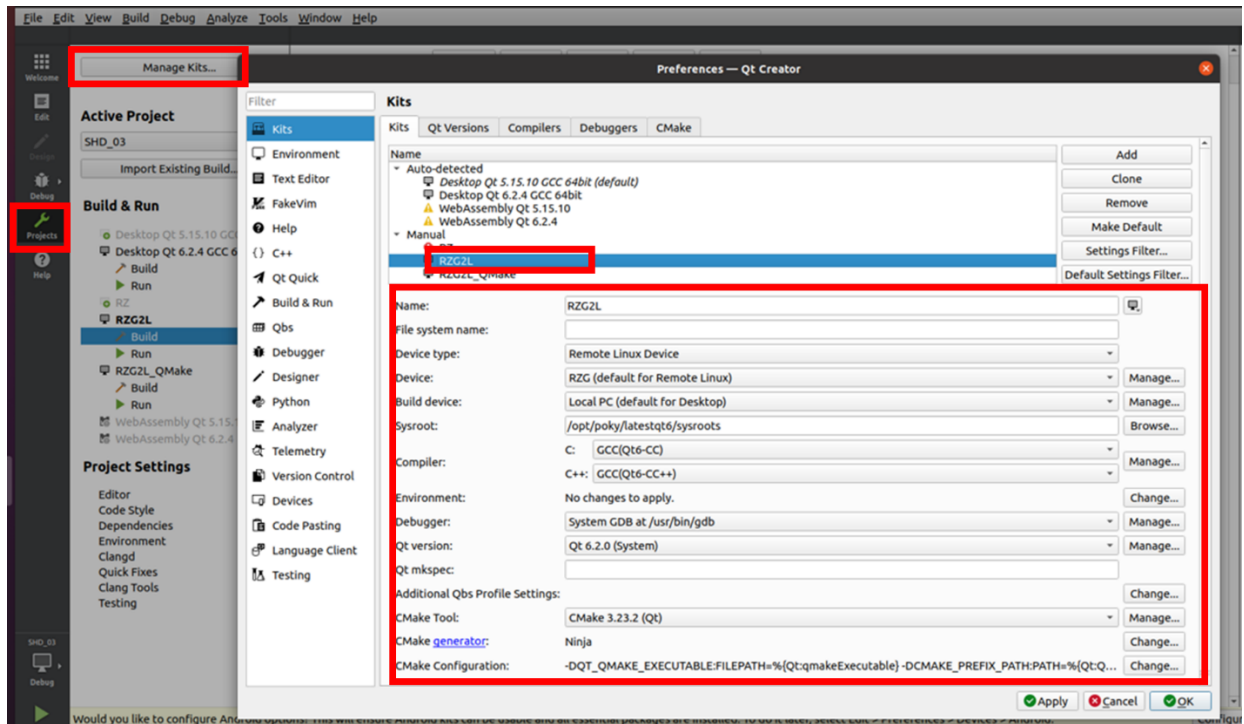


Figure 13. Qt Creator – Kits

7.1.3 Open the project and build the application.

Configure the Qt creator as described in section 7.1.2. Open the project File -> Open File or Project (^O). An instance is shown below figure for the reference

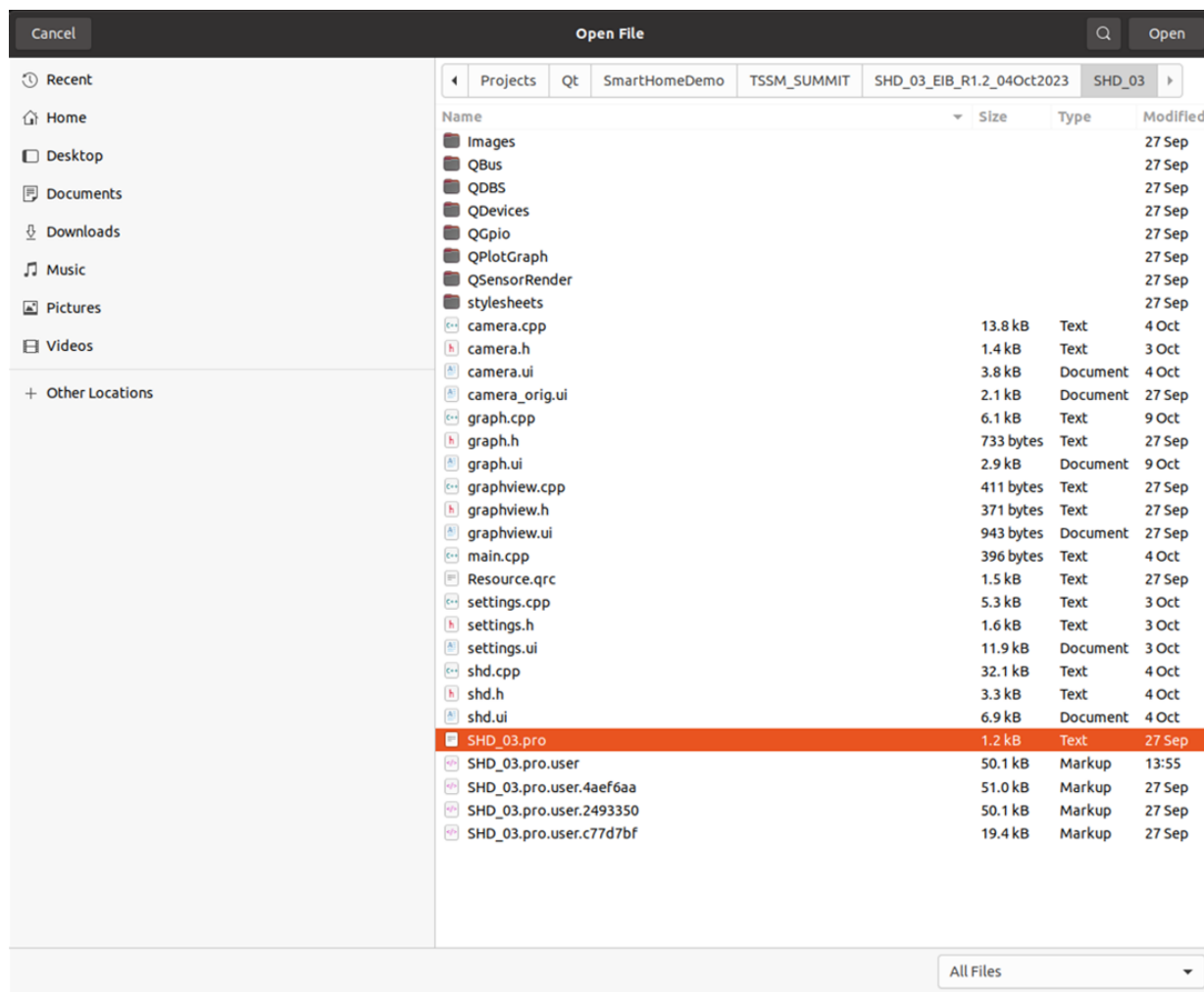


Figure 14. Qt Creator – Open Project

Compile the project by clicking the build button as shown in the below figure. Copy the Qt application in SD Card as explained in the section 5.2.3, SD Card Load/Program.

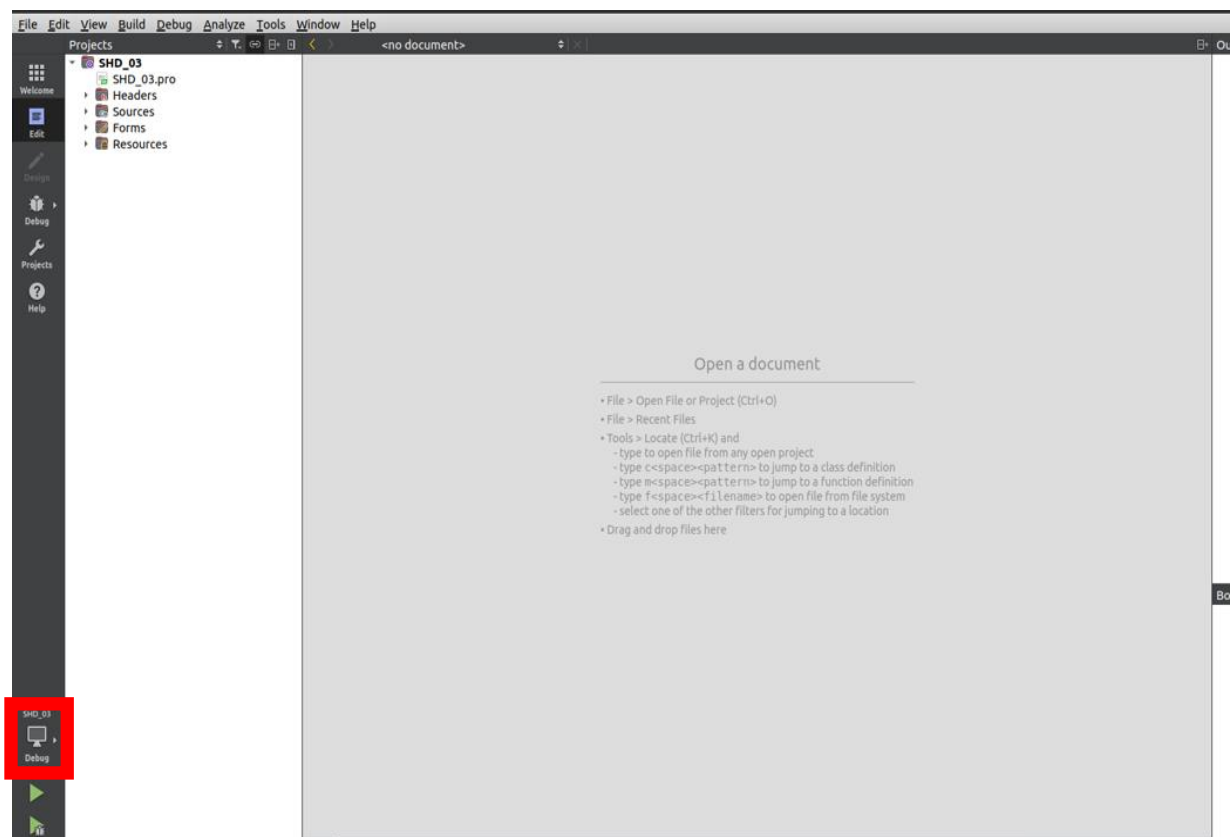


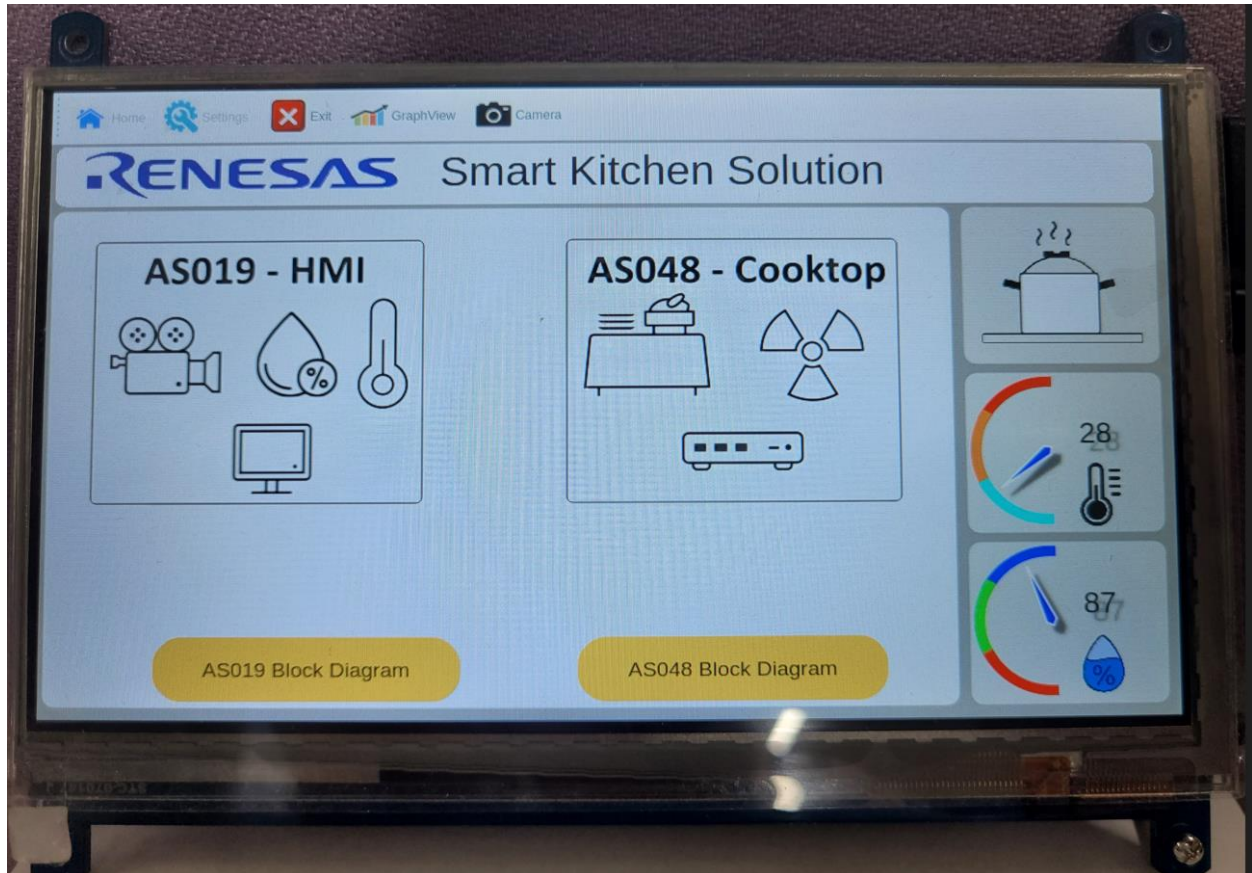
Figure 15. Qt Creator – Build

7.2 UI Screens

This section explains the various GUI screens in the Qt application and its usage

7.2.1 Home Screen

When Qt application is loaded and executed, when system is powered on, below screen appears. This is the home screen.



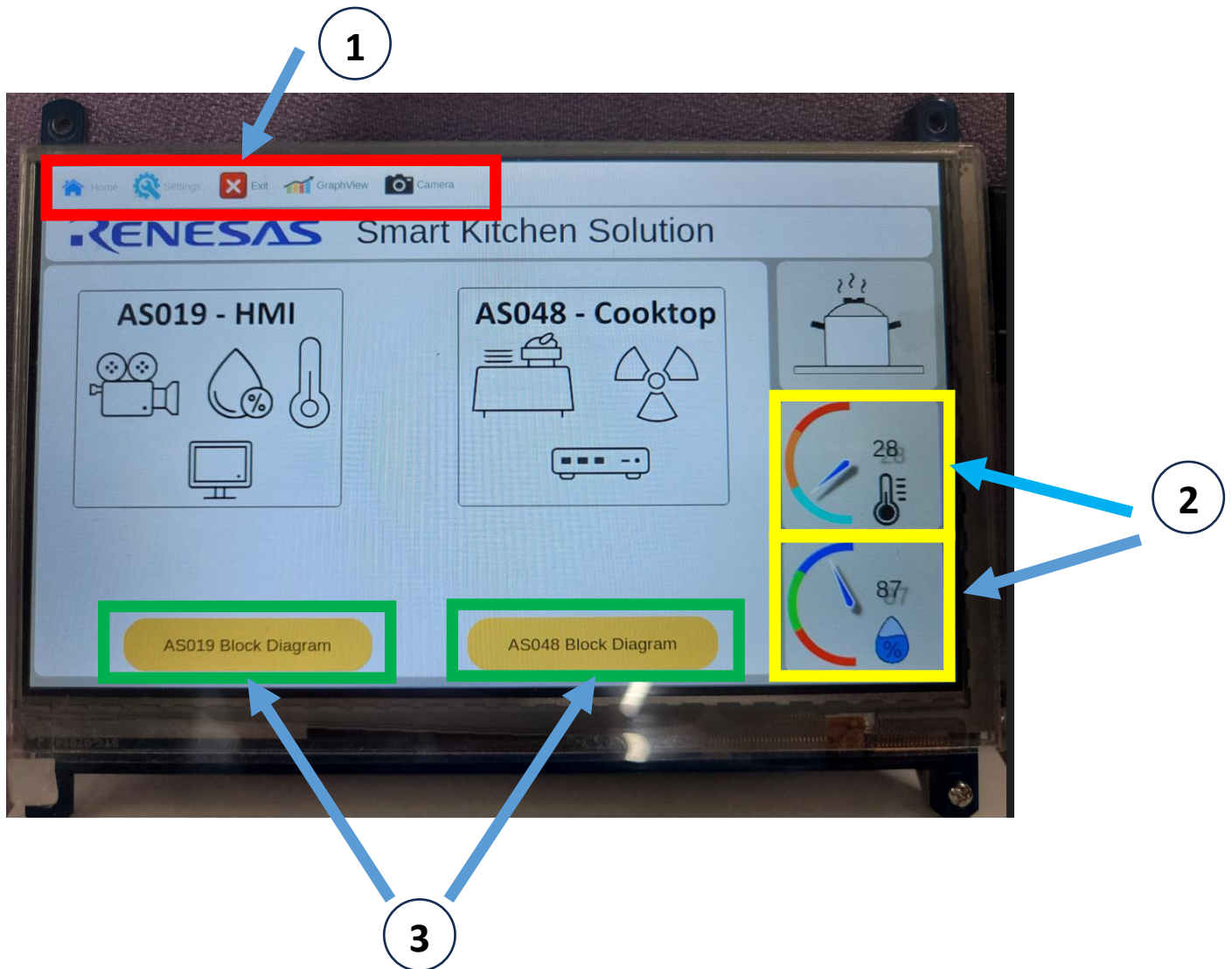


Figure 16. Home Screen

1. The menu tool bar is in the top left corner highlighted in red color.
 - a. Clicking home icon will navigate to home screen
 - b. Clicking setting icon will navigate to setting screen, please refer section 7.1.2
 - c. Clicking graphical icon will navigate to graphical screen, please refer section 7.1.3
 - d. Clicking camera icon will navigate to camera screen, please refer section 7.1.4
 - e. Clicking exit icon will close Qt application.
2. The live sensor data appears in bottom left corner of the home screen, highlighted in yellow color.

3. Navigation to AS048INDCKTP-D-POCZ and AS019PMODEXP-POCZ system block diagram is highlighted in green color at the bottom of the home screen.
 - a. Clicking “**AS048 Block Diagram**” button will navigate to AS048INDCKTP-D-POCZ system diagram, as below

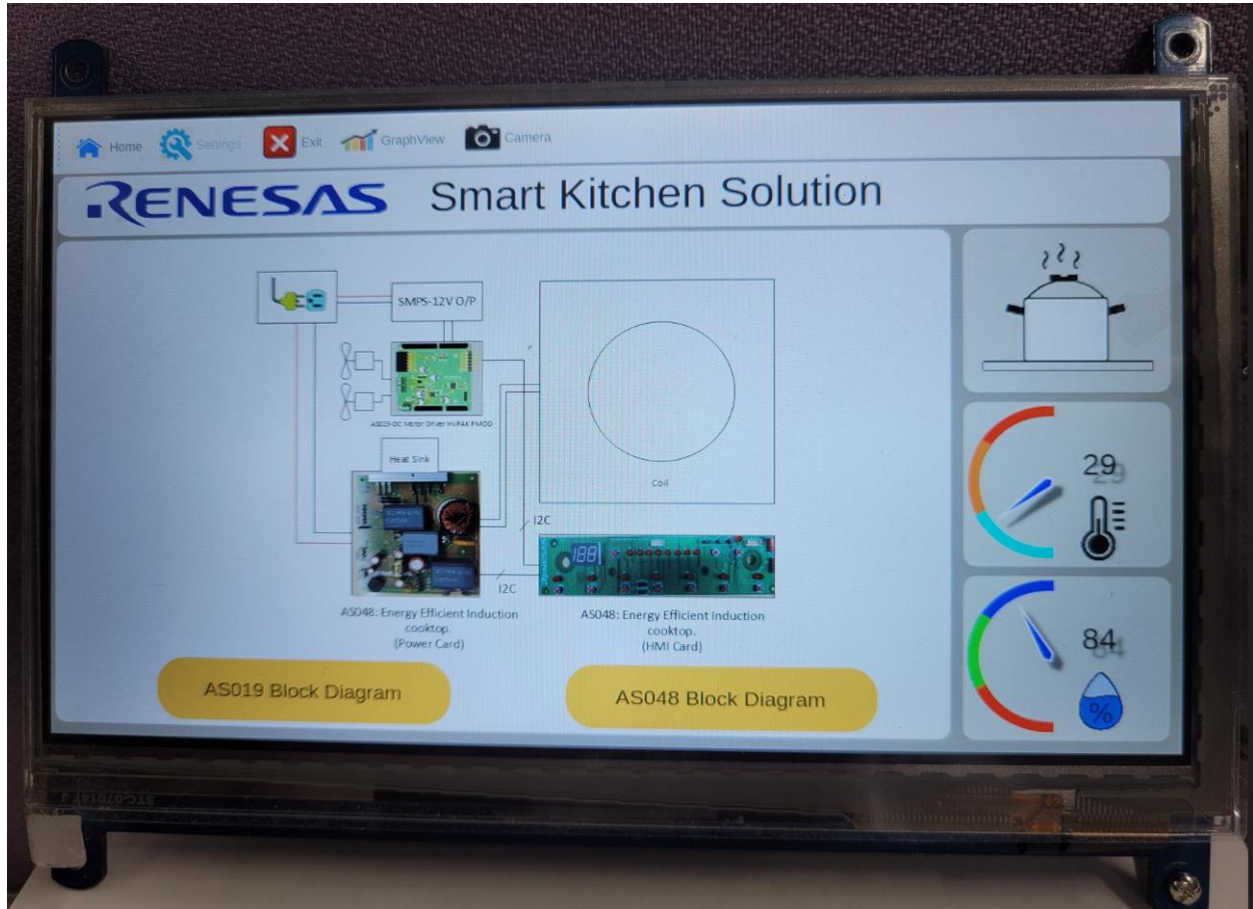


Figure 17. AS048INDCKTP-D-POCZ Block Diagram

- b. Clicking “**AS019 Block Diagram**” button will navigate to AS019PMODEXP-POCZ system diagram, as below



Figure 18. AS019PMODEXP-POCZ Block Diagram

7.2.2 Settings Screen

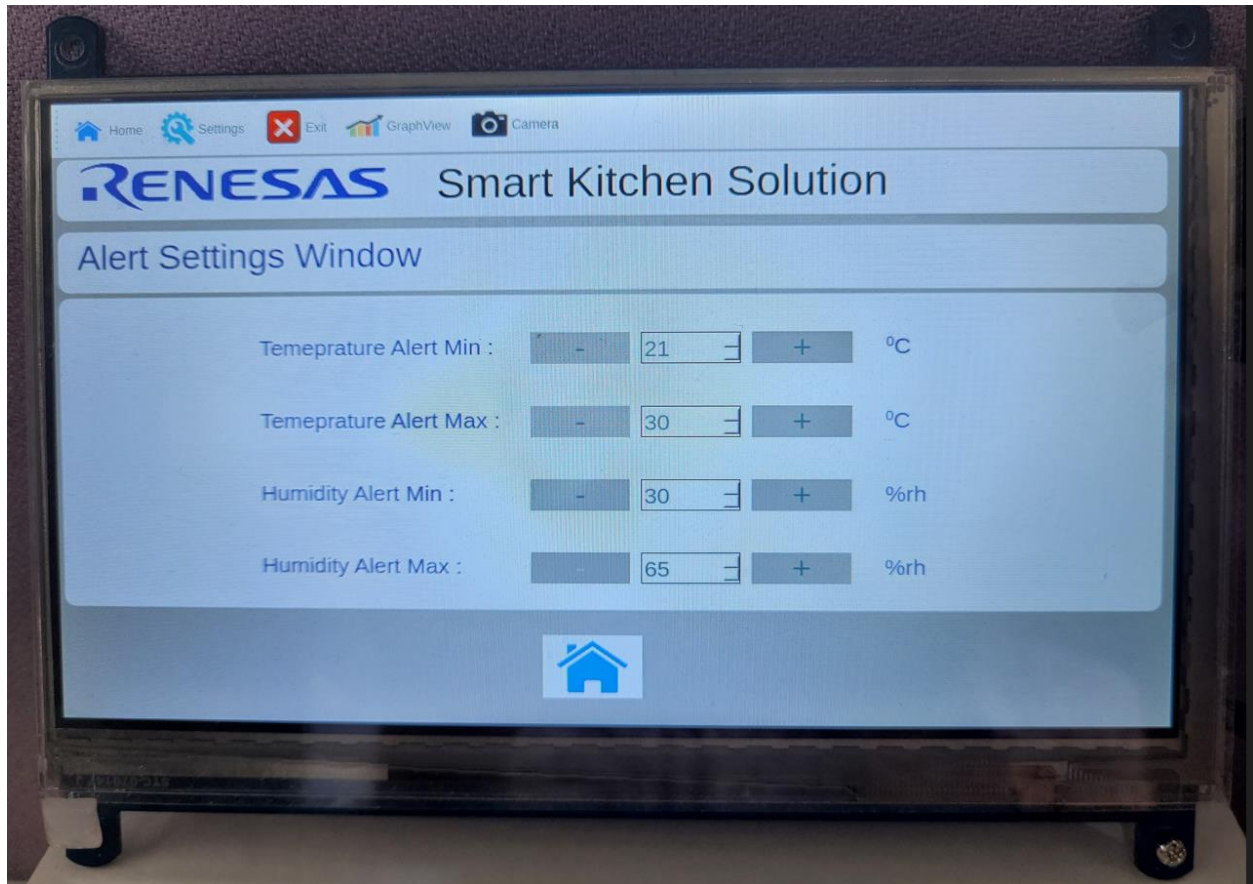


Figure 19. Settings Screen

The screen in the figure 14, allows you to set the maximum and minimum settings for temperature and humidity, which will trigger the alarm/alert event logic to execute.

- On clicking “+” button will increase the corresponding parameter value by one
- On clicking “-” button will decrease the corresponding parameter value by one

Clicking on the home button will navigate to home screen.

7.2.3 Graphical Screen

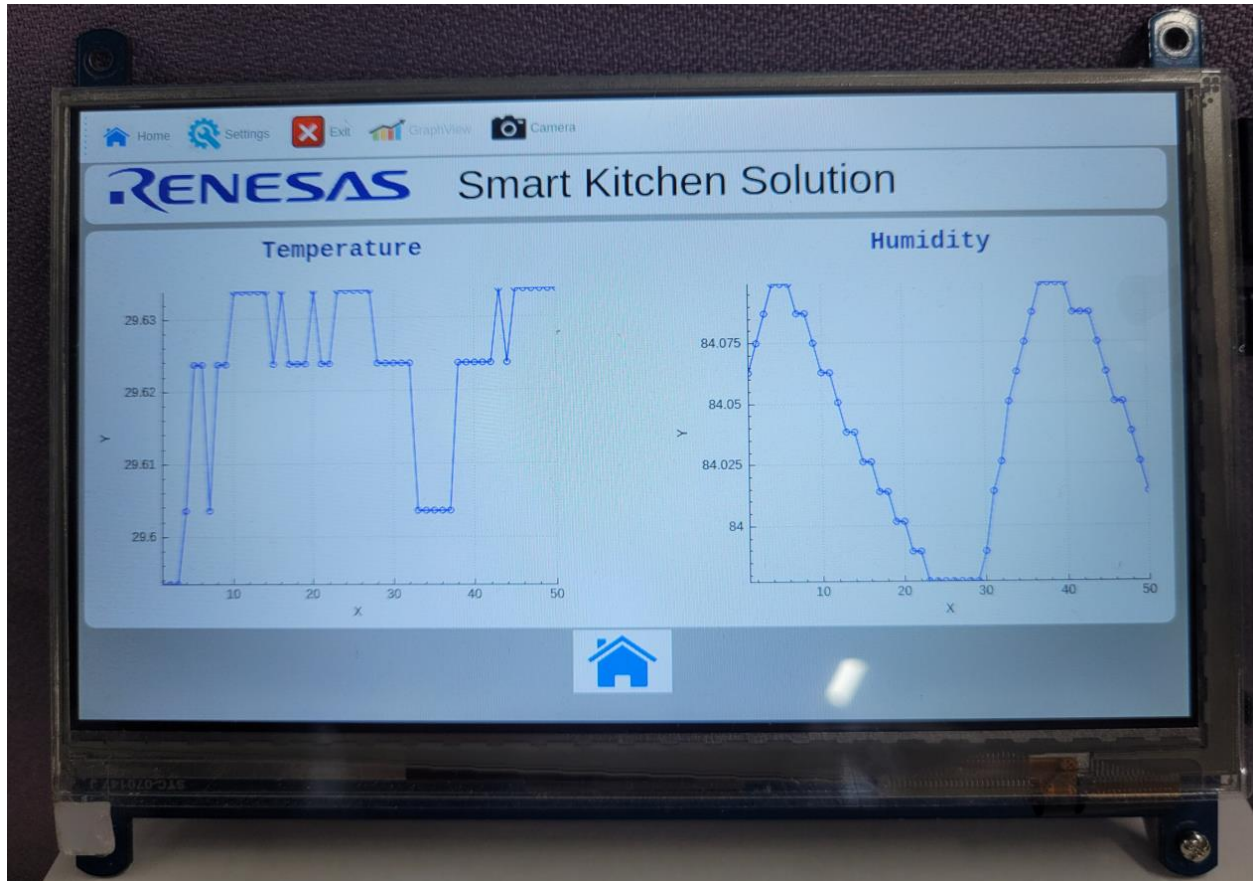


Figure 20. Graphical View Screen

The screen in figure 15, will plot the latest live sensor data on y-axis and sample numbers on x-axis. The temperature plot is on the left side of the screen, The humidity plot on right side of the screen.

Clicking on the home button in the graphical screen will navigate to home screen.

7.2.4 Alarm/Alert Screen

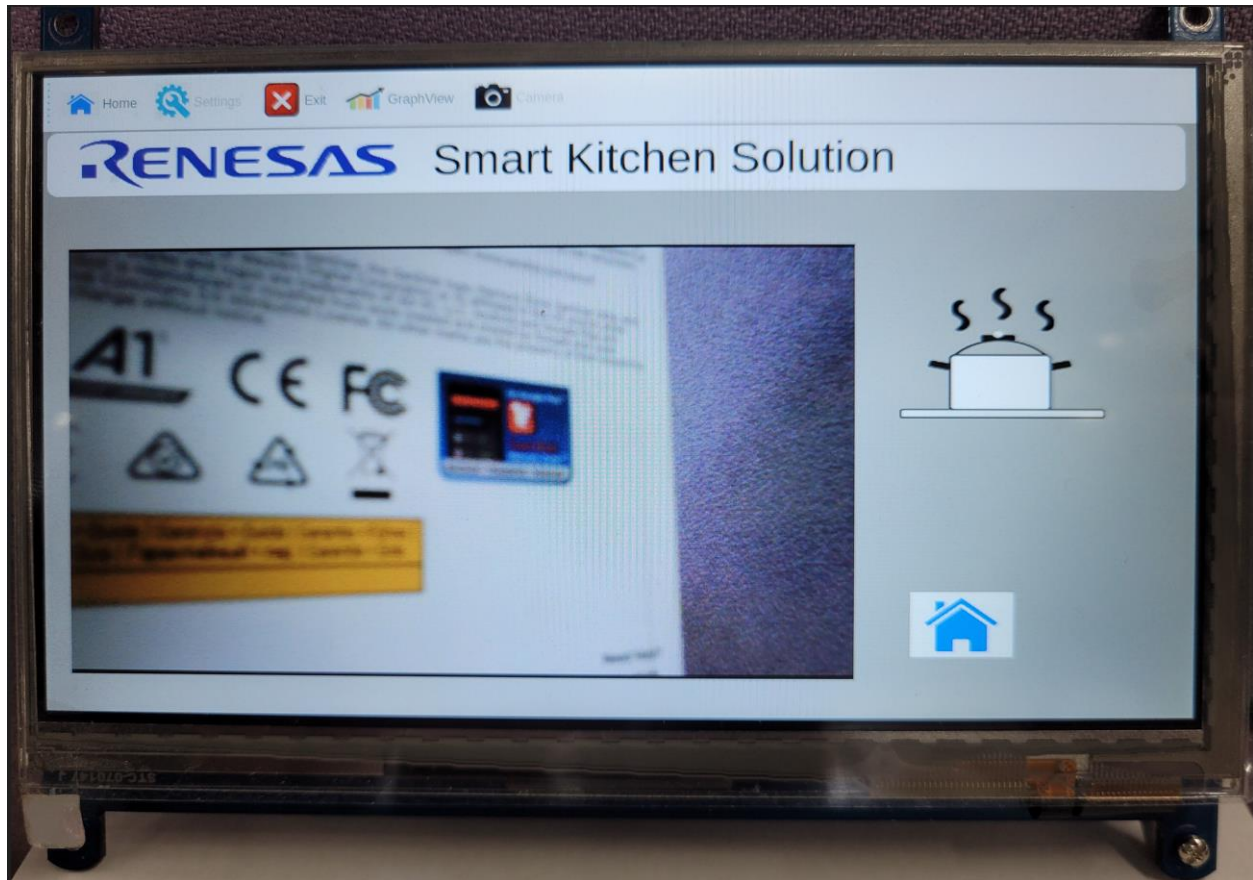


Figure 21. Alert Screen – Normal Condition

The screen in Figure 16, will appear when the alarm/alert icon is clicked in the menu tool bar. The normal cooking image will be displayed on right top corner of the screen. The camera live streaming data will appear on the left side of the screen.

Clicking on the home button in the Alert screen will navigate to home screen.

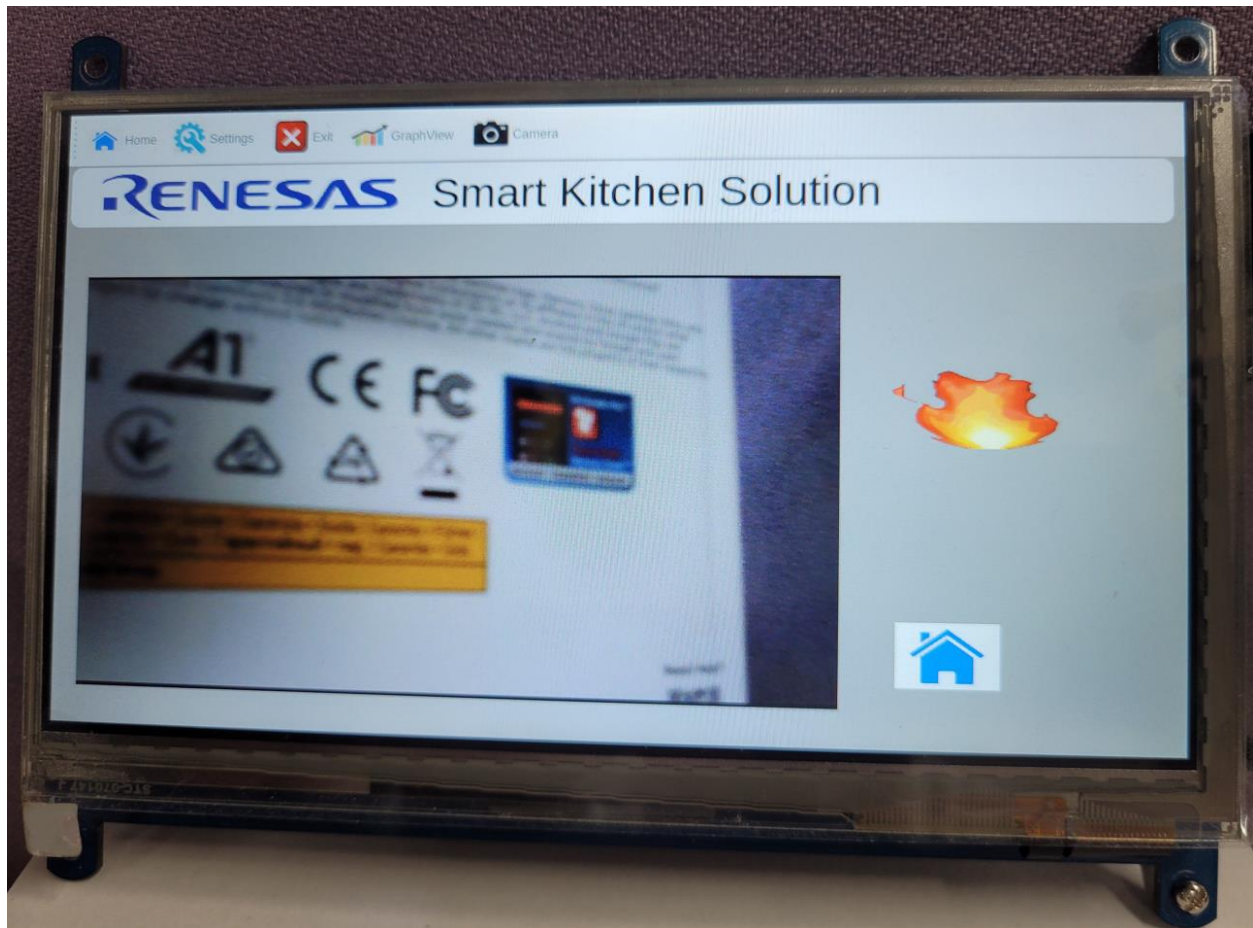


Figure 22. Alert Screen – Alert Condition

The screen in Figure 17, will appear when the alarm/alert event occurs. The alarm/alert event occurs when live current sensor data is more than the maximum value set in the setting screen of either temperature or humidity. The camera live streaming data will appear on the left side of the screen. And fire image will appear on the top right corner of the screen.

The PMOD1's PIN#7 voltage level is changed for temperature alarm/alert event. The PMOD1's PIN#8 voltage level is changed for humidity alarm/alert event

Clicking on the home button in the Alert screen will navigate to home screen.

8 Limitation & Know Issues

- The graphics and video encoder library are evaluation version, there is a time restriction on its execution.
- Occasionally when switching to camera screen under normal or alarm/alert condition the camera streaming sometimes doesn't happen. Under this scenario, if we navigate between camera screen and home screen once, the streaming will continue/resume.