EU070 Smoke Detector

Software Guide

Contents

1. Introduction

1.1 Overview

This document provides detailed information about the EU070 Smoke Detector PoC software architecture and how the functionality was implemented. The Smoke Detector demonstrator board is dedicated to detecting the surrounding smoke or dust, using a photo transistor and a photo diode, and enable an audible piezo transducer horn if the smoke intensity is over the selected threshold. Besides this, the demonstrator board may verify the battery load, and enable a horn test.

The whole system is built around a Renesas RL78/G12 microcontroller.

The IDE used for this Software project is E2Studio for RL78 version which also include the configurator tool.

1.2 Tools and Software Versions required

1.2.1 RL78/G12 Microcontroller

For RL78/G12 Microcontroller following versions are required:

- Renesas e^2 studio Integrated Solution Development Environment (ISDE) v 2020-10 or greater
- Code Generator Plug-in for RL78
- Renesas CC-RL v1.09.00 or greater

2. Resource Constraints and Usage

2.1 Memory resources

The chosen microcontroller, RL78/G12 (p/n R5F10267), includes:

- 4KB Code Flash Memory
- 2KB Data Flash Memory
- 512 Bytes RAM
- 256 Bytes + 2 KB Special Function Register (SFR)

2.2 CPU Load

The operation mode of the Smoke Detector board has a ratio of 0.4:999.6 for cyclic monitoring, this means that the CPU will work in normal operation mode for 0.04% of the time, and 99.96% of the time will work in stop mode.

2.3 Timing Constraints

The following timing constraints should be considered:

2.3.1 Timing constraints given by AFE IC

Parameter	Description	Constraint
SPI	SPI frequency	4MHz AFE required
AFE Ready pin	AFE Enable Unit	Wait for AFE Ready pin (max 50us)
LED1_EN pin	LED1 enable time	Wait for LED1 transceiver rise time (~70us)
BATT_TEST_EN bit	Battery Load time	Wait for battery load charging (~9us)
ADC_CFG AFE register (Bit 0 - *ADC_CONV_DONE*)	A/D conversion	Wait for ADC conversion end with timeout to avoid lock

2.3.2 Timing constraints given by MCU

Parameter	Description	Constraint
SPI flags	Transmission/Reception confirmation	Wait for SPI transfer to be complete, with timeout to avoid lock
ADIF MCU register	A/D conversion	Wait for A/D conversion of channel or internal reference end.

3. Physical Structure

3.1 Structure Overview

The EU070 Smoke Detector PoC board provides the following structure:

- RL78/G12 microcontroller 20 pins – low power and high performance
- Analog Front-End IC – provides the smoke detection drivers like:
    - LDO for microcontroller supply
    - Battery load test
    - SPI Interface
    - 2 LED drivers with 8-Bit DAC adjustable current
    - Photodetector amplifier, TIA, PGA and 10-Bit ADC for Photodetector Input
    - GPIO
- A green presence LED
- A red alarm LED
- An audible piezo horn
- A debugger connector – available for debugging or flashing
- An 'OR' type power supply configuration
- A boost converter up to 9V for Horn driver power supply

3.2 Hardware block diagram

The following diagram (Fig. 1) shows the hardware blocks available for the entire EU070 Smoke Detector board presented below. The diagram contains the general blocks used to implement the smoke detection, CO detection and battery monitoring algorithm. The RL78/G12 master unit establishes a SPI and GPIO communication with AFE IC, which provides the information about the smoke presence via an IR Emitter and an IR Receiver. Also, the AFE Unit provides information about the CO presence, using a CO sensor, or battery status by its internal resistance. As user input, a button is used, which has two functions: *start* an alarm test for 3 seconds (if alarms are disabled) or *stop* the smoke alarm. Also, for user information there are 2 LEDs used, a green one (presence LED), and a red one. The last one is enabled together with an audible piezo horn, just when a smoke or battery alarm is triggered, or a horn test is enabled. The power supply may be provided from multiple sources like debugger, 3V Battery, 9V battery, 5V Micro-USB connector or 220V AC, using an 'OR' circuit to select just one of them.

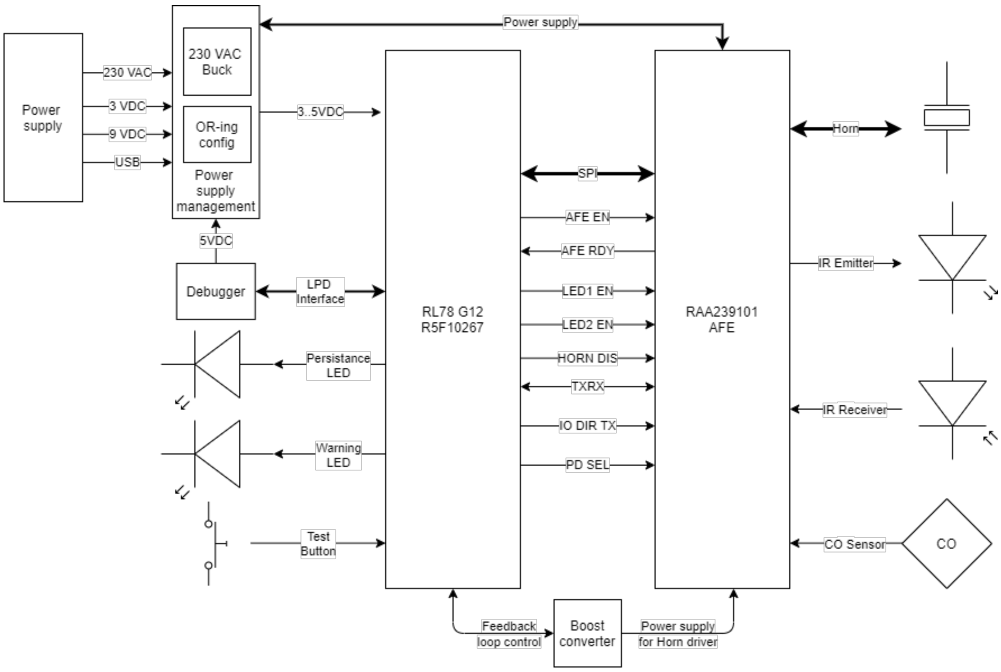

Fig. 1 Hardware block diagram

3.3 RL78/G12 microcontroller Building Blocks

In the current configuration are used: 4 Timer Array Unit channels, GPIOs, A/D converter – channel 0 and internal reference, Interrupt control – for user button, 12-Bit Interval Timer – for cyclic trigger and SPI communication – for communication with AFE IC.

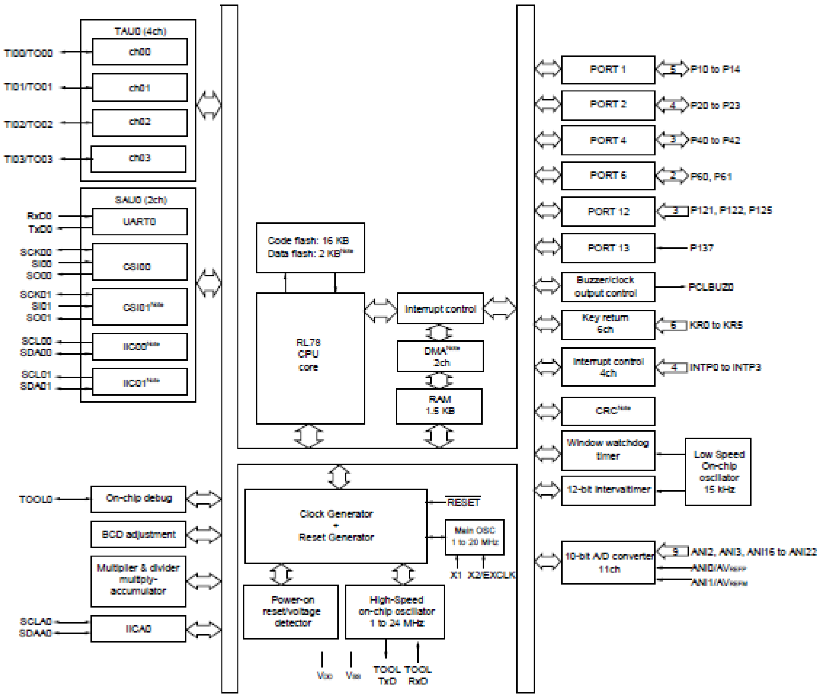

Fig. 2 Renesas RL78/G12 building blocks

4. Software Layer

4.1 Layering

The following diagram (Fig. 3) represents the software layer model.

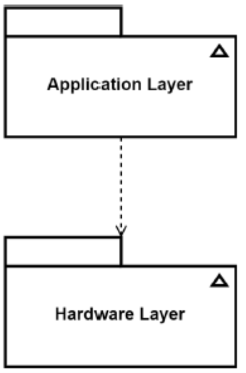

Fig. 3 Software layer model

4.1.1 Application Layer

The **Application Layer** implements the functionality of the Smoke Detector. This layer contains:

- System initialization (SINI).
- Specific functionalities implementation: smoke monitoring, battery status monitoring, boost converter output monitoring, data flash communication (APP).
- Services provided to the AFE module: read/write AFE IC registers, AFE GPIOs.

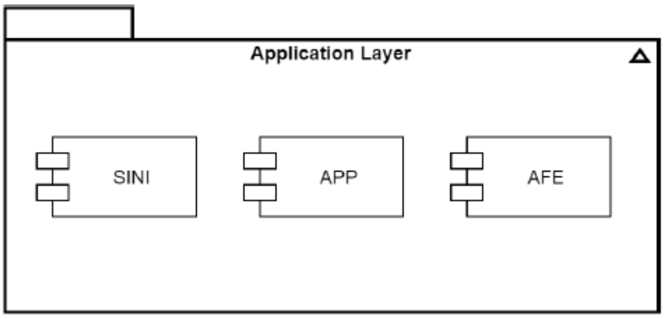

Fig. 4 Application layer blocks

4.1.2 Hardware Layer

The **Hardware Layer** is responsible for interfacing the MCU hardware components of the system with the rest of application. The main content of this layer are the drivers of the MCU, which are generated and configurated into E2Studio IDE.

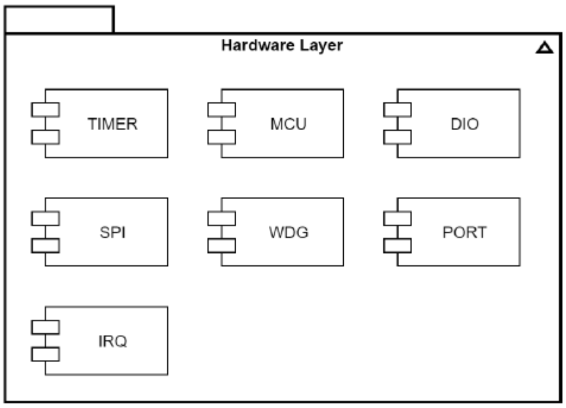

Fig. 5 Hardware layer blocks

4.2 Subsystems

Next images (Fig. 6) represent the software components diagram, and for each subsystem, the detailed modules:

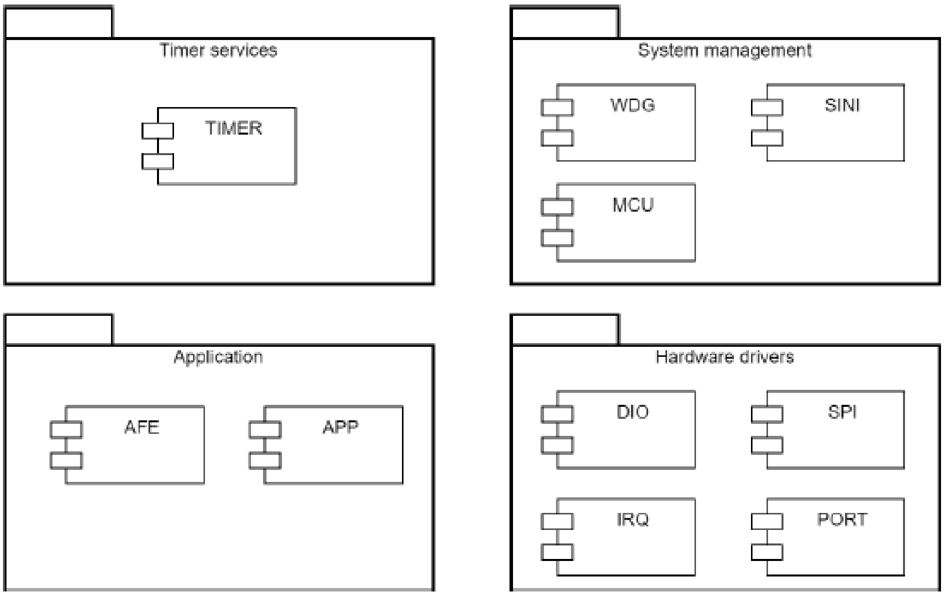

Fig. 6 Software components

4.2.1 Timer services

This module is responsible for providing the correct timing for measurements, like the timing used for smoke and battery monitoring, the timing used for horn pulses, or the timing used for boost converter PWM pulses.

4.2.2 System management

This software module contains modules and functions used for the initialization and management of the system.

4.2.3 Application

This component includes all customer specific functionalities.

4.2.4 Hardware drivers

This module contains all hardware drivers and hardware-linked modules in the system.

5. Context Management

5.1 System Initialization

Fig. 7 is showing the initialization sequence after power-on reset:

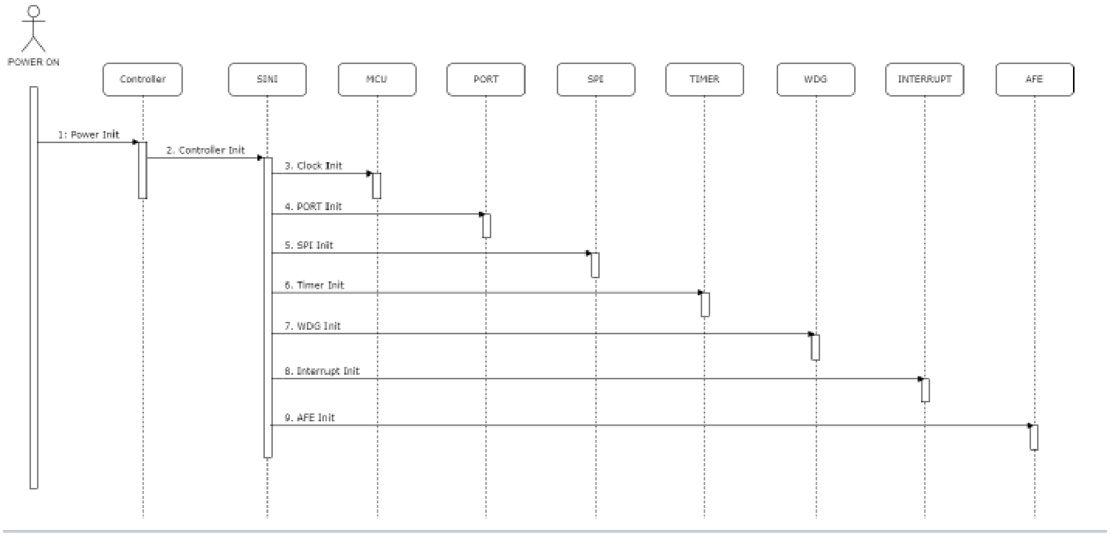

Fig. 7 Software initialization

The first step after power-on reset is the basic controller initialization. This is followed by the initialization the clock tree and microcontroller hardware components. The sequence is finished with initialization of AFE IC ports.

5.2 Interrupts

Below table presents all the interrupts used in the Smoke Detector algorithm:

Interrupt	Priority	Description
Reset	-	Microcontroller reset
r_it_interrupt	Low	Interval timer interrupt, generated at each 250ms
r_csi00_interrupt	Level2	SPI end data transmission/ reception
r_intc3_interrupt	Low	User button interrupt, generated on the rising edge
r_tau0_channel2_interrupt	High	End of channel 2 timer, used as interval timer for boost feedback
r_tau0_channel3_interrupt	Level1	End of channel 3 timer, used as interval timer for the width of horn pulses

5.3 Critical Sections

The critical section must be protected by disabling the interrupts during their run-time to avoid data mismatch across multiple modules or even software dead locks. As a result, the length of the interrupt locked section should be maintained to minimum.

In the current algorithm, the most sensitive sections are smoke measurement, CO measurement and battery monitoring. These are protected by disabling the interrupts for channel 3 timer and user button while the functions are running.

5.4 Synchronization

A software synchronization is required for cyclic monitoring / smoke detection and battery test which are called from the main module, based on flags, as follow:

The AFE ADC registers counter is cleared after the AFE Unit is disabled. In this case, after each smoke, CO or battery readings, the A/D conversion is processed and the AFE unit is disabled. On the other hand, the AFE Unit should be enabled when the horn driver is used. In this case, the cyclic monitorization of smoke, CO, or battery functions are called only if the horn pulse is disabled and the horn pulses are enabled, but only if the smoke/CO/battery functions are completed.

6. Mechanism

6.1 System State Model

6.1.1 Interval Timer State Model

Below diagram describes the states of one cycle of 12-bit Interval Timer, available for smoke, CO and battery monitoring.

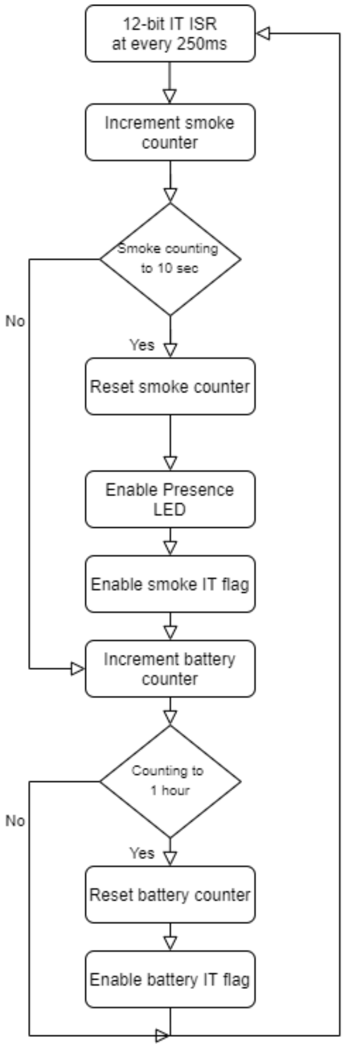

Fig. 8 Interval Timer state model

6.1.2 Main State Model

The Interval Timer flags, one for smoke and CO measurement and another for battery monitoring, enable the measurement functions as follow:

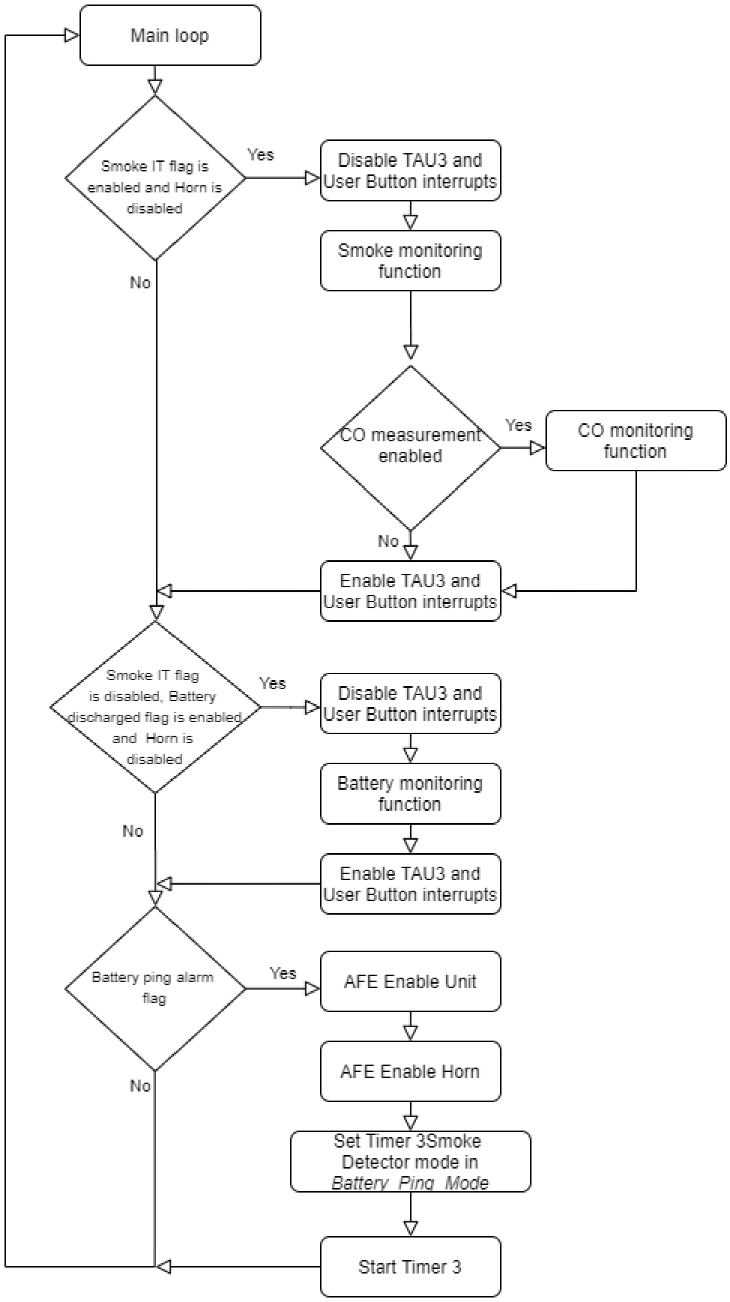

Fig. 9 Main state model

If the *battery discharged* flag is enabled, the battery alarm ping is called at every 2 minutes.

6.2 Cyclic Monitoring – Smoke Detection

The diagram below describes the measurement algorithm for the smoke intensity.

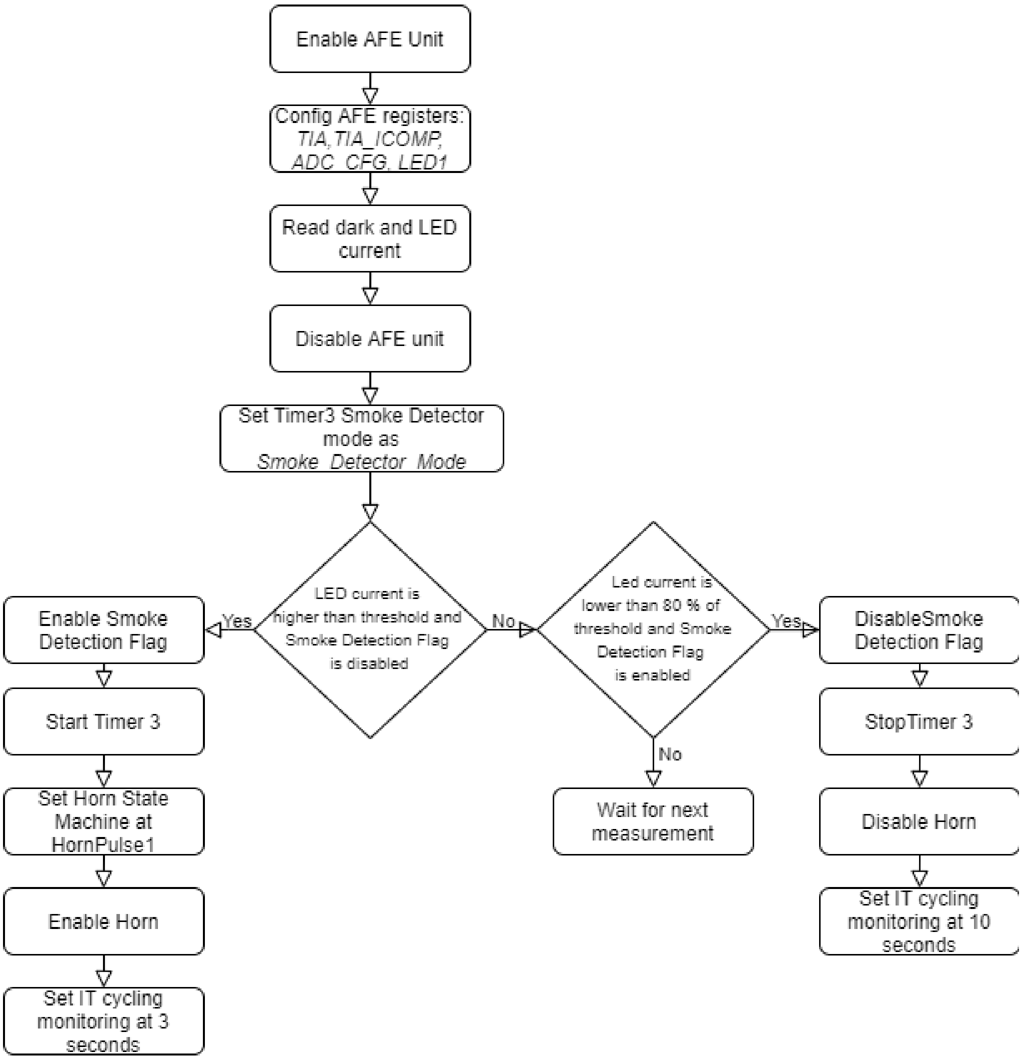

Fig. 10 Detailed representation of smoke monitoring

6.3 Cyclic Monitoring – Battery Test

The diagram below describes the measurement algorithm for the battery resistance.

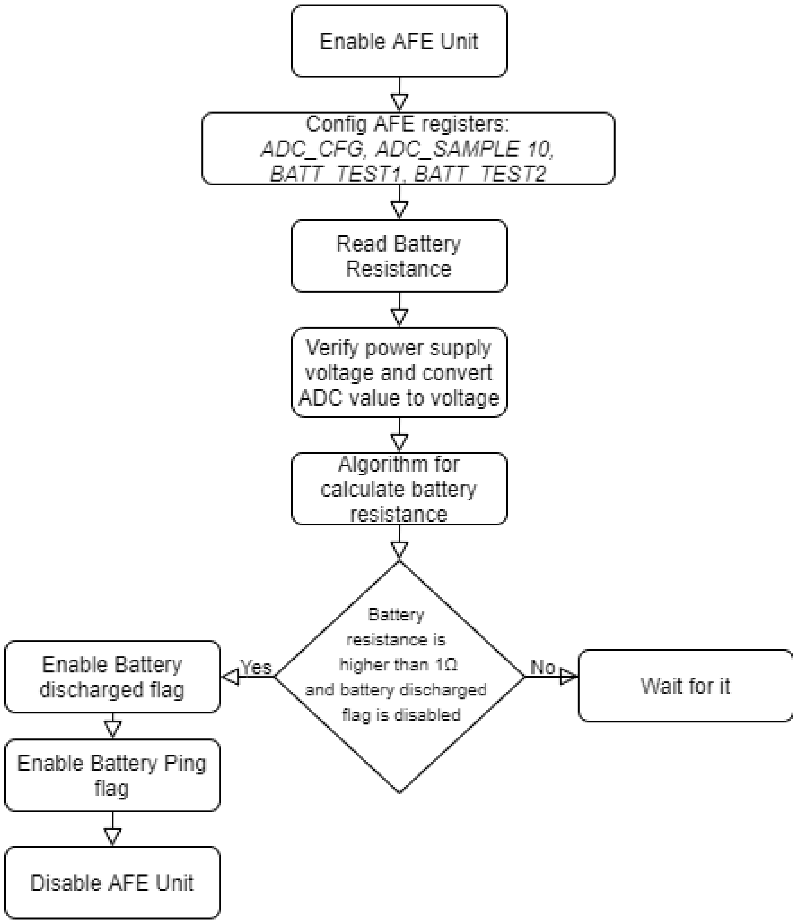

Fig. 11 Detailed representation of battery monitoring

6.4 Cyclic Monitoring – CO Sensor

The Fig. 12 describes the measurement algorithm for the CO monitoring.

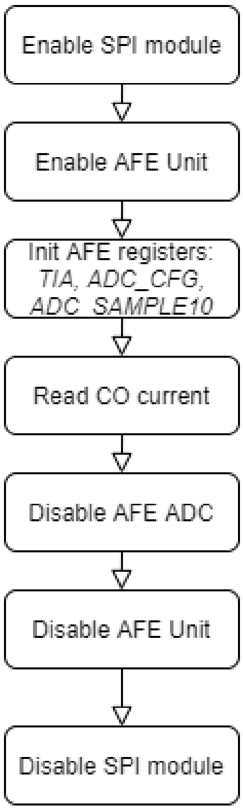

Fig. 12 Detailed representation of CO monitoring

7. Peripherals Configuration

7.1 Fixed Function Assignments

This chapter presents the configuration of the peripherals used in the system. Before starting to configure the peripherals, it is necessary to assign fixed functions to some pins of the MCU. The setup for the fixed pins is:

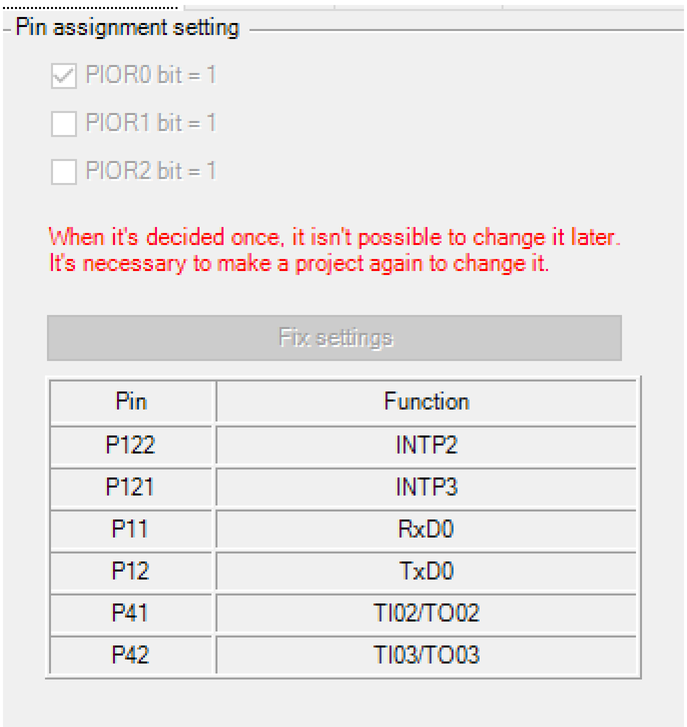

Fig. 13 Pin assignment setting

Once these settings have been applied, no more changes are allowed, only if a new configuration is started.

7.2 Clock tree

The RL78/L1C clock tree is based on the internal high-speed on-chip oscillator configured at 16MHz working frequency. This frequency value provides the lower power consumption, but also the main system clock may be configured at 8Mhz – low speed main mode. It is available for CPU and peripherals clock.

Interval Timer is an exception, because it uses the internal low-speed crystal oscillator, configured at 15kHz.

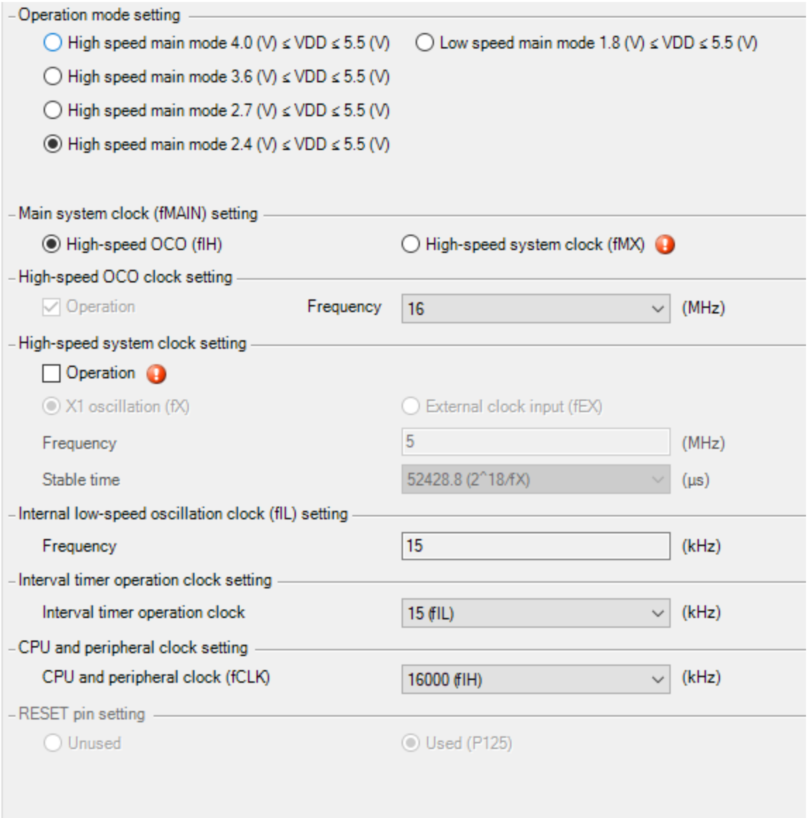

Fig. 14 Clock tree configuration

7.3 On-Chip Debugging

In the same configuration window, only for debugging mode it is neccesary to set the On-chip debug operation setting to *Used* and select the emulator model for debugging or flashing.

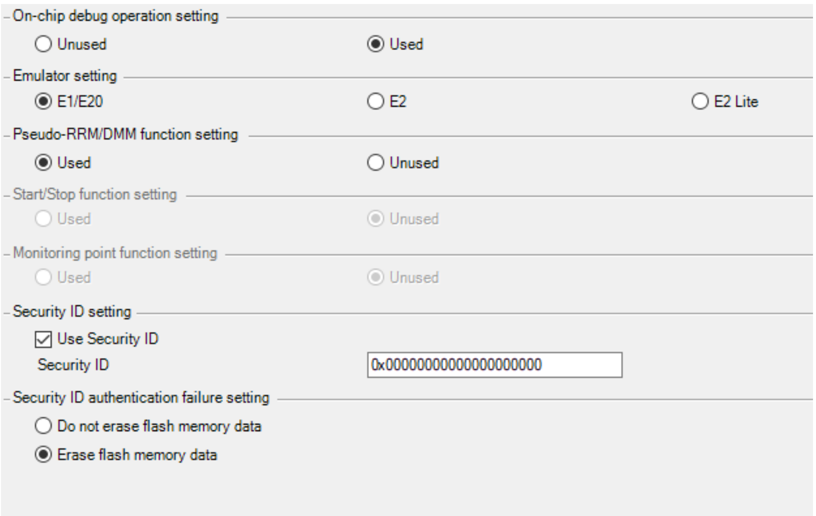

Fig. 15 On-chip Debug configuration

7.4 Reset Source / Output Function

It is recommended to enable the reset source of the system, to be able to determine the last reset source. The option *Output the function for confirming reset source* must be selected form the Confirming reset source tab.

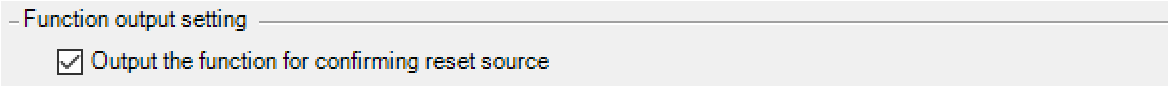

Fig. 16 Function output setting

7.5 Port function/ Pin assignments

The RL78/G12, R5F10267 model, provides 20 digital I/O pins which can control a variety of operations. In addition, these pins have several alternative functions.

7.5.1 Port 1 setup:

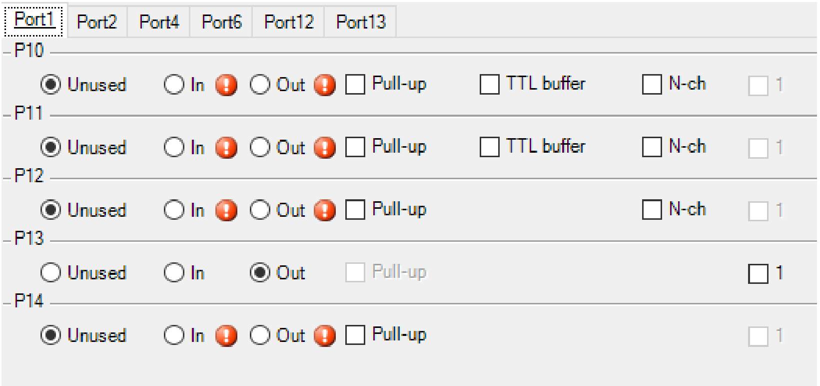

Fig. 17 Port1 configuration

7.5.2 Port 2 setup:

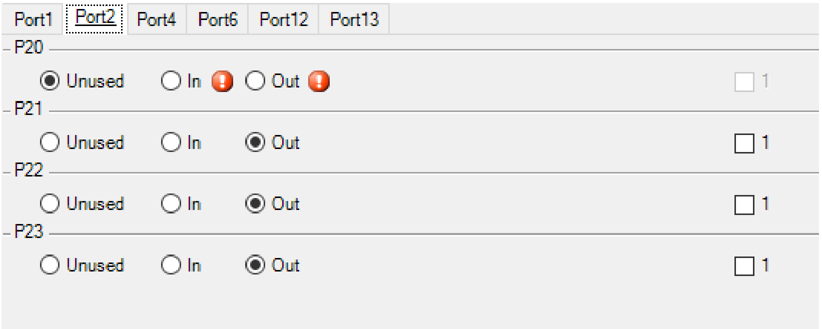

Fig. 18 Port2 configuration

7.5.3 Port 4 setup:

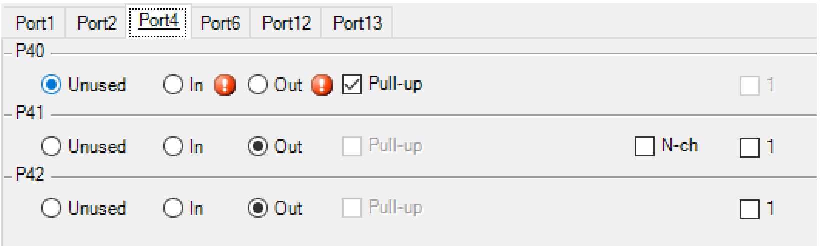

Fig. 19 Port4 configuration

7.5.4 Port 6 setup:

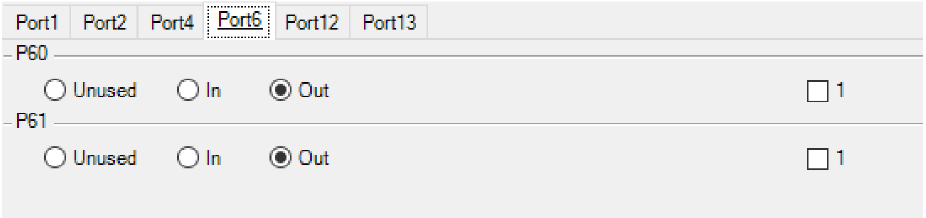

Fig. 20 Port6 configuration

7.5.5 Port 12 setup:

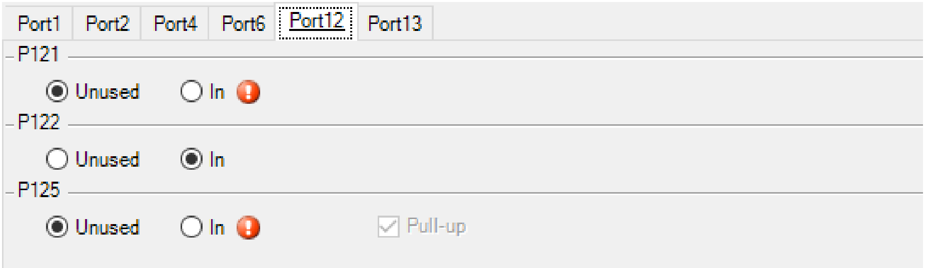

Fig. 21 Port12 configuration

7.5.6 Port 13 setup:

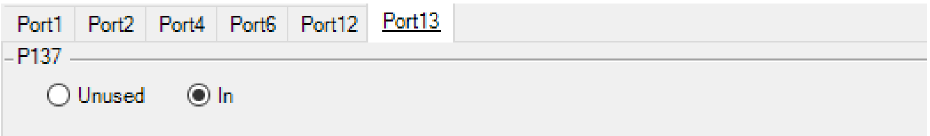

Fig. 22 Port13 configuration

7.5.7 Port Configuration Summary

The table below represents the description of port configuration.

PORT	State	Description
P13	OUT	AFE – SPI Enable Pin
P21	OUT	AFE- LED1 Pin
P22	OUT	AFE- AFE Enable Unit Pin
P23	OUT	AFE – AFE Horn Enable Pin
P41	OUT	Green Presence LED
P42	OUT	Red Alarm LED
P60	OUT	AFE- PD_SEL Pin
P61	OUT	AFE- IODIR TX Pin (*not used*)
P122	IN	AFE- AFE Ready Pin
P137	IN	AFE -TXRX Pin (*not used*)

7.6 External Interrupt

In the present configuration a single external interrupt is necessary, used for User Button, which is configured to rising edge mode.

Fig. 23 Interrupt configuration

7.7 Serial Array Unit

The Serial function is used to enable the CSI communication between MCU and AFE IC by Channel 0. It is open to both transmission and reception, using interrupt and callback functions. The baud rate is limited by AFE.

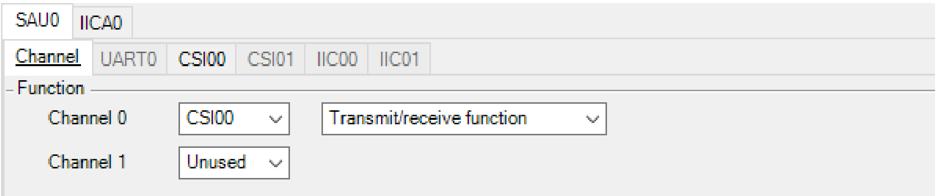

Fig. 24 SAU Channel configuration

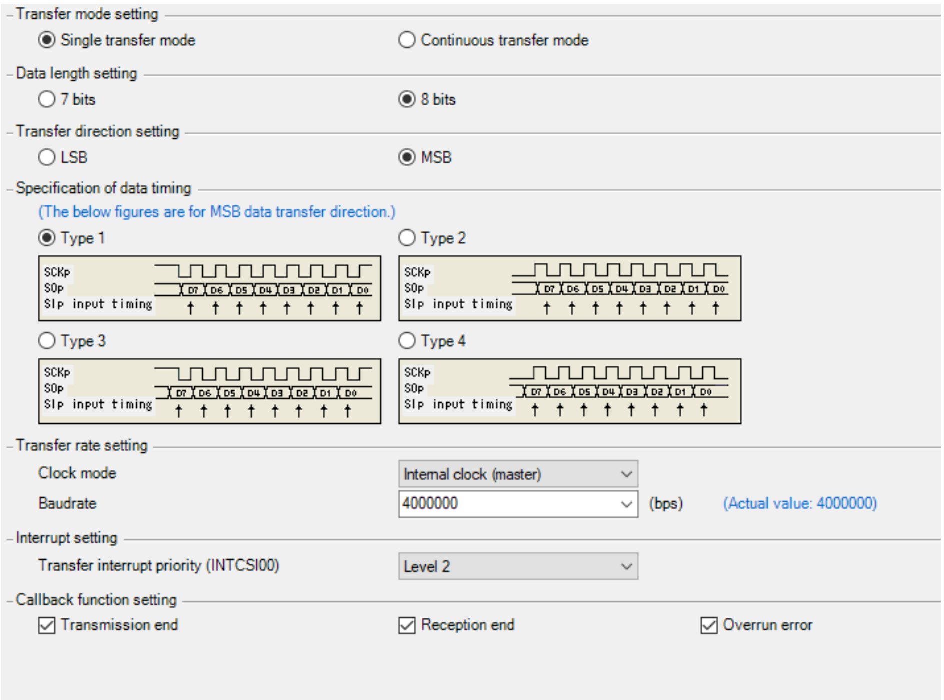

Fig. 25 SAU0 settings

7.8 A/D Converter

The A/D Converter of the RL78/G12 Microcontroller is used to:

- control the feedback loop of boost converter – *Analog Input channel 0*
- determine the power supply voltage - *Internal reference*

A software trigger is available for both cases.

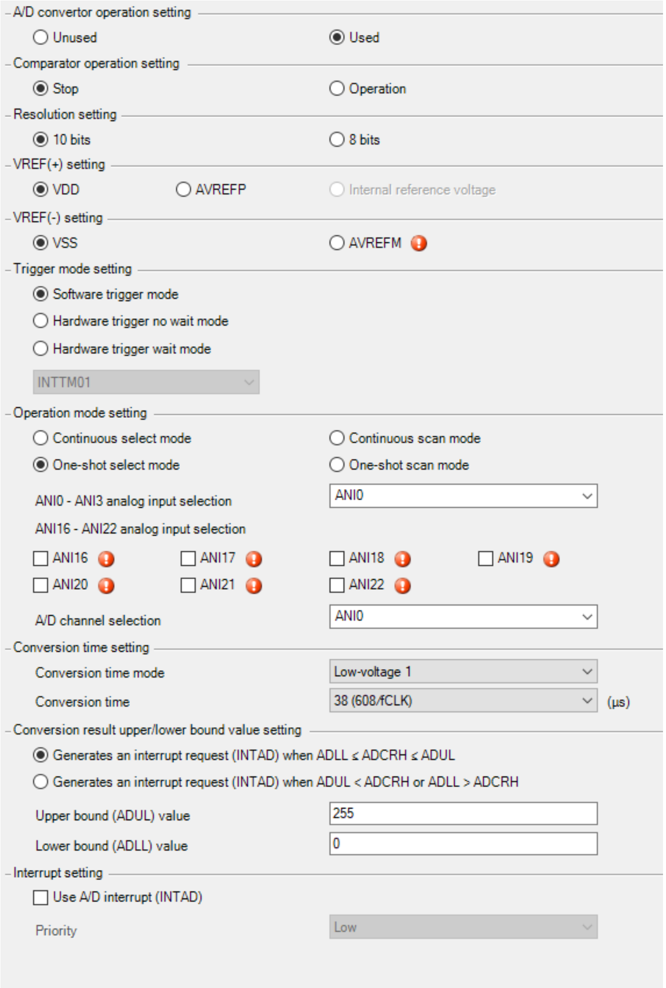

Fig. 26 A/D Converter configuration

7.9 Timer Array Unit

7.9.1 TAU Overview

The Timer Array Unit has 4 channels, each of them configured independently as follows:

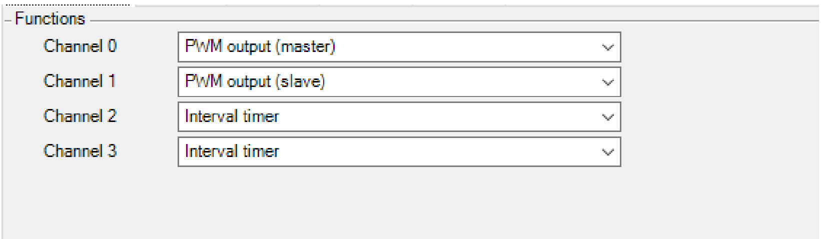

Fig. 27 Timer channels configuration

7.9.2 Channel 0 & 1

Channel 0(master) and Channel 1(slave) are set for the PWM control of boost converter at 166.6Khz frequency.

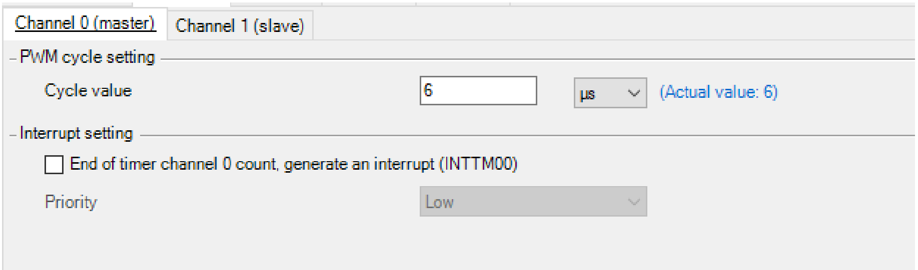

Fig. 28 Channel0 configuration

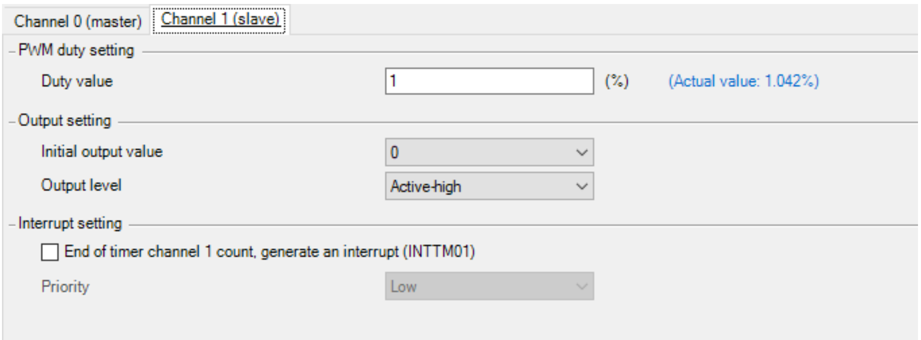

Fig. 29 Channel1 configuration

7.9.3 Channel 2

Channel 2 is configured as interval timer to trigger the boost feedback loop control.

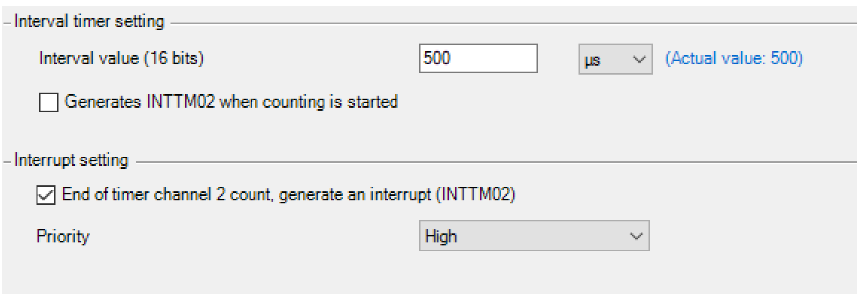

Fig. 30 Channel2 configuration

7.9.4 Channel 3

Channel 3 is configured as interval timer to trigger the smoke/battery alarm pulses. The register value is updated for each alarm case as follows:

1. **Smoke alarm**: 3 bursts of sound lasting 500ms with a 500ms of silence between them. The cycle of the three pulses is separated by a gap of 1500ms between them.

2. **Battery alarm**: a pulse lasting 500ms at every 2 minutes.

3. **Horn test**: a single burst of 3000ms.

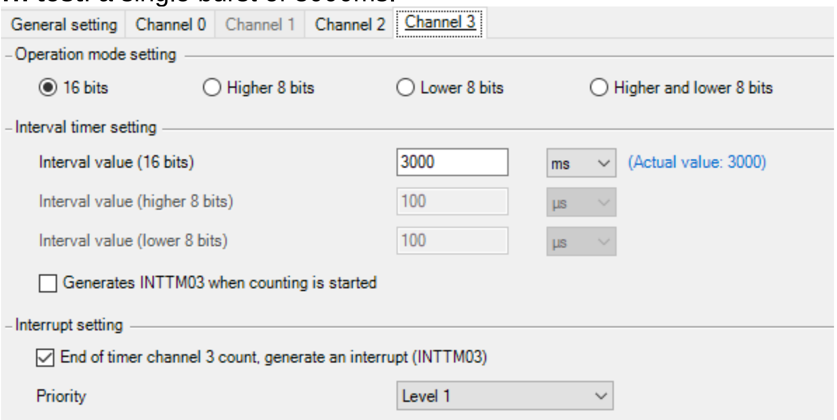

Fig. 31 Channel3 configuration

7.10 12-Bit Interval Timer

The 12-Bit Interval Timer is used to trigger a new cyclic measurement and to set the MCU in normal mode (SNOOZE), using the interrupt function. The value of cyclic interval is 250ms, the necessary time to implement the counter flags for smoke and battery measurements.

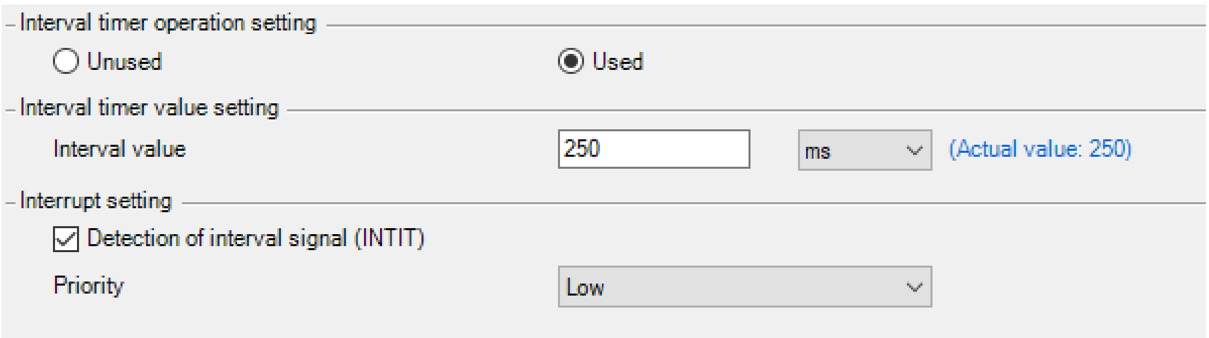

Fig. 32 Interval Timer configuration

8. Functional Design

8.1 Component Interfaces

This chapter contains information about the implemented modules and their interfaces.

8.1.1 Application Module Interface

The *Application module* is responsible for the communication with data flash (read and write data), to read power supply voltage and to implement smoke and battery cyclic monitoring.

Interface	Description
APP_vInit()	Initialization function
App_vGetADCReferenceVoltage	Function used to calculate ADC voltage reference
App_vGetPowerSupplyVoltage	Function used at initialization to determine the power supply voltage
APP_vBatteryMonitoring	Function used to determine battery resistance value
APP_vCyclicMonitoring	Function used to determine smoke presence
APP_vCOMonitoring	Function used to determine CO presence
APP_vInitMeasurement	Function used to enable SPI module and AFE unit before measurement
APP_vDinitMeasurement	Function used to disable SPI module and AFE unit after measurement
APP_vGetValueFlash	Function used to read data from data flash memory
APP_vSaveValuesToFlash(T_CALIB_DATA* inputBuffer)	Function used to write data in data flash memory
APP_delay	Software delay necessary after changing the ADC channel into internal reference

8.1.2 AFE Module Interface

The *AFE module* functions which access directly the AFE pins and Hardware module:

Interface	Description
AFE_vInit ()	Initialization function
AFE_enRead(AFE_tenReg reg, uint8_t * rx_buf, uint8_t counter)	Function used to read AFE registers
AFE_enWrite(AFE_tenReg reg, uint8_t * tx_buf, uint8_t counter)	Function used to write AFE registers
AFE_Driver_SPI_Error_Notification()	Callback from SPI driver in case the transmission error occurred
AFE_Driver_SPI_RX_Confirmation()	Callback from SPI driver to notify the end of reception
AFE_Driver_SPI_TX_Confirmation()	Callback from SPI driver to notify the end of transmission
AFE_vEnableUnit()	Function used to enable the AFE IC Unit
AFE_vDisableUnit()	Function used to disable the AFE IC Unit
AFE_vEnableHorn()	Function used to enable the AFE Horn Driver
AFE_vDisableHorn()	Function used to disable the AFE Horn Driver
AFE_vGetStatus()	Function used for a AFE A/D dummy conversion
AFE_vReadPD1()	Function used to read dark current and LED1 current
AFE_vReadPD2()	Function used to read current from CO sensor
AFE_vBatteryCheck()	Function used to read battery load value

9. PoC Software-Update

The application can be updated using the E1 Renesas debugger and Renesas Flash Programmer tool. This feature is available both for data and code flash. To be able to do this, it is required to have a *.hex file that contains the data flash parameter values or a new version of firmware. The next steps should be followed:

A new project must be created.

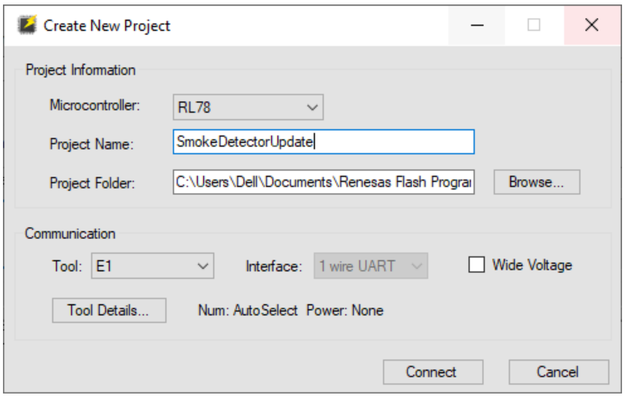

Fig. 33 Create a new project in Renesas Flash Programmer tool

Connect to the development board. Load the .hex file using *Browse* button and finally press *Start* to start flashing the new dataset.

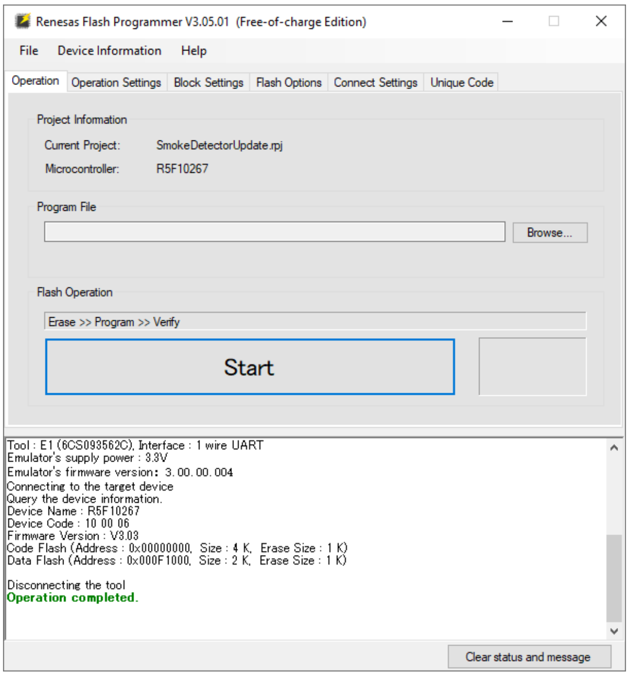

Fig. 34 Connect to the board and flash .hex file using Renesas Flash Programmer tool

10. Nomenclature

- IC Integrated Circuit
- PoC Proof of Concept
- SW Software
- HW Hardware
- MCU Microcontroller
- AFE Analog Front-End
- LED Light Emitting Diode
- IR Infrared
- TIA Transimpedance Input Amplifier
- PGA Programmable Gain Amplifier
- DAC D/A converter or Digital to Analog Converter
- ADC A/D converter or Analog to Digital Converter
- SPI Serial Peripheral Interface
- DC Direct Current
- AC Alternating Current
- LDO Low-Dropout Voltage Regulator
- GPIO General-Purpose Input/Output
- SINI System Initialization
- APP Application
- IDE Integrated development environment
- CO Carbon Monoxide

References

[1] Renesas Electronics, “EU070 – Software User Guide”. *[>> THIS DOCUMENT <<]*

[2] Renesas Electronics, “EU070 – Quick Start Guide”.

[3] Renesas Electronics, “EU070 – Hardware User Guide”.

[4] Renesas Electronics, “RL78/G12 - User’s Manual: Hardware”.

[5] Renesas Electronics, “E1/E20/E2 Emulator, E2 Emulator Lite - Additional Document for User’s Manual - (Notes on Connection for RL78)”

Revision History

Rev.	Date	Description / Summary
0.1	13.01.2021	Initial REE version

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

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

www.renesas.com

Trademarks

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/.