Bio Sensing Software Platform

Software Library for Measuring Human Activity

R11AN0323EJ0101
Rev.1.01
Sep 14, 2018

Abstract

This document describes the sample program to obtain the human activity level using accelerometer.

Target Device

RX231

Target Board

- Renesas Starter Kit for RX231 (R0K505231S000BE) (Renesas Electronics)
  Hereafter, it is abbreviated as RX231 RSK.
- Accelerometer module: Pmod™ ACL2 (Digilent Inc.)
  SPI interface module included Analog Devices ADXL362

Contents

1. Introduction

This application note describes how to measure human activity level with RX231 and accelerometer.

The human activity level means number of walking/running steps and consumption energy.

1.1 Terminology

Table 1-1 Terminology

Term	Meaning
ADP	Adenosine DiPhosphate
ALM	Activity Level Measuring
API	Application Programming Interface
ATP	Adenosine TriPhosphoate
DSP	Digital Signal Processor.
FIFO	First-In First-Out type buffer
FPU	Floating-Point number processing Uni
LSB	The Least Significant Bit
MEMS	Micro Electro Mechanical Systems
MET	Metabolic Equivalent of Task
MSB	The Most Significant Bit

1.2 Overview

This application note answers the following topics:

- How to calculate number of steps from acceleration
- How to calculate consumption energy from steps
- How to use an accelerometer device
- Algorithm of converting from sensor output to the activity level

1.3 Devices

In this application note, the system is constructed following major devices:

- MCU: RX231 series 32bit microcontroller by Renesas Electronics

  RX231 series CPU leverages a 32bit RXv2 CPU core with DSP/FPU and low power consumption technology to realize extreme power efficiency.

- Sensor: ADXL362 by Analog Devices, Inc.

  ADXL362 is an ultralow power, 3-axis MEMS accelerometer.

  - 1.8[uA] at 100[Hz] ODR, 2.0[V] supply
  - 1[mg/LSB]
  - Adjustable threshold sleep/wake mode for motion activation
  - Autonomous interrupt processing
  - Deep embedded FIFO
  - Low noise down to 175[ug/√Hz]
  - Selectable measurement range
  - External trigger for sampling
  - SPI digital interface

2. Functional Purpose

Monitoring of Human activity is simple and inexpensive body-worn motion sensors that are readily being used by researchers and practitioners to assess and motivate physical activity behaviors. Human activity Monitor-determined physical activity indices which are needed to guide individual’s efforts.

2.1 Step count of walking/running

In ordinary, the velocity of walking is within a certain range. It is said to be from 60 steps/minute (stroll) to 200 steps/minute (Race walking). And some sprinters run 50 steps a 100 m in 10 seconds (300 steps/minute).

These values figure out this systems requirement. Namely the system must cover the range of steps from 60 steps/minute to 300 steps/minute.

2.2 Consumption energy

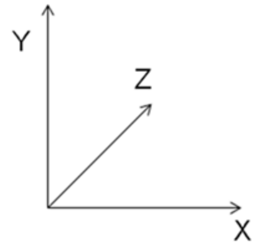
The system estimates consumption energy when above action was executed. Other than that, the system does not calculate consumption energy.

Because the exercise strength is different walking and running, this system must be able to classify these exercise modes. In this case, persons are under walking mode when the step count is less than 150 count a minute. Otherwise, they are under running mode.

3. Measurement Principle

3.1 Total Acceleration calculation from 3-axis point

The 3-axis accelerometer will give the acceleration applied on these three axes. The combined acceleration (mag) is the total acceleration applied on the object and that can be calculated using the formula mentioned in the below figure.



$$|mag| = \sqrt{(x^2+y^2+z^2)}$$

Figure 3-1 Acceleration calculation from 3-axis point

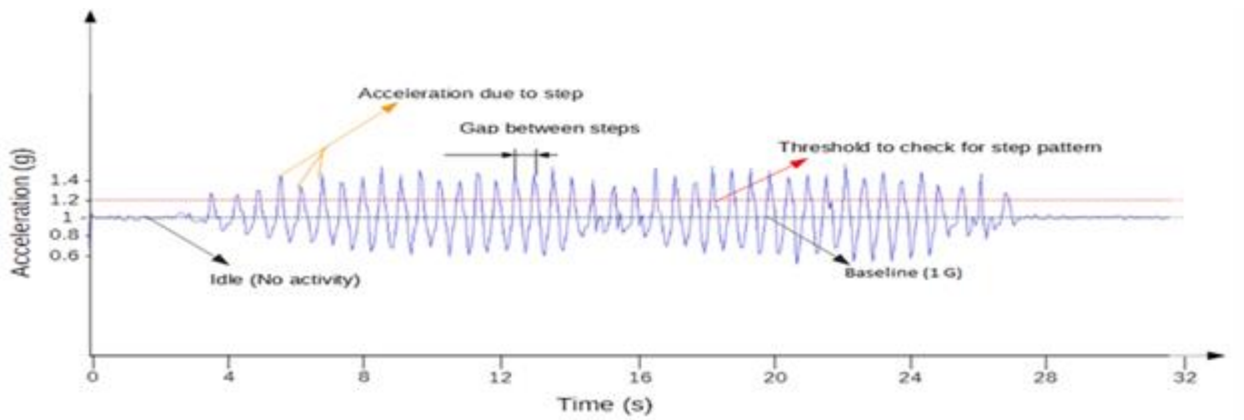
3.2 Measurement Value

3.2.1 Equations

(1) **Calculation the step count from the acceleration Characteristics**

(a) **Typical pattern of acceleration measured while walking:**

The following diagram shows the combined acceleration pattern of walking against time when the ADXL362 placed on chest.



Note: The acceleration data (X, Y and Z) is captured from ADXL362.
Acceleration = sqrt($x^2 + y^2 + z^2$)

Figure 3-2 Typical pattern of acceleration measured while walking

(b) **Step Pattern:**

From the above figures a step will have the following properties

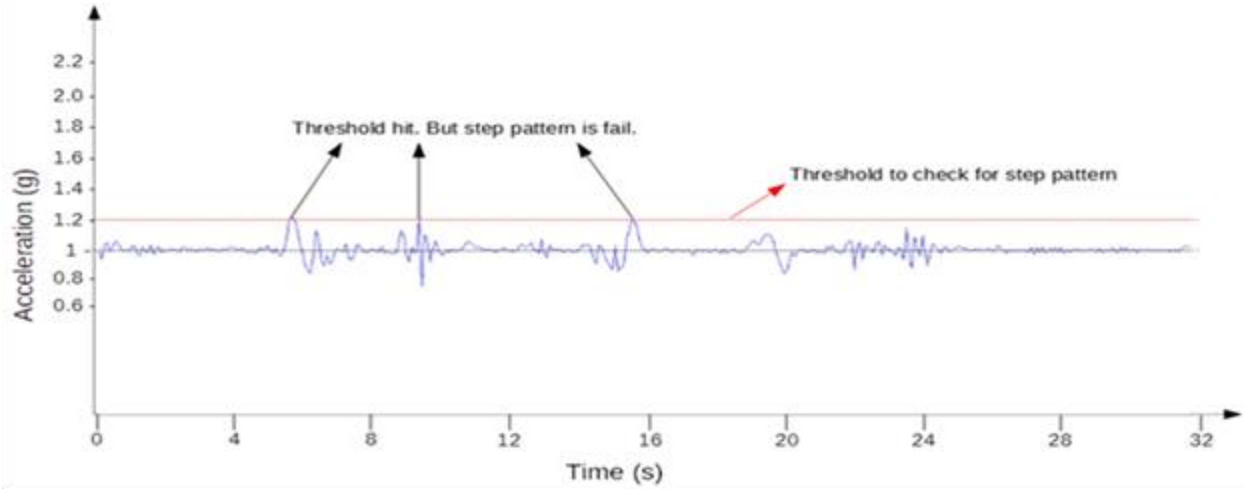
- The acceleration peaks crossing the step threshold happens for every step taken.
- The step peaks occur at uniform interval with deviation less than 500 ms

(c) **Step Pattern Fail Conditions:**

- Acceleration peak not crossing the step threshold.
- The step peak not occurring at uniform interval with deviation more than 500 ms.

(2) **Typical pattern of acceleration measured while doing activities**

(Activities include standing, sitting on chair, preparing coffee, arranging things in drawer)



Note: The acceleration data (X, Y and Z) is captured from ADXL362.
Acceleration = sqrt($x^2 + y^2 + z^2$)

Figure 3-3 Typical pattern of acceleration measured while doing activities

From the above figures the step calculation algorithm can be improved as below to avoid the false detection of step.

- Calculate number of peaks crossing the threshold.
- Recognize the pattern of peaks occurring at uniform interval (± 500ms).
- If pattern is observed, then set the pattern observed flag as true.
- If pattern observed flag is true, then increment step count for each peak that crosses the threshold.

(3) How to Count Steps

The step calculation threshold is the minimum acceleration threshold that a step has to cross for taking that acceleration data into step detection algorithm. This minimum acceleration threshold is kept between noise level and the minimum value that a step will produce.

The maximum noise produced peak produced during resting state is 0.15 G (150 mG)

The minimum acceleration produced by the step (slow walking) is 0.2 G (200 mG)

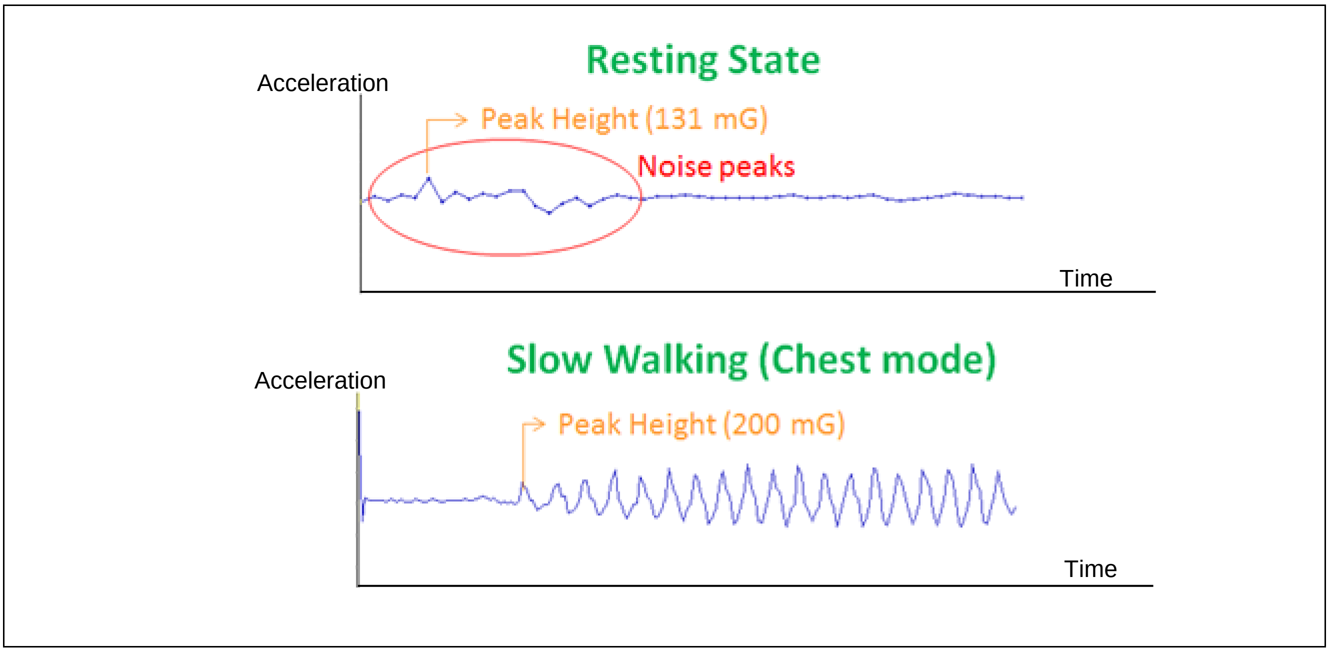


Figure 3-4 Acceleration threshold of Noise and Step

(4) **Accelerometer range requirement**

Table 3-1 ADXL Range Options

Range	Resolution
±2G	1 mG / LSB

ADXL Configuration:
Data length: 12-bit (0 to 4095)
Range: 2G

The acceleration ranges from slow walking to fast running with range setting as ±2G is below,

Table 3-2 ADXL Range Check

The acceleration range: ±2G	LSB Value (For lowest step peak height)		
Range	Low value	High value	Peak Height
Slow walking	872	1063	191
Normal walking	587	1868	1281
Brisk walking (with fast hand swing)	2097	3543	1446
Jogging	831	3220	2389
No Action (Noise)	955	973	18

The ration of step peak to noise peak for 2G range for slow walking is given below,

Table 3-3 Signal to Noise Ratio

Range setting	Step peak height	Noise peak height	Ratio (Step peak / Noise peak)
±2G	191	18	10.6

Since the acceleration information resolution and quality is better in the 2G range setting, it is chosen for the step counter. The below table shows the result of step count actual vs calculated for 2G range for slow walking,

Table 3-4 Test Results (Slow Walking)

Range	Actual steps	Calculated steps	Error %
±2G	40	40	0 %
±2G	50	53	6 %
±2G	60	63	5 %

(5) **Calorie Calculation Formula**

(a) **Basic Definitions**

Metabolism:

Metabolism is a term used to refer particularly to the breakdown of food and its subsequent transformation into energy the person's body needs. In the field of Biology, Metabolism refers to all of the body's chemical processes, the digestion of food, and the elimination of waste.

Calorie:

Calorie is the amount of chemical energy used by the body to heat one gram of water by one degree.

Metabolic Rate:

The term, 'Metabolic Rate,' refers to the amount of chemical energy a person frees from their body per unit time. A person's metabolic rate is commonly expressed in terms of kcal per hour or day. Any physical activities of the body will raise the metabolic rate which in turn raises the calories burnt by the body.

Relationship between moving person and calories burnt:

The below diagram shows the relationship between force applied by using chemical energy (MET Rate), acceleration produced, and distance moved with the work done (in calories).

Metabolic Rate for a step = Chemical Energy consumed for a step (in Calories)

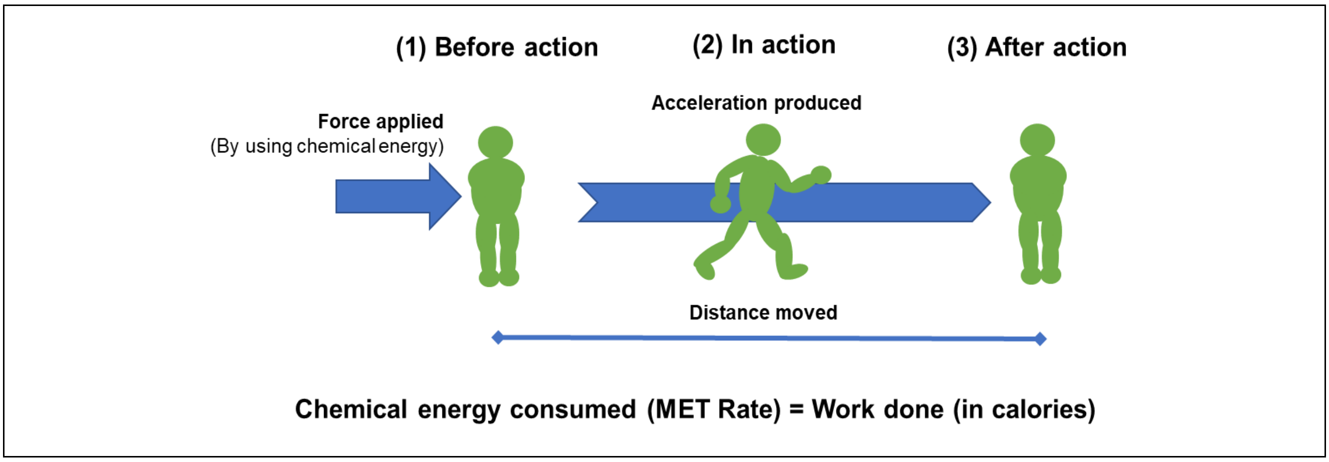


Figure 3-5 Relationship between Force, Person, Acceleration and Work Done

Metabolic Equivalent:

A unit of metabolic equivalent, or MET, is defined as the ratio of a person's working metabolic rate relative to the resting metabolic rate. METS values correlate with oxygen requirements. Starting with 1, which is the least amount of activity (such as resting), the values increase with the amount of activity. Here the calories are estimated based on metabolic equivalent (METS) conversion method.

MET = Working Metabolic Rate / Resting Metabolic Rate

For each activity, there is a corresponding MET value. The MET value of Idle, walking and running is given in the below table.

Table 3-5 The Metabolic Equivalent (MET) for activities

Context	METS
Idle [Note2]	0.9
Walking [Note1]	0.0272 * Walking Speed(m/min) + 1.2
Running [Note1]	0.093 * Running Speed(m/min) - 4.7

Note1: This is referred from a research paper on "Monitoring_Daily_Energy_Expenditure_using_a_3-Axis_Accelerometer_with_a_Low-Power_Microprocessor.pdf" (Refer to "6 References".)

Note2: This is referred from "Compendium of physical activities: an update of activity codes and MET intensities" (Refer to "6 References".) where sleeping MET is 0.9. Since 0.9 is the lowest METS value, for Idle state (comprises of Sleeping, Sitting and Standing) 0.9 is used as METS value.

The energy (Calories burnt) can be calculated from the MET value by using below formula,

Energy (kcal) = 1.05×METS×Weight (kg) ×Exercise time (hr)

The following values are used for calculating energy.

The default human weight = 50 [kg]

The standard step width = 0.74 [m]

4. Libraries

4.1 API List

Each API functions' role is shown in the below table:

Table 4-1 API List

Functional Name	Function
R_ALM_StartSampling(void)	Starts the operation of MCU peripheral devices.
R_ALM_StopSampling(void)	Stops the MCU peripheral devices' operation. It is used only if required.
R_ALM_Calculate(void)	Resets sampling complete flag. Calculates the Step Counts and calorie burnt values and stores the values in the respective global variables.
R_ALM_Calculation_Init(void)	Initializes step_count result and calorie burnt result values.
R_ALM_IsSampleDataReady(void)	Returns the Sampling complete flag. If the flag is set, the function returns true indicating samples are ready for calculation, otherwise returns false.

4.2 Global Variables

The Step Counts and Calories burnt values are stored in the below global variables:

Table 4-2 List of Global Variables

Global Variables	Function
g_step_count	integer value indicating number of step counts [step counts]
g_calorie_burnt	float value indicating the calories burnt [kcal]
g_bytes_read	Size of FIFO data sampled
g_sample_buffer[]	sampled data buffer arranged each axis by axis

4.3 Memory Size

Table 4-3 Memory Size

Memory	Size
ROM	3,804 bytes for ALM library [Note1]
RAM	2,156 bytes for ALM Library [Note1]
User Stack	256 bytes
Interrupt Stack	82 bytes

Note1: Refer to "5.2.2 Software Architecture Overview". It does not include device drivers.

5. Applications

5.1 Hardware

5.1.1 Hardware Design Policy

Hardware design policy of the ALM is

- The Hardware components used in the system shall be commercially available in the market. This helps the user to reproduce the development environment easily.
- Accelerometer ADXL362, an ultralow power, 3-axis MEMS accelerometer that consumes less than 2 µA at a 100 Hz output data rate, 2.0 V supply and 270 nA when in motion triggered wake-up mode is used for activity detection. In addition to its ultra-low power operation, it provides below features which makes it suitable for the activity detection:
- High resolution: 1 mg/LSB
- Adjustable threshold sleep/wake modes for motion activation
- Autonomous interrupt processing, without need for microcontroller intervention
- Deep embedded FIFO minimizes host processor load
- Acceleration sample synchronization via external trigger – host can control the sampling timing
- A Bi-directional Voltage level translator circuit is inserted between the RX231 RSK and the ADXL362 to give an external clock to the ADXL362 to maintain accuracy and to translate the voltage levels of the interface signals to the power supply voltage.

5.1.2 Hardware Block Diagram

The ALM will be developed to run on RX231 RSK (Renesas Starter Kit for RX231) and Accelerometer ADXL362.

The Block Diagram for the overall system is shown below:

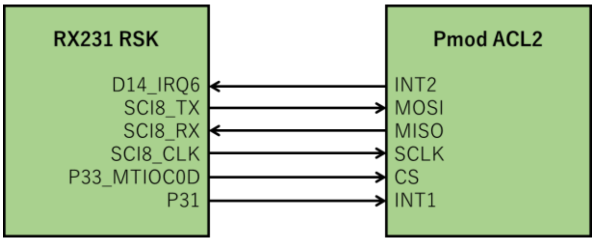


Figure 5-1 Hardware Block Diagram

The Specifications of the RX231 RSK and ADXL362 Accelerometer are listed in the below tables respectively:

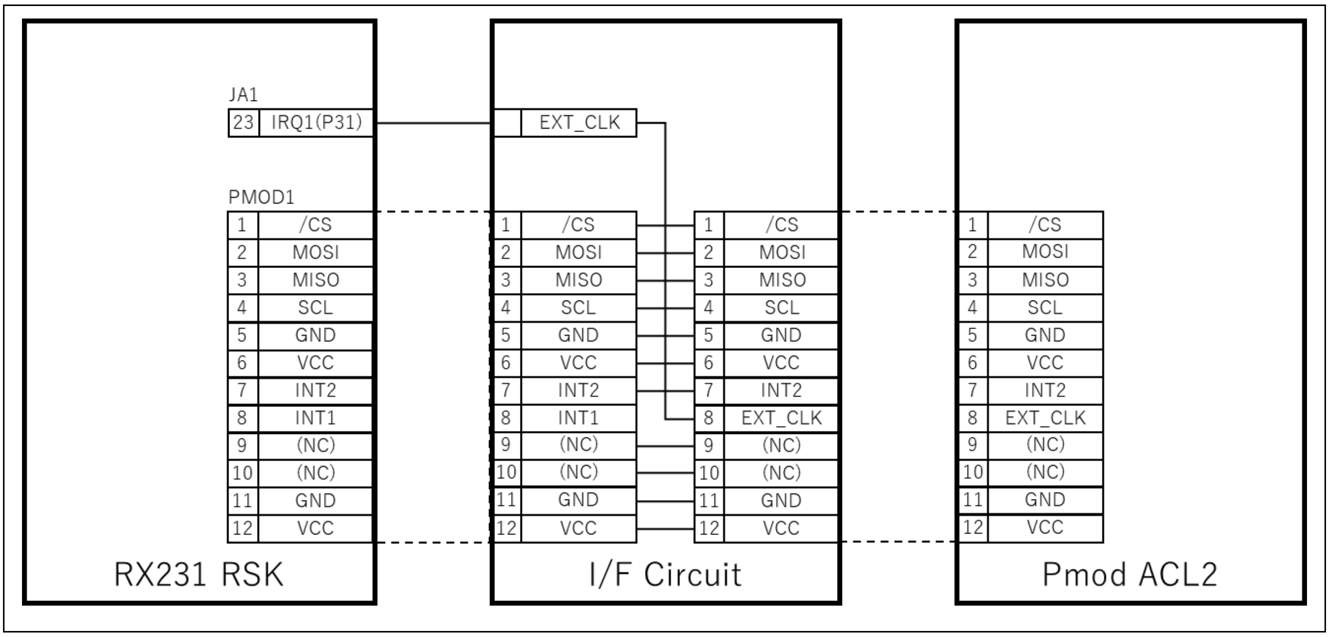


Figure 5-2 Hardware Wiring Chart

5.1.3 RX231 MCU Digital Interconnect

The functional block diagram for the ALM library with the RX231 peripheral blocks is shown below:

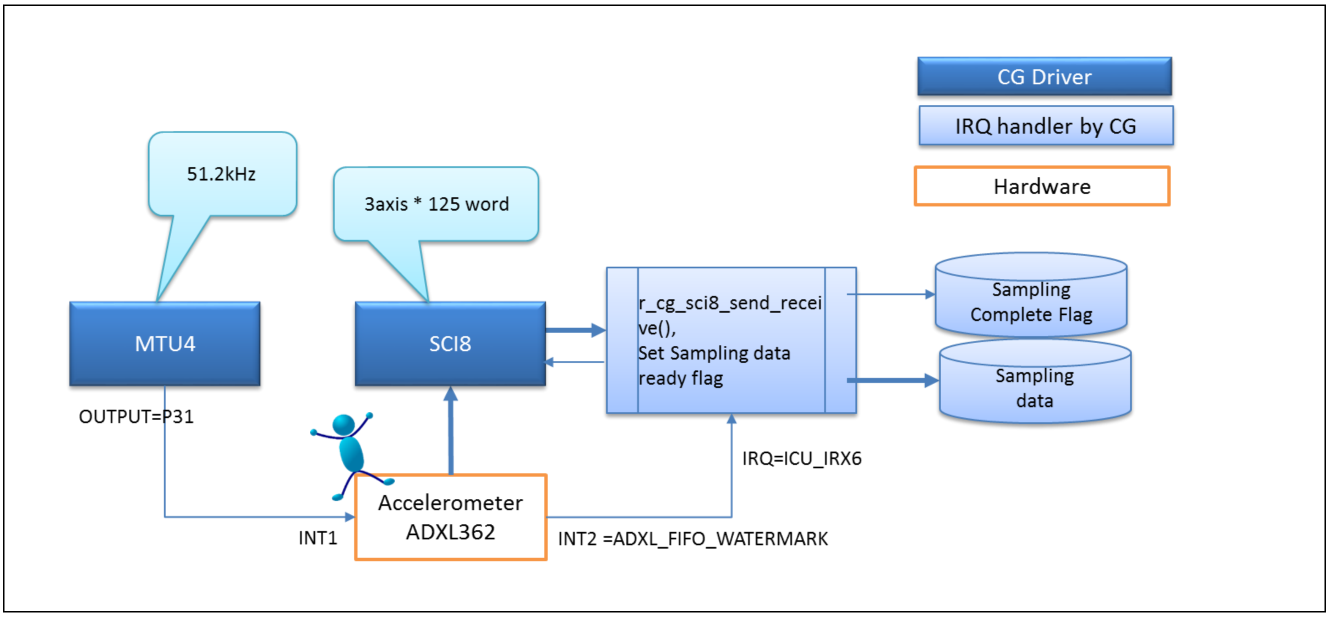


Figure 5-3 ALM Digital Interconnect

- The EXTCLK for ADXL362 is oscillated by MTU4 which creates oscillation of 51.2 kHz.
- ADXL362 will be configured with Range= ±2G, Sampling Rate= 12.5Hz.
- ADXL362 will be configured to use X, Y and Z axis value. T (Temperature) is not used.

- ADXL362 will use FIFO for storing Data, WatermarkIRQ (10sec) for raising interrupt and will use ETXCLK for data Synchronization.
- Set Power Mode = Measure.
- ADXL362 is connected to SCI8 channel.
- When WatermarkIRQ occurs, SCI8 sends Read FIFO Command to ADXL362.
- On reading FIFO sampling complete flag is set.

5.2 Software

5.2.1 Software Design Policy

The Software Design Policy is:

- ALM Software is designed for Non-OS environment
- Code Generator Utility is used to create device drivers and hence the software framework is driven by the Code Generator Utility
- The Accelerometer Data Sampling is automatically started on software initialization, immediately after all the peripherals are initialized and configured
- The sensor data is sampled continuously unless the system is powered off. The sampling can be stopped temporarily during the calculation, if needed to get better accuracy
- The data sampling and calculation of the Step Counts and Calories Burnt is executed continuously using an infinite loop (while(1U) {}) in the main () function
- The calculate () function calculates Step Counts and Calories burnt and stores the calculated values in global variables
- The measured values are not displayed on any display or stored in any file
- The software does not include power management
- The ALM Software Architecture is designed to be a simple Layered Architecture where each layer exposes a set of APIs to the layer above it. The Software Layered Architecture is described in the next section

5.2.2 Software Architecture Overview

The Software architecture is a layered architecture, where each layer provides a set of APIs for the above layers to access it.

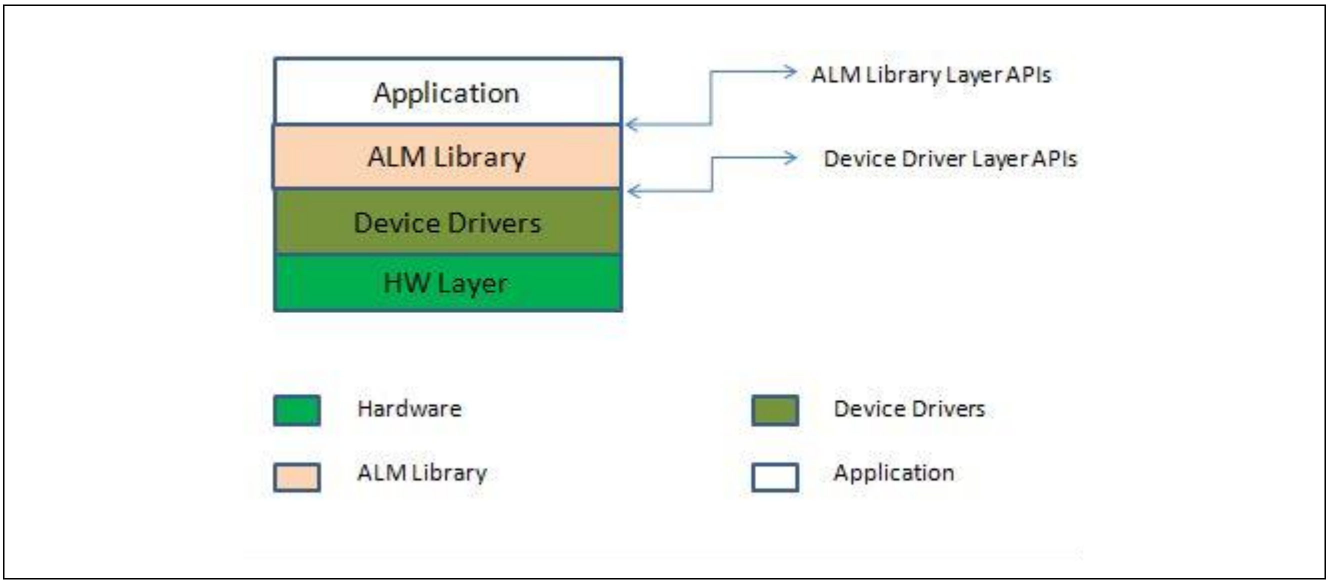


Figure 5-4 ALM Software Architecture

5.2.3 Measurement Signal Flow

The Activity Count is measured using ADXL362 to indicate the step counts and calories burnt. The activity count measurement flow is shown in the diagram below:

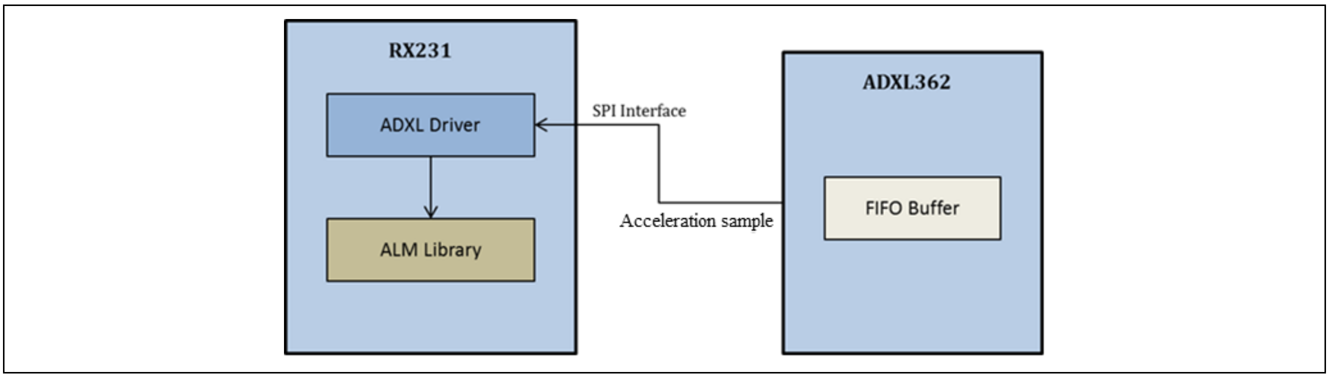


Figure 5-5 Interface of the ADXL362 Sensor with the MCU

5.2.4 Measurement Control Flow

The ALM Software implements a simple application to use the ALM Library. The application is automatically started on reset. The Activity Level data sampling w done continuously until the device is powered off.

The Control Flow of the main application is shown in the flow chart below:

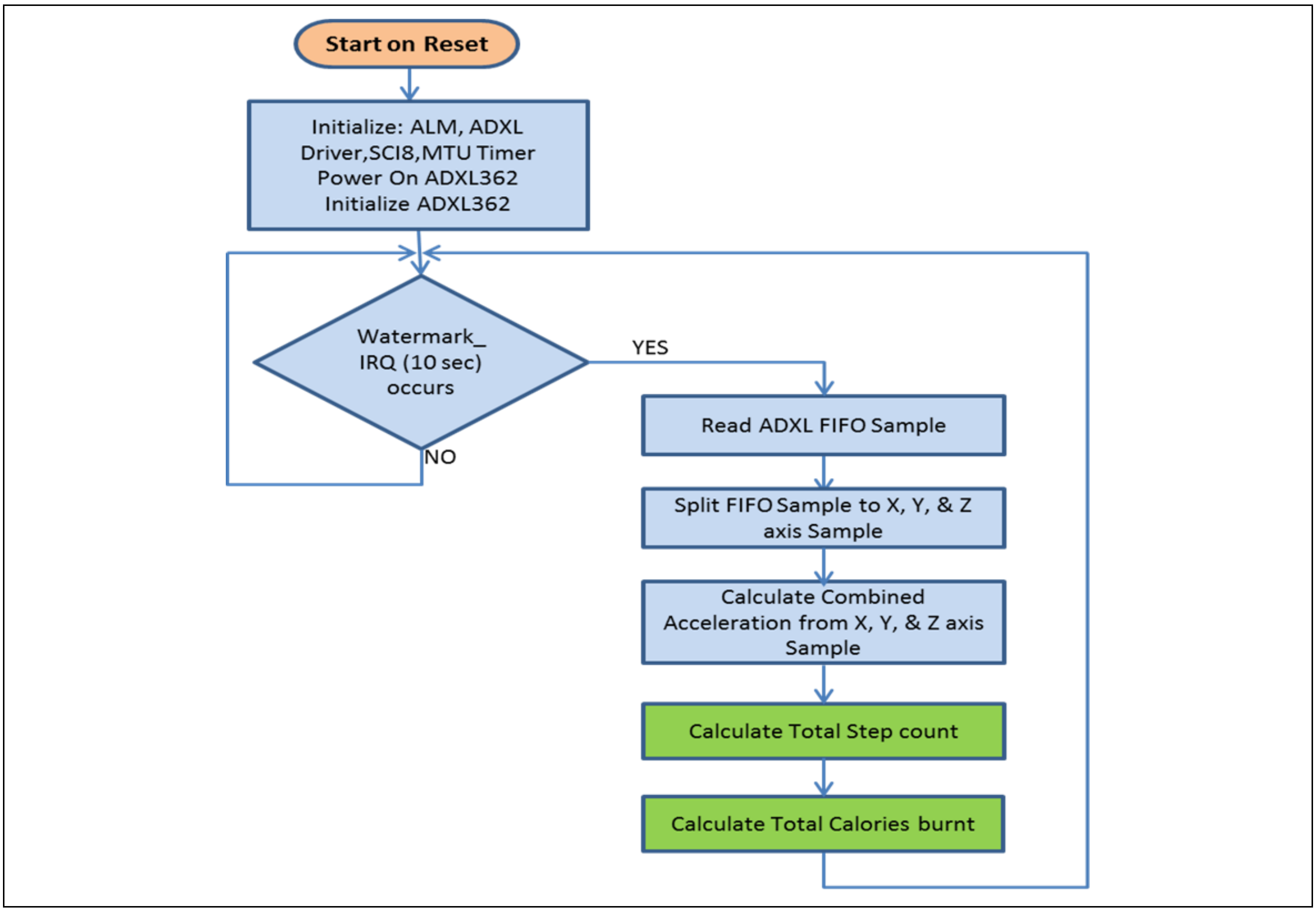


Figure 5-6 ALM Software Control Flow

5.3 Device Drivers

5.3.1 ADXL driver

(1) **Driver Function List**

Each API functions' role is shown in the below table:

Table 5-1 Driver Function List

Functional Name	Function
R_ADXL362_Initialize(void)	Initializes the accelerometer sensor (ADXL362) by setting range, sampling rate, watermark interrupt etc.
R_ADXL362_Fetch(void)	Receives the sample data through SCI and Sets the sampling complete flag.

(2) **Conditions**

For ADXL Configuration, refer to "5.1.3 RX231 MCU Digital Interconnect".

5.3.2 Code Generator

Table 5-2 shows the used peripheral function.

Table 5-3 shows the functions generated by Code Generator.

Table 5-2 Used Peripheral function List

Peripheral function	Macro	Sub Macro	Setting	Status
Clock Generator	CGC		VCC setting	2.7 (V) = VCC = 5.5 (V)
			Main clock oscillation source	Resonator
			Main clock oscillation source Frequency	8(MHz)
			Oscillator wait time	8192cycles2048 (µs)
			Oscillation stop detection function	Disabled
			PLL circuit setting	
			Input frequency division ratio	x 1/2
			Frequency multiplication factor	x 8
			PLL Frequency	32 (MHz)
			Sub-clock oscillator drive capacity	Drive capacity for low CL
			Sub-clock oscillator and RTC (RTCSCLK) setting	32.768 (kHz)
			Low speed clock oscillator (LOCO) setting	4 (MHz)
			Clock source	Main clock oscillator
			System clock (ICLK)	x 18 (MHz)
			Peripheral module clock (PCLKA)	x 18 (MHz)
			Peripheral module clock (PCLKB)	x 18 (MHz)
			Peripheral module clock (PCLKD)	x 18 (MHz)
			External bus clock (BCLK)	x 18 (MHz)
			Flash IF clock (FCLK)	x 18 (MHz)

Interrupt Controller Unit	ICU	IRQ6	Pin	PE6
			Digital filter	No filter 0(MHz)
			Valid edge	Falling
			Priority	Level 15 (highest)
I/O Ports	Port3	P33	Mode	Out
			CMOS output	Used
			output value	1
Multi-Function Timer Pulse Unit 2	MTU2_U0	MTU4	Counter clock selection	PCLK/4
			Clock edge setting	Rising edge
			Counter clear source	TGRD4 compare match/input capture
			TGRA4 (Output compare register)	9.765625μs, (Actual value: 10)
			TGRB4 (Output compare register)	9.765625μs, (Actual value: 10)
			TGRC4 (Output compare register)	9.765625μs, (Actual value: 10)
			TGRD4 (Output compare register)	19.5μs, (Actual value: 19.5)
			MTIOC4A pin (PA0)	MTIOC4A pin output disabled
			MTIOC4B pin (P30)	MTIOC4B pin output disabled
			MTIOC4C pin (PB1)	MTIOC4C pin output disabled
			MTIOC4D pin (P31)	Initial output of MTIOC4D pin is 0. Toggle output at compare match.
			Enable TGRA4 input capture/compare match interrupt (TGIA4)	Used
			(TGIA/TGIB/TGIC/TGID) Priority	Level 15 (highest)
Serial Communications Interface	SCI8		Function setting	Simple SPI bus (Master transmit/receive)
			SMOSI8	PC7
			SMISO8	PC6
			Transfer direction setting	MSB-first
			Data inversion setting	Normal
			Transfer clock	Internal clock
			Bit rate	5000 (bps)
			SCK8 pin function selection	Clock output
			SCK8	PC5
			Clock delay	Clock is not delayed
			Transmit data handling	Data handled in interrupt service routine
			Receive data handling	Data handled in interrupt service routine
			TXI8, RXI8, TEI8, ERI8 priority	Level 15 (highest)
			Enable error interrupt (ERI8)	Used
			Transmission end	Used
			Reception end	Used
			Reception error	Used

Table 5-3 Functions Generated by Code Generator

Peripheral function	File	Macro	Function
Common	r_cg_main.c		void main(void)
			void R_MAIN_UserInit(void)
	r_cg_dbsct.c		-
	r_cg_intprg.c		void r_privileged_exception(void)
			void r_floatingpoint_exception(void)
			void r_access_exception(void)
			void r_undefined_exception(void)
			void r_reserved_exception(void)
			void r_nmi_exception(void)
			void r_brk_exception(void)
	r_cg_resetprg.c		void PowerON_Reset_PC(void)
	r_cg_sbrk.c		-
	r_cg_vecttbl.c		-
	r_cg_sbrk.h		-
	r_cg_stacksct.h		-
	r_cg_vect.h		-
	r_cg_hardware_setup.c		void R_Systeminit(void)
			void HardwareSetup(void)
	r_cg_macrodriver.h		-
	r_cg_userdefine.h		-
Clock Generator	r_cg_cgc.c		void R_CGC_Create(void)
	r_cg_cgc_user.c		-
	r_cg_cgc.h		-
Interrupt Controller Unit	r_cg_icu.c		void R_ICU_Create(void)
		IRQ6	void R_ICU_IRQ6_Start(void)
			void R_ICU_IRQ6_Stop(void)
	r_cg_icu_user.c	IRQ6	static void r_icu_irq6_interrupt(void)
	r_cg_icu.h		-
I/O Ports	r_cg_port.c		void R_PORT_Create(void)
	r_cg_port_user.c		-
	r_cg_port.h		-
Multifunction timer pulse unit 2	r_cg_mtu2.c		void R_MTU2_Create(void)
		MTU4	void R_MTU2_C4_Start(void)
			void R_MTU2_C4_Stop(void)
	r_cg_mtu2_user.c	MTU4	static void r_mtu2_tgia4_interrupt(void)
	r_cg_mtu2.h		-
Serial Communication Interface	r_cg_sci.c	SCI8	void R_SCI8_Create(void)
			void R_SCI8_Start(void)
			void R_SCI8_Stop(void)
			MD_STATUS R_SCI8_SPI_Master_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, uint8_t * const rx_buf, uint16_t rx_num)
	r_cg_sci_user.c	SCI8	static void r_sci8_transmit_interrupt(void)
			static void r_sci8_transmitend_interrupt(void)
			static void r_sci8_receive_interrupt(void)
			static void r_sci8_receiveerror_interrupt(void)
			void r_sci8_callback_transmitend(void)
			void r_sci8_callback_receiveend(void)
			void r_sci8_callback_receiveerror(void)
	r_cg_sci.h		-

For details, refer to the following files.

- Function.html

  It is stored in the “an-r11an0323ej0100-bsspf-apl/workspace/ALM/doc” folder
- Macro.html

  It is stored in the “an-r11an0323ej0100-bsspf-apl/workspace/ALM/doc” folder

5.4 Application Framework

The application framework generated by Code Generator is shown in the flow chart below:

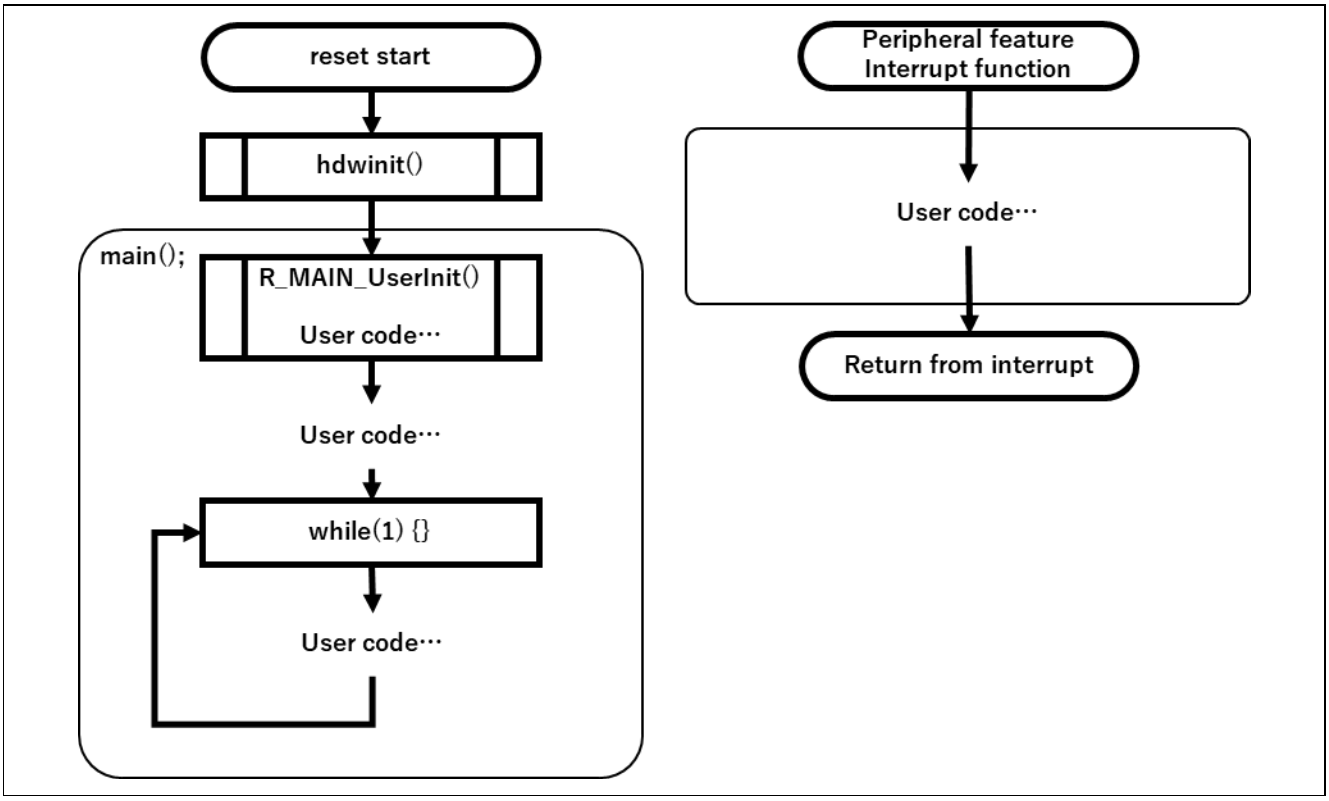


Figure 5-7 Application Framework by the Code Generator

The application program is constructed on this framework generated by the Code Generator.

In the main routine, all the peripherals of the MCU will be initialized before the main () function is executed. The hdwinit () is automatically invoked from the compiler on reset and all the peripherals are initialized according to the Code Generator Configuration.

The main () function will have three sections:

1. R_MAIN_UserInit () – This function is invoked in the beginning of main () function. User code to initialize all the devices outside the MCU, such as ADXL362 will be implemented in R_MAIN_UseInit () function.
2. User Code Section - User code to start each of the device operation will be implemented here.
3. While loop – main () function in the application framework will have an infinite while loop. User code can be implemented in this loop.

The Code Generator generates Peripheral Interrupt Framework for each of the Interrupt configured in the Code Generator. The user code to handle each of the enabled interrupts shall be implemented in the respective interrupt template generated by the code generator.

5.5 Application Flow

The application flow is shown in the flow chart below:

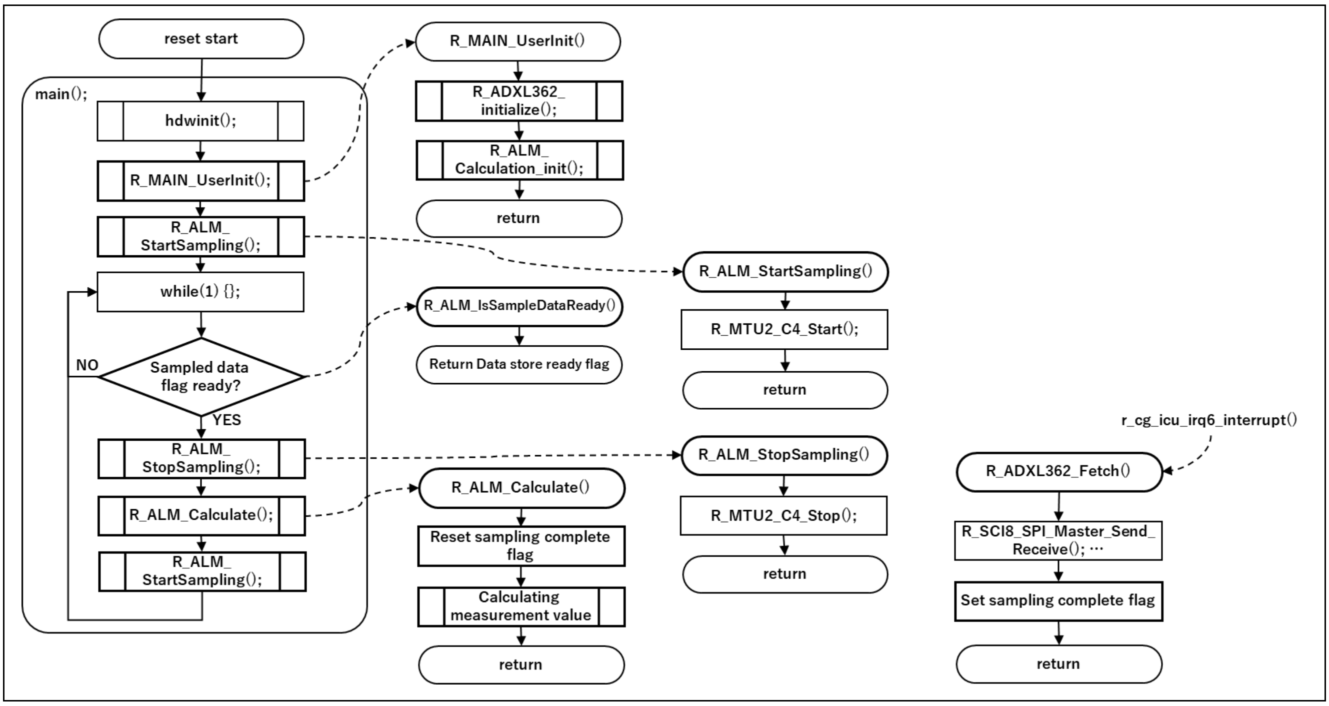


Figure 5-8 Software Flow Chart

The application does the below operations:

- Initialize the MCU, Peripherals and the ADXL362 on reset.
- Do below operations repeatedly

  Start Data Sampling

  Wait for sampling complete flag to be set (sampled data ready)

  Stop Sampling once the sampled data flag is ready, if required.

  Reset Sampling complete flag

  Calculate Step Counts and Calories burnt from the collected samples

  Start Data Sampling again

5.6 File Configurations

Figure 5-9 and Figure 5-10 show the file structure.

```
an-r11an0323ej0101-bsspf-apl
|     r11an0323ej0101-bsspf.pdf                       : This application note
|
+--- workspace                                        : Workspace folder
      |
      +--- ALM                                        : Project folder
            |     .cproject                           : ProjectDescription
            |     .project                            : ProjectDescription
            |     SM HardwareDebug.launch             : Launch Configuration
            |
            +--- .settings                            : Configuration folder of e2studio (Omit details)
            |
            +--- demo
            |          r11an0323ej0101-bsspf-alm.zip  : Archived file of this project
            |
            +--- doc
            |     |     Function.html                 : Function Table file for CG
            |     |     Macro.html                    : Macro Table file for CG
            |     |
            |     +--- Macro.files                    : Macro.files folder (Omit details)
            |
            +--- generate                             : generate folder
            |          iodefine.h                     : IO definition file
            |
            +--- src                                  : Source folder
                  |
                  +--- cg_src                         : CG Source folder
                  |          r_cg_cgc.c
                  |          r_cg_cgc.h
                  |          r_cg_cgc_user.c
                  |          r_cg_dbsct.c
                  |          r_cg_hardware_setup.c
                  |          r_cg_icu.c
                  |          r_cg_icu.h
                  |          r_cg_icu_user.c
                  |          r_cg_intprg.c
                  |          r_cg_macrodriver.h
                  |          r_cg_main.c
                  |          r_cg_mtu2.c
                  |          r_cg_mtu2.h
                  |          r_cg_mtu2_user.c
                  |          r_cg_port.c
                  |          r_cg_port.h
                  |          r_cg_port_user.c
                  |          r_cg_resetprg.c
```

Figure 5-9 File Structure (1/2)

```
|           r_cg_sbrk.c
|           r_cg_sbrk.h
|           r_cg_sci.c
|           r_cg_sci.h
|           r_cg_sci_user.c
|           r_cg_stacksct.c
|           r_cg_userdefine.h
|           r_cg_vect.h
|           r_cg_vecttbl.h
|
+--- r_alm_src                                        : ALM Source folder
     |     r_alm_if.h
     |
     +--- src
                r_alm.c
                r_alm_cgw.h
                r_alm_driver.c
                r_alm_driver.h
                r_alm_private.h
```

Figure 5-10 File Structure (2/2)

5.7 System Requirement

The following are required for the execution of the sample project:

- e2studio version: 6.2.0 or above
- RX Family C/C++Compiler Package version: CC-RX 2.07.00 or later
- Language Configuration: C(C99) (-lang=c99)

5.8 Procedure to Execute the Sample Application

The following describes the steps to build and execute the sample application.

(1) **Import the sample project into e2studio workspace by clicking "Import" from File Menu.**

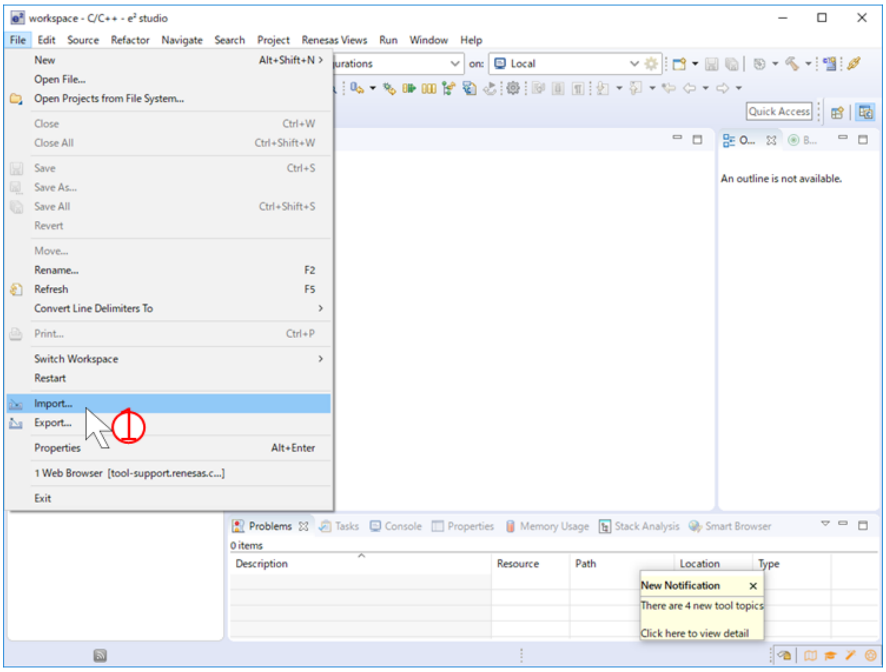


Figure 5-11 Select "Import" Menu

(2) **Select "Existing Projects into Workspace".**

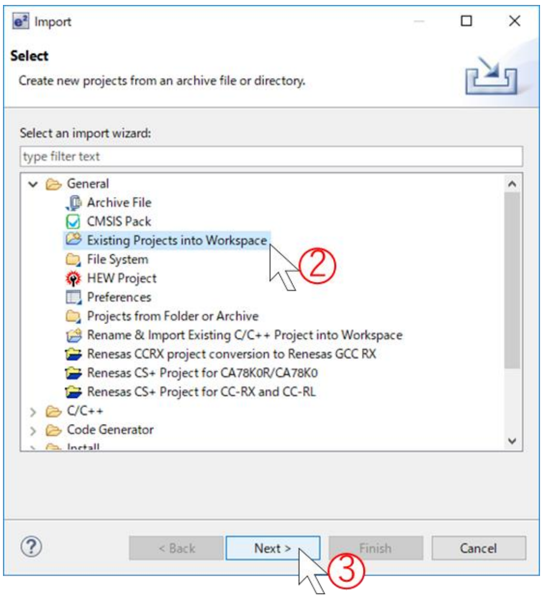


Figure 5-12 Select "Existing Projects into Workspace"

(3) **In the next window, choose "Select archive file:" and browse to the directory of "r11an0323ej0101-bsspf-alm.zip".**

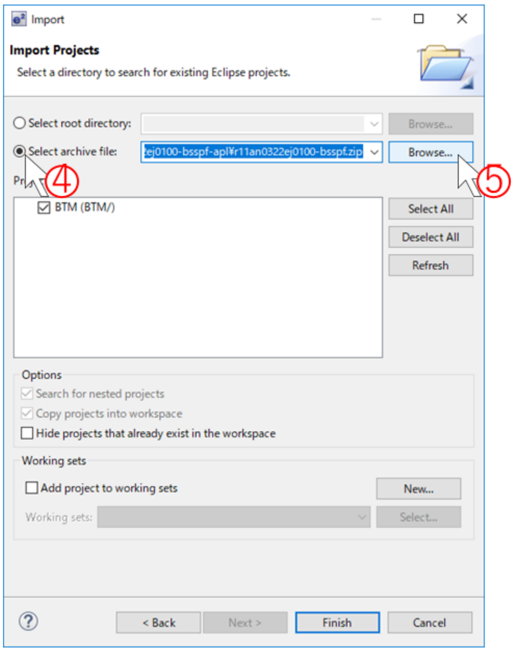


Figure 5-13 Choose "Select archive file:" and Select the Archived File

(4) **After selecting the archive file, the projects it contains will be listed down as shown. Click "Finish" to finish the importing.**

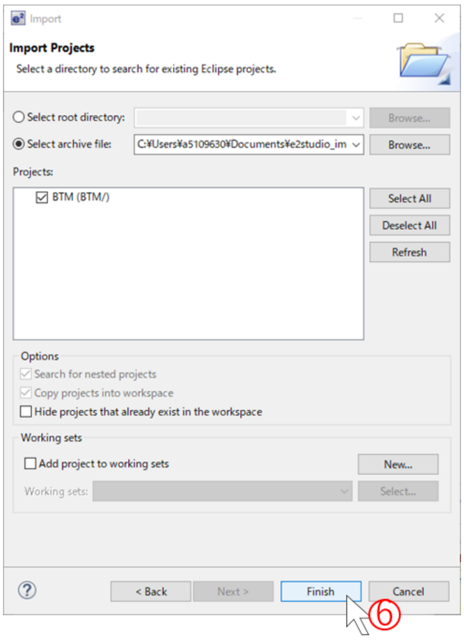


Figure 5-14 Click "Finish"

6. References

- User's Manual for RX231:

  The latest version can be downloaded from the Renesas Electronics website.
- User's Manual for Renesas Starter Kit for RX231 (R0K505231S020BE):

  The latest version can be downloaded from the Renesas Electronics website.
- Pmod ACL2:

  https://store.digilentinc.com/pmod-acl2-3-axis-mems-accelerometer/
- ADXL362 Datasheet:

  http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf
- Monitoring_Daily_Energy_Expenditure_using_a_3-Axis_Accelerometer_with_a_Low-Power_Microprocessor.pdf:

  https://www.researchgate.net/publication/220366667_Monitoring_Daily_Energy_Expenditure_using_a_3-Axis_Accelerometer_with_a_Low-Power_Microprocessor
- Compendium of physical activities: an update of activity codes and MET intensities

  https://www.ncbi.nlm.nih.gov/pubmed/10993420

  (U.S. National Library of Medicine)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Pmod^{TM} is a trademark of Digilent Inc.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul 31, 2018	-	1st Released
1.01	Sep 14, 2018	23 to 24	Changed “5.6 File Configurations”.
		24	Added “5.7 System Requirement” and “5.8 Import procedure”.
		25 to 26	Updated SALES OFFICE page.

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

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

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.2