RL78/I1C(512KB) Fast Prototyping Board Quick Start Guide (Software)

Quick Start Example Project

Table of Contents

Table of Figures

Table of Tables

1. Introduction

This Quick Start Guide (Software) provides:

- An overview of the Quick Start Example Project for the RL78/I1C (512KB) Fast Prototyping Board (FPB).
- Instructions for powering, connecting, and running the Quick Start Example Project.
- Instructions for creating, modifying, and building the Quick Start Example Project using the CS+ Integrated Development Environment (CS+ IDE).

1.1. Assumptions and Advisory Notes

1. Tool experience: It is assumed that the user has prior experience working with IDEs such as CS+ and terminal emulation programs such as Tera Term.
2. Subject Knowledge: It is assumed that the user has basic knowledge about microcontrollers, embedded systems, and Code Generator in CS+ to create and modify the example project as described in this document.
3. The screenshots provided throughout this document are for reference. The actual screen content may differ depending on the version of software and development tools.

2. Required Environments

Hardware Requirements:

1. RL78/I1C (512KB) Fast Prototyping Board [RTK5RL10N0CPL000BJ]
2. Coin-cell battery [CR2032 (3V)]
3. Micro USB Device Cable
4. PC with at least 1 USB port

Software Requirements:

- Windows® 10 operating system
- USB Serial Drivers (included in Windows 10)
- Tera Term (or similar) terminal console application
- CS+ Ver. 8.05.00 (or above)

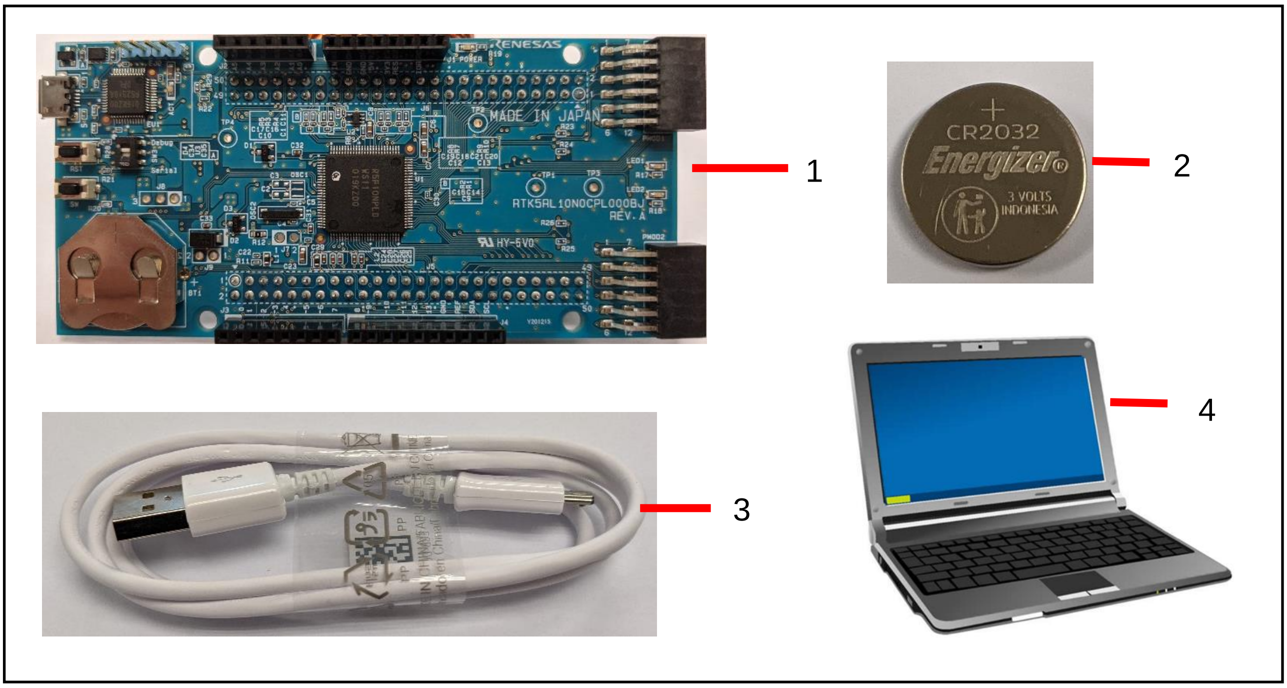


Figure 1: Hardware components

3. Overview of Quick Start Example Project

The Quick Start Example Project allows the user to:

- Display and set the Real-Time Clock (RTC), and
- Toggle the on-board LED 1 and 2 through UART.
- Toggle LED 2 by pressing **SW**.

When running the Quick Start Example Project through a “Serial communication” with a PC via USB port, the RTC and LEDs can be controlled through the terminal console.

3.1. Quick Start Example Flowchart

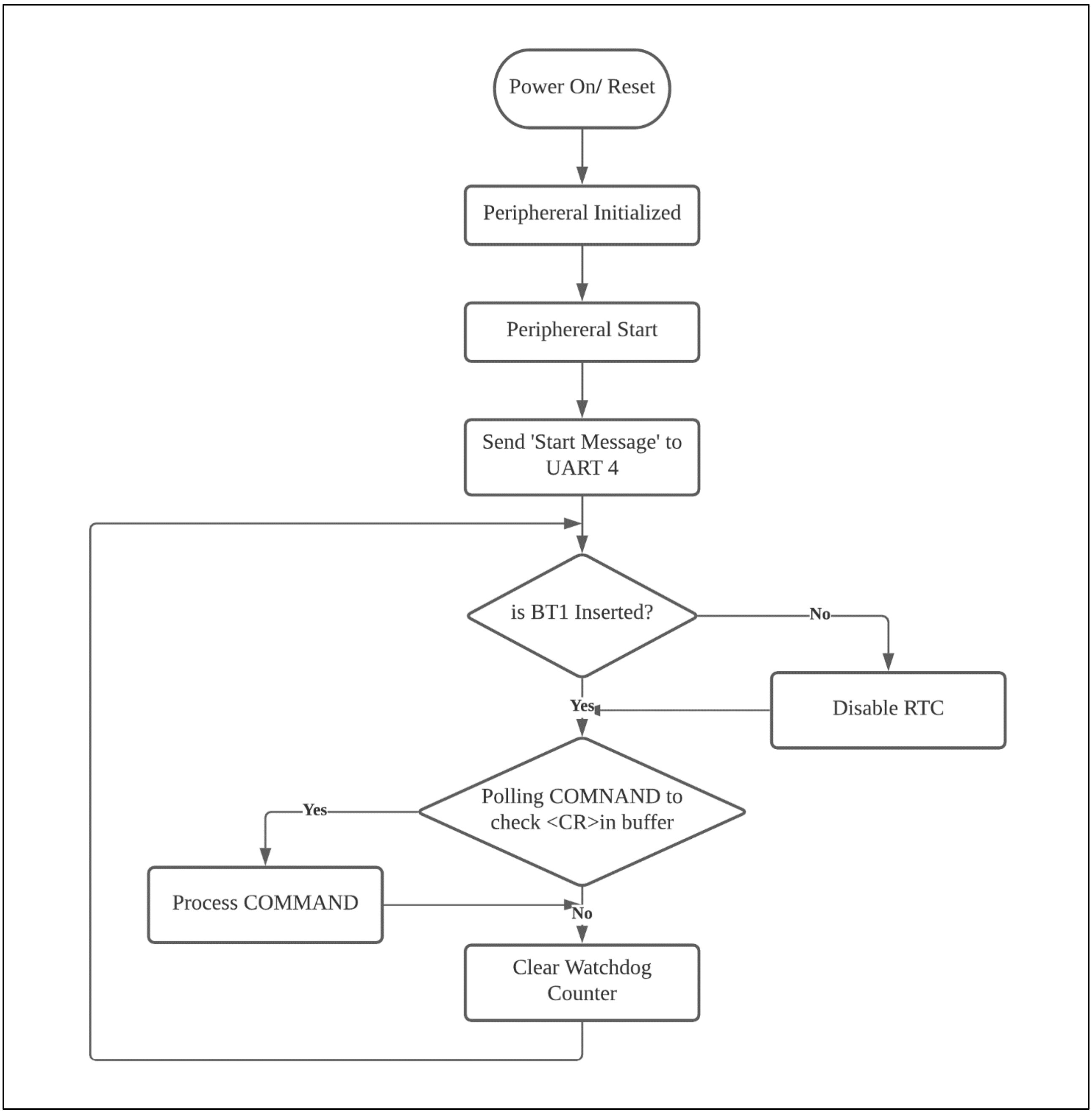


Figure 2: FPB Quick Start Example Project Flowchart

3.2. Quick Start Example Project State Diagram

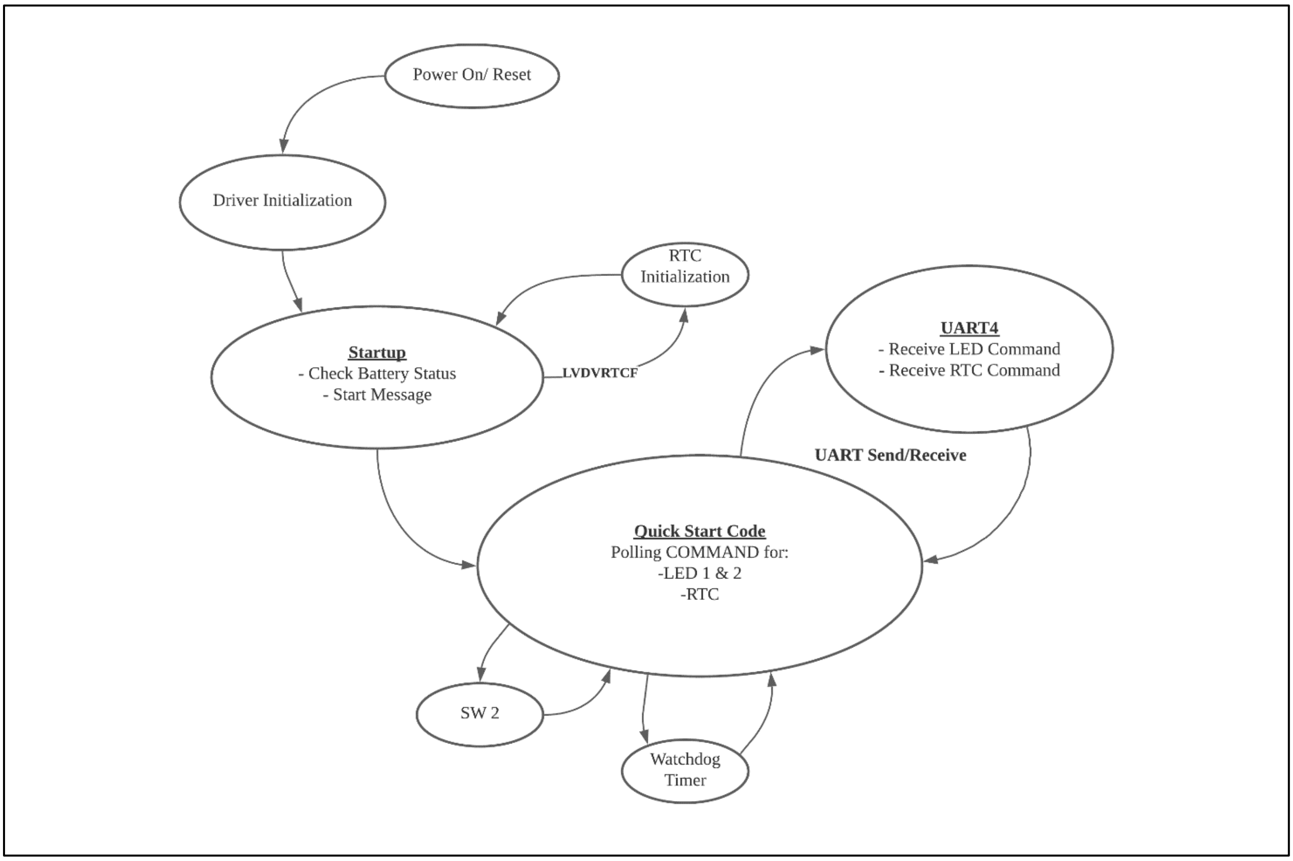


Figure 3: FPB Quick Start Example Project State Diagram

4. Running the Quick Start Example Project

This section lists the instructions to power, communicate and program the RL78/I1C (512KB) FPB, and run the Quick Start Example Project.

4.1. RTK5RL10N0CPL000BJ Board Outline

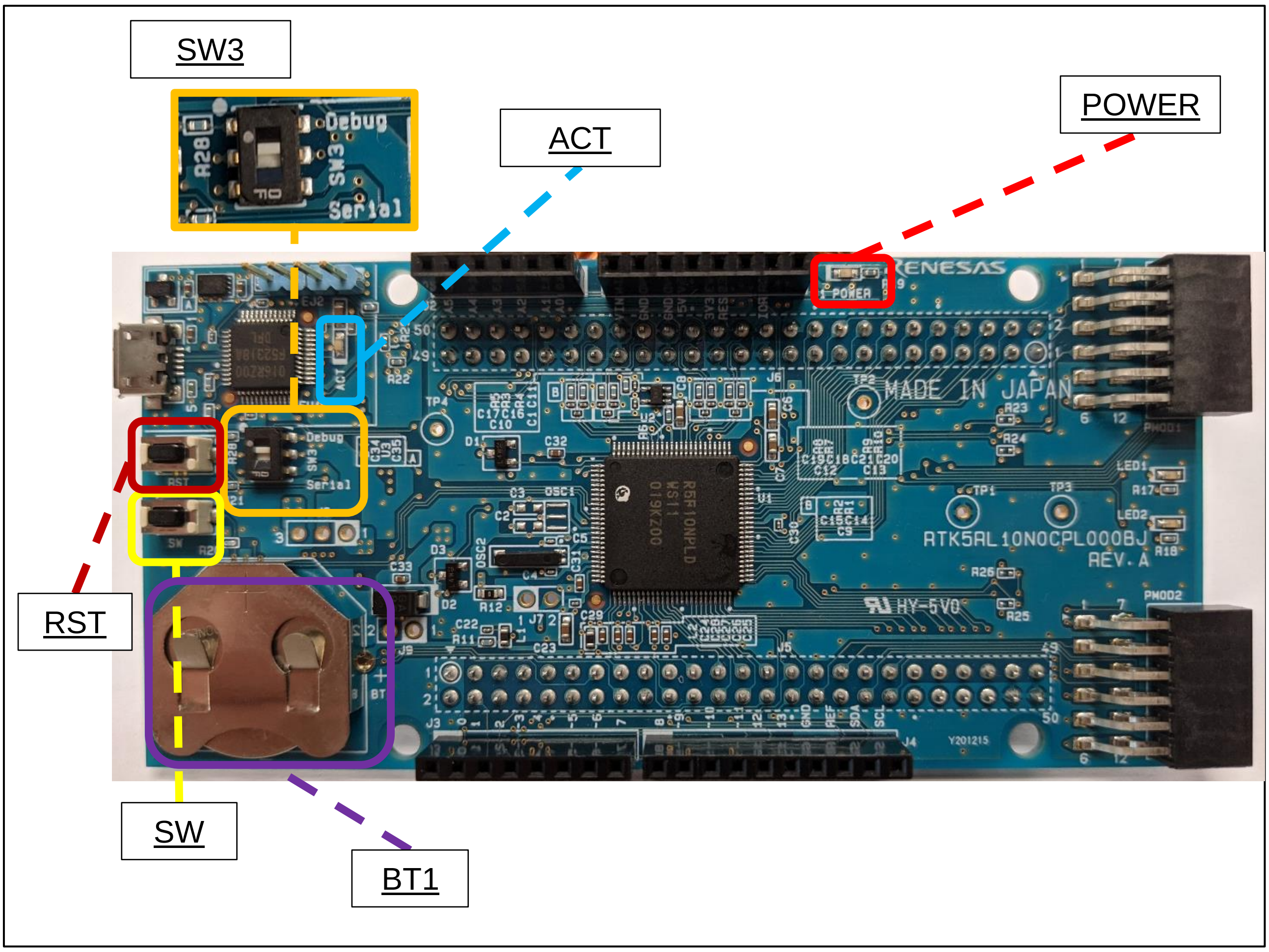


Figure 4: Outline of Board when Connecting to Host PC

DIP switch (SW3) determines the following communication method:

- **Debug**: Programming and Running the board through a debugger
- **Serial**: Communicating with the board on a terminal console application

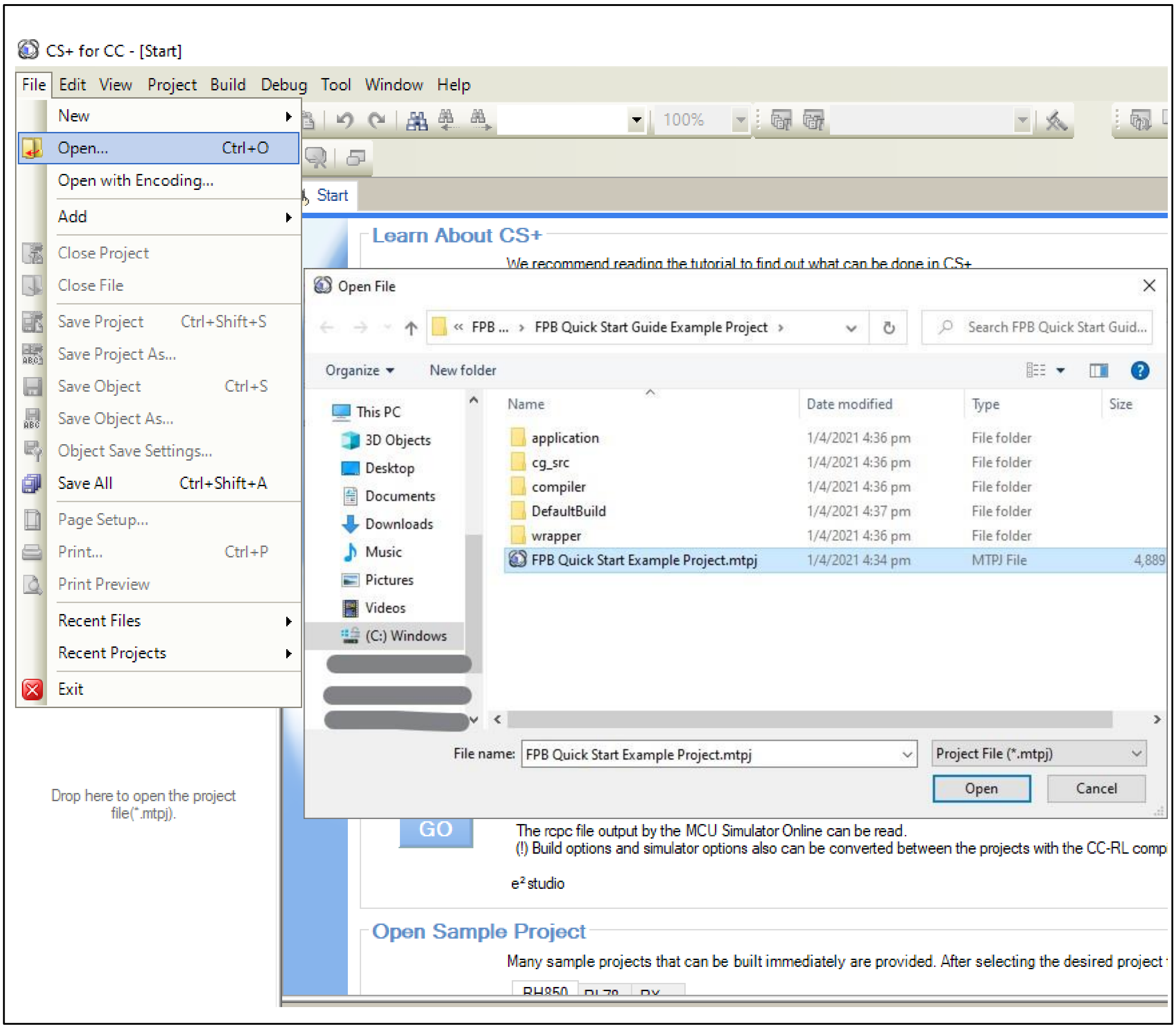
4.2. Connecting and Running the Quick Start Example Project with a Debugger

1. Set the on-board dip switch (SW3) into “**Debug**” and connect the Micro USB cable into the Micro USB connector (EJ1) on the board.
2. Connect the other end of the Micro USB cable (USB Head) into the host PC. LED1 (ACT) will start blinking, indicating that the board is in “**Debug**” mode.
3. While LED3 (POWER) will light up solid green, indicating that the board is powered.

4.3. Downloading the Quick Start Example Project

1. Download the **r01an5860es0100-rl78i1c512fota-sample-project.zip** and extract the **FPB Quick Start Guide Example Project** folder to a known location.
2. Launch CS+ IDE.
3. Under **File**, **Open…**, browse to the **Quick Start Example Project** directory to locate the **FPB Quick Start Example Project.mtpj**, and **Open**.

Note: An example directory can be **e.g. C:\[PC Name]\[Username]\Desktop\FPB Test Code\FPB Quick Start Guide Example Project**



We recommend reading the tutorial to find out what can be done in CS+

Name	Date modified	Type	Size
application	1/4/2021 4:36 pm	File folder	
cg_src	1/4/2021 4:36 pm	File folder	
compiler	1/4/2021 4:36 pm	File folder	
DefaultBuild	1/4/2021 4:37 pm	File folder	
wrapper	1/4/2021 4:36 pm	File folder	
FPB Quick Start Example Project.mtpj	1/4/2021 4:34 pm	MTPJ File	4,889

File name: FPB Quick Start Example Project.mtpj

Project File (*.mtpj)

The rcpc file output by the MCU Simulator Online can be read.
(!) Build options and simulator options also can be converted between the projects with the CC-RL comp

e² studio

Many sample projects that can be built immediately are provided. After selecting the desired project

Figure 5: Opening the Quick Start Guide Example Project

4. Ensure the **Debug Tool** is set at **RL78 E2 Lite**. Else configure by right clicking **Debug tool**, **Using Debug Tool**, **RL78 E2 Lite.**

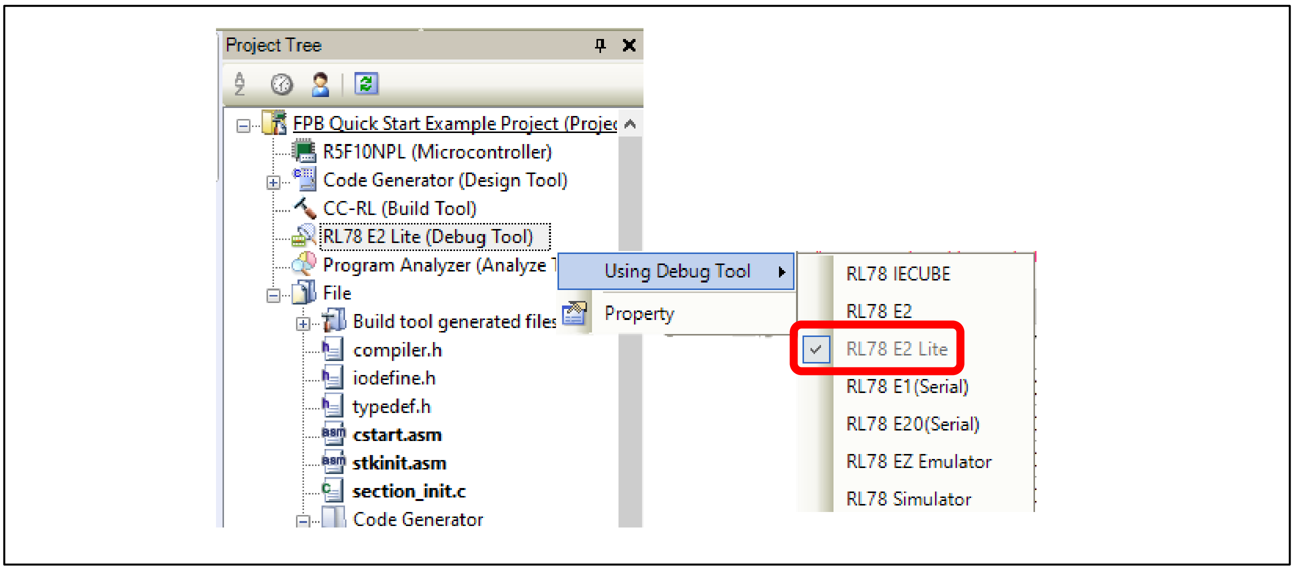
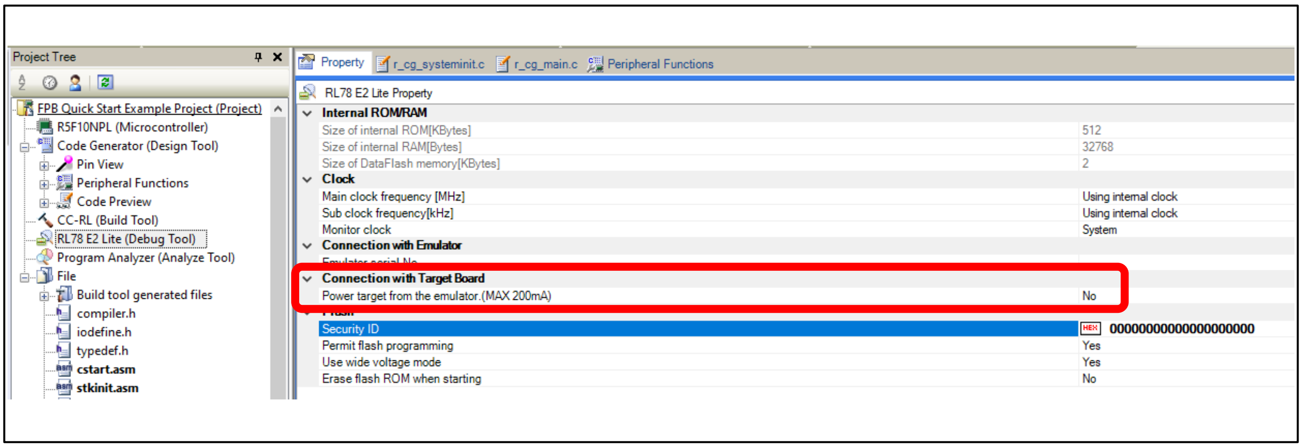


Figure 6: Configuring CS+ Debug Tool

5. Ensure **Power target from the emulator** is set to **No**, under **Connection with Target Board** to as shown below.



RL78 E2 Lite Property

Internal ROM/RAM
Size of internal ROM[KBytes] 512
Size of internal RAM[Bytes] 32768
Size of DataFlash memory[KBytes] 2

Clock
Main clock frequency [MHz] Using internal clock
Sub clock frequency[kHz] Using internal clock
Monitor clock System

Connection with Emulator

Connection with Target Board
Power target from the emulator.(MAX 200mA) No

Security ID HEX 00000000000000000000
Permit flash programming Yes
Use wide voltage mode Yes
Erase flash ROM when starting No

Figure 7: Configuring Power to Target Board

6. Build and Download the project into the FPB by clicking the **build & download** icon.

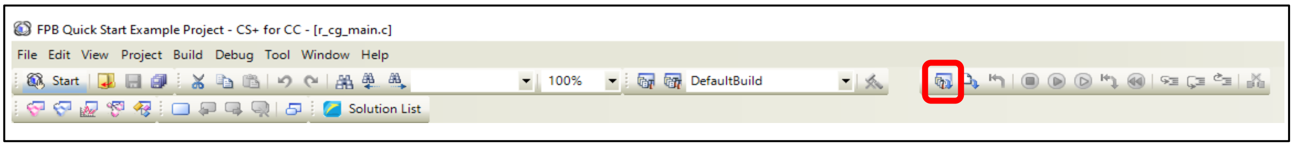
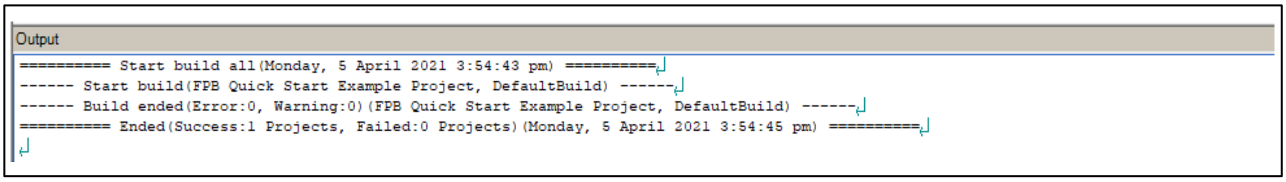

FPB Quick Start Example Project - CS+ for CC - [r_cg_main.c]
File Edit View Project Build Debug Tool Window Help
Start
100%
DefaultBuild
Solution List

Figure 8: Build and Download into FPB

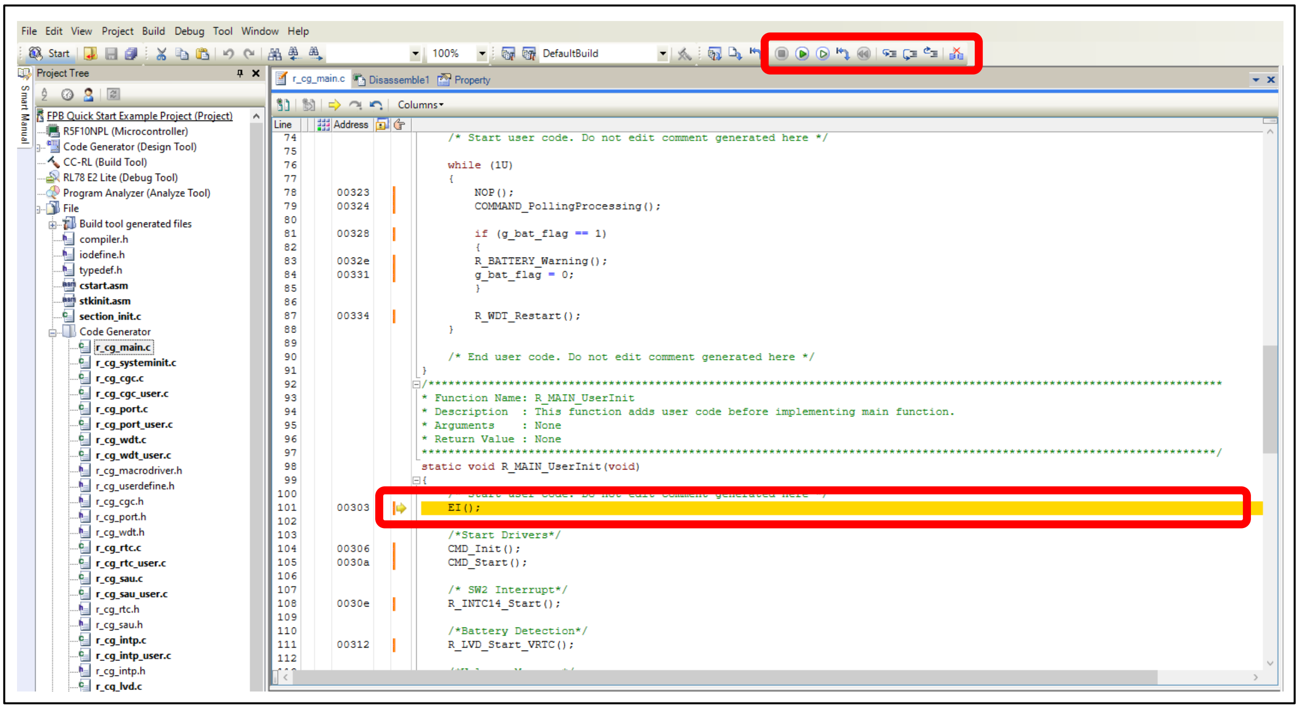
7. A successful build produces an output as follows:



```
========== Start build all(Monday, 5 April 2021 3:54:43 pm) ==========
------ Start build(FPB Quick Start Example Project, DefaultBuild) ------
------ Build ended(Error:0, Warning:0)(FPB Quick Start Example Project, DefaultBuild) ------
========== Ended(Success:1 Projects, Failed:0 Projects)(Monday, 5 April 2021 3:54:45 pm) ==========
```

Figure 9: Successful Build of Project

8. A successful download is shown as a **yellow start point** and an active **Debug tool bar.**



```
             /* Start user code. Do not edit comment generated here */

             while (1U)
             {
00323            NOP();
00324            COMMAND_PollingProcessing();

00328            if (g_bat_flag == 1)
                 {
0032e            R_BATTERY_Warning();
00331            g_bat_flag = 0;
                 }

00334            R_WDT_Restart();
             }

             /* End user code. Do not edit comment generated here */
         }
         /******************************************************************************************************
         * Function Name: R_MAIN_UserInit
         * Description  : This function adds user code before implementing main function.
         * Arguments    : None
         * Return Value : None
         ******************************************************************************************************/
         static void R_MAIN_UserInit(void)
         {
             /* Start user code. Do not edit comment generated here */
00303        EI();

             /*Start Drivers*/
00306        CMD_Init();
0030a        CMD_Start();

             /* SW2 Interrupt*/
0030e        R_INTC14_Start();

             /*Battery Detection*/
00312        R_LVD_Start_VRTC();
```

Figure 10: Successful download into FPB

9. The project is now downloaded into the FPB. Next, disconnect the FPB from the CS+ IDE by clicking the **Disconnect** and **Unplug** the USB Cable.

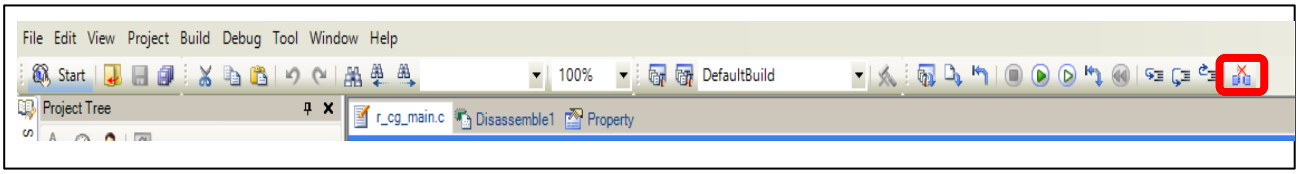


Figure 11: Disconnect CS+ from FPB

4.4. Connecting and Running the Quick Start Example Project

1. Set the on-board dip switch (SW3) into "**Serial**" and connect the Micro USB cable into the Micro USB connector (EJ1) on the board.
2. Insert a coin-cell battery into the battery holder (BT1) with the correct polarity.
3. Connect the other end of the Micro USB cable (USB Head) into the host PC. LED3 (POWER) will light up solid green, indicating that the board is powered.

4.5. Running the Quick Start Example Project

To run the Quick Start Example Project, use the following instructions:

1. On the host PC, open Windows Device Manager. Expand **Ports (Com & LPT)**, located **USB Serial Device (COMxx)** and note down the COM port number for reference in the next step.

Note: USB Serial Device drivers are required to communicate between the board and the terminal application on the host PC.

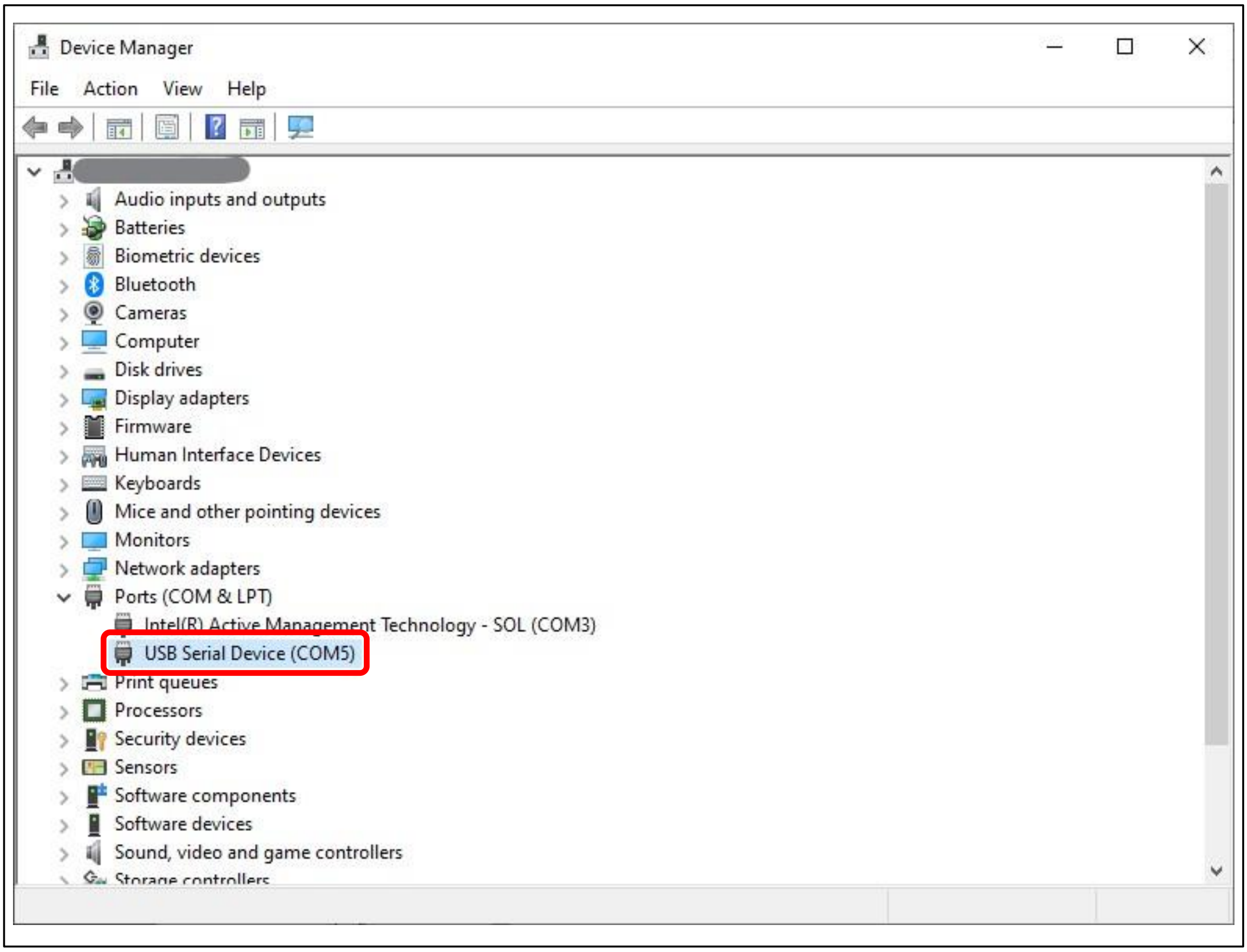
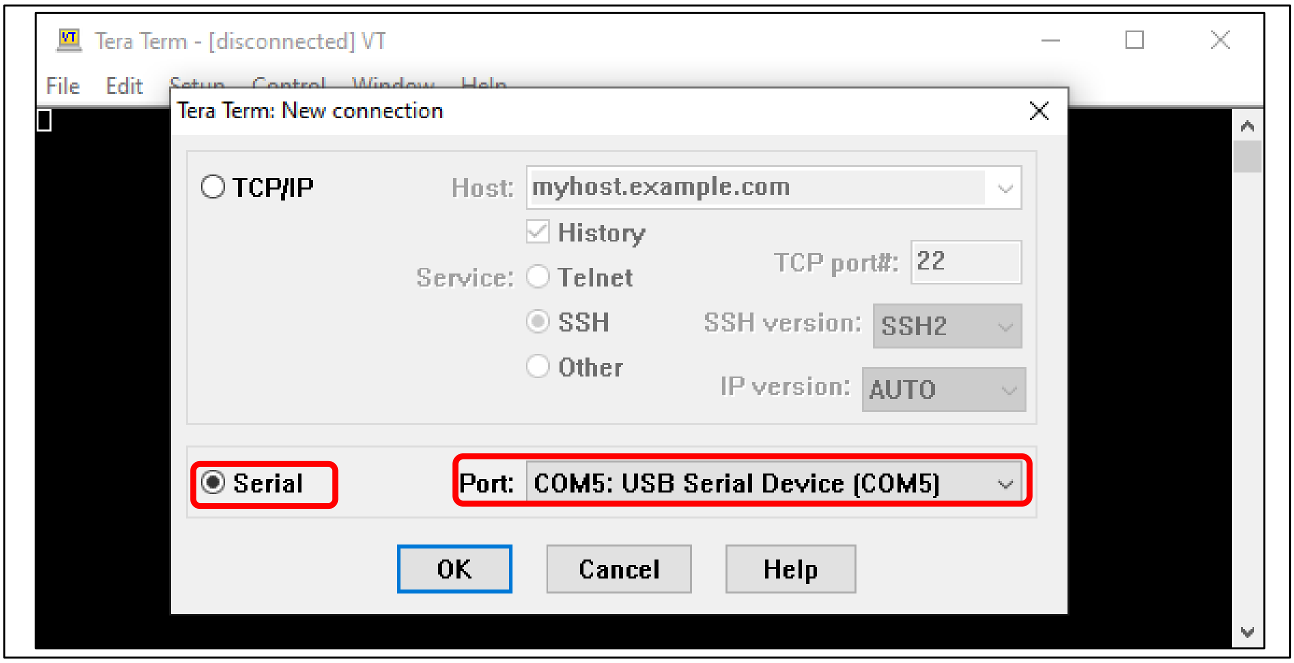


Figure 12: USB Serial Device in Windows Device Manager

2. Open Tera Term, select **Serial** and **COMxx: Serial Device (COMxx)** and click **OK.**


Tera Term - [disconnected] VT
File Edit Setup Control Window Help
Tera Term: New connection
OK
Cancel
Help

○ TCP/IP

Host: myhost.example.com

☑ History

Service: ○ Telnet

● SSH

○ Other

TCP port#: 22

SSH version: SSH2

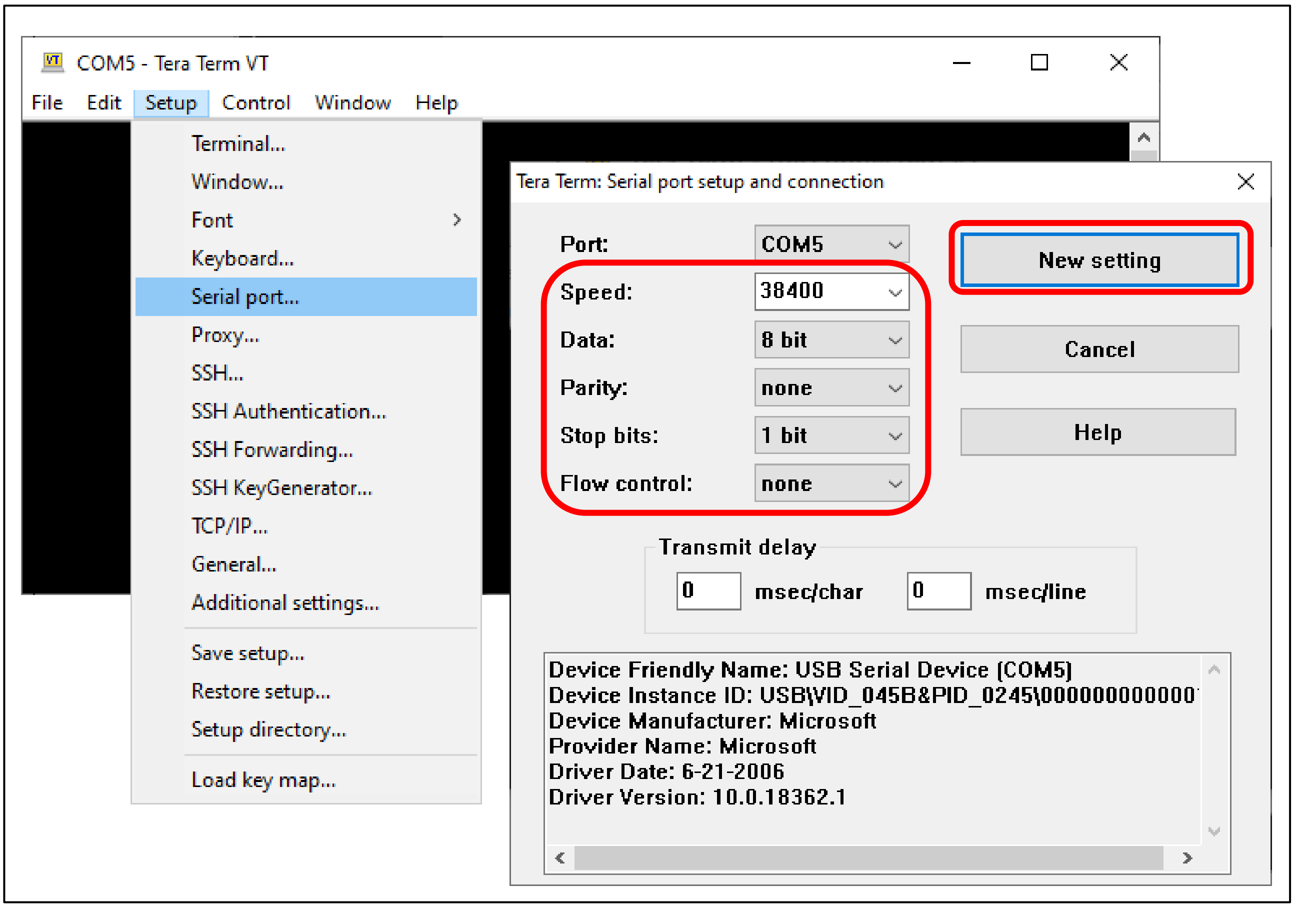
IP version: AUTO

● Serial

Port: COM5: USB Serial Device (COM5)

Figure 13: Selecting the Serial Port on Tera Term

3. In Tera Term, select **Setup** and **Serial Port…** for the **Tera Term: Serial port setup and connection** window. Configure the setup as follows and click **New setting**.



Port: COM5

Speed: 38400

Data: 8 bit

Parity: none

Stop bits: 1 bit

Flow control: none

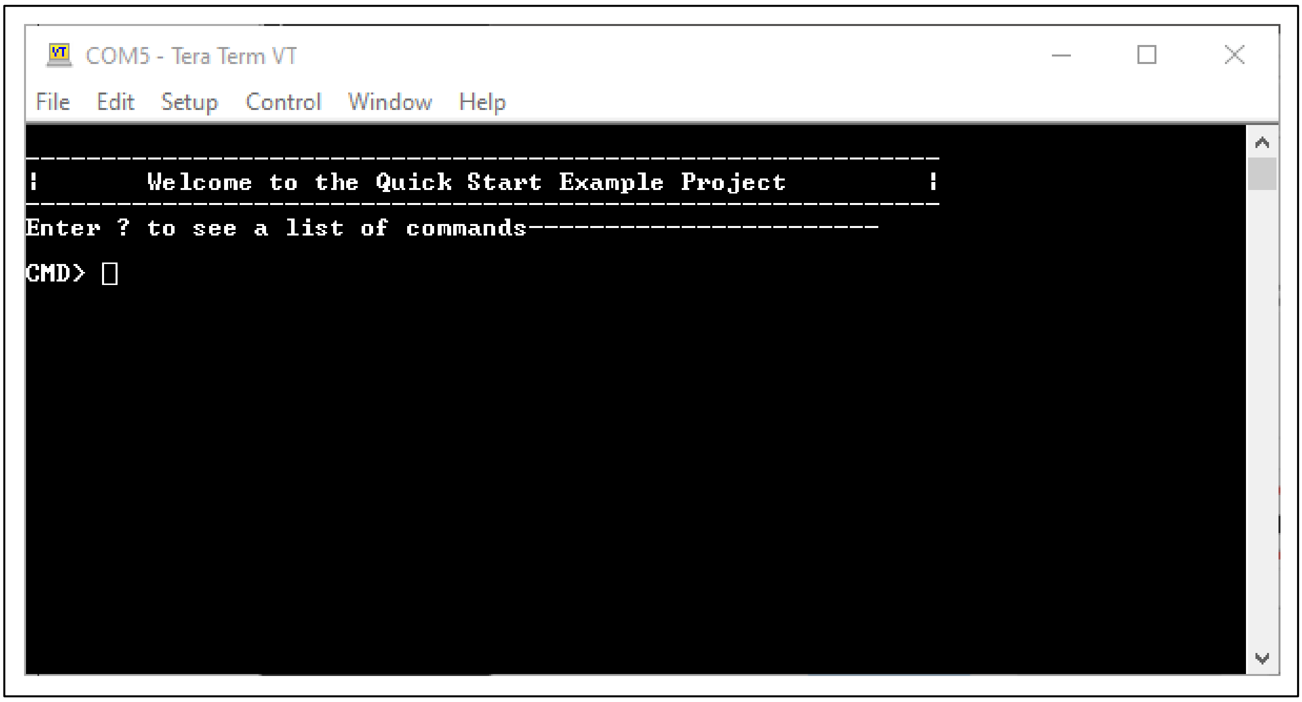
Transmit delay

0 msec/char 0 msec/line

```
Device Friendly Name: USB Serial Device (COM5)
Device Instance ID: USB\VID_045B&PID_0245\000000000000
Device Manufacturer: Microsoft
Provider Name: Microsoft
Driver Date: 6-21-2006
Driver Version: 10.0.18362.1
```

Figure 14: Setting Up the Serial Port in Tera Term

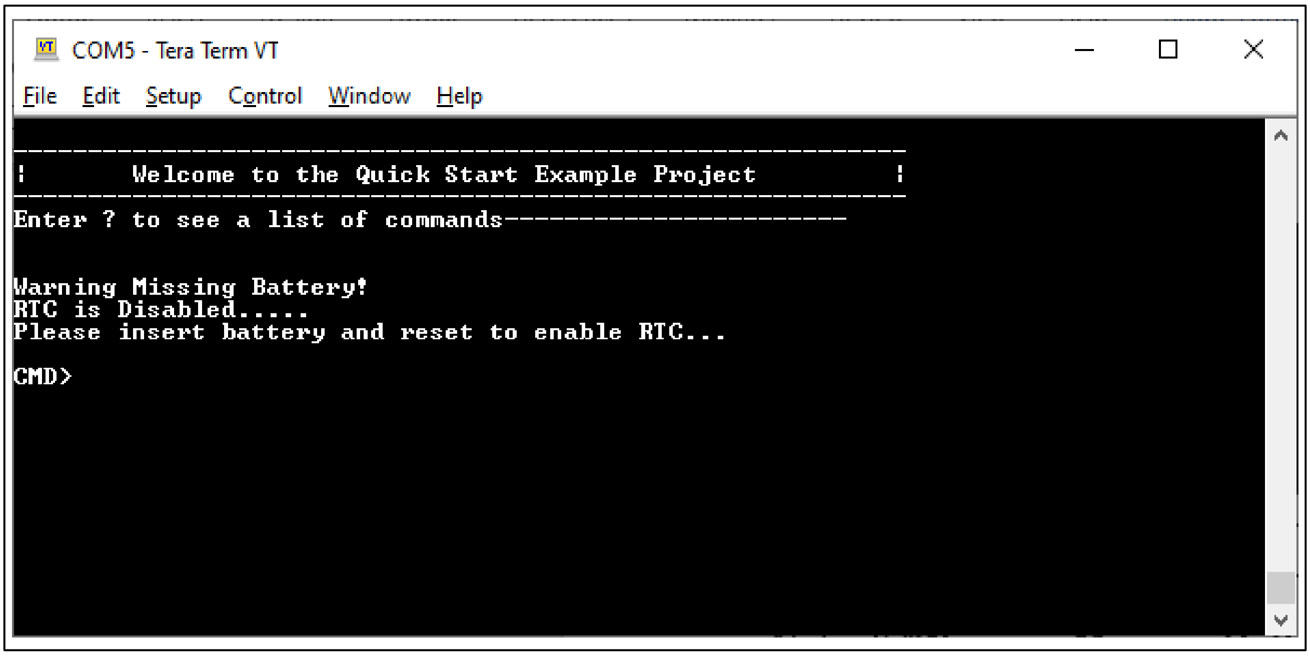
4. Follow the connection of Chapter 4.4, to connect and power the board. Press the on-board **RST** push button once to reset the board.
5. Wait for the welcome message to be displayed.



```
------------------------------------------------------------
|        Welcome to the Quick Start Example Project        |
------------------------------------------------------------
Enter ? to see a list of commands-----------------------

CMD> 
```

Figure 15: Start Message Displayed on Tera Term

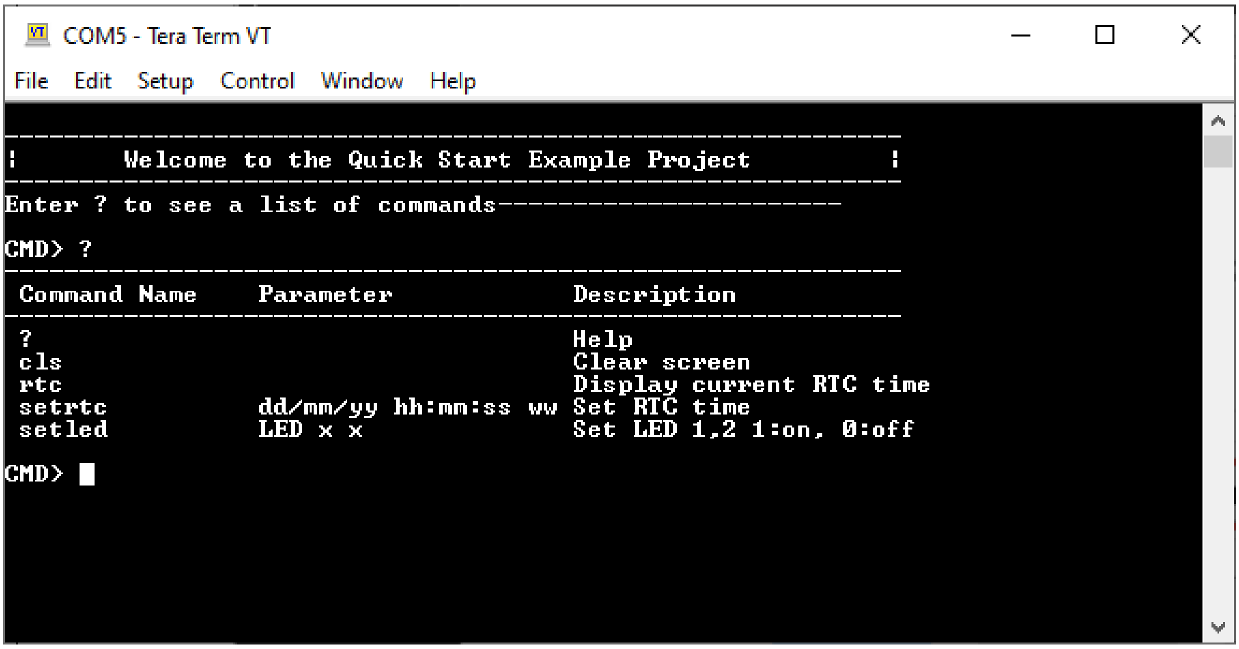
Note: If “**Warning Missing Battery!**” is displayed, insert battery, and reset the board. An example is shown below:



```
------------------------------------------------------------
|        Welcome to the Quick Start Example Project        |
------------------------------------------------------------
Enter ? to see a list of commands-----------------------

Warning Missing Battery!
RTC is Disabled.....
Please insert battery and reset to enable RTC...

CMD>
```

Figure 16: Missing Battery Warning

6. Type “**?**” and press **Enter** key to observe the possible functions.



```
------------------------------------------------------------
|        Welcome to the Quick Start Example Project        |
------------------------------------------------------------
Enter ? to see a list of commands----------------------

CMD> ?
------------------------------------------------------------
 Command Name     Parameter                Description
------------------------------------------------------------
 ?                                         Help
 cls                                       Clear screen
 rtc                                       Display current RTC time
 setrtc           dd/mm/yy hh:mm:ss ww     Set RTC time
 setled           LED x x                  Set LED 1,2 1:on, 0:off

CMD>
```

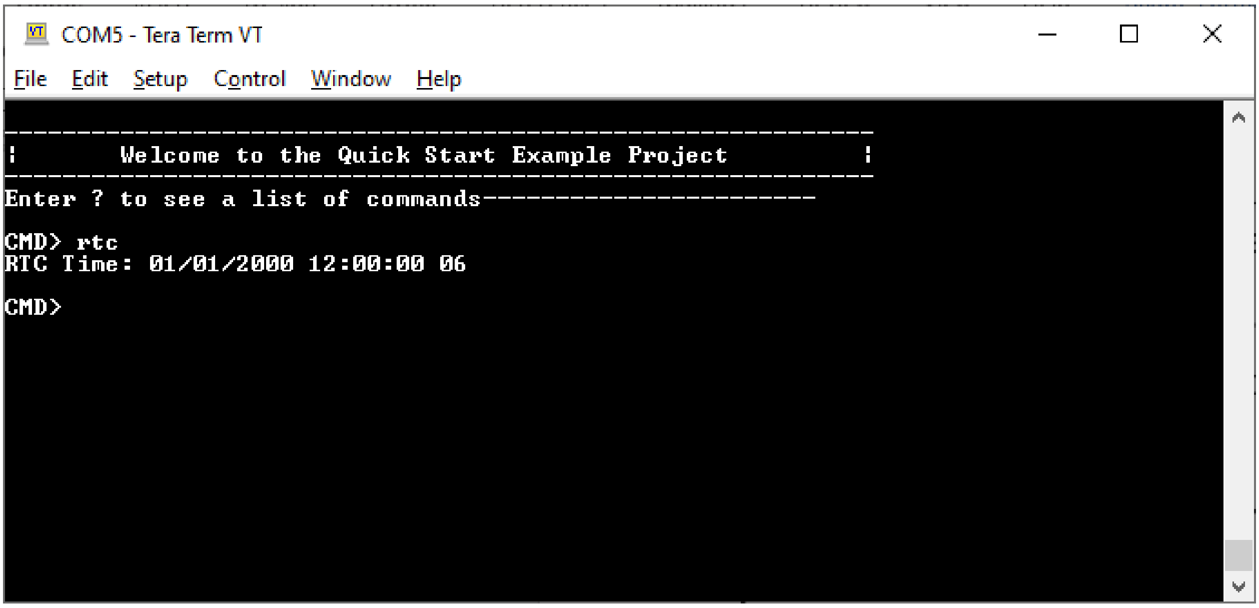
Figure 17: Available Commands in Quick Start Guide Example Project

7. Enter the commands as defined to observe each function.

Note: Commands can be repeatedly entered after the previous command ends.

8. To display current RTC Time, enter **rtc** into the terminal, as shown below:

Note: **RTC Time: 01/01/2000 12:00:00 06** is the default start time of the RTC when not configured



```
------------------------------------------------------------
|        Welcome to the Quick Start Example Project        |
------------------------------------------------------------
Enter ? to see a list of commands----------------------

CMD> rtc
RTC Time: 01/01/2000 12:00:00 06

CMD>
```

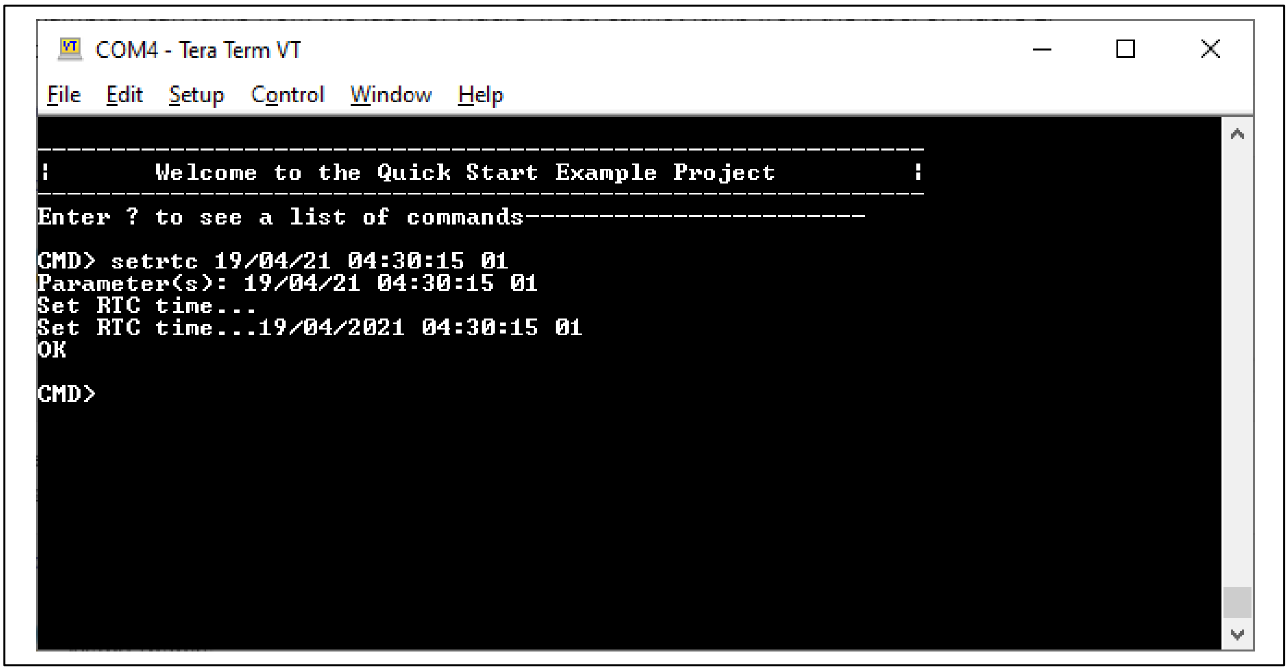
Figure 18: RTC Display Command

9. To set the RTC, enter **setrtc dd/mm/yy hh:mm:ss ww** where:

dd	=	Day
mm	=	Month
yy	=	Year
hh	=	Hour
mm	=	Min
ss	=	Sec
ww	=	Day-of-week [0 = Sunday to 6 = Saturday]

An example is **19/04/21 04:30:15 01**, or 19 April 2021 4:30:15pm week 1 (Monday)

Note: Users are advice to **verify the date and time** they are entering into the system, as no additional checks are available to verify their information

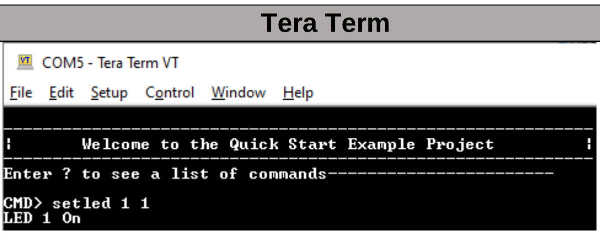
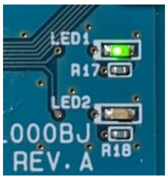
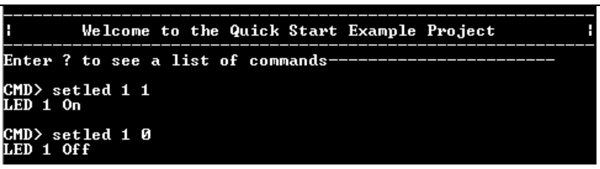
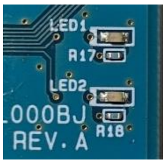
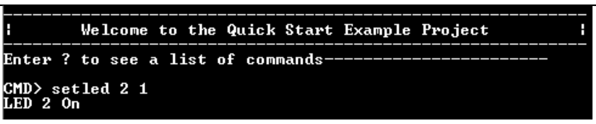
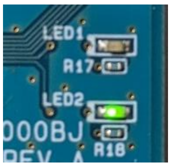
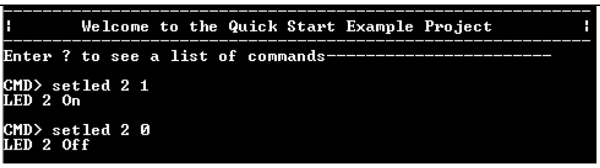
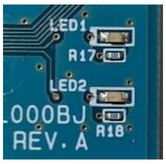


```
------------------------------------------------------------
|        Welcome to the Quick Start Example Project        |
------------------------------------------------------------
Enter ? to see a list of commands----------------------

CMD> setrtc 19/04/21 04:30:15 01
Parameter(s): 19/04/21 04:30:15 01
Set RTC time...
Set RTC time...19/04/2021 04:30:15 01
OK

CMD>
```

Figure 19: Setting RTC through Tera Term

10. LED 2 can be triggered by pressing **SW**.
11. LED 1 and LED 2 can also be triggered by entering **setled x x**, as shown in the table below:

Table 1: LED States with Commands

Enter	Tera Term	FPB Output
setled 1 1	 | Welcome to the Quick Start Example Project | Enter ? to see a list of commands----------------------- CMD> setled 1 1 LED 1 On	
setled 1 0	 | Welcome to the Quick Start Example Project | Enter ? to see a list of commands----------------------- CMD> setled 1 1 LED 1 On CMD> setled 1 0 LED 1 Off	
setled 2 1	 | Welcome to the Quick Start Example Project | Enter ? to see a list of commands----------------------- CMD> setled 2 1 LED 2 On	
setled 2 0	 | Welcome to the Quick Start Example Project | Enter ? to see a list of commands----------------------- CMD> setled 2 1 LED 2 On CMD> setled 2 0 LED 2 Off	

5. Modifying Your Quick Start Example Project

This section provides instructions on 2 methods of customizing the Quick Start Example Project.

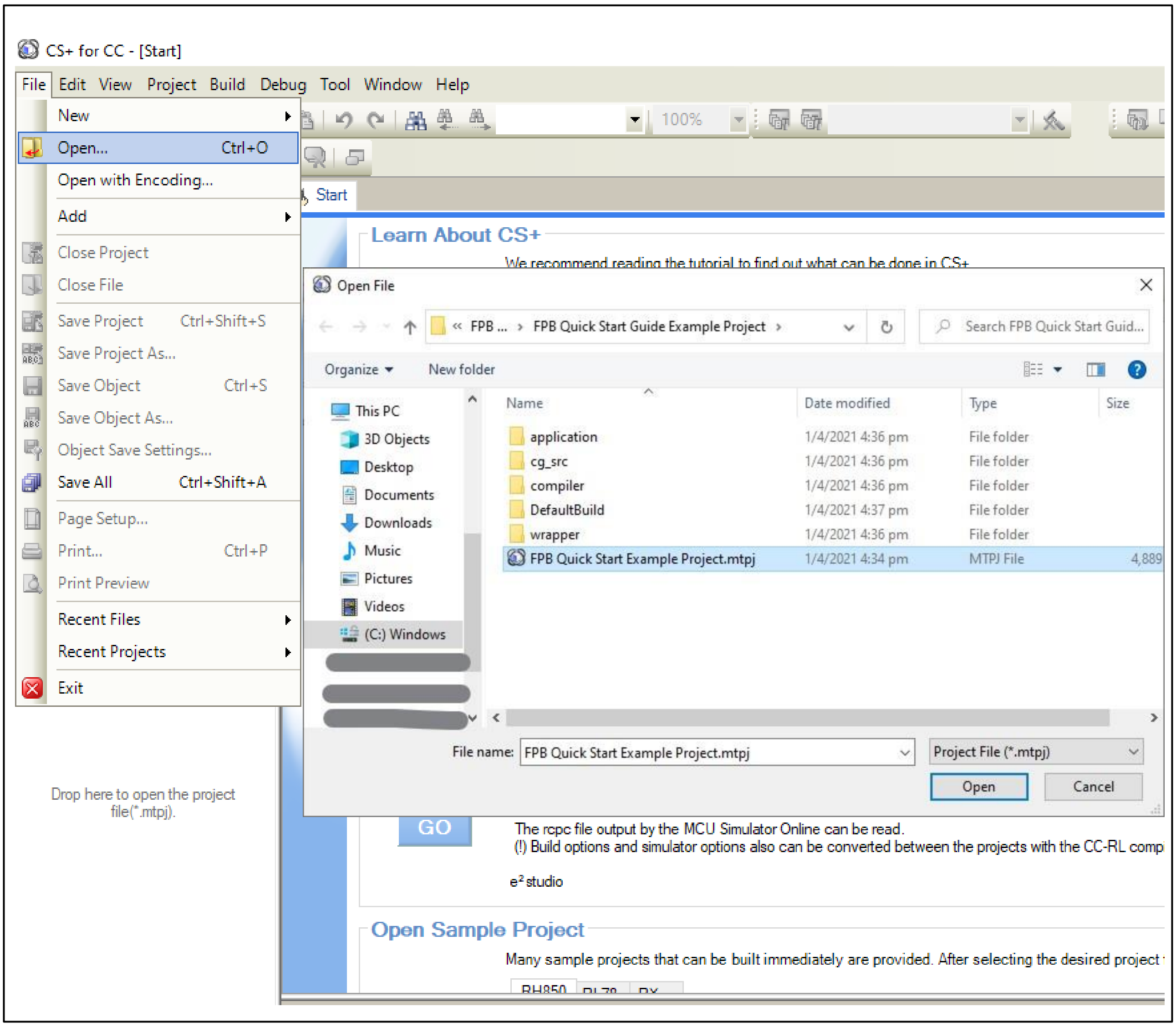
5.1. Required Software

Before the Quick Start Example Project can be modified or created, it is necessary to download and install the CS+ IDE software on the host PC.

CS+ (Formerly CubeSuite+) integrated development environment can be downloaded through the Renesas webpage at renesas.com/us/en/software-tool/cs.

5.2. Opening the Quick Start Example Project

1. Launch CS+ IDE
2. Under **File**, **Open…**, browse to the Quick Start Example Project's directory to locate the **FPB Quick Start Example Project.mtpj**, and **Open**.



We recommend reading the tutorial to find out what can be done in CS+

Name	Date modified	Type	Size
application	1/4/2021 4:36 pm	File folder	
cg_src	1/4/2021 4:36 pm	File folder	
compiler	1/4/2021 4:36 pm	File folder	
DefaultBuild	1/4/2021 4:37 pm	File folder	
wrapper	1/4/2021 4:36 pm	File folder	
FPB Quick Start Example Project.mtpj	1/4/2021 4:34 pm	MTPJ File	4,889

File name: FPB Quick Start Example Project.mtpj

Project File (*.mtpj)

Drop here to open the project file(*.mtpj).

The rcpc file output by the MCU Simulator Online can be read.

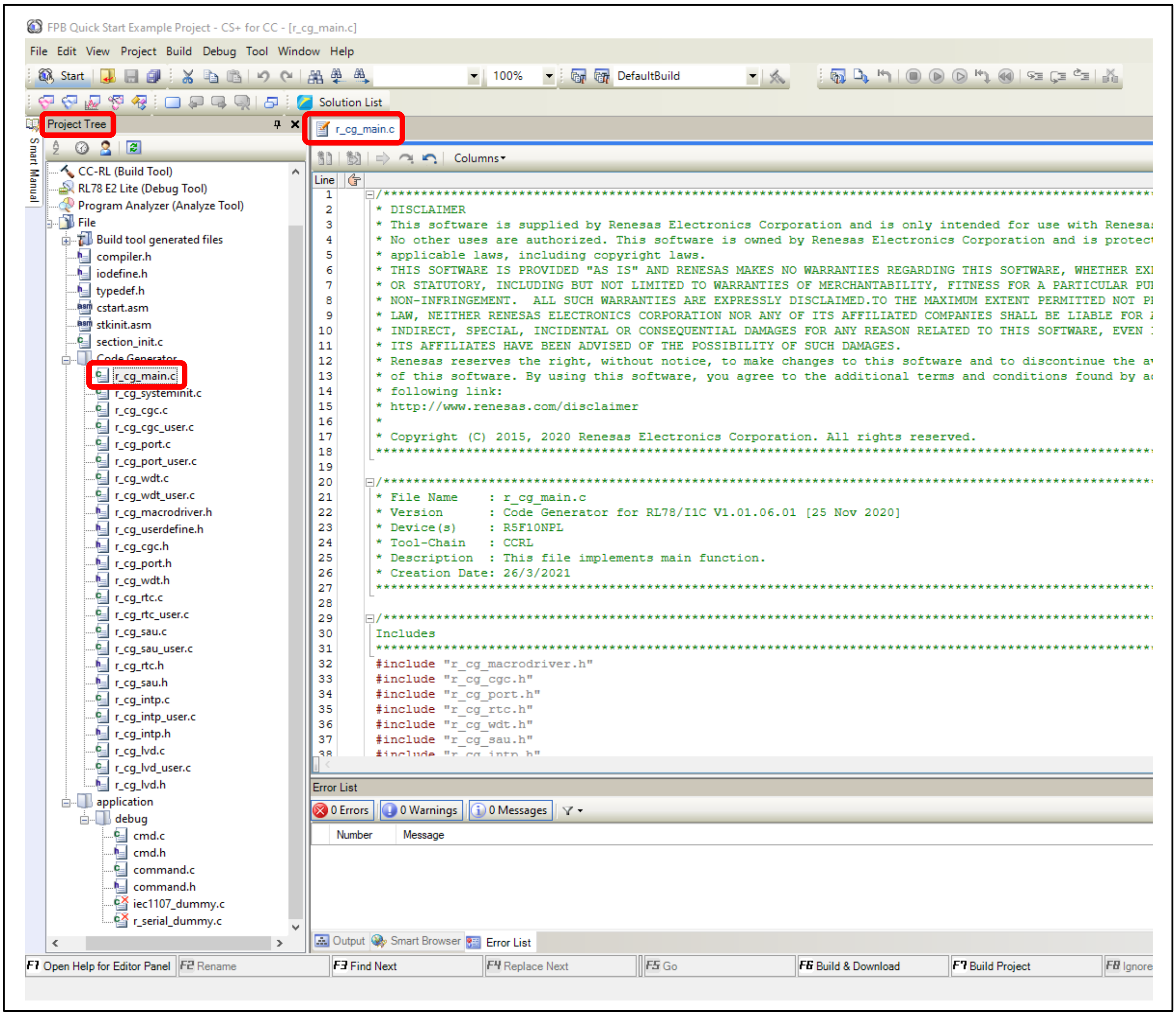
(!) Build options and simulator options also can be converted between the projects with the CC-RL comp

Many sample projects that can be built immediately are provided. After selecting the desired project

Figure 20: Opening the Quick Start Example Project

5.3. Method 1: Modifying the Quick Start Example Project

1. Open individual source files in the project under **Project Tree** by clicking the file names. For example, **r_cg_main.c** file.



```
/********************************************************************************************************
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only intended for use with Renesa
* No other uses are authorized. This software is owned by Renesas Electronics Corporation and is protect
* applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING THIS SOFTWARE, WHETHER EXI
* OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PUI
* NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.TO THE MAXIMUM EXTENT PERMITTED NOT PI
* LAW, NEITHER RENESAS ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE FOR A
* INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN I
* ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software and to discontinue the av
* of this software. By using this software, you agree to the additional terms and conditions found by ac
* following link:
* http://www.renesas.com/disclaimer
*
* Copyright (C) 2015, 2020 Renesas Electronics Corporation. All rights reserved.
*********************************************************************************************************

/********************************************************************************************************
* File Name    : r_cg_main.c
* Version      : Code Generator for RL78/I1C V1.01.06.01 [25 Nov 2020]
* Device(s)    : R5F10NPL
* Tool-Chain   : CCRL
* Description  : This file implements main function.
* Creation Date: 26/3/2021
*********************************************************************************************************

/********************************************************************************************************
Includes
*********************************************************************************************************
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_port.h"
#include "r_cg_rtc.h"
#include "r_cg_wdt.h"
#include "r_cg_sau.h"
#include "r_cg_intp.h"
```

Figure 21: Opening a source file in CS+

2. Modify the source files as needed and save the changes.
3. Build and Download the project into the FPB by clicking the **build & download** icon.

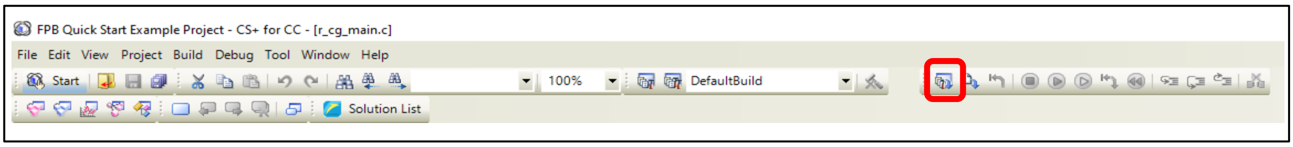
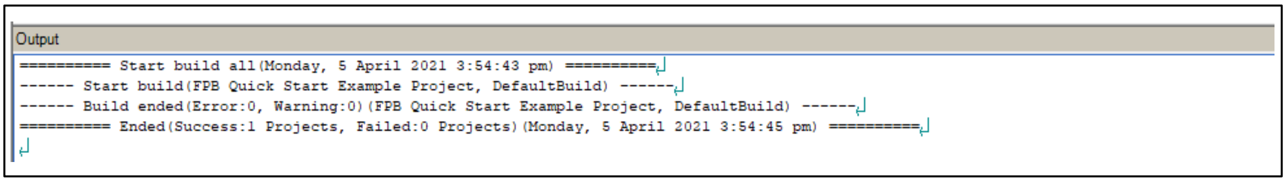

FPB Quick Start Example Project - CS+ for CC - [r_cg_main.c]
File Edit View Project Build Debug Tool Window Help
Start
100%
DefaultBuild
Solution List

Figure 22: Building and downloading the Project into FPB

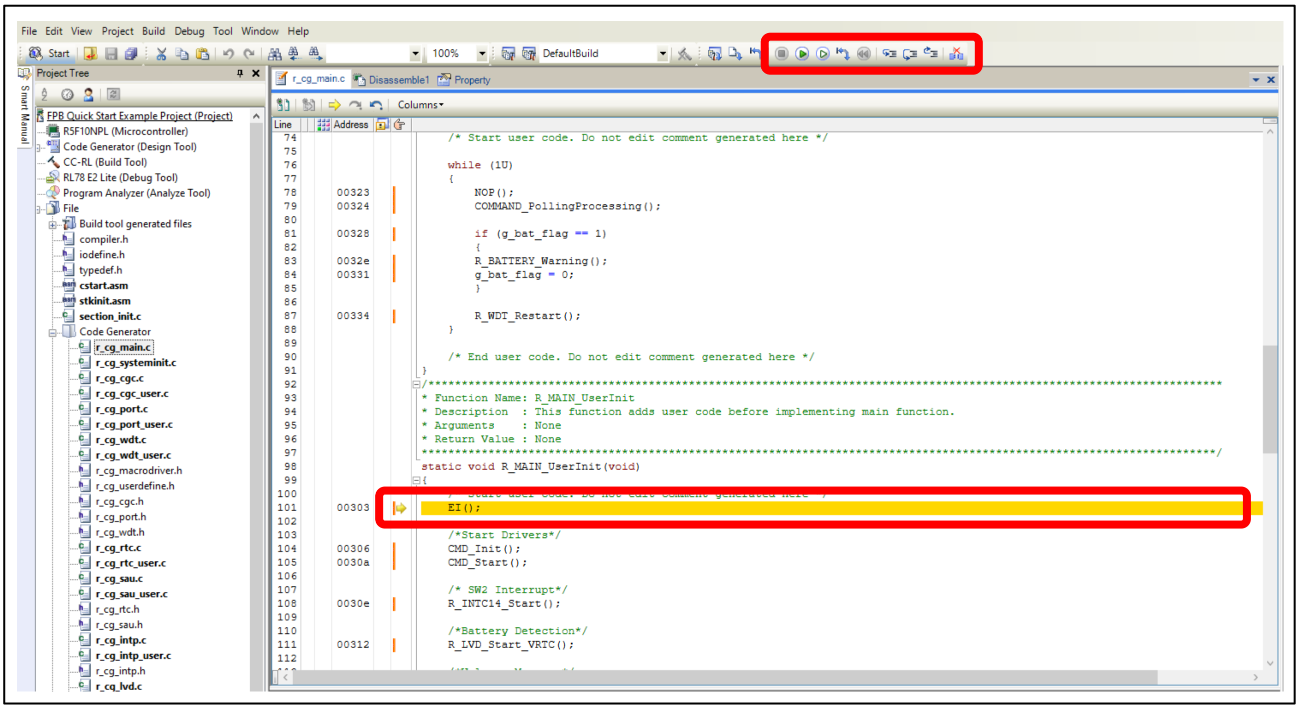
4. A successful build produces an output as follows:



```
========== Start build all(Monday, 5 April 2021 3:54:43 pm) ==========
------ Start build(FPB Quick Start Example Project, DefaultBuild) ------
------ Build ended(Error:0, Warning:0)(FPB Quick Start Example Project, DefaultBuild) ------
========== Ended(Success:1 Projects, Failed:0 Projects)(Monday, 5 April 2021 3:54:45 pm) ==========
```

Figure 23: Successful Project Build

5. A successful download is shown as a **yellow start point** and an active **Debug tool bar.**

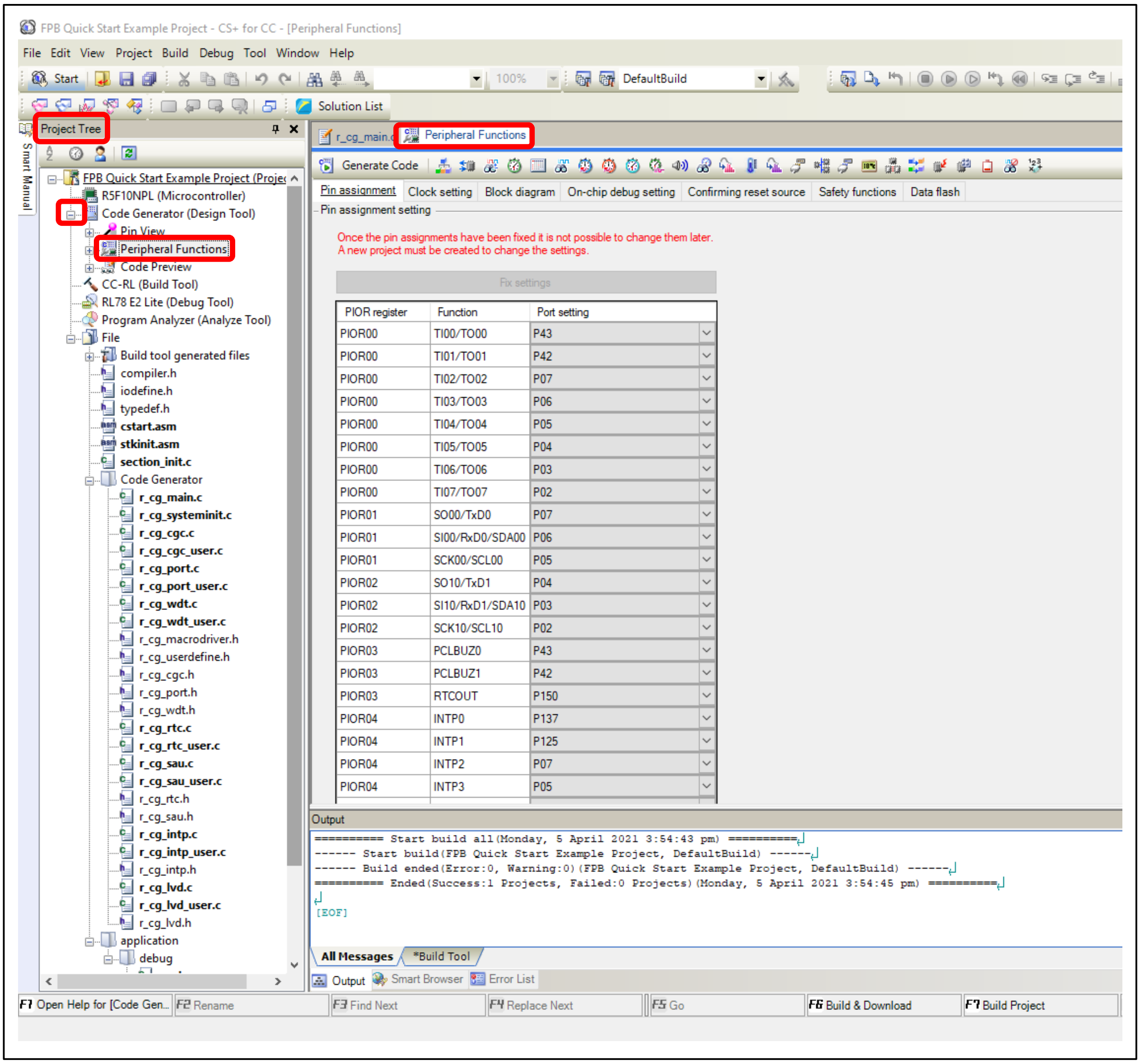


```
            /* Start user code. Do not edit comment generated here */

            while (1U)
            {
00323           NOP();
00324           COMMAND_PollingProcessing();

00328           if (g_bat_flag == 1)
                {
0032e           R_BATTERY_Warning();
00331           g_bat_flag = 0;
                }

00334           R_WDT_Restart();
            }

            /* End user code. Do not edit comment generated here */
        }
        /***********************************************************************************************************************
        * Function Name: R_MAIN_UserInit
        * Description  : This function adds user code before implementing main function.
        * Arguments    : None
        * Return Value : None
        ***********************************************************************************************************************/
        static void R_MAIN_UserInit(void)
        {
            /* Start user code. Do not edit comment generated here */
00303       EI();

            /*Start Drivers*/
00306       CMD_Init();
0030a       CMD_Start();

            /* SW2 Interrupt*/
0030e       R_INTC14_Start();

            /*Battery Detection*/
00312       R_LVD_Start_VRTC();
```

Figure 24: Successful Download into FPB

5.4. Method 2: Modifying the Quick Start Example Project with CS+ Code Generator

1. Open Code Generator from **Project Tree** click the dropdown for **Code Generator (Design Tool)** and click on **Peripheral Functions**.



Once the pin assignments have been fixed it is not possible to change them later.
A new project must be created to change the settings.

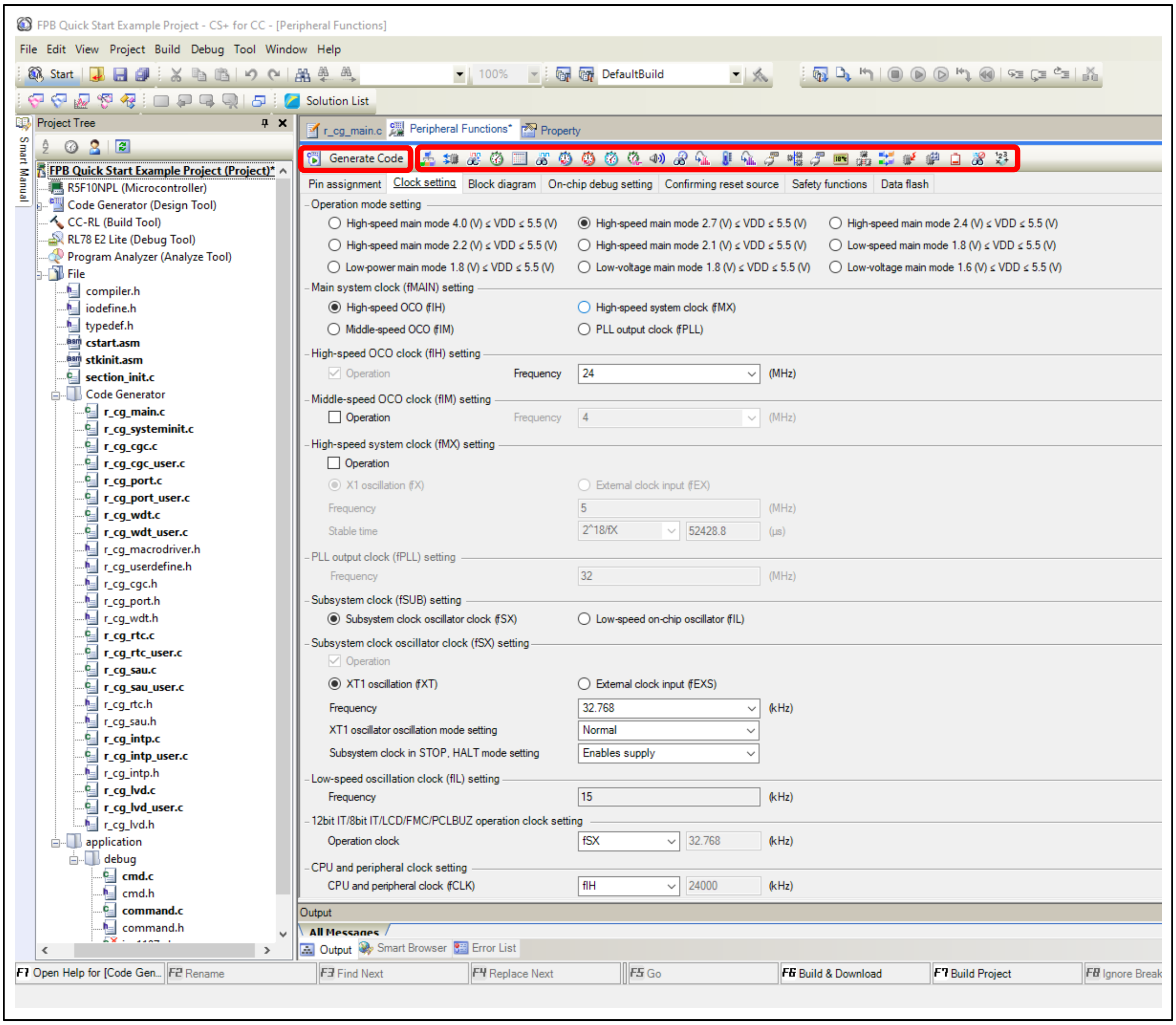
PIOR register	Function	Port setting
PIOR00	TI00/TO00	P43
PIOR00	TI01/TO01	P42
PIOR00	TI02/TO02	P07
PIOR00	TI03/TO03	P06
PIOR00	TI04/TO04	P05
PIOR00	TI05/TO05	P04
PIOR00	TI06/TO06	P03
PIOR00	TI07/TO07	P02
PIOR01	SO00/TxD0	P07
PIOR01	SI00/RxD0/SDA00	P06
PIOR01	SCK00/SCL00	P05
PIOR02	SO10/TxD1	P04
PIOR02	SI10/RxD1/SDA10	P03
PIOR02	SCK10/SCL10	P02
PIOR03	PCLBUZ0	P43
PIOR03	PCLBUZ1	P42
PIOR03	RTCOUT	P150
PIOR04	INTP0	P137
PIOR04	INTP1	P125
PIOR04	INTP2	P07
PIOR04	INTP3	P05

```
========== Start build all(Monday, 5 April 2021 3:54:43 pm) ==========
------ Start build(FPB Quick Start Example Project, DefaultBuild) ------
------ Build ended(Error:0, Warning:0)(FPB Quick Start Example Project, DefaultBuild) ------
========== Ended(Success:1 Projects, Failed:0 Projects)(Monday, 5 April 2021 3:54:45 pm) ==========

[EOF]
```

Figure 25: Opening Code Generator

2. Configure each function and their tabs to the required functionality and click **Generate Code**.

Note: The generated codes will appear as source files in the **Project Tree**, under **Code Generator**



Operation mode setting

○ High-speed main mode 4.0 (V) ≤ VDD ≤ 5.5 (V) ● High-speed main mode 2.7 (V) ≤ VDD ≤ 5.5 (V) ○ High-speed main mode 2.4 (V) ≤ VDD ≤ 5.5 (V)

○ High-speed main mode 2.2 (V) ≤ VDD ≤ 5.5 (V) ○ High-speed main mode 2.1 (V) ≤ VDD ≤ 5.5 (V) ○ Low-speed main mode 1.8 (V) ≤ VDD ≤ 5.5 (V)

○ Low-power main mode 1.8 (V) ≤ VDD ≤ 5.5 (V) ○ Low-voltage main mode 1.8 (V) ≤ VDD ≤ 5.5 (V) ○ Low-voltage main mode 1.6 (V) ≤ VDD ≤ 5.5 (V)

Main system clock (fMAIN) setting

● High-speed OCO (fIH) ○ High-speed system clock (fMX)

○ Middle-speed OCO (fIM) ○ PLL output clock (fPLL)

High-speed OCO clock (fIH) setting

☑ Operation Frequency 24 (MHz)

Middle-speed OCO clock (fIM) setting

□ Operation Frequency 4 (MHz)

High-speed system clock (fMX) setting

□ Operation

● X1 oscillation (fX) ○ External clock input (fEX)

Frequency 5 (MHz)

Stable time 2^18/fX 52428.8 (µs)

PLL output clock (fPLL) setting

Frequency 32 (MHz)

Subsystem clock (fSUB) setting

● Subsystem clock oscillator clock (fSX) ○ Low-speed on-chip oscillator (fIL)

Subsystem clock oscillator clock (fSX) setting

☑ Operation

● XT1 oscillation (fXT) ○ External clock input (fEXS)

Frequency 32.768 (kHz)

XT1 oscillator oscillation mode setting Normal

Subsystem clock in STOP, HALT mode setting Enables supply

Low-speed oscillation clock (fIL) setting

Frequency 15 (kHz)

12bit IT/8bit IT/LCD/FMC/PCLBUZ operation clock setting

Operation clock fSX 32.768 (kHz)

CPU and peripheral clock setting

CPU and peripheral clock (fCLK) fIH 24000 (kHz)

Figure 26: Configuring the Code Generator

3. Build and Download the project into the FPB by clicking the **build & download** icon.

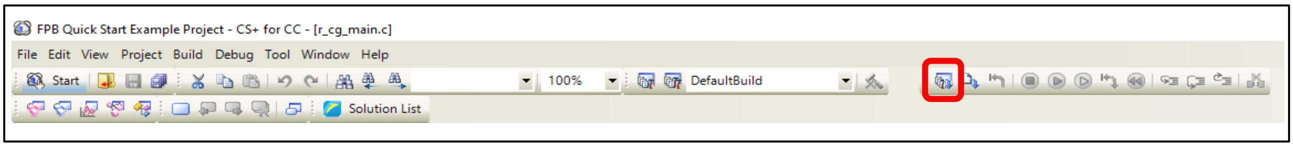
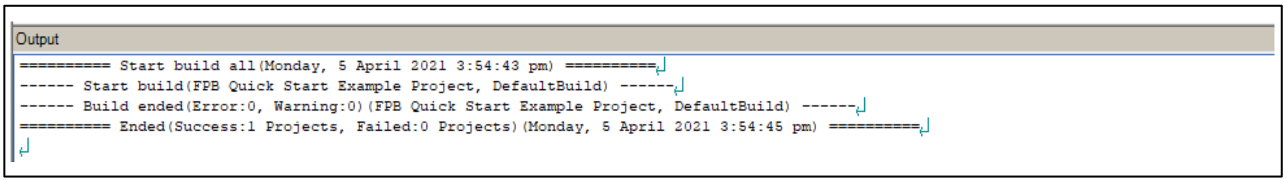

FPB Quick Start Example Project - CS+ for CC - [r_cg_main.c]
File Edit View Project Build Debug Tool Window Help
Start
100%
DefaultBuild
Solution List

Figure 27: Building the Project

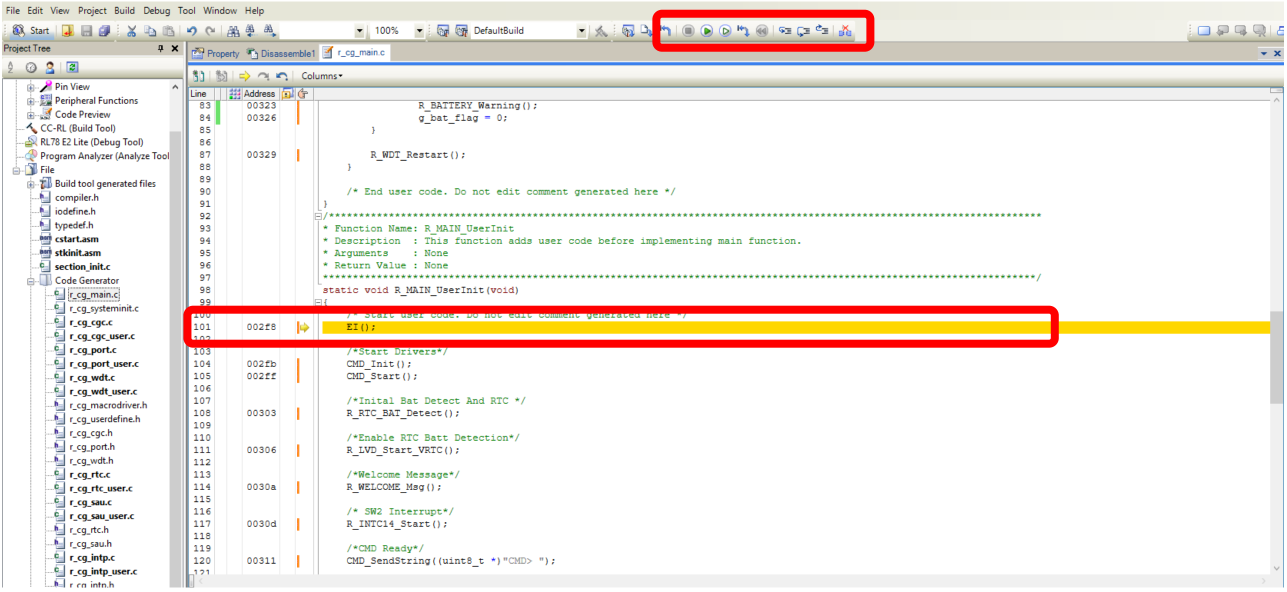
4. A successful build produces an output as follows:



```
========== Start build all(Monday, 5 April 2021 3:54:43 pm) ==========
------ Start build(FPB Quick Start Example Project, DefaultBuild) ------
------ Build ended(Error:0, Warning:0)(FPB Quick Start Example Project, DefaultBuild) ------
========== Ended(Success:1 Projects, Failed:0 Projects)(Monday, 5 April 2021 3:54:45 pm) ==========
```

Figure 28: Code Generator Successful Build

5. A successful download is shown as a **yellow start point** and an active **Debug tool bar.**



```
00323                   R_BATTERY_Warning();
00326                   g_bat_flag = 0;
                    }

00329           R_WDT_Restart();
            }

        /* End user code. Do not edit comment generated here */
}
/***********************************************************************************************************************
* Function Name: R_MAIN_UserInit
* Description  : This function adds user code before implementing main function.
* Arguments    : None
* Return Value : None
***********************************************************************************************************************/
static void R_MAIN_UserInit(void)
{
    /* Start user code. Do not edit comment generated here */
002f8    EI();

    /*Start Drivers*/
002fb    CMD_Init();
002ff    CMD_Start();

    /*Inital Bat Detect And RTC */
00303    R_RTC_BAT_Detect();

    /*Enable RTC Batt Detection*/
00306    R_LVD_Start_VRTC();

    /*Welcome Message*/
0030a    R_WELCOME_Msg();

    /* SW2 Interrupt*/
0030d    R_INTC14_Start();

    /*CMD Ready*/
00311    CMD_SendString((uint8_t *)"CMD> ");
```

Figure 29: Successful Download into FPB

6. Building the Quick Start Example Project

This section shows the required settings in Code Generator and source file to build the Quick Start Example Project.

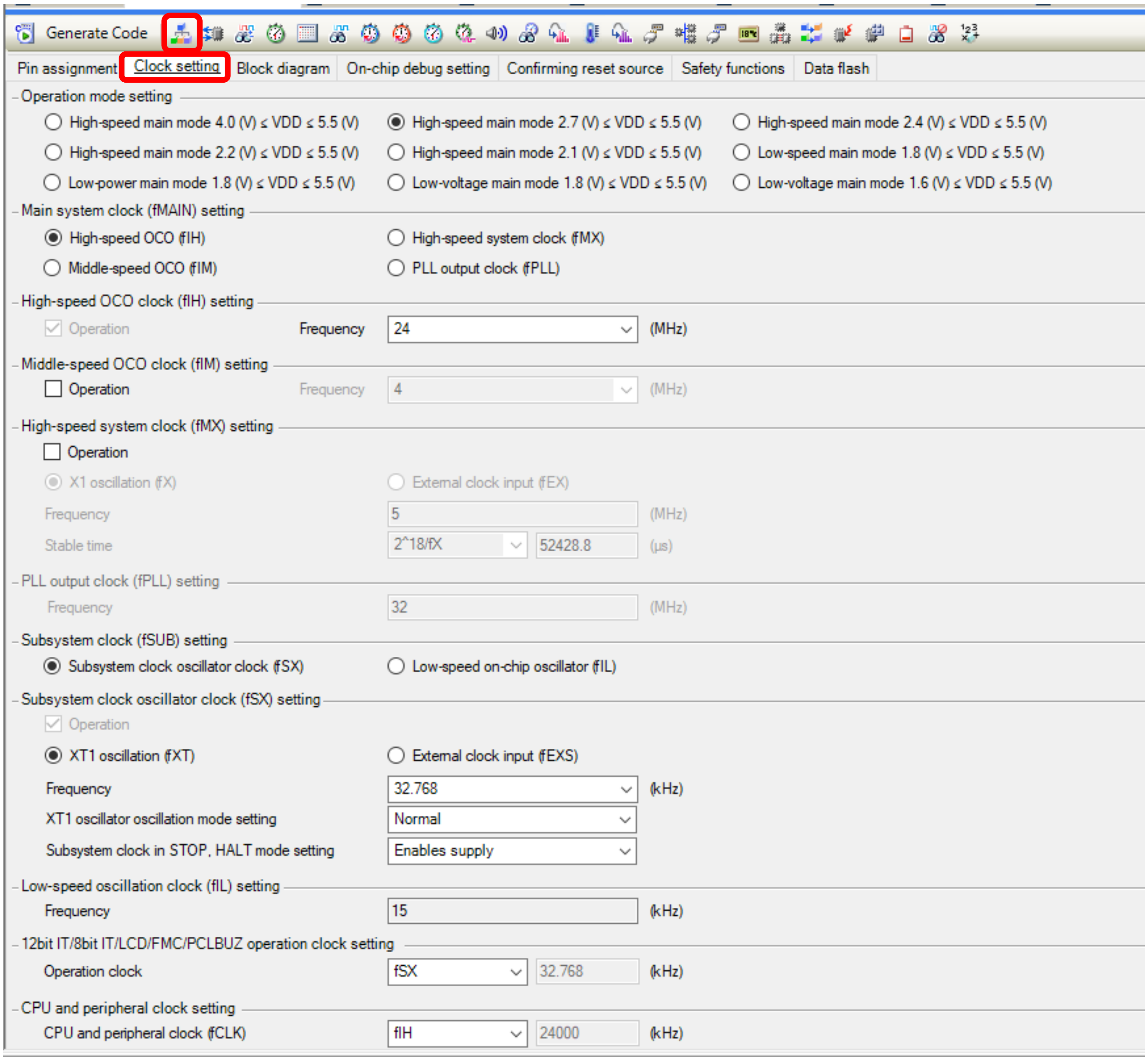
6.1. Additional Files Required

1. Compiler.h
2. Iodefine.h
3. Section_init.c
4. Cmd.c & Cmd.h
5. Command.c & Command.h

Note: The additional files stated can be imported from the supplied Quick Start Example Project

6.2. Configurating the individual functions using Code Generator

1. Configurations of **Common/ Clock Generator**, under **Clock settings**



Operation mode setting
○ High-speed main mode 4.0 (V) ≤ VDD ≤ 5.5 (V) ● High-speed main mode 2.7 (V) ≤ VDD ≤ 5.5 (V) ○ High-speed main mode 2.4 (V) ≤ VDD ≤ 5.5 (V)
○ High-speed main mode 2.2 (V) ≤ VDD ≤ 5.5 (V) ○ High-speed main mode 2.1 (V) ≤ VDD ≤ 5.5 (V) ○ Low-speed main mode 1.8 (V) ≤ VDD ≤ 5.5 (V)
○ Low-power main mode 1.8 (V) ≤ VDD ≤ 5.5 (V) ○ Low-voltage main mode 1.8 (V) ≤ VDD ≤ 5.5 (V) ○ Low-voltage main mode 1.6 (V) ≤ VDD ≤ 5.5 (V)

Main system clock (fMAIN) setting
● High-speed OCO (fIH) ○ High-speed system clock (fMX)
○ Middle-speed OCO (fIM) ○ PLL output clock (fPLL)

High-speed OCO clock (fIH) setting
☑ Operation Frequency 24 (MHz)

Middle-speed OCO clock (fIM) setting
□ Operation Frequency 4 (MHz)

High-speed system clock (fMX) setting
□ Operation
● X1 oscillation (fX) ○ External clock input (fEX)
Frequency 5 (MHz)
Stable time 2^18/fX 52428.8 (µs)

PLL output clock (fPLL) setting
Frequency 32 (MHz)

Subsystem clock (fSUB) setting
● Subsystem clock oscillator clock (fSX) ○ Low-speed on-chip oscillator (fIL)

Subsystem clock oscillator clock (fSX) setting
☑ Operation
● XT1 oscillation (fXT) ○ External clock input (fEXS)
Frequency 32.768 (kHz)
XT1 oscillator oscillation mode setting Normal
Subsystem clock in STOP, HALT mode setting Enables supply

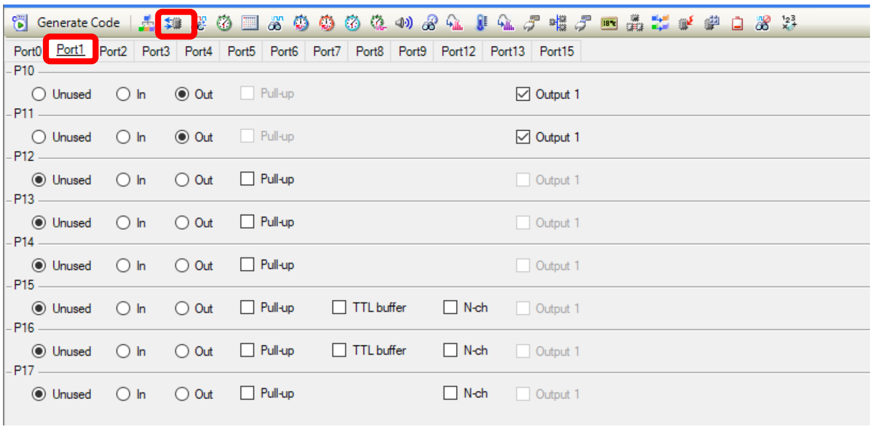
Low-speed oscillation clock (fIL) setting
Frequency 15 (kHz)

12bit IT/8bit IT/LCD/FMC/PCLBUZ operation clock setting
Operation clock fSX 32.768 (kHz)

CPU and peripheral clock setting
CPU and peripheral clock (fCLK) fIH 24000 (kHz)

Figure 30: Clock Setting in Code Generator

2. Configurations of **Port Function**, in **Port1**



Generate Code

Port0 | Port1 | Port2 | Port3 | Port4 | Port5 | Port6 | Port7 | Port8 | Port9 | Port12 | Port13 | Port15

P10
○ Unused ○ In ● Out □ Pull-up ☑ Output 1

P11
○ Unused ○ In ● Out □ Pull-up ☑ Output 1

P12
● Unused ○ In ○ Out □ Pull-up □ Output 1

P13
● Unused ○ In ○ Out □ Pull-up □ Output 1

P14
● Unused ○ In ○ Out □ Pull-up □ Output 1

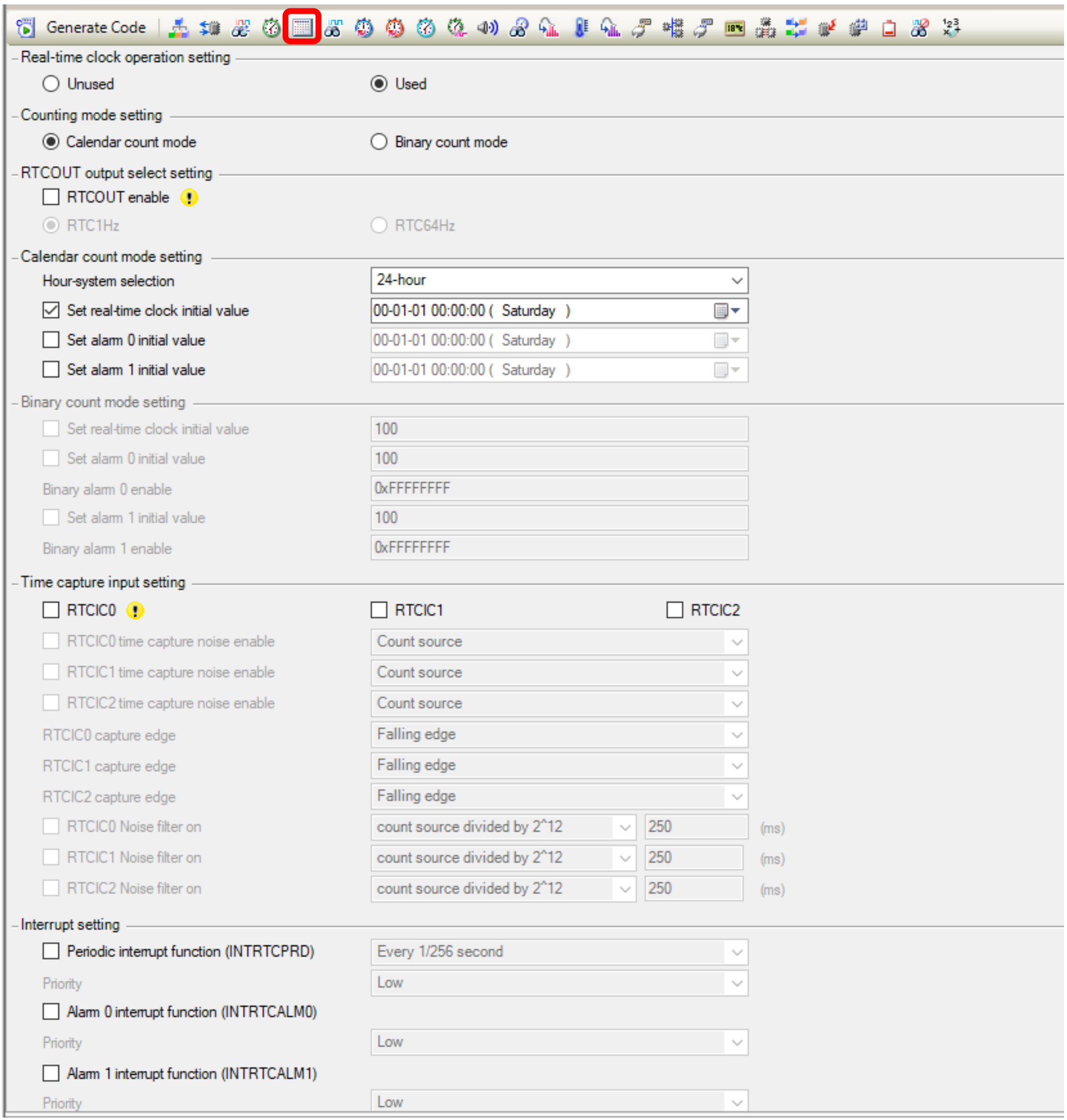
P15
● Unused ○ In ○ Out □ Pull-up □ TTL buffer □ N-ch □ Output 1

P16
● Unused ○ In ○ Out □ Pull-up □ TTL buffer □ N-ch □ Output 1

P17
● Unused ○ In ○ Out □ Pull-up □ N-ch □ Output 1

Figure 31: Port 1 Setting in Code Generator

3. Configuration of **Real-Time Clock**



Generate Code

Real-time clock operation setting
○ Unused ● Used

Counting mode setting
● Calendar count mode ○ Binary count mode

RTCOUT output select setting
□ RTCOUT enable
● RTC1Hz ○ RTC64Hz

Calendar count mode setting
Hour-system selection: 24-hour
☑ Set real-time clock initial value: 00-01-01 00:00:00 (Saturday)
□ Set alarm 0 initial value: 00-01-01 00:00:00 (Saturday)
□ Set alarm 1 initial value: 00-01-01 00:00:00 (Saturday)

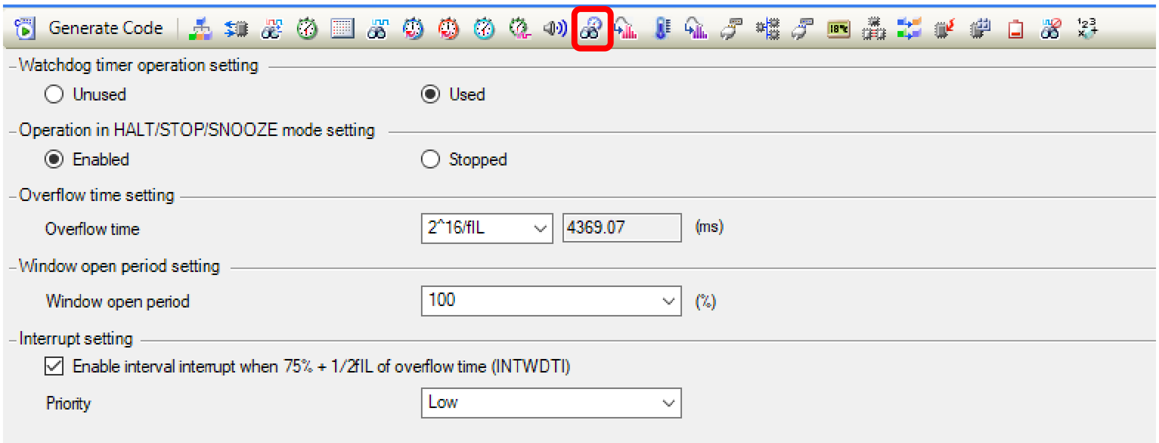
Binary count mode setting
□ Set real-time clock initial value: 100
□ Set alarm 0 initial value: 100
Binary alarm 0 enable: 0xFFFFFFFF
□ Set alarm 1 initial value: 100
Binary alarm 1 enable: 0xFFFFFFFF

Time capture input setting
□ RTCIC0 □ RTCIC1 □ RTCIC2
□ RTCIC0 time capture noise enable: Count source
□ RTCIC1 time capture noise enable: Count source
□ RTCIC2 time capture noise enable: Count source
RTCIC0 capture edge: Falling edge
RTCIC1 capture edge: Falling edge
RTCIC2 capture edge: Falling edge
□ RTCIC0 Noise filter on: count source divided by 2^12 250 (ms)
□ RTCIC1 Noise filter on: count source divided by 2^12 250 (ms)
□ RTCIC2 Noise filter on: count source divided by 2^12 250 (ms)

Interrupt setting
□ Periodic interrupt function (INTRTCPRD): Every 1/256 second
Priority: Low
□ Alarm 0 interrupt function (INTRTCALM0)
Priority: Low
□ Alarm 1 interrupt function (INTRTCALM1)
Priority: Low

Figure 32: Real-Time Clock Setting in Code Generator

4. Configuration of **Watchdog Timer**



Watchdog timer operation setting

○ Unused ● Used

Operation in HALT/STOP/SNOOZE mode setting

● Enabled ○ Stopped

Overflow time setting

Overflow time 2^16/fIL 4369.07 (ms)

Window open period setting

Window open period 100 (%)

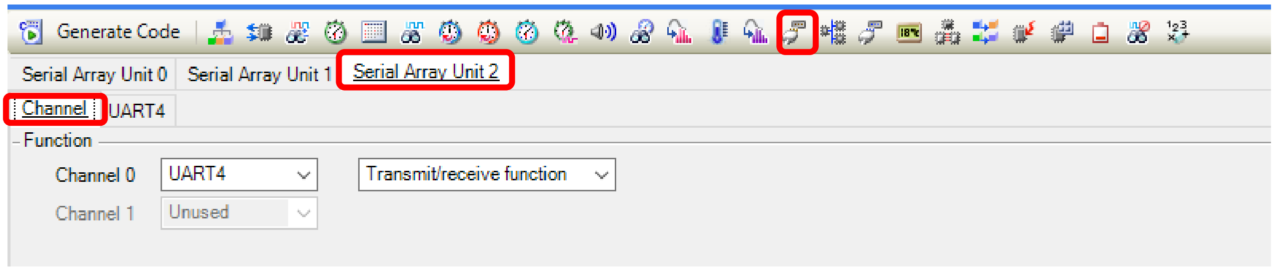
Interrupt setting

☑ Enable interval interrupt when 75% + 1/2fIL of overflow time (INTWDTI)

Priority Low

Figure 33: Watchdog Timer Setting in Code Generator

5. Configuration of **Serial Array Unit**, inside **Serial Array Unit 2**, click **Channel**



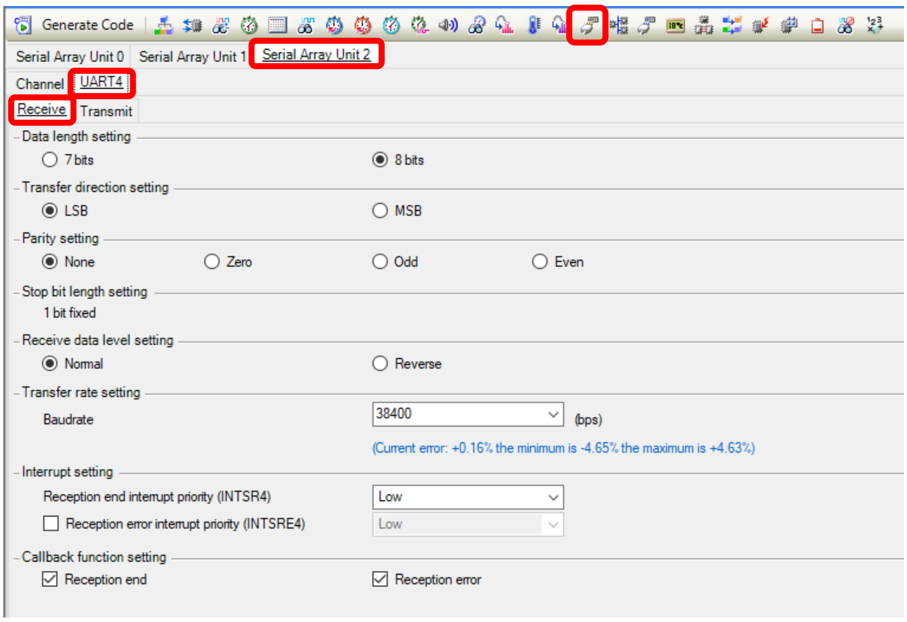
Function

Channel 0 UART4 Transmit/receive function

Channel 1 Unused

Figure 34: Serial Array Unit Channel Setting inside Code Generator

6. Configuration of **Serial Array Unit**, inside **Serial Array Unit 2**, click **UART4**, inside **Receive** and **Transmit**



Data length setting

○ 7 bits ● 8 bits

Transfer direction setting

● LSB ○ MSB

Parity setting

● None ○ Zero ○ Odd ○ Even

Stop bit length setting

1 bit fixed

Receive data level setting

● Normal ○ Reverse

Transfer rate setting

Baudrate 38400 (bps)

(Current error: +0.16% the minimum is -4.65% the maximum is +4.63%)

Interrupt setting

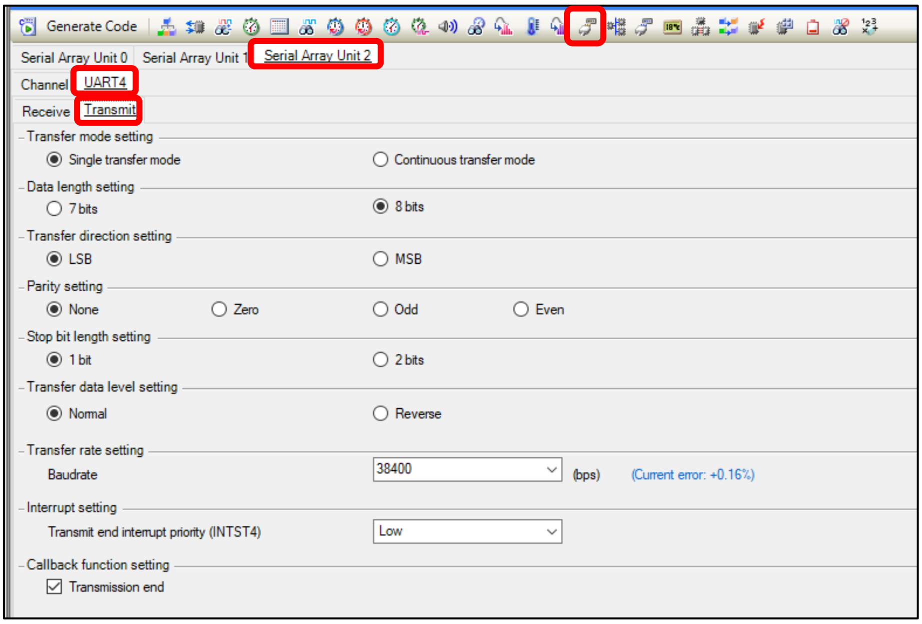
Reception end interrupt priority (INTSR4) Low

□ Reception error interrupt priority (INTSRE4) Low

Callback function setting

☑ Reception end ☑ Reception error

Figure 35: UART 4 Receive Setting in Code Generator



Channel UART4

Receive Transmit

Transfer mode setting
● Single transfer mode ○ Continuous transfer mode

Data length setting
○ 7 bits ● 8 bits

Transfer direction setting
● LSB ○ MSB

Parity setting
● None ○ Zero ○ Odd ○ Even

Stop bit length setting
● 1 bit ○ 2 bits

Transfer data level setting
● Normal ○ Reverse

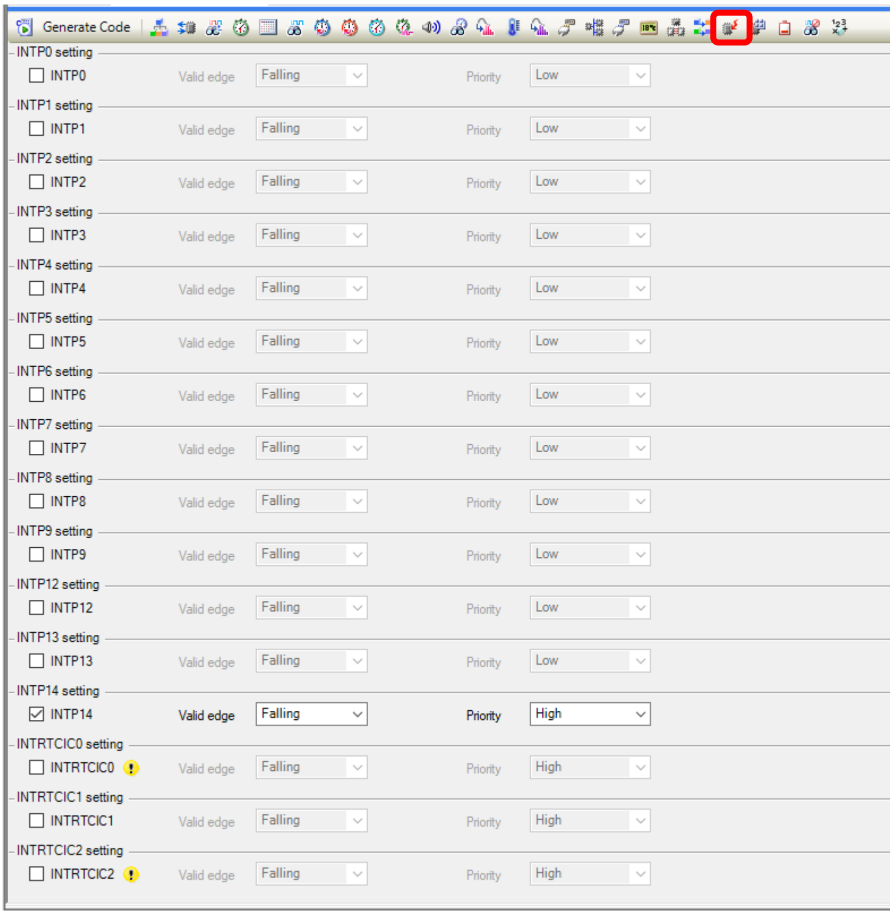
Transfer rate setting
Baudrate 38400 (bps) (Current error: +0.16%)

Interrupt setting
Transmit end interrupt priority (INTST4) Low

Callback function setting
☑ Transmission end

Figure 36: UART 4 Transmit Setting in Code Generator

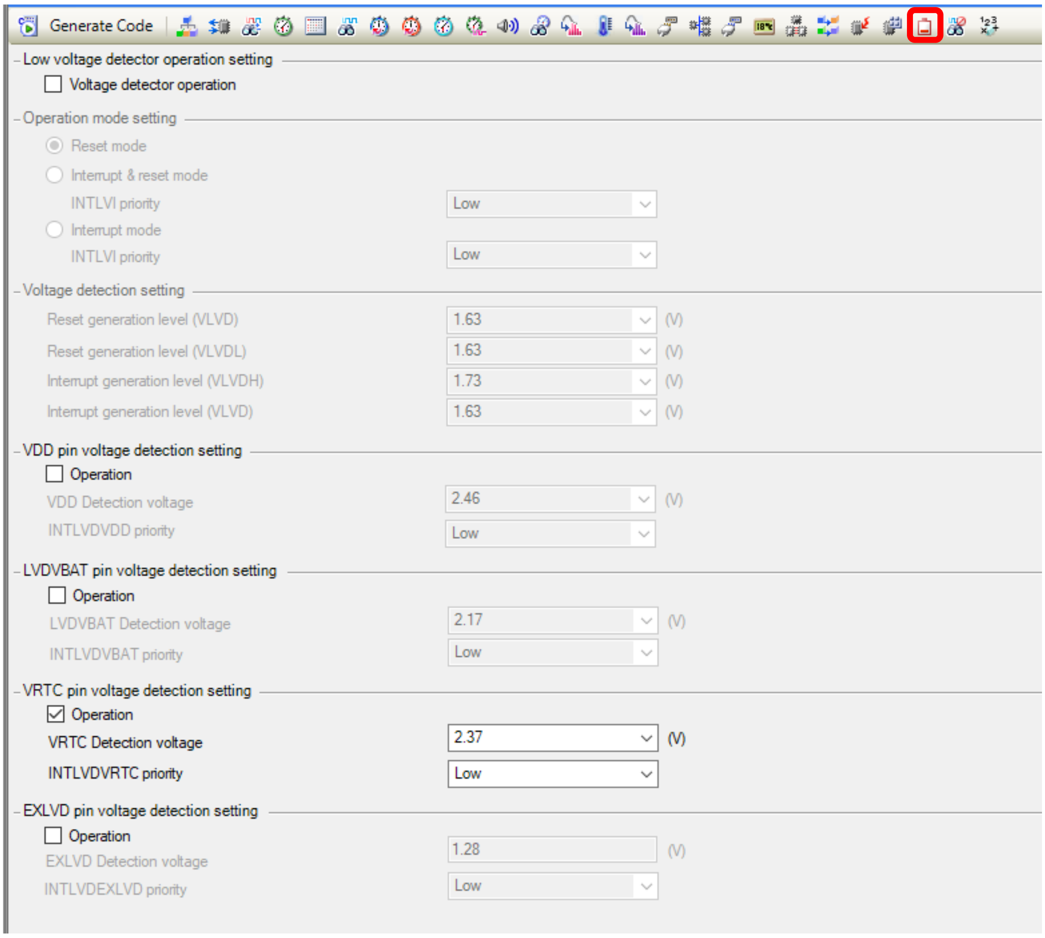
7. Configuration of **Interrupt Function**



INTP0 setting				
□ INTP0	Valid edge	Falling	Priority	Low
INTP1 setting				
□ INTP1	Valid edge	Falling	Priority	Low
INTP2 setting				
□ INTP2	Valid edge	Falling	Priority	Low
INTP3 setting				
□ INTP3	Valid edge	Falling	Priority	Low
INTP4 setting				
□ INTP4	Valid edge	Falling	Priority	Low
INTP5 setting				
□ INTP5	Valid edge	Falling	Priority	Low
INTP6 setting				
□ INTP6	Valid edge	Falling	Priority	Low
INTP7 setting				
□ INTP7	Valid edge	Falling	Priority	Low
INTP8 setting				
□ INTP8	Valid edge	Falling	Priority	Low
INTP9 setting				
□ INTP9	Valid edge	Falling	Priority	Low
INTP12 setting				
□ INTP12	Valid edge	Falling	Priority	Low
INTP13 setting				
□ INTP13	Valid edge	Falling	Priority	Low
INTP14 setting				
☑ INTP14	Valid edge	Falling	Priority	High
INTRTCIC0 setting				
□ INTRTCIC0 !	Valid edge	Falling	Priority	High
INTRTCIC1 setting				
□ INTRTCIC1	Valid edge	Falling	Priority	High
INTRTCIC2 setting				
□ INTRTCIC2 !	Valid edge	Falling	Priority	High

Figure 37: Interrupt Function Setting in Code Generator

8. Configuration of **Voltage Detection**



Low voltage detector operation setting
□ Voltage detector operation

Operation mode setting
● Reset mode
○ Interrupt & reset mode
INTLVI priority: Low
○ Interrupt mode
INTLVI priority: Low

Voltage detection setting
Reset generation level (VLVD): 1.63 (V)
Reset generation level (VLVDL): 1.63 (V)
Interrupt generation level (VLVDH): 1.73 (V)
Interrupt generation level (VLVD): 1.63 (V)

VDD pin voltage detection setting
□ Operation
VDD Detection voltage: 2.46 (V)
INTLVDVDD priority: Low

LVDVBAT pin voltage detection setting
□ Operation
LVDVBAT Detection voltage: 2.17 (V)
INTLVDVBAT priority: Low

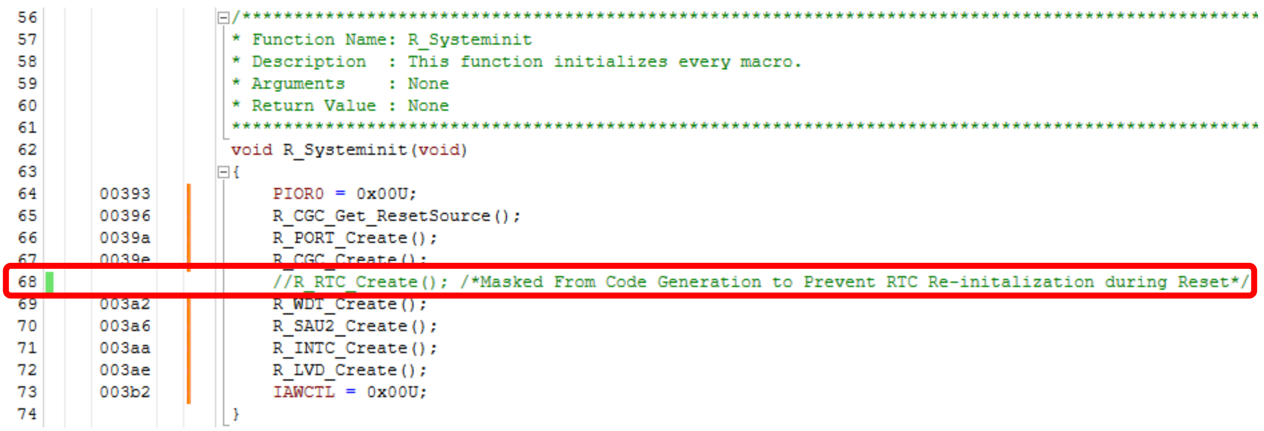
VRTC pin voltage detection setting
☑ Operation
VRTC Detection voltage: 2.37 (V)
INTLVDVRTC priority: Low

EXLVD pin voltage detection setting
□ Operation
EXLVD Detection voltage: 1.28 (V)
INTLVDEXLVD priority: Low

Figure 38: Voltage Detection Setting for Code Generator

6.3. Configuration of Source File

In the **r_cg_systeminit.c** source file, **Mask R_RTC_Create( )** at **line 68** of the **R_Systeminit**( ) function as shown below before building and downloading the code into the FPB.



```
56              /**************************************************************************************
57              * Function Name: R_Systeminit
58              * Description  : This function initializes every macro.
59              * Arguments    : None
60              * Return Value : None
61              ***************************************************************************************
62              void R_Systeminit(void)
63              {
64    00393         PIOR0 = 0x00U;
65    00396         R_CGC_Get_ResetSource();
66    0039a         R_PORT_Create();
67    0039e         R_CGC_Create();
68                  //R_RTC_Create(); /*Masked From Code Generation to Prevent RTC Re-initalization during Reset*/
69    003a2         R_WDT_Create();
70    003a6         R_SAU2_Create();
71    003aa         R_INTC_Create();
72    003ae         R_LVD_Create();
73    003b2         IAWCTL = 0x00U;
74              }
```

Figure 39: R_Systeminit Source File Configuration

7. Diving Deeper

1. To learn more about the RL78/I1C (512KB) Fast Prototyping Board, refer to the RL78/I1C (512KB) User's Manual available in the User Guides & Manuals of the RL78/I1C webpage at renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/rl78i1c-ultra-low-power-microcontrollers-high-end-smart-electricity-meter-market
2. Renesas provides several example projects that demonstrate different capabilities of the RL78/I1C (512KB) Fast Prototyping Board. These example projects can serve as a good starting point for users to develop custom applications. Example projects (source code and project files) are available in the RL78/I1C (512KB) Fast Prototyping Board Example Project Bundle.

8. Website and Support

Visit the following URLs to learn about the kit and the RA family of microcontrollers, download tools and documentation, and get support.

- RL78/I1C Resource renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/rl78i1c-ultra-low-power-microcontrollers-high-end-smart-electricity-meter-market
- RL78 Product Information renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus
- RL78 Knowledge Base en-support.renesas.com/knowledgeBase#31025
- Renesas Support en-support.renesas.com/dashboard

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	May 10th, 2021	−	Initial release

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.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

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/.

Trademarks

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