Integrated Development Environment e2 studio

R20AN0322EJ0101
Rev.1.01
Oct 01, 2014

How to use CUnit in e2 studio

Introduction

CUnit is a system for writing, administering, and running unit tests in C. It is built as a library (static or dynamic) which is linked with the user's testing code.

This document describes how to use CUnit in Renesas e^2 studio environment.

Contents

1. Introduction 2
2. How to use CUnit with Renesas e^2 studio environment 2
3. Reference Information 8
3.1 CUnit Web site 8

1. Introduction

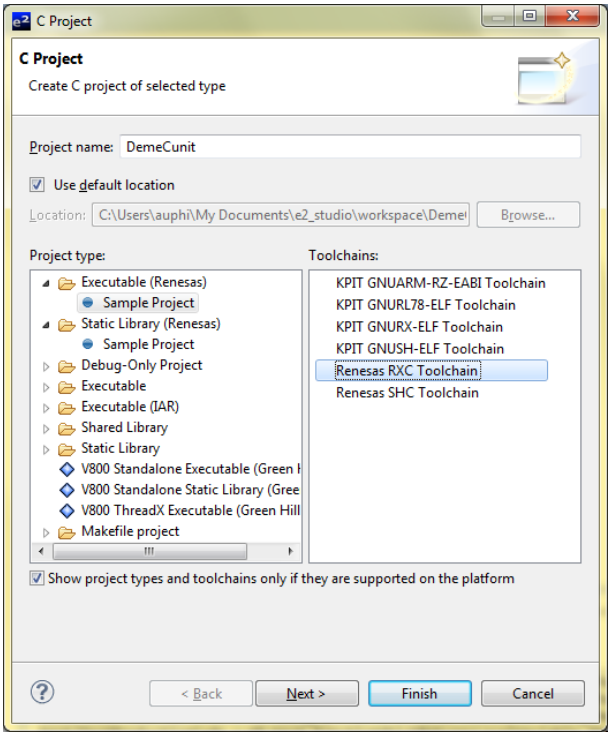
CUnit is a system for writing, administering, and running unit tests in C. It is provided as a (static or dynamic) library linkable with test target code.

CUnit uses a simple framework for building test structures, and provides a rich set of assertions for testing common data types. In addition, several different interfaces are provided for running tests and reporting results. These include automated interfaces for code-controlled testing and reporting without user inputs, as well as interactive interfaces allowing the user to run tests and view results dynamically.

This document describes how to use CUnit in Renesas e^2 studio environment. If you would like to know more about CUnit, please refer to http://cunit.sourceforge.net/doc/index.html.

2. How to use CUnit with Renesas e^2 studio environment

1. Download CUnit from hhttp://sourceforge.net/projects/cunit/ . Extract zip file to get CUnit package.
2. Open e^2 studio, create C project with Renesas toolchain by:
   a. Select ‘File’ > ‘New’ > ‘C Project’
   b. On the ‘C Project’ dialog, specify name of project. Select ‘Executable (Renesas)’ category > ‘Sample Project’ > ‘Renesas RXC Toolchain’ > Press the ‘Finish’ button



3. Add new source files to project. For example:

source.c (This file contains the function to be tested.)

```
int add(int a, int b) {
  return a+b;
}

int subtract(int a, int b) {
  return a-b;
}
```

source.h (This file contains the prototype declaration of the function to be tested.)

```
#ifndef SOURCE_H_
#define SOURCE_H_

int add(int a, int b);
int subtract(int a, int b);

#endif /* SOURCE_H_ */
```

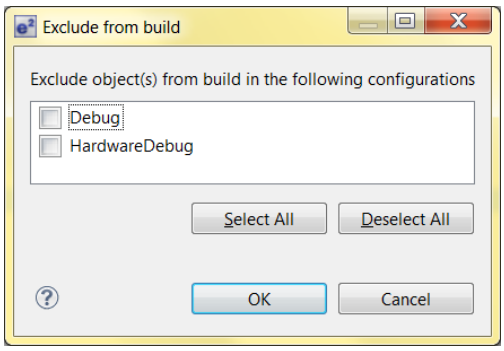
4. Copy CUnit files to project (contain CUnit assertion in order to use in your test).

- 'Basic.c' (CUnit-2.1-2\CUnit\Sources\Basic)

5. Copy folders from CUnit package to project (contain header and framework of CUnit)

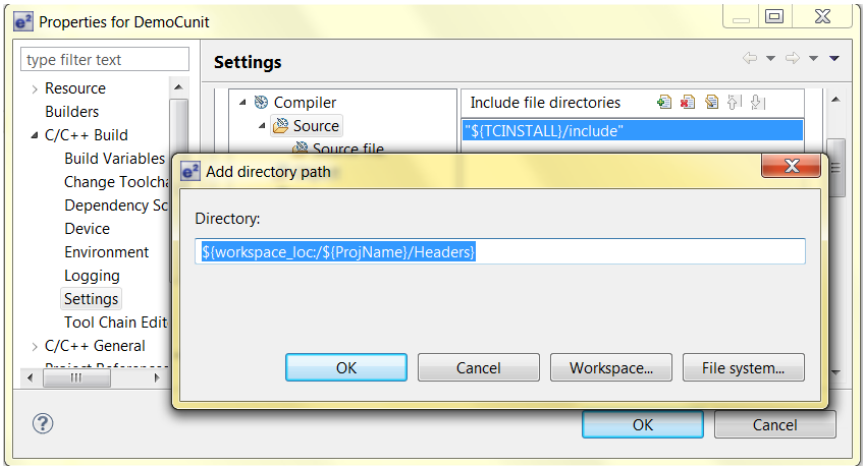
- 'Header' folder (CUnit-2.1-2\CUnit)
- 'Framework' folder (CUnit-2.1-2\CUnit\Sources)

'Header' and 'Framework' folders are not recognized as source folders because they are excluded from build. You can include these folders to build by: Right click the 'Header' and 'Framework' folders > Select 'Exclude from build' > Select the 'Deselect All' button > select the 'OK' button



6. 'include file directories' for 'Header' folders (to use header files)

You can add include file directories by: On 'Properties for DemoCunit' dialog, select 'C/C++ Build' > 'Settings' > select the 'Tool Settings' tab > 'Compiler' > 'Source' > press the 'Add…' button on toolbar > enter directory path on the 'Edit directory path' dialog > select the 'OK' button



7. Copy files to project (do this step to display result on Renesas Debug Virtual Console.) For these low-level interface routines, please refer to the compiler manual.

- lowlvl.src, lowsrc.c, lowsrc.h

8. Remove the comments in reset_program.c which contains //Use SIM I/O (to initialize C library Functions)

```
#ifdef __cplusplus                  // Use SIM I/O
extern "C" {
#endif
extern void _INIT_IOLIB(void);
extern void _CLOSEALL(void);
#ifdef __cplusplus
}
#endif

    _INIT_IOLIB();                  // Use SIM I/O

    _CLOSEALL();                    // Use SIM I/O
```

9. Add ‘testsource.c’ to project (This file contains functions to test functions in ‘source.c’.)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "CUnit.h"
#include "source.h"
//This is a test case used to test add() function in source.c
static void test_Add_01(void)
{
  //Equal Assertion is used in this test case.
  //1 is expected value, and add(1,0) is actual return value.
  //If expected value is not same assertion occurs.
  //we can refer the Reference document for the other useful assertion
  CU_ASSERT_EQUAL(1,add(1,0));
}
static void test_Add_02(void)
{
   CU_ASSERT_EQUAL(10,add(1,9));
}
//This is a test case used to test subtract() function in source.c
static void test_Subtract(void)
{
  //0 is expected value, and subtract(1,1) is actual return value.
  //If expected value is not same assertion occurs.
  CU_ASSERT_EQUAL(1,subtract(1,1));
}
//This is a test suite
static CU_TestInfo tests_Add[] = {
  //Register test case to test suite
  { "testAdd_01", test_Add_01 },
  { "testAdd_02", test_Add_02 },
  CU_TEST_INFO_NULL,
};
static CU_TestInfo tests_Subtract[] = {
   { "testSubtract", test_Subtract },
  CU_TEST_INFO_NULL,
};
//Declare the test suites in SuiteInfo
```

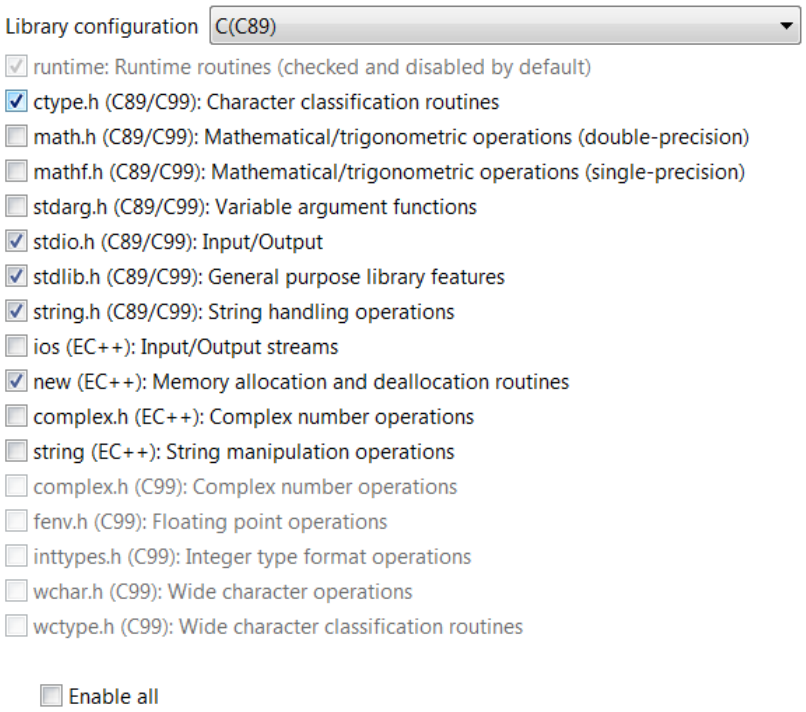
```
static CU_SuiteInfo suites[] = {
 { "TestSimpleAssert_AddSuite",  NULL,  NULL,   tests_Add },
 { "TestSimpleAssert_SubtractSuite", NULL,  NULL,    tests_Subtract },
  CU_SUITE_INFO_NULL,
};
void AddTests(void)
{
  //Retrieves a pointer to the current test registry.
  assert(NULL != CU_get_registry());
  //Flag for whether a test run is in progress
  assert(!CU_is_test_running());
  //Registers the suites in a single CU_SuiteInfo array
  if (CU_register_suites(suites) != CUE_SUCCESS) {
   //Get error message
   printf("suite registration failed - %s\n",CU_get_error_msg());
   exit(EXIT_FAILURE);
  }
}
```

10. Open ‘DemoCUnit.c’ (call test function, and run test case)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Basic.h"
void exit(long);
void abort(void);
int main(void);
extern AddTests();
int main(void)
{
  //Define the run mode for the basic interface.
  //Verbose mode - maximum output of run details.
     CU_BasicRunMode mode = CU_BRM_VERBOSE;
     //Define error action
     //Runs should be continued when an error condition occurs (if possible)
     CU_ErrorAction error_action = CUEA_IGNORE;
     //Initializes the framework test registry
     if (CU_initialize_registry()) {
       printf("\nInitialization of Test Registry failed.");
     }
     else {
       //call add test function
       AddTests();
       //Sets the basic run mode, which controls the output during test runs
       CU_basic_set_mode(mode);
       //Sets the error action
       CU_set_error_action(error_action);
       //Runs all tests in all registered suites
       printf("\nTests completed with return value %d.\n",
                CU_basic_run_tests());
       //Call this function to clean up
       //and release memory used by the framework
       CU_cleanup_registry();
     }
     return 0;
}
void abort(void){}
void exit(long exitcode){}
```

11. Open ‘sbrk.h’ file to re-define heap size (increase memory size for dynamic memory allocation on malloc and alloc functions)
    - `Old: #define HEAPSIZE 0x400`
    - `New: #define HEAPSIZE 0x800`
12. Add ‘ctype.h’ to build ‘Standard Library’ phase.

To add ‘ctype.h’: On ‘Properties for DemoCunit’ dialog, select ‘C/C++ Build’ > ‘Settings’ > select the ‘Tool Settings’ tab > ‘Standard Library’ > ‘Contents’ > Select the ‘ctype.h (C89/C99): Character classification routines’ entry > select the ‘OK’ button



13. Open ‘Framework\TestRun.c’ and ‘Framework\Util.c’ files. Change function ‘_snprintf’ to ‘printf’ (This step is optional, use for C89 only)

For example:

- `Old:  snprintf(buf, 33, "%d", number);`
- `New:  printf("%d", number);`

14. Add dummy files to project to build the project.

Add the ‘time.h’ to the ‘Headers’ folder

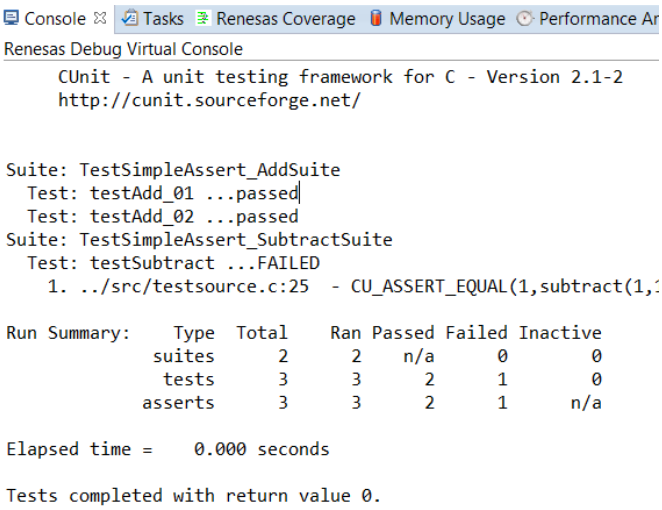
Eg:

```
typedef int clock_t;
#define CLOCKS_PER_SEC 1000
#define clock() (0)
```

15. Build project

16. Connect to the debugger, and then run the program.

17. Open ‘Renesas Debug Virtual Console’ panel to see the result of debug test



3. Reference Information

3.1 CUnit Web site

http://cunit.sourceforge.net/doc/index.html

Website and Support

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

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

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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 an MPU or MCU 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.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

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

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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 906, 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 Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0