

# RZ/T2M, RZ/T2ME, RZ/N2L, RZ/T2L Group

## Quick Start Guide: lwIP Protocol Stack Sample Program

### Introduction

This document describes the setup procedure of the sample program that implements lwIP (lightweight IP) protocol stack on the Renesas RZ/T2M, RZ/T2ME, RZ/N2L, RZ/T2L platform.

### Target Device

RZ/T2M, RZ/T2ME, RZ/N2L, RZ/T2L

### Contents

1. Overview .....	3
1.1 Abbreviations / Definitions .....	3
1.2 Reference .....	3
1.3 Limitation / Known Issue .....	3
2. Features .....	4
2.1 Package folder structure .....	4
3. Requirements (Software&Hardware) .....	5
3.1 RZ/T2M.....	5
3.2 RZ/N2L .....	6
3.3 RZ/T2L.....	7
4. Hardware Setup.....	8
4.1 RZ/T2M RSK Board.....	8
4.2 RZ/N2L RSK Board .....	9
4.3 RZ/T2L RSK Board.....	10
4.4 Connecting the Debugger .....	11
5. Host PC Settings .....	12
6. Running the Sample Application .....	13
6.1 Setting Up the e <sup>2</sup> studio Sample Project .....	13
6.1.1 Starting e <sup>2</sup> studio .....	13
6.1.2 Board IP address setting .....	15
6.1.3 How to generate source code and how to build .....	16
6.1.4 Downloading the application and running the debugger.....	19
6.2 Setting Up the EWARM Sample Project .....	21
6.2.1 Starting EWARM .....	21
6.2.2 Board IP address setting .....	22
6.2.3 How to generate source code and how to build .....	23
6.2.4 Downloading the application and running the debugger (single-core version).....	26

6.2.5	Downloading the application and running the debugger (Dual-Core Version))	28
7.	Demonstration of the Sample Application	30
7.1	Connecting to a PC	30
7.2	Echo Server	31
7.3	lwIP application	33
7.3.1	Steps to enable the application.	33
7.3.2	Execution steps for each application	36
8.	Software Specifications	38
8.1	Software Structure	38
8.1.1	Components Overview	39
8.1.2	Drivers Used	39
8.2	Task	40
8.2.1	Resources	40
8.2.2	Task Operations	41
8.2.3	Mirroring function for received packets	42
8.2.4	Ethernet Sending API	44
8.3	Memory Allocation	45
8.3.1	e <sup>2</sup> studio project	45
8.3.2	EWARM project	52
8.4	CPU Utilization	58
	Revision History	59

## 1. Overview

This document describes the setup procedure of the sample program for RZ/T2M, RZ/T2ME, RZ/N2L, RZ/T2L of Industrial Network

Please note that since RZ/T2ME can run using RZ/T2M project files, we will omit the specific description for RZ/T2ME.

### 1.1 Abbreviations / Definitions

Table 1.1. Abbreviations/Definitions

Index	Abbreviations /Definitions	Description
1	IP	Internet Protocol
2	TCP	Transmission Control Protocol
3	USB	Universal Serial Bus
4	PC	Personal Computer
5	SW	Switch
6	EWARM	Embedded Workbench® for ARM
7	lwIP	lightweight IP

### 1.2 Reference

Technical information about RZ/T2M, RZ/N2L, RZ/T2L is available via Renesas.

Table 1.2 Technical Inputs for RZ/T2M

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZT2M hardware.	Renesas Starter Kit+ for RZ/T2M,RZ/T2ME User's Manual	<a href="#">r20out4939ej****</a>
User's Manual	Provides technical details of the RZ/T2M microprocessor.	RZ/T2M Group User's Manual Hardware	<a href="#">r01uh0916ej****</a>

Table 1.3 Technical Inputs for RZ/N2L

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZN2L hardware.	Renesas Starter Kit+ for RZ/N2L User's Manual	<a href="#">r20out4984ej****</a>
User's Manual	Provides technical details of the RZ/N2L microprocessor.	RZ/N2L Group User's Manual Hardware	<a href="#">r01uh0955ej****</a>

Table 1.4 Technical Inputs for RZ/T2L

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+RZT2L hardware.	Renesas Starter Kit+ for RZ/T2L User's Manual	<a href="#">r20out5164ej****</a>
User's Manual	Provides technical details of the RZ/T2L microprocessor.	RZ/T2L Group User's Manual Hardware	<a href="#">r01uh0985ej****</a>

### 1.3 Limitation / Known Issue

None

## 2. Features

lwIP (lightweight IP) is an open-source embedded TCP/IP protocol stack. lwIP can implement full-scale TCP functionality with few resources and is suitable for use in embedded systems. lwIP was originally developed by Adam Dunkels at the Swedish Institute of Computer Science (SICS). It is now developed and maintained by a worldwide network of developers. For information on licensing lwIP, see "\common\oss\lwip\COPYING" in the sample program.

### 2.1 Package folder structure

Here is a folder overview of the main files:

Please refer to the release notes for the folder tree.

**Table 2.1. Folder structure**

Folder name	Overview
common/oss	lwip+freeRTOS package file
common/renesas/application	Applications
common/renesas/module/ether_netif	Ethernet API
common/renesas/module/lwip_port	lwip API
common/renesas/module/serial_io	Serial API for log output
SRC	Entry functions, etc.

### 3. Requirements (Software&Hardware)

This project was developed and tested in an environment using the following boards and tools:

#### 3.1 RZ/T2M

Table 3.1. RZ/T2M environment

Category	Name	Version	Description
Board	Renesas Starter Kit+ for RZ/T2M, RZ/T2ME	User's Manual 2.00	Renesas <a href="#">RZ/T2M-RSK - Renesas Starter Kit Plus for RZ/T2M   Renesas</a>
IDE	EWARM	9.60.3	IAR Systems <a href="#">IAR Embedded Workbench for Arm   IAR</a>
	e <sup>2</sup> studio	2025-12	RZ FSP <a href="#">About Flexible Software Package (FSP) for Renesas RZ series</a>
Configurator	FSP Smart Configurator	2025-12	
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.Rel1	
	GNU ARM A-Profile (AArch64 bare-metal)	—	
Emulator	J-Link™	8.60	SEGGER <a href="#">SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace</a>
	I-jet		IAR Systems <a href="#">IAR debug probes   IAR</a>
	Tera Term		Tera Term <a href="#">Tera Term Open Source Project</a>
Software	Wire shark		WIRESHARK <a href="#">Wireshark • Go Deep   Download</a>

## 3.2 RZ/N2L

Table 3.2. RZ/N2L environment

Category	Name	Version	Description
Board	Renesas Starter Kit+ for RZ/N2L	User's Manual 1.04	Renesas <a href="#">RZ/N2L-RSK - Renesas Starter Kit+ for RZ/N2L   Renesas</a>
IDE	EWARM	9.60.3	IAR Systems <a href="#">IAR Embedded Workbench for Arm   IAR</a>
	e <sup>2</sup> studio	2025-12	RZ FSP <a href="#">About Flexible Software Package (FSP) for Renesas RZ series</a>
Configurator	FSP Smart Configurator	2025-12	
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.Rel1	
	GNU ARM A-Profile (AArch64 bare-metal)	—	
Emulator	J-Link™	8.60	SEGGER <a href="#">SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace</a>
	I-jet		IAR Systems <a href="#">IAR debug probes   IAR</a>
	Tera Term		Tera Term <a href="#">Tera Term Open Source Project</a>
Software	Wire shark		WIRESHARK <a href="#">Wireshark • Go Deep   Download</a>

### 3.3 RZ/T2L

Table 3.3. RZ/T2L environment

Category	Name	Version	Description
Board	Renesas Starter Kit+ for RZ/T2L	User's Manual 1.21	Renesas <a href="#">RZ/T2L-RSK - Renesas Starter Kit+ for RZ/T2L   Renesas</a>
IDE	EWARM	9.60.3	IAR Systems <a href="#">IAR Embedded Workbench for Arm   IAR</a>
	e <sup>2</sup> studio	2025-12	RZ FSP <a href="#">About Flexible Software Package (FSP) for Renesas RZ series</a>
Configurator	FSP Smart Configurator	2025-12	
Flexible Software Package	FSP	4.0.0	
GCC Compiler	GNU ARM Embedded Toolchain	13.3.Rel1	
	GNU ARM A-Profile (AArch64 bare-metal)	—	
Emulator	J-Link™	8.60	SEGGER <a href="#">SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace</a>
	I-jet		IAR Systems <a href="#">IAR debug probes   IAR</a>
	Tera Term		Tera Term <a href="#">Tera Term Open Source Project</a>
Software	Wire shark		WIRESHARK <a href="#">Wireshark • Go Deep   Download</a>

## 4. Hardware Setup

This section explains the main hardware. For details about the board you are using, please refer to the [user's manual](#).

### 4.1 RZ/T2M RSK Board

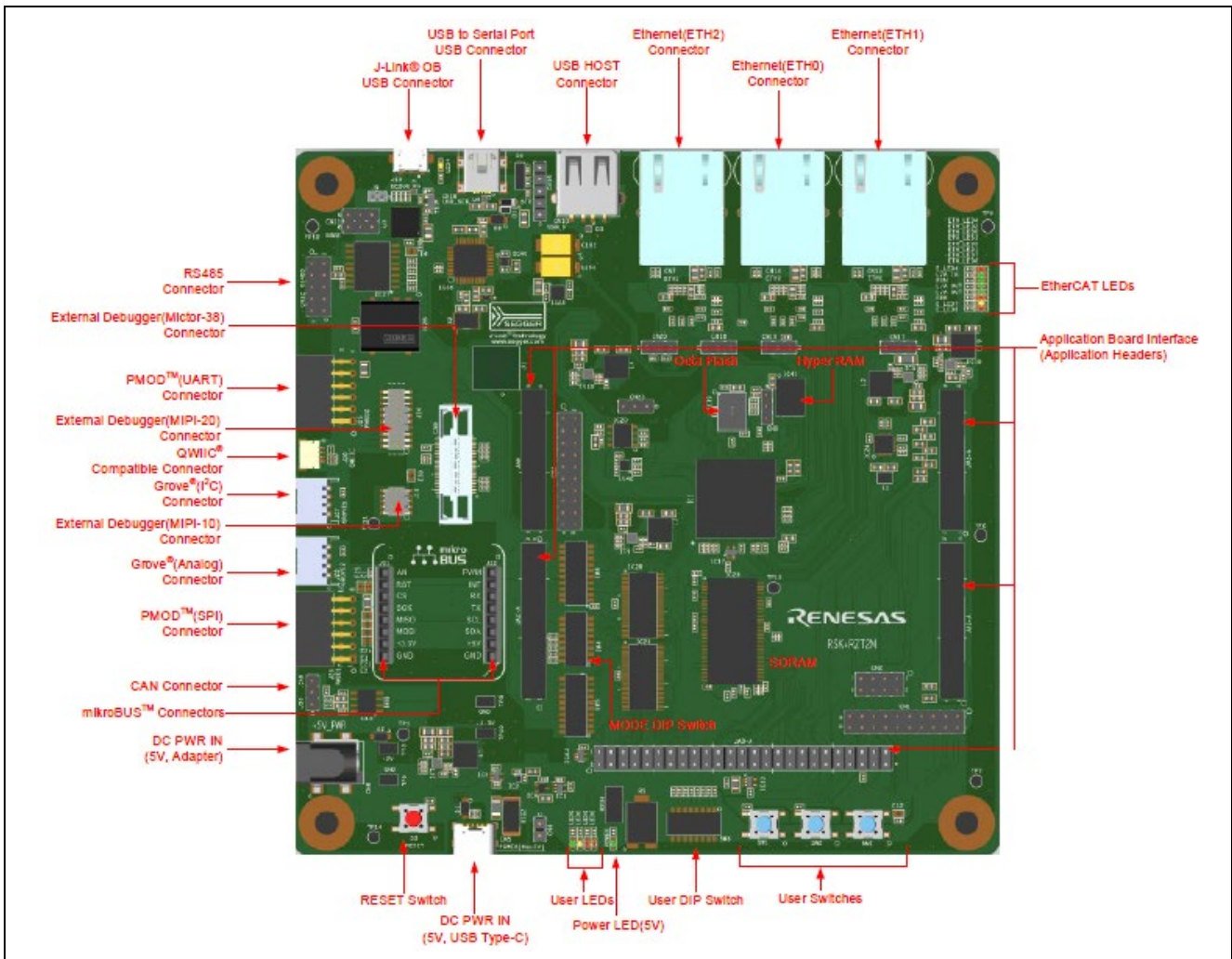


Figure 4.1 RZ/T2M RSK board layout

#### 1. DIP SW and Jumper SW settings

This is the initial setting in "[6. Configuration](#)" of the Renesas Starter Kit+ for RZ/T2M User's Manual.

- Connect the J-Link to micro USB or J20 with I-jet, switching between them using J9.
- Use a USB cable (Type-C) or an AC/DC adapter to supply power.
  - When using a USB cable, connect it to CN5.
  - When using an AC/DC adapter, connect it to CN6.
- Connect the Ethernet cable to ETH0, ETH1, or ETH2.

## 4.2 RZ/N2L RSK Board

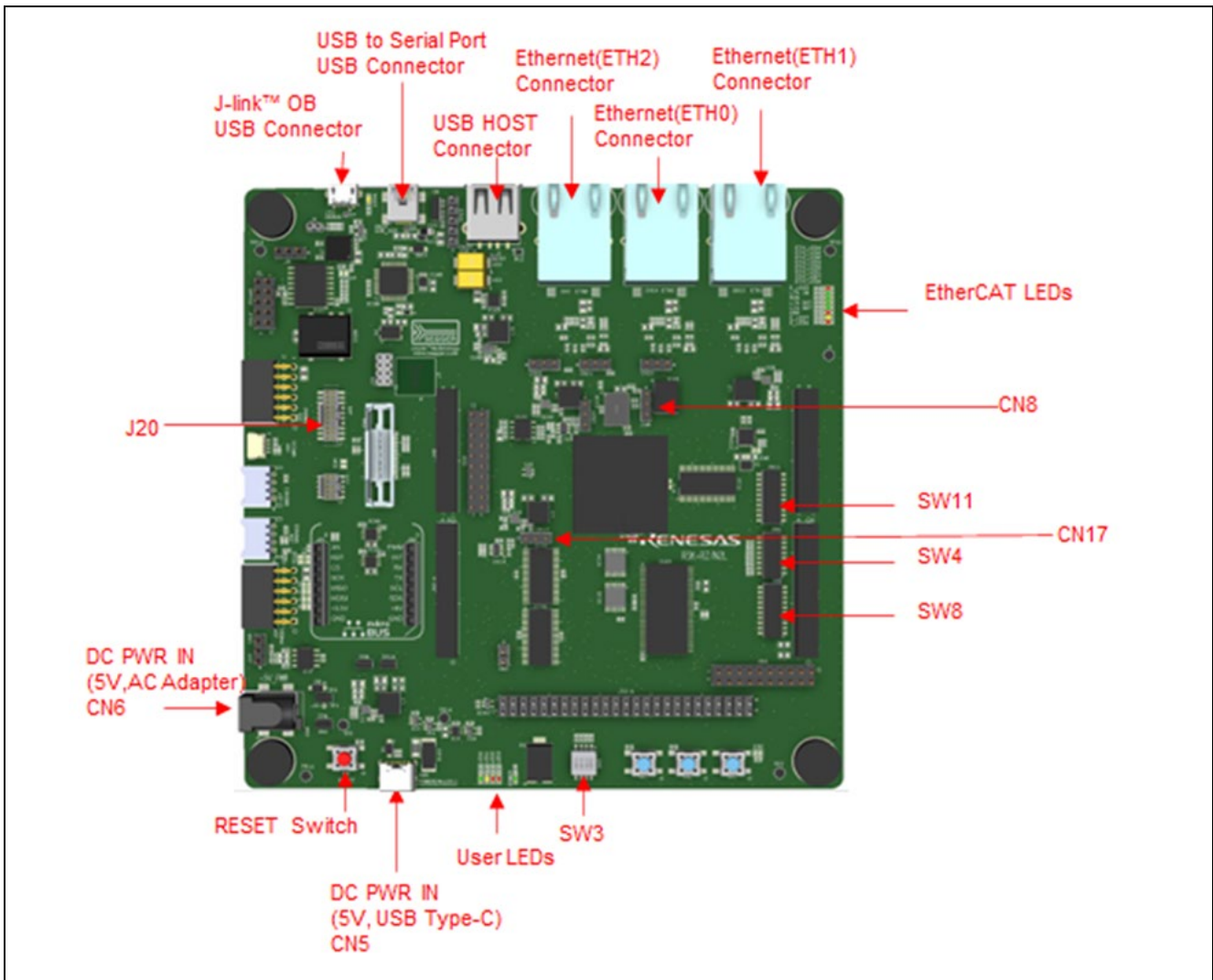


Figure 4.2 RZ/N2L RSK board layout

### 1. DIP Switch and Jumper Switch Settings

This setting is described in "[6. Configuration](#)" in the Renesas Starter Kit+ for RZ/T2L User's Manual.

2. Connect the J-Link to micro USB or J20 with I-jet, switching between them using J9.
3. Use a USB cable (Type-C) or an AC/DC adapter to supply power.
  - When using a USB cable, connect it to CN5.
  - When using an AC/DC adapter, connect it to CN6.
4. Connect the Ethernet cable to ETH0, ETH1, or ETH2.

### 4.3 RZ/T2L RSK Board

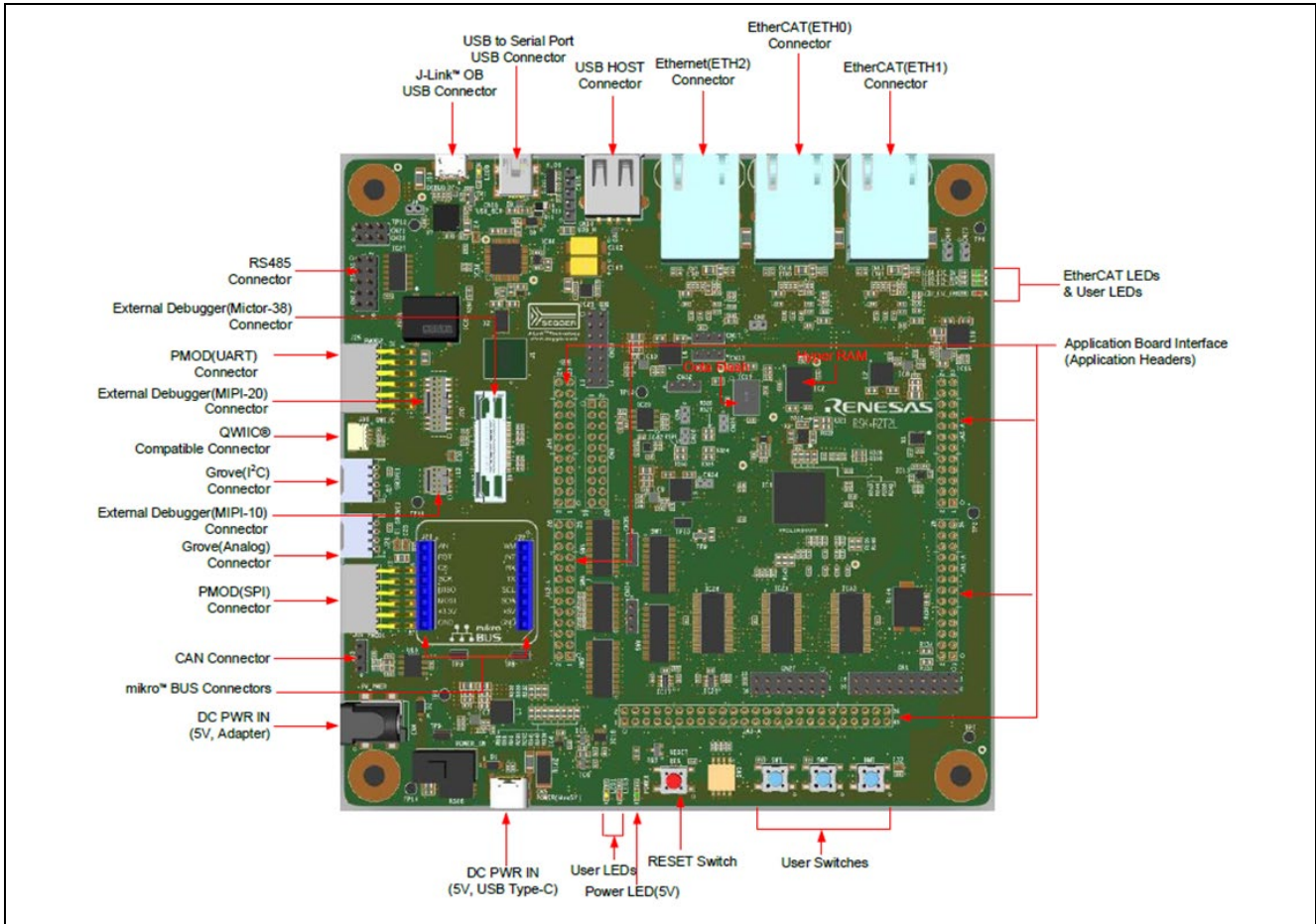


Figure 4.3 RZ/T2L RSK board layout

#### 1. DIP Switch and Jumper Switch Settings

This setting is described in "[6. Configurations](#)" in the Renesas Starter Kit+ for RZ/T2L User's Manual.

2. Connect the J-Link to micro USB or J20 with I-jet, switching between them using J9.
3. Use a USB cable (Type-C) or an AC/DC adapter to supply power.
  - When using a USB cable, connect it to CN5.
  - When using an AC/DC adapter, connect it to CN6.
4. Connect the Ethernet cable to ETH2.

### 4.4 Connecting the Debugger

Connect the evaluation board and debugger according to the tool you are using.

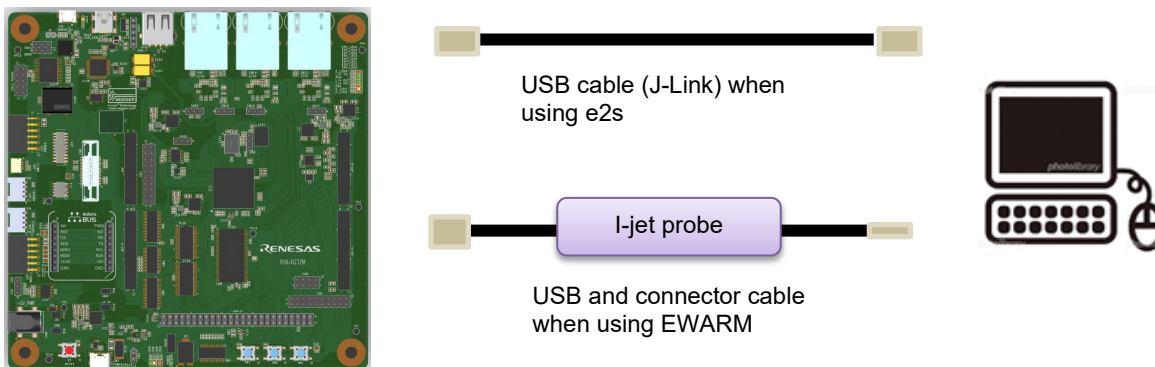


Figure 4. 4 Debugger Connection

## 5. Host PC Settings

See "[7. Demonstration of the Sample Application](#)"

## 6. Running the Sample Application

To run an EWARM project, proceed to [6.2. Setting Up the EWARM Sample Project.](#)

### 6.1 Setting Up the e<sup>2</sup> studio Sample Project

#### 6.1.1 Starting e<sup>2</sup> studio

1. Open e<sup>2</sup> studio and select the workspace directory.
2. Click "Open Projects from File System..." on the File tab.

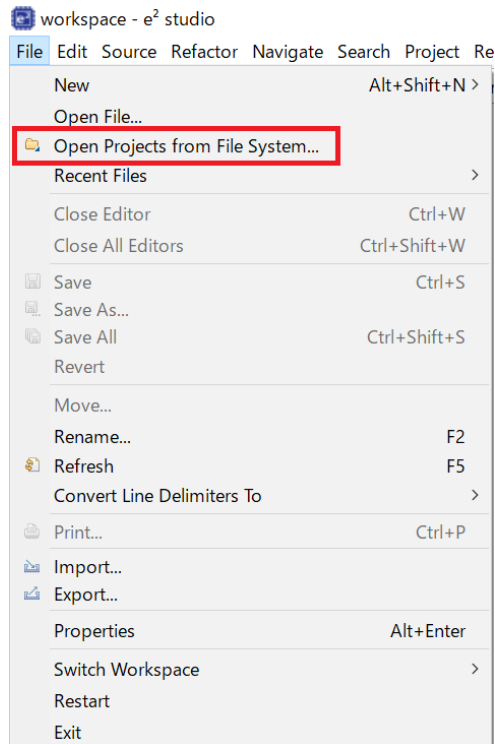


Figure 6.1 e<sup>2</sup>studio File tab

3. Import the project folder.

> RZ/T2M

Import the following two projects.

- The following project has no functionality other than blinking the LED.  
Project Name: RZT2M\_RSK\_lwIP\_cpu0 (hereafter referred to as "CPU0")  
Folder : project\rzt2m\_rsk\e2studio\RZT2M\_RSK\_lwIP\_cpu0
- The following project includes functionality related to "[7. Sample Application Demonstration.](#)"  
Project Name: RZT2M\_RSK\_lwIP\_cpu1 (hereafter referred to as "CPU1")  
Folder: project\rzt2m\_rsk\e2studio\RZT2M\_RSK\_lwIP\_cpu1

> RZ/N2L

Project Name: RZN2L\_RSK\_lwIP\_single (hereafter referred to as "CPU0")

Folder : project\rzn2l\_rsk\e2studio

> RZ/T2L

Project Name: RZT2L\_RSK\_lwIP\_single (hereafter referred to as "CPU0")

Folder : project\rzt2l\_rsk\e2studio

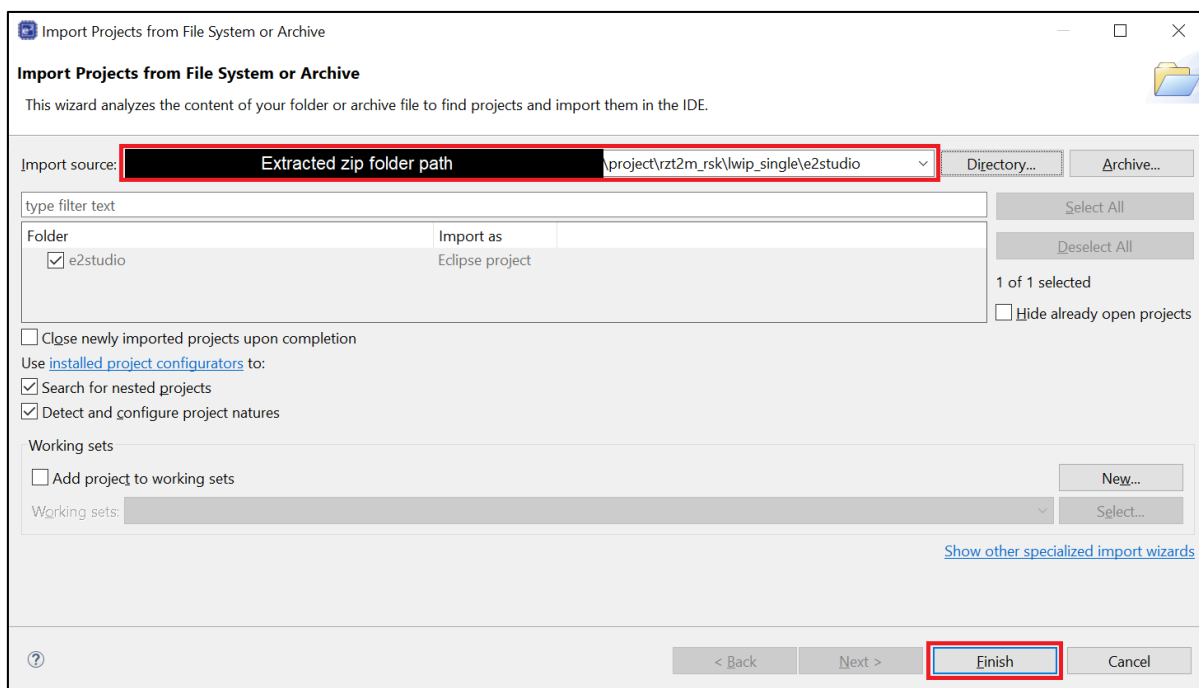


Figure 6.2 Import project on e<sup>2</sup>studio

## 6.1.2 Board IP address setting

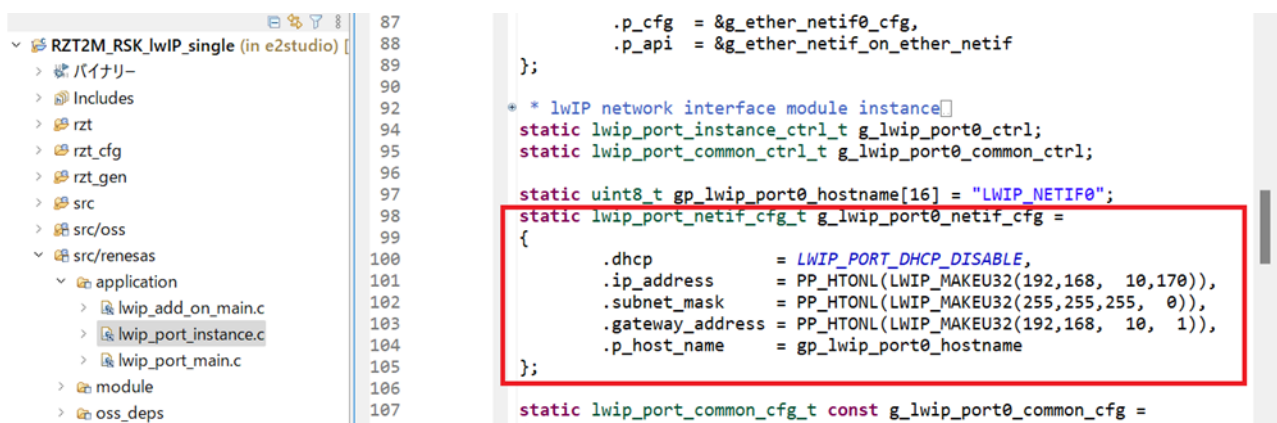
Set the IP address in the following procedure.

1. The IP address is set in `src/renesas/application/lwip_port_instance.c`.

The port number is set in `src/renesas/application/lwip_port_main.c`.

The following addresses are used in the example:

IP address : 192.168.10.170  
 Subnet mask : 255.255.255.065  
 Default gateway : 192.168.10.1  
 Port number : 65000

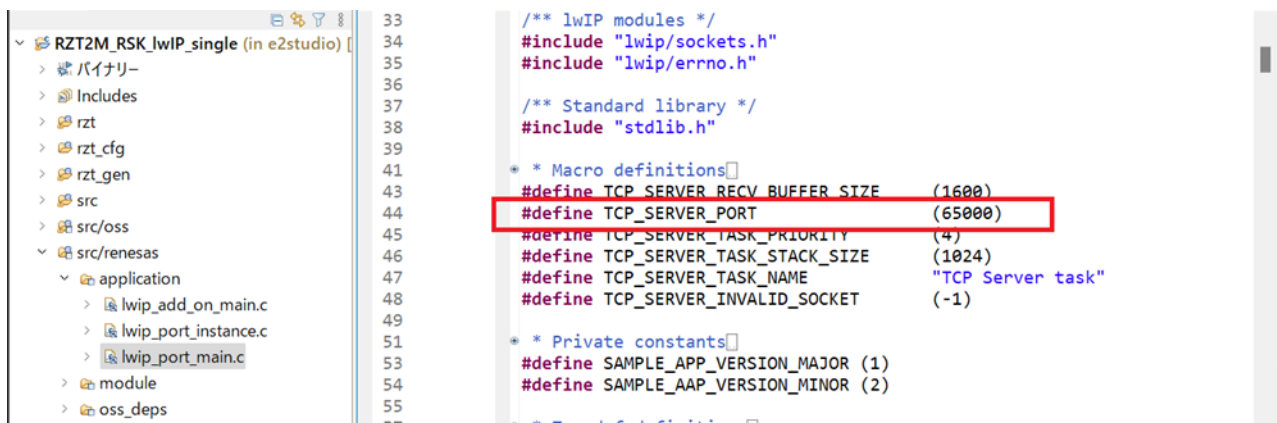


```

87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895

```

Figure 6.3 Static IP address



```

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895

```

Figure 6.4 Port number

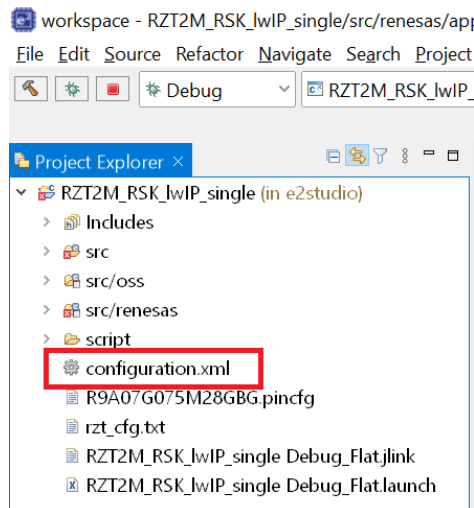
### 6.1.3 How to generate source code and how to build

If there is a script folder in the Project Explorer window before code generation, the following precautions are required.

- If you use anything other than the “ld” file in the folder, unexpected results may occur.
- If you change the boot mode or FSP version, the “ld” file will be rewritten at the time of code generation, so please save it in advance and restore it to the script before building.

If there is no change, there is no rewriting.

1. Click the “Configuration.xml”.



**Figure 6.5 Configuration**

2. On the "FSP Smart Configurator" screen that appears, check the following settings in the "BSP" tab and click "Generate Project Content". The generated folder will be added to your workspace.

For RZ/T2M, always start with CPU0.doku

- RAM boot (RAM execution without flash memory)
  - RZ/T2M
    - CPU0 and CPU1 have the same settings.
      - Board : RSK+RZT2M (RAM execution without flash memory)
  - RZ/N2L
    - Board : RSK+RZN2L (RAM execution without flash memory)
  - RZ/T2L
    - Board : RSK+RZT2L (RAM execution without flash memory)

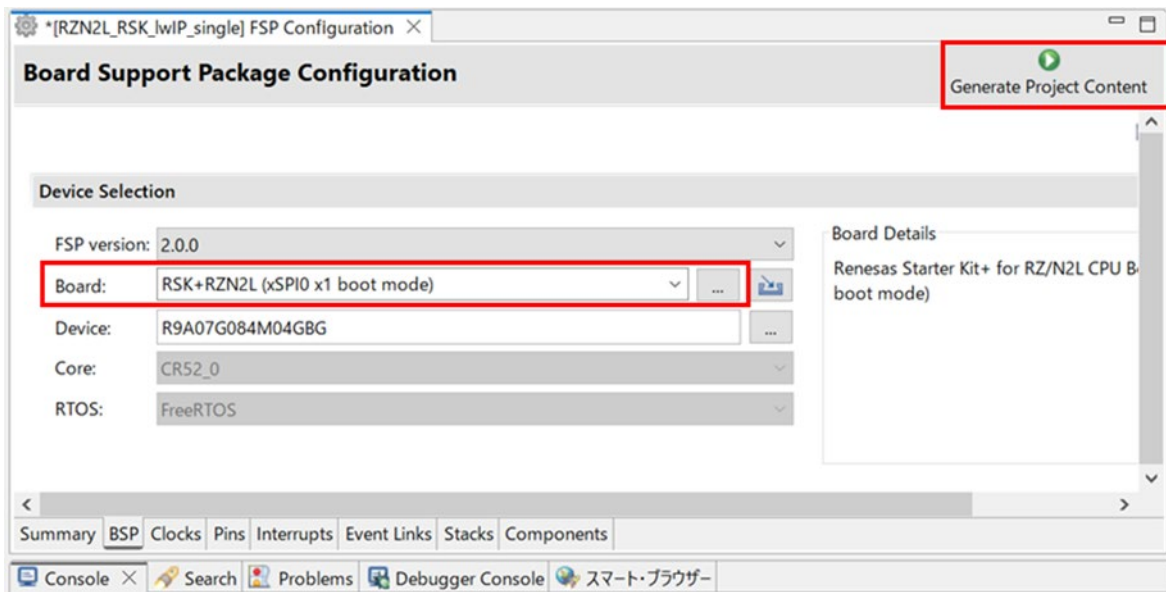


Figure 6.6 Generate Project Content

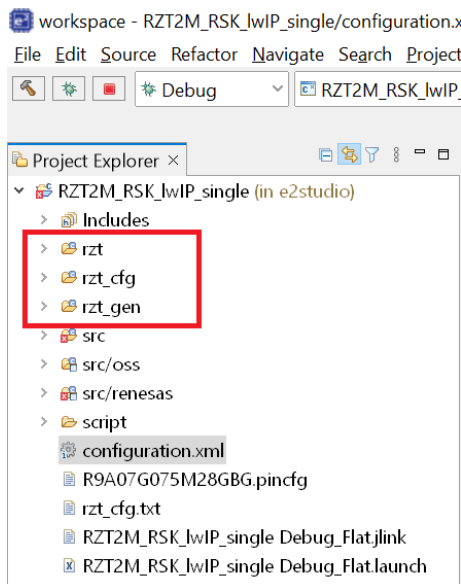


Figure 6.7 Generate project folder

- Click the Build button in tool bar to build the project and confirm that there is no error message in build message log.  
For RZ/T2M, always start with CPU0.

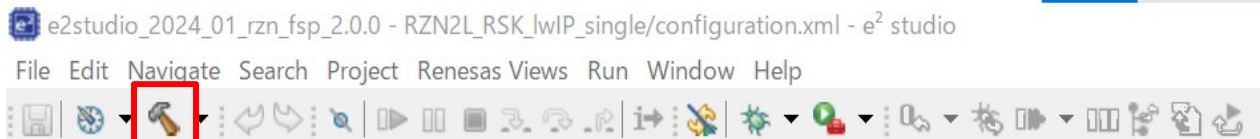
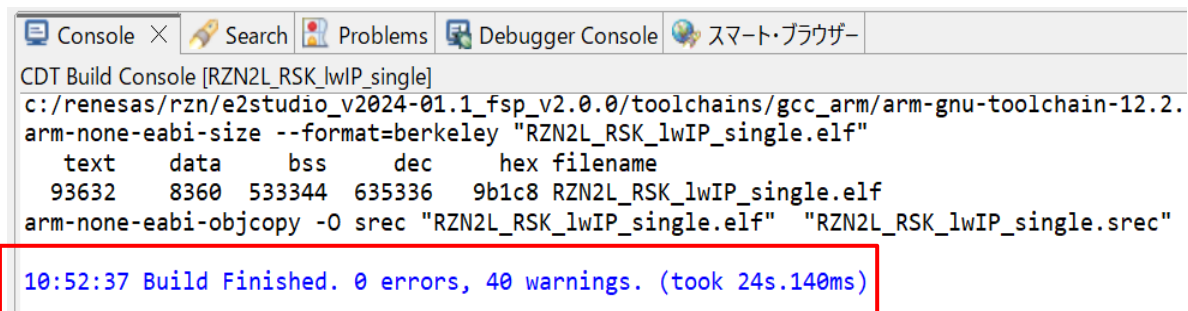


Figure 6.8 Build button



```
CDT Build Console [RZN2L_RSK_lwIP_single]
c:/renesas/rzn/e2studio_v2024-01.1_fsp_v2.0.0/toolchains/gcc_arm/arm-gnu-toolchain-12.2.
arm-none-eabi-size --format=berkeley "RZN2L_RSK_lwIP_single.elf"
  text  data  bss   dec   hex filename
 93632  8360 533344 635336 9b1c8 RZN2L_RSK_lwIP_single.elf
arm-none-eabi-objcopy -O srec "RZN2L_RSK_lwIP_single.elf" "RZN2L_RSK_lwIP_single.srec"

10:52:37 Build Finished. 0 errors, 40 warnings. (took 24s.140ms)
```

**Figure 6.9 Build message**

### 6.1.4 Downloading the application and running the debugger

Please follow these steps for each project.

1. Focus on the CPU0 project and click the Debug button on the toolbar to download the CPU0 program and launch the debugger.

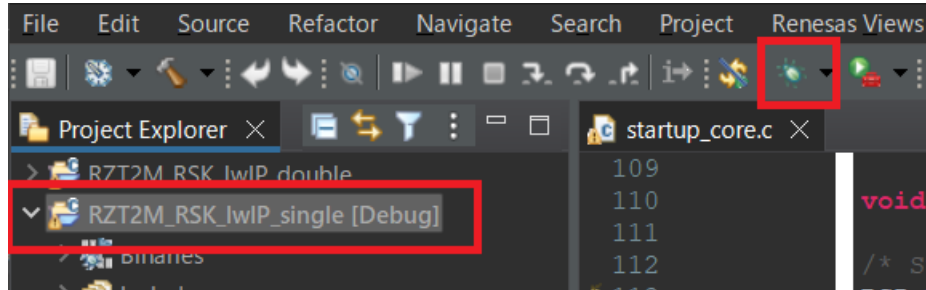


Figure 6.10 CPU0 Debug button

2. Dialog response  
When starting the debugger, if a dialog other than an error display appears depending on the usage environment, select "Switch", "Yes", "OK", etc. to close it.
3. The program will stop at the breakpoint set at startup, so simply click the Run button on the toolbar. For RZ/T2M, continue with CPU1.  
Before running, connect your PC as described in "[7. Sample Application Demonstration.](#)"

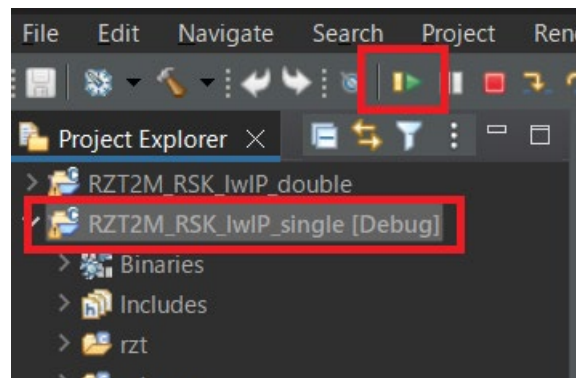


Figure 6.11 CPU0 Run button

- Next, focus on the CPU1 project and click the Debug button on the toolbar to download the CPU1 program and launch the debugger.  
At that time, the following dialog box will appear, so please click "No".

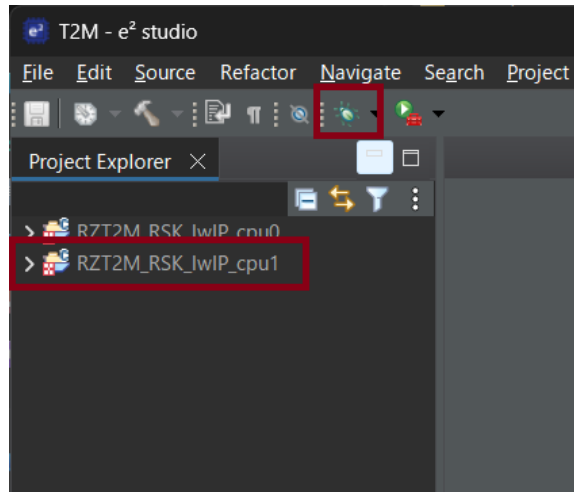


Figure 6.12 CPU1 Debug button

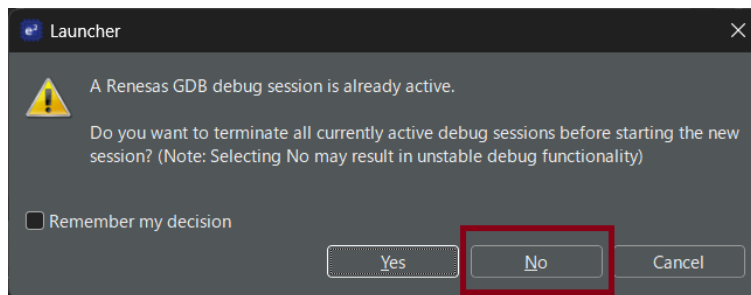


Figure 6.13 CPU1 Duplicate confirmation

- Other Dialogs**  
If a dialog other than an error message appears due to your environment or other reasons, close it by selecting "Switch," "Yes," or "OK."
- The program will stop at the breakpoint set at startup, so simply click the Run button on the toolbar.

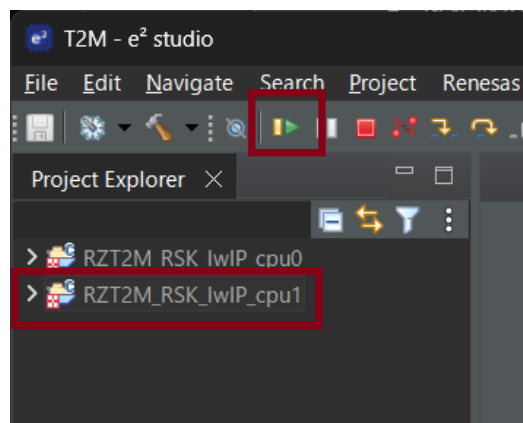


Figure 6.14 CPU1 Run button

## 6.2 Setting Up the EWARM Sample Project

For each of the project files listed below, please perform steps 1 through 3 below.

### 6.2.1 Starting EWARM

1. Open the EWARM.
2. Click "Open Workspace..." in File tab.

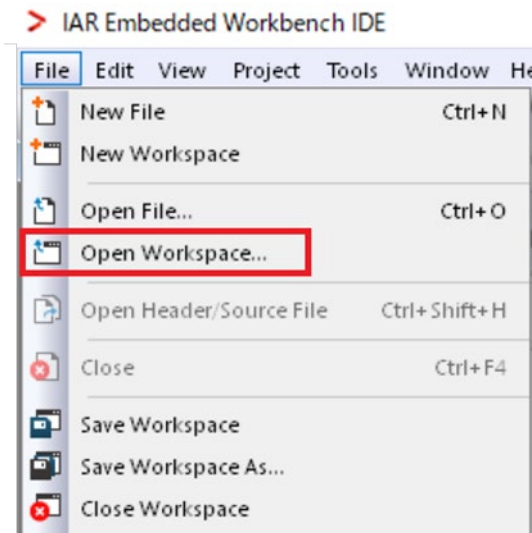


Figure 6.15 EWARM file tab

3. Select the Workspace File(.eww) and click the Open button.
  - RZ/T2M  
Open the following two projects in separate EWARMs.
    - The following project has no functionality other than blinking the LED.  
Project file: RZT2M\_RSK\_lwIP\_cpu0.eww (hereafter referred to as "CPU0")  
Folder: project\rzt2m\_rsk\ewarm\RZT2M\_RSK\_lwIP\_cpu0
    - The following project includes functionality related to "[7. Demonstration of the Sample Application](#)"  
Project file: RZT2M\_RSK\_lwIP\_cpu1.eww (hereafter referred to as "CPU1")  
Folder: project\rzt2m\_rsk\ewarm\RZT2M\_RSK\_lwIP\_cpu1
  - RZ/N2L  
Project file: RZN2L\_RSK\_lwIP\_single.eww (hereafter referred to as "CPU0")  
Import the folder: project\rzn2l\_rsk\ewarm.
  - RZ/T2L  
Project file: RZT2L\_RSK\_lwIP\_single.eww (hereafter referred to as "CPU0")  
Import the folder: project\rzt2l\_rsk\ewarm.

## 6.2.2 Board IP address setting

Set the IP address on the following procedure.

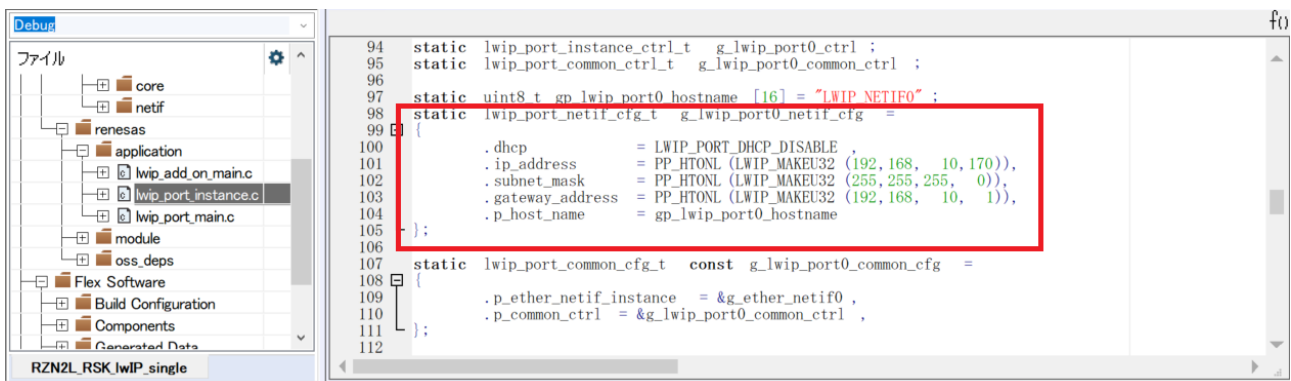
The default settings are "192.168.10.1" for the host PC and "192.168.10.170" for the device.

1. The IP address is set in `\common\renesas\application\lwip_port_instance.c`.

The port number is set in `\common\renesas\application\lwip_port_main.c`.

The following addresses are used in the example:

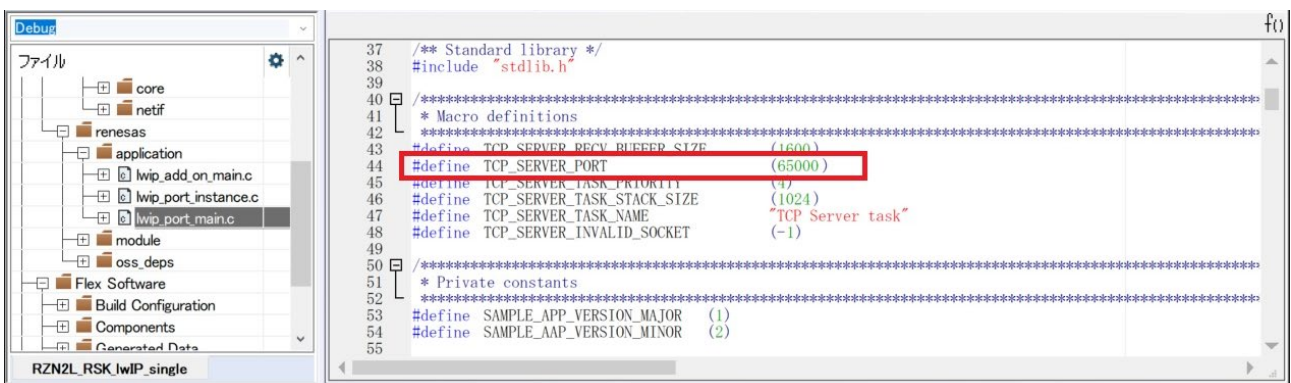
IP address : 192.168.10.170  
 Subnet mask : 255.255.255.0  
 Default gateway : 192.168.10.1  
 Port number : 65000



```

94 static lwip_port_instance_ctrl_t g_lwip_port0_ctrl ;
95 static lwip_port_common_ctrl_t g_lwip_port0_common_ctrl ;
96
97 static uint8_t gp_lwip_port0_hostname [16] = "LWIP_NETIF0" ;
98 static lwip_port_netif_cfg_t g_lwip_port0_netif_cfg =
99 {
100     .dhcp          = LWIP_PORT_DHCP_DISABLE ,
101     .ip_address    = PP_HTONL (LWIP_MAKEU32 (192, 168, 10, 170)),
102     .subnet_mask   = PP_HTONL (LWIP_MAKEU32 (255, 255, 255, 0)),
103     .gateway_address = PP_HTONL (LWIP_MAKEU32 (192, 168, 10, 1)),
104     .p_host_name   = gp_lwip_port0_hostname
105 };
106
107 static lwip_port_common_cfg_t const g_lwip_port0_common_cfg =
108 {
109     .p_ether_netif_instance = &g_ether_netif0 ,
110     .p_common_ctrl        = &g_lwip_port0_common_ctrl ,
111 };
112
  
```

Figure 6.16 Static IP address



```

37 /** Standard library */
38 #include "stdlib.h"
39
40 /* Macro definitions
41 *****
42 #define TCP_SERVER_RECV_BUFFER_SIZE (1600)
43 #define TCP_SERVER_PORT (65000)
44 #define TCP_SERVER_TASK_PRIORITY (4)
45 #define TCP_SERVER_TASK_STACK_SIZE (1024)
46 #define TCP_SERVER_TASK_NAME "TCP Server task"
47 #define TCP_SERVER_INVALID_SOCKET (-1)
48 *****
49
50 /* Private constants
51 *****
52 #define SAMPLE_APP_VERSION_MAJOR (1)
53 #define SAMPLE_APP_VERSION_MINOR (2)
54 *****
55
  
```

Figure 6.17 Port number

### 6.2.3 How to generate source code and how to build

#### 1. Notes on Code Generation.

- Using an icf file other than the one in the script folder may result in unexpected results.
- If you change the boot mode or FSP version, the icf file will be rewritten during code generation. Therefore, save it beforehand and restore it to the previously saved icf file before building. If there are no changes, the file will not be rewritten.
- For RZ/T2M, build projects for CPU0 and CPU1.
- For RZ/T2M, do not open two "FSP Smart Configurator" projects simultaneously.
- If you installed "FSP Smart Configurator" in a location other than the default installation path, open the "buildinfo.ipcf" file in the project folder and edit the path set in RASC\_EXE\_PATH to match the actual location of rasc.exe.

If you modify "buildinfo.ipcf," restart EWARM to reflect the changes.

- Click "FSP Smart Configurator" in the Tools tab.  
If you have not yet configured the FSP Smart Configurator in EWARM, see RZ/T2M, RZ/N2L, RZ/T2L Getting Started with Flexible Software Package (R01AN6434EJ). Section 5.4 explains how to set it up.

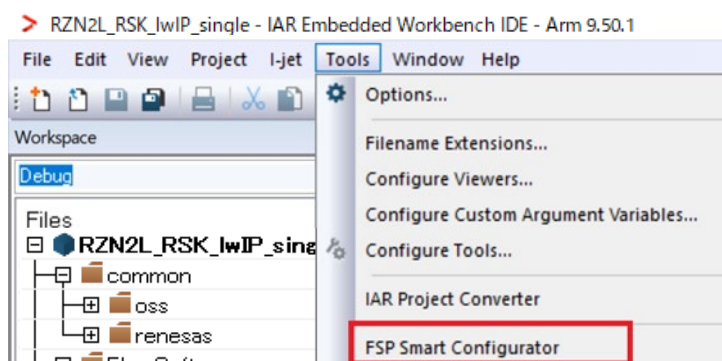


Figure 6.18 Tools tab

#### 2. Code Generation Procedure

After clicking "FSP Smart Configurator," check the following settings under the "BSP" tab on the screen that appears, and then click "Generate Project Content." The generated folder will be added.

##### ➤ RZ/T2M

CPU0 and CPU1 have the same settings.

Board : : RSK+RZT2M (RAM execution without flash memory)

##### ➤ RZ/N2L

Board : : RSK+RZN2L (RAM execution without flash memory)

##### ➤ RZ/T2L

Board : : RSK+RZT2L (RAM execution without flash memory)

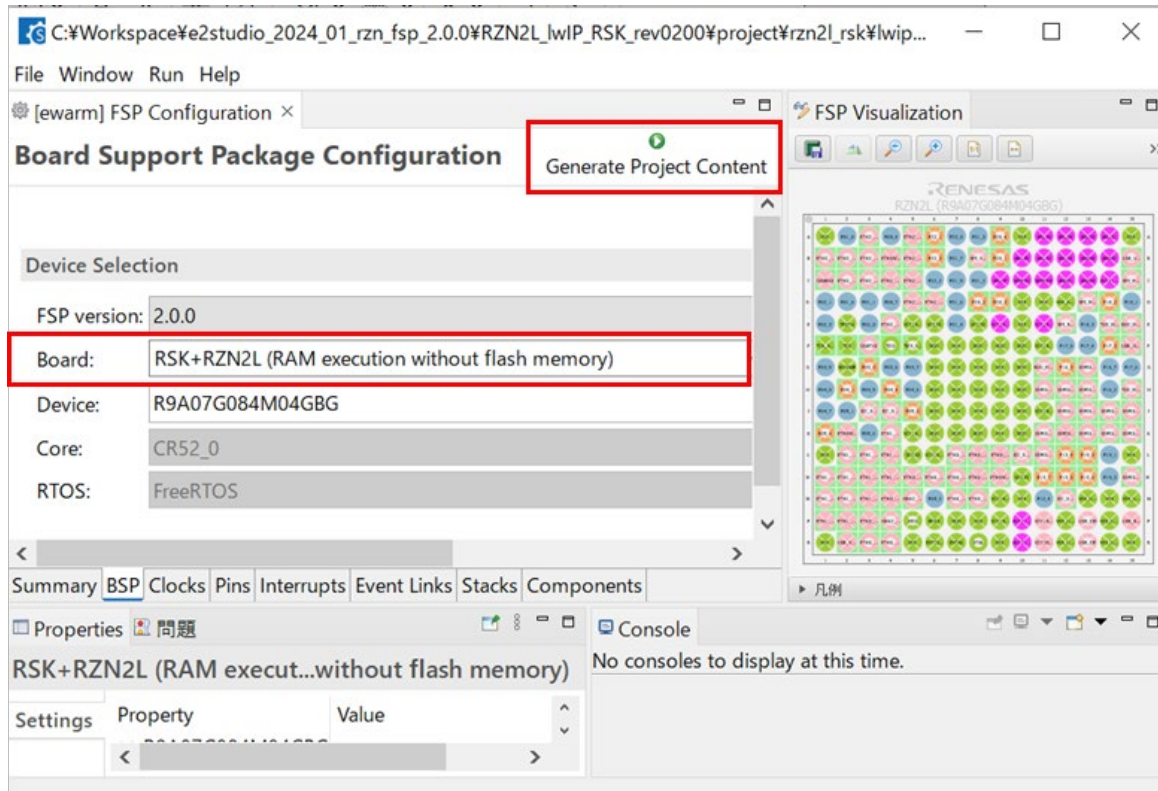


Figure 6.19 FSP SC Smart Configurator

### 3. Build Procedure

Click "Make" on the Project tab or click the button to build. When the build is complete, build messages will be displayed in the Build Console window, displaying the compilation target files and the number of errors/warnings.

For RZ/T2M models, please start execution from CPU0.

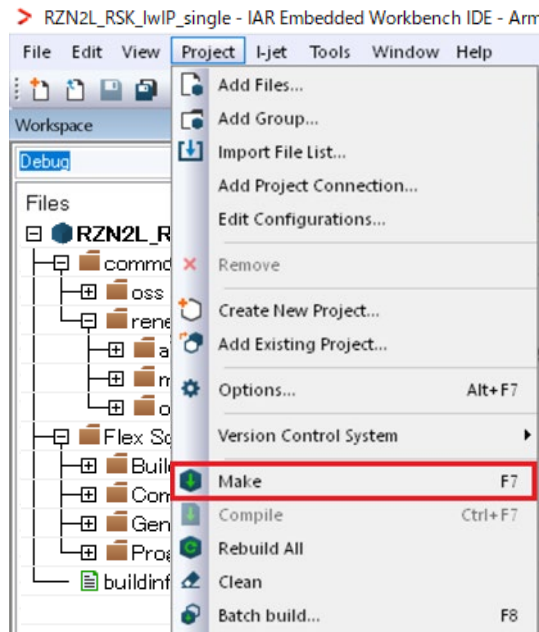


Figure 6.20 Make button 1

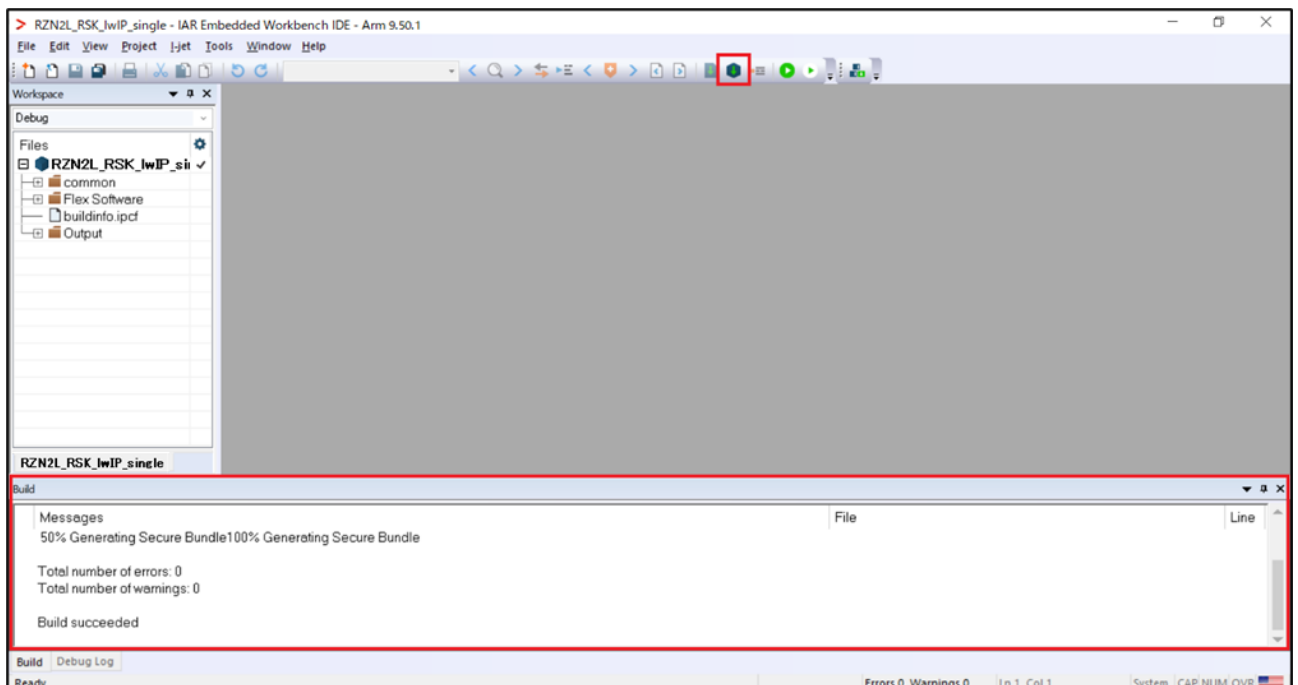


Figure 6.21 Make button 2 and Build console

### 6.2.4 Downloading the application and running the debugger (single-core version)

1. Click "Download and Debug" under "Project" on the menu bar, or click the Debug button to download the application program. After the debugger starts, the program will stop at the set breakpoint.

For RZ/T2M, skip this section and proceed to "[6.2.5. Downloading the Application and Running the Debugger \(Dual-Core Version\)](#)"

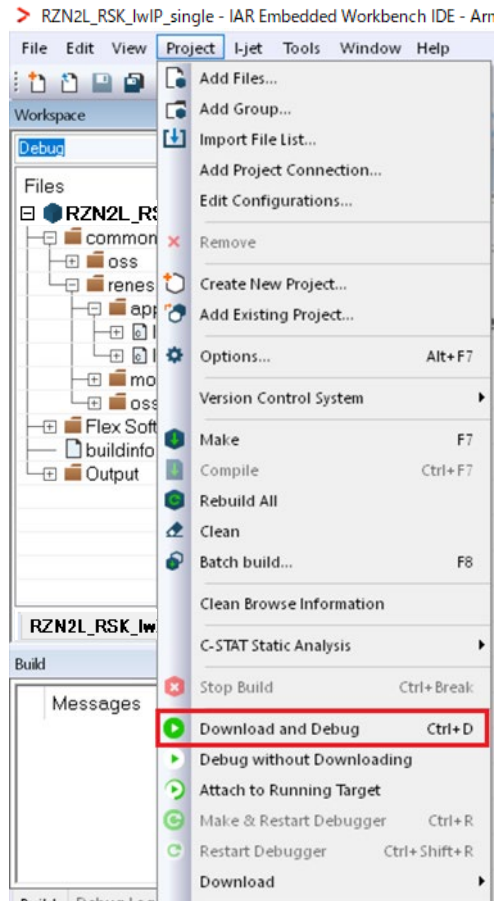


Figure 6.22 Download and Debug button

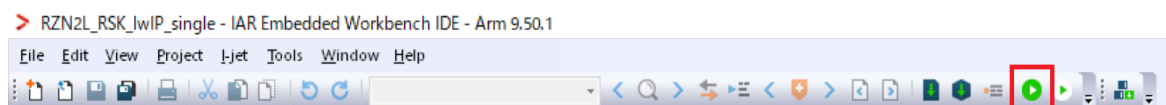


Figure 6.23 Debug button

- Click the Go button. If the program is running correctly, it will wait for a TCP/IP connection request.  
Before running, please connect your PC as described in "[7. Sample Application Demonstration.](#)"

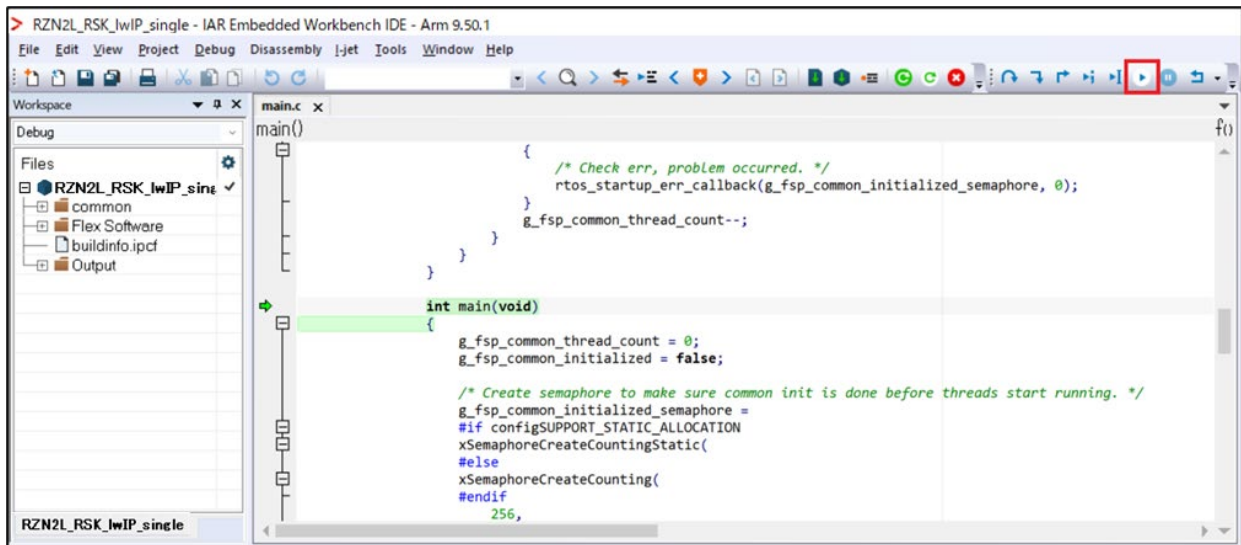


Figure 6.24 Break point and Go button

## 6.2.5 Downloading the application and running the debugger (Dual-Core Version)

Please perform the following steps.

1. Close EWARM for the CPU1 project.
2. Click "Download and Debug" in the "Project" menu bar, or click the Debug button to download the application program. After the debugger starts, the program will stop at the set breakpoint.

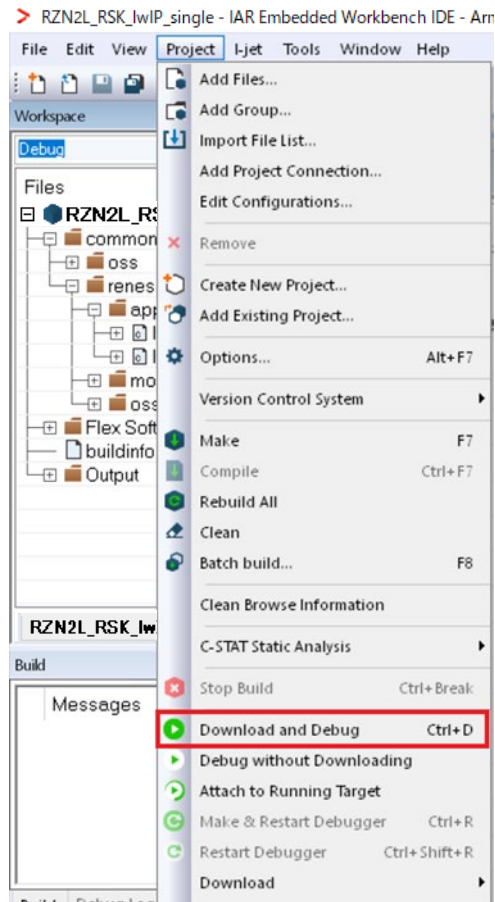


Figure 6.25 Download and Debug button

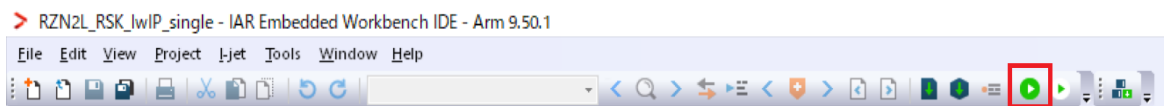


Figure 6.26 Debug button

- As CPU0 downloads, CPU0 will become the master and the CPU1 project will automatically start as the slave, and both projects will stop at the set breakpoints.
- Click the Run button in the CPU0 project, then run the CPU1 project in the same way.

Before running, connect via UART to a PC running terminal software.

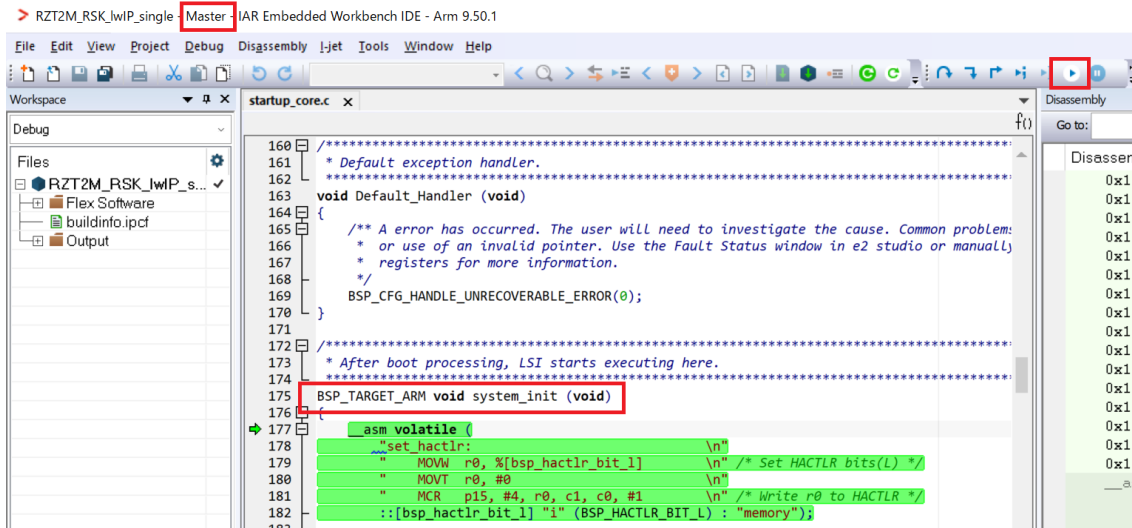


Figure 6.27 CPU0 Go button

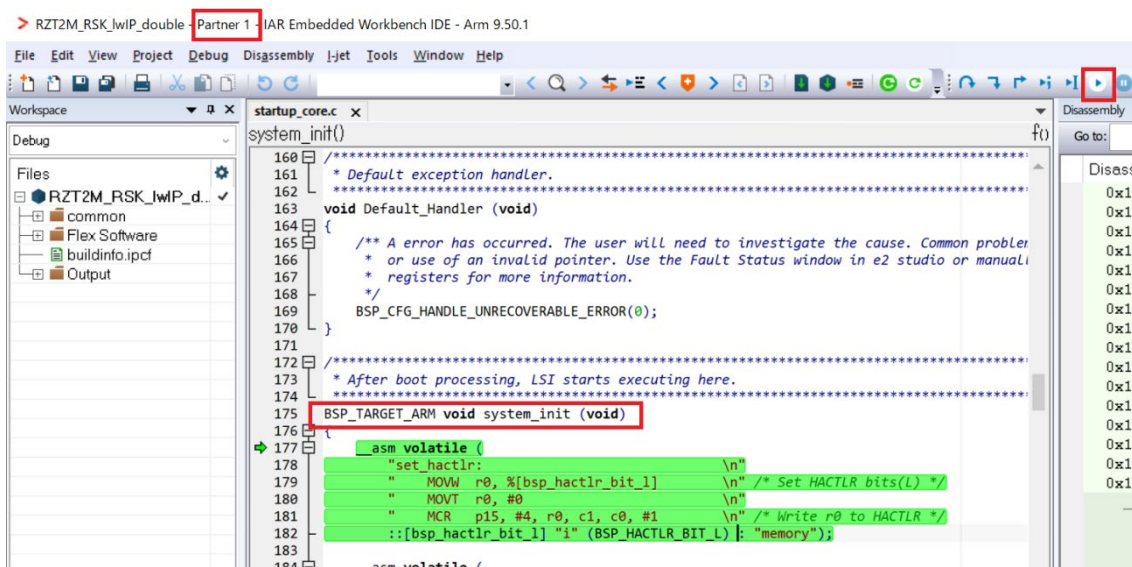


Figure 6.28 CPU1 Go button

## 7. Demonstration of the Sample Application

The sample program demonstrates the echo server functionality and the LWIP add-ons "iperf" and "SNTP" in the "apps" folder, which can be run with a conditional compiler.

### 7.1 Connecting to a PC

Follow the steps below to connect the evaluation board to the communication PC.

1. Connect the "USB to Serial Port" port shown in [4. Hardware Settings](#) to your PC using a USB cable.
2. To display log data from the evaluation board, launch terminal software such as "Tera Term" using the following protocol.

We will use "Tera Term" here.

- Port: Port added by connecting to the evaluation board
  - Speed: 115200
  - Other: DATA8 / NONE / 1 bit / Local Echo OFF
3. Connect the Ethernet port shown in [4. Hardware Settings](#) to your PC's NIC using a LAN cable.

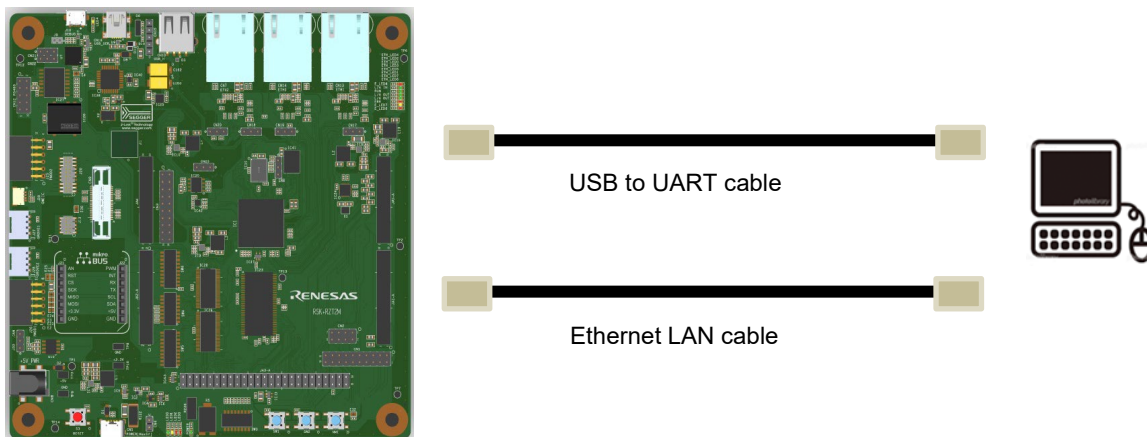
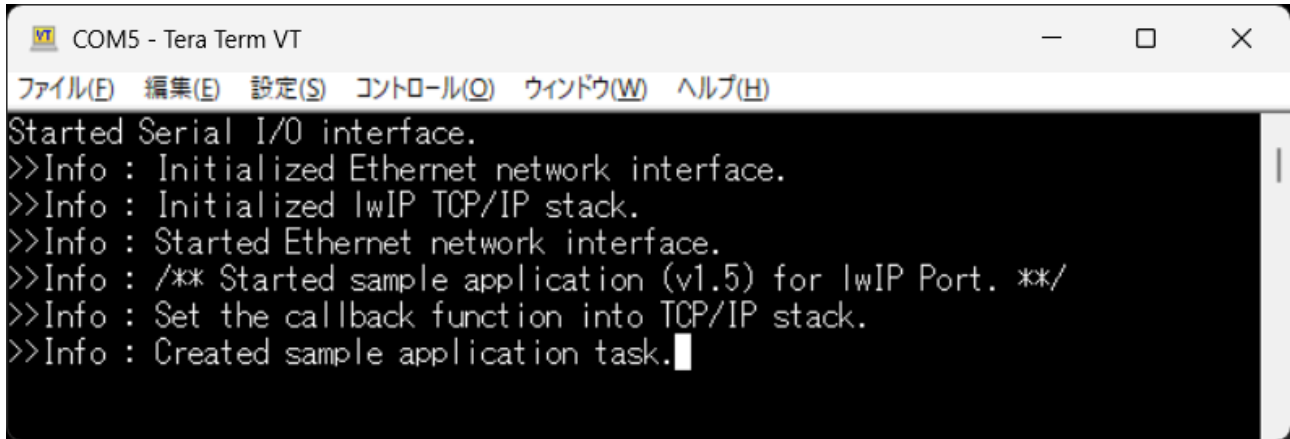


Figure 7.1 Communication terminal connection

## 7.2 Echo Server

After running the sample application, the following screen will appear and the echo server will start, allowing you to send characters from your PC via telnet.

Please send the following steps.



```
COM5 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
Started Serial I/O interface.
>>Info : Initialized Ethernet network interface.
>>Info : Initialized lwIP TCP/IP stack.
>>Info : Started Ethernet network interface.
>>Info : /** Started sample application (v1.5) for lwIP Port. **/
>>Info : Set the callback function into TCP/IP stack.
>>Info : Created sample application task.
```

Figure 7.2 Startup screen

1. Open "Tera Term" on your PC. Set it up as shown below.

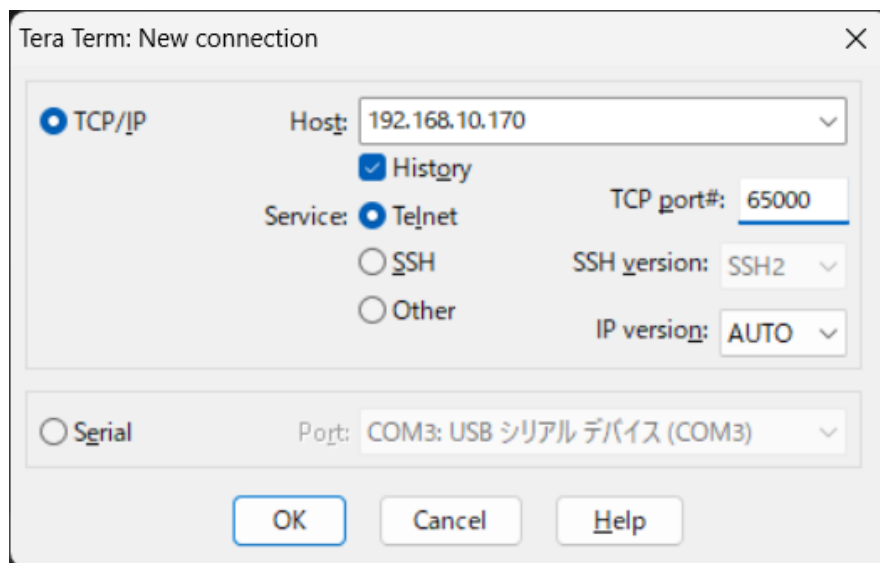


Figure 7.3 Setting connection for Tera Term

2. Enter any characters and press Enter.

In the following screen capture, the characters you entered are returned by the board.

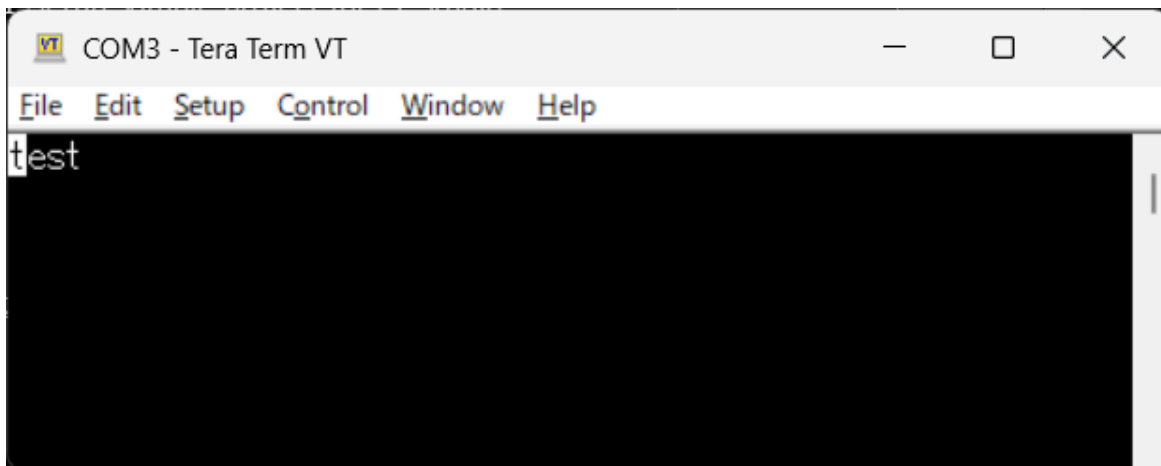


Figure 7.4 Demonstration

### 7.3 lwIP application.

This sample program can run the following applications as lwIP add-ons.  
Please note that this function is available in sample program version 1.2 or later.

As a demonstration, a working application.

- SNTP client function
- lwiperf (iperf2) server function
- lwiperf (iperf2) client function

#### 7.3.1 Steps to enable the application.

Each application can be started by enabling conditional compilation, and the settings are shown below.

1. Setting values for conditional compilation

Application	Setting values
SNTP client	Bit0=1: valid integer : 1 0: invalid integer : 0
lwiperf (iperf2) server	Bit1=1: valid integer : 2 0: invalid integer : 0
lwiperf (iperf2) client	Bit2=1: valid integer : 4 0: invalid integer : 0

2. Settings at build time

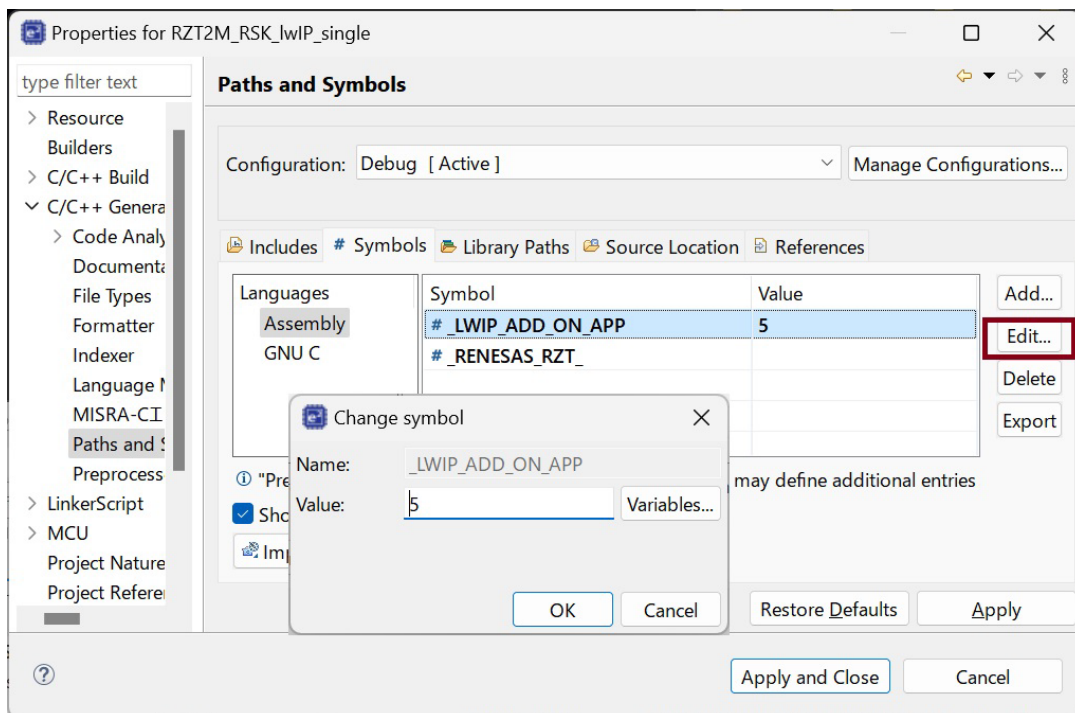
In each build environment, set the above setting values for the application you want to enable in the builder.

➤ e<sup>2</sup> studio

Follow the steps below to set it up.

In addition, when enabling multiple functions, please set with OR of each setting value.

- Project menu "Properties" → "C/C++ General" → "Paths and symbols" → "\_LWIP\_ADD\_ON\_APP" select it and press the edit button on the right to set it.
- After setting, follow the procedure in Section 3 of "6.1.3" to build.



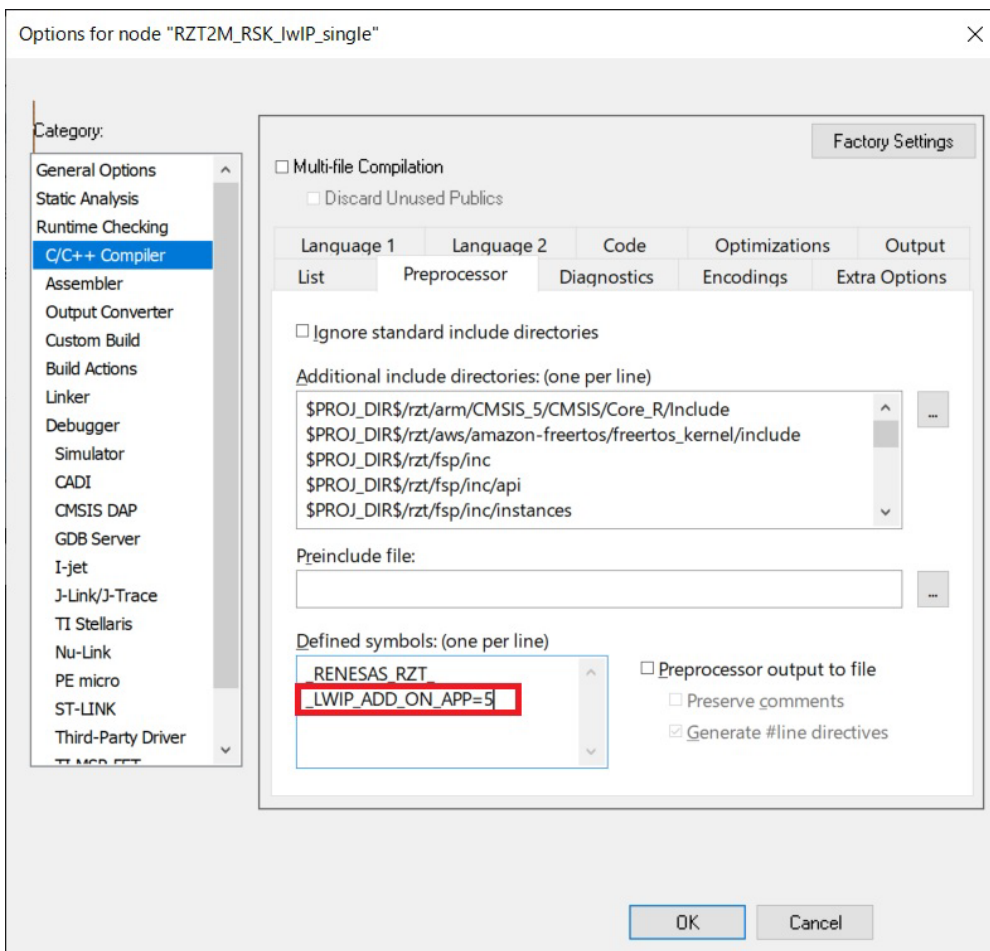
**Figure 7.5 e<sup>2</sup> studio**

➤ EWARM

Follow the steps below to set it up.

In addition, when enabling multiple functions, OR each setting value and set it.

- Project menu "Option" → "C/C++ Compiler" → "Preprocessor" → Set "\_LWIP\_ADD\_ON\_APP=5" directly in the symbol definition field.
- After setting, follow the procedure in Section 3 of "6.2.3" to build.



**Figure 7.6 EWARM**

3. Setting items required by the application

The required settings for each application are listed below.

Note that the IP addresses shown in red frames in Figure 7.7 and Figure 7.8 are setting examples, so be sure to set them according to your usage environment.

Application	Defined name	Setting contents
SNTP client	SNTP_SERVER_IP <a href="#">Figure 7.7</a> <a href="#">Figure 7.8</a>	IP address of the SNTP (NTP) server to connect to
lwiperf (iperf2) server	Use the "IP address" setting in 6.1.2 or 6.2.2	iperf2(lwiperf) server IP address
lwiperf (iperf2) client	IPERF_SERVER_IP <a href="#">Figure 7.7</a> <a href="#">Figure 7.8</a>	IP address of the iperf2 server to connect to

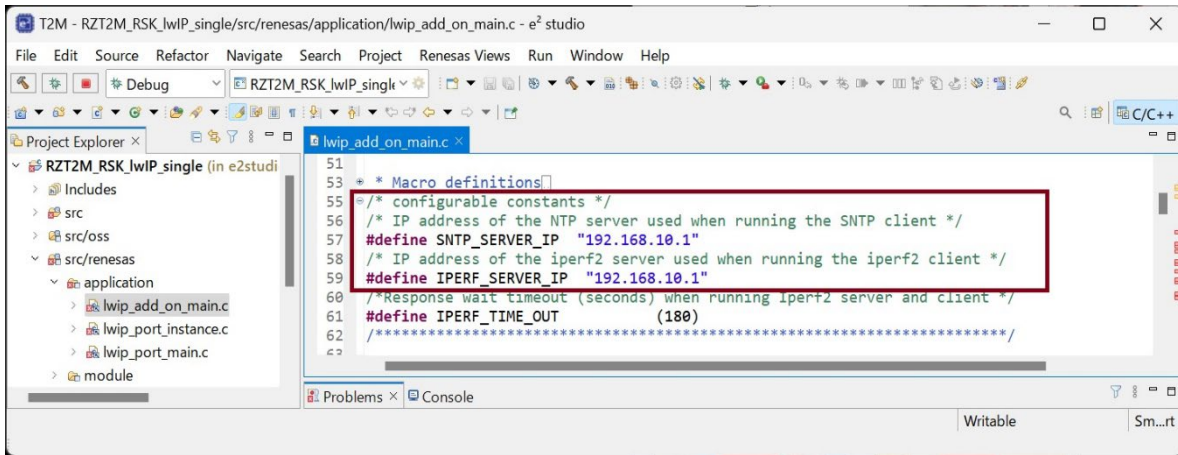


Figure 7.7 e2 studio

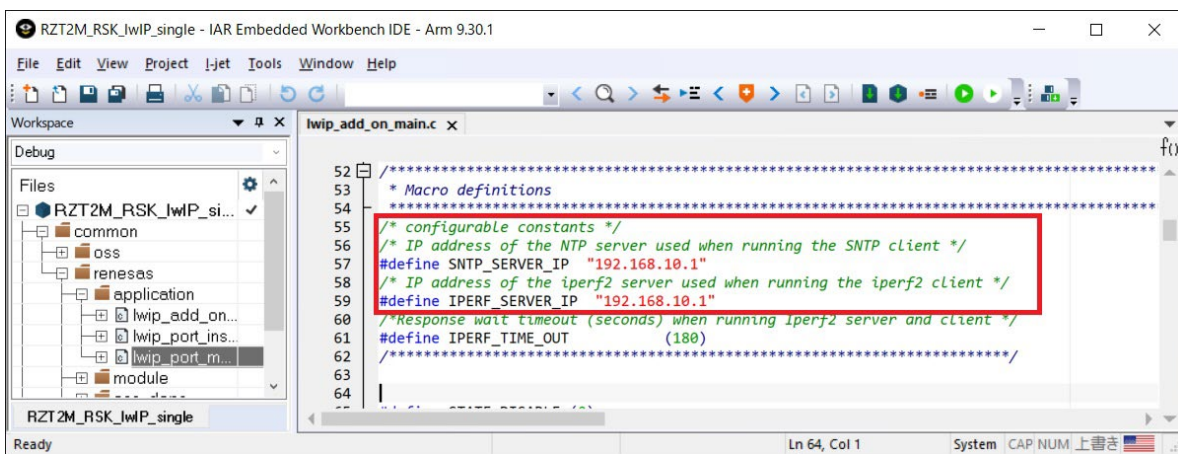


Figure 7.8 EWARM

- Reference information about connection destinations for each application  
The table below shows the applications to which each application connects.  
Each application does not specify the connection destination application.

Application	Connection destination application	how to get
SNTP client	Use Windows NTP server function	Enable the normally disabled NTP server functionality with a registry change. The setting method is introduced on the website.
Iwiperf (iperf2) server Iwiperf (iperf2) client	"Iperf 2" is a free software for Windows.	Please do a web search for "Iperf 2". Example of site <a href="https://sourceforge.net/projects/iperf2/">https://sourceforge.net/projects/iperf2/</a>

### 7.3.2 Execution steps for each application

The applications set in 2 of "7.3" will start in the following order according to the steps in 6.1.4 or 6.2.4. Here are the steps:

There is no difference in the procedure between e<sup>2</sup> studio and EWARM.

1. In the application startup order, the startup order decreases in the right direction.  
SNTP client > lwiperf (iperf2) server > lwiperf (iperf2) client

2. The steps to start each application are shown below.

➤ SNTP client

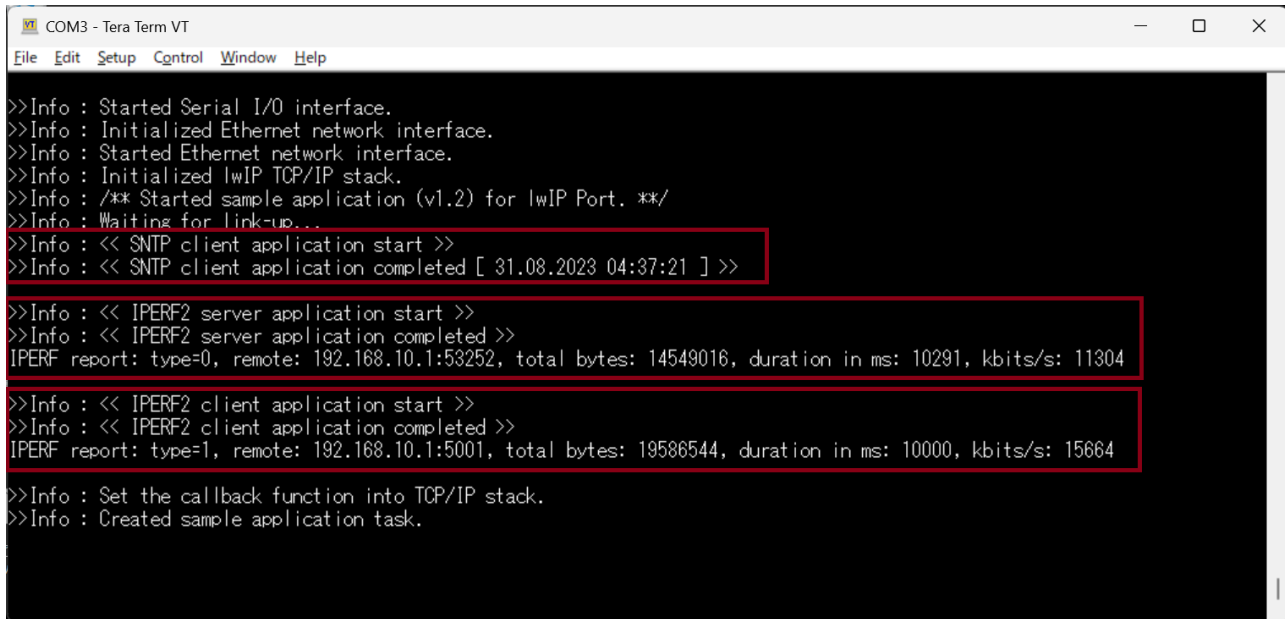
- Before starting the sample program, start the NTP server on the connected Windows PC.
- After displaying the sample program startup LOG, the SNTP client function starts, displays the next LOG, and waits for a response from the NTP server.  
"<< SNTP client application start >>"
- It depends on the communication environment, but normally the notification time will be displayed as shown below after waiting for a response for about 5 seconds, and the process will end.  
"<< SNTP client application completed [ 09.08.2023 00:00:00 ] >>"
- If 150 seconds have passed while waiting for a response from the NTP server, the following LOG will be displayed and the process will end.  
"<< SNTP client application end with error >>"

➤ lwiperf (iperf2) server

- After displaying the startup LOG of the sample program, if the lwiperf (iperf2) server function is enabled, the following LOG will be displayed, waiting for a connection from the lwiperf (iperf2) client.  
"<< IPERF2 server application start >>"
- After the above start LOG is displayed, start the lwiperf (iperf2) client from the connected Windows PC.
- Although it depends on the communication environment, it usually takes about 15 seconds after starting communication from the connection destination Windows PC, and the communication report is displayed as shown below and ends.  
"<< IPERF2 server application completed>>"  
"IPERF report: type=0, remote: 192.168.10.1:56861, total bytes: xxxxxxx, duration in ms: 10000, kbits/s: xxxxx "
- If there is no connection from the lwiperf (iperf2) client within 180 seconds, the following LOG will be displayed and the process will end.  
" << IPERF2 server application end with error >>"

➤ lwiperf (iperf2) client

- Before starting the sample program, start the iperf2 server on the connected Windows PC.
- After the sample program startup LOG is displayed, the lwiperf (iperf2) client function starts, the next LOG is displayed, and it waits for a response from the iperf2 server.  
"<< IPERF2 client application start >>"
- Depending on the communication environment, it usually takes about 15 seconds to display a communication report as shown below and finish.  
"<< IPERF2 client application completed >>"  
"IPERF report: type=1, remote: 192.168.10.1:5001, total bytes: xxxxxxx, duration in ms: 10000, kbits/s: xxxxx "



```
COM3 - Tera Term VT
File Edit Setup Control Window Help

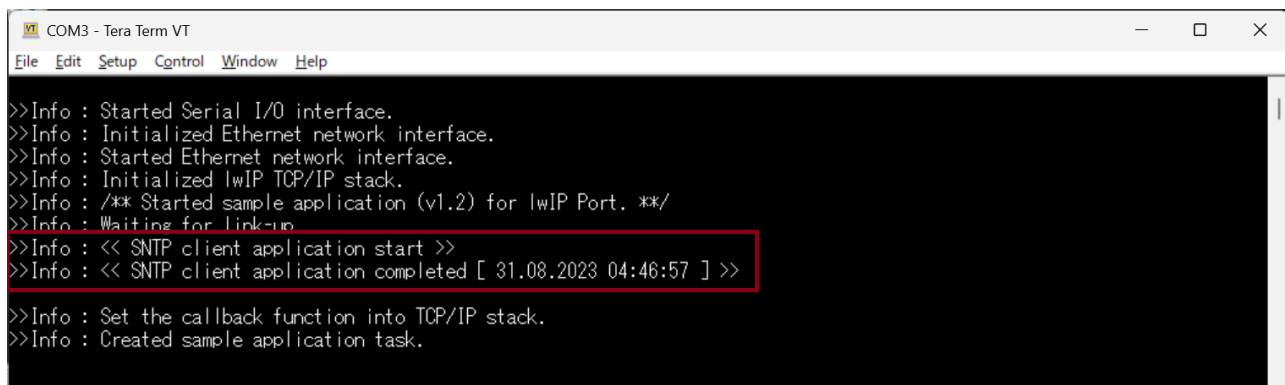
>>Info : Started Serial I/O interface.
>>Info : Initialized Ethernet network interface.
>>Info : Started Ethernet network interface.
>>Info : Initialized lwIP TCP/IP stack.
>>Info : /** Started sample application (v1.2) for lwIP Port. **/
>>Info : Waiting for link-up...
>>Info : << SNTP client application start >>
>>Info : << SNTP client application completed [ 31.08.2023 04:37:21 ] >>

>>Info : << IPERF2 server application start >>
>>Info : << IPERF2 server application completed >>
IPERF report: type=0, remote: 192.168.10.1:53252, total bytes: 14549016, duration in ms: 10291, kbits/s: 11304

>>Info : << IPERF2 client application start >>
>>Info : << IPERF2 client application completed >>
IPERF report: type=1, remote: 192.168.10.1:5001, total bytes: 19586544, duration in ms: 10000, kbits/s: 15664

>>Info : Set the callback function into TCP/IP stack.
>>Info : Created sample application task.
```

**Figure 7.9**  
Execution example with SNTP client, lwiperf (iperf2) server and lwiperf (iperf2) client enabled



```
COM3 - Tera Term VT
File Edit Setup Control Window Help

>>Info : Started Serial I/O interface.
>>Info : Initialized Ethernet network interface.
>>Info : Started Ethernet network interface.
>>Info : Initialized lwIP TCP/IP stack.
>>Info : /** Started sample application (v1.2) for lwIP Port. **/
>>Info : Waiting for link-up...
>>Info : << SNTP client application start >>
>>Info : << SNTP client application completed [ 31.08.2023 04:46:57 ] >>

>>Info : Set the callback function into TCP/IP stack.
>>Info : Created sample application task.
```

**Figure 7.10** Execution example with lwiperf (iperf2) client enabled

## 8. Software Specifications

### 8.1 Software Structure

An overview of the software is shown below.

Legend in the figure: ( ) Destination of the corresponding source file

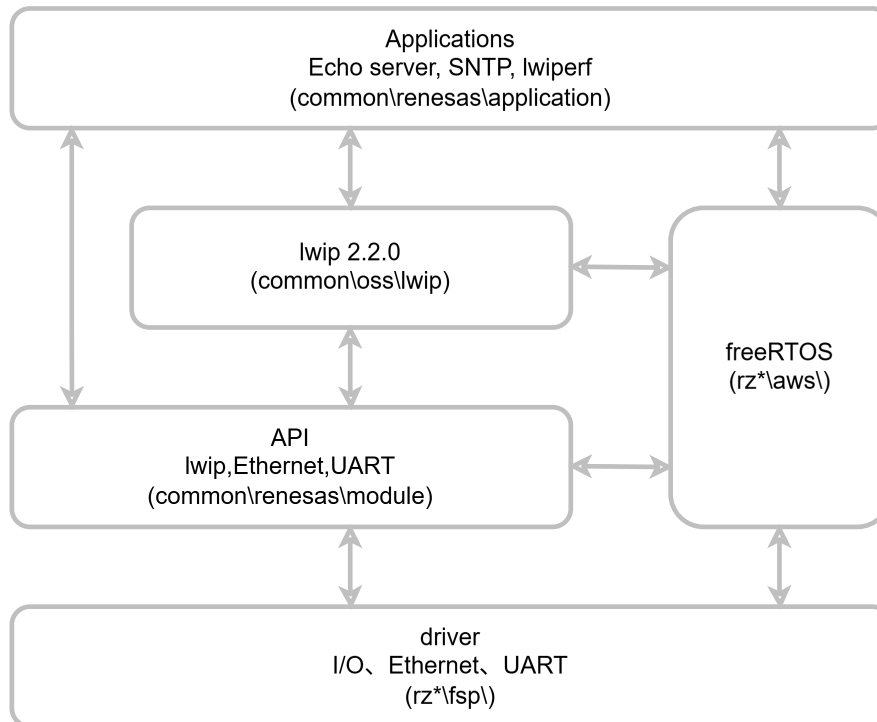


Figure 8.1. Software Architecture Diagram

### 8.1.1 Components Overview

Functionality overview in the overview diagram

Table 8.1. Overview of Each Component

Group	Folder name	Overview
Applications	common/renesas/application	Demonstration App
	SRC	Entry functions etc.
API	common/renesas/module/ether_netif	Ethernet API
	common/renesas/module/lwip_port	lwip API
	common/renesas/module/serial_io	
OS	Rz*/aws/	freeRTOS
driver	rz*/fsp/	Ethernet driver
	rz*/fsp/	UART driver

### 8.1.2 Drivers Used

The stacks used in FSP are shown below.

Table 8.2. Sample Software Support Status

Stack Name	function
g_ioport I/O Port Driver	Pin Settings, Control
g_ether0 Ethernet	Ethernet communication
g_uart0 UART	Serial communication
Heap 4	Allocating heap for FreeRTOS

## 8.2 Task

The task properties used in the sample program are shown below.

**Table 8.3. Task List**

Module	Task name	Priority	Stack size [bytes]	Synchronous mode	Overview
Main	Main thread	5	1024	Pause after initialization	Main task
Ethernet Port	task_code	3	1024	Wake up every 1 second	Ethernet link status monitoring
	task_code	4	2048	Received events	Waiting for received Ethernet data
lwIP Port	task_code	3	1024	Queue notification events	LINK status notification processing
lwIP	lwIP	4	1024	Received events	Reception processing within lwIP

**Table 8.4. RTOS Max Priority**

Item	Value
Max Priority	10

### 8.2.1 Resources

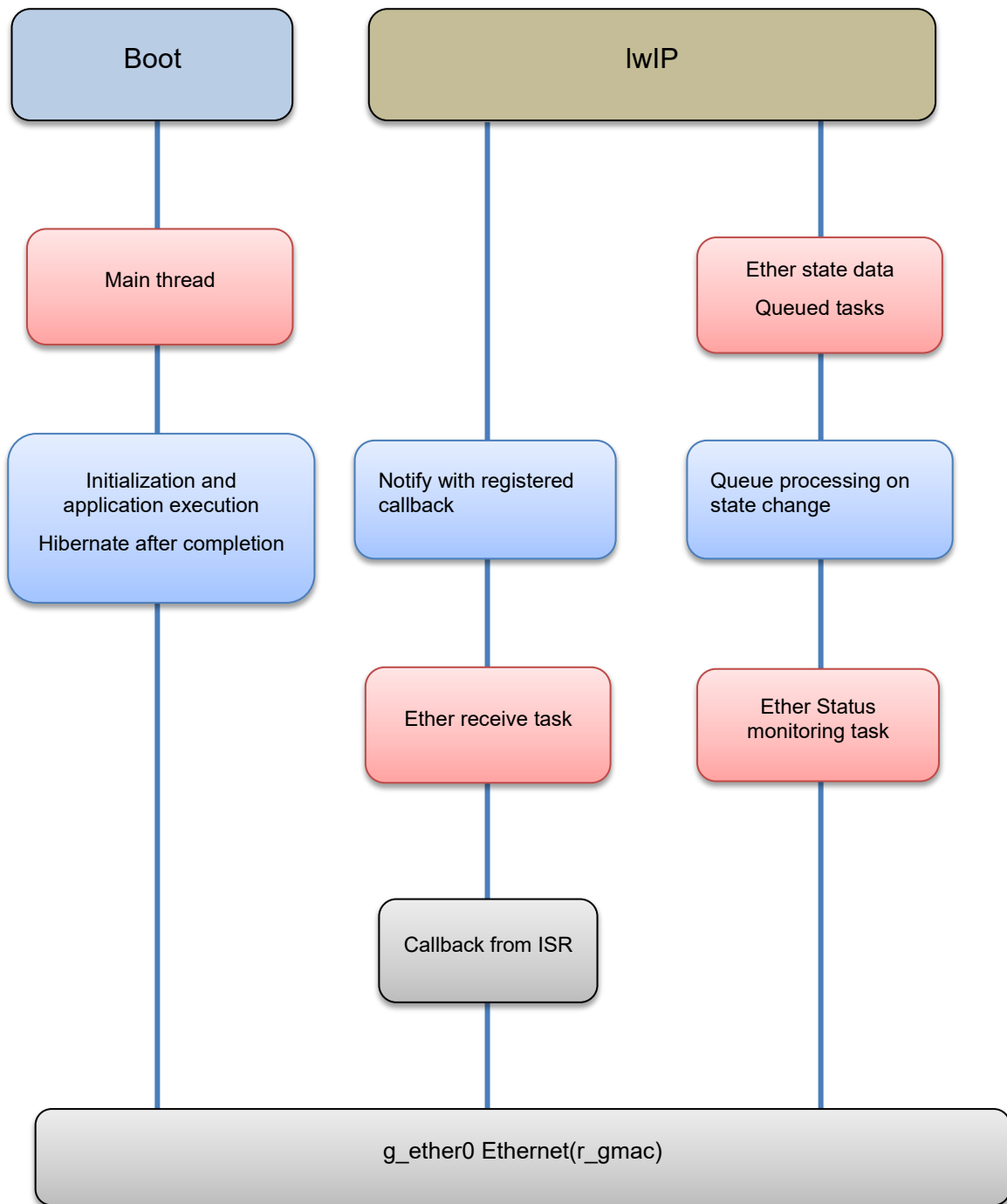
The resources used are shown below.

**Table 8.5. Used Resource**

Resources used	Format	Module	Use
Heap4	Heap memory	All modules	Ethernet transmission and reception data
queue	Queue	Ethernet port	Link status notification

### 8.2.2 Task Operations

The following diagram shows the relationships between tasks.



Legend :



Driver internal processing



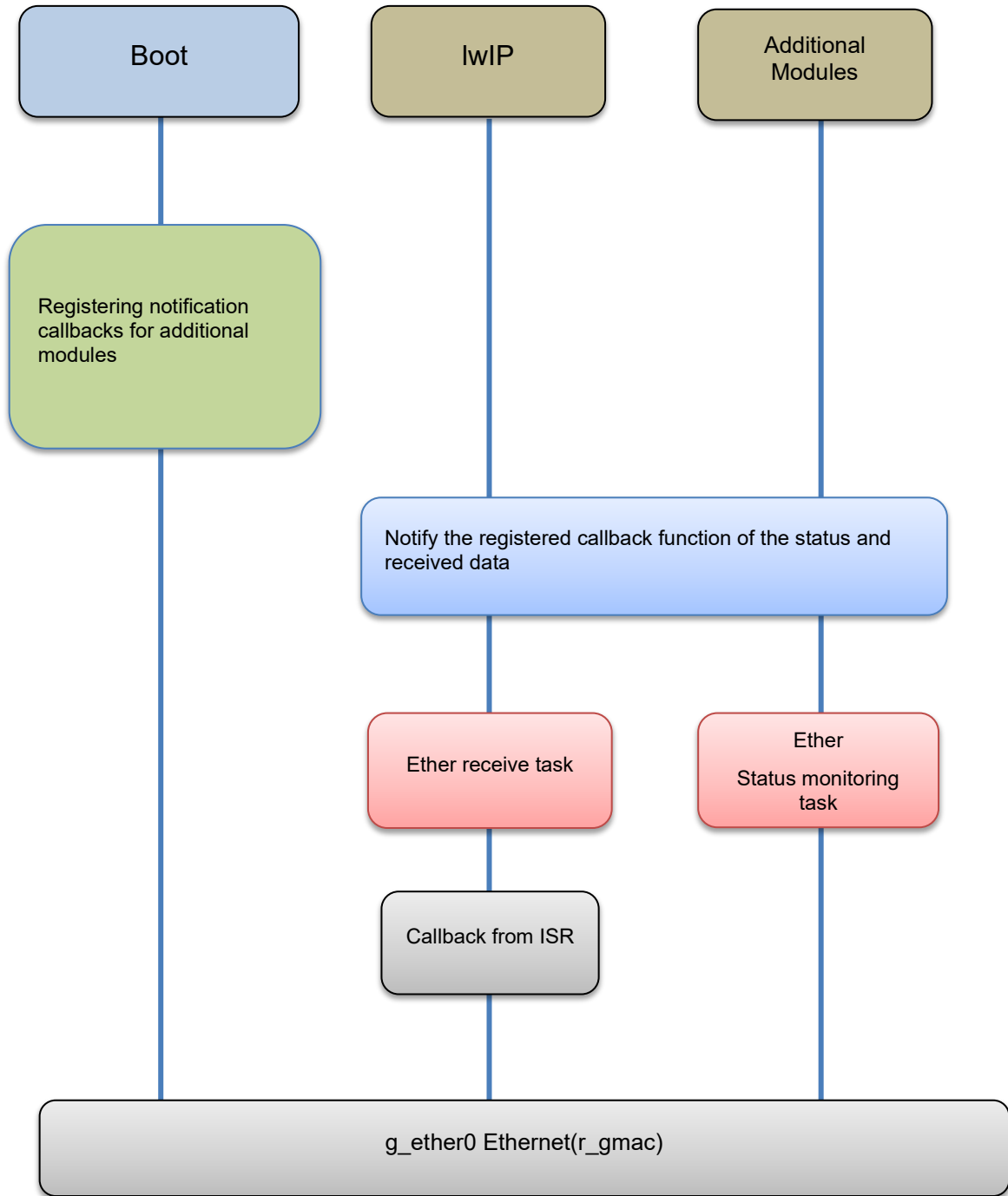
Task(thread)

Figure 8.2. sequence

### 8.2.3 Mirroring function for received packets

This shows the function to notify additional modules of data received from the driver, similar to lwIP.

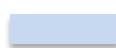
1. Overview



Legend :



Added processing



lwIP API

Figure 8.3. sequence

## 2. Interface

To enable the mirroring feature, register a callback function with the lwIP API to receive received data (MAC frames).

### ► Registering the Callback Function

Sample source code is commented out in the lwIP function "um\_lwip\_port\_ether\_netif.c / um\_lwip\_port\_ether\_netif\_open."

### ► Notification Types

- `ETHER_NETIF_CALLBACK_EVENT_LINK_UP`  
Ethernet status: Notified when a link is established. No notification data.
- `ETHER_NETIF_CALLBACK_EVENT_LINK_DOWN`  
Ethernet status: Notified when a link is disconnected. No notification data.
- `ETHER_NETIF_CALLBACK_EVENT_RECEIVE_ETHER_FRAME`  
An Ethernet receive event notifies the received MAC frame (RAW) as a parameter.  
Note that the parameter buffer "p\_frame\_packet" must be freed after use.

### 8.2.4 Ethernet Sending API

If you want to send MAC frames (RAW) directly without going through lwIP, a sample code is commented out in the source code below.

The sample code is commented out in the lwIP function "um\_lwip\_port\_mainf.c/115".

## 8.3 Memory Allocation

The following three groups are placed in a project.

- Loader Program  
Startup processing and initialization of the clock, interrupts, IO, etc. are placed in the BTCM.
- User Program  
Processing other than the loader program, such as the RTOS, interrupt vectors, drivers, and applications, is placed in the ATCM or system RAM.
- Shared Non-Cache  
Used as a storage location for Ethernet transmission and reception data, etc., this is placed in the mirror area of system RAM.

### 8.3.1 e<sup>2</sup> studio project

These definition files are located in the "script" folder.

- xxxxxxxx.ld  
This is a link script file; the file name changes depending on the boot mode, etc.
- xxxxxxxx.h  
This is a physical memory definition file, and defines the address of the implemented device, etc.

1. RZ/T2M

The memory usage by group is shown below.

➤ CPU0(CR52\_0)

The only application that runs on CPU0 is the LED blinking function.

**RZT2M Memory Map**

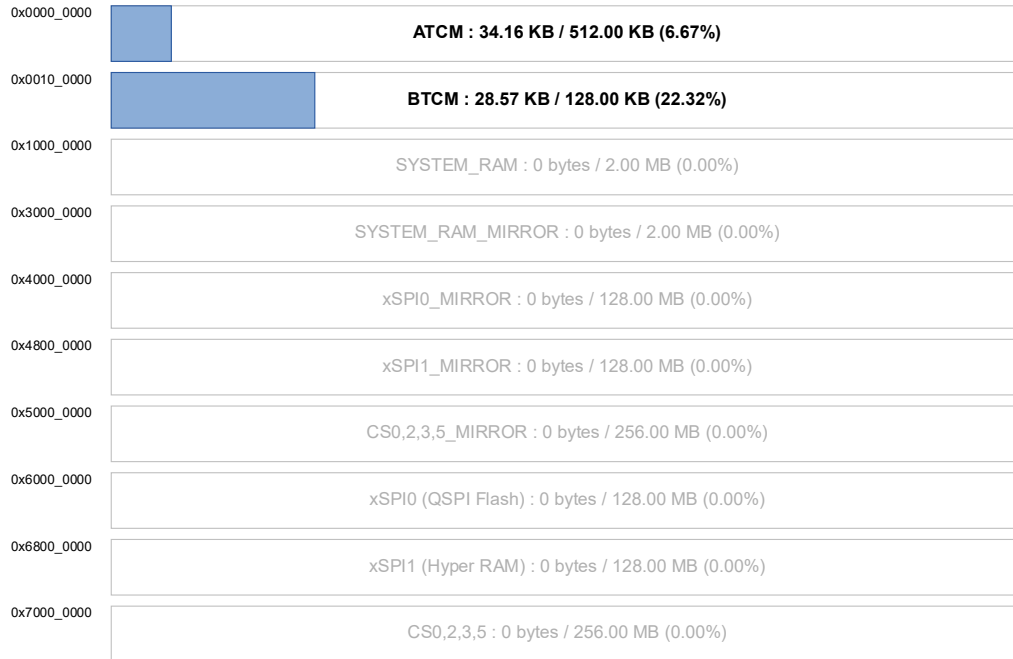


Figure 8.4. Memory footprint of the CPU0 project (e<sup>2</sup> studio)

**ATCM**

Total Usage : 34.16 KB

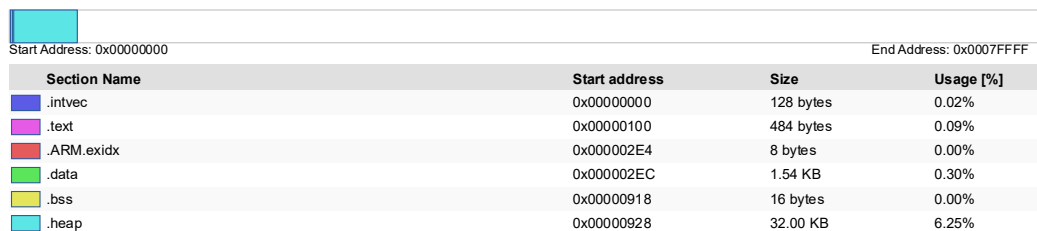


Figure 8.5. ATCM footprint of the CPU0 project (e<sup>2</sup> studio)

**BTCM**

Total Usage : 28.57 KB

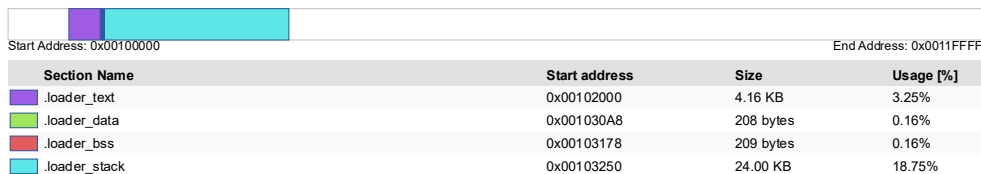
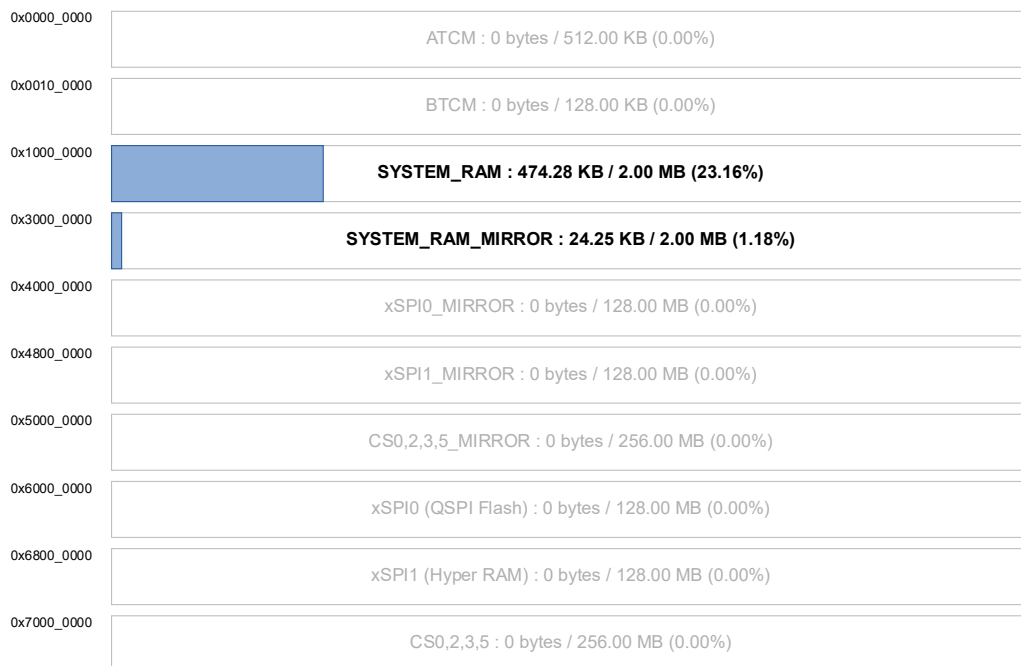


Figure 8.6. BTCM footprint of the CPU0 project (e<sup>2</sup> studio)

- CPU1(CR52\_1)  
The application running on CPU1 will perform functions other than blinking the LED.

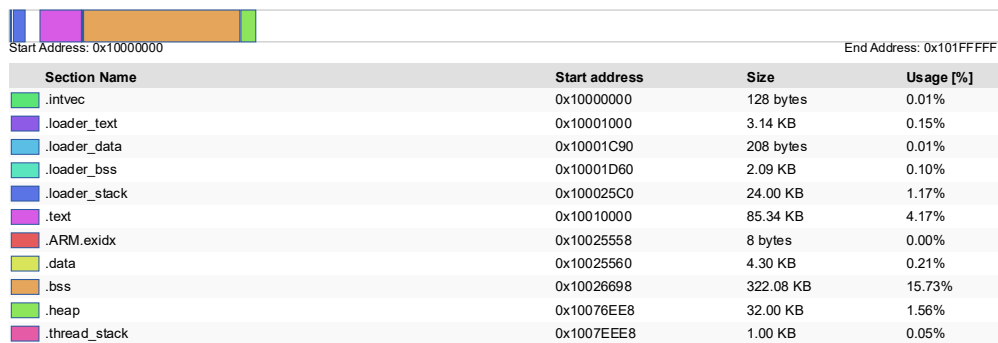
**RZT2M Memory Map**



**Figure 8.7. Memory footprint of the CPU1 project (e<sup>2</sup> studio)**

**SYSTEM\_RAM**

Total Usage : 474.28 KB



**Figure 8.8. SYSTEM\_RAM footprint of the CPU1 project (e<sup>2</sup> studio)**

**SYSTEM\_RAM\_MIRROR**

Total Usage : 24.25 KB



**Figure 8.9. SYSTEM\_RAM (MIRROR) footprint of the CPU1 project (e<sup>2</sup> studio)**

## 2. RZ/N2L

In the case of RZ/N2L, the size of the ATCM is small, so the user program is placed in SYSTEM\_RAM.

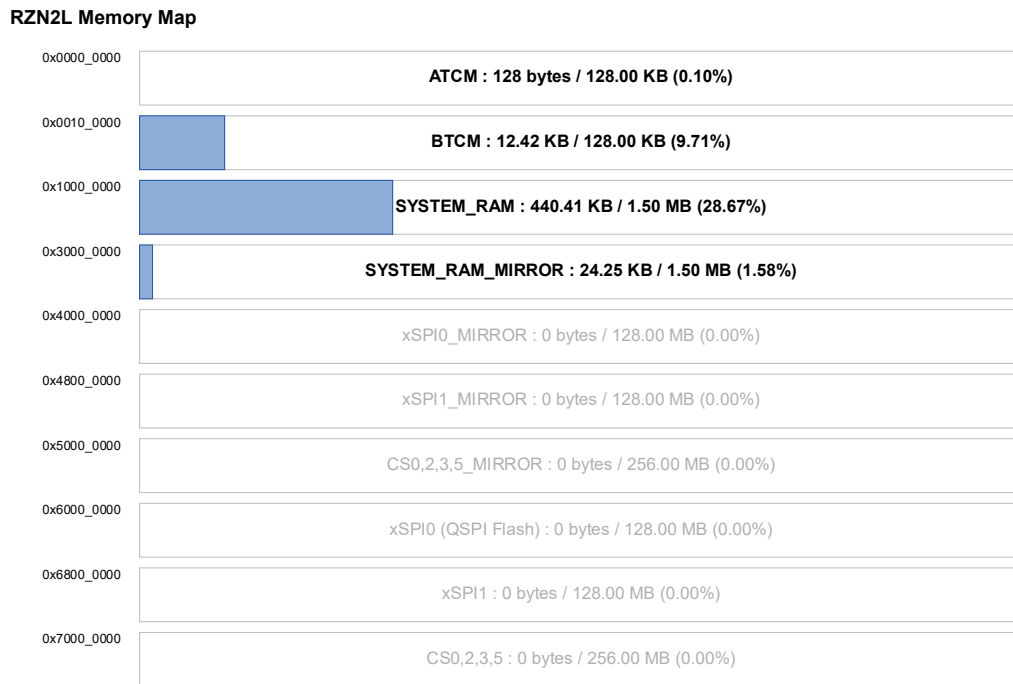


Figure 8.10. Memory footprint of the CPU0 project (e<sup>2</sup> studio)

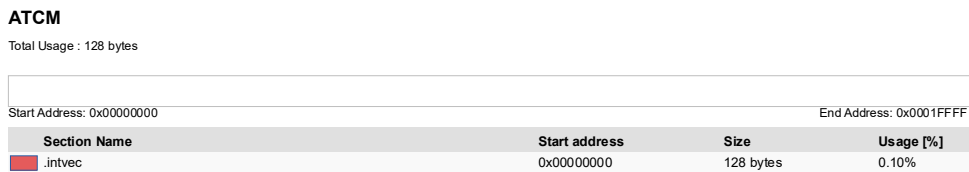


Figure 8.11. ATCM footprint of the CPU0 project (e<sup>2</sup> studio)

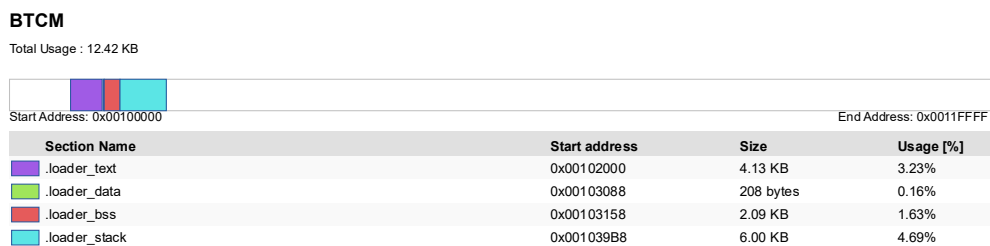
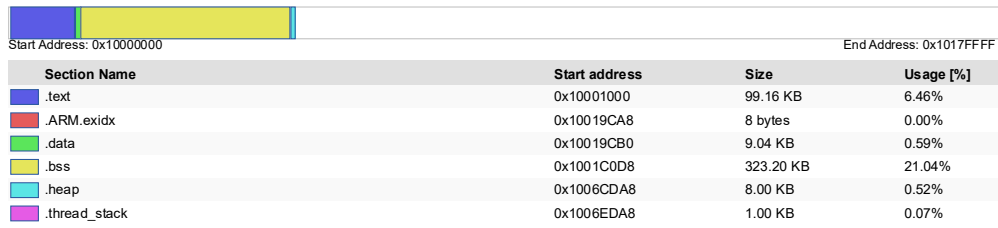


Figure 8.12. BTCM footprint of the CPU0 project (e<sup>2</sup> studio)

**SYSTEM\_RAM**

Total Usage : 440.41 KB



**Figure 8.13. SYSTEM\_RAM footprint of the CPU0 project (e<sup>2</sup> studio)**

**SYSTEM\_RAM\_MIRROR**

Total Usage : 24.25 KB



**Figure 8.14. SYSTEM\_RAM (MIRROR) footprint of the CPU0 project (e<sup>2</sup> studio)**

### 3. RZ/T2L

The memory usage by group is shown below.

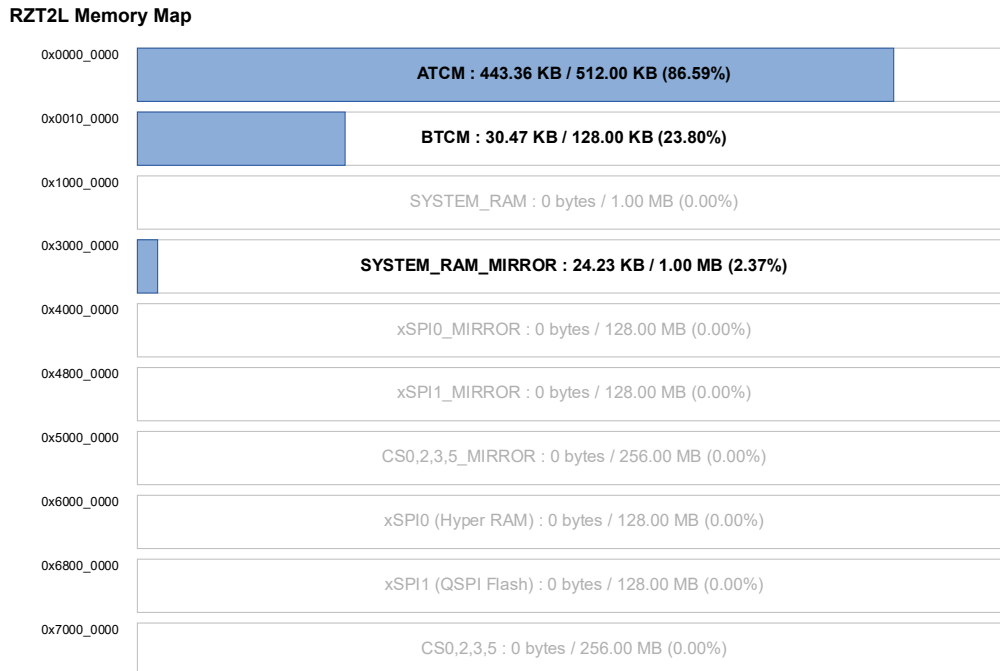


Figure 8.15. Memory footprint of the CPU0 project (e<sup>2</sup> studio)

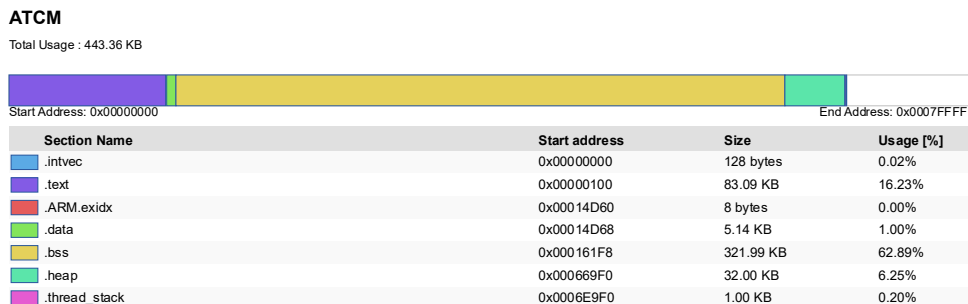


Figure 8.16. ATCM footprint of the CPU0 project (e<sup>2</sup> studio)

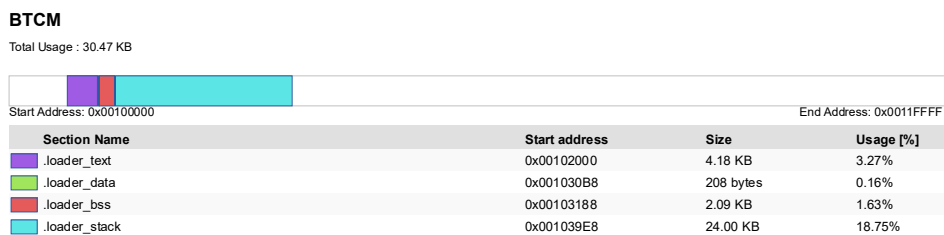
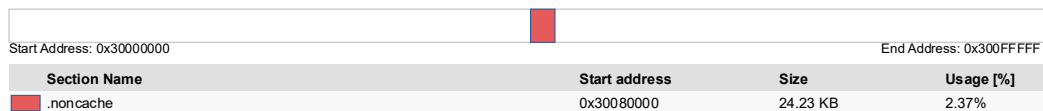


Figure 8.17. BTCM footprint of the CPU0 project (e<sup>2</sup> studio)

**SYSTEM\_RAM\_MIRROR**

Total Usage : 24.23 KB



**Figure 8.18. SYSTEM\_RAM (MIRROR) footprint of the CPU0 project (e<sup>2</sup> studio)**

### 8.3.2 EWARM project

It is placed in the definition file below in the "script" folder.

▶ xxxxxxxx.icf

This is a link script file. The file name changes depending on the boot mode, etc.

▶ xxxxxxxx.h

This is the physical memory definition file, and defines the addresses of the implemented devices.

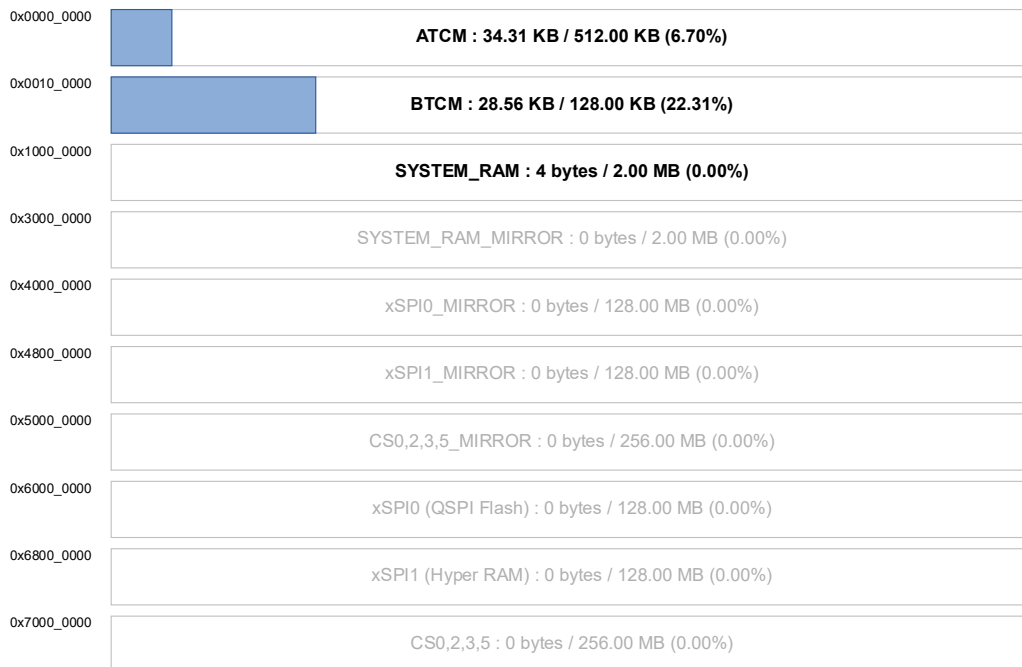
1. RZ/T2M

The memory usage by group is shown below.

➤ CPU0(CR52\_0)

The only application running on CPU0 is the LED blinking function.

**RZT2M Memory Map**



**Figure 8.19. Memory footprint of the CPU0 project (EWARM)**

**ATCM**

Total Usage : 34.31 KB



**Figure 8.20. ATCM footprint of the CPU0 project (EWARM)**

**SYSTEM\_RAM**

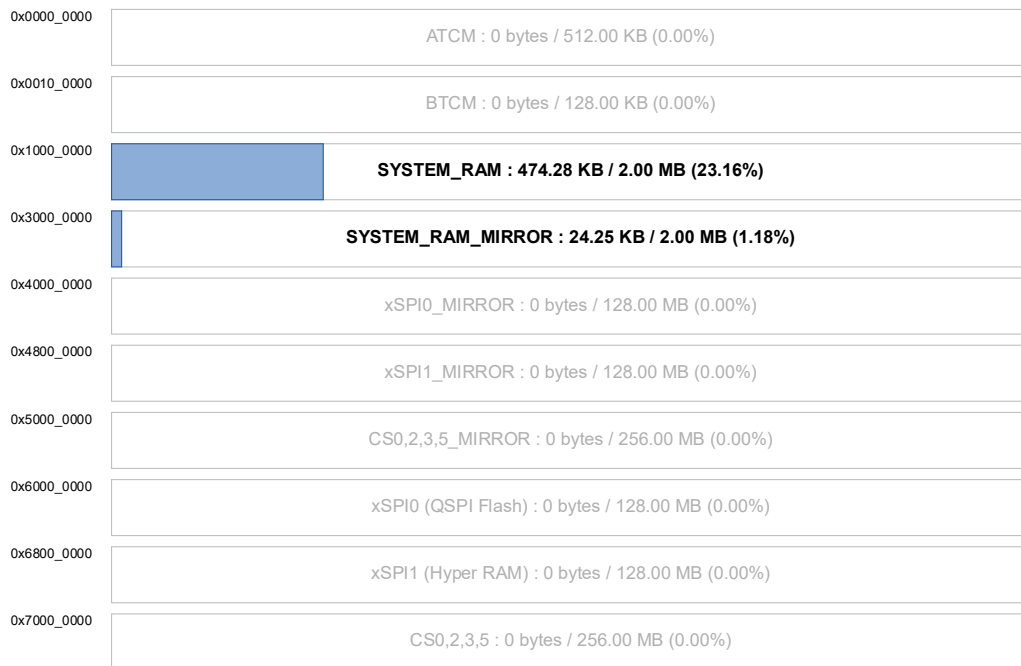
Total Usage : 4 bytes



**Figure 8.21. BTCM footprint of the CPU0 project (EWARM)**

- CPU1(CR52\_1)  
The application running on CPU1 will perform functions other than blinking the LED.

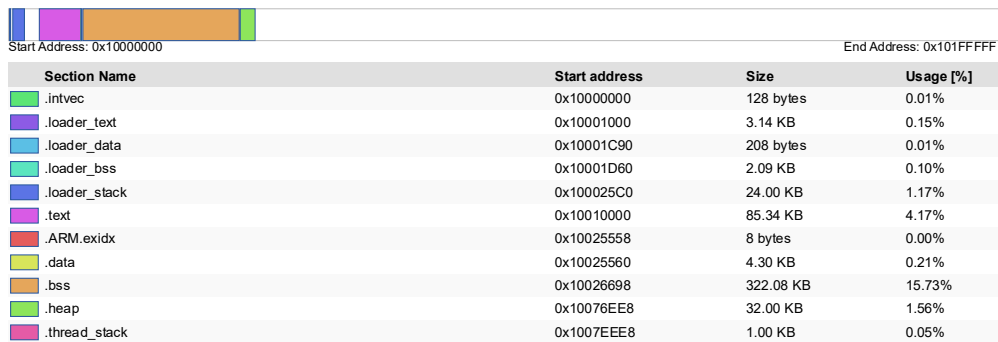
**RZT2M Memory Map**



**Figure 8.22. Memory footprint of the CPU1 project (EWARM)**

**SYSTEM\_RAM**

Total Usage : 474.28 KB



**Figure 8.23. SYSTEM\_RAM footprint of the CPU1 project (EWARM)**

**SYSTEM\_RAM\_MIRROR**

Total Usage : 24.25 KB



**Figure 8.24. SYSTM\_RAM (MIRROR) footprint of the CPU1 project (EWARM)**

2. RZ/N2L

In the case of RZ/N2L, the size of the ATCM is small, so the user program is placed in SYSTEM\_RAM.

RZ/N2L Memory Map

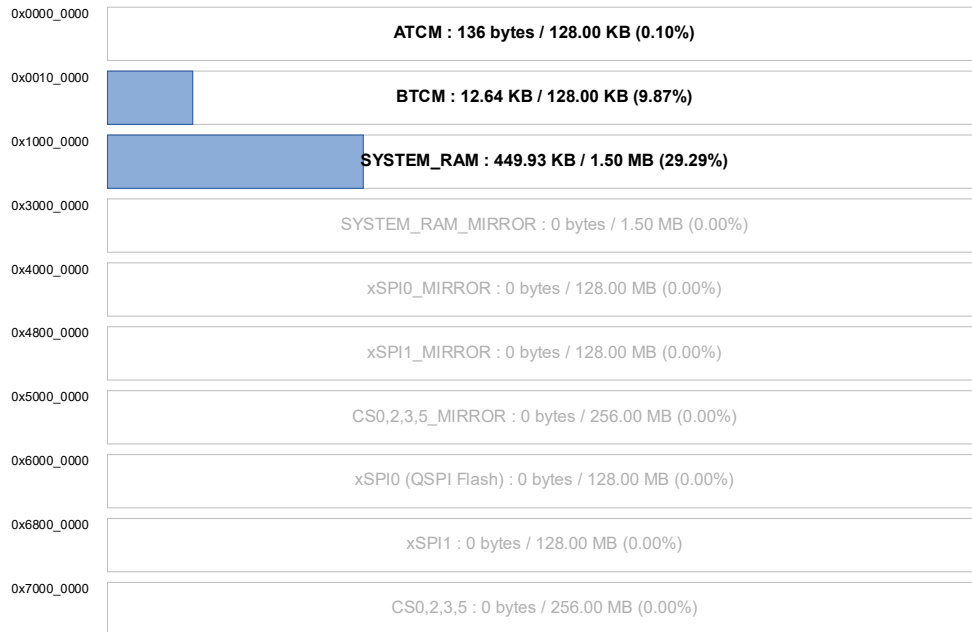


Figure 8.25. Memory footprint of the CPU0 project (EWARM)

ATCM

Total Usage : 136 bytes

Section Name	Start address	Size	Usage [%]
VECTOR_RBLOCK	0x00000000	136 bytes	0.10%

Figure 8.26. ATCM footprint of the CPU0 project (EWARM)

BTCM

Total Usage : 12.64 KB

Section Name	Start address	Size	Usage [%]
LDR_PRG_RBLOCK	0x00102000	4.36 KB	3.41%
LDR_DATA_WBLOCK	0x00103178	204 bytes	0.16%
LOADER_STACK	0x00103248	6.00 KB	4.69%
LDR_DATA_ZBLOCK	0x00104A48	2.08 KB	1.62%
SOFTWARE_RESET_ZBLOCK	0x00105295	1 bytes	0.00%

Figure 8.27. BTCM footprint of the CPU0 project (EWARM)

ATCM

Total Usage : 128 bytes

Section Name	Start address	Size	Usage [%]
.intvec	0x00000000	128 bytes	0.10%

Figure 8.28. SYSTEM\_RAM footprint of the CPU0 project (EWARM)

### 3. RZ/T2L

The memory usage by group is shown below.

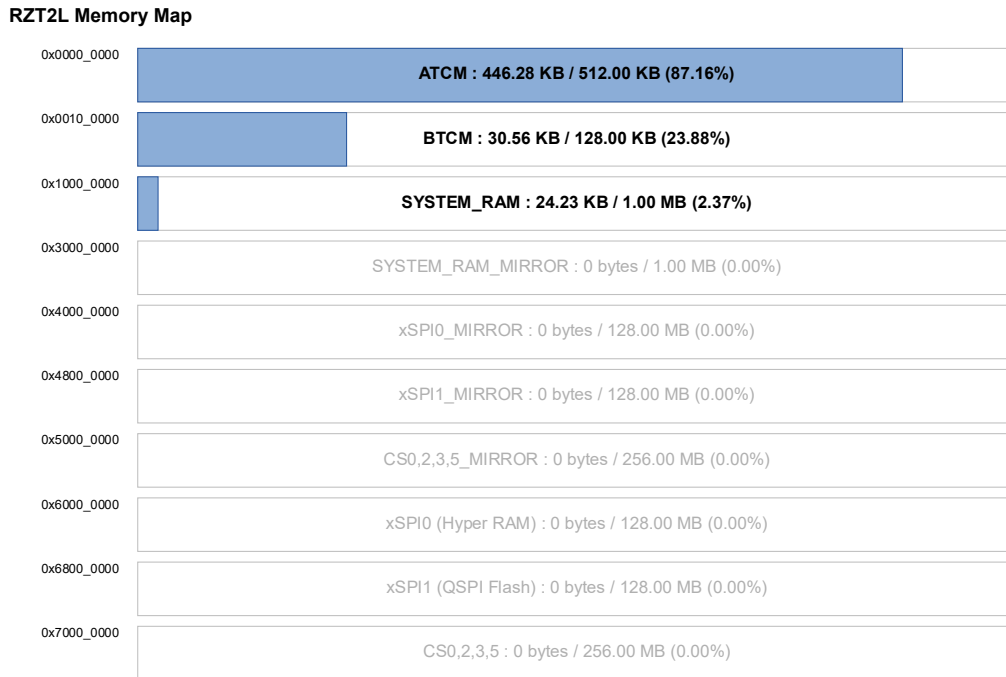


Figure 8.29. Memory footprint of the CPU0 project (EWARM)

#### ATCM

Total Usage : 446.28 KB

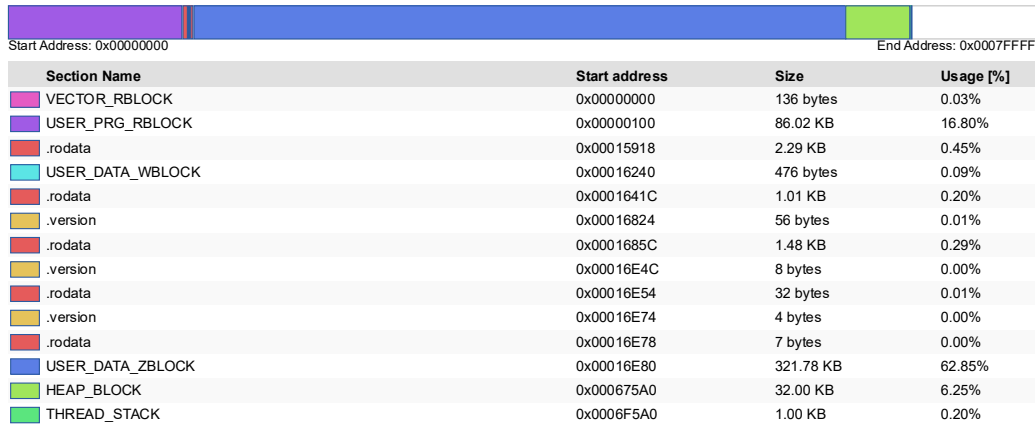
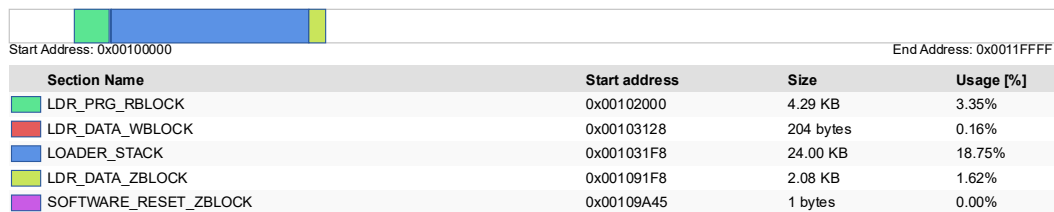


Figure 8.30. ATCM footprint of the CPU0 project (EWARM)

**BTCM**

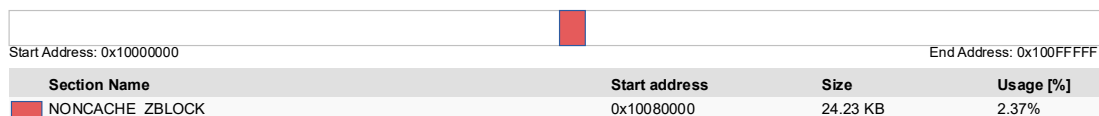
Total Usage : 30.56 KB



**Figure 8.31. BTCM footprint of the CPU0 project (EWARM)**

**SYSTEM\_RAM**

Total Usage : 24.23 KB



**Figure 8.32. SYSTEM\_RAM footprint of the CPU0 project (EWARM)**

## 8.4 CPU Utilization

This shows the CPU Utilization (task operation rate) of each device.

Please note that the measurements were performed using a simplified calculation method with the FreeRTOS API in e<sup>2</sup> studio, and therefore should be considered as reference values only.

1. **RZ/T2M**
  - iperf communication in progress  
CPU Utilization 72%
  - When the echo server is waiting for a client  
CPU Utilization 18%
  
2. **RZ/N2L**
  - iperf communication in progress  
CPU Utilization 74%
  - When the echo server is waiting for a client  
CPU Utilization 35%
  
3. **RZ/T2L**
  - iperf communication in progress  
CPU Utilization 65%
  - When the echo server is waiting for a client  
CPU Utilization 26%

## Revision History

Rev.	Date	Description	
		Page	Summary
1.50	Mar, 20 2026	-	Revised New Edition

# 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.

- Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.
- Ethernet is a registered trademark of Fujitsu Xerox Co. Ltd.
- IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc
- Additionally all product names and service names in this document are a trademark or a registered trademark which belongs to the respective owners. a trademark or a registered trademark which belongs to the respective owners.

## 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](http://www.renesas.com)

## Trademarks

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

## 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/](http://www.renesas.com/contact/).