



Customer training workshop: Device Configurator MCWDT

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.0 2023-7



Please read the [Important notice and warnings](#) at the end of this document

Scope of work

- This document helps application developers understand how to use the MCWDT configuration of the Device Configurator as part of creating a ModusToolbox™ application
 - The Device Configurator is part of a collection of tools included with the ModusToolbox™ software. It provides a GUI to configure the target device.

- ModusToolbox™ tools package version
 - 3.1.0
- Device Configurator version
 - 4.10
- Device
 - This code example uses the TRAVEO™ T2G CYT4BF8CDS device.
- Board
 - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing.

Introduction

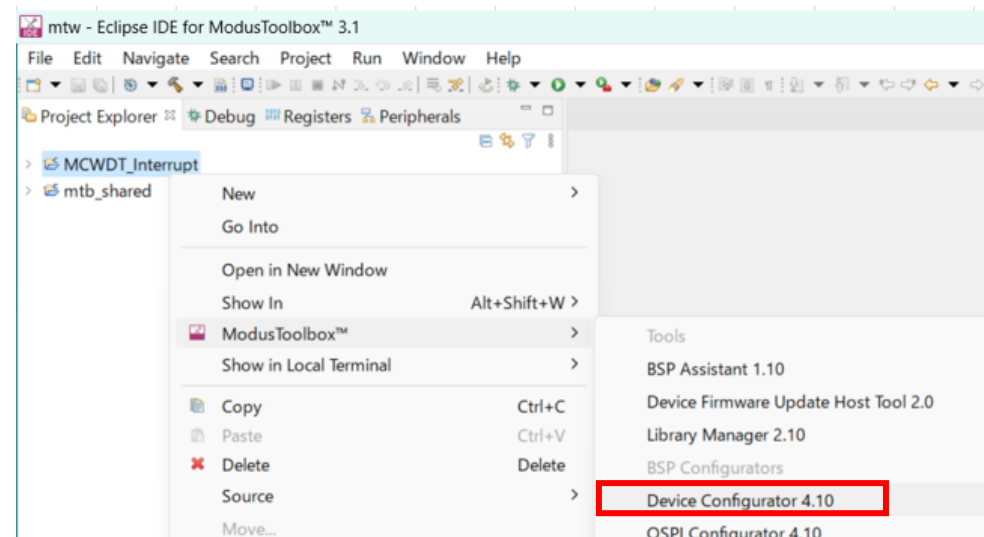
- **MCWDT has the following features:**
 - Up to four MCWDTs, each supporting the following:
 - LFCLK (ILO0, ILO1, WCO, LPECO, or ECO) as the input clock source
 - Fault and device reset generation, if not serviced, within a configurable interval
 - Periodic interrupt/wakeup generation in Active, Sleep, and DeepSleep power modes
 - Three independent counters: two 16-bit counters and one 32-bit counter
 - Warning threshold generates an interrupt to request servicing
 - Window mode
 - Running and freezing timers during DeepSleep mode
 - Debug

Launch the Device Configurator

– From Eclipse IDE

You can launch the Device Configurator by either of the following methods:

- a. Right-click on the project in the Project Explorer and select **ModusToolbox™ > Device Configurator <version>**



- b. Click the Device Configurator link in the Quick Panel



Device Configurator view for MCWDT config

– Peripherals tab

- Set each MCWDT on the Peripheral tab

Select used Multi-Counter Watchdog Timer (MCWDT0)

The screenshot shows the Infineon Device Configurator interface. The 'Peripherals' tab is selected, displaying a list of resources. The 'Multi-Counter Watchdog Timer (MCWDT) 0 (MCWDT_0)' is selected and highlighted with a red box. A red arrow points from a text box to this selection. To the right, the 'Multi-Counter Watchdog Timer (MCWDT) 0 (MCWDT_0) - Parameters' window is open, showing configuration options for Counter0 and Counter1. A red box highlights this window, and a red arrow points from a text box to it.

Name	Value
Overview	
Configuration Help	Open MCWDT Documentation
Counter0	
C0 Lower Limit	0
C0 Upper Limit	0xFFFF
C0 Warn Limit	32000
C0 Lower Action	None
C0 Upper Action	None
C0 Warn Action	Interrupt
C0 Auto Service Enable	<input checked="" type="checkbox"/>
C0 Deep Sleep Pause Enable	<input checked="" type="checkbox"/>
C0 Debug Run Enable	<input checked="" type="checkbox"/>
Counter1	
C1 Lower Limit	0
C1 Upper Limit	0xFFFF
C1 Warn Limit	32000
C1 Lower Action	None
C1 Upper Action	None
C1 Warn Action	Interrupt
C1 Auto Service Enable	<input checked="" type="checkbox"/>
C1 Deep Sleep Pause Enable	<input checked="" type="checkbox"/>
C1 Debug Run Enable	<input checked="" type="checkbox"/>

Configure selected MCWDT parameters

Quick start

– To use the Device Configurator for MCWDT setting

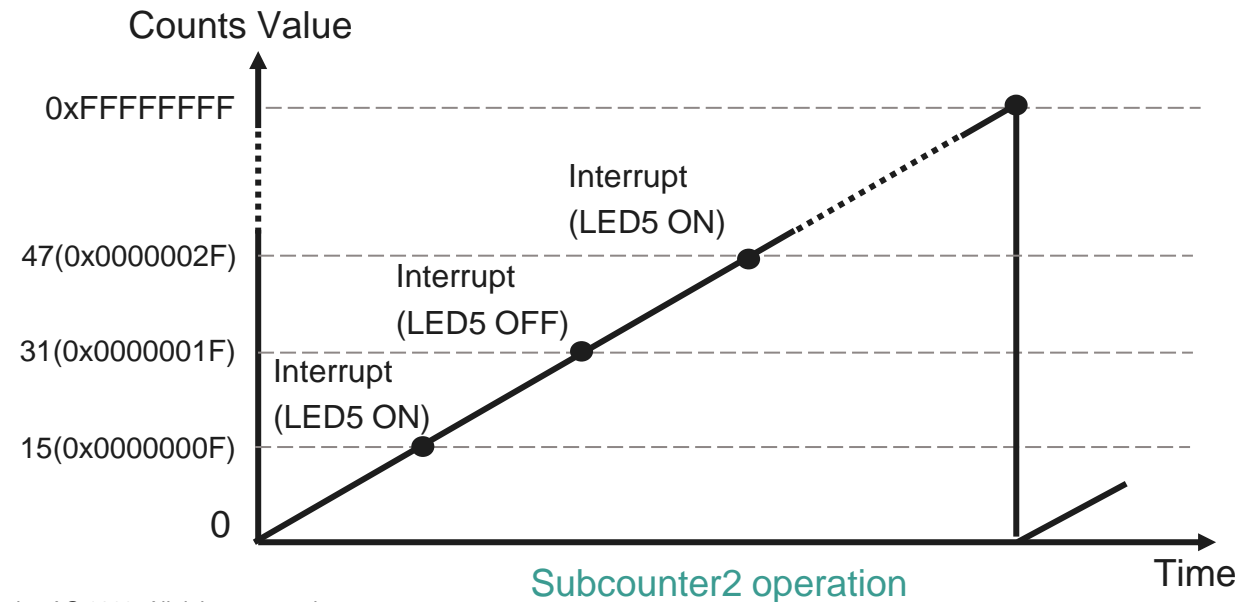
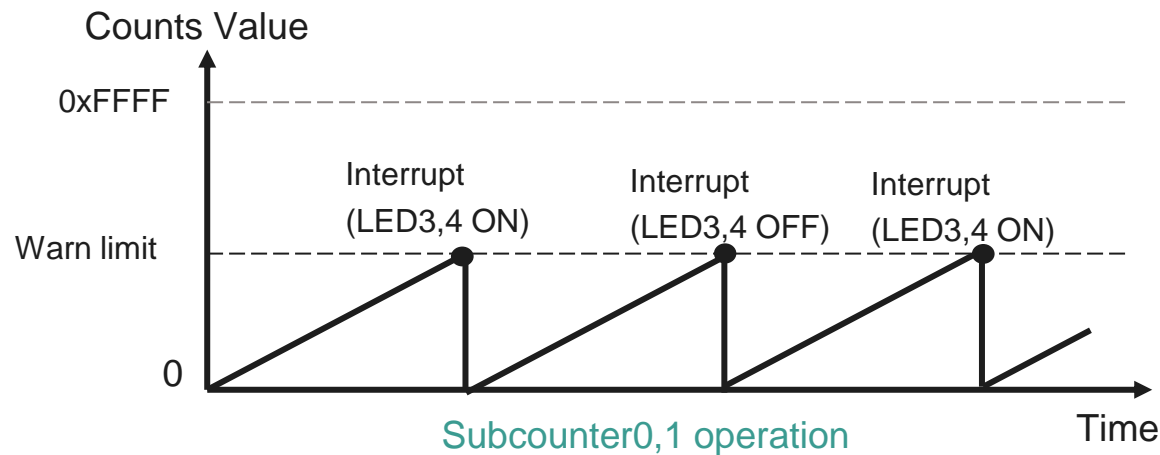
- Launch the Device Configurator.
- Use the various pull-down menus to configure signals. Refer to the descriptions in the Routing tab section for more details.
- Save the file to generate the source code.
- The Device Configurator generates code into a "GeneratedSource" directory in your Eclipse IDE application, or in the same location where you saved the *.modus file for non-IDE applications. That directory contains the necessary source (.c) and header (.h) files for the generated firmware, which uses the relevant driver APIs to configure the hardware.
- Use the generated structures as input parameters for MCWDT configuration functions in your application.

Use case

– Use case operation overview

- Use MCWDT Subcounter0, 1, and 2. Subcounter0 and 1 count up to the Warn limit (32000). Then an interrupt is generated and the count value restarts from zero. When Subcounter2 counts up to the setting value in the toggle bit, an interrupt is generated. The count value continuously counts up to a maximum value (0xFFFFFFFF). Then the count value restarts from zero.
- When each interrupt generates, the LED is repeatedly turned on and off. Combinations of Subcounter and LED are shown in the following diagrams.
 - Subcounter0: LED3, Subcounter1: LED4, Subcounter2: LED5

– Subcounter operation



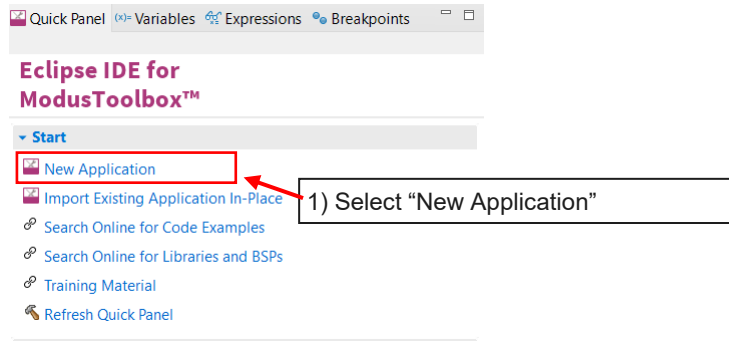
Use case (contd.)

- CLK_LF frequency set to 32.8 kHz
 - Source clock uses ILO
- MCWDT0 Subcounter0, Subcounter1, and Subcounter2 operations with interrupt.
 - Subcounter0
 - Upper limit: 0xFFFF, Lower limit: 0, Warn limit: 32000, Warn limit action: Interrupt
 - Enabling Auto service (Count up to Warn limit, then restart at zero), Pausing in DeepSleep mode, Enabling debugger connection run
 - Subcounter1
 - Upper limit: 0xFFFF, Lower limit: 0, Warn limit: 32000, Warn limit action: Interrupt
 - Enabling Auto service (Count up to Warn limit, then restart at zero), Pausing in DeepSleep mode, Enabling debugger connection run
 - Subcounter2
 - Toggle bit position value: 15, Toggle bit position action: Interrupt
 - Pausing in DeepSleep mode, Enabling debugger connection run
 - See the MCWDT_Interrupt application for operation

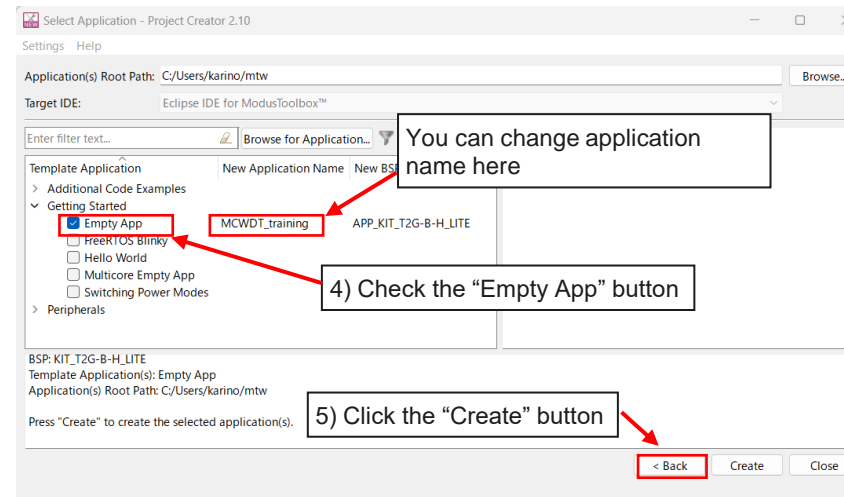
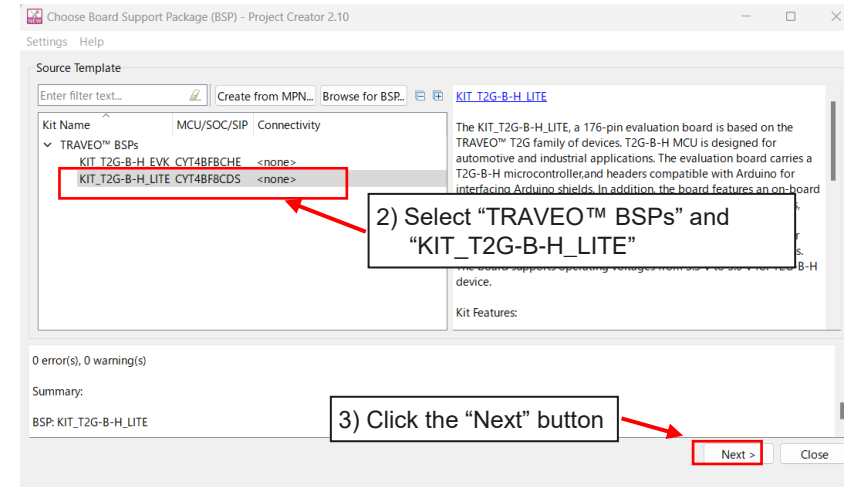
MCWDT configuration

– Create project

1. Click **New Application** in the Quick Panel and open the **Choose Board Support Package (BSP)** window



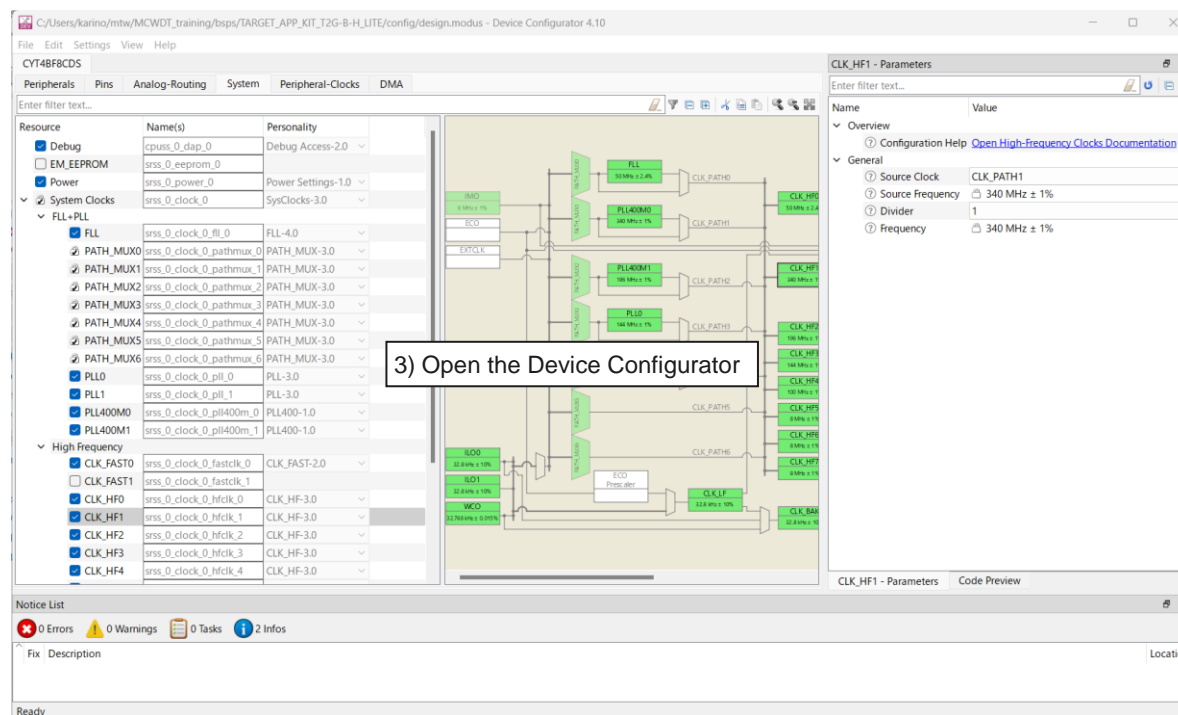
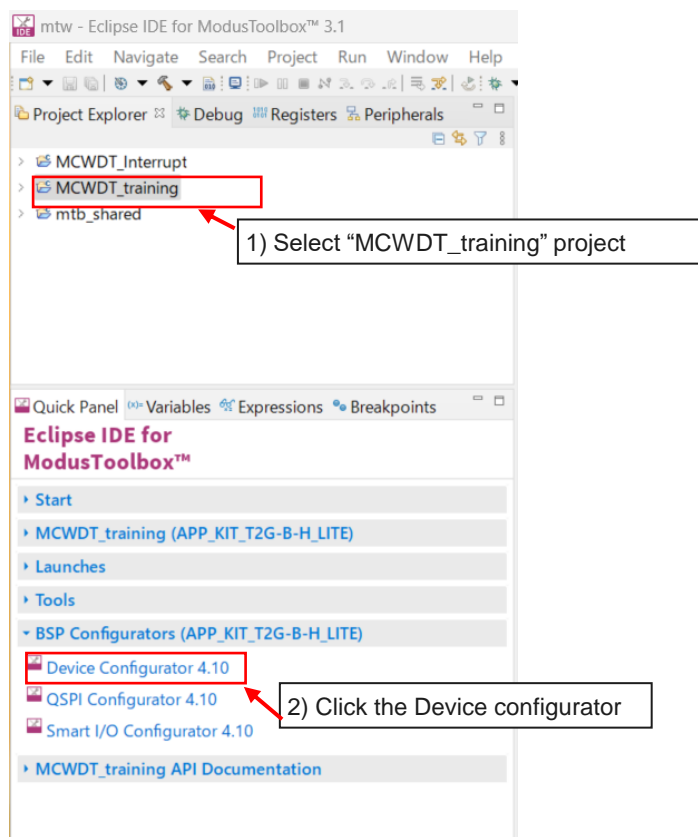
2. Select **TRAVEO™ BSPs** and **KIT_T2G-B-H_LITE**
3. Click the **Next** button and open the Application window
4. In this use case, it changes to **MCWDT_training**
5. Click the **Create** button, and then start application creation



MCWDT configuration (contd.)

– Launch the Device configurator

1. Select the **MCWDT_training** project.
2. Click the Device configurator in the Quick Panel
3. Then, open the Device Configurator window



MCWDT configuration (contd.)

- **Configure MCWDT Subcounter0 and Subcounter1**
 - Open the Peripherals tab and make the following settings
 - The values without description are default settings.

1) Select Multi-Counter Watchdog Timer(MCWDT)0

2) Fill the Name to "MCWDT_0"

Subcounter0

Subcounter1

Name	Value
Overview	
Configuration Help	Open MCWDT Documentation
Counter0	
C0 Lower Limit	0
C0 Upper Limit	0xFFFF
C0 Warn Limit	32000
C0 Lower Action	None
C0 Upper Action	None
C0 Warn Action	Interrupt
C0 Auto Service Enable	<input checked="" type="checkbox"/>
C0 Deep Sleep Pause Enable	<input checked="" type="checkbox"/>
C0 Debug Run Enable	<input checked="" type="checkbox"/>
Counter1	
C1 Lower Limit	0
C1 Upper Limit	0xFFFF
C1 Warn Limit	32000
C1 Lower Action	None
C1 Upper Action	None
C1 Warn Action	Interrupt
C1 Auto Service Enable	<input checked="" type="checkbox"/>
C1 Deep Sleep Pause Enable	<input checked="" type="checkbox"/>
C1 Debug Run Enable	<input checked="" type="checkbox"/>
Counter2	
C2 Toggle Bit	15
C2 Action	Interrupt
C2 Deep Sleep Pause Enable	<input checked="" type="checkbox"/>
C2 Debug Run Enable	<input checked="" type="checkbox"/>

Set C0 Upper Limit to 0xFFFF

Set C0 Warn limit to 32000

Select C0 Warn Action to Interrupt

Check all items(Auto service, DeepSleep pause, Debug run)

Set C1 Upper Limit to 0xFFFF

Set C1 Warn limit to 32000

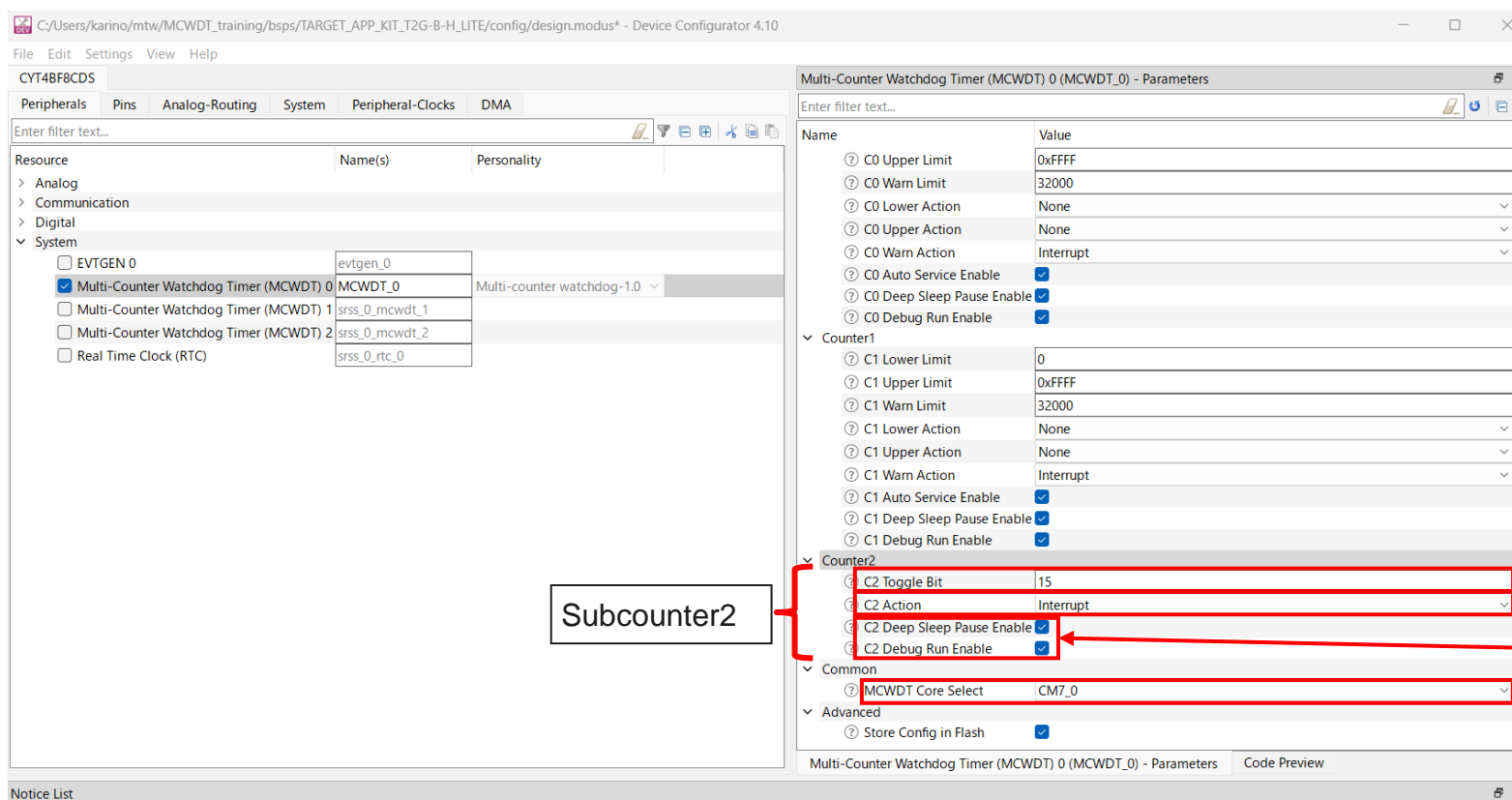
Select C1 Warn Action to Interrupt

Check all items(Auto service, DeepSleep pause, Debug run)

MCWDT configuration (contd.)

– Configure MCWDT Subcounter2

– The values without description are default settings.



Subcounter2

Set C2 Toggle Bit to 15

Set C2 Action to Interrupt

Check all items (DeepSleep pause, Debug run)

Select MCWDT Core Select to CM7_0

MCWDT configuration (contd.)

– System tab

- Set Low frequency clock (CLK_LF) on the System tab.

The screenshot shows the Infineon Device Configurator 4.10 interface. The 'System' tab is selected, and the 'System Clocks' section is expanded. The 'CLK_LF' resource is highlighted with a red box. A red arrow points from a text box labeled 'Select CLK_LF' to this resource. Another red box highlights the 'CLK_LF - Parameters' dialog box, which is open and shows the 'General' tab. A red arrow points from a text box labeled 'Configure parameters of Source Clock' to this dialog box. The dialog box shows the 'Source Clock' set to 'ILO0' and the 'Frequency' set to '32.8 kHz ± 10%'.

Resource	Name(s)	Personality
<input checked="" type="checkbox"/> Debug	cpuss_0_dap_0	Debug Access-2.0
<input type="checkbox"/> EM_EEPROM	srss_0_eeeprom_0	
<input checked="" type="checkbox"/> Power	srss_0_power_0	Power Settings-1.0
<input checked="" type="checkbox"/> System Clocks	srss_0_clock_0	SysClocks-3.0
> FLL+PLL		
> High Frequency		
> Input		
<input type="checkbox"/> ECO	srss_0_clock_0_eco_0	
<input type="checkbox"/> EXTCLK	srss_0_clock_0_ext_0	
<input checked="" type="checkbox"/> ILO0	srss_0_clock_0_ilo_0	ILO-3.0
<input checked="" type="checkbox"/> ILO1	srss_0_clock_0_ilo_1	ILO-3.0
<input checked="" type="checkbox"/> IMO	srss_0_clock_0_imo_0	IMO-3.0
<input checked="" type="checkbox"/> WCO	srss_0_clock_0_wco_0	WCO-3.0
> Miscellaneous		
<input type="checkbox"/> CLK_ALT_SYS_TICK	srss_0_clock_0_altsystick_0	
<input checked="" type="checkbox"/> CLK_BAK	srss_0_clock_0_bakclk_0	CLK_BAK-3.0
<input checked="" type="checkbox"/> CLK_LF	srss_0_clock_0_lfclk_0	CLK_LF-3.0
<input checked="" type="checkbox"/> CLK_TIMER	srss_0_clock_0_timerclk_0	CLK_TIMER-3.0
<input type="checkbox"/> ECO prescaler	srss_0_clock_0_ecoprescaler_0	

MCWDT configuration (contd.)

– Confirm configuration result

- Check the configuration result in the “Code Preview” tab of the Device Configurator

MCWDT

```

Code Preview
Enter search text...

/* NOTE: This is a preview only. It combines elements of the
 * cycfg_peripherals.c and cycfg_peripherals.h files located in the folder
 * C:/Users/Karino/mtw_EVK/MCWDT_Interrupt_2/beps/TARGET_APP_KIT_T2G-B-H_EVK/config/GeneratedSource.
 */

#include "cy_mcwdt.h"
#if defined (CY_USING_HAL)
    #include "cyhal_hwmgr.h"
#endif //defined (CY_USING_HAL)

#define MCWDT_0_HW MCWDT0
#define MCWDT_0_IRQ srss_interrupt_mcwdt_0_IRQn

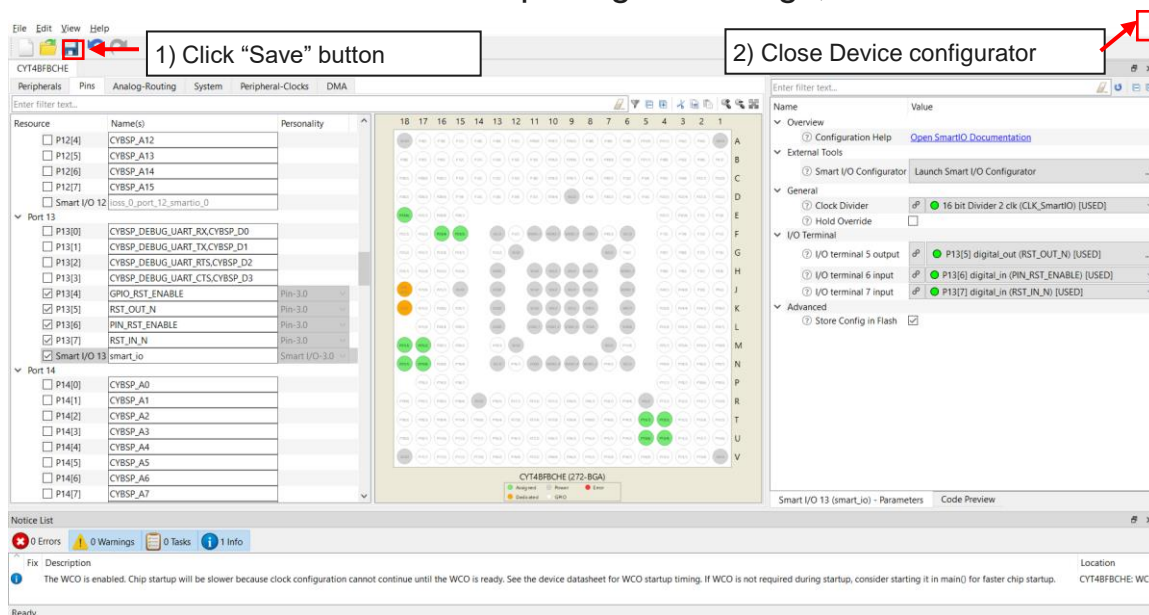
const cy_stc_mcwdt_config_t MCWDT_0_config =
{
    .c0LowerLimit = 0U,
    .c0UpperLimit = 0xFFFFU,
    .c0WarnLimit = 32000U,
    .c0LowerAction = CY_MCWDT_ACTION_NONE,
    .c0UpperAction = CY_MCWDT_ACTION_NONE,
    .c0WarnAction = CY_MCWDT_WARN_ACTION_INT,
    .c0AutoService = CY_MCWDT_ENABLE,
    .c0SleepDeepPause = CY_MCWDT_ENABLE,
    .c0DebugRun = CY_MCWDT_ENABLE,
    .c1LowerLimit = 0U,
    .c1UpperLimit = 0xFFFFU,
    .c1WarnLimit = 32000U,
    .c1LowerAction = CY_MCWDT_ACTION_NONE,
    .c1UpperAction = CY_MCWDT_ACTION_NONE,
    .c1WarnAction = CY_MCWDT_WARN_ACTION_INT,
    .c1AutoService = CY_MCWDT_ENABLE,
    .c1SleepDeepPause = CY_MCWDT_ENABLE,
    .c1DebugRun = CY_MCWDT_ENABLE,
    .c2ToggleBit = 15U,
    .c2Action = CY_MCWDT_CNT2_ACTION_INT,
    .c2SleepDeepPause = CY_MCWDT_ENABLE,
    .c2DebugRun = CY_MCWDT_ENABLE,
    .coreSelect = CY_MCWDT_PAUSED_BY_DPSLP_CM7_0,
};

```

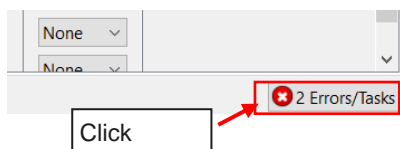
MCWDT configuration (contd.)

– Close Device configurator

- Click the **Save** button after completing all settings, then close the Device configurator



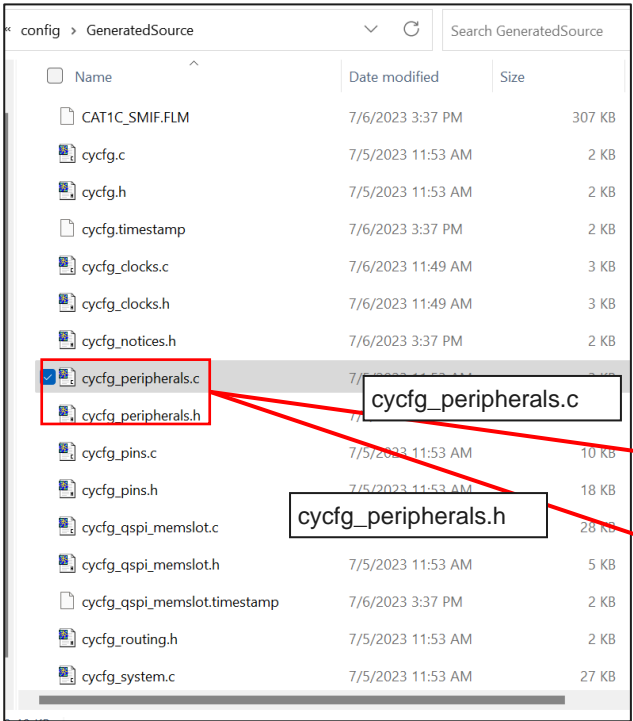
- If an **Errors/Tasks** message appears, resolve that according to the instructions as shown in the following screenshot



MCWDT configuration (contd.)

– Configuration file

- The Device Configurator generates code into a "GeneratedSource" directory in your Eclipse IDE application, or in the same location you saved the *.modus file for non-IDE applications.
- In this example, the following code is generated:



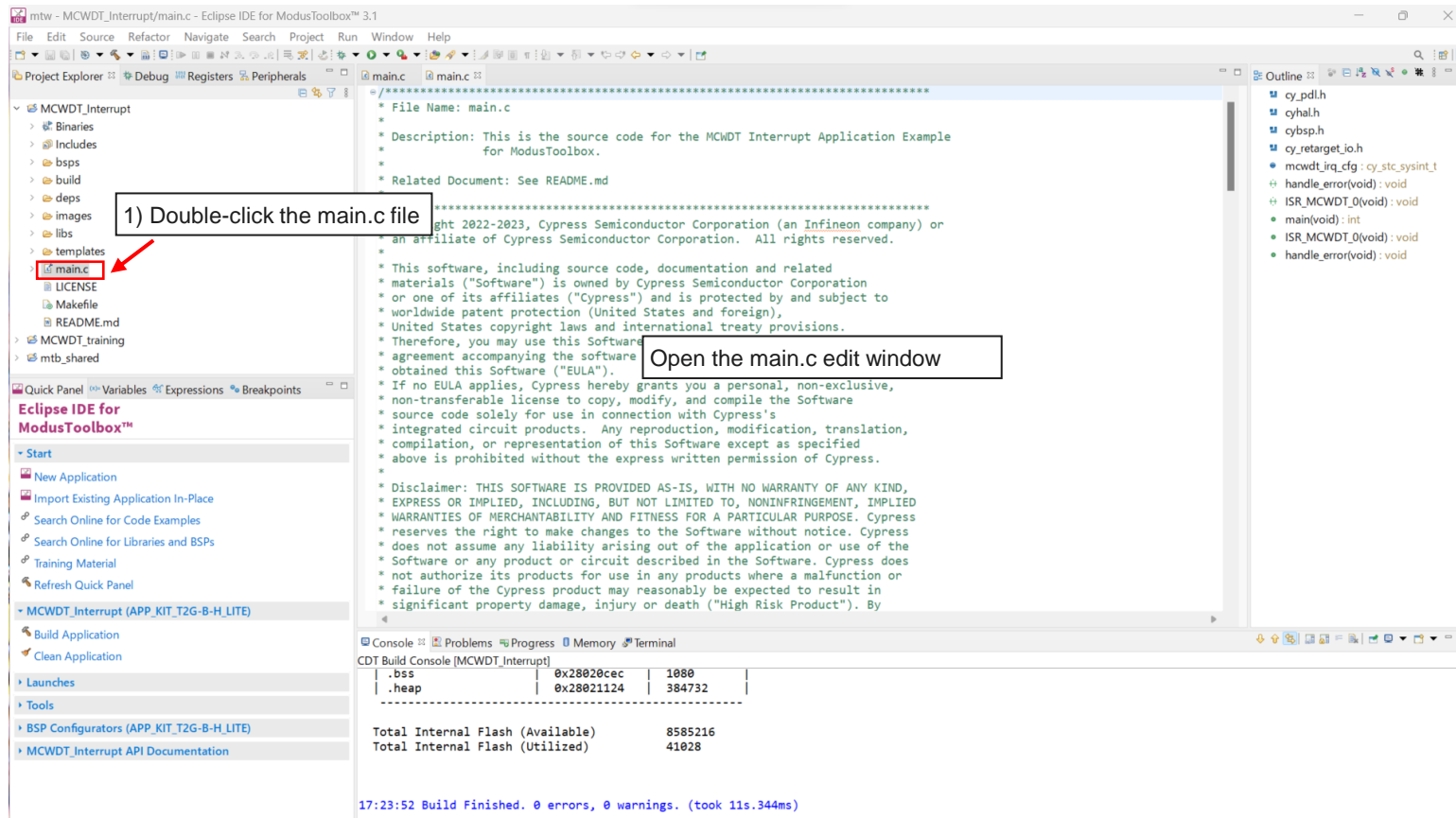
```

30 ↓
31 const cy_stc_mcwdt_config_t MCWDT_0_config = ↓
32 {↓
33     .c0LowerLimit = 0U,↓
34     .c0UpperLimit = 0xFFFFU,↓
35     .c0WarnLimit = 32000U,↓
36     .c0LowerAction = CY_MCWDT_ACTION_NONE,↓
37     .c0UpperAction = CY_MCWDT_ACTION_NONE,↓
38     .c0WarnAction = CY_MCWDT_WARN_ACTION_INT,↓
39     .c0AutoService = CY_MCWDT_ENABLE,↓
40     .c0SleepDeepPause = CY_MCWDT_ENABLE,↓
41     .c0DebugRun = CY_MCWDT_ENABLE,↓
42     .c1LowerLimit = 0U,↓
43     .c1UpperLimit = 0xFFFFU,↓
44     .c1WarnLimit = 32000U,↓
45     .c1LowerAction = CY_MCWDT_ACTION_NONE,↓
46     .c1UpperAction = CY_MCWDT_ACTION_NONE,↓
47     .c1WarnAction = CY_MCWDT_WARN_ACTION_INT,↓
48     .c1AutoService = CY_MCWDT_ENABLE,↓
49     .c1SleepDeepPause = CY_MCWDT_ENABLE,↓
50     .c1DebugRun = CY_MCWDT_ENABLE,↓
51     .c2ToggleBit = 15U,↓
52     .c2Action = CY_MCWDT_CNT2_ACTION_INT,↓
53     .c2SleepDeepPause = CY_MCWDT_ENABLE,↓
54     .c2DebugRun = CY_MCWDT_ENABLE,↓
55     .coreSelect = CY_MCWDT_PAUSED_BY_DPSLP_CM7_0,↓
56 };↓
57 #if defined(CY_USING_HAL)↓
58 const cyhal_resource_inst_t MCWDT_0_obj = ↓
59 {↓
60     .type = CYHAL_RSC_LPTIMER,↓
61     .block_num = 0U,↓
62     .channel_num = 0U,↓
63 };↓
64 #endif //defined(CY_USING_HAL)
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 ↓
896 ↓
897 ↓
898 ↓
899 ↓
900 ↓
901 ↓
902 ↓
903 ↓
904 ↓
905 ↓
906 ↓
907 ↓
908 ↓
909 ↓
910 ↓
911 ↓
912 ↓
913 ↓
914 ↓
915 ↓
916 ↓
917 ↓
918 ↓
919 ↓
920 ↓
921 ↓
922 ↓
923 ↓
924 ↓
925 ↓
926 ↓
927 ↓
928 ↓
929 ↓
930 ↓
931 ↓
932 ↓
933 ↓
934 ↓
935 ↓
936 ↓
937 ↓
938 ↓
939 ↓
940 ↓
941 ↓
942 ↓
943 ↓
944 ↓
945 ↓
946 ↓
947 ↓
948 ↓
949 ↓
950 ↓
951 ↓
952 ↓
953 ↓
954 ↓
955 ↓
956 ↓
957 ↓
958 ↓
959 ↓
960 ↓
961 ↓
962 ↓
963 ↓
964 ↓
965 ↓
966 ↓
967 ↓
968 ↓
969 ↓
970 ↓
971 ↓
972 ↓
973 ↓
974 ↓
975 ↓
976 ↓
977 ↓
978 ↓
979 ↓
980 ↓
981 ↓
982 ↓
983 ↓
984 ↓
985 ↓
986 ↓
987 ↓
988 ↓
989 ↓
990 ↓
991 ↓
992 ↓
993 ↓
994 ↓
995 ↓
996 ↓
997 ↓
998 ↓
999 ↓
1000 ↓

```


Implementation

- The structure generated by the Device Configurator can be used by implementing the following function in your application code.



Implementation (contd.)

- Add MCWDT initialization enable function

```

int main(void)
{
    cy_rslt_t result;
    cy_en_mcwdt_status_t mcwdt_init_status = CY_MCWDT_SUCCESS;

    /* Initialize the device and board peripherals */
    result = cybsp_init();

    /* BSP initialization failed. Stop program execution */
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }

    /* Enable global interrupts */
    _enable_irq();

    /* Initialize retarget-io to use the debug UART port */
    result = cy_retarget_io_init(CYBSP_DEBUG_UART_TX, CYBSP_DEBUG_UART_RX,
                                CY_RETARGET_IO_BAUDRATE);

    /* retarget-io initialization failed. Stop program execution */
    if (result != CY_RSLT_SUCCESS)
    {
        handle_error();
    }

    /* Initialize the MCWDT_0 */
    mcwdt_init_status = Cy_MCWDT_Init(MCWDT_0_HW, &MCWDT_0_config);

    if (mcwdt_init_status != CY_MCWDT_SUCCESS)
    {
        handle_error();
    }

    /* Sets up the interrupt handler */
    Cy_SysInt_Init(&mcwdt_irq_cfg, ISR_MCWDT_0);

    /* Enable the MCWDT interrupt in NVIC */
    NVIC_EnableIRQ((IRQn_Type) NvicMux3_IRQn);

    /* Enable the MCWDT_0 counters */
    Cy_MCWDT_Unlock(MCWDT_0_HW);
    Cy_MCWDT_SetInterruptMask(MCWDT_0_HW, CY_MCWDT_CTR_Msk);
    Cy_MCWDT_Enable(MCWDT_0_HW, CY_MCWDT_CTR_Msk,
                    0u);
    Cy_MCWDT_Lock(MCWDT_0_HW);

    /* Print a message on UART */
    /* \x1b[2J\x1b[H - ANSI ESC sequence for clear screen */
    printf("\x1b[2J\x1b[H");

    printf("*****
           *XMC7000 MCU: Multi-Counter Watchdog Timer Example *
           ***** \r\n\n");

    printf("\r\nMCWDT initialization is complete. USE LED blinking \r\n");

    for(;;)
    {
    }
}
    
```

```

const cy_stc_mcwdt_config_t MCWDT_0_config =
{
    .c0LowerLimit = 0U,
    .c0UpperLimit = 0xFFFFFU,
    .c0WarnLimit = 32000U,
    .c0LowerAction = CY_MCWDT_ACTION_NONE,
    .c0UpperAction = CY_MCWDT_ACTION_NONE,
    .c0WarnAction = CY_MCWDT_WARN_ACTION_INT,
    .c0AutoService = CY_MCWDT_ENABLE,
    .c0SleepDeepPause = CY_MCWDT_ENABLE,
    .c0DebugRun = CY_MCWDT_ENABLE,
    .c1LowerLimit = 0U,
    .c1UpperLimit = 0xFFFFFU,
    .c1WarnLimit = 32000U,
    .c1LowerAction = CY_MCWDT_ACTION_NONE,
    .c1UpperAction = CY_MCWDT_ACTION_NONE,
    .c1WarnAction = CY_MCWDT_WARN_ACTION_INT,
    .c1AutoService = CY_MCWDT_ENABLE,
    .c1SleepDeepPause = CY_MCWDT_ENABLE,
    .c1DebugRun = CY_MCWDT_ENABLE,
    .c2ToggleBit = 15U,
    .c2Action = CY_MCWDT_CNT2_ACTION_INT,
    .c2SleepDeepPause = CY_MCWDT_ENABLE,
    .c2DebugRun = CY_MCWDT_ENABLE,
    .coreSelect = CY_MCWDT_PAUSED_BY_DPSLP_CM7_0,
};

#if defined(CY_USING_HAL)
const cyhal_resource_inst_t MCWDT_0_obj =
{
    .type = CYHAL_RSC_LPTIMER,
}
    
```

Add MCWDT initialization function

Add MCWDT enable function

You can use the "MCWDT_0_HW" (MCWDT#0) to specify the hardware

```

#define MCWDT_0_ENABLED 1U
#define MCWDT_0_HW MCWDT0
#define MCWDT_0_IRQ srs_interruption_mcwdt_0_IRQn
    
```

Implementation (contd.)

Initialize the MCWDT_0

- Call the [Cy_MCWDT_Init\(\)](#) function to initialize the MCWDT_0.
 - The MCWDT configuration that is set in **MCWDT_0_config** can be changed using Device Configurator
 - Subcounter0/1 is set to generate interrupt when the counter matches to warning threshold value (=32000)
 - Subcounter2 is set to generate an interrupt when bit15 of the counter toggles

Set up the interrupt handler

- Call the [Cy_SysInt_Init\(\)](#) function to set up **ISR_MCWDT_0()** as the ISR.
 - The ISR reads the status of the interrupt by calling the [Cy_MCWDT_GetInterruptStatusMasked\(\)](#) function
 - Then it controls each user LED by calling [Cy_GPIO_Inv\(\)](#) depends on cause of interrupt
 - If the cause is MCWDT Subcounter0, user LED3 (P5.0) is toggled.
 - If the cause is MCWDT Subcounter1, user LED4 (P5.1) is toggled.
 - If the cause is MCWDT Subcounter2, user LED5 (P5.2) is toggled.
 - Clear interrupt by calling the [Cy_MCWDT_ClearInterrupt\(\)](#) function.

Implementation (contd.)

Enable the MCWDT_0 counters

- Unlocks the MCWDT_0 configuration registers by calling the [Cy_MCWDT_Unlock\(\)](#) function.
- Set MCWDT_0 interrupt mask register by calling the [Cy_MCWDT_SetInterruptMask\(\)](#) function.
- Enables MCWDT_0 counters by calling the [Cy_MCWDT_Enable\(\)](#) function.
- Locks out configuration changes to the MCWDT_0 registers by calling the [Cy_MCWDT_Lock\(\)](#) function.

References

Datasheet

- [CYT4BF datasheet 32-bit Arm® Cortex® -M7 microcontroller TRAVEO™ T2G family](#)

Architecture Technical reference manual

- [TRAVEO™ T2G automotive body controller high family architecture technical reference manual](#)

Registers Technical reference manual

- [TRAVEO™ T2G Automotive body controller high registers technical reference manual](#)

PDL/HAL

- [PDL](#)

- [HAL](#)

Training

- [TRAVEO™ T2G Training](#)

Revision History

Revision	ECN	Submission Date	Description of Change
**	7933288	2023/07/27	Initial release

Important notice and warnings

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-07

**Published by
Infineon Technologies AG
81726 Munich, Germany**

**© 2023 Infineon Technologies
AG.
All Rights Reserved.**

**Do you have a question about
this document?**

Go to:
www.infineon.com/support

**Document reference
002-38198 Rev. ****

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenhheitsgarantie”).

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.

