

RX 族

使用 MTU3/GPTW 的 PWM 输出方法

介绍

本篇应用说明描述了使用 MTU3d 和 GPTW 的 PWM 输出方法。

RX66T 系列微控制器（MCUs）配备了可生成脉宽调制（PWM）波形的多功能定时器脉冲单元 3（MTU3d）和通用 PWM 定时器（GPTW）。

本篇应用说明中的描述针对配备 MTU3 和 GPTW 的 RX 族器件。当使用本篇应用说明用于 RX66T 群以外的瑞萨 MCU 时，建议在进行符合备用的 MCU 的修改后进行仔细的评估。

适用设备

配备 MTU3 和 GPTW 的 RX 系列设备

目标设备

RX66T 群

多功能定时器脉冲单元 3 在本文档中被称为“MTU”。

目录

1	MTU 和 GPTW 的规格	12
1.1	运行模式和输出波形	14
1.2	缓冲功能	17
1.3	左右不对称互补 PWM 输出	23
1.4	死区时间自动设置功能	24
1.5	占空比 0%/100%输出功能	26
1.6	周期和周期寄存器之间的关系	27
2	运行确认条件	28
3	MCU 示例代码	29
3.1	共通点	29
3.1.1	示例代码列表	29
3.1.2	文件夹结构	31
3.1.3	文件结构	32
3.1.4	添加组件	33
3.1.5	引脚设置	34
3.1.6	中断设置	35
3.2	PWM 模式 1 比较匹配	36
3.2.1	概要	36
3.2.2	运行说明	37
3.2.3	Smart Configurator 设置	38
3.2.4	流程图	39
3.2.5	使用时的注意事项	40
3.2.5.1	缓冲寄存器修改运行和比较匹配之间的竞争	40
3.2.5.2	缓冲寄存器修改延迟	40
3.2.5.3	设置一个大于占空比寄存器的值	40
3.3	PWM 模式 1 计数清除	41
3.3.1	概要	41
3.3.2	运行说明	42
3.3.3	Smart Configurator 设置	43
3.3.4	流程图	44
3.3.5	使用时的注意事项	45

3.3.5.1	缓冲寄存器修改运行和计数器清除之间的竞争	45
3.3.5.2	缓冲寄存器修改延迟	45
3.4	PWM 模式 1 无缓冲寄存器	46
3.4.1	概要	46
3.4.2	运行说明	47
3.4.3	Smart Configurator 设置	48
3.4.4	流程图	49
3.4.5	使用时的注意事项	51
3.4.5.1	TGR 寄存器写入和比较匹配之间的竞争	51
3.4.5.2	设置一个大于占空比寄存器的值	51
3.5	PWM 模式 2 比较匹配	52
3.5.1	概要	52
3.5.2	运行说明	54
3.5.3	Smart Configurator 设置	55
3.5.4	流程图	57
3.5.5	使用时的注意事项	59
3.5.5.1	多通道计数开始	59
3.5.5.2	缓冲寄存器修改运行和比较匹配之间的竞争	59
3.5.5.3	缓冲寄存器修改延迟	59
3.5.5.4	设置一个大于占空比寄存器的值	59
3.6	PWM 模式 2 计数清除	60
3.6.1	概要	60
3.6.2	运行说明	62
3.6.3	Smart Configurator 设置	63
3.6.4	流程图	65
3.6.5	使用时的注意事项	67
3.6.5.1	多通道计数开始	67
3.6.5.2	缓冲寄存器修改运行和计数器清除之间的竞争	67
3.6.5.3	缓冲寄存器修改延迟	67
3.7	PWM 模式 2 无缓冲寄存器	68
3.7.1	概要	68
3.7.2	运行说明	69

3.7.3	Smart Configurator 设置	70
3.7.4	流程图	72
3.7.5	使用时的注意事项	75
3.7.5.1	多通道计数开始	75
3.7.5.2	TGR 寄存器写入和比较匹配之间的竞争	75
3.7.5.3	设置一个大于占空比寄存器的值	75
3.8	复位同步 PWM 模式	76
3.8.1	概要	76
3.8.2	运行说明	78
3.8.3	Smart Configurator 设置	79
3.8.4	流程图	81
3.8.5	使用时的注意事项	84
3.8.5.1	使用 4 个通道时计数开始	84
3.8.5.2	Smart Configurator 使用的组件	84
3.9	互补 PWM 模式 带双缓冲器	85
3.9.1	概要	85
3.9.2	运行说明	88
3.9.3	Smart Configurator 设置	90
3.9.4	流程图	92
3.9.5	使用时的注意事项	94
3.9.5.1	引脚设置	94
3.9.5.2	缓冲寄存器值更新	94
3.9.5.3	缓冲运行设置	94
3.9.5.4	输出电平设置	94
3.9.5.5	TGR 寄存器初始值	95
3.9.5.6	左右不对称 PWM 输出方法	95
3.10	互补 PWM 模式 无双缓冲器	96
3.10.1	概要	96
3.10.2	运行说明	99
3.10.3	Smart Configurator 设置	101
3.10.4	流程图	103
3.10.5	使用时的注意事项	105

3.10.5.1	引脚设置	105
3.10.5.2	缓冲寄存器值更新	105
3.10.5.3	缓冲运行设置	105
3.10.5.4	输出电平设置	105
3.11	PWM 模式 1 占空比 0%到 100%（在比较匹配时修改比较寄存器）	106
3.11.1	概要	106
3.11.2	运行说明	107
3.11.3	Smart Configurator 设置	109
3.11.4	流程图	110
3.11.5	相关运行	113
3.11.5.1	占空比 0%到 100%（无缓冲器）	113
3.11.6	使用时的注意事项	114
3.11.6.1	缓冲寄存器修改延迟	114
3.11.6.2	设置一个大于占空比寄存器的值	114
3.11.6.3	从 0%占空比切换	114
3.12	PWM 模式 1 占空比 0%到 100%（在计数器清除时修改比较寄存器）	115
3.12.1	概要	115
3.12.2	运行说明	116
3.12.3	Smart Configurator 设置	118
3.12.4	流程图	119
3.12.5	相关运行	121
3.12.5.1	占空比 0%到 100%（无缓冲器）	121
3.12.6	使用时的注意事项	123
3.12.6.1	缓冲寄存器修改和比较匹配之间的竞争	123
3.12.6.2	缓冲寄存器修改延迟	123
3.12.6.3	设置一个大于占空比寄存器的值	123
3.12.6.4	从 0%占空比切换	123
3.13	PWM 模式 1 占空比 0%和 100%	124
3.13.1	概要	124
3.13.2	运行说明	125
3.13.3	Smart Configurator 设置	126
3.13.4	流程图	127

3.13.5	相关运行	128
3.13.5.1	占空比 0%和 100%（无缓冲器）	128
3.13.5.2	在初始值低时保持 0%后更改占空比到 100%.....	129
3.13.5.3	在初始值低时保持 100%后更改占空比到 0%.....	131
3.13.5.4	在初始值高时保持 0%后更改占空比到 100%.....	133
3.13.5.5	在初始值高时保持 100%后更改占空比到 0%.....	135
3.13.6	使用时的注意事项	137
3.13.6.1	TIOR 寄存器修改延迟.....	137
3.13.6.2	TIOR 寄存器修改运行和比较匹配之间的竞争	137
3.13.6.3	修改 MTU3, MTU4, MTU6, MTU7 的 TIOR 寄存器的使用说明	137
3.13.6.4	缓冲寄存器修改运行和比较匹配之间的竞争	137
3.13.6.5	从 0%占空比切换.....	137
3.13.6.6	当占空比寄存器设置为 0 时一个周期的输出	137
3.13.6.7	当占空比寄存器设置为与周期寄存器相同值时波形不发生改变	138
3.14	互补 PWM 模式 占空比 0%到 100%.....	139
3.14.1	概要	139
3.14.2	运行说明	142
3.14.3	Smart Configurator 设置	144
3.14.4	流程图	146
3.14.5	使用时的注意事项	148
3.14.5.1	引脚设置	148
3.14.5.2	缓冲寄存器值更新	148
3.14.5.3	缓冲运行设置	148
3.14.5.4	输出电平设置	148
3.15	互补 PWM 模式 占空比 0%和 100%.....	149
3.15.1	概要	149
3.15.2	运行说明	152
3.15.3	Smart Configurator 设置	154
3.15.4	流程图	156
3.15.5	相关运行	157
3.15.5.1	在初始值低时保持 0%后更改占空比到 100%.....	157
3.15.5.2	在初始值低时保持 100%后更改占空比到 0%.....	158

3.15.5.3	在初始值高时保持 0%后更改占空比到 100%.....	159
3.15.5.4	在初始值高时保持 100%后更改占空比到 0%.....	160
3.15.6	使用时的注意事项	161
3.15.6.1	引脚设置	161
3.15.6.2	缓冲寄存器值更新	161
3.15.6.3	缓冲运行设置	161
3.15.6.4	输出电平设置	161
4	GPTW 示例代码.....	162
4.1	通用代码.....	162
4.1.1	示例代码列表	162
4.1.2	文件夹结构	165
4.1.3	文件结构	166
4.1.4	添加组件	167
4.1.5	引脚设置	168
4.1.6	中断设置	169
4.2	锯齿波 PWM 模式	170
4.2.1	概要	170
4.2.2	运行说明	172
4.2.3	Smart Configurator 设置.....	173
4.2.4	流程图.....	176
4.2.5	使用时的注意事项	178
4.2.5.1	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	178
4.3	锯齿波单触发脉冲	179
4.3.1	概要	179
4.3.2	运行说明	181
4.3.3	Smart Configurator 设置.....	183
4.3.4	流程图.....	186
4.3.5	相关运行	188
4.3.5.1	当不使用死区时间自动设置功能	188
4.3.6	使用时的注意事项	190
4.3.6.1	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	190
4.4	锯齿波 PWM 模式 三相互补	191

4.4.1	概要	191
4.4.2	运行说明	193
4.4.3	Smart Configurator 设置	194
4.4.4	流程图	196
4.4.5	使用时的注意事项	197
4.4.5.1	多通道计数开始	197
4.4.5.2	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	197
4.5	锯齿波单触发脉冲 三相互补	198
4.5.1	概要	198
4.5.2	运行说明	200
4.5.3	Smart Configurator 设置	202
4.5.4	流程图	205
4.5.5	相关运行	208
4.5.5.1	每个间隔的单独死区时间自动设置	208
4.5.6	使用时的注意事项	209
4.5.6.1	多通道计数开始	209
4.5.6.2	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	209
4.6	三角波 PWM 模式 1	210
4.6.1	概要	210
4.6.2	运行说明	212
4.6.3	Smart Configurator 设置	214
4.6.4	流程图	217
4.6.5	相关运行	220
4.6.5.1	每个间隔的单独死区时间自动设置	220
4.6.5.2	具有死区时间自动设置功能的缓冲运行	220
4.6.5.3	当不使用死区时间自动设置功能	221
4.6.6	使用时的注意事项	223
4.6.6.1	多通道计数开始	223
4.6.6.2	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	223
4.7	三角波 PWM 模式 2	224
4.7.1	概要	224
4.7.2	运行说明	226

4.7.3	Smart Configurator 设置	228
4.7.4	流程图	231
4.7.5	相关运行	235
4.7.5.1	每个间隔的单独死区时间自动设置	235
4.7.5.2	具有死区时间自动设置功能的缓冲运行	235
4.7.5.3	当不使用死区时间自动设置功能	236
4.7.6	使用时的注意事项	238
4.7.6.1	多通道计数开始	238
4.7.6.2	比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)	238
4.8	三角波 PWM 模式 3	239
4.8.1	概要	239
4.8.2	运行说明	241
4.8.3	Smart Configurator 设置	243
4.8.4	流程图	246
4.8.5	相关运行	249
4.8.5.1	每个间隔的单独死区时间自动设置	249
4.8.5.2	具有死区时间自动设置功能的缓冲运行	249
4.8.5.3	当不使用死区时间自动设置功能	250
4.8.6	使用时的注意事项	252
4.8.6.1	多通道计数启动	252
4.8.6.2	比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)	252
4.9	锯齿波 PWM 模式 占空比 0%到 100%.....	253
4.9.1	概要	253
4.9.2	运行说明	255
4.9.3	Smart Configurator 设置	256
4.9.4	流程图	258
4.9.5	使用时的注意事项	260
4.9.5.1	比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)	260
4.9.5.2	比较匹配时 100%占空比输出	260
4.9.5.3	占空比 0%和 100%输出期间比较匹配运行	261
4.10	锯齿波单触发脉冲 占空比 0%到 100%	262
4.10.1	概要	262

4.10.2	运行说明	264
4.10.3	Smart Configurator 设置	267
4.10.4	流程图	270
4.10.5	使用时的注意事项	272
4.10.5.1	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	272
4.10.5.2	比较匹配时 100%占空比输出	272
4.10.5.3	占空比 0%和 100%输出期间比较匹配运行	272
4.10.5.4	从占空比 0%和 100%切换	273
4.11	锯齿波 PWM 模式 占空比 0%和 100%	274
4.11.1	概要	274
4.11.2	运行说明	275
4.11.3	Smart Configurator 设置	276
4.11.4	流程图	278
4.11.5	相关运行	280
4.11.5.1	在初始值低时保持 0%后更改占空比到 100%	280
4.11.5.2	在初始值低时保持 100%后更改占空比到 0%	281
4.11.5.3	在初始值高时保持 0%后更改占空比到 100%	282
4.11.5.4	在初始值高时保持 100%后更改占空比到 0%	283
4.11.5.5	在保持 50%后更改占空比到 100%	284
4.11.5.6	在保持 50%后更改占空比到 0%	285
4.11.6	使用时的注意事项	286
4.11.6.1	比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)	286
4.11.6.2	计数开始时 GTUDDTYC.OADTY 设置的反映	286
4.11.6.3	使用 Smart Configurator 在计数开始时占空比的反映	287
4.11.6.4	比较匹配时 100%占空比输出	287
4.11.6.5	占空比 0%和 100%输出期间的比较匹配运行	287
4.12	三角波 PWM 模式 1 占空比 0%到 100%	288
4.12.1	概要	288
4.12.2	运行说明	290
4.12.3	Smart Configurator 设置	293
4.12.4	流程图	296
4.12.5	使用时的注意事项	298

4.12.5.1	比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)	298
4.12.5.2	比较匹配时 100%占空比输出	298
4.12.5.3	占空比 0%和 100%输出期间比较匹配运行	299
4.12.5.4	从占空比 0%和 100%切换	299
4.13	三角波 PWM 模式 1 占空比 0%和 100%	300
4.13.1	概要	300
4.13.2	运行说明	302
4.13.3	Smart Configurator 设置	304
4.13.4	流程图	307
4.13.5	相关运行	309
4.13.5.1	在初始值低时保持 0%后更改占空比到 100%	309
4.13.5.2	在初始值低时保持 100%后更改占空比到 0%	310
4.13.5.3	在初始值高时保持 0%后更改占空比到 100%	311
4.13.5.4	在初始值高时保持 100%后更改占空比到 0%	313
4.13.5.5	在保持 50%后更改占空比到 100%	314
4.13.5.6	在保持 50%后更改占空比到 0%	315
4.13.6	使用时的注意事项	316
4.13.6.1	比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)	316
4.13.6.2	计数开始时 GTUDDTYC.OADTY 设置的反映	316
4.13.6.3	使用 Smart Configurator 在计数开始时占空比的反映	317
4.13.6.4	比较匹配时的 100%占空比输出	317
4.13.6.5	占空比 0%和 100%输出期间比较匹配运行	317
5	如何导入项目	318
5.1	使用 e ² studio 导入	318
5.2	使用 CS+导入	319
6	参考文献	320

1 MTU 和 GPTW 的规格

下表列出了与 MTU 和 GPTW 的 PWM 输出相关的主要功能。

表 1.1 PWM 输出功能

项目		MTU	GPTW
基本功能	通道数量	9 通道	10 通道
	计数器精度	16 位 ^{注1}	32 位
	最大工作频率	与 CPU 频率相同	与 CPU 频率相同
	同步运行	✓	✓
通用功能	PWM 输出	✓	✓
	互补 PWM 输出	✓	✓
	PWM 输出保护功能 (POE)	✓	✓
	左右不对称三角 PWM 波	✓	✓
	0%或 100%占空比输出	✓	✓
GPTW 特有功能	锯齿波互补 PWM	—	✓
	左/右死区时间控制	—	✓

注 1: 级联时 32 位可用, 但 PWM 输出期间仅 16 位可用。

MTU 每个通道支持不同的功能。下面列出了每个通道的功能。

表 1.2 功能（按通道）

项目	MTU									GPTW 注 1
	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5	MTU6	MTU7	MTU9	
计数时钟	14	11	12	11	11	10	11	11	14	13
同步运行	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
I/O 引脚 数量	4	2	2	4	4	3	4	4	4	2
缓冲运行	✓	-	-	✓	✓	-	✓	✓	✓	✓
PWM 模式 1 注 2	2	1	1	2	2	-	2	2	2	-
PWM 模式 2 注 2	4	2	2	-	-	-	-	-	4	-
互补 PWM 模式 1/2/3 注 2	-	-	-	2	4	-	2	4	-	-
复位同步 PWM 模式 注 2	-	-	-	2	4	-	2	4	-	-
锯齿波 PWM 模式 注 2	-	-	-	-	-	-	-	-	-	2
锯齿波 单触发 脉冲模式 注 2	-	-	-	-	-	-	-	-	-	2
三角波 PWM 模式 1/2/3 注 2	-	-	-	-	-	-	-	-	-	2
死区时间自 动设置功能	-	-	-	-	-	-	-	-	-	✓

注 1：表示 1 个通道的功能。GPTW 有 10 个相同功能的通道。

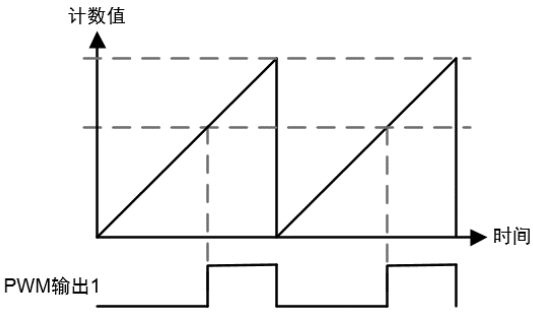
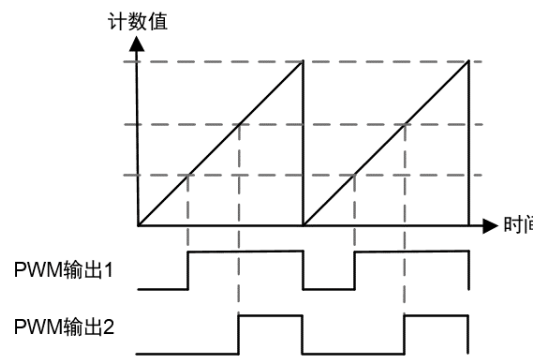
注 2：数字表示 PWM 输出引脚的数量。正相和反相各算一个引脚。

以下章节提供了 MTU 和 GPTW 的 PWM 功能详细信息以及规格差异。

1.1 运行模式和输出波形

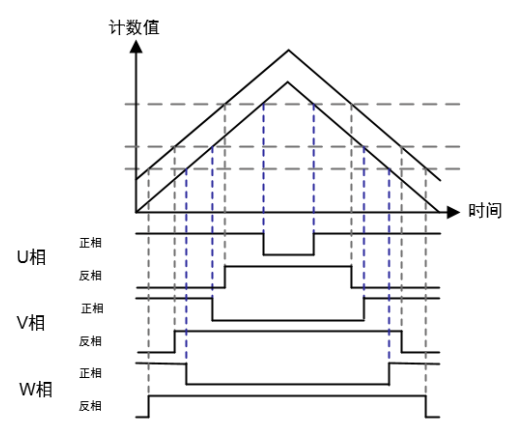
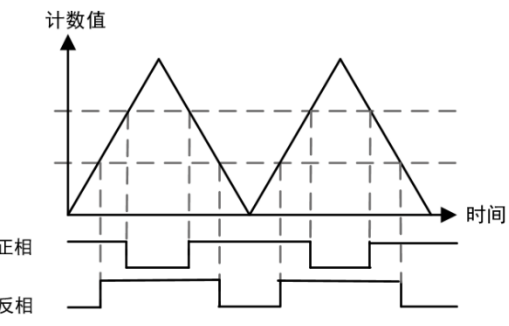
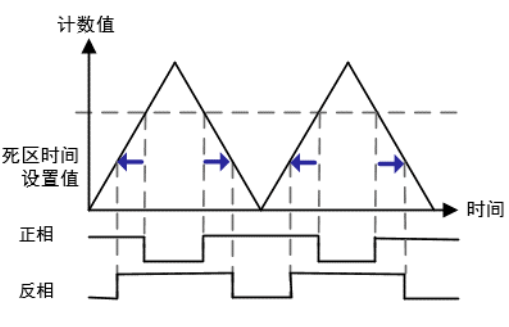
本节描述了 MTU 和 GPTW 的模式和输出波形之间的关系。MTU 和 GPTW 的 PWM 输出模式和输出波形如下表所示。

表 1.3 PWM 输出

输出波形	定时器	描述
	MTU	• PWM 模式 1 - 最多可输出 14 相 PWM 波形 - 每个 PWM 输出使用一个周期寄存器和一个占空比寄存器 - 可以通过同步两个或多个相来使能同步运行
	GPTW	• 锯齿波 PWM 模式 - 最多可输出 20 相 PWM 波形 - 每两个 PWM 输出使用一个周期寄存器和与 PWM 输出相同数量的占空比寄存器 - 可以通过同步三个或多个相来使能同步运行
	MTU	• PWM 模式 2 - 最多可输出 12 相 PWM 波形 - 每多个 PWM 输出使用一个周期寄存器和与 PWM 输出相同数量的占空比寄存器 - 可以通过同步三个或多个相来使能同步运行
	GPTW	• 锯齿波 PWM 模式 - 最多可输出 20 相 PWM 波形 - 每两个 PWM 输出使用一个周期寄存器和与 PWM 输出相同数量的占空比寄存器 - 可以通过同步三个或多个相来使能同步运行

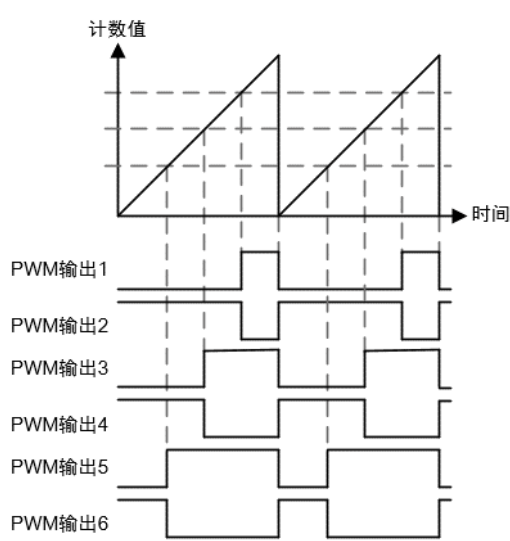
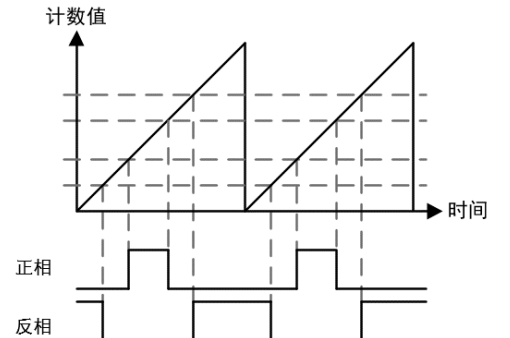
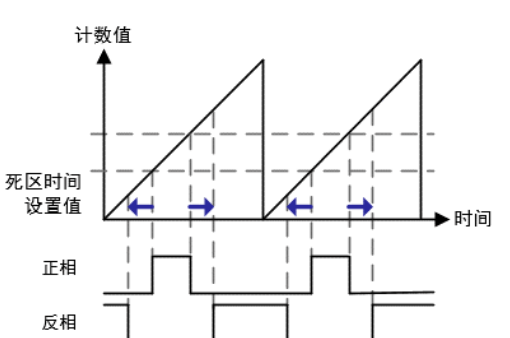
下面描述了 MTU 和 GPTW 互补 PWM 输出（三角波）的模式和输出波形。

表 1.4 互补 PWM 模式（三角波）

输出波形	定时器	描述
	MTU	• 互补 PWM 模式 1/2/3 - 三相互补最多 2 个通道 - 可以通过改变每 1 相的一个比较值来改变占空比 - 死区时间自动设置功能
	GPTW	• 三角波 PWM 模式 1/2/3 - 单相互补最多 10 个通道 - 可以通过分别改变正相和反相中的一个比较值来改变占空比 - 死区时间是每个正相和反相比较值之间的差 - 同步操作可用于同步两个或多个通道（示例：三个通道用于三相互补 + 一个通道用于单相，两个通道用于五相互补）
		• 三角波 PWM 模式 1/2/3 + 死区时间自动设置功能 - 单相互补最多 10 个通道 - 正相占空比可以通过改变正相比较值来改变 - 反相占空比可以通过设置左右死区时间来改变 - 同步运行可用于同步两个或多个通道（示例：三个通道用于三相互补 + 一个通道用于单相，两个通道用于五相互补）

下面描述了 MTU 和 GPTW 互补 PWM 输出（锯齿波）的模式和输出波形。

表 1.5 互补 PWM 模式（锯齿波）

输出波形	定时器	描述
	MTU	- 复位同步 PWM 模式 - 三相互补最多两个通道
	GPTW	- 锯齿波 PWM 模式 + 同步运行 - 单相互补最多 10 个通道 - 每个正相和反相设置相同比较值 - GTIOCnA 引脚输出设置：初始输出为低电平，在 GTCCRA 寄存器比较匹配时输出高电平，周期结束时输出低电平 - GTIOCnB 引脚输出设置：初始输出为高电平，在 GTCCRB 寄存器比较匹配时输出低电平，周期结束时输出高电平
	GPTW	- 锯齿波单触发脉冲模式 - 单相互补最多 10 个通道 - 使用双缓冲器设置正相和反相的比较值 - 死区时间是每个正相和反相比较值之间的差 - 同步运行可用于同步两个或多个通道（示例：三个通道用于三相互补 + 一个通道用于单相，两个通道用于五相互补）
	GPTW	- 锯齿波单触发脉冲模式 + 死区时间自动设置功能 - 单相互补最多 10 个通道 - 反相占空比可以通过设置左右死区时间来改变 - 同步运行可用于同步两个或多个通道（示例：三个通道用于三相互补 + 一个通道用于单相，两个通道用于五相互补）

1.2 缓冲功能

MTU 和 GPTW 都具有缓冲功能。缓冲结构和传送时间因模式而异。

GPTW 支持在缓冲寄存器写入时禁止缓冲传送的功能。详情参考 RX66T Group User's Manual: Hardware, 章节 24.8.2 禁止缓冲运行。

PWM 模式 1/2 和复位同步 PWM 模式下 MTU 的缓冲结构和传送时间如下。传送时间是指从缓冲寄存器传送到寄存器的时间。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.3 缓冲运行。

表 1.6 MTU PWM 模式 1/2

通道	寄存器	缓冲寄存器	传送时间
MTU0, MTU9	TGRA	TGRC	在 PWM 模式 1 或 2 下 • 比较匹配 • 计数器清除
	TGRB	TGRD	
	TGRE	TGRF	
MTU3, MTU4, MTU6, MTU7	TGRA	TGRC	在 PWM 模式 1 下 • 比较匹配 • 计数器清除
	TGRB	TGRD	

复位同步 PWM 模式下 MTU 的缓冲结构和传送时间如下。传送时间是指从缓冲寄存器传送到寄存器的时间。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.15 复位同步 PWM 模式下的缓冲运行和比较匹配。

表 1.7 MTU 复位同步 PWM 模式

通道	寄存器	缓冲寄存器	传送时间
MTU3, MTU6	TGRA	TGRC	• 周期结束 (MTU3.TGRA, MTU6.TGRA 比较匹配)
	TGRB	TGRD	

互补 PWM 模式（单缓冲）下 MTU 的缓冲结构和传送时间如下。MTU3.TGRA，MTU6.TGRA，TCDRA 和 TCDRB 寄存器的传送时间是指数据从缓冲寄存器传送到寄存器的时间。对于所有其他寄存器，是指数据从暂存器传送到寄存器的时间。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式，(b) 寄存器运行。

表 1.8 MTU 互补 PWM 模式 1/2/3

通道	寄存器	暂存器	缓冲寄存器	传送时间
MTU3	TGRA	-	TGRC	互补 PWM 模式 1 • 波峰 互补 PWM 模式 2 • 波谷 互补 PWM 模式 3 • 波峰和波谷
	TGRB	TEMP1A	TGRD	
MTU4	TGRA	TEMP2A	TGRC	
	TGRB	TEMP3A	TGRD	
MTU6	TGRA	-	TGRC	
	TGRB	TEMP4A	TGRD	
MTU7	TGRA	TEMP5A	TGRC	
	TGRB	TEMP6A	TGRD	
MTU	TCDRA	-	TCBRA	
	TCDRB	-	TCBRB	

互补 PWM 模式（双缓冲）下 MTU 的缓冲结构和传送时间如下。MTU3.TGRA, MTU6.TGRA, TCDRA 和 TCDRB 寄存器的传送时间是指数据从缓冲寄存器传送到寄存器的时间。对于所有其他寄存器，请参考下表中的时间。

设置缓冲寄存器 A（MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC 和 MTU7.TGRD）时，请同时设置缓冲寄存器 B（MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE 和 MTU7.TGRF）。

缓冲寄存器 A 是递减计数时的比较值；缓冲寄存器 B 是递增计数时的比较值。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式, (s) 互补 PWM 模式下的双缓冲功能。

表 1.9 MTU 互补 PWM 模式 3（使用双缓冲）

通道	寄存器	暂存器	缓冲寄存器	双缓冲寄存器	传送时间
MTU3	TGRB	TEMP1A	TGRD	-	当写入 MTU4.TGRD 或 MTU7.TGRD 时, TGRC, TGRD, TGRE 和 TGRF 的值被传送到相应的暂存器。
		TEMP1B	-	TGRE	
MTU4	TGRA	TEMP2A	TGRC	-	
		TEMP2B	-	TGRE	
	TGRB	TEMP3A	TGRD	-	从 TEMP1A, TEMP2A, TEMP3A, TEMP4A, TEMP5A 和 TEMP6A 到相应的寄存器的传送时间是在波峰。
		TEMP3B	-	TGRF	
MTU6	TGRB	TEMP4A	TGRD	-	从 TEMP1B, TEMP2B, TEMP3B, TEMP4B, TEMP5B 和 TEMP6B 到相应的寄存器的传送时间是在波谷。
		TEMP4B	-	TGRE	
MTU7	TGRA	TEMP5A	TGRC	-	
		TEMP5B	-	TGRE	
	TGRB	TEMP6A	TGRD	-	
		TEMP6B	-	TGRF	
MTU3	TGRA	-	TGRC	-	波峰和波谷
MTU6	TGRA	-	TGRC	-	
MTU	TCDRA	-	TCBRA	-	
	TCDRB	-	TCBRB	-	

GPTW 在锯齿波 PWM 模式和三角波 PWM 模式 1/2 下的缓冲结构和传送时间如下。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.2 缓冲运行。

表 1.10 GPTW 锯齿波 PWM 模式和三角波 PWM 模式 1/2

寄存器	缓冲寄存器	双缓冲寄存器	传送时间
GTPR	GTPBR	GTPDBR	锯齿波 PWM 模式 • 溢出/下溢 • 计数器清除 - 硬件 - 软件 三角波 PWM 模式 1/2 • 波谷
GTCCRA	GTCCRC	GTCCRD	锯齿波 PWM 模式 • 溢出/下溢 • 计数器清除 - 硬件 - 软件
GTCCRB	GTCCRE	GTCCRF	• 强制缓冲传送 三角波 PWM 模式 1 • 波谷 • 强制缓冲传送 三角波 PWM 模式 2 • 波峰和波谷 • 强制缓冲传送
GTDVU ^注	GTDBU	-	三角波 PWM 模式 1/2
GTDVD ^注	GTDBD	-	• 计数器清除 - 清除 GTCNT 计数器 • 波谷

注：不能在锯齿波 PWM 模式下使用。

GPTW 在锯齿波单触发脉冲模式下的缓冲结构和传送时间如下。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.3 PWM 输出运行模式, (2) 锯齿波单触发脉冲模式。

表 1.11 GPTW 锯齿波单触发脉冲模式

寄存器	缓冲寄存器	暂存器	双缓冲寄存器	传送时间
GTPR	GTPBR	-	GTPDBR	• 溢出/下溢 • 计数器清除 - 硬件 - 软件
GTCCRA	GTCCRC	-	-	在周期结束时 (溢出/下溢): 从 GTCCRC 传送到 GTCCRA, 从 GTCCRE 传送到 GTCCRB, 从 GTCCRD 传送到暂存器 A 和从 GTCCRF 传送到暂存器 B
	-	暂存器 A	GTCCRD	
GTCCRB	GTCCRE	-	-	在 GTCCRA 比较匹配时, 从暂存器 A 传送到 GTCCRA; 在 GTCCRB 比较匹配时, 从暂存器 B 传送到 GTCCRB
	-	暂存器 B	GTCCRF	
GTDVU	GTDBU	-	-	• 计数器清除 - 清除 GTCNT 计数器 • 溢出/下溢

GPTW 在三角波 PWM 模式 3 下的缓冲结构和传送时间如下。

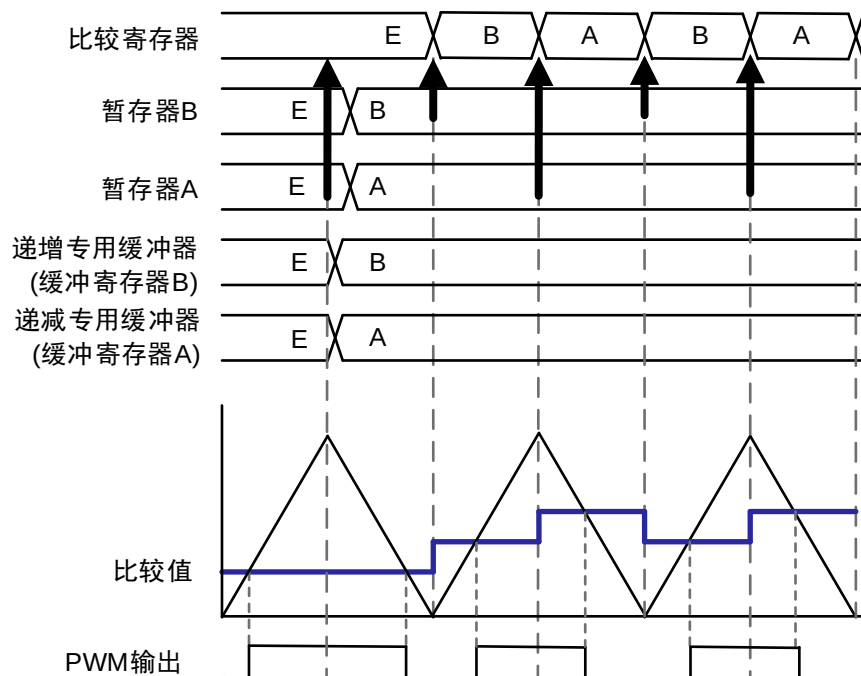
详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.3 PWM 输出操作模式, (5) 三角波 PWM 模式 3 (波谷 64 位传送)。

表 1.12 GPTW 三角波 PWM 模式 3

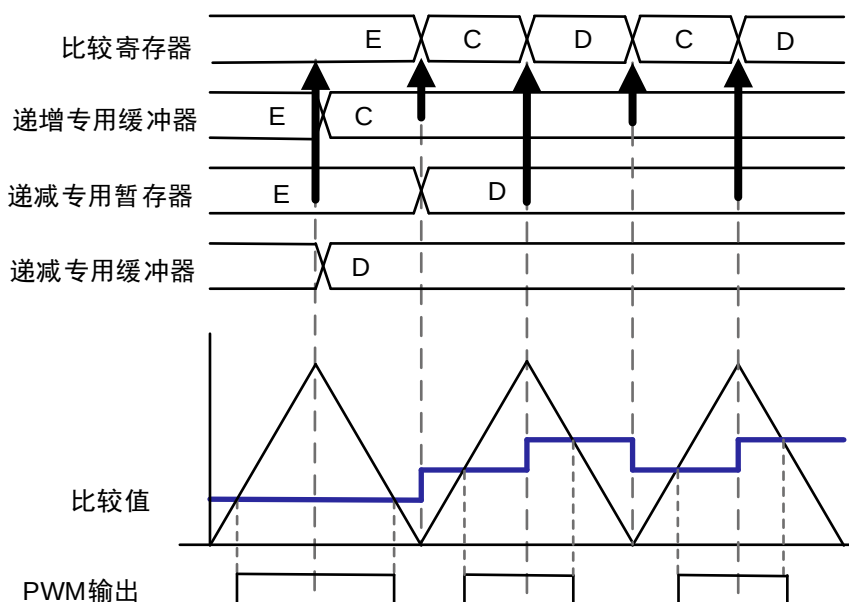
寄存器	缓冲寄存器	暂存器	双缓冲寄存器	传送时间
GTPR	GTPBR	-	GTPDBR	• 波谷
GTCCRA	GTCCRC	-	-	波谷: 从 GTCCRC 传送到 GTCCRA, 从 GTCCRE 传送到 GTCCRB, 从 GTCCRD 传送到暂存器 A 和从 GTCCRF 传送到暂存器 B
	-	暂存器 A	GTCCRD	
GTCCRB	GTCCRE	-	-	波峰: 从暂存器 A 传送到 GTCCRA 和从暂存器 B 传送到 GTCCRB
	-	暂存器 B	GTCCRF	
GTDVU	GTDBU	-	-	• 计数器清除 - 清除 GTCNT 计数器 • 波谷
GTDVD	GTDBD	-	-	

MTU 和 GPTW 的双缓冲结构和传送时间如下图所示。

该示例显示了在波峰中断发生时，在递增计数期间专用缓冲（递增专用缓冲）或递减计数期间专用缓冲（递减专用缓冲）中设置下一个比较值的情况。尽管结构不同，但在这两种情况下，左右不对称互补 PWM 是在每个载波周期的一个中断处理中实现的。



(1) MTU互补PWM模式3

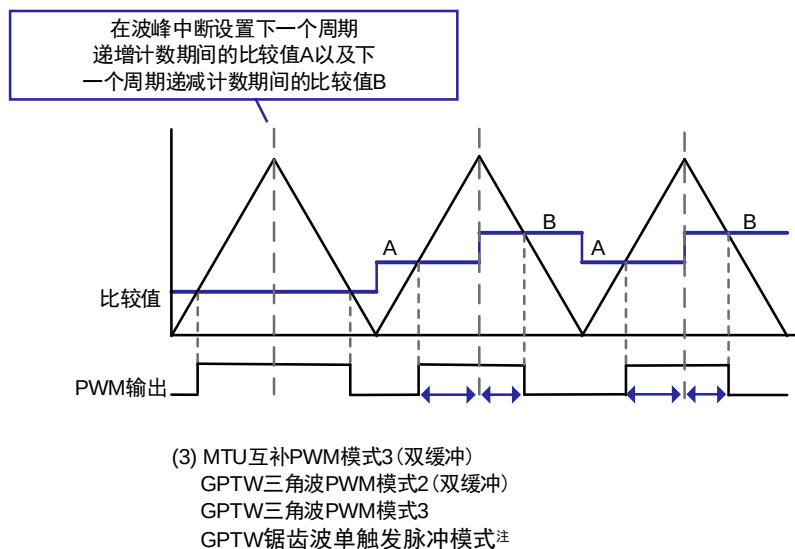
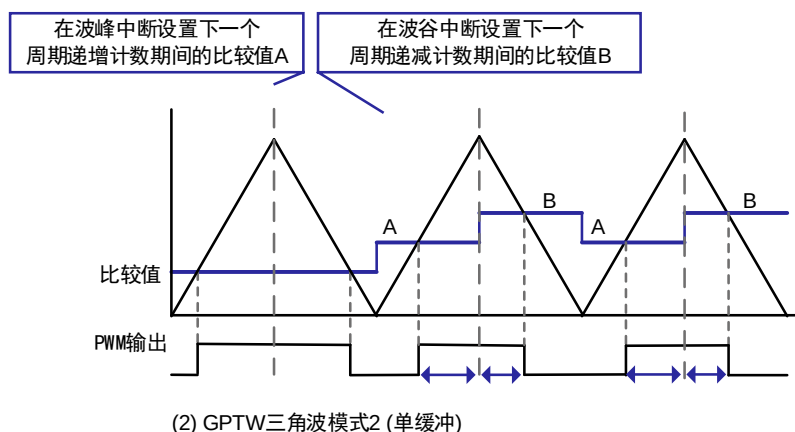
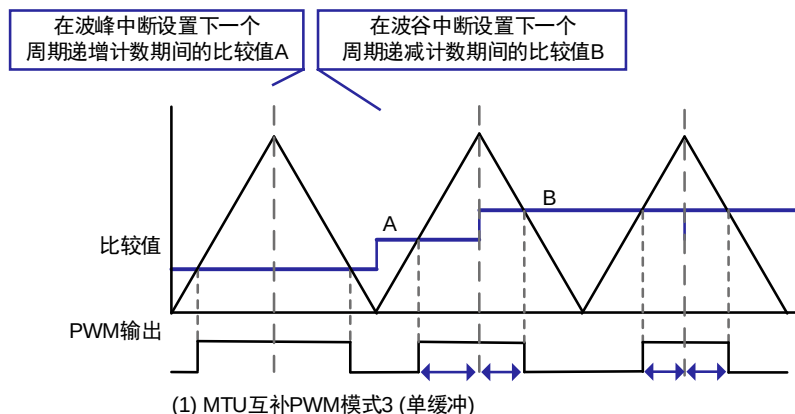


(2) GPTW三角波PWM模式3

图 1.1 双缓冲结构

1.3 左右不对称互补 PWM 输出

MTU 和 GPTW 能够输出左右不对称互补 PWM。要实现左右不对称互补 PWM 输出，在下图所示的时间设置比较寄存器。



注：在锯齿波单触发脉冲模式中，在周期结束时设置比较值（溢出/下溢）。

图 1.2 左右不对称互补 PWM 输出

1.4 死区时间自动设置功能

MTU 和 GPTW 的死区时间自动设置如下所示。对于 MTU，递增计数和递减计数设置相同的值，但 GPTW 可以设置两个不同的值。

有关 MTU 的详情，请参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式，(e) 死区时间设置。

有关 GPTW 的详情，请参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

表 1.13 死区时间自动设置功能

项目	MTU	GPTW
可用模式	互补 PWM 模式 1/2/3	- 锯齿波单触发脉冲模式 - 三角波 PWM 模式 1/2/3
设置寄存器	递增和递减计数同用一个寄存器	一个递增计数专用寄存器 一个递减计数专用寄存器
在运行中更改设置	禁止	使能
输出保护功能	-	✓

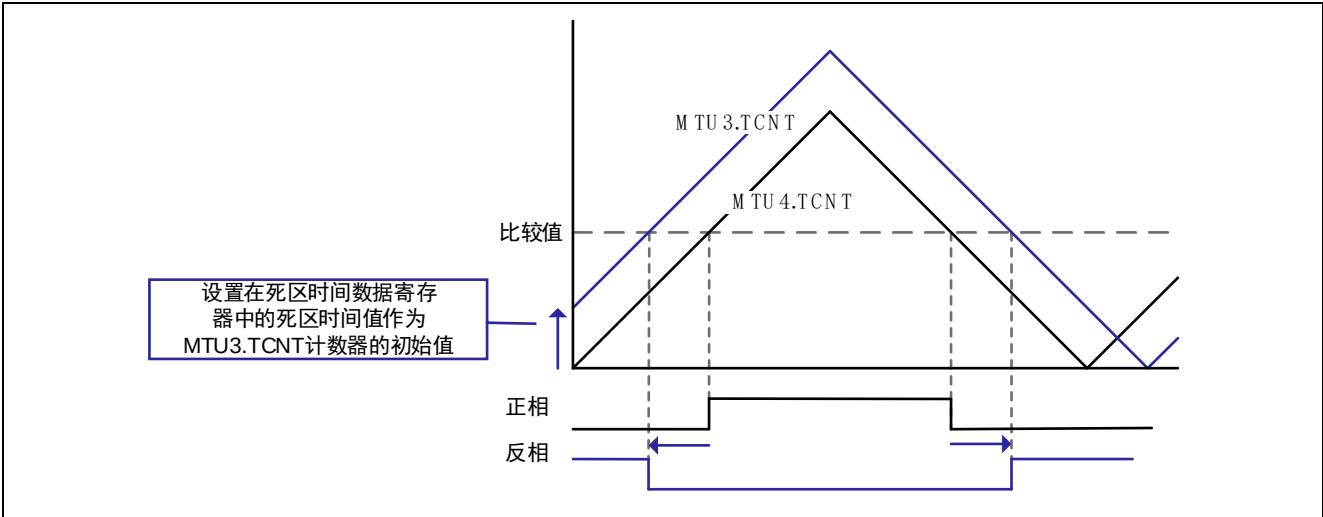
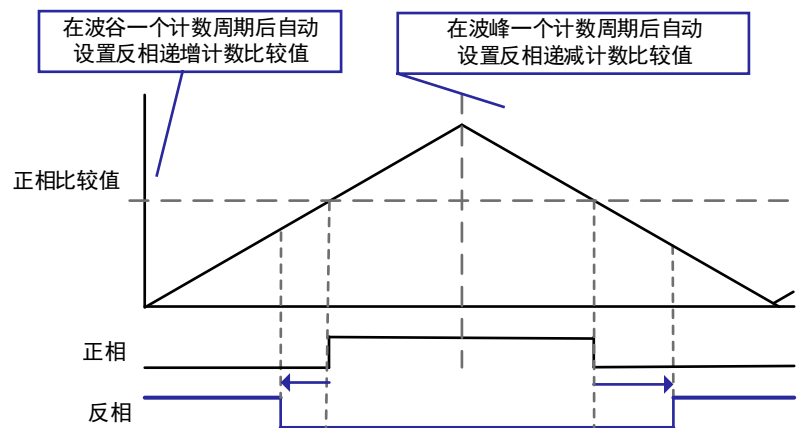
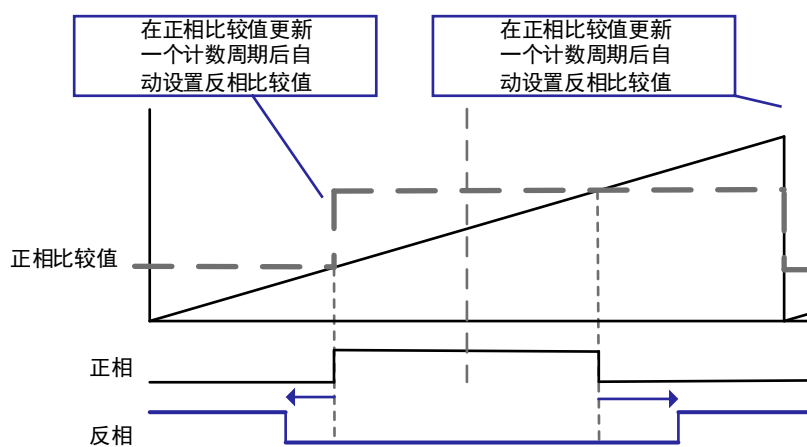


图 1.3 MTU 死区时间自动设置 (MTU3, MTU4)



(1) 三角波PWM模式1



(2) 锯齿波单触发脉冲模式

图 1.4 GPTW 死区时间自动设置

1.5 占空比 0%/100%输出功能

下面描述了 MTU 和 GPTW 输出占空比 0%和 100%的设置方法。

表 1.14 占空比 0%和 100%的设置方法

设置方法	MTU	GPTW
比较寄存器设置 MTU: TGR 寄存器 GPTW: GTCCR 寄存器	- PWM 模式 1/2 100%: 比较值 = 周期寄存器值 0%: 比较值 > 周期寄存器值 - 互补 PWM 模式 1/2/3 100%: 比较值 = 0000h 0%: 比较值 = 周期寄存器值 对于某些情况, 上面的设置可能不充分。如果有必要, 联合使用 TIOR。	根据模式, GTCCR 寄存器和 GTIOR 被用作实现占空比 0%/100%输出。 详细信息, 请参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间的 GTCCRM 寄存器设置 (m = A 到 F)。
引脚输出占空比设置的位设置 GPTW: 设置 GTUDDTYC.OADTY[1:0] 和 OBDTY[1:0]	-	详细信息, 请参考 RX66T Group User's Manual: Hardware, 章节 24.3.6, 占空比 0%/100%输出功能。
I/O 控制寄存器设置 MTU: TIOR 寄存器 GPTW: GTIOR 寄存器	详细信息, 请参考 RX66T Group User's Manual: Hardware, 章节 22.2.6, 定时器 I/O 控制寄存器 (TIOR)。	详细信息, 请参考 RX66T Group User's Manual: Hardware, 章节 24.2.14, 通用 PWM 定时器 I/O 控制寄存器(GTIOR)。

1.6 周期和周期寄存器之间的关系

下面描述了 MTU 和 GPTW 中周期和周期寄存器之间的关系。

表 1.15 周期和周期寄存器的关系

关系表达式	MTU	GPTW
周期 = 周期寄存器值 + 1	• PWM 模式 1/2 • 复位-同步 PWM 模式	• 锯齿波 PWM 模式 • 锯齿波单触发脉冲
周期 = 周期寄存器值 x 2	• 互补 PWM 模式 1/2/3	• 三角波 PWM 模式 1/2/3

2 运行确认条件

本应用说明中包含的示例代码已在以下运行条件下得到确认。

表 2.1 运行确认环境

项目	描述
MCU	R5F566TEADFP (包含在 Renesas Starter Kit for RX66T)
运行频率	主时钟: 8MHz PLL: 160MHz (主时钟 x 1/1 x 20) HOCO: 停止 LOCO: 停止 系统时钟(ICLK): 160MHz (PLL x 1/1) 外围模块时钟 A (PCLKA): 80MHz (PLL x 1/2) 外围模块时钟 B (PCLKB): 40MHz (PLL x 1/4) 外围模块时钟 C (PCLKC): 160MHz (PLL x 1/1) 外围模块时钟 D (PCLKD): 40MHz (PLL x 1/4) FlashIF 时钟(FCLK): 40MHz (PLL x 1/4)
运行电压	3.3V
集成开发环境(IDE)	Renesas Electronics e ² studio Version 2021-07
C 编译器 ^注	Renesas Electronics C/C++ Compiler Package for RX Family V3.03.00 编译器选项 使用集成开发环境的默认设置
iodefine.h 版本	V1.00
字节序	小端法
运行模式	单片机模式
处理器模式	管理器模式
示例代码版本	V1.00
硬件板	Renesas Starter Kit for RX66T (Product number: RTK50566T0CxxxxxBE)
仿真器	E2-Lite

注: 导入与原始项目中指定的相同版本的工具链 (C 编译器)。如果导入目的地中没有相同的工具链, 则无法选择该工具链, 会出现错误。检查项目设置界面上的工具链选择状态。

设置方法参考 FAQ3000404。

FAQ3000404: 尝试构建导入工程 (e² studio) 时出现 “在 PATH 中找不到程序 ‘make’ ” 错误。

3 MCU 示例代码

3.1 共通点

3.1.1 示例代码列表

本应用说明提供的以下代码都是使用 Smart Configurator 创建的。如果未标明占空比，则输出波形以占空比 20%→40%→60%→80%→20%→...重复。

示例代码可以从瑞萨电子网站下载。

表 3.1 MTU 示例代码列表 (1/2)

名称	描述	参考
PWM 模式 1 比较匹配 r01an5995_rx66t_mtu3_pwm1_cmp.zip	使用 PWM 模式 1 TGRA 比较匹配发生时缓冲传送，输出 PWM 波形	3.2
PWM 模式 1 计数清除 r01an5995_rx66t_mtu3_pwm1_cntclear.zip	使用 PWM 模式 1 计数器清除发生时缓冲传送，输出 PWM 波形	3.3
PWM 模式 1 无缓冲寄存器 r01an5995_rx66t_mtu3_pwm1.zip	使用 PWM 模式 1 不使用缓冲器，输出 PWM 波形	3.4
PWM 模式 2 比较匹配 r01an5995_rx66t_mtu3_pwm2_cmp.zip	使用 PWM 模式 2 TGRA 比较匹配发生时缓冲传送，输出 2 相 PWM 波形	3.5
PWM 模式 2 计数清除 r01an5995_rx66t_mtu3_pwm2_cntclear.zip	使用 PWM 模式 2 计数器清除发生时缓冲传送，输出 2 相 PWM 波形	3.6
PWM 模式 2 无缓冲寄存器 r01an5995_rx66t_mtu3_pwm2.zip	使用 PWM 模式 2 不使用缓冲器，输出 2 相 PWM 波形	3.7
复位同步 PWM 模式 r01an5995_rx66t_mtu3_reset_sync_pwm.zip	使用复位同步 PWM 模式 输出无死区时间的三相互补 PWM 波形 每相输出 25%，50%和 75%固定占空比	3.8
互补 PWM 模式 带双缓冲器 r01an5995_rx66t_mtu3_complementary_pwm_dblbuf.zip	使用互补 PWM 模式 3（在波峰和波谷传送） 使用双缓冲器产生左右不对称占空比并输出带死区时间的三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	3.9
互补 PWM 模式 无双缓冲器 r01an5995_rx66t_mtu3_complementary_pwm.zip	使用互补 PWM 模式 3（在波峰和波谷传送） 无需使用双缓冲器即可生成对称占空比，并输出带死区时间的三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	3.10

表 3.2 MTU 示例代码列表 (2/2)

名称	描述	参考
PWM 模式 1 占空比 0%到 100% (在比较匹配时修改比较寄存器) r01an5995_rx66t_mtu3_pwm1_50to100.zip	使用 PWM 模式 1 输出包括占空比 0%和 100%的 PWM 波形 在比较匹配时修改比较寄存器 输出占空比: 50%→80%→100%→80%→50%→0%→...	3.11
PWM 模式 1 占空比 0%到 100% (在计数器清除时修改比较寄存器) r01an5995_rx66t_mtu3_pwm1_50to100_rwcc.zip	使用 PWM 模式 1 输出包括占空比 0%和 100%的 PWM 波形 在计数器清除时修改比较寄存器 输出占空比: 50%→80%→100%→80%→50%→0%→...	3.12
PWM 模式 1 占空比 0%和 100% r01an5995_rx66t_mtu3_pwm1_0to100.zip	使用 PWM 模式 1 输出占空比 0%和 100%之间交替的 PWM 波形 输出占空比: 0%→100%→0%→100%→...	3.13
互补 PWM 模式 占空比 0%到 100% r01an5995_rx66t_mtu3_complementary_pwm_50to100.zip	使用互补 PWM 模式 2 (在波谷传送) 输出包括占空比 0%和 100%的 PWM 波形 输出占空比: 50%→80%→100%→80%→50%→0%→...	3.14
互补 PWM 模式 占空比 0%和 100% r01an5995_rx66t_mtu3_complementary_pwm_0to100.zip	使用互补 PWM 模式 2 (在波谷传送) 输出占空比 0%和 100%之间交替的 PWM 波形 输出占空比: 0%→100%→0%→100% →...	3.15

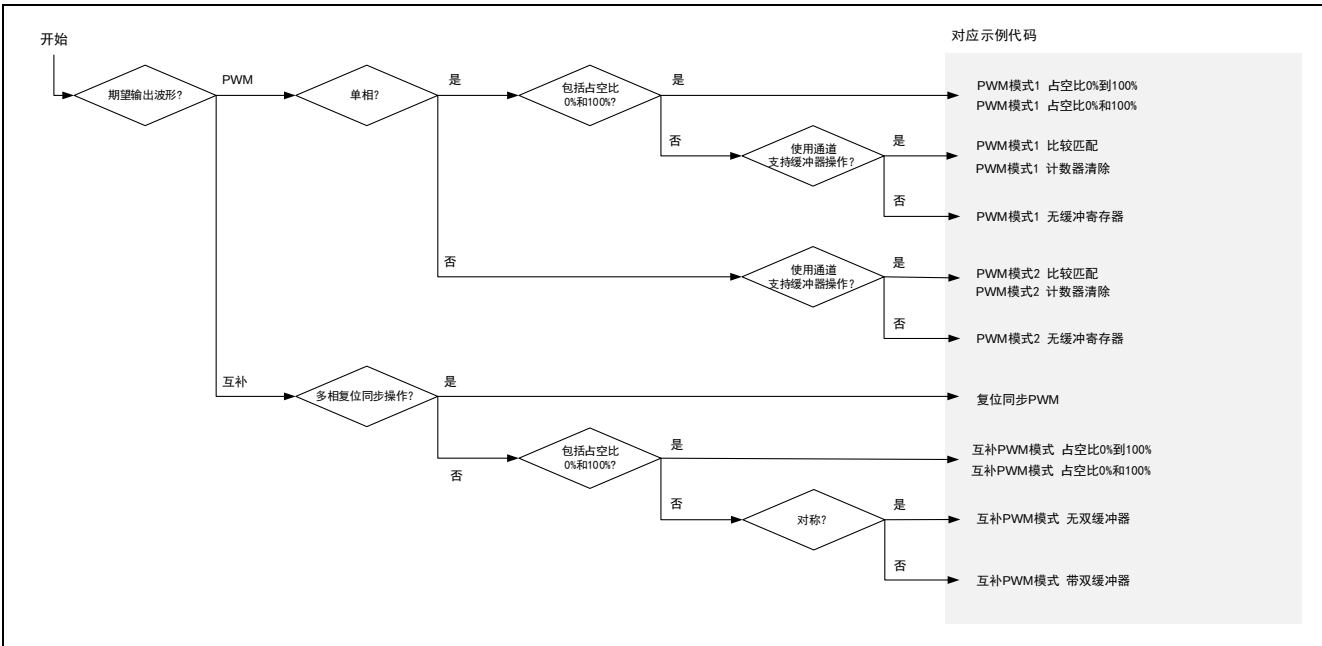


图 3.1 根据用途的示例代码

3.1.2 文件夹结构

示例代码的主要文件夹结构如下。

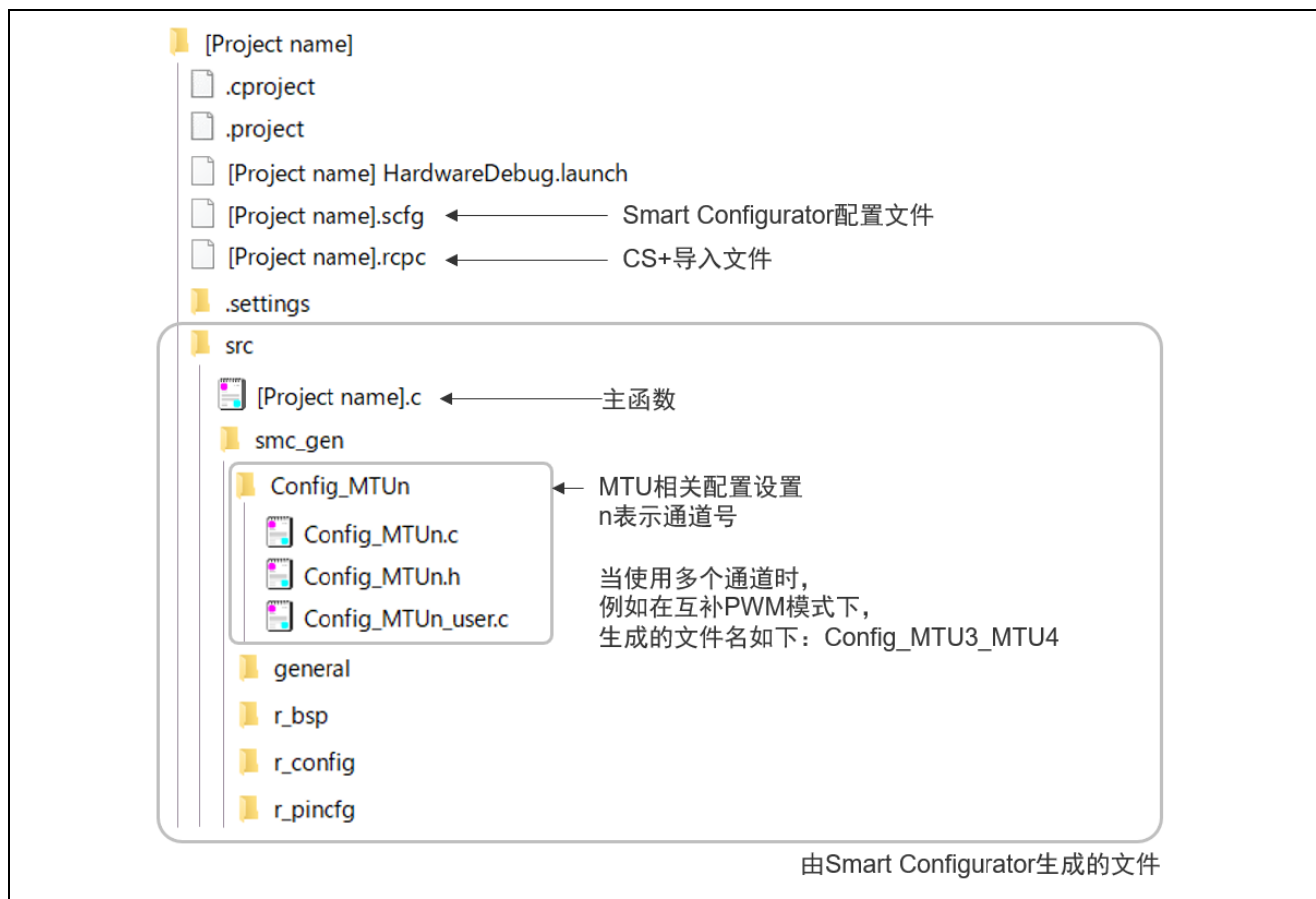


图 3.2 MTU 文件夹结构

3.1.3 文件结构

示例代码的主要文件结构如下。

表 3.3 MTU 文件结构

文件名称	描述
[Project name].c	<u>main function</u> 这是主函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。
Config_MTUn.c*	<u>R_Config_MTUn_Create function</u> 这是 MTU 的初始化函数。 基于 Smart Configurator 中设置的初始化函数由 Smart Configurator 生成。 此函数的调用由 Smart Configurator 生成。该函数在主函数之前执行的 R_SystemInit 函数中调用。
	<u>R_Config_MTUn_Start function</u> 这是 MTU 的计数开始函数。 此函数由 Smart Configurator 生成。 在示例代码中，这个函数在主函数中调用。
	<u>R_Config_MTUn_Stop function</u> 这是 MTU 的计数停止函数。 此函数由 Smart Configurator 生成。 示例代码中未使用此函数。
Config_MTUn_user.c*	<u>r_Config_MTUn_Create_UserInit function</u> 这是 MTU 的用户初始化函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。 这是由 Smart Configurator 生成的 R_Config_MTUn_Create 函数中调用的最后一个函数。
	<u>r_Config_MTUn [interrupt name] interrupt function</u> 这是中断处理函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。
Config_MTUn.h*	这是定义 MTU 相关函数的头文件。 该文件包含在 Smart Configurator 生成的 r_smc_entry.h 文件中。 要使用 MTU 相关函数，请务必包含 r_smc_entry.h 文件。

*: n 表示通道号

3.1.4 添加组件

如下所述，示例代码使用 Smart Configurator 添加 MTU。

表 3.4 添加组件

项目	描述
组件	参考每个示例代码的章节（下图中（1））
配置名称	示例代码使用默认设置名称
运行	参考每个示例代码的章节（下图中（2））
资源	参考每个示例代码的章节（下图中（3））

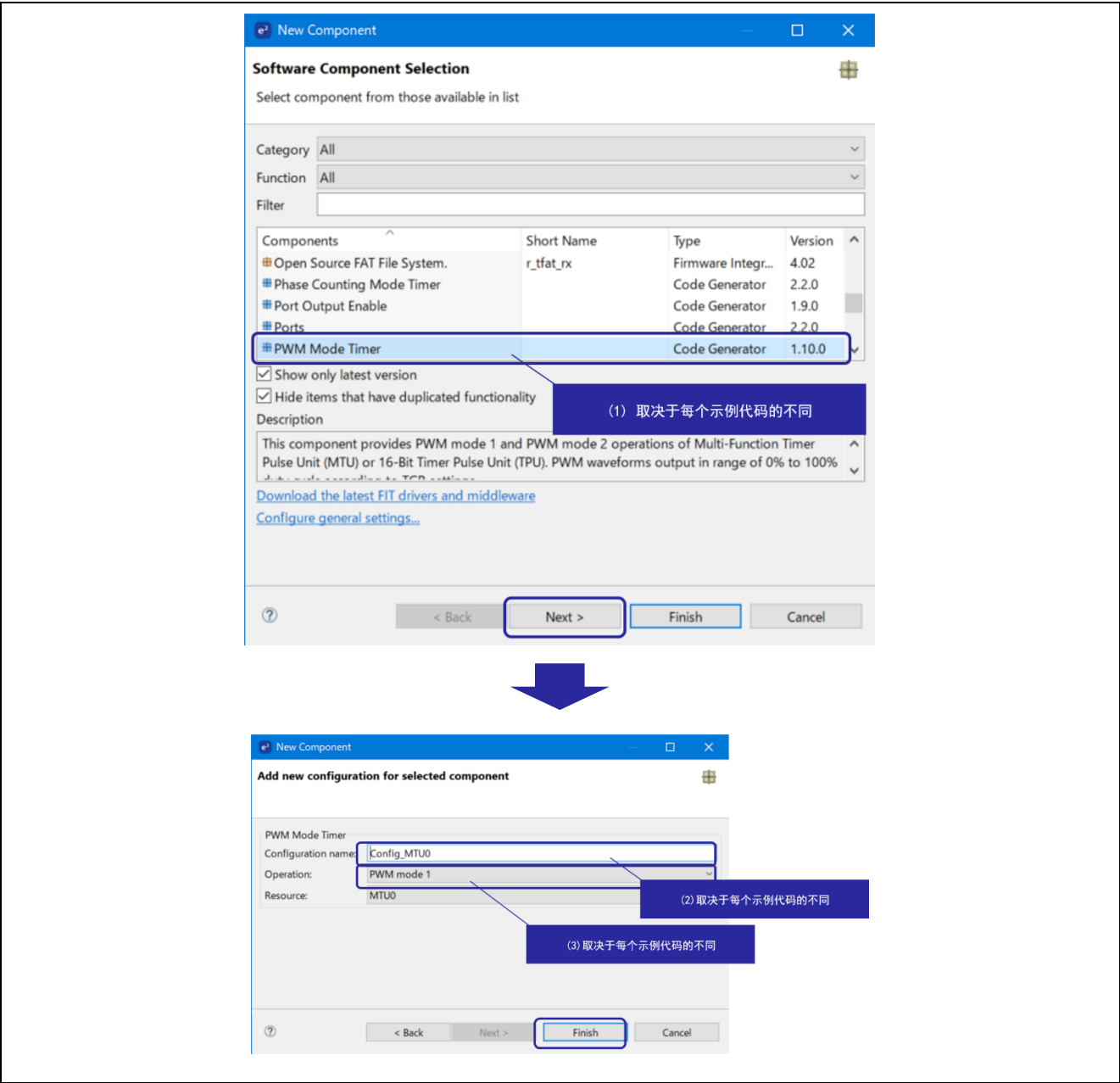


图 3.3 添加组件

3.1.5 引脚设置

图 3.4 显示了使用 Smart Configurator 进行引脚设置的示例。
设置 MTU 后配置引脚。有关 MTU 设置，请参考每个示例代码的“Smart Configurator Settings”。
引脚设置在由 Smart Configurator 生成的 R_Config_MTUn_Create 函数中进行。

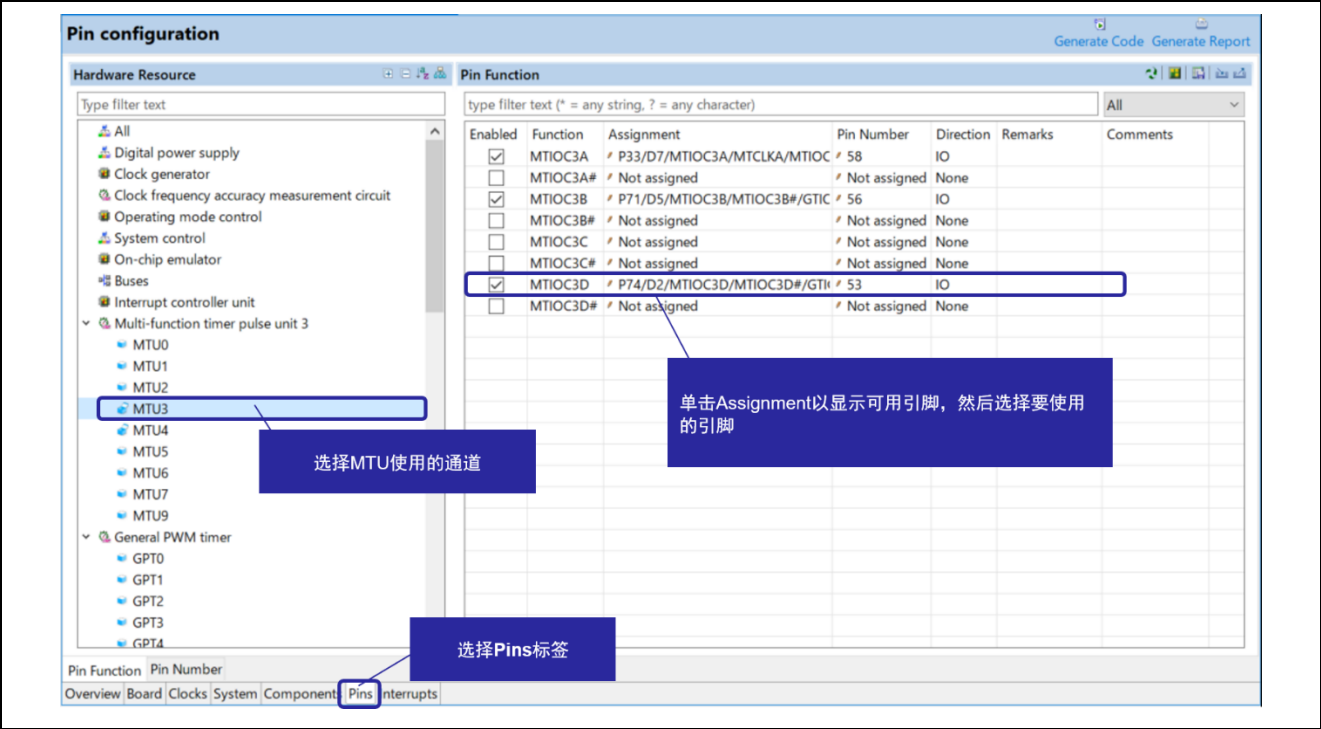


图 3.4 引脚设置

3.1.6 中断设置

图 3.5 显示了使用 Smart Configurator 设置中断的示例。有关软件可配置中断 A 的详细信息，请参考瑞萨 RX66T Group User's Manual Hardware，章节 14.4.5.1 软件可配置中断 A。

设置 MTU 后配置中断。有关 MTU 设置，请参考每个示例代码的“Smart Configurator Settings”。

可以在由 Smart Configurator 生成的 R_Config_MTUn_Create 函数，R_Config_MTUn_Start 函数和 R_Config_MTUn_Stop 函数中配置中断设置。

中断处理函数在由 Smart Configurator 生成的 Config_MTUn_user.c 文件中创建，名称为 r_Config_MTUn_[interrupt name]_interrupt。

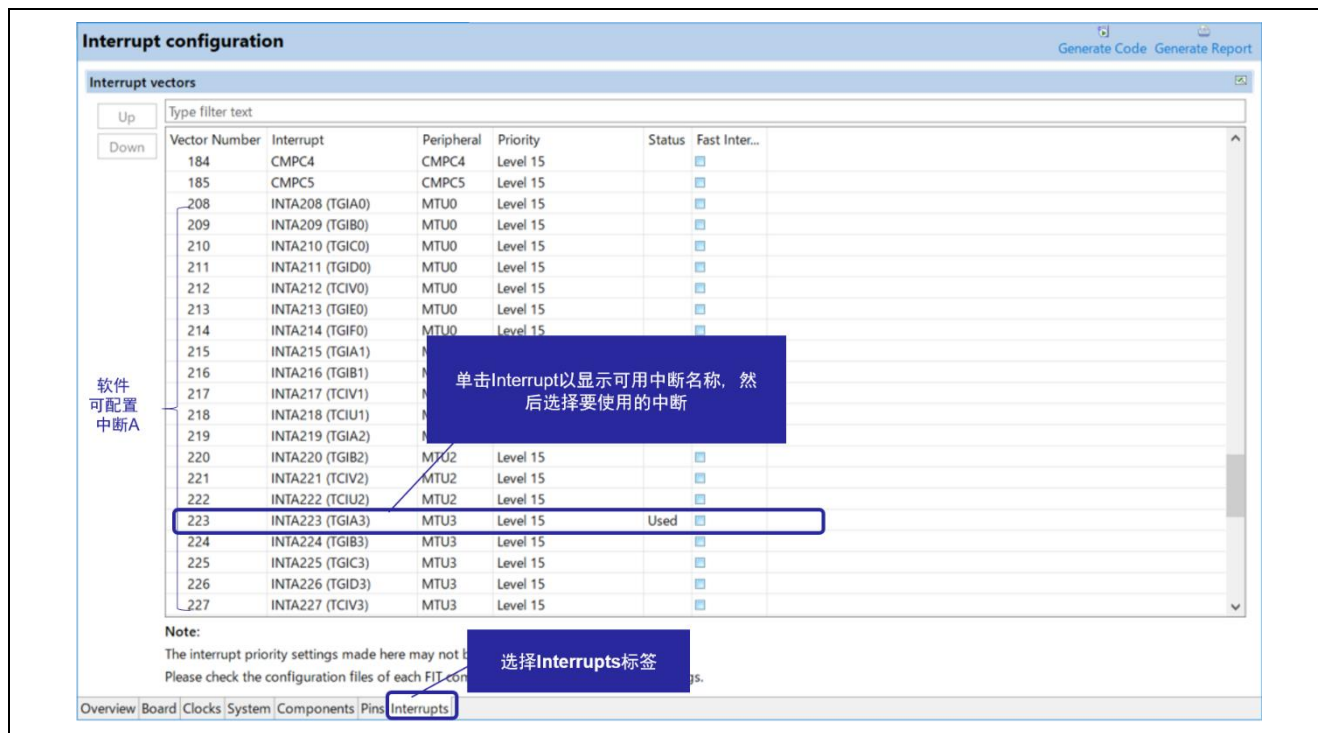


图 3.5 中断设置

3.2 PWM 模式 1 比较匹配

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm1_cmp.zip

3.2.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 1 并重复输出占空比为 20% → 40% → 60% → 80% → 20% → ...的波形。当 TGRA 比较匹配发生时,使用缓冲寄存器 TGRC 将 TGRC 的值传送到占空比寄存器 TGRA 来改变占空比。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - 在 TGRB 比较匹配时输出低
- 使用 TGRA 作为占空比寄存器
 - 在 TGRA 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 TGRC 作为 TGRA 的缓冲寄存器
 - 在 TGRA 比较匹配时缓冲传送
- 每个周期的占空比改变
 - 在 TGRB 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息,请参考图 3.7

在 Smart Configurator 中设置。
设置方法请参考章节 3.2.3。

此示例代码的 PWM 模式 1 输出如下所示。

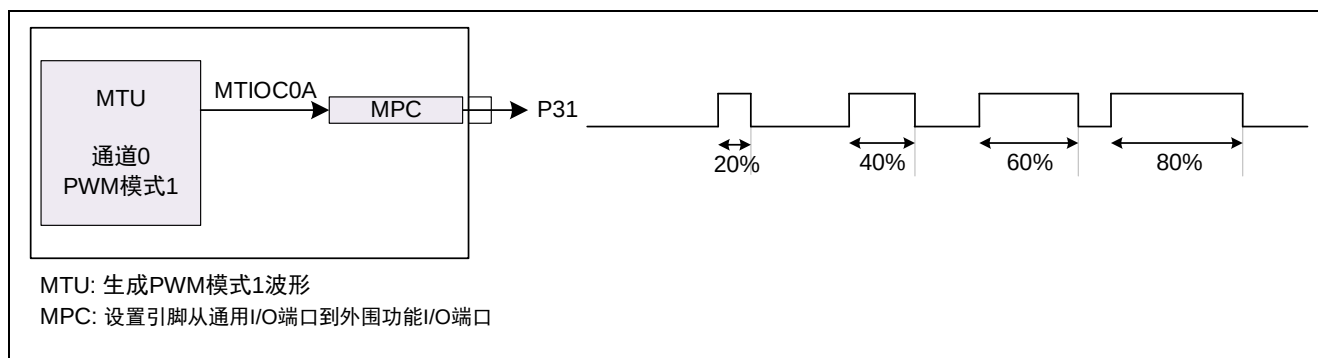
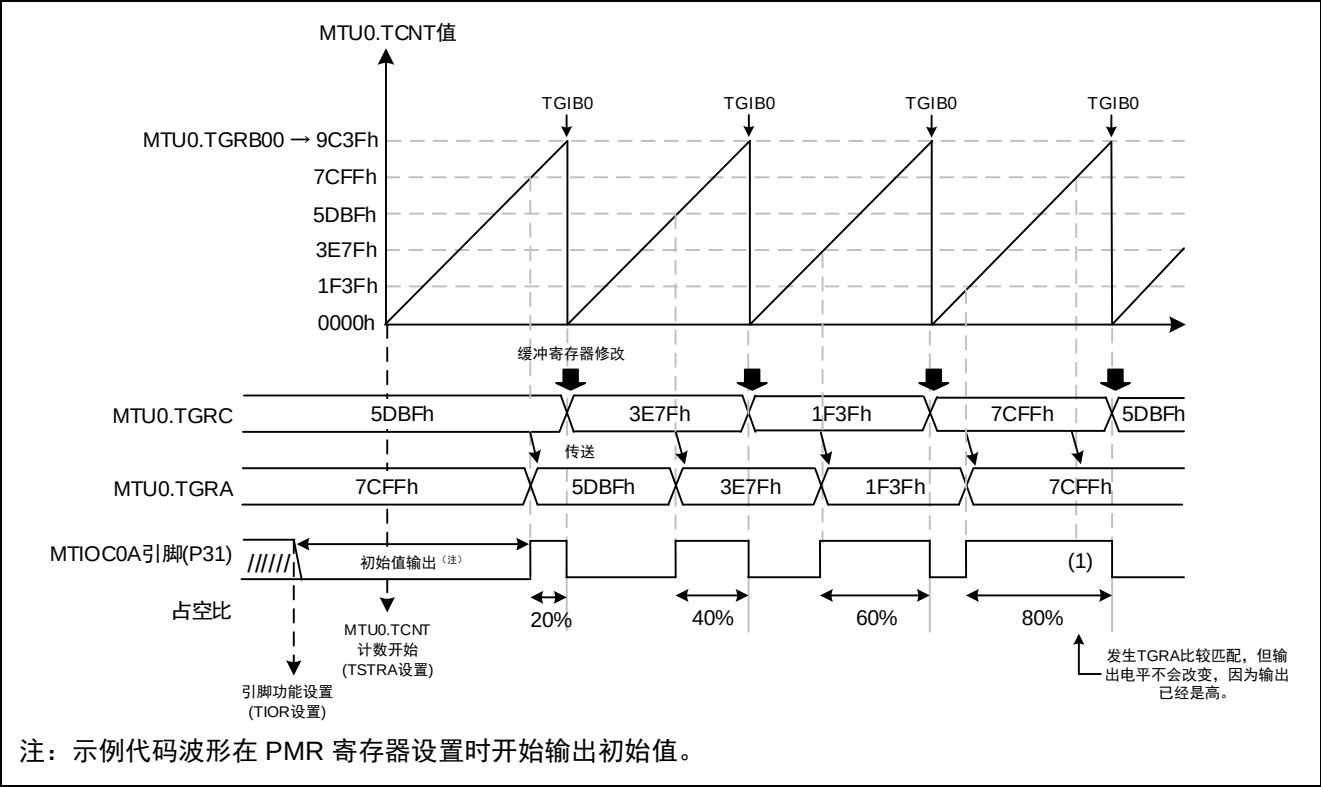


图 3.6 PWM 模式 1 输出

3.2.2 运行说明

示例代码的运行如图 3.7 所示。通过在周期寄存器 TGRB 的 TGRB 比较匹配中断（TGIB0）中修改缓冲寄存器 TGRC 的值，可以在每个周期改变占空比的设置。当 TGRA 比较匹配发生时，TGRC 的值被传送到占空比寄存器 TGRA。

当占空比从 80%切换到 20%时，同一周期内发生两次 TGRA 比较匹配，但波形不会改变，因为第二个比较匹配发生在输出高期间（下图中（1））。



3.2.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.5 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU0
运行	PWM 模式 1
资源	MTU0

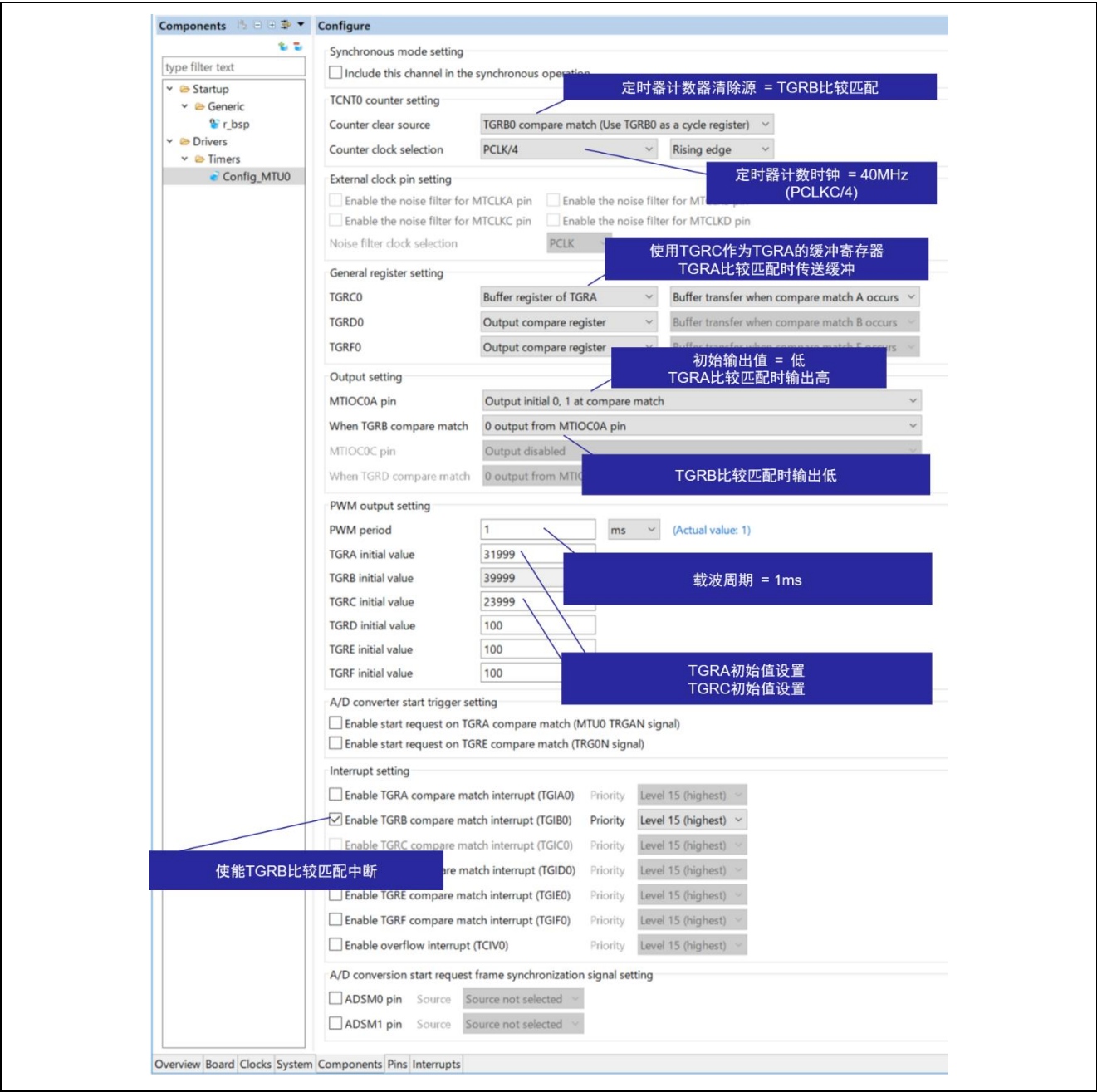


图 3.8 MTU0 设置

3.2.4 流程图

下面显示了主函数流程和当 TGIB0 中断发生时的流程，这两者都是在 Smart Configurator 生成代码后添加的。

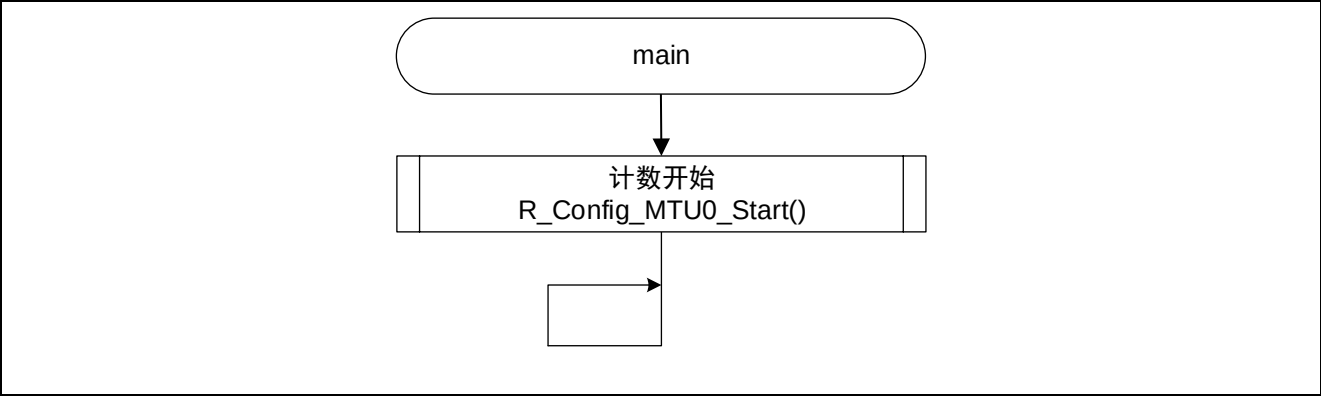


图 3.9 主函数

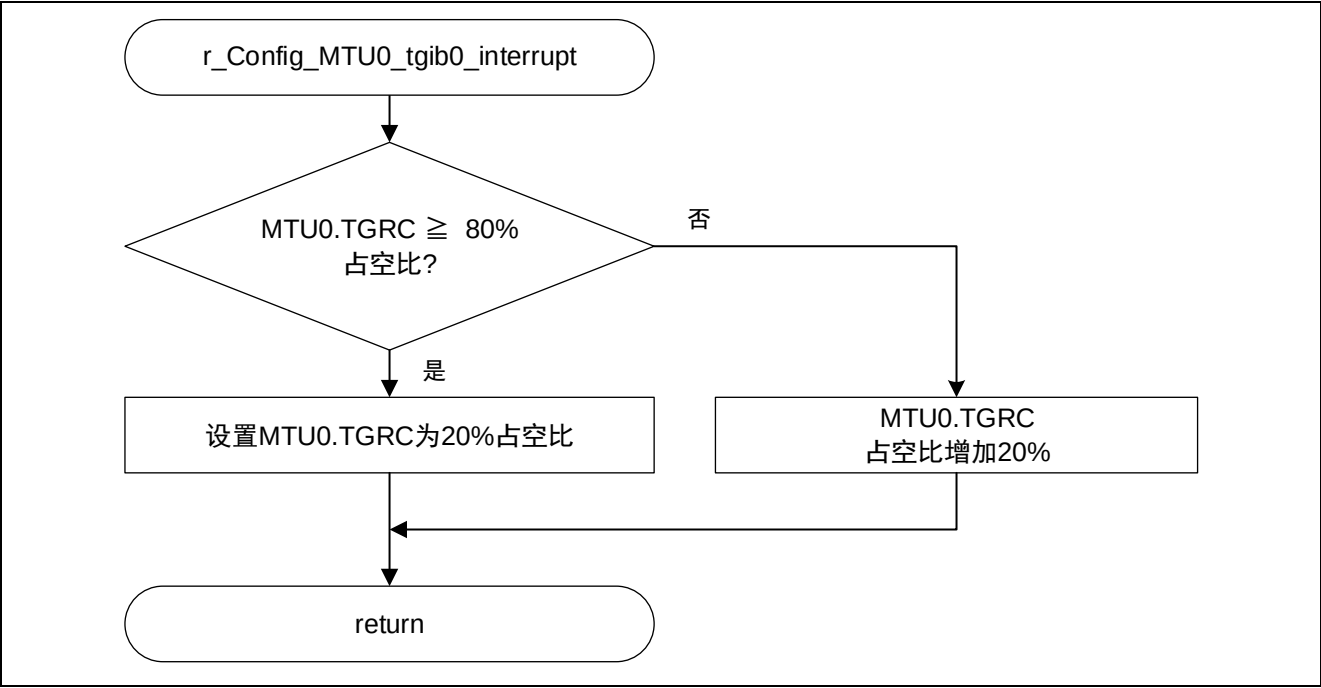


图 3.10 TGIB0 中断处理函数

3.2.5 使用时的注意事项

3.2.5.1 缓冲寄存器修改运行和比较匹配之间的竞争

当在比较匹配时设置缓冲传送时间，如果在缓冲寄存器写周期中发生比较匹配，则修改前的缓冲寄存器中的数据将传送到占空比寄存器。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.7 缓冲寄存器写操作和比较匹配之间的竞争。

3.2.5.2 缓冲寄存器修改延迟

在此示例代码中，当 TGRA 比较匹配发生时，将执行从缓冲寄存器 TGRC 到 TGRA 的传送。必须在下一次比较匹配发生之前完成 TGRC 值的修改。

如果缓冲寄存器修改延迟，则到下一个周期中 TGRA 比较匹配发生时，TGRA 值可能不会更新。在这种情况下，请考虑在计数器清除发生时传送缓冲。

有关 PWM 模式 1 在计数器清除时执行缓冲传送的示例代码，请参考 3.3 PWM 模式 1 计数清除。

3.2.5.3 设置一个大于占空比寄存器的值

在此示例代码中，当 TGRA 比较匹配发生时，将执行从缓冲寄存器 TGRC 到 TGRA 的传送。

在 TGRA 比较匹配发生后，如果在 TGRA 中设置了一个大于当前设置在 TGRA 中的值，则一个周期内可能会发生两次 TGRA 比较匹配。

有关详细信息，请参考图 3.7 中 (1)。

3.3 PWM 模式 1 计数清除

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm1_cntclear.zip

3.3.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比从 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 1 并重复输出占空比为 20% → 40% → 60% → 80% → 20% → ...的波形。当计数器清除发生时,使用缓冲寄存器 TGRC 将 TGRC 的值传送到占空比寄存器 TGRA 来改变占空比。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - TGRB 比较匹配时输出低
- 使用 TGRA 作为占空比寄存器
 - TGRA 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 TGRC 作为 TGRA 的缓冲寄存器
 - 计数器清除发生时缓冲传送
- 每个周期的占空比改变
 - 在 TGRB 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息,请参考图 3.12

在 Smart Configurator 中设置。
设置方法请参考章节 3.3.3。

此示例代码的 PWM 模式 1 输出如下所示。

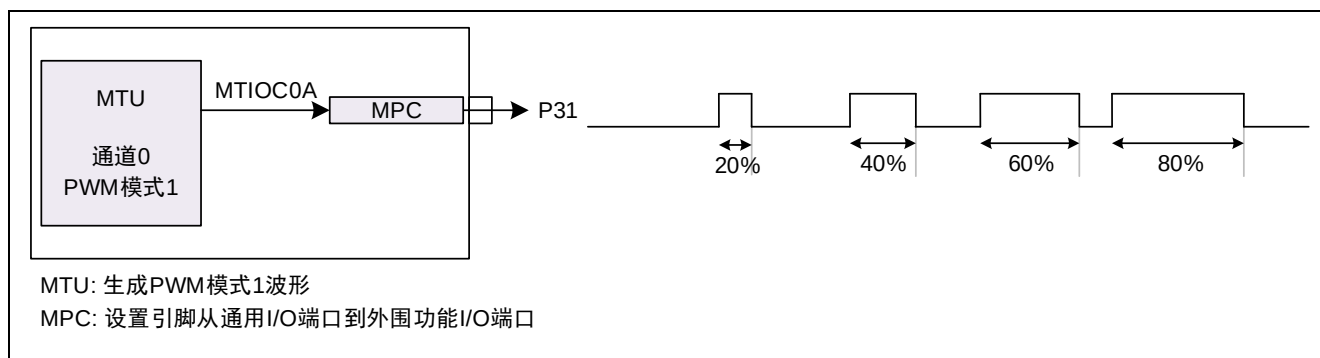


图 3.11 PWM 模式 1 输出

3.3.2 运行说明

示例代码的运行如图 3.12 所示。通过在周期寄存器 TGRB 的 TGRB 比较匹配中断（TGIB0）中修改缓冲寄存器 TGRC 的值，可以在每个周期改变占空比的设置。当计数器清除发生时，TGRC 的值被传送到占空比寄存器 TGRA。

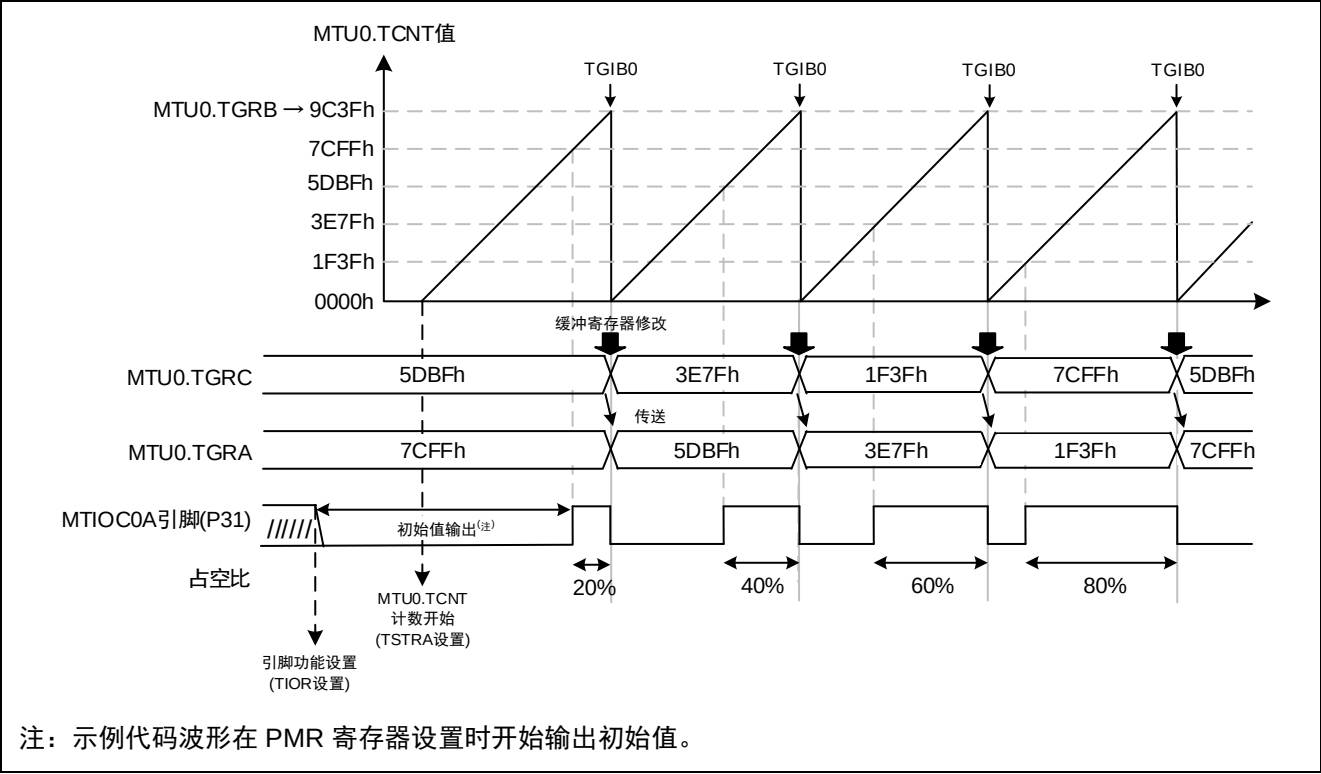


图 3.12 示例代码运行

3.3.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考章节 3.1.4 添加组件。

表 3.6 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU0
运行	PWM 模式 1
资源	MTU0

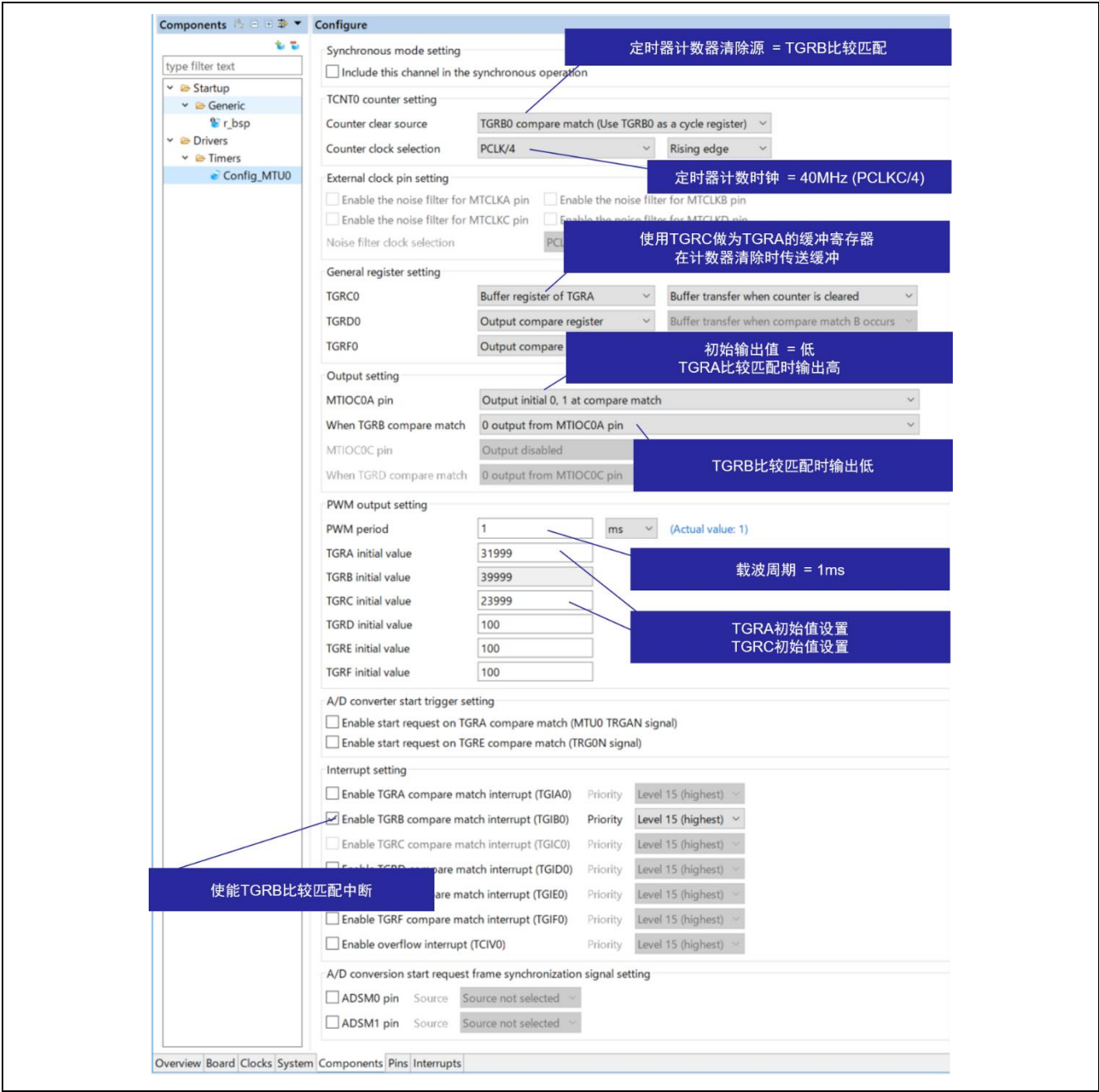


图 3.13 MTU0 设置

3.3.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。

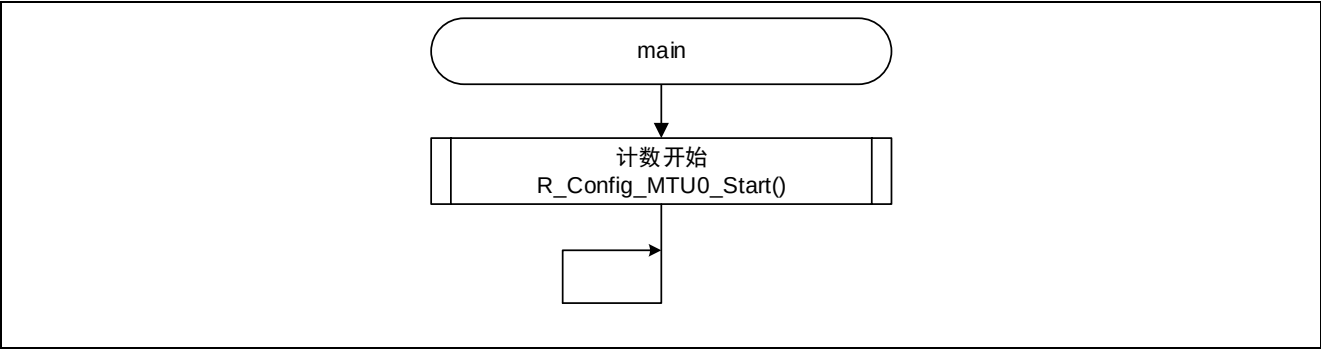


图 3.14 主函数

TGIB0 中断处理函数根据 TGRC 寄存器的当前值来改变数值。

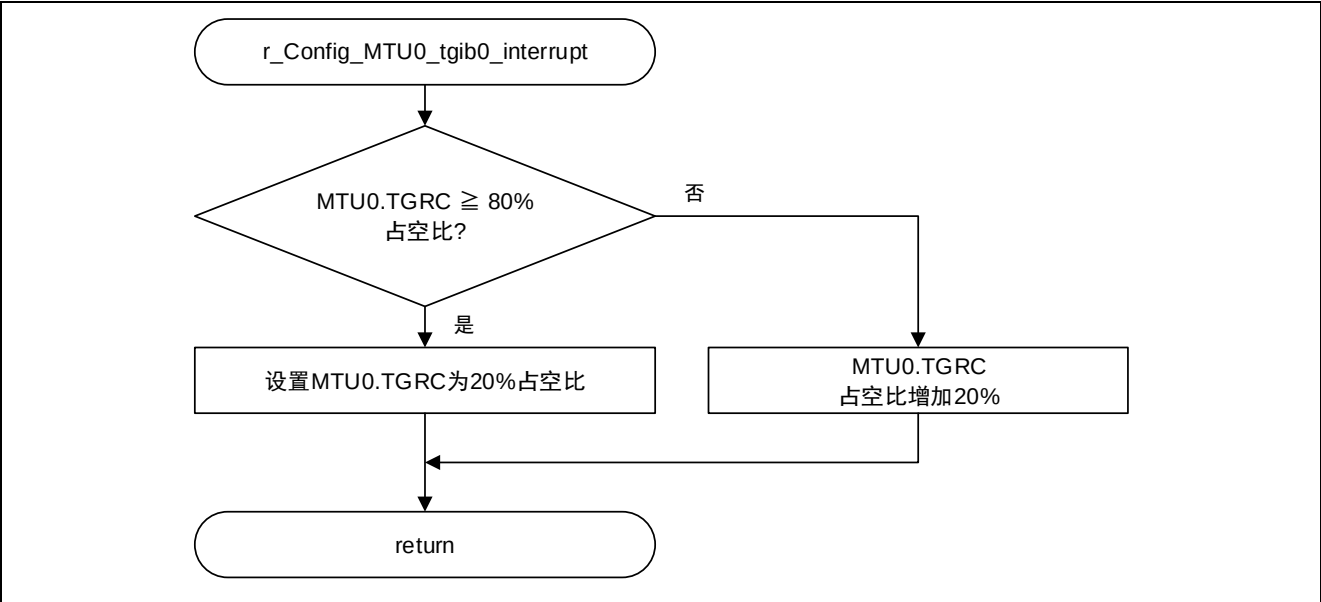


图 3.15 TGIB0 中断处理函数

3.3.5 使用时的注意事项

3.3.5.1 缓冲寄存器修改运行和计数器清除之间的竞争

如果在计数器清除时设置缓冲传送时间，在缓冲寄存器写周期中发生计数器清除，修改前的缓冲寄存器数据被传送到占空比寄存器。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.8 缓冲寄存器写和 TCNT 清除运行之间的竞争。

3.3.5.2 缓冲寄存器修改延迟

在此示例代码中，当 TGRA 计数器清除发生时，执行从缓冲寄存器 TGRC 到 TGRA 的传送。TGRC 值的修改必须在下一个计数器清除发生前完成。

如果缓冲寄存器 TGRC 的修改延迟，则无法输出所需的占空比。

3.4 PWM 模式 1 无缓冲寄存器

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm1.zip

3.4.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 1 并重复占空比 20% → 40% → 60% → 80% → 20% → ...的波形输出。不使用缓冲寄存器, 通过更新占空比寄存器 TGRA 的值来改变占空比。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 1
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - TGRB 比较匹配时输出低
- 使用 TGRA 作为占空比寄存器
 - TGRA 比较匹配时输出高
- 每个周期的占空比改变
 - 在 TGRA 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 3.17

在 Smart Configurator 中设置。
设置方法请参考章节 3.4.3。

此示例代码的 PWM 模式 1 输出如下所示。

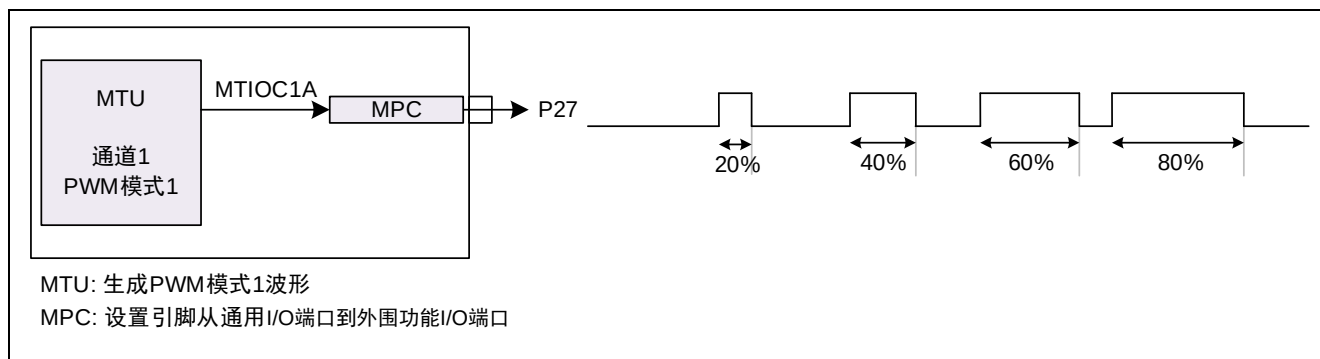


图 3.16 PWM 模式 1 输出

3.4.2 运行说明

示例代码的运行如图 3.17 所示。通过在占空比寄存器 TGRA 的比较匹配中断（TGIA1）处修改 TGRA 的值，可以在每个周期改变占空比的设置。

当占空比从 80%切换到 20%时，同一个周期内发生两次 TGRA 比较匹配，但波形不会变化，因为第二个比较匹配发生在高电平输出期间（下图中（1））。占空比设置随着每个周期而改变，因此需要调整修改 TGRA 值的时间（下图中（2））。

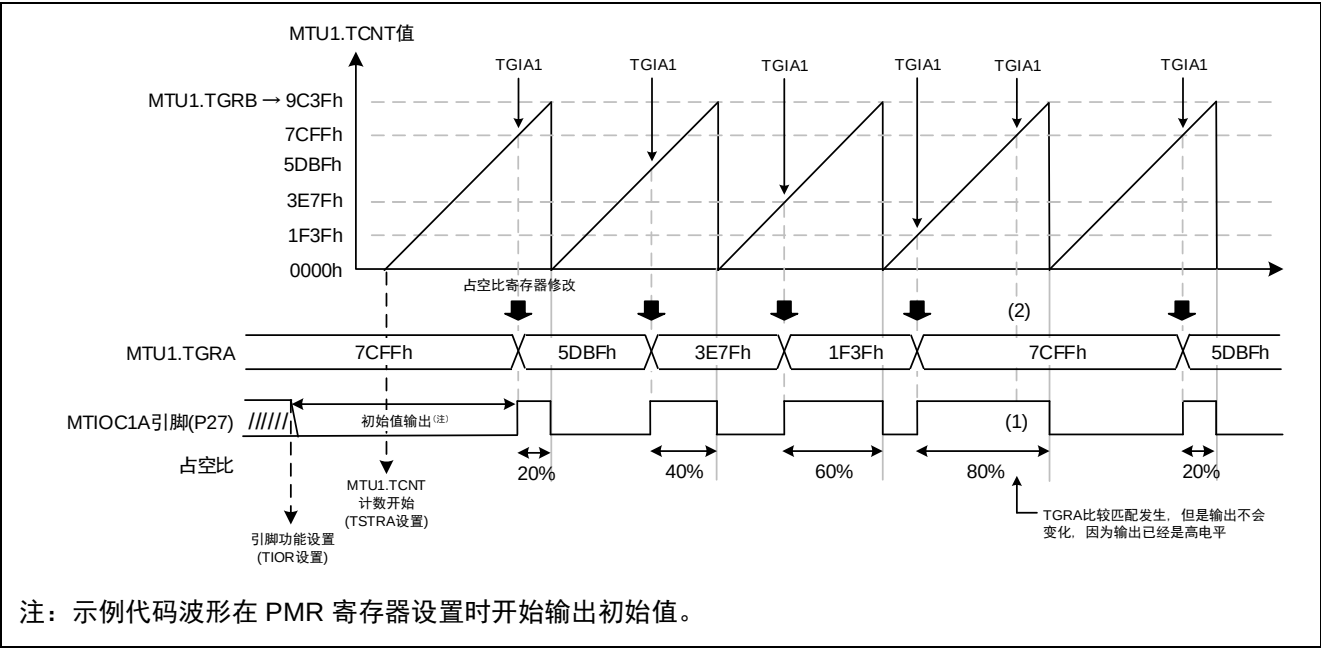


图 3.17 示例代码运行

3.4.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考章节 3.1.4 添加组件。

表 3.7 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU1
运行	PWM 模式 1
资源	MTU1

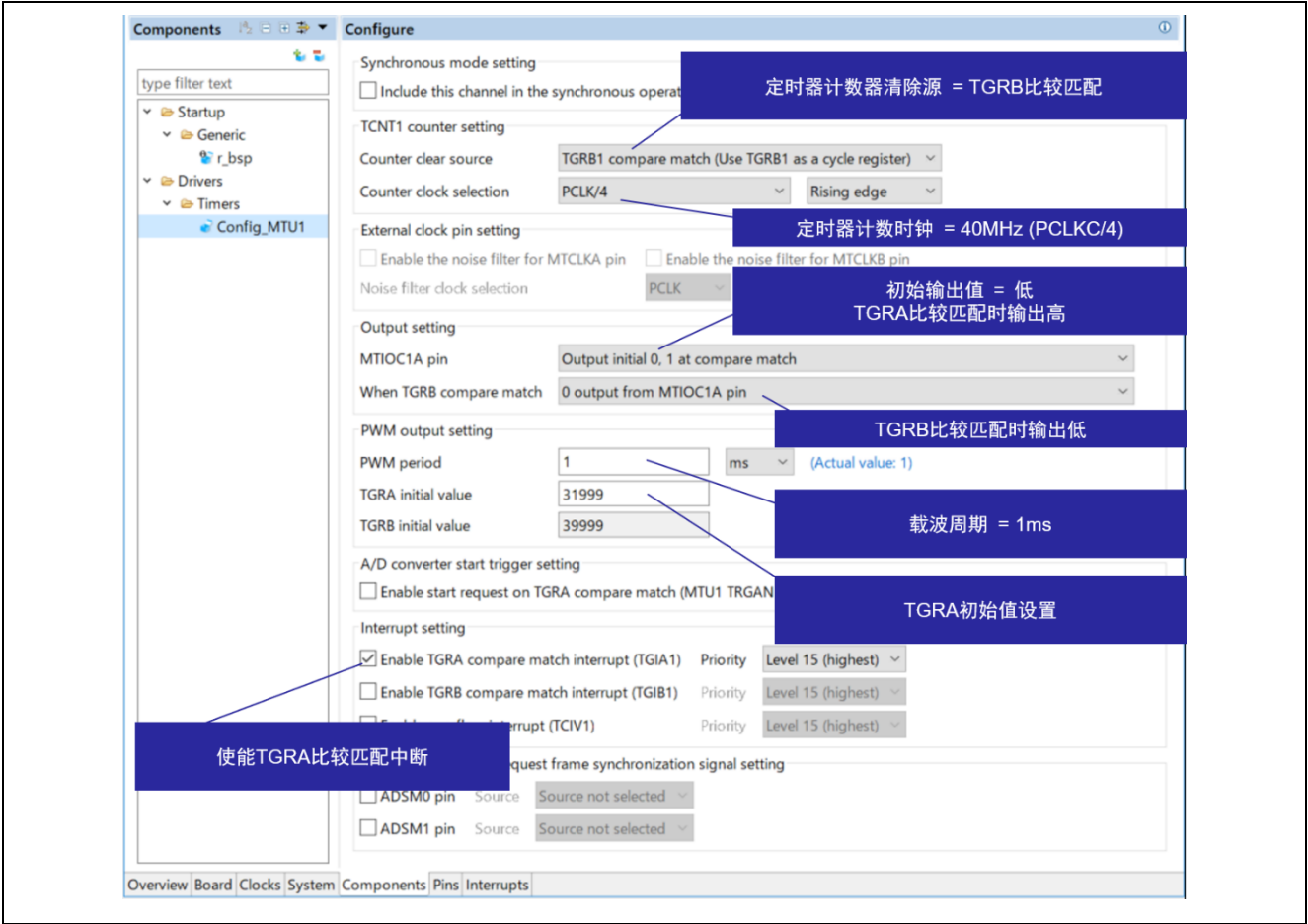


图 3.18 MTU1 设置

3.4.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。

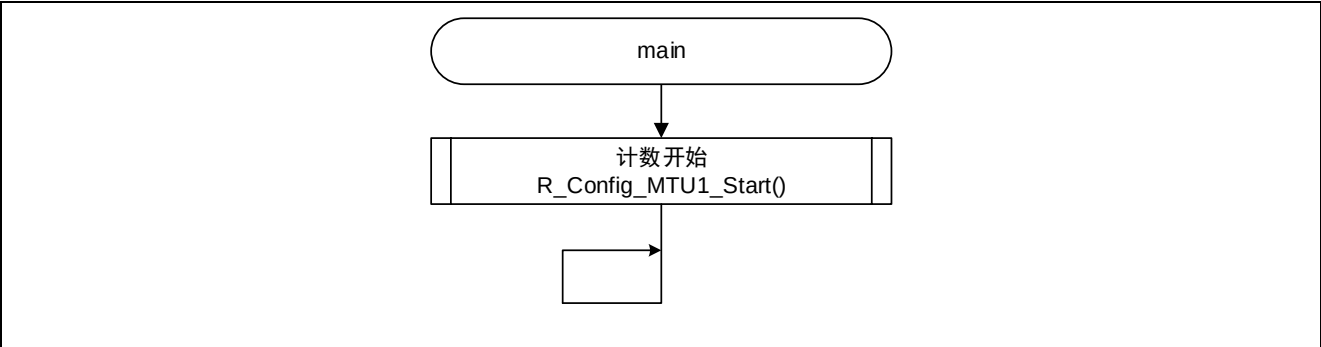


图 3.19 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU1_Create_UserInit 中初始化变量。此函数在 R_Config_MTU1_Create 函数中调用。

此函数初始化示例代码中使用的以下变量。

- s_duty_prv: 用于保存上一次 TGRA 寄存器值的变量。

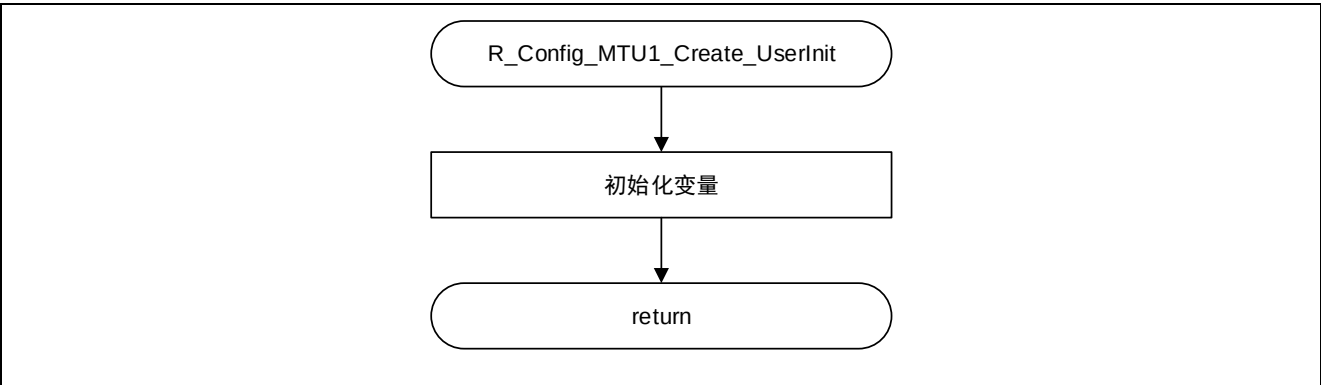


图 3.20 用户初始化函数

TGIA1 中断处理函数根据 TGRA 寄存器的当前值和上一次 TGRA 寄存器中的设置值来改变 TGRA 寄存器的值。

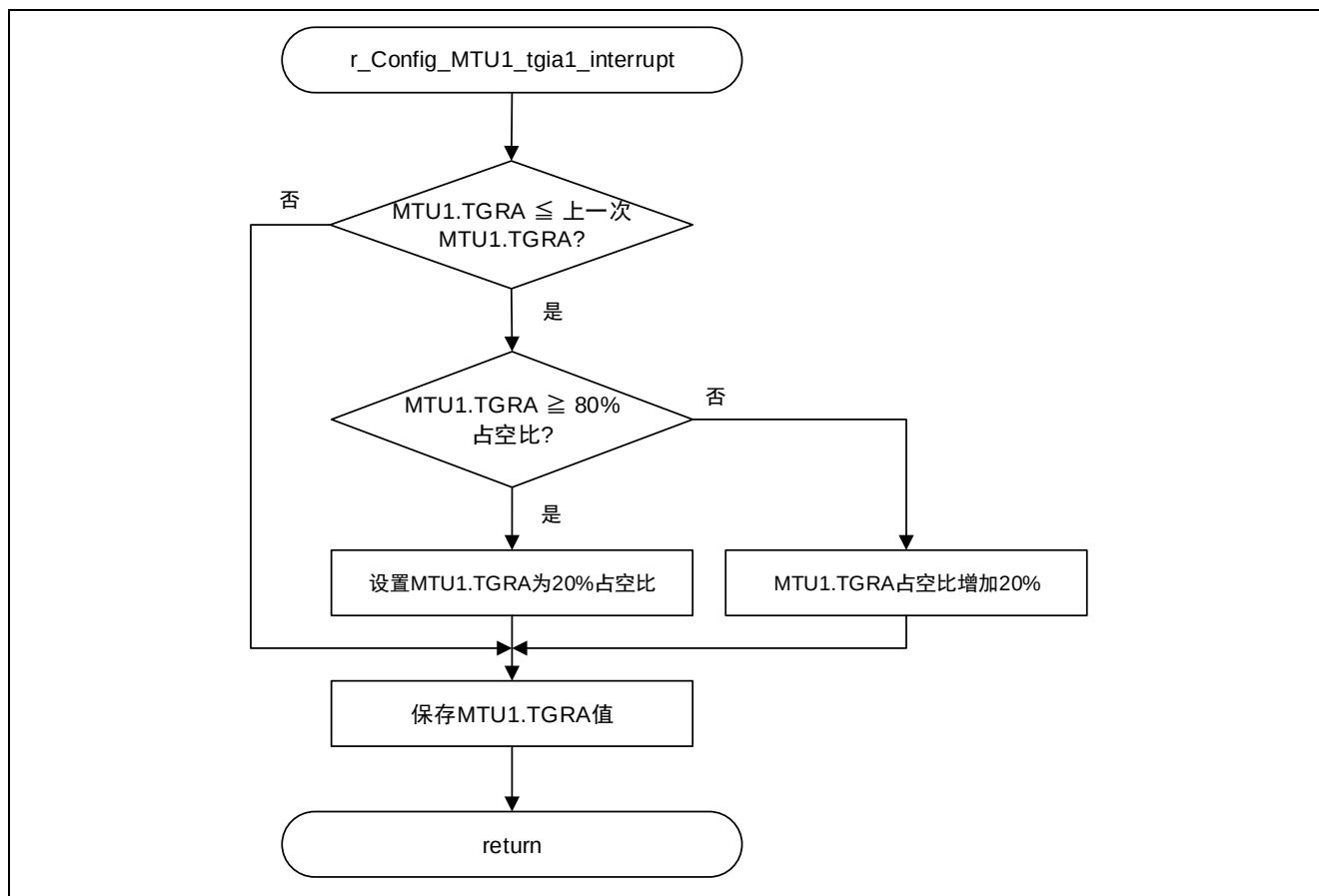


图 3.21 TGIA1 中断处理函数

3.4.5 使用时的注意事项

3.4.5.1 TGR 寄存器写入和比较匹配之间的竞争

如果在 TGR 寄存器写周期中发生比较匹配，则执行 TGR 的写操作，同时也会产生比较匹配信号。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.6 TGR 写操作和比较匹配之间的竞争。

3.4.5.2 设置一个大于占空比寄存器的值

在此示例代码中，占空比寄存器 TGRA 在产生 MTU1.TGRA 比较匹配中断（TGIA1）时被修改。

在 TGRA 比较匹配发生后，如果在 TGRA 中设置一个大于当前设置在 TGRA 的值，则一个周期内可能会发生两次 TGRA 比较匹配。

详情参考图 3.17 中（1）。

3.5 PWM 模式 2 比较匹配

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm2_cmp.zip

3.5.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 2 并重复输出占空比为 20% → 40% → 60% → 80% → 20% → ...的波形。当占空比寄存器 TGRA 和 TGRB 之间发生比较匹配时, 使用缓冲寄存器 TGRC 和 TGRD 将 TGRC 和 TGRD 的值传送到占空比寄存器 TGRA 和 TGRB 来改变占空比。由于 MTU 通道 0 的 TGRA 到 TGRD 寄存器用于占空比寄存器和缓冲寄存器, 因此使用通道 1 的 TGRA 寄存器作为周期寄存器。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 2
- 使用通道 0 和 1
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 MTU1.TGRA 作为周期寄存器
 - 定时器计数器清除源 = MTU1.TGRA 比较匹配
 - 在 MTU1.TGRA 比较匹配时输出低
- 使用 MTU0.TGRA 作为占空比寄存器
 - 在 MTU0.TGRA 比较匹配时输出高
- 使用 MTU0.TGRB 作为占空比寄存器
 - 在 MTU0.TGRB 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 MTU0.TGRC 作为 MTU0.TGRA 的缓冲寄存器
 - 使用 MTU0.TGRD 作为 MTU0.TGRB 的缓冲寄存器
 - 在比较匹配发生时缓冲传送
- 每个周期的占空比改变
 - 在 MTU1.TGRA 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 3.23

在 Smart Configurator 中设置。
设置方法请参考章节 3.5.3。

此示例代码的 PWM 模式 2 输出如下所示。

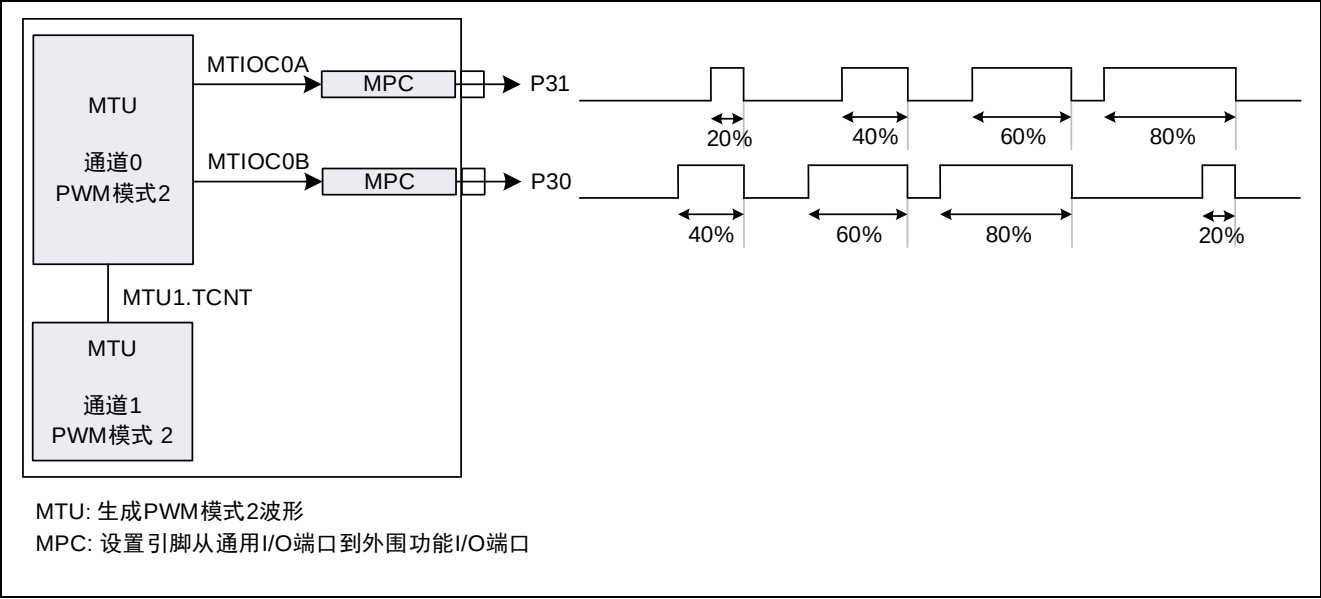


图 3.22 PWM 模式 2 输出

3.5.2 运行说明

示例代码的运行如图 3.23 所示。通过在周期寄存器 MTU1.TGRA 的比较匹配中断（TGIA1）中修改缓冲寄存器 MTU0.TGRC 和 MTU0.TGRD 的值，可以在每个周期改变占空比的设置。比较匹配发生时，MTU0.TGRC 和 MTU0.TGRD 的值将传送到占空比寄存器 MTU0.TGRA 和 MTU0.TGRB。

当占空比从 80%切换到 20%时，同一周期内发生两次 TGRA 比较匹配，但波形不会改变，因为第二个比较匹配发生在输出高期间（下图中（1））。

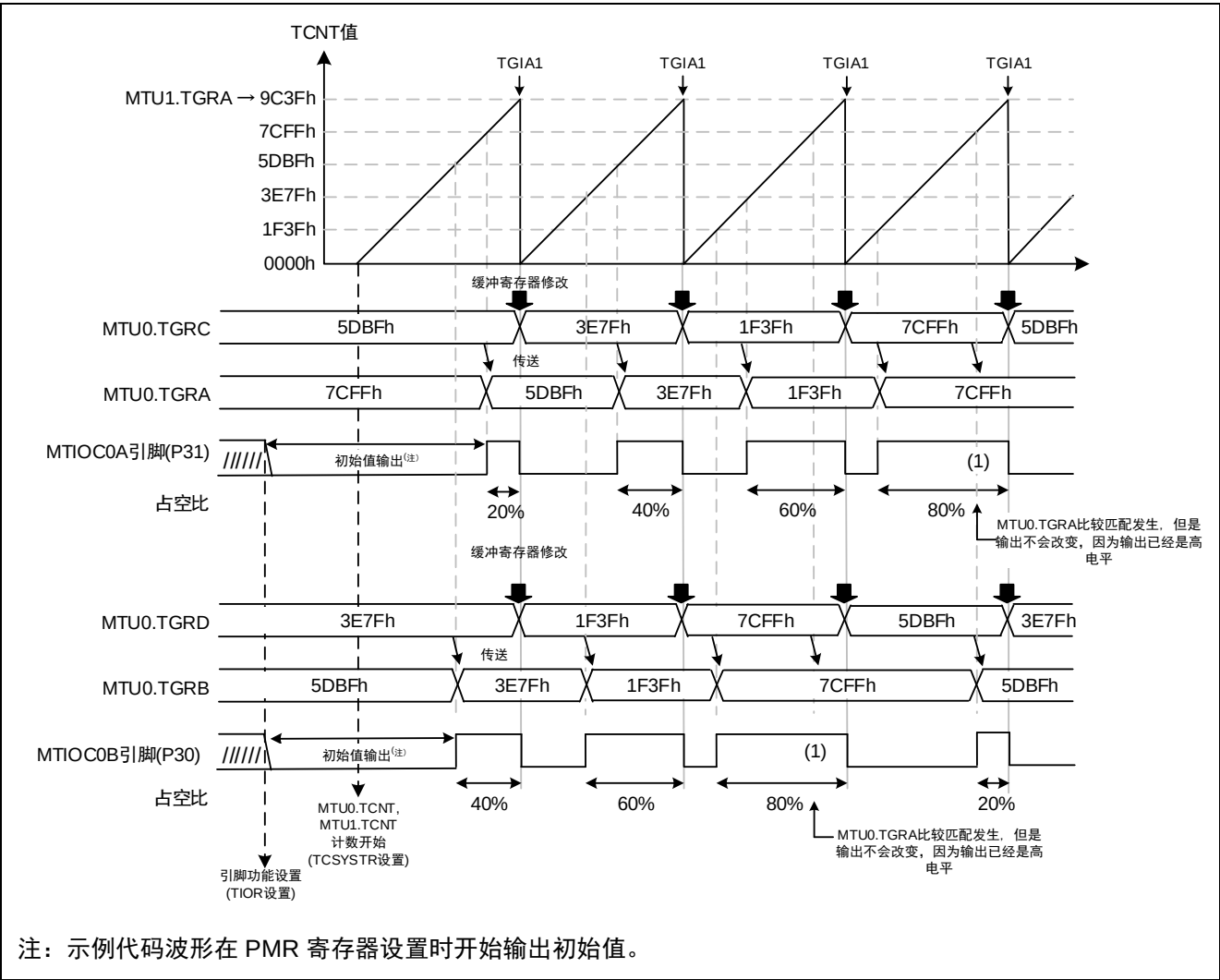


图 3.23 示例代码运行

3.5.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.8 添加组件

项目	描述	
组件	PWM 模式定时器	
配置名称	Config_MTU0	Config_MTU1
运行	PWM 模式 2	
资源	MTU0	MTU1

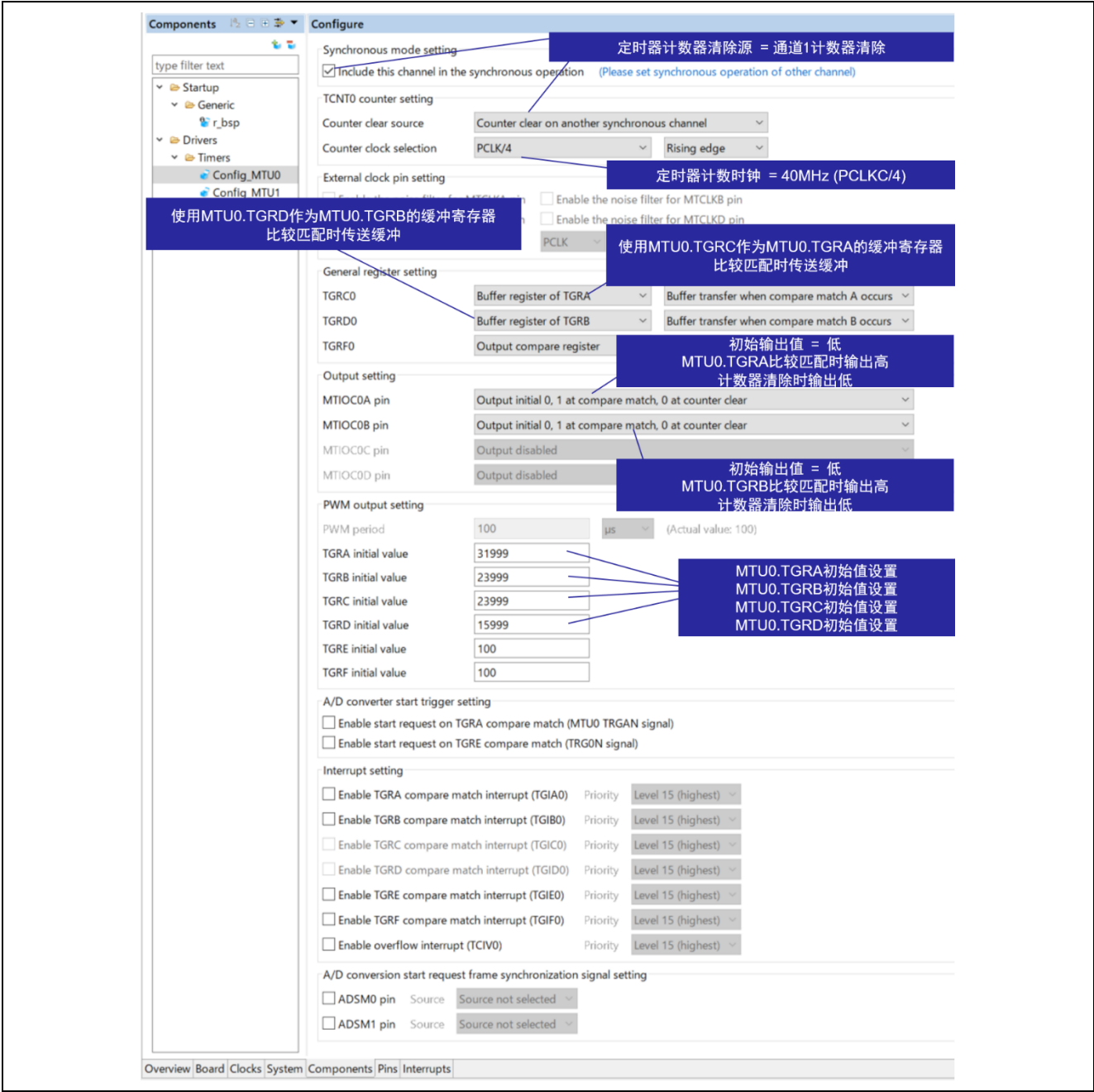


图 3.24 MTU0 设置

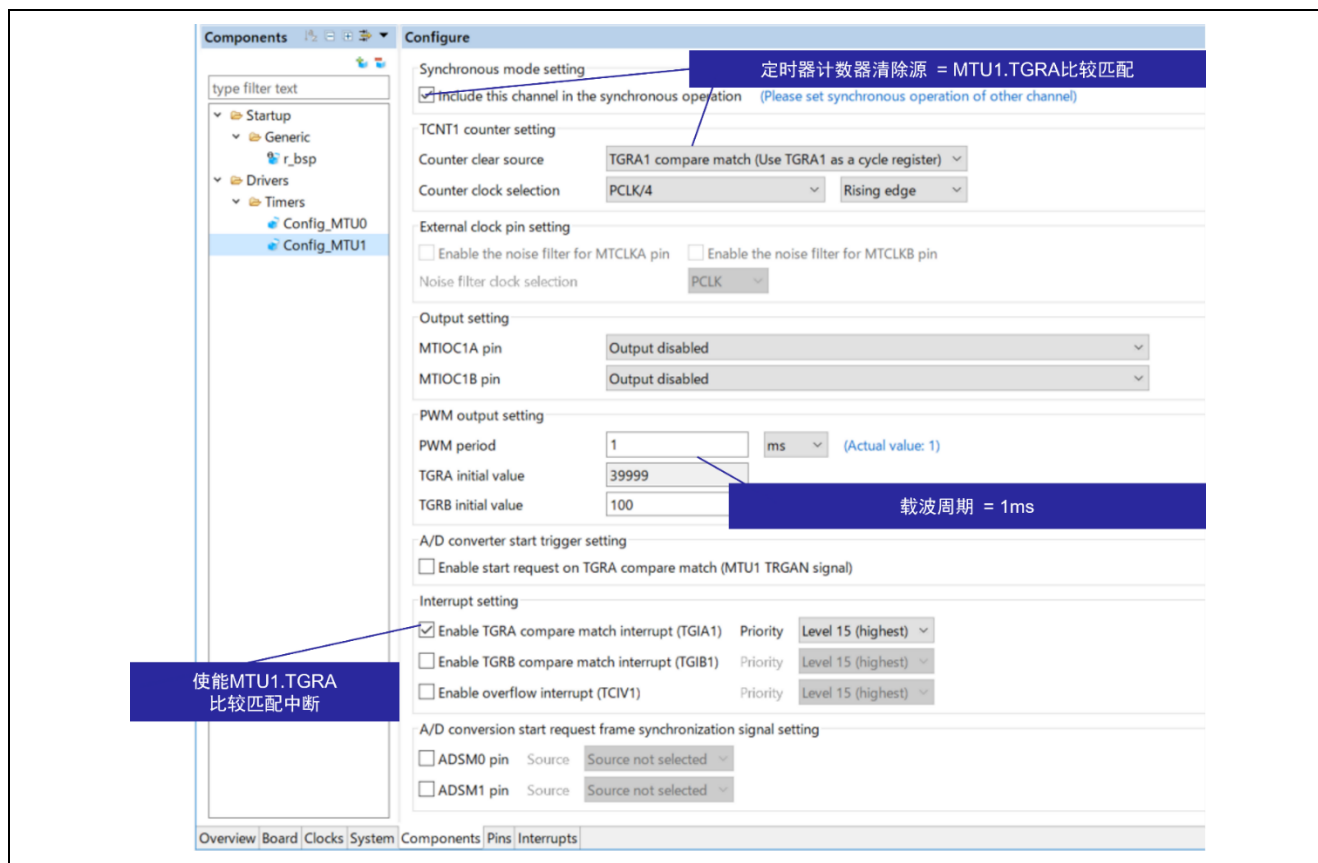


图 3.25 MTU1 设置

3.5.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 `mtu0_mtu1_start` 并开始计数。

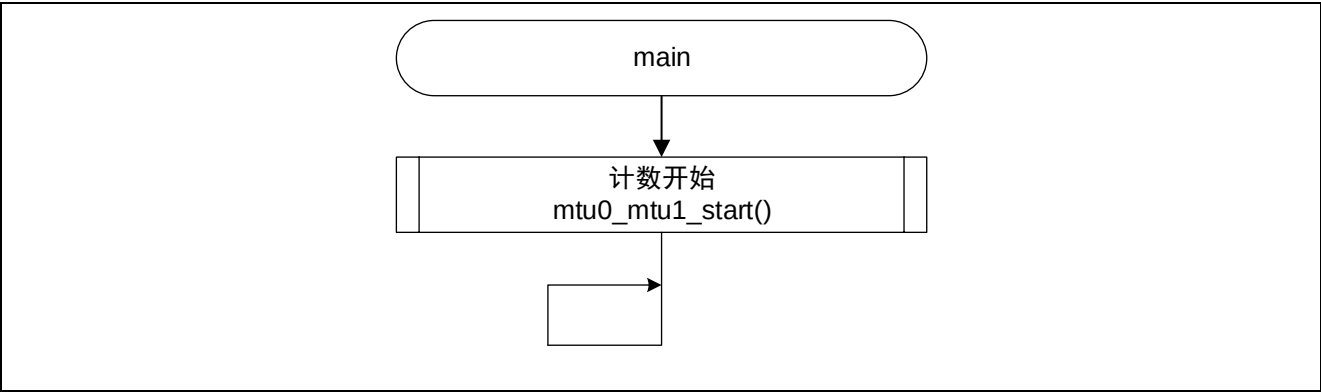


图 3.26 主函数

在计数开始函数中，MTU0 和 MTU1 计数在 TGIA1 中断使能后启动。
此函数是在 Smart Configurator 生成代码后新创建的。

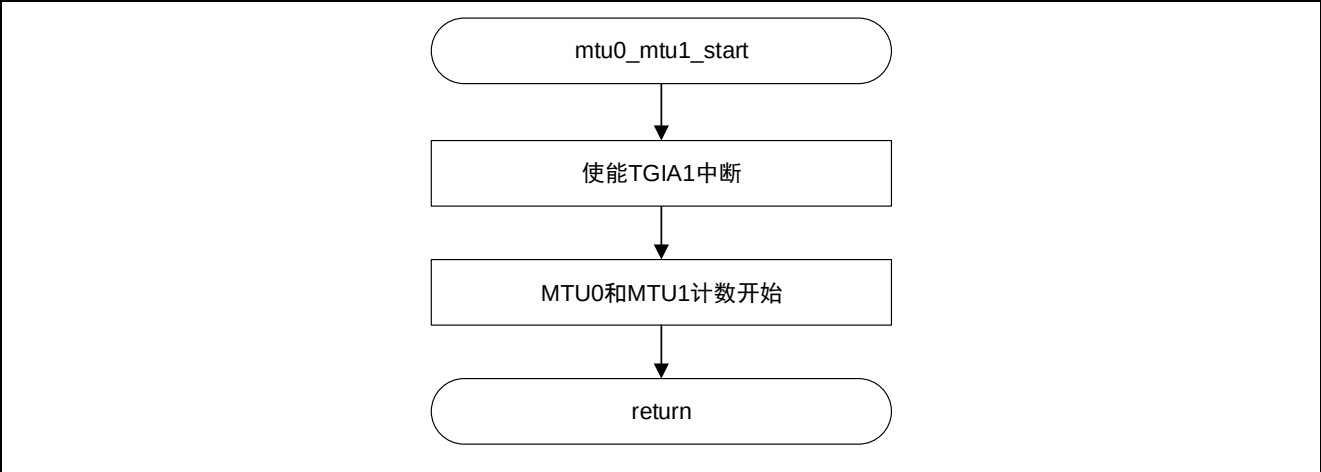


图 3.27 计数开始函数

TGIA1 中断处理函数根据 MTU0.TGRC 和 MTU0.TGRD 的当前值来改变占空比。

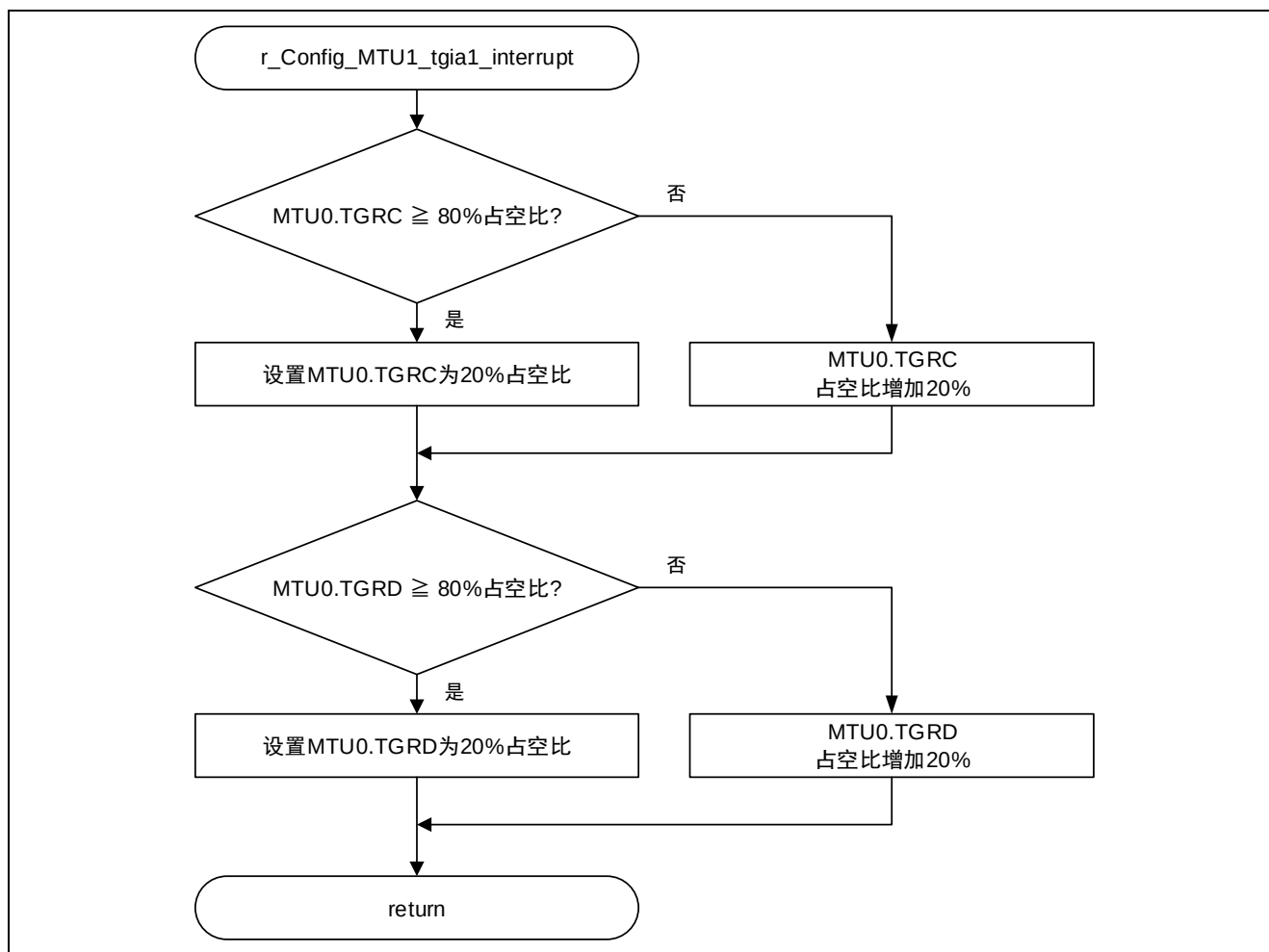


图 3.28 TGIA1 中断处理函数

3.5.5 使用时的注意事项

3.5.5.1 多通道计数开始

在此示例代码中，在 `mtu0_mtu1_start` 函数中同时设置了定时器计数器同步启动寄存器 `TCSYSTR` 的 `SCH0` 和 `SCH1` 位，以同时启动 `MTU0` 和 `MTU1` 计数。

当使用 Smart Configurator 生成的 `R_Config_MTU0_Start` 和 `R_Config_MTU1_Start` 函数时，因为每个函数被分别调用，计数开始的时间可能不相同。通过同时设置 `TSTRA` 定时器启动寄存器的 `CST0` 和 `CST1` 位，可以同时启动 `MTU0` 和 `MTU1` 计数。

详情参考 *RX66T Group User's Manual: Hardware*, 章节 22.2.17 定时器启动寄存器（`TSTRA`, `TSTRB`, `TSTR`）和章节 22.2.19 定时器计数器同步启动寄存器（`TCSYSTR`）。

3.5.5.2 缓冲寄存器修改运行和比较匹配之间的竞争

当在比较匹配时设置缓冲传送时间，如果在缓冲寄存器写周期中发生比较匹配，则修改前的缓冲寄存器中的数据将传送到占空比寄存器。

详情参考 *RX66T Group User's Manual: Hardware*, 章节 22.6.7 缓冲寄存器写操作和比较匹配之间的竞争。

3.5.5.3 缓冲寄存器修改延迟

在此示例代码中，当 `TGRA` (`TGRB`)比较匹配发生时，将执行从缓冲寄存器 `TGRC` (`TGRD`)到 `TGRA` (`TGRB`)的传送。必须在下一次比较匹配之前完成 `TGRC` (`TGRD`)值的修改。

如果缓冲寄存器修改延迟，则到下一个周期中 `TGRA` (`TGRB`)比较匹配发生时，`TGRA` (`TGRB`)值可能不会更新。在这种情况下，请考虑在计数器清除发生时传送缓冲。

有关 PWM 模式 2 在计数器清除时执行缓冲传送的示例代码，请参考 3.6 PWM 模式 2 计数清除。

3.5.5.4 设置一个大于占空比寄存器的值

在此示例代码中，当 `TGRA` 比较匹配发生时，将执行从缓冲寄存器 `TGRC` 到 `TGRA` 的传送。

在 `TGRA` 比较匹配发生后，如果在 `TGRA` 中设置了一个大于当前设置在 `TGRA` 中的值，则一个周期内可能会发生两次 `TGRA` 比较匹配。

有关详细信息，请参考图 3.23 中 (1)。

3.6 PWM 模式 2 计数清除

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm2_cntclear.zip

3.6.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 2 并重复占空比为 20% → 40% → 60% → 80% → 20% →的波形输出。当计数器清除发生时,通过使用缓冲寄存器 TGRC 和 TGRD 将 TGRC 和 TGRD 值传送到占空比寄存器 TGRA 和 TGRB 来改变占空比。由于 MTU 通道 0 的寄存器 TGRA 到 TGRD 被用作占空比寄存器和缓冲寄存器,使用通道 1 的 TGRA 寄存器作为周期寄存器。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 2
- 使用通道 0 和 1
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 MTU1.TGRA 作为周期寄存器
 - 定时器计数器清除源 = MTU1.TGRA 比较匹配
 - MTU1.TGRA 比较匹配时输出低
- 使用 MTU0.TGRA 作为占空比寄存器
 - MTU0.TGRA 比较匹配时输出高
- 使用 MTU0.TGRB 作为占空比寄存器
 - MTU0.TGRB 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 MTU0.TGRC 作为 MTU0.TGRA 的缓冲寄存器
 - 使用 MTU0.TGRD 作为 MTU0.TGRB 的缓冲寄存器
 - 当计数器清除发生时缓冲传送
- 每个周期的占空比改变
 - 在 MTU1.TGRA 比较匹配中断时更改占空比
 - 关于占空比改变的时间的详细信息,请参考图 3.30

在 Smart Configurator 中设置。
设置方法请参考章节 3.6.3。

此示例代码的 PWM 模式 2 输出如下所示。

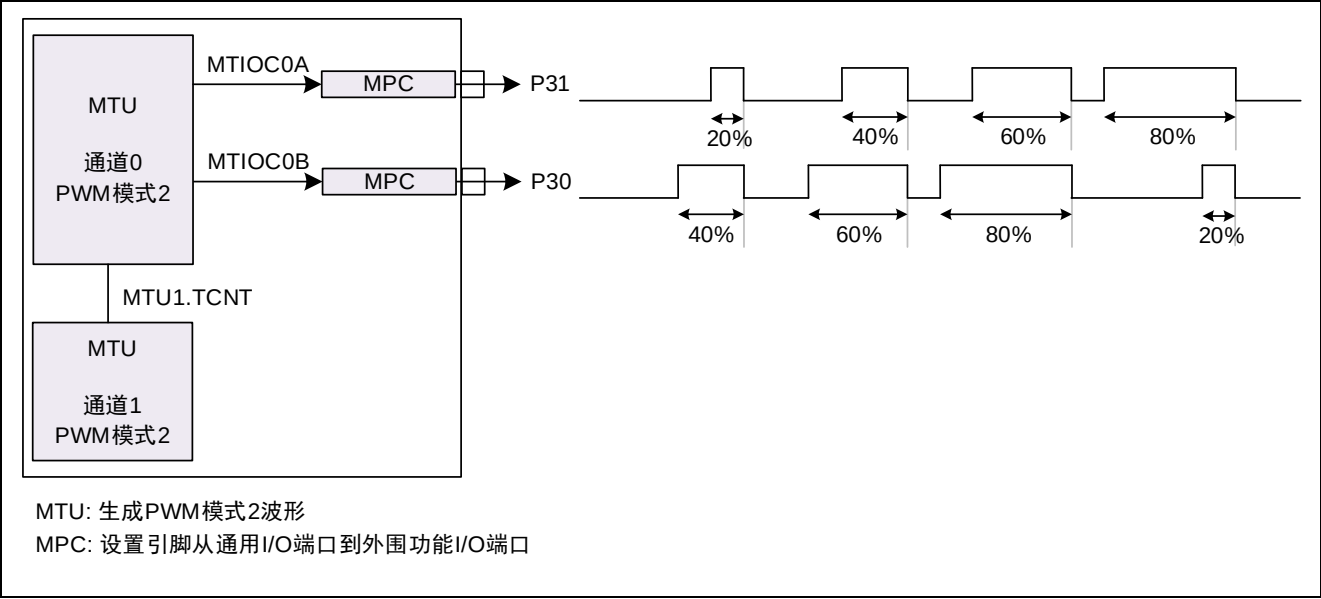


图 3.29 PWM 模式 2 输出

3.6.2 运行说明

示例代码的运行如图 3.30 所示。通过在周期寄存器 MTU1.TGRA 的比较匹配中断(TGIA1)处修改缓冲寄存器 MTU0.TGRC 和 MTU0.TGRD 的值，可以在每个周期改变占空比的设置。当发生计数器清除时，MTU0.TGRC 和 MTU0.TGRD 的值被传送到占空比寄存器 MTU0.TGRA 和 MTU0.TGRB。

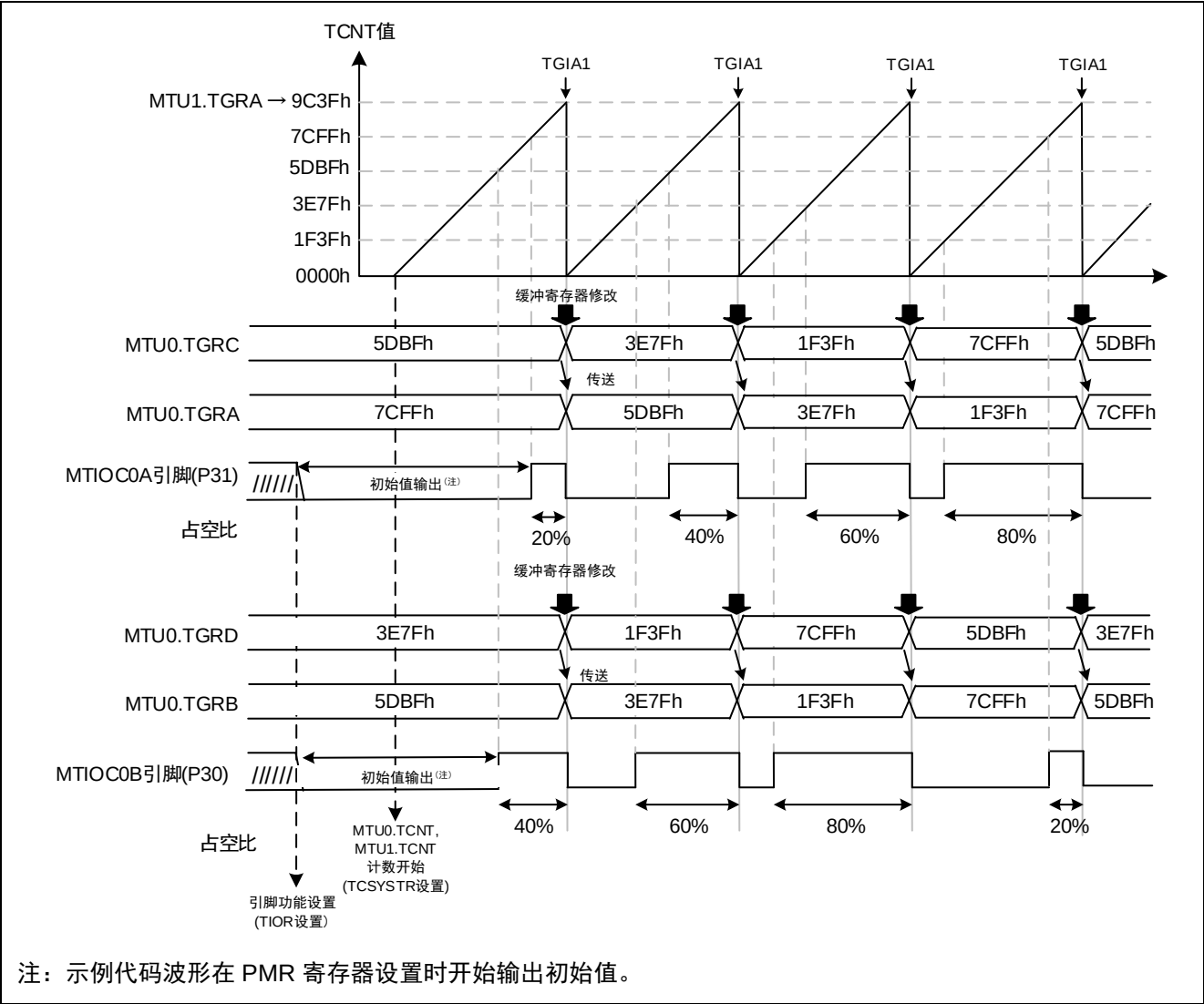


图 3.30 示例代码运行

3.6.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考章节 3.1.4 添加组件。

表 3.9 添加组件

项目	描述	
组件	PWM 模式定时器	
配置名称	Config_MTU0	Config_MTU1
运行	PWM 模式 2	
资源	MTU0	MTU1

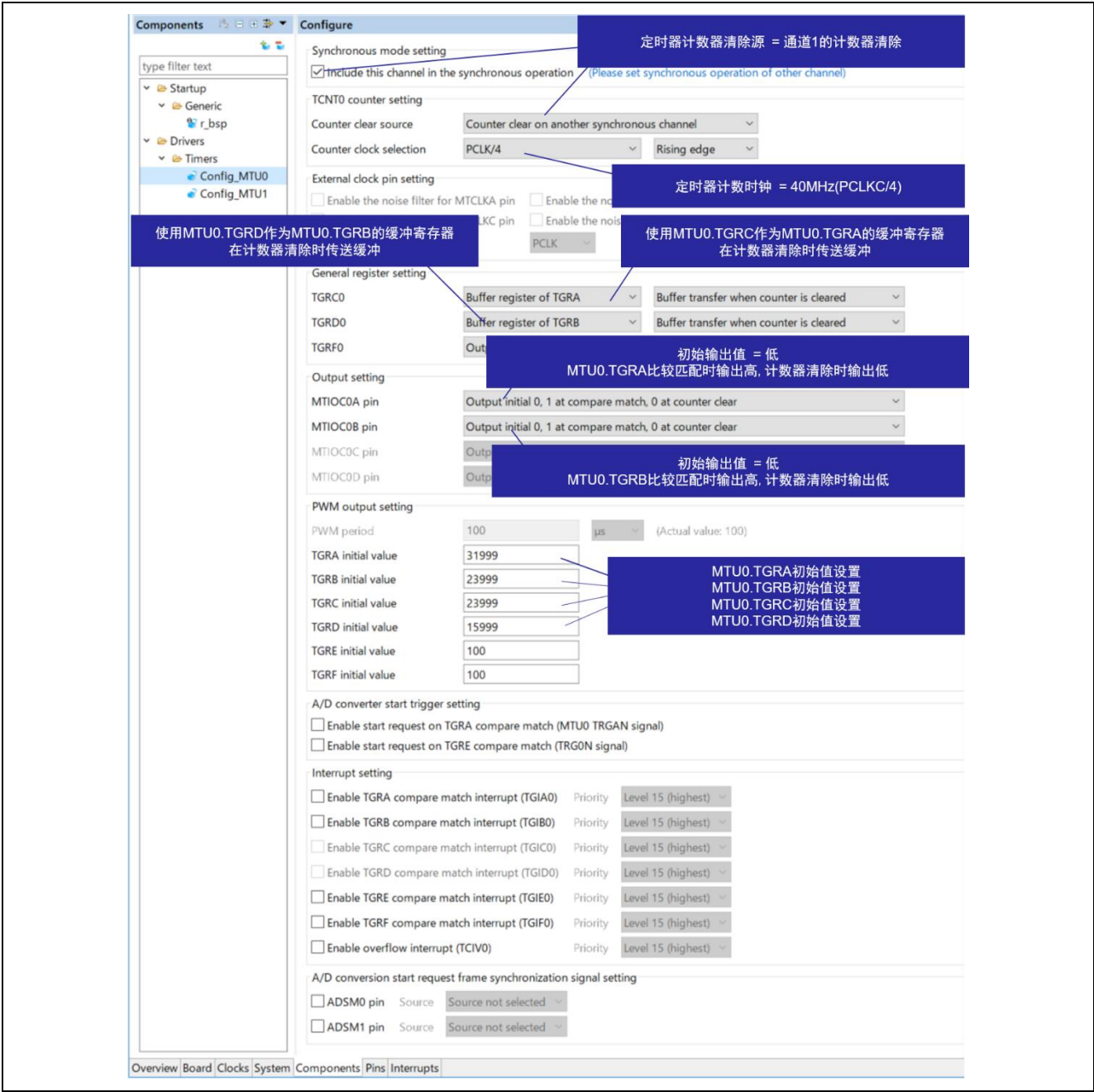


图 3.31 MTU0 设置

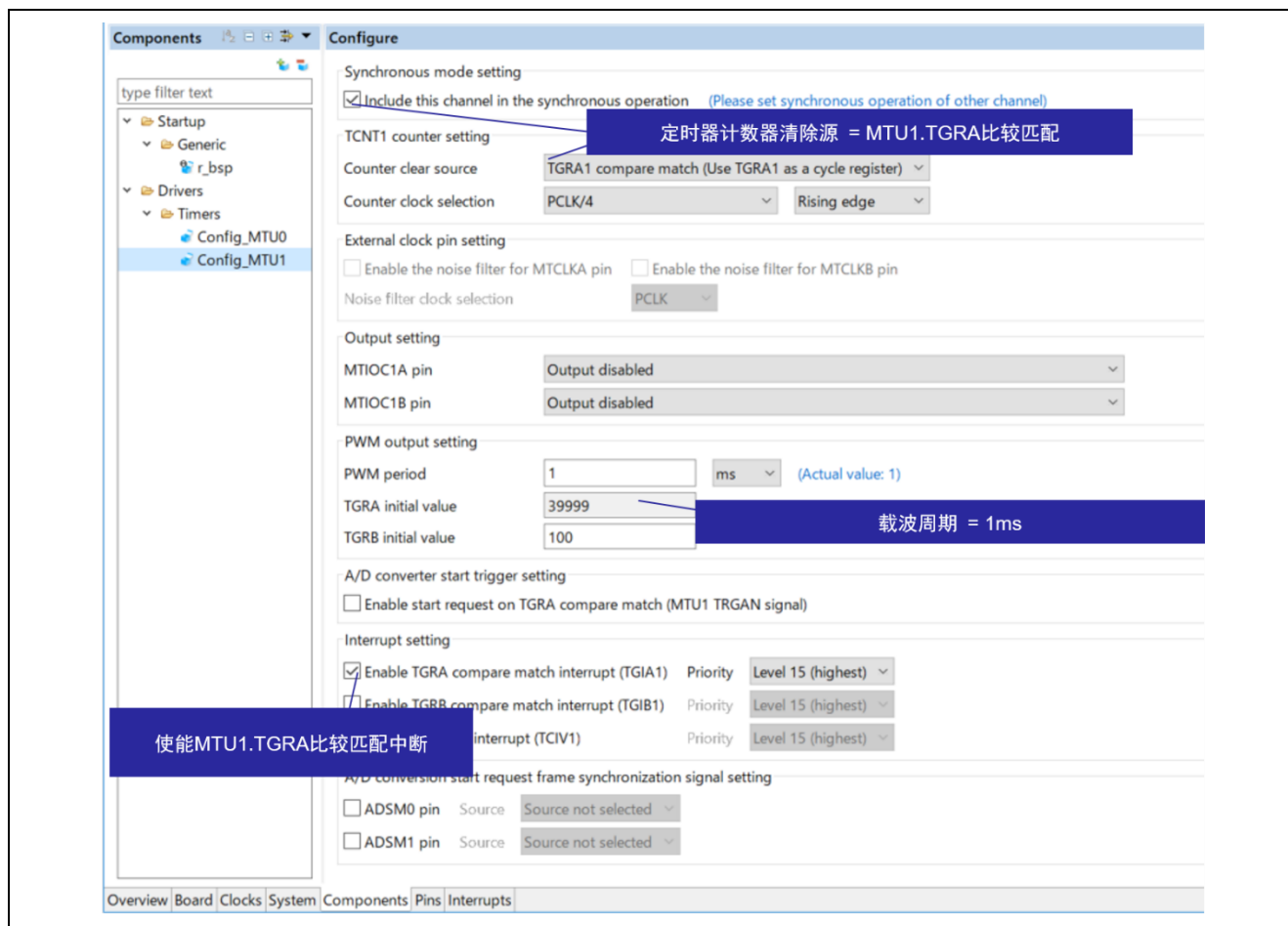


图 3.32 MTU1 设置

3.6.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 `mtu0_mtu1_start` 并开始计数。

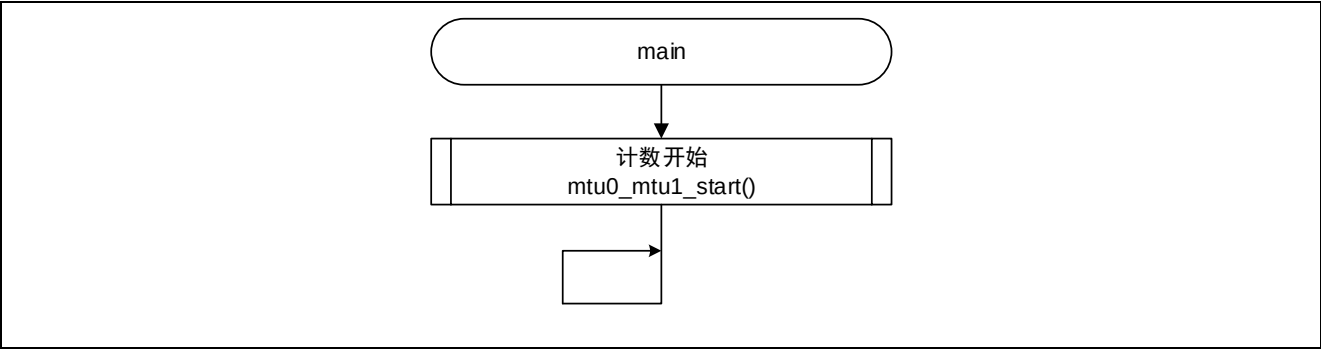


图 3.33 主函数

在计数开始函数中, MTU0 和 MTU1 在 TGIA1 中断使能后开始计数。
此函数是在 Smart Configurator 生成代码后新创建的。

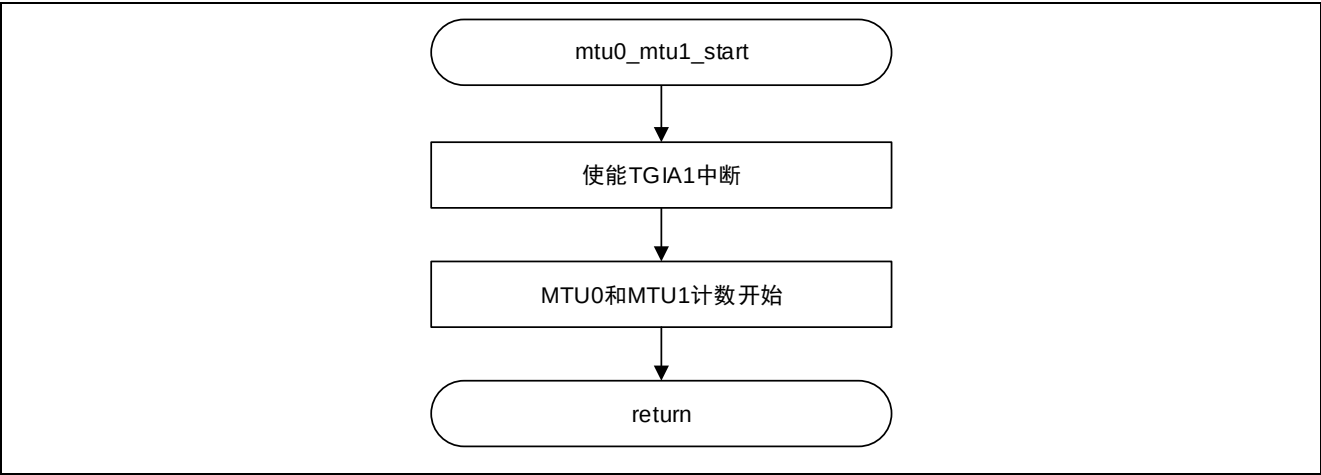


图 3.34 计数开始函数

TGIA1 中断处理函数根据 MTU0.TGRA 和 MTU0.TGRB 寄存器的当前值来改变数值。

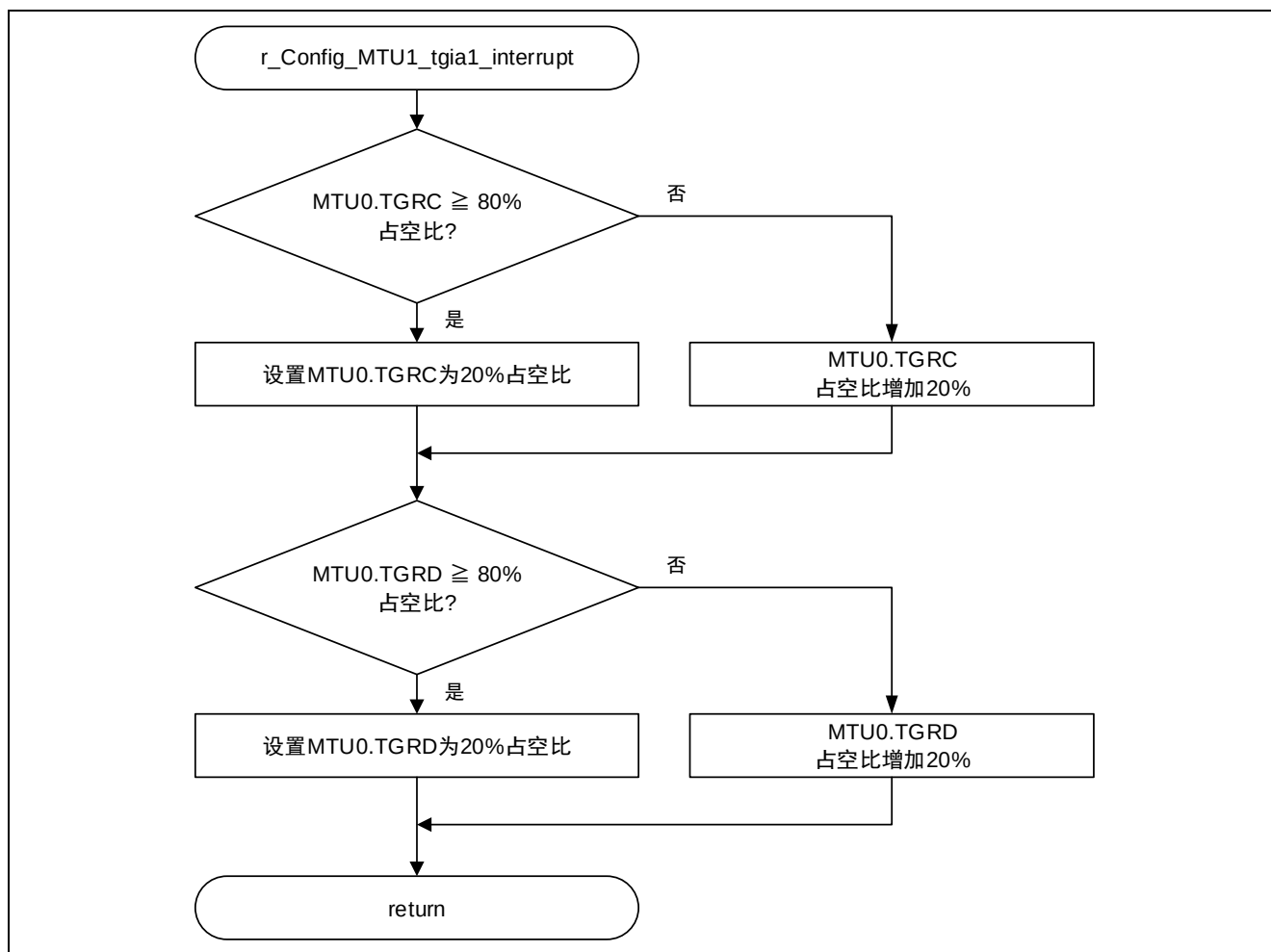


图 3.35 TGIA1 中断处理函数

3.6.5 使用时的注意事项

3.6.5.1 多通道计数开始

在此示例代码中，在 `mtu0_mtu1_start` 函数中同时设置了定时器的计数器同步启动寄存器 `TCSYSTR` 的 `SCH0` 和 `SCH1` 位，以同时启动 `MTU0` 和 `MTU1` 计数。

当使用 Smart Configurator 生成的 `R_Config_MTU0_Start` 和 `R_Config_MTU1_Start` 函数时，因为每个函数被分别调用，计数开始的时间可能不相同。通过同时设置 `TSTRA` 定时器启动寄存器 `CST0` 和 `CST1` 位，可以同时启动 `MTU0` 和 `MTU1` 计数。

详情参考 *RX66T Group User's Manual: Hardware*，章节 22.2.17 定时器启动寄存器（`TSTRA`，`TSTRB`，`TSTR`）和章节 22.2.19 定时器计数器同步启动寄存器（`TCSYSTR`）。

3.6.5.2 缓冲寄存器修改运行和计数器清除之间的竞争

如果在计数器清除时设置缓冲传送时间，在缓冲寄存器写周期中发生计数器清除，修改前的缓冲寄存器数据被传送到占空比寄存器。

详情参考 *RX66T Group User's Manual: Hardware*，章节 22.6.8 缓冲寄存器写和 `TCNT` 清除运行之间的竞争。

3.6.5.3 缓冲寄存器修改延迟

在此示例代码中，当计数器清除发生时，执行从缓冲寄存器 `TGRC`（`TGRD`）到 `TGRA`（`TGRB`）的传送。`TGRC`（`TGRD`）值的修改必须在下一个计数器清除发生前完成。

如果缓冲寄存器 `TGRC`（`TGRD`）的修改延迟，则无法输出所需的占空比。

3.7 PWM 模式 2 无缓冲寄存器

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm2.zip

3.7.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 2 并重复占空比 20% → 40% → 60% → 80% → 20% → ... 的波形输出。不使用缓冲寄存器, 通过更新占空比寄存器 TGRA 和 TGRB 的值来改变占空比。由于 MTU 通道 1 的 TGRA 和 TGRB 寄存器用作占空比寄存器和缓冲寄存器, 周期寄存器使用通道 0 的 TGRA 寄存器。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 2
- 使用通道 0 和 1
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 MTU0.TGRA 作为周期寄存器
 - 定时器计数器清除源 = MTU0.TGRA 比较匹配
 - MTU0.TGRA 比较匹配时输出低
- 使用 MTU1.TGRA 作为占空比寄存器
 - MTU1.TGRA 比较匹配时输出高
- 使用 MTU1.TGRB 作为占空比寄存器
 - MTU1.TGRB 比较匹配时输出高
- 每个周期的占空比改变
 - 在 MTU1.TGRA 比较匹配中断时改变 MTU1.TGRA 占空比
 - 在 MTU1.TGRB 比较匹配中断时改变 MTU1.TGRB 占空比
 - 关于占空比改变的时间的详细信息, 请参考图 3.37

在 Smart Configurator 中设置。
设置方法请参考章节 3.7.3。

此示例代码的 PWM 模式 2 输出如下所示。

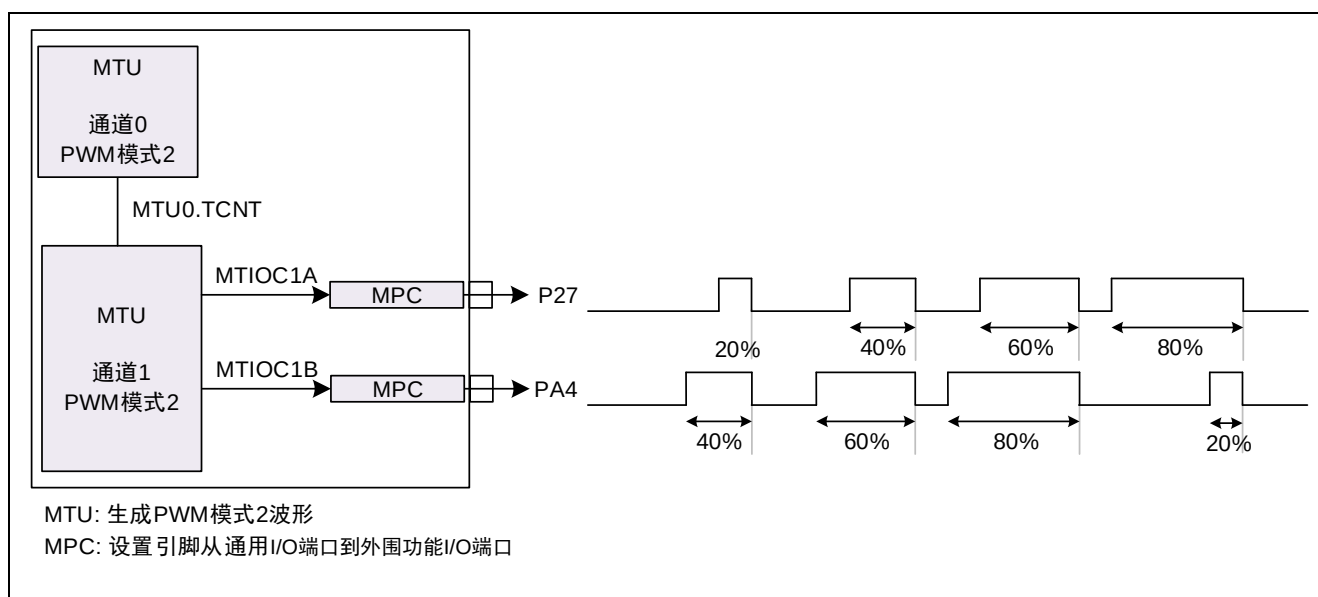


图 3.36 PWM 模式 2 输出

3.7.2 运行说明

示例代码的运行如图 3.37 所示。通过在 MTU1.TGRA 比较匹配中断（TGIA1）中修改占空比寄存器 MTU1.TGRA 的值和在 MTU1.TGRB 比较匹配中断（TGIB1）中修改占空比寄存器 MTU1.TGRB 的值，可以在每个周期改变占空比的设置。

当占空比从 80%切换到 20%时，在同一个周期发生两次比较匹配，但是波形没有变化，因为第二次比较匹配发生在高电平输出期间（下图中（1））。占空比设置随着每个周期而改变，因此需要调整修改 MTU1.TGRA 和 MTU1.TGRB 值的时间（下图中（2））。

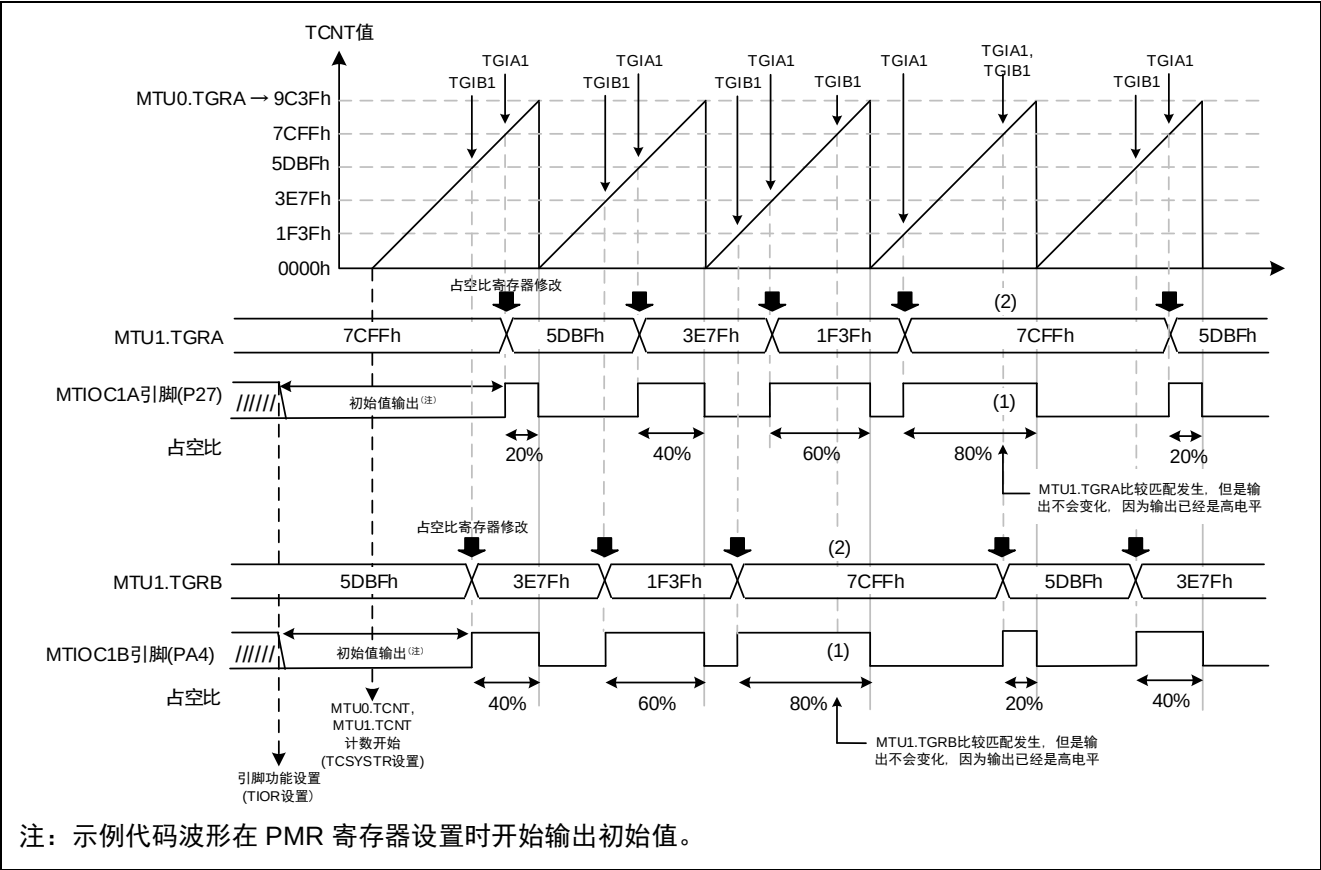


图 3.37 示例代码运行

3.7.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.10 添加组件

项目	描述	
组件	PWM 模式定时器	
配置名称	Config_MTU0	Config_MTU1
运行	PWM 模式 2	
资源	MTU0	MTU1

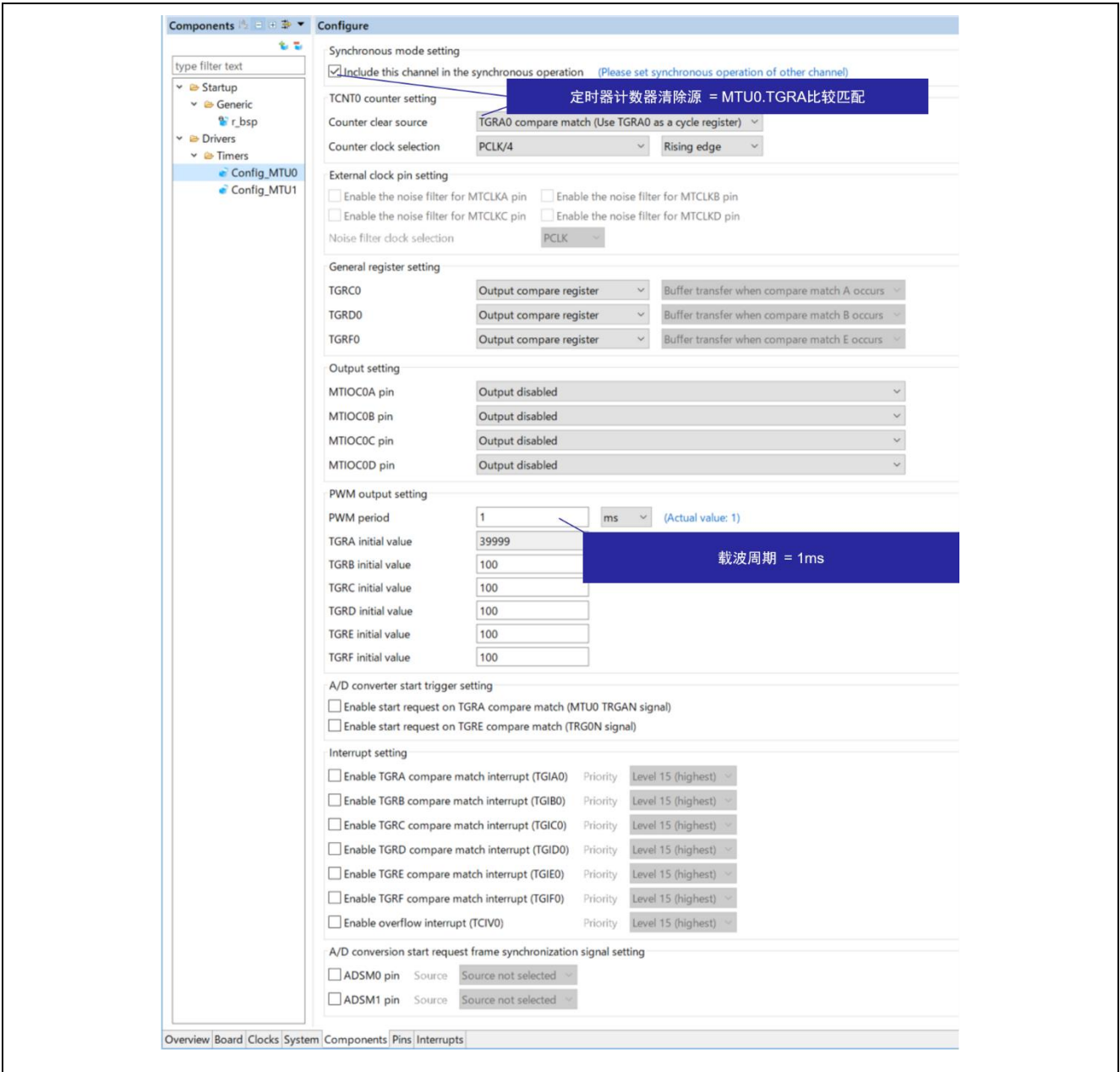


图 3.38 MTU0 设置

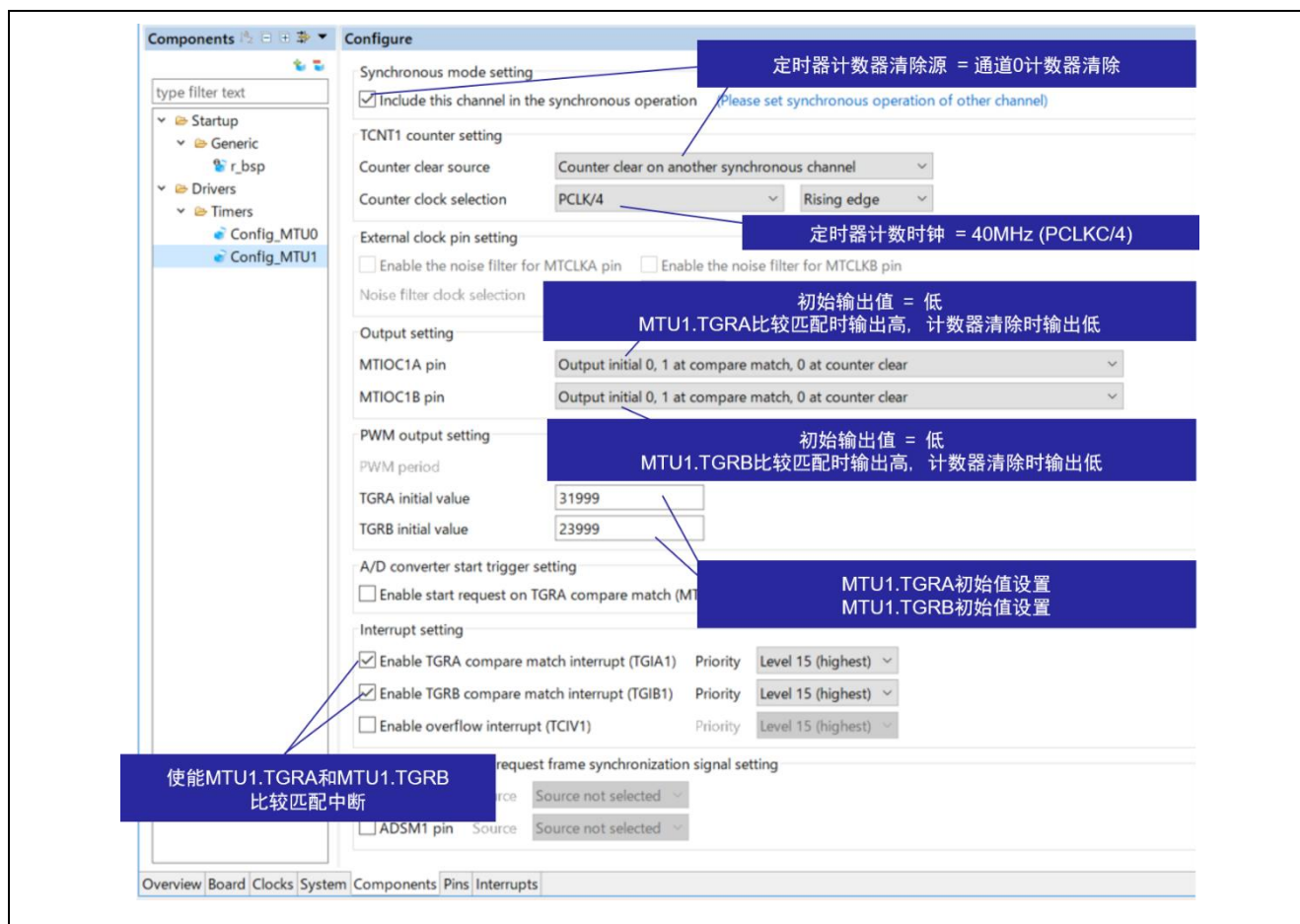


图 3.39 MTU1 设置

3.7.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。

在主函数中，调用计数开始函数 `mtu0_mtu1_start` 并开始计数。

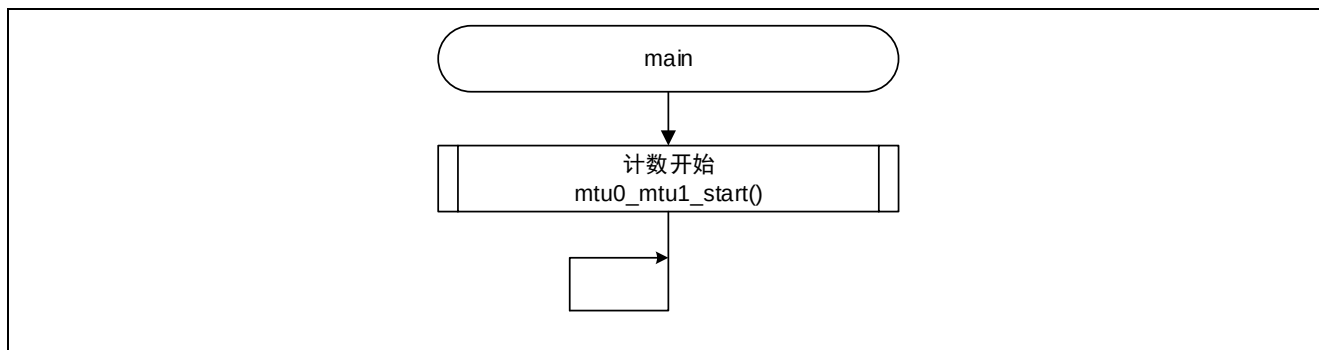


图 3.40 主函数

在计数开始函数中，在 TGIA1 和 TGIB1 中断使能后开始 MTU0 和 MTU1 计数。

此函数是在 Smart Configurator 生成代码后新创建的。

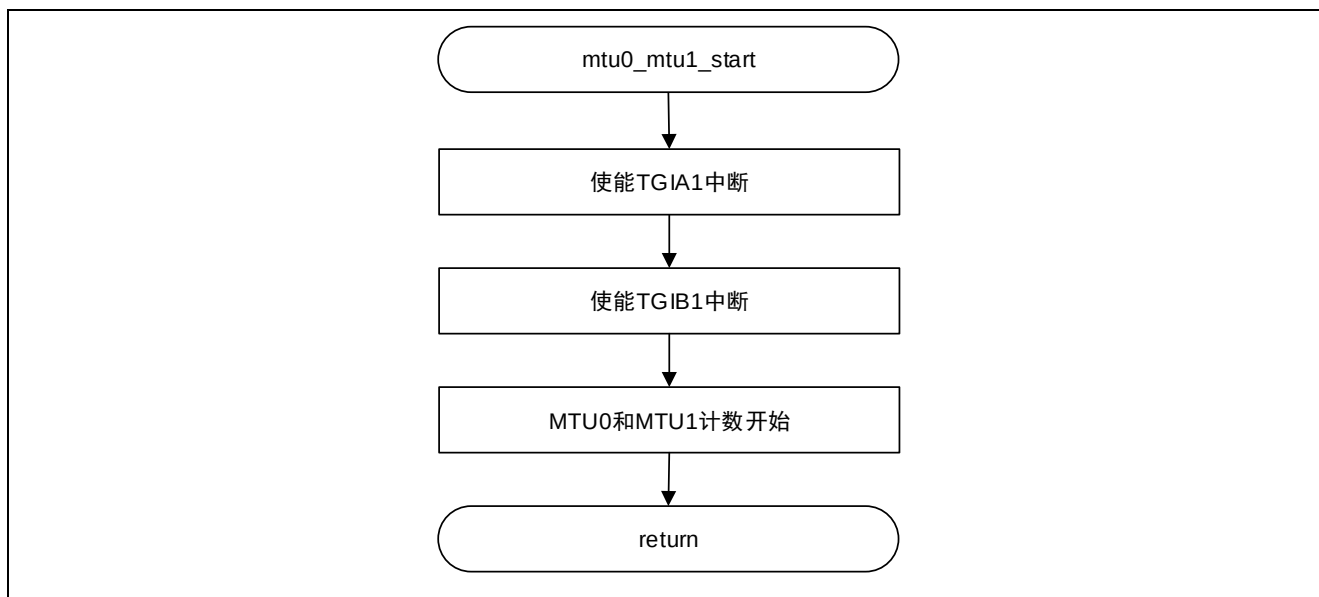


图 3.41 计数开始函数

在主函数之前执行的用户初始化函数 `R_Config_MTU1_Create_UserInit` 初始化变量。此函数在 `R_Config_MTU1_Create` 函数中调用。

此函数初始化示例代码中使用的以下变量。

- `s_duty_prv_a`: 用于保存上一次 MTU1.TGRA 寄存器值的变量
- `s_duty_prv_b`: 用于保存上一次 MTU1.TGRB 寄存器值的变量

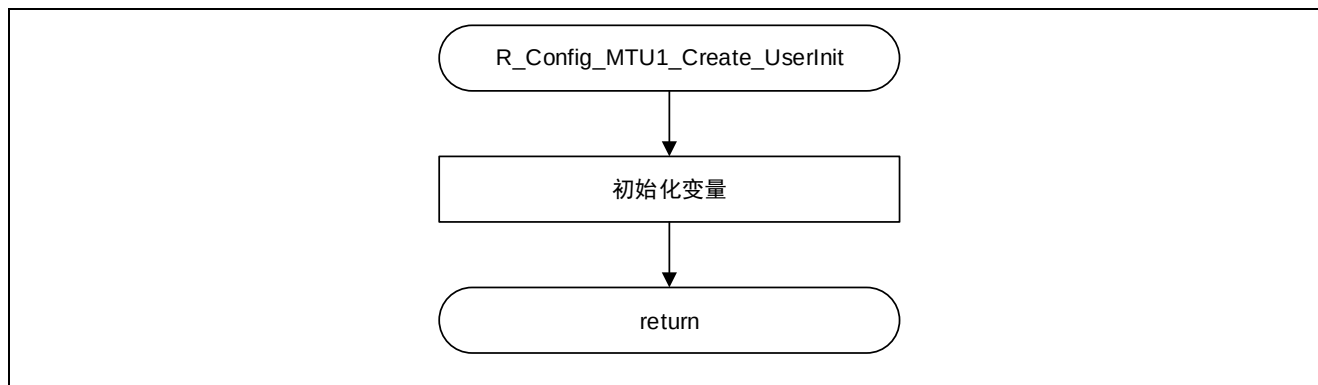


图 3.42 用户初始化函数

TGIA1 中断处理函数根据 MTU1.TGRA 寄存器的当前值和上一次 MTU1.TGRA 寄存器中的设置值来改变 MTU1.TGRA 寄存器的值。

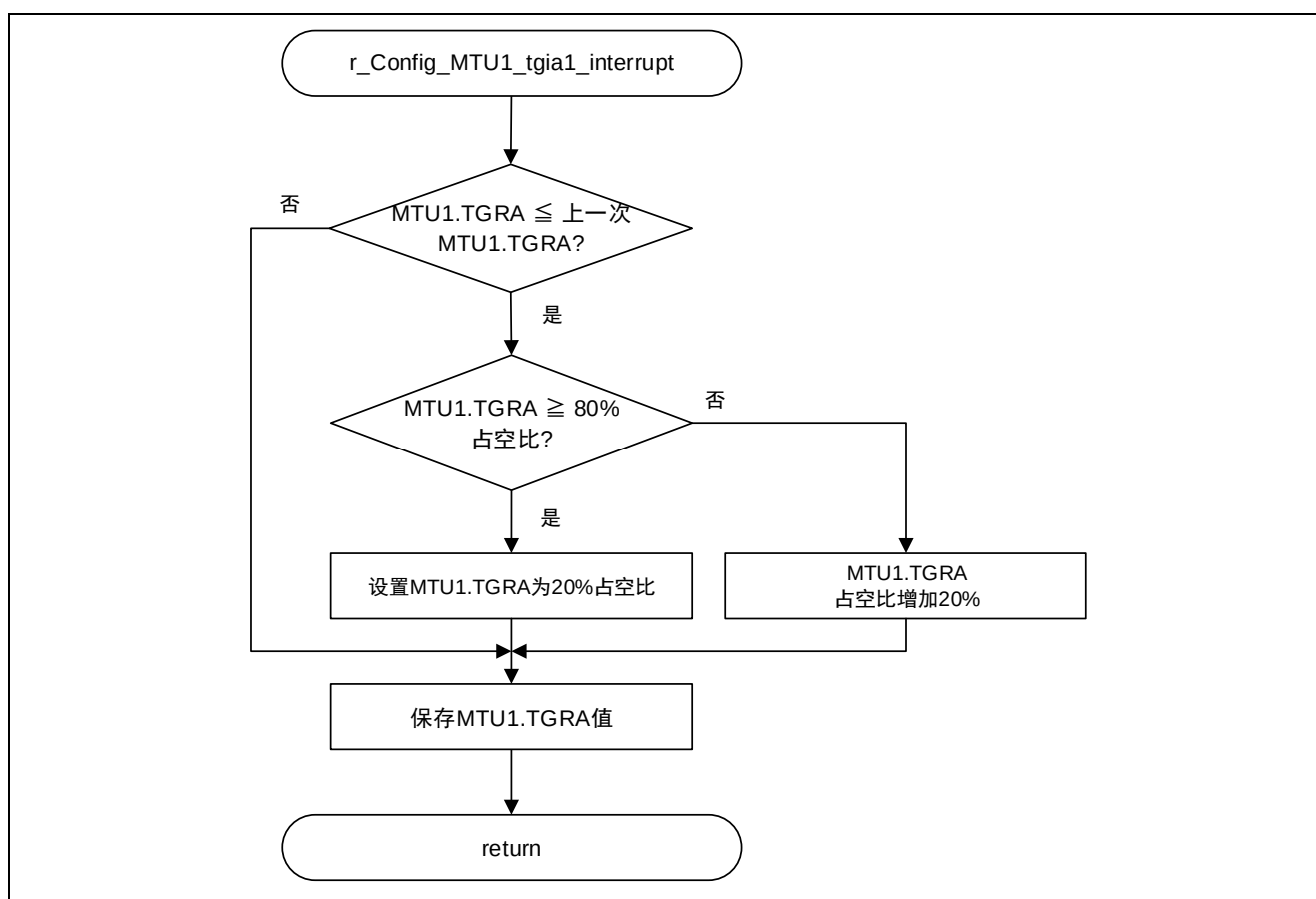


图 3.43 TGIA1 中断处理函数

TGIB1 中断处理函数根据 MTU1.TGRB 寄存器的当前值和上一次 MTU1.TGRB 寄存器中的设置值来改变 MTU1.TGRB 寄存器的值。

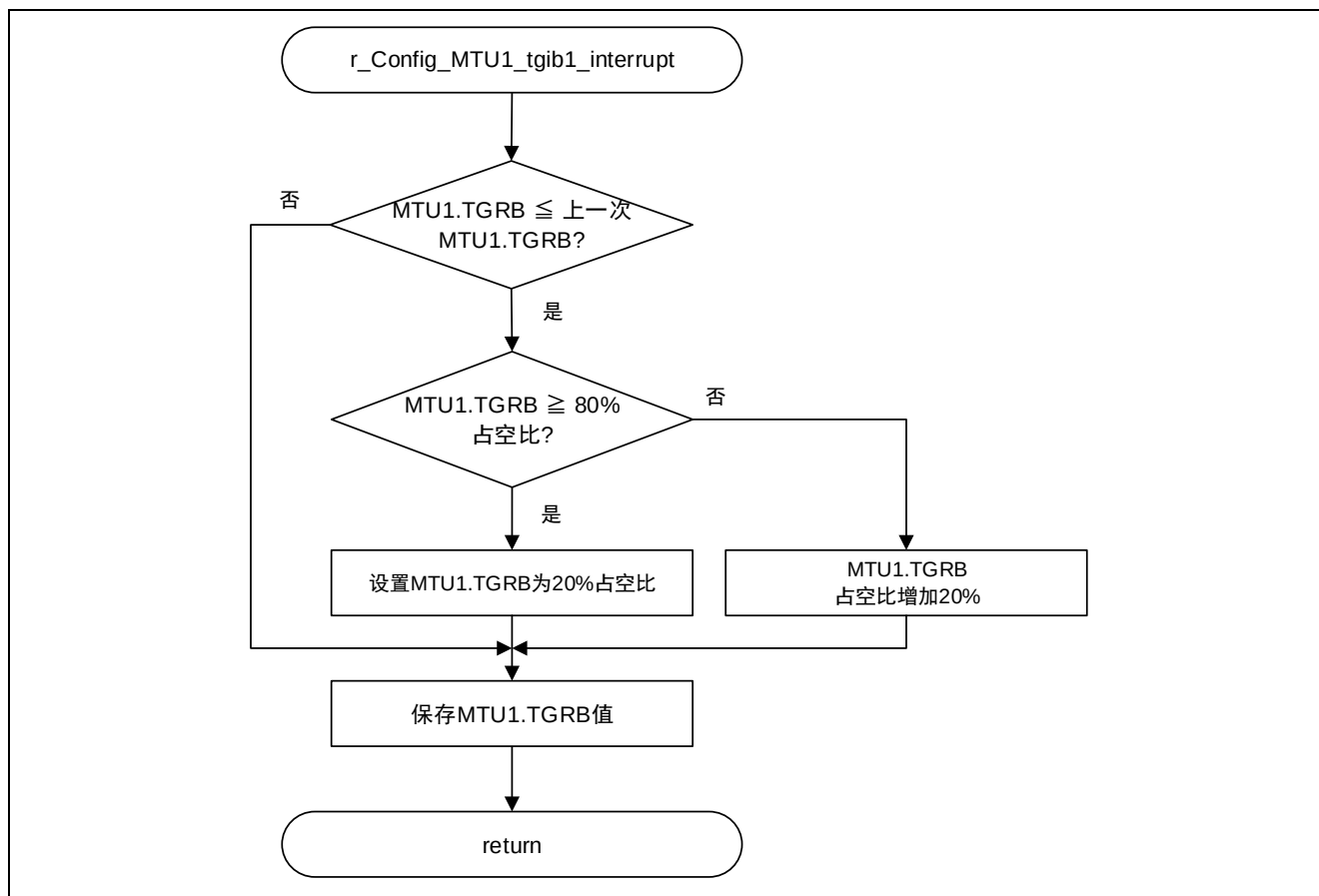


图 3.44 TGIB1 中断处理函数

3.7.5 使用时的注意事项

3.7.5.1 多通道计数开始

在本示例代码中，在 `mtu0_mtu1_start` 函数中同时设定定时器的计数器同步启动寄存器 `TCSYSTR` 的 `SCH0` 和 `SCH1` 位，以同时启动 `MTU0` 和 `MTU1` 计数。

当使用 Smart Configurator 生成的 `R_Config_MTU0_Start` 和 `R_Config_MTU1_Start` 函数时，因为每个函数被分别调用，计数开始的时间可能不相同。通过同时设置 `TSTRA` 定时器启动寄存器 `CST0` 和 `CST1` 位，可以同时启动 `MTU0` 和 `MTU1` 计数。

详情参考 *RX66T Group User's Manual: Hardware*，章节 22.2.17 定时器启动寄存器（`TSTRA`，`TSTRB`，`TSTR`）和章节 22.2.19 定时器计数器同步启动寄存器（`TCSYSTR`）。

3.7.5.2 TGR 寄存器写入和比较匹配之间的竞争

如果在 `TGR` 寄存器写周期中发生比较匹配，则执行 `TGR` 的写操作，同时也会产生比较匹配信号。

详情参考 *RX66T Group User's Manual: Hardware*，章节 22.6.6 `TGR` 写操作和比较匹配之间的竞争。

3.7.5.3 设置一个大于占空比寄存器的值

在此示例代码中，占空比寄存器 `TGRA` 在产生 `MTU1.TGRA` 比较匹配中断（`TGIA1`）时被修改；占空比寄存器 `TGRB` 在产生 `MTU1.TGRB` 比较匹配中断（`TGIB1`）时被修改。

在 `TGRA`（`TGRB`）比较匹配发生后，如果在 `TGRA`（`TGRB`）中设置的值大于当前设置在 `TGRA`（`TGRB`）的值，则可能在一个周期内发生两次 `TGRA`（`TGRB`）比较匹配。

详情参考图 3.37 中（1）。

3.8 复位同步 PWM 模式

- 目标示例代码文件名: r01an5995_rx66t_mtu3_reset_sync_pwm.zip

3.8.1 概要

在 MTU 的复位同步 PWM 模式下, 通过将 MTU3 和 MTU4 与 MTU6 和 MTU7 组合, 可以输出共享一个公共的波形转换点的 6 相正和反的 PWM 波形 (共 12 相)。

此示例代码使用 MTU3 和 MTU4 的复位同步 PWM 模式, 并重复输出各 3 相 (正相和反相) 的 PWM 波形, 共 6 相, 占空比恒定。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用通道 3 和 4
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 MTU3.TGRA 作为周期寄存器
 - 定时器计数器清除源 = MTU3.TGRA 比较匹配
- 使用 MTU3.TGRB 作为占空比寄存器
 - MTIOC3B 引脚 (正) 初始输出为低;
MTU3.TGRB 比较匹配时输出高
 - MTIOC3D 引脚 (反) 初始输出为高;
MTU3.TGRB 比较匹配时输出低
- 使用 MTU4.TGRA 作为占空比寄存器
 - MTIOC4A 引脚 (正) 初始输出为低;
MTU4.TGRA 比较匹配时输出高
 - MTIOC4C 引脚 (反) 初始输出为高;
MTU4.TGRA 比较匹配时输出低
- 使用 MTU4.TGRB 作为占空比寄存器
 - MTIOC4B 引脚 (正) 初始输出为低;
MTU4.TGRB 比较匹配时输出高
 - MTIOC4D 引脚 (反) 初始输出为高;
MTU4.TGRB 比较匹配时输出低
- 无刷直流电机控制无效
- 使用复位同步 PWM 模式

在 Smart Configurator 中设置。
设置方法请参考章节 3.8.3。

此示例代码的复位同步 PWM 模式输出如下所示。

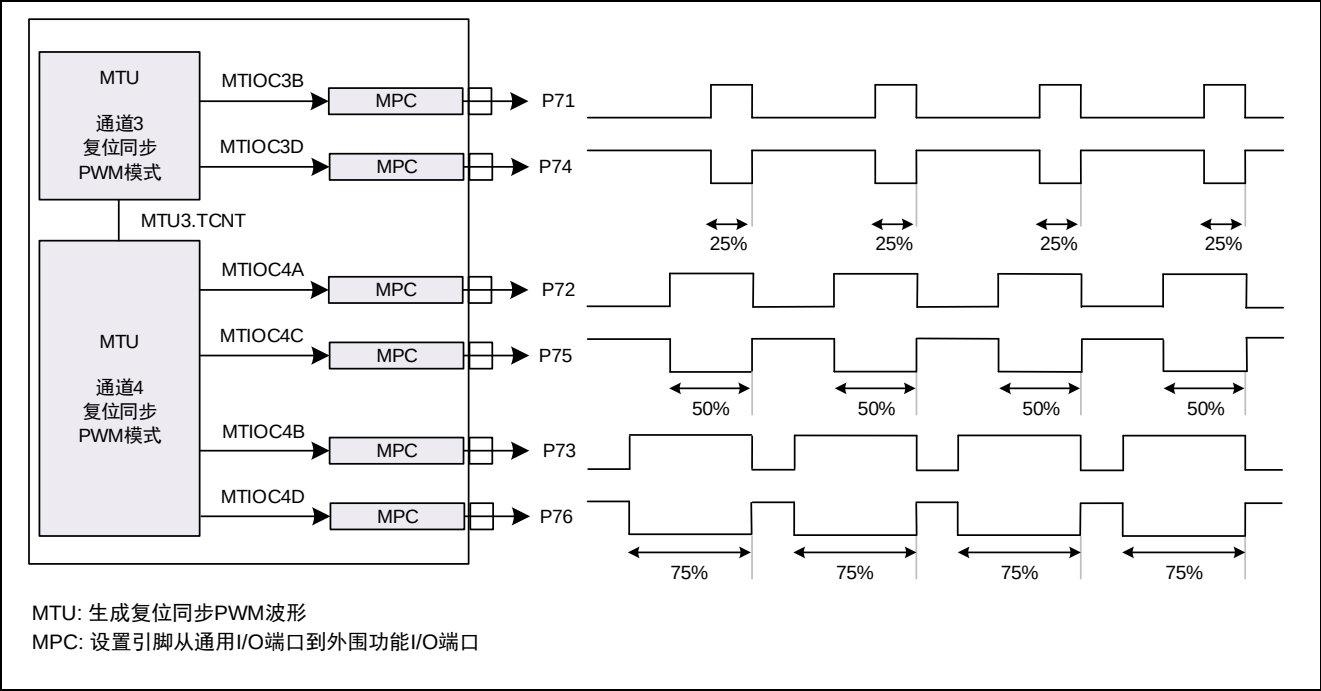


图 3.45 复位同步 PWM 模式输出

3.8.2 运行说明

示例代码的运行如图 3.46 所示。MTU3.TCNT 和 MTU4.TCNT 使用复位同步 PWM 模式作为递增计数器运行。当 MTU3.TCNT 和 MTU3.TGRA 之间发生比较匹配时，将清除计数器，然后从 0000h 开始递增。每次当 MTU3.TGRB、MTU4.TGRA 和 MTU4.TGRB 发生比较匹配和计数器清除发生时，PWM 输出引脚的输出都会翻转。

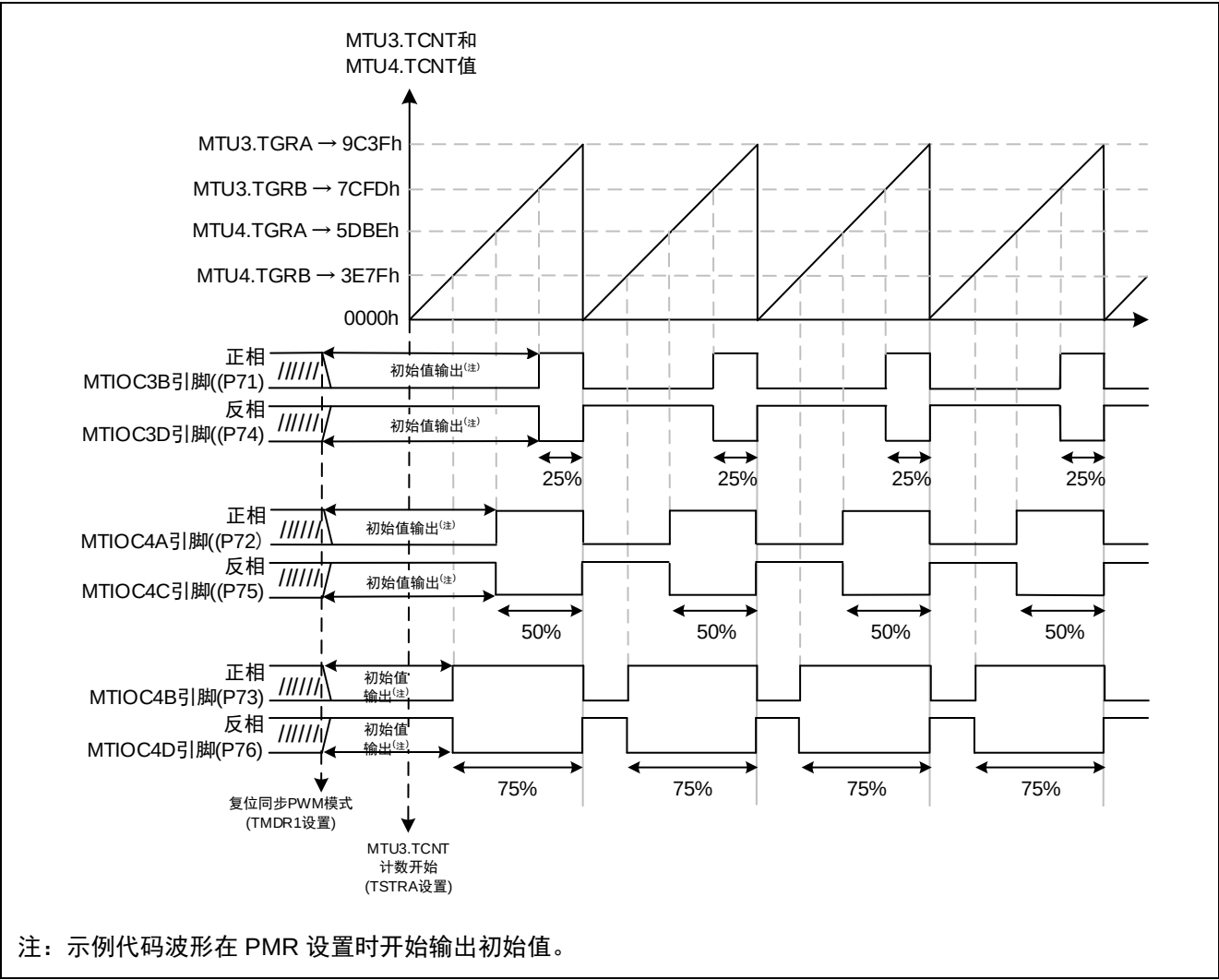


图 3.46 示例代码运行

3.8.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。
由于没有复位同步 PWM 模式组件，因此使用正常模式定时器组件生成代码。
添加组件的具体说明请参考 3.1.4 添加组件。

表 3.11 添加组件

项目	描述	
组件	正常模式定时器 ^注	
配置名称	Config_MTU3	Config_MTU3
输入捕获/输出比较引脚	2 个引脚	4 个引脚
资源	MTU3	MTU4

注：使用正常模式定时器组件生成代码后，在 MTU3 用户初始化函数 R_Config_MTU3_Create_UserInit 中设置复位同步 PWM 模式。有关寄存器设置的详细信息，请参考图 3.50。

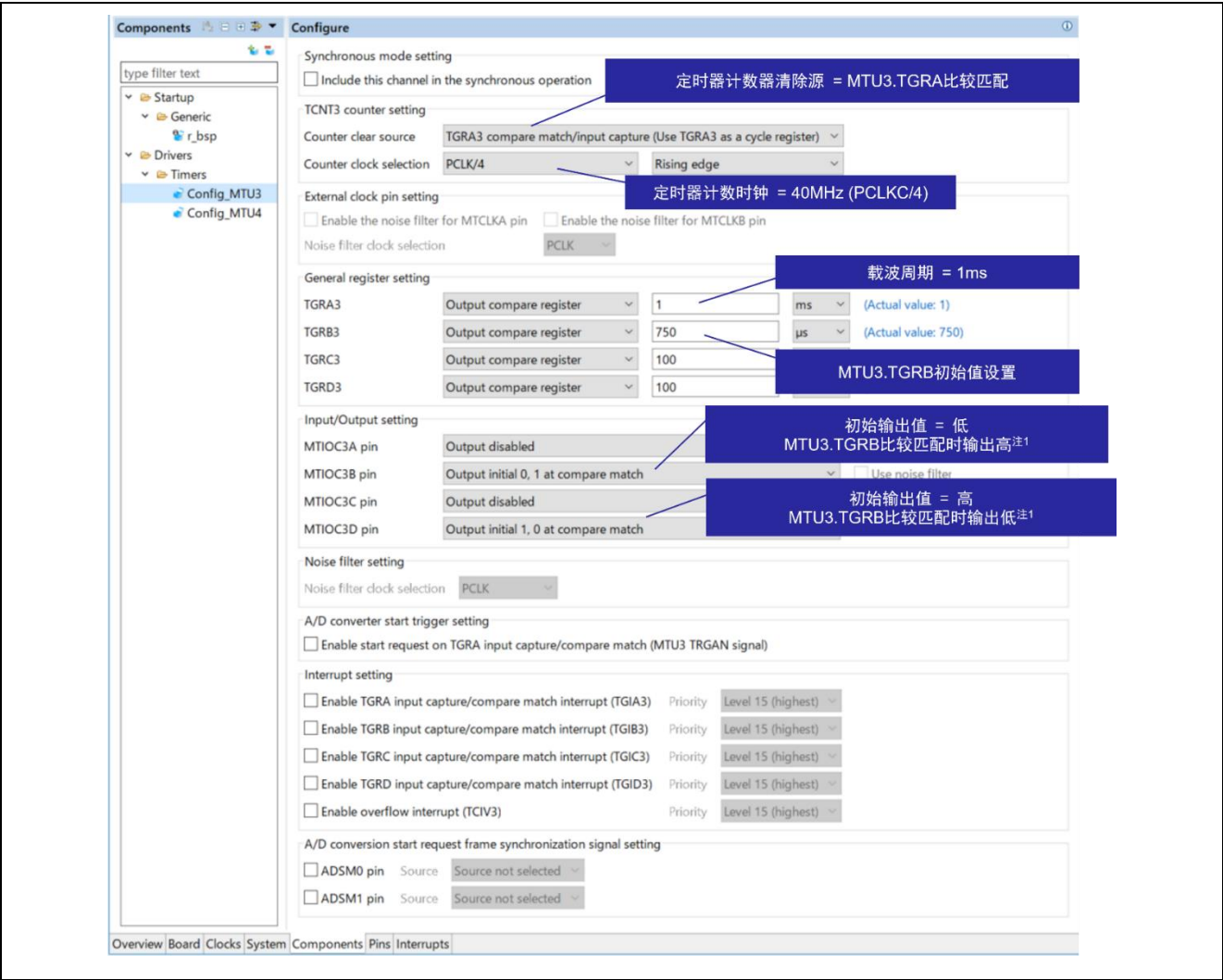


图 3.47 MTU3 设置

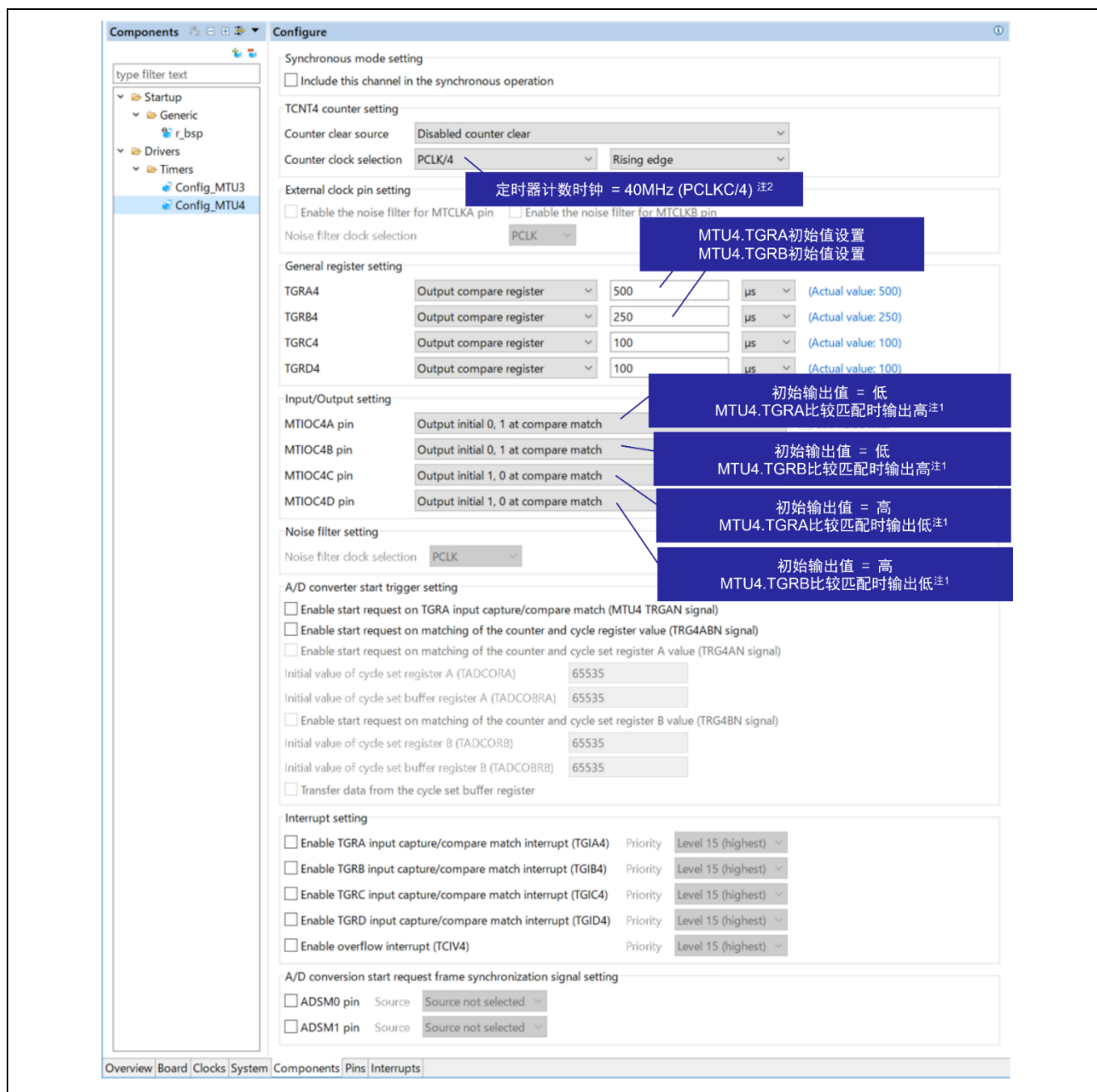


图 3.48 MTU4 设置

- 注 1: 使用正常模式定时器组件生成的代码输出 TIOR 寄存器设置。复位同步 PWM 模式的 PWM 输出电平在 TOCR1A 寄存器中设置, 而不是在 TIOR 寄存器中设置。因此, 不需要初始化 TIOR 寄存器, 并分别在用户初始化函数 `R_Config_MTU3_Create_UserInit` 和 `R_Config_MTU4_Create_UserInit` 中设置 TOCR1A 寄存器。有关 `R_Config_MTU3_Create_UserInit` 函数的详细信息, 请参考图 3.50, 有关 `R_Config_MTU4_Create_UserInit` 函数的详细信息, 请参考图 3.51。
- 注 2: 此设置会导致生成的代码输出 MTU4 预分频器选择位 (TCR.TPSC) 设置。复位同步 PWM 模式计数器时钟在 MTU3 的设置中使能, 但对 MTU4 的设置不是必需的。因此, 在用户初始化函数 `R_Config_MTU4_Create_UserInit` 中不需要初始化 TCR 寄存器。有关 `R_Config_MTU4_Create_UserInit` 函数的详细信息, 请参考图 3.51。

3.8.4 流程图

以下流程图显示了 Smart Configurator 生成代码后添加的函数的处理。

计数在主函数中启动。在复位同步 PWM 模式下，MTU3.TCNT 计数启动也会启动 MTU4.TCNT 计数。

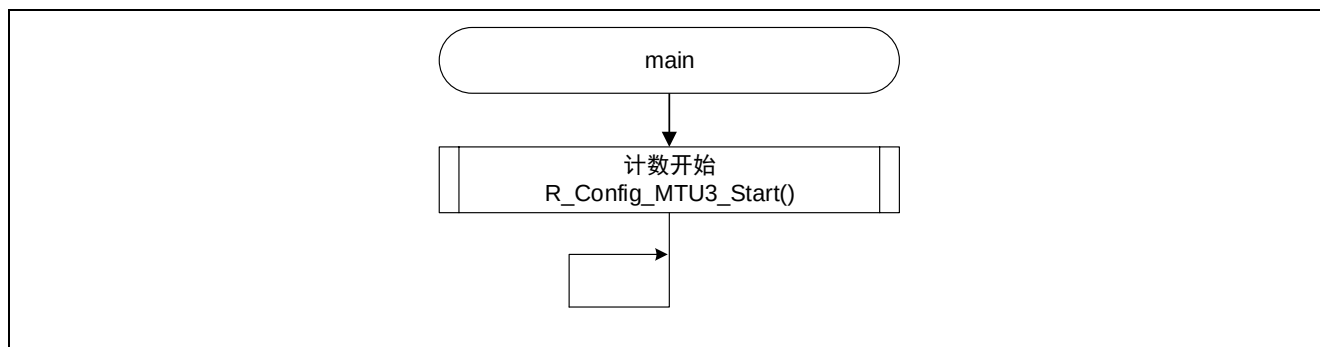


图 3.49 主函数

在主函数执行之前由 R_Config_MTU3_Create 函数调用用户初始化函数

R_Config_MTU3_Create_UserInit 来设置无刷直流电机控制、PWM 输出电平和定时器模式，并且在复位同步 PWM 模式下不需要初始化 TIOR 寄存器。

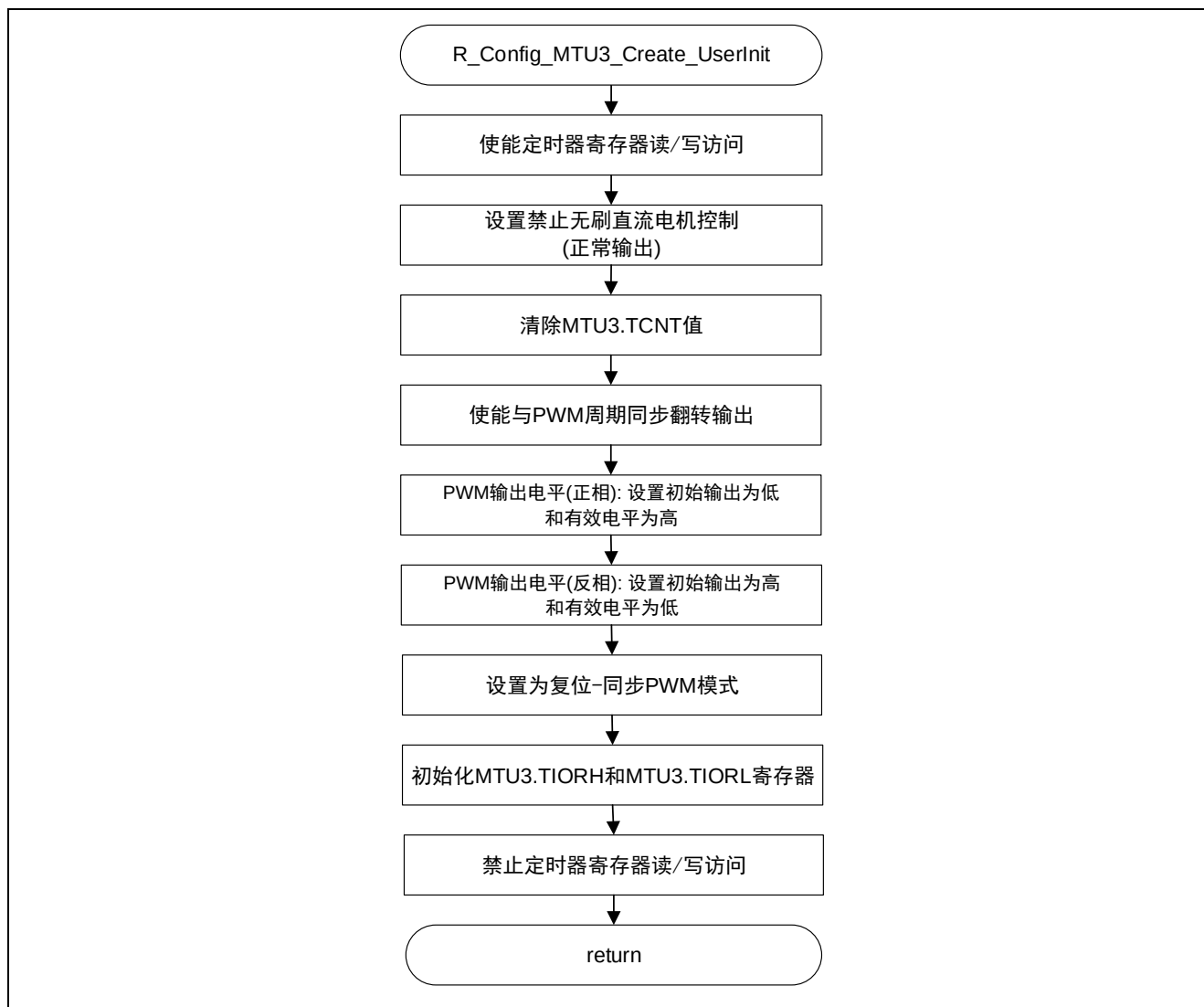


图 3.50 MTU3 用户初始化函数

在主函数执行之前由 R_Config_MTU4_Create 函数调用用户初始化函数 R_Config_MTU4_Create_UserInit，在复位同步 PWM 模式下不需要初始化 TIOR 寄存器。



图 3.51 MTU4 用户初始化函数

3.8.5 使用时的注意事项

3.8.5.1 使用 4 个通道时计数开始

在这种模式下，通过将 MTU3 和 MTU4 与 MTU6 和 MTU7 组合，可以输出共享一个公共波转换点的六相正反 PWM 波形（共 12 相）。

当使用 Smart Configurator 生成的 R_Config_MTU3_Start 和 R_Config_MTU6_Start 函数开始 MTU3 和 MTU6 计数时，因为每个函数被调用，计数开始时间可能不同。通过同时设置定时器计数器同步启动寄存器 TCSYSTR 的 SCH3 和 SCH6 位，可以同时启动 MTU3 和 MTU6 计数。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.19 定时器计数器同步启动寄存器（TCSYSTR）。

3.8.5.2 Smart Configurator 使用的组件

Smart Configurator 没有复位同步 PWM 模式组件。

在此示例代码中，对于使用正常模式定时器组件生成的代码，必须在 MTU3 用户初始化函数 R_Config_MTU3_Create_UserInit 中将定时器模式设置改变为复位同步 PWM 模式。有关 R_Config_MTU3_Create_UserInit 函数的详细信息，请参考图 3.50。

此外，当使用互补 PWM 模式组件时，按照定时器操作周期的设置自动固定了周期寄存器的值，并且由于无法指定计数器清除源，因此生成实现递增计数和递减计数的代码。因此，互补 PWM 模式组件不能用于创建生成复位同步 PWM 波形的代码。

3.9 互补 PWM 模式 带双缓冲器

- 目标示例代码文件名: r01an5995_rx66t_mtu3_complementary_pwm_dblbuf.zip

3.9.1 概要

使用 MTU 的互补 PWM 模式可输出带死区时间的三相互补 PWM 波形。

此示例代码描述了重复以下在互补 PWM 模式 3（在波峰和波谷传送）下带死区时间的输出波形。每个占空比使用双缓冲器生成左右不对称的 PWM 波形。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

当 TCNT 计数器溢出发生时，通过从缓冲寄存器 A 和 B 传送到暂存器 A 和 B 并且从暂存器 A 传送到比较寄存器来改变占空比；以及当下溢发生时从暂存器 B 传送到比较寄存器。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用互补 PWM 模式 3（在波峰和波谷传送）
- 使用通道 3 和 4
- 载波周期 = 1ms
- 死区时间 = 30 μ s
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 设置 MTU3.TGRA 为 MTU3.TCNT 上限值
（1/2 载波周期 + 死区时间）
— MTIOC3A 引脚翻转输出设置
- 设置缓冲传送时间
— 在计数的波峰和波谷传送数据
- 初始输出值高，有效电平低
- 使用 MTU3.TGRB 作为 U 相周期寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用 MTU4.TGRA 作为 V 相占空比寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用 MTU4.TGRB 作为 W 相占空比寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用双缓冲寄存器
— 使用 MTU3.TGRD 和 MTU3.TGRE 作为 MTU3.TGRB 的缓冲寄存器
— 使用 MTU4.TGRC 和 MTU4.TGRE 作为 MTU4.TGRA 的缓冲寄存器
— 使用 MTU4.TGRD 和 MTU4.TGRF 作为 MTU4.TGRB 的缓冲寄存器
— 有关左右不对称 PWM 输出波形的缓冲寄存器的初始值的详细信息，请参考图 3.5.4
- 每个周期的占空比更改
— 在 MTU3.TGRA 比较匹配中断时更改占空比
— 关于占空比改变的时间的详细信息，请参考图 3.5.4

在 Smart Configurator 中设置。
设置方法请参考章节 3.9.3。

此示例代码的互补 PWM 模式输出如下所示。

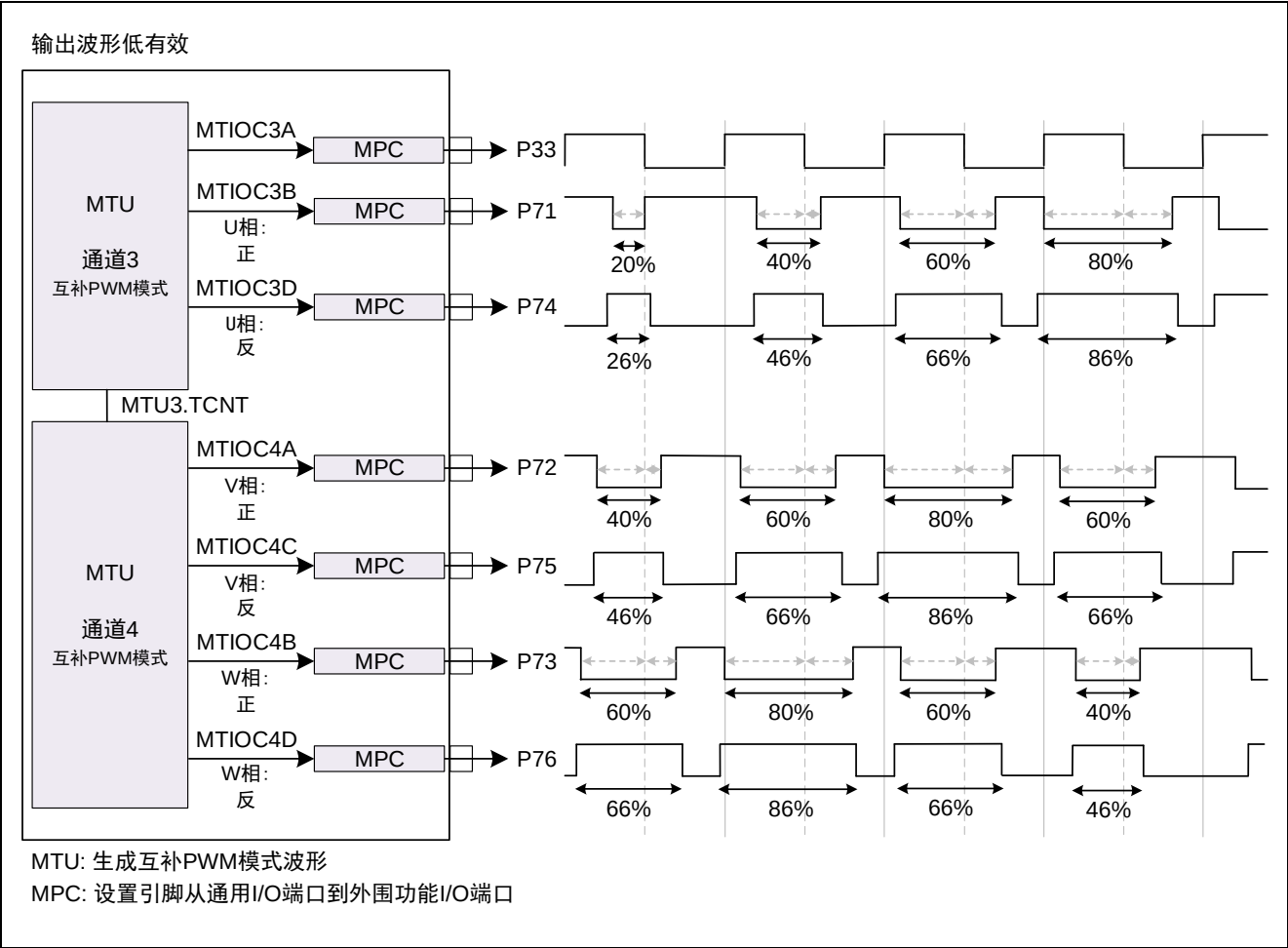


图 3.52 互补 PWM 模式输出

3.9.2 运行说明

示例代码的运行如图 3.54 所示。通过在设置为 MTU3.TCNT 上限值的 MTU3.TGRA 寄存器的比较匹配中断 (TGIA3) 处修改缓冲寄存器 A (MTU3.TGRD, MTU4.TGRC 和 MTU4.TGRD) 和缓冲寄存器 B (MTU3.TGRE, MTU4.TGRE 和 MTU4.TGRF) 的值, 可以在每个周期改变占空比的设置 (图 3.54 中 (1))。

在缓冲寄存器修改结束时写入 MTU4.TGRD 可以使能从缓冲寄存器 A 和 B 到暂存器 A 和 B 的数据传送。当在 Ta 间隔内写入 MTU4.TGRD 时, 写入缓冲寄存器的数据立即传送到暂存器, 但是在本示例代码中, MTU4.TGRD 是在 Tb1 间隔内写入, 因此在 Tb1 间隔结束后数据会传送到暂存器 (图 3.54 中 (2))。

由于此示例代码使用互补 PWM 模式 3 进行波峰波谷传送, 比较寄存器通过在 Tb1 间隔结束时 (图 3.54 中 (3)) 从暂存器 A 和 Tb2 间隔结束时 (图 3.54 中 (4)) 从暂存器 B 传送来更新数据。

根据 TOCR2A 的 OLS1P, OLS1N, OLS2P, OLS2N, OLS3P 和 OLS3N 位的设置, 正相和反相的输出初始值均为 OFF。在 MTU3.TMDR1 中设置互补 PWM 模式后, 输出继续直到 MTU4.TCNT 大于 TDDRA 寄存器的值 (图 3.54 中 (5))。

在上述操作之后, 重复以下步骤 1 至 3。

1. Ta 间隔内发生比较寄存器和计数器寄存器之间的比较匹配, 反相输出 OFF, 然后正相输出 ON (图 3.54 中 (6))。
2. Tb1 间隔内使能比较寄存器和暂存器 A。U 相发生比较匹配, 正相输出 OFF, 然后反相输出 ON。当 V 相和 W 相不发生比较匹配时, 波形不会改变 (图 3.54 中 (7))。
3. Tb2 间隔内使能比较寄存器和暂存器 B。当 U 相和 V 相的计数器寄存器和比较寄存器之间不发生比较匹配时, 波形不会改变。W 相发生比较匹配, 正相输出 OFF, 然后反相输出 ON。 (图 3.54 中 (8))。

- 左右不对称 PWM 波形输出

每个周期的占空比在递增计数期间和递减计数期间产生不同的占空比。

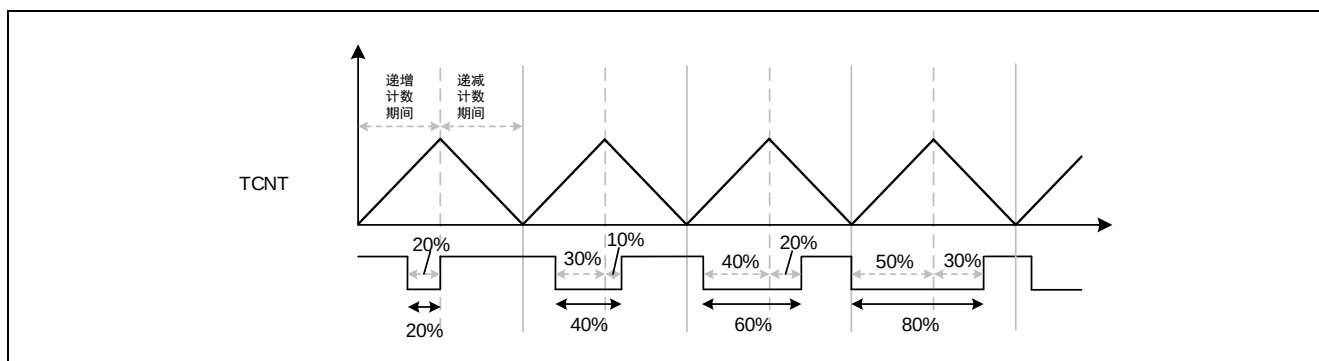


图 3.53 左右不对称 PWM 输出波形

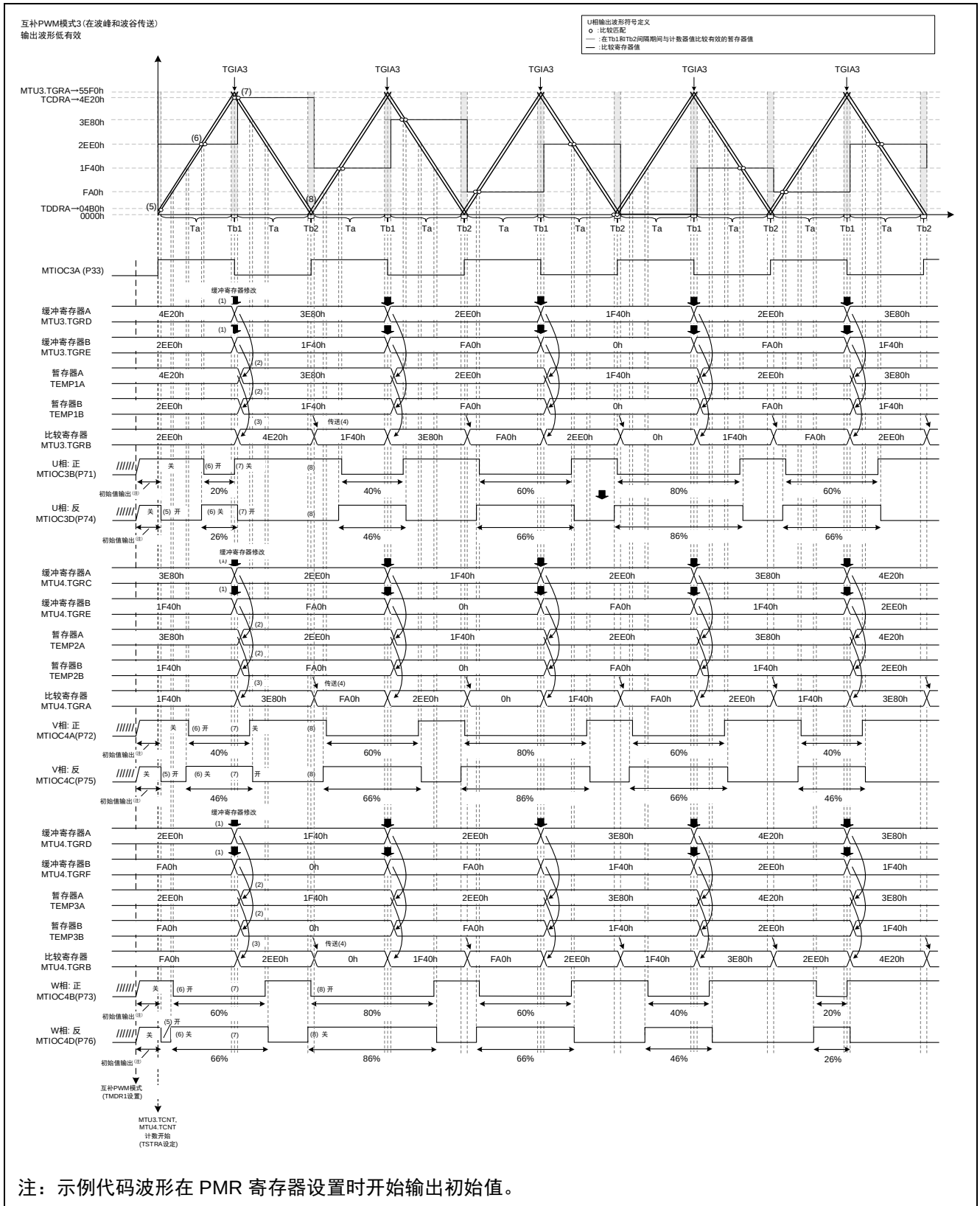


图 3.54 示例代码运行

3.9.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考章节 3.1.4 添加组件。

表 3.12 添加组件

项目	描述
组件	互补 PWM 模式定时器
配置名称	Config_MTU3_MTU4
运行	互补 PWM 模式 3（在波峰和波谷传送）
资源	MTU3_MTU4

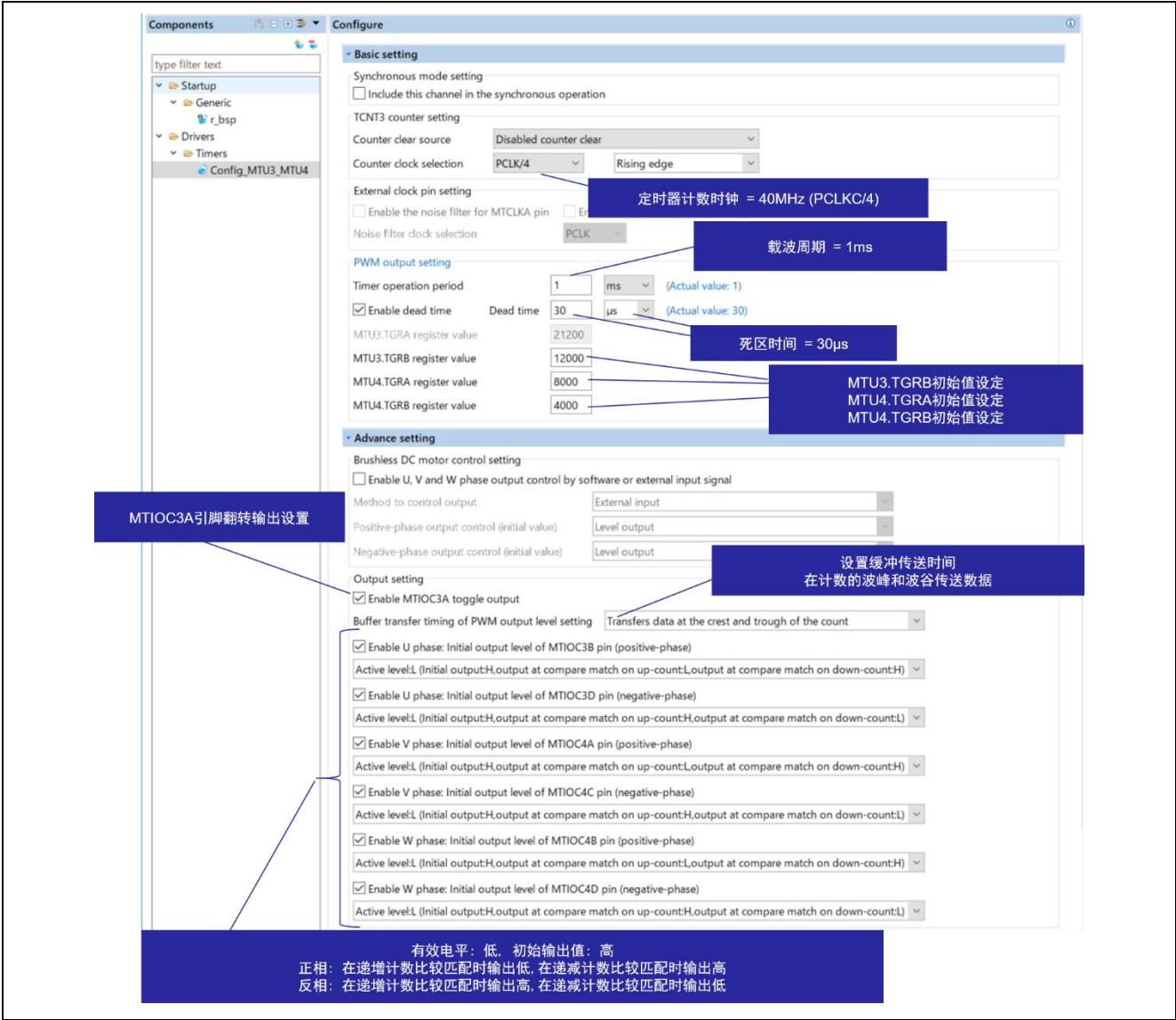


图 3.55 MTU3 和 MTU4 设置（1/2）

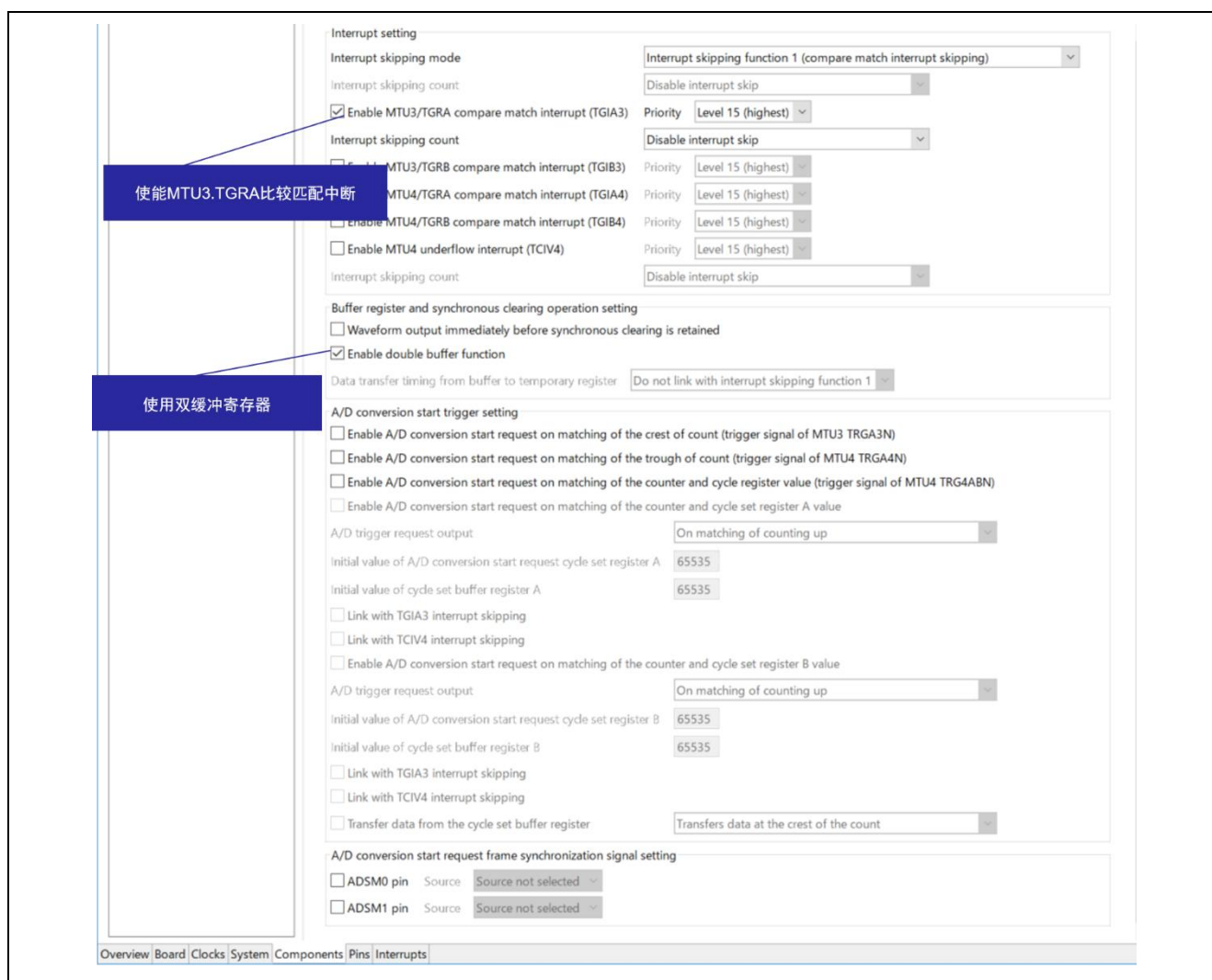


图 3.56 MTU3 和 MTU4 设置 (2/2)

3.9.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
MTU3.TCNT 和 MTU4.TCNT 计数在主函数中启动。

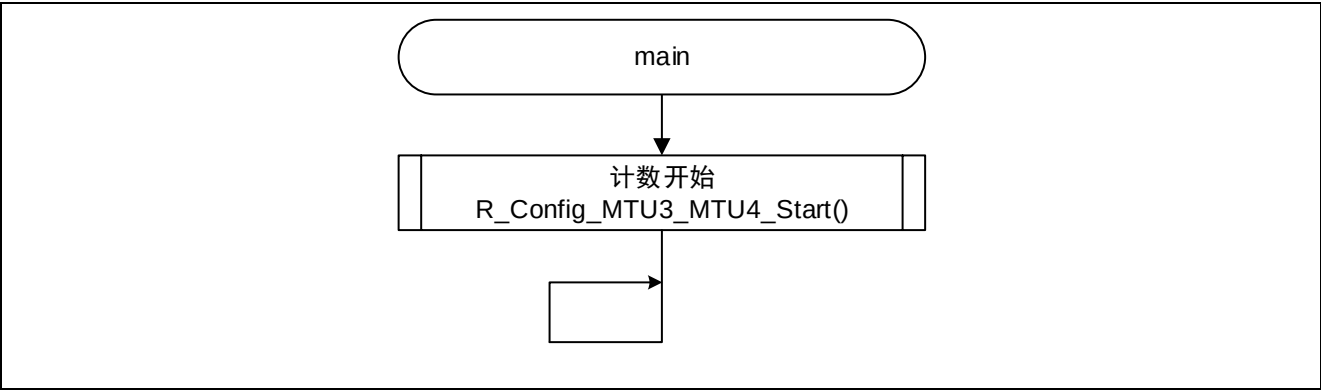


图 3.57 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU3_MTU4_Create_UserInit 中设置缓冲寄存器初始值并初始化变量。此函数在 R_Config_MTU3_MTU4_Create 函数中调用。

此函数初始化示例代码中使用的以下变量。

- s_u_ucduty_prv: 用于保存上一次 MTU3.TGRD 寄存器值的变量
- s_v_ucduty_prv: 用于保存上一次 MTU4.TGRC 寄存器值的变量
- s_w_ucduty_prv: 用于保存上一次 MTU4.TGRD 寄存器值的变量

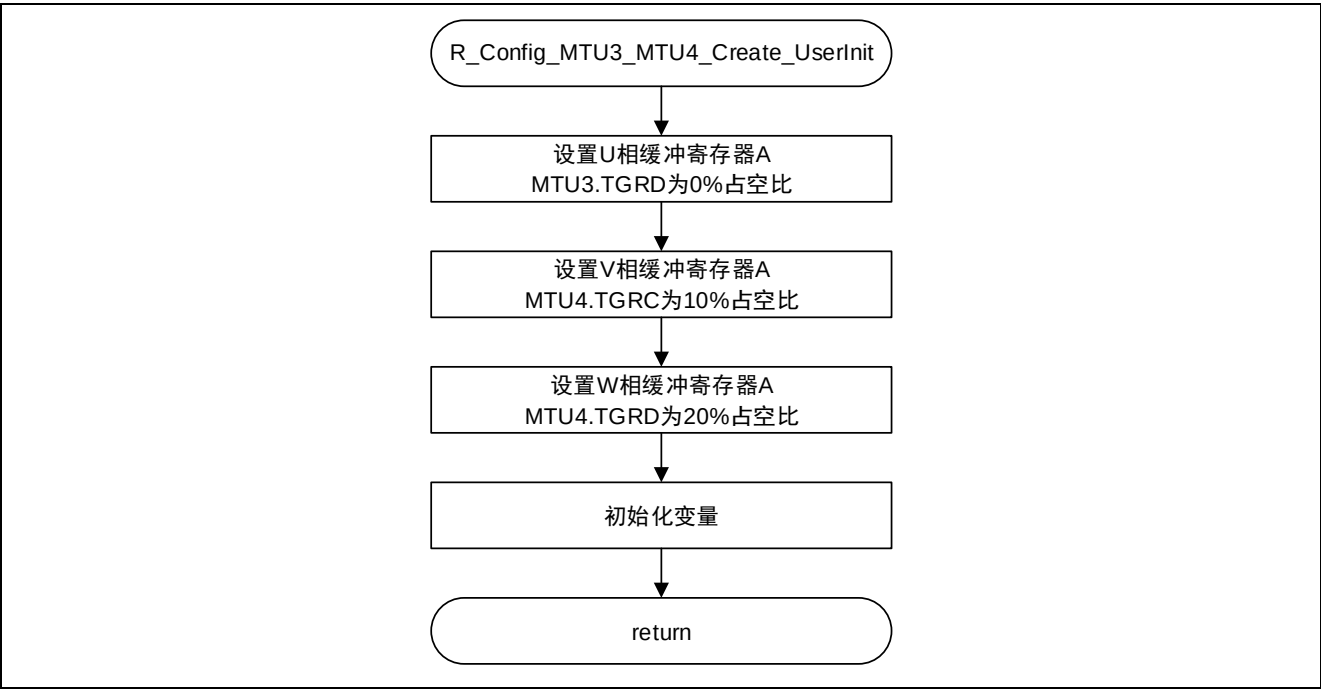


图 3.58 用户初始化函数

TGIA3 中断处理函数根据缓冲寄存器 A（MTU3.TGRD，MTU4.TGRC 和 MTU4.TGRD）的当前值和上一次缓冲寄存器 A 中的设置值来改变缓冲寄存器 A 和缓冲寄存器 B（MTU3.TGRE，MTU4.TGRE 和 MTU4.TGRF）的值。

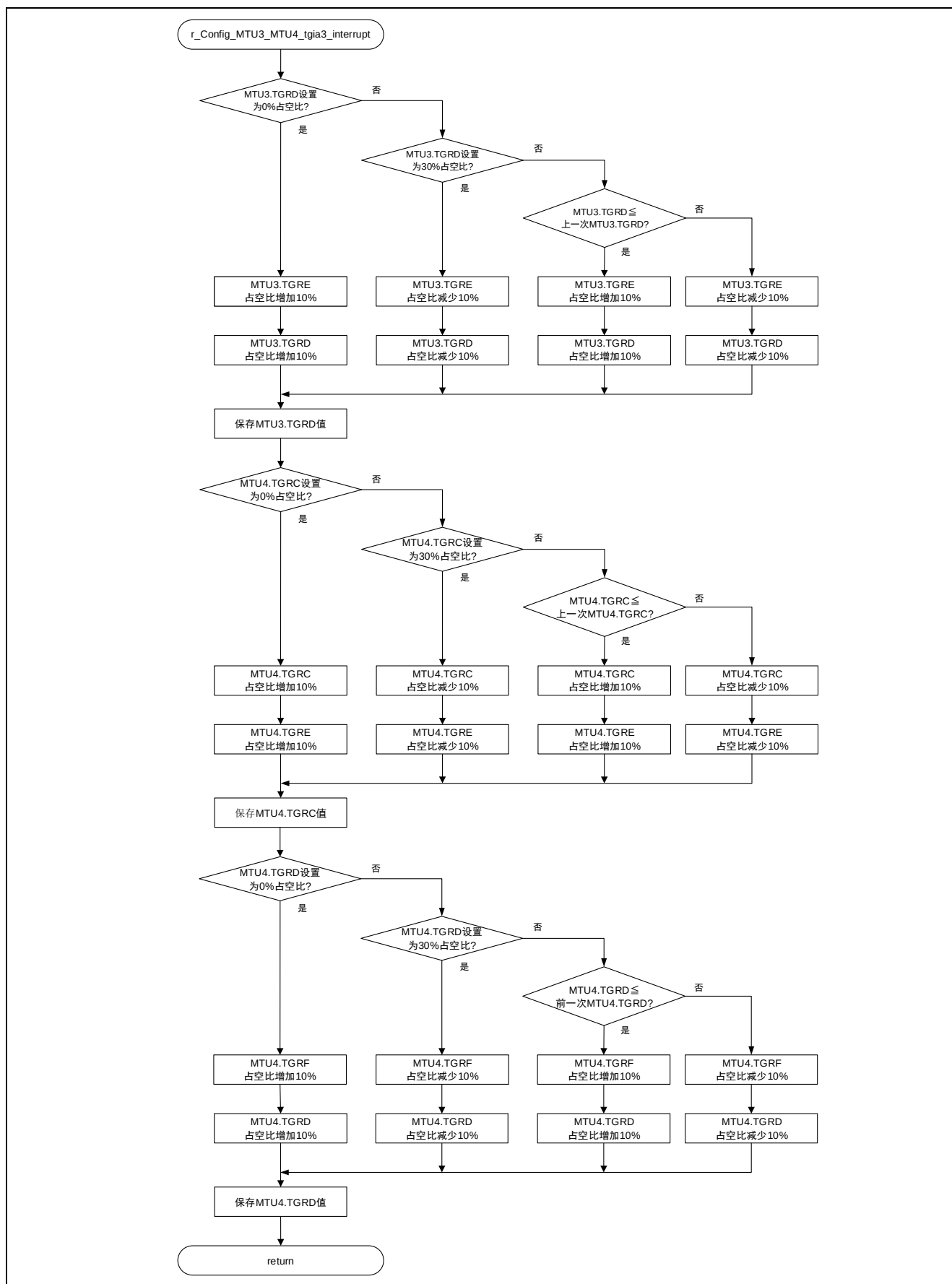


图 3.59 TGIA3 中断处理函数

3.9.5 使用时的注意事项

3.9.5.1 引脚设置

在互补 PWM 模式下，MTIOC3C 和 MTIOC6C 引脚不能用作定时器输入/输出引脚。将它们设置为 I/O 端口。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式, 表 22.74 互补 PWM 模式的输出引脚。

3.9.5.2 缓冲寄存器值更新

当修改缓冲寄存器中的数据时，确保最后修改 MTU4.TGRD (MTU7.TGRD)。在写入 MTU4.TGRD (MTU7.TGRD) 之后，数据从缓冲寄存器同时传送到所有的五个暂存器。

即使当不更新所有五个寄存器或者不更新 MTU4.TGRD (MTU7.TGRD) 中的数据时，也要确保在将数据写入要更新的寄存器后再写入 MTU4.TGRD (MTU7.TGRD)。在这种情况下，写入 MTU4.TGRD (MTU7.TGRD) 的数据应该与写入操作之前的数据相同。

3.9.5.3 缓冲运行设置

在互补 PWM 模式下，使用缓冲运行来修改 PWM 周期设置寄存器 (MTU3.TGRA 和 MTU6.TGRA)，定时器的周期数据寄存器 (TCDRA 和 TCDRB) 和占空比设置寄存器 (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA 和 MTU7.TGRB)。

当缓冲运行位 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 置 1 时，MTIOC4C (MTIOC7C) 和 MTIOC4D (MTIOC7D) 引脚无法输出波形。将 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 位置 0。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.14 互补 PWM 模式中的缓冲运行设置。

3.9.5.4 输出电平设置

当 MTU3 和 MTU4 (或 MTU6 和 MTU7) 为互补 PWM 模式时，通过 TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP 和 TOCR1B.OLSN 位，和 TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP 和 TOCR2B.OLSnN (n = 0 到 3) 位来设置 PWM 波形的输出电平。设置 TIOR 寄存器为 00h。

在互补 PWM 模式下，若将 TDERA.TDER (TDERB.TDER) 位设置为 0 (不生成死区时间)，反相输出电平按照 TOCR1A.OLSP (TOCR1B.OLSP) 和 TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 到 3) 位设置为正相输出的取反电平，与 TOCR1A.OLSN (TOCR1B.OLSN) 和 TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 到 3) 位的设置无关。

在不产生死区时间的情况下，如果只使能了负相的输出，而正相的输出在 TOER 寄存器中被禁止，则负相不会被输出。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.22 定时器输出控制寄存器 1 (TOCR1A, TOCR1B) 和章节 22.2.23 定时器输出控制寄存器 2 (TOCR2A, TOCR2B)。

3.9.5.5 TGR 寄存器初始值

Smart Configurator 生成的代码中，缓冲寄存器 A 和缓冲寄存器 B 的值被设置成与比较寄存器相同的值。设置缓冲寄存器 A 和 B 的初始值的代码，被增加在用户初始化函数 `R_Config_MTU3_MTU4_Create_UserInit` 中。

3.9.5.6 左右不对称 PWM 输出方法

当缓冲寄存器 A 的值被设置为缓冲寄存器 B 的值，PWM 输出是对称的。为了输出左右不对称 PWM 波形，设置缓冲寄存器 B 的值和缓冲寄存器 A 的值不同。

详情参考 *RX66T Group User's Manual: Hardware*，章节 22.3.8 互补 PWM 模式，(2) 互补 PWM 运行概要，段落 (s) 互补 PWM 模式下双缓冲功能。

3.10 互补 PWM 模式 无双缓冲器

- 目标示例代码文件名: r01an5995_rx66t_mtu3_complementary_pwm.zip

3.10.1 概要

使用 MTU 的互补 PWM 模式可输出带死区时间的三相互补 PWM 波形。

此示例代码描述了重复以下在互补 PWM 模式 3（在波峰和波谷传送）下带死区时间的输出波形。每个占空比使用缓冲器（不是双缓冲器）生成对称的 PWM 波形。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

当 TCNT 计数器溢出发生时，通过从缓冲寄存器传送到暂存器并且从暂存器传送到比较寄存器来改变占空比；以及当下溢发生时从暂存器传送到比较寄存器。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用互补 PWM 模式 3（在波峰和波谷传送）
- 使用通道 3 和 4
- 载波周期 = 1ms
- 死区时间 = 30 μ s
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 设置 MTU3.TGRA 为 MTU3.TCNT 上限值
（1/2 载波周期 + 死区时间）
— MTIOC3A 引脚翻转输出设置
- 设置缓冲器传送时间
— 在计数的波峰和波谷传送
- 初始输出值高，有效电平低
- 使用 MTU3.TGRB 作为 U 相占空比寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用 MTU4.TGRA 作为 V 相占空比寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用 MTU4.TGRB 作为 W 相占空比寄存器
— 正相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
— 反相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
- 使用缓冲寄存器
— 使用 MTU3.TGRD 作为 MTU3.TGRB 的缓冲寄存器
— 使用 MTU4.TGRC 作为 MTU4.TGRA 的缓冲寄存器
— 使用 MTU4.TGRD 作为 MTU4.TGRB 的缓冲寄存器
— 有关缓冲寄存器初始值的详细信息，请参考图 3.62
- 每个周期的占空比改变
— 在 MTU3.TGRA 比较匹配中断时改变占空比
— 关于占空比改变的时间的详细信息，请参考图 3.62

在 Smart Configurator 中设置。
设置方法请参考章节 3.10.3。

此示例代码的互补 PWM 模式输出如下所示。

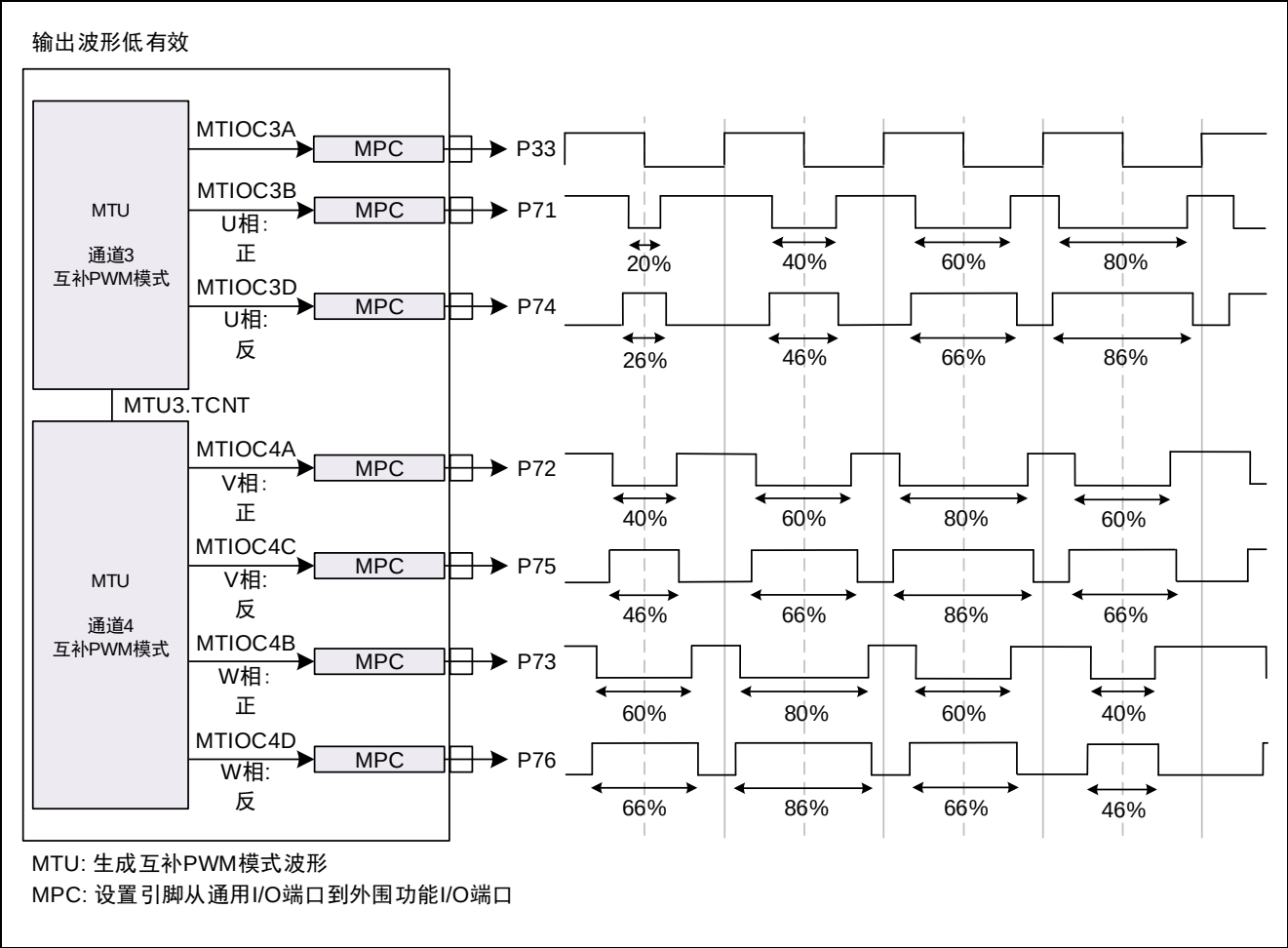


图 3.60 互补 PWM 模式输出

3.10.2 运行说明

示例代码的运行如图 3.62 所示。通过在设置为 MTU3.TCNT 上限值的 MTU3.TGRA 寄存器的比较匹配中断 (TGIA3) 处修改缓冲寄存器 (MTU3.TGRD, MTU4.TGRC 和 MTU4.TGRD) 的值, 可以在每个周期改变占空比的设置 (图 3.62 中 (1))。

在缓冲寄存器修改结束时写入 MTU4.TGRD 可以使能从缓冲寄存器到暂寄存器的数据传送。当在 Ta 间隔内写入 MTU4.TGRD 时, 写入缓冲寄存器的数据立即传送到暂寄存器, 但是在本示例代码中, MTU4.TGRD 是在 Tb1 间隔内写入, 因此在 Tb1 间隔结束后数据会传送到暂寄存器 (图 3.62 中 (2))。

由于此示例代码使用互补 PWM 模式 3 进行波峰波谷传送, 比较寄存器通过在 Tb1 间隔结束时 (图 3.62 中 (3)) 和 Tb2 间隔结束时 (图 3.62 中 (4)) 从暂寄存器传送来更新数据。

根据 TOCR2A 的 OLS1P, OLS1N, OLS2P, OLS2N, OLS3P 和 OLS3N 位的设置, 正相和反相的输出初始值均为 OFF。在 MTU3.TMDR1 中设置互补 PWM 模式后, 输出继续直到 MTU4.TCNT 大于 TDDRA 寄存器的值 (图 3.62 中 (5))。

在上述操作之后, 重复以下步骤 1 至 3。

1. Ta 间隔内发生比较寄存器和计数器寄存器之间的比较匹配, 反相输出 OFF, 然后正相输出 ON (图 3.62 中 (6))。
2. Tb1 间隔内使能比较寄存器和暂寄存器, 但是因为与计数器寄存器的比较匹配没有发生, 波形不会改变 (图 3.62 中 (7))。
3. Tb2 间隔内使能比较寄存器和暂寄存器, 但是因为与计数器寄存器的比较匹配没有发生, 波形不会改变 (图 3.62 中 (8))。

- 对称 PWM 输出波形

每个周期的占空比在递增计数期间和递减计数期间产生 1/2 占空比。

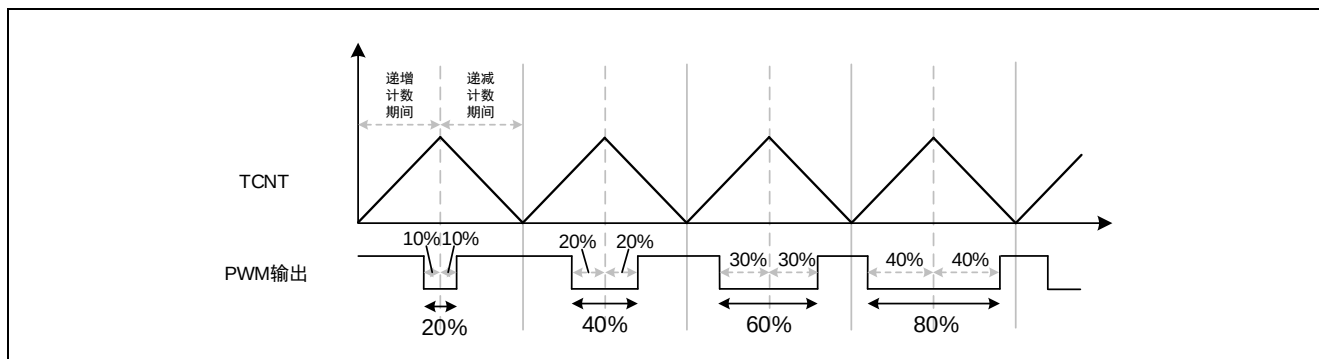


图 3.61 对称 PWM 输出波形

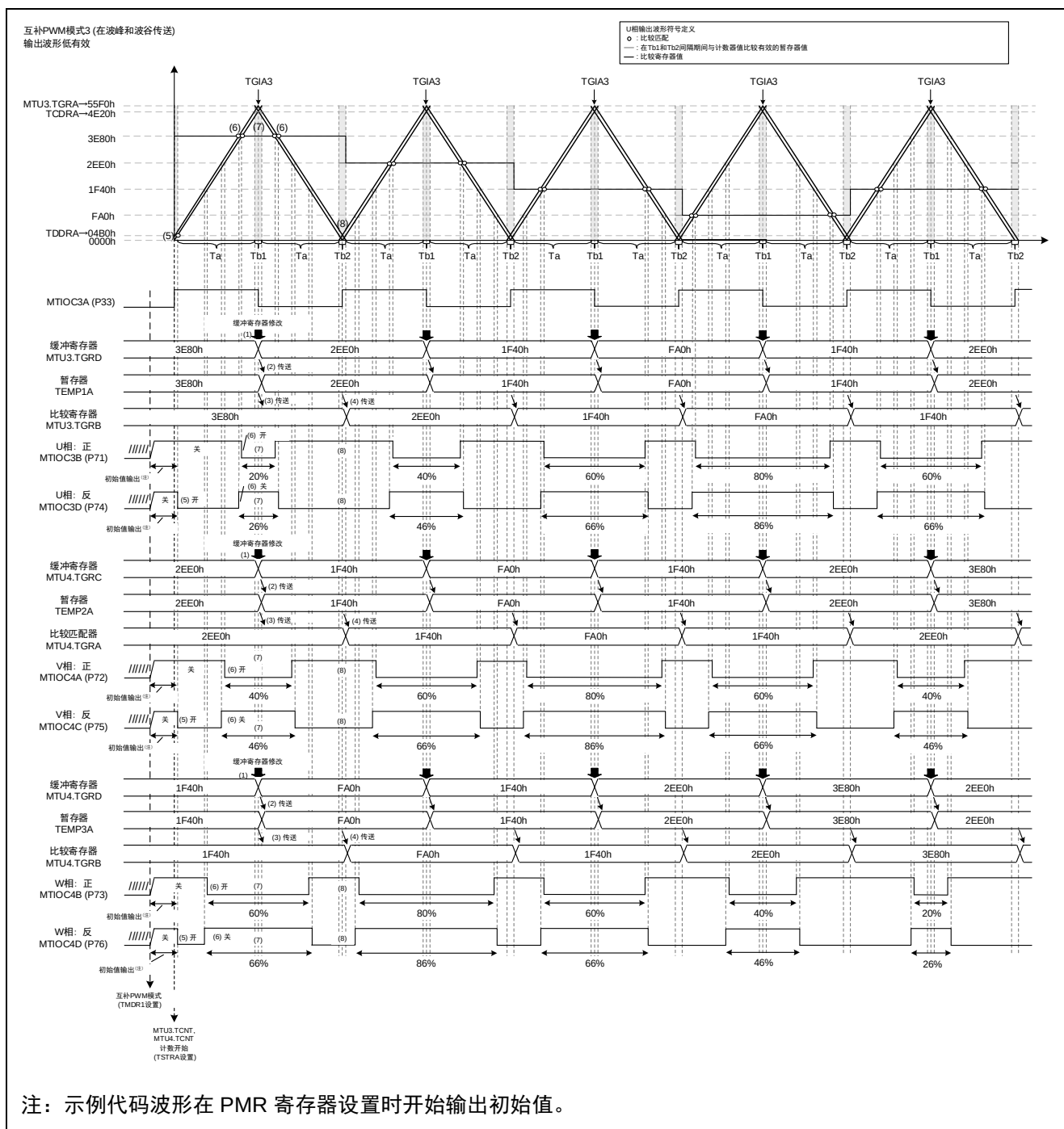


图 3.62 示例代码运行

3.10.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.13 添加组件

项目	描述
组件	互补 PWM 模式定时器
配置名称	Config_MTU3_MTU4
运行	互补 PWM 模式 3（在波峰和波谷传送）
资源	MTU3_MTU4



图 3.63 MTU3 和 MTU4 设置（1/2）

Interrupt setting

Interrupt skipping mode: Interrupt skipping function 1 (compare match interrupt skipping)

Interrupt skipping count: Disable interrupt skip

☒ Enable MTU3/TGRA compare match interrupt (TGIA3) Priority: Level 15 (highest)

☐ Enable MTU3/TGRB compare match interrupt (TGIB3) Priority: Level 15 (highest)

☐ Enable MTU3/TGRC compare match interrupt (TGIC3) Priority: Level 15 (highest)

☐ Enable MTU3/TGRD compare match interrupt (TGID3) Priority: Level 15 (highest)

☐ Enable MTU4 underflow interrupt (TCIV4) Priority: Level 15 (highest)

Interrupt skipping count: Disable interrupt skip

Buffer register and synchronous clearing operation setting

☐ Waveform output immediately before synchronous clearing is retained

☐ Enable double buffer function

Data transfer timing from buffer to temporary register: Do not link with interrupt skipping function 1

A/D conversion start trigger setting

☐ Enable A/D conversion start request on matching of the crest of count (trigger signal of MTU3 TRGA3N)

☐ Enable A/D conversion start request on matching of the trough of count (trigger signal of MTU4 TRGA4N)

☐ Enable A/D conversion start request on matching of the counter and cycle register value (trigger signal of MTU4 TRG4ABN)

☐ Enable A/D conversion start request on matching of the counter and cycle set register A value

A/D trigger request output: On matching of counting up

Initial value of A/D conversion start request cycle set register A: 65535

Initial value of cycle set buffer register A: 65535

☐ Link with TGIA3 interrupt skipping

☐ Link with TCIV4 interrupt skipping

☐ Enable A/D conversion start request on matching of the counter and cycle set register B value

A/D trigger request output: On matching of counting up

Initial value of A/D conversion start request cycle set register B: 65535

Initial value of cycle set buffer register B: 65535

☐ Link with TGIA3 interrupt skipping

☐ Link with TCIV4 interrupt skipping

☐ Transfer data from the cycle set buffer register: Transfers data at the crest of the count

A/D conversion start request frame synchronization signal setting

☐ AD5M0 pin Source: Source not selected

☐ AD5M1 pin Source: Source not selected

Overview Board Clocks System Components Pins Interrupts

图 3.64 MTU3 和 MTU4 设置 (2/2)

3.10.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
MTU3.TCNT 和 MTU4.TCNT 计数在主函数中启动。

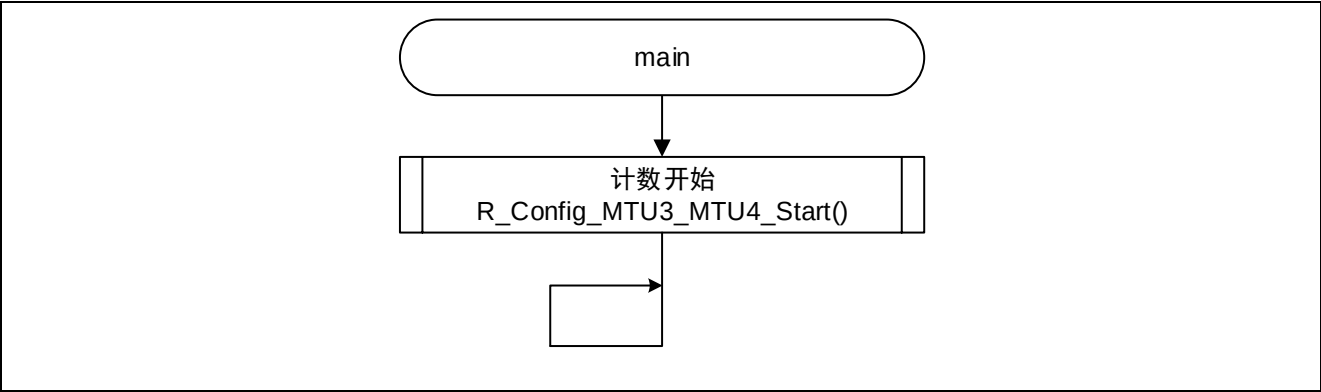


图 3.65 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU3_MTU4_Create_UserInit 中初始化变量。
此函数在 R_Config_MTU3_MTU4_Create 函数中调用。

此函数初始化示例代码中使用的以下变量。

- s_u_ucduty_prv: 用于保存上一次 MTU3.TGRD 寄存器值的变量
- s_v_ucduty_prv: 用于保存上一次 MTU4.TGRC 寄存器值的变量
- s_w_ucduty_prv: 用于保存上一次 MTU4.TGRD 寄存器值的变量

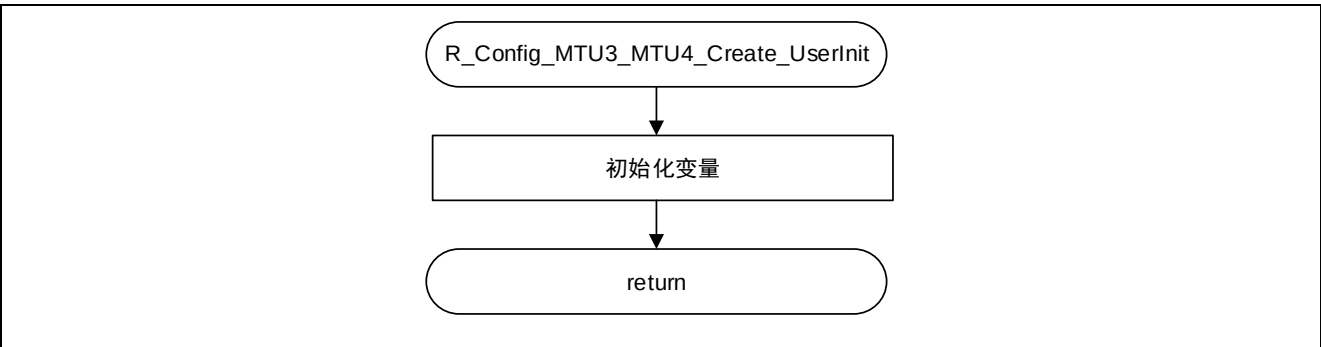


图 3.66 用户初始化函数

TGIA3 中断处理函数根据缓冲寄存器（MTU3.TGRD，MTU4.TGRC 和 MTU4.TGRD）的当前值和上一次缓冲寄存器中的设置值来改变缓冲寄存器的值。

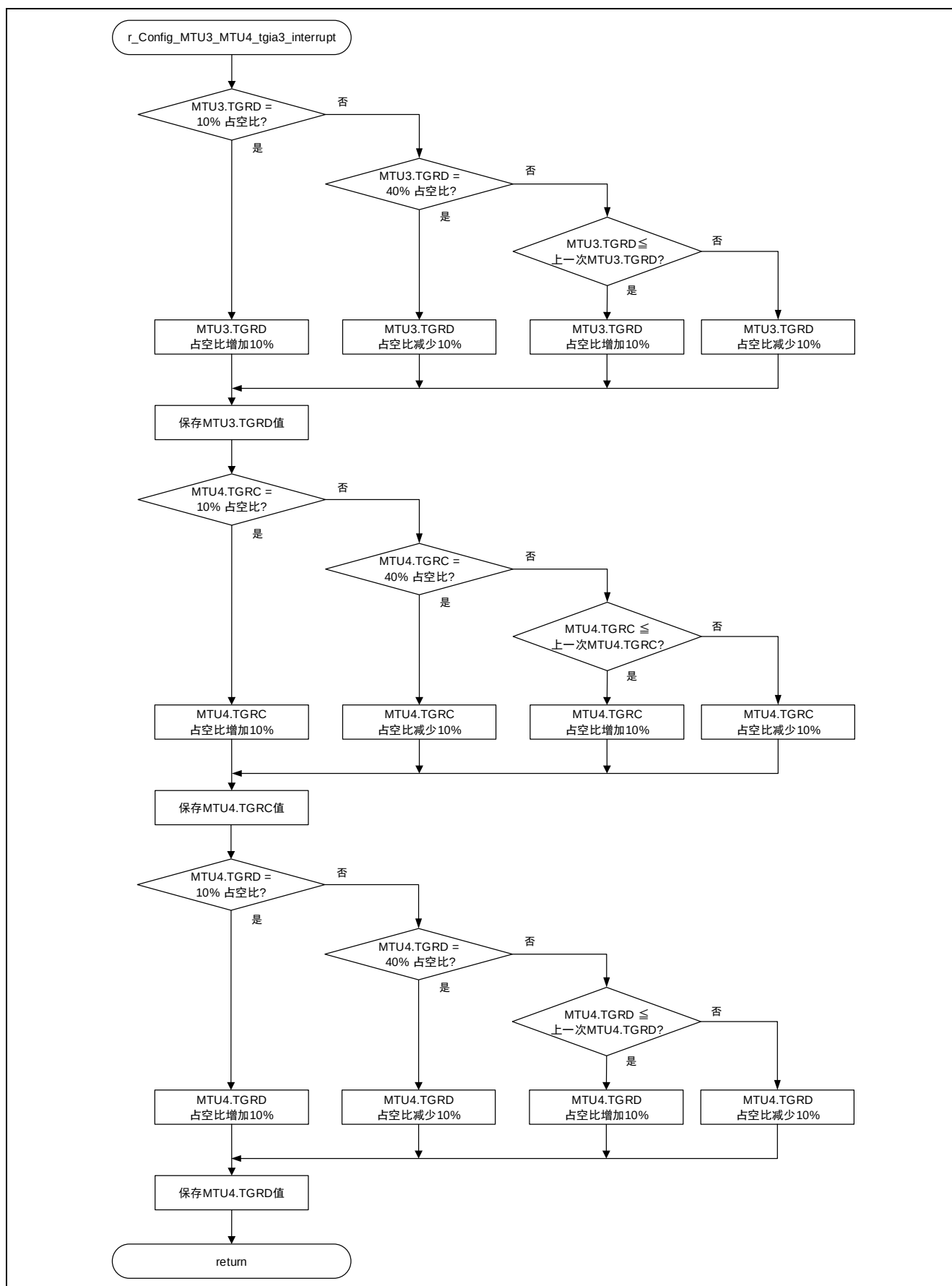


图 3.67 TGIA3 中断处理函数

3.10.5 使用时的注意事项

3.10.5.1 引脚设置

在互补 PWM 模式下，MTIOC3C 和 MTIOC6C 引脚不能用作定时器输入/输出引脚。将它们设置为 I/O 端口。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式, 表 22.74 互补 PWM 模式的输出引脚。

3.10.5.2 缓冲寄存器值更新

当修改缓冲寄存器中的数据时，确保最后修改 MTU4.TGRD (MTU7.TGRD)。在写入 MTU4.TGRD (MTU7.TGRD) 之后，数据从缓冲寄存器同时传送到所有的五个暂存器。

即使当不更新所有五个寄存器或者不更新 MTU4.TGRD (MTU7.TGRD) 中的数据时，也要确保在将数据写入要更新的寄存器后再写入 MTU4.TGRD (MTU7.TGRD)。在这种情况下，写入 MTU4.TGRD (MTU7.TGRD) 的数据应该与写入操作之前的数据相同。

3.10.5.3 缓冲运行设置

在互补 PWM 模式下，使用缓冲运行来修改 PWM 周期设置寄存器 (MTU3.TGRA 和 MTU6.TGRA)，定时器的周期数据寄存器 (TCDRA 和 TCDRB) 和占空比设置寄存器 (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA 和 MTU7.TGRB)。

当缓冲运行位 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 置 1 时，MTIOC4C (MTIOC7C) 和 MTIOC4D (MTIOC7D) 引脚无法输出波形。将 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 位置 0。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.14 互补 PWM 模式下的缓冲运行设置。

3.10.5.4 输出电平设置

当 MTU3 和 MTU4 (或 MTU6 和 MTU7) 为互补 PWM 模式时，通过 TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP 和 TOCR1B.OLSN 位，和 TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP 和 TOCR2B.OLSnN (n = 0 到 3) 位来设置 PWM 波形的输出电平。设置 TIOR 寄存器为 00h。

在互补 PWM 模式下，若将 TDERA.TDER (TDERB.TDER) 位设置为 0 (不生成死区时间)，按照 TOCR1A.OLSP (TOCR1B.OLSP) 和 TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 到 3) 位的设置，反相输出为正相输出的取反电平，且不依赖于 TOCR1A.OLSN (TOCR1B.OLSN) 和 TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 到 3) 位的设置。

在不产生死区时间的情况下，如果只使能了负相的输出，而正相的输出在 TOER 寄存器中被禁止，则负相不会被输出。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.22 定时器输出控制寄存器 1 (TOCR1A, TOCR1B) 和章节 22.2.23 定时器输出控制寄存器 2 (TOCR2A, TOCR2B)。

3.11 PWM 模式 1 占空比 0%到 100%（在比较匹配时修改比较寄存器）

- 目标示例代码文件名：01an5995_rx66t_mtu3_pwm1_50to100.zip

3.11.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述使用 PWM 模式 1 并重复输出以下波形，包括占空比 0%和 100%。

- 占空比切换：50% → 80% → 100% → 80% → 50% → 0% → . . .

基本操作是当使用缓冲寄存器 TGRC 在占空比寄存器 TGRA 比较匹配发生时，通过将 TGRC 的值传送到 TGRA 来改变占空比。当从 0%切换到 50%时，在周期寄存器 TGRB 比较匹配发生时，将执行直接修改 TGRA 寄存器的流程。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - 在 TGRB 比较匹配时输出低
- 使用 TGRA 作为占空比寄存器
 - 在 TGRA 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 TGRC 作为 TGRA 的缓冲寄存器
 - 在 TGRA 比较匹配时缓冲传送
- 每个周期的占空比改变
 - 在 TGRA 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 3.69

在 Smart Configurator 中设置。
设置方法请参考章节 3.11.3。

此示例代码的 PWM 模式 1 输出如下所示：

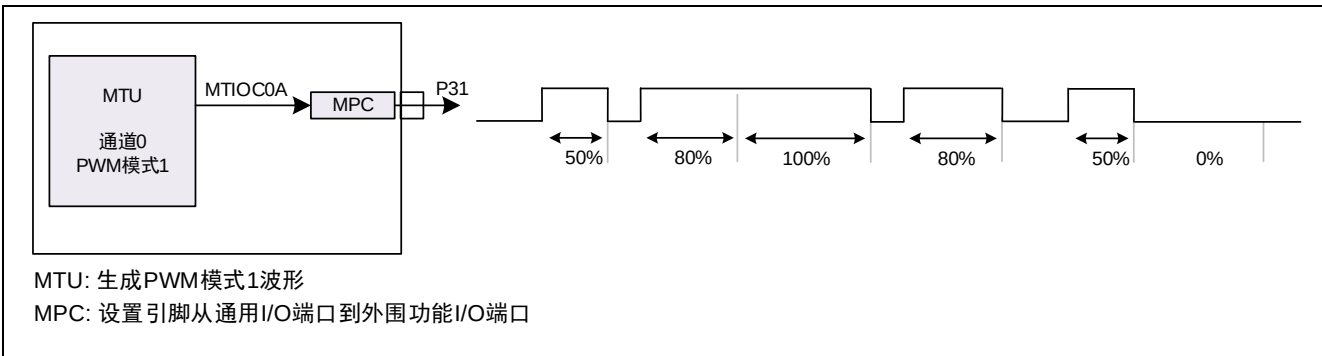
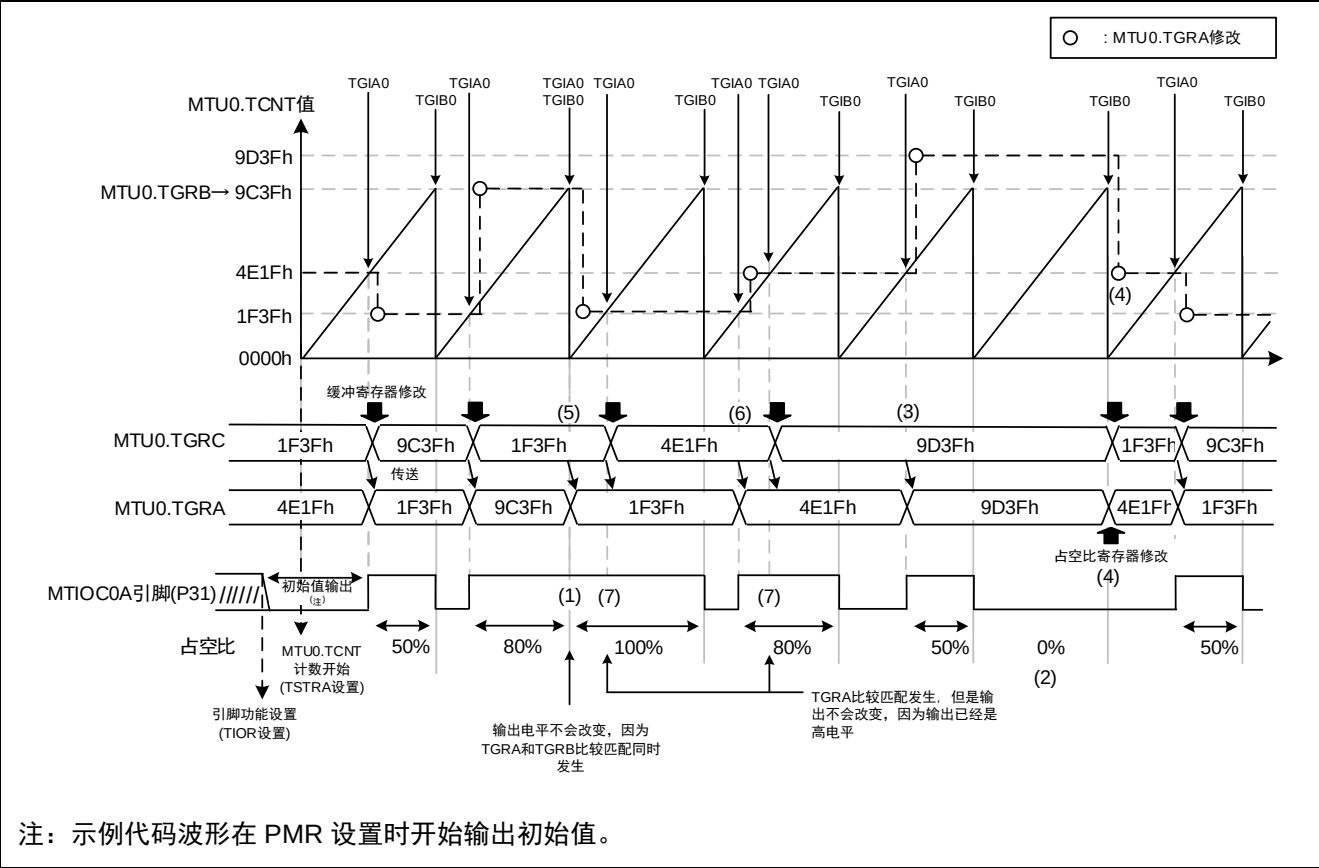


图 3.68 PWM 模式 1 输出

3.11.2 运行说明

示例代码的运行如图 3.69 所示。基本操作是通过在占空比寄存器 TGRA 的比较匹配中断 (TGIA0) 中修改缓冲寄存器 TGRC 的值, 并在 TGRA 比较匹配发生时将 TGRC 的值传送到占空比寄存器 TGRA, 从而改变占空比的设置。

- 100%占空比输出
当输出 100%占空比时, 将 TGRA 设置为与 TGRB 相同的值。
当 TGRA 的值与 TGRB 的值相同时, 两个比较匹配将同时发生。由于当两个比较匹配同时发生时输出不会改变, 因此保持高电平从而输出 100%占空比 (图 3.69 中 (1))。
- 0%占空比输出
当输出 0%占空比时, 将 TGRA 设置为大于 TGRB 的值。
当 TGRA 的值大于 TGRB 时, 保持低电平并输出 0%占空比, 因为没有发生 TGRA 比较匹配 (图 3.69 中 (2))。
为了在一个周期内继续保持低电平并输出 0%占空比, TGRC 的值不会被 TGIA0 中断修改, 在中断中大于 TGRB 的值被传送到 TGRA。当 TGIA0 发生时修改 TGRC 值的时间需要调整 (图 3.69 中 (3))。
- 从 0%占空比切换
当输出 0%占空比时, 因为 TGRA 大于 TGRB, 不会发生 TGRA 比较匹配。因此, TGIA0 不会发生, 并且没有修改缓冲寄存器 TGRC 的值或将 TGRC 的值传送到 TGRA 的时机。当从占空比 0%切换到 50%时, TGRC 和 TGRA 寄存器在 TGIB0 发生时直接修改 (图 3.69 中 (4))。
- TGRA 比较匹配在一个周期内发生两次
当从占空比 100%切换到 80%时, 将发生两次比较匹配, 包括与 TGRB 比较匹配同时发生的 TGRA 比较匹配, 因此必须调整修改 TGRC 值的时间 (图 3.69 中 (5))。
要从占空比 80%切换到 50%, 请将 TGRA 设置为大于当前值的值。
这将导致在一个周期内发生两次比较匹配, 因此必须调整修改 TGRC 值的时间 (图 3.69 中 (6))。
因为第二次 TGRA 比较匹配发生在高输出期间, 所以波形不会改变。(图 3.69 中 (7))。



注：示例代码波形在 PMR 设置时开始输出初始值。

图 3.69 示例代码运行

3.11.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.14 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU0
运行	PWM 模式 1
资源	MTU0

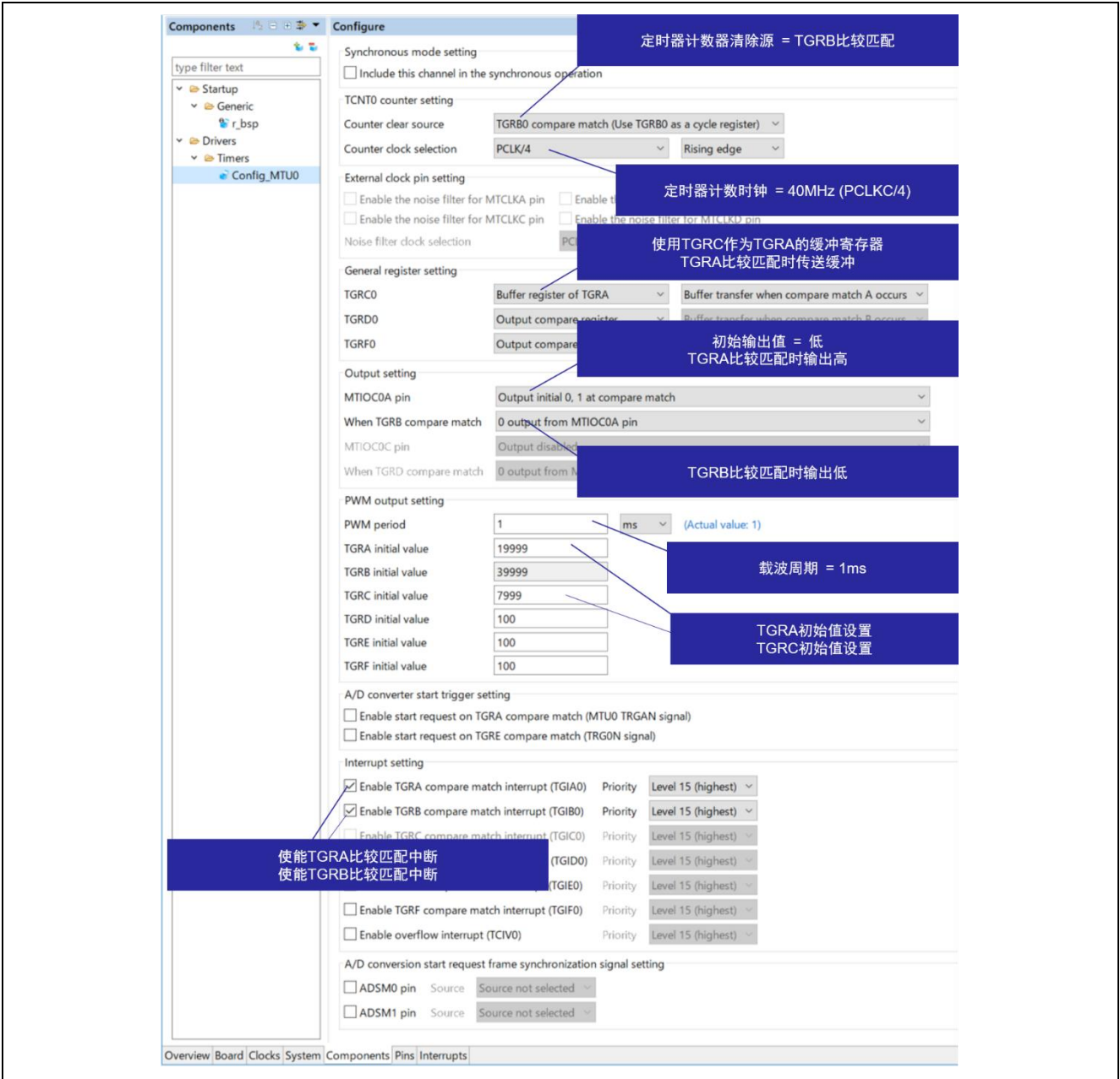


图 3.70 MTU0 设置

3.11.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

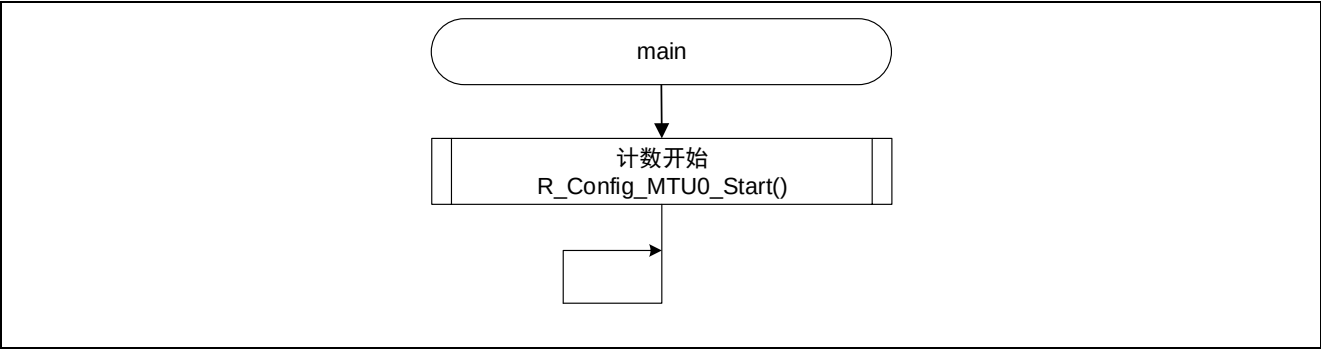


图 3.71 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU0_Create_UserInit 中初始化变量。此函数被 R_Config_MTU0_Create 函数调用。

此函数初始化此示例代码中使用的以下变量。

- s_duty_list_counter: 用于从占空比列表中读取的计数器变量
- s_wait_cnt: 等待变量，用于管理 0%占空比输出期间 TGIB0 的发生次数
- s_duty_prv: 用于保存上一次 TGRA 寄存器值的变量

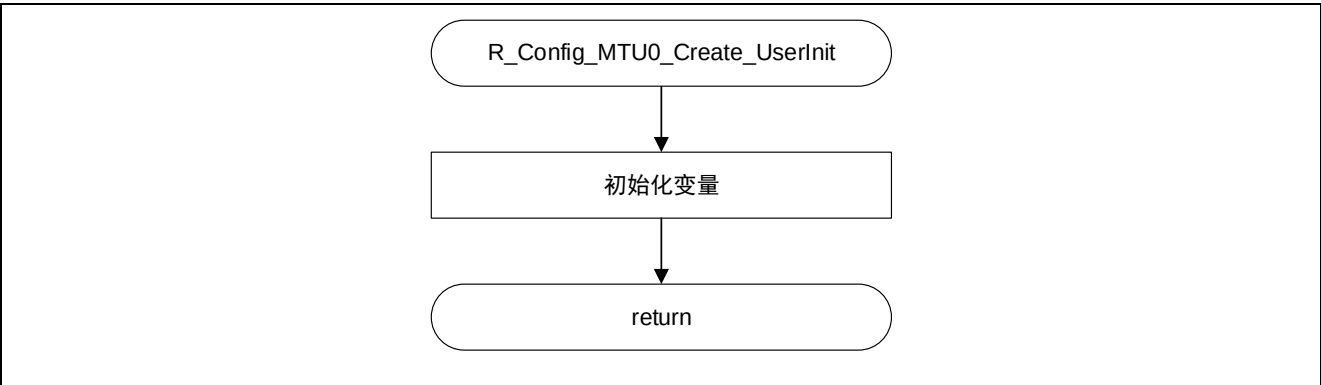


图 3.72 用户初始化函数

TGIA0 中断处理函数根据 TGRA 寄存器的当前值和上一次 TGRA 寄存器中设置的值来改变 TGRC 寄存器的值。

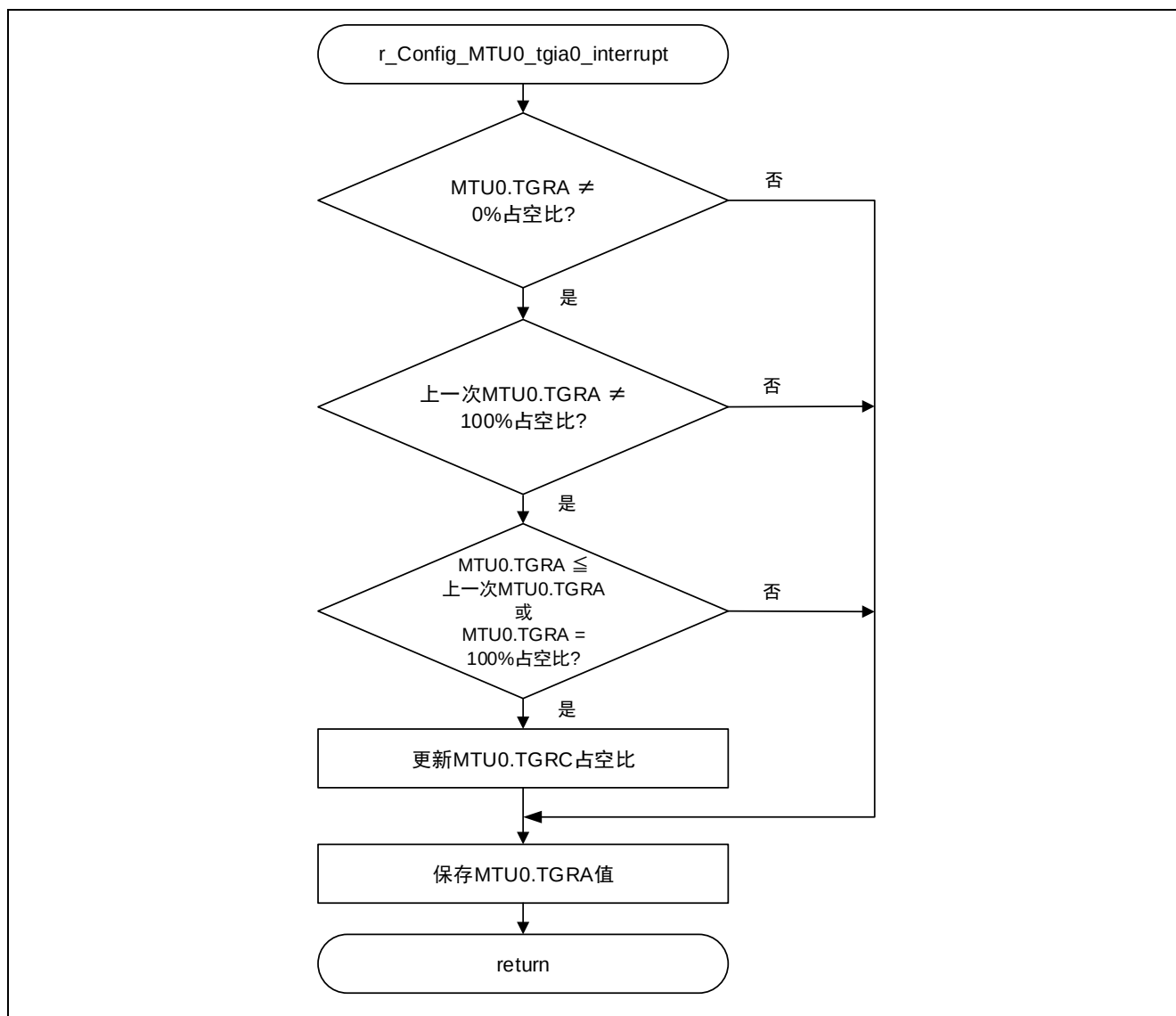


图 3.73 TGIA0 中断处理函数

TGIB0 中断处理函数在从 0%占空比切换时直接将占空比设置到 TGRA 和 TGRC。如果在将 0%占空比传送到 TGRA 后，当第一次 TGIB0 发生时修改 TGRA，则不会输出 0%，因此需要在第二次 TGIB0 发生时设置占空比，并且需要管理等待计数。

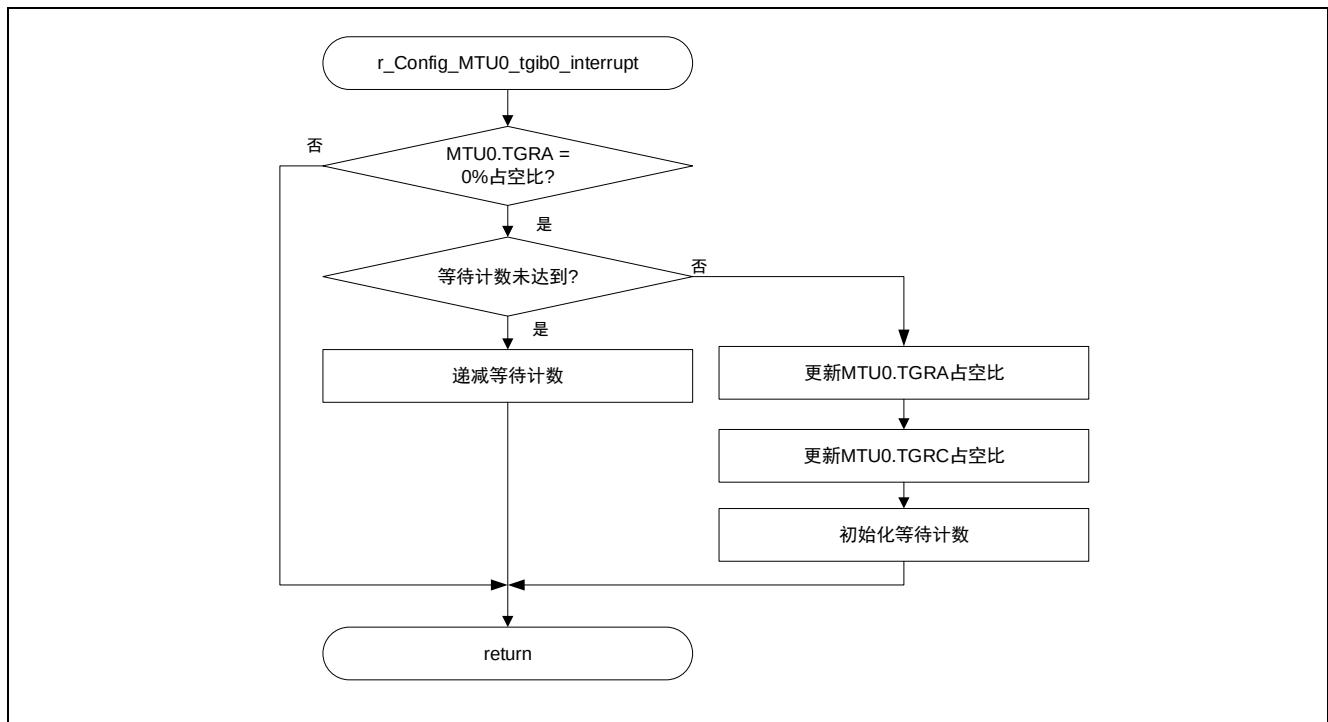


图 3.74 TGIB0 中断处理函数

3.11.5 相关运行

3.11.5.1 占空比 0%到 100%（无缓冲器）

下图显示了在不使用缓冲器的情况下，占空比为 0%到 100%的 PWM 波形输出示例，如图 3.69 所示。

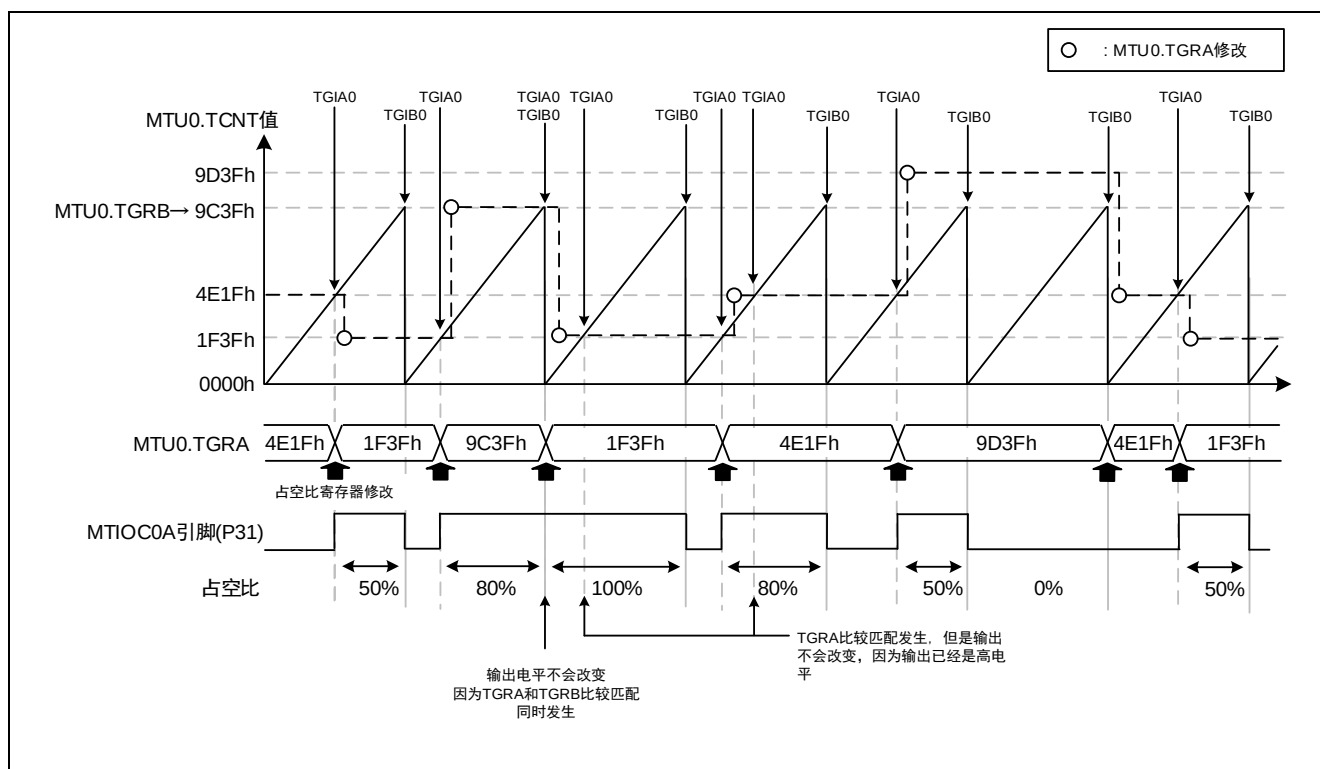


图 3.75 占空比 0%到 100%的 PWM 输出（无缓冲器）

3.11.6 使用时的注意事项

3.11.6.1 缓冲寄存器修改延迟

在此示例代码中，当 TGRA 比较匹配发生时，将执行从缓冲寄存器 TGRC 到 TGRA 的传送。必须在下一次 TGRA 比较匹配发生之前完成 TGRC 值的修改。

如果缓冲寄存器 TGRC 修改延迟，则无法输出预期的占空比。

3.11.6.2 设置一个大于占空比寄存器的值

在此示例代码中，当 MTU0.TGRA 的比较匹配中断（TGIA0）发生时，修改缓冲寄存器 TGRC，并在 TGRA 比较匹配发生时将值从 TGRC 传送到 TGRA。如果在 TGRA 比较匹配发生后，在 TGRA 中设置的值大于当前 TGRA 的值，则在一个周期内可能会发生两次 TGRA 比较匹配。

有关详细信息，请参考图 3.69 中（6）。

3.11.6.3 从 0%占空比切换

在此示例代码中，当 MTU0.TGRA 的比较匹配中断（TGIA0）发生时，修改缓冲寄存器 TGRC，并在 TGRA 比较匹配发生时将值从 TGRC 传送到 TGRA。如果在 TGRA 中设置了大于 TGRB 的值，并且输出 0%占空比，则不会发生 TGRA 比较匹配，因为在达到 TGRA 值之前计数器就被清除了。

从 0%占空比切换时，需要直接在 TGRA 寄存器中设置该值。

有关详细信息，请参考图 3.69 中（4）。

3.12 PWM 模式 1 占空比 0%到 100%（在计数器清除时修改比较寄存器）

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm1_50to100_rwcc.zip

3.12.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述使用 PWM 模式 1 并重复输出以下波形，包括占空比 0%和 100%。

- 占空比切换: 50% → 80% → 100% → 80% → 50% → 0% → . . .

基本操作是当使用缓冲寄存器 TGRC 在占空比寄存器 TGRA 比较匹配发生时，通过将 TGRC 的值传送到 TGRA 来改变占空比。当从 0%切换到 50%时，在周期寄存器 TGRB 比较匹配发生时，将直接修改 TGRA 寄存器。当周期寄存器 TGRB 比较匹配发生时，TGRC 被修改。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - 在 TGRB 比较匹配时输出低
- 使用 TGRA 作为占空比寄存器
 - 在 TGRA 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 TGRC 作为 TGRA 的缓冲寄存器
 - 在 TGRA 比较匹配时缓冲传送
- 每个周期的占空比改变
 - 在 TGRB 比较匹配中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 3.77

在 Smart Configurator 中设置。
设置方法请参考章节 3.12.3。

此示例代码的 PWM 模式 1 输出如下所示。

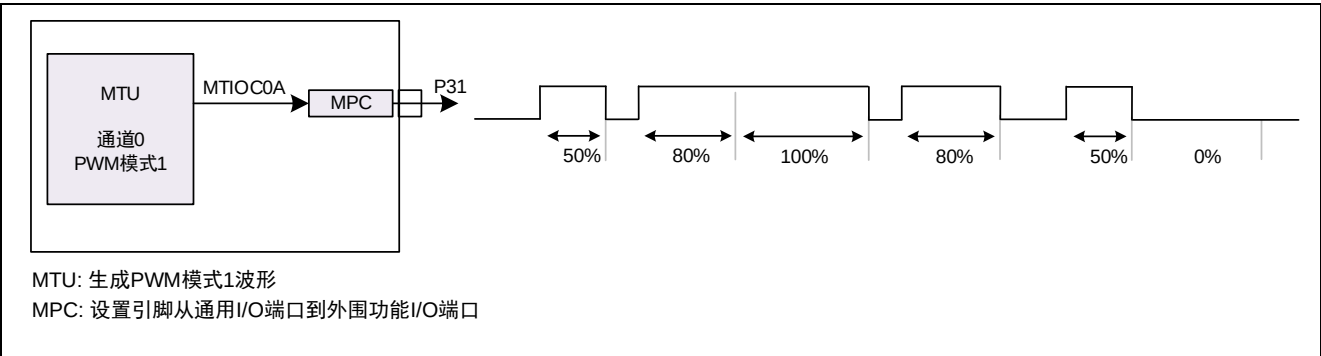


图 3.76 PWM 模式 1 输出

3.12.2 运行说明

示例代码的运行如图 3.77 所示。基本操作是通过在周期寄存器 TGRB 的比较匹配中断 (TGIB0) 中修改缓冲寄存器 TGRC 的值, 并在 TGRA 比较匹配发生时将 TGRC 的值传送到占空比寄存器 TGRA, 从而改变每个周期的占空比的设置。

- 100%占空比输出
当输出 100%占空比时, 将 TGRA 设置为与 TGRB 相同的值。
当 TGRA 的值与 TGRB 的值相同时, 两个比较匹配将同时发生。由于当两个比较匹配同时发生时输出不会改变, 因此保持高电平从而输出 100%占空比 (图 3.77 中 (1))。
- 0%占空比输出
当输出 0%占空比时, 将 TGRA 设置为大于 TGRB 的值。
当 TGRA 的值大于 TGRB 时, 保持低电平并输出 0%占空比, 因为没有发生 TGRA 比较匹配 (图 3.77 中 (2))。
- 从 100%占空比切换
当输出 100%占空比时, TGRA 和 TGRB 比较匹配将同时发生。因此, 没有修改 TGRC 到一个新的值的时机。当从占空比 100%切换到 0%时, TGRA 寄存器的值在 TGIB0 发生时直接修改 (图 3.77 中 (3))。
- 从 0%占空比切换
当输出 0%占空比时, 因为 TGRA 大于 TGRB, 不会发生 TGRA 比较匹配。因此, 没有将 TGRC 的值传送到 TGRA 的时机。当从占空比 0%切换到 50%时, TGRC 和 TGRA 寄存器的值在 TGIB0 发生时直接修改 (图 3.77 中 (4))。
- TGRA 比较匹配在一个周期内发生两次
当从占空比 100%切换到 80%时, 将发生两次比较匹配, 包括与 TGRB 比较匹配同时发生的 TGRA 比较匹配 (图 3.77 中 (5))。
要从占空比 80%切换到 50%, 请将 TGRA 设置为大于当前值的值。
这将导致在一个周期内发生两次 TGRA 比较匹配 (图 3.77 中 (6))。
因为第二次 TGRA 比较匹配发生在高输出期间, 所以波形不会改变。(图 3.77 中 (7))。

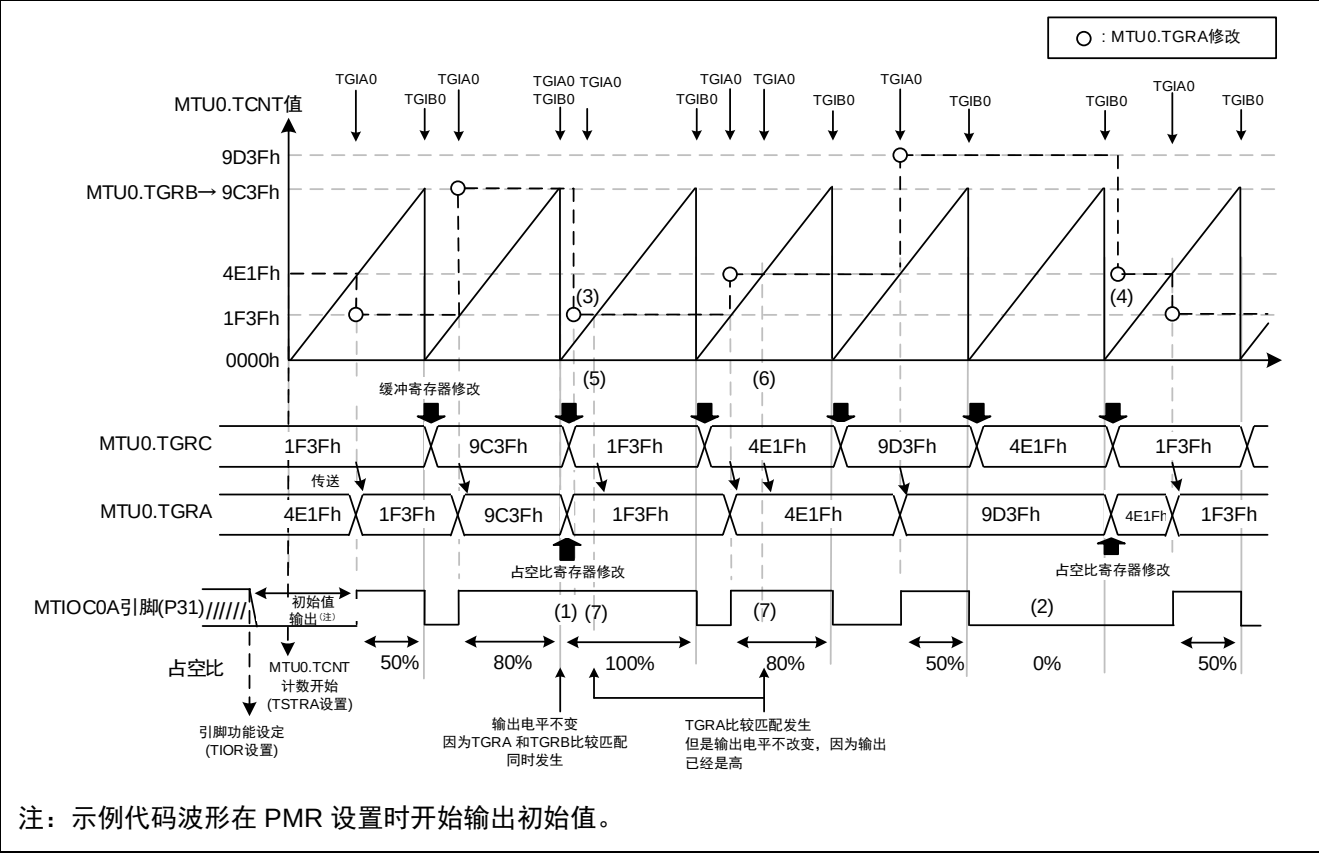


图 3.77 示例代码运行

3.12.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.15 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU0
运行	PWM 模式 1
资源	MTU0

Components

type filter text

Startup

Generic

r_bsp

Drivers

Timers

Config_MTU0

Configure

Synchronous mode setting

☐ Include this channel in the synchronous operation

TCNT0 counter setting

Counter clear source

TGRB0 compare match (Use TGRB0 as a cycle register)

Counter clock selection

PCLK/4

Rising edge

External clock pin setting

☐ Enable the noise filter for MTCLKA pin

☐ Enable the noise filter for MTCLKC pin

Noise filter clock selection

PCLK

General register setting

TGRC0

Buffer register of TGRA

Buffer transfer when compare match A occurs

TGRD0

Output compare register

Buffer transfer when compare match B occurs

TGRF0

Output compare register

Output setting

MTIOC0A pin

Output initial 0, 1 at compare match

When TGRB compare match

0 output from MTIOC0A pin

MTIOC0C pin

Output disabled

When TGRD compare match

0 output from MTIOC0C pin

PWM output setting

PWM period

1

ms

(Actual value: 1)

TGRA initial value

19999

TGRB initial value

39999

TGRC initial value

7999

TGRD initial value

100

TGRE initial value

100

TGRF initial value

100

A/D converter start trigger setting

☐ Enable start request on TGRA compare match (MTU0 TRGAN signal)

☐ Enable start request on TGRE compare match (TRGON signal)

Interrupt setting

☐ Enable TGRA compare match interrupt (TGIA0)

Priority

Level 15 (highest)

☒ Enable TGRB compare match interrupt (TGIB0)

Priority

Level 15 (highest)

☐ Enable TGRC compare match interrupt (TGIC0)

Priority

Level 15 (highest)

☐ Enable TGRD compare match interrupt (TGID0)

Priority

Level 15 (highest)

☐ Enable TGRE compare match interrupt (TGIE0)

Priority

Level 15 (highest)

☐ Enable TGRF compare match interrupt (TGIF0)

Priority

Level 15 (highest)

☐ Enable overflow interrupt (TCIV0)

Priority

Level 15 (highest)

A/D conversion start request frame synchronization signal setting

☐ AD5M0 pin

Source

Source not selected

☐ AD5M1 pin

Source

Source not selected

Overview

Board

Clocks

System

Components

Pins

Interrupts

定时器计数器清除源 = TGRB0比较匹配

定时器计数时钟 = 40MHz(PCLKC/4)

使用TGRC作为TGRA的缓冲寄存器
TGRA比较匹配时传送缓冲

初始输出值 = 低
TGRA比较匹配时输出高

TGRB比较匹配时输出低

载波周期 = 1ms

TGRA初始值设置
TGRC初始值设置

使能TGRB比较匹配中断

图 3.78 MTU0 设置

3.12.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

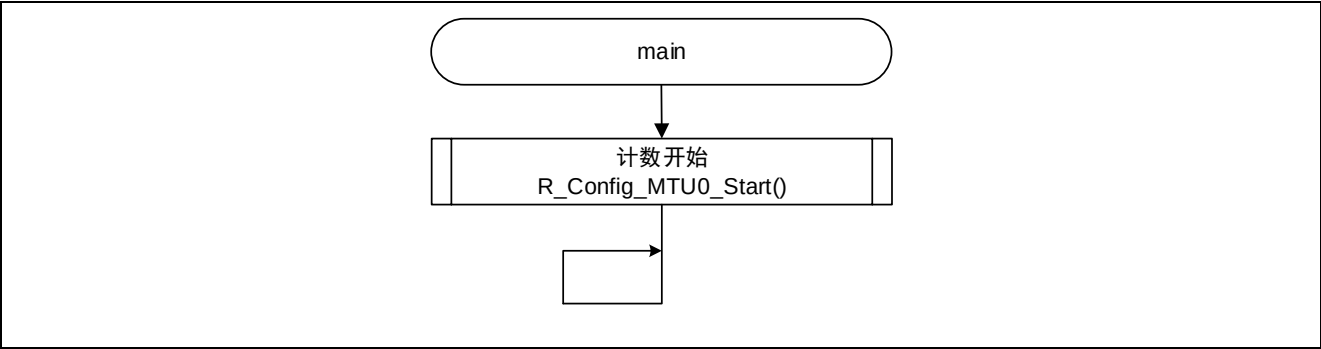


图 3.79 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU0_Create_UserInit 中初始化变量。此函数被 R_Config_MTU0_Create 函数调用。

此函数初始化此示例代码中使用的以下变量。

- s_duty_list_counter: 用于从占空比列表中读取的计数器变量
- s_duty_prv: 用于保存上一次 TGRA 寄存器值的变量

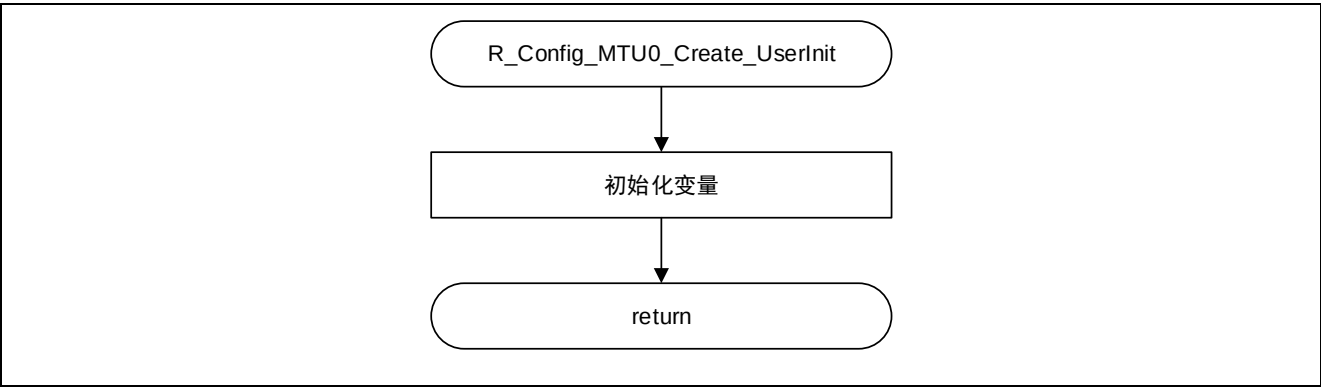


图 3.80 用户初始化函数

当占空比由 100%或者 0%切换时，TGIB0 中断处理函数直接设置 TGRA 寄存器的值。TGRC 寄存器的值每次被更改。

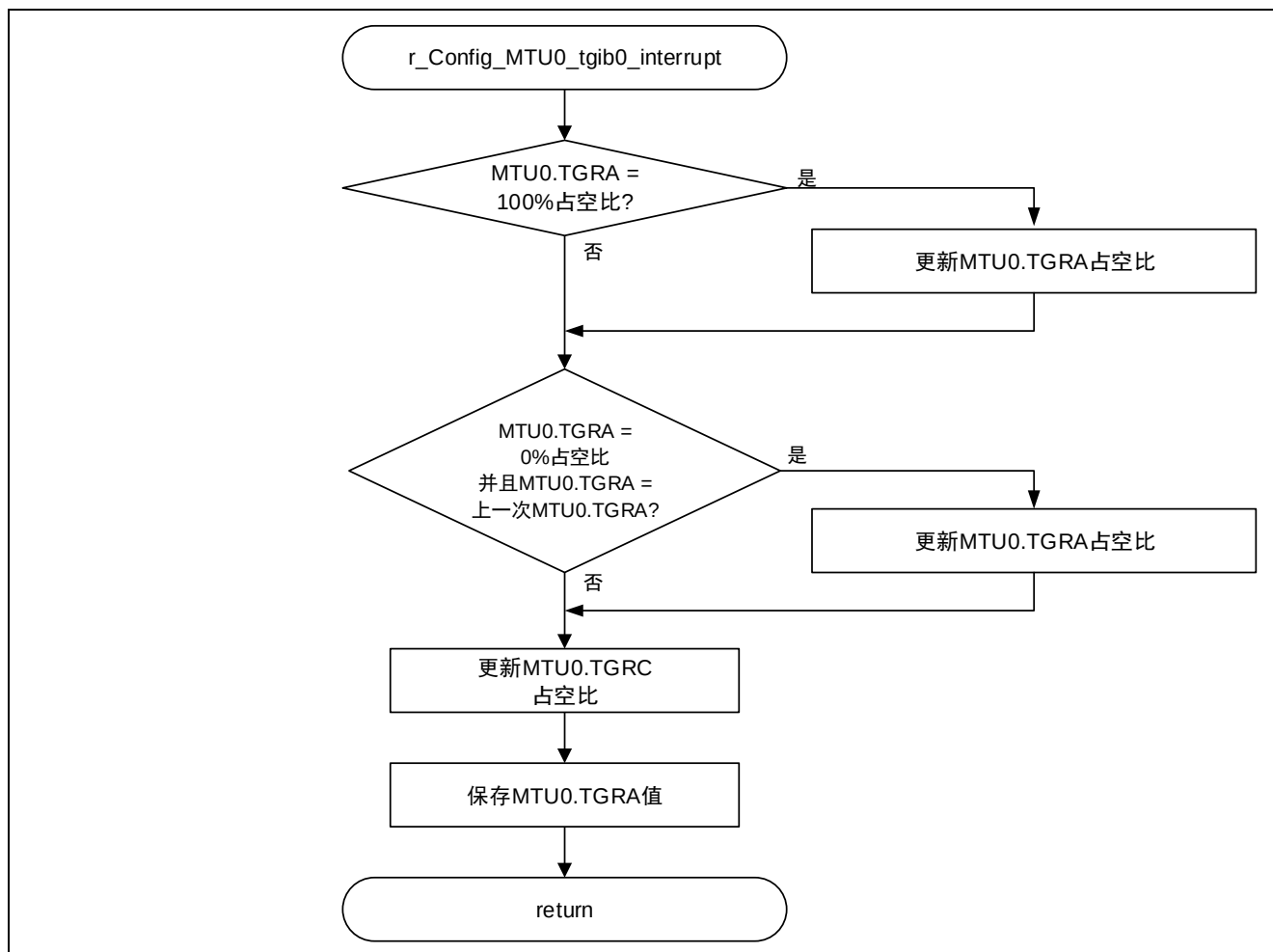


图 3.81 TGIB0 中断处理函数

3.12.5 相关运行

3.12.5.1 占空比 0%到 100%（无缓冲器）

下图显示了在不使用缓冲器的情况下，占空比为 0%到 100%的 PWM 波形输出示例，如图 3.69 所示。

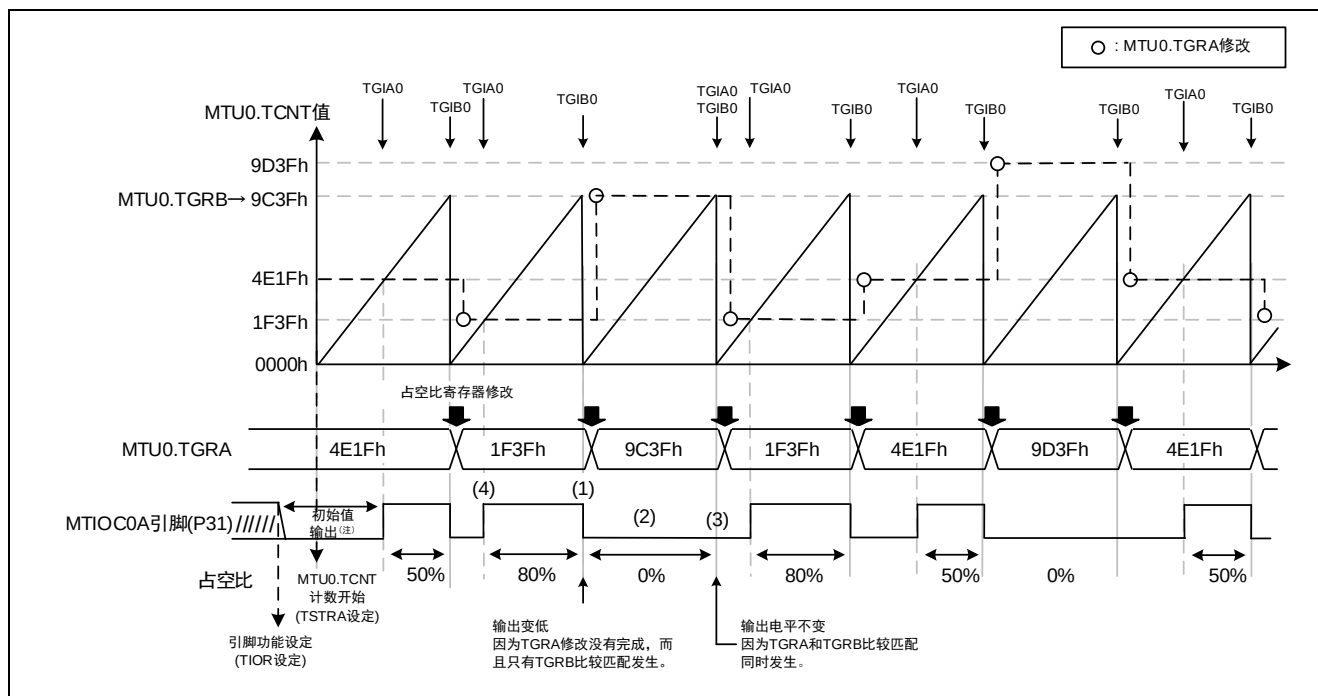


图 3.82 占空比 0%到 100%的 PWM 输出 (无缓冲器)

下面是使用或者不使用缓冲器时输出占空比 0%到 100%的 PWM 波形的不同。

- 100%占空比输出

当输出 100% 占空比时，设置 TGRA 为与 TGRB 相同的值。

如图 3.82 中 (1) 所示时间, 当 TGIB0 发生时, TGRA 值被修改, 并且输出变为低, 因为只有 TGRB 比较匹配发生。因此, 即使希望输出 100%, 输出仍为 0% (图 3.82 中 (2))。此外, 当 TGRA 和 TGRB 比较匹配同时发生时, 因为输出没有改变, 所以保持低 (图 3.82 中 (3))。

图 3.84 显示了如何在指定时间 (4) 修改 TGRA, 以避免图 3.82 中 (2) 的状况。

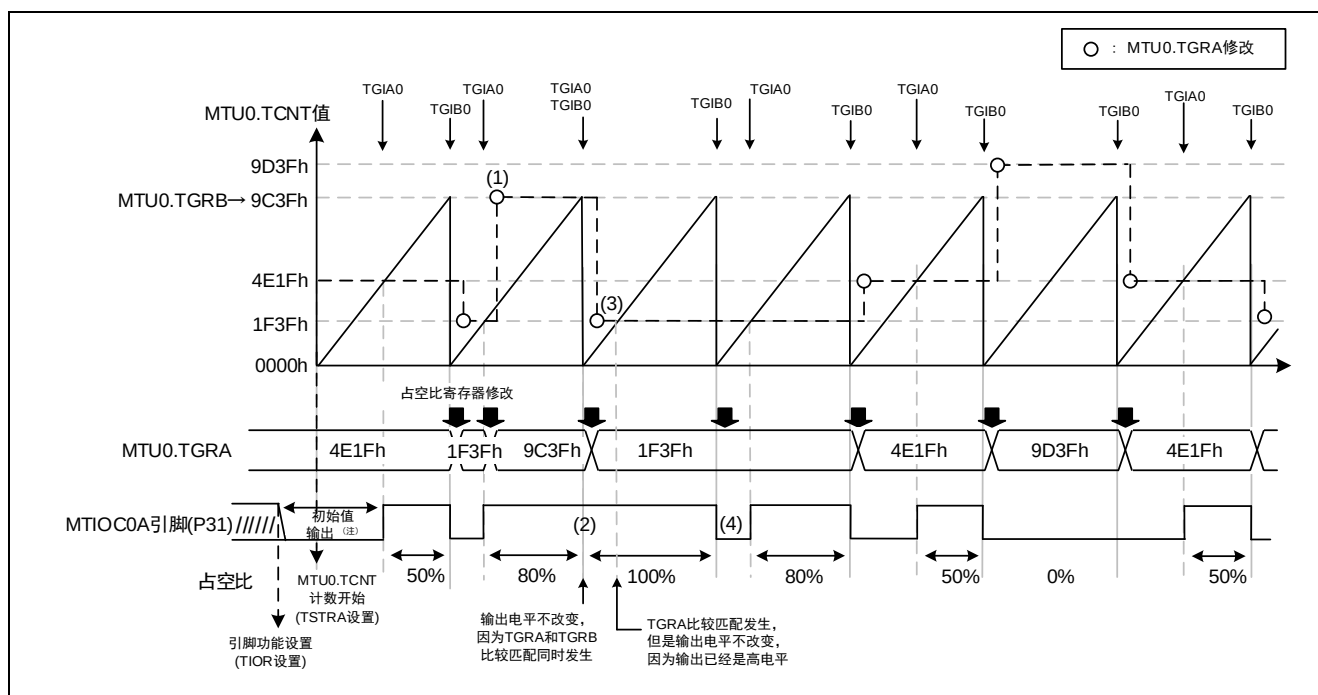


图 3.83 占空比 0%到 100%的 PWM 输出（无缓冲器）

- 100%占空比输出
当输出 100%占空比时，设置 TGRA 为与 TGRB 相同的值。
当 TGIA0 发生（图 3.83 中（1））时，修改 TGRA 的值。
当 TGRA 的值与 TGRB 的值相同时，两个比较匹配同时发生。因为在两个比较匹配同时发生时输出不改变，保持高电平并且输出 100%占空比（图 3.83 中（2））。
当 TGIB0 发生（图 3.83 中（3））时，TGRA 改变为 80%。
- 从 100%占空比切换
当 100%占空比 TGRB 比较匹配发生（图 3.83 中（4））时，引脚输出低。同时，TGRA 再次保持 80%。
- TGRA 比较匹配在一个周期内发生两次
当占空比从 100%切换到 80%，将发生两次比较匹配，包括与 TGRB 比较匹配同时发生的 TGRA 比较匹配（图 3.83 中（3））。
因为第二次 TGRA 比较匹配发生在高电平输出时，所以波形不改变。

3.12.6 使用时的注意事项

3.12.6.1 缓冲寄存器修改和比较匹配之间的竞争

当缓冲传送时间设置为比较匹配时，如果发生比较匹配，则修改前的缓冲寄存器中的数据将传送到占空比寄存器。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.7 缓冲寄存器写操作和比较匹配之间的竞争。

3.12.6.2 缓冲寄存器修改延迟

在此示例代码中，当发生 TGRA 比较匹配时，将执行从缓冲寄存器 TGRC 到 TGRA 的传送。必须在下一次比较匹配发生之前完成 TGRC 值的修改。

如果缓冲寄存器修改延迟，TGRA 的值可能不能在下一个周期 TGRA 比较匹配发生时被更新。在这种情况下，考虑在计数器清除发生时缓冲传送。

3.12.6.3 设置一个大于占空比寄存器的值

在此示例代码中，当 MTU0.TGRB 的比较匹配中断（TGIB0）发生时，修改缓冲寄存器 TGRC，并在发生 TGRA 比较匹配时将值从 TGRC 传送到 TGRA。如果在发生 TGRA 比较匹配后，在 TGRA 中设置的值大于当前 TGRA 的值，则在一个周期内可能会发生两次 TGRA 比较匹配。

有关详细信息，请参考图 3.69 中（7）。

3.12.6.4 从 0%占空比切换

在此示例代码中，当 MTU0.TGRB 的比较匹配中断（TGIB0）发生时，修改缓冲寄存器 TGRC，并在发生 TGRA 比较匹配时将值从 TGRC 传送到 TGRA。如果在 TGRA 中设置了大于 TGRB 的值，并且输出 0%占空比，则不会发生 TGRA 比较匹配，因为在达到 TGRA 值之前计数器被清除。

从 0%占空比切换时，需要直接在 TGRA 寄存器中设置该值。

有关详细信息，请参考图 3.69 中（4）。

3.13 PWM 模式 1 占空比 0%和 100%

- 目标示例代码文件名: r01an5995_rx66t_mtu3_pwm1_0to100.zip

3.13.1 概要

MTU PWM 模式可用于根据 TGR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用 PWM 模式 1 并重复占空比 0%和 100%之间交替的波形输出。

使用占空比寄存器 TGRA, 波形输出不能在每个周期占空比 0%和 100%之间交替, 因此, 当周期寄存器 TGRB 比较匹配发生时, 通过将输出切换为高和低, 输出占空比 0%和 100%的波形。用户应意识到与其它进程的潜在冲突, 中断延迟等, 并在使用前对产品进行彻底评估样本代码程序是否适应用户系统, 因为 TIOR 寄存器设置在定时器运行期间被修改。

注意当 TGRB 比较匹配发生时, 通过设置输出翻转, 可以容易地实现占空比 0%和 100%之间的交替。在此示例代码中, TIOR 寄存器被修改为允许与 0%或 100%以外的占空比组合。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用 PWM 模式 1
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 40MHz (PCLKC/4)
- 使用 TGRB 作为周期寄存器
 - 定时器计数器清除源 = TGRB 比较匹配
 - TGRB 比较匹配时输出高
- 使用 TGRA 作为占空比寄存器
 - TGRA 比较匹配时输出高
- 使用缓冲寄存器
 - 使用 TGRC 作为 TGRA 的缓冲寄存器
 - TGRA 比较匹配发生时缓冲传送
- 每个周期的占空比改变
 - 在 TGRB 比较匹配中断时设置 TGRB 比较匹配输出
 - 关于占空比改变的时间的详细信息, 请参考图 3.85

在 Smart Configurator 中设置。
设置方法请参考章节 3.11.3。

此示例代码的 PWM 模式 1 输出如下所示。

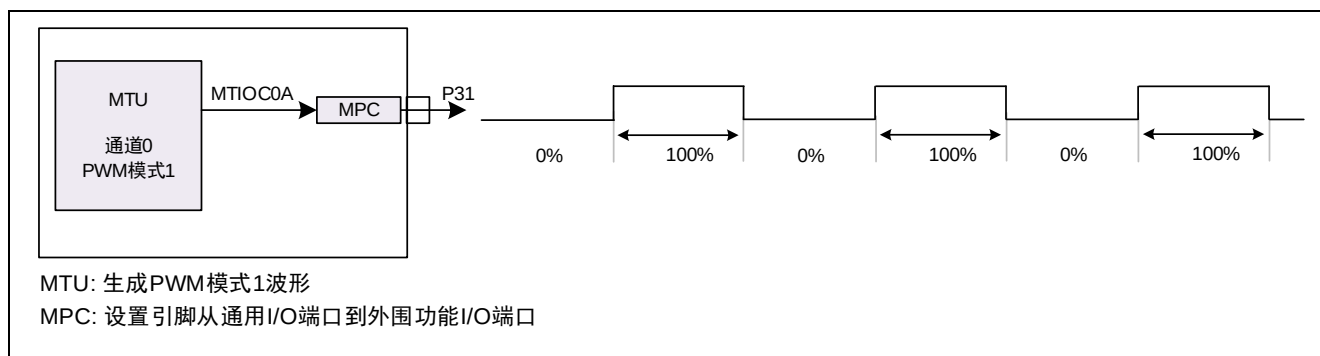


图 3.84 PWM 模式 1 输出

3.13.2 运行说明

示例代码的运行如图 3.85 所示。为了确保输出波形不因占空比寄存器 TGRA 的比较匹配而改变，TGRA 和 TGRC 被设置为大于周期寄存器 TGRB 的值。

如下所述，通过在 TGRB 比较匹配中断（TGIB0）处修改 TIOR 寄存器的值，占空比可以在 0% 和 100%之间交替。

- MTU0.TIORH.IOB = 5h: 在 TGRB 比较匹配时输出低
- MTU0.TIORH.IOB = 6h: 在 TGRB 比较匹配时输出高

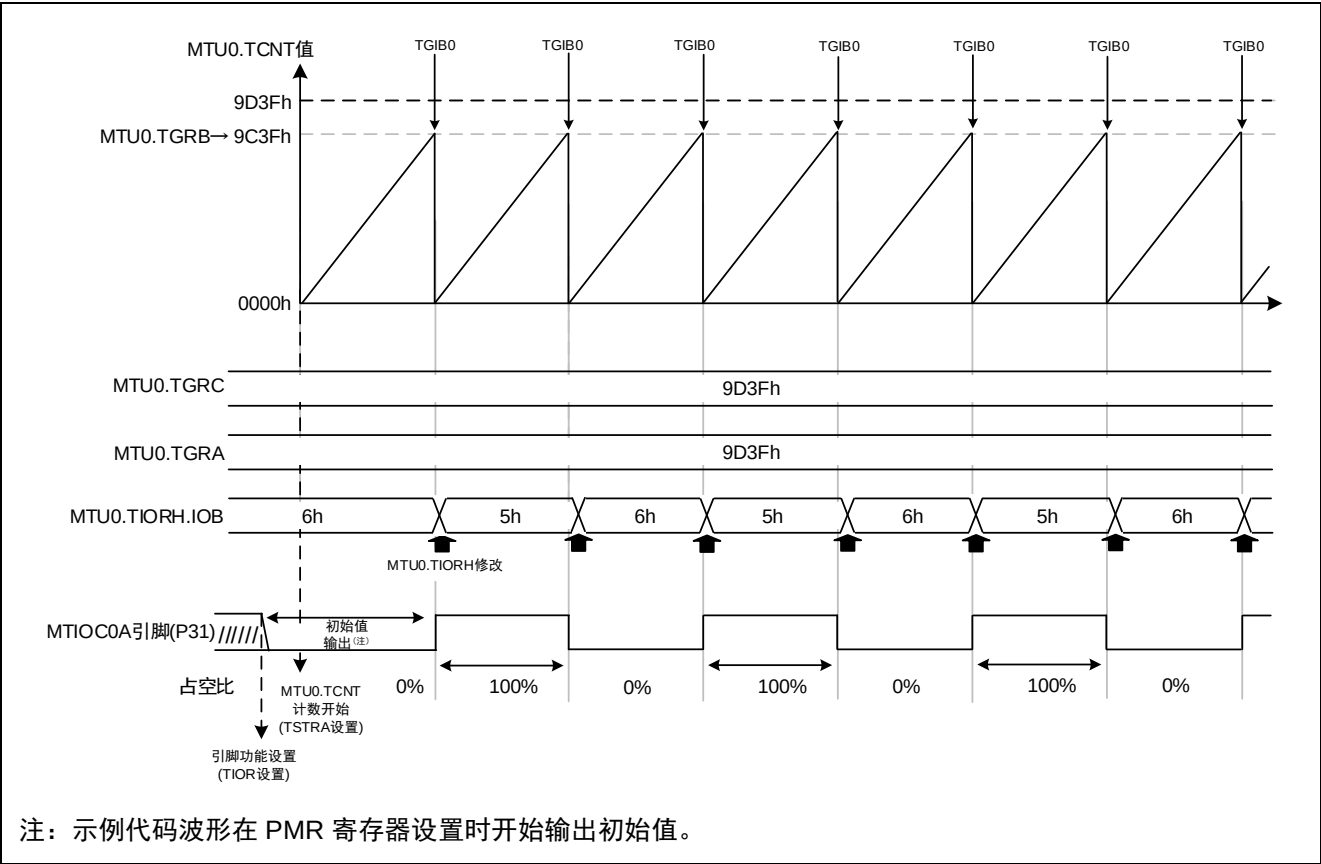


图 3.85 示例代码运行

3.13.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考章节 3.1.4 添加组件。

表 3.16 添加组件

项目	描述
组件	PWM 模式定时器
配置名称	Config_MTU0
运行	PWM 模式 1
资源	MTU0

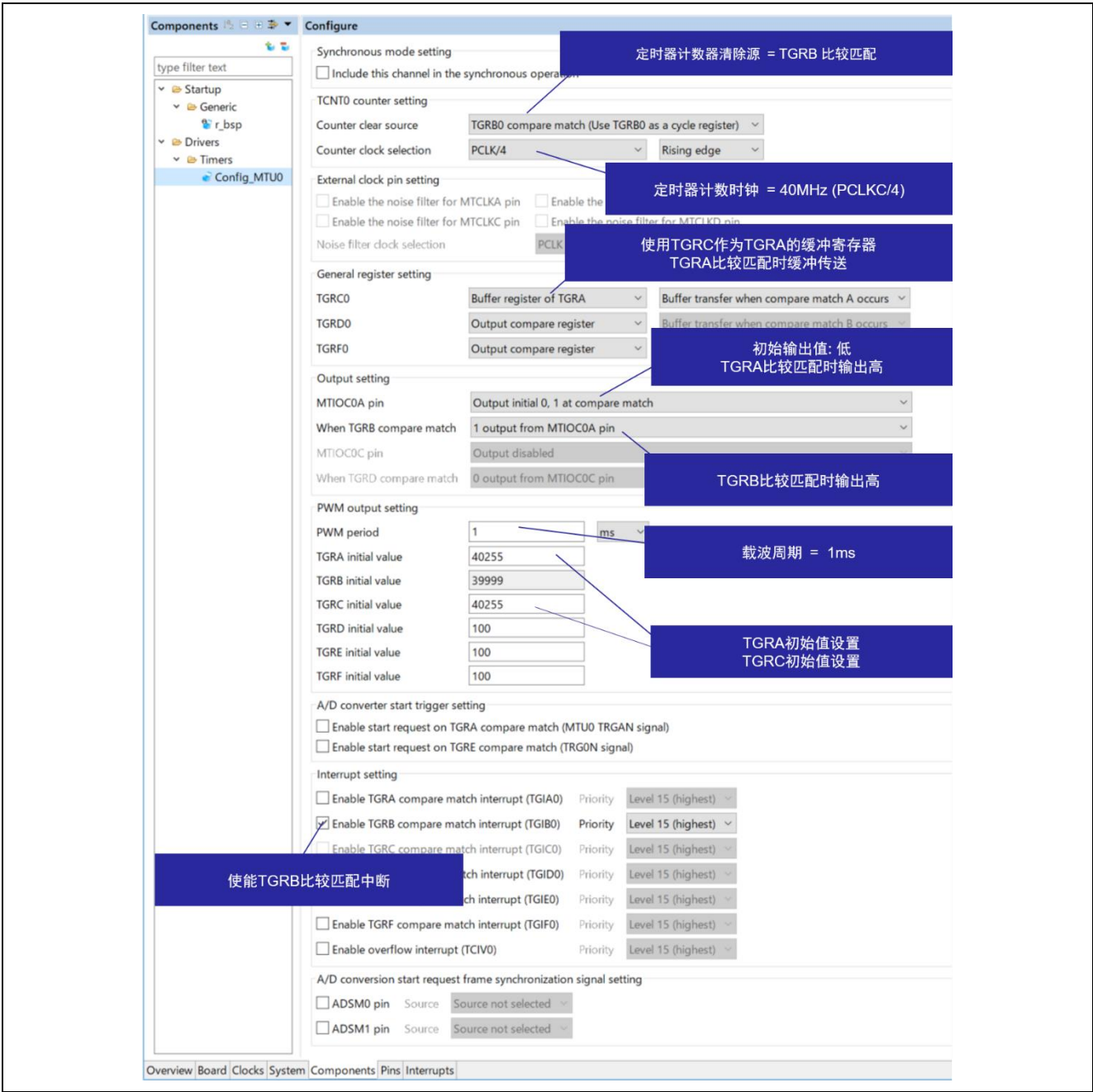


图 3.86 MTU0 设置

3.13.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

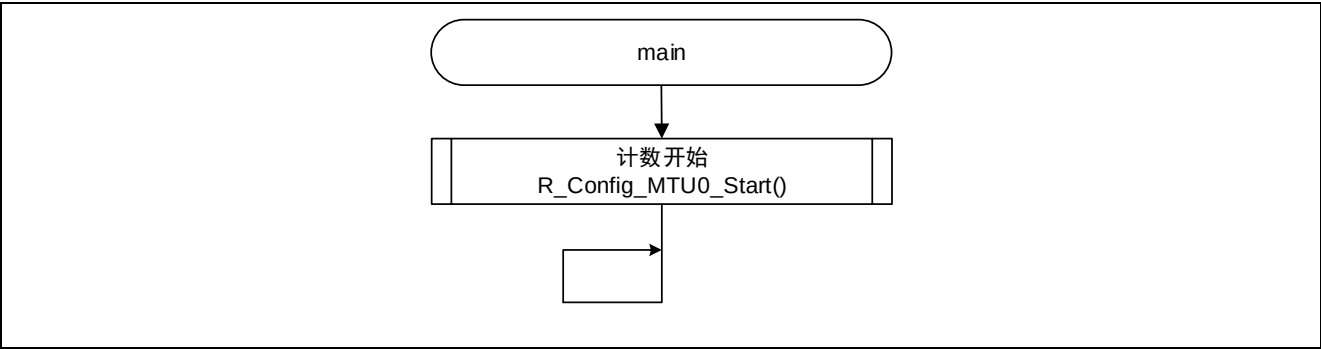


图 3.87 主函数

当 TGRB 比较匹配发生时，TGIB0 中断处理函数更改 TIOR 寄存器的值并更改输出设置。

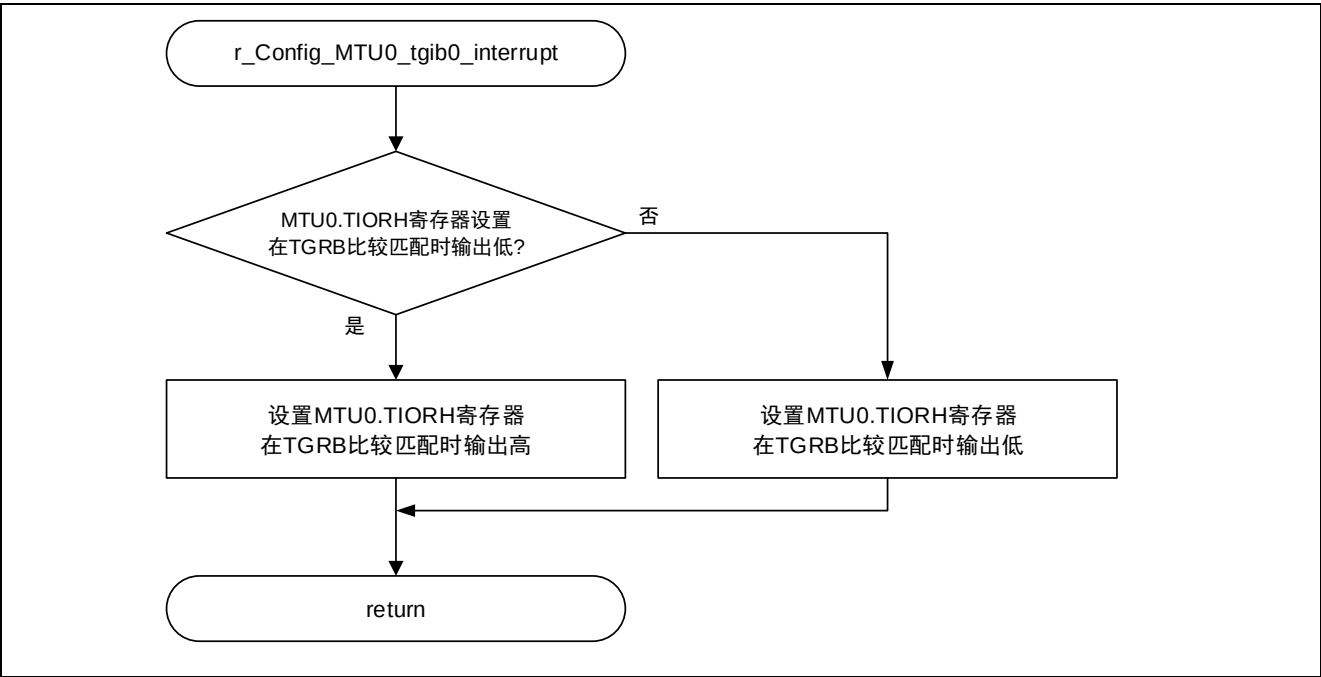


图 3.88 TGIB0 中断处理函数

3.13.5 相关运行

3.13.5.1 占空比 0%和 100%（无缓冲器）

下面显示了占空比为 0%和 100%的 PWM 波形输出示例，如图 3.85 所示，不使用缓冲器。因为缓冲寄存器和占空比寄存器不用于切换占空比，所以程序和使用寄存器的相同。

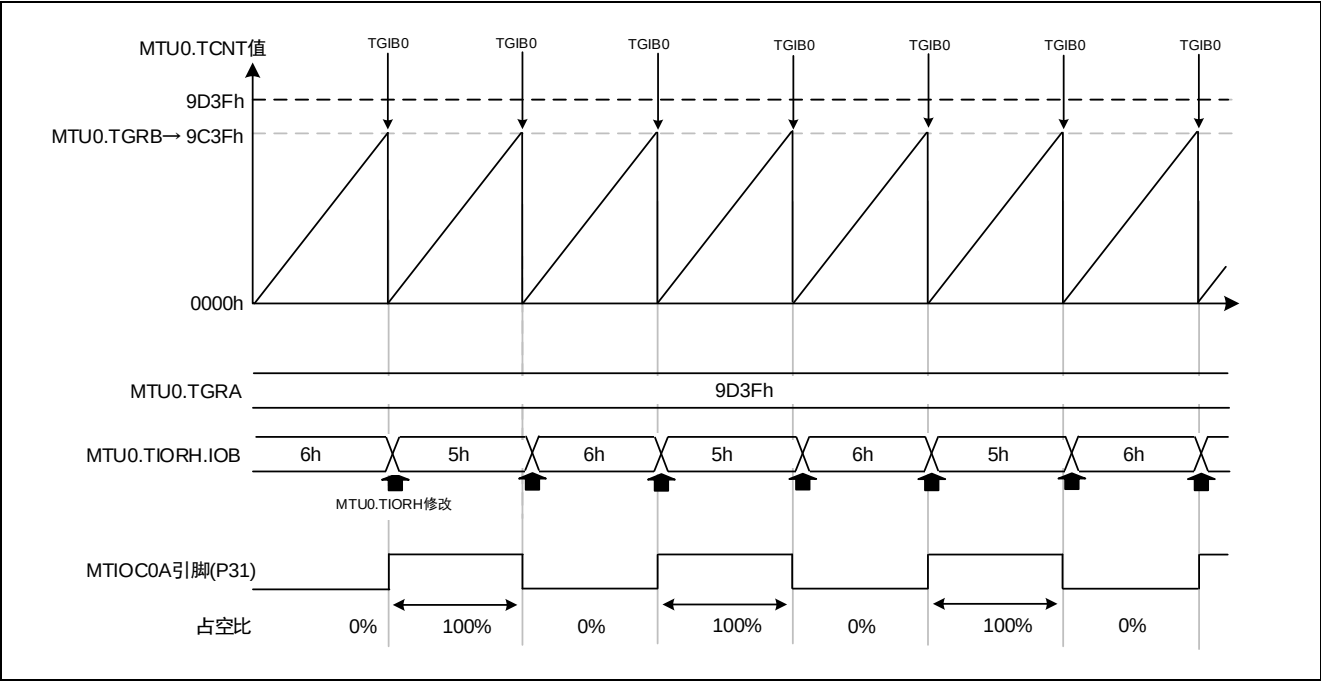


图 3.89 占空比 0%和 100%的 PWM 输出（无缓冲器）

3.13.5.2 在初始值低时保持 0%后更改占空比到 100%

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例。

当 TIORH.IOA 设置为 2h 以及 TIORH.IOB 设置为 5h 时，MITOC0A 引脚在运行开始后输出初始值低和 0%占空比。然后，当 TIORH.IOB 设置为 6h 时（下图中（1）），引脚在 TGRB 比较匹配时输出高；在下一个 TGIB0 发生时，当 TGRA = TGRB = TGRB 并且 TIORH.IOB 设置为 5h 时（下图中（2）），占空比变为 100%。

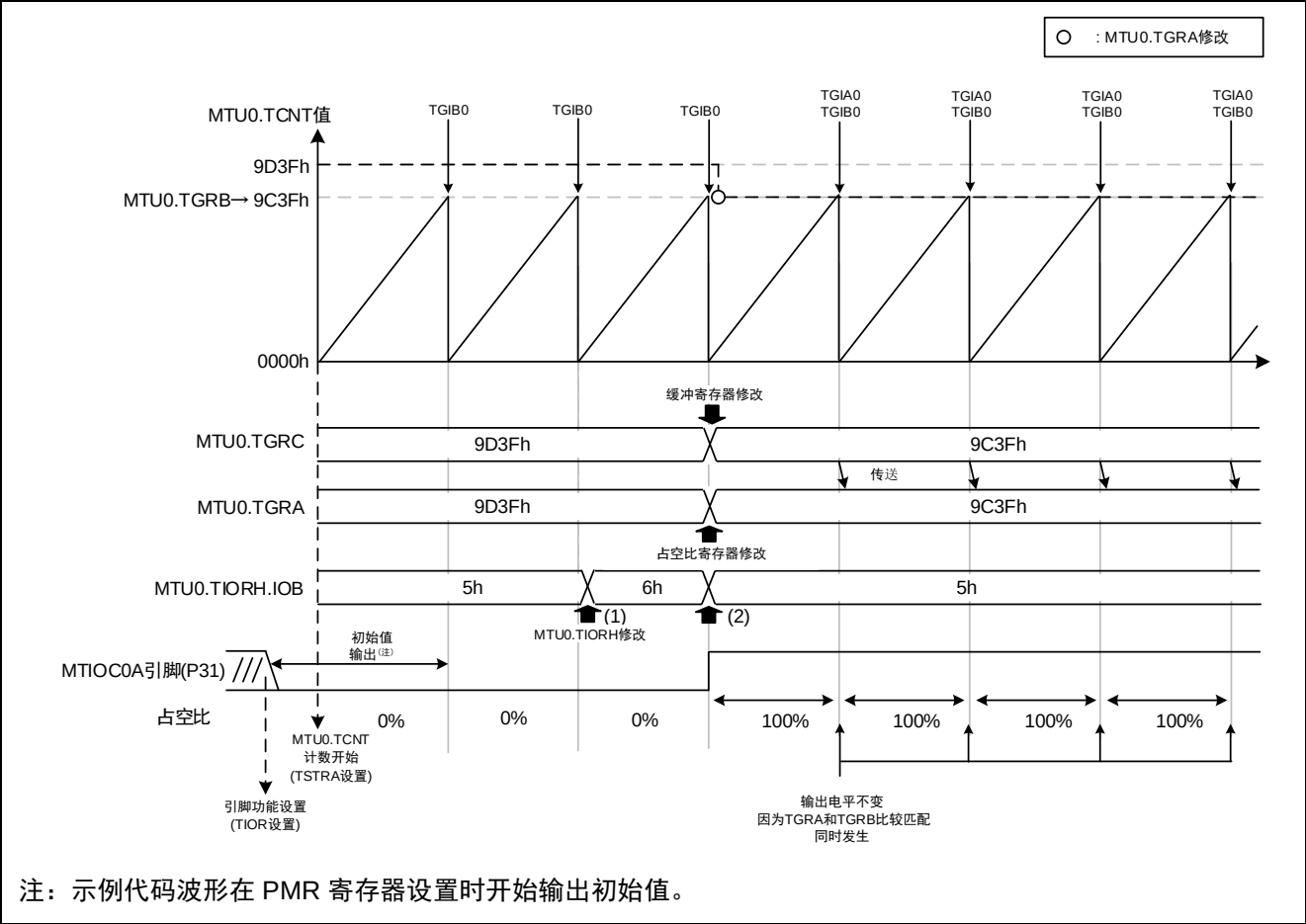


图 3.90 在保持 0%后更改占空比到 100%（有缓冲器）

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例，如图 3.90 所示，不使用缓冲器。程序和使用寄存器相同。

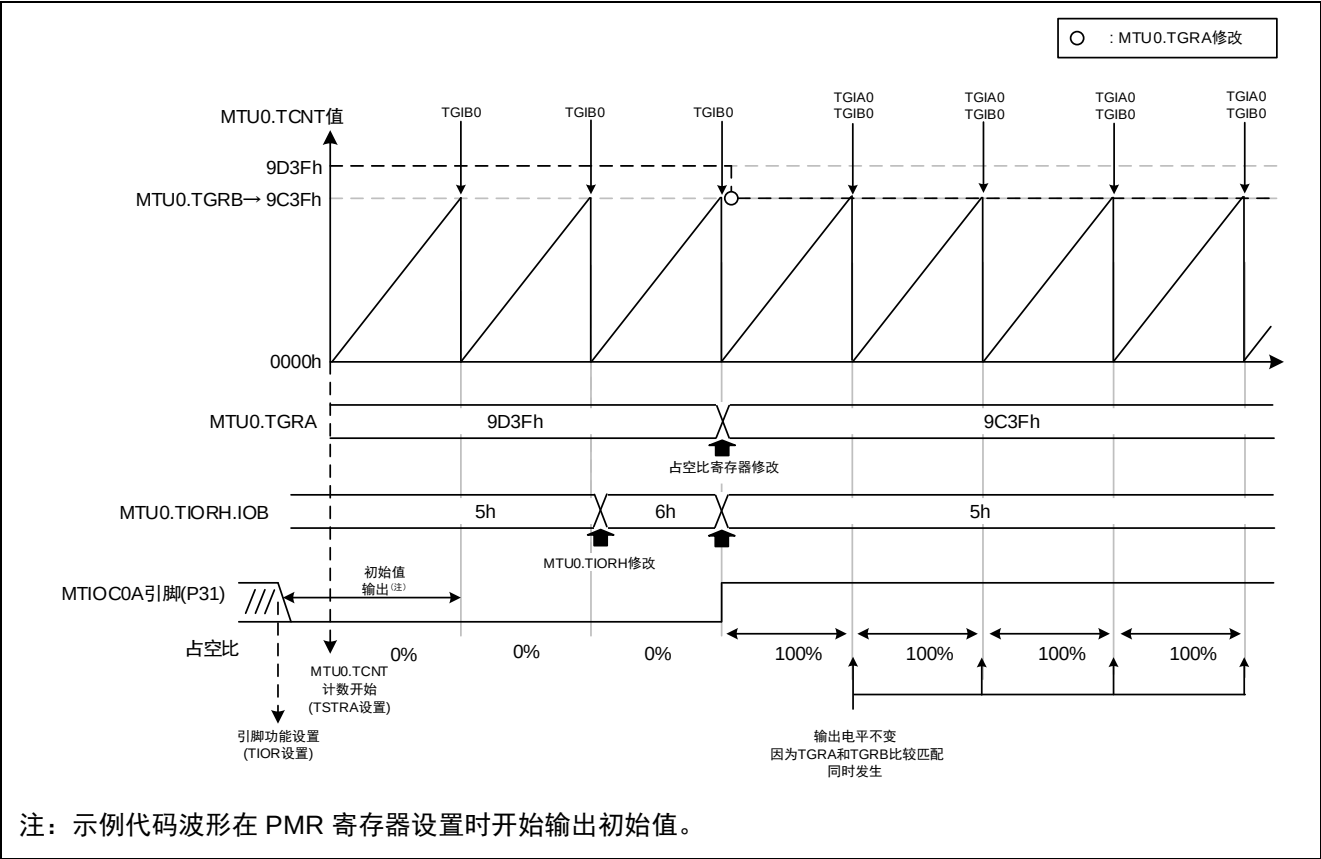


图 3.91 在保持 0%后更改占空比到 100%（无缓冲器）

3.13.5.3 在初始值低时保持 100%后更改占空比到 0%

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例。

当 TIORH.IOA 设置为 2h 以及 TIORH.IOB 设置为 5h 时，MITOC0A 引脚初始设置为低。TGRA 初始值 0000h，当 TCNT 从 0000h 计数到 0001h 时，发生比较匹配并输出高，从而产生 100%占空比输出（下图中（1））。在计数清除时，与 TGRB 相同的值从 TGRC 传送到 TGRA（下图中（2））。当在任意时间将 TGRC 设置为大于 TGRB 的值时，在 2 个周期后输出低以及 0%占空比（下图中（3））。

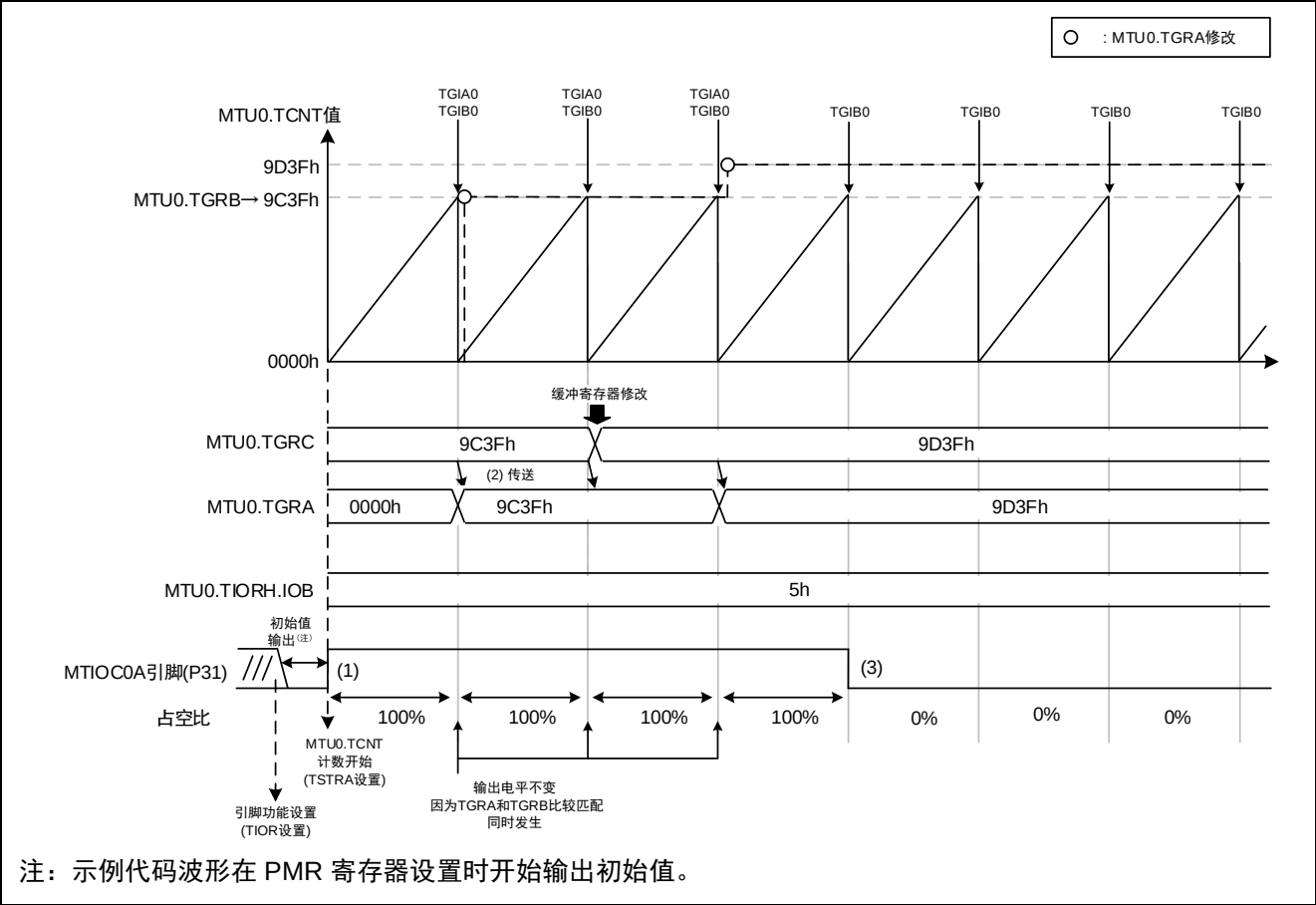


图 3.92 在保持 100%后更改占空比到 0%（有缓冲器）

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例，如图 3.92 所示，不使用缓冲器。程序和使用寄存器相同，但将 TGRA 的值设定为大于 TGRB 的值后，在下个周期输出 0%占空比。

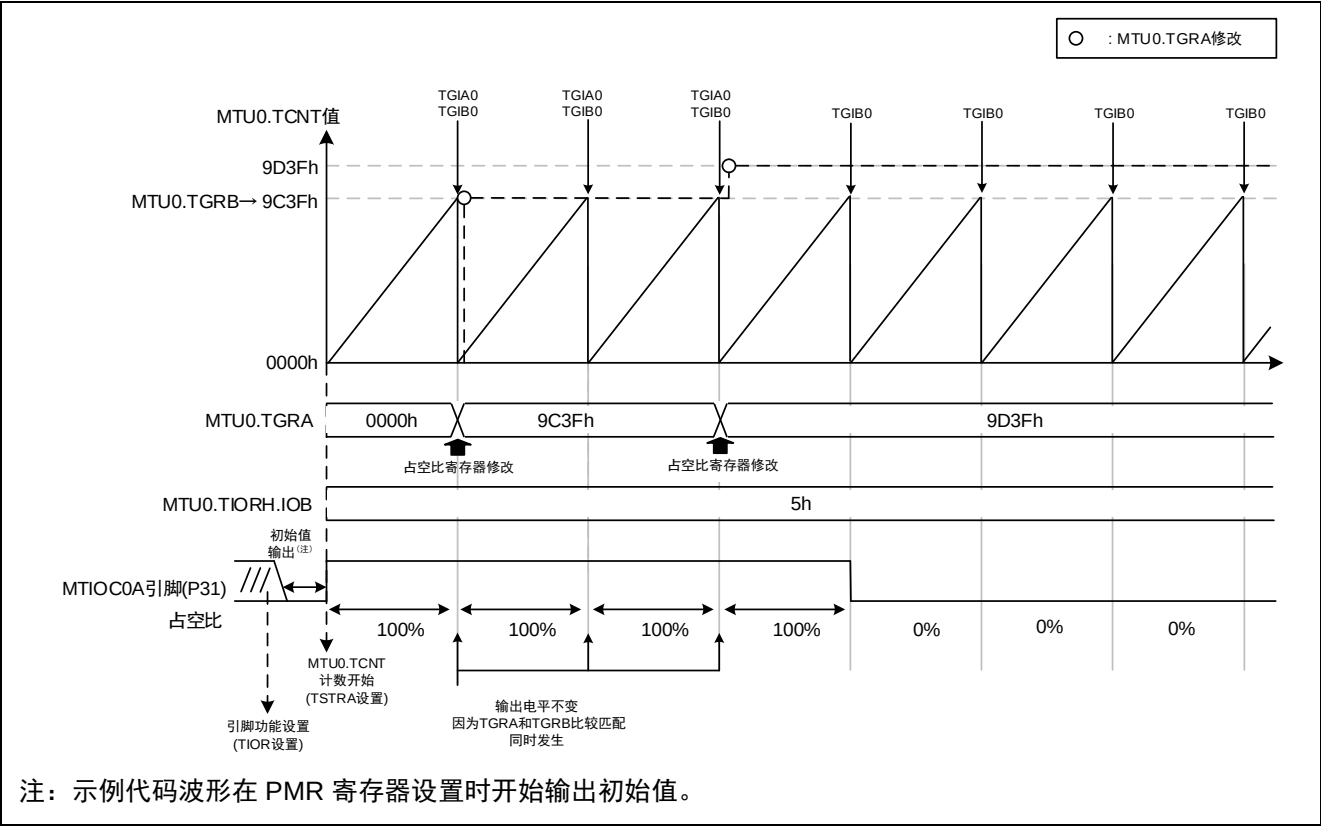


图 3.93 在保持 100%后更改占空比到 0%（无缓冲器）

3.13.5.4 在初始值高时保持 0%后更改占空比到 100%

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例。

当 TIORH.IOA 设置为 5h 以及 TIORH.IOB 设置为 5h 时，MITOC0A 引脚初始设置为低。TGRA 初始值 0000h，当 TCNT 从 0000h 计数到 0001h 时，发生比较匹配并输出低，从而产生 0%占空比输出（下图中（1））。然后，当 TIORH.IOB 设置为 6h 时（下图中（2）），引脚在 TGRB 比较匹配时输出高；在下一个 TGIB0 发生时，当 TGRA = TGRC = TGRB 并且 TIORH.IOB 设置为 5h 时（下图中（3）），占空比变为 100%。

此外，当 TIORH.IOA 在（2）指定的时间设置为 2h 时，可以输出高有效 PWM。

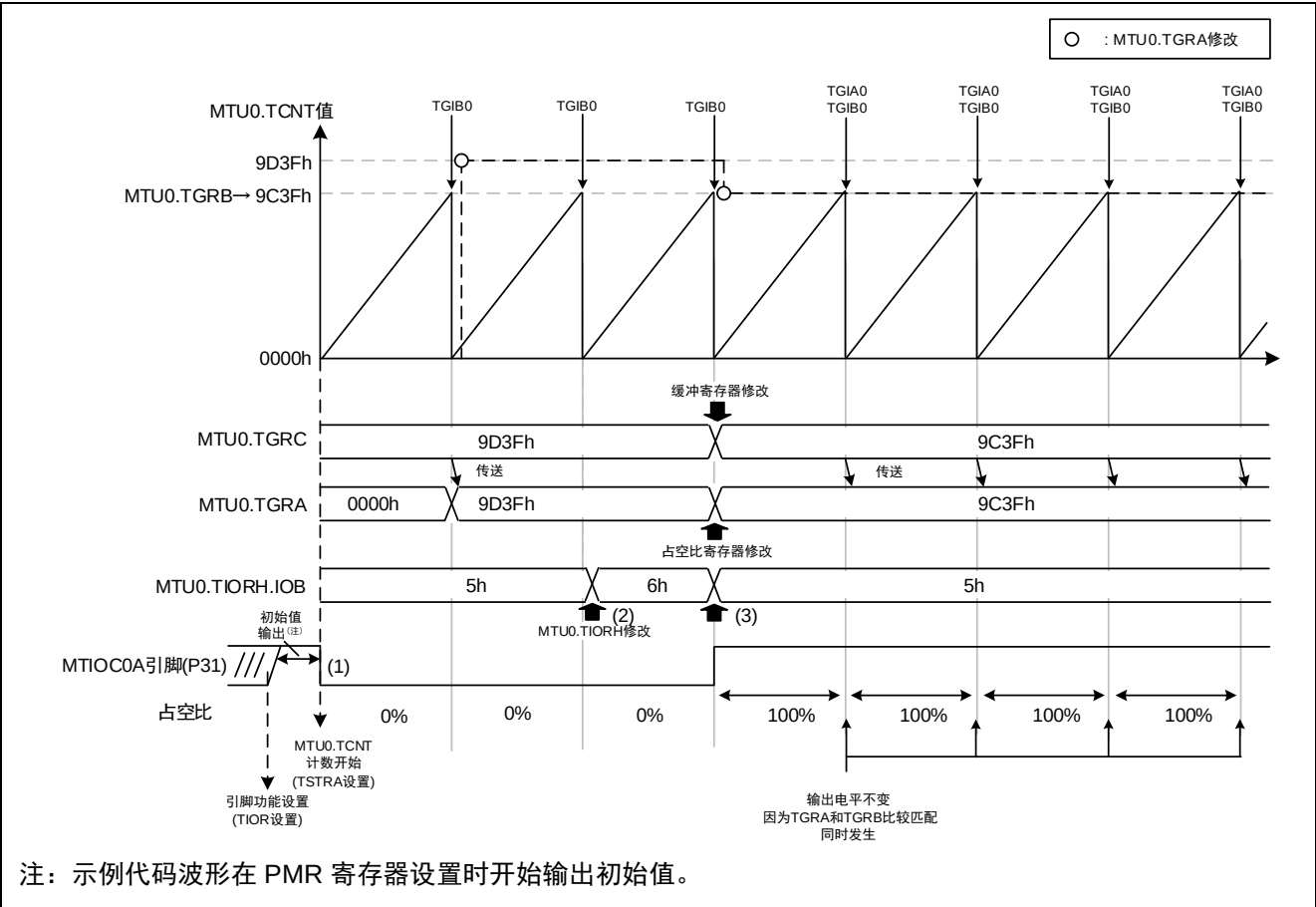


图 3.94 在保持 0%后更改占空比到 100%（有缓冲器）

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例，如图 3.94 所示，不使用缓冲器。程序和使用寄存器相同。

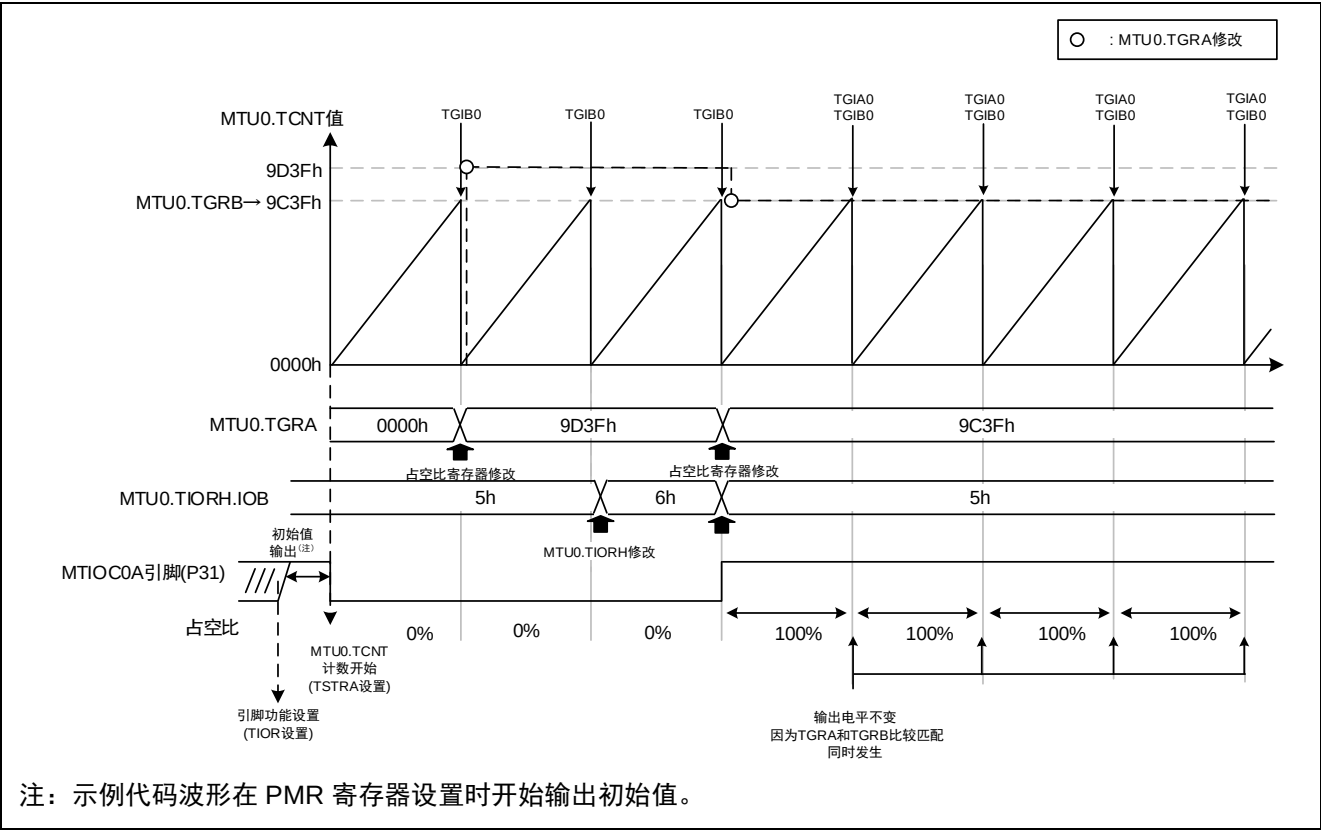


图 3.95 在保持 0%后更改占空比到 100%（无缓冲器）

3.13.5.5 在初始值高时保持 100%后更改占空比到 0%

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例。

当 TIORH.IOA 设置为 6h 以及 TIORH.IOB 设置为 5h 时，MITOC0A 引脚在运行开始后输出初始值高和 100%占空比。当在任意时间将 TGRC 设置为大于 TGRB 的值时，在 2 个周期后输出低并输出 0%占空比（下图中（1））。

此外，当 TIORH.IOA 在运行开始后任意时间设置为 2h 时，可以输出高有效 PWM。

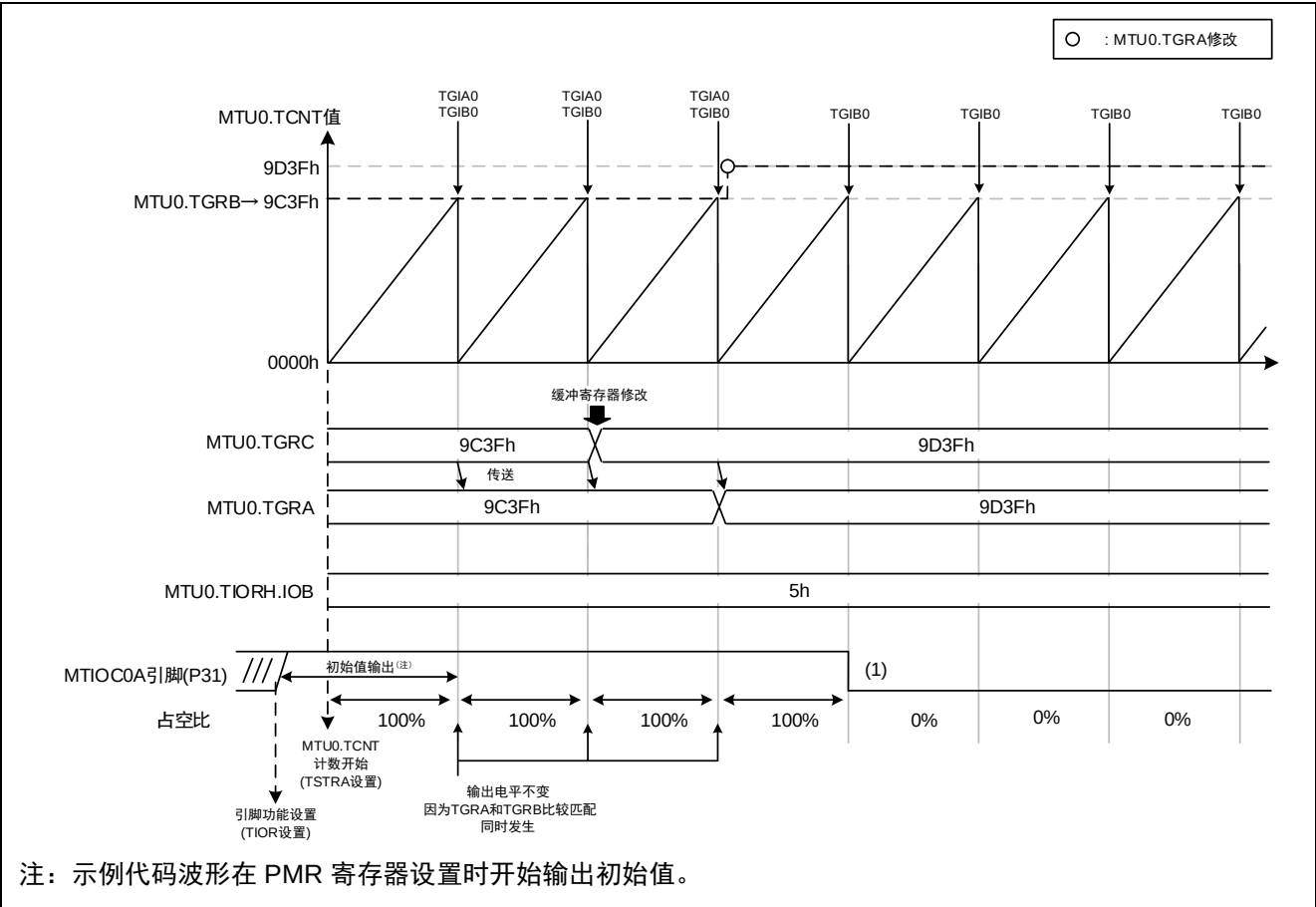


图 3.96 在保持 100%后更改占空比到 0%（有缓冲器）

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例，如图 3.96 所示，不使用缓冲器。程序和使用寄存器相同，但将 TGRA 的值设定为大于 TGRB 的值后，在下个周期输出 0%占空比。

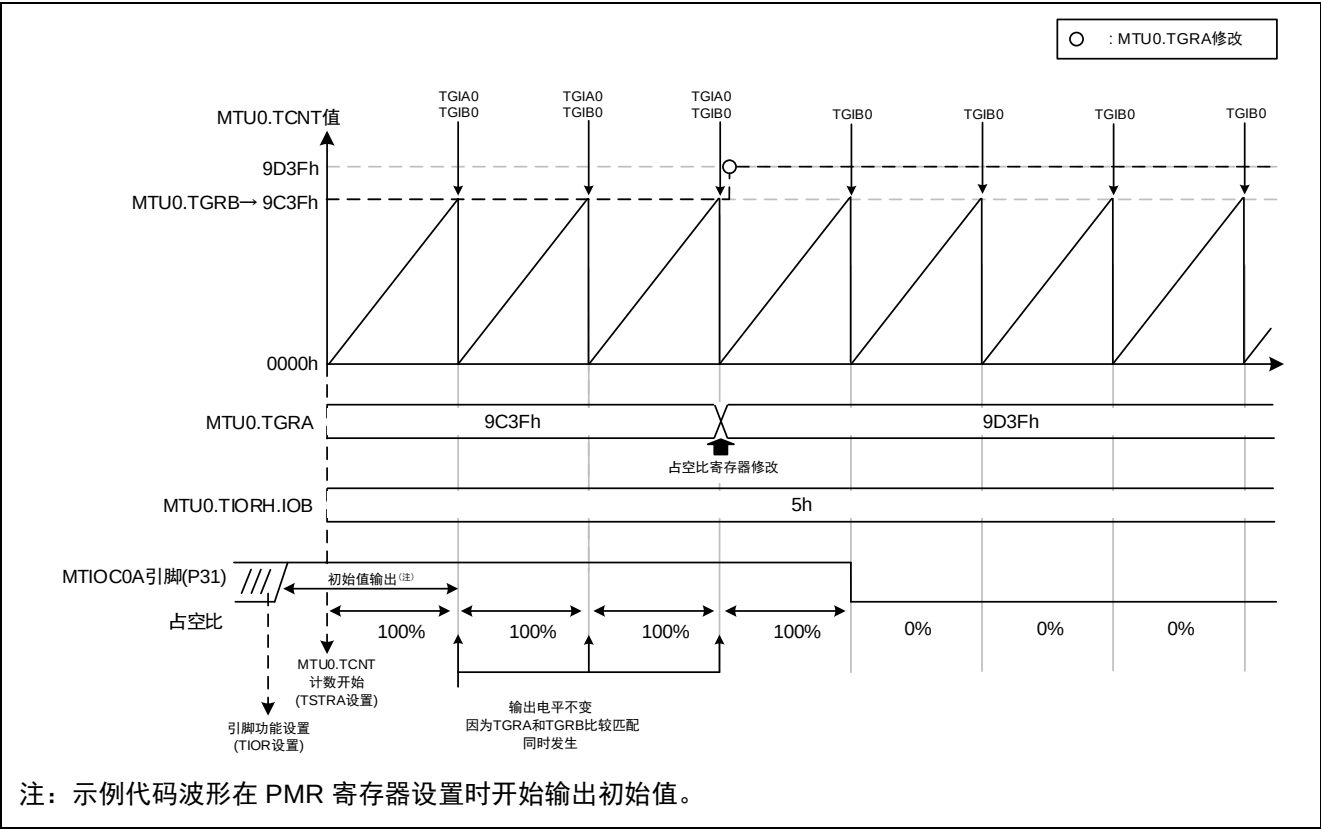


图 3.97 在保持 100%后更改占空比到 0%（无缓冲器）

3.13.6 使用时的注意事项

在此示例代码中，当产生 MTU0.TGRB 比较匹配中断（TGIB0）时，将执行修改 TIOR 寄存器的处理。假如将相同处理用于客户的系统，客户应当意识到与其它进程的潜在冲突，中断延迟等，并在使用前对产品进行彻底评估。

3.13.6.1 TIOR 寄存器修改延迟

如果在下一个 TGRB 比较匹配发生之前不能完成 TIOR 寄存器的修改，则无法输出所需要的占空比。

3.13.6.2 TIOR 寄存器修改运行和比较匹配之间的竞争

如果产生 MTU0.TGRB 比较匹配中断（TGIB0）并且在 TIOR 寄存器修改运行期间修改 TIOR 寄存器，则无法输出所需要的占空比。

3.13.6.3 修改 MTU3, MTU4, MTU6, MTU7 的 TIOR 寄存器的使用说明

当在 PWM 模式 1 下修改 MTU3, MTU4, MTU6 或 MTU7 的 TIOR 寄存器，将寄存器 TRWERA 和 TRWERB 的 RWE 设置为 1。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.20 定时器读写使能寄存器（TRWERA, TRWERB）。

3.13.6.4 缓冲寄存器修改运行和比较匹配之间的竞争

当缓冲传送时间被设置在比较匹配时，如果在缓冲寄存器写周期内发生比较匹配，则修改之前的缓冲寄存器中的数据被传送到占空比寄存器。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.7 缓冲寄存器写操作和比较匹配之间的竞争。

3.13.6.5 从 0%占空比切换

当 TGRA 设置为大于 TGRB 的值并且输出 0%占空比时，不会发生 TGRA 比较匹配，因为计数器在 TGRA 值达到之前被清除。

如果缓冲寄存器 TGRC 设置为在发生 TGRA 比较匹配时将值传送到 TGRA，并且占空比寄存器更改为从 0%占空比切换，则必须在 TGRA 寄存器中直接设置该值。

详情参考图 3.90。

3.13.6.6 当占空比寄存器设置为 0 时一个周期的输出

当占空比寄存器设置为 0 时，在发生周期寄存器比较匹配后输出一个周期的波形，然后将发生占空比寄存器比较匹配。

当每个周期在占空比 0%和 100%之间重复切换时，考虑通过更改 TIOR 寄存器进行切换，如示例代码中所做。

3.13.6.7 当占空比寄存器设置为与周期寄存器相同值时波形不发生改变

当占空比寄存器设置为与周期寄存器相同值时，两个寄存器同时发生比较匹配，波形不变。

当每个周期在占空比 0%和 100%之间重复切换时，考虑通过更改 TIOR 寄存器进行切换，如示例代码中所做。

3.14 互补 PWM 模式 占空比 0%到 100%

- 目标示例代码文件名: r01an5995_rx66t_mtu3_complementary_pwm_50to100.zip

3.14.1 概要

MTU 互补 PWM 模式可用于输出带死区时间的三相互补 PWM 波形。

此示例代码描述了在互补 PWM 模式 2（在波谷传送）下重复输出以下带死区时间的波形，包括占空比 0%和 100%。

- 占空比切换: 50% → 80% → 100% → 80% → 50% → 0% → . . .

基本操作是在 TCNT 计数器溢出发生时从缓冲寄存器传送到暂存器，并在下溢发生时从暂存器传送到比较寄存器，从而改变占空比。当周期寄存器 MTU3.TGRA 比较匹配发生时，缓冲寄存器被修改。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用互补 PWM 模式 2（在波谷传送）
- 使用通道 3 和 4
- 载波周期 = 1ms
- 死区时间 = 50us
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 设置 MTU3.TGRA 为 MTU3.TCNT 上限值
（1/2 载波周期 + 死区时间）
— MTIOC3A 引脚翻转输出设置
- 设置缓冲传送时间
— 在计数的波谷传送
- 初始输出值为低，有效电平为高
- 使用 MTU3.TGRB 作为 U 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用 MTU4.TGRA 作为 V 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用 MTU4.TGRB 作为 W 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用缓冲寄存器
— 使用 MTU3.TGRD 作为 MTU3.TGRB 的缓冲寄存器
— 使用 MTU4.TGRC 作为 MTU4.TGRA 的缓冲寄存器
— 使用 MTU4.TGRD 作为 MTU4.TGRB 的缓冲寄存器
— 有关缓冲寄存器初始值，请参考图 3.100
- 每个周期的占空比改变
— 在 MTU3.TGRA 比较匹配中断时改变占空比
— 关于占空比改变的时间的详细信息，请参考图 3.100

在 Smart Configurator 中设置。
设置方法请参考章节 3.14.3。

此示例代码的互补 PWM 模式输出如下所示。

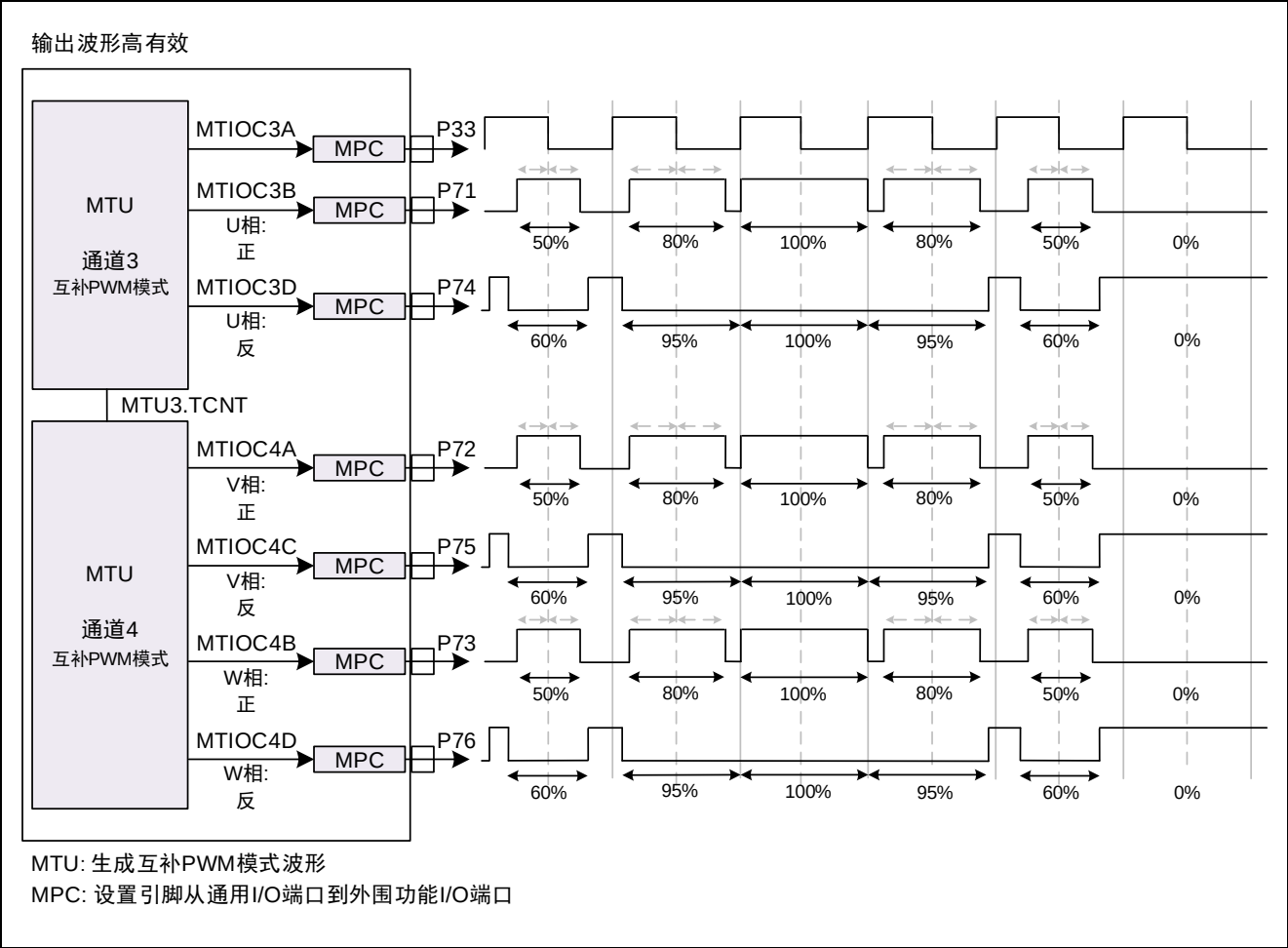


图 3.98 互补 PWM 模式输出

3.14.2 运行说明

示例代码的运行如图 3.100 所示。设置 MTU3.TGRA 寄存器为 MTU3.TCNT 的上限值（图 3.100 中（1）），通过在 MTU3.TGRA 寄存器的比较匹配中断（TGIA3）修改缓冲寄存器（MTU3.TGRD，MTU4.TGRC 和 MTU4.TGRD）的值，可以在每个周期改变占空比的设置。

在缓冲寄存器修改结束时写 MTU4.TGRD 来使能从缓冲寄存器到暂寄存器的数据传送。当在 Ta 间隔中写入 MTU4.TGRD，写入缓冲寄存器的数据立即传送到暂寄存器，但在此示例代码中，MTU4.TGRD 在 Tb1 间隔中写入，因此数据在 Tb1 间隔结束（图 3.100 中（2））之后传送到暂寄存器。

由于此示例代码使用在波谷传送的互补 PWM 模式 2，因此在 Tb2 间隔（图 3.100 中（3））结束时数据从暂寄存器传送到比较寄存器中。

根据 TOCR2A 位 OLS1P、OLS1N、OLS2P、OLS2N、OLS3P 和 OLS3N 的设置，正相输出和反相输出的初始输出均为 OFF。在 MTU3.TMDR1 中设置互补 PWM 模式后，输出持续直到 MTU4.TCNT 大于 TDDRA 寄存器的值（图 3.100 中（4））。

每个周期的占空比为递增计数期间和递减计数期间各生成 1/2 的占空比。在递增计数和递减计数期间中产生的占空比如图 3.99 所示。

- 100%占空比输出

当输出 100%占空比时，提前一个周期设置缓冲寄存器为 0000h，以使比较寄存器的值为 0000h。

比较寄存器和暂寄存器在 Tb2 间隔中启用，当暂寄存器和计数器寄存器之间发生比较匹配时，正相变成 ON（图 3.100 中（5））。由于当两个比较匹配发生时，反相不会变成 OFF，因此保留了前一个周期的低输出（图 3.100 中（6））。

- 0%占空比输出

当输出 0%占空比时，提前一个周期设置缓冲寄存器与 MTU3.TGRA 相同的值，以使比较寄存器与 MTU3.TGRA 的值相同。以下比较匹配（a、b、c 和 d）发生，但波形不变。正相波保持低输出，而反相波保持高输出（图 3.100 中（7））。

- 对于比较匹配 c，比较寄存器和计数器寄存器之间发生比较匹配，正相变成 OFF，但波形不会改变，因为正相从 OFF 变为 OFF。
- 对于比较匹配 a 和 d，反相输出变成 ON/OFF 的比较匹配同时发生。但是，当把同一相位的输出变成 ON/OFF 的比较匹配同时发生时，这两个比较匹配将被忽略，波形不会改变。
- 对于比较匹配 b，由于把输出变成 OFF 的比较匹配 c 发生在同一区间中，因此 b 被忽略，波形不会改变。

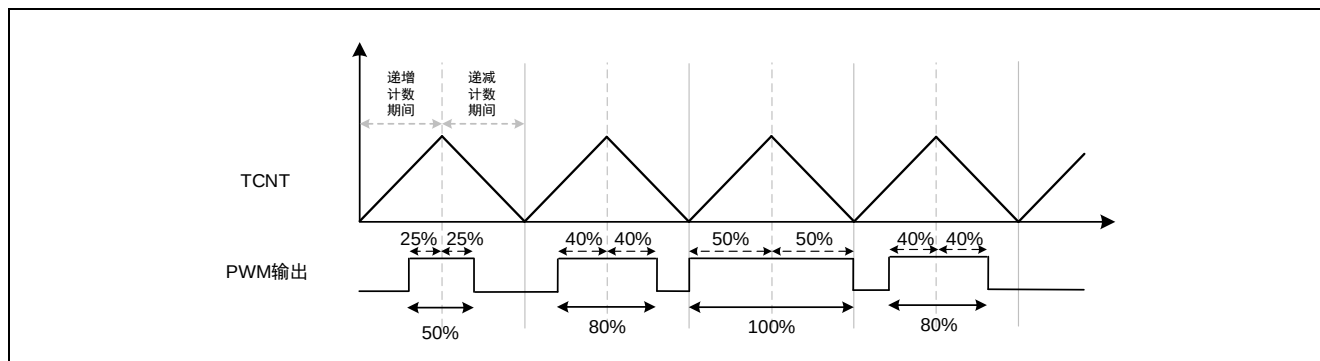


图 3.99 对称 PWM 输出波形

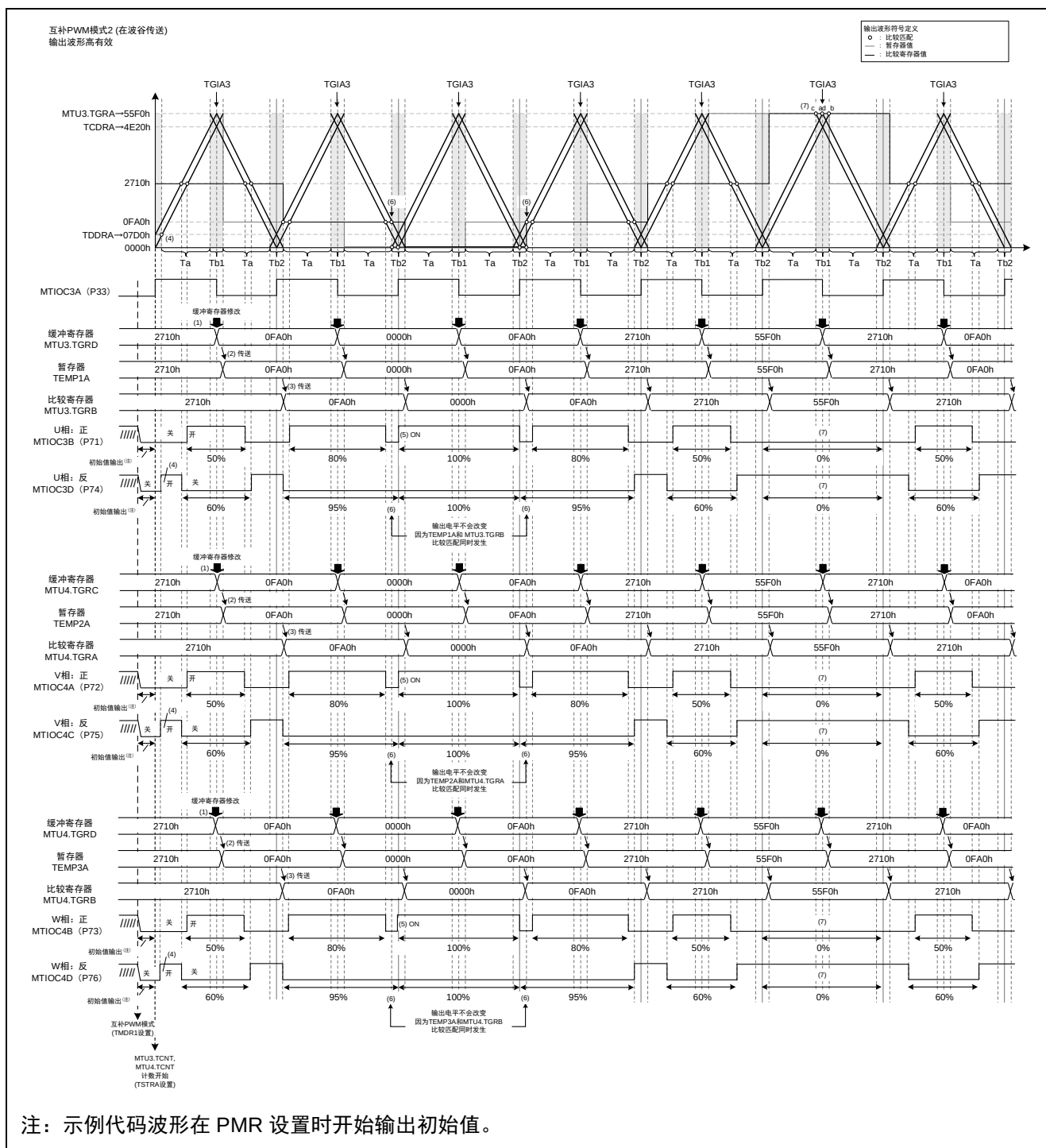


图 3.100 示例代码运行

3.14.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.17 添加组件

项目	描述
组件	互补 PWM 模式定时器
配置名称	Config_MTU3_MTU4
运行	互补 PWM 模式 2（在波谷传送）
资源	MTU3_MTU4

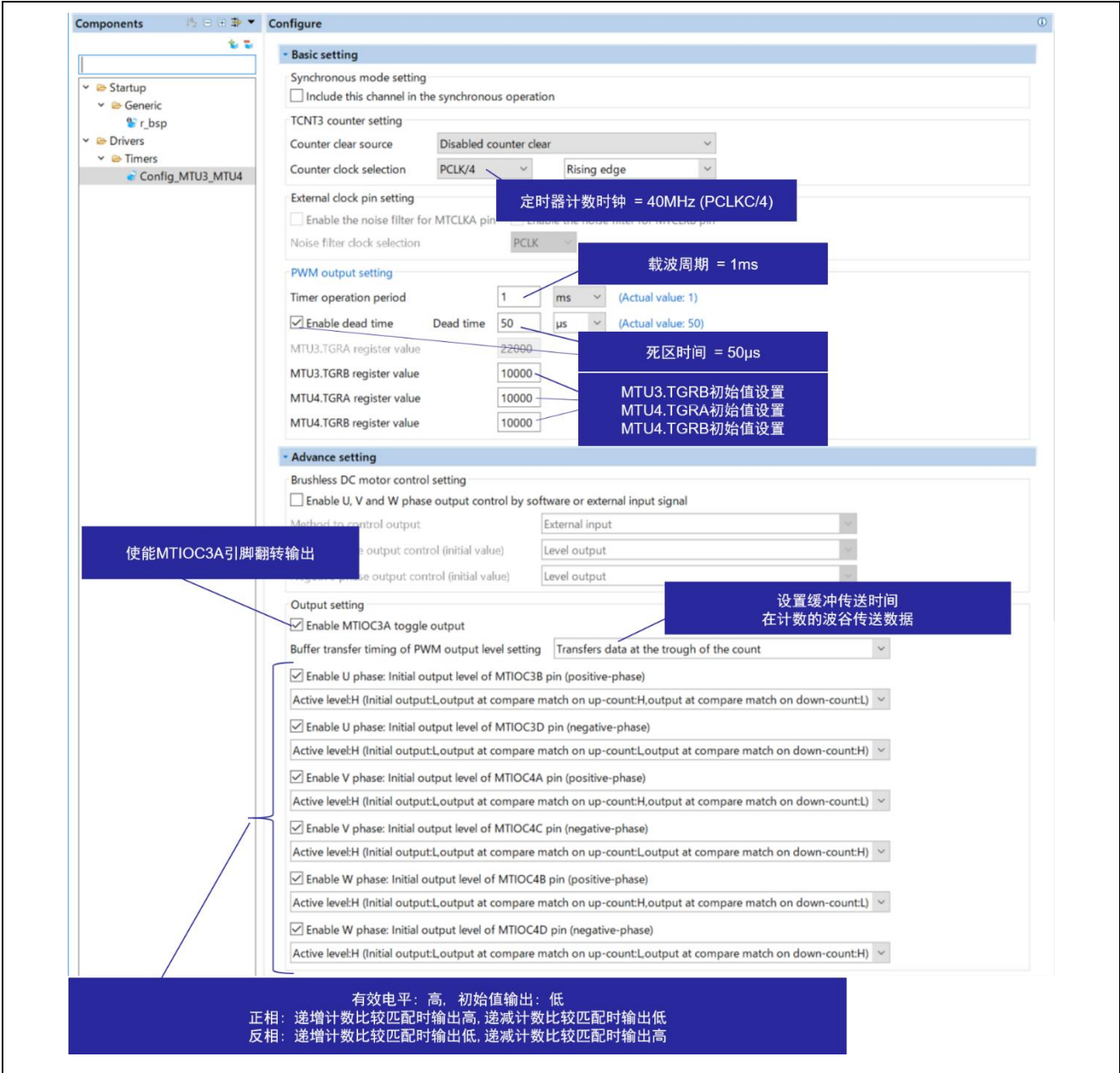


图 3.101 MTU3 和 MTU4 设置（1/2）

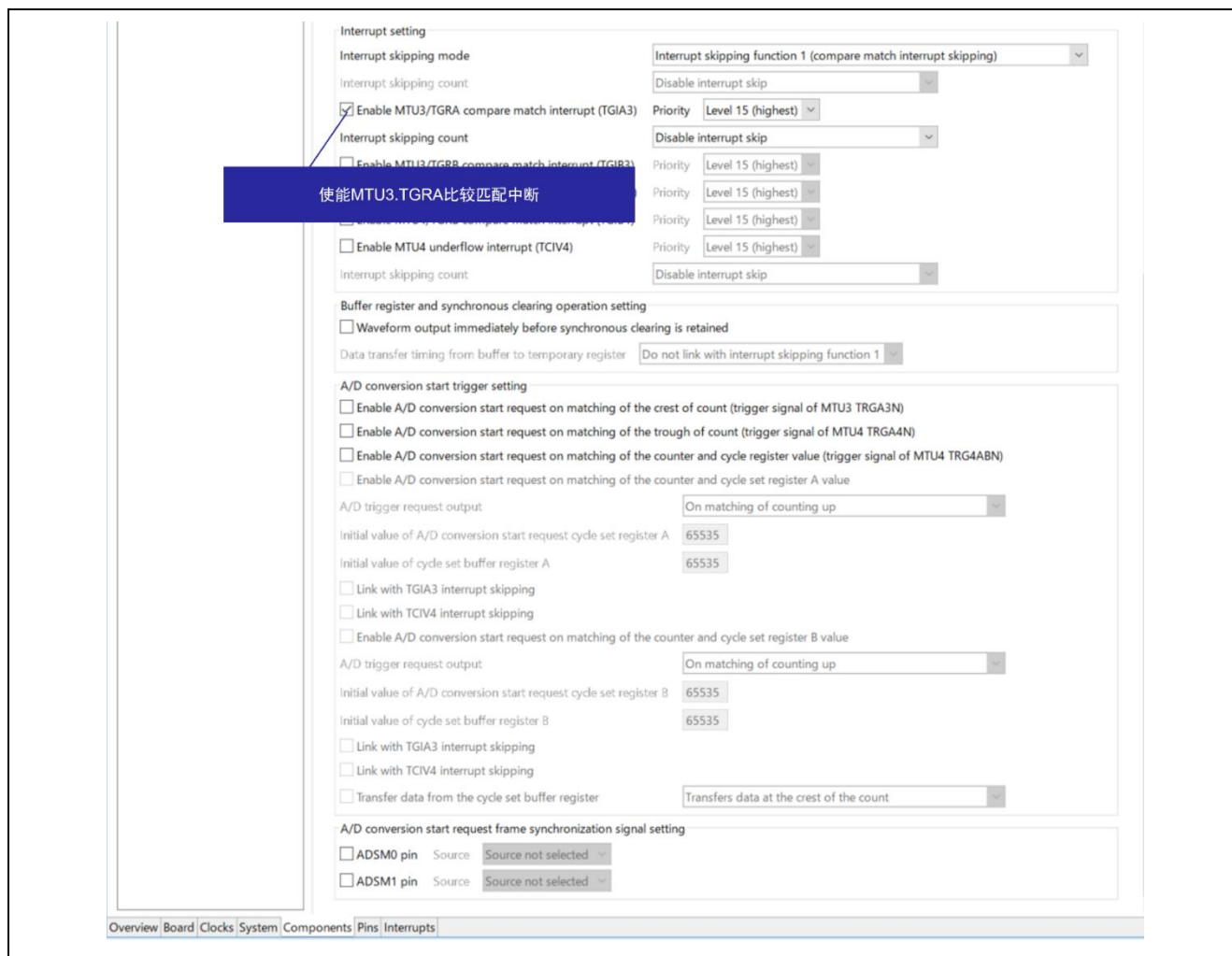


图 3.102 MTU3 和 MTU4 设置 (2/2)

3.14.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
MTU3.TCNT 和 MTU4.TCNT 计数在主函数中启动。

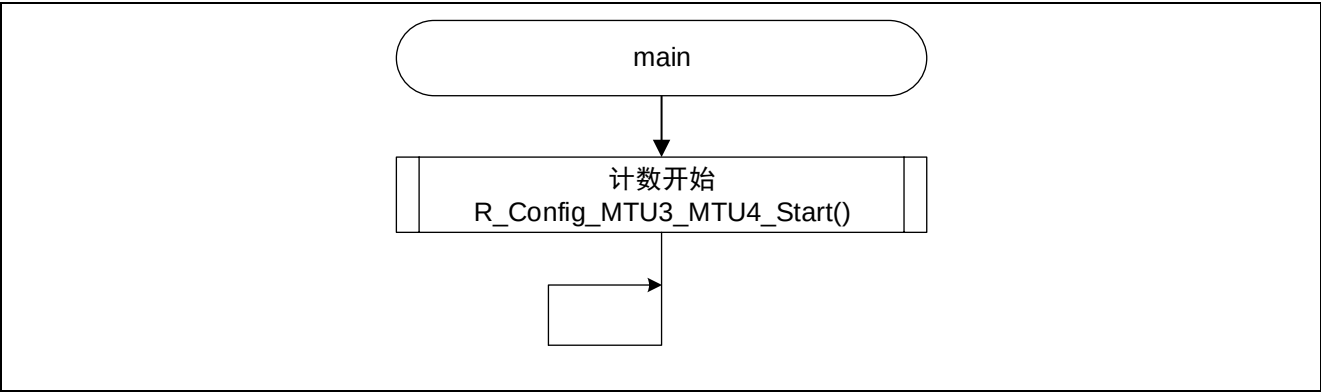


图 3.103 主函数

在主函数之前执行的用户初始化函数 R_Config_MTU0_Create_UserInit 中初始化变量。此函数在 R_Config_MTU0_Create 函数中调用。

此函数初始化此示例代码中使用的以下变量。

- s_duty_list_counter: 用于从占空比列表中读取的计数器变量

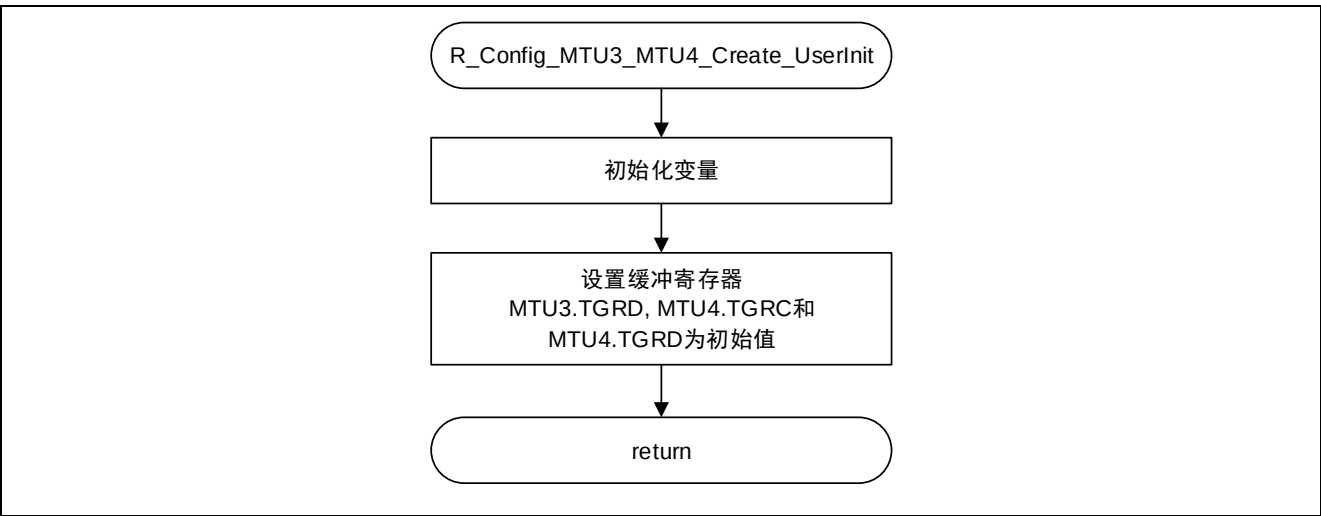


图 3.104 用户初始化函数

TGIA3 中断处理函数按照从占空比列表中读取的值依次改变下一次缓冲寄存器 (MTU3.TGRD, MTU4.TGRC 和 MTU4.TGRD) 的值。

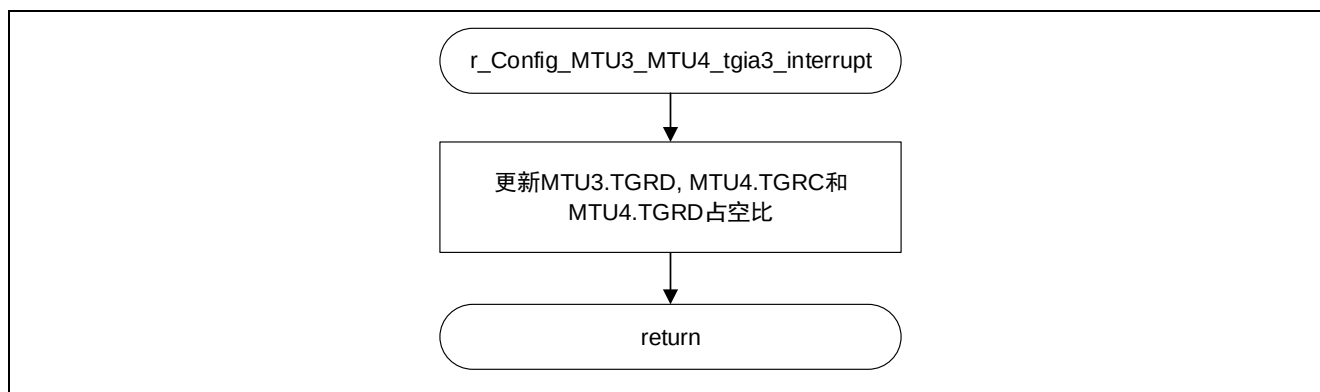


图 3.105 TGIA3 中断处理函数

3.14.5 使用时的注意事项

3.14.5.1 引脚设置

在互补 PWM 模式下，MTIOC3C 和 MTIOC6C 引脚不能用作定时器输入/输出引脚。将它们设置为 I/O 端口。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式, 表 22.74 互补 PWM 模式的输出引脚。

3.14.5.2 缓冲寄存器值更新

当修改缓冲寄存器中的数据时，确保最后修改 MTU4.TGRD (MTU7.TGRD)。写入 MTU4.TGRD (MTU7.TGRD) 后，数据同时从缓冲寄存器传送到所有五个暂存器。

即使不更新所有五个寄存器或不更新 MTU4.TGRD (MTU7.TGRD) 中的数据，在将数据写入到需更新的寄存器后确保写入 MTU4.TGRD (MTU7.TGRD)。在这种情况下，写入 MTU4.TGRD (MTU7.TGRD) 的数据应与写入操作之前的数据相同。

3.14.5.3 缓冲运行设置

在互补 PWM 模式下，使用缓冲运行修改 PWM 周期设置寄存器 (MTU3.TGRA 和 MTU6.TGRA)、定时器周期数据寄存器 (TCDRA 和 TCDRB) 和占空比设置寄存器 (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA 和 MTU7.TGRB)。

当缓冲运行位 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA), MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 设置为 1 时，MTIOC4C (MTIOC7C) 和 MTIOC4D (MTIOC7D) 引脚无法输出波形。设置 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 位为 0。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.14 互补 PWM 模式下的缓冲运行设置。

3.14.5.4 输出电平设置

当 MTU3 和 MTU4 (或 MTU6 和 MTU7) 为互补 PWM 模式时，PWM 波形输出电平由 TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP 和 TOCR1B.OLSN 位以及 TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP 和 TOCR2B.OLSnN (n = 0 到 3) 决定。将 TIOR 寄存器设置为 00h。

如果 TDERA.TDER (TDERB.TDER) 位在互补 PWM 模式下设置为 0 (不产生死区时间)，根据 TOCR1A.OLSP (TOCR1B.OLSP) 和 TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 到 3) 位的设置，反相输出是正相输出的取反电平，并且不依赖于 TOCR1A.OLSN (TOCR1B.OLSN) 和 TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 到 3) 位的设置。

在不产生死区时间的情况下，如果只使能了负相的输出，而正相的输出在 TOER 寄存器中被禁止，则负相不会被输出。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.22 定时器输出控制寄存器 1 (TOCR1A, TOCR1B) 和章节 22.2.23 定时器输出控制寄存器 2 (TOCR2A, TOCR2B)。

3.15 互补 PWM 模式 占空比 0%和 100%

- 目标示例代码文件名: r01an5995_rx66t_mtu3_complementary_pwm_0to100.zip

3.15.1 概要

MTU 互补 PWM 模式可用于输出带死区时间的三相互补 PWM 波形。

此示例代码描述了使用互补 PWM 模式 2（在波谷传送）并重复输出在占空比 0%和 100%之间交替的波形。

基本操作是在 TCNT 计数器溢出发生时从缓冲寄存器传送到暂存器，并在下溢发生时从暂存器传送到比较寄存器，从而改变占空比。当周期寄存器 MTU3.TGRA 比较匹配发生时，缓冲寄存器被修改。

以下列表提供了示例代码中使用的 MTU 设置。

- 使用互补 PWM 模式 2（在波谷传送）
- 使用通道 3 和 4
- 载波周期 = 1ms
- 死区时间 = 50us
- 定时器计数时钟 = 40MHz（PCLKC/4）
- 设置 MTU3.TGRA 为 MTU3.TCNT 上限值
（1/2 载波周期 + 死区时间）
— MTIOC3A 引脚翻转输出设置
- 设置缓冲传送时间
— 在计数的波谷传送数据
- 初始输出值为低，有效电平为高
- 使用 MTU3.TGRB 作为 U 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用 MTU4.TGRA 作为 V 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用 MTU4.TGRB 作为 W 相占空比寄存器
— 正相：
 递增计数比较匹配时输出高
 递减计数比较匹配时输出低
— 反相：
 递增计数比较匹配时输出低
 递减计数比较匹配时输出高
- 使用缓冲寄存器
— 使用 MTU3.TGRD 作为 MTU3.TGRB 的缓冲寄存器
— 使用 MTU4.TGRC 作为 MTU4.TGRA 的缓冲寄存器
— 使用 MTU4.TGRD 作为 MTU4.TGRB 的缓冲寄存器
— 有关缓冲寄存器初始值，请参考图 3.107
- 每个周期的占空比改变
— 在 MTU3.TGRA 比较匹配中断时改变占空比
— 关于占空比改变的时间的详细信息，请参考图 3.107

在 Smart Configurator 中设置。
设置方法请参考章节 3.15.3。

此示例代码的互补 PWM 模式输出如下所示。

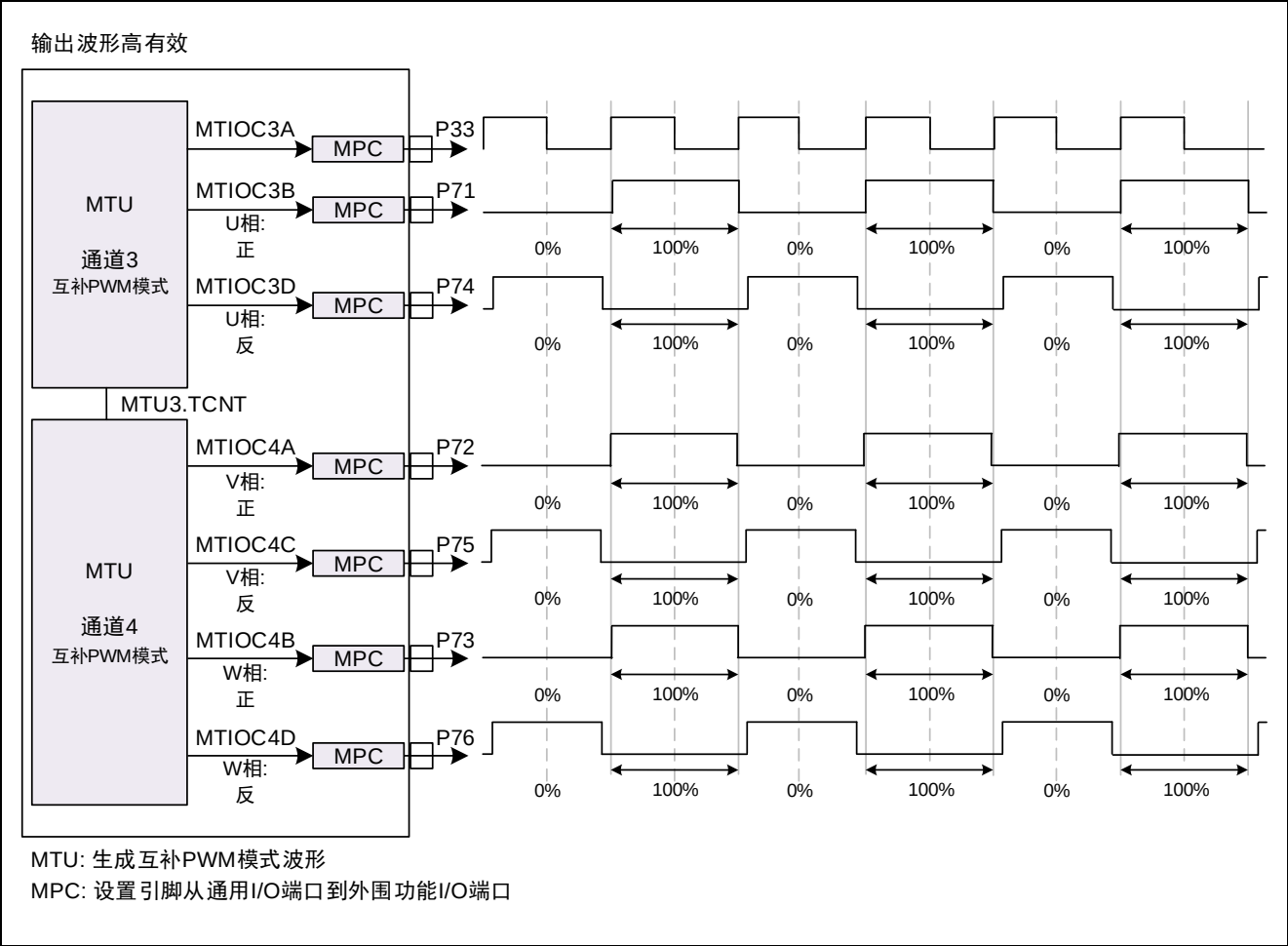


图 3.106 互补 PWM 模式输出

3.15.2 运行说明

示例代码的运行如图 3.107 所示。设置 MTU3.TGRA 寄存器为 MTU3.TCNT 的上限值（图 3.107 中（1）），通过在 MTU3.TGRA 寄存器的比较匹配中断（TGIA3）修改缓冲寄存器（MTU3.TGRD，MTU4.TGRC 和 MTU4.TGRD）的值，每个周期的占空比在 0%和 100%之间切换。

根据 TOCR2A 位 OLS1P、OLS1N、OLS2P、OLS2N、OLS3P 和 OLS3N 的设置，正相输出和反相输出的初始输出均为 OFF。在 MTU3.TMDR1 中设置互补 PWM 模式后，输出持续直到 MTU4.TCNT 大于 TDDRA 寄存器的值（图 3.107 中（2））。

- 0%占空比输出

当输出 0%占空比时，设置缓冲寄存器为与 MTU3.TGRA 相同的值，以使比较寄存器与 MTU3.TGRA 的值相同。

反相 ON 时间比正相 OFF 时间短死区时间。

发生以下比较匹配（a、b、c 和 d），但波形不变。正相波保持低输出，反相波保持高输出（图 3.107 中（3））。

- 对于比较匹配 c，比较寄存器和计数器寄存器之间发生比较匹配，正相变成 OFF，但波形不会改变，因为正相从 OFF 变为 OFF。
- 当反相输出变成 ON/OFF 的比较匹配同时发生时，这两个比较匹配将被忽略，波形不会改变。
- 对于比较匹配 b，由于把输出变成 OFF 的比较匹配 c 发生在同一区间中，因此 b 被忽略，波形不会改变。

- 100%占空比输出

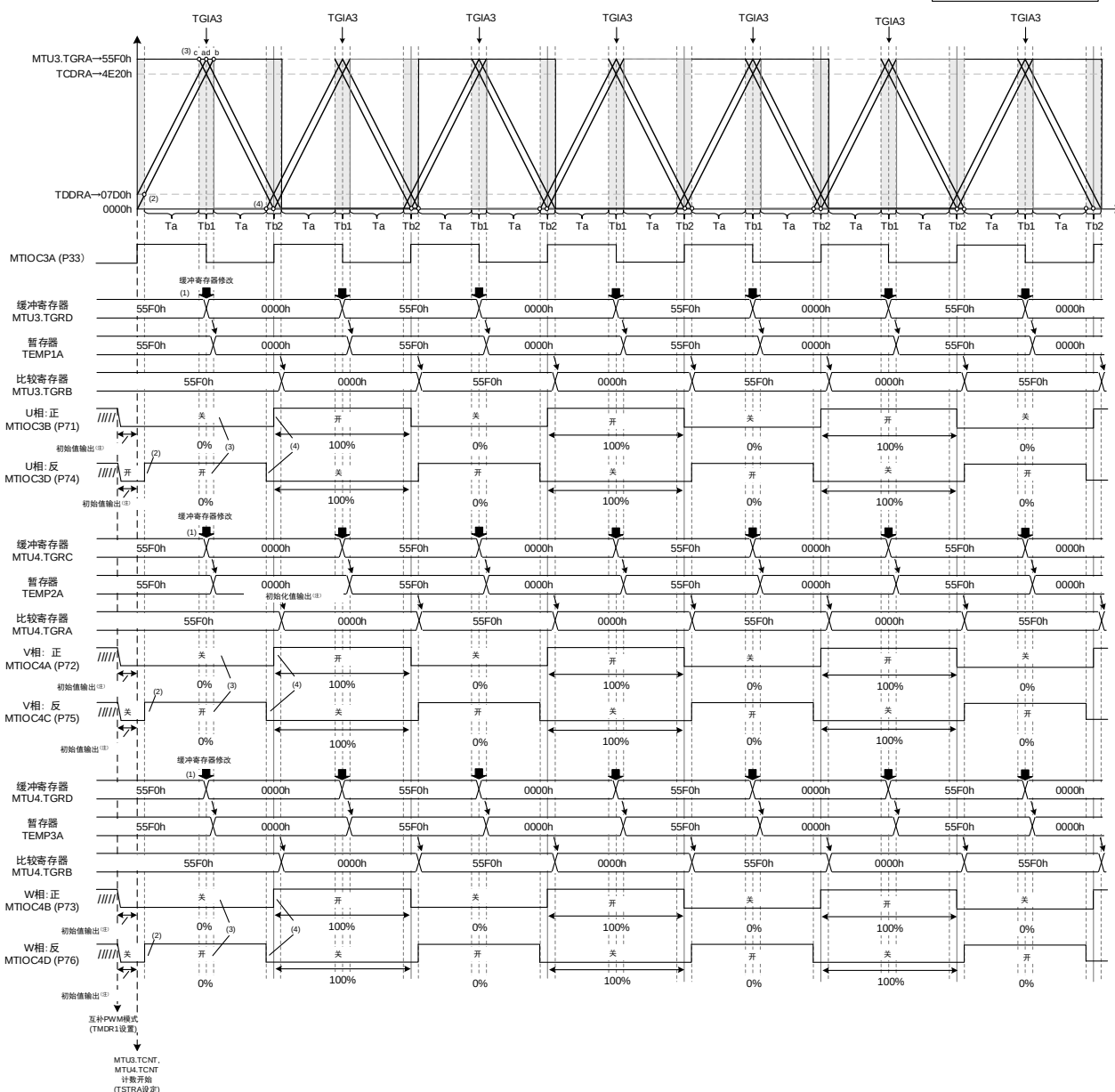
当输出 100%占空比时，提前一个周期设置缓冲寄存器为 0000h，以使比较寄存器的值为 0000h。

比较寄存器和暂寄存器在 Tb2 间隔中启用。当暂寄存器和计数器寄存器之间发生比较匹配时，正相变高并且反相变低（图 3.107 中（4））。

反相 OFF 时间比正相 ON 时间长死区时间。

互补PWM模式2(在波谷传递)
输出波形高有效

输出波形符号定义
○ : 比较匹配
--- : 寄存器值
--- : 比较寄存器值



注: 示例代码波形在 PMR 设置时开始输出初始值。

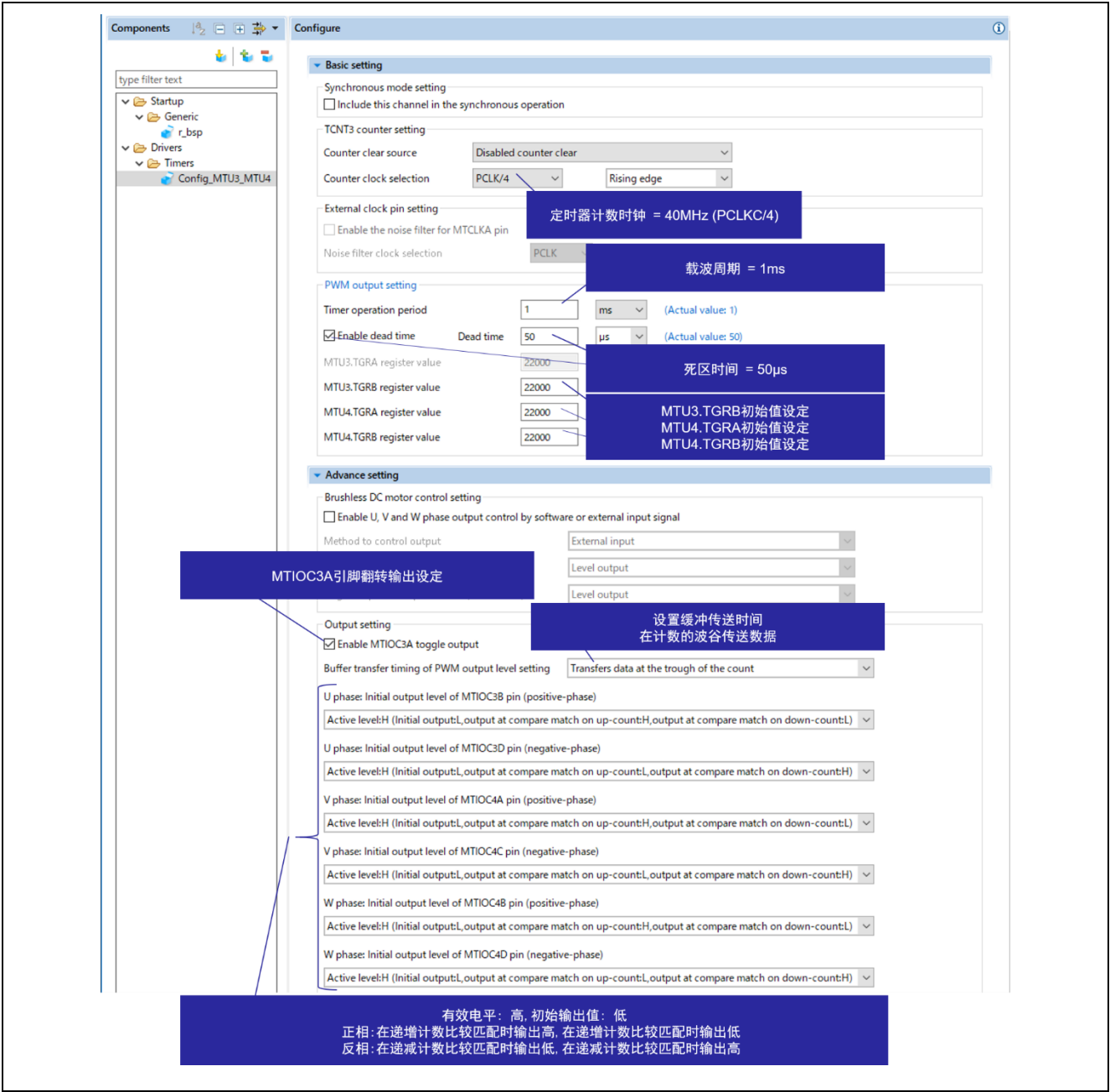
图 3.107 示例代码运行

3.15.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 MTU。添加组件的具体说明请参考 3.1.4 添加组件。

表 3.18 添加组件

项目	描述
组件	互补 PWM 模式定时器
配置名称	Config_MTU3_MTU4
运行	互补 PWM 模式 2（在波谷传送）
资源	MTU3_MTU4



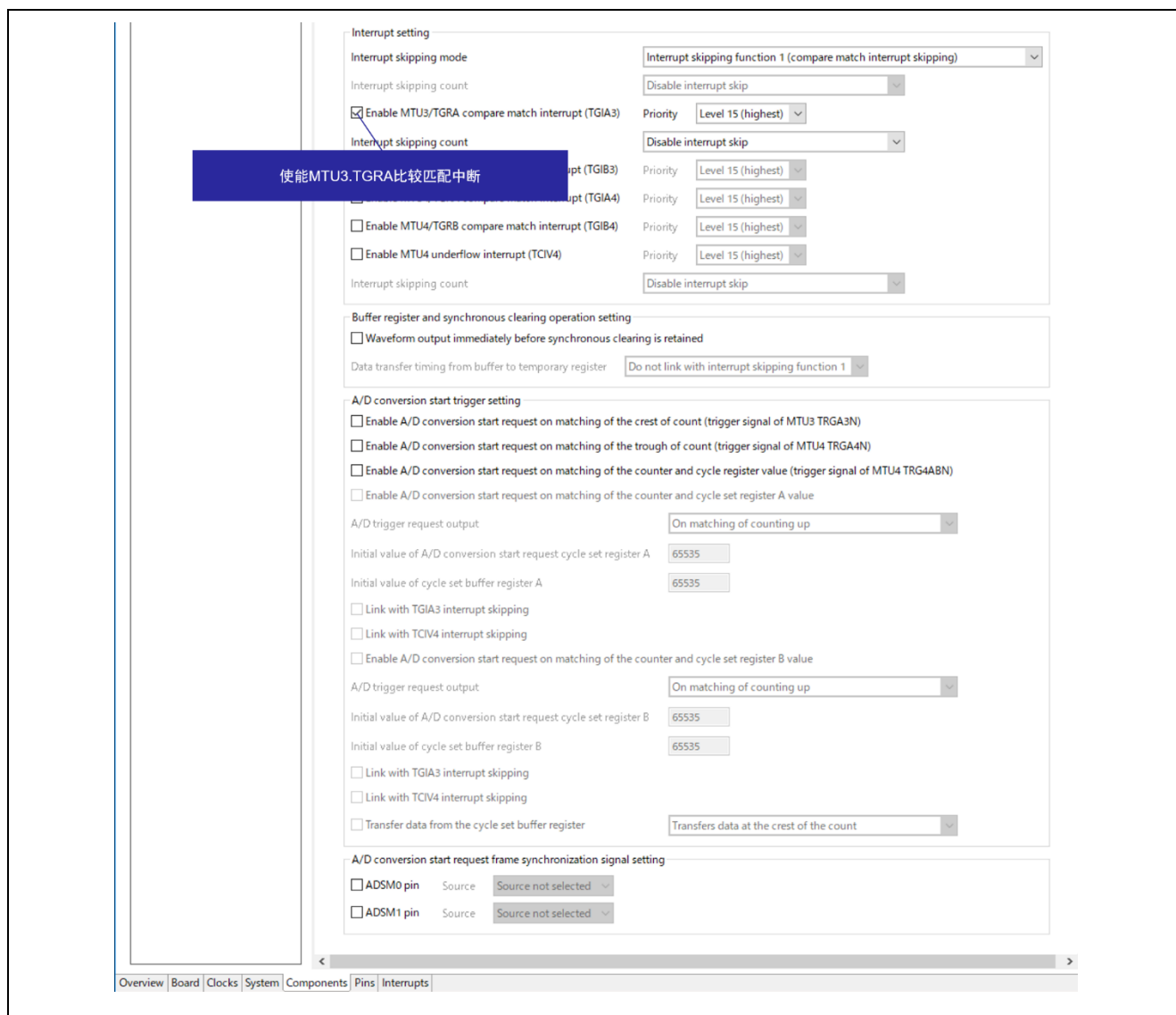


图 3.109 MTU3 和 MTU4 设置 (2/2)

3.15.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
MTU3.TCNT 和 MTU4.TCNT 计数在主函数中启动。

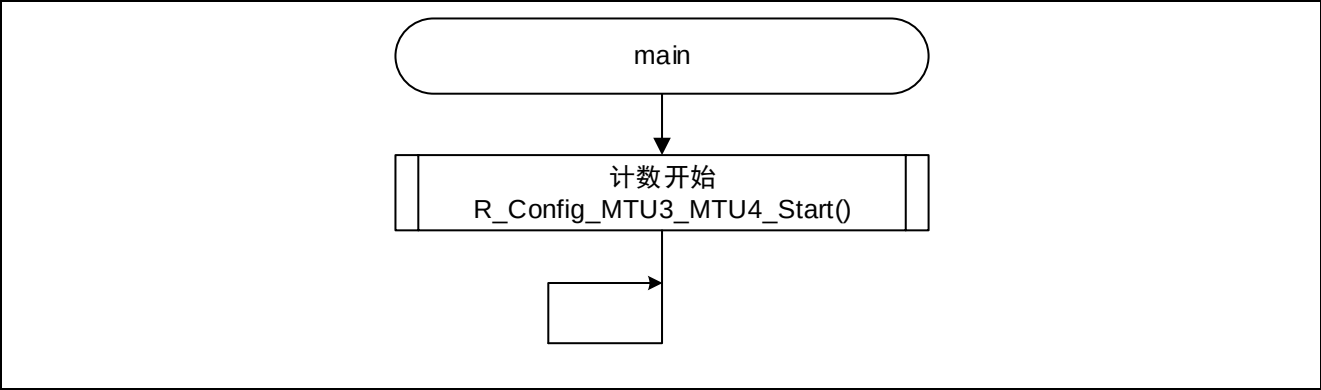


图 3.110 主函数

TGIA3 中断处理函数按照缓冲寄存器（MTU3.TGRD，MTU4.TGRC 和 MTU4.TGRD）中设置的占空比来更改要设置的下一缓冲寄存器的值。

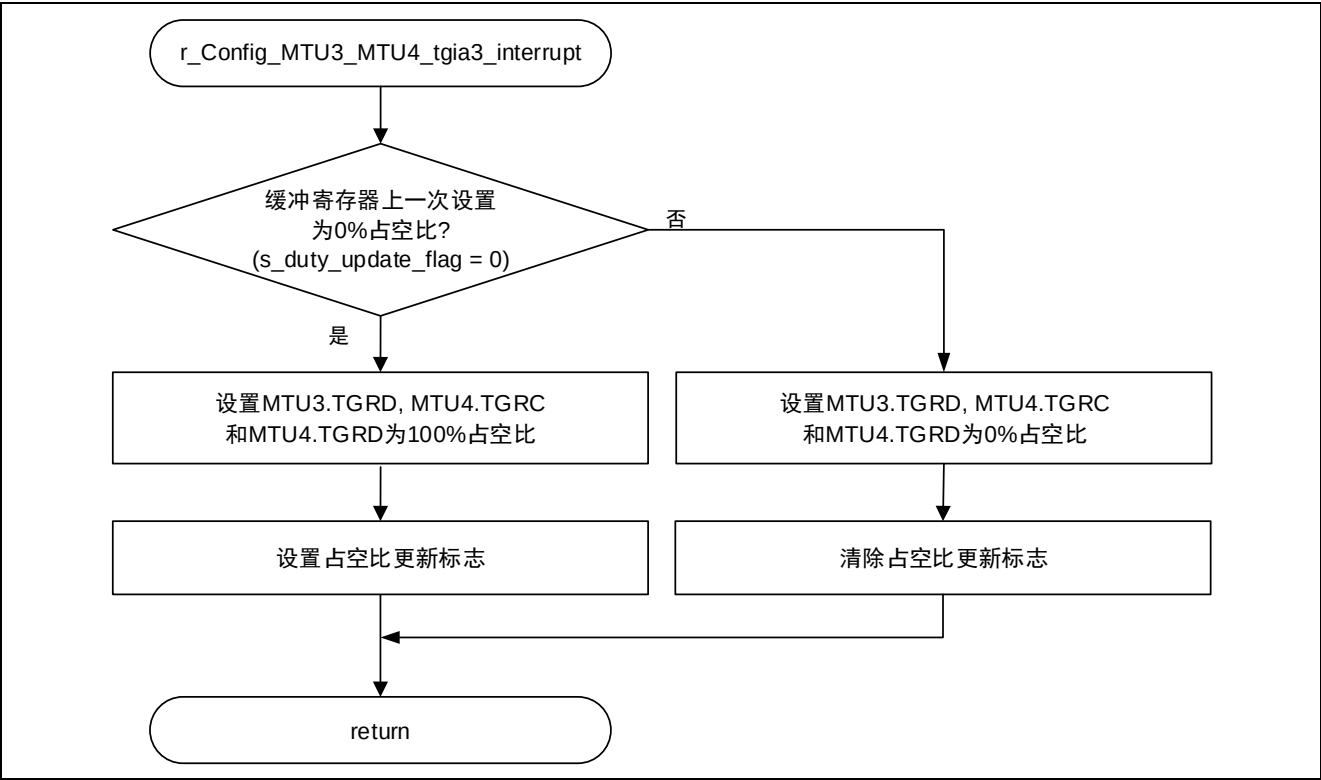


图 3.111 TGIA3 中断处理函数

3.15.5 相关运行

3.15.5.1 在初始值低时保持 0%后更改占空比到 100%

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例。

比较寄存器和暂寄存器在 Tb2 间隔中启用，当暂寄存器和计数器寄存器之间发生比较匹配时，反相输出变低，然后正相输出变高（下图中（1））。

反相 OFF 时间比正相 ON 时间长死区时间。

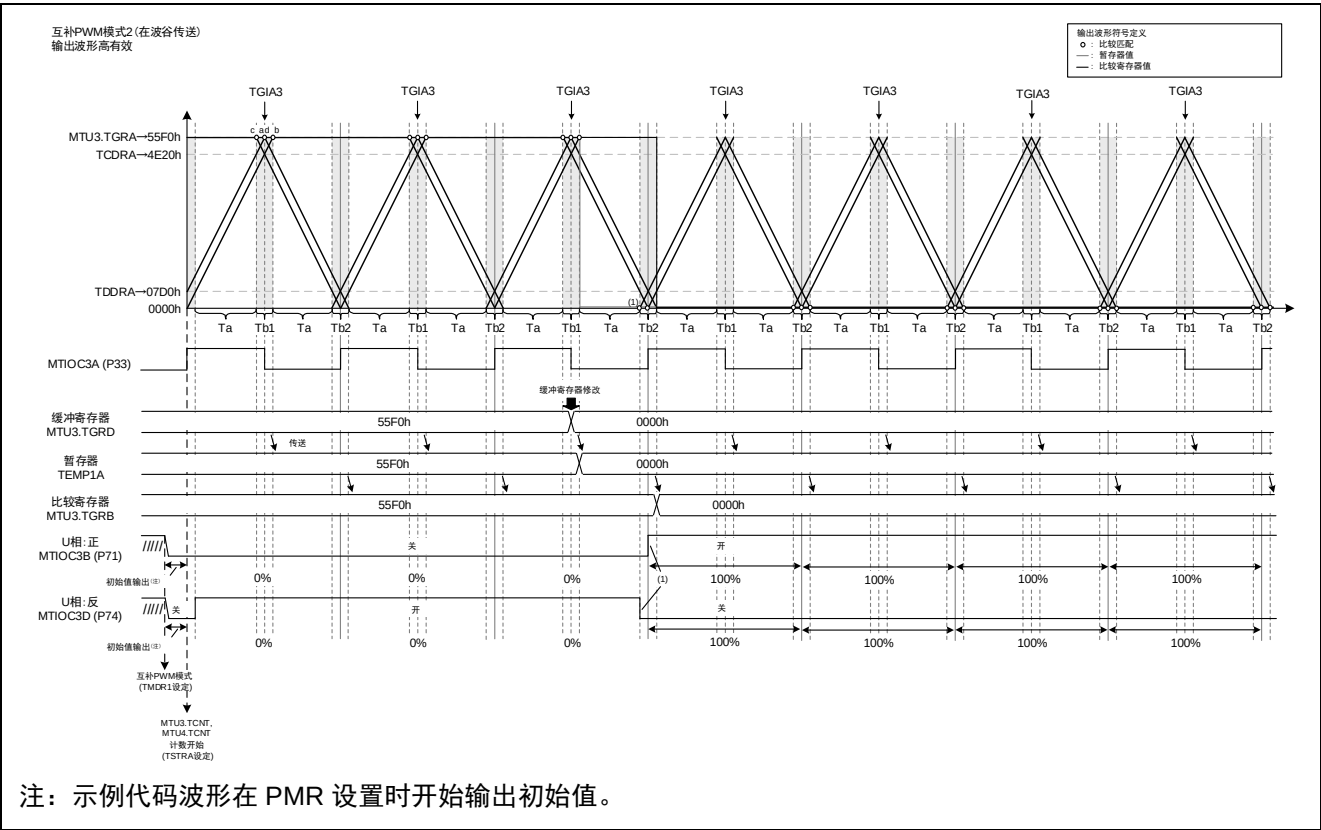


图 3.112 在保持 0%后更改占空比到 100%

3.15.5.2 在初始值低时保持 100%后更改占空比到 0%

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例。

比较寄存器和暂寄存器在 Tb2 间隔中启用，当暂寄存器和计数器寄存器之间发生比较匹配时，正相输出变低，然后反相输出变高（下图中（1））。

反相 ON 时间比正相 OFF 时间短死区时间。

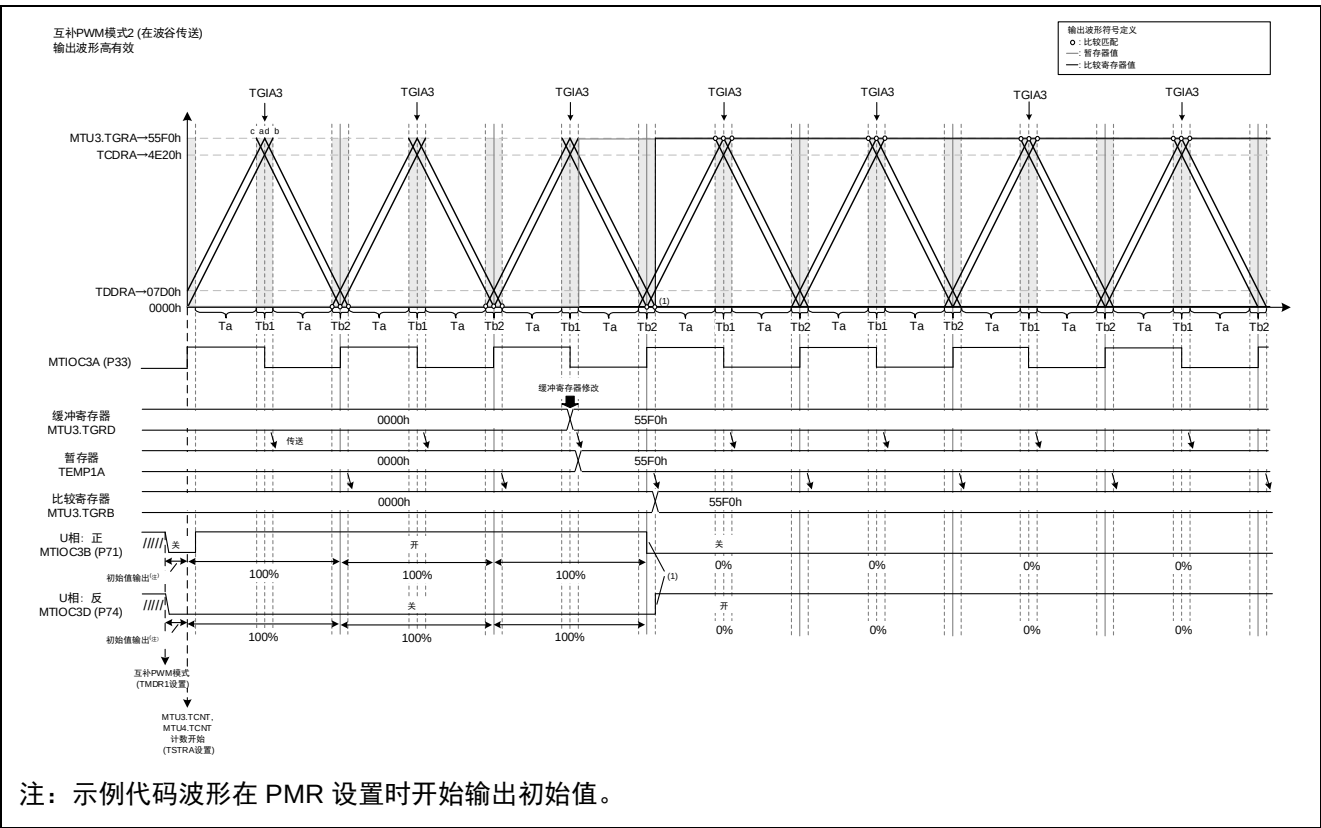


图 3.113 在保持 100%后更改占空比到 0%

3.15.5.3 在初始值高时保持 0%后更改占空比到 100%

下面显示了在输出几个周期 0%后切换占空比到 100%的运行示例。

比较寄存器和暂寄存器在 Tb2 间隔中启用，当暂寄存器和计数器寄存器之间发生比较匹配时，反相输出变高，然后正相输出变低（下图中（1））。

反相 OFF 时间比正相 ON 时间长死区时间。

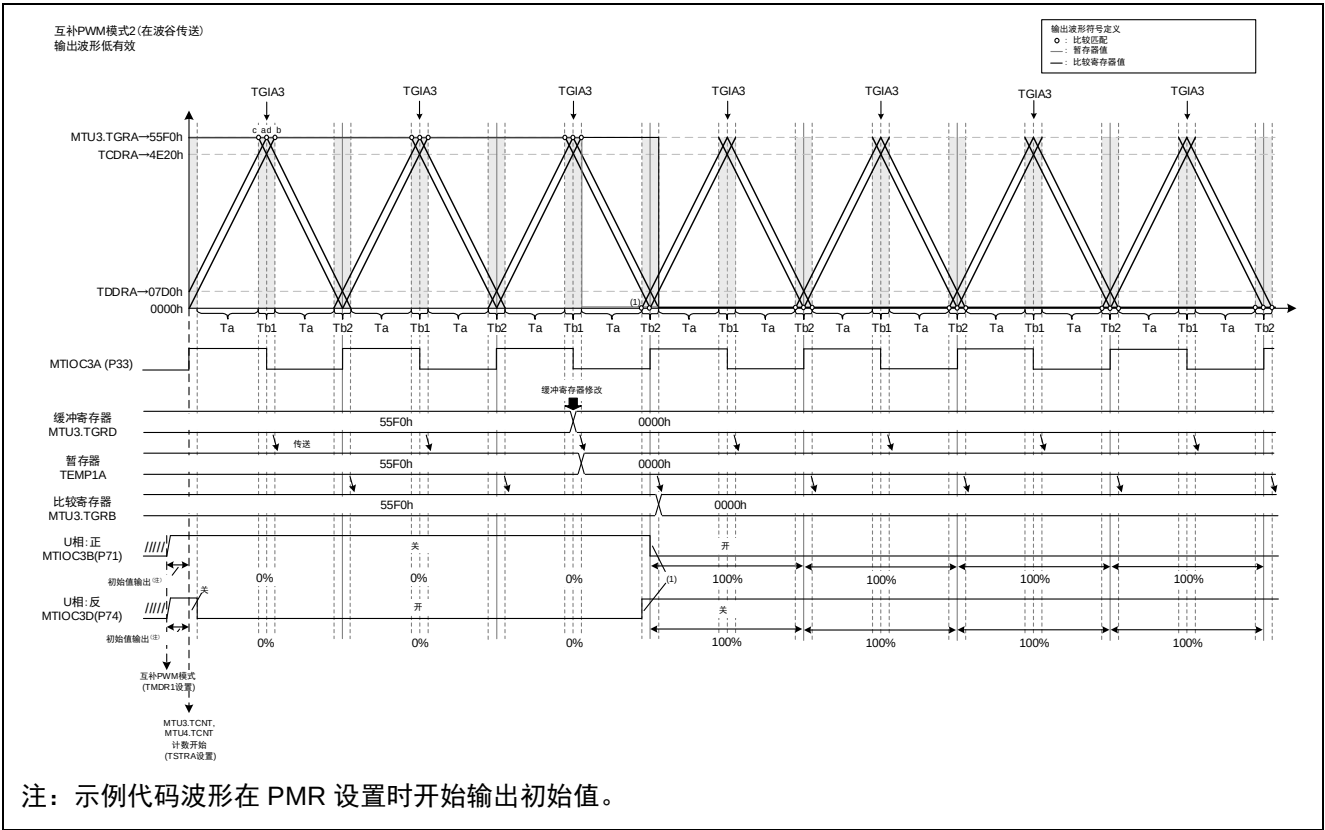


图 3.114 在保持 0%后更改占空比到 100%

要使初始值变高，请将每个引脚的“初始输出电平”设置为“有效电平：L”。在互补 PWM 模式下，当初始输出为高时，有效电平变为低。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.23 定时器输出控制寄存器 2 (TOCR2A, TOCR2B)。

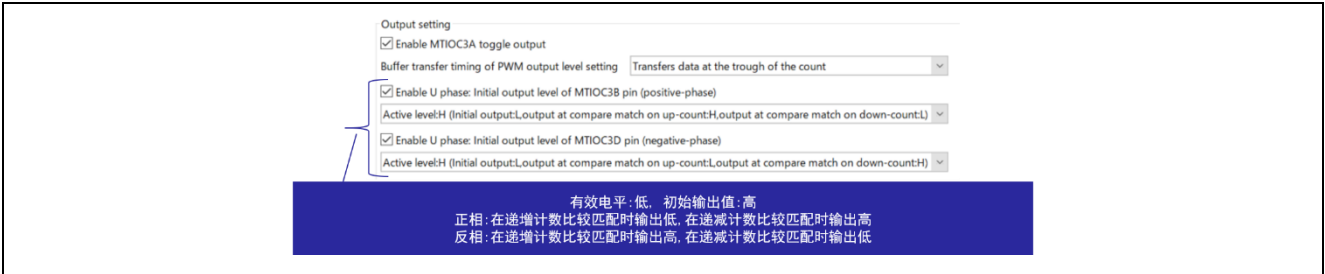


图 3.115 保持高的 Smart Configurator 设置

3.15.5.4 在初始值高时保持 100%后更改占空比到 0%

下面显示了在输出几个周期 100%后切换占空比到 0%的运行示例。

比较寄存器和暂寄存器在 Tb2 间隔中启用，当暂寄存器和计数器寄存器之间发生比较匹配时，正相输出变高，然后反相输出变低（下图中（1））。

反相 ON 时间比正相 OFF 时间短死区时间。

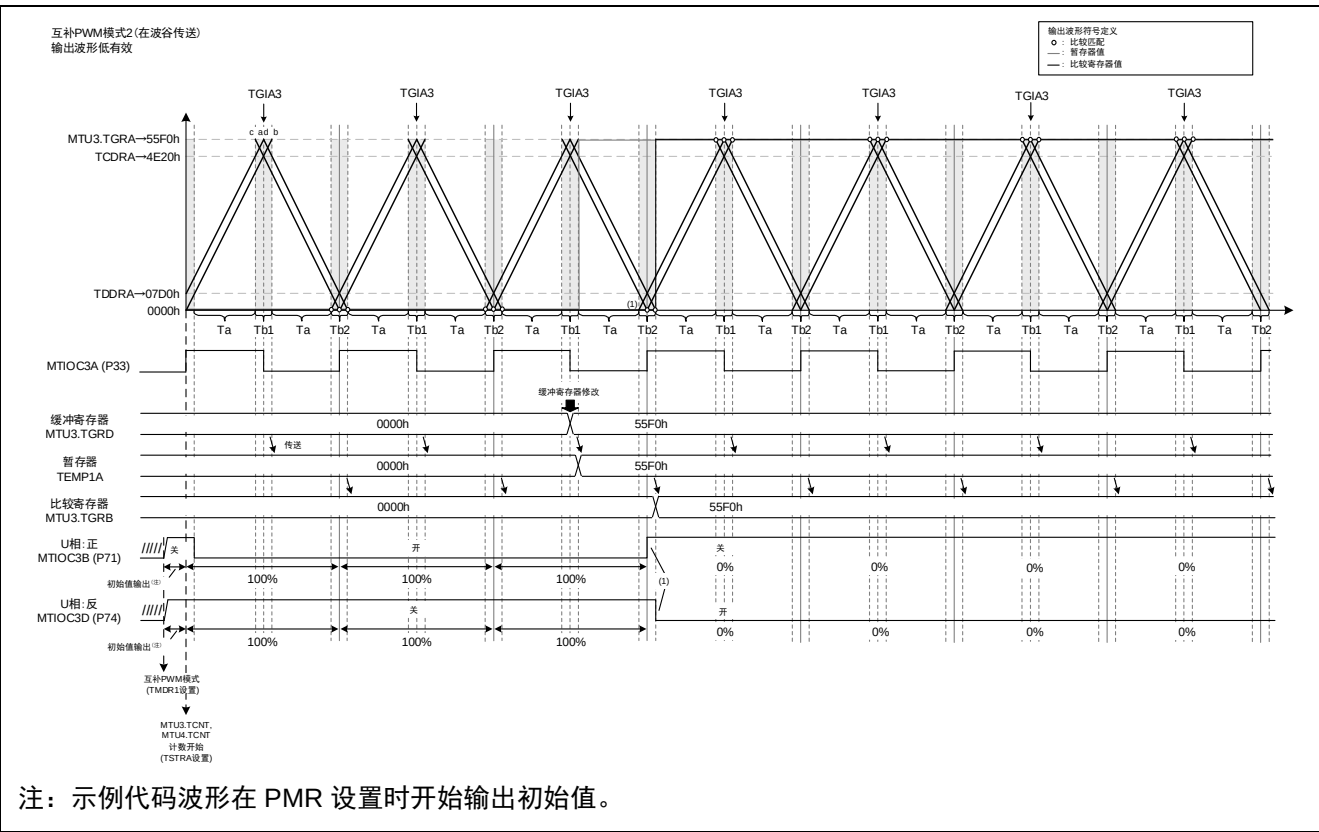


图 3.116 在保持 100%后更改占空比到 0%

要使初始值变高，请将每个引脚的“初始输出电平”设置为“有效电平：L”。详情参考图 3.115。

3.15.6 使用时的注意事项

3.15.6.1 引脚设置

在互补 PWM 模式下，MTIOC3C 和 MTIOC6C 引脚不能用作定时器输入/输出引脚。将它们设置为 I/O 端口。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.3.8 互补 PWM 模式, 表 22.74 互补 PWM 模式的输出引脚。

3.15.6.2 缓冲寄存器值更新

当修改缓冲寄存器中的数据时，确保最后修改 MTU4.TGRD (MTU7.TGRD)。写入 MTU4.TGRD (MTU7.TGRD) 后，数据同时从缓冲寄存器传送到所有五个暂存器。

即使不更新所有五个寄存器或不更新 MTU4.TGRD (MTU7.TGRD) 中的数据，在将数据写入到需更新的寄存器后确保写入 MTU4.TGRD (MTU7.TGRD)。在这种情况下，写入 MTU4.TGRD (MTU7.TGRD) 的数据应与写入操作之前的数据相同。

3.15.6.3 缓冲运行设置

在互补 PWM 模式下，使用缓冲运行修改 PWM 周期设置寄存器 (MTU3.TGRA 和 MTU6.TGRA)、定时器周期数据寄存器 (TCDRA 和 TCDRB) 和占空比设置寄存器 (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA 和 MTU7.TGRB)。

当缓冲运行位 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA), MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 设置为 1 时，MTIOC4C (MTIOC7C) 和 MTIOC4D (MTIOC7D) 引脚无法输出波形。设置 MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) 和 MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) 位为 0。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.6.14 互补 PWM 模式下的缓冲运行设置。

3.15.6.4 输出电平设置

当 MTU3 和 MTU4 (或 MTU6 和 MTU7) 为互补 PWM 模式时，PWM 波形输出电平由 TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP 和 TOCR1B.OLSN 位以及 TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP 和 TOCR2B.OLSnN (n = 0 到 3) 决定。将 TIOR 寄存器设置为 00h。

如果 TDERA.TDER (TDERB.TDER) 位在互补 PWM 模式下设置为 0 (不产生死区时间)，根据 TOCR1A.OLSP (TOCR1B.OLSP) 和 TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 到 3) 位的设置，反相输出是正相输出的取反电平，并且不依赖于 TOCR1A.OLSN (TOCR1B.OLSN) 和 TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 到 3) 位的设置。

在不产生死区时间的情况下，如果只使能了负相的输出，而正相的输出在 TOER 寄存器中被禁止，则负相不会被输出。

详情参考 RX66T Group User's Manual: Hardware, 章节 22.2.22 定时器输出控制寄存器 1 (TOCR1A、TOCR1B) 和章节 22.2.23 定时器输出控制寄存器 2 (TOCR2A、TOCR2B)。

4 GPTW 示例代码

4.1 通用代码

4.1.1 示例代码列表

本应用说明提供的以下代码都是使用 Smart Configurator 创建的。如果未标明占空比，则输出波形以占空比 80%→60%→40%→20%→... 重复。

示例代码可以从瑞萨电子网站下载。

表 4.1 GPTW 示例代码列表 (1/3)

名称	描述	参考
锯齿波 PWM 模式 r01an5995_rx66t_gptw_sawtooth_pwm.zip	使用锯齿波 PWM 模式 TCNT 溢出发生时缓冲传送，输出 2 相 PWM 波形	4.2
锯齿波单触发脉冲	使用锯齿波单触发脉冲模式 TCNT 溢出和比较匹配发生时缓冲传送，输出 1 相 PWM 波形	4.3
r01an5995_rx66t_gptw_sawtooth_1shotpls_dt.zip	使用死区时间自动设置功能	4.3
r01an5995_rx66t_gptw_sawtooth_1shotpls.zip	不使用死区时间自动设置功能	4.3.5.1
锯齿波 PWM 模式三相互补 r01an5995_rx66t_gptw_sawtooth_pwm_3phase.zip	使用锯齿波 PWM 模式 输出无死区时间的三相互补 PWM 波形 每相输出 25%, 50% 和 75% 固定占空比	4.4
锯齿波单触发脉冲 三相互补 r01an5995_rx66t_gptw_sawtooth_1shotpls_3phase_dt.zip	使用锯齿波单触发脉冲模式 TCNT 溢出和比较匹配发生时缓冲传送，输出三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	4.5
三角波 PWM 模式 1	使用三角波 PWM 模式 1 (在波谷 32 位传送) 输出带缓冲的对称三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	4.6
r01an5995_rx66t_gptw_triangle_pwm1_dt.zip	使用死区时间自动设置功能	4.6
r01an5995_rx66t_gptw_triangle_pwm1.zip	不使用死区时间自动设置功能	4.6.5.3

表 4.2 GPTW 示例代码列表 (2/3)

名称	描述	参考
三角波 PWM 模式 2	使用三角波 PWM 模式 2 (在波峰和波谷 32 位传送) 输出带缓冲的左右不对称三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	4.7
r01an5995_rx66t_gptw_triangle_pwm2_dt.zip	使用死区时间自动设置功能	4.7
r01an5995_rx66t_gptw_triangle_pwm2.zip	不使用死区时间自动设置功能	4.7.5.3
三角波 PWM 模式 3	使用三角波 PWM 模式 3 (在波谷 64 位传送) 输出带双缓冲的左右不对称三相互补 PWM 波形 输出占空比: 20%→40%→60%→80%→60%→...	4.8
r01an5995_rx66t_gptw_triangle_pwm3_dt.zip	使用死区时间自动设置功能	4.8
r01an5995_rx66t_gptw_triangle_pwm3.zip	不使用死区时间自动设置功能	4.8.5.3
锯齿波 PWM 模式 占空比 0%到 100% r01an5995_rx66t_gptw_sawtooth_pwm_50to100.zip	使用锯齿波 PWM 模式 输出包括占空比 0%和 100%的 PWM 波形 输出占空比: 50%→80%→100%→80%→50%→0%→...	4.9
锯齿波单触发脉冲 占空比 0%到 100% r01an5995_rx66t_gptw_sawtooth_1shotpls_50to100.zip	使用锯齿波单触发脉冲模式 输出包括占空比 0%和 100%的 PWM 波形 输出占空比: 50%→80%→100%→80%→50%→0%→...	4.10
锯齿波单触发脉冲 占空比 0%和 100% r01an5995_rx66t_gptw_sawtooth_pwm_0to100.zip	使用锯齿波 PWM 模式 输出在占空比 0%和 100%之间交替的 PWM 波形 输出占空比: 0%→100%→0%→100%→...	4.11
三角波 PWM 模式 1 占空比 0%到 100% r01an5995_rx66t_gptw_triangle_pwm_50to100_dt.zip	使用三角波 PWM 模式 1 (在波谷 32 位传送) 输出包括占空比 0%和 100%的 PWM 波形 输出占空比: 50%→80%→100%→80%→50%→0%→...	4.12

表 4.2 GPTW 示例代码列表（3/3）

名称	描述	参考
三角波 PWM 模式 1 占空比 0%和 100% r01an5995_rx66t_gptw_triangle_pwm_0to100_dt.zip	使用三角波 PWM 模式 1 (在波谷 32 位传送) 输出在占空比 0%和 100%之间交替的 PWM 波形 输出占空比: 0%→100%→0%→100%→...	4.13

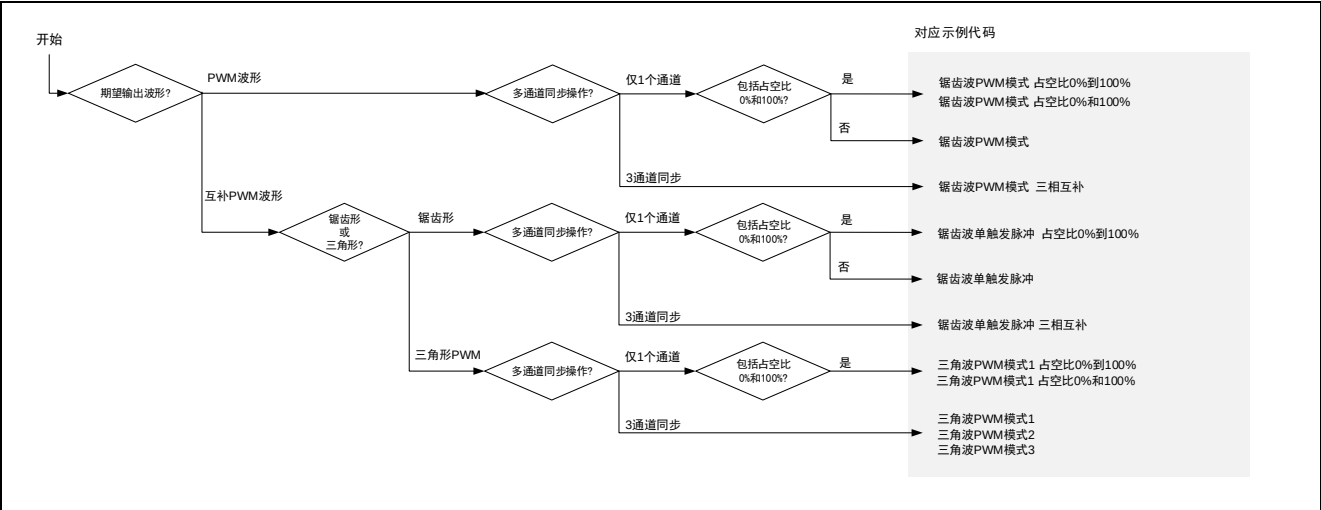


图 4.1 根据用途的示例代码

4.1.2 文件夹结构

示例代码的主要文件夹结构如下。

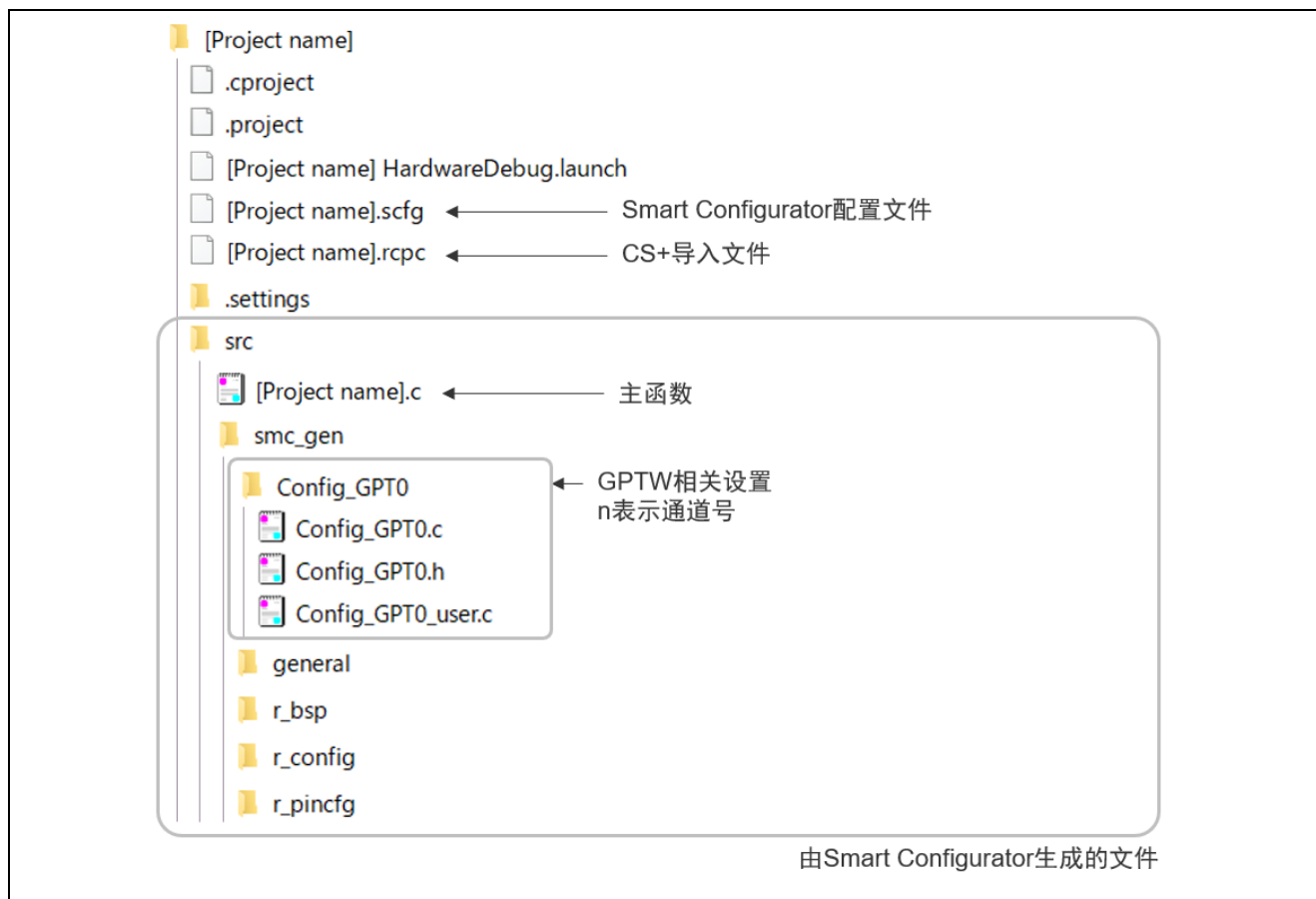


图 4.2 GPTW 文件夹结构

4.1.3 文件结构

示例代码的主要文件结构如下。

表 4.3 GPTW 文件结构

文件名称	描述
[Project name].c	<u>main function</u> 这是主函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。
Config_GPTn.c*	<u>R_Config_GPTn_Create function</u> 这是 GPTW 的初始化函数。 基于 Smart Configurator 中设置的初始化函数由 Smart Configurator 生成。 此函数的调用由 Smart Configurator 生成。该函数在主函数之前执行的 R_SystemInit 函数中调用。
	<u>R_Config_GPTn_Start function</u> 这是 GPTW 的计数开始函数。 此函数由 Smart Configurator 生成。 在示例代码中，这个函数在主函数中调用。
	<u>R_Config_GPTn_Stop function</u> 这是 GPTW 的计数停止函数。 此函数由 Smart Configurator 生成。 示例代码中未使用此函数。
Config_GPTn_user.c*	<u>r_Config_GPTn_Create_UserInit function</u> 这是 GPTW 的用户初始化函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。 这是由 Smart Configurator 生成的 R_Config_MTUn_Create 函数中调用的最后一个函数。
	<u>r_Config_GPTn [interrupt name] interrupt function</u> 这是中断处理函数。 Smart Configurator 生成一个空函数。此处描述了每个示例代码的必要处理。
Config_GPTn.h*	这是定义 GPTW 相关函数的头文件。 该文件包含在 Smart Configurator 生成的 r_smc_entry.h 文件中。 要使用 GPTW 相关函数，请务必包含 r_smc_entry.h 文件。

*: n 表示通道号

4.1.4 添加组件

如下所述，示例代码使用 Smart Configurator 添加 GPTW。

表 4.4 添加组件

项目	描述
组件	General PWM Timer（下图中（1））
配置名称	示例代码使用默认设置名称
工作模式	参考每个示例代码的章节（下图中（2））
资源	参考每个示例代码的章节（下图中（3））

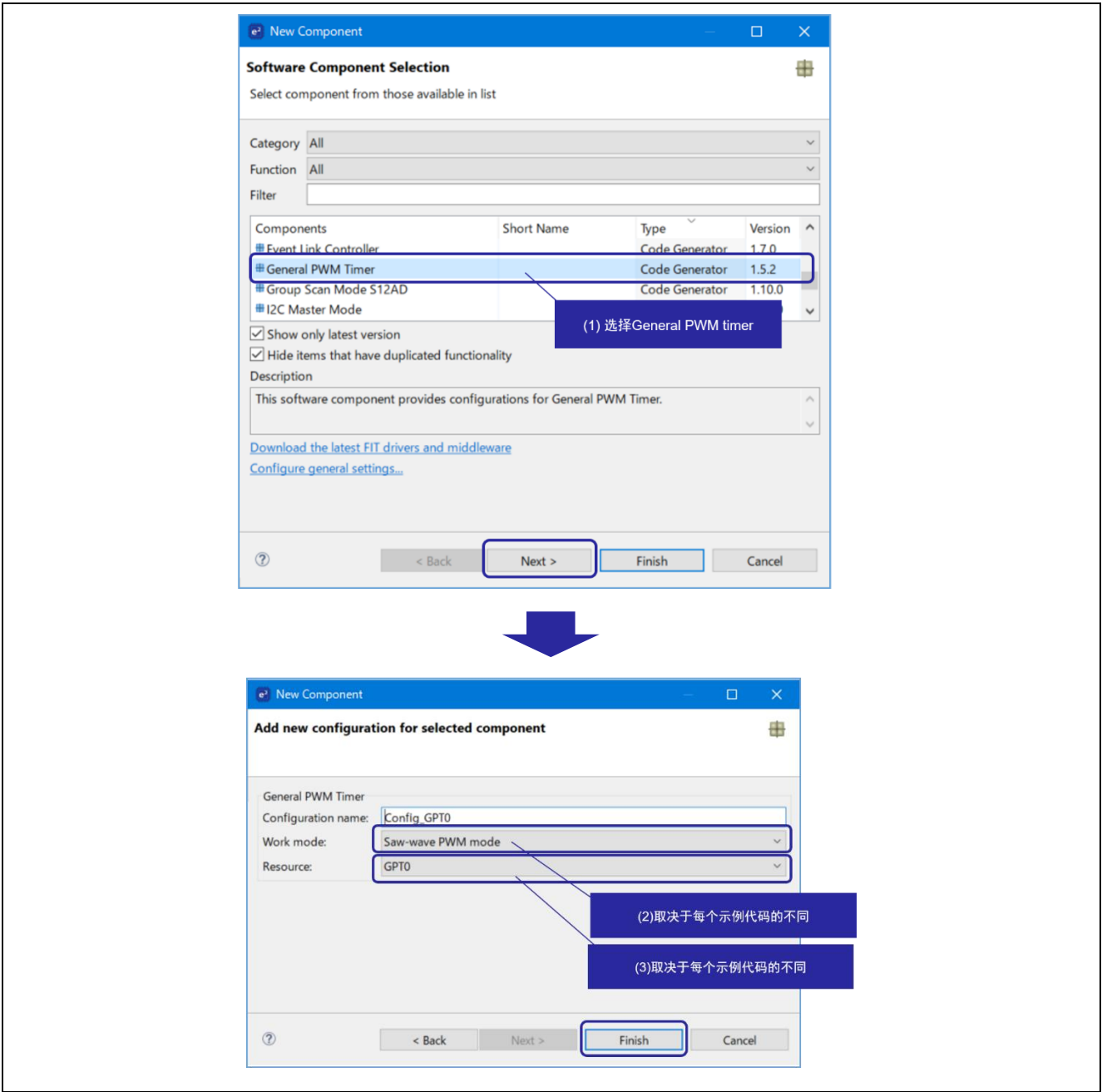


图 4.3 添加组件

4.1.5 引脚设置

图 4.4 显示了使用 Smart Configurator 进行引脚设置的示例。

设置 GPTW 后配置引脚。有关 GPTW 设置，请参考每个示例代码的“Smart Configurator Settings”。

引脚设置在由 Smart Configurator 生成的 R_Config_GPTn_Create 函数中进行。

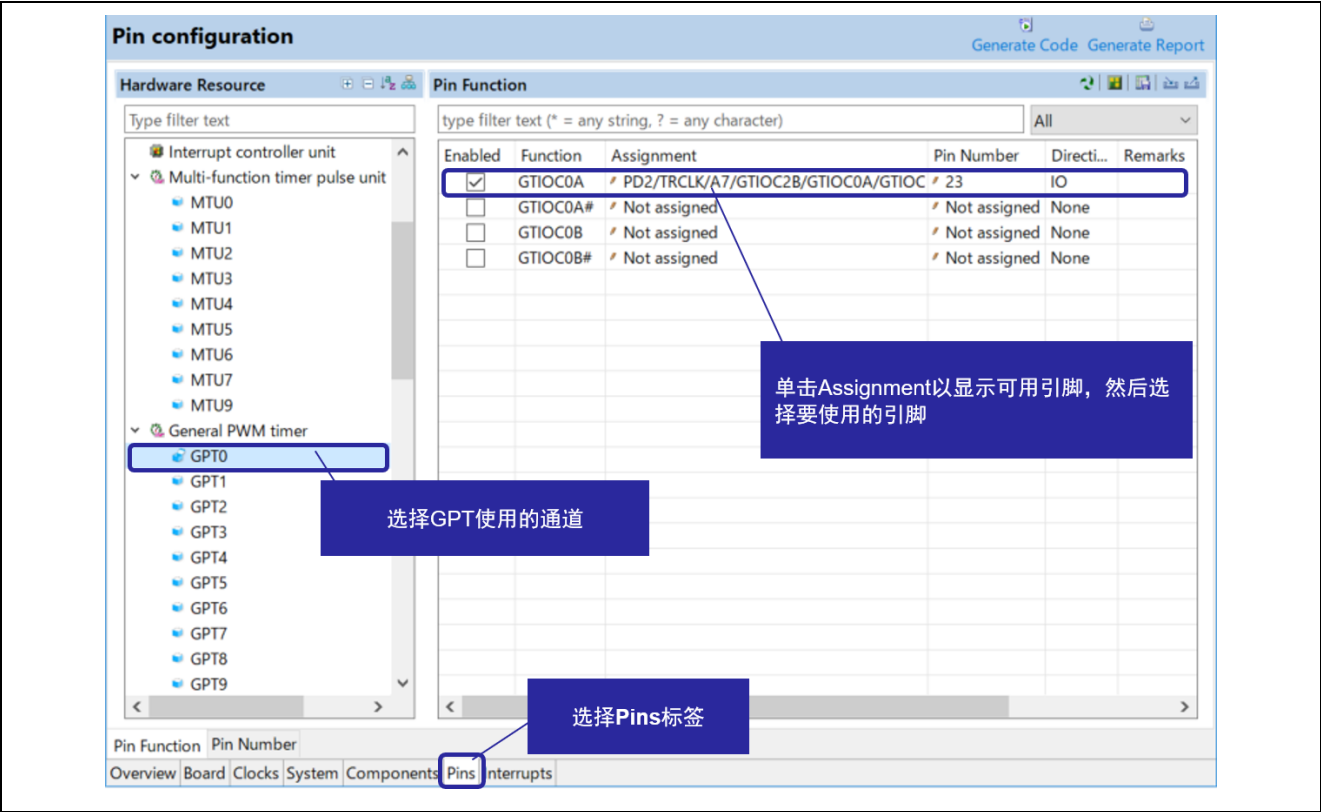


图 4.4 引脚设置

4.1.6 中断设置

图 4.5 中断设置显示了使用 Smart Configurator 设置中断的示例。有关软件可配置中断 A 的详细信息，请参考瑞萨 RX66T Group User's Manual Hardware，章节 14.4.5.1 软件可配置中断 A。

设置 GPTW 后配置中断。有关 GPTW 设置，请参考每个示例代码的“Smart Configurator Settings”。

可以在由 Smart Configurator 生成的 R_Config_GPTn_Create 函数，R_Config_GPTn_Start 函数和 R_Config_GPTn_Stop 函数中配置中断设置。

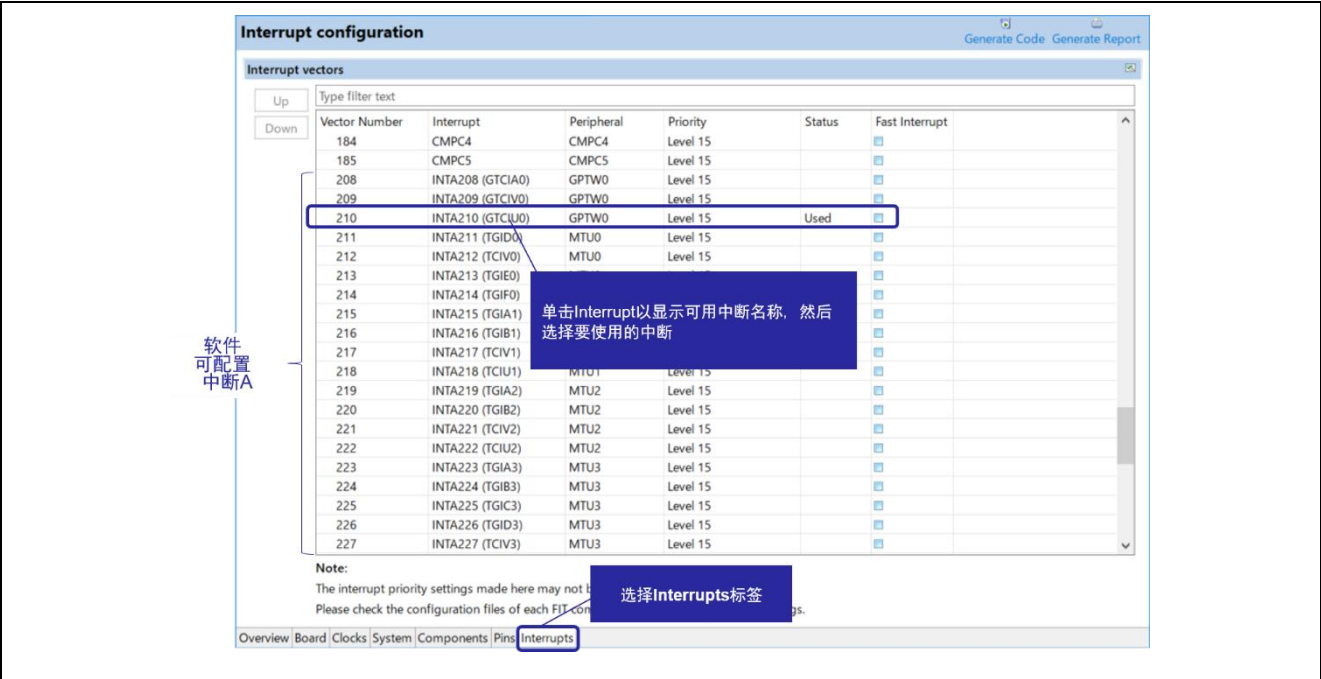


图 4.5 中断设置

在 Smart Configurator Interrupt 标签中，GPTW 中断默认只选择 GTCIE0, GTCIF0 和 GDTE0。要使用在 Components 标签中配置的中断，必须在 Interrupt 标签中选择。以下显示了缺失选择时的状态和错误信息。

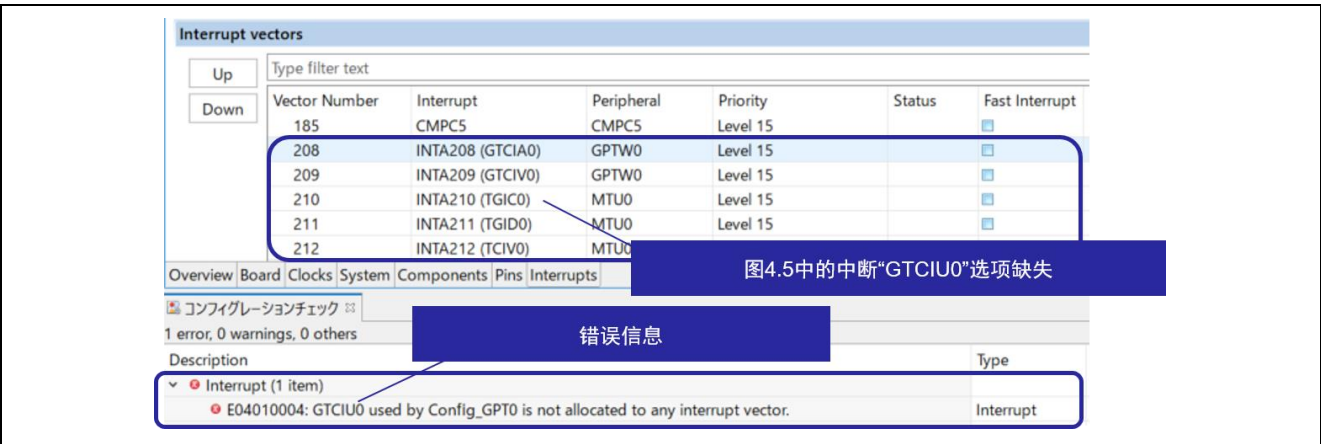


图 4.6 中断设置（缺少中断选择）

4.2 锯齿波 PWM 模式

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_pwm.zip

4.2.1 概要

GPTW 锯齿波 PWM 模式可用于根据 GTCCR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用锯齿波 PWM 模式并重复输出占空比为 80%→60%→40%→20%→80%→...的波形。使用缓冲寄存器 GTCCRC 和 GTCCRE, 当溢出发生时传送 GTCCRC 和 GTCCRE 的值到比较寄存器 GTCCRA 和 GTCCRB 中来改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波 PWM 模式
- 使用通道 0
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOC0A 引脚为 PWM 输出引脚
 - 计数开始时输出低
 - GTCCRA 比较匹配时输出高
 - 周期结束时输出低
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOC0B 引脚为 PWM 输出引脚
 - 计数开始时输出低
 - GTCCRB 比较匹配时输出高
 - 周期结束时输出低
- 使用缓冲寄存器
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
 - 使用 GTCCRE 作为 GTCCRB 的缓冲寄存器
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.8

在 Smart Configurator 中设置。
设置方法请参考章节 4.2.3。

此示例代码的锯齿波 PWM 模式输出如下所示。

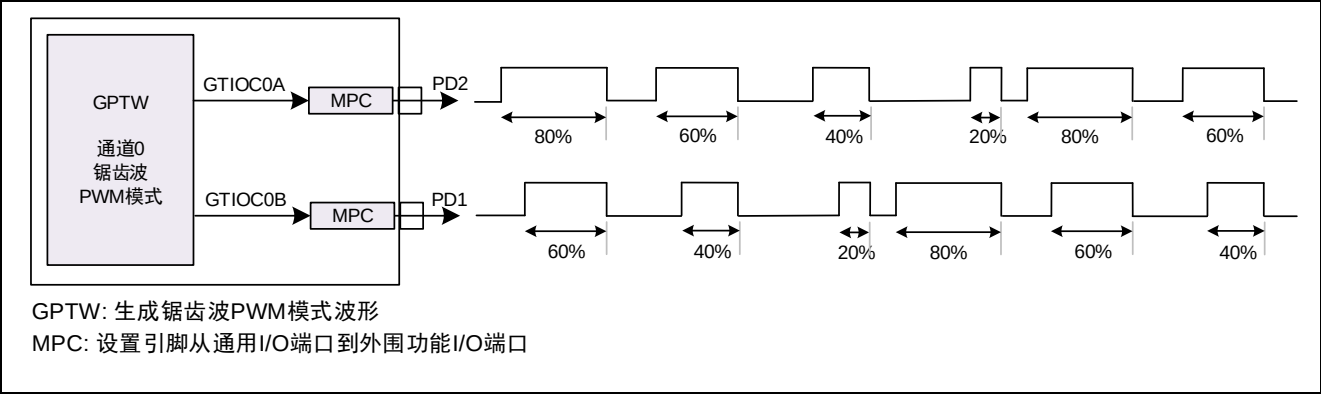


图 4.7 锯齿波 PWM 模式输出

4.2.2 运行说明

示例代码的运行如图 4.8 所示。通过在 GTCNT 溢出中断（GTCIV0）中修改缓冲寄存器 GTCCRC 和 GTCCRE 的值来改变每个周期的占空比设置。当 CTCNT 溢出发生时，GTCCRC 和 GTCCRE 的值被传送到比较寄存器 GTCCRA 和 GTCCRB。

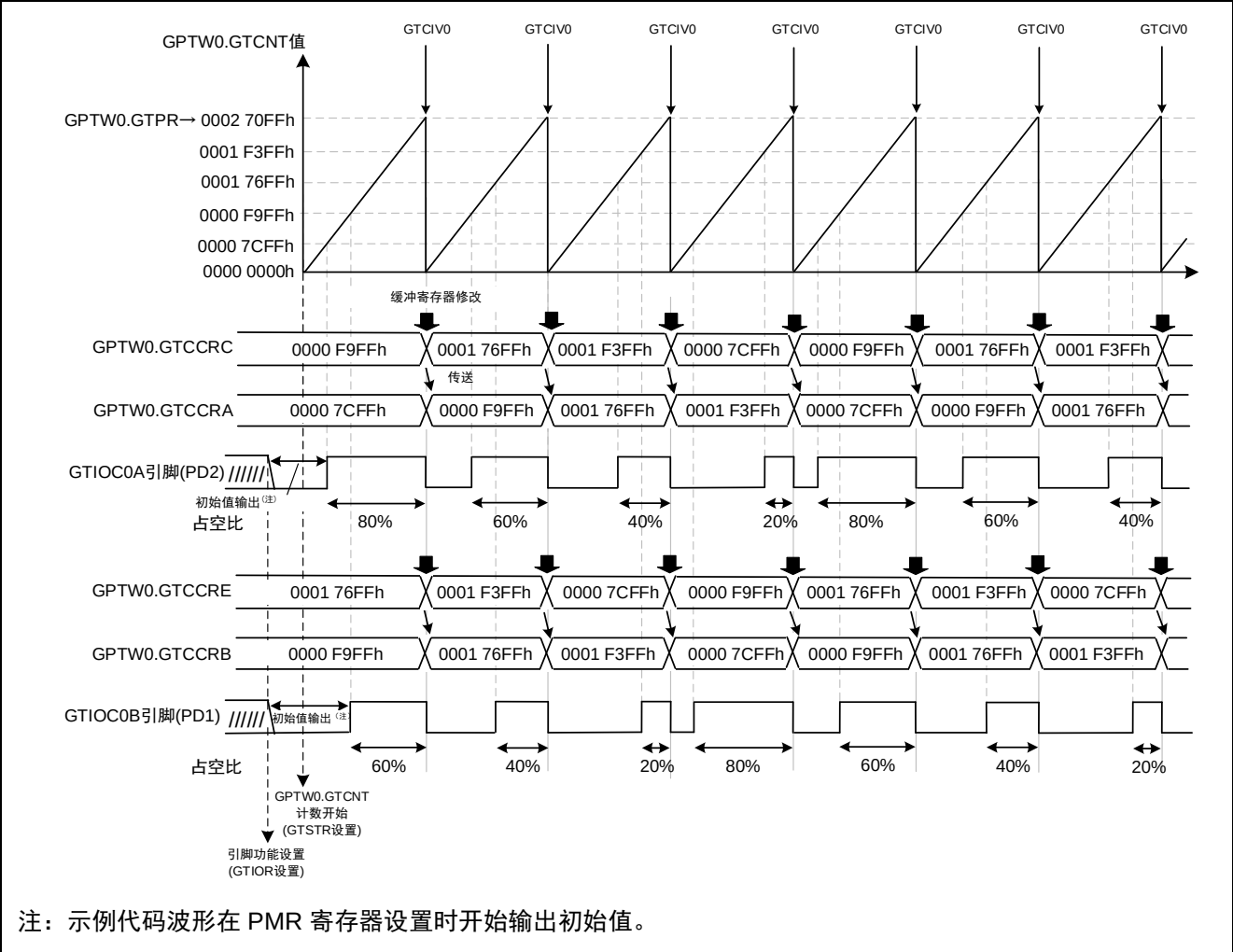


图 4.8 示例代码运行

4.2.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考 4.1.4 添加组件。

表 4.5 添加组件

项目	描述
组件	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	锯齿波 PWM 模式
资源	GPT0

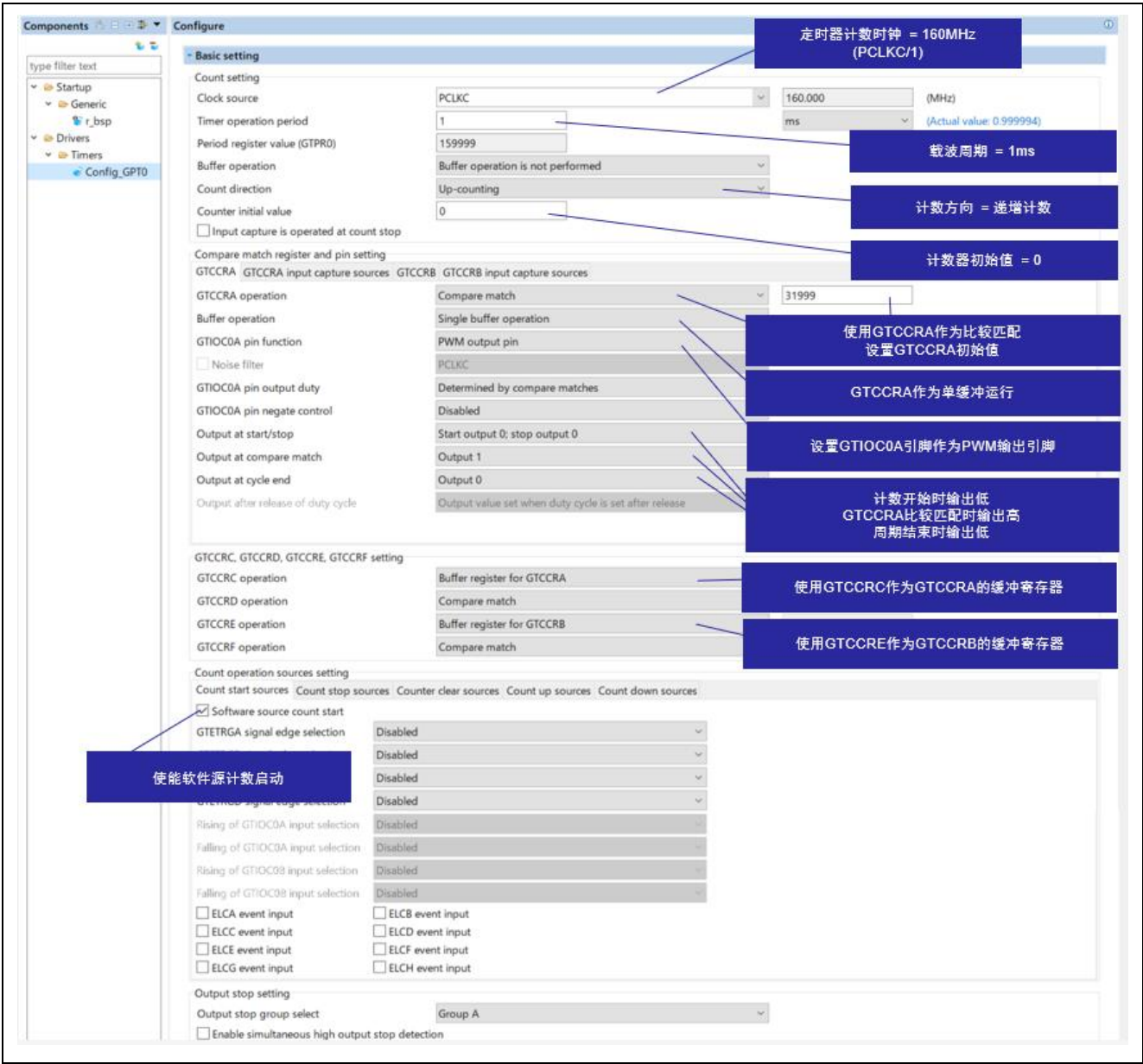


图 4.9 GPT0 设置（1/2）

Advance setting

A/D conversion start request setting

GTADTRA GTADTRB

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA) 100

Buffer operation Buffer operation is not performed

Buffer transfer timing setting No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTCIA0) Priority Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTCIB0) Priority Level 15 (highest)

☐ Enable GTCCRC compare match interrupt (GTCIC0) Priority Level 15 (highest)

☐ Enable GTCCRD compare match interrupt (GTCID0) Priority Level 15 (highest)

☐ Enable GTCCRE compare match interrupt (GTCIE0) Priority Level 15 (highest)

☐ Enable GTCCRF compare match interrupt (GTCIF0) Priority Level 15 (highest)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority Level 15 (highest)

☐ Enable GTCNT underflow interrupt (GTCIU0) Priority Level 15 (highest)

Extended interrupt skipping setting

☐ Link GTCIA0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIB0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIC0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCID0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIE0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIF0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source Skipping is not performed

Skip count Skip count of 1

Extended interrupt skipping counter 2 count source Skipping is not performed

Skip count Skip count of 1

Counter 2 initial skip count Skip count of 1

GTCCRA interrupt extended skipping function No extended interrupt skipping

GTCCRB interrupt extended skipping function No extended interrupt skipping

GTCCRC interrupt extended skipping function No extended interrupt skipping

GTCCRD interrupt extended skipping function No extended interrupt skipping

GTCCRE interrupt extended skipping function No extended interrupt skipping

GTCCRF interrupt extended skipping function No extended interrupt skipping

Overflow interrupt extended skipping function No extended interrupt skipping

Underflow interrupt extended skipping function No extended interrupt skipping

GTADTRA interrupt extended skipping function No extended interrupt skipping

GTADTRB interrupt extended skipping function No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function No extended interrupt skipping

GTCCRB buffer transfer extended skipping function No extended interrupt skipping

GTPR buffer transfer extended skipping function No extended interrupt skipping

GTADTRA buffer transfer extended skipping function No extended interrupt skipping

GTADTRB buffer transfer extended skipping function No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTCOA pin rising edge delay select Apply delay of 0/32 times PCLKC period

GTCOA pin falling edge delay select Apply delay of 0/32 times PCLKC period

GTCOB pin rising edge delay select Apply delay of 0/32 times PCLKC period

GTCOB pin falling edge delay select Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.10 GPT0 设置 (2/2)

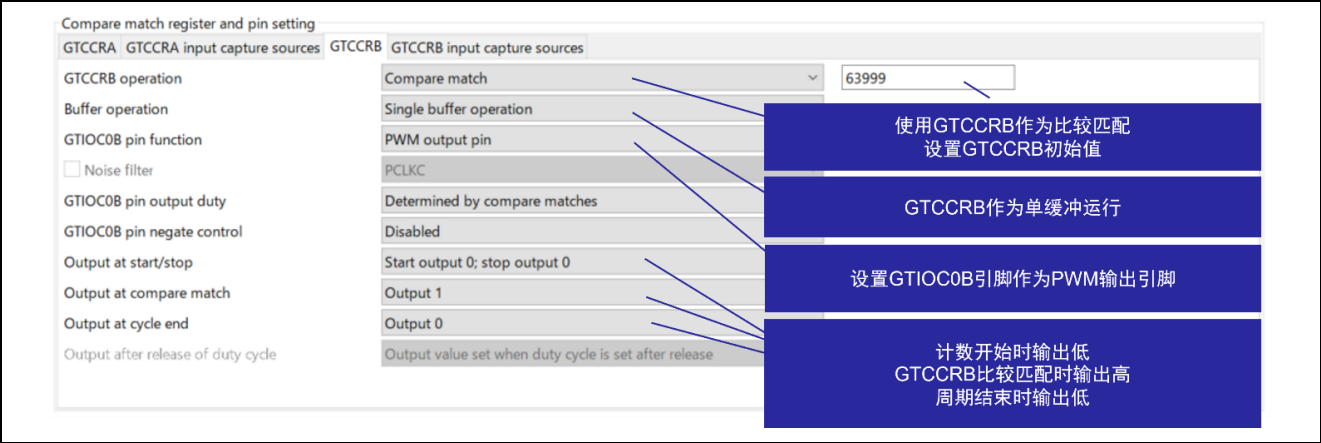


图 4.11 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.2.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
计数在主函数中启动。

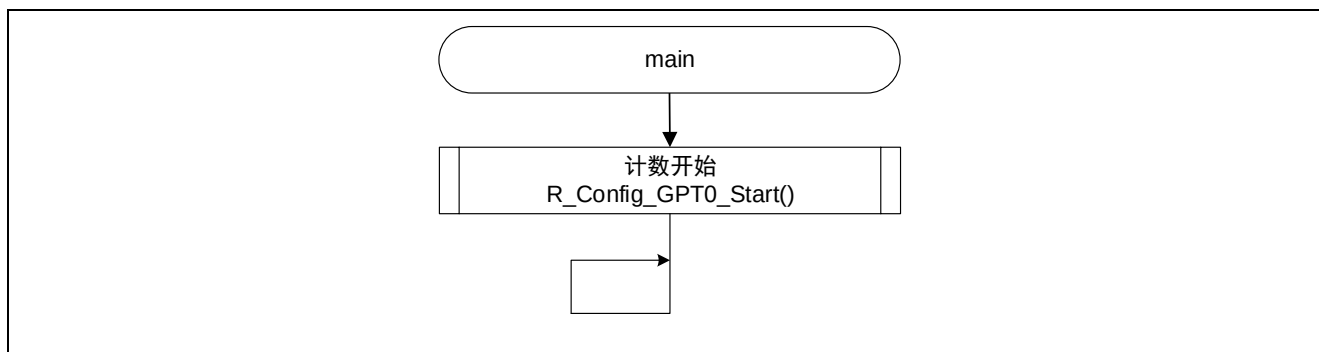


图 4.12 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 中设置缓冲寄存器的初始值。此函数在 R_Config_GPT0_Create 函数中调用。

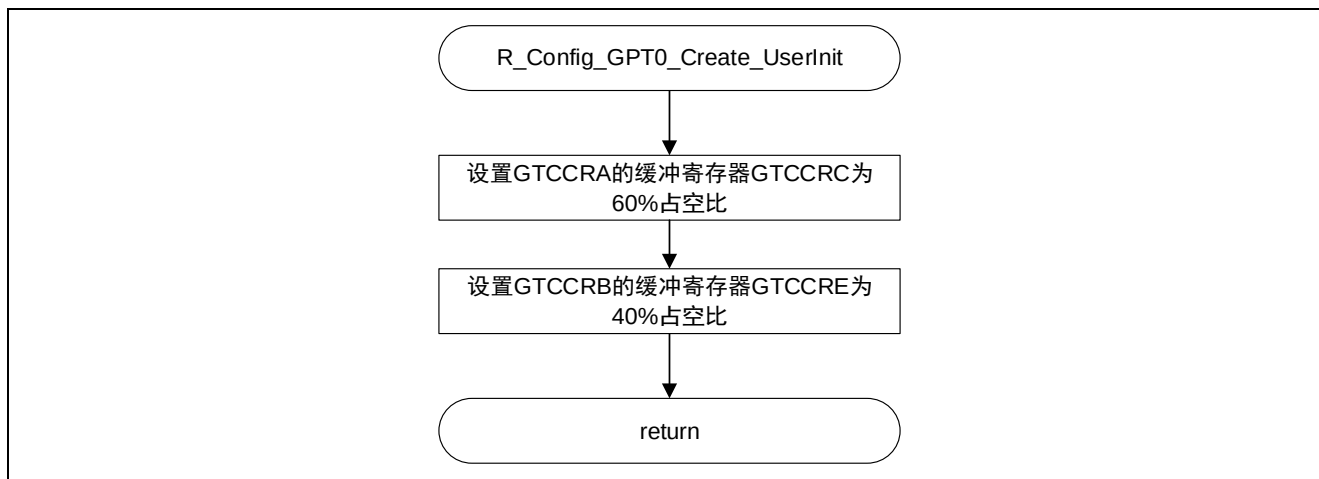


图 4.13 用户初始化函数

GTCIV0 中断处理函数根据当前占空比来改变缓冲寄存器 GTCCRC 和 GTCCRE 的值。

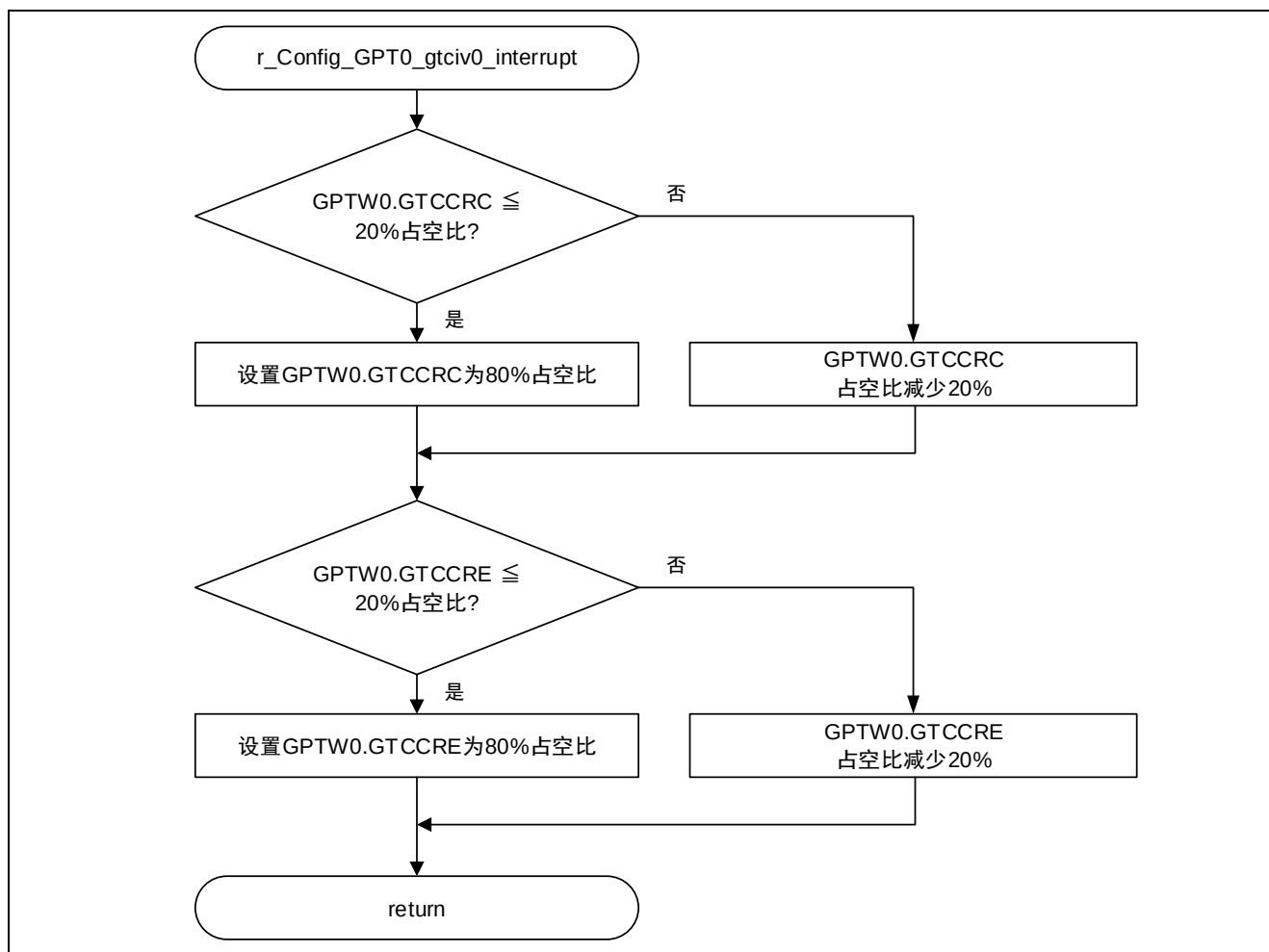


图 4.14 GTCIV0 中断处理函数

4.2.5 使用时的注意事项

4.2.5.1 比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)

将比较寄存器 GTCCRA 和 GTCCRB 设置为大于 0000 0001h 但小于 GTPR 寄存器的设置值。如果设置为 0000 0000h 或与 GTPR 寄存器的值相同，则仅当比较匹配寄存器的值为 0000 0000h 或比较寄存器设置为与 GTPR 寄存器相同的值时，才会在周期内发生比较匹配。如果将比较寄存器的值设置为超过 GTPR 寄存器的设置值，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 GTCCRm 寄存器在比较匹配运行期间的设置 (m = A 到 F)，(5) 在锯齿波 PWM 模式下。

4.3 锯齿波单触发脉冲

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_1shotpls_dt.zip

4.3.1 概要

在 GPTW 锯齿波单触发脉冲模式下, 通过设置 GTPR 寄存器的周期, GTCNT 计数器运行在锯齿波 (半波), 通过 GTCCRA 和 GTCCRB 寄存器之间的比较匹配, 可以从 GTIOCNB 和 GTIOCnA 引脚 (n = 0 到 9) 输出 PWM 波形。

此示例代码描述了在锯齿波单触发脉冲模式下使用死区时间自动设置功能并重复输出以下波形。

- GTIOC0A 引脚高带宽切换: 80% → 60% → 40% → 20% → 80% → 60% → ...
- GTIOC0B 引脚低带宽切换: 90% → 70% → 50% → 30% → 90% → 70% → ...

当 GTCCRA 比较匹配发生时, 暂存器 A 的值被传送到比较寄存器 GTCCRA。当 GTCNT 计数器溢出发生时, 通过将值从缓冲寄存器 GTCCRD 传送到暂存器 A 以及从缓冲寄存器 GTCCRC 到比较寄存器 GTCCRA 来改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波单触发脉冲模式
- 使用通道 0
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCnA 引脚为 PWM 输出引脚
 - 计数开始时输出低
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCnB 引脚为 PWM 输出引脚
 - 计数开始时输出高
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用双缓冲寄存器
 - 使用 GTCCRC 和 GTCCRD 作为 GTCCRA 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.16

在 Smart Configurator 中设置。
设置方法请参考章节 4.3.3。

此示例代码的锯齿波单触发脉冲模式输出如下所示。

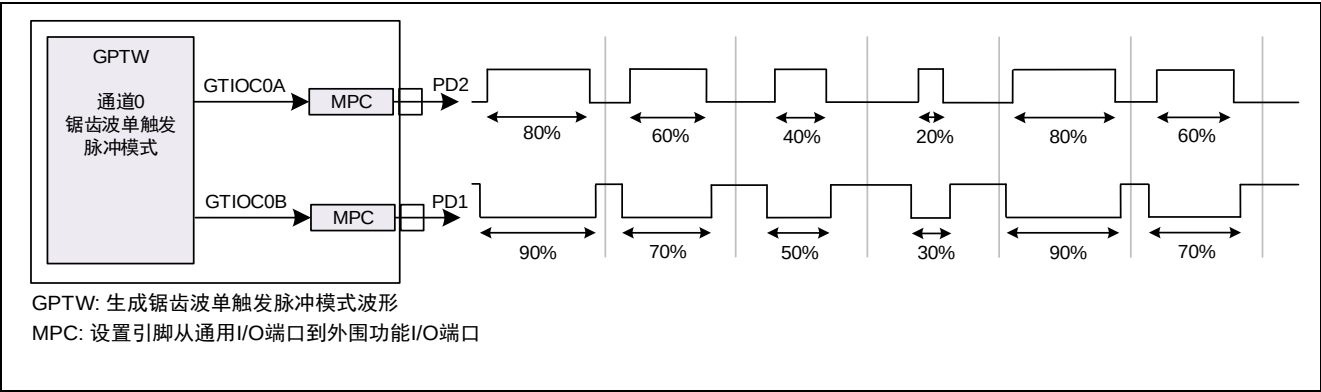


图 4.15 锯齿波单触发脉冲模式输出

4.3.2 运行说明

示例代码运行如图 4.16 所示。通过在 GTCNT 计数器溢出中断（GTCIV0）时修改缓冲寄存器 GTCCRC 和 GTCCRD 的值，在每个周期改变占空比的设置（图 4.16 中（1））。

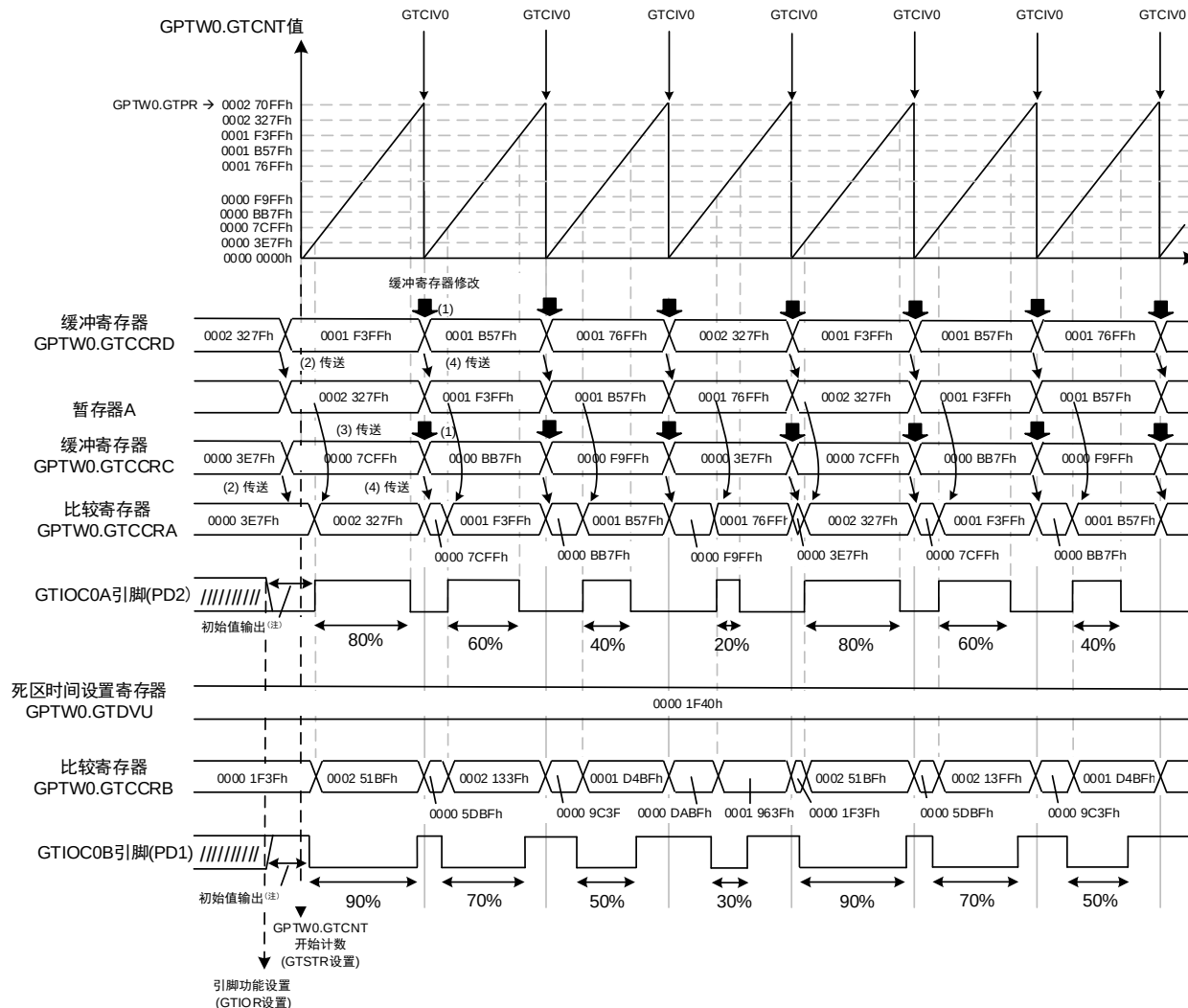
在此示例代码中，使用 Smart Configurator 生成的代码中缓冲寄存器的初始值设置为与比较寄存器相同的值。因此，在计数开始之前，在用户初始化函数 R_Config_GPT0_Create_UserInit 中设置缓冲寄存器的值。

通过强制缓冲传送，寄存器中的设置值分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.16 中（2））。

在计数开始后，通过重复下面步骤 1 和 2 来更新比较寄存器中的数据。

1. 当 GTCCRA 比较匹配发生时，从暂存器 A 传送到比较寄存器 GTCCRA（图 4.16 中（3））。
2. 当 GTCNT 计数器溢出发生时，分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.16 中（4））。

此外，因为使用了死区时间自动设置功能，GTCCRB 寄存器根据 GTCCRA 的更新自动设置。GTDVU 和 GTDVD 设置为相同的值。



注：示例代码波形在 PMR 寄存器设置时开始输出初始值。

图 4.16 示例代码运行

4.3.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参见章节 4.1.4 添加组件。

表 4.6 添加组件

项目	描述
组件	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	锯齿波单触发脉冲模式
资源	GPT0

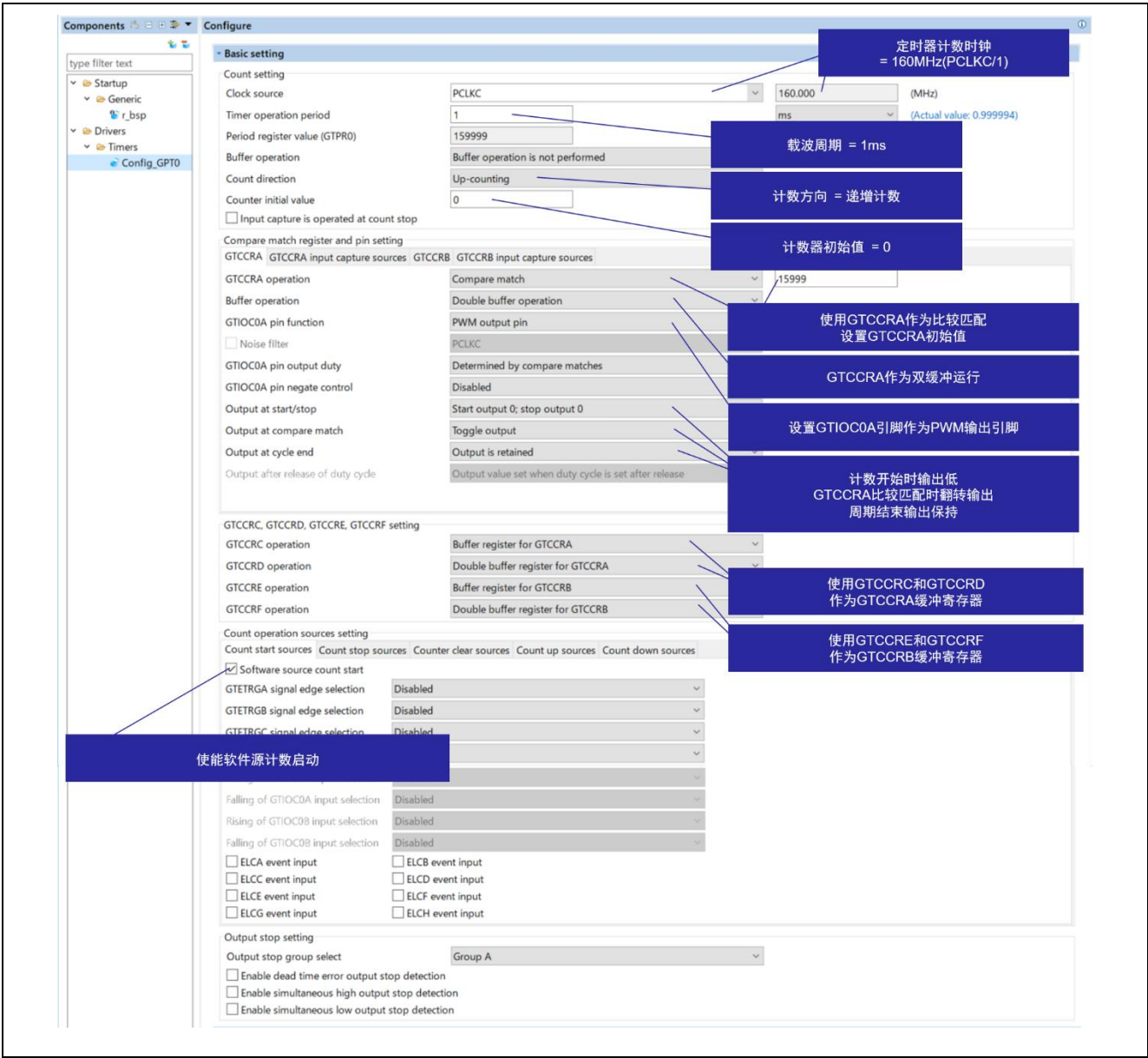


图 4.17 GPT0 设置 (1/2)

Advance setting

Automatic dead time setting

☒ Automatically set GTCCRB0 using GTCCRA0 value and dead time



GTDVU value: 8000 ☐ Enable buffer (GTDVU)

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0 ☐ Enable buffer (GTDVD)

A/D conversion start request setting

GTADTRA GTADTRB

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA): 100

Buffer operation: Buffer operation is not performed

Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTCAI0) Priority: Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTCTB0) Priority: Level 15 (highest)

☐ Enable dead time error interrupt (GDTE0) Priority: Level 15 (highest)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority: Level 15 (highest)

☐ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

GTADTRA/GTADTRB interrupt skipping function: Skip count of 1

Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCCRA interrupt extended skipping function: No extended interrupt skipping

GTCCRB interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.18 GPT0 设置 (2/2)

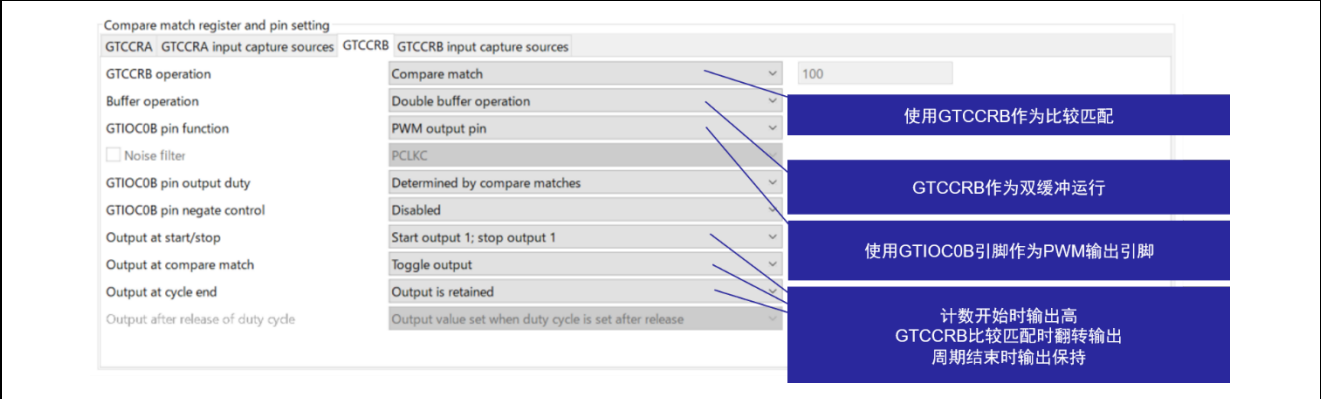


图 4.19 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.3.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

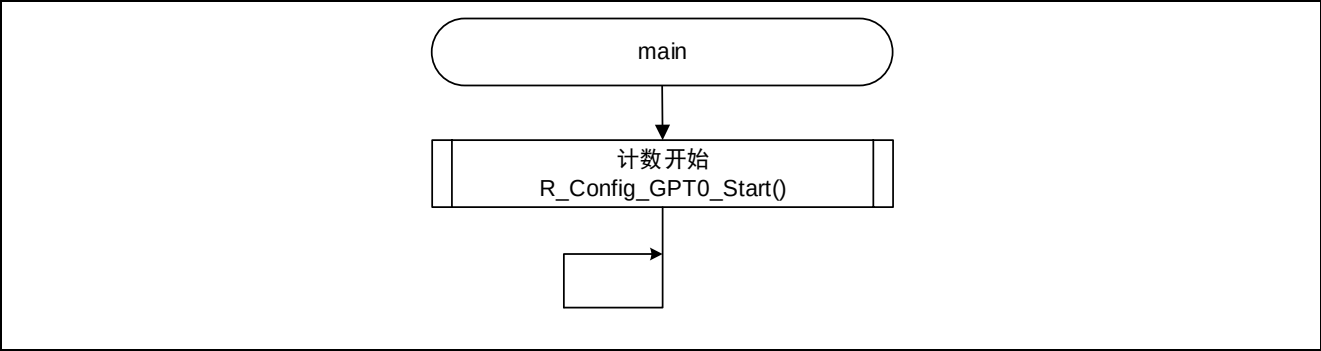


图 4.20 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 用来设置缓冲寄存器的初始值。为了在第一个周期中设置第二个比较匹配寄存器值，在设置缓冲寄存器值后执行强制缓冲传送，然后设置暂存器和比较寄存器的值。此函数在 R_Config_GPT0_Create 函数中调用。

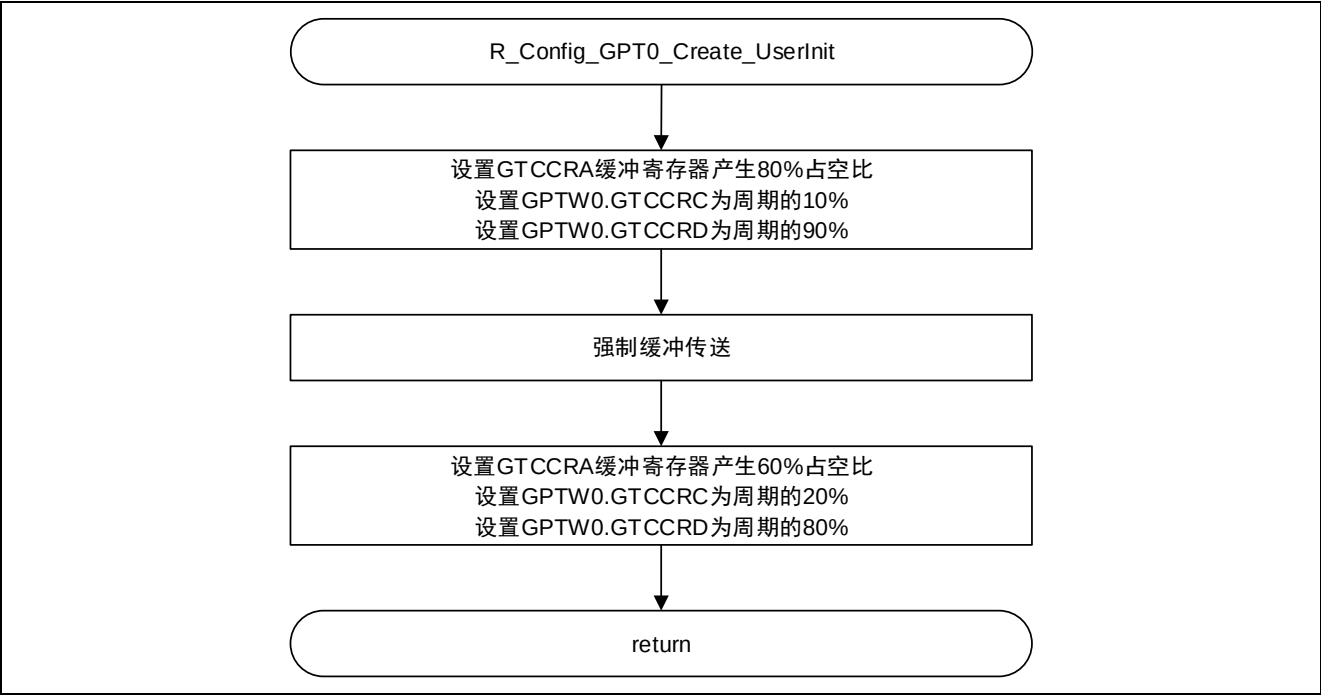


图 4.21 用户初始化函数

GTCIV0 中断处理函数根据当前寄存器值更改缓冲寄存器 GTCCRC 和 GTCCRD 的值。

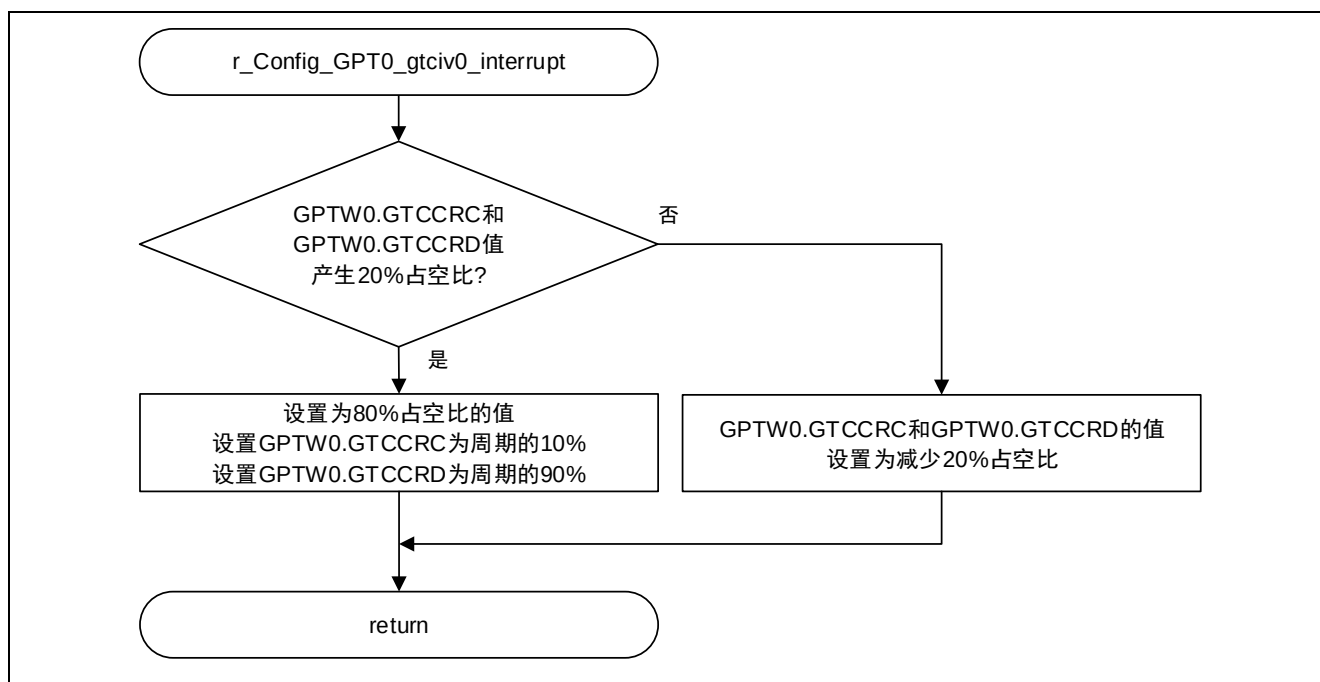


图 4.22 GTCIV0 中断处理函数

4.3.5 相关运行

4.3.5.1 当不使用死区时间自动设置功能

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_1shotpls.zip

图 4.23 显示了不使用死区时间自动设置功能 (GTDTCR.TDE 位为 0) 的运行示例。

在上述示例代码中, 由于周期占空比随周期变化, 所以缓冲寄存器 GTCCRC, GTCCRD, GTCCRE 和 GTCCRF 的值通过 GTCNT 计数器溢出中断 (GTCIV0) 修改 (图 4.23 中 (1))。

在此示例代码中, 使用 Smart Configurator 生成的代码中缓冲寄存器的初始值设置为与比较寄存器相同的值。因此, 在计数开始之前, 在用户初始化函数 R_Config_GPT0_Create_UserInit 中设置缓冲寄存器的值。通过强制缓冲传送, 寄存器中的设置值分别从缓冲寄存器 GTCCRD (GTCCRF) 传送到暂存器 A (B) 和从缓冲寄存器 GTCCRC (GTCCRE) 传送到比较寄存器 GTCCRA (GTCCRB) (图 4.23 中 (2))。

在计数开始后, 通过重复下面步骤 1 和 2 来更新比较寄存器中的数据。

1. 当 GTCCRA (GTCCRB) 比较匹配发生时, 从暂存器 A (B) 传送到比较寄存器 GTCCRA (GTCCRB) (图 4.23 中 (3))。
2. 当 GTCNT 计数器溢出发生时, 分别从缓冲寄存器 GTCCRD (GTCCRF) 传送到暂存器 A (B) 和从缓冲寄存器 GTCCRC (GTCCRE) 传送到比较寄存器 GTCCRA (GTCCRB) (图 4.23 中 (4))。

此外, 如图 4.16 中的运行所示, 确保相同的死区时间段。

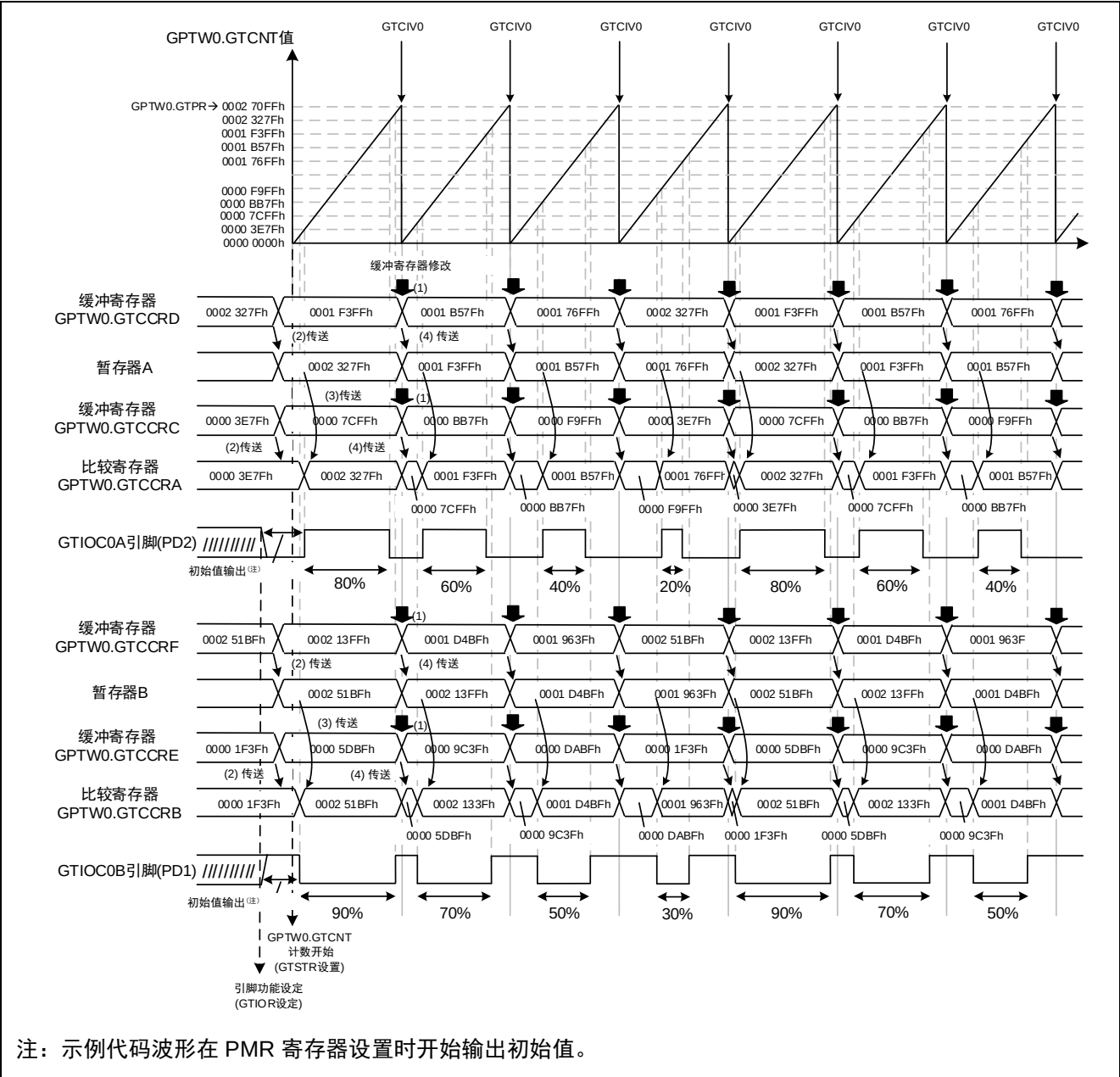


图 4.23 当不使用死区时间自动设置功能（锯齿波单触发脉冲模式）

4.3.6 使用时的注意事项

4.3.6.1 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较匹配寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

缓冲寄存器 GTCCRC 和 GTCCRD 的值应设置为满足以下限制。如果不满足这些限制，可能无法获得具有安全死区时间的正确输出波形。

- 在递增计数中：
 $GTCCRC < GTCCRD$
 $GTCCRC > GTDVU$
 $GTCCRD < GTPR - GTDVD$
- 在递减计数中：
 $GTCCRC > GTCCRD$
 $GTCCRC < GTPR - GTDVU$
 $GTCCRD > GTDVD$

此外，如果死区时间没有自动设置，应设置缓冲寄存器 GTCCRC 和 GTCCRD（或 GTCCRE 和 GTCCRF）满足以下限制。如果不满足这些限制，则不会发生两次比较匹配并且无法执行脉冲输出。

- 在递增计数中： $0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR$
- 在递减计数中： $GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0$

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)，(3) 在锯齿波单触发脉冲模式下进行死区时间自动设置时和 (4) 在锯齿波单触发脉冲模式下未进行死区时间自动设置时。

4.4 锯齿波 PWM 模式 三相互补

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_pwm_3phase.zip

4.4.1 概要

通道 0, 1 和 2 可用于 GPTW 锯齿波 PWM 来根据 GTCCR 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用锯齿波 PWM 模式来重复输出每个 3 相也就是总共 6 相的恒定占空比的 PWM 波形（正反相）。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波 PWM 模式
- 使用通道 0, 1 和 2（通道号: n = 0, 1 和 2）
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz（PCLKC/1）
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCnA 引脚为 PWM 输出引脚
 - 计数开始时输出低
 - GTCCRA 比较匹配时输出高
 - 周期结束时输出低
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCnB 引脚为 PWM 输出引脚
 - 计数开始时输出高
 - GTCCRB 比较匹配时输出低
 - 周期结束时输出高
- 不使用缓冲运行
- 使能软件源计数启动

在 Smart Configurator 中设置。
设置方法请参考章节 4.4.3。

此示例代码的锯齿波 PWM 模式输出如下所示。

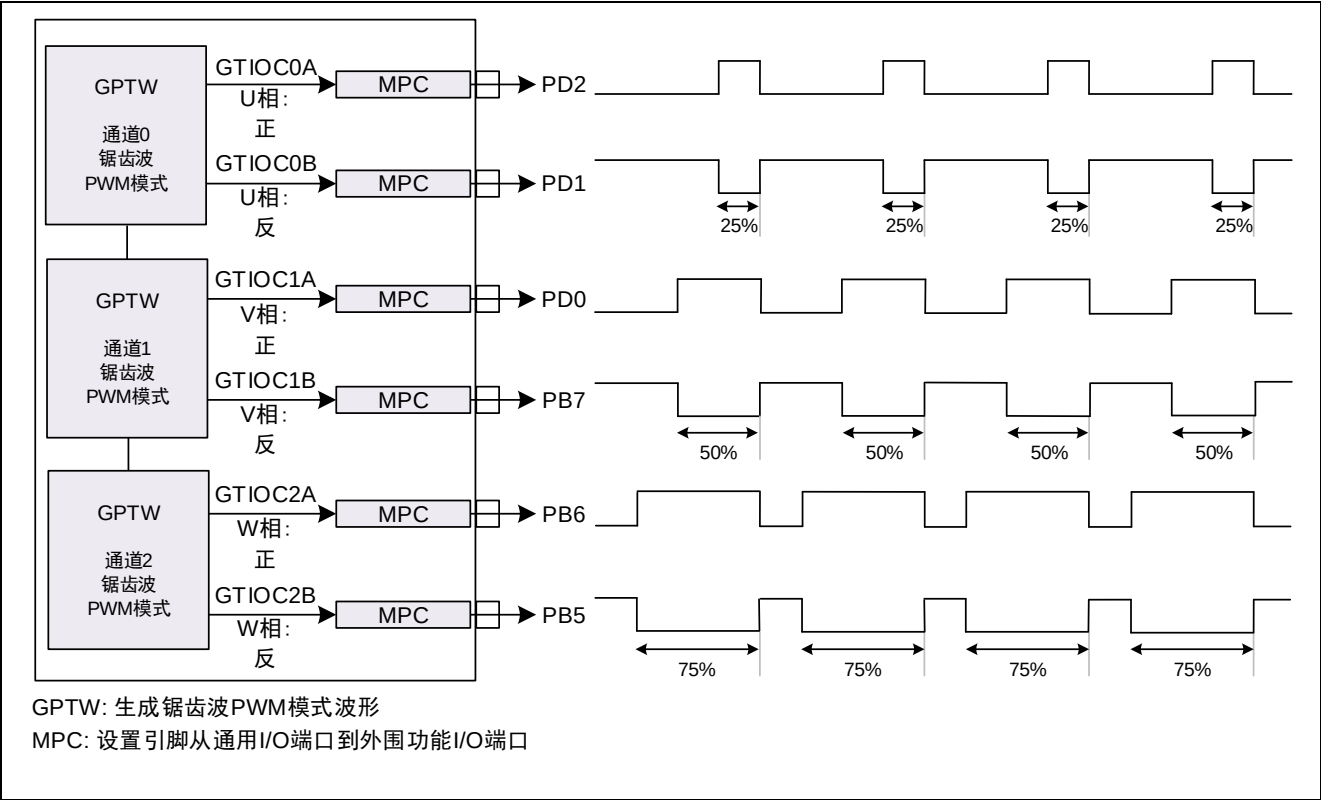


图 4.24 锯齿波 PWM 模式输出

4.4.2 运行说明

示例代码的运行如图 4.25 所示。每次每个通道的 GTCCRA 和 GTCCRB 寄存器比较匹配发生以及 GTCNT 计数器溢出（计数器清除）发生时，PWM 输出引脚翻转输出。获得与 MTU 复位同步 PWM 模式相同的输出波形。详情参考章节 3.8。

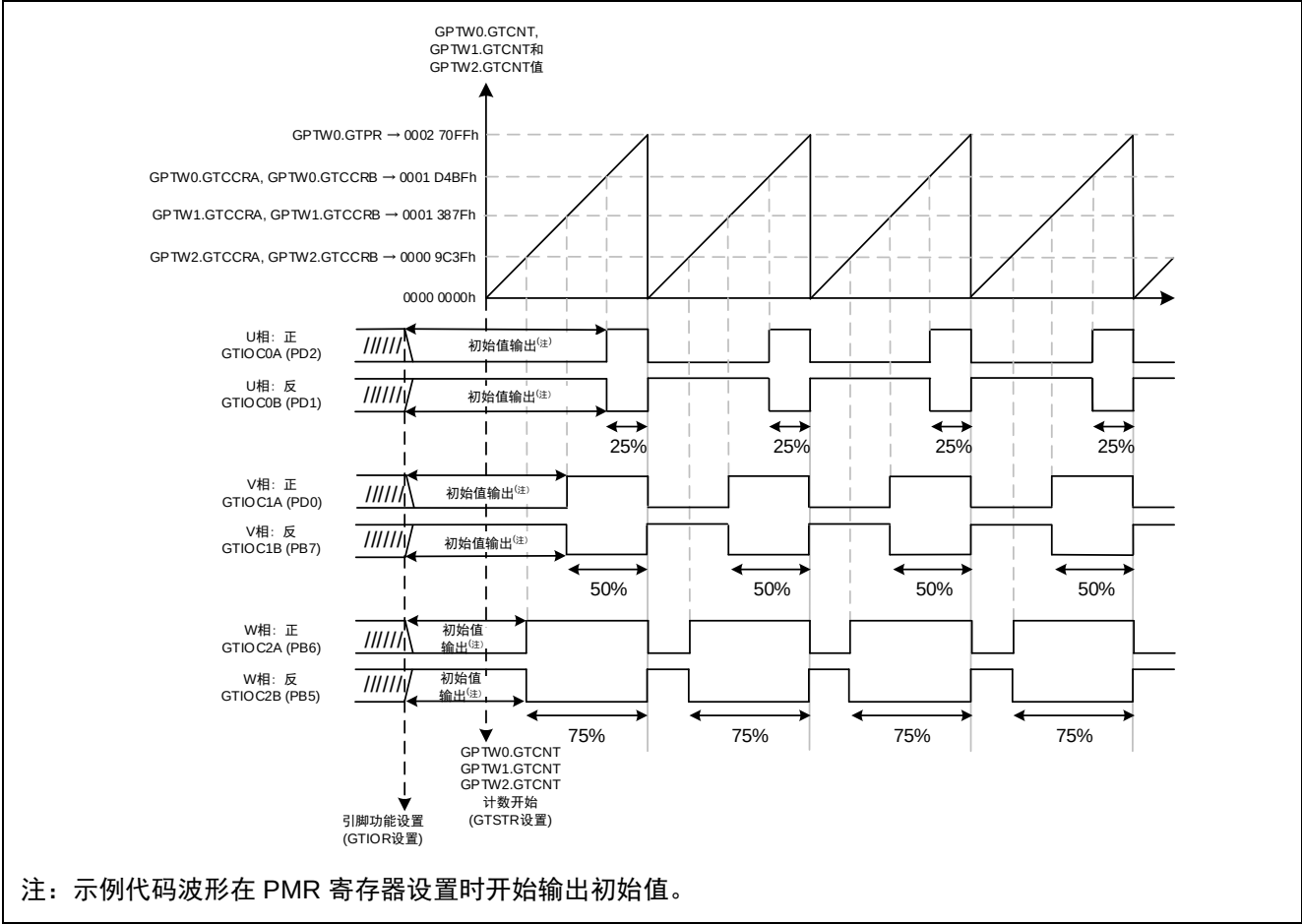


图 4.25 示例代码运行

4.4.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.7 添加组件

项目	描述		
组件	通用 PWM 定时器		
配置名称	Config_GPT0	Config_GPT1	Config_GPT2
工作模式	锯齿波 PWM 模式		
资源	GPT0	GPT1	GPT2

图 4.26 和图 4.27 显示 Config_GPT0 设置。相同的设置适用于 GPT1 和 GPT2。

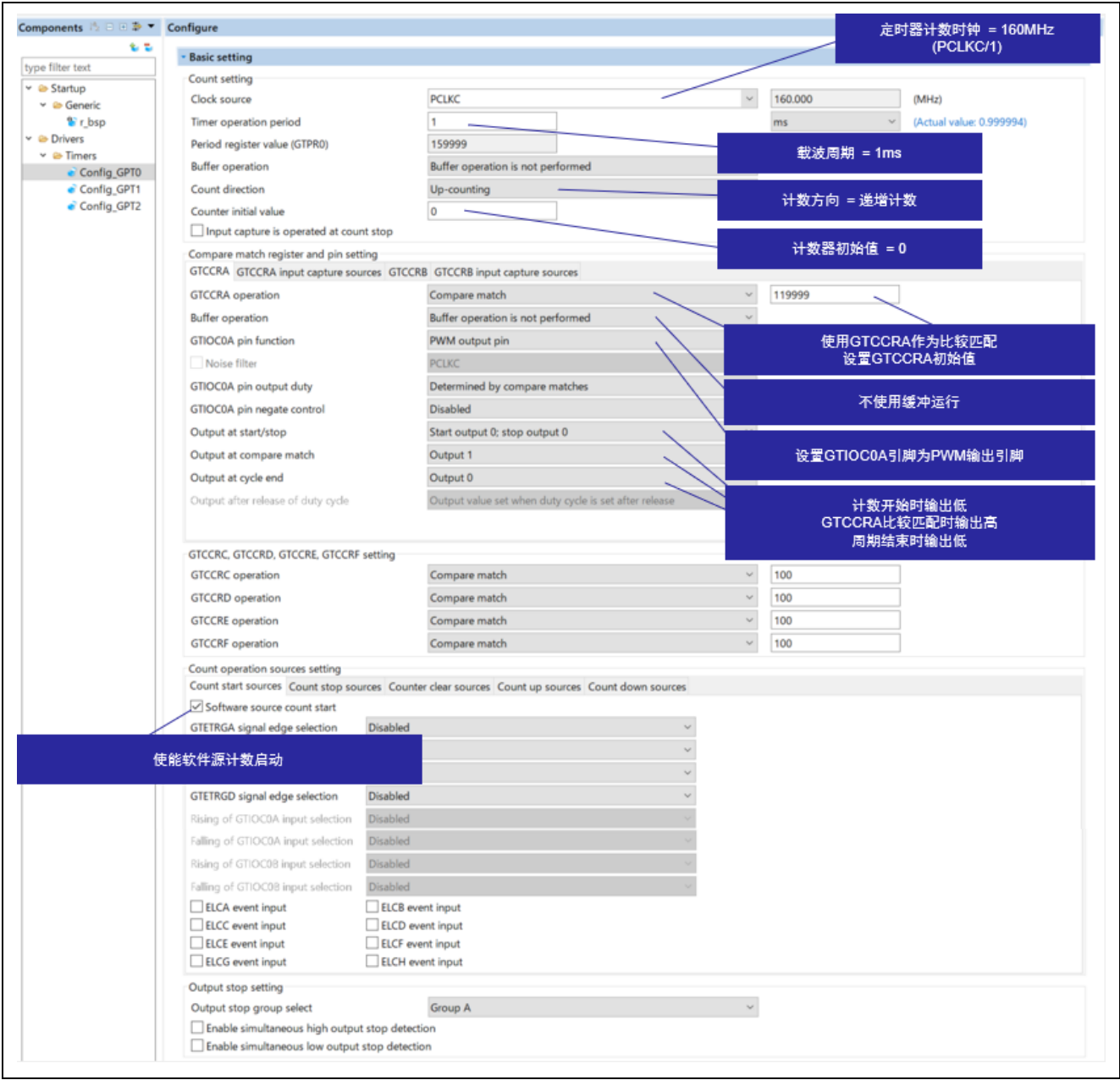


图 4.26 GPT0 设置

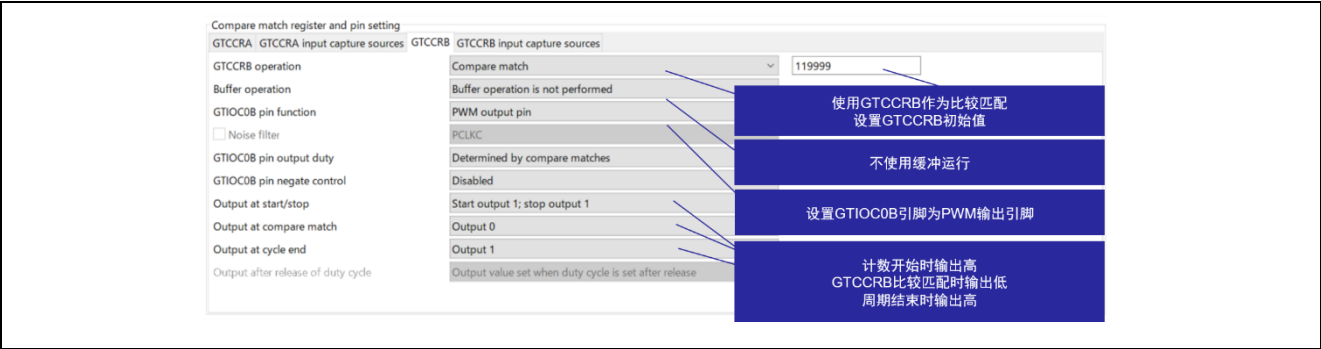


图 4.27 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.4.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 gpt0_gpt1_gpt2_start 并开始计数。

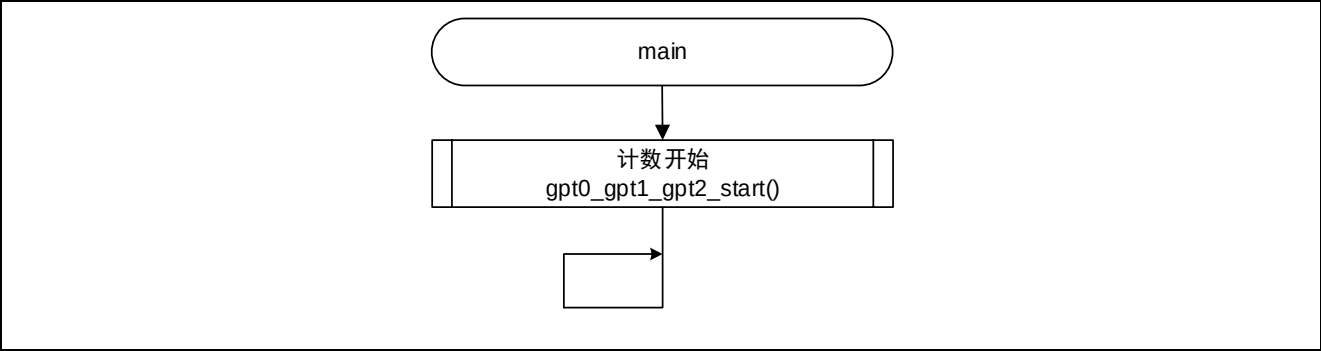


图 4.28 主函数

在计数开始函数中，GPT0，GPT1 和 GPT2 开始计数。
此函数是在 Smart Configurator 生成代码后新创建的。

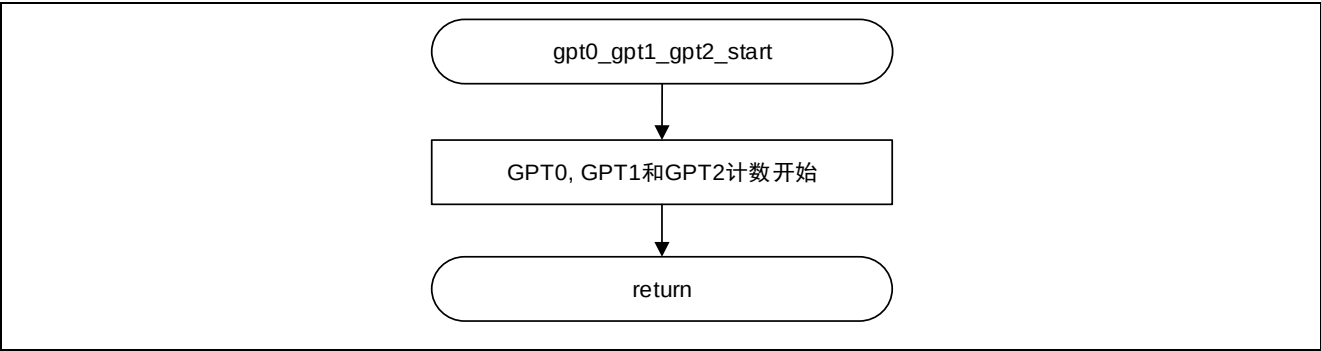


图 4.29 计数开始函数

4.4.5 使用时的注意事项

4.4.5.1 多通道计数开始

在此示例代码中，定时器软件启动寄存器 GTSTR 的 CSTR0, CSTR1 和 CSTR2 位在 gpt0_gpt1_gpt2_start 函数中同时设置，以同时启动 GPTW0, GPTW1 和 GPTW2 计数。

当使用 Smart Configurator 生成的 R_Config_GPT0_Start, R_Config_GPT1_Start 和 R_Config_GPT2_Start 函数时，因为每个函数调用时间不同，计数开始时间可能不同。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.8.1 软件同步操作。

4.4.5.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

设置比较寄存器 GTCCRA 和 GTCCRB 的值大于 0000 0001h 但小于 GTPR 寄存器的值。如果寄存器设置为 0000 0000h 或者与 GTPR 寄存器相同的值，则只有比较寄存器设置为 0000 0000h 或者比较寄存器设置为与 GTPR 寄存器相同的值时，才会在周期内发生比较匹配。如果比较寄存器被设置为超过 GTPR 寄存器的设置值时，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 至 F)，(5) 锯齿波 PWM 模式下。

4.5 锯齿波单触发脉冲 三相互补

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_1shotpls_3phase_dt.zip

4.5.1 概要

通道 0, 1 和 2 可用于 GPTW 锯齿波单触发脉冲模式来输出带死区时间的三相互补 PWM 波形。

此示例代码描述了在锯齿波单触发脉冲模式下使用死区时间自动设置功能并重复输出以下波形。双缓冲器用于生成具有左右不对称 PWM 波形的每个占空比。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

在比较寄存器 GTCCRA 的比较匹配时将暂存器 A 的值传送到比较寄存器 GTCCRA 来改变占空比, 并在 GTCNT 计数器溢出发生时将缓冲寄存器 GTCCRD 的值传送到暂存器 A 并将缓冲寄存器 GTCCRC 的值传送到比较寄存器 GTCCRA。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波单触发脉冲模式
- 使用通道 0, 1 和 2 (通道号: n = 0, 1 和 2)
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCNnA 引脚为 PWM 输出引脚
 - 计数开始时输出高, 计数停止时输出高
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCNnB 引脚为 PWM 输出引脚
 - 计数开始时输出低, 计数停止时输出低
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用缓冲寄存器
 - GTCCRA 和 GTCCRB 作为双缓冲运行
 - 使用 GTCCRC 和 GTCCRD 作为 GTCCRA 的缓冲寄存器
 - 使用 GTCCRE 和 GTCCRF 作为 GTCCRB 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.32

在 Smart Configurator 中设置。
设置方法请参考章节 4.5.3。

此示例代码的锯齿波单触发脉冲模式输出如下所示。

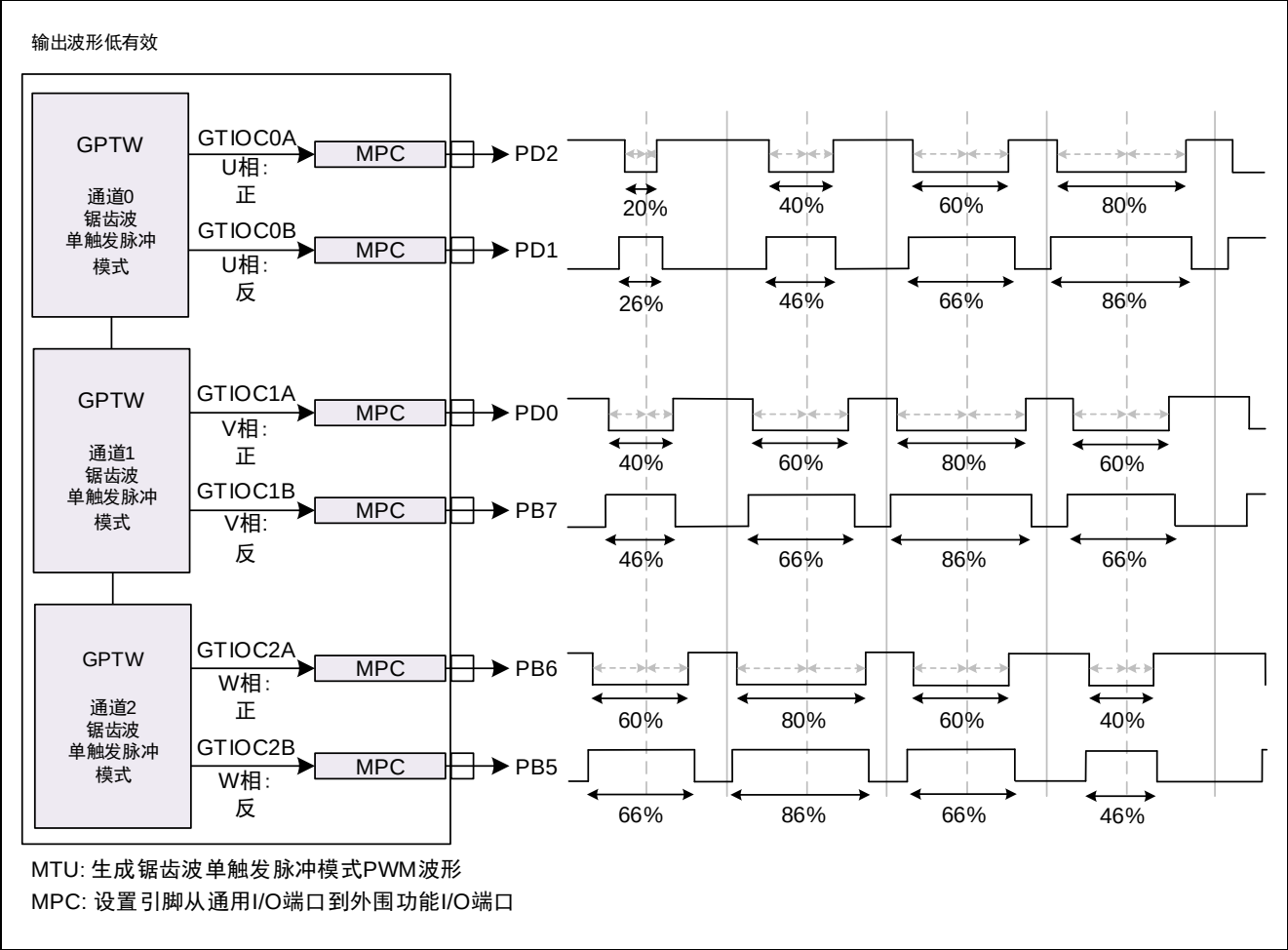


图 4.30 锯齿波单触发脉冲模式输出

4.5.2 运行说明

示例代码的运行如图 4.32 所示。通过在 GTCNT 计数器溢出中断（GTCIV0）中修改缓冲寄存器 GTCCRC 和 GTCCRD 的值在每个周期改变占空比的设置（图 4.32 中（1））。

在此示例代码中，使用 Smart Configurator 生成的代码中缓冲寄存器的初始值设置为与比较寄存器相同的值。因此，缓冲寄存器的值在计数开始之前在用户初始化函数 R_Config_GPT0_Create_UserInit 中设置。

通过强制缓冲传送，寄存器中设置的值分别从缓冲寄存器 GTCCRD 传送到暂存器 A，并从缓冲区寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.32 中（2））。

计数开始后，重复下面的步骤 1 和 2 以更新比较寄存器中的数据。

1. 当 GTCCRA 比较匹配发生时，从暂存器 A 传送到比较寄存器 GTCCRA（图 4.32 中（3））。
2. 当 GTCNT 计数器溢出发生时，分别从缓冲区寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.32 中（4））。

此外，由于使用了死区时间自动功能，GTCCRB 寄存器根据 GTCCRA 的更新自动设置。GTDVU 和 GTDVD 设置为相同的值。

计数开始后，GTCCRA 和 GTCCRB 比较匹配发生，反相输出 OFF，正相输出 ON（图 4.32 中（5））。

- 左右不对称 PWM 波形输出
每个周期的前半部分和后半部分生成不同的占空比。

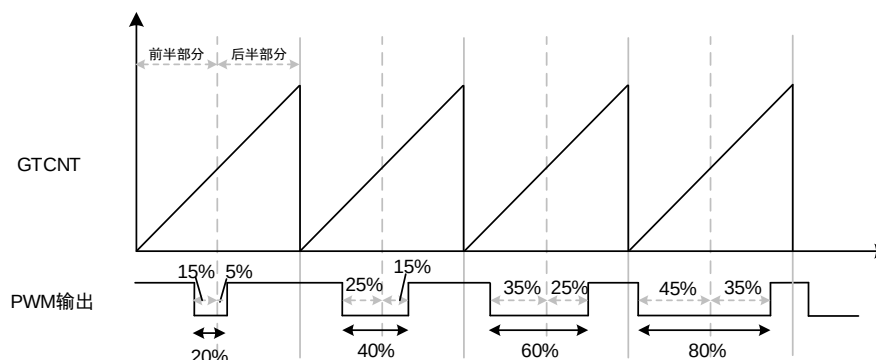


图 4.31 左右不对称 PWM 波形输出



注：示例代码波形在 PMR 设置时开始输出初始值。

图 4.32 示例代码运行

4.5.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考 4.1.4 添加组件。

表 4.8 添加组件

项目	描述		
组件	通用 PWM 定时器		
配置名称	Config_GPT0	Config_GPT1	Config_GPT2
工作模式	锯齿波单触发脉冲模式		
资源	GPT0	GPT1	GPT2

图 4.33 至图 4.35 显示了 Config_GPT0 设置。相同的设置适用于 GPT1 和 GPT2。

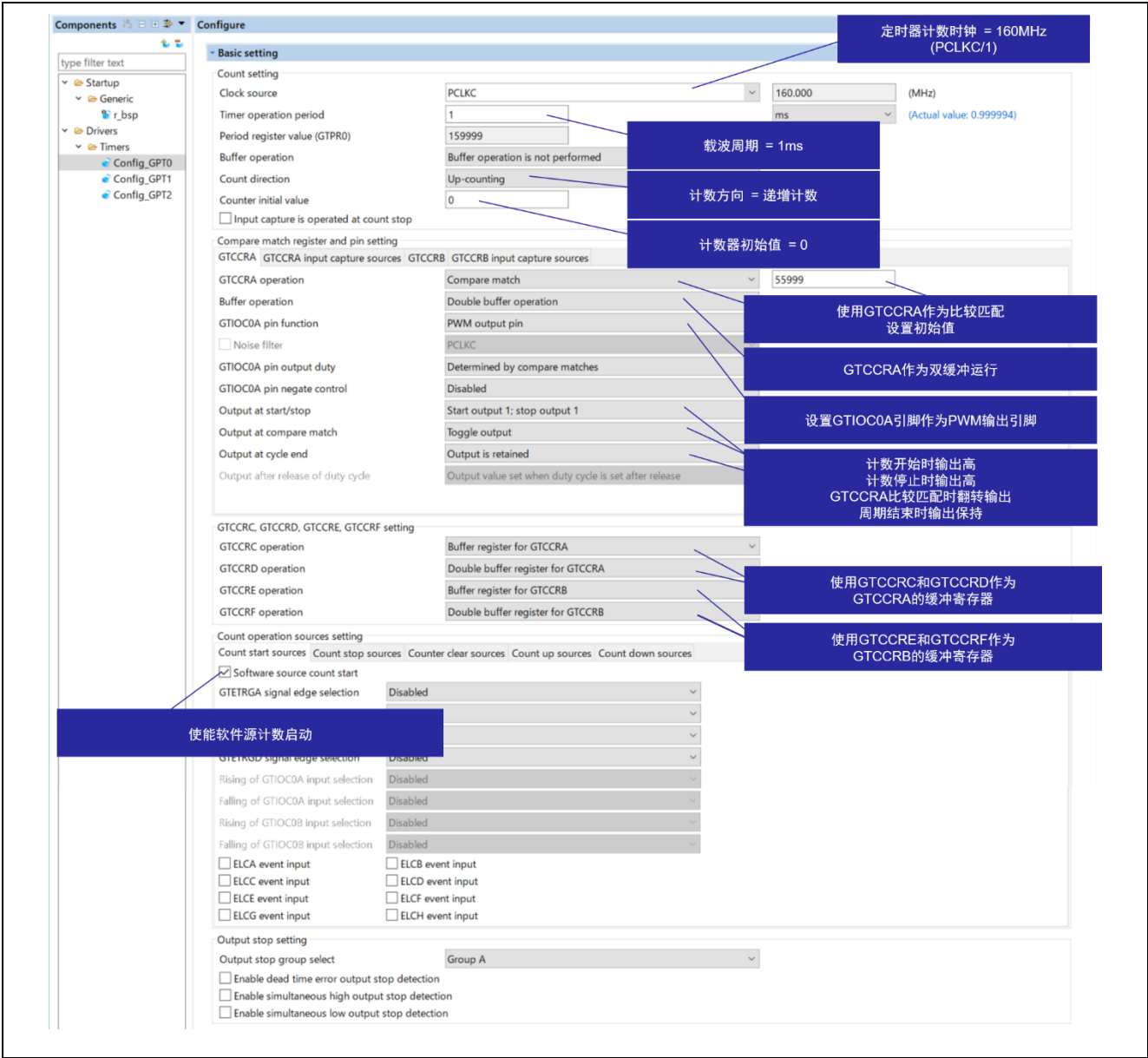
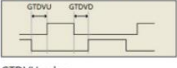


图 4.33 GPT0 设置 (1/2)

Advance setting

Automatic dead time setting

☒ Automatically set GTCCRB0 using GTCCRA0 value and dead time



GTDVU value: 4800 ☐ Enable buffer (GTDBU)

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0 ☐ Enable buffer (GTDBD)

A/D conversion start request setting

GTADTRA GTADTRB

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRB)

Compare match value (GTADTRA): 100

Buffer operation: Buffer operation is not performed

Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTICIA0) Priority: Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTICIB0) Priority: Level 15 (highest)

☐ Enable dead time error interrupt (GDTED) Priority: Level 15 (highest)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority: Level 15 (highest)

☐ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

Skip count of 1

Link GTCIB0 with GTCIV0/GTCIU0 interrupt skipping function

Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCCRA interrupt extended skipping function: No extended interrupt skipping

GTCCRB interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.34 GPT0 设置 (2/2)

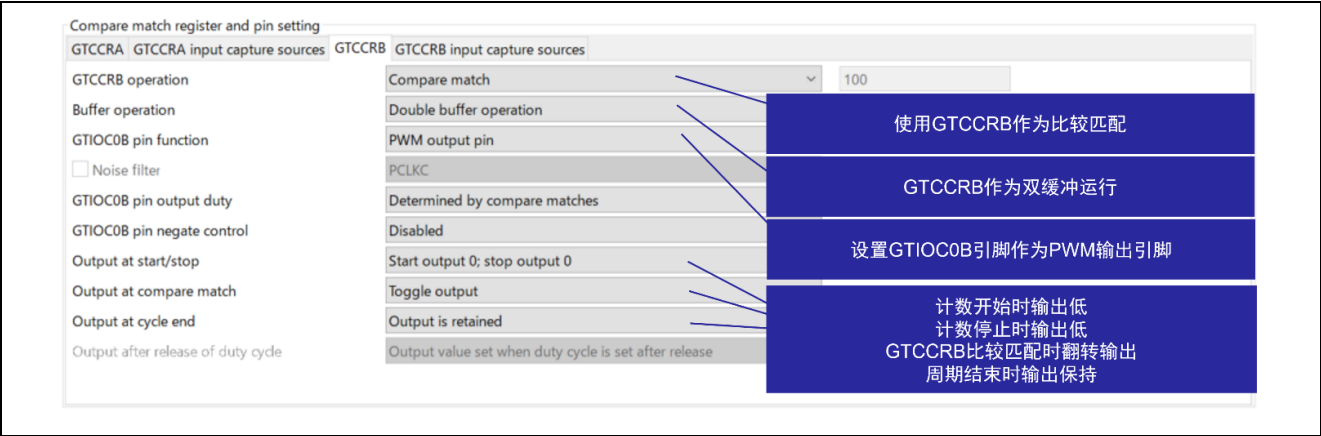


图 4.35 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.5.4 流程图

以下流程图显示了 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 gpt0_gpt1_gpt2_start 并开始计数。

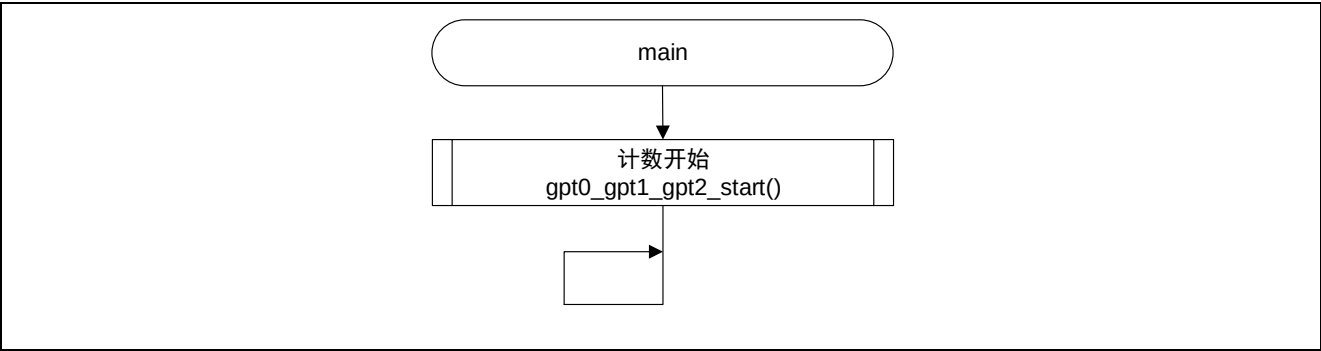


图 4.36 主函数

在计数开始函数中，GPT0、GPT1 和 GPT2 计数在使能 GTCIV0 中断后启动。
此功能是在 Smart Configurator 生成代码后新创建的。

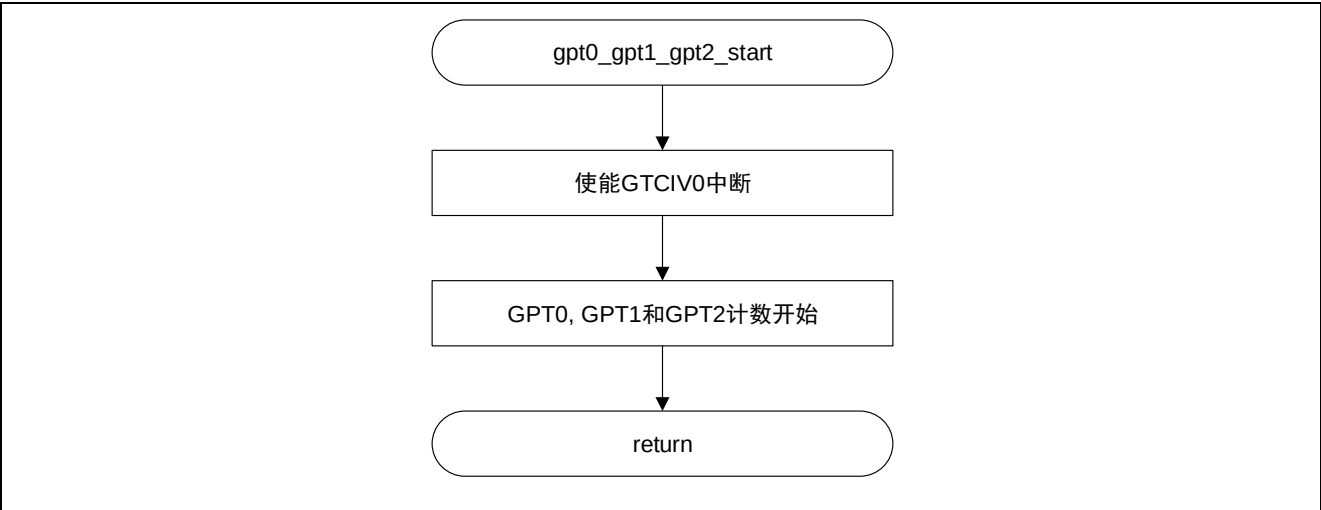


图 4.37 计数开始函数

在主函数之前执行的用户初始化函数 `R_Config_GPT0_Create_UserInit` 中设置缓冲寄存器的初始值并初始化变量。此函数在 `R_Config_GPT0_Create` 函数中调用。

此示例代码使用以下变量。

- `s_uptmg_prv`: 用于保存上一次 `GPTW0.GTCCRC` 寄存器的值（在 `GTIOC0A` 引脚输出上升时间）

`R_Config_GPT1_Create_UserIni` 和 `R_Config_GPT2_Create_UserInit` 也执行相同的流程。

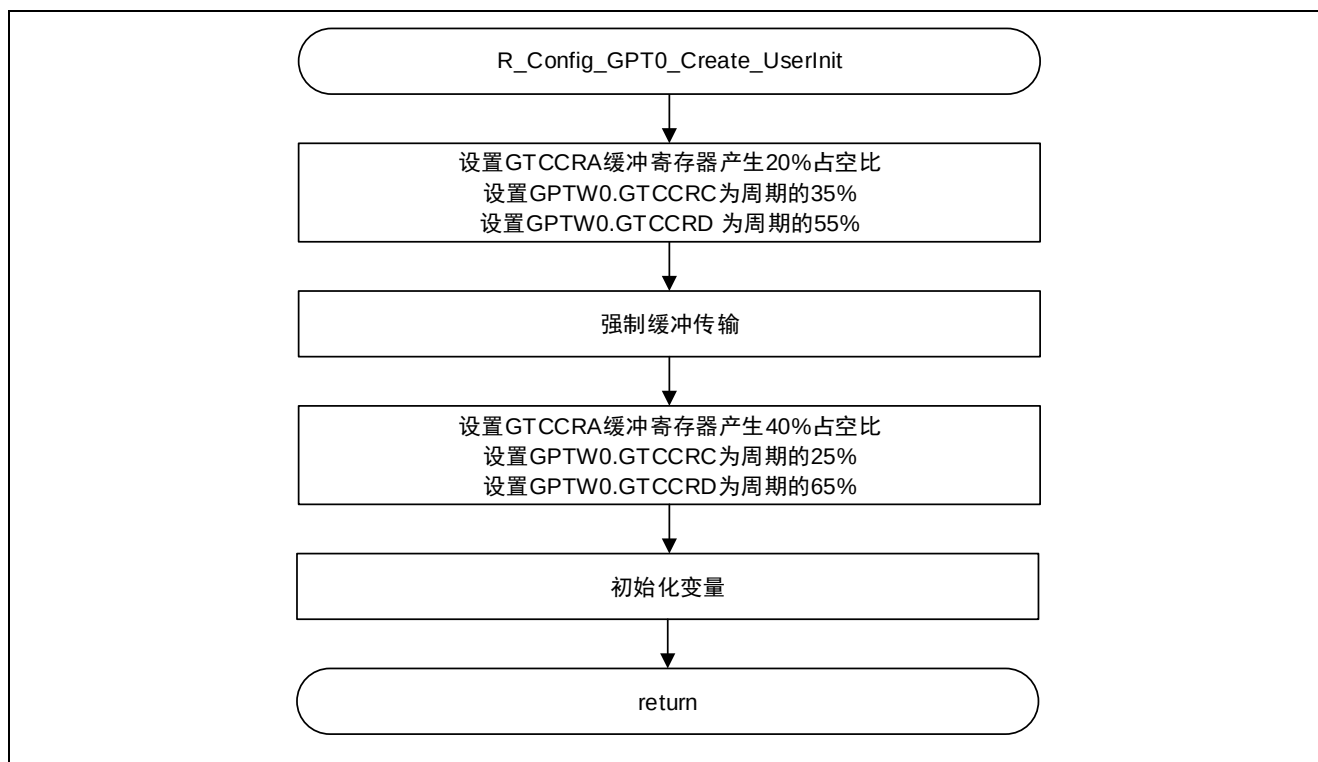


图 4.38 用户初始化函数

GTCIV0 中断处理函数根据缓冲寄存器 GPTW0.GTCCRC, GPTW1.GTCCRC 和 GPTW2.GTCCRC 的当前值以及上一次缓冲寄存器中设置的值来改变缓冲寄存器的值。

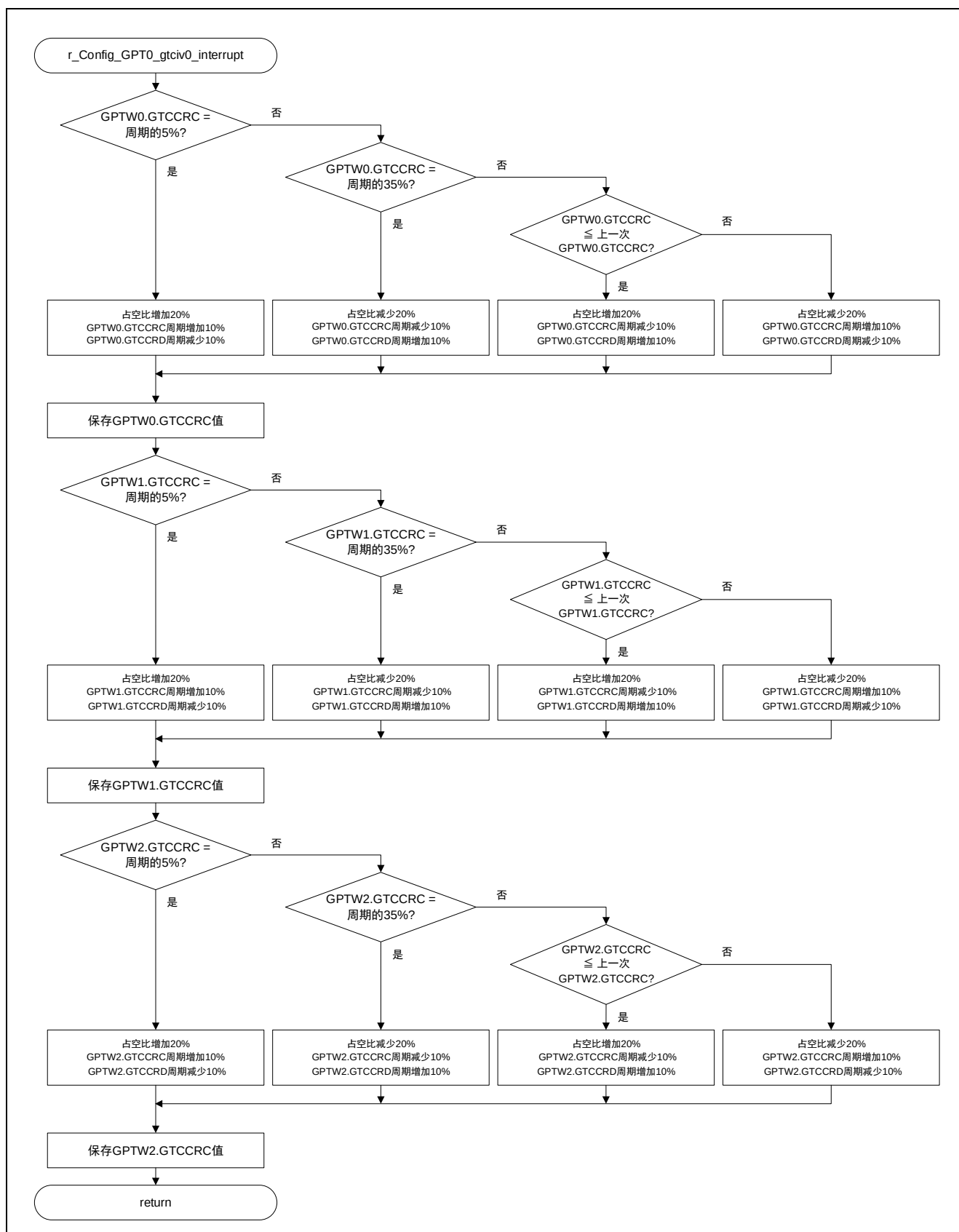


图 4.39 GTCIV0 中断处理函数

4.5.5 相关运行

4.5.5.1 每个间隔的单独死区时间自动设置

本示例代码使用死区时间自动设置功能，通过设置 GTDTCR.TDFER 位为 1，在反相的前侧和后侧生成具有公共转换时序的死区时间。

在死区时间自动设置功能中，可以分别设置波形的前侧和后侧的死区时间。反相的前侧转换时序的死区时间在 GTDVU 寄存器中设置，后侧的死区时间在 GTDVD 寄存器中设置。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

4.5.6 使用时的注意事项

4.5.6.1 多通道计数开始

在此示例代码中，定时器软件启动寄存器 GTSTR 的 CSTR0、CSTR1 和 CSTR2 位在 gpt0_gpt1_gpt2_start 函数中同时设置，以同时启动 GPTW0、GPTW1 和 GPTW2 计数。

使用 Smart Configurator 生成的 R_Config_GPT0_Start、R_Config_GPT1_Start 和 R_Config_GPT2_Start 函数时，因为每个函数调用时间不同，计数开始时间可能不同。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.8.1 软件同步操作。

4.5.6.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较匹配寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

缓冲寄存器 GTCCRC 和 GTCCRD 的值应设置为满足以下限制。如果不满足限制，可能无法获得具有安全死区时间的正确输出波形。

- 在递增计数中：
 $GTCCRC < GTCCRD$
 $GTCCRC > GTDVU$
 $GTCCRD < GTPR - GTDVD$
- 在递减计数中：
 $GTCCRC > GTCCRD$
 $GTCCRC < GTPR - GTDVU$
 $GTCCRD > GTDVD$

此外，如果未自动设置死区时间，则应将缓冲寄存器 GTCCRC 和 GTCCRD（或 GTCCRE 和 GTCCRF）设置为满足以下限制。如果不满足限制，则不会发生两个比较匹配，并且无法执行脉冲输出。

- 在递增计数中： $0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR$
- 在递减计数中： $GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0$

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 在比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)，(3) 在锯齿波单触发脉冲模式下进行死区时间自动设置时以及 (4) 在锯齿波单触发脉冲模式下未进行死区时间自动设置时。

4.6 三角波 PWM 模式 1

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm1_dt.zip

4.6.1 概要

GPTW 三角波 PWM 模式 1 可用于输出带死区时间的三相互补 PWM 波形。

此示例代码描述了在三角波 PWM 模式 1（在波谷 32 位传送）下使用死区时间自动设置功能并重复输出以下波形。每个占空比使用缓冲器（不是双缓冲器）生成对称的 PWM 波形。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

当 GTCNT 计数器下溢发生时，通过将缓冲寄存器 GTCCRC 的值传送到比较寄存器 GTCCRA 来改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用三角波 PWM 模式 1
- 使用通道 0, 1 和 2（通道号: n = 0, 1 和 2）
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz（PCLKC/1）
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCN_A 引脚为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出高，计数结束时输出高
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCN_B 引脚为 PWM 输出引脚
 - 使用 GTCCRB 作为比较匹配
 - 计数开始时输出低，计数结束时输出低
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器下溢中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 4.42

在 Smart Configurator 中设置。
设置方法请参考章节 4.6.3。

此示例代码的三角波 PWM 模式 1 输出如下所示。

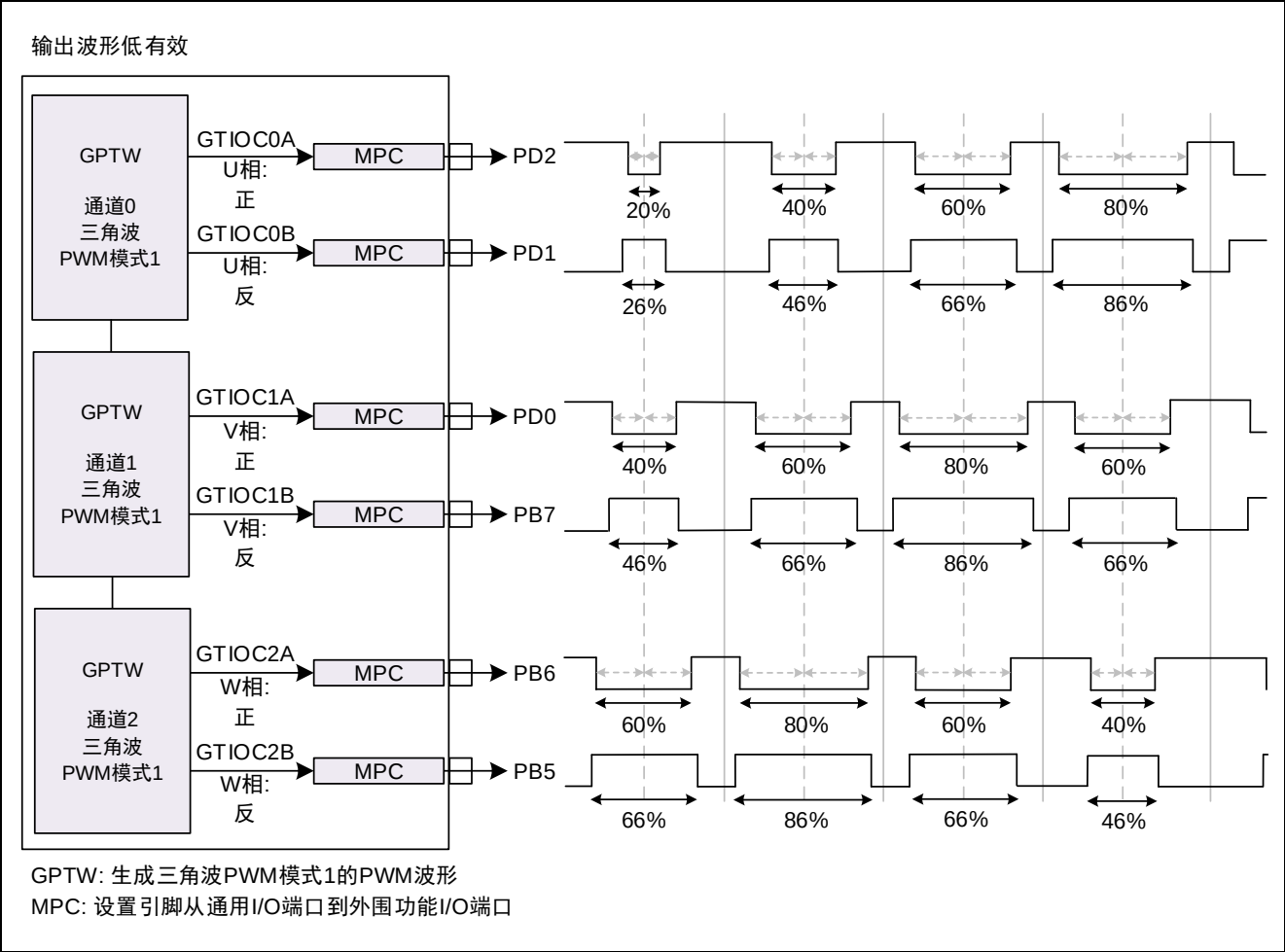


图 4.40 三角波 PWM 模式 1 输出

4.6.2 运行说明

示例代码的运行如图 4.42 所示。当 GTCNT 计数器下溢中断（GTCIU0）产生时，通过修改缓冲寄存器 GTCCRC 的值来改变每个周期的占空比的设置。（图 4.42 中（1））。

当 GTCNT 计数器下溢发生时，通过传送缓冲寄存器的值到比较寄存器 GTCCRA，示例代码使用三角波 PWM 模式 1（在波谷传送）来更新数据（图 4.42 中（2））。

此外，因为使用死区时间自动设置功能，GTCCRB 寄存器根据 GTCCRA 的更新自动设置。GTDVU 和 GTDVD 设置为相同值。此示例代码在计数开始时自动设置 GTCCRB 的值（图 4.42 中（3））。

在计数开始后，比较寄存器和计数器寄存器之间发生比较匹配，反相输出 OFF，然后正相输出 ON（图 4.42 中（4））。

- 对称 PWM 波形输出

每个周期的占空比在递增计数期间和递减计数期间产生 1/2 占空比。

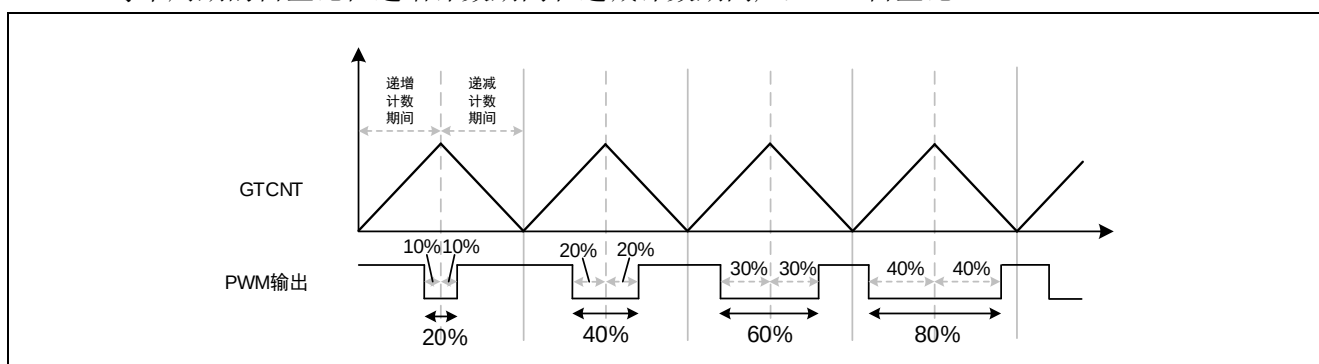


图 4.41 对称 PWM 输出波形

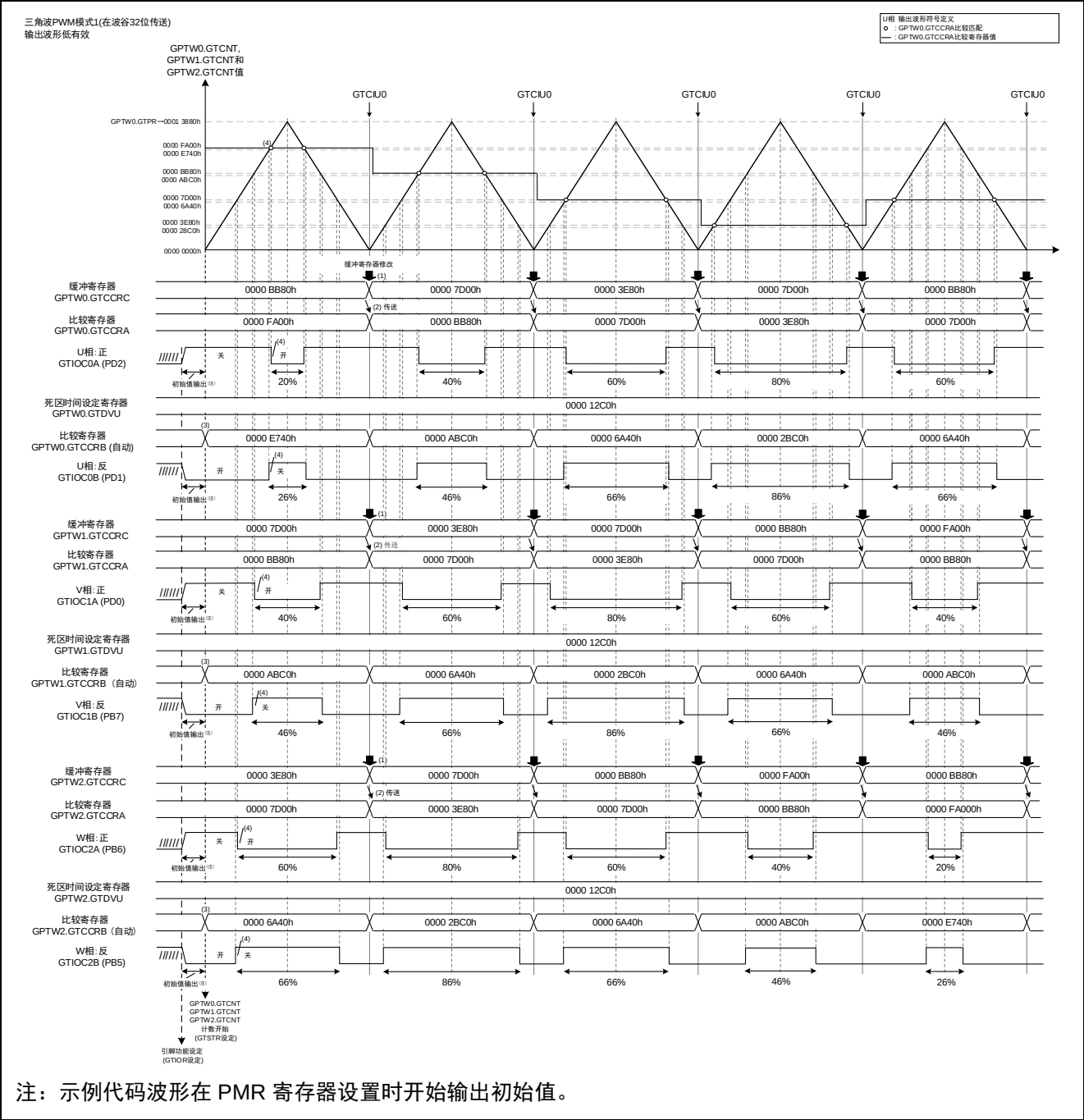


图 4.42 示例代码运行

4.6.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.9 添加组件

项目	描述		
组件	通用 PWM 定时器		
配置名称	Config_GPT0	Config_GPT1	Config_GPT2
工作模式	三角波 PWM 模式 1		
资源	GPT0	GPT1	GPT2

图 4.43 至图 4.45 显示 Config_GPT0 设置。相同的设置适用于 GPT1 和 GPT2。

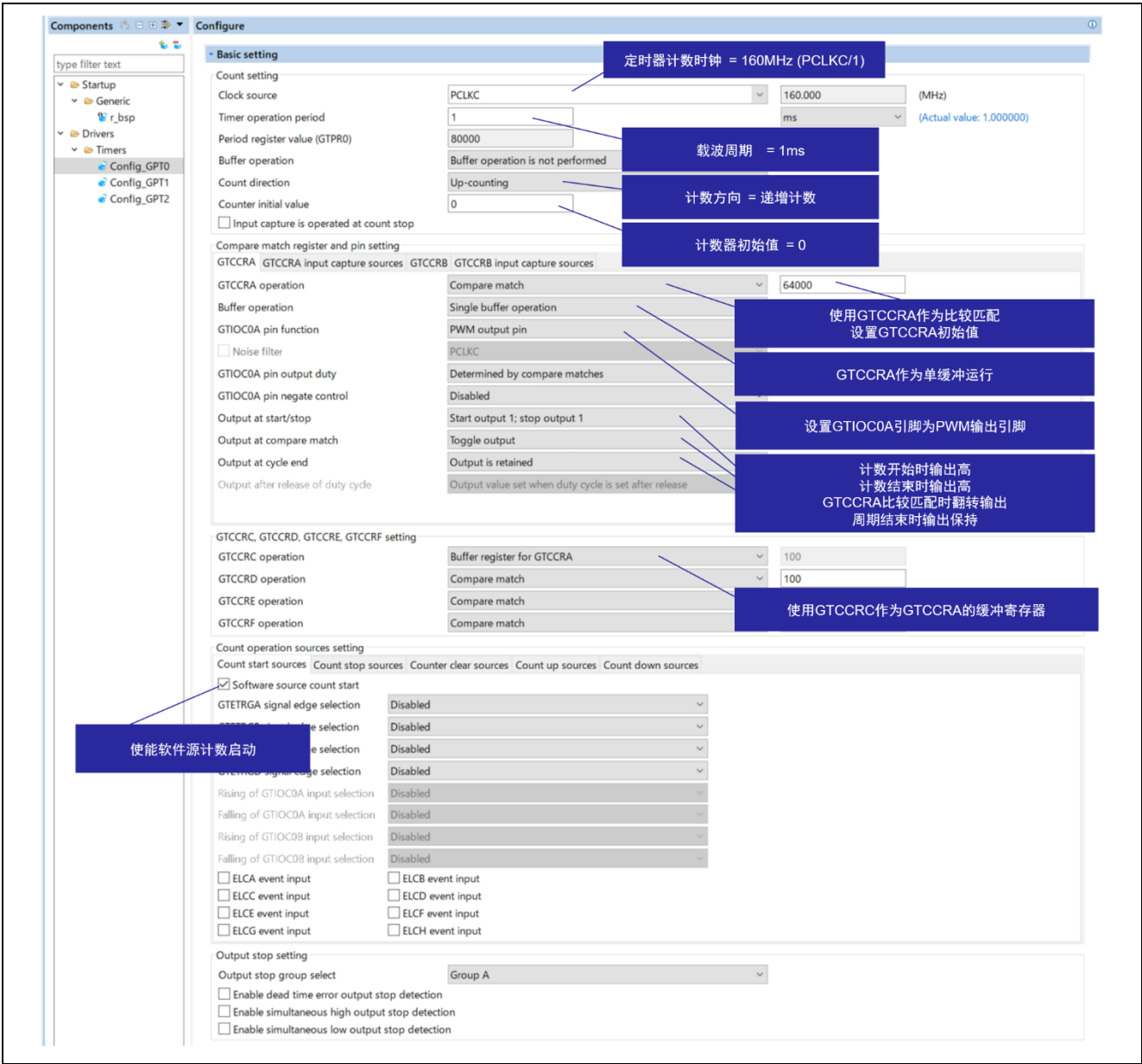


图 4.43 GPT0 设置（1/2）

Advance setting

Automatic dead time setting

- ☒ Automatically set GTCRBO using GTCRA0 value and dead time

GTDVU value: 4800 ☐ Enable buffer (GTDBU)

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0 ☐ Enable buffer (GTDBD)

A/D conversion start request setting

GTADTRA GTADTRB

- ☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)
- ☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

value (GTADTRA): 100

Buffer operation is not performed

Buffer transfer timing setting

No transfer

A/D converter start request signal monitor setting

- ☐ Enable S12AD0 monitor Monitor signal select GTADTRA compare match during up-counting
- ☐ Enable S12AD1 monitor Monitor signal select GTADTRA compare match during up-counting

Interrupt setting

- ☐ Enable GTCRA input capture/compare match interrupt (GTCIA0) Priority Level 15 (highest)
- ☐ Enable GTCRB input capture/compare match interrupt (GTCIB0) Priority Level 15 (highest)
- ☐ Enable GTCRC compare match interrupt (GTCIC0) Priority Level 15 (highest)
- ☐ Enable GTCRD compare match interrupt (GTCID0) Priority Level 15 (highest)
- ☐ Enable GTCRE compare match interrupt (GTCIE0) Priority Level 15 (highest)
- ☐ Enable GTCRF compare match interrupt (GTCIF0) Priority Level 15 (highest)
- ☐ Enable dead time error interrupt (GDE0) Priority Level 15 (highest)
- ☐ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority Level 15 (highest)
- ☒ Enable GTCNT underflow interrupt (GTCIU0) Priority Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

GTCIV0/GTCIU0 interrupt skipping count: Skip count of 1

☐ Link GTCIA0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIB0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIC0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCID0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIE0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIF0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCRA interrupt extended skipping function: No extended interrupt skipping

GTCRB interrupt extended skipping function: No extended interrupt skipping

GTCRC interrupt extended skipping function: No extended interrupt skipping

GTCRD interrupt extended skipping function: No extended interrupt skipping

GTCRE interrupt extended skipping function: No extended interrupt skipping

GTCRF interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

- ☐ Enable output high resolution PWM waveform
- ☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview | Board | Clocks | System | Components | Pins | Interrupts

图 4.44 GPT0 设置 (2/2)

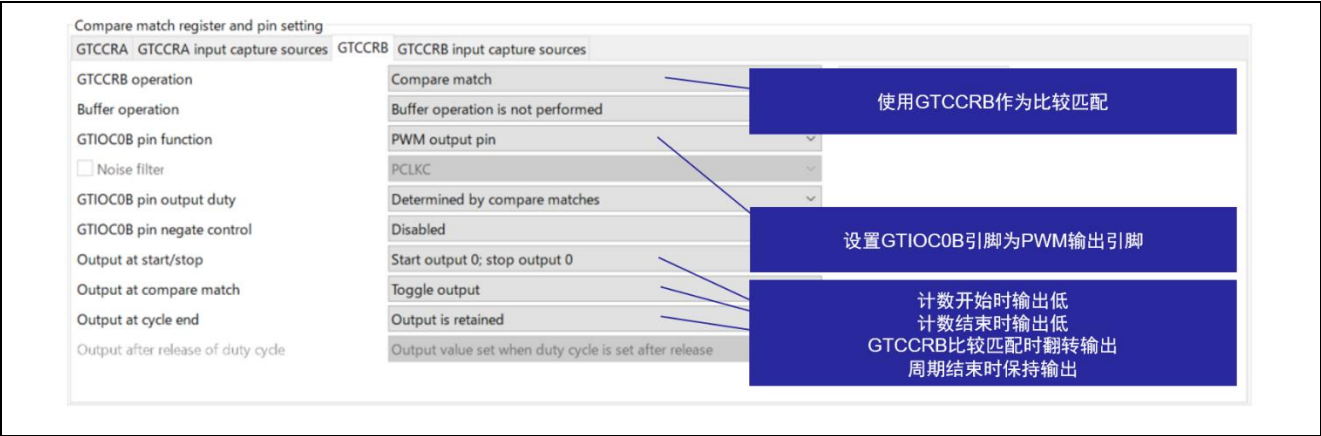


图 4.45 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.6.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。

在主函数中，调用计数开始函数 gpt0_gpt1_gpt2_start 并开始计数。

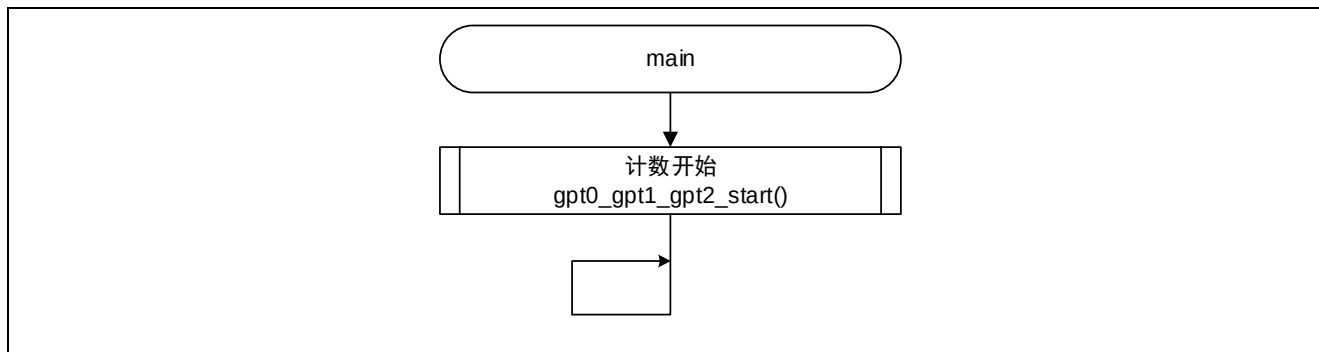


图 4.46 主函数

在计数开始函数中，GPT0，GPT1 和 GPT2 在 GTCIU0 中断使能后开始计数。

此函数是在 Smart Configurator 生成代码后新创建的。

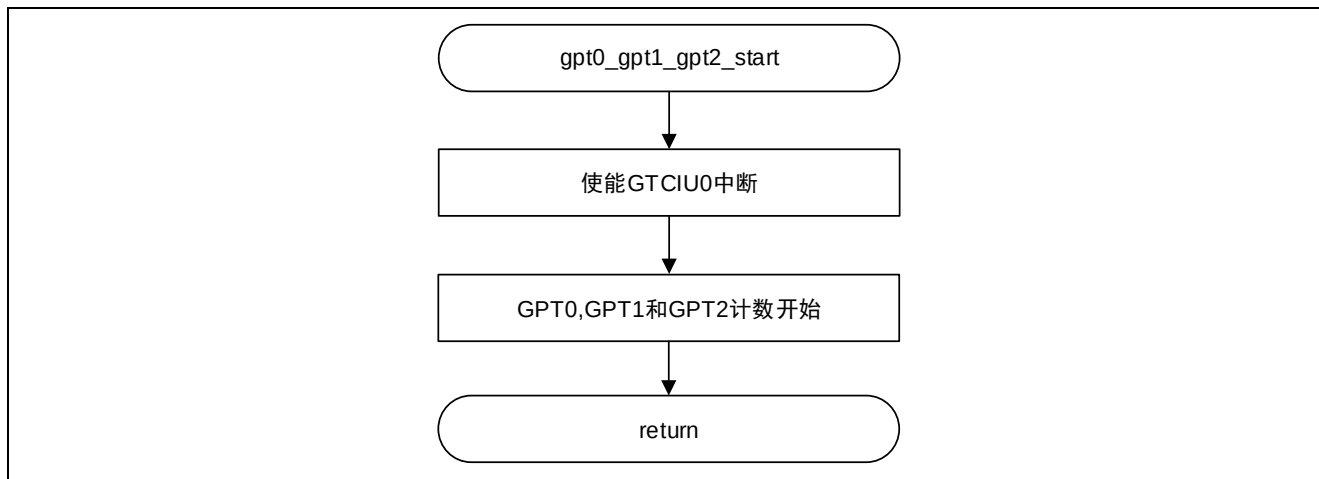


图 4.47 计数开始函数

在主函数之前执行的用户初始化函数 `R_Config_GPT0_Create_UserInit` 用来设置缓冲寄存器的初始值和初始化变量。此函数在 `R_Config_GPT0_Create` 函数中调用。

此示例代码使用以下变量。

- `g_ucduty_prv0`: 用于保存上一次 GPTW0.GTCCRC 寄存器值的变量
- `R_Config_GPT1_Create_UserInit` 和 `R_Config_GPT2_Create_UserInit` 也执行同样的过程。

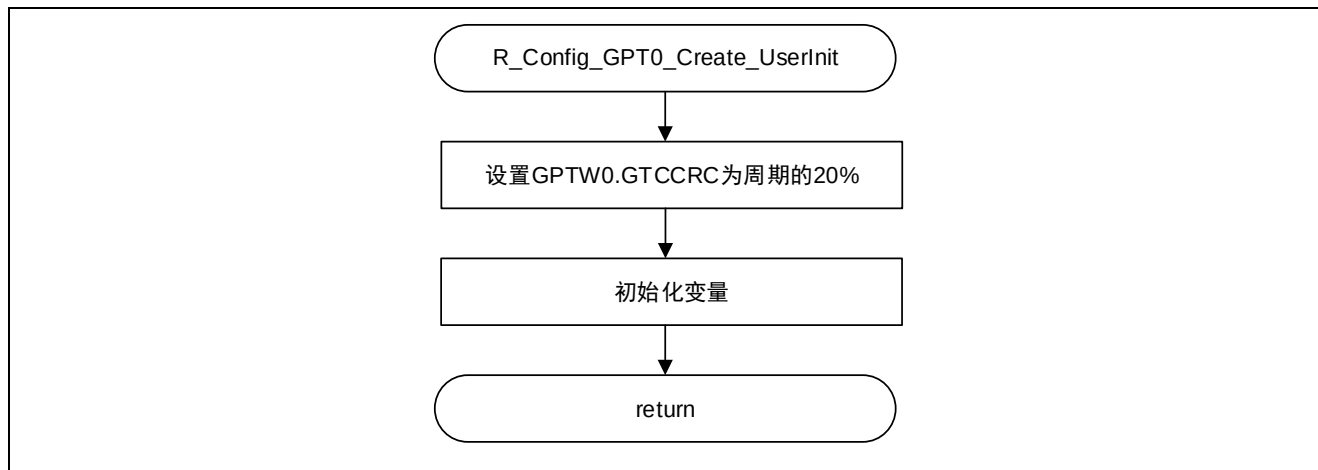


图 4.48 用户初始化函数

GTCIU0 中断处理函数根据缓冲寄存器 GPTW0.GTCCRC, GPTW1.GTCCRC 和 GPTW2.GTCCRC 的当前值以及上一次缓冲寄存器中设置的值来改变缓冲寄存器的值。

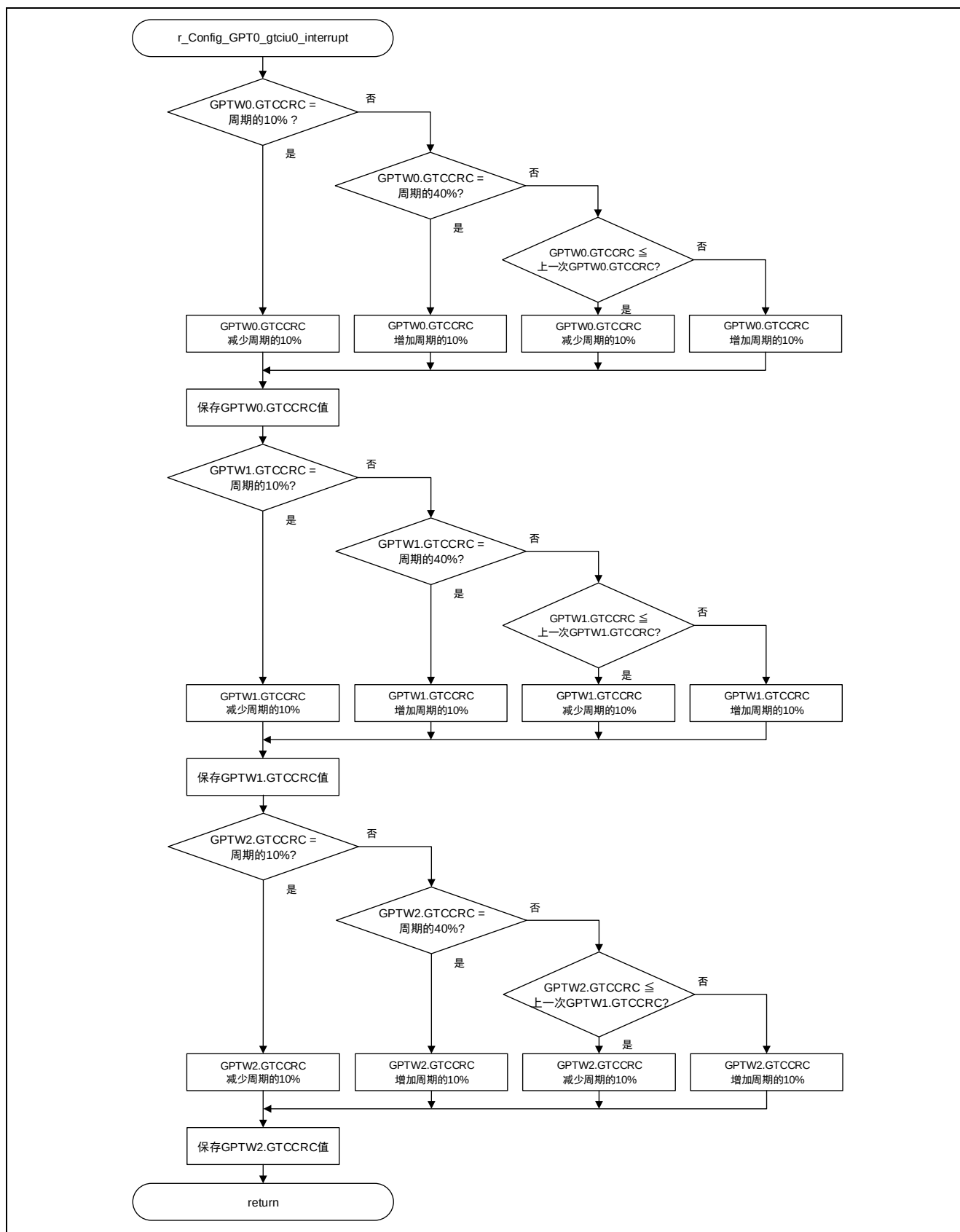


图 4.49 GTCIU0 中断处理函数

4.6.5 相关运行

4.6.5.1 每个间隔的单独死区时间自动设置

本示例代码使用死区时间自动设置功能，通过将 GTDTCR.TDFER 位设为 1，在反相的前侧和后侧生成具有公共转换时序的死区时间。

在死区时间自动设置功能中，可以对波形的前侧和后侧分别设置死区时间。反相的前侧转换时序的死区时间设置在 GTDVU 寄存器中，后侧设置在 GTDVD 寄存器中。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

4.6.5.2 具有死区时间自动设置功能的缓冲运行

本示例代码使用死区时间自动设置功能生成由 Smart Configurator 设置的确保固定死区时间段的反相波形。

死区时间自动设置功能使用 GTDBU 作为 GTDVU 的缓冲寄存器以及 GTDBD 作为 GTDVD 的缓冲寄存器，通过在计数周期结束时（当 GTCNT 计算器下溢发生时（波谷））执行缓冲传送来更新死区时间段。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

Smart Configurator 设置如下所示。

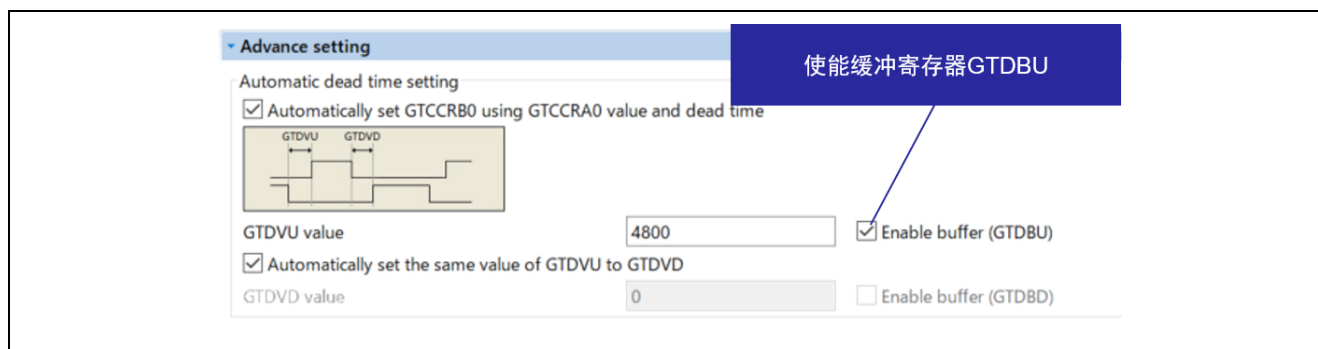


图 4.50 Smart Configurator 死区时间自动设置

图 4.51 显示了使用死区时间值的缓冲寄存器的死区时间自动设置功能的运行示例。

GTDBU 的值在 GTCNT 计数器下溢中断（GTCIU0）（图 4.51 中（1））中被修改。当 GTCNT 计数器下溢发生（波谷）（图 4.51 中（2））时，GTDBU 的值被传送到死区时间设置寄存器 GTDVU，并且输出确保更新后的死区时间段的波形（图 4.51 中（3））。

在第三个周期中，在第一个周期结束时被修改的 GTDBU 的值被传送到 GTDVU（图 4.51 中（4））。

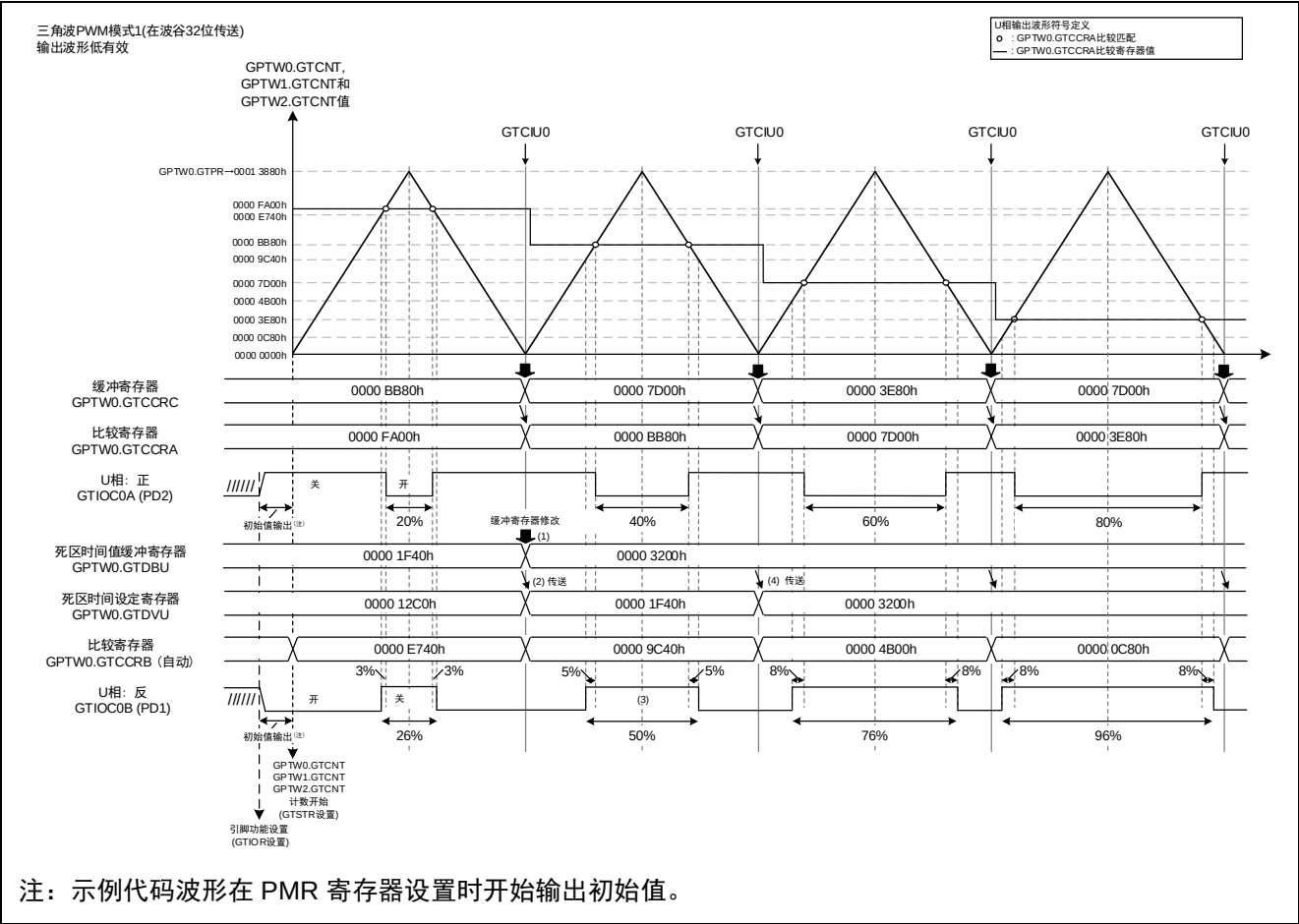


图 4.51 具有死区时间自动设置功能的缓冲运行

4.6.5.3 当不使用死区时间自动设置功能

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm1.zip

图 4.52 显示了不使用死区时间自动设置功能 (GTDTCR.TDE 位为 0) 的运行示例。

在上述示例代码中, 通过设置一个值到反相波形的比较匹配寄存器 GTCCRB 和缓冲寄存器 GTCCRE, 生成确保死区时间段的波形。

与正相相似, 在反相中通过修改缓冲寄存器 GTCCRE (图 4.52 中 (1)), 并在 GTCNT 计数器下溢发生 (波谷) 时将缓冲寄存器传送到比较寄存器 GTCCRB 中时 (图 4.52 中 (2)), 在每个周期中更新比较值。

此外, 如图 4.16 中运行所示, 确保相同的死区时间段。

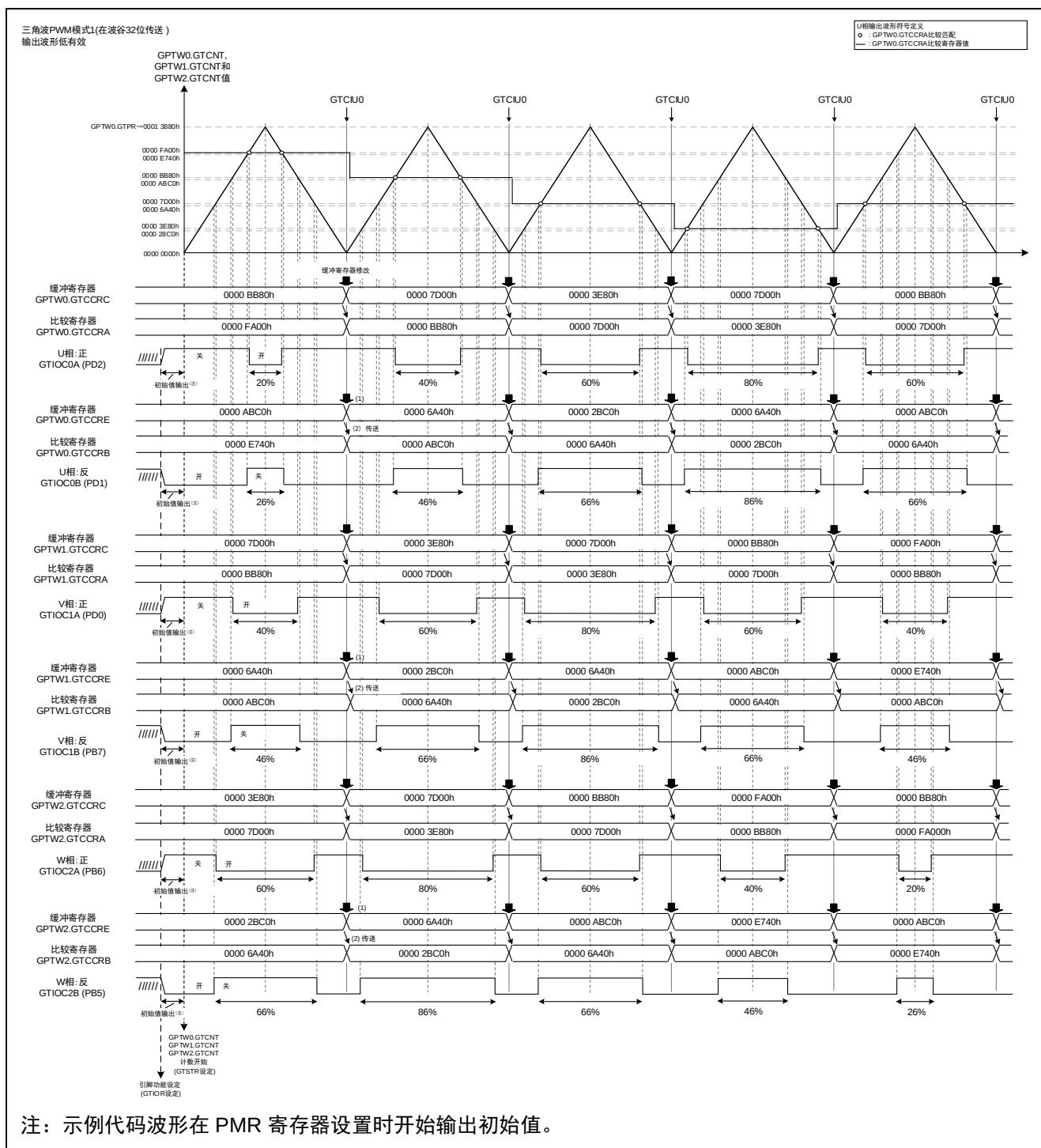


图 4.52 当不使用死区时间自动设置功能(三角波 PWM 模式 1)

4.6.6 使用时的注意事项

4.6.6.1 多通道计数开始

在此示例代码中，定时器软件启动寄存器 GTSTR 的 CSTRTO, CSTRTO1 和 CSTRTO2 位在 gpt0_gpt1_gpt2_start 函数中同时设置，以同时启动 GPTW0, GPTW1 和 GPTW2 计数。

当使用 Smart Configurator 生成的 R_Config_GPT0_Start, R_Config_GPT1_Start 和 R_Config_GPT2_Start 函数时，因为每个函数调用时间不同，计数开始时间可能不同。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.8.1 软件同步操作。

4.6.6.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较匹配寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

比较寄存器 GTCCRA 的值应设置为满足以下限制。

$GTCCRA > GTDVU$

$GTCCRA > GTDVD$

$GTCCRA < GTPR$

如果在计数运行期间，GTCCRA 设置为 0000 0000h 或大于 GTPR 的设置值，则输出保护功能被激活。

但是，如果不满足以下条件，则功能不能正常运行。

当计数运行开始时的 GTCCRA 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设置值时

有关输出保护功能的详细信息，请参考 RX66T Group User's Manual: Hardware, 章节 24.8.4 GTIOCNm 引脚输出保护功能 (n = 0 到 9; m = A, B)。

当不使用死区时间自动设置功能时，GTCCRA (GTCCRB) 寄存器的值设置为大于 0000 0001h 且小于 GTPR 寄存器的设置值。当 GTCCRA 寄存器中设置了 0000 0000h 或与 GTPR 寄存器相同的值时，仅当满足 $[GTCCRA (GTCCRB) = 0000 0000h]$ 或 $[GTCCRA (GTCCRB) = GTPR]$ 时，才在一个周期中发生比较匹配。另外，如果 GTCCRA 被设置为超过 GTPR 寄存器设置值的值时，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)，(1) 在三角波 PWM 模式下进行死区时间自动设置时和 (2) 在三角波 PWM 模式下未进行死区时间自动设置时。

4.7 三角波 PWM 模式 2

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm2_dt.zip

4.7.1 概要

GPTW 三角波 PWM 模式 2 可用于输出带死区时间的三相互补 PWM 波形。

此示例代码描述了在三角波 PWM 模式 2（在波峰和波谷 32 位传送）下使用死区时间自动设置功能并重复输出以下波形。每个占空比使用缓冲器（不是双缓冲器）生成左右不对称的 PWM 波形。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

当 GTCNT 计数器溢出和下溢发生时，通过从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA 来改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用三角波 PWM 模式 2
- 使用通道 0, 1 和 2（通道号: n = 0, 1 和 2）
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz（PCLKC/1）
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCN_A 引脚为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出高，计数结束时输出高
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCN_B 引脚为 PWM 输出引脚
 - 使用 GTCCRB 作为比较匹配
 - 计数开始时输出低，计数结束时输出低
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断和下溢中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 4.55

在 Smart Configurator 中设置。
设置方法请参考章节 4.7.3。

此示例代码的三角波 PWM 模式 2 输出如下所示。

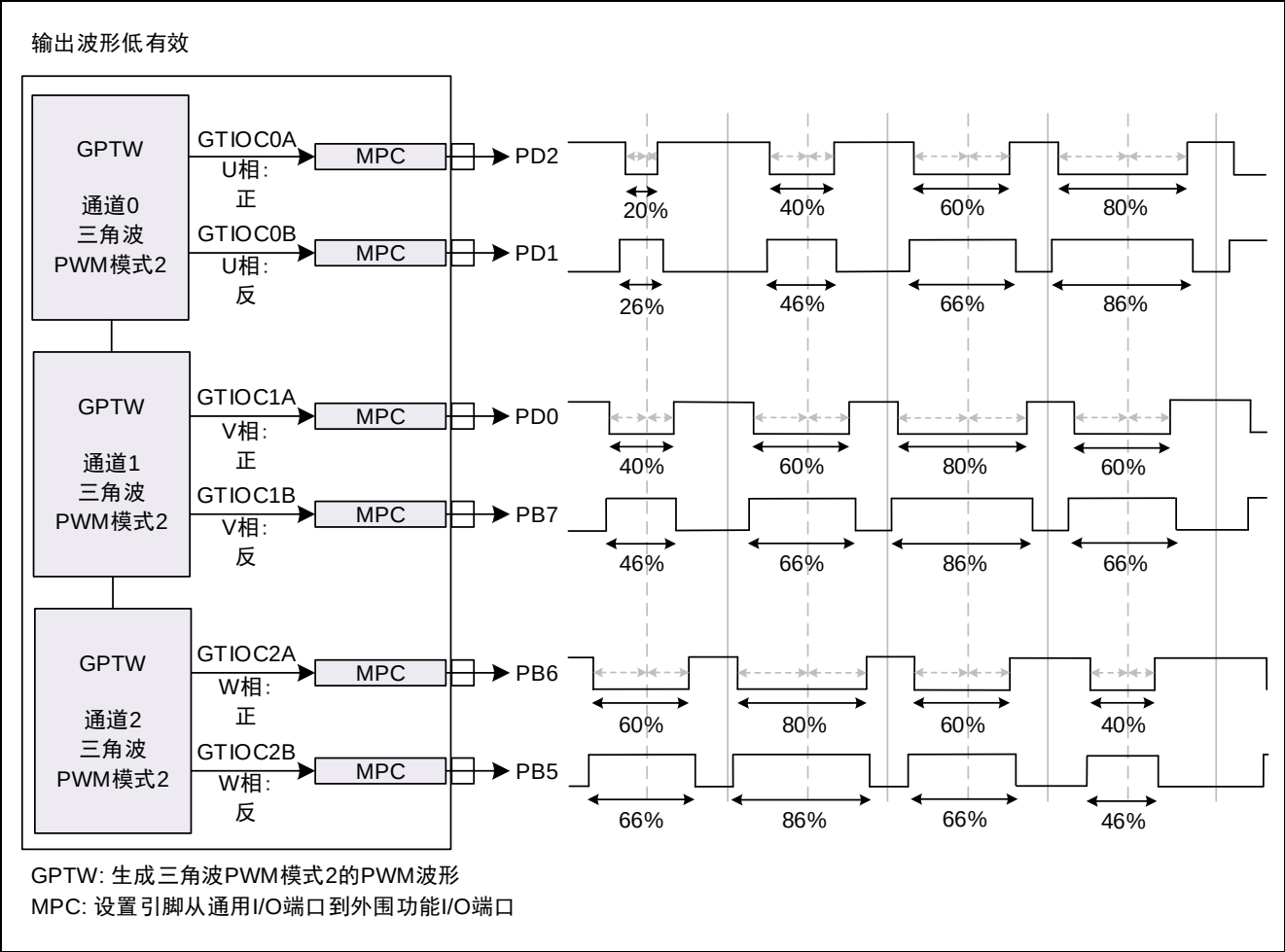


图 4.53 三角波 PWM 模式 2 输出

4.7.2 运行说明

示例代码的运行如图 4.55 所示。当 GTCNT 计数器溢出中断（GTCIV0）和 GTCNT 计数器下溢中断（GTCIU0）产生时，通过修改缓冲寄存器 GTCCRC 的值来改变每个周期的占空比的设置。（图 4.55 中（1））。

当 GTCNT 计数器溢出（波峰）和 GTCNT 计数器下溢（波谷）发生时，通过缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA，示例代码使用三角波 PWM 模式 2 来更新数据（图 4.55 中（2））。

此外，因为使用死区时间自动设置功能，GTCCRB 寄存器根据 GTCCRA 的更新自动设置。GTDVU 和 GTDVD 设置为相同值。此示例代码在计数开始时自动设置 GTCCRB 的值（图 4.55 中（3））。

在计数开始后，比较寄存器和计数器寄存器之间发生比较匹配，反相输出 OFF，然后正相输出 ON（图 4.55 中（4））。

- 左右不对称 PWM 波形输出
每个周期在递增计数期间和递减计数期间产生不同的占空比。

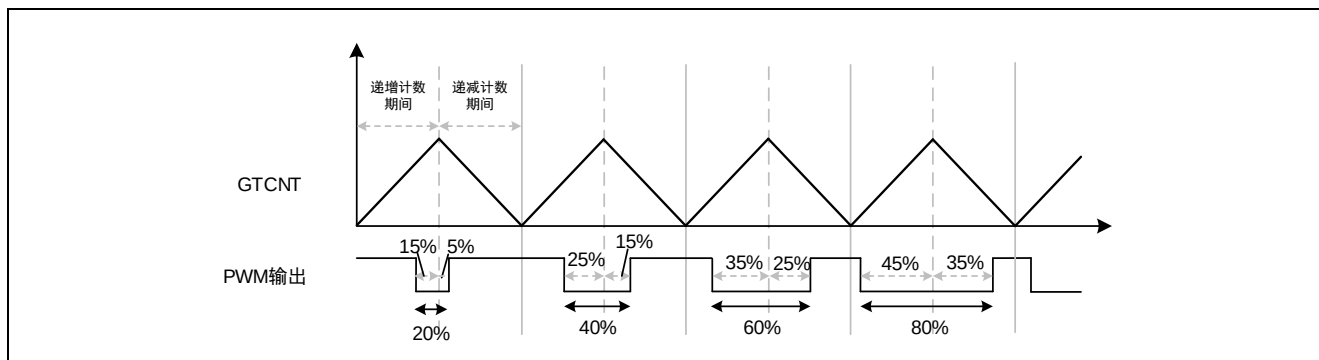


图 4.54 左右不对称 PWM 输出波形



图 4.55 示例代码运行

4.7.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.10 添加组件

项目	描述		
组件	通用 PWM 定时器		
配置名称	Config_GPT0	Config_GPT1	Config_GPT2
工作模式	三角波 PWM 模式 2		
资源	GPT0	GPT1	GPT2

图 4.56 至图 4.58 显示 Config_GPT0 设置。相同的设置适用于 GPT1 和 GPT2。

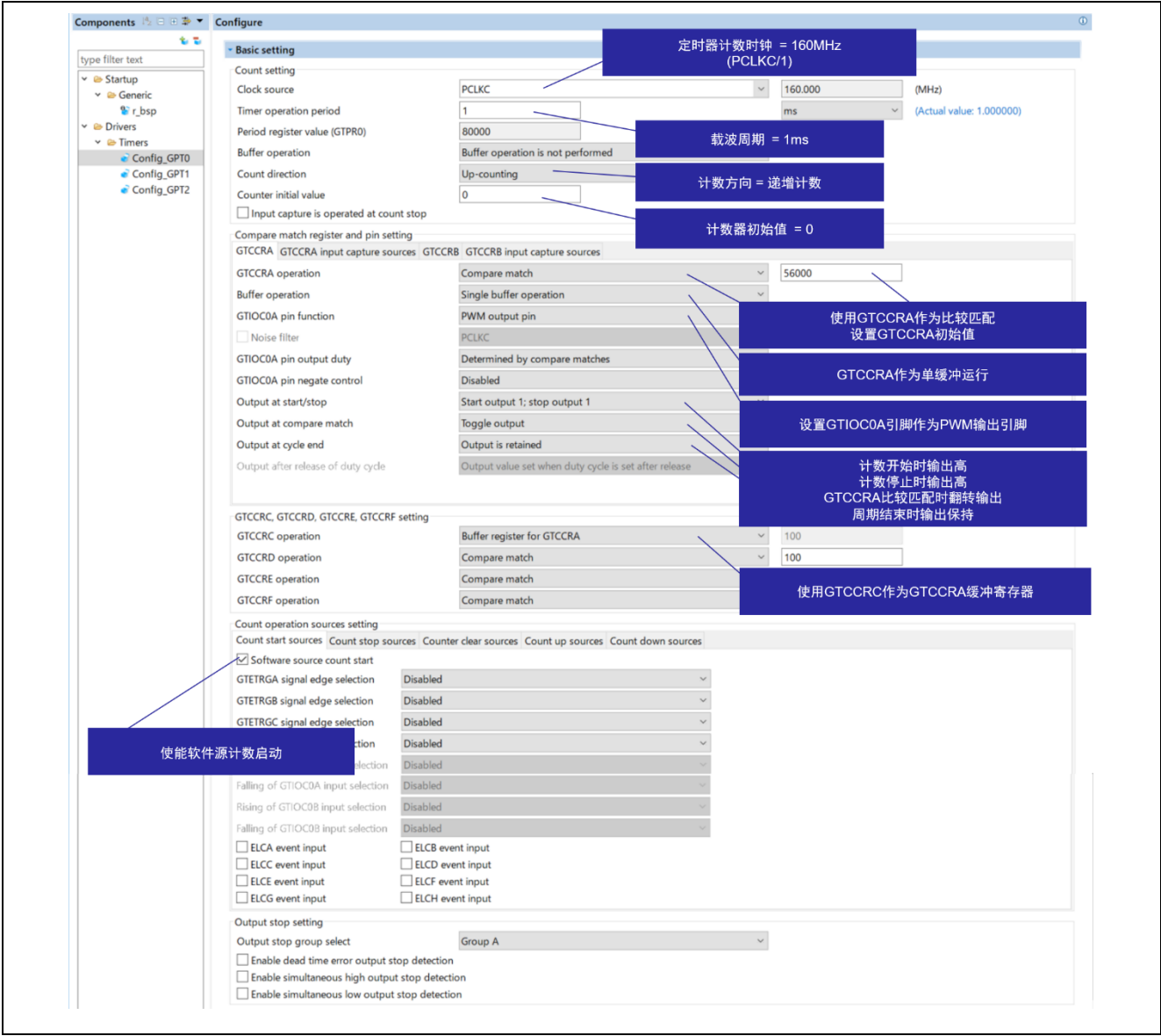


图 4.56 GPT0 设置 (1/2)

Advance setting

Automatic dead time setting

☒ Automatically set GTCR80 using GTCRA0 value and dead time

GTDVU value: 4800

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0

☐ Enable buffer (GTD8U)

☐ Enable buffer (GTD8D)

Start request setting

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA): 100

Buffer operation: Buffer operation is not performed

Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCRA input capture/compare match interrupt (GTCIA0) Priority: Level 15 (highest)

☐ Enable GTCRB input capture/compare match interrupt (GTCIB0) Priority: Level 15 (highest)

☐ Enable GTCRC compare match interrupt (GTCIC0) Priority: Level 15 (highest)

☐ Enable GTCRD compare match interrupt (GTCID0) Priority: Level 15 (highest)

☐ Enable GTCRE compare match interrupt (GTCIE0) Priority: Level 15 (highest)

☐ Enable GTCRF compare match interrupt (GTCIF0) Priority: Level 15 (highest)

☐ Enable dead time error interrupt (GDTE0) Priority: Level 15 (highest)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority: Level 15 (highest)

☒ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

GTCIV0/GTCIU0 interrupt skipping function: Skip count of 1

☐ Link GTCID0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIE0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTCIF0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCRA interrupt extended skipping function: No extended interrupt skipping

GTCRB interrupt extended skipping function: No extended interrupt skipping

GTCRC interrupt extended skipping function: No extended interrupt skipping

GTCRD interrupt extended skipping function: No extended interrupt skipping

GTCRE interrupt extended skipping function: No extended interrupt skipping

GTCRF interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.57 GPT0 设置 (2/2)

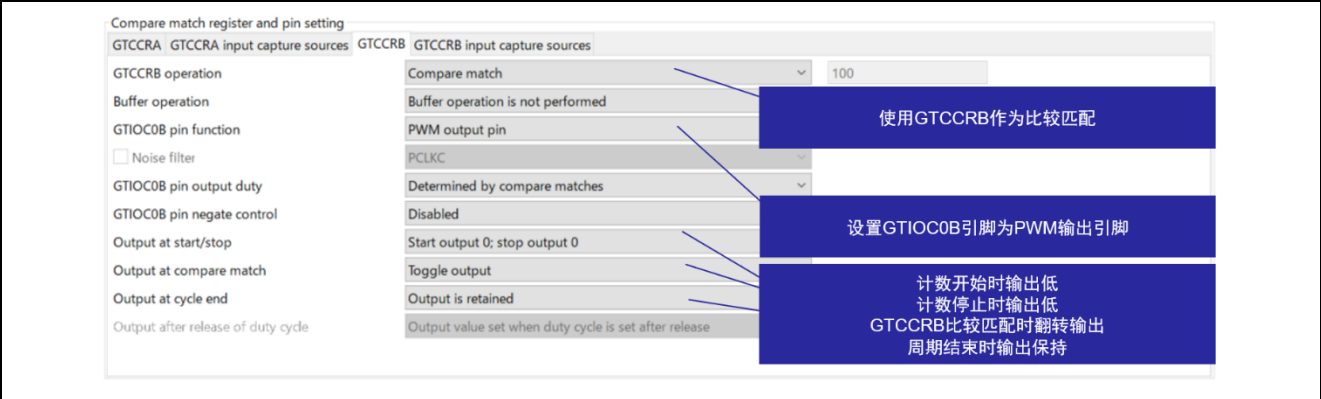


图 4.58 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.7.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 gpt0_gpt1_gpt2_start 并开始计数。

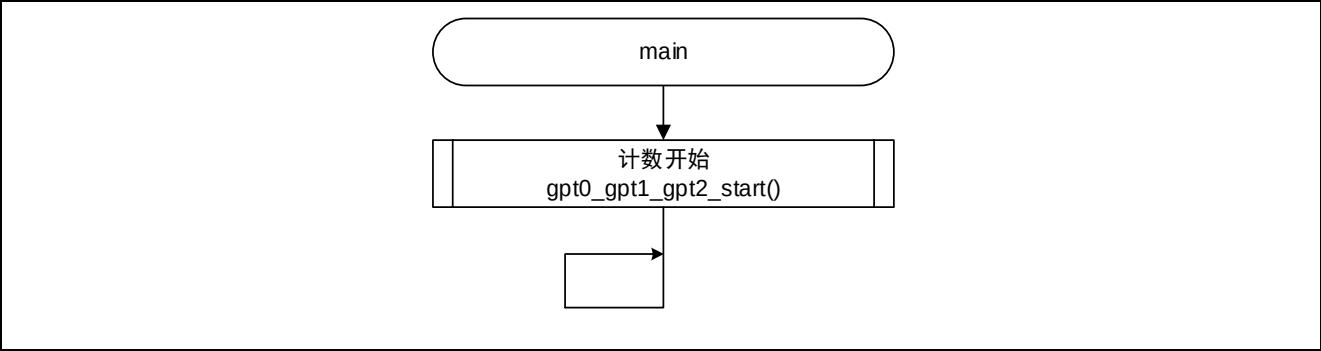


图 4.59 主函数

在计数开始函数中，GPT0，GPT1 和 GPT2 在 GTCIV0 和 GTCIU0 中断使能后开始计数。
此函数是在 Smart Configurator 生成代码后新创建的。

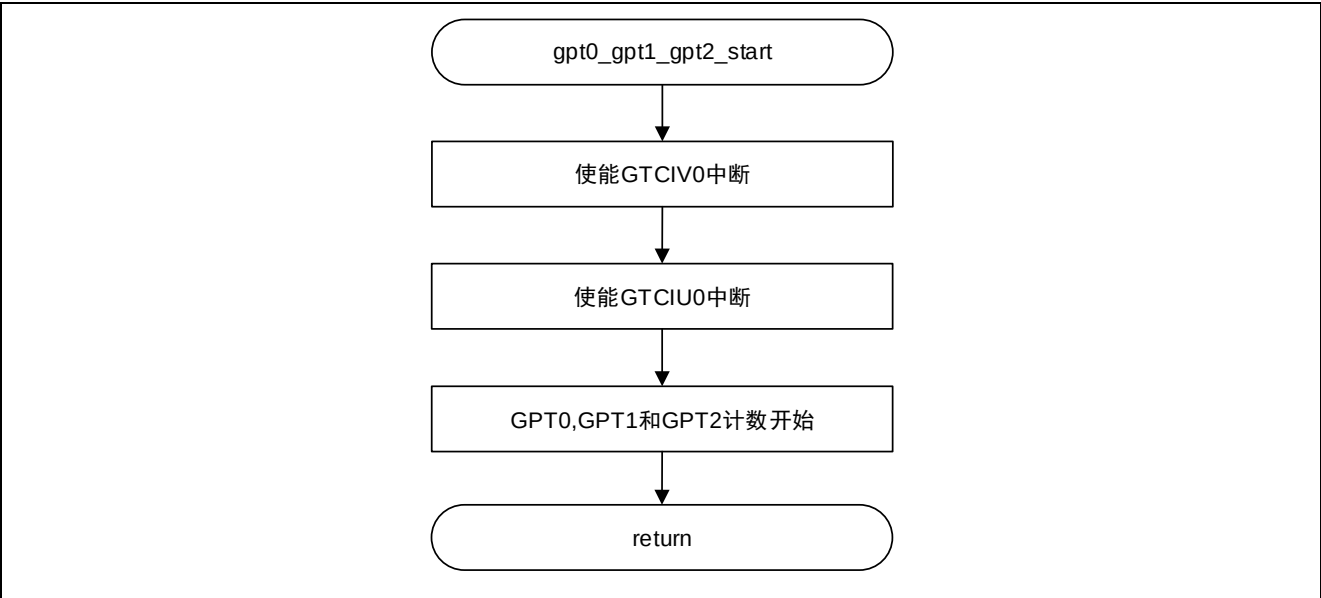


图 4.60 计数开始函数

在主函数之前执行的用户初始化函数 `R_Config_GPT0_Create_UserInit` 用来设置缓冲寄存器的初始值。此函数在 `R_Config_GPT0_Create` 函数中调用。

`R_Config_GPT1_Create_UserInit` 和 `R_Config_GPT2_Create_UserInit` 也执行同样的过程。

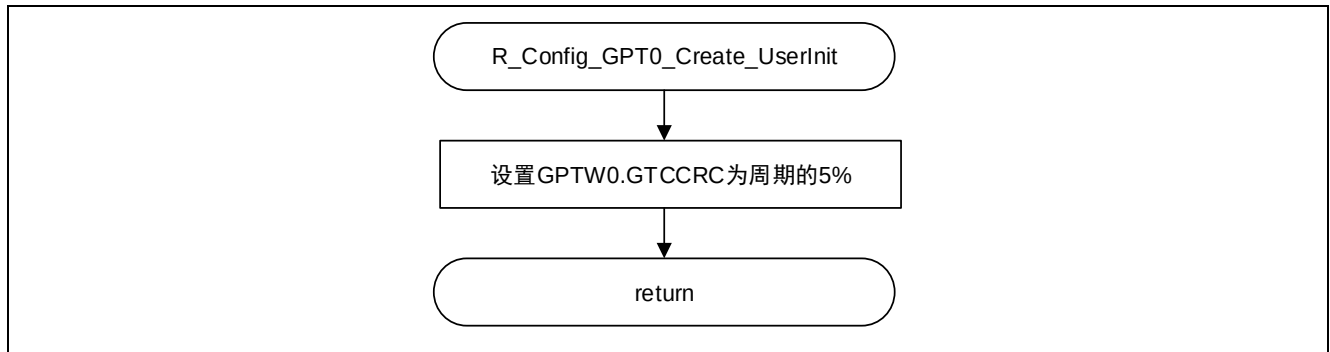


图 4.61 用户初始化函数

GTCIV0 中断处理函数根据输出占空比的变化来更改缓冲寄存器的值。

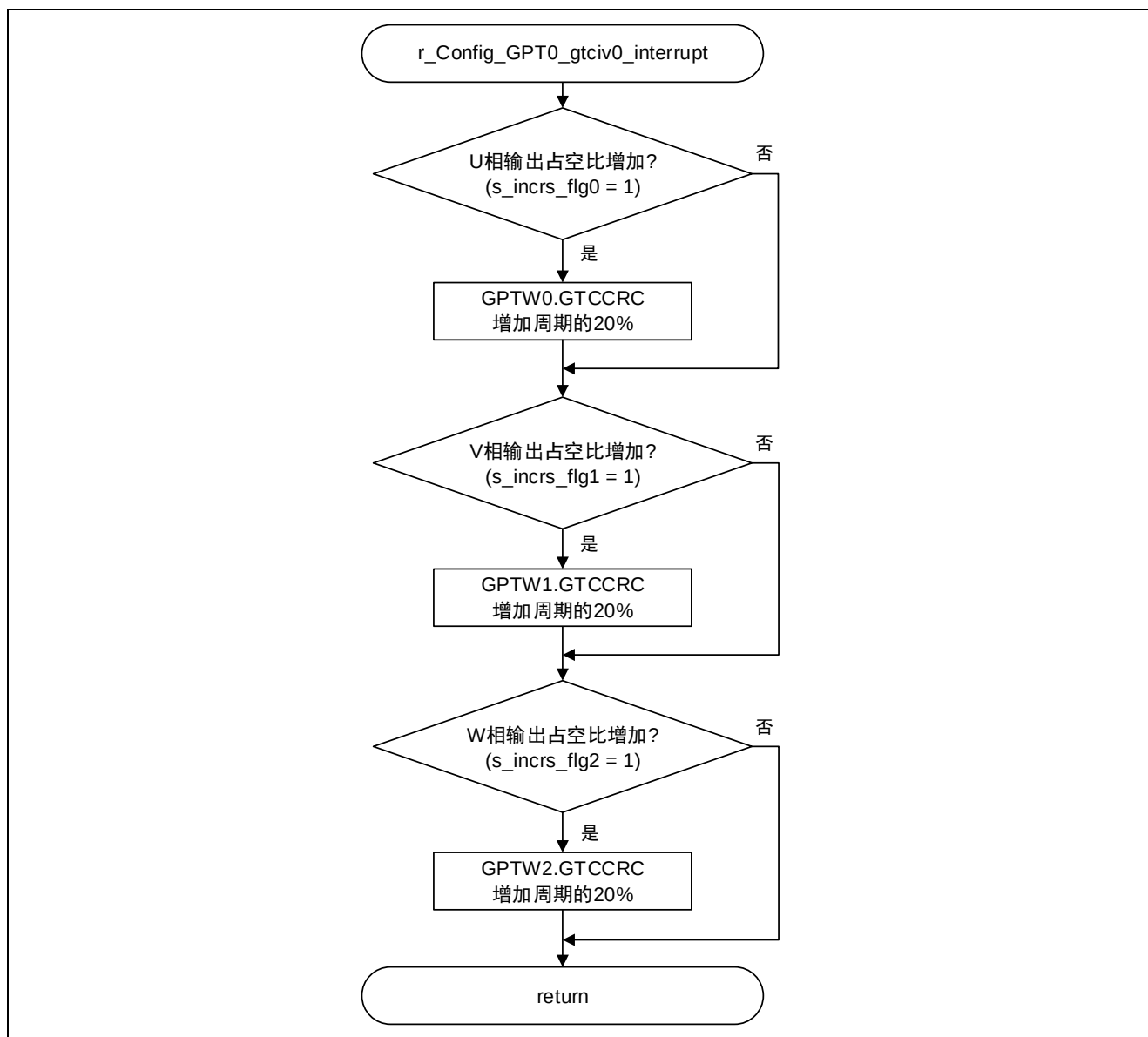


图 4.62 GTCIV0 中断处理函数

GTCIU0 中断处理函数根据缓冲寄存器的当前值来更改缓冲寄存器的值。

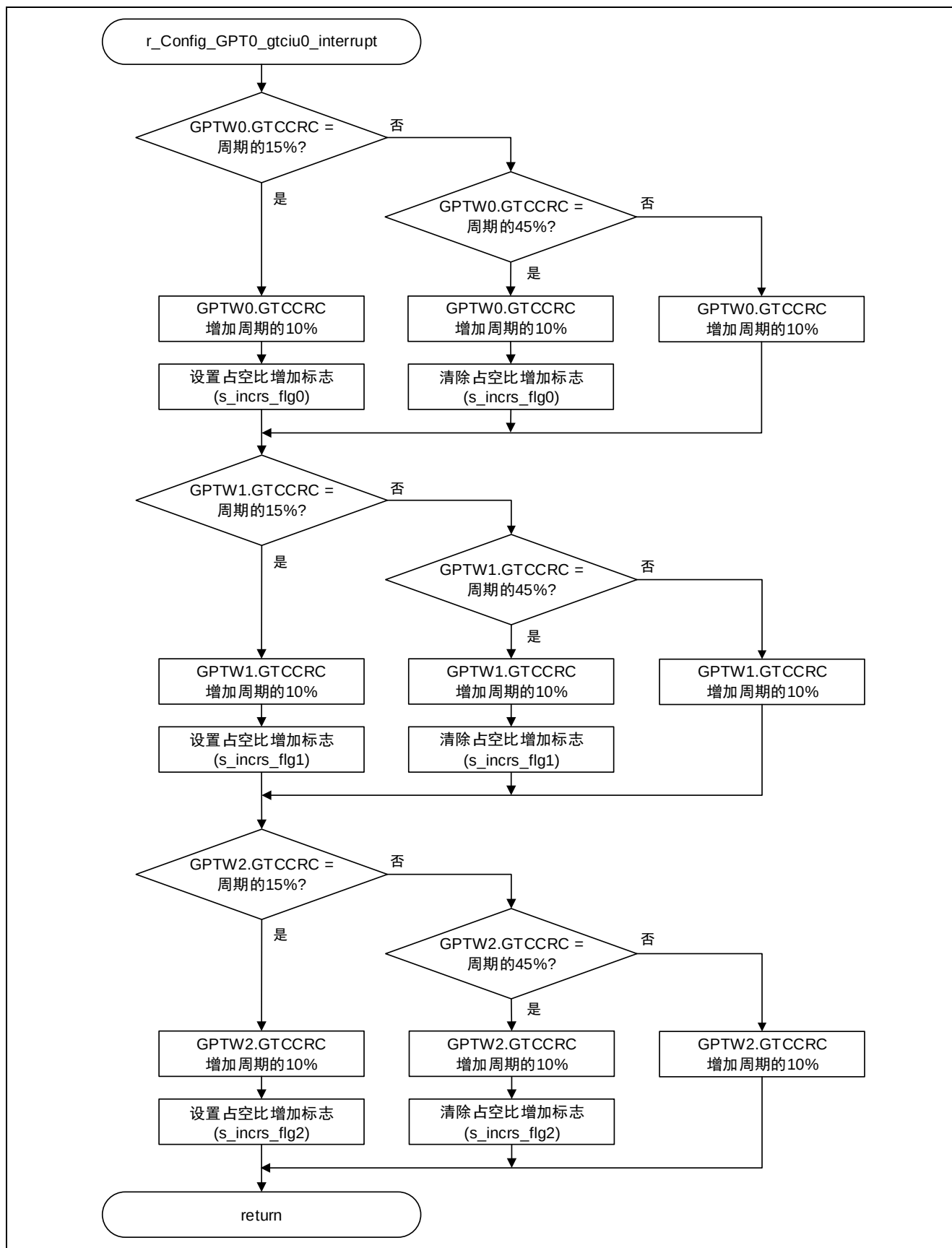


图 4.63 GTCIU0 中断处理函数

4.7.5 相关运行

4.7.5.1 每个间隔的单独死区时间自动设置

本示例代码使用死区时间自动设置功能，通过将 GTDTCR.TDFER 位设为 1，在反相的前侧和后侧生成具有公共转换时序的死区时间。

在死区时间自动设置功能中，可以在波形的前侧和后侧分别设置死区时间。反相的前侧转换时序的死区时间设置在 GTDVU 寄存器中，后侧设置在 GTDVD 寄存器中。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

4.7.5.2 具有死区时间自动设置功能的缓冲运行

本示例代码使用死区时间自动设置功能生成由 Smart Configurator 设置的确保固定死区时间段的反相波形。

死区时间自动设置功能使用 GTDBU 作为 GTDVU 的缓冲寄存器以及 GTDBD 作为 GTDVD 的缓冲寄存器，通过在计数周期结束时（当 GTCNT 计算器下溢发生时（波谷））执行缓冲传送来更新死区时间段。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

Smart Configurator 设置如下所示。

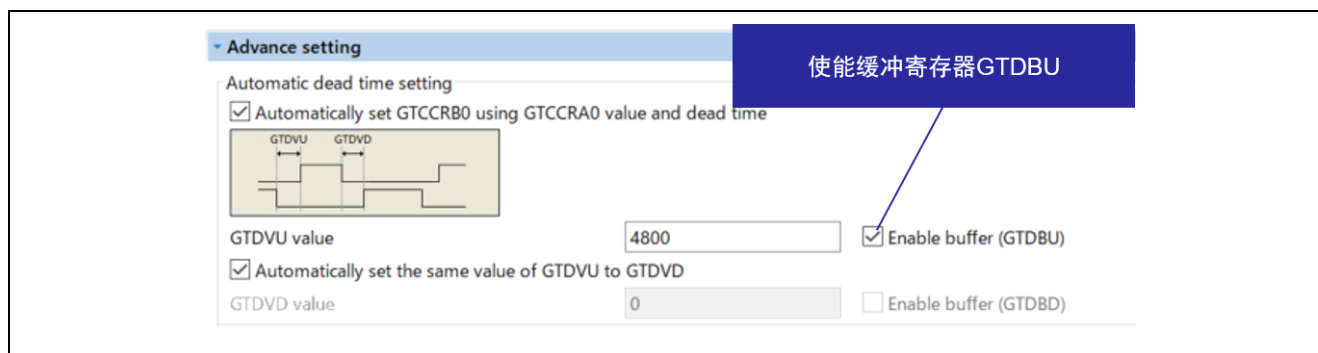


图 4.64 Smart Configurator 死区时间自动设置

图 4.65 显示了使用死区时间值的缓冲寄存器的死区时间自动设置功能的运行示例。

GTDBU 的值在 GTCNT 计数器下溢中断（GTCIU0）（图 4.65 中（1））中被修改。当 GTCNT 计数器下溢发生（波谷）（图 4.65 中（2））时，GTDBU 的值被传送到死区时间设置寄存器 GTDVU，并且输出确保更新后的死区时间段的波形（图 4.65 中（3））。

在第三个周期中，在第一个周期结束时被修改的 GTDBU 的值被传送到 GTDVU（图 4.65 中（4））。

第四个周期的死区时间设置导致在反相前侧的转换时序产生死区时间误差（GTCCRA-GTDVU<0），因此输出具有校准转换时序的正相和反相波形（图 4.65 中（5））。

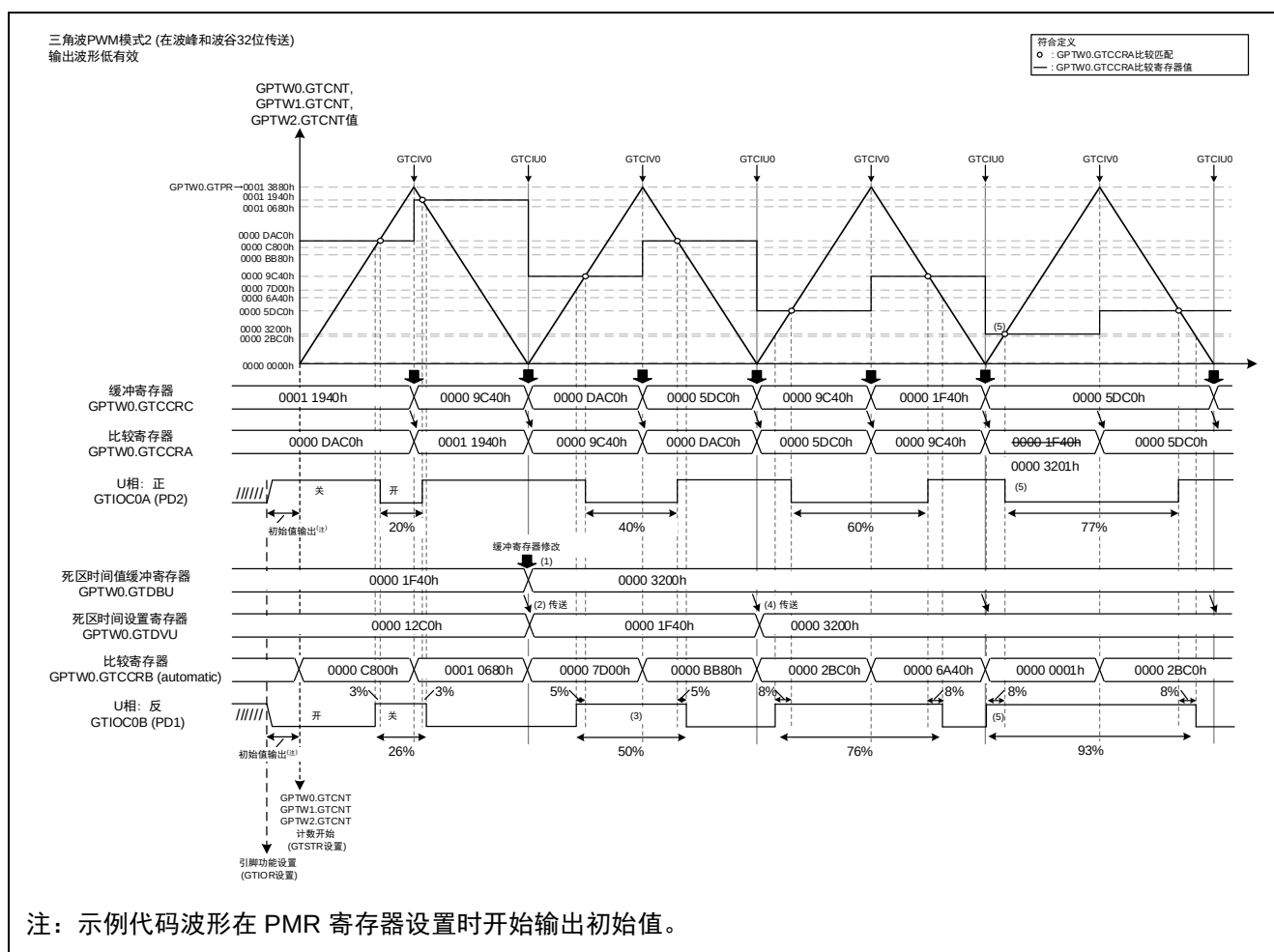


图 4.65 具有死区时间自动设置功能的缓冲运行

4.7.5.3 当不使用死区时间自动设置功能

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm2.zip

图 4.66 显示了不使用死区时间自动设置功能 (GTDTCCR.TDE 位为 0) 的运行示例。

当不使用死区时间自动设置功能时, 通过设置到比较匹配寄存器 GTCCRB (反相波形寄存器) 和缓冲寄存器 GTCCRE 的值, 生成确保死区时间段的波形。

与正相相似, 在反相中通过在产生 GTCNT 计数器下溢中断 (GTCIV0) 和 GTCNT 计数器溢出中断 (GTCIU0) 时修改缓冲寄存器 GTCCRE (图 4.66 中 (1)), 并在 GTCNT 计数器溢出发生 (波峰) 时将缓冲寄存器 GTCCRE 传送到比较寄存器 GTCCRB 中 (图 4.66 中 (2)), 在每个周期中更新比较值。

此外, 如图 4.55 中运行所示, 确保相同的死区时间段。

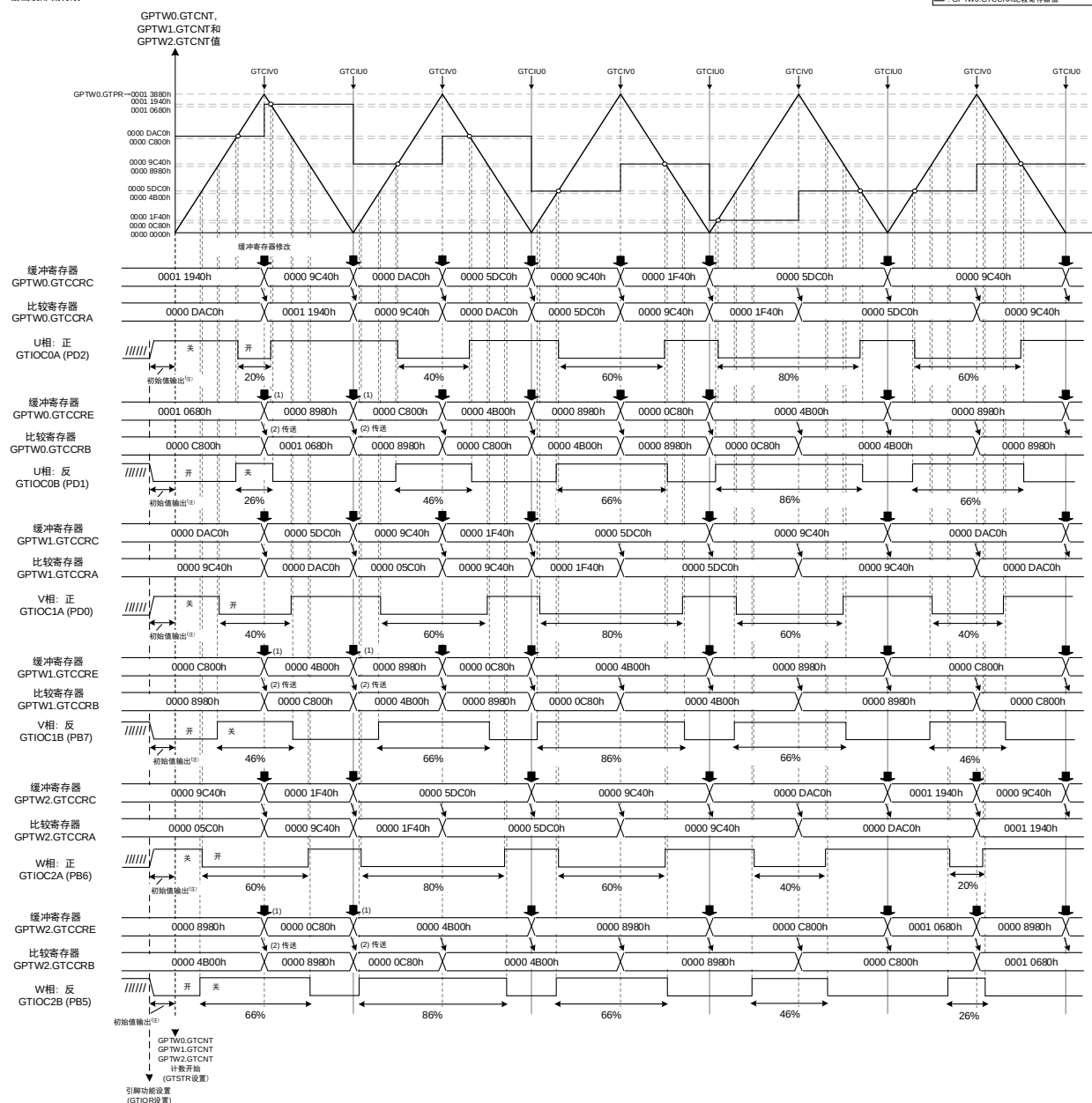


图 4.66 当不使用死区时间自动设置功能（三角波 PWM 模式 2）

4.7.6 使用时的注意事项

4.7.6.1 多通道计数开始

在此示例代码中，定时器软件启动寄存器 GTSTR 的 CSTRTO, CSTRTO1 和 CSTRTO2 位在 gpt0_gpt1_gpt2_start 函数中同时设置，以同时启动 GPTW0, GPTW1 和 GPTW2 计数。

当使用 Smart Configurator 生成的 R_Config_GPT0_Start, R_Config_GPT1_Start 和 R_Config_GPT2_Start 函数时，因为每个函数调用时间不同，计数开始时间可能不同。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.8.1 软件同步操作。

4.7.6.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较匹配寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

比较寄存器 GTCCRA 的值应该设置为满足以下限制

$$GTCCRA > GTDVU$$

$$GTCCRA > GTDVD$$

$$GTCCRA < GTPR$$

如果在计数运行期间，GTCCRA 设置为 0000 0000h 或大于 GTPR 的设置值，则输出保护功能被激活。

但是，如果不满足以下条件，则功能不能正常运行。

当计数运行开始时的 GTCCRA 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设置值时

有关输出保护功能的详细信息，请参考 RX66T Group User's Manual: Hardware, 章节 24.8.4 GTIOcnm 引脚输出保护功能 (n = 0 到 9; m = A, B)。

当不使用死区时间自动设置功能时，GTCCRA (GTCCRB) 寄存器的值设置为大于 0000 0001h 且小于 GTPR 寄存器的设置值。当 GTCCRA 寄存器中设置了 0000 0000h 或与 GTPR 寄存器相同的值时，仅当满足 $[GTCCRA (GTCCRB) = 0000 0000h]$ 或 $[GTCCRA (GTCCRB) = GTPR]$ 时，才在一个周期中发生比较匹配。另外，如果 GTCCRA 被设置为超过 GTPR 寄存器设置值的值时，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 至 F)，(1) 在三角波 PWM 模式下进行死区时间自动设置时和 (2) 再三角波 PWM 模式下未进行死区时间自动设置时。

4.8 三角波 PWM 模式 3

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm3_dt.zip

4.8.1 概要

GPTW 三角波 PWM 模式 3 可用于输出带死区时间的三相互补 PWM 波形。

此示例代码描述了在三角波 PWM 模式 3（在波谷 64 位传送）中使用死区时间自动设置功能并重复以下波形输出。每个占空比使用双缓冲生成左右不对称 PWM 波形。

- U 相占空比切换: 20% → 40% → 60% → 80% → 60% → 40% → ...
- V 相占空比切换: 40% → 60% → 80% → 60% → 40% → 20% → ...
- W 相占空比切换: 60% → 80% → 60% → 40% → 20% → 40% → ...

当 GTCNT 计数器溢出发生时, 将暂存器 A 的值传送到比较寄存器 GTCCRA, 并在下溢发生时将缓冲寄存器 GTCCRD 的值传送到暂存器 A 以及将缓冲区寄存器 GTCRCC 的值传送到比较寄存器 GTCCRA, 从而改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用三角波 PWM 模式 3
- 使用通道 0, 1 和 2 (通道号: n = 0, 1 和 2)
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCnA 引脚为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出高, 计数停止时输出高
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCnB 引脚为 PWM 输出引脚
 - 使用 GTCCRB 作为比较匹配
 - 计数开始时输出低, 计数停止时输出低
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用缓冲寄存器
 - GTCCRA 和 GTCCRB 作为双缓冲运行
 - 使用 GTCCRC 和 GTCCRD 作为 GTCCRA 的缓冲寄存器
 - 使用 GTCCRE 和 GTCCRF 作为 GTCCRB 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器下溢中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.69

在 Smart Configurator 中设置。
设置方法请参考章节 4.8.3。

此示例代码的三角波 PWM 模式 3 输出如下所示。

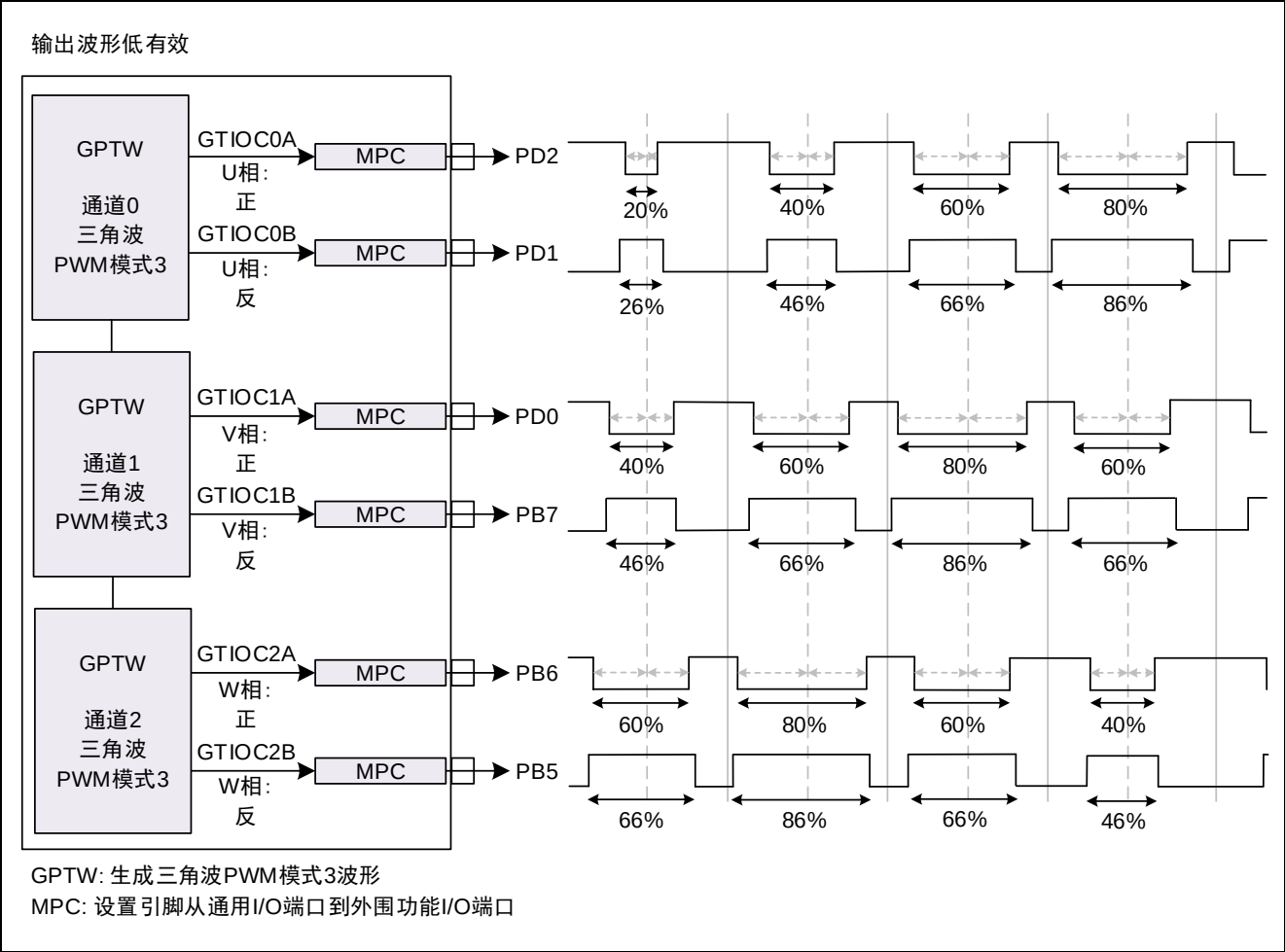


图 4.67 三角波 PWM 模式 3 输出

4.8.2 运行说明

示例代码的运行如图 4.69 所示。通过在 GTCNT 计数器下溢中断（GTCIU0）中修改缓冲寄存器 GTCCRC 和 GTCCRD 的值，在每个周期改变占空比的设置（图 4.69 中（1））。

使用 Smart Configurator 生成的代码中缓冲寄存器的初始值设置为与比较寄存器相同的值。因此，缓冲寄存器的值在计数开始之前为用户初始化函数 `R_Config_GPT0_Create_UserInit` 中设置。寄存器中设置的值通过强制缓冲传送分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.69 中（2））。

此示例代码使用三角波 PWM 模式 3，在 GTCNT 计数器溢出（波峰）发生时传送暂存器 A 到比较的寄存器 GTCCRA（图 4.69 中（3））并在 GTCNT 计数器下溢（波谷）发生时将缓冲寄存器 GTCCRD 传送到暂存器 A 以及缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.69 中（4））来更新数据。

此外，由于使用了死区时间自动功能，GTCCRB 寄存器的值会根据 GTCCRA 更新自动设置。设置 GTDVU 和 GTDVD 为相同的值。此示例代码在计数开始时自动设置 GTCCRB 的值（图 4.55 中（5））。

计数开始后，比较寄存器和计数器寄存器之间发生比较匹配，反相输出 OFF，然后正相输出 ON（图 4.69 中（6））。

- 左右不对称 PWM 波形输出
每个周期在递增计数和递减计数期间产生不同的占空比。

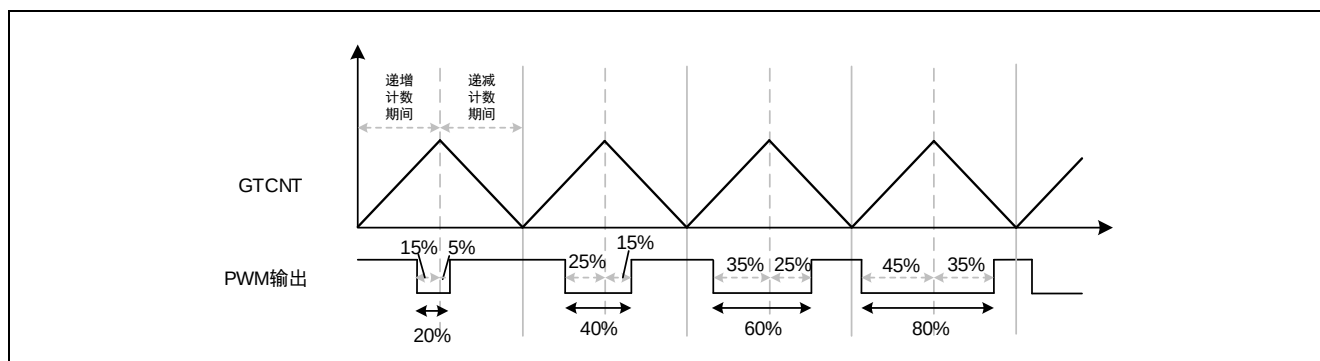
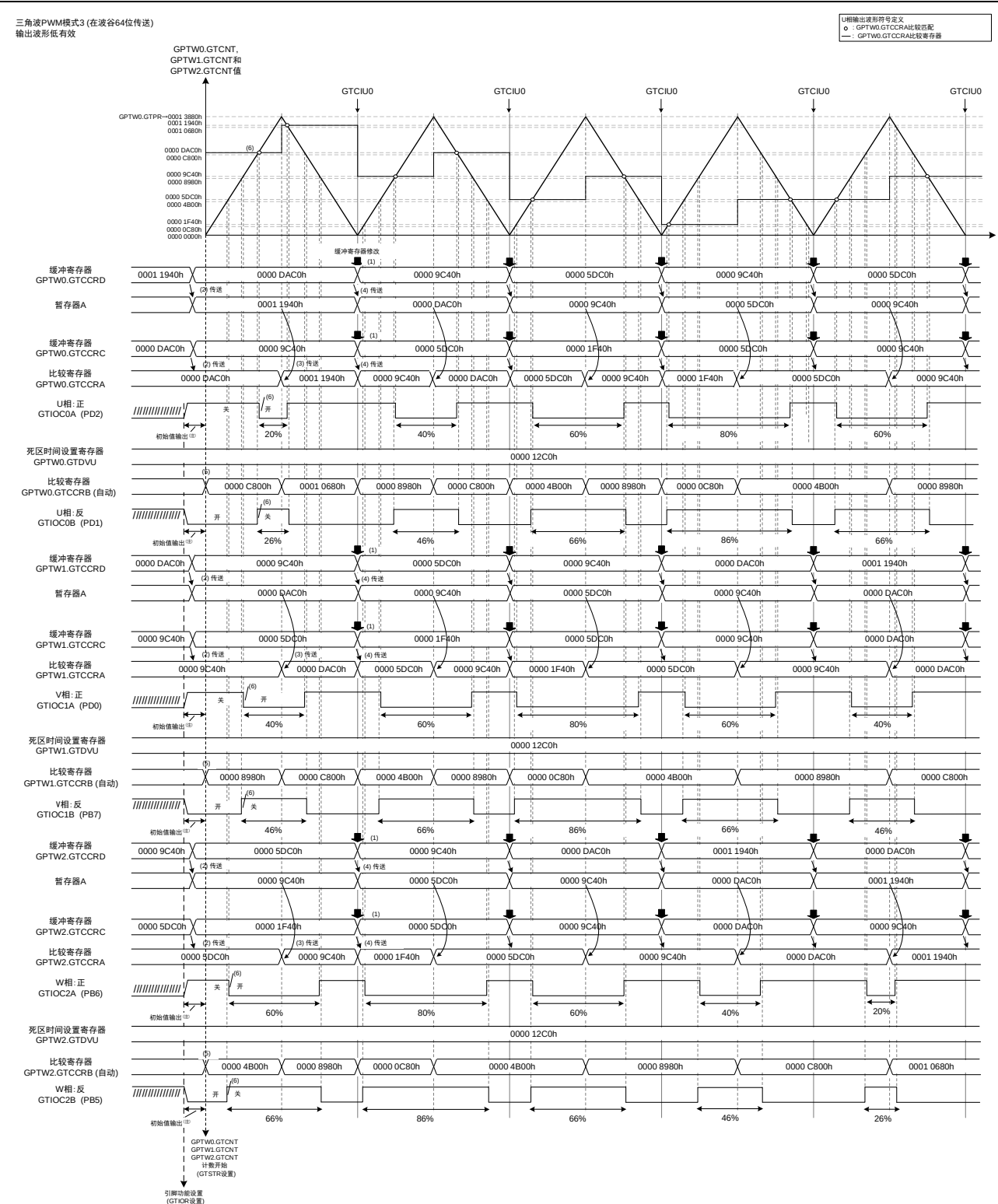


图 4.68 左右不对称 PWM 波形输出



注：示例代码波形在 PMR 寄存器设置时开始输出初始值。

图 4.69 示例代码运行

4.8.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.11 添加组件

项目	描述		
组件	通用 PWM 定时器		
配置名称	Config_GPT0	Config_GPT1	Config_GPT2
工作模式	三角波 PWM 模式 3		
资源	GPT0	GPT1	GPT2

图 4.70 至图 4.72 显示了 Config_GPT0 设置。相同的设置适用于 GPT1 和 GPT2。

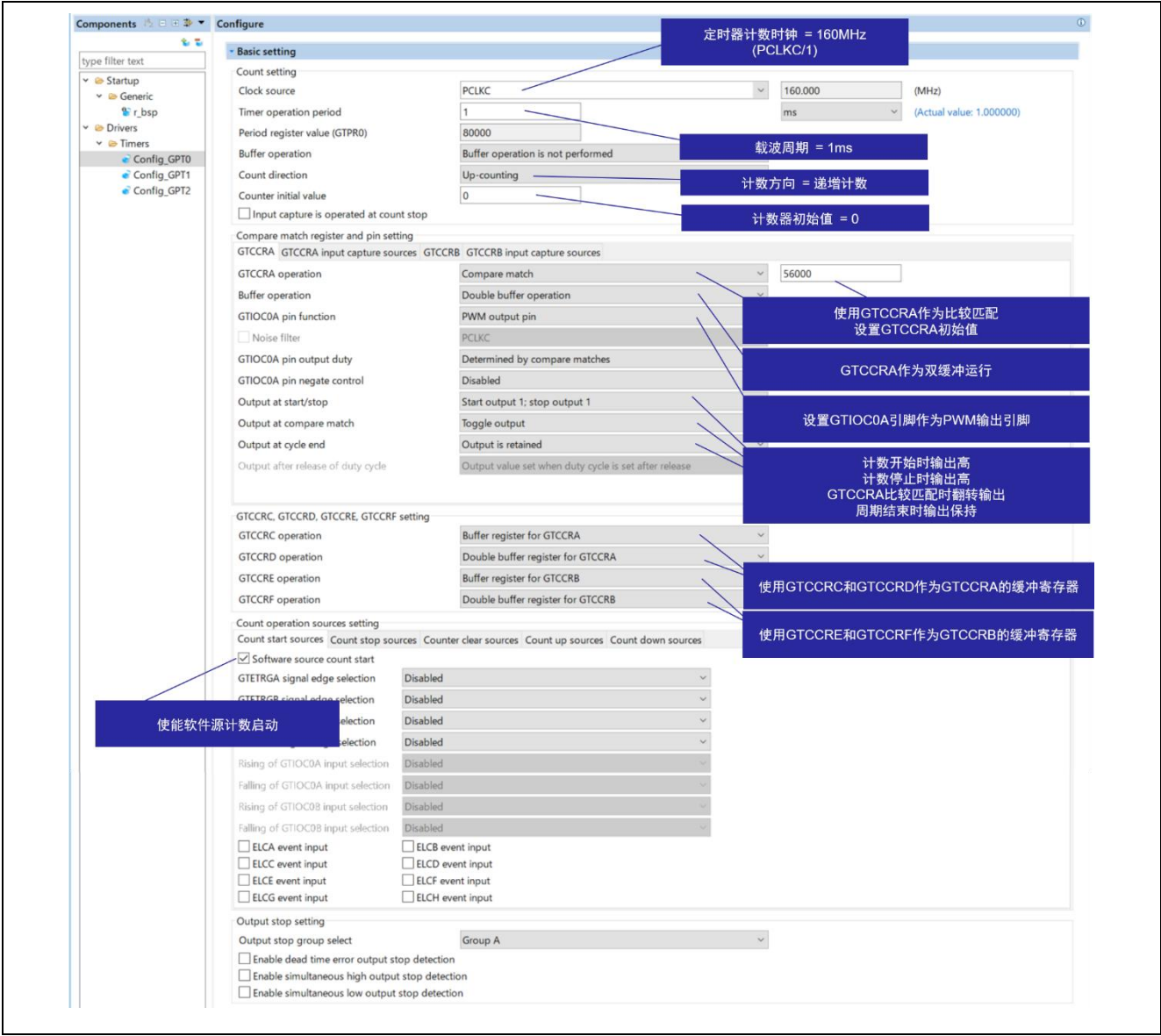


图 4.70 GPT0 设置（1/2）

Advance setting

Automatic dead time setting

☒ Automatically set GTCCRB0 using GTCCRA0 value and dead time

GTDVU value: 4800

☐ Enable buffer (GTDBU)

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0

☐ Enable buffer (GTDBD)

Conversion start request setting

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA): 100

Buffer operation: Buffer operation is not performed

Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTCIA0) Priority: Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTCIB0) Priority: Level 15 (highest)

☐ Enable dead time error interrupt (GDTE0) Priority: Level 15 (highest)

☐ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority: Level 15 (highest)

☒ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

GTCIV0/GTCIU0 interrupt skipping count: Skip count of 1

☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCCRA interrupt extended skipping function: No extended interrupt skipping

GTCCRB interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview | Board | Clocks | System | Components | Pins | Interrupts

图 4.71 GPT0 设置 (2/2)

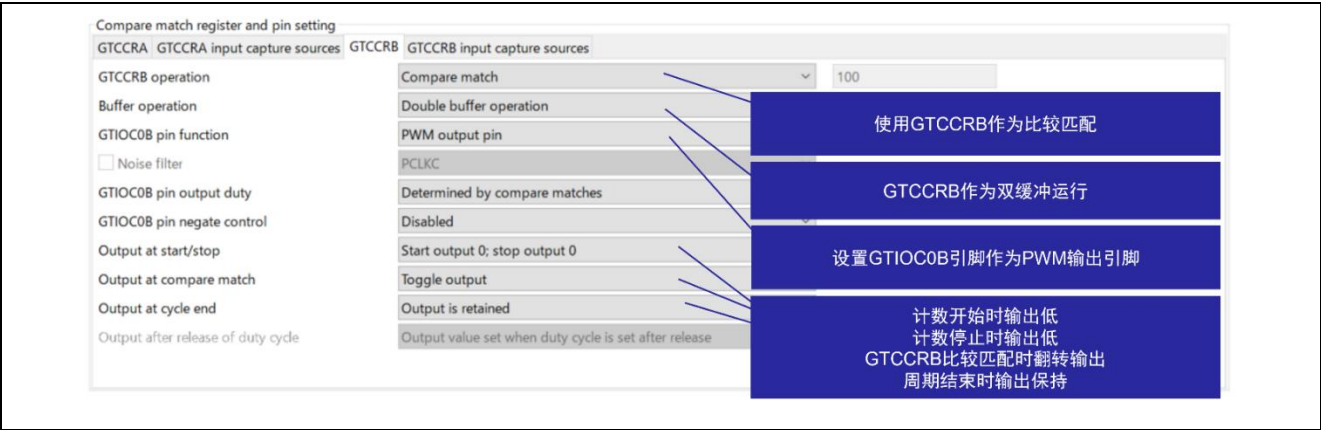


图 4.72 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.8.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中，调用计数开始函数 gpt0_gpt1_gpt2_start 并开始计数。

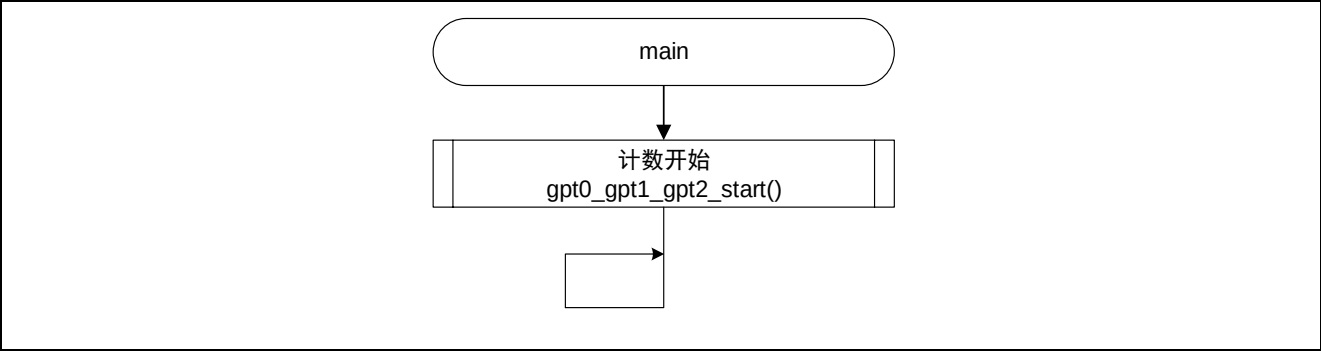


图 4.73 主函数

在计数开始函数中，GPT0、GPT1 和 GPT2 计数在使能 GTCIU0 中断后启动。
此功能是在 Smart Configurator 生成代码后新创建的。

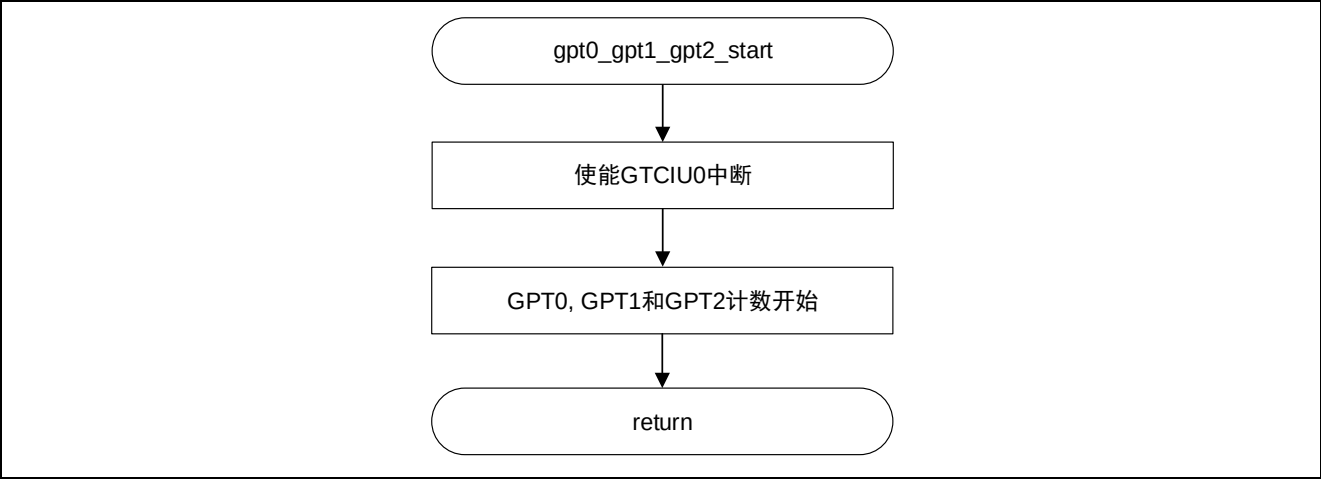


图 4.74 计数开始函数

在主函数之前执行的用户初始化函数 `R_Config_GPT0_Create_UserInit` 中设置缓冲寄存器的初始值并初始化变量。为了在第一个周期中设置第二个比较匹配寄存器的值，在设置缓冲寄存器的值后执行强制缓冲传送，然后设置暂存器和比较寄存器的值。此函数在 `R_Config_GPT0_Create` 函数中调用。

此函数初始化此示例代码中使用的以下变量。

- `g_ucduty_prv0`: 用于保存上一次 GPTW0.GTCCRC 寄存器值的变量
- `R_Config_GPT1_Create_UserInit` 和 `R_Config_GPT2_Create_UserInit` 也执行相同的流程。

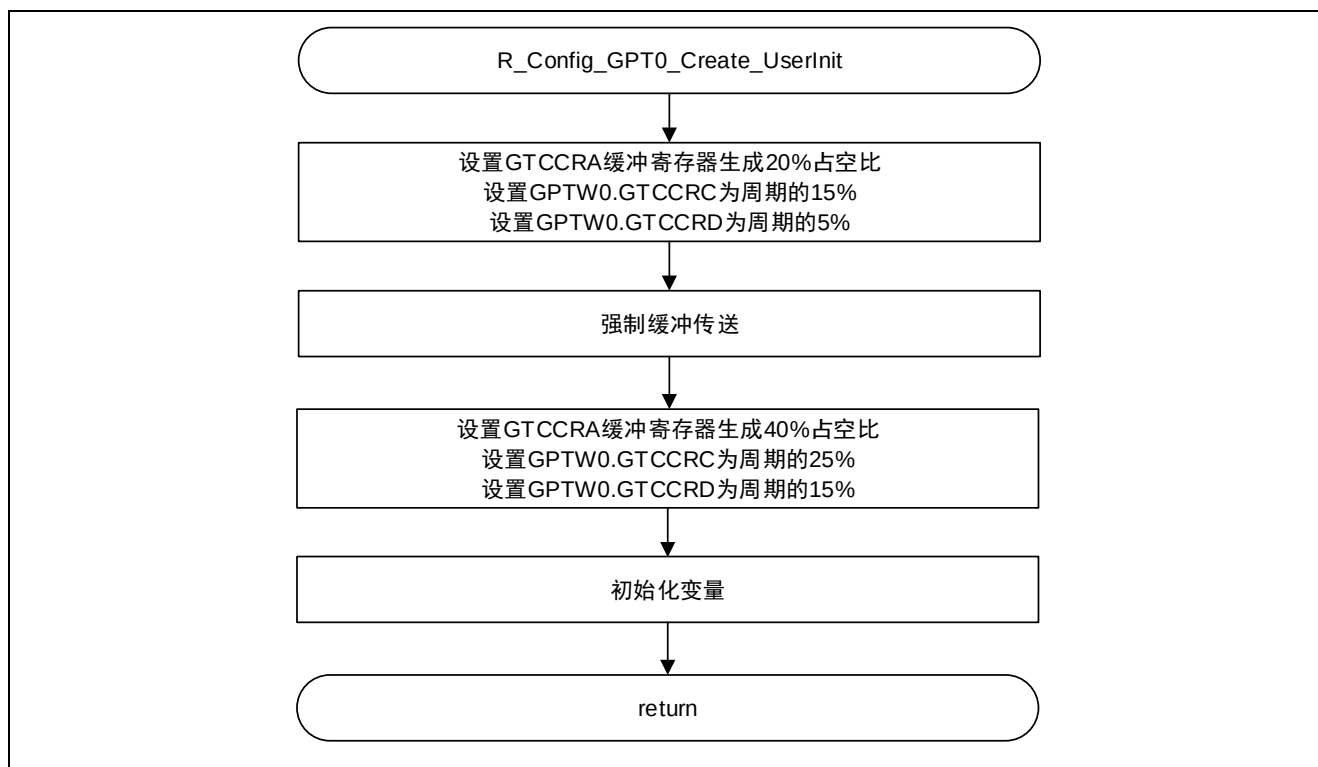


图 4.75 用户初始化函数

GTCIU0 中断处理函数根据缓冲寄存器 GPTW0.GTCCRC 的当前值以及上一次缓冲寄存器中设置的值来改变缓冲寄存器的值。

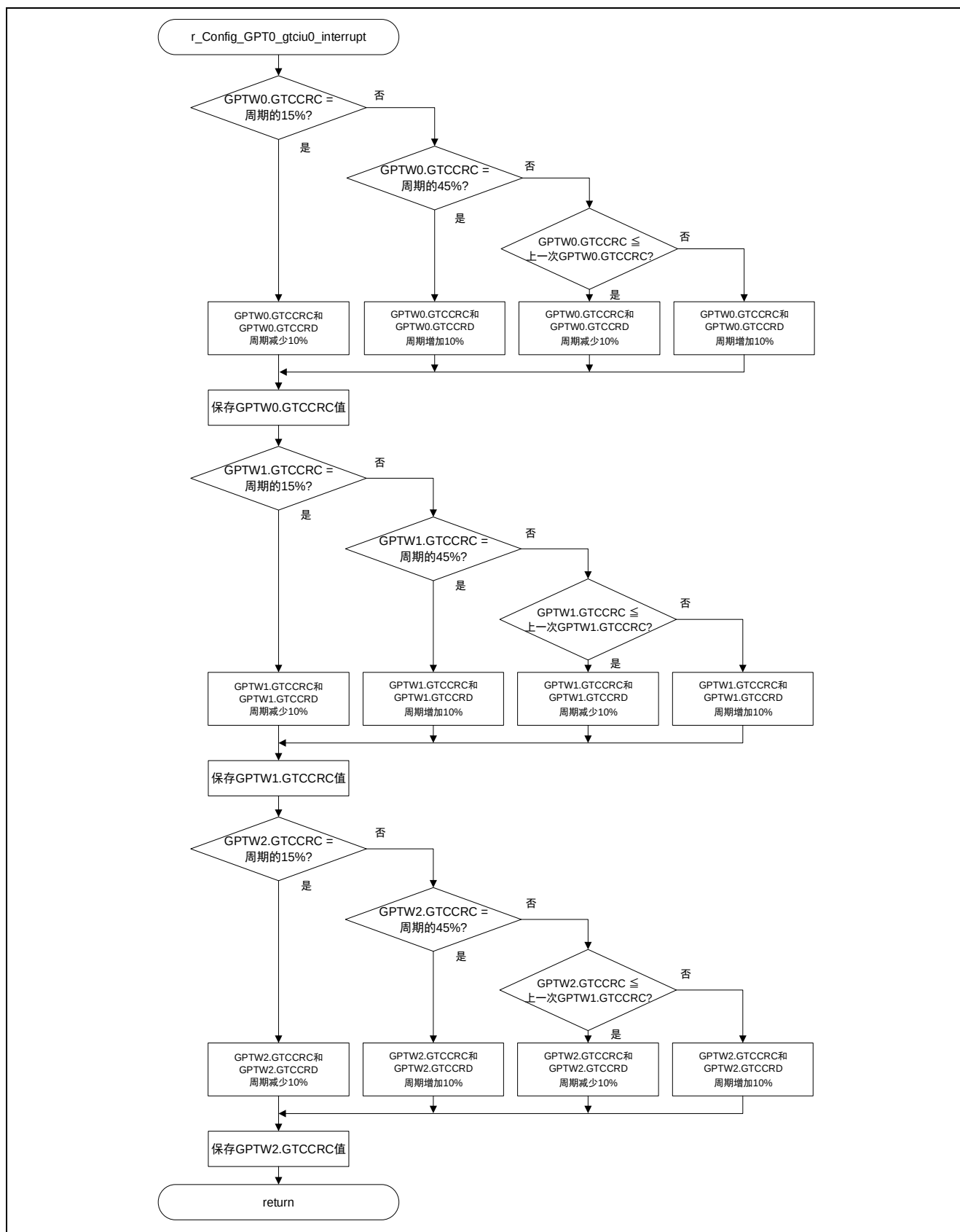


图 4.76 GTCIU0 中断处理函数

4.8.5 相关运行

4.8.5.1 每个间隔的单独死区时间自动设置

本示例代码使用死区时间自动设置功能，通过设置 GTDTCR.TDFER 位为 1，在反相的前侧和后侧生成具有公共转换时序的死区时间。

在死区时间自动设置功能中，可以分别设置波形的前侧和后侧的死区时间。反相前侧转换时序的死区时间在 GTDVU 寄存器中设置，后侧的死区时间在 GTDVD 寄存器中设置。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

4.8.5.2 具有死区时间自动设置功能的缓冲运行

此示例代码使用死区时间自动设置功能生成由 Smart Configurator 设置的确保固定死区时间段的反相波形。

死区时间自动设置功能使用 GTDBU 作为 GTDVU 的缓冲寄存器以及 GTDBD 作为 GTDVD 的缓冲寄存器，通过在计数周期结束时（当 GTCNT 计数器下溢发生（波谷）时）执行缓冲区传送来更新计数期间的死区时间段。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.4 死区时间自动设置功能。

Smart Configurator 设置如下所示。

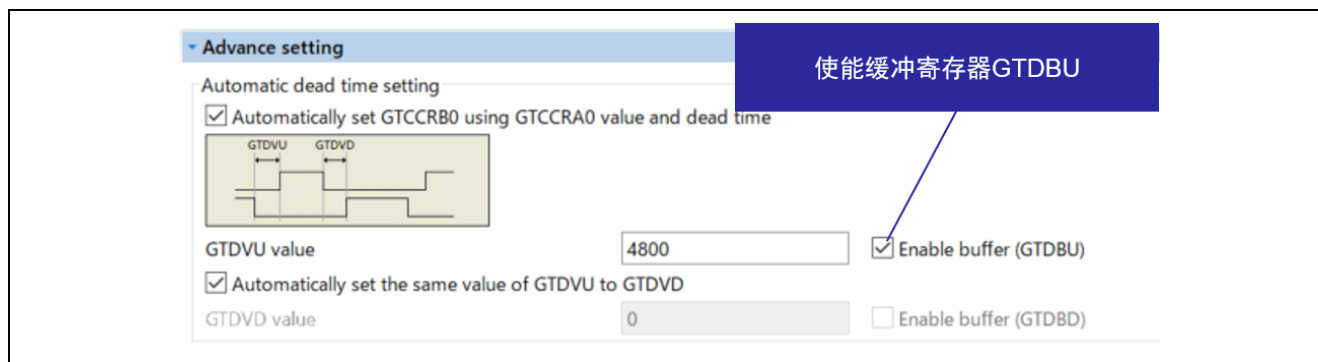


图 4.77 Smart Configurator 死区时间自动设置

下图显示了使用死区时间值的缓冲寄存器的死区时间自动设置功能的运行示例。

GTDBU 的值在 GTCNT 计数器下溢中断（GTCIU0）中修改（图 4.77 中（1））。当 GTCNT 计数器下溢发生（在波谷）时，GTDBU 的值被传送到死区时间设置寄存器 GTDVU（图 4.77 中（2）），并输出确保更新后的死区时间段的波形（图 4.77 中（3））。

在第三个周期中，在第一个周期结束时修改的 GTDBU 的值将传送到 GTDVU（图 4.77 中（4））。

第四个周期的死区时间设置导致反相前侧的转换时序产生死区时间错误（GTCCRA-GTDVU<0），因此输出具有校正转换时序的正相和反相波形（图 4.77 中（5））。

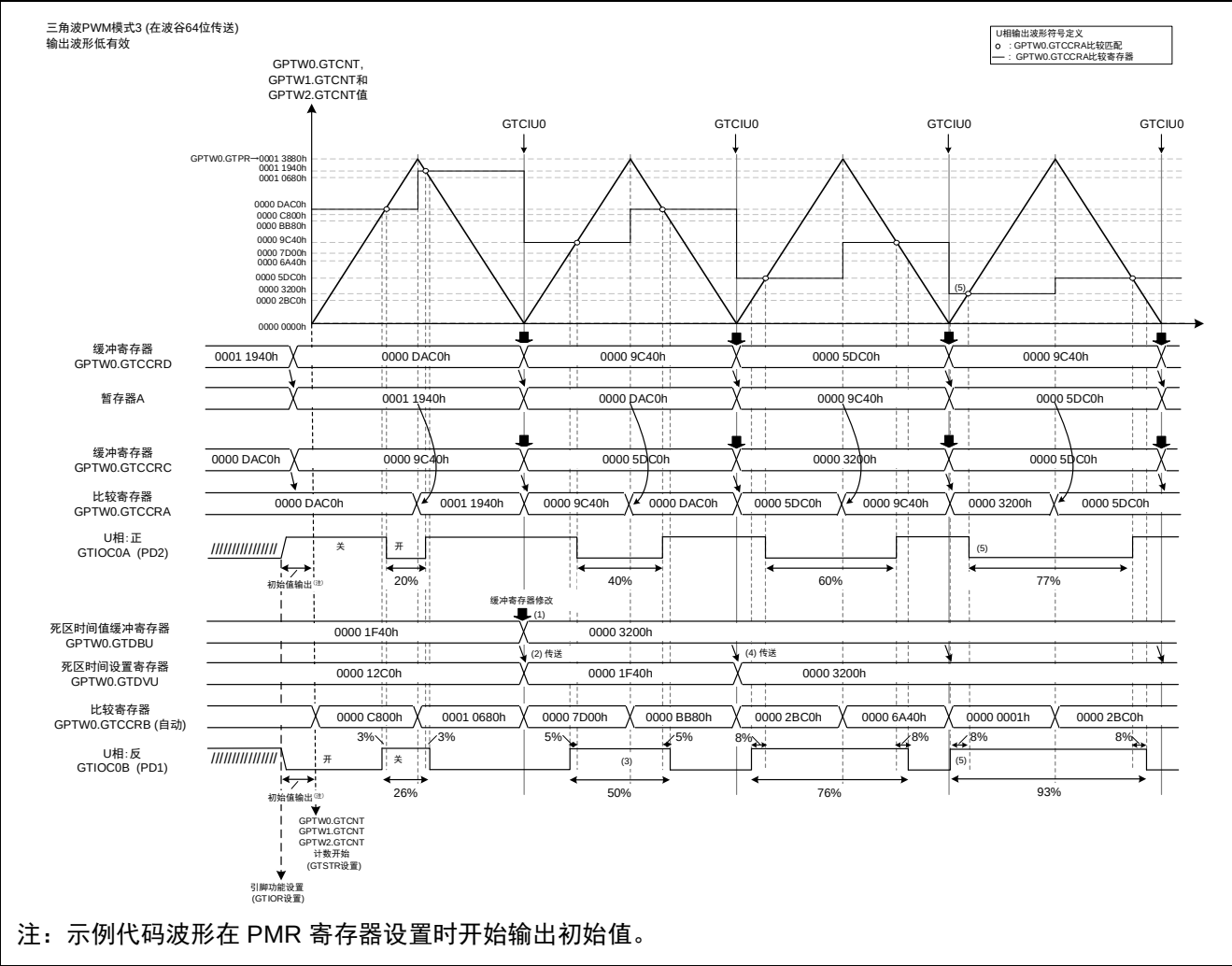


图 4.78 具有死区时间自动设置功能的缓冲运行

4.8.5.3 当不使用死区时间自动设置功能

- 目标示例代码文件名：r01an5995_rx66t_gptw_triangle_pwm3.zip。

图 4.79 显示了不使用死区时间自动设置功能（GTDTCR.TDE 位设为 0）的运行示例。

不使用死区时间自动设置功能时，通过设置比较匹配寄存器 GTCCRB（反相波形寄存器）以及缓冲寄存器 GTCCRE 和 GTCCRF 的值，生成确保死区时间段的波形。

与正相类似，在反相中当 GTCNT 计数器下溢（波谷）发生时通过修改缓冲寄存器 GTCCRE 和 GTCCRF（图 4.79 中（1）），当 GTCNT 计数器溢出（波峰）发生时传送暂存器 B 到比较寄存器 GTCCRB（图 4.79 中（2）），以及当 GTCNT 计数器下溢（波谷）发生时将缓冲寄存器 GTCCRF 传送到暂存器 B 和缓冲寄存器 GTCCRE 到比较寄存器 GTCCRB（图 4.79 中（3）），以在每个周期更新比较值。

此外，如图 4.69 中运行所示，确保相同的死区时间段。

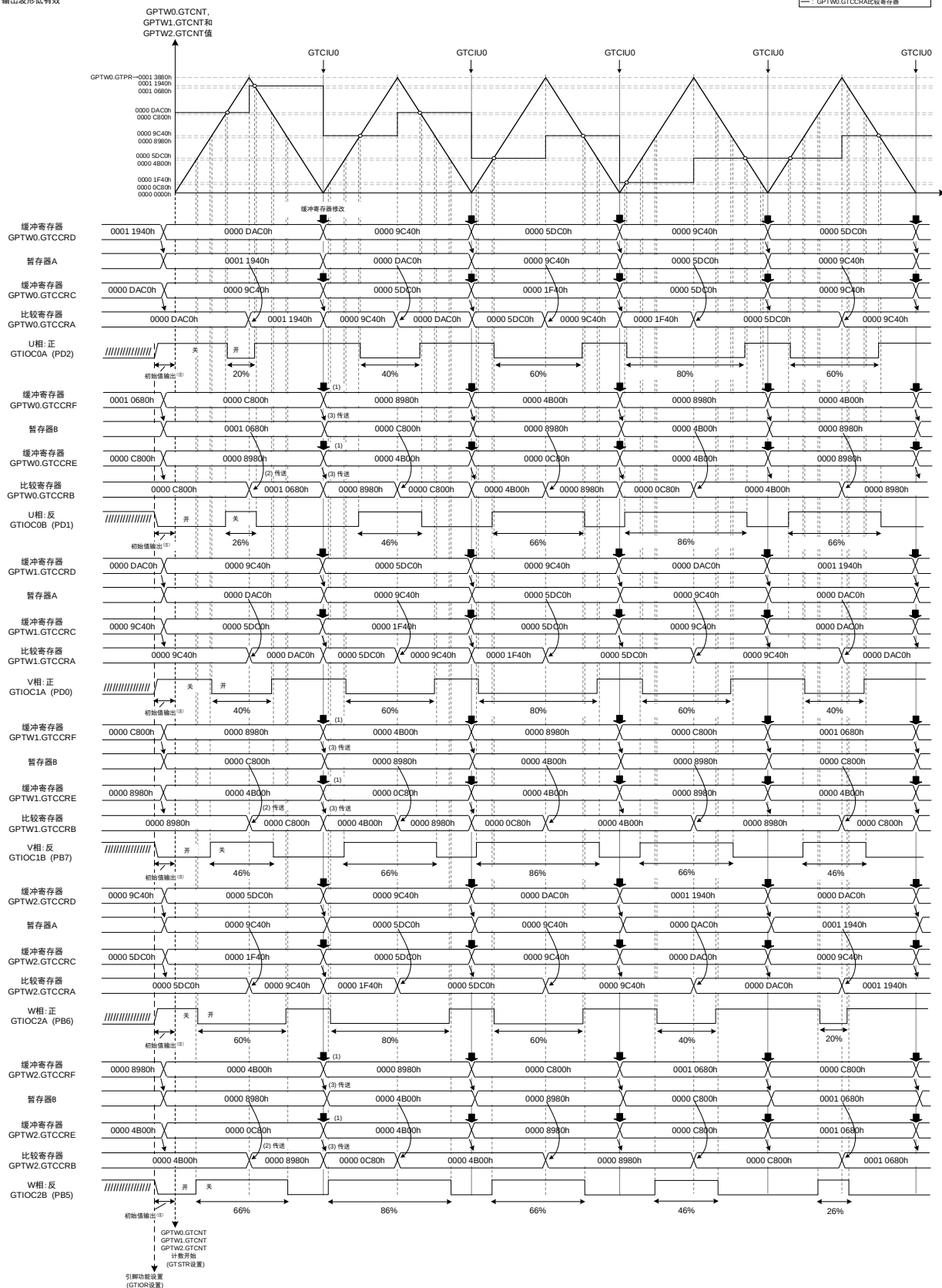


图 4.79 不使用死区时间自动设置功能时 (三角波 PWM 模式 3)

4.8.6 使用时的注意事项

4.8.6.1 多通道计数启动

在此示例代码中，定时器软件启动寄存器 GTSTR 的 CSTRT0、CSTRT1 和 CSTRT2 位在 gpt0_gpt1_gpt2_start 函数中同时设置，以同时启动 GPTW0、GPTW1 和 GPTW2 计数。

使用 Smart Configurator 生成的 R_Config_GPT0_Start、R_Config_GPT1_Start 和 R_Config_GPT2_Start 函数时，因为每个函数调用时间不同，计数开始时间可能不同。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.8.1 软件同步操作。

4.8.6.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较匹配寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

设置比较寄存器 GTCCRA 的值应满足以下限制。

$GTCCRA > GTDVU$

$GTCCRA > GTDVD$

$GTCCRA < GTPR$

如果在计数操作期间将 GTCCRA 设置为 0000 0000h 或大于 GTPR 的设置值，则输出保护功能被激活。

但是，如果不满足以下条件，则功能将无法正常运行。

当计数操作开始时 GTCCRA 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设定值时。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.8.4 GTIOCnm 引脚输出的输出保护功能 (n = 0 到 9; m = A、B)。

不使用死区时间自动设置功能时，设置 GTCCRA (GTCCRB) 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设定值。当在 GTCCRA 寄存器中设置 0000 0000h 或与 GTPR 寄存器相同的值时，仅当满足 $[GTCCRA (GTCCRB) = 0000 0000h]$ 或 $[GTCCRA (GTCCRB) = GTPR]$ 时，才会在一个周期中发生比较匹配。此外，如果在 GTCCRA 中设置的值超过 GTPR 寄存器的设置值，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 GTCCRM 寄存器在比较匹配操作期间的设置 (m = A 到 F)，(1) 在三角波 PWM 模式下进行死区时间自动设置时，以及 (2) 在三角波 PWM 模式下未进行死区时间自动设置时。

4.9 锯齿波 PWM 模式 占空比 0%到 100%

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_pwm_50to100.zip

4.9.1 概要

GPTW 锯齿波 PWM 模式可用于根据 GTCCRA 寄存器比较匹配和 GTUDDTYC 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用锯齿波 PWM 模式并重复输出以下波形，包括占空比 0%和 100%。

- 占空比切换: 50% → 80% → 100% → 80% → 50% → 0% → ...

基本操作是当使用缓冲寄存器 GTCCRC 在 GTCNT 计数器溢出发生时，通过将缓冲寄存器的值传送到 GTCCRA 来改变占空比。当在占空比 0%和 100%之间切换时，GTCNT 计数器溢出发生时执行修改 GTUDDTYC 寄存器的过程。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波 PWM 模式
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOC0A 引脚为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出低
 - GTCCRA 比较匹配时输出高
 - 周期结束时输出低
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 4.81

在 Smart Configurator 中设置。
设置方法请参考章节 4.9.3。

此示例代码的锯齿波 PWM 模式输出如下所示。

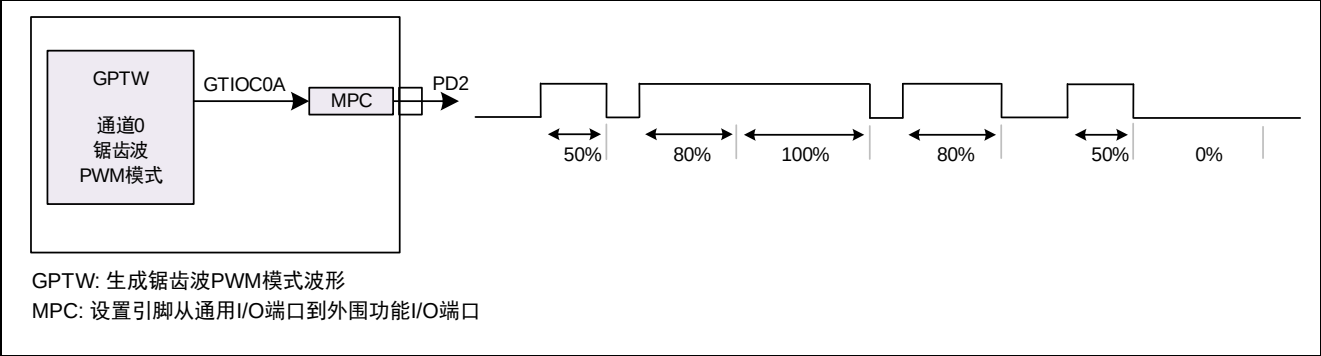


图 4.80 锯齿波 PWM 模式输出

4.9.2 运行说明

示例代码的运行如图 4.81 所示。基本操作是通过在 GTCNT 计数器溢出中断（GTCIV0）中修改缓冲寄存器 GTCCRC 的值，并在 GTCNT 计数器溢出发生时将 GTCCRC 的值传送到 GTCCRA，从而改变每个周期的占空比的设置。

- 100%占空比输出（下图中（1））
通过将 GTUDDTYC.OADTY 位设为 11b，从下一个周期输出变为高。即使发生 GTCCRA 比较匹配，波形也不会改变。
- 0%占空比输出（下图中（2））
通过将 GTUDDTYC.OADTY 位设为 10b，从下一个周期输出变为低。即使发生比较匹配，波形也不会改变。
- 从占空比 100%或 0%切换（下图中（3））
通过将 GTUDDTYC.OADTY 位设为 00b，通过比较从下一个周期开始可以改变占空比输出。

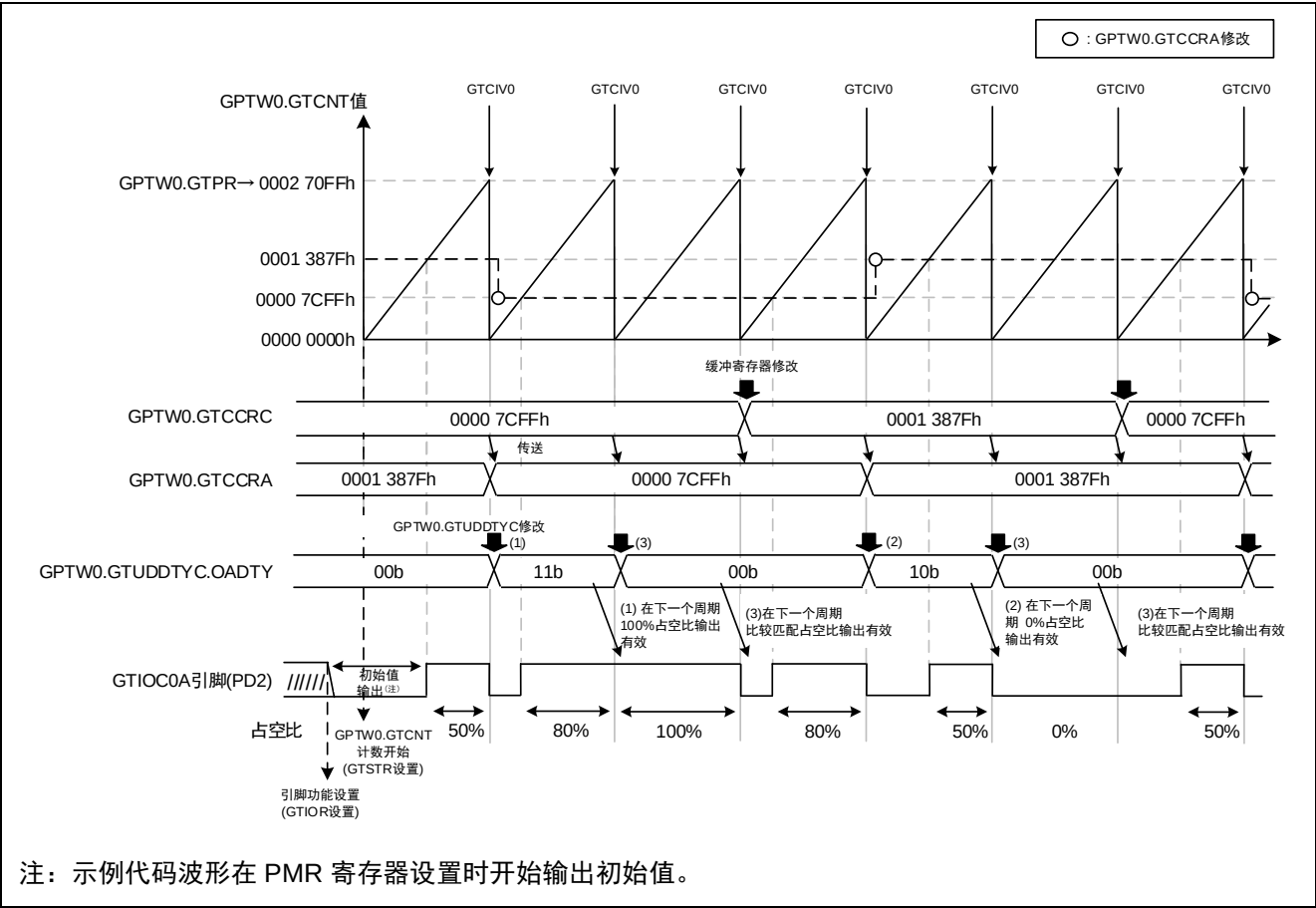


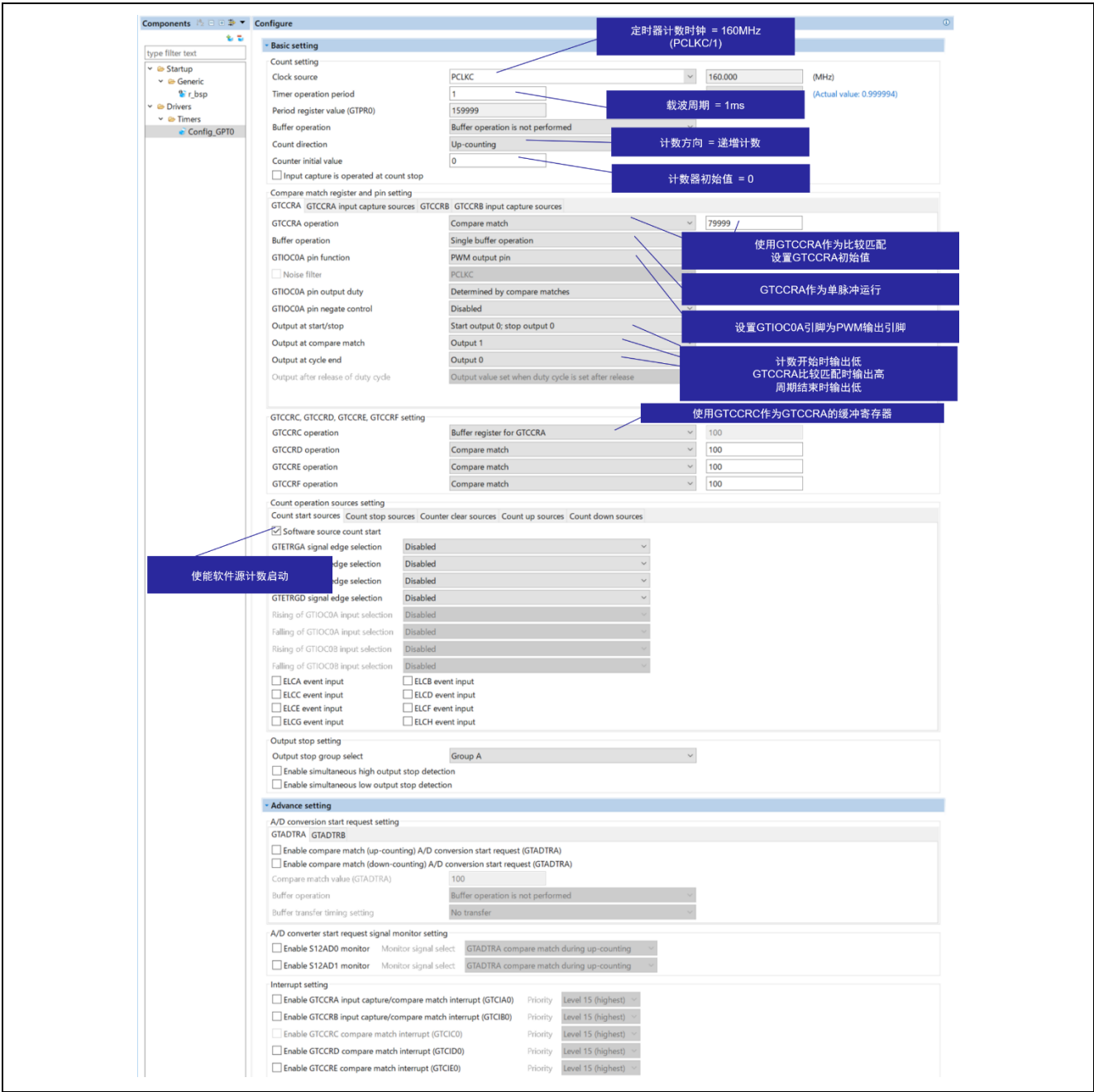
图 4.81 示例代码运行

4.9.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考 4.1.4 添加组件。

表 4.12 添加组件

项目	描述
项目	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	锯齿波 PWM 模式
资源	GPT0



Advance setting

A/D conversion start request setting
 GTADTRA GTADTRB
☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)
☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)
 Compare match value (GTADTRA) 100
 Buffer operation Buffer operation is not performed
 Buffer transfer timing setting No transfer

A/D converter start request signal monitor setting
☐ Enable S12AD0 monitor Monitor signal select GTADTRA compare match during up-counting
☐ Enable S12AD1 monitor Monitor signal select GTADTRA compare match during up-counting

Interrupt setting
☐ Enable GTCRA input capture/compare match interrupt (GTCIA0) Priority Level 15 (highest)
☐ Enable GTCRB input capture/compare match interrupt (GTCIB0) Priority Level 15 (highest)
☐ Enable GTCRC compare match interrupt (GTCIC0) Priority Level 15 (highest)
☐ Enable GTCRD compare match interrupt (GTCID0) Priority Level 15 (highest)
☐ Enable GTCRE compare match interrupt (GTCIE0) Priority Level 15 (highest)
☐ Enable GTCRF compare match interrupt (GTCIF0) Priority Level 15 (highest)
☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority Level 15 (highest)
☐ Enable GTCNT underflow interrupt (GTCIU0) Priority Level 15 (highest)

Interrupt and A/D converter start request skipping setting
 GTCIV0/GTCIU0 interrupt skipping function Skipping is not performed
 Skip count of 1

Link GTCIB0 with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTCIC0 with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTCID0 with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTCIE0 with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTCIF0 with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function
☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting
 Extended interrupt skipping counter 1 count source Skipping is not performed
 Skip count Skip count of 1
 Extended interrupt skipping counter 2 count source Skipping is not performed
 Skip count Skip count of 1
 Counter 2 initial skip count Skip count of 1
 GTCRA interrupt extended skipping function No extended interrupt skipping
 GTCRB interrupt extended skipping function No extended interrupt skipping
 GTCRC interrupt extended skipping function No extended interrupt skipping
 GTCRD interrupt extended skipping function No extended interrupt skipping
 GTCRE interrupt extended skipping function No extended interrupt skipping
 GTCRF interrupt extended skipping function No extended interrupt skipping
 Overflow interrupt extended skipping function No extended interrupt skipping
 Underflow interrupt extended skipping function No extended interrupt skipping
 GTADTRA interrupt extended skipping function No extended interrupt skipping
 GTADTRB interrupt extended skipping function No extended interrupt skipping

Extended buffer transfer skipping setting
 GTCRA buffer transfer extended skipping function No extended interrupt skipping
 GTCRB buffer transfer extended skipping function No extended interrupt skipping
 GTPR buffer transfer extended skipping function No extended interrupt skipping
 GTADTRA buffer transfer extended skipping function No extended interrupt skipping
 GTADTRB buffer transfer extended skipping function No extended interrupt skipping

HRPWM setting
 High Resolution PWM setting
☐ Enable output high resolution PWM waveform
☐ Enable operation of rising and falling edge adjustment circuit
 GTIOC0A pin rising edge delay select Apply delay of 0/32 times PCLKC period
 GTIOC0A pin falling edge delay select Apply delay of 0/32 times PCLKC period
 GTIOC0B pin rising edge delay select Apply delay of 0/32 times PCLKC period
 GTIOC0B pin falling edge delay select Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.83 GPT0 设置 (2/2)

4.9.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

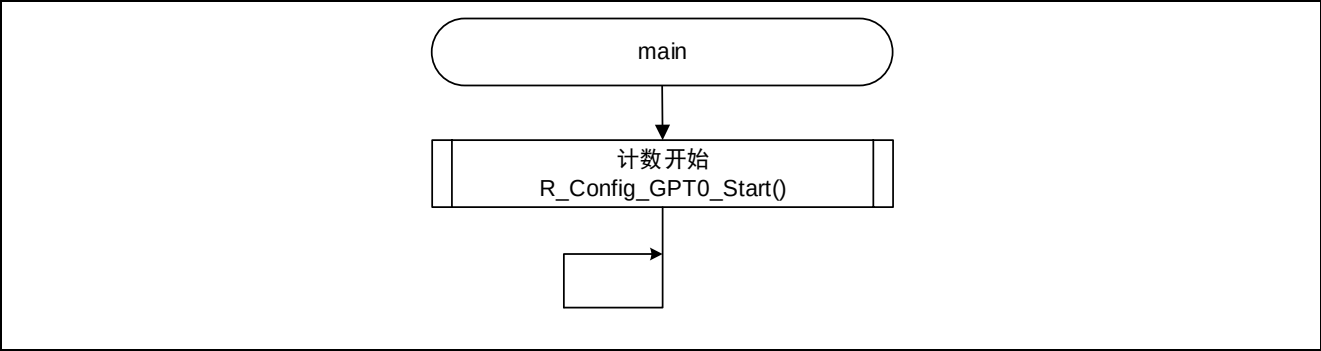


图 4.84 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 用来设置缓冲寄存器的初始值和初始化变量。此函数在 R_Config_GPT0_Create 函数中调用。

此示例代码使用以下变量。

- s_duty_list_counter: 用于从占空比列表中读取的计数器变量

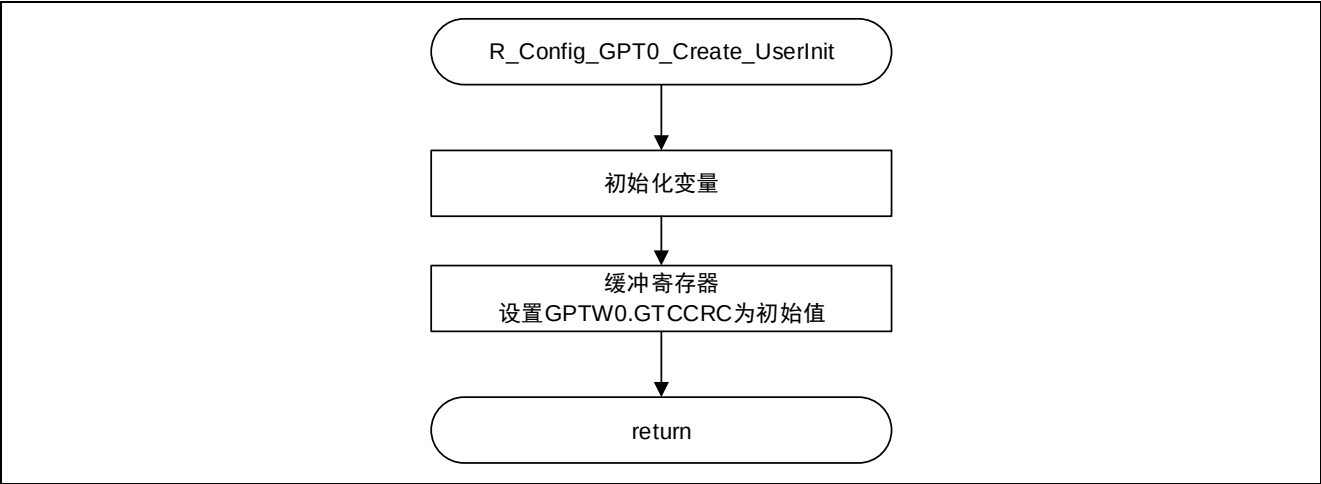


图 4.85 用户初始化函数

GTCIV0 中断处理函数根据要设置的下一个占空比更改 GTCCRC 寄存器和 GTUDDTYC 寄存器的值。

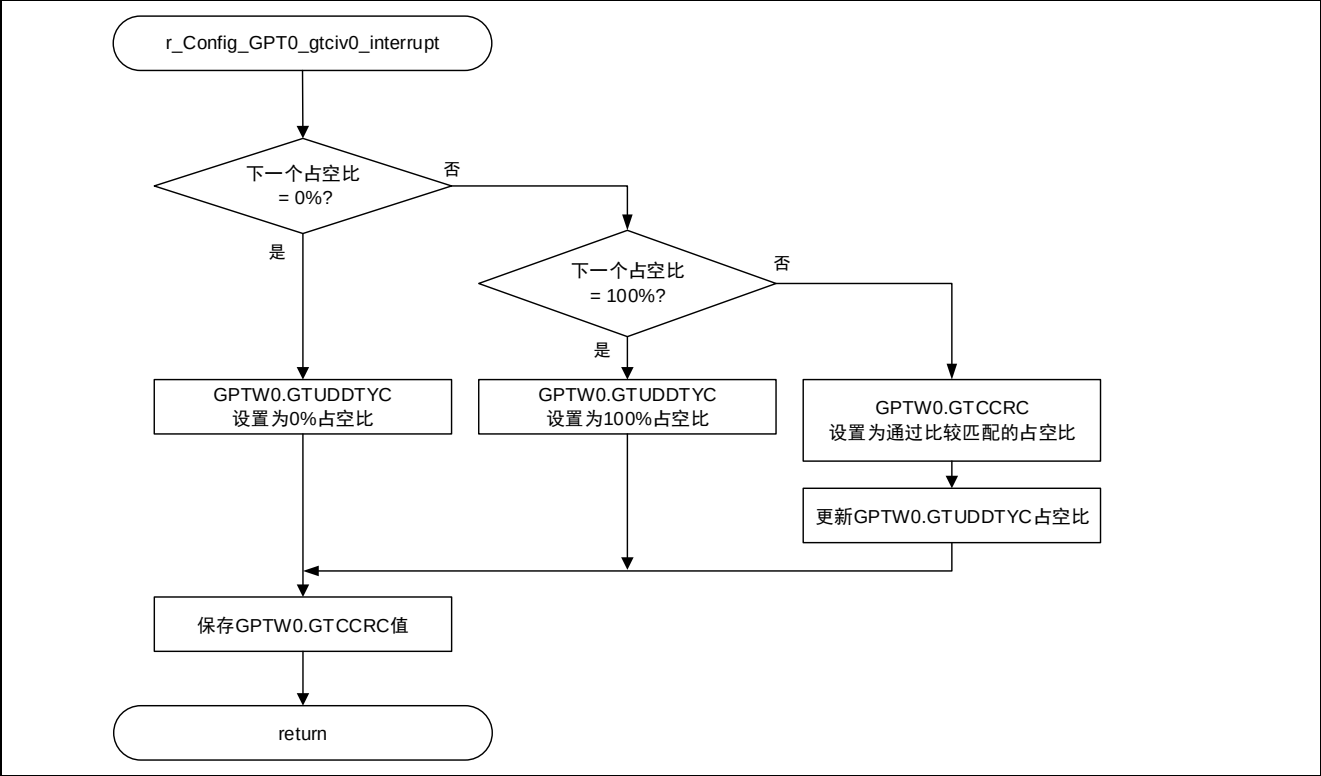


图 4.86 GTCIV0 中断处理函数

4.9.5 使用时的注意事项

4.9.5.1 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间, 并使用比较寄存器 GTCCRA (正相波形寄存器) 的值来更新比较匹配寄存器 GTCCRB (反相波形寄存器) 的值。

缓冲寄存器 GTCCRC 和 GTCCRD 的值应设置为满足以下限制。如果不满足这些限制, 可能无法获得具有固定死区时间的正确输出波形。

- 在递增计数中:
 $GTCCRC < GTCCRD$
 $GTCCRC > GTDVU$
 $GTCCRD < GTPR - GTDVD$
- 在递减计数中:
 $GTCCRC > GTCCRD$
 $GTCCRC < GTPR - GTDVU$
 $GTCCRD > GTDVD$

此外, 如果死区时间没有自动设置, 应设置缓冲寄存器 GTCCRC 和 GTCCRD (或 GTCCRE 和 GTCCRF) 满足以下限制。如果不满足这些限制, 则不会发生两次比较匹配并且无法执行脉冲输出。

- 在递增计数中: $0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR$
- 在递减计数中: $GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0$

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F), (3) 在锯齿波单触发脉冲模式下进行死区时间自动设置时和 (4) 在锯齿波单触发脉冲模式下未进行死区时间自动设置时。

4.9.5.2 比较匹配时 100% 占空比输出

若不更改 GTUDDTYC 寄存器, 则无法在比较匹配时将输出占空比切换为高有效 100% 占空比。为了输出高有效 100% 占空比, 设置 GTUDDTYC.OADTY 位为 11b。

如果在此示例代码设置中将 GTCCRA 寄存器设置为 0, 则无法输出 100% 占空比, 因为在 GTCNT 计数器溢出发生后, 输出一个时钟周期的低电平, 然后输出高电平。

MTU 可以输出 100% 占空比, 因为占空比寄存器和周期寄存器被设置为相同的值, 并且当计数器清除和比较匹配同时发生时波形不会改变。在 GPTW 锯齿波 PWM 模式下, 100% 占空比不能以相同方式输出, 因为当一个周期结束和比较匹配同时发生时, 周期结束时的输出设置优先, 并且波形发生变化。

有关 GPTW 中同时发生 GTCNT 溢出中断和比较匹配时的波形输出的详情, 请参考 RX66T Group User's Manual: Hardware, 章节 24.2.14 通用 PWM 定时器 I/O 控制寄存器 (GTIOR) 中表 24.4 下的注释。

4.9.5.3 占空比 0%和 100%输出期间比较匹配运行

在此示例代码中，输出占空比 0%和 100%是基于 GTUDDTYC.OADTY 位的设置值决定的。当占空比设置为 0%或 100%时，GPTW 中的比较匹配继续，并执行中断输出和缓冲传送运行。

此示例代码不使用比较匹配中断，因此如果你要使用比较匹配中断，请在占空比 0%和 100%输出期间谨慎使用中断。

4.10 锯齿波单触发脉冲 占空比 0%到 100%

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_1shotpls_50to100_dt.zip

4.10.1 概要

当使用 GPTW 锯齿波单触发脉冲模式时, 通过设置 GTPR 寄存器中的周期, GTCNT 计数器可以在锯齿波 (半波) 下运行, 并通过 GTCCRA 和 GTCCRB 寄存器之间的比较匹配, 可以从 GTIOCNB 和 GTIOCnA 引脚 (n = 0 到 9) 输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了在锯齿波单触发脉冲模式下使用死区时间自动设置功能并重复输出以下波形, 包括占空比 0%和 100%。

- GTIOC0A 引脚高带宽切换: 50% → 80% → 100% → 80% → 50% → 0% → ...
- GTIOC0B 引脚低带宽切换: 60% → 90% → 100% → 90% → 60% → 0% → ...

当 GTCCRA 比较匹配发生时, 暂存器 A 的值被传送到比较寄存器 GTCCRA。当 GTCNT 计数器溢出发生时, 通过分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA 来改变占空比。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波单触发脉冲模式
- 使用通道 0
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOCnA 引脚为 PWM 输出引脚
 - 计数开始时输出低
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用 GTCCRB 作为占空比输出比较匹配
 - 设置 GTIOCnB 引脚为 PWM 输出引脚
 - 计数开始时输出高
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
- 使用双缓冲寄存器
 - 使用 GTCCRC 作为 GTCCRD 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.89

在 Smart Configurator 中设置。
设置方法请参考章节 4.10.3。

此示例代码的锯齿波单触发脉冲模式输出如下所示。

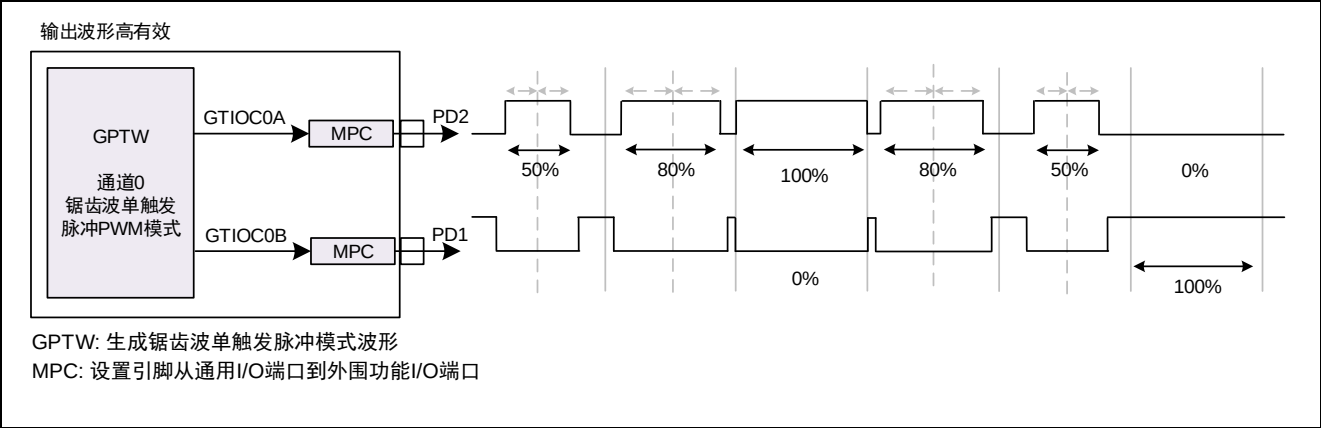


图 4.87 锯齿波单触发脉冲模式输出

4.10.2 运行说明

示例代码运行如图 4.89 所示。通过在 GTCNT 计数器溢出中断（GTCIV0）时修改缓冲寄存器 GTCCRC 和 GTCCRD 的值来改变每个周期的占空比设置（图 4.89 中（1））。

使用 Smart Configurator 生成的代码中缓冲寄存器的初始值设置为与此示例代码中的比较寄存器相同的值。因此，在计数开始之前，在用户初始化函数 `R_Config_GPT0_Create_UserInit` 中设置缓冲寄存器的值。通过强制缓冲传送，寄存器中的设置值分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.89 中（2））。

在计数开始后，基本运行是通过重复下面步骤 1 和 2 来更新比较寄存器中的数据。

1. 当 GTCCRA 比较匹配发生时，从暂存器 A 传送到比较寄存器 GTCCRA（图 4.89 中（3））。
2. 当 GTCNT 计数器溢出发生时，分别从缓冲寄存器 GTCCRD 传送到暂存器 A 和从缓冲寄存器 GTCCRC 传送到比较寄存器 GTCCRA（图 4.89 中（4））。

此外，因为使用了死区时间自动设置功能，GTCCRB 寄存器根据 GTCCRA 更新自动设置。GTDVU 和 GTDVD 设置相同的值。

- 100%占空比输出（图 4.89 中（5））

通过设置 GTUDDTYC.OADTY 位为 11b，GTIOC0A 引脚从下一次 GTCNT 计数器溢出发生时输出高电平，并且即使发生 GTCCRA 比较匹配，波形也不改变。

通过设置 GTUDDTYC.OBDTY 位为 10b，GTIOC0B 引脚从下一次 GTCNT 计数器溢出发生时输出低电平，并且即使发生 GTCCRB 比较匹配，波形也不改变。在此示例代码中，此位被设置为 10b，使得 GTIOC0B 引脚在一个周期内输出低电平。

- 0%占空比输出（图 4.89 中（6））

通过设置 GTUDDTYC.OADTY 位为 10b，GTIOC0A 引脚从下一次 GTCNT 计数器溢出发生时输出低电平，并且即使发生 GTCCRA 比较匹配，波形也不改变。

通过设置 GTUDDTYC.OBDTY 位为 11b，GTIOC0B 引脚从下一次 GTCNT 计数器溢出发生时输出高电平，并且即使发生 GTCCRB 比较匹配，波形也不改变。在此示例代码中，此位被设置为 11b，使得 GTIOC0B 引脚在一个周期内输出高电平。

- 从占空比 100%和 0%切换（图 4.89 中（7））

通过设置 GTUDDTYC.OADTY 位为 00b，GTIOC0A 引脚根据 GTCCRA 比较匹配从下一次 GTCNT 计数器溢出发生时输出占空比。在占空比 100%和 0%解除后的输出由 GTIOR.GTIOA[3:2]和 GTUDDTYC.OADTYR 位决定。在此示例代码中，输出被屏蔽的比较匹配输出值低。

通过设置 GTUDDTYC.OBDTY 位为 00b，GTIOC0B 引脚根据 GTCCRB 比较匹配从下一次 GTCNT 计数器溢出发生时输出占空比。在占空比 100%和 0%解除后的输出由 GTIOR.GTIOB[3:2]和 GTUDDTYC.OBDTYR 位决定。在此示例代码中，输出被屏蔽的比较匹配输出值高。

以下显示解除占空比 100%和 0%后 GTIOC0A 引脚输出的示例。

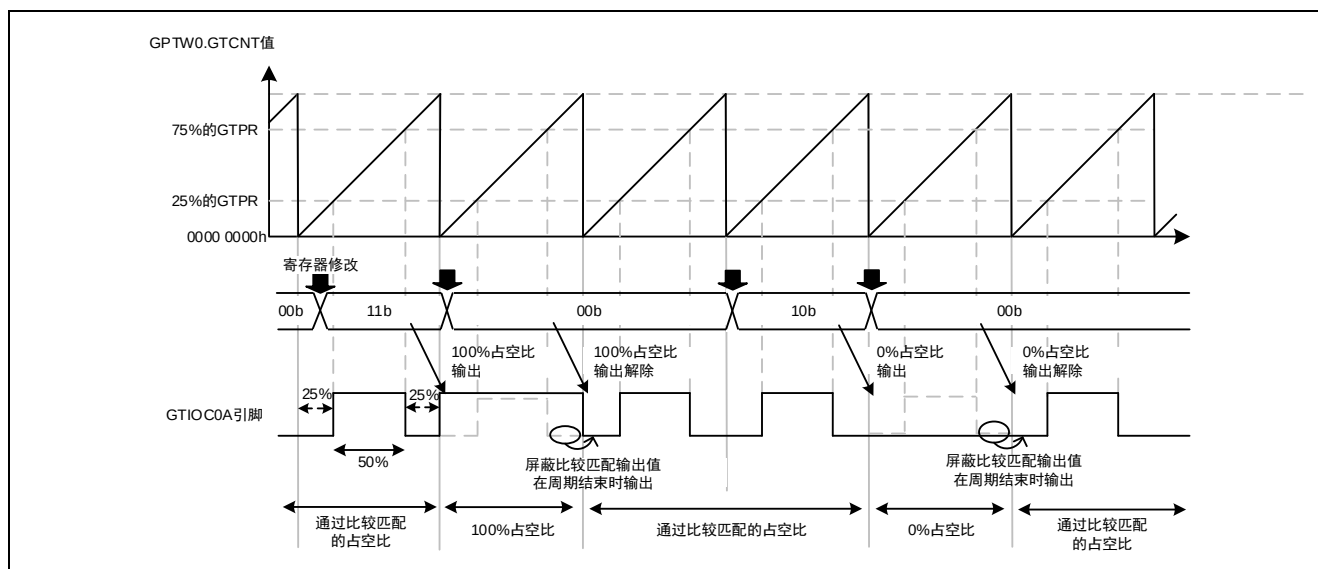


图 4.88 占空比 100%和 0%解除后的占空比产生和输出

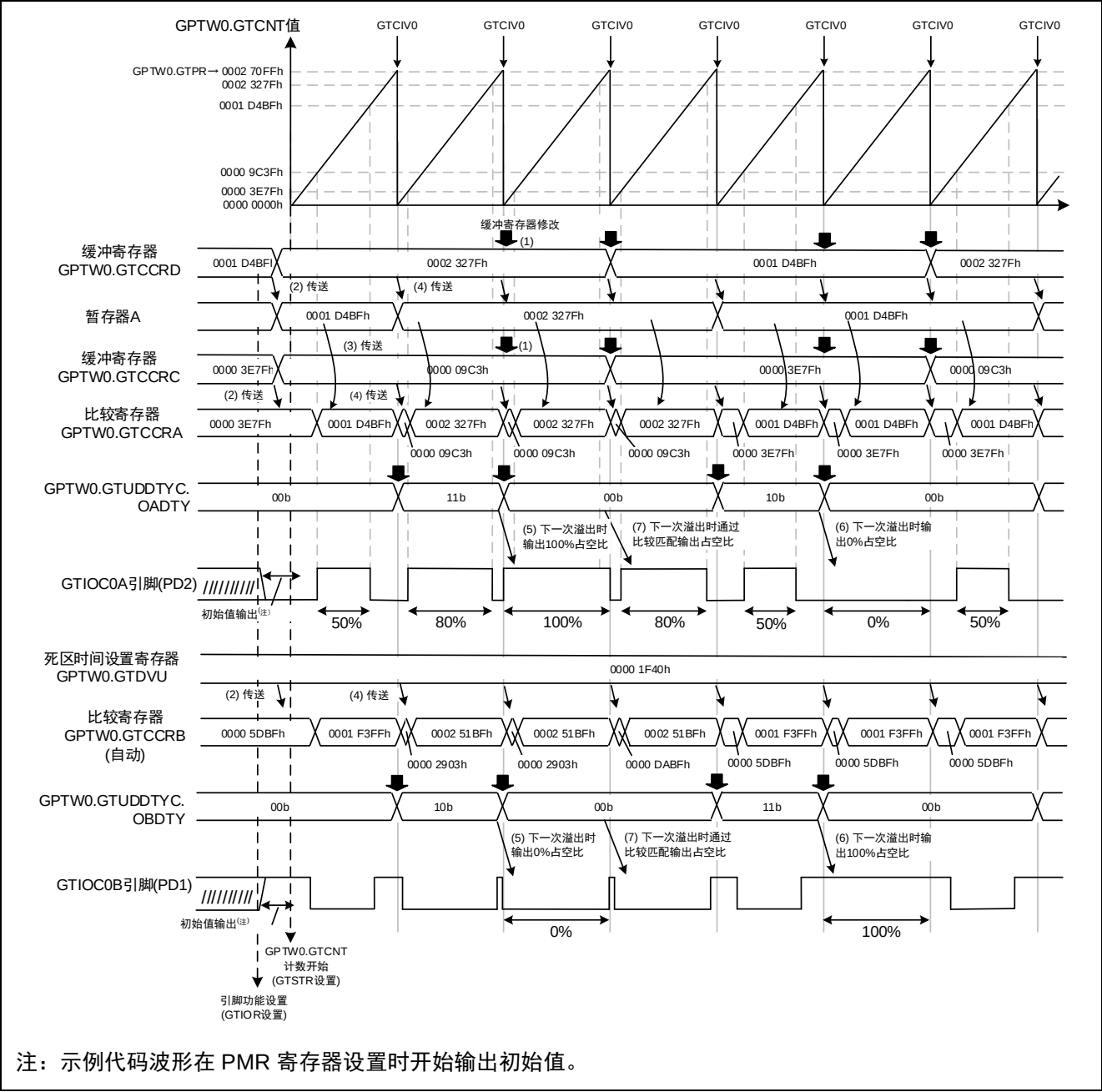


图 4.89 示例代码运行

4.10.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.13 添加组件

项目	描述
组件	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	锯齿波单触发脉冲模式
资源	GPT0

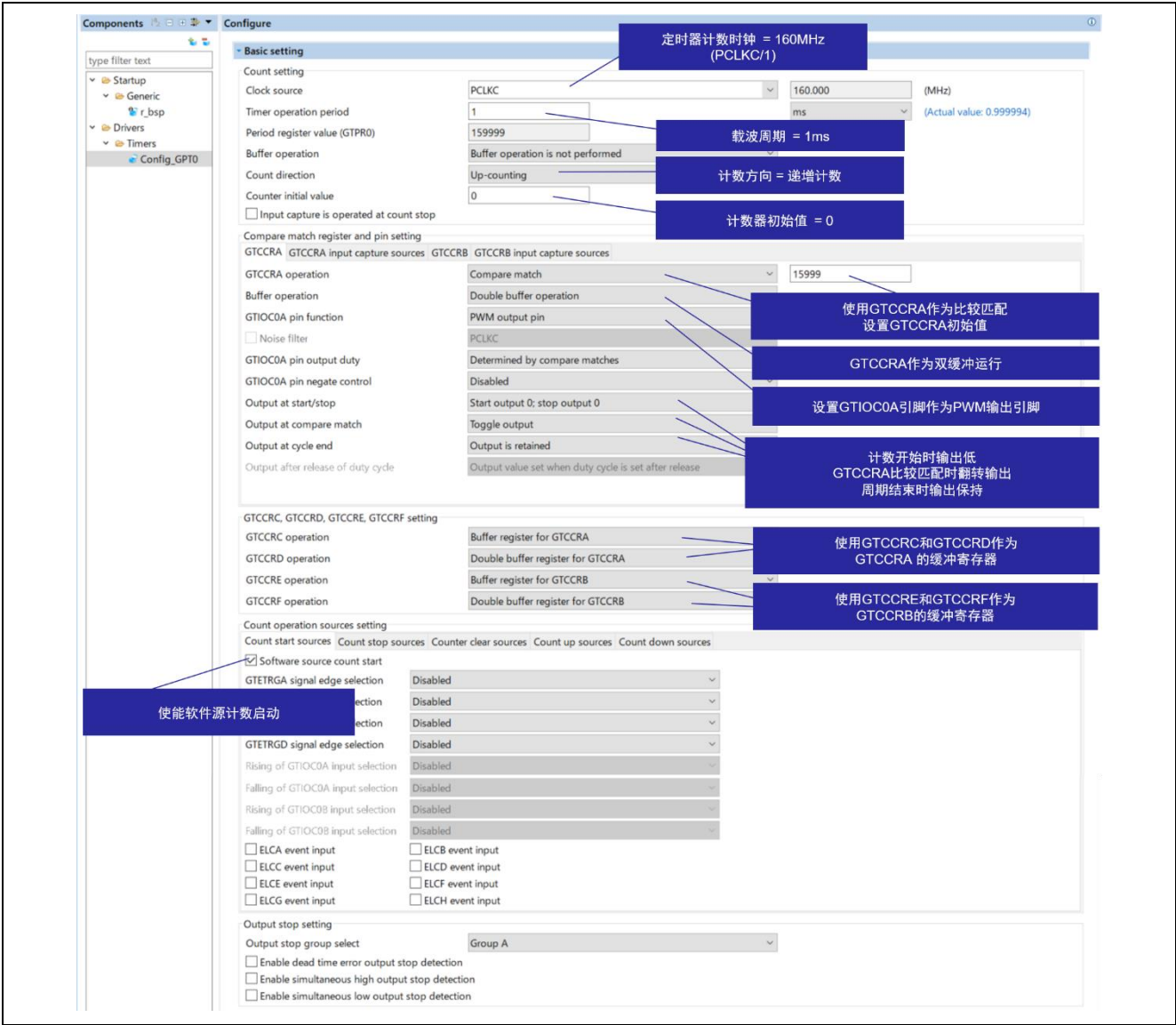


图 4.90 GPT0 设置 (1/2)

使能死区时间自动设置

设置GTDVU值 设置相同的值到GTDVD

使能GTCNT溢出中断

Advance setting

Automatic dead time setting

☒ Automatically set GTCCR80 using GTCCRA0 value and dead time



GTDVU value: 8000 ☐ Enable buffer (GTDBU)

☒ Automatically set the same value of GTDVU to GTDVD

GTDVD value: 0 ☐ Enable buffer (GTDBD)

A/D conversion start request setting

GTADTRA GTADTRB

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA): 100

Buffer operation: Buffer operation is not performed

Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTICIA0) Priority: Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTICIB0) Priority: Level 15 (highest)

☐ Enable dead time error interrupt (GDTED)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTCIV0) Priority: Level 15 (highest)

☐ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

Interrupt and A/D converter start request skipping setting

GTCIV0/GTCIU0 interrupt skipping function: Skipping is not performed

GTCIV0/GTCIU0 interrupt skipping count: Skip count of 1

☐ Link GTICIA0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTICIB0 with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRA A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRB A/D converter start request with GTCIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source: Skipping is not performed

Skip count: Skip count of 1

Extended interrupt skipping counter 2 count source: Skipping is not performed

Skip count: Skip count of 1

Counter 2 initial skip count: Skip count of 1

GTCCRA interrupt extended skipping function: No extended interrupt skipping

GTCCRB interrupt extended skipping function: No extended interrupt skipping

Overflow interrupt extended skipping function: No extended interrupt skipping

Underflow interrupt extended skipping function: No extended interrupt skipping

GTADTRA interrupt extended skipping function: No extended interrupt skipping

GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function: No extended interrupt skipping

GTCCRB buffer transfer extended skipping function: No extended interrupt skipping

GTPR buffer transfer extended skipping function: No extended interrupt skipping

GTADTRA buffer transfer extended skipping function: No extended interrupt skipping

GTADTRB buffer transfer extended skipping function: No extended interrupt skipping

GTDVU buffer transfer extended skipping function: No extended interrupt skipping

GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.91 GPT0 设置 (2/2)

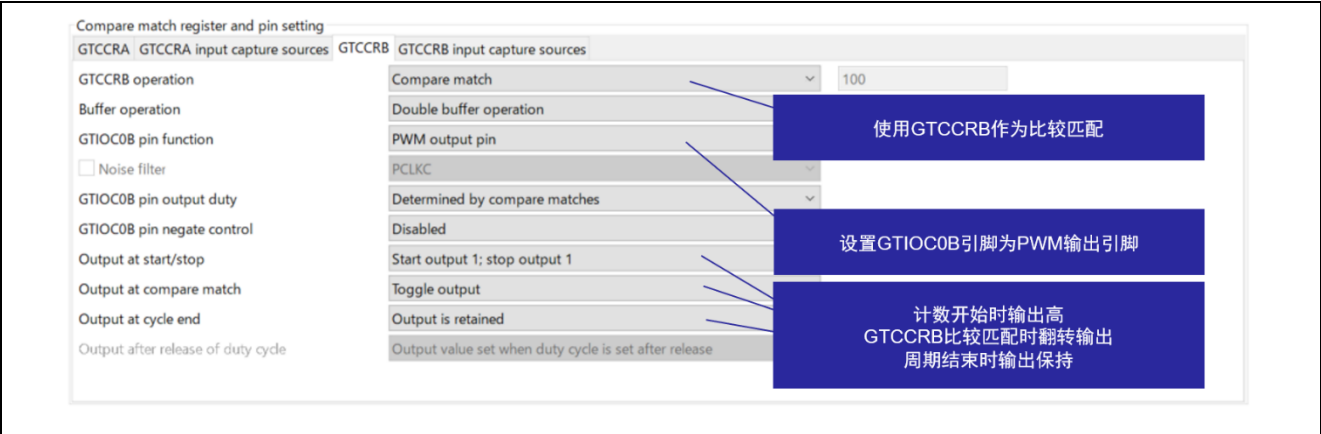


图 4.92 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.10.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

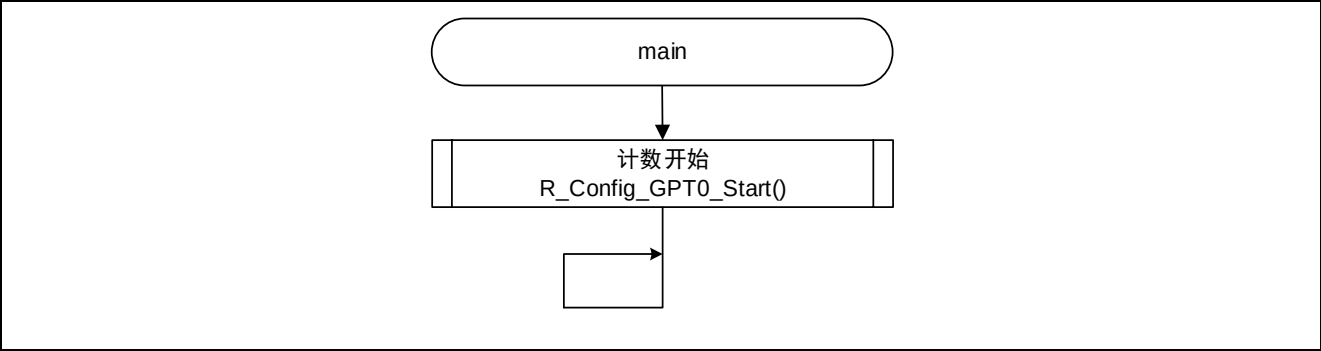


图 4.93 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 用来设置缓冲寄存器的初始值。为了在第一个周期中设置第二个比较匹配寄存器值，在设置缓冲寄存器值后执行强制缓冲传送，然后设置暂存器和比较寄存器的值。此函数在 R_Config_GPT0_Create 函数中调用。

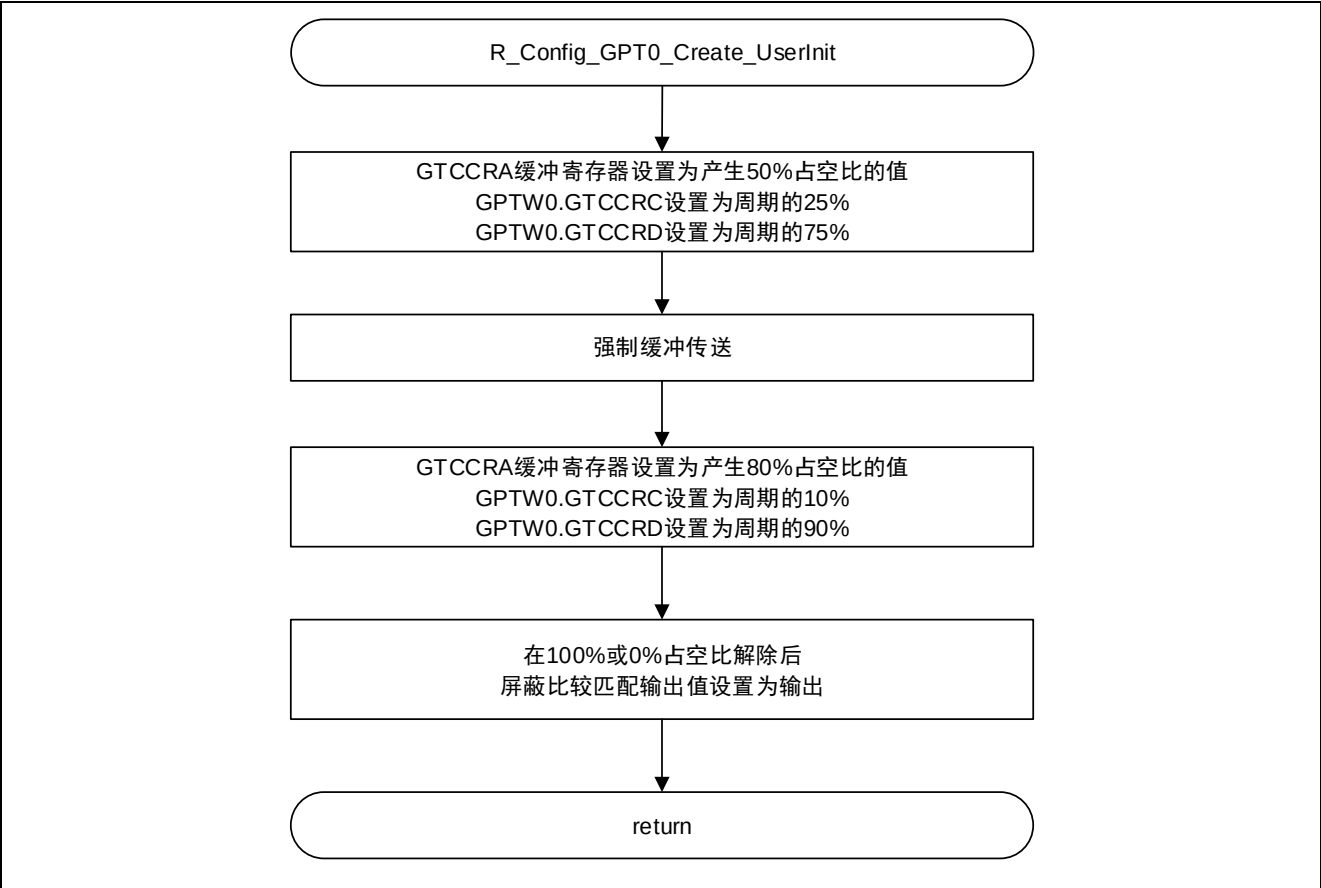


图 4.94 用户初始化函数

GTCIV0 中断处理函数更改缓冲寄存器 GTCCRC 和 GTCCRD 以及 GTUDDTY 寄存器的值。

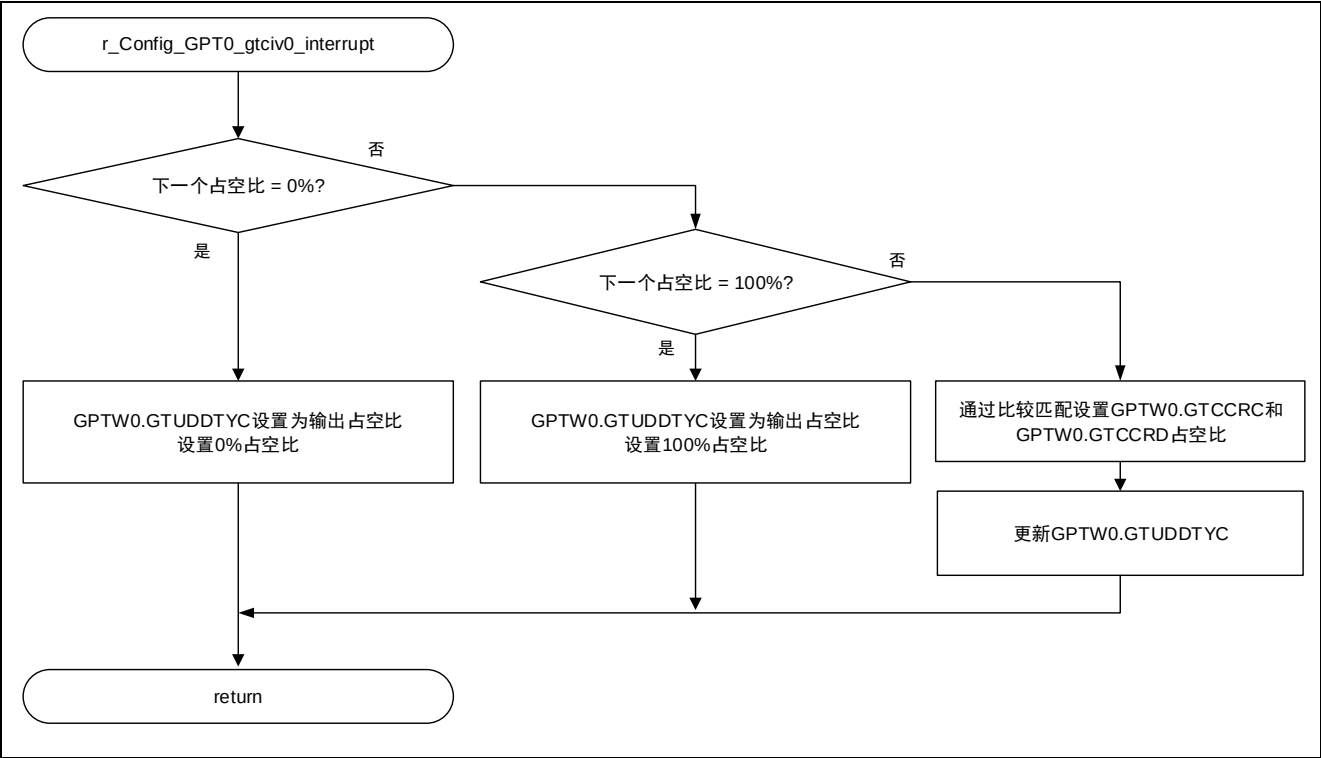


图 4.95 GTCIV0 中断处理函数

4.10.5 使用时的注意事项

4.10.5.1 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间, 并使用比较寄存器 GTCCRA (正相波形寄存器) 的值来更新比较匹配寄存器 GTCCRB (反相波形寄存器) 的值。

比较寄存器 GTCCRA 的值应设置为满足以下限制。如果不满足这些限制, 可能无法获得具有确保固定死区时间的正确输出波形。

- 在递增计数中:
 $GTCCRC < GTCCRD$
 $GTCCRC > GTDVU$
 $GTCCRD < GTPR - GTDVD$
- 在递减计数中:
 $GTCCRC > GTCCRD$
 $GTCCRC < GTPR - GTDVU$
 $GTCCRD > GTDVD$

此外, 如果死区时间没有自动设置, 应设置缓冲寄存器 GTCCRC 和 GTCCRD 满足以下限制。如果不满足这些限制, 则不会发生两次比较匹配并且无法执行脉冲输出。

- 在递增计数中: $0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR$
- 在递减计数中: $GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0$

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F), (3) 在锯齿波单触发脉冲模式下进行死区时间自动设置时和 (4) 在锯齿波单触发脉冲模式下未进行死区时间自动设置时。

4.10.5.2 比较匹配时 100% 占空比输出

若不更改 GTUDDTYC 寄存器, 则无法在比较匹配时将输出占空比切换为 100% 占空比。为了输出 100% 占空比, 设置 GTUDDTYC.OADTY 位为 11b 和 GTUDDTYC.OBDTY 位为 10b。

如果在此示例代码设置中将 GTCCRA 和 GTCCRB 设置为 0, 则无法输出 100% 占空比, 因为在发生 GTCNT 计数器溢出后输出一个时钟周期。

当 GTCCRA 和 GTCCRB 寄存器设置为与 GTPR 相同的值并且同时发生 GTCNT 计数溢出和比较匹配时, 在比较匹配时间的输出设置被优先并且翻转, 禁止 100% 占空比输出。

有关 GPTW 中同时发生 GTCNT 计数器溢出和比较匹配时的波形输出的详情, 请参考 RX66T Group User's Manual: Hardware, 章节 24.2.14 通用 PWM 定时器 I/O 控制寄存器 (GTIOR) 中表 24.4 下的注释。

4.10.5.3 占空比 0% 和 100% 输出期间比较匹配运行

在此示例代码中, 输出占空比 0% 和 100% 是基于 GTUDDTYC.OADTY 和 GTUDDTYC.OBDTY 位的设置值决定的。当占空比设置为 0% 或 100% 时, GPTW 中的比较匹配继续, 并执行中断输出和缓冲传送运行。

此示例代码不使用比较匹配中断, 因此如果你要使用比较匹配中断, 请在占空比 0% 和 100% 输出期间谨慎使用中断。

4.10.5.4 从占空比 0%和 100%切换

如果在设置占空比 0%或 100%后通过比较匹配更改输出设置，则输出值在周期结束时通过 GTIOR.GTIOA[3:2]和 GTUDDTYC.OADTYR 位的值决定。

以下是示例代码的设置。注意，如果 GTUDDTYC.OADTYR 位的初始硬件值为 0，无法执行与此示例代码相同的运行。同样适用于 GTIOC0B 引脚。

- GTIOR.GTIOA[3:2] = 00b: 周期结束时保持输出
- GTUDDTYC.OADTYR = 1: 在解除占空比 0%/100%后，将 GTIOA[3:2]应用于屏蔽的比较匹配输出值。
- GTIOR.GTIOB[3:2] = 00b: 周期结束时保持输出
- GTUDDTYC.OBDTYR = 1: 在解除占空比 0%/100%后，将 GTIOA[3:2]应用于屏蔽的比较匹配输出值。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.6 占空比 0%/100%输出功能。

4.11 锯齿波 PWM 模式 占空比 0%和 100%

- 目标示例代码文件名: r01an5995_rx66t_gptw_sawtooth_pwm_0to100.zip

4.11.1 概要

GPTW 锯齿波 PWM 模式可用于根据 GTCCRA 寄存器比较匹配和 GTUDDTYC 寄存器设置输出占空比为 0%到 100%的 PWM 波形。

此示例代码描述了使用锯齿波 PWM 模式并在占空比 0%和 100%之间交替重复波形输出。

当 GTCNT 计数器溢出发生时, 通过修改 GTUDDTYC 寄存器, 输出的占空比在 0%和 100%之间切换。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用锯齿波 PWM 模式
- 使用通道 0
- 初始输出值 = 低
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数方向 = 递增计数
 - 计数器初始值 = 0
- 使用 GTCCRA 作为占空比输出比较匹配
 - 设置 GTIOC0A 引脚为 PWM 输出引脚
 - GTCCRA 作为比较匹配
 - 计数开始时输出低
 - GTCCRA 比较匹配时输出高
 - 周期结束时输出低
 - 计数开始时强制输出 0%占空比*
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器溢出中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.97

在 Smart Configurator 中设置。
设置方法请参考章节 4.11.3。
标记*的项目除外。

此示例代码的锯齿波 PWM 模式输出如下所示。

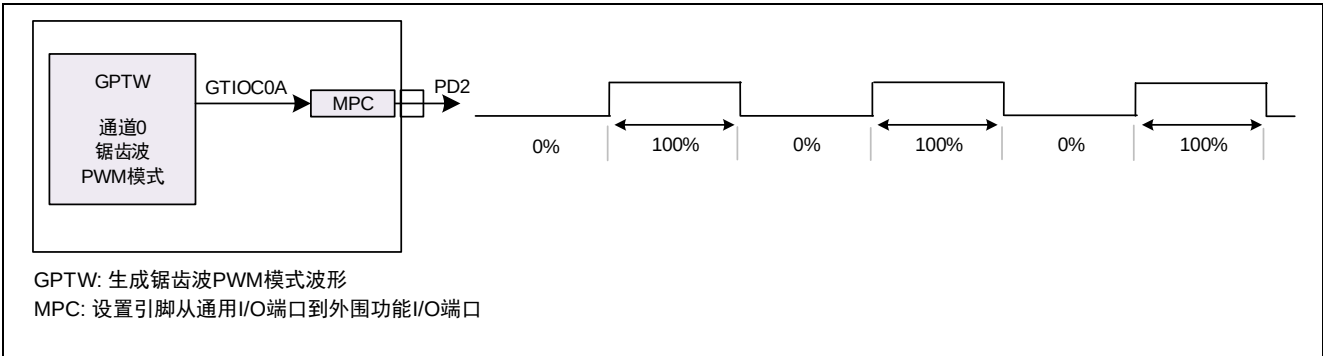
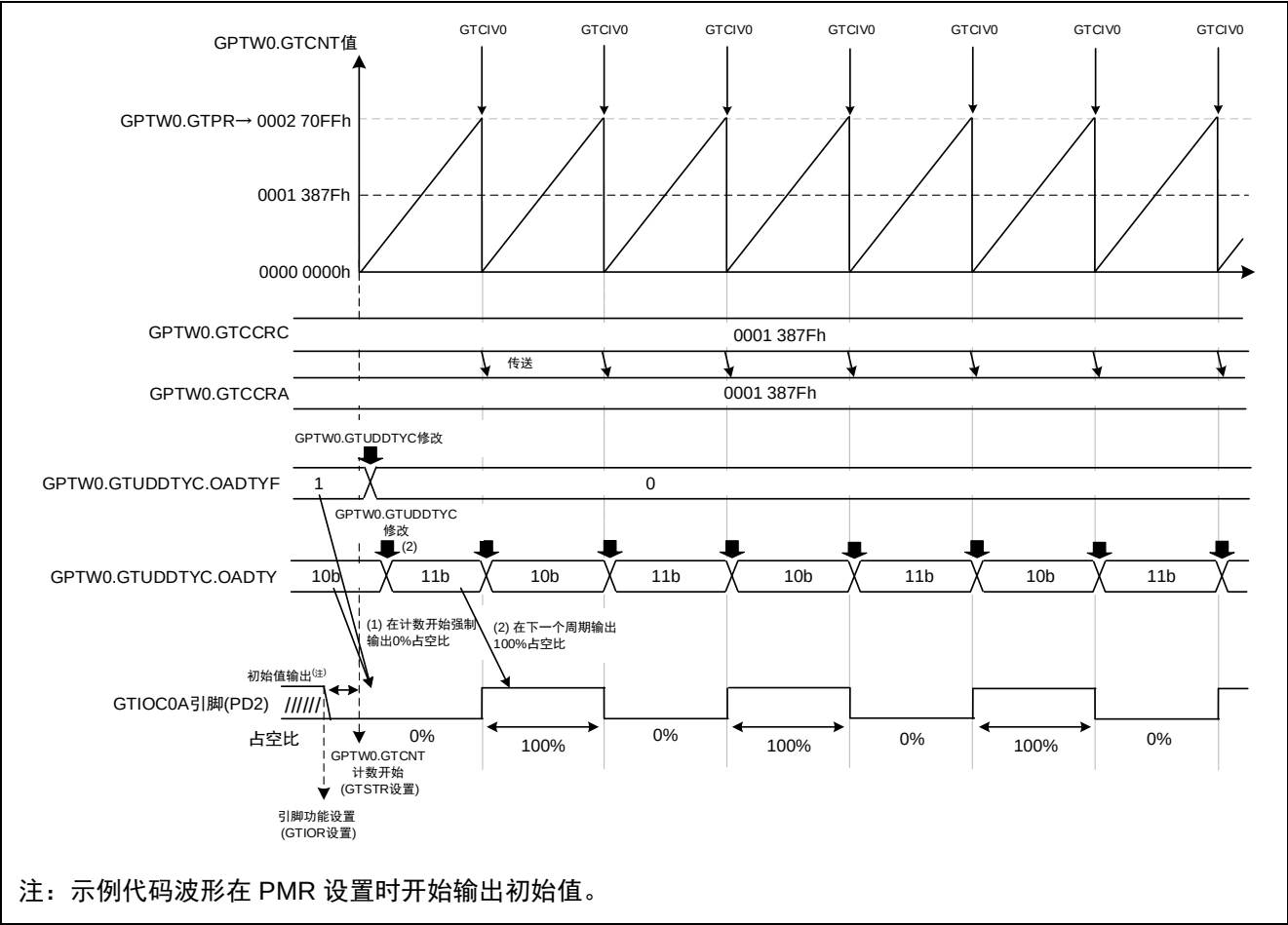


图 4.96 锯齿波 PWM 模式输出

4.11.2 运行说明

示例代码的运行如图 4.97 所示。在 GTCNT 计数器溢出中断（GTCIV0）中通过修改 GTUDDTYC 寄存器的值，每个周期的占空比在 0%和 100%之间交替。

- 计数开始后 0%占空比输出（下图中（1））
在 GTUDDTYC.OADTYF 位为 1b 且计数停止时通过设置 GTUDDTYC.OADTY 位为 10b，计数开始后立即输出低电平。即使发生 GTCCRA 比较匹配，波形也不会改变。
- 第 2 个周期 0%占空比输出（下图中（2））
在计数开始后发生第一个 GTCNT 计数器溢出之前，通过设置 GTUDDTYC.OADTY 位为 11b，从第二个周期输出高电平。即使发生 GTCCRA 比较匹配，波形也不会改变。



注：示例代码波形在 PMR 设置时开始输出初始值。

图 4.97 示例代码运行

4.11.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考 4.1.4 添加组件。

表 4.14 添加组件

项目	描述
组件	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	锯齿波 PWM 模式
资源	GPT0

定时计数器时钟 = 160MHz (PCLKC/1)

载波周期 = 1ms

计数方向 = 递增计数

计数器初始值 = 0

使用GTCRA作为比较匹配 设置 GTCRA初始值

GTCRA作为单缓冲运行

设置GTIOC0A引脚作为PWM输出引脚 输出占空比由比较匹配决定^(注)

计数开始时输出低 GTCRA比较匹配时翻转输出 周期结束时输出低

使用GTCRC作为GTCRA的缓冲寄存器

使能软件源计数启动

注：在计数开始时设置为强制输出占空比后，输出占空比需要设置为 0%。
在 R_Config_GPT0_Create_UserInit 函数中设置 0%占空比输出。

图 4.98 GPT0 设置 (1/2)

Advance setting

A/D conversion start request setting

GTADTRA GTADTRB

☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)

☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)

Compare match value (GTADTRA) 100

Buffer operation Buffer operation is not performed

Buffer transfer timing setting No transfer

A/D converter start request signal monitor setting

☐ Enable S12AD0 monitor Monitor signal select GTADTRA compare match during up-counting

☐ Enable S12AD1 monitor Monitor signal select GTADTRA compare match during up-counting

Interrupt setting

☐ Enable GTCCRA input capture/compare match interrupt (GTICIA0) Priority Level 15 (highest)

☐ Enable GTCCRB input capture/compare match interrupt (GTICIB0) Priority Level 15 (highest)

☐ Enable GTCCRC compare match interrupt (GTICIC0) Priority Level 15 (highest)

☐ Enable GTCCRD compare match interrupt (GTICID0) Priority Level 15 (highest)

☐ Enable GTCCRE compare match interrupt (GTICIE0) Priority Level 15 (highest)

☐ Enable GTCCRF compare match interrupt (GTICIF0) Priority Level 15 (highest)

☒ Enable GTCNT overflow (GTPR compare match) interrupt (GTICIV0) Priority Level 15 (highest)

☐ Enable GTCNT underflow interrupt (GTICIU0) Priority Level 15 (highest)

Interrupt skipping setting

Interrupt skipping on Skipping is not performed

GTICIV0/GTCIU0 interrupt skipping count Skip count of 1

☐ Link GTICIA0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTICIB0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTICIC0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTICID0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTICIE0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTICIF0 with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRA A/D converter start request with GTICIV0/GTCIU0 interrupt skipping function

☐ Link GTADTRB A/D converter start request with GTICIV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting

Extended interrupt skipping counter 1 count source Skipping is not performed

Skip count Skip count of 1

Extended interrupt skipping counter 2 count source Skipping is not performed

Skip count Skip count of 1

Counter 2 initial skip count Skip count of 1

GTCCRA interrupt extended skipping function No extended interrupt skipping

GTCCRB interrupt extended skipping function No extended interrupt skipping

GTCCRC interrupt extended skipping function No extended interrupt skipping

GTCCRD interrupt extended skipping function No extended interrupt skipping

GTCCRE interrupt extended skipping function No extended interrupt skipping

GTCCRF interrupt extended skipping function No extended interrupt skipping

Overflow interrupt extended skipping function No extended interrupt skipping

Underflow interrupt extended skipping function No extended interrupt skipping

GTADTRA interrupt extended skipping function No extended interrupt skipping

GTADTRB interrupt extended skipping function No extended interrupt skipping

Extended buffer transfer skipping setting

GTCCRA buffer transfer extended skipping function No extended interrupt skipping

GTCCRB buffer transfer extended skipping function No extended interrupt skipping

GTPR buffer transfer extended skipping function No extended interrupt skipping

GTADTRA buffer transfer extended skipping function No extended interrupt skipping

GTADTRB buffer transfer extended skipping function No extended interrupt skipping

HRPWM setting

High Resolution PWM setting

☐ Enable output high resolution PWM waveform

☐ Enable operation of rising and falling edge adjustment circuit

GTIOC0A pin rising edge delay select Apply delay of 0/32 times PCLKC period

GTIOC0A pin falling edge delay select Apply delay of 0/32 times PCLKC period

GTIOC0B pin rising edge delay select Apply delay of 0/32 times PCLKC period

GTIOC0B pin falling edge delay select Apply delay of 0/32 times PCLKC period

Overview Board Clocks System Components Pins Interrupts

图 4.99 GPT0 设置 (2/2)

4.11.4 流程图

以下流程图显示了 Smart Configurator 生成代码后添加的函数的处理。

在主函数中，计数开始并且下一个周期的输出设置为 100% 占空比。

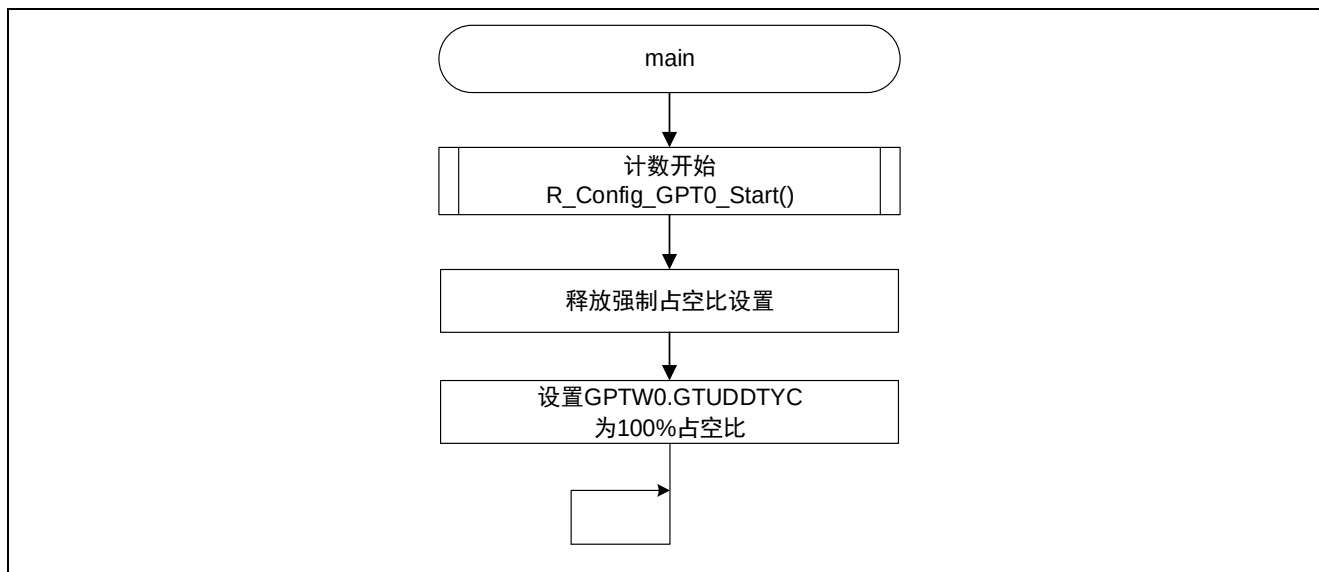


图 4.100 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 在计数启动时设置强制输出占空比并将占空比设置为 0%。此函数在 R_Config_GPT0_Create 函数中调用。

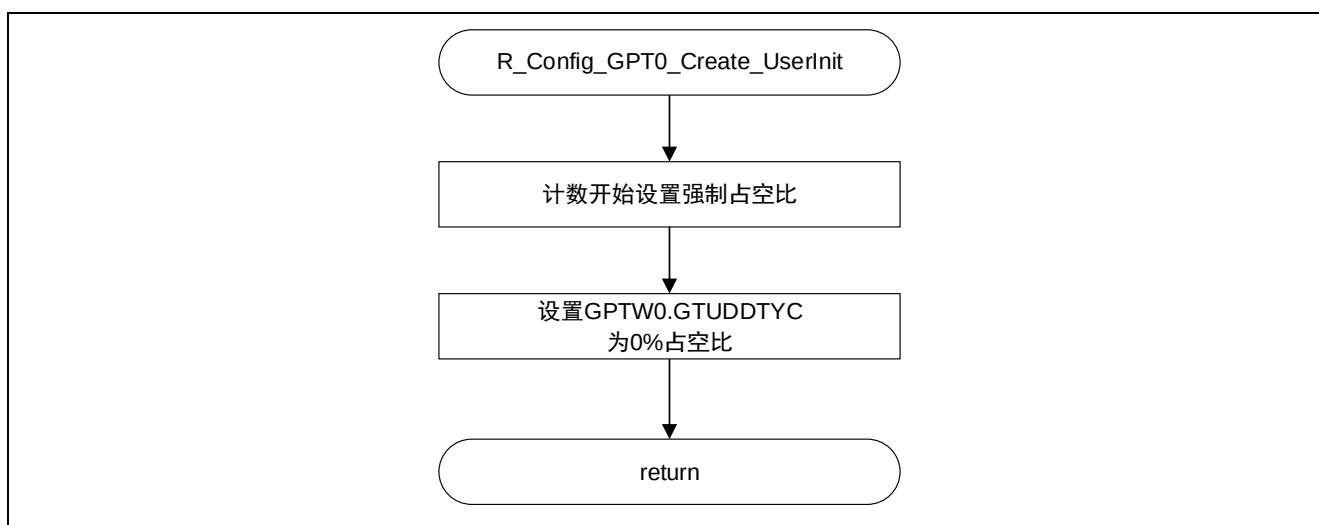


图 4.101 用户初始化函数

GTCIV0 中断处理函数改变 GTUDDTYC 寄存器的值。

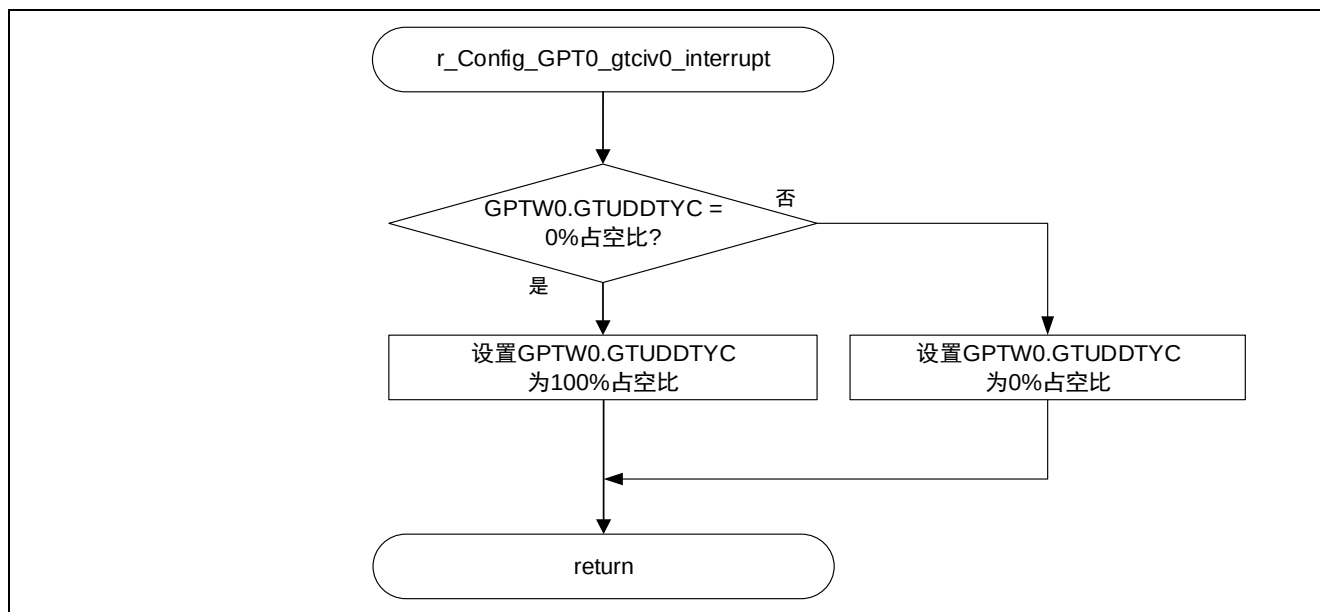


图 4.102 GTCIV0 中断处理函数

4.11.5 相关运行

4.11.5.1 在初始值低时保持 0%后更改占空比到 100%

下面显示了在计数开始后并输出几个周期 0%后将占空比切换到 100%的运行示例。

低电平在计数开始后立即输出，当 GTUDDTYC.OADTYF 位为 1b 且计数停止时通过设置 GTUDDTYC.OADTY 位为 10b，输出占空比切换为 0%（下图中（1））。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，输出变为高电平，占空比在下一个周期变为 100%（下图中（2））。

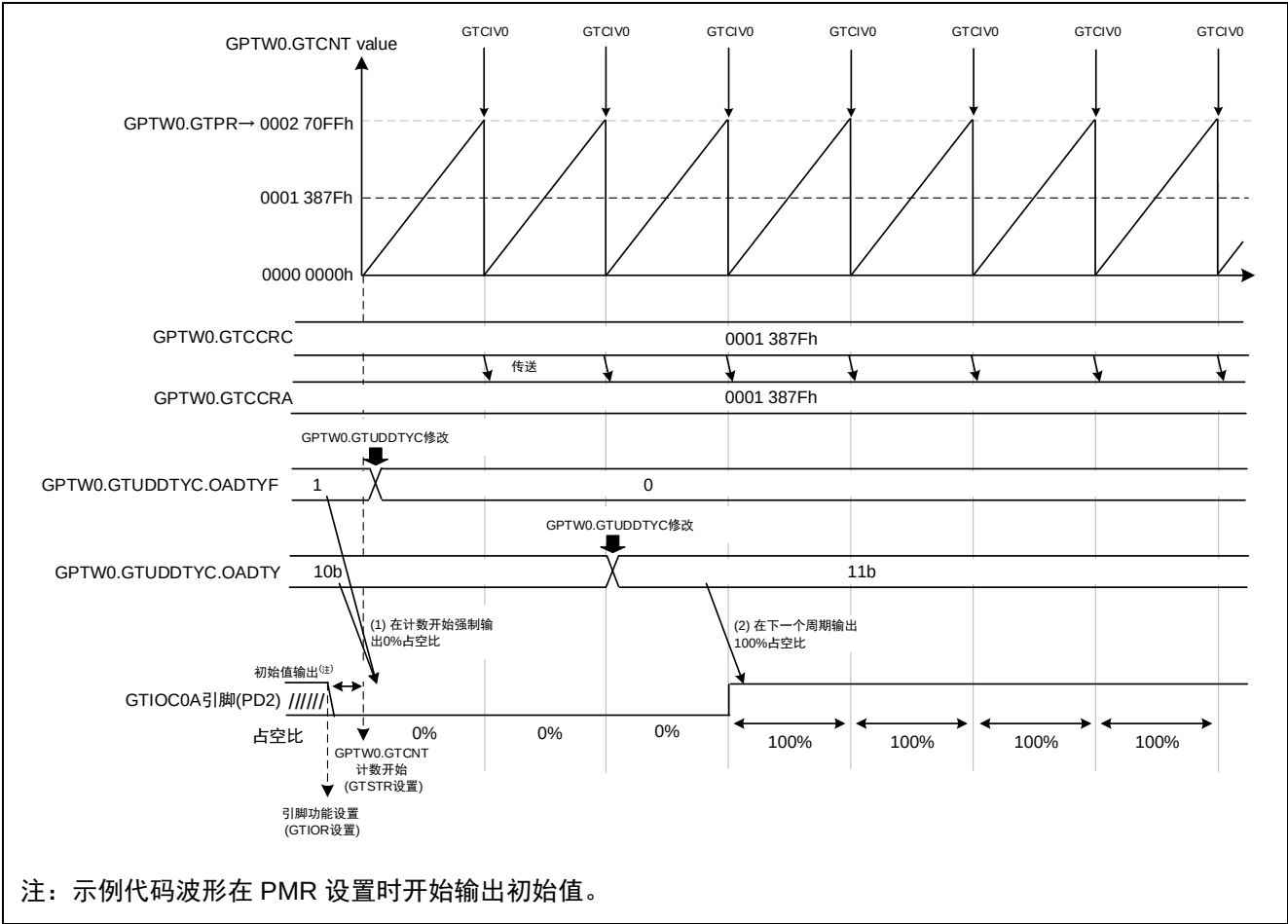


图 4.103 在保持 0%后更改占空比到 100%

4.11.5.2 在初始值低时保持 100%后更改占空比到 0%

下面显示了在计数开始后并输出几个周期 100%后将占空比切换到 0%的运行示例。

高电平在计数开始后立即输出，当 GTUDDTYC.OADTYF 位为 1b 且计数停止时通过设置 GTUDDTYC.OADTY 位为 11b，输出占空比切换到 100%（下图中（1））。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 10b，输出变为低电平，占空比在下一个周期变为 0%（下图中（2））。

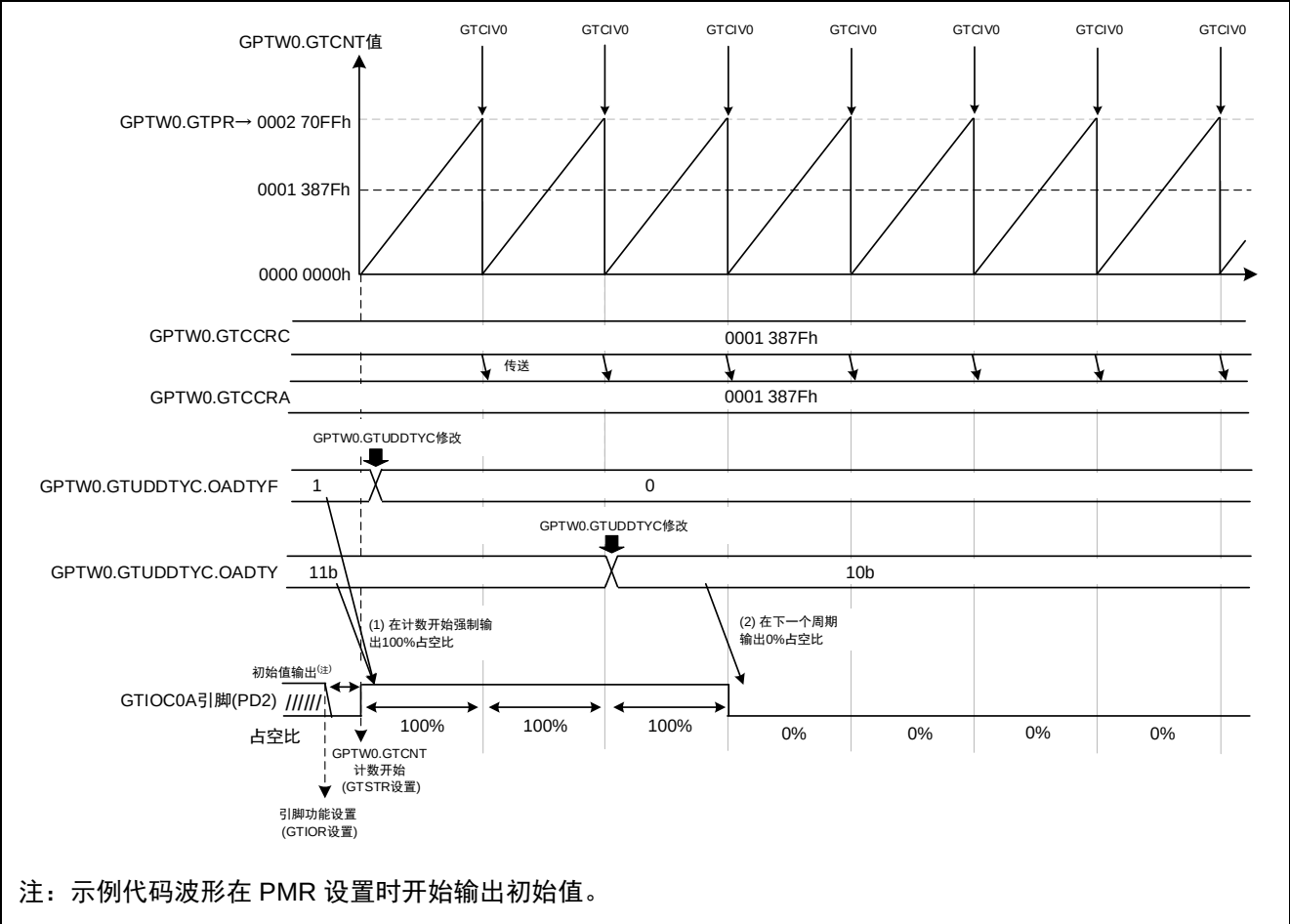


图 4.104 在保持 100%后更改占空比到 0%

4.11.5.3 在初始值高时保持 0%后更改占空比到 100%

下面显示了在计数开始后并输出几个周期 0%后将占空比切换到 100%的运行示例。

低电平在计数开始后立即输出，当 GTUDDTYC.OADTYF 位为 1b 且计数停止时通过设置 GTUDDTYC.OADTY 位为 10b，输出占空比切换为 0%（下图中（1））。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，输出变为高电平，占空比在下一个周期变为 100%（下图中（2））。

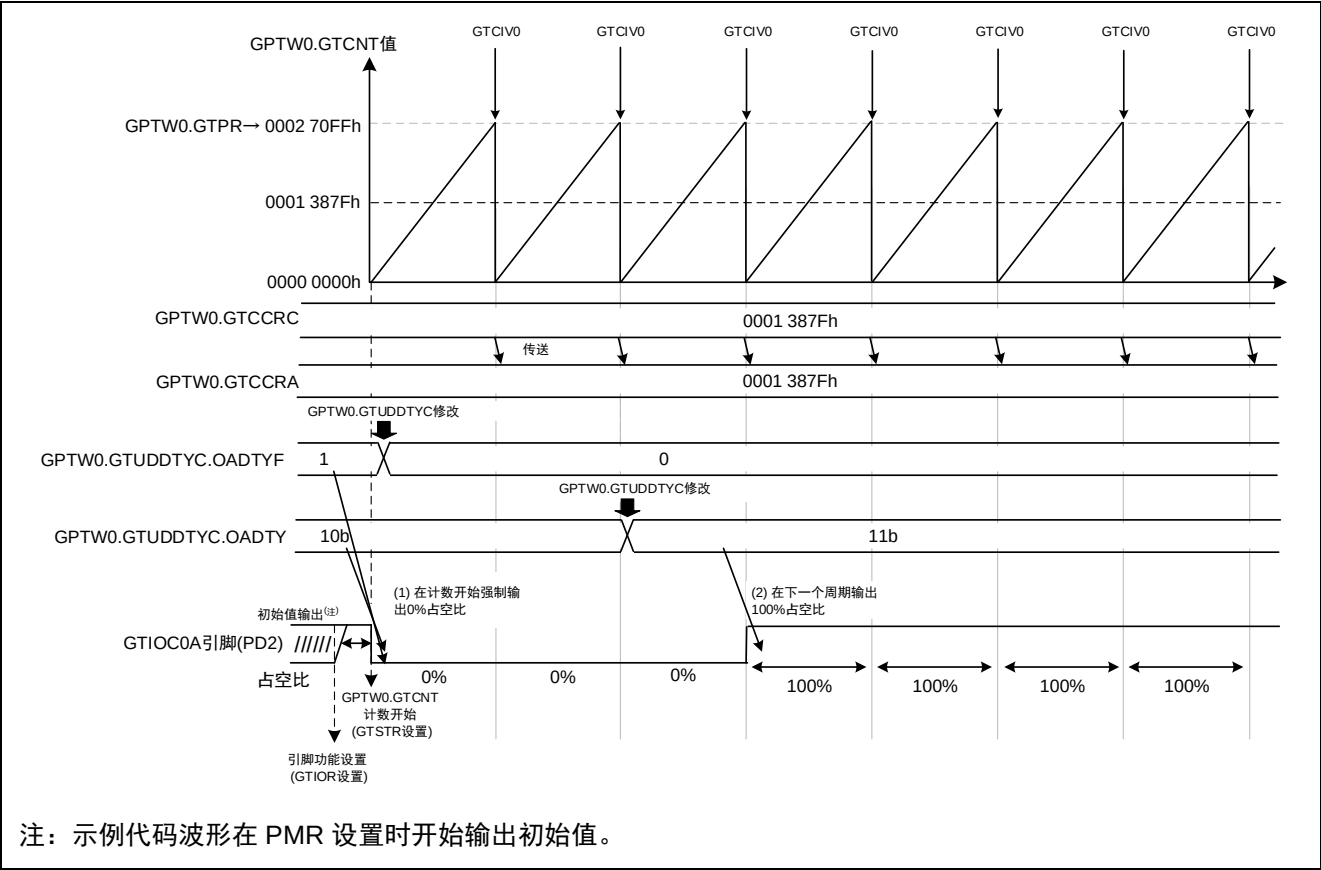


图 4.105 在保持 0%后更改占空比到 100%

要将初始值设置为高，将“Output at start/stop”设置为“stop output 1”。

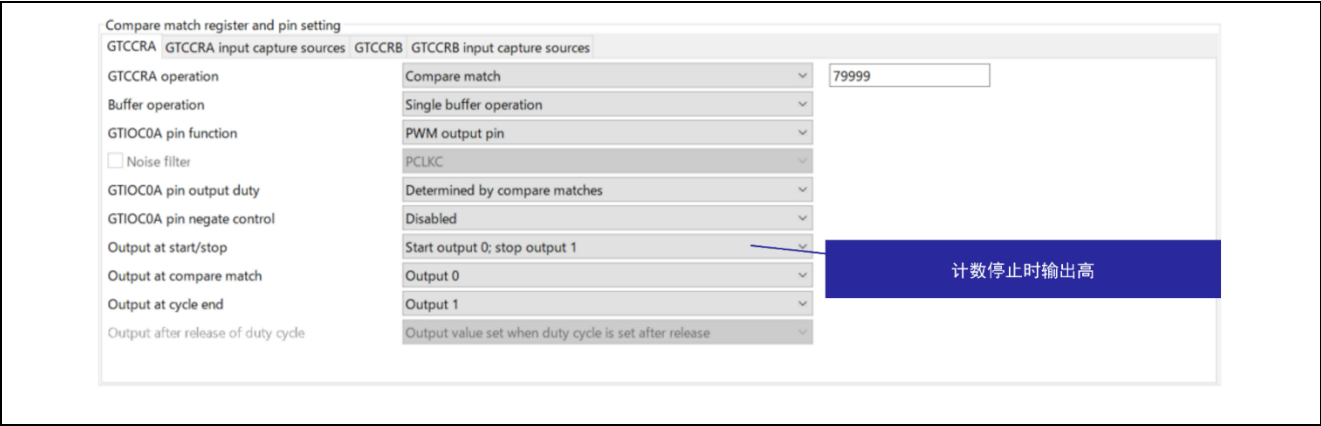


图 4.106 初始值高的 Smart Configurator 设置

4.11.5.4 在初始值高时保持 100%后更改占空比到 0%

下面显示了在计数开始后并输出几个周期 100%后将占空比切换到 0%的运行示例。

高电平在计数开始后立即输出，当 GTUDDTYC.OADTYF 位为 1b 且计数停止时通过设置 GTUDDTYC.OADTY 位为 11b，输出占空比切换到 100%（下图中（1））。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 10b，输出变为低电平，占空比在下一个周期变为 0%（下图中（2））。

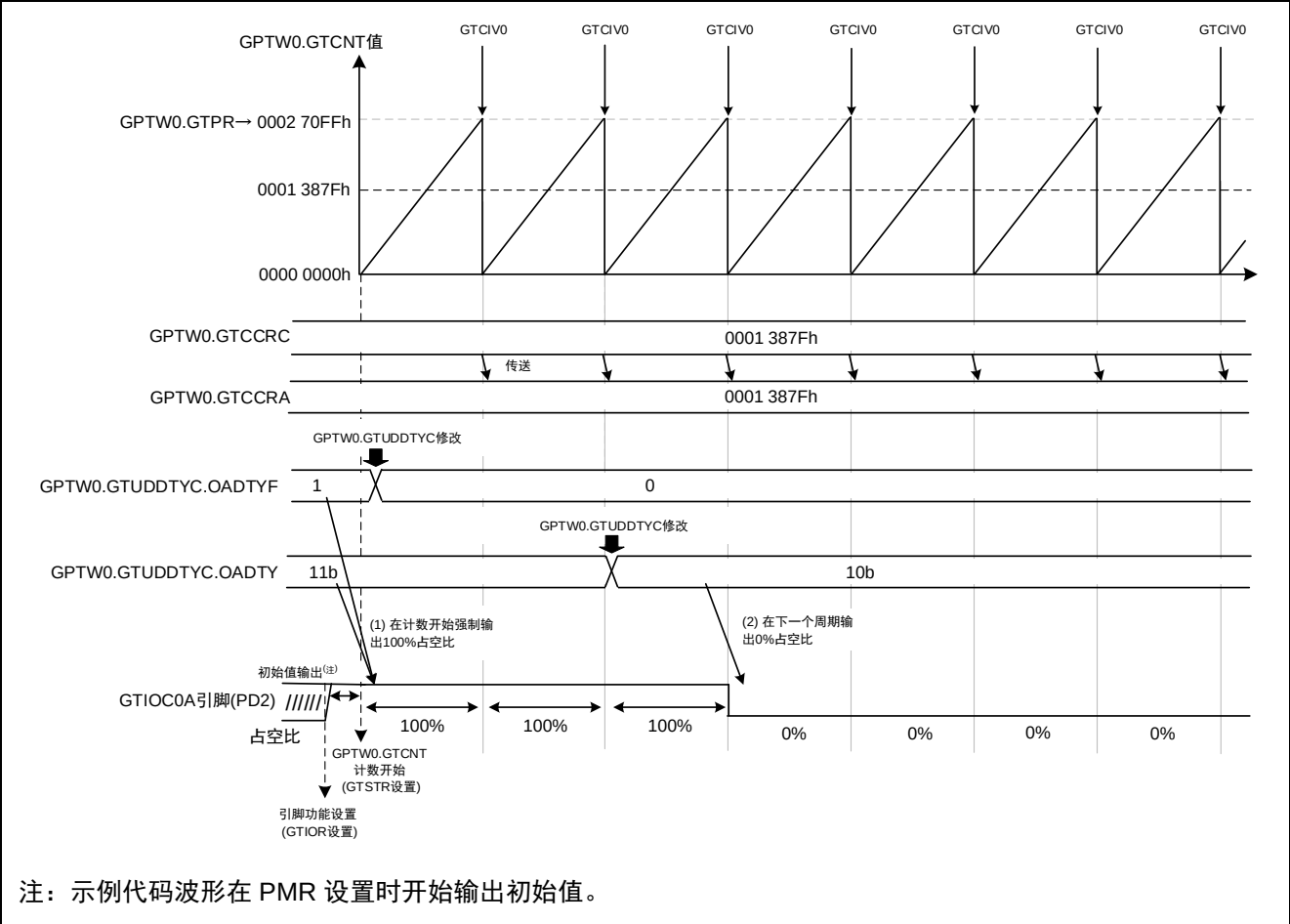


图 4.107 保持 100%后将占空比改变为 0%

要将初始值设置为高，请将“Output at start/stop”设置为“start output 1”。有关详细信息，请参考图 4.106。

4.11.5.5 在保持 50%后更改占空比到 100%

下面显示了当 GTUDDTYC.OADTY 位改变，占空比在输出几个周期 50%后切换到 100%的运行示例。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，占空比在下一个周期输出 100%（下图中（1））。

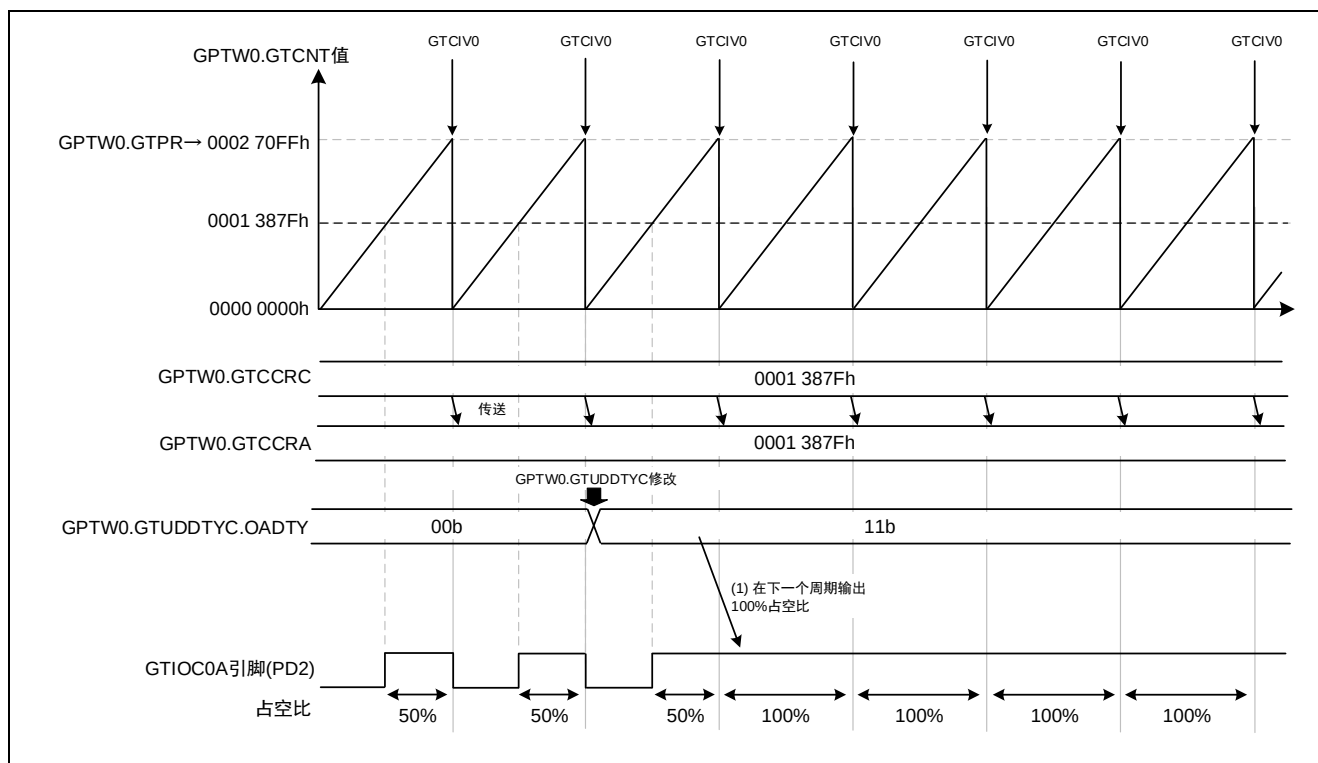


图 4.108 在保持 50%后更改占空比到 100%

在计数运行期间不改变 GTUDDTYC.OADTY 位的情况下，无法通过比较匹配将占空比切换到 100%。

4.11.5.6 在保持 50%后更改占空比到 0%

下面显示了当 GTUDDTYC.OADTY 位改变，占空比在输出几个周期 50%后切换到 0%的运行示例。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 10b，占空比在下一个周期输出 0%（下图中（1））。

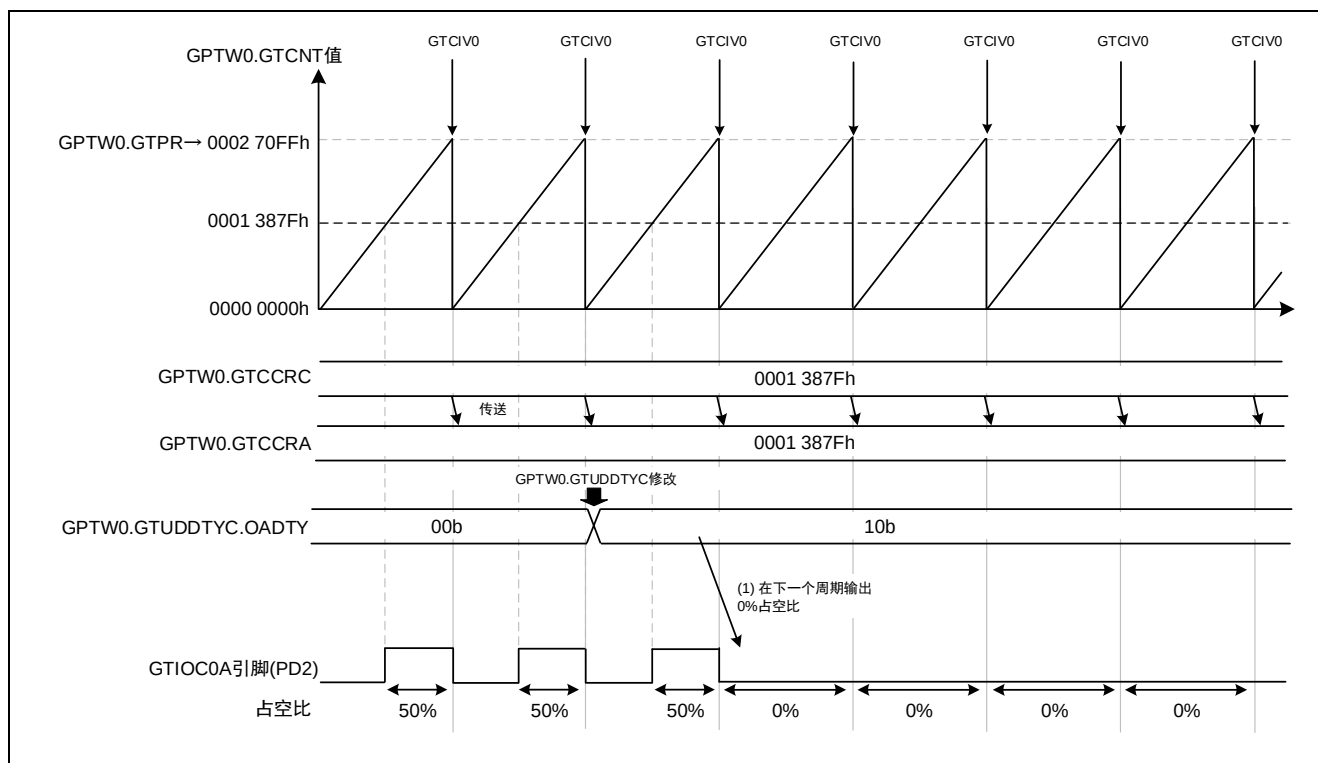


图 4.109 在保持 50%后更改占空比到 0%

4.11.6 使用时的注意事项

4.11.6.1 比较匹配运行期间 GTCCRm 寄存器的设置 (m = A 到 F)

将比较寄存器 GTCCRA 和 GTCCRB 设置为高于 0000 0001h 但小于 GTPR 寄存器的设置值。如果设置为 0000 0000h 或与 GTPR 寄存器的值相同，则仅当比较匹配寄存器值为 0000 0000h 或比较寄存器设置为与 GTPR 寄存器相同的值时，才会在周期内发生比较匹配。如果将比较寄存器的值设置为超过 GTPR 寄存器的设置值，则不会发生比较匹配。

以下是在比较寄存器中设置超过 GTPR 寄存器设置值时的运行示例。由于未发生比较匹配，GTIOC0A 引脚保持低输出（下图中（1））。

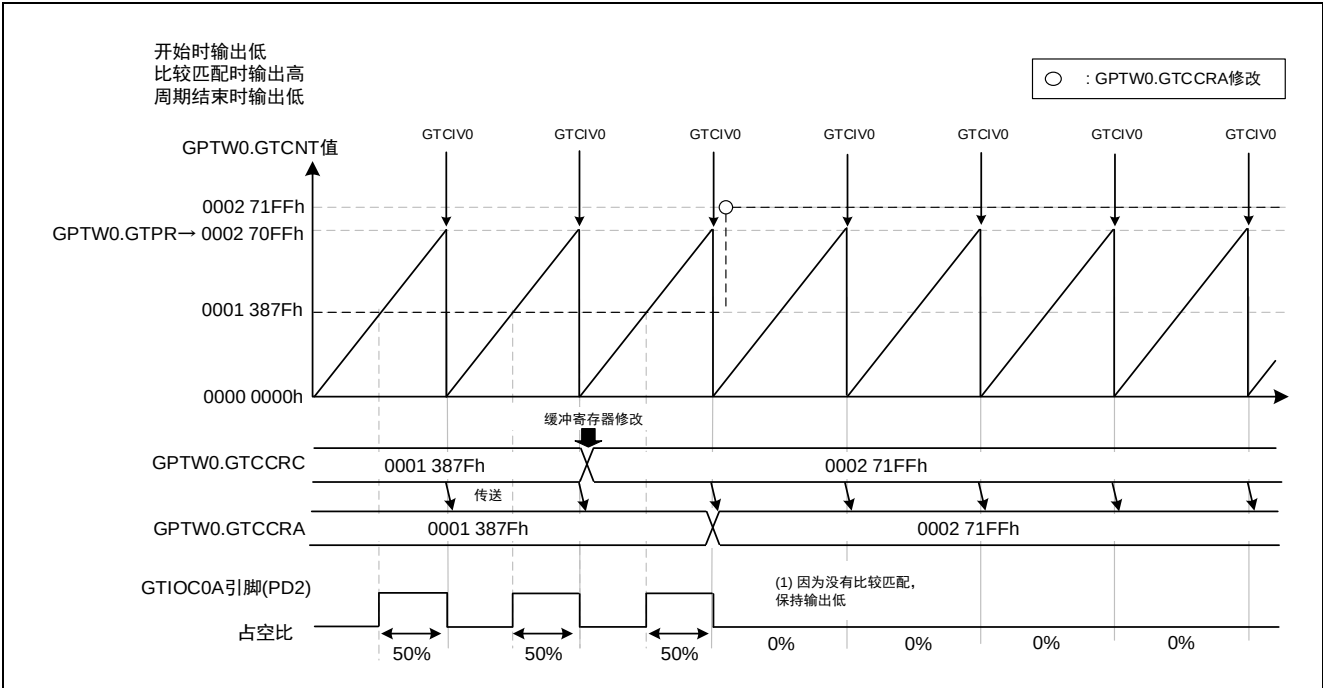


图 4.110 设置超过 GTPR 寄存器设置值时的运行

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 GTCCRm 寄存器在比较匹配运行期间的设置 (m = A 到 F)，(5) 在锯齿波 PWM 模式下。

4.11.6.2 计数开始时 GTUDDTYC.OADTY 设置的反映

在此示例代码中，计数运行占空比由在计数停止时设置的 GTUDDTYC.OADTYF 和 GTUDDTYC.OADTY 位确定。

在计数停止且 GTUDDTYC.OADTYF 位为 0b 时，如果 GTUDDTYC.OADTY 位的值被改变，计数开始时改变的输出占空比设置不会被反映。为了反映计数开始时的设置，有必要在计数停止且 GTUDDTYC.OADTYF 位为 1b 时改变 GTUDDTYC.OADTY 位的值，然后开始计数。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.6 占空比 0%/100% 输出功能。

4.11.6.3 使用 Smart Configurator 在计数开始时占空比的反映

如图 4.111，当使用 Smart Configurator 并且引脚输出占空比设置为 0%或 100%时，由于 GTUDDTYC.OADTYF 位为 0b，占空比不会在计数开始时反映。为了在计数开始时反映占空比，用户必须创建一个代码在计数停止且 GTUDDTYC.OADTYF 位为 1b 时来设置 GTUDDTYC.OADTY 位为 10b 或 11b。

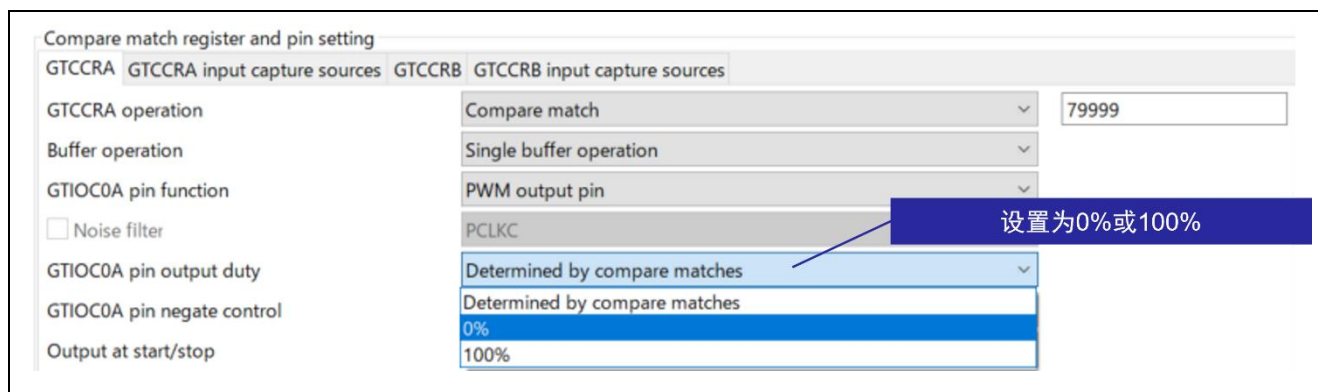


图 4.111 设置引脚输出占空比为 0%或 100%

有关 Smart Configurator 设置的详细信息，请参考 4.11.3 Smart Configurator 设置，有关用户生成代码的示例，请参考 4.11.4 流程图。

4.11.6.4 比较匹配时 100%占空比输出

在不改变 GTUDDTYC 寄存器的情况下，输出占空比无法在比较匹配时切换到 100%。要输出 100%占空比，设置 GTUDDTYC.OADTY 位为 11b。

如果在此示例代码中将 GTCCRA 寄存器设置为 0，则无法输出 100%占空比，因为在 GTCNT 计数器溢出发生后，输出一个时钟周期的低电平，然后输出高电平。

MTU 可以输出 100%占空比，因为占空比寄存器和周期寄存器设置为相同的值，当计数器清除和比较匹配同时发生时，波形不会改变。GPTW 无法以相同的方式输出 100%占空比，因为即使 GTCNT 计数器溢出和比较匹配同时发生，波形也会发生变化。

有关 GPTW 中 GTCNT 计数器溢出和比较匹配同时发生时的波形输出的详细信息，详情参考 RX66T Group User's Manual: Hardware, 章节 24.2.14 通用 PWM 定时器 I/O 控制寄存器 (GTIOR) 中表 24.4 下的注释。

4.11.6.5 占空比 0%和 100%输出期间的比较匹配运行

在此示例代码中，输出占空比 0%和 100%是根据 GTUDDTYC.OADTY 位设置的值确定的。当占空比设置为 0%或 100%时，GPTW 中的比较匹配运行将继续，并执行中断输出和缓冲传送运行。

此示例代码不使用比较匹配中断，因此，如果使用比较匹配中断，请在占空比 0%和 100%输出期间谨慎使用中断。

4.12 三角波 PWM 模式 1 占空比 0%到 100%

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm_50to100_dt.zip

4.12.1 概要

GPTW 三角波 PWM 模式 1 可用于根据 GTCCRA 寄存器的比较匹配和 GTUDDTYC 寄存器设置输出占空比 0%到 100%的 PWM 波形。

此示例代码描述了在三角波 PWM 模式 1 下使用死区时间自动设置功能并重复输出以下波形，包括占空比 0%和 100%。

- GTIOC0A 引脚高带宽切换: 50% → 80% → 100% → 80% → 50% → 0% → ...
- GTIOC0B 引脚低带宽切换: 60% → 90% → 100% → 90% → 60% → 0% → ...

基本操作是当使用缓冲寄存器 GTCCRC 在 GTCNT 计数器下溢发生时，通过将缓冲寄存器的值传送到 GTCCRA 来改变占空比。当在占空比 0%和 100%之间切换时，GTCNT 计数器下溢发生时执行修改 GTUDDTYC 寄存器的过程。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用三角波 PWM 模式 1
- 使用通道 0
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 使用 GTIOC0A 引脚作为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出低
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时保持输出
 - 在解除占空比 0%/100%后，输出被屏蔽的比较匹配输出值*
- 使用 GTCCRB 作为占空比输出比较匹配
 - 使用 GTIOC0B 引脚作为 PWM 输出引脚
 - 使用 GTCCRB 作为比较匹配
 - 计数开始时输出高
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时保持输出
 - 在解除占空比 0%/100%后，输出被屏蔽的比较匹配输出值*
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器下溢中断时改变占空比
 - 关于占空比改变的时间的详细信息，请参考图 4.114

在 Smart Configurator 中设置。
设置方法请参考章节 4.12.3。
标记*的项目除外。

此示例代码的三角波 PWM 模式 1 输出如下所示。

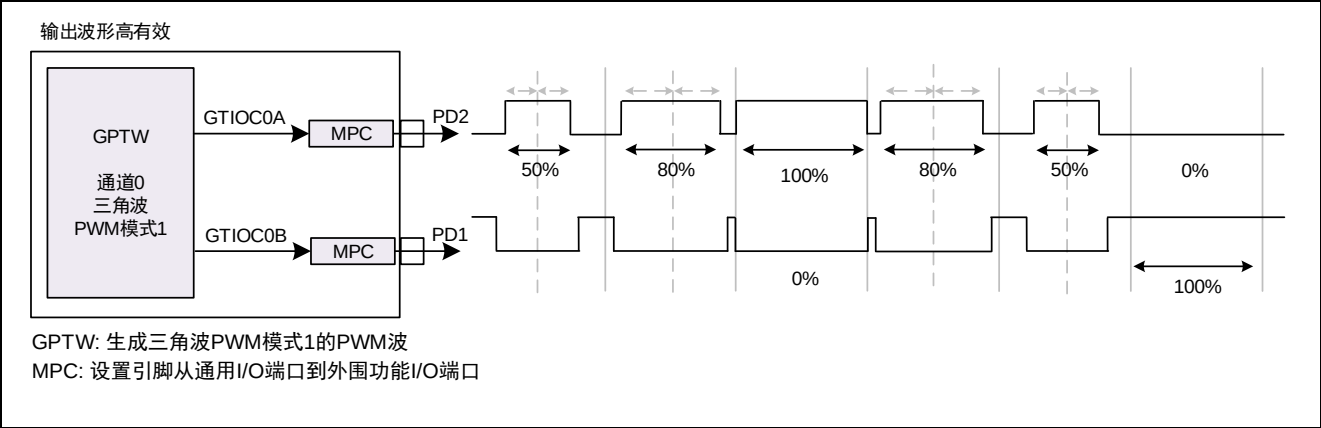


图 4.112 三角波 PWM 模式 1 输出

4.12.2 运行说明

示例代码的运行如图 4.114 所示。基本操作是通过在 GTCNT 计数器下溢中断（GTCIU0）中修改缓冲寄存器 GTCCRC 的值，并在 GTCNT 计数器下溢发生时将缓冲寄存器的值传送到 GTCCRA，从而改变每个周期的占空比的设置。

此外，因为使用了死区时间自动设置功能，GTCCRB 寄存器根据 GTCCRA 更新自动设置。GTDVU 和 GTDVD 设置相同的值。

每个周期的占空比在递增计数期间和递减计数期间产生 1/2 占空比。在递增计数和递减计数期间产生的占空比如图 4.113 所示。

- 100%占空比输出（图 4.114 中（1））

通过设置 GTUDDTYC.OADTY 位为 11b，GTIOC0A 引脚从下一次 GTCNT 计数器下溢（波谷）发生时输出高电平，并且即使发生 GTCCRA 比较匹配，波形也不改变。

通过设置 GTUDDTYC.OBDTY 位为 10b，GTIOC0B 引脚从下一次 GTCNT 计数器下溢（波谷）发生时输出低电平，并且即使发生 GTCCRB 比较匹配，波形也不改变。在此示例代码中，此位被设置为 10b，使得 GTIOC0B 引脚在一个周期内输出低电平。

- 0%占空比输出（图 4.114 中（2））

通过设置 GTUDDTYC.OADTY 位为 10b，GTIOC0A 引脚从下一次 GTCNT 计数器下溢（波谷）发生时输出低电平，并且即使发生 GTCCRA 比较匹配，波形也不改变。

通过设置 GTUDDTYC.OBDTY 位为 11b，GTIOC0B 引脚从下一次 GTCNT 计数器下溢（波谷）发生时输出高电平，并且即使发生 GTCCRB 比较匹配，波形也不改变。在此示例代码中，此位被设置为 11b，使得 GTIOC0B 引脚在一个周期内输出高电平。

- 从占空比 100%和 0%切换（图 4.114 中（3））

通过设置 GTUDDTYC.OADTY 位为 00b，GTIOC0A 引脚根据 GTCCRA 比较匹配从下一次 GTCNT 计数器下溢（波谷）发生时输出占空比。在占空比 100%和 0%解除后的输出由 GTIOR.GTIOA[3:2]和 GTUDDTYC.OADTYR 位决定。在此示例代码中，输出被屏蔽的比较匹配输出值低。

- 通过设置 GTUDDTYC.OBDTY 位为 00b，GTIOC0B 引脚根据 GTCCRB 比较匹配从下一次 GTCNT 计数器下溢发生时输出占空比。在占空比 100%和 0%解除后的输出由 GTIOR.GTIOB[3:2]和 GTUDDTYC.OBDTYR 位决定。在此示例代码中，输出被屏蔽的比较匹配输出值高。

图 4.113 显示解除占空比 100%和 0%后 GTIOC0A 引脚输出的示例。

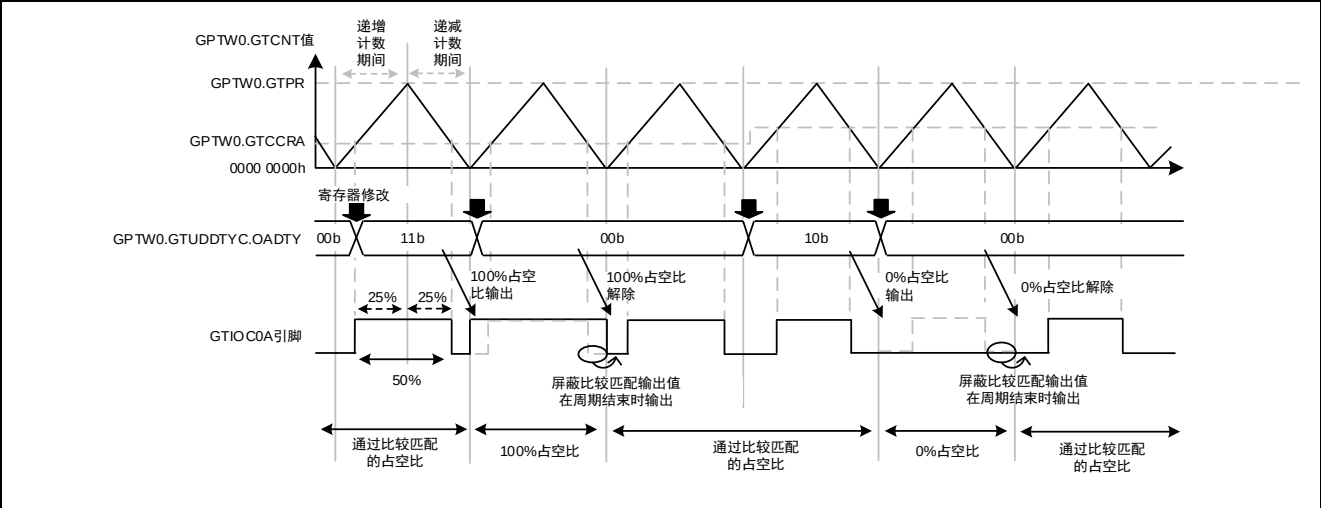


图 4.113 占空比 100%和 0%解除后的 1/2 占空比产生和输出

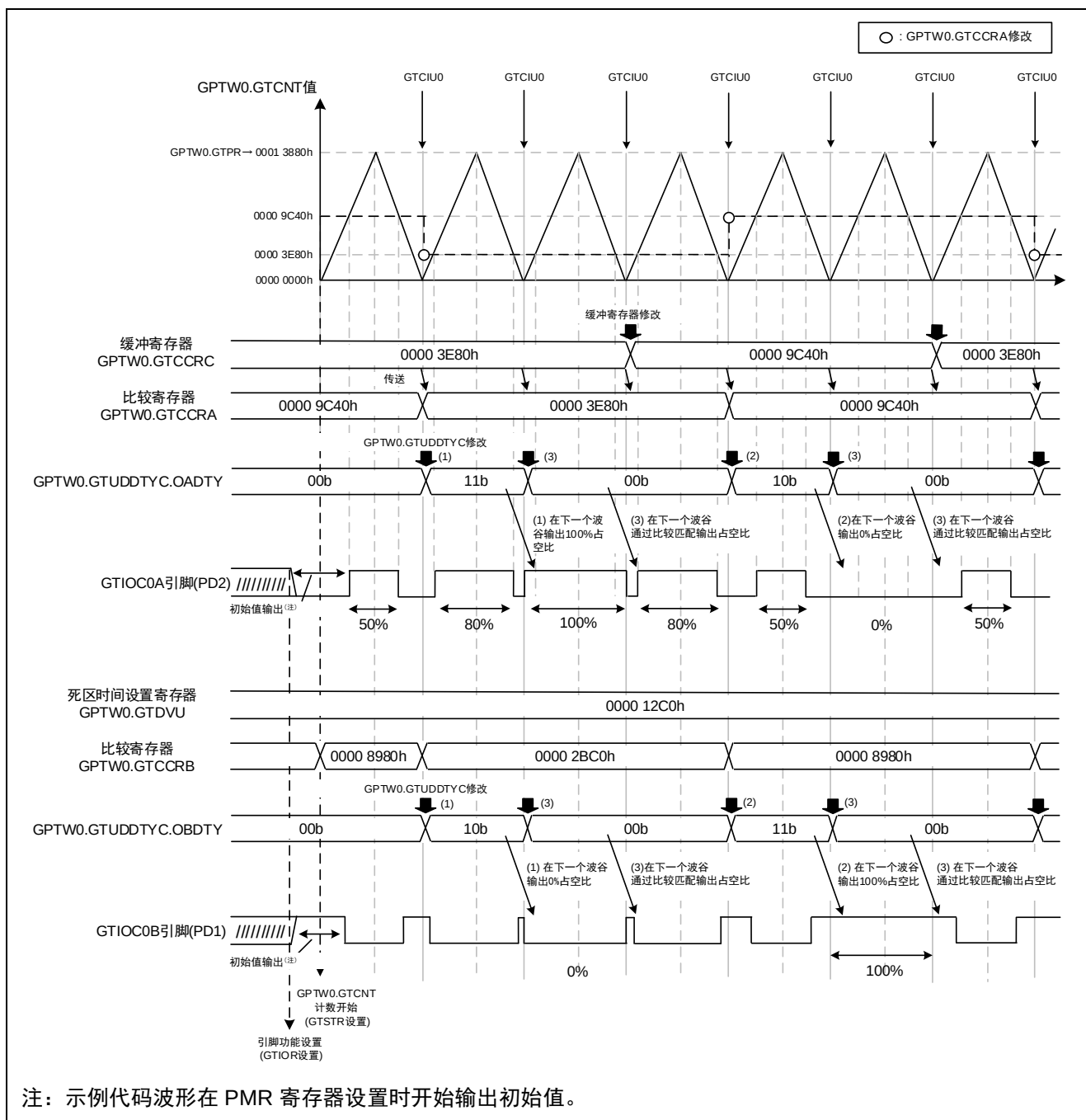


图 4.114 示例代码运行

4.12.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.15 添加组件

项目	描述
组件	通用 PWM 定时器
配置名称	Config_GPT0
工作模式	三角波 PWM 模式 1
资源	GPT0

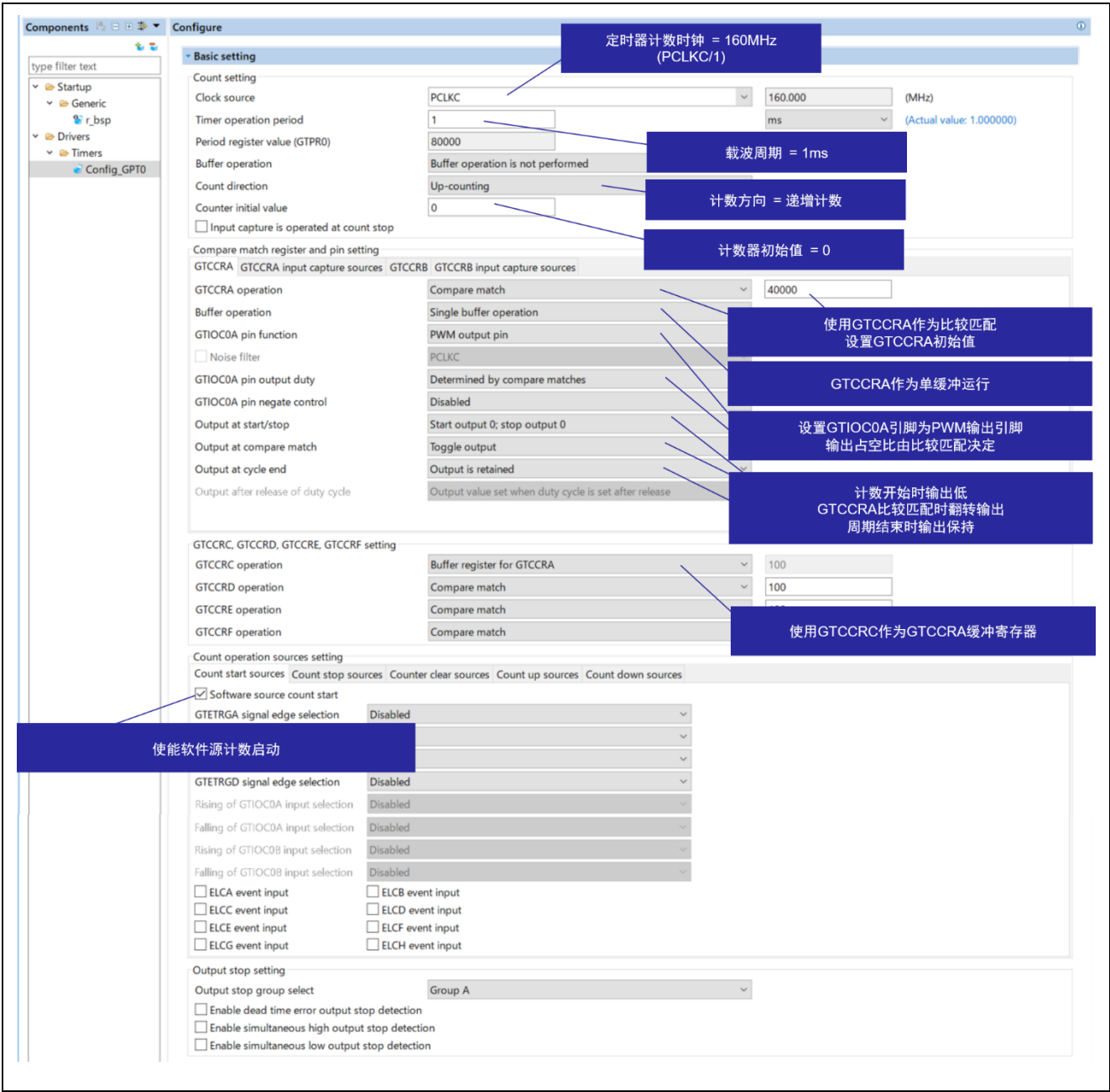


图 4.115 GPT0 设置 (1/2)

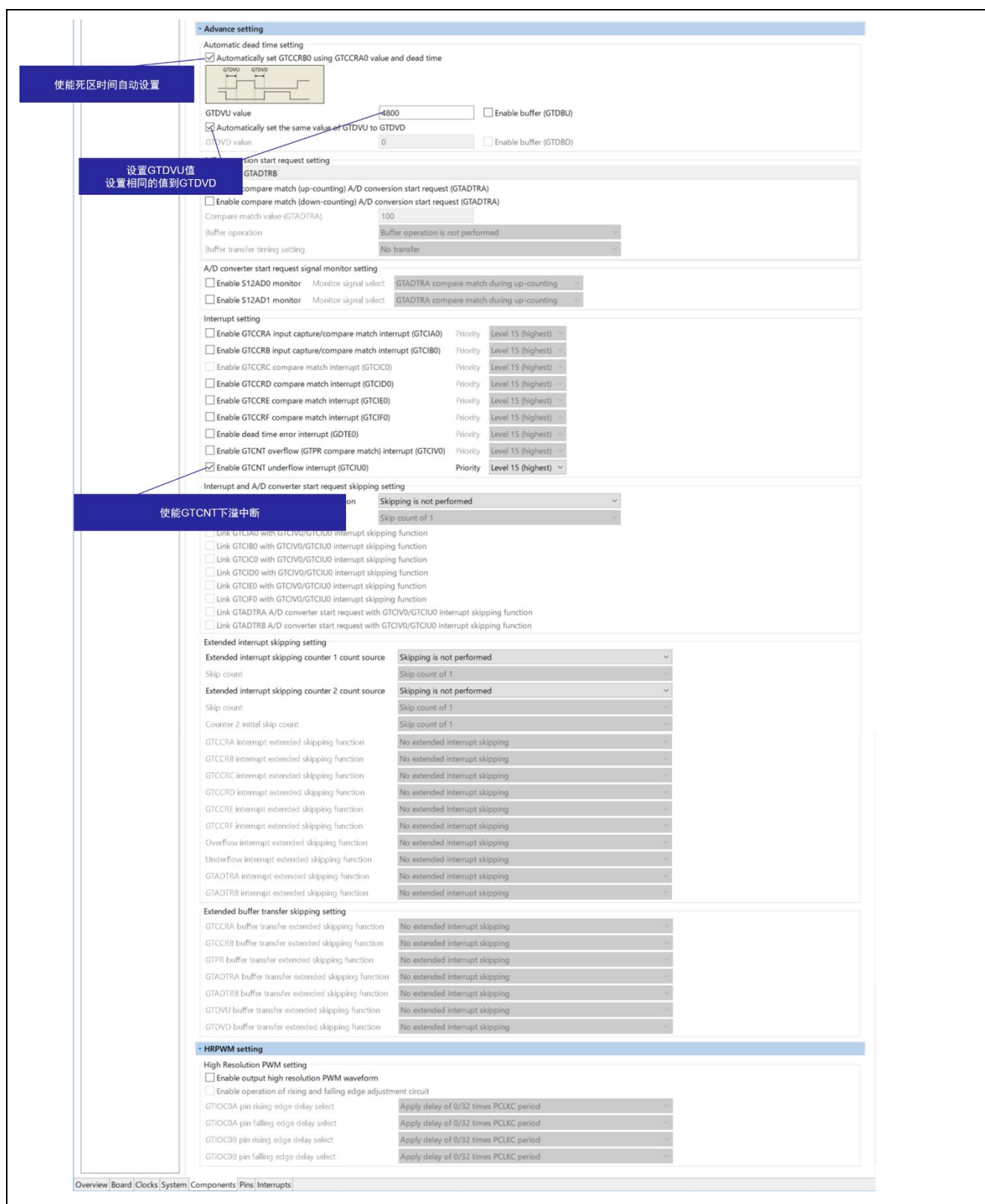


图 4.116 GPT0 设置 (2/2)

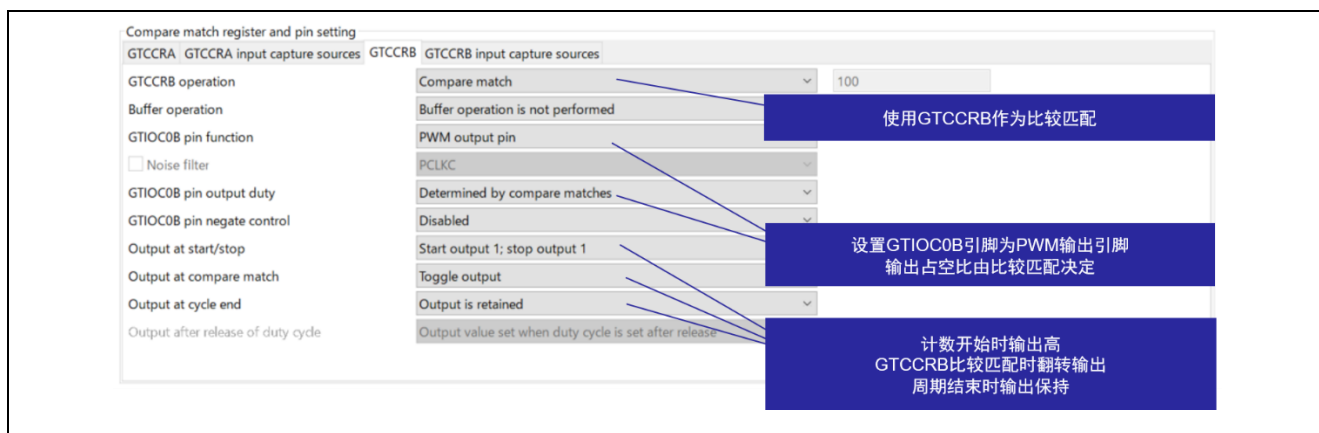


图 4.117 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.12.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。
在主函数中开始计数。

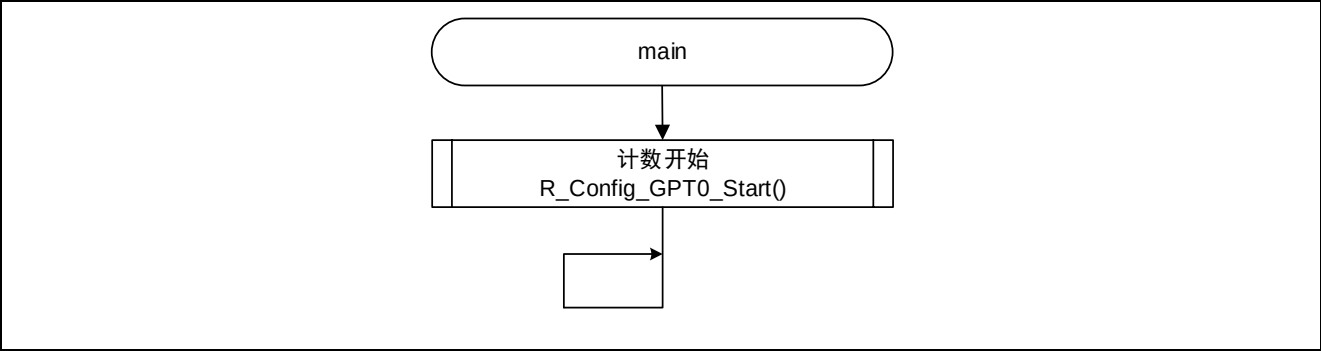


图 4.118 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 用来初始化变量、设置缓冲寄存器的初始值和解除占空比 100%和 0%后设置输出值。此函数在 R_Config_GPT0_Create 函数中调用。

此示例代码使用以下变量。

- s_duty_list_counter: 用于从占空比列表中读取的计数器变量

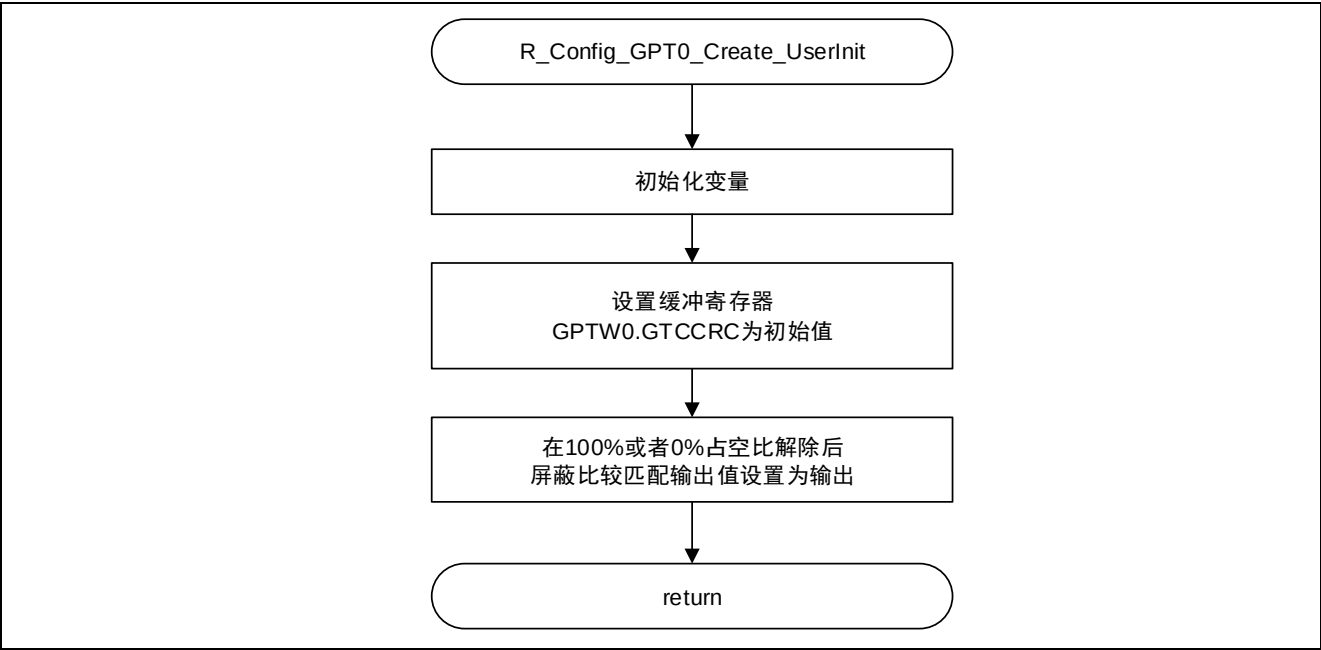


图 4.119 用户初始化函数

GTCIU0 中断处理函数根据要设置的下一个占空比更改缓冲寄存器和 GTUDDTYC 寄存器的值。

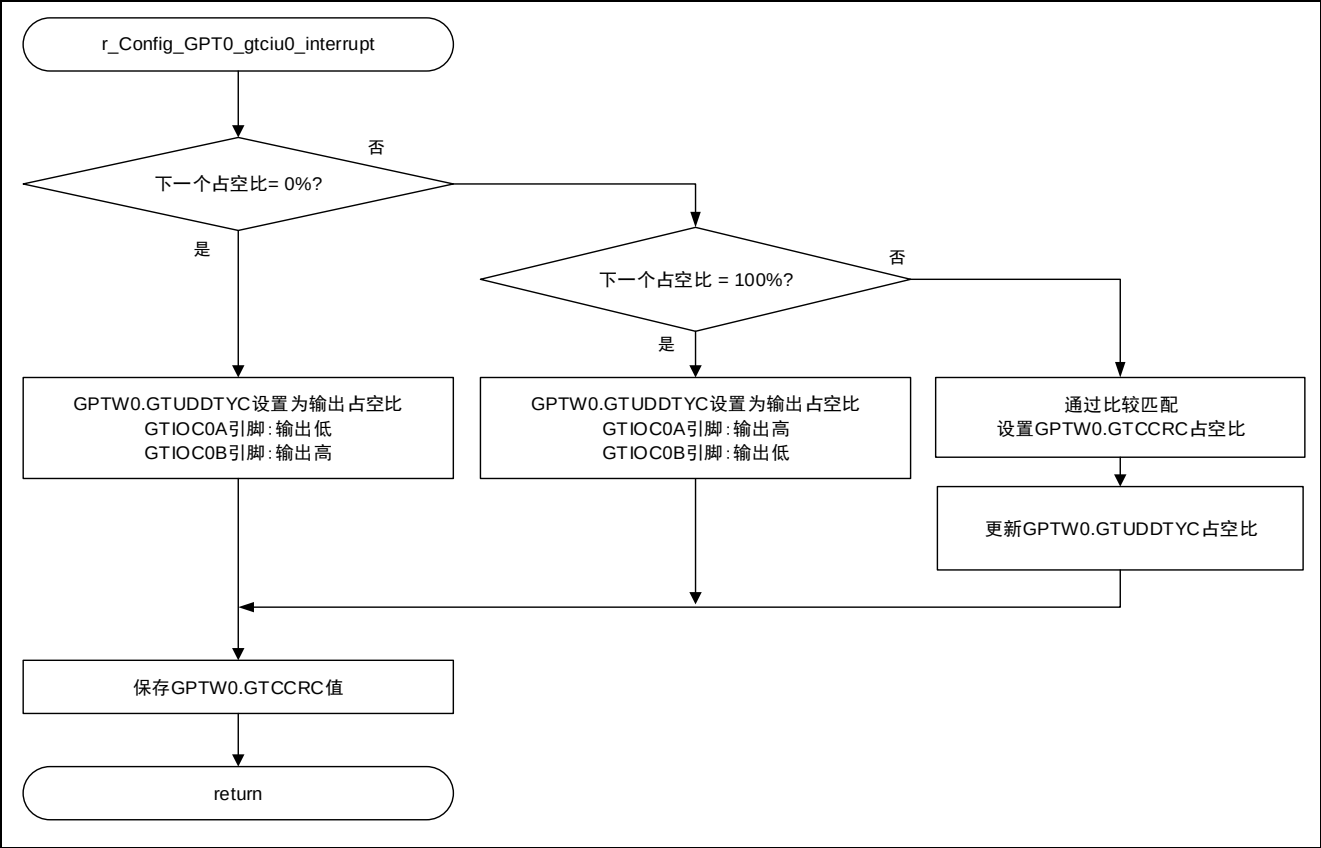


图 4.120 GTCIU0 中断处理函数

4.12.5 使用时的注意事项

4.12.5.1 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间, 并使用比较匹配寄存器 GTCCRA (正相波形寄存器) 的值来更新比较匹配寄存器 GTCCRB (反相波形寄存器) 的值。

设置比较寄存器 GTCCRA 的值应满足以下限制。

$GTCCRA > GTDVU$

$GTCCRA > GTDVD$

$GTCCRA < GTPR$

如果在计数运行期间将 GTCCRA 设置为 0000 0000h 或大于 GTPR 的设置值, 则输出保护功能被激活。

但是, 如果不满足以下条件, 则功能将无法正常运行。

当计数运行开始时 GTCCRA 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设定值时。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.8.4 GTIOCnm 引脚输出的输出保护功能 (n = 0 到 9; m = A、B)。

不使用死区时间自动设置功能时, 设置 GTCCRA (GTCCRB) 寄存器的值大于 0000 0001h 且小于 GTPR 寄存器的设定值。当在 GTCCRA 寄存器中设置 0000 0000h 或与 GTPR 寄存器相同的值时, 仅当满足 $[GTCCRA (GTCCRB) = 0000 0000h]$ 或 $[GTCCRA (GTCCRB) = GTPR]$ 时, 才会在一个周期中发生比较匹配。此外, 如果在 GTCCRA 中设置的值超过 GTPR 寄存器的设置值, 则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 GTCCRM 寄存器在比较匹配运行期间的设置 (m = A 到 F), (1) 在三角波 PWM 模式下进行死区时间自动设置时, 以及 (2) 在三角波 PWM 模式下未进行死区时间自动设置时。

4.12.5.2 比较匹配时 100% 占空比输出

若不更改 GTUDDTYC 寄存器, 则无法在比较匹配时将输出占空比切换为 100% 占空比。为了输出 100% 占空比, 设置 GTUDDTYC.OADTY 位为 11b 和 GTUDDTYC.OBDTY 位为 10b。

如果在此示例代码设置中将 GTCCRA 和 GTCCRB 设置为 0, 则无法输出 100% 占空比, 因为在发生 GTCNT 计数器溢出后输出一个时钟周期。

当 GTCCRA 和 GTCCRB 寄存器设置为与 GTPR 相同的值并且同时发生 GTCNT 计数溢出和比较匹配时, 比较匹配时间的输出设置被优先并且翻转, 禁止 100% 占空比输出。

有关 GPTW 中同时发生 GTCNT 计数器溢出和比较匹配时的波形输出的详情, 请参考 RX66T Group User's Manual: Hardware, 章节 24.2.14 通用 PWM 定时器 I/O 控制寄存器 (GTIOR) 中表 24.4 下的注释。

4.12.5.3 占空比 0%和 100%输出期间比较匹配运行

在此示例代码中，输出占空比 0%和 100%是基于 GTUDDTYC.OADTY 位的设置值决定的。当占空比设置为 0%或 100%时，GPTW 中的比较匹配继续，并执行中断输出和缓冲传送运行。

此示例代码不使用比较匹配中断，因此如果你要使用比较匹配中断，请在占空比 0%和 100%输出期间谨慎使用中断。

4.12.5.4 从占空比 0%和 100%切换

如果在设置占空比 0%和 100%后通过比较匹配更改输出设置，则输出值在周期结束时通过 GTIOR.GTIOA[3:2]和 GTUDDTYC.OADTYR 位的值决定。

以下是示例代码的设置。注意，如果 GTUDDTYC.OADTYR 位的初始硬件值为 0，无法执行与此示例代码相同的运行。同样适用于 GTIOC0B 引脚。

- GTIOR.GTIOA[3:2] = 00b: 周期结束时保持输出
- GTUDDTYC.OADTYR = 1: 在解除占空比 0%/100%后，将 GTIOA[3:2]应用于屏蔽的比较匹配输出值。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.6 占空比 0%/100%输出功能。

4.13 三角波 PWM 模式 1 占空比 0%和 100%

- 目标示例代码文件名: r01an5995_rx66t_gptw_triangle_pwm_0to100_dt.zip

4.13.1 概要

GPTW 三角波 PWM 模式 1 可用于根据 GTCCRA 和 GTCCRB 寄存器的比较匹配和设置 GTUDDTYC 寄存器输出占空比 0%到 100%。

此示例代码描述了在三角波 PWM 模式 1 下使用死区时间自动设置功能并重复输出在占空比 0%和 100%之间交替的波形。

当 GTCNT 计数器下溢发生时, 通过修改 GTUDDTYC 寄存器, 输出在占空比 0%和 100%之间切换。

以下列表提供了示例代码中使用的 GPTW 设置。

- 使用三角波 PWM 模式 1
- 使用通道 0
- 载波周期 = 1ms
- 定时器计数时钟 = 160MHz (PCLKC/1)
- 使用 GTPR 作为周期寄存器
 - 计数器从初始值 0 开始递增计数
- 使用 GTCCRA 作为占空比输出比较匹配
 - 使用 GTIOC0A 引脚作为 PWM 输出引脚
 - 使用 GTCCRA 作为比较匹配
 - 计数开始时输出低
 - GTCCRA 比较匹配时翻转输出
 - 周期结束时输出保持
 - 在计数开始时强制输出 0%占空比*
- 使用 GTCCRB 作为占空比输出比较匹配
 - 使用 GTIOC0B 引脚作为 PWM 输出引脚
 - 使用 GTCCRB 作为比较匹配
 - 计数开始时输出高
 - GTCCRB 比较匹配时翻转输出
 - 周期结束时输出保持
 - 在计数开始时强制输出 0%占空比*
- 使用缓冲寄存器
 - GTCCRA 作为单缓冲运行
 - 使用 GTCCRC 作为 GTCCRA 的缓冲寄存器
- 使用死区时间自动生成
- 使能软件源计数启动
- 每个周期的占空比改变
 - 在 GTCNT 计数器下溢中断时改变占空比
 - 关于占空比改变的时间的详细信息, 请参考图 4.122

在 Smart Configurator 中设置。
设置方法请参考章节 4.13.3。
标记*的项目除外。

此示例代码的三角波 PWM 模式 1 输出如下所示。

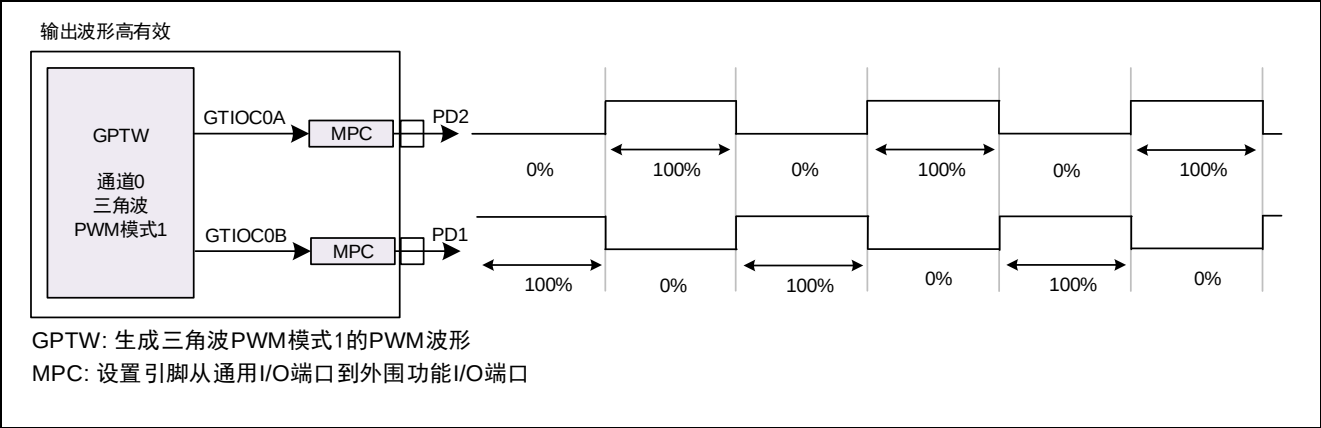


图 4.121 三角波 PWM 模式 1 输出

4.13.2 运行说明

示例代码的运行如图 4.122 所示。通过在 GTCNT 计数器下溢中断（GTCIU0）时修改 GTUDDTYC 寄存器的值，每个周期的占空比在 0%和 100%之间切换。

- 计数开始后 0%占空比输出（图 4.122 中（1））
当 GTUDDTYC.OADTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OADTY 位为 10b，在计数开始后 GTIOC0A 引脚立即输出低电平。即使发生 GTCCRA 比较匹配，波形也不会改变。
当 GTUDDTYC.OBDTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OBDTY 位为 11b，在计数开始后 GTIOC0B 引脚将立即输出高电平。即使发生 GTCCRB 比较匹配，波形也不会改变。在此示例代码中，GTUDDTYC.OBDTY 位设置为 11b，以使 GTIOC0B 引脚在一个周期内输出高电平。
- 第二个周期 100%占空比输出（图 4.122 中（2））
在计数开始后第一次 GTCNT 计数器下溢发生之前，通过设置 GTUDDTYC.OADTY 位为 11b，GTIOC0A 引脚将从第二个周期输出高电平。即使发生 GTCCRA 比较匹配，波形也不会改变。
在计数开始后第一次 GTCNT 计数器下溢发生之前，通过设置 GTUDDTYC.OBDTY 位为 10b，GTIOC0B 引脚将从第二个周期输出低电平。即使发生 GTCCRB 比较匹配，波形也不会改变。在此示例代码中，GTUDDTYC.OBDTY 位设置为 10b，以使 GTIOC0B 引脚在一个周期内输出低电平。

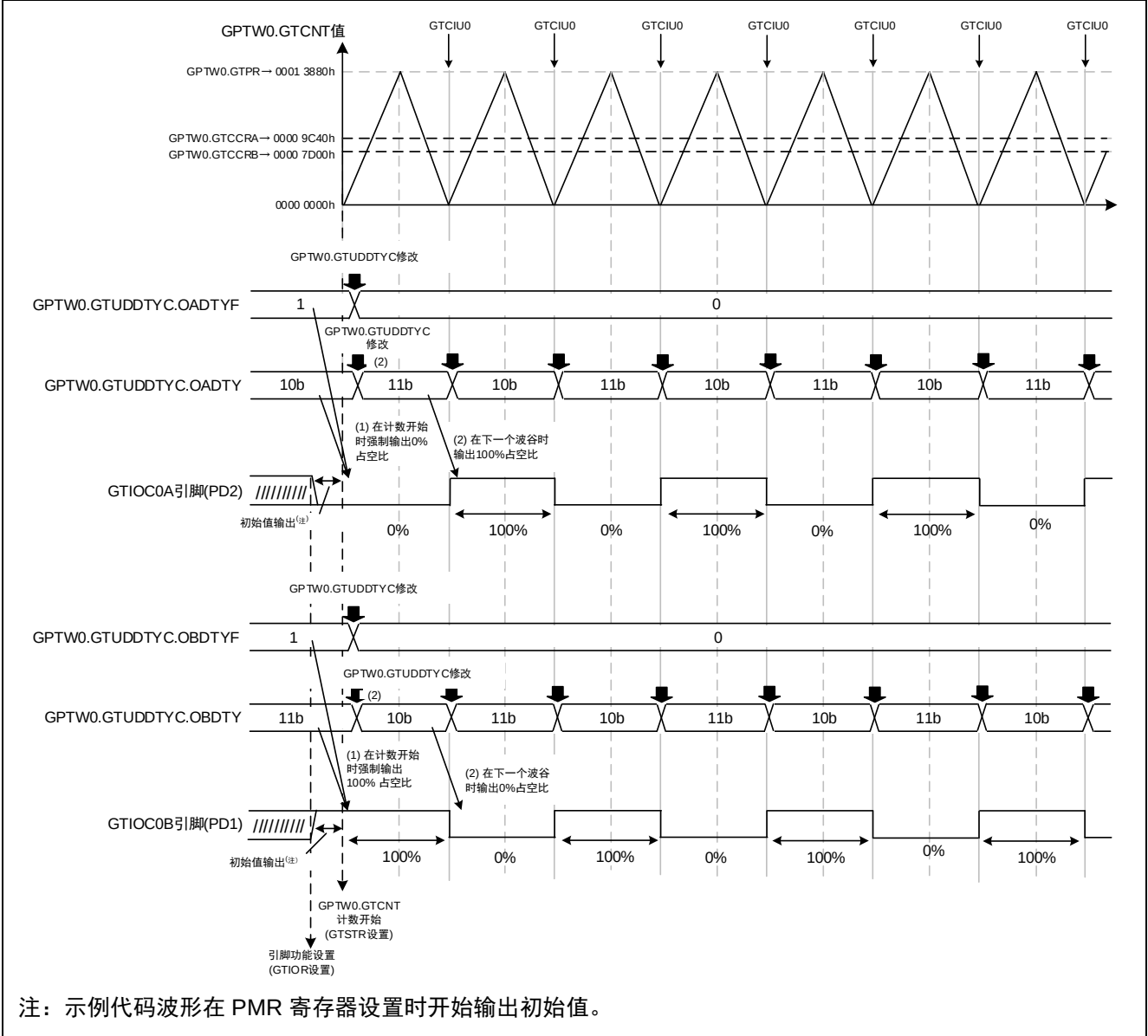


图 4.122 示例代码运行

4.13.3 Smart Configurator 设置

如下所述，示例代码使用 Smart Configurator 添加 GPTW。添加组件的具体说明请参考章节 4.1.4 添加组件。

表 4.16 添加组件

项目	描述
组件	通用 PWM 定时器（GPTW）
配置名称	Config_GPT0
工作模式	三角波 PWM 模式 1
资源	GPT0

定时计数器时钟 = 160MHz (PCLKC/1)

载波周期 = 1ms

计数方向 = 递增计数

计数器初始值 = 0

使用 GTCCRA 作为比较匹配 设置 GTCCRA 初始值

GTCCRA 作为单缓冲运行

使用 GTIOC0A 引脚作为 PWM 输出 通过比较匹配确定输出占空比 (注)

计数开始时输出低 GTCCRA 比较匹配时翻转输出 周期结束时输出保持

使用 GTCCRC 作为 GTCCRA 缓冲寄存器

使能软件源计数启动

注：在计数开始时设置为强制输出占空比后，占空比输出需要设置为 0%。
在 R_Config_GPT0_Create_UserInit 函数中设置 0% 占空比输出。

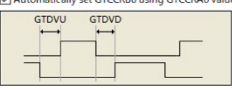
图 4.123 GPT0 设置（1/2）

Advance setting

Automatic dead time setting
☒ Automatically set GTCCRB0 using GTCCRA0 value and dead time

使能死区时间自动设置

**设置GTDVU值
设置相同的值到GTDVD**



GTDVU value: 4800 ☐ Enable buffer (GTD8U)
 GTDVD value: 0 ☐ Enable buffer (GTD8D)

A/D conversion start request setting
 GTADTRA GTADTRB
☐ Enable compare match (up-counting) A/D conversion start request (GTADTRA)
☐ Enable compare match (down-counting) A/D conversion start request (GTADTRA)
 Compare match value (GTADTRA): 100
 Buffer operation: Buffer operation is not performed
 Buffer transfer timing setting: No transfer

A/D converter start request signal monitor setting
☐ Enable S12AD0 monitor Monitor signal select: GTADTRA compare match during up-counting
☐ Enable S12AD1 monitor Monitor signal select: GTADTRA compare match during up-counting

Interrupt setting
☐ Enable GTCCRA input capture/compare match interrupt (GTCIA0) Priority: Level 15 (highest)
☐ Enable GTCCRB input capture/compare match interrupt (GTCIB0) Priority: Level 15 (highest)
☐ Enable GTCCRC compare match interrupt (GTCIC0) Priority: Level 15 (highest)
☐ Enable GTCCRD compare match interrupt (GTCID0) Priority: Level 15 (highest)
☐ Enable GTCCRE compare match interrupt (GTCIE0) Priority: Level 15 (highest)
☐ Enable GTCCRF compare match interrupt (GTCIF0) Priority: Level 15 (highest)
☐ Enable dead time error interrupt (GTE0) Priority: Level 15 (highest)
☐ Enable GTCNT overflow (GTPR compare match) interrupt (GTCV0) Priority: Level 15 (highest)
☒ Enable GTCNT underflow interrupt (GTCIU0) Priority: Level 15 (highest)

使能GTCNT下溢中断

Interrupt and A/D converter start request skipping setting
 GTCIU0 interrupt skipping function: Skipping is not performed
 GTCIU0 interrupt skipping count: Skip count of 1
☐ Link GTCIA0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTCIB0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTCIC0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTCID0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTCIE0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTCIF0 with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTADTRA A/D converter start request with GTCV0/GTCIU0 interrupt skipping function
☐ Link GTADTRB A/D converter start request with GTCV0/GTCIU0 interrupt skipping function

Extended interrupt skipping setting
 Extended interrupt skipping counter 1 count source: Skipping is not performed
 Skip count: Skip count of 1
 Extended interrupt skipping counter 2 count source: Skipping is not performed
 Skip count: Skip count of 1
 Counter 2 initial skip count: Skip count of 1
 GTCCRA interrupt extended skipping function: No extended interrupt skipping
 GTCCRB interrupt extended skipping function: No extended interrupt skipping
 GTCCRC interrupt extended skipping function: No extended interrupt skipping
 GTCCRD interrupt extended skipping function: No extended interrupt skipping
 GTCCRE interrupt extended skipping function: No extended interrupt skipping
 GTCCRF interrupt extended skipping function: No extended interrupt skipping
 Overflow interrupt extended skipping function: No extended interrupt skipping
 Underflow interrupt extended skipping function: No extended interrupt skipping
 GTADTRA interrupt extended skipping function: No extended interrupt skipping
 GTADTRB interrupt extended skipping function: No extended interrupt skipping

Extended buffer transfer skipping setting
 GTCCRA buffer transfer extended skipping function: No extended interrupt skipping
 GTCCRB buffer transfer extended skipping function: No extended interrupt skipping
 GTPR buffer transfer extended skipping function: No extended interrupt skipping
 GTADTRA buffer transfer extended skipping function: No extended interrupt skipping
 GTADTRB buffer transfer extended skipping function: No extended interrupt skipping
 GTDVU buffer transfer extended skipping function: No extended interrupt skipping
 GTDVD buffer transfer extended skipping function: No extended interrupt skipping

HRPWM setting

High Resolution PWM setting
☐ Enable output high resolution PWM waveform
☐ Enable operation of rising and falling edge adjustment circuit
 GTIO0A pin rising edge delay select: Apply delay of 0/32 times PCLKC period
 GTIO0A pin falling edge delay select: Apply delay of 0/32 times PCLKC period
 GTIO0B pin rising edge delay select: Apply delay of 0/32 times PCLKC period
 GTIO0B pin falling edge delay select: Apply delay of 0/32 times PCLKC period

Overview | Board | Clocks | System | Components | Pins | Interrupts

图 4.124 GPT0 设置 (2/2)

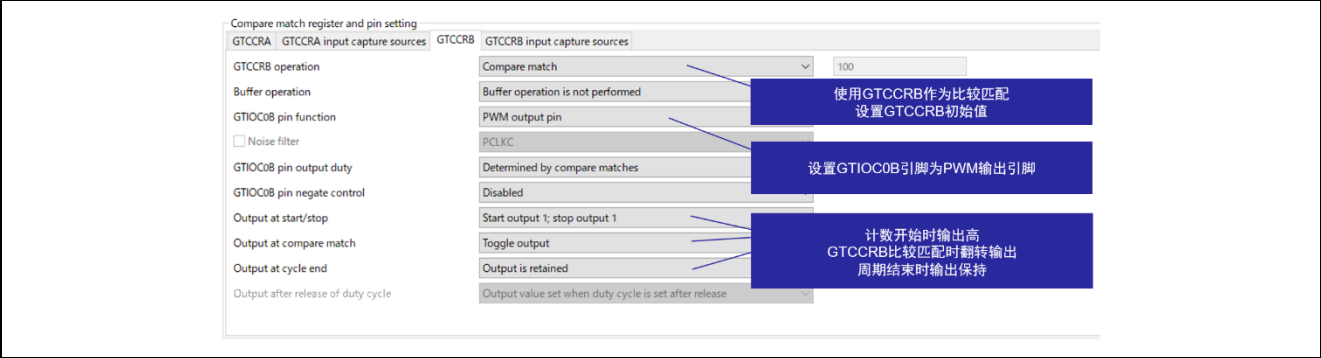


图 4.125 GPT0 设置（比较匹配寄存器和 GTCCRB 引脚设置）

4.13.4 流程图

以下流程图显示了由 Smart Configurator 生成代码后添加的函数的处理。

在主函数中，开始计数并设置下一个周期的占空比为 100%。

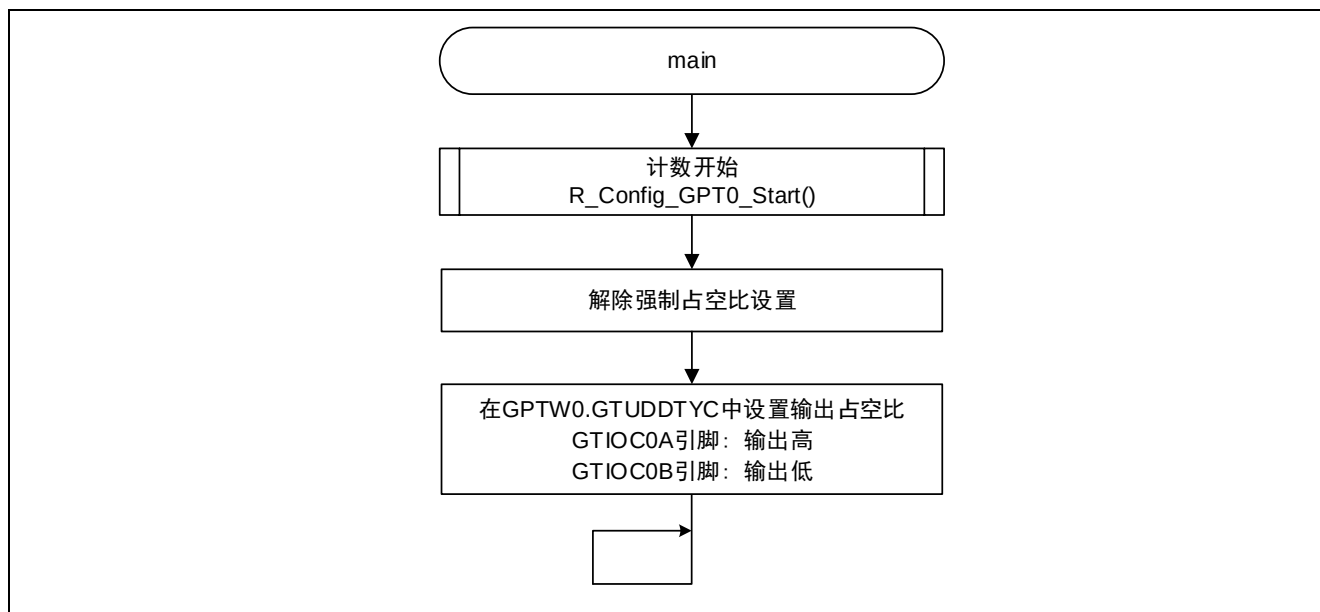


图 4.126 主函数

在主函数之前执行的用户初始化函数 R_Config_GPT0_Create_UserInit 在计数开始时设置为强制输出占空比并将输出占空比设置为 0%。此函数在 R_Config_GPT0_Create 函数中调用。

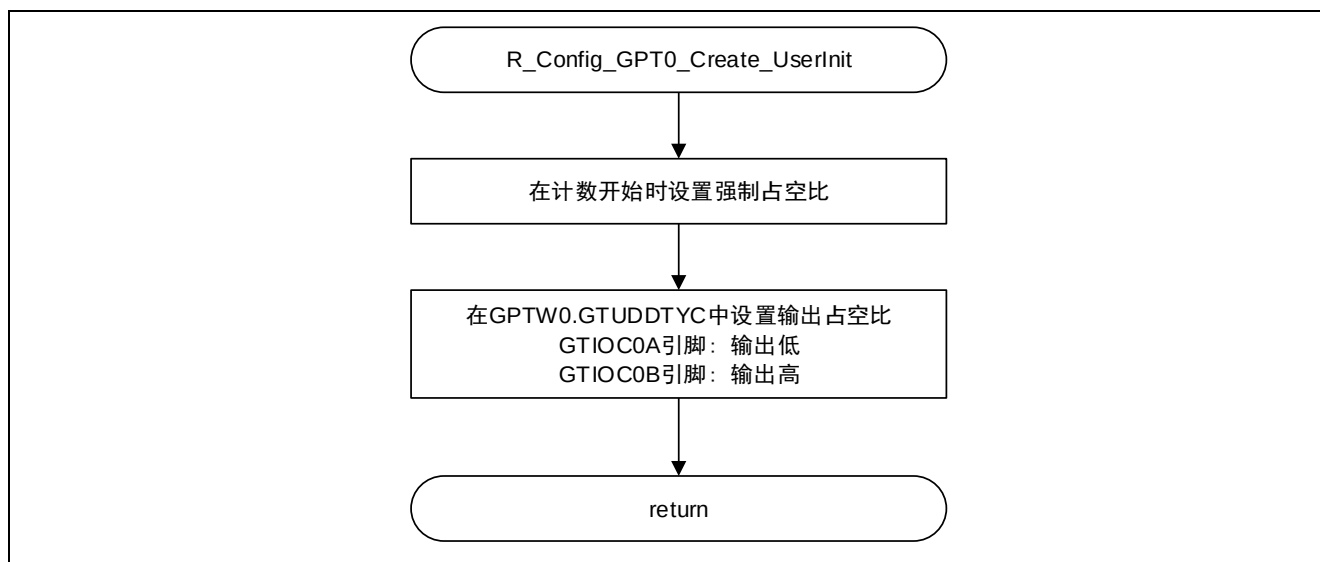


图 4.127 用户初始化函数

GTCIU0 中断处理函数更改 GTUDDTYC 寄存器的值。

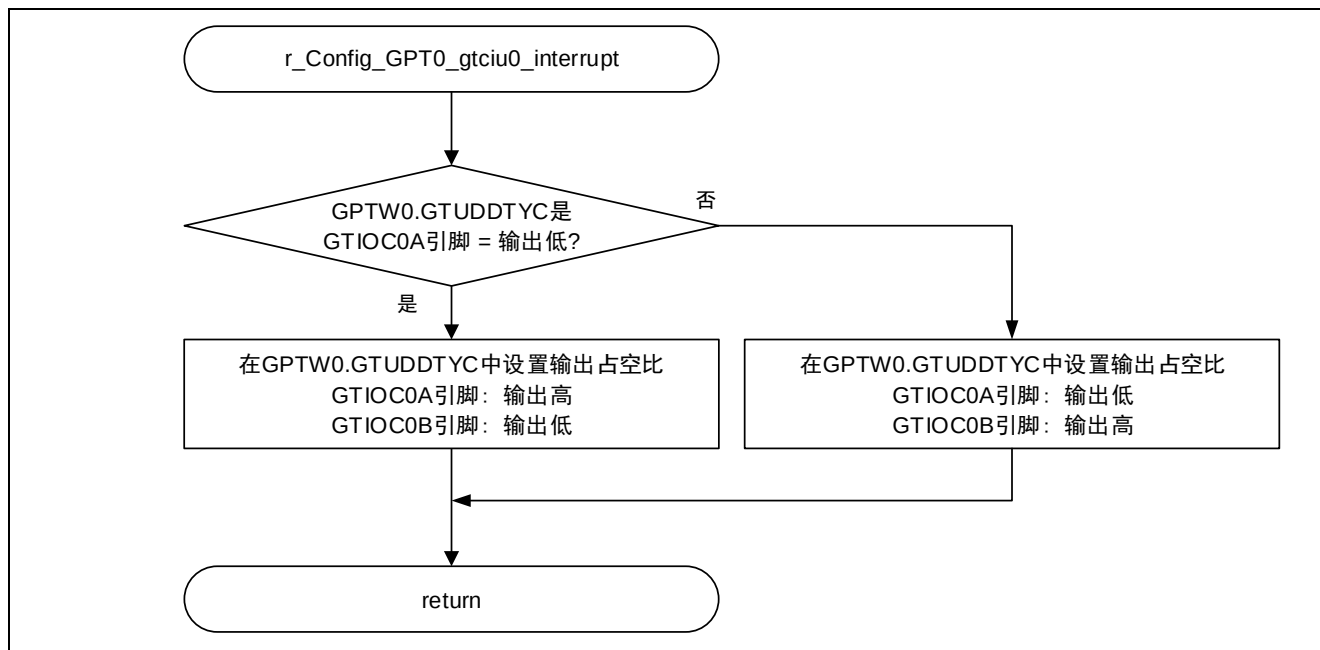


图 4.128 GTCIU0 中断处理函数

4.13.5 相关运行

4.13.5.1 在初始值低时保持 0%后更改占空比到 100%

下面显示了在计数开始并输出几个周期 0%后切换占空比到 100%的运行示例。

当 GTUDDTYC.OADTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OADTY 位为 10b，在计数开始后 GTIOC0A 引脚立即输出低电平并且输出占空比切换到 0%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，在下一个周期中输出变高并且占空比变为 100%（下图中（2））。

当 GTUDDTYC.OBDTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OBDTY 位为 11b，在计数开始后 GTIOC0B 引脚立即输出高电平并且输出占空比切换到 100%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OBDTY 位为 10b，在下一个周期中输出变低并且占空比变为 0%（下图中（2））。在此示例中，GTUDDTYC.OBDTY 位被设置为 11b 以使 GTIOC0B 引脚输出固定为高并输出 100%占空比，而被设置为 10b 以使 GTIOC0B 引脚输出固定为低并输出 0%占空比。

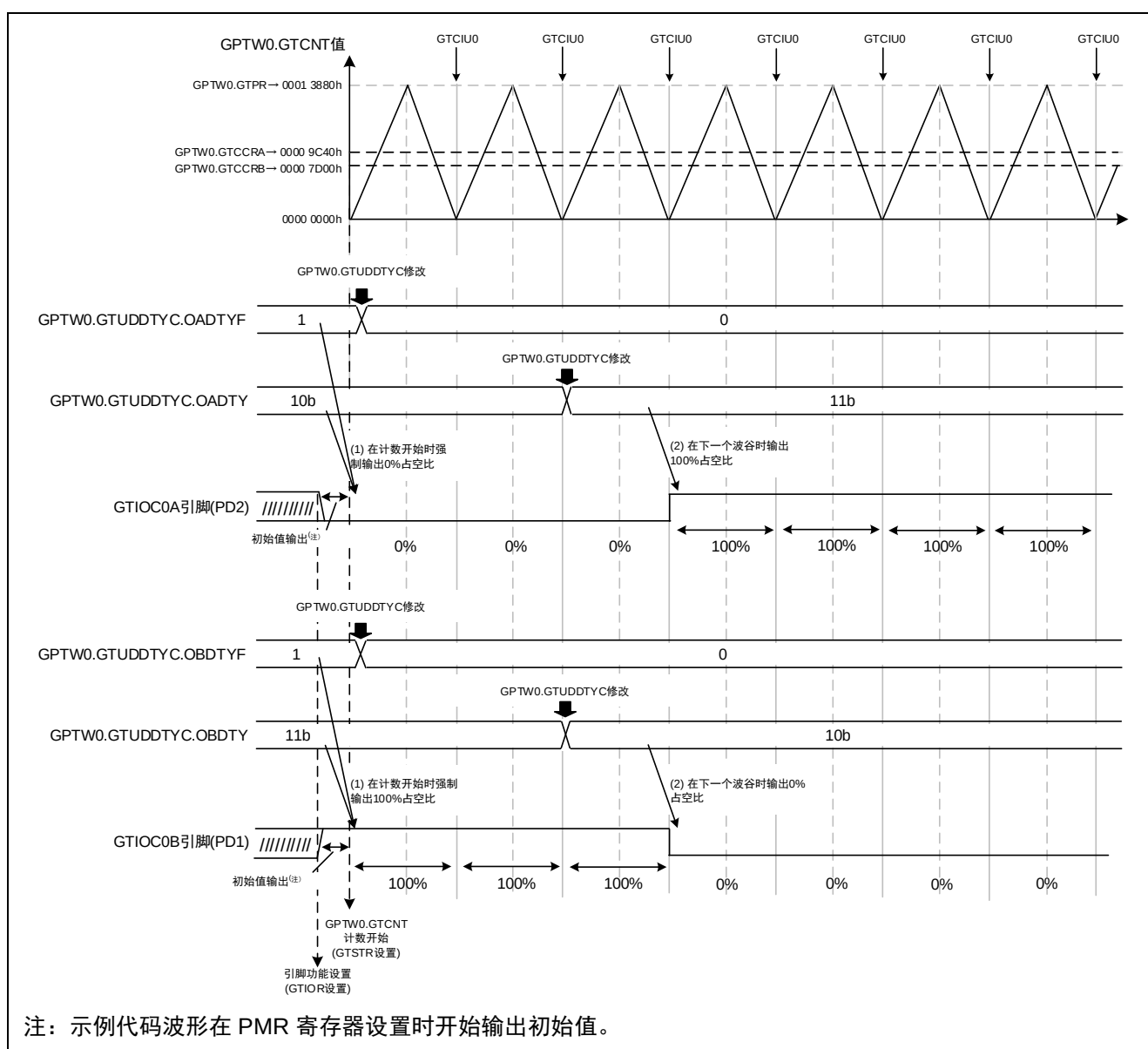


图 4.129 在保持 0%后更改占空比到 100%

4.13.5.2 在初始值低时保持 100%后更改占空比到 0%

下面显示了在计数开始并输出几个周期 100%后切换占空比到 0%的运行示例。

当 GTUDDTYC.OADTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OADTY 位为 11b，在计数开始后 GTIOC0A 引脚立即输出高电平并且输出占空比切换到 100%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 10b，在下一个周期中输出变低并且占空比变为 0%（下图中（2））。

当 GTUDDTYC.OBDTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OBDTY 位为 10b，在计数开始后 GTIOC0B 引脚立即输出低电平并且输出占空比切换到 0%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OBDTY 位为 11b，在下一个周期中输出变高并且占空比变为 100%（下图中（2））。在此示例中，GTUDDTYC.OBDTY 位被设置为 10b 以使 GTIOC0B 引脚输出固定为高并输出 100%占空比，而被设置为 11b 以使 GTIOC0B 引脚输出固定为低并输出 0%占空比。

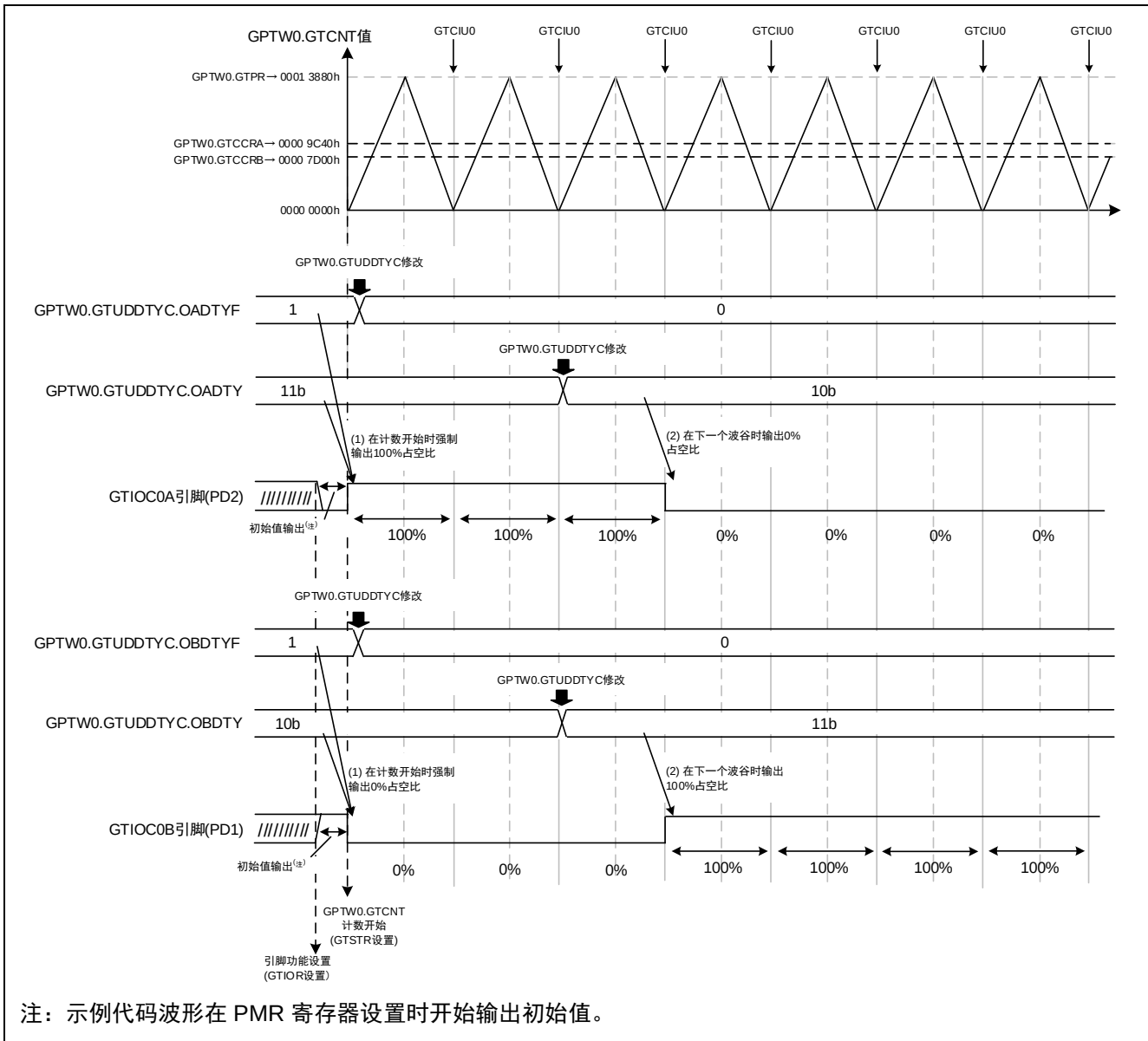


图 4.130 在保持 100%后更改占空比到 0%

4.13.5.3 在初始值高时保持 0%后更改占空比到 100%

下面显示了在计数开始并输出几个周期 0%后切换占空比到 100%的运行示例。

当 GTUDDTYC.OADTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OADTY 位为 10b，在计数开始后 GTIOC0A 引脚立即输出低电平并且输出占空比切换到 0%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，在下一个周期中输出变高并且占空比变为 100%（下图中（2））。

当 GTUDDTYC.OBDTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OBDTY 位为 11b，在计数开始后 GTIOC0B 引脚立即输出高电平并且输出占空比切换到 100%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OBDTY 位为 10b，在下一个周期中输出变低并且占空比变为 0%（下图中（2））。在此示例中，GTUDDTYC.OBDTY 位被设置为 11b 以使 GTIOC0B 引脚输出固定为高并输出 100%占空比，而被设置为 10b 以使 GTIOC0B 引脚输出固定为低并输出 0%占空比。

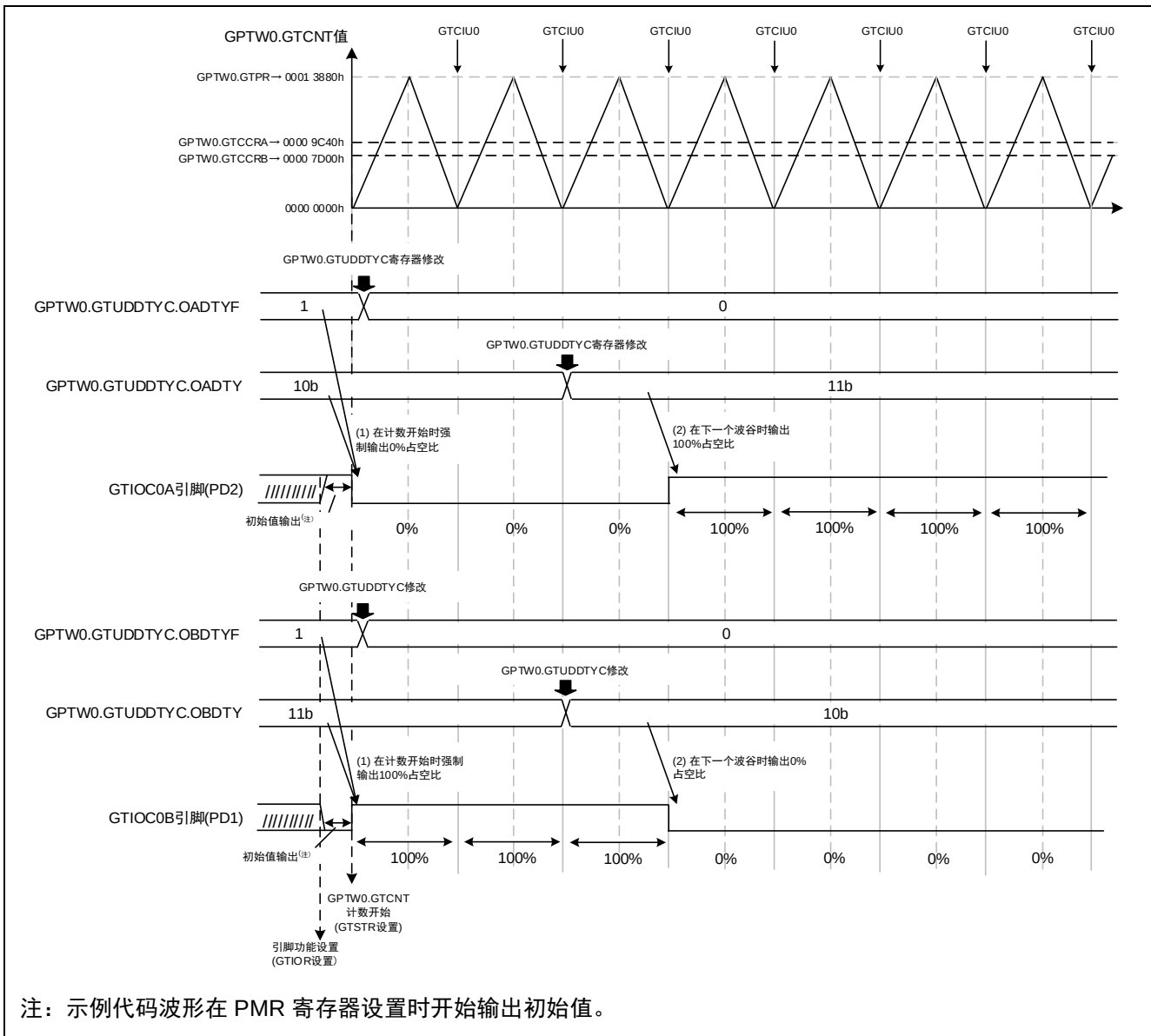


图 4.131 在保持 0%后更改占空比到 100%

要使 GTIOC0A 引脚的初始值为高，设置“Output at start/stop”为“stop output 1”。

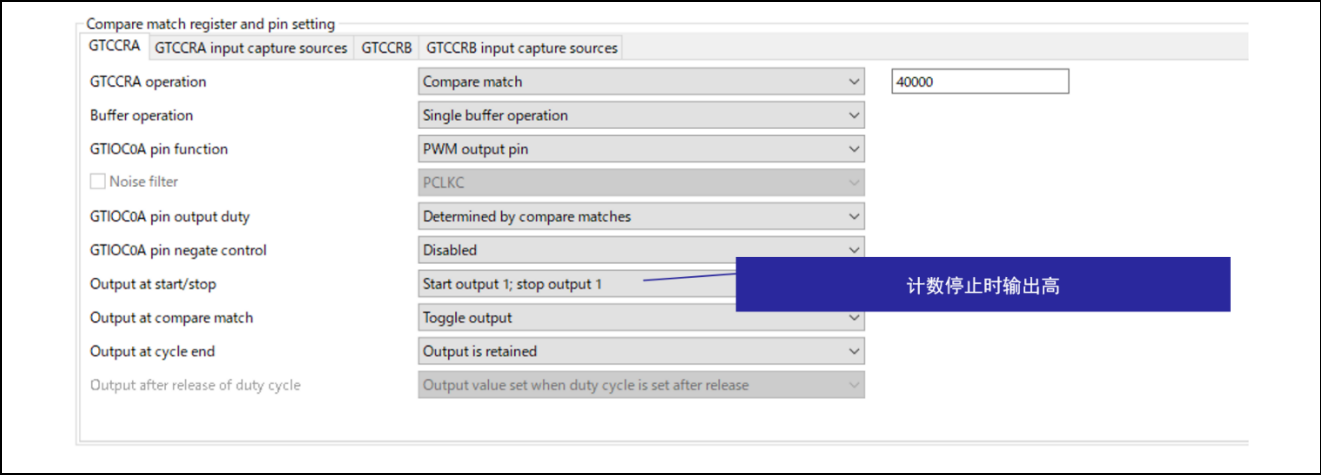


图 4.132 初始值高的 Smart Configurator 设置

要使 GTIOC0B 引脚的初始值为低，设置“Output at start/stop”为“stop output 0”。

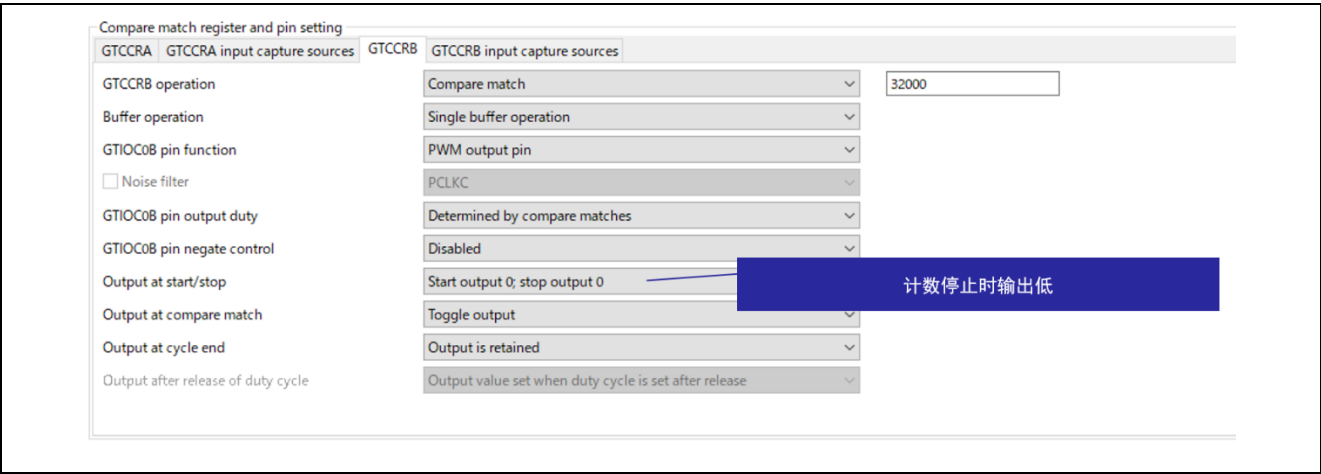


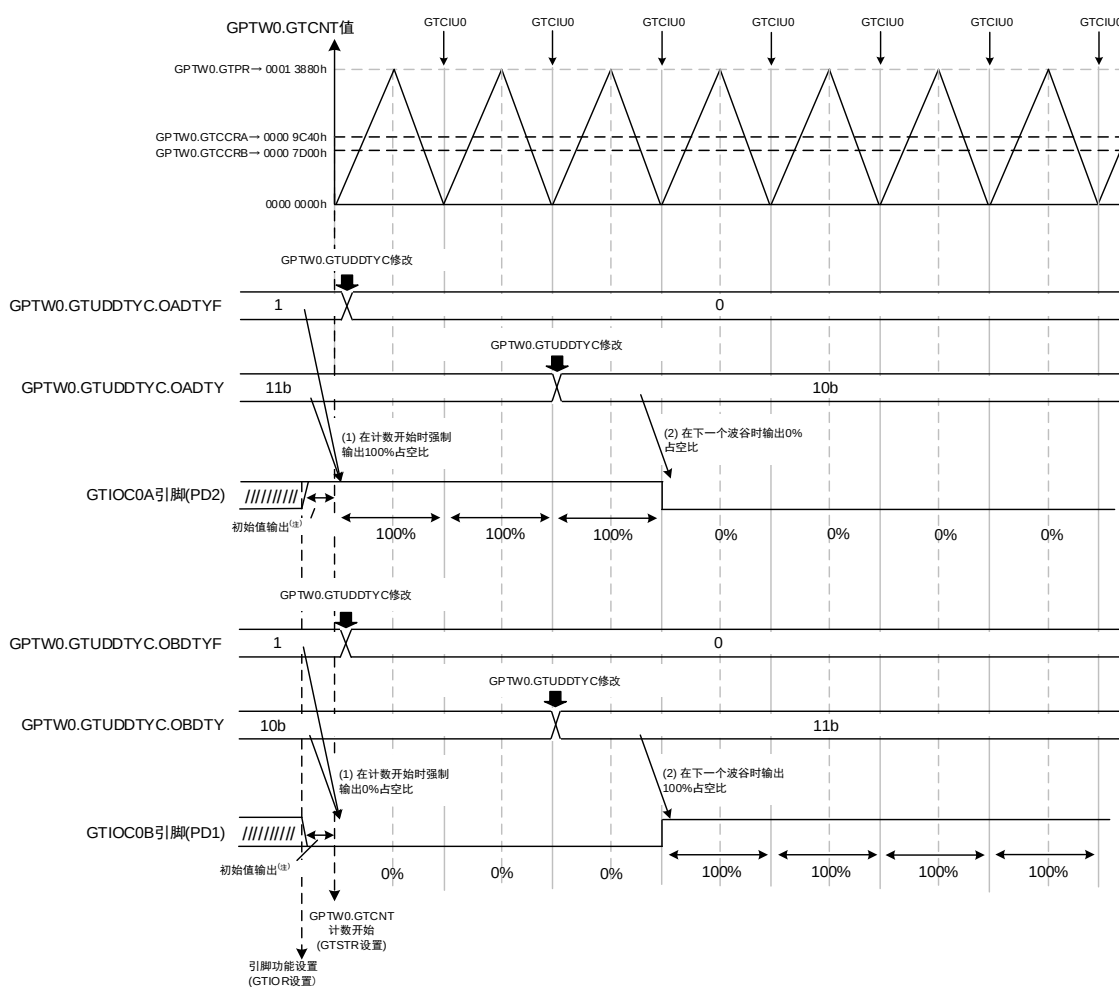
图 4.133 初始值低的 Smart Configurator 设置

4.13.5.4 在初始值高时保持 100%后更改占空比到 0%

下面显示了在计数开始并输出几个周期 100%后切换占空比到 0%的运行示例。

当 GTUDDTYC.OADTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OADTY 位为 11b，在计数开始后 GTIOC0A 引脚立即输出高电平并且输出占空比切换到 100%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 10b，在下一个周期中输出变低并且占空比变为 0%（下图中（2））。

当 GTUDDTYC.OBDTYF 位为 1b 且计数停止时，通过设置 GTUDDTYC.OBDTY 位为 10b，在计数开始后 GTIOC0B 引脚立即输出低电平并且输出占空比切换到 0%（下图中（1））。在计数运行期间，通过设置 GTUDDTYC.OBDTY 位为 11b，在下一个周期中输出变高并且占空比变为 100%（下图中（2））。在此示例中，GTUDDTYC.OBDTY 位被设置为 10b 以使 GTIOC0B 引脚输出固定为低并输出 0%占空比，而被设置为 11b 以使 GTIOC0B 引脚输出固定为高并输出 100%占空比。



注：示例代码波形在 PMR 寄存器设置时开始输出初始值。

图 4.134 在保持 100%后更改占空比到 0%

要使 GTIOC0A 引脚的初始值为高，设置“Output at start/stop”为“start output 1”。详情参考图 4.132。

以同样的方式，要使 GTIOC0B 引脚的初始值为低，设置“Output at start/stop”为“start output 0”。详情参考图 4.133。

4.13.5.5 在保持 50%后更改占空比到 100%

下面显示了在输出几个周期 50%占空比后 GTUDDTYC.OADTY 和 GTUDDTYC.OBDTY 位被更改为切换到 100%占空比的运行示例。

在计数运行期间，通过设置 GTUDDTYC.OADTY 位为 11b，在下一个周期中 GTIOC0A 引脚输出变高并且变为 100%占空比（下图中（1））。

在计数运行期间，通过设置 GTUDDTYC.OBDTY 位为 10b，在下一个周期中 GTIOC0B 引脚输出变低并且变为 0%占空比（下图中（1））。

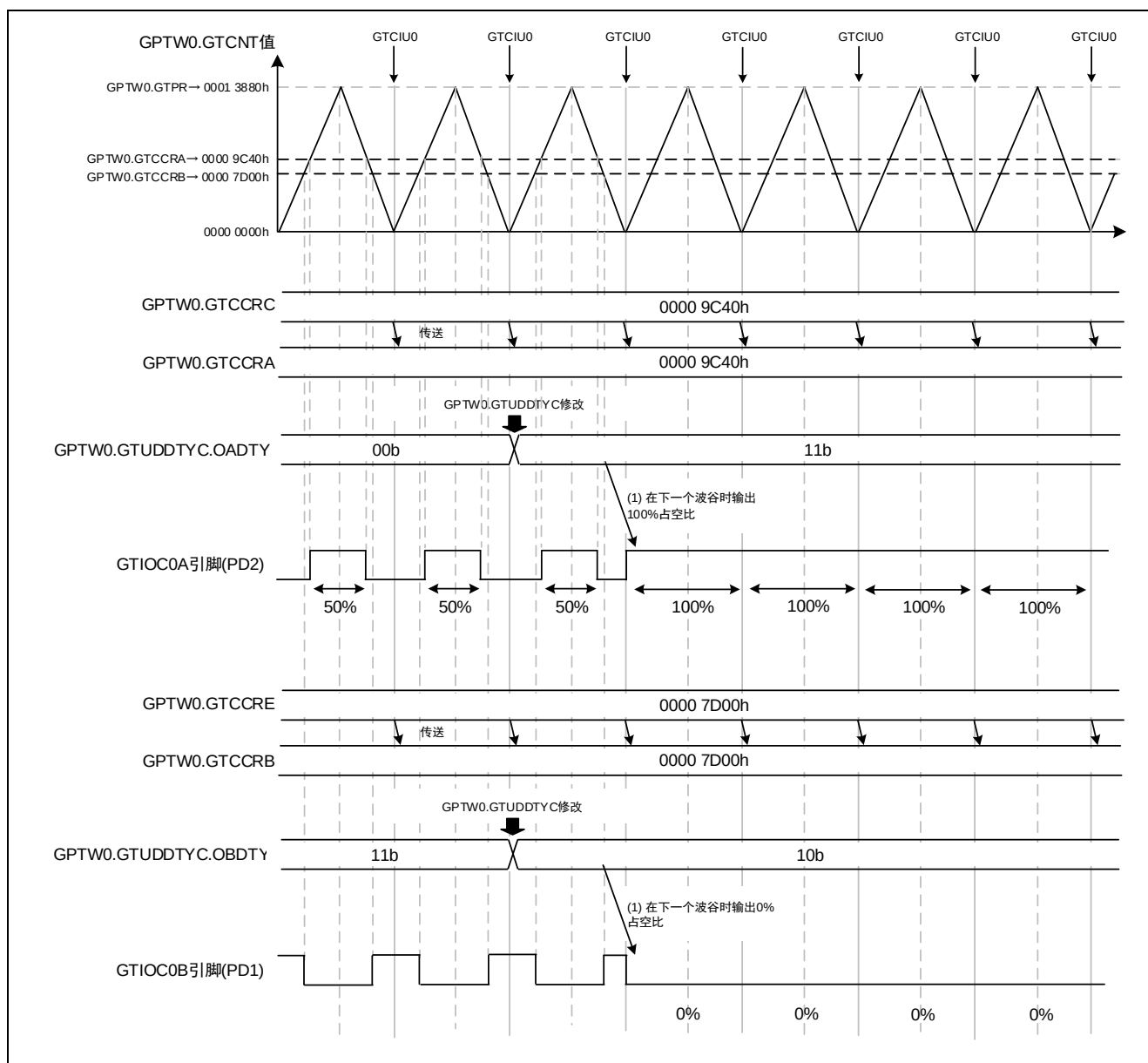


图 4.135 在保持 50%后更改占空比到 100%

在计数运行期间，不更改 GTUDDTYC.OADTY 和 GTUDDTYC.OBDTY 位的情况下，比较匹配时占空比不能切换到 100%。

4.13.5.6 在保持 50%后更改占空比到 0%

下面显示了在不更改 GTUDDTYC.OADTY 和 GTUDDTYC.OBDTY 位的情况下输出几个周期 50%占空比后切换到 0%占空比的运行示例。

当 GTCCRA 和 GTCCRB 寄存器的值设置为大于 GTPR 的值时，可以输出 0%占空比，因为没有发生比较匹配。

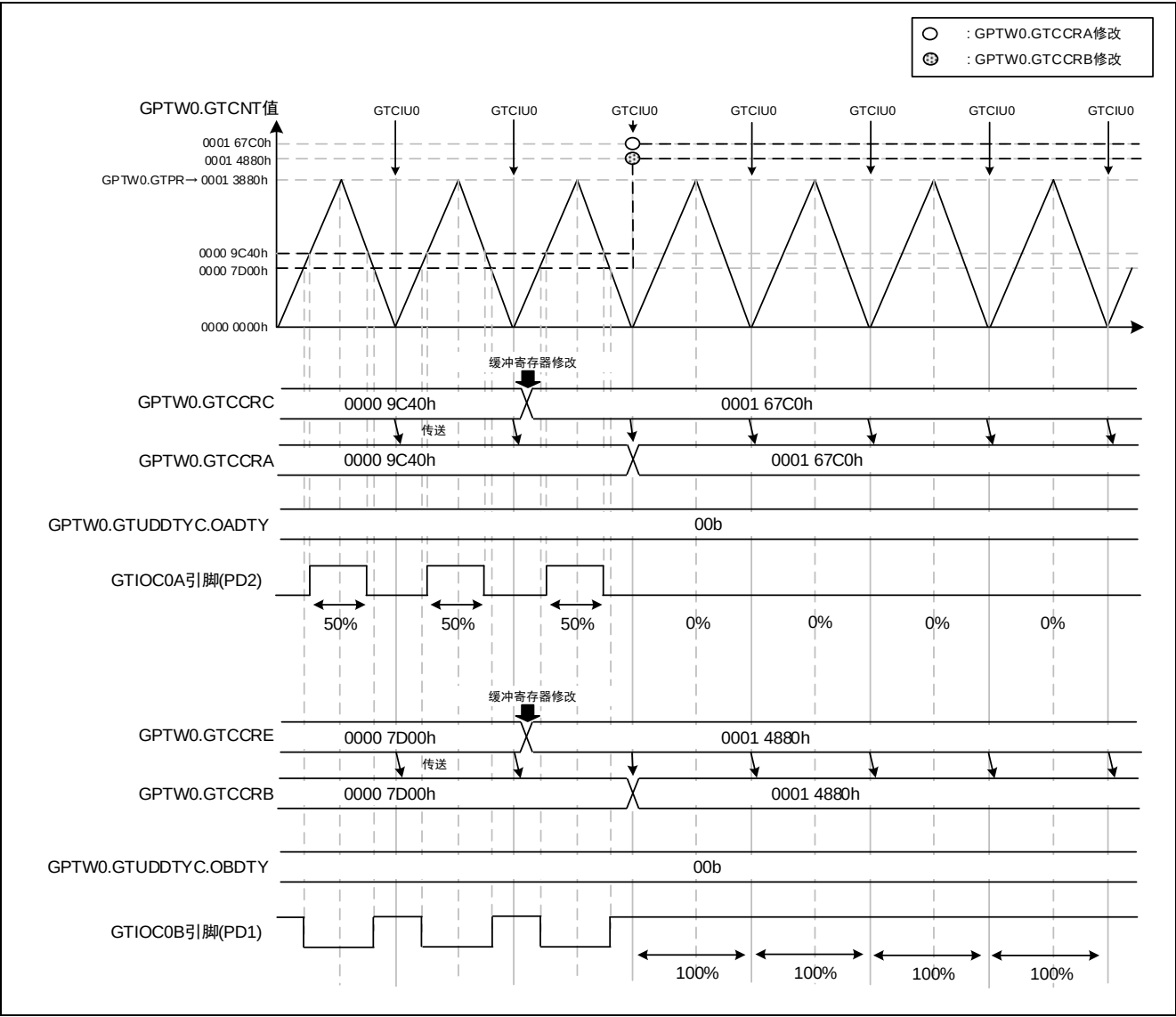


图 4.136 在保持 50%后更改占空比到 0%

在计数运行期间，GTUDDTYC.OADTY 和 GTUDDTYC.OBDTY 位分别设置为 10b 和 11b，以在下一个周期使能 0%占空比。

4.13.6 使用时的注意事项

4.13.6.1 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)

此示例代码自动设置死区时间，并使用比较寄存器 GTCCRA（正相波形寄存器）的值来更新比较匹配寄存器 GTCCRB（反相波形寄存器）的值。

设置比较寄存器 GTCCRA 的值以满足以下限制。

GTCCRA > GTDVU

GTCCRA > GTDVD 和

GTCCRA < GTPR

在计数运行期间，当 GTCCRA 寄存器设置为 0000 0000h 或大于等于 GTPR 的设置值，则输出保护功能被激活。

但是，如果不满足以下条件，输出保护不能正常运行。

当计数开始时的 GTCCRA 寄存器的值是 0000 0001h 或更大且小于 GTPR 寄存器的设置值时

有关输出保护功能的详细信息，请参考 RX66T Group User's Manual: Hardware, 章节 24.8.4 GTIOcnm 引脚输出保护功能 (n = 0 到 9; m = A, B)。

当不使用死区时间自动设置功能时，GTCCRA (GTCCRB) 寄存器的值设置为大于 0000 0001h 且小于 GTPR 寄存器的设置值。当 GTCCRA 寄存器中设置了 0000 0000h 或与 GTPR 寄存器相同的值时，仅当满足[GTCCRA (GTCCRB) 寄存器 = 0000 0000h]或[GTCCRA (GTCCRB) 寄存器 = GTPR 寄存器]时，才在一个周期中发生一次比较匹配。另外，如果 GTCCRA 被设置为超过 GTPR 寄存器设置值的值时，则不会发生比较匹配。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.10.2 比较匹配运行期间 GTCCRM 寄存器的设置 (m = A 到 F)，(1) 当三角波 PWM 模式下进行死区时间自动设置时和 (2) 当三角波 PWM 模式下未进行死区时间自动设置时。

4.13.6.2 计数开始时 GTUDDTYC.OADTY 设置的反映

在此示例代码中，计数运行的占空比取决于计数停止时的 GTUDDTYC.OADTYF 和 GTUDDTYC.OADTY 位的设置。

假如在 GTUDDTYC.OADTYF 位设为 0b 且计数停止时更改 GTUDDTYC.OADTY 位的值，计数开始时的更改的输出占空比设置不被反映。为了反映计数开始时的设置，需要在 GTUDDTYC.OADTYF 位设为 1b 且计数停止时更改 GTUDDTYC.OADTY 的值，然后开始计数。

详情参考 RX66T Group User's Manual: Hardware, 章节 24.3.6 占空比 0%/100%输出功能。

4.13.6.3 使用 Smart Configurator 在计数开始时占空比的反映

当使用 Smart Configurator 且引脚输出占空比设置为 0%或 100%时（图 4.111），占空比不会在计数开始时反映，因为 GTUDDTYC.OADTYF 位为 0b。为了反映计数开始时的占空比，用户必须创建代码在 GTUDDTYC.OADTYF 位为 1b 且计数停止时设置 GTUDDTYC.OADTY 位为 10b 或 11b。

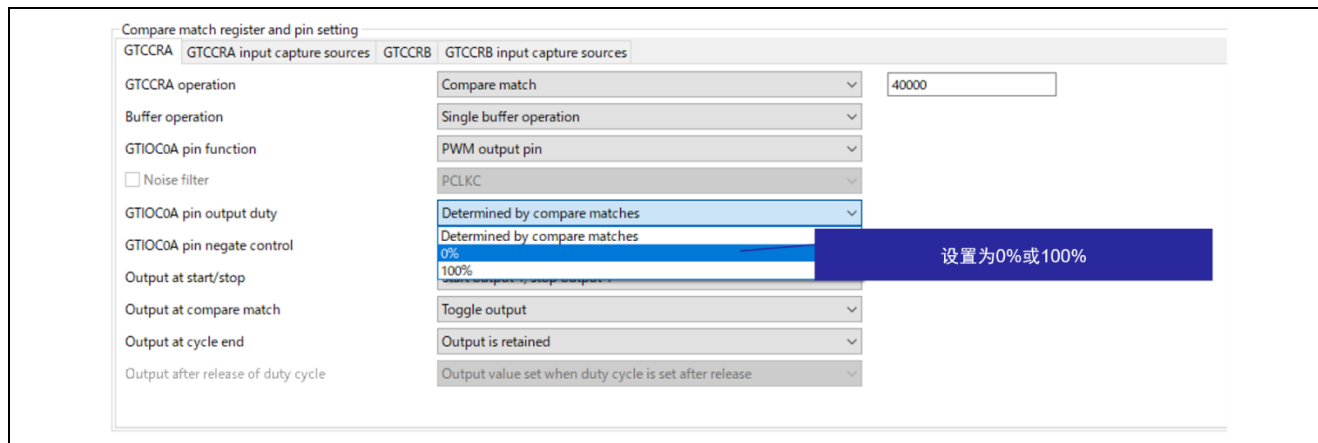


图 4.137 设置引脚输出占空比为 0%或 100%

有关 Smart Configurator 设置的详情，请参考章节 4.13.3 Smart Configurator 设置，以及用户生成代码的示例请参考章节 4.13.4 流程图。

4.13.6.4 比较匹配时的 100%占空比输出

在不改变 GTUDDTYC 寄存器的情况下，输出占空比在比较匹配时不能切换到 100%。为了输出 100%占空比，设置 GTUDDTYC.OADTY 位为 11b 以及 GTUDDTYC.OBDTY 位为 10b。

如果在此示例代码设置中将 GTRRCA 和 GTRRCB 寄存器设置为 0，则无法输出 100%占空比，因为在 GTCNT 计数器下溢发生后输出一个时钟周期。

当 GTCCRA 和 GTCCRB 寄存器设置为与 GTPR 相同的值并且同时发生 GTCNT 计数溢出和比较匹配时，比较匹配时间的输出设置被优先并且翻转，禁止 100%占空比输出。

有关 GPTW 中同时发生 GTCNT 定时器下溢和比较匹配时的波形输出的详情，请参考 RX66T Group User's Manual: Hardware, 章节 24.2.14 通用 PWM 定时器 I/O 控制寄存器 (GTIOR) 中表 24.4 下的注释。

4.13.6.5 占空比 0%和 100%输出期间比较匹配运行

在此示例代码中，输出占空比 0%和 100%是基于 GTUDDTYC.OADTY 位的设置值决定的。当占空比设置为 0%或 100%时，GPTW 内部继续执行比较匹配运行，中断输出和缓冲传送运行。

此示例代码不使用比较匹配中断，因此如果你要使用比较匹配中断，请在占空比 0%和 100%输出期间谨慎使用中断。

5 如何导入项目

示例代码是以 e² studio 工程的格式提供。本章描述了如何将项目导入 e² studio 和 CS+。在导入完成后，确认编译和调试器设置。

5.1 使用 e² studio 导入

在 e² studio 中使用示例代码时，请使用以下步骤将其导入 e² studio。

(实际屏幕可能因您使用的 e² studio 版本而异。)

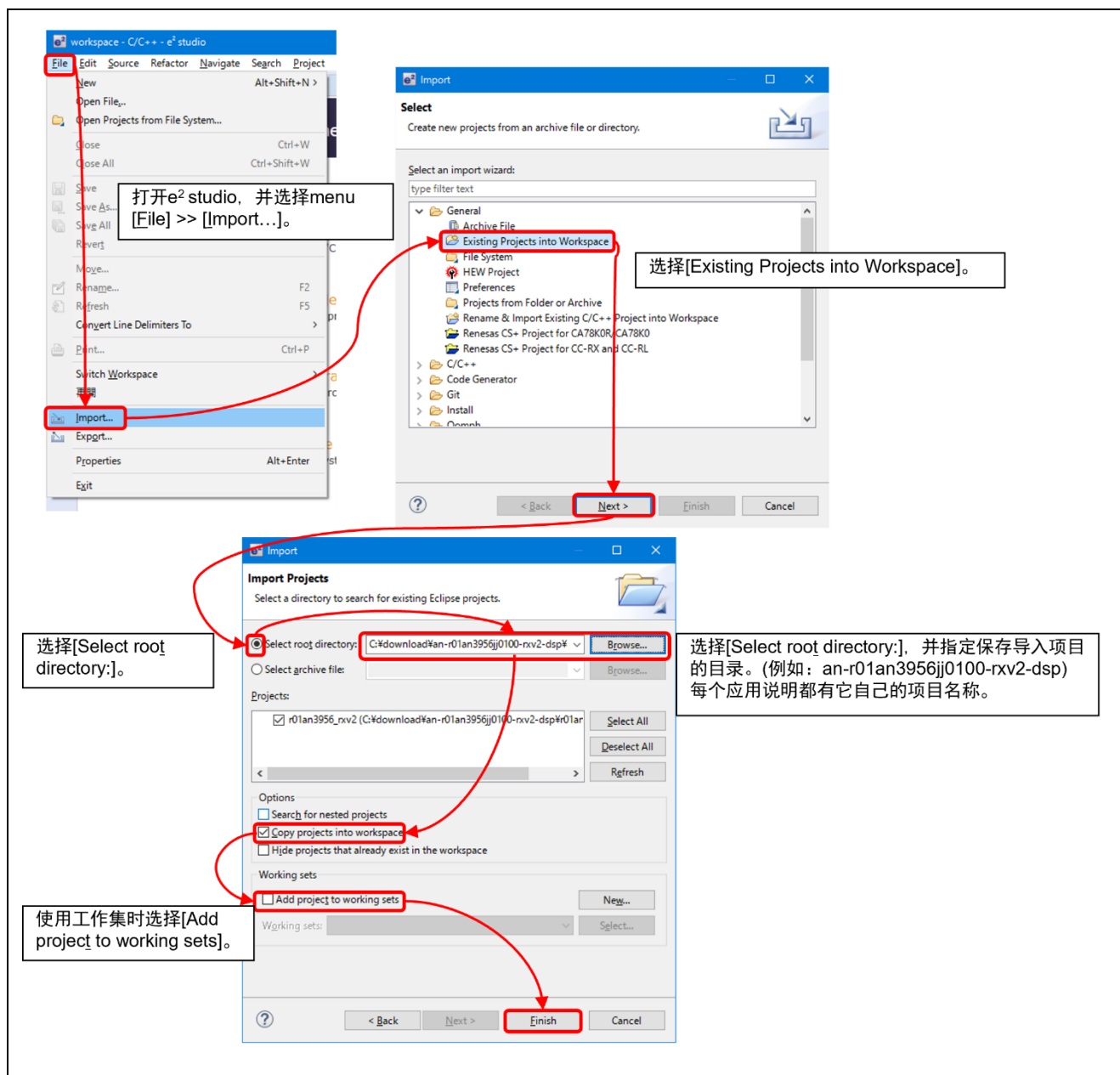


图 5.1 如何将项目导入 e² studio

5.2 使用 CS+导入

在 CS+中使用示例代码时, 请使用以下步骤将其导入 CS+。

(实际屏幕可能因您使用的 CS+版本而异。)

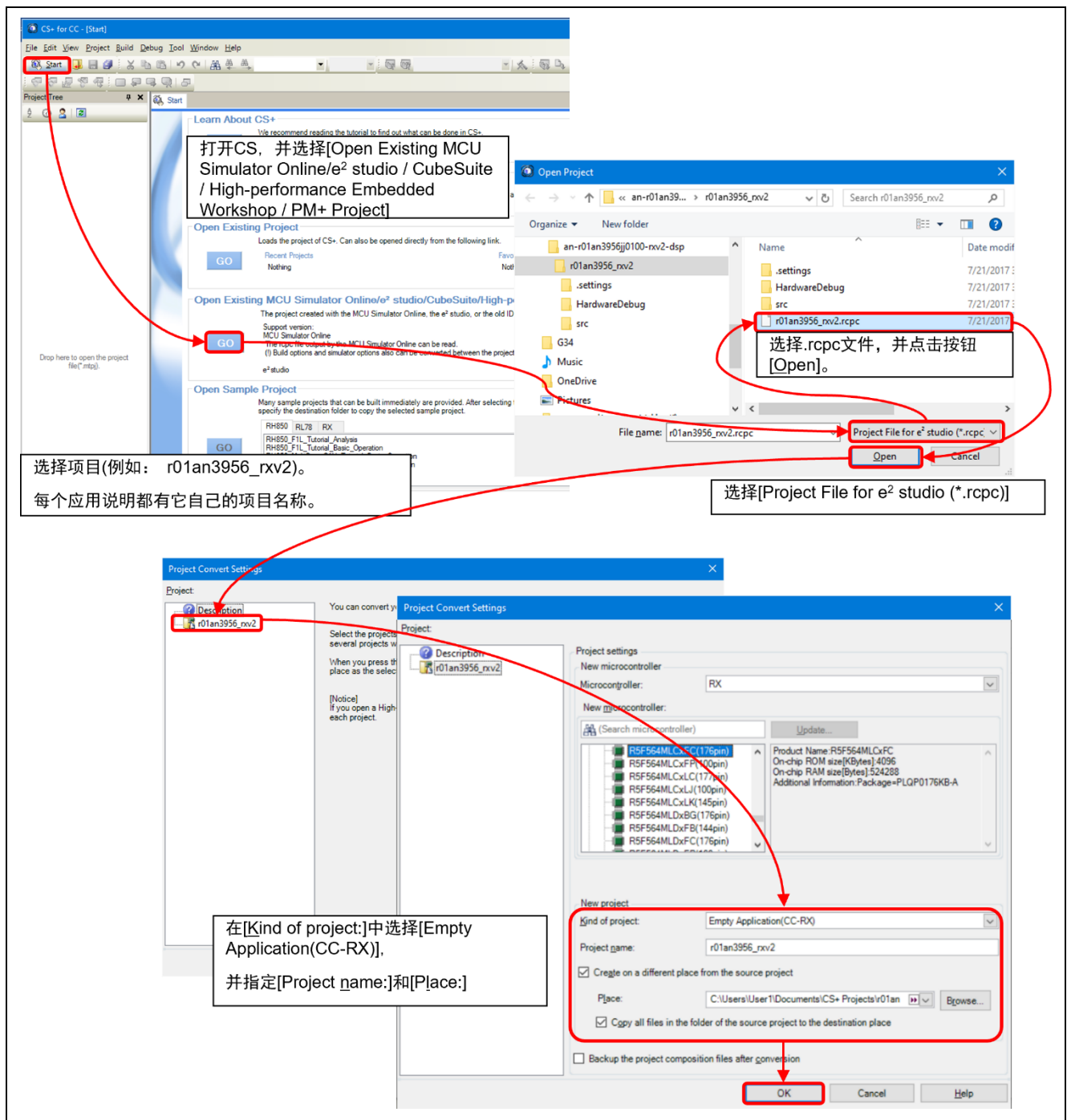


图 5.2 如何将项目导入 CS+

6 参考文献

用户手册：硬件

RX66T Group User's Manual: Hardware (R01UH0749)

(请从瑞萨电子网站获取最新版本。)

技术更新/技术新闻

(请从瑞萨电子网站获取最新版本。)

用户手册：开发环境

RX Family CC-RX Compiler User's Manual (R20UT3248)

(请从瑞萨电子网站获取最新版本。)

用户手册：开发环境

RX66T Group Renesas Starter Kit User's Manual (R20UT4150)

(请从瑞萨电子网站获取最新版本。)

更新履历

版本号	日期	描述	
		页数	概要
1.00	2023 年 2 月 10 日	-	初版发布
1.10	2023 年 6 月 10 日	26	修改“表 1.14 占空比 0%和 100%的设置方法”。
		35	修改“图 3.5 中断设置”。
		94, 105, 148, 161	修改“输出电平设置”。
		173	修改“图 4.9 GPT0 设置 (1/2)”。
		182	修改“图 4.17 GPT0 设置 (1/2)”。
		194	修改“图 4.26 GPT0 设置”。

注意

1. 本文档中电路、软件和其他相关信息的描述仅用于说明半导体产品的操作和应用示例。用户应对产品或系统设计中电路、软件和信息纳入或任何其他用途承担全部责任。对于您或第三方因使用这些电路、软件或信息而引起的任何损失和损害，Renesas Electronics 不承担任何责任。
2. Renesas Electronics 特此声明，对于因使用本文档中所述的 Renesas Electronics 产品或技术信息（包括但不限于产品数据、图纸、图表、程序、算法和应用示例）而引起的侵权或与第三方有关的专利、版权或其他知识产权的任何其他索赔，概不承担任何责任和赔偿。
3. 对 Renesas Electronics 或其他公司的任何专利、版权或其他知识产权均不授予任何明示、暗示或其他形式的许可。
4. 您应负责确定需要从任何第三方获得哪些许可，并在需要时为合法进口、出口、制造、销售、使用、分销或以其他方式处置包含 Renesas Electronics 产品的任何产品获得此类许可。
5. 不得对 Renesas Electronics 产品的全部或部分进行更改、修改、复制或逆向工程。对于因更改、修改、复制或逆向工程而导致您或第三方蒙受的任何损失或损害，Renesas Electronics 不承担任何责任。
6. Renesas Electronics 产品根据以下两个质量等级进行分类：“标准”和“优质”。Renesas Electronics 每种产品的预期应用取决于产品的质量等级，具体如下所示。

“标准”：计算机、办公设备、通信设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备、工业机器人等

“优质”：运输设备（汽车、火车、轮船等）；交通管制（交通信号灯）；大型通信设备；关键金融终端系统；安全控制设备等

除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中明确指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 产品不适合或不授权用于可能对人类生命构成直接威胁或造成人身伤害（人造生命支持设备或系统；手术植入物等），或者可能造成严重的财产损失（空间系统、海底中继器、核动力控制系统、飞机控制系统、关键设备系统、军事装备等）的产品或系统。对于因使用任何与 Renesas Electronics 数据手册、用户手册或其他 Renesas Electronics 文档不一致的 Renesas Electronics 产品而引起的您或任何第三方所造成的任何损坏或损失，Renesas Electronics 不承担任何责任。
7. 没有任何半导体产品是绝对安全的。尽管 Renesas Electronics 的硬件或软件产品中可能实施了任何安全措施或功能，Renesas Electronics 对因任何漏洞或侵袭（包括但不限于以任何未经授权的方式访问或使用 Renesas Electronics 产品或使用 Renesas Electronics 产品的系统）而产生的任何后果概不负责。RENESAS ELECTRONICS 不担保或保证 RENESAS ELECTRONICS 产品或使用 RENESAS ELECTRONICS 产品创建的任何系统不会被破坏，或者可免于数据损坏、攻击、病毒、干扰、黑客攻击、数据丢失或失窃或其他安全入侵（“漏洞问题”）。RENESAS ELECTRONICS 不承担任何由任何漏洞问题引起的或与之相关的任何和所有责任或义务。此外，在适用法律允许的范围内，RENESAS ELECTRONICS 不对本文件和任何相关或附带的软件或硬件提供任何和所有明示或暗示的保证，包括但不限于对适销性或对特定用途的适用性的暗示保证。
8. 使用 Renesas Electronics 产品时，请参见最新的产品信息（数据手册、用户手册、应用笔记、可靠性手册中的“处理和使用半导体器件的一般说明”等），并确保使用条件符合 Renesas Electronics 在最大额定值、工作电源电压范围、散热特性和安装等方面的规定。对于因在超出上述规定范围的条件上使用 Renesas Electronics 产品而引起的任何失常、故障或事故，Renesas Electronics 不承担任何责任。
9. 尽管 Renesas Electronics 致力于提高 Renesas Electronics 产品的质量和可靠性，但半导体产品具有特定的特性，例如在特定速率下发生故障以及在某些使用条件下出现故障。除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 的产品将不受抗辐射设计的约束。用户应负责采取安全措施，以防止人身伤害、火灾造成的伤害，和/或因 Renesas Electronics 产品发生故障或失常而对公众造成的危险，例如硬件和设备的安全设计，包括但不限于冗余、火控和故障预防、针对老化退化的适当处理或任何其他适当的措施。由于对微型计算机软件进行评估非常困难且无实操性，因此用户有责任评估自己生产的最终产品或系统的安全性。
10. 请联系 Renesas Electronics 销售办事处，以获取有关环境事宜的详细信息，例如每个 Renesas Electronics 产品的环境相容性。用户有责任认真、充分地研究有关纳入或使用受控物质的适用法律和法规（包括但不限于欧盟 RoHS 指令），并按照所有适用法律和法规使用 Renesas Electronics 产品。对于因您未遵守适用的法律和法规而造成的损坏或损失，Renesas Electronics 不承担任何责任。
11. Renesas Electronics 产品和技术不得被用于或纳入为任何适用的本国或外国法律、法规所禁止制造、使用或销售的产品或系统范围内。用户应遵守由对当事方或交易拥有管辖权的任何国家/地区的政府颁布和管理的任何可适用的出口控制法律和法规。
12. 应由 Renesas Electronics 产品的购买方或分销商，或者对产品进行分发、处置或以其他方式出售或转让给第三方的任何其他当事方，负责将本文档中阐明的内容和条件提前通知前述第三方。
13. 未经 Renesas Electronics 事先书面同意，不得以任何形式全部或部分重印、再现或复制本文档。
14. 如果对本文档中包含的信息或 Renesas Electronics 产品有任何疑问，请联系 Renesas Electronics 销售办事处。

（注 1）本文档中的“Renesas Electronics”是指 Renesas Electronics Corporation，也包括其直接或间接控制的子公司。

（注 2）“Renesas Electronics 产品”是指 Renesas Electronics 开发或制造的任意产品。

（版本 5.0-1 2020 年 10 月）

公司总部

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

商标

Renesas 和 Renesas 徽标是 Renesas Electronics Corporation 的商标。所有商标和注册商标都是各自所有者的财产。

联系信息

有关产品、技术、文档最新版本或离您最近的销售办事处的更多信息，请访问：www.renesas.com/contact/。