

# BEFORE

```

20001E38      .word 0x20001E38
--- main.c --- 576 ---
static void app_pwm_init(void)
{
0003015C      B500      push {lr}
0003015E      B085      sub sp, sp, #20
--- main.c --- 586 ---
/* a way that increase and decrease
* continuously on succeeding chanr
*/
nrf_drv_pwm_config_t const config0
00030160      4A10      ldr r2, =0x00034DC
00030162      AB01      add r3, sp, #4
00030164      CA07      ldm r2, {r0-r2}
00030166      E8830007  stm r3, {r0-r2}
--- main.c --- 603 ---
.load_mode = NRF_PWM_LOAD_INDIVIDU
.step_mode = NRF_PWM_STEP_AUTO
};
//APP_ERROR_CHECK(
nrf_drv_pwm_init(&m_pwm0, &config0
0003016A      AB01      add r3, sp, #4
0003016C      4A0F      ldr r2, =0x000300F

```

```

main.c sdk_config.h nrf_pwm.h app_error_weak.c nrfx_pwm.c nrfx_pwm.h app_error_handler_gcc.c
apply_old_config.h nrfx_prs.c
void app_pwm_init0
SER_APP_SPIM0_SCK_PIN | NRF_DRV_PWM_PIN_INVERTED, // channel 0
NRF_DRV_PWM_PIN_NOT_USED, // channel 1
NRF_DRV_PWM_PIN_NOT_USED, // channel 2
NRF_DRV_PWM_PIN_NOT_USED, // channel 3
},
.irq_priority = APP_IRQ_PRIORITY_LOWEST,
.base_clock = NRF_PWM_CLK_1MHz,
.count_mode = NRF_PWM_MODE_UP,
.top_value = 53125,
.load_mode = NRF_PWM_LOAD_INDIVIDUAL,
.step_mode = NRF_PWM_STEP_AUTO
};
//APP_ERROR_CHECK(
nrf_drv_pwm_init(&m_pwm0, &config0, pwm0_handler);//);
m_used |= USED_PWM(0);
610 p_channels[0] = servo_cur_value; /* Open */
(void)nrf_drv_pwm_simple_playback(&m_pwm0, &m_pwm_seq, 1,
NRF_PWM_FLAG_LOOP);

```

Expression	Value
m_pwm0	<struct>
p_registers	0x4001c000
drv_inst_idx	0x00
config0	<struct>
output_pins	"\233\377\377"
[0]	0x9b
[1]	0xff
[2]	0xff
[3]	0xff
irq_priority	0x07
base_clock	NRF_PWM_CLK_1MHz
count_mode	NRF_PWM_MODE_UP
top_value	0xcfc85
load_mode	NRF_PWM_LOAD_INDIVIDUAL
step_mode	NRF_PWM_STEP_AUTO

ble\_app\_blinky\_c\_pca10056\_s140 - SEGGER Embedded Studio for ARM V4.52c (64-bit) - Licensed to Ashok ra - Embedded Club (Stopped)

# AFTER

```

app_error_fault_handler + 0:
00028E00      4E1E      ldr r3, =0x000338E
00028E02      1A13      subs r3, r2, r3
00028E04      08DB      lsrs r3, r3, #3
00028E06      041B      lsls r3, r3, #16
00028E08      F0430301  orr r3, r3, #1
00028E0A      9A06      ldr r2, [sp, #24]
00028E0C      4911      ldr r1, =0x00033D6
00028E0E      4618      mov r0, r3
00028E10      F7FEFFC7  bl 0x00027DC4 <nrf
--- app_error_weak.c --- 96 ---
break;
00028E12      BF00      nop
--- app_error_weak.c --- 97 ---
}
#endif
NRF_BREAKPOINT_COND;
00028E14      4B0F      ldr r3, =0xE00EDF
00028E16      681B      ldr r3, [r3]
00028E18      F0030301  and r3, r3, #1
00028E1A      2B00      cmp r3, #0
00028E1C      D000      beq 0x00028E46
00028E1E      BE00      bkpt #0
--- app_error_weak.c --- 103 ---
#ifdef DEBUG
NRF_LOG_WARNING("System reset");
NVIC_SystemReset();
#else
app_error_save_and_stop(id, pc, ir
00028E20      9A05      ldr r2, [sp, #20]
00028E22      9906      ldr r1, [sp, #24]
00028E24      9807      ldr r0, [sp, #28]
00028E26      F7FFFF08  bl 0x00028C60 <app
--- app_error_weak.c --- 108 ---
#endif // DEBUG

```

```

main.c sdk_config.h nrf_pwm.h app_error_weak.c nrfx_pwm.c nrfx_pwm.h app_error_handler_gcc.c apply_old_config.h
nrfx_prs.c
void app_error_fault_handler(unsigned int id=0x00000001, unsigned int pc=0x00
90      nrf_strerror_get(p_info->err_code),
p_info->p_file_name,
p_info->line_num,
pc);
NRF_LOG_ERROR("End of error report");
break;
}
default:
NRF_LOG_ERROR("UNKNOWN FAULT at 0x%08X", pc);
break;
}
#endif
100 NRF_BREAKPOINT_COND;
// On assert, the system can only recover with a reset.
#ifdef DEBUG
NRF_LOG_WARNING("System reset");
NVIC_SystemReset();
#else
app_error_save_and_stop(id, pc, info);
#endif // DEBUG
}
110 /*lint -restore */

```

Expression	Value
m_pwm0	<struct>
p_registers	0x4001c000
drv_inst_idx	0x00
config0	<struct>
output_pins	"\360\021"
[0]	0xf0
[1]	0x11
[2]	0x00
[3]	0x20
irq_priority	0x6b
base_clock	NRF_PWM_CLK_16MHz
count_mode	NRF_PWM_MODE_UP
top_value	0x7edf
load_mode	0x84
step_mode	0x2b

Output

Show: Target Output

Programming 149.3 KB of addresses 00001000 - 00026597  
l-link: Flash download: Bank 0 @ 0x00000000: 1 range affe

Call Stack

Function	Call Address
void app_error_fault_handler(unsigned int id=0x00000001, unsigned int pc=0x00028E44)	0x00028E44