Obstacle Avoiding Car System Design

By : Team 4

Contents

Flowchart Table

Detailed Requirements

System Requirements:

1. The car starts initially from 0 speed
2. The default rotation direction is to the right
3. Press PB2 to start or stop the robot
4. After Pressing Start:
    1) The LCD will display a centered message in line 1 “Set Def. Rot.”
    2) The LCD will display the selected option in line 2 “Right”
    3) The robot will wait for 5 seconds to choose between Right and Left
        (1) When PB1 is pressed once, the default rotation will be Left and the LCD line 2 will be updated
        (2) When PB1 is pressed again, the default rotation will be Right and the LCD line 2 will be updated
        (3) For each press, the default rotation will be changed and the LCD line 2 is updated
        (4) After the 5 seconds, the default value of rotation is set
    4) The robot will move after 2 seconds from setting the default direction of rotation.
    5) For No obstacles or object is far than 70 centimeters:
        (1) The robot will move forward with 30% speed for 5 seconds
        (2) After 5 seconds it will move with 50% speed as long as there was no object or objects are located at more than 70 centimeters distance
        (3) The LCD will display the speed and moving direction in line 1: “Speed:00% Dir: F/B/R/S”, F: forward, B: Backwards, R: Rotating, and S: Stopped
        (4) The LCD will display Object distance in line 2 “Dist.: 000 Cm”
    6) For Obstacles located between 30 and 70 centimeters
        (1) The robot will decrease its speed to 30%
        (2) LCD data is updated
    7) For Obstacles located between 20 and 30 centimeters
        (1) The robot will stop and rotates 90 degrees to right/left according to the chosen configuration
        (2) The LCD data is updated
    8) 8. For Obstacles located less than 20 centimeters
        (1) The robot will stop, move backwards with 30% speed until distance is greater than 20 and less than 30
        (2) The LCD data is updated
        (3) Then preform point 8
    9) Obstacles surrounding the robot (Bonus)
        (1) If the robot rotated for 360 degrees without finding any distance greater than 20 it will stop
        (2) LCD data will be updated.
        (3) The robot will frequently (each 3 seconds) check if any of the obstacles was removed or not and move in the direction of the furthest object

Layered architecture

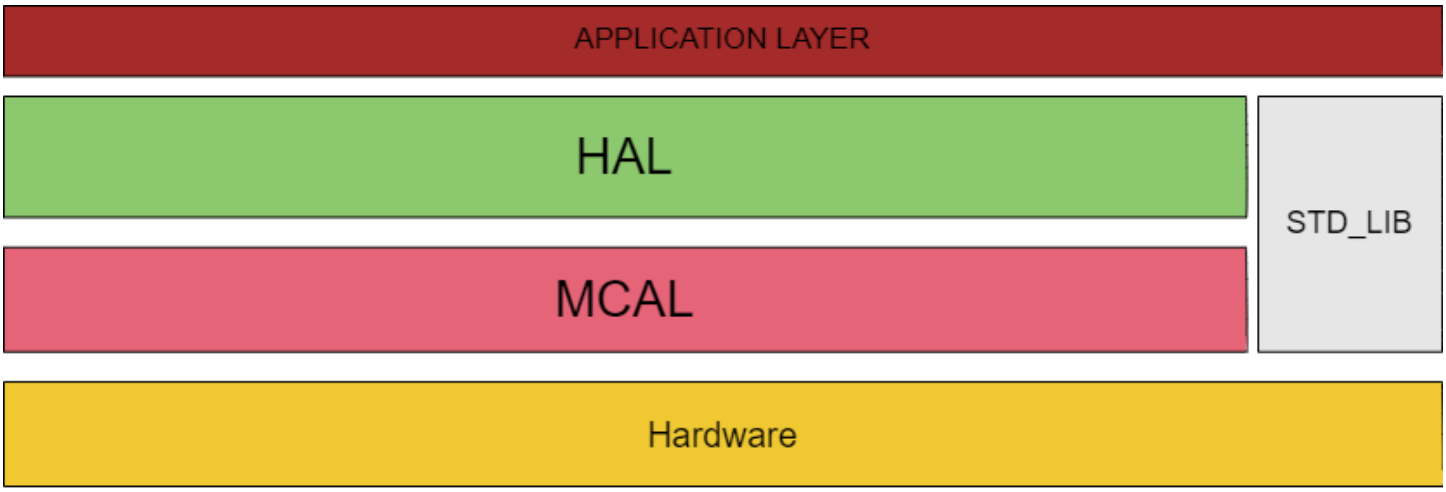

Figure 1 Desgin_arch

System moduls

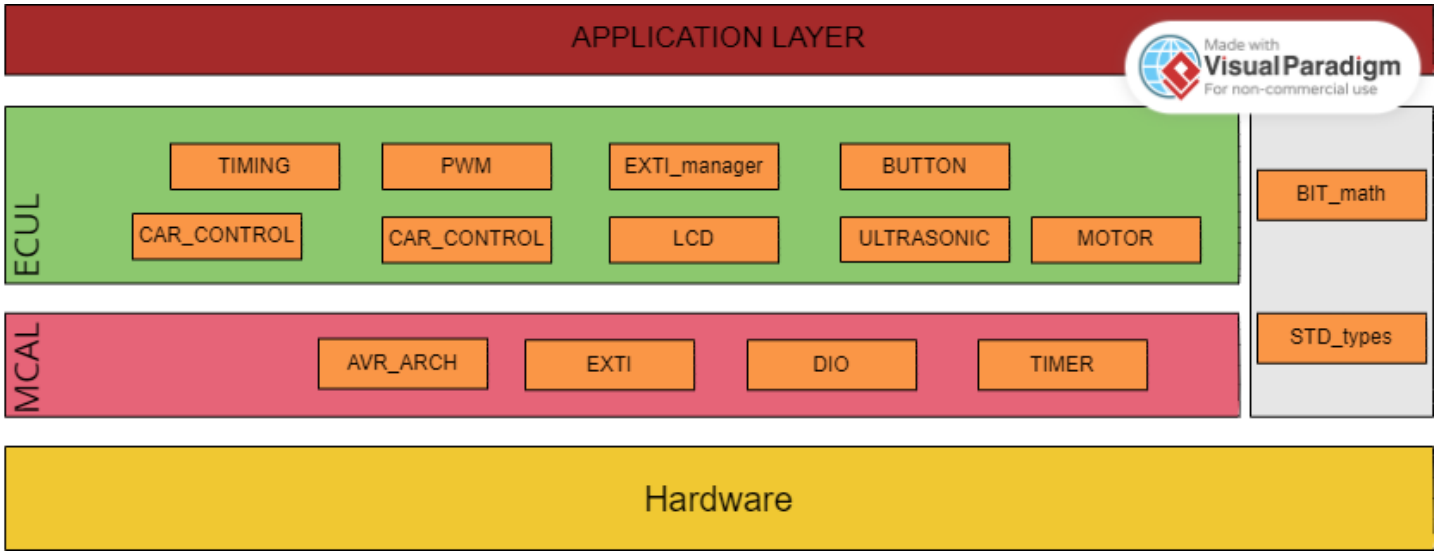

Figure 2 Obstical_Avoiding_Car_Design_Modules

MCAL

DIO API :

Flowcharts:

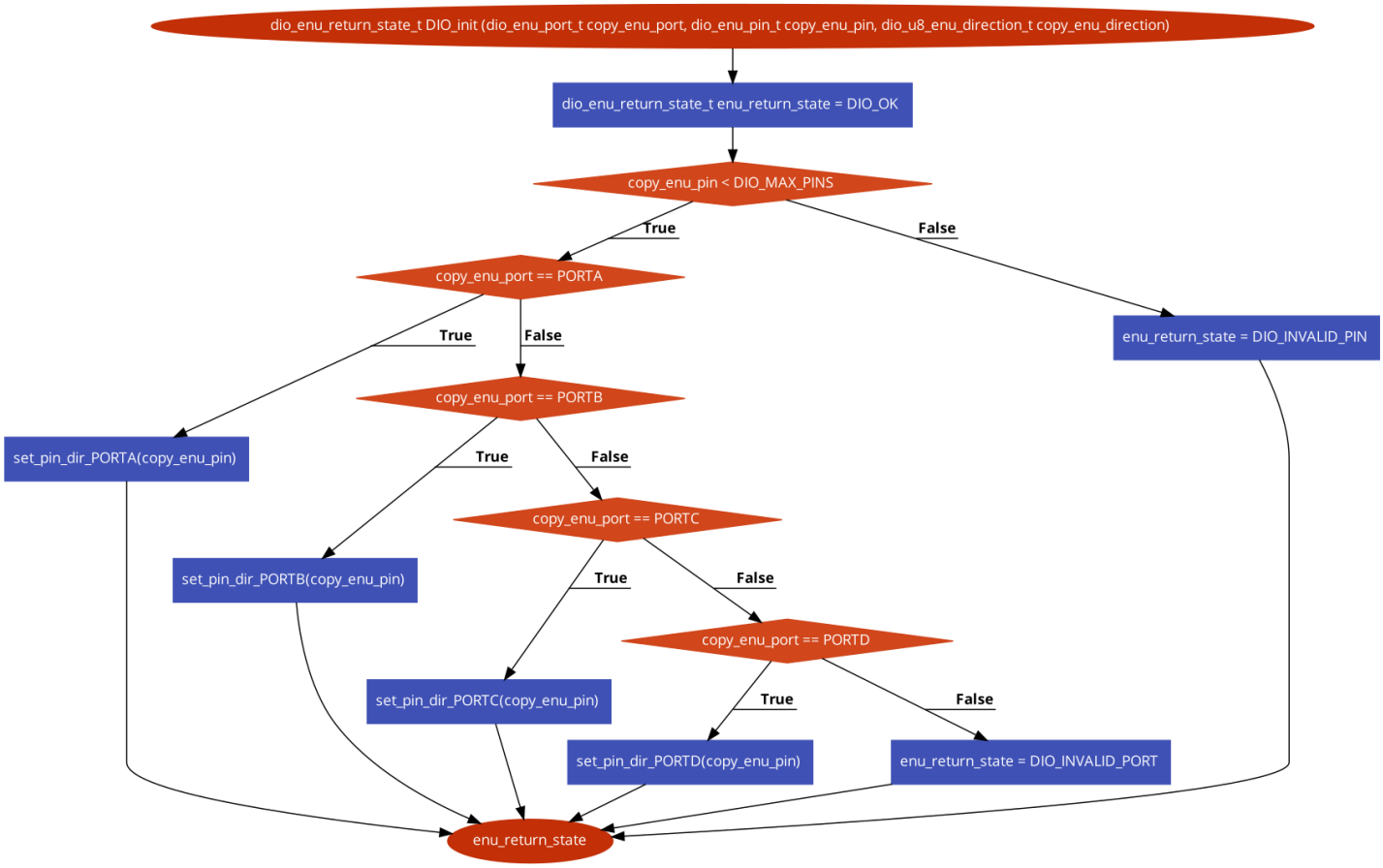

Figure 3 DIO_init()

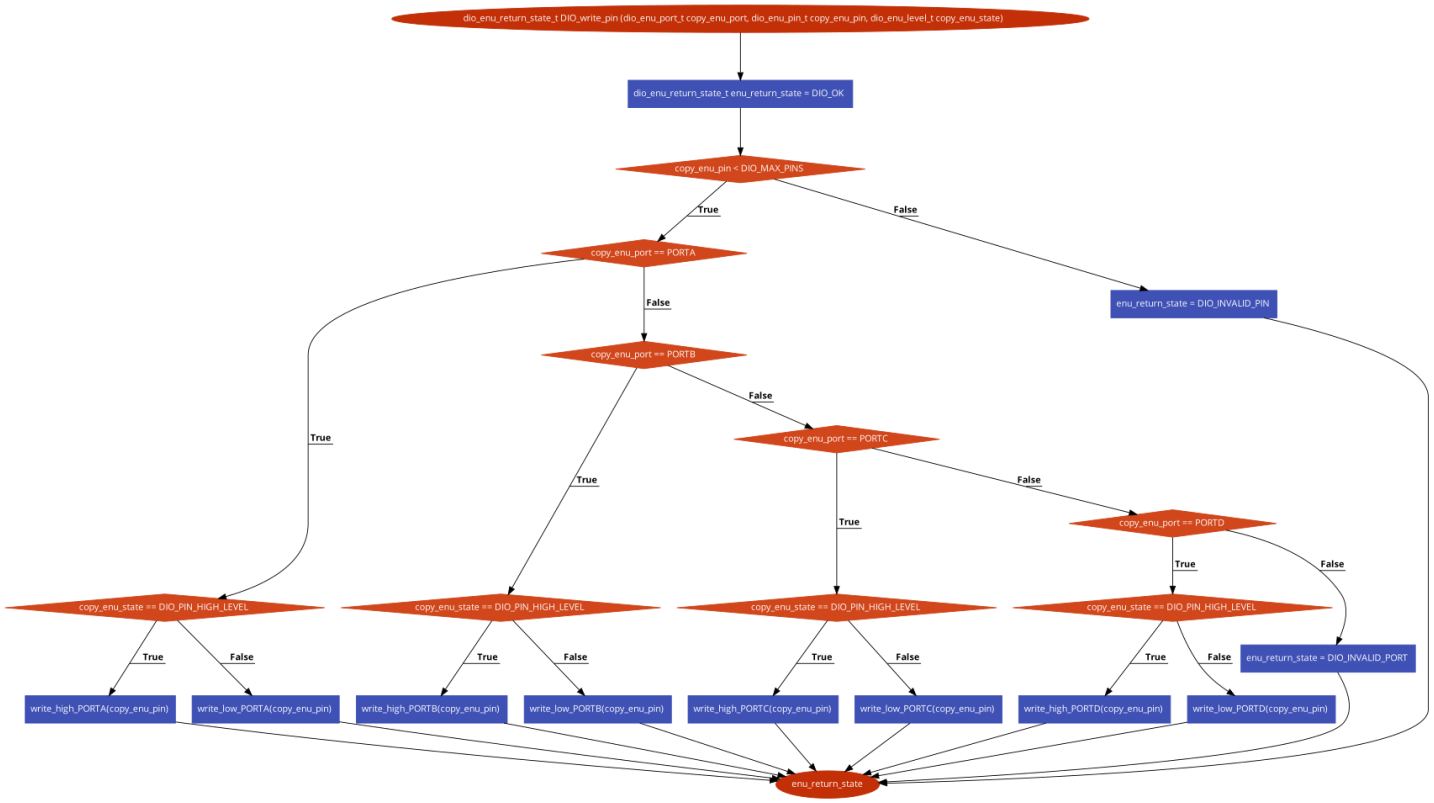

Figure 4 DIO_write_pin()

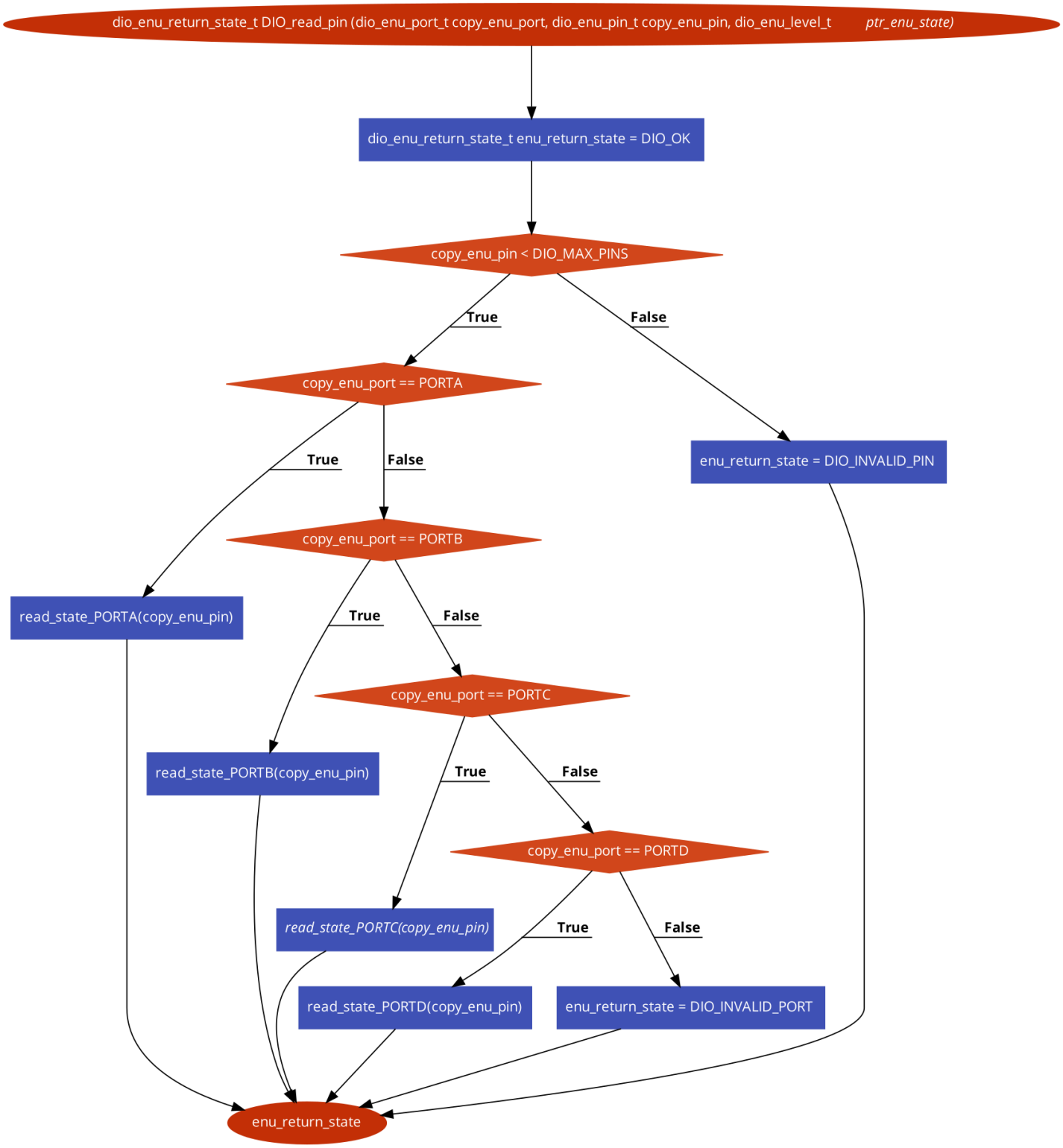

Figure 5 DIO_read_pin()

External Interrupt API :

Flowcharts:

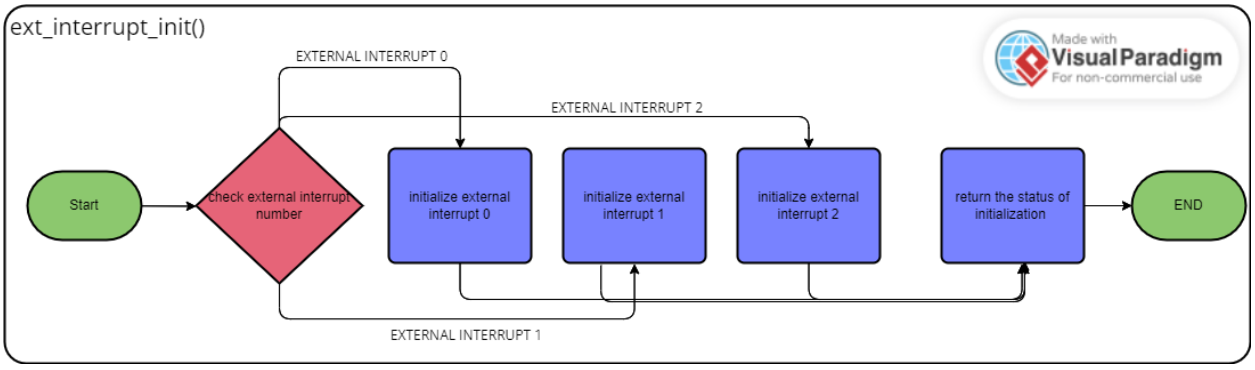

Figure 6 ext_interrupt_int()

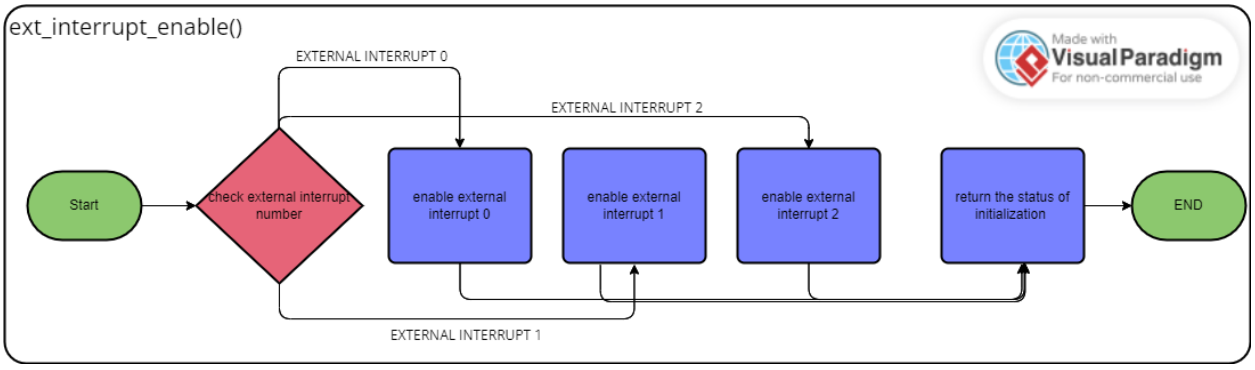

Figure 7 ext_interrupt_enable()

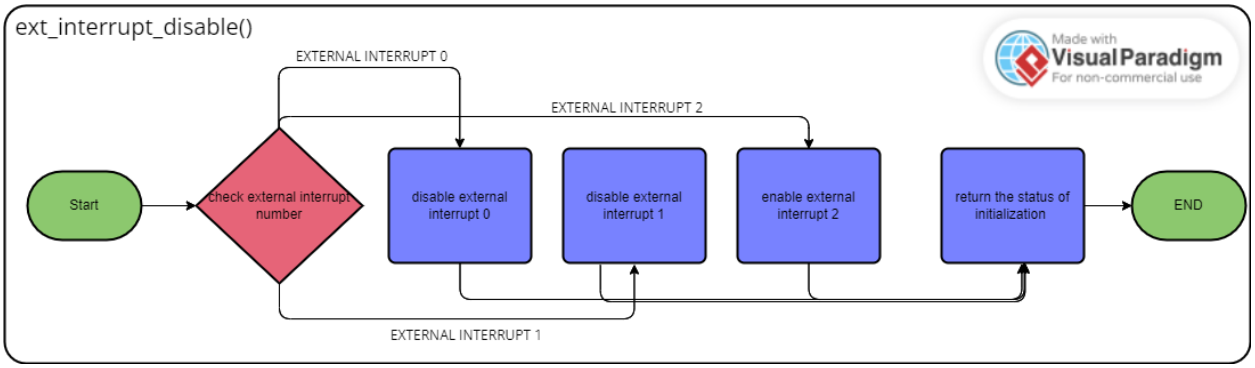

Figure 8 ext_interrupt_disable()

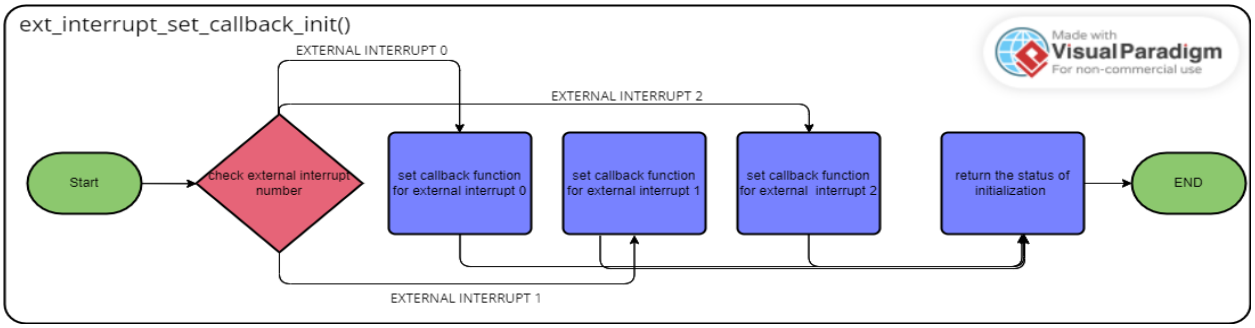

Figure 9 ext_interrupt_set_callback_init()

TIMER API :

Flowcharts:

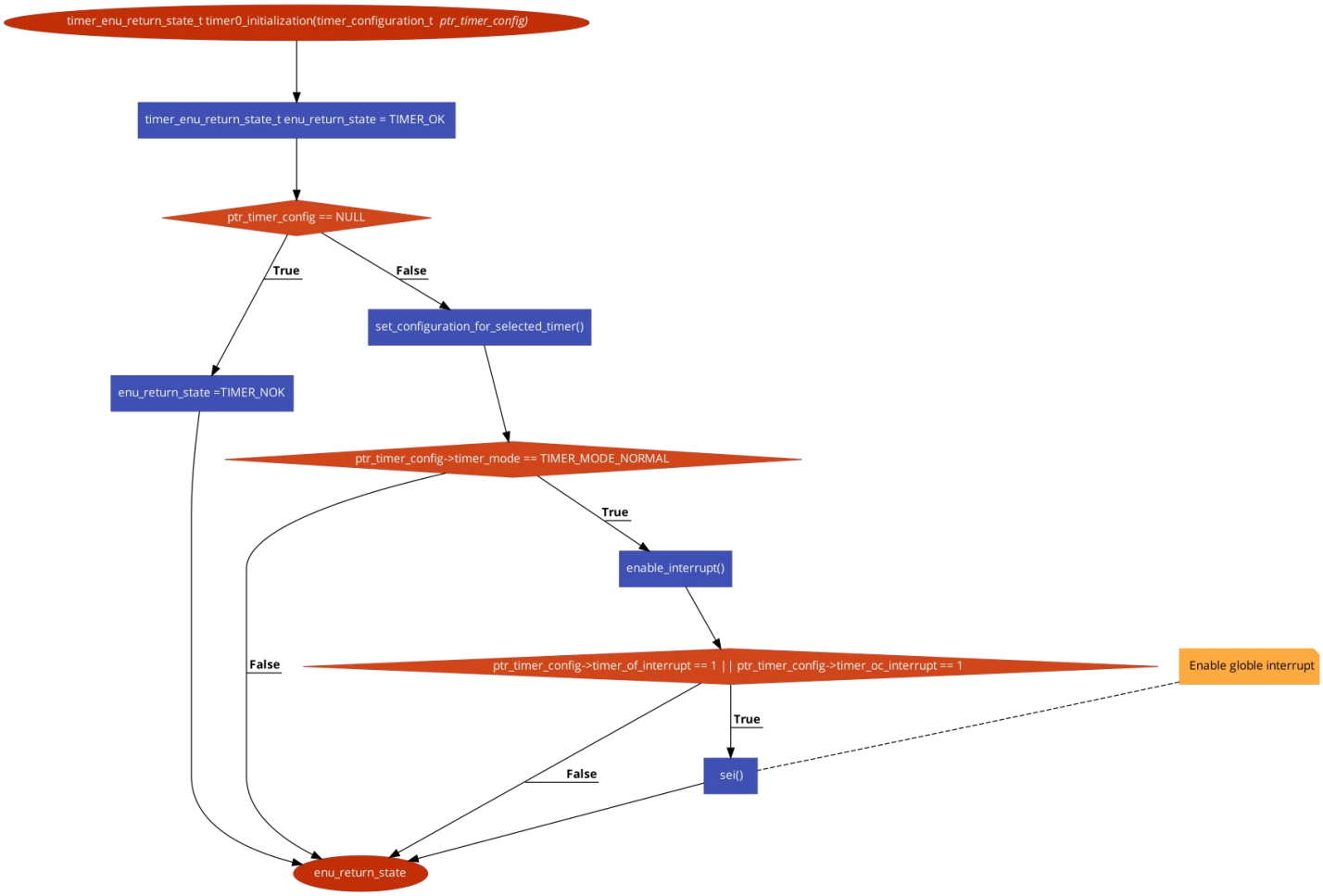

Figure 10 timer_x_initialization()

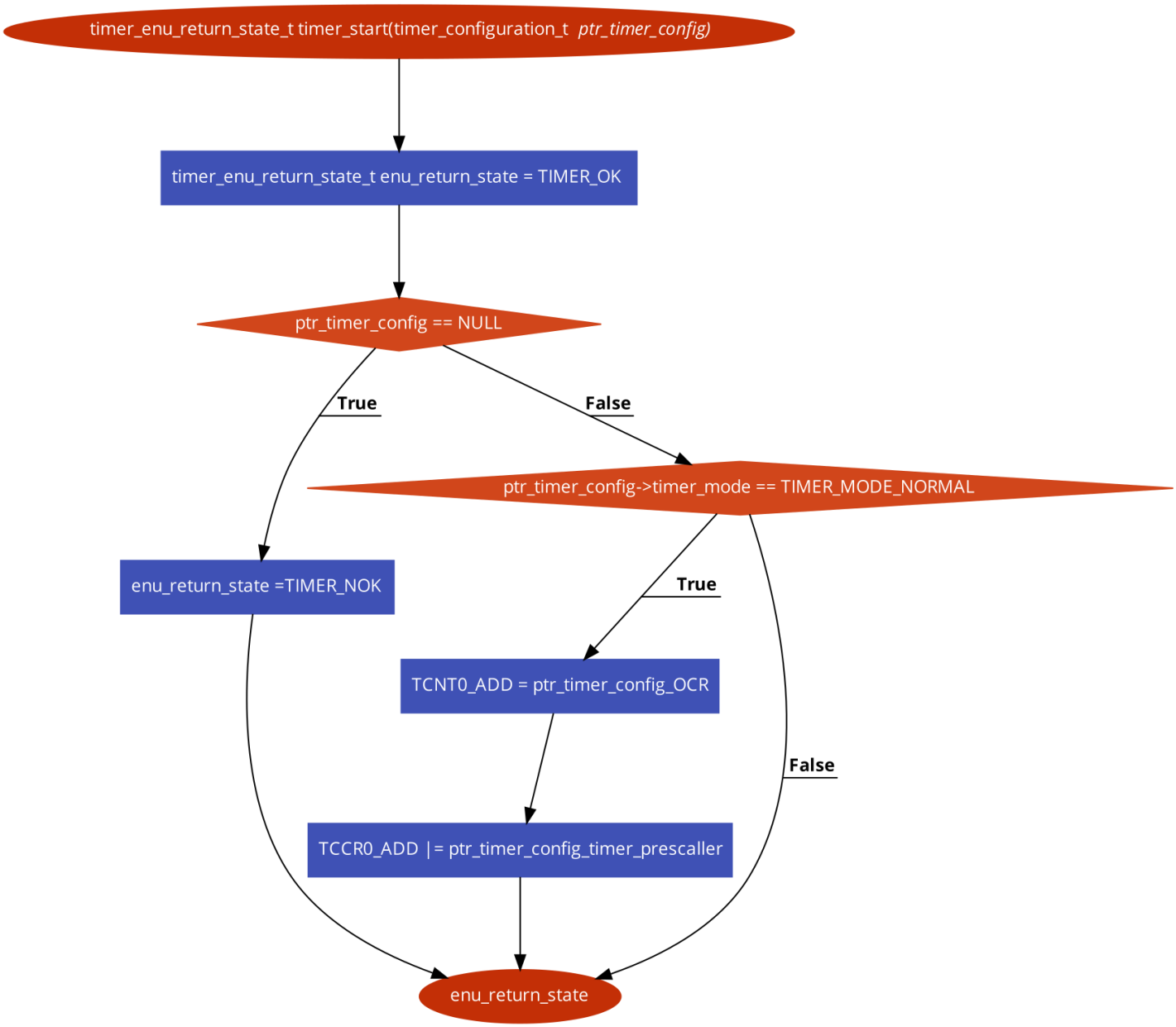

Figure 11 timer_x_start()

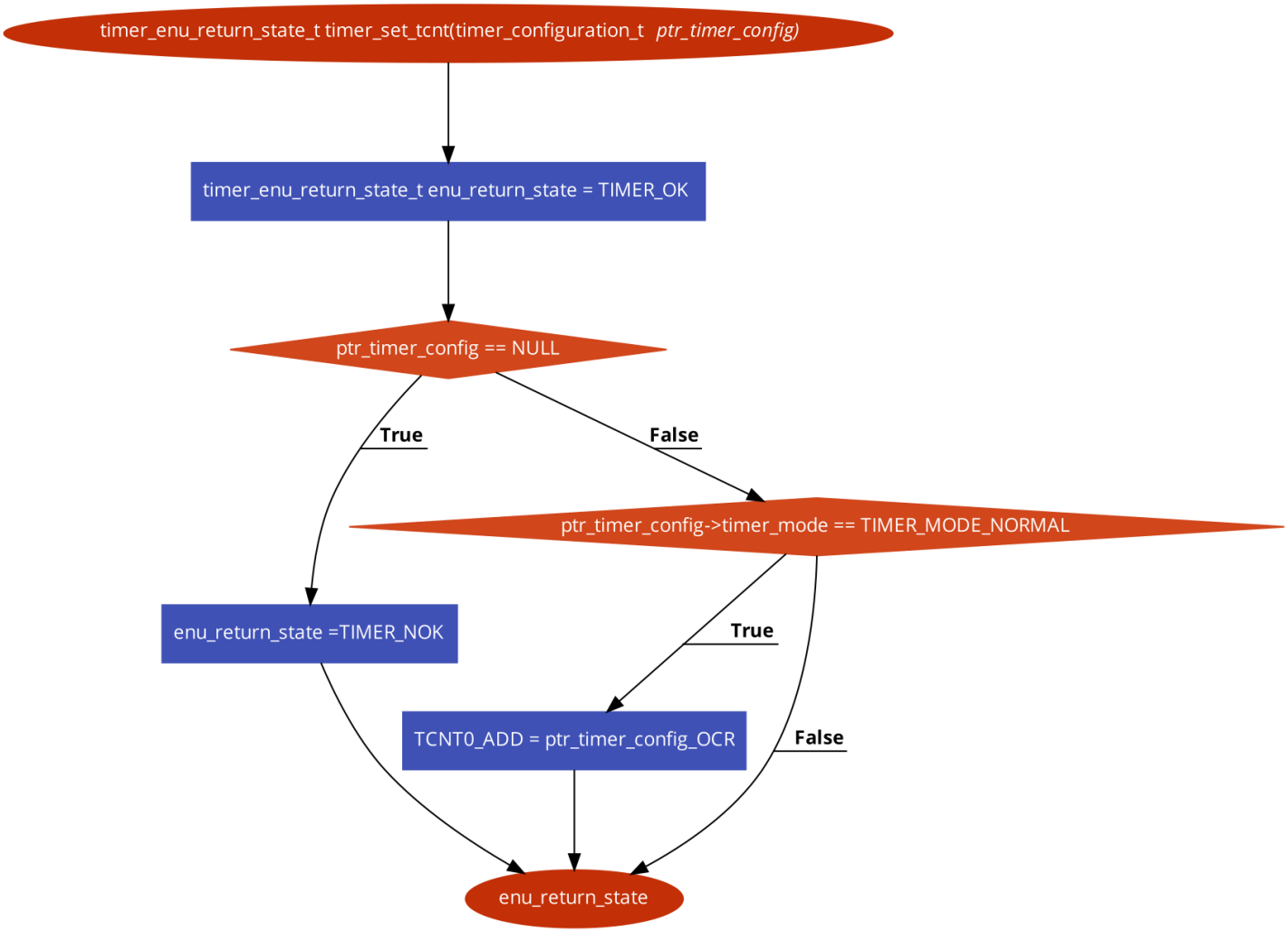

Figure 12 timer_x_set_tcnt()

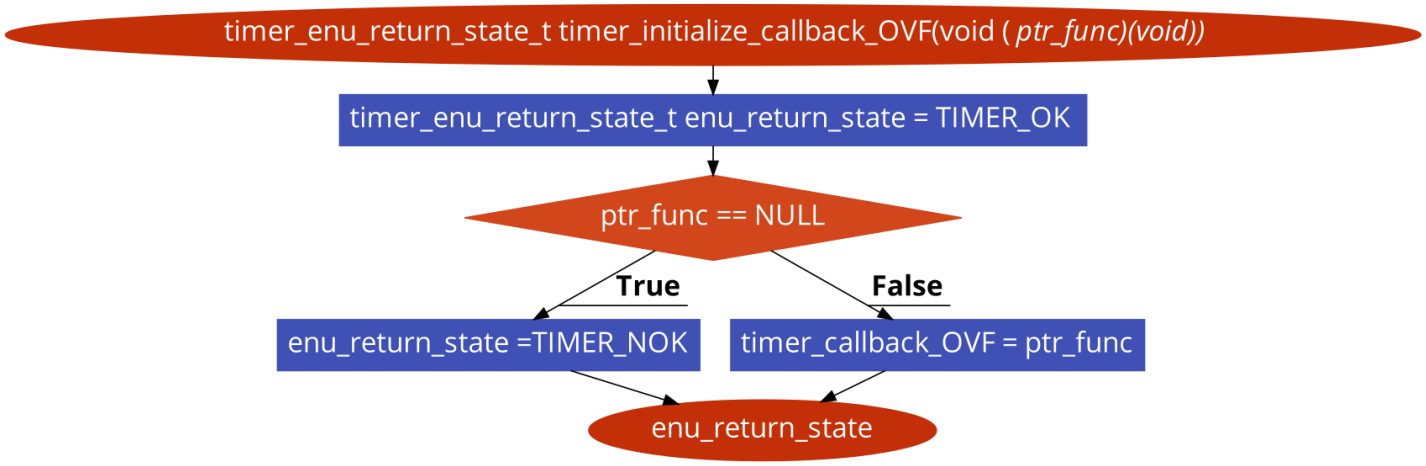

Figure 13 timer_x_initialize_callback_OVF()

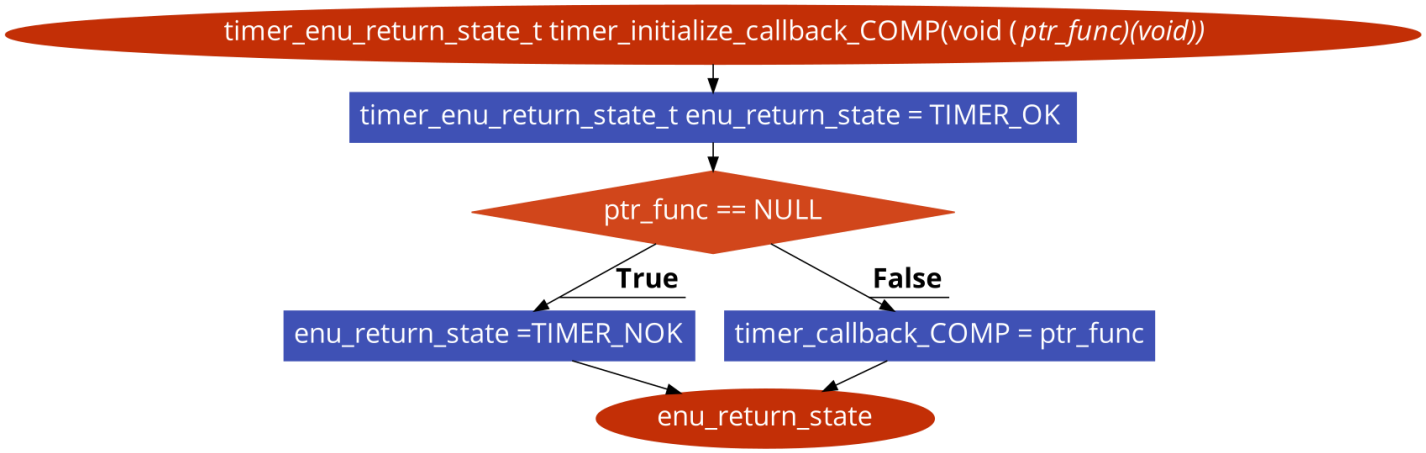

Figure 14 timer_x_initialize_callback_COMP()

HAL

ULTRASONIC API:

Flowcharts:

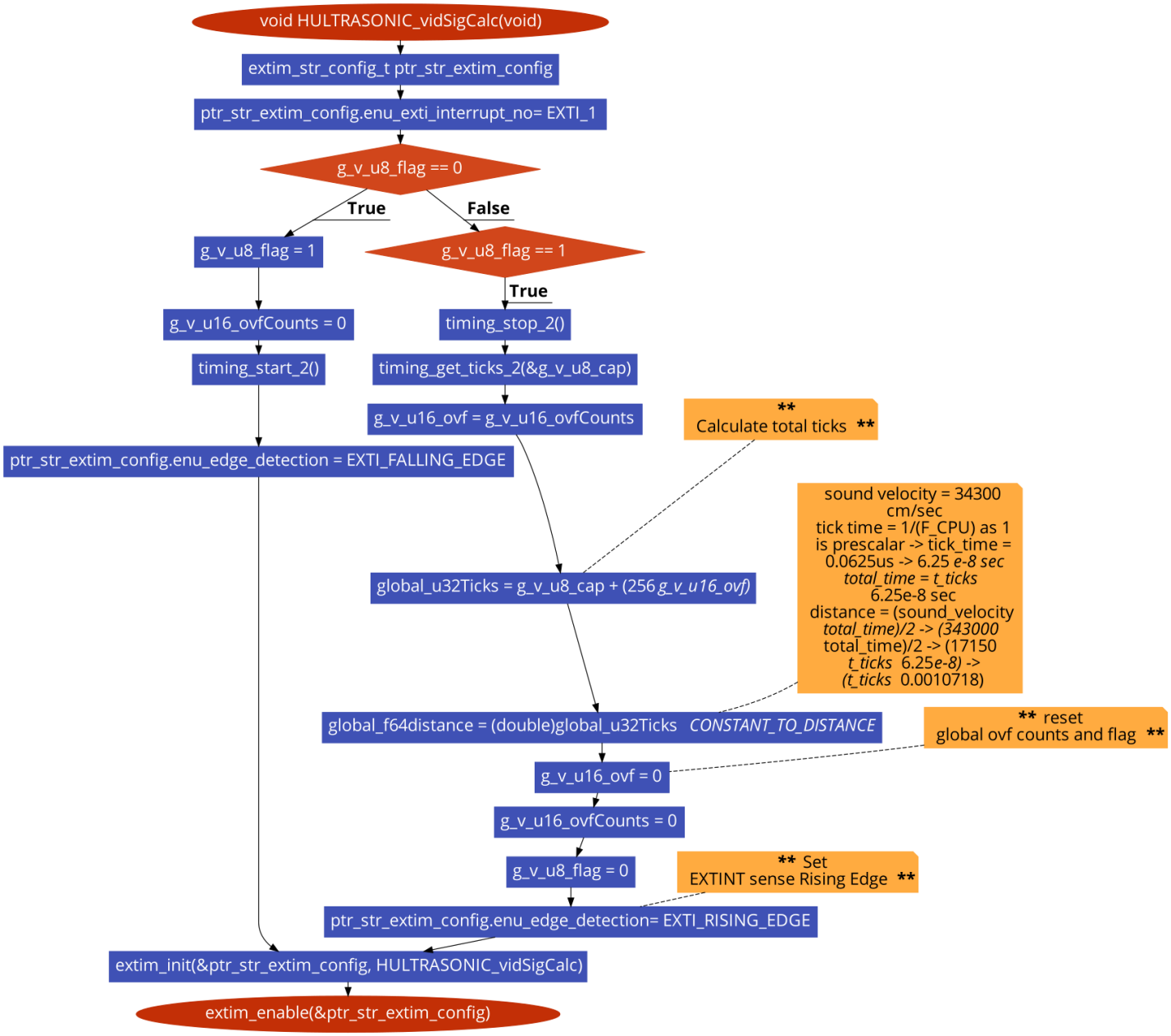

Figure 15 HULTRASONIC_vidSigCalc()

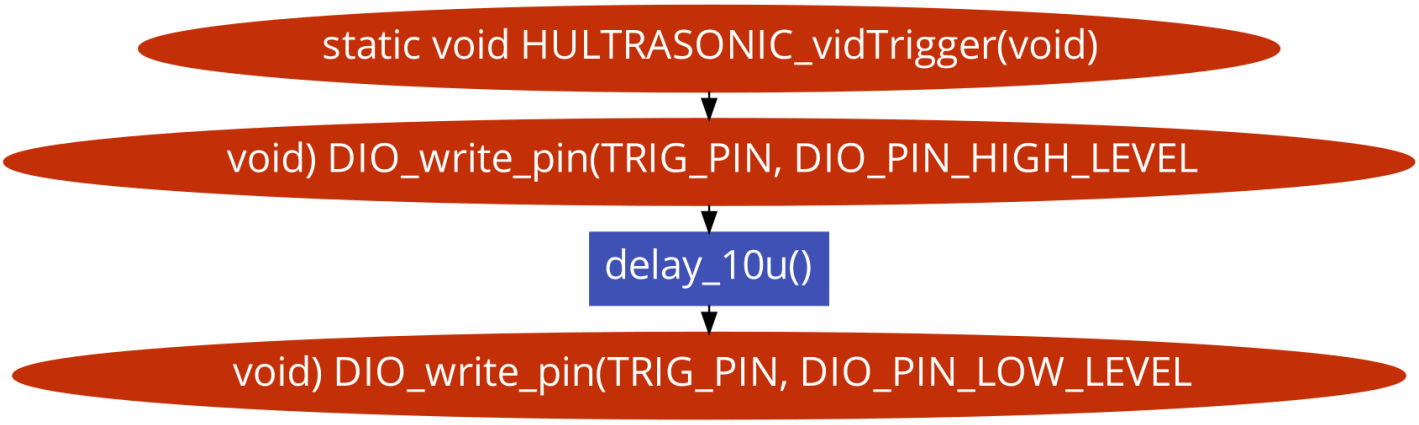

Figure 16 HULTRASONIC_vidTrigger()

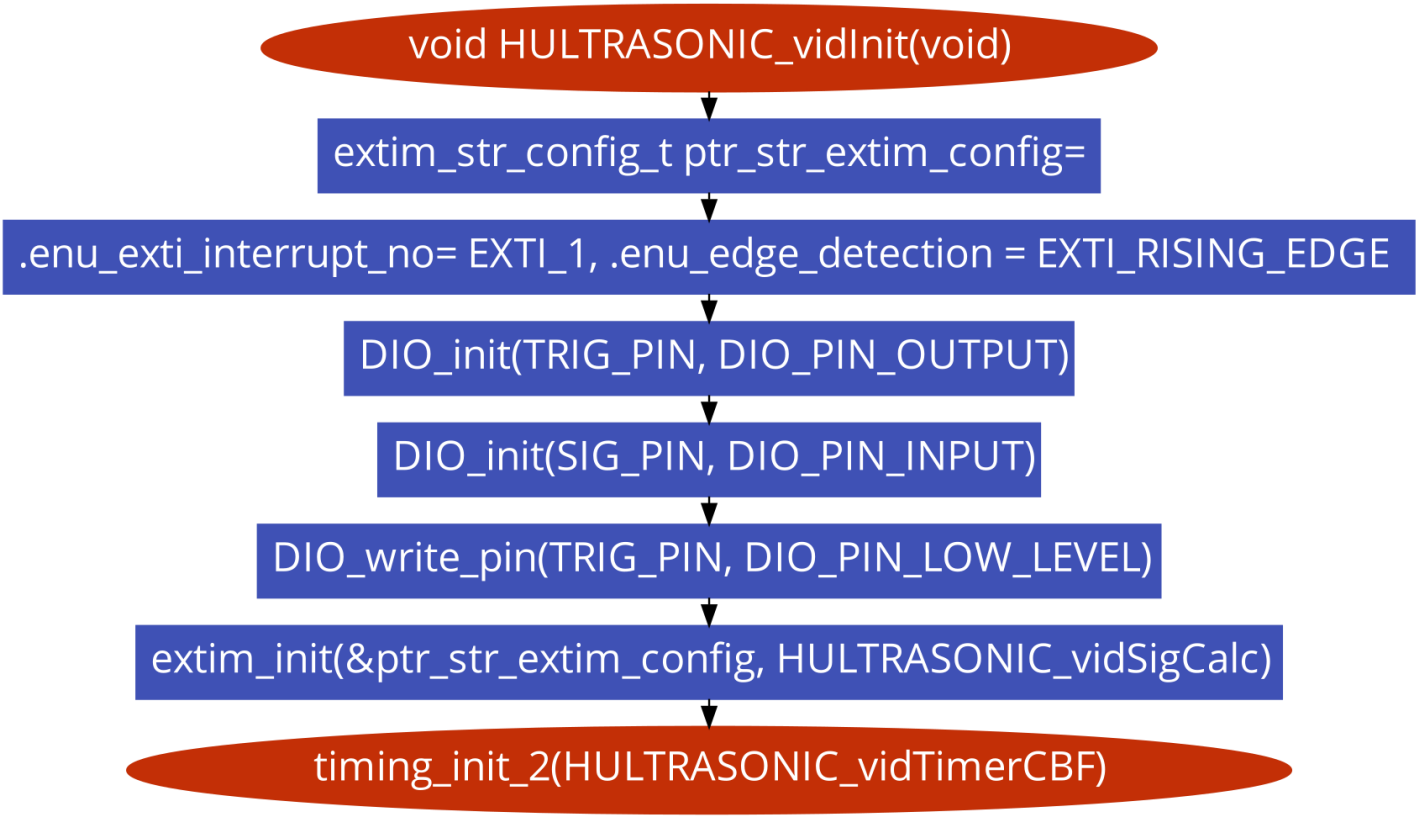

Figure 17 HULTRASONIC_vidInit()

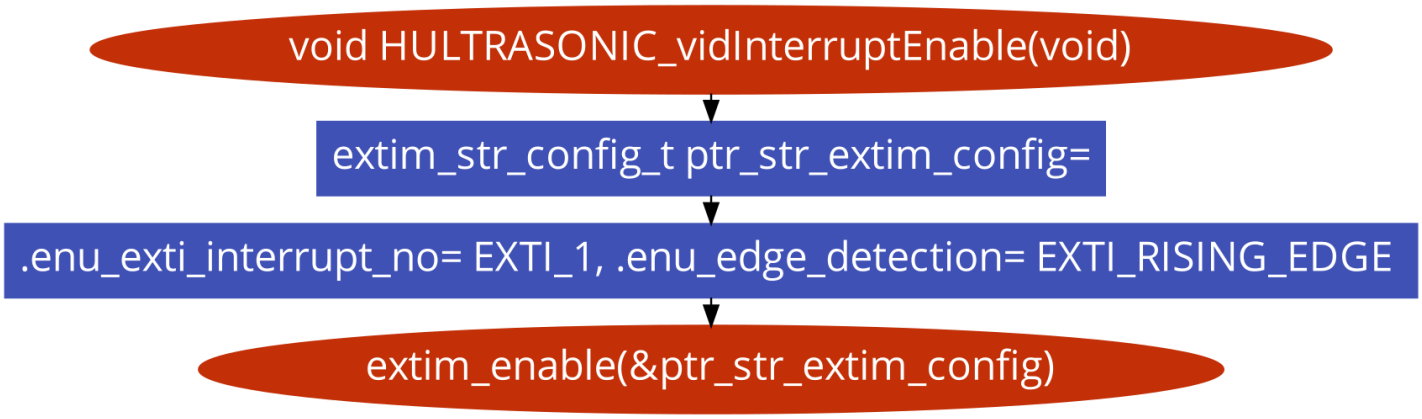

Figure 18 HULTRASONIC_vidInterruptEnable()

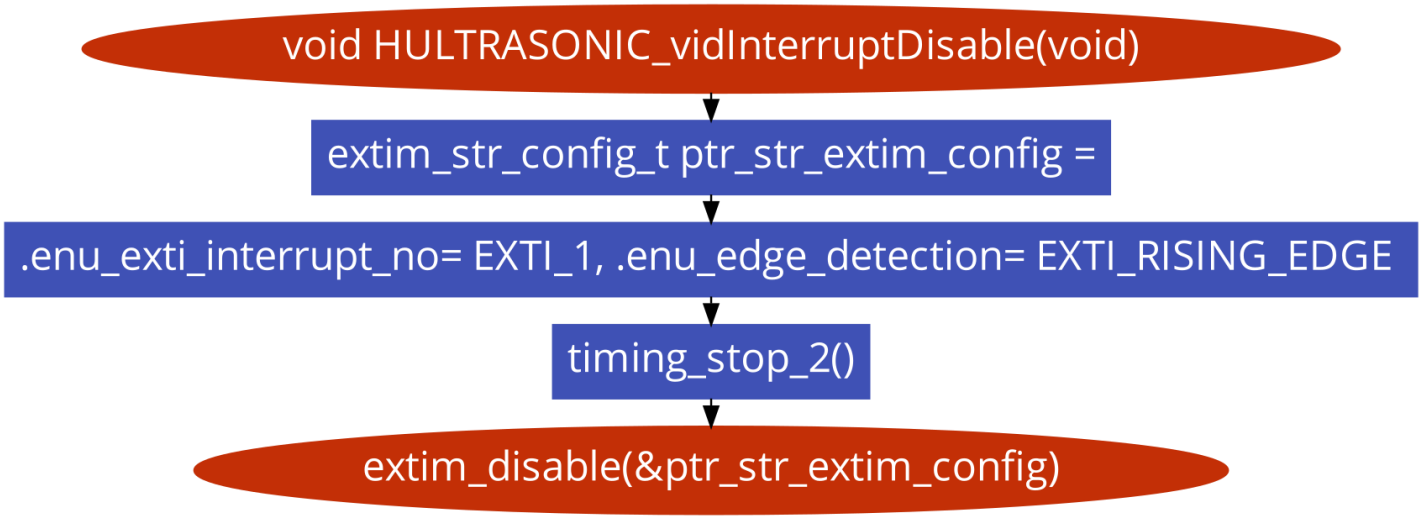

Figure 20 HULTRASONIC_vidInterruptDisable()

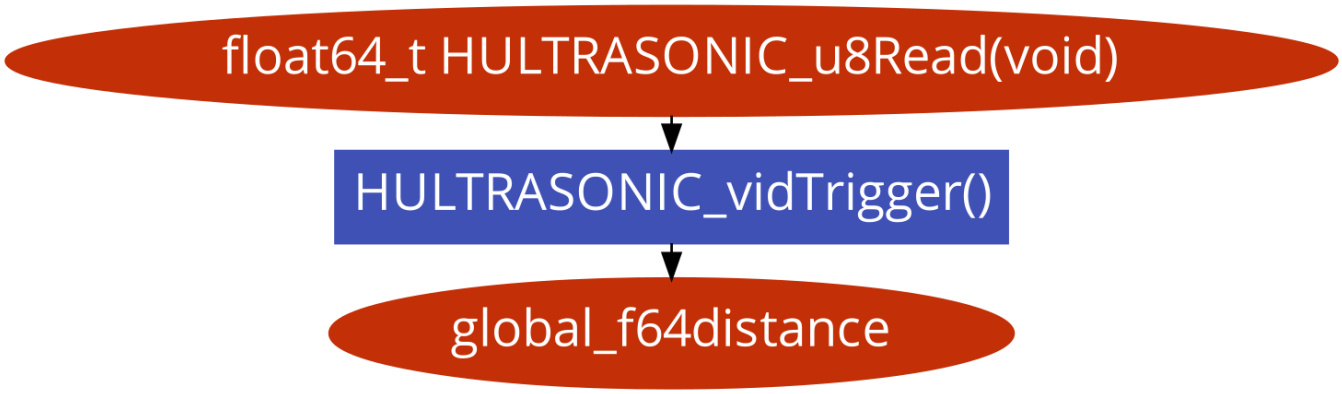

Figure 19 HULTRASONIC_u8Read()

LCD API:

Flowcharts:

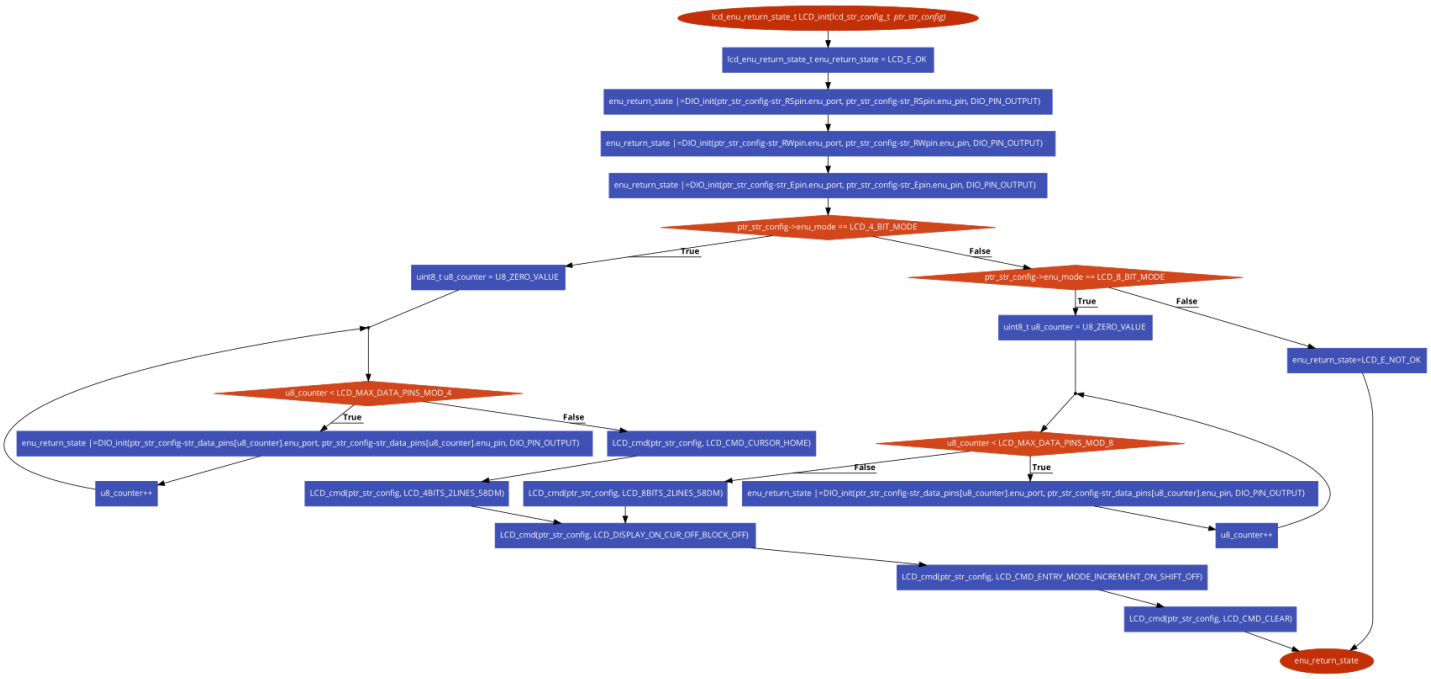

Figure 22 LCD_init ()

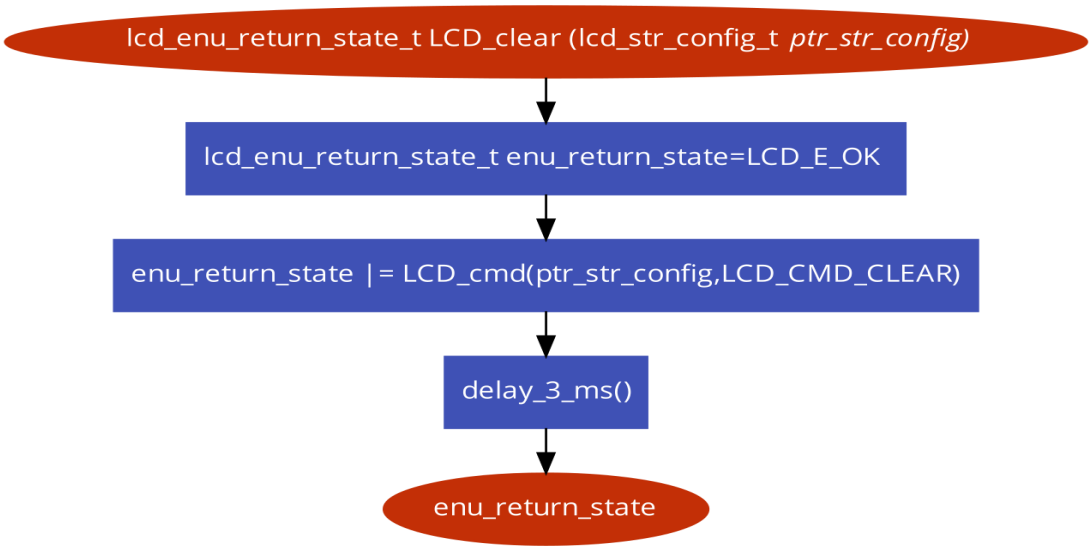

Figure 21 LCD_clear()

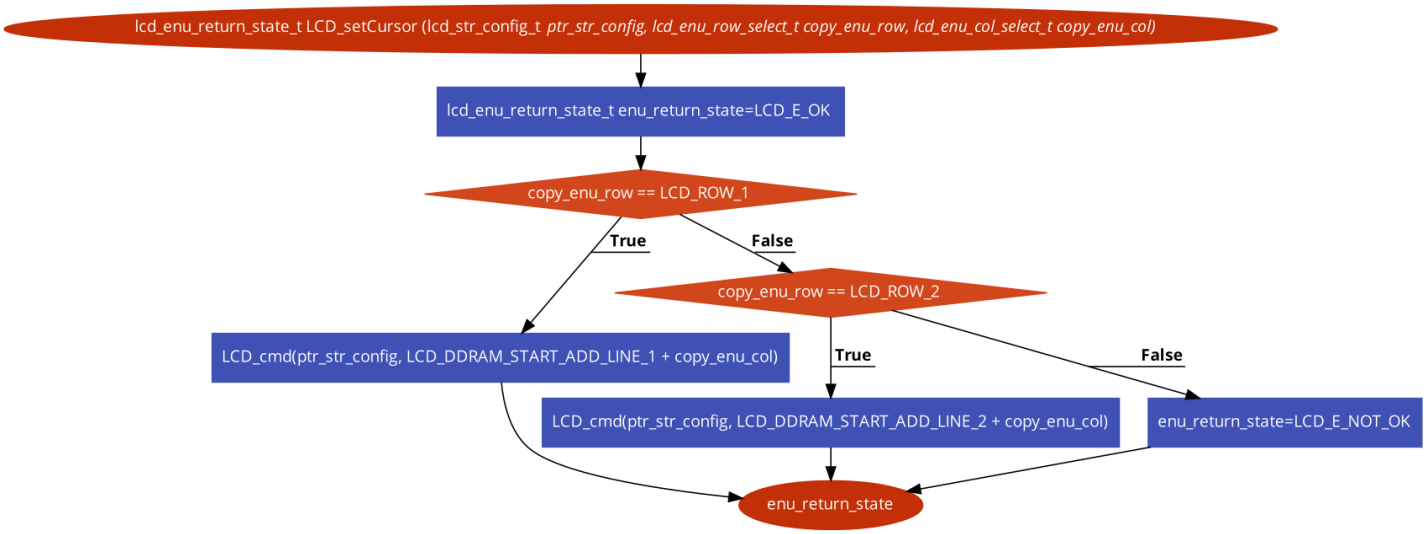

Figure 24 LCD_setCursor()

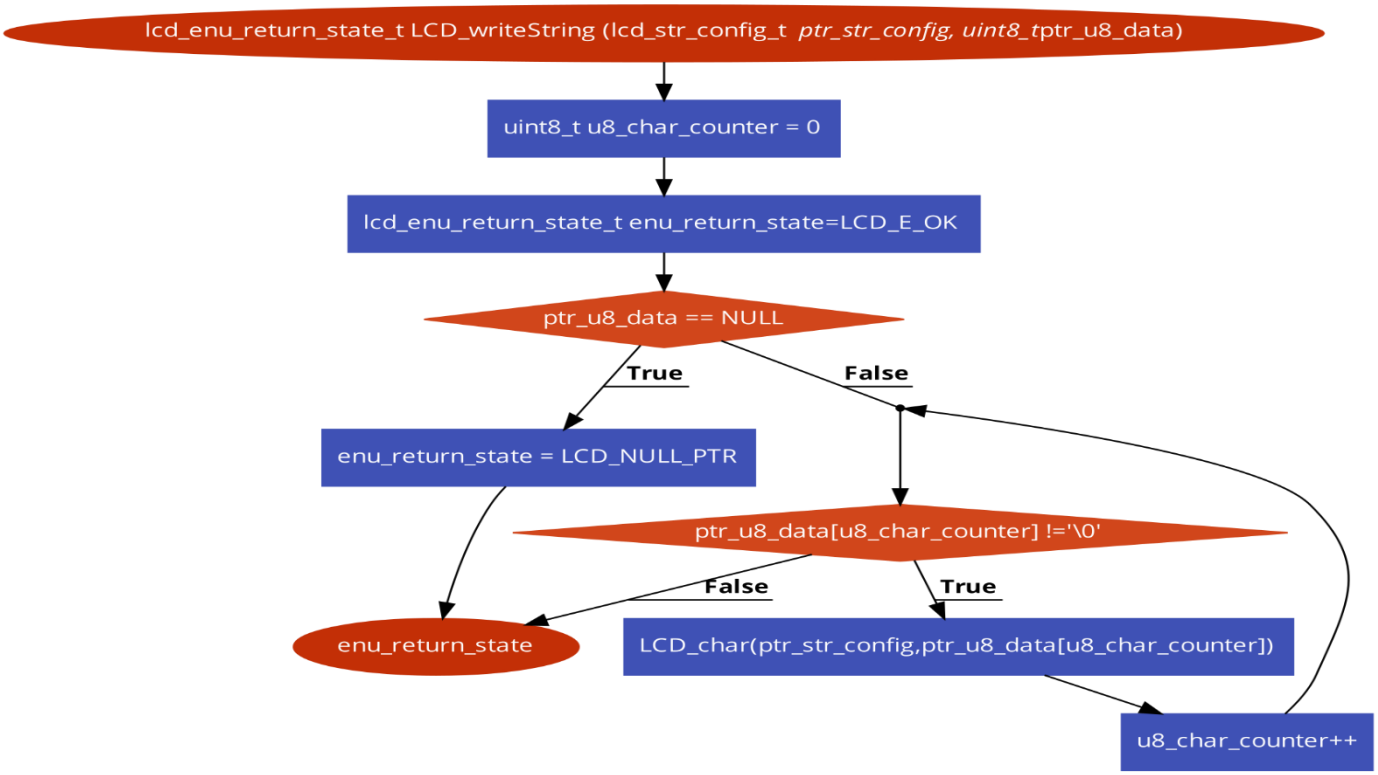

Figure 23 LCD_writeString()

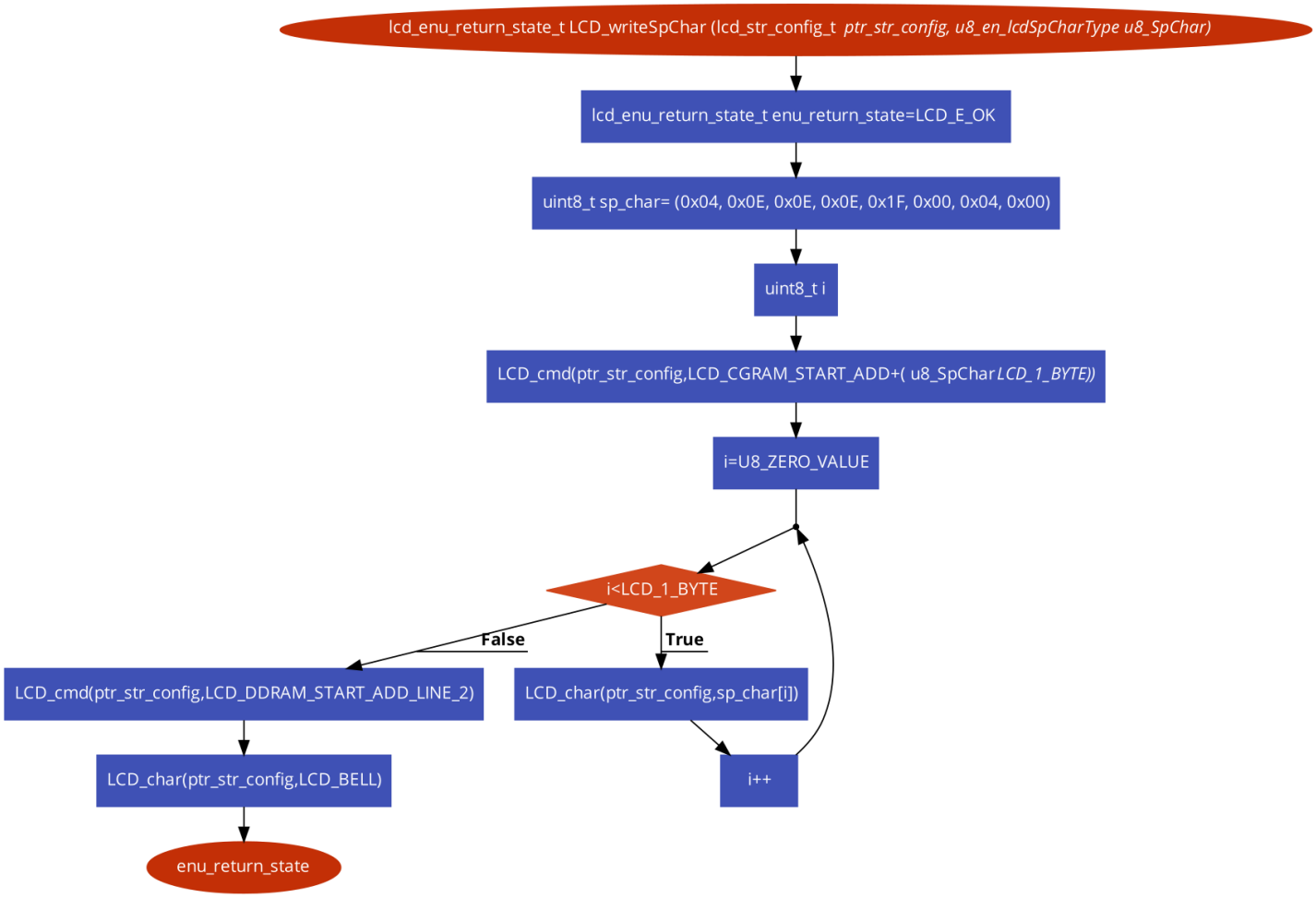

Figure 25 LCD_writeSpChar()

Motor API:

Flowcharts:

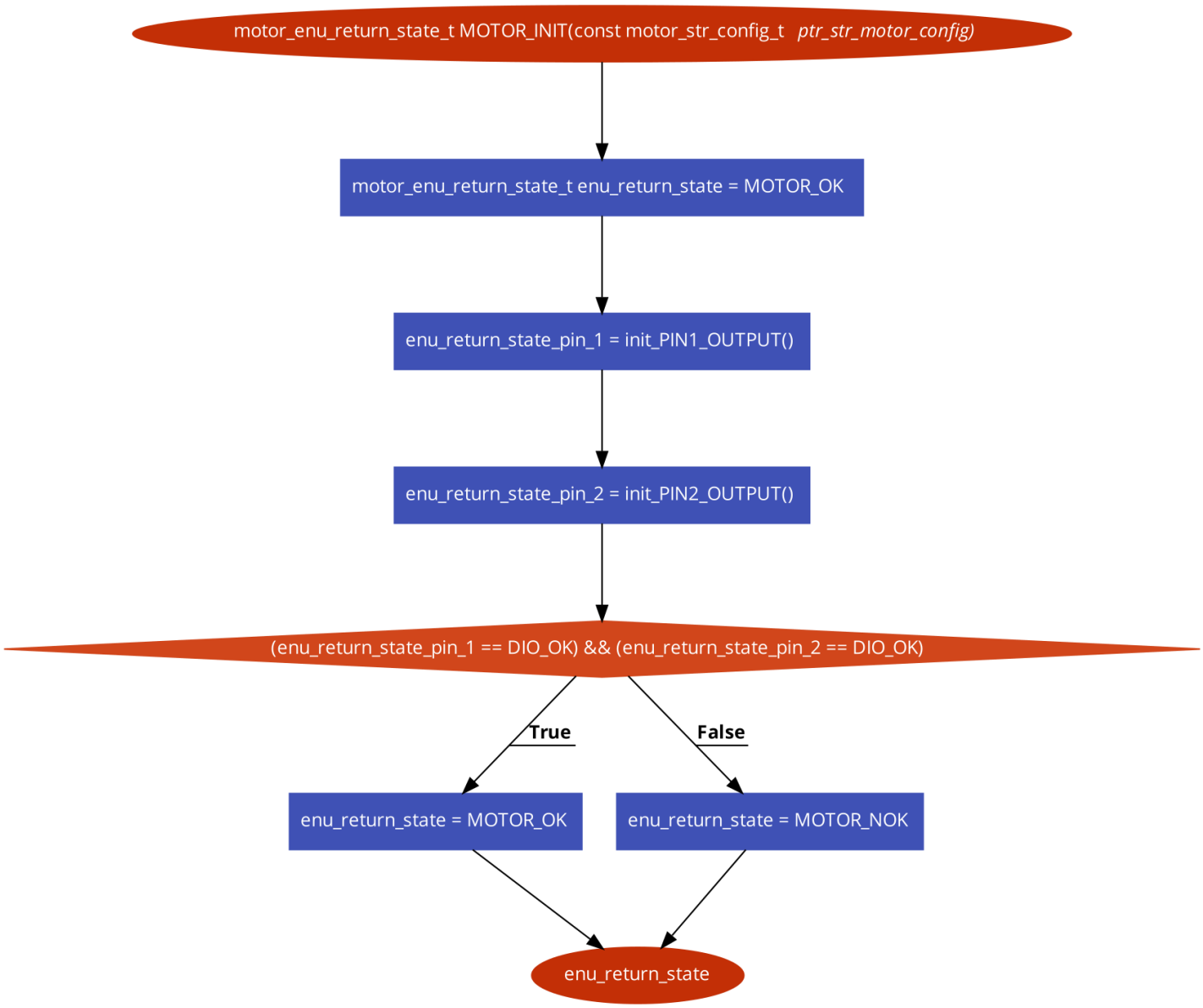

Figure 26 MOTOR_INIT ()

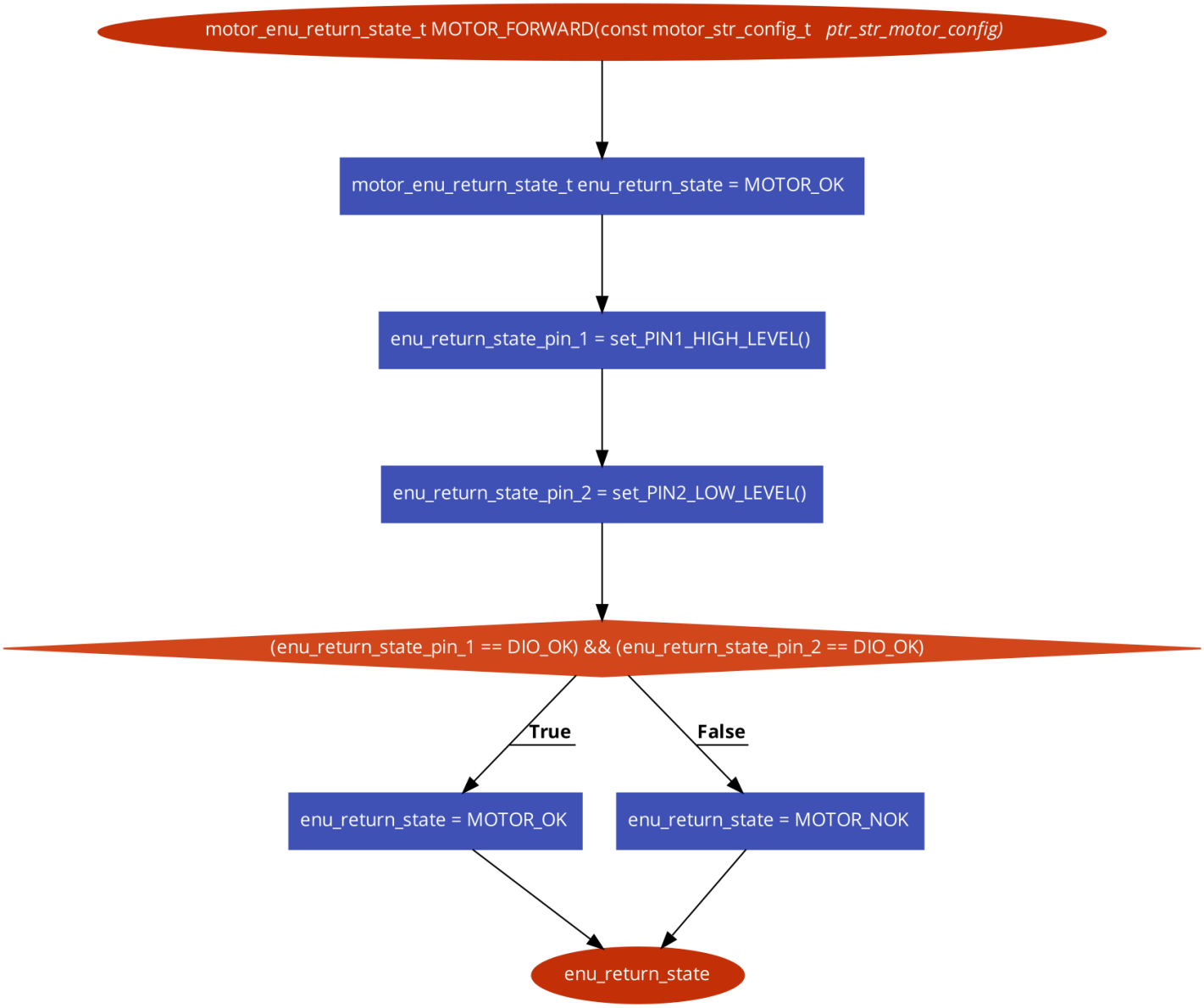

Figure 27 MOTOR_FORWARD()

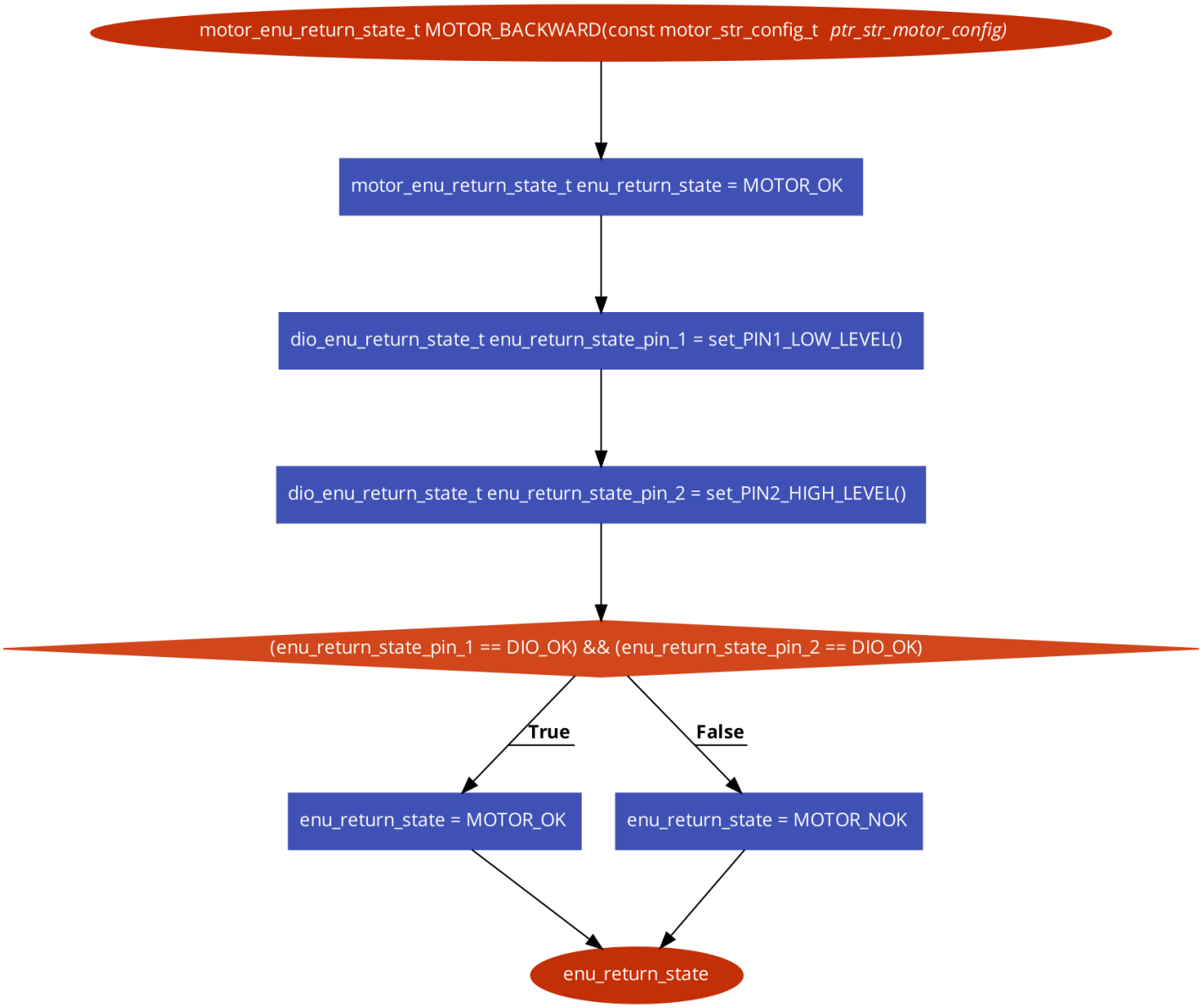

Figure 28 MOTOR_BACKWARD()

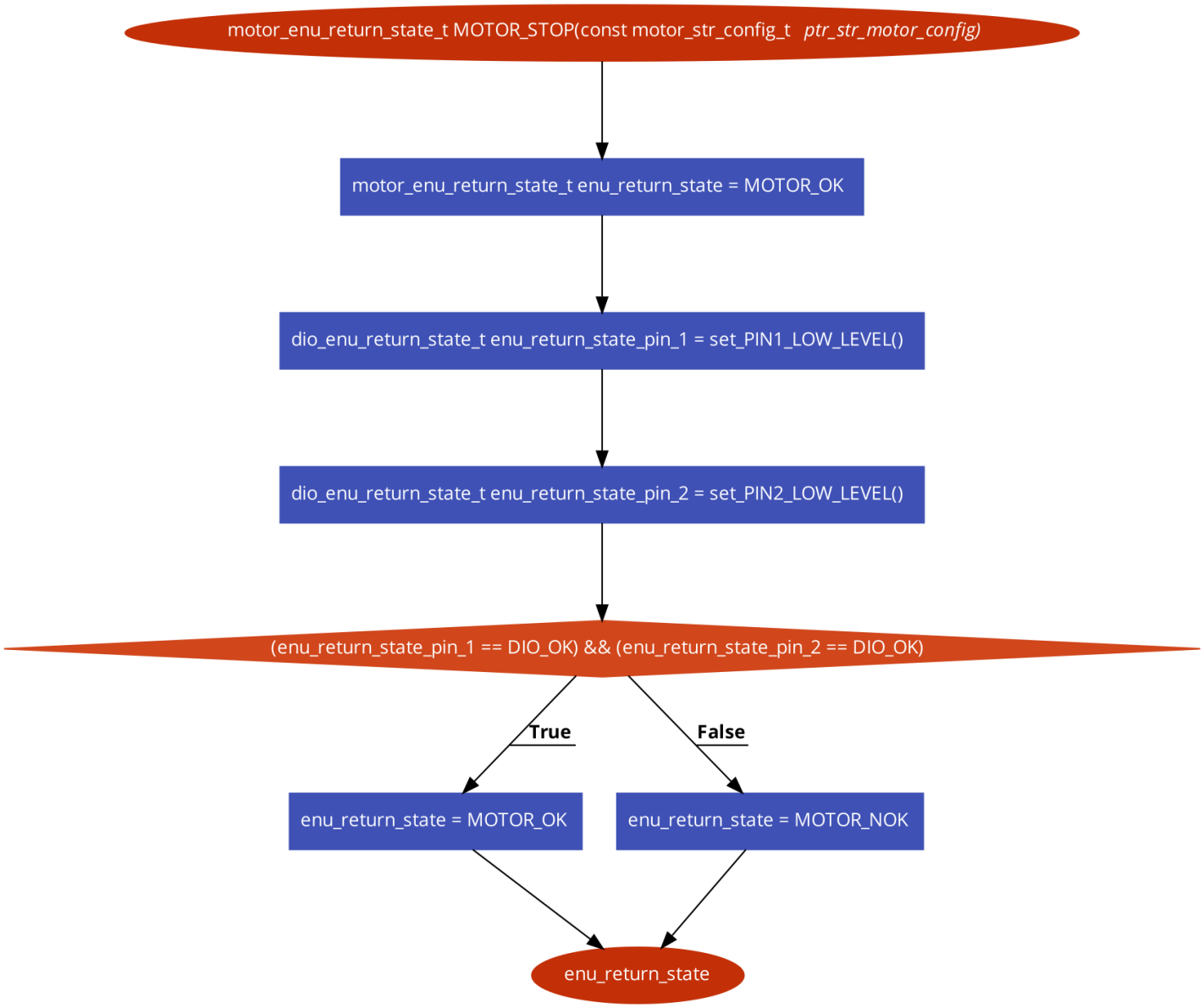

Figure 29 MOTOR_STOP()

Car Control API :

Flowcharts:

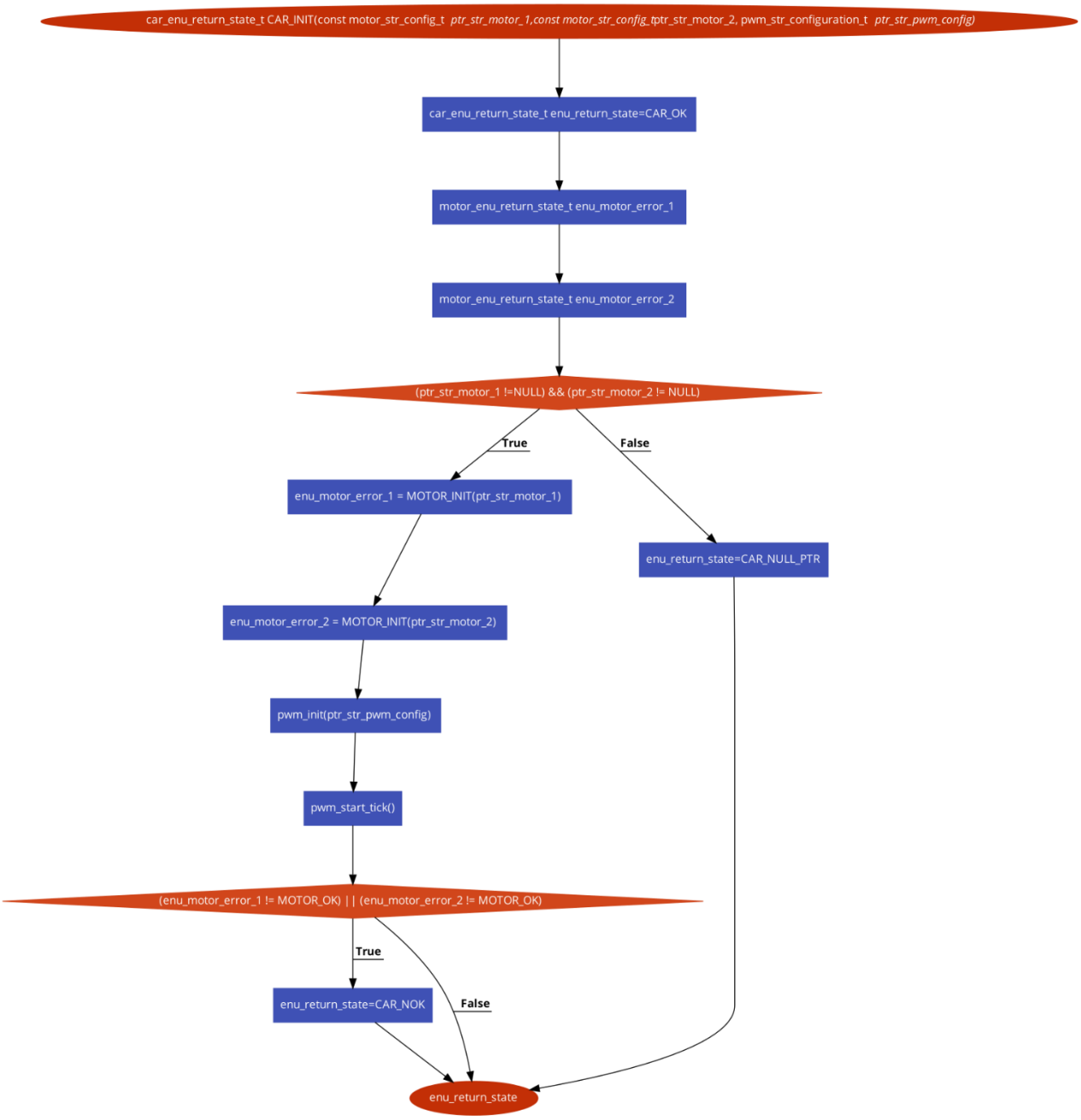

Figure 30 CAR_INIT()

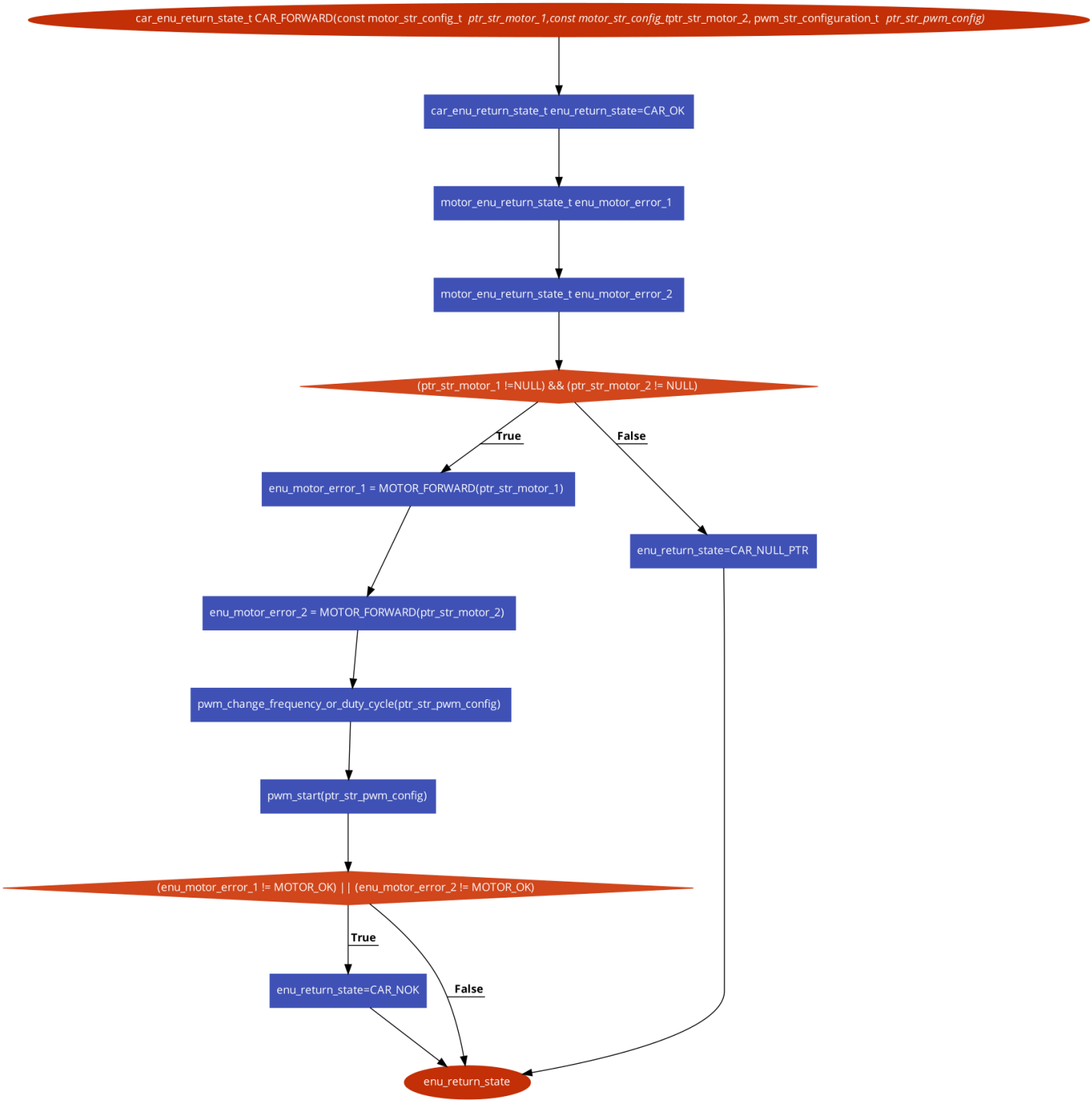

Figure 31 CAR_FORWARD()

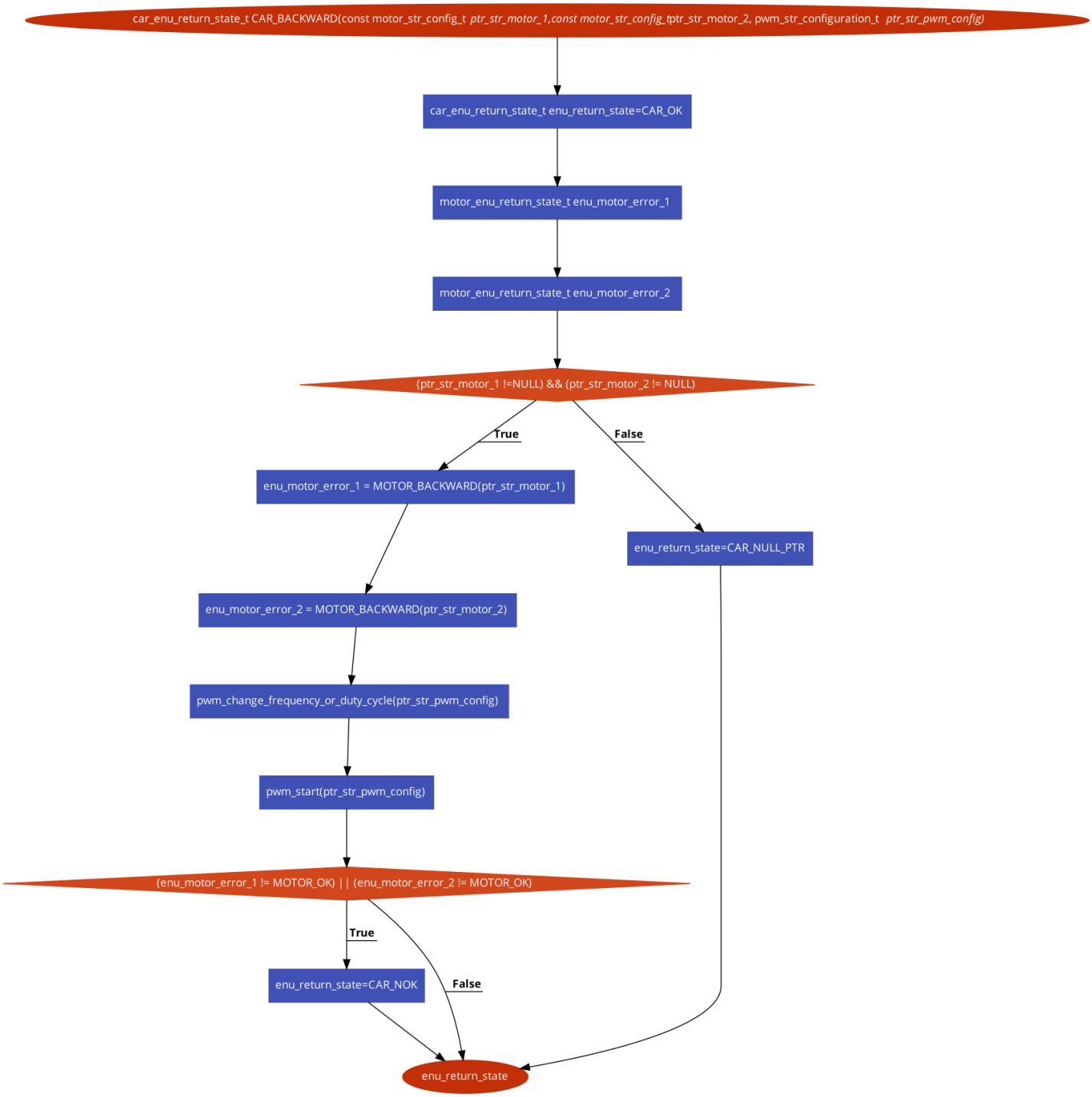

Figure 32 CAR_BACKWARD()

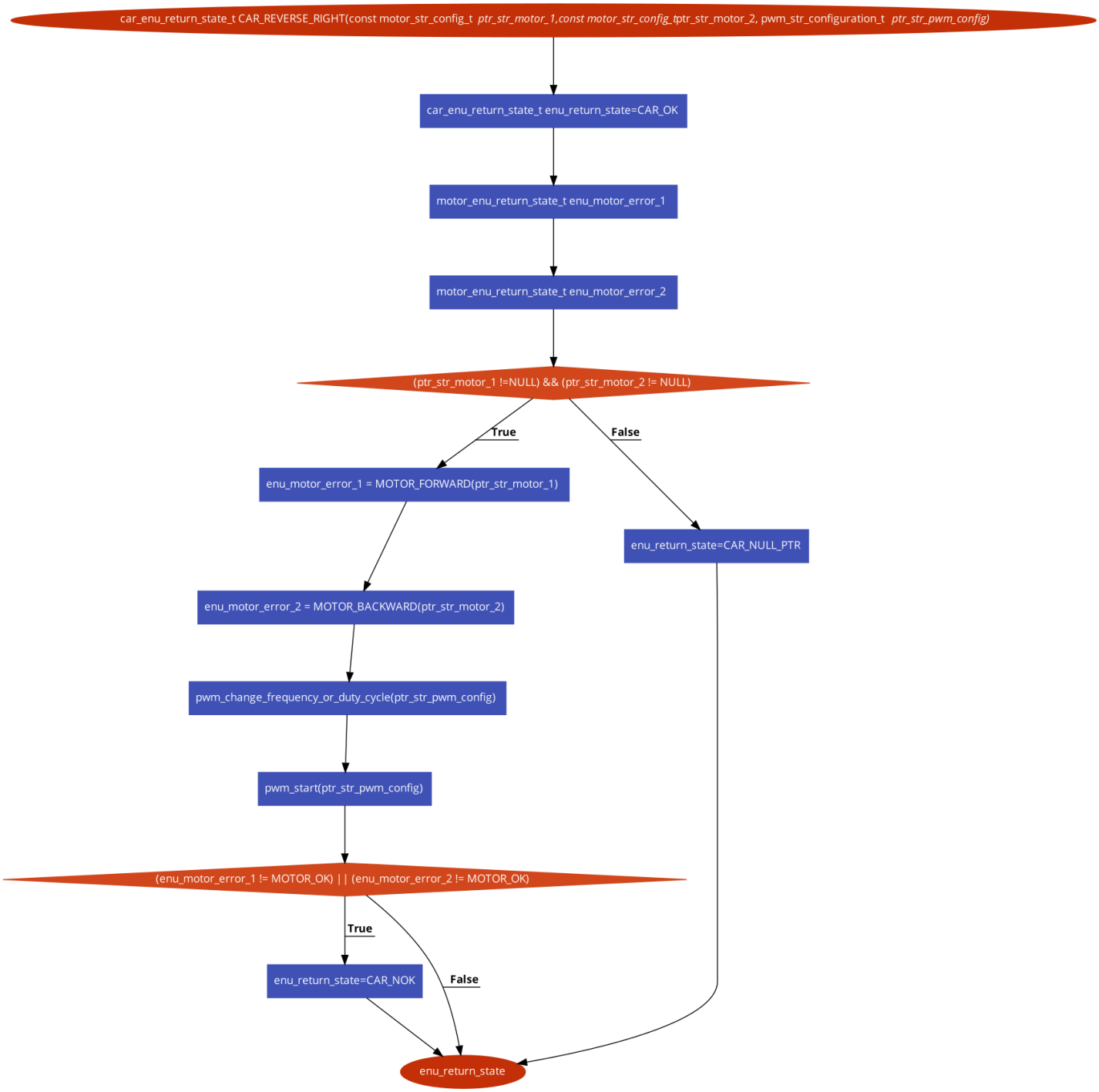

Figure 33 CAR_REVERSE_RIGHT()

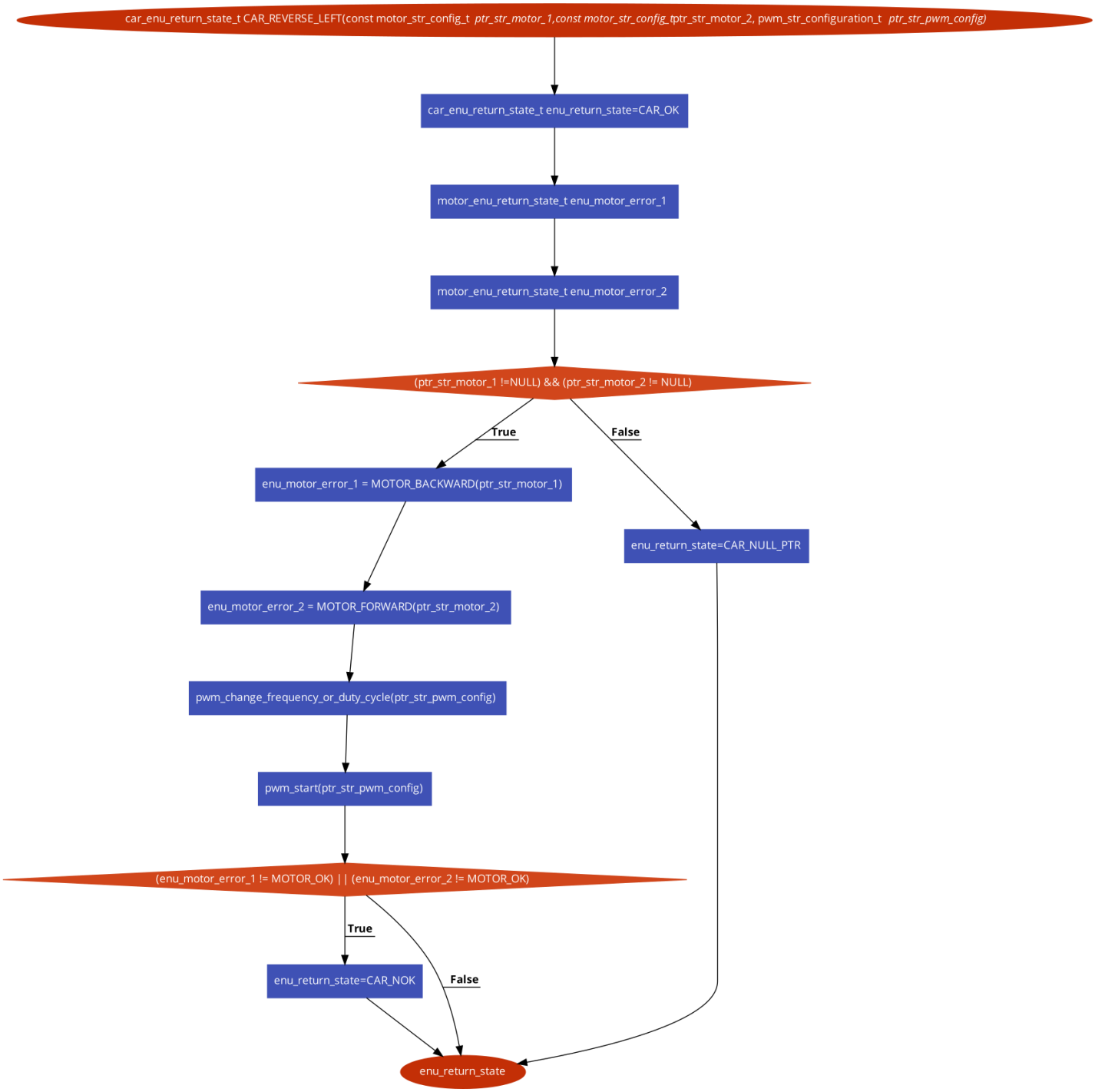

Figure 34 CAR_REVERSE_LEFT()

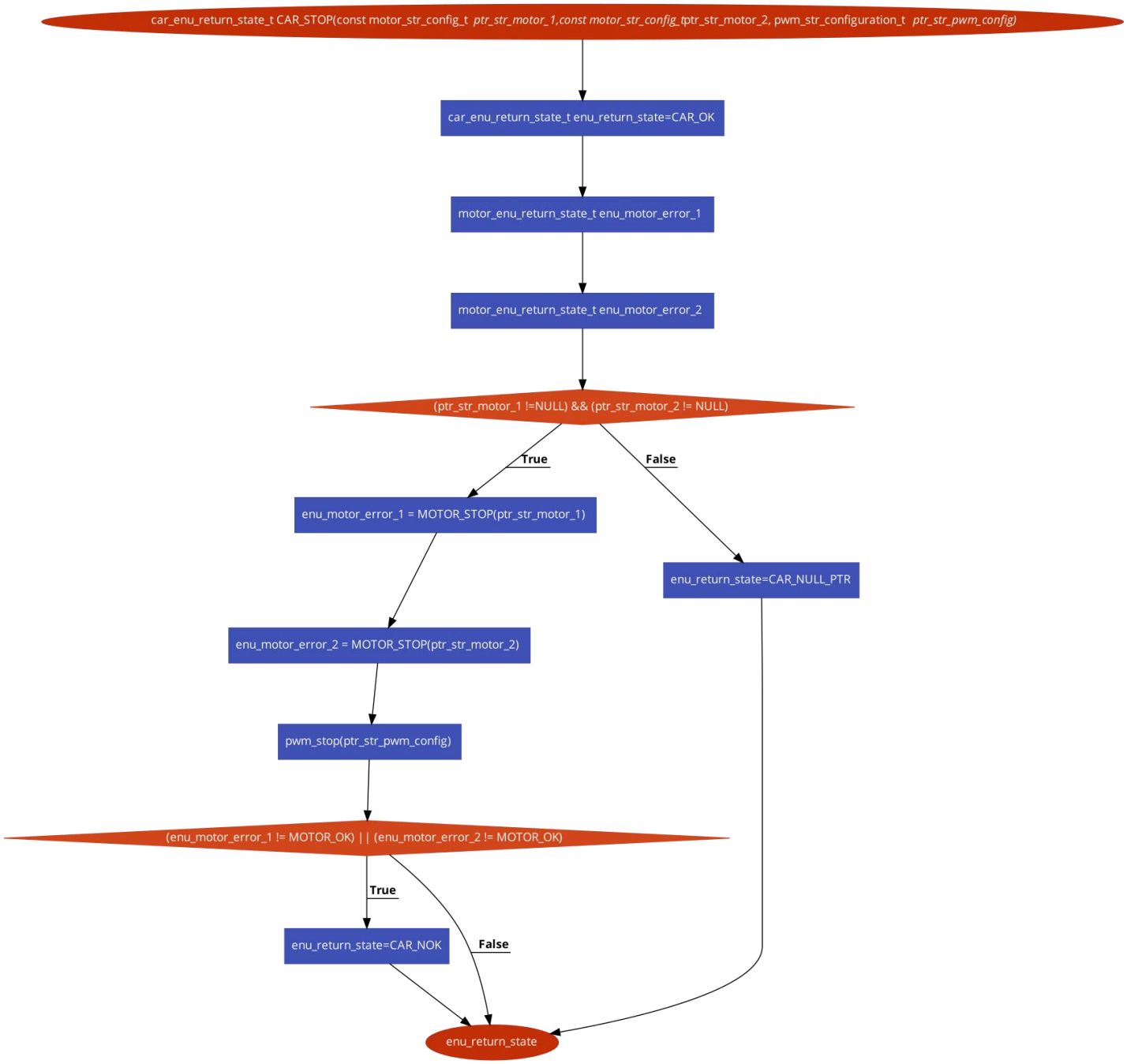

Figure 35 CAR_STOP

TIMING API :

Flowcharts:

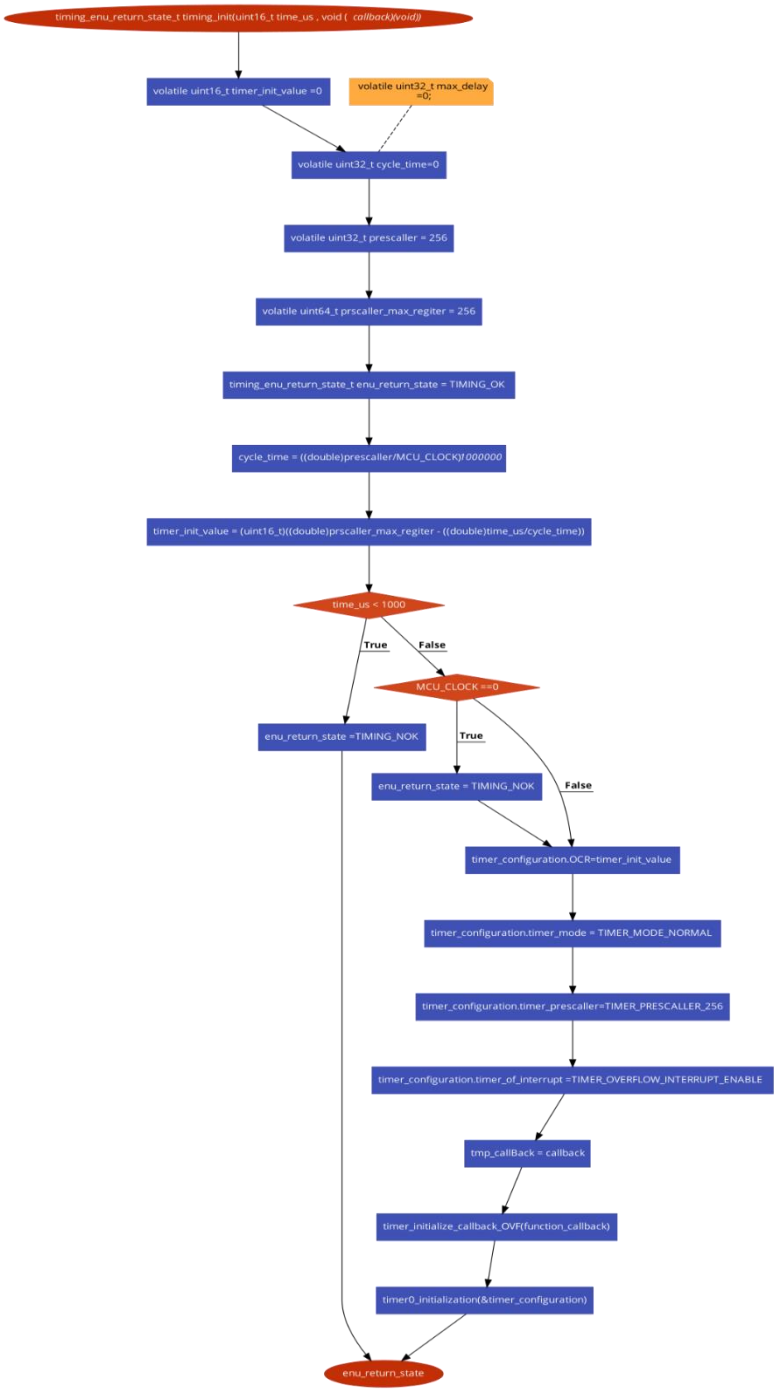

Figure 36 `timing_init ()`

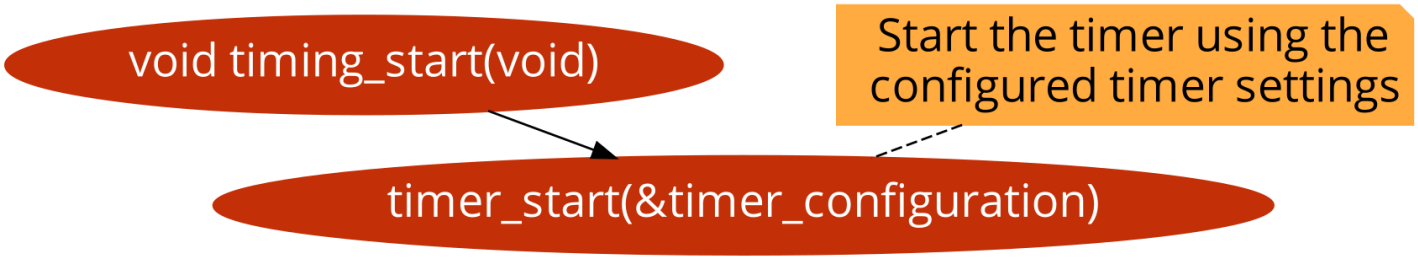

Figure 37 `timing_star()`

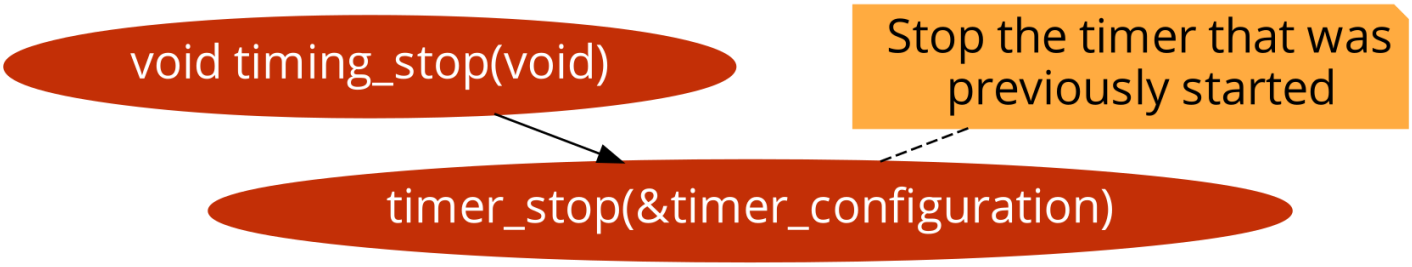

Figure 38 timing_stop

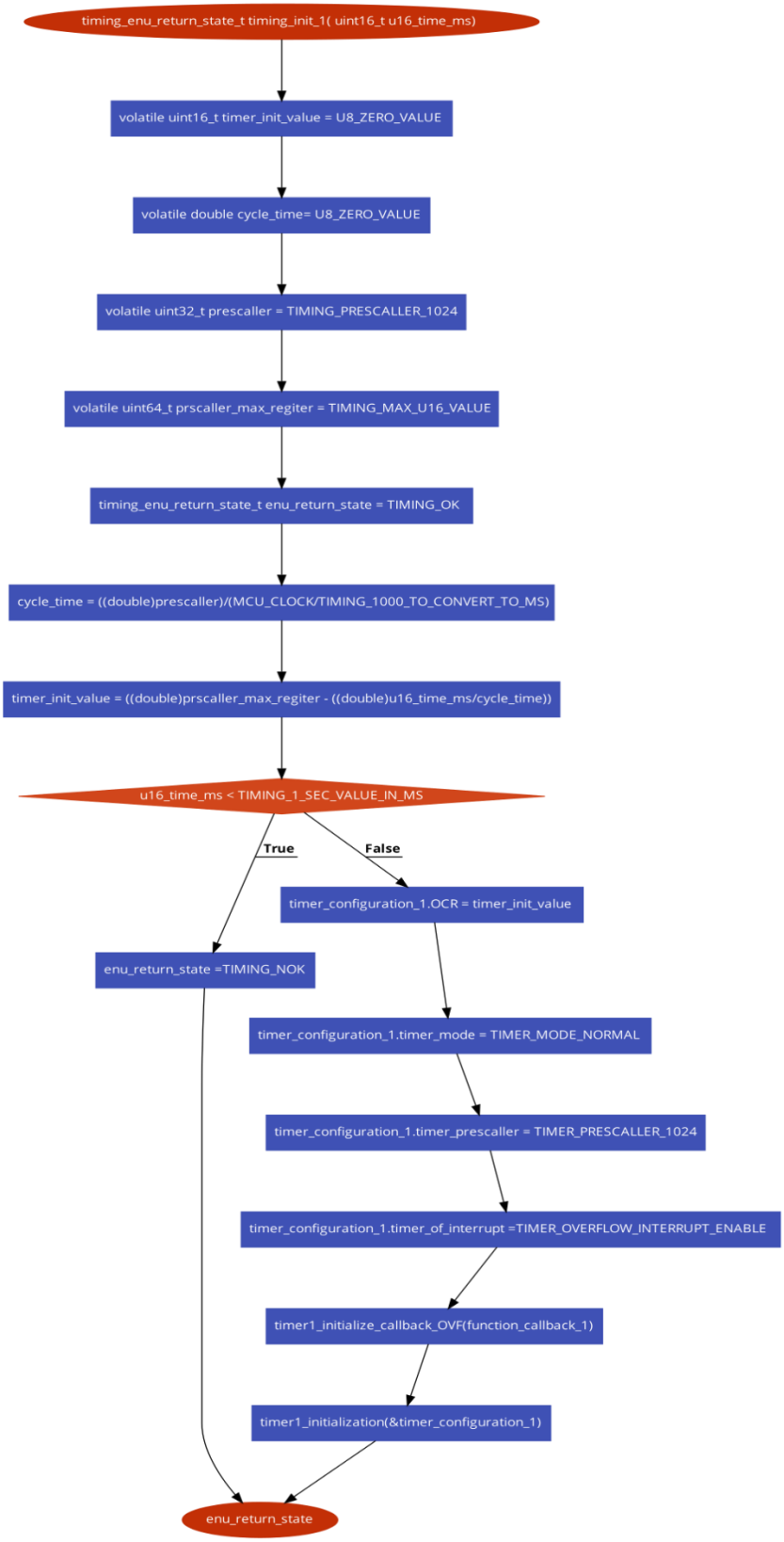

Figure 39 `timing_init_1()`

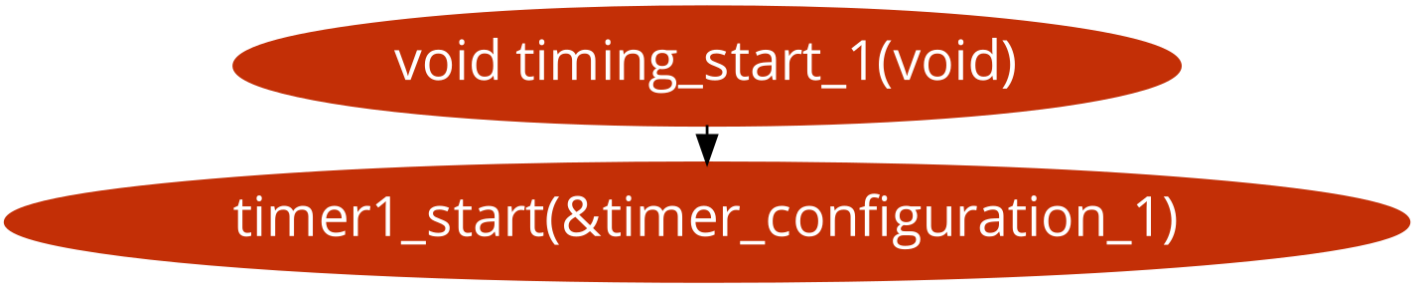

Figure 40 timing_start_1()

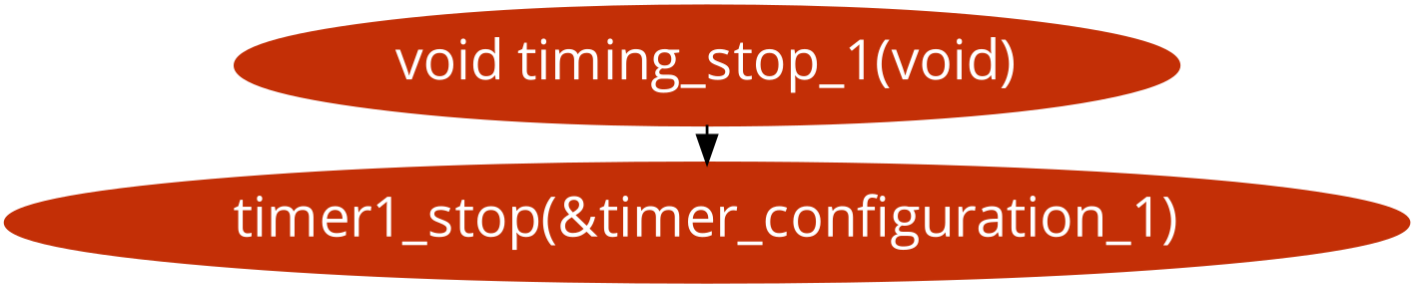

Figure 42 timing_stop_1()

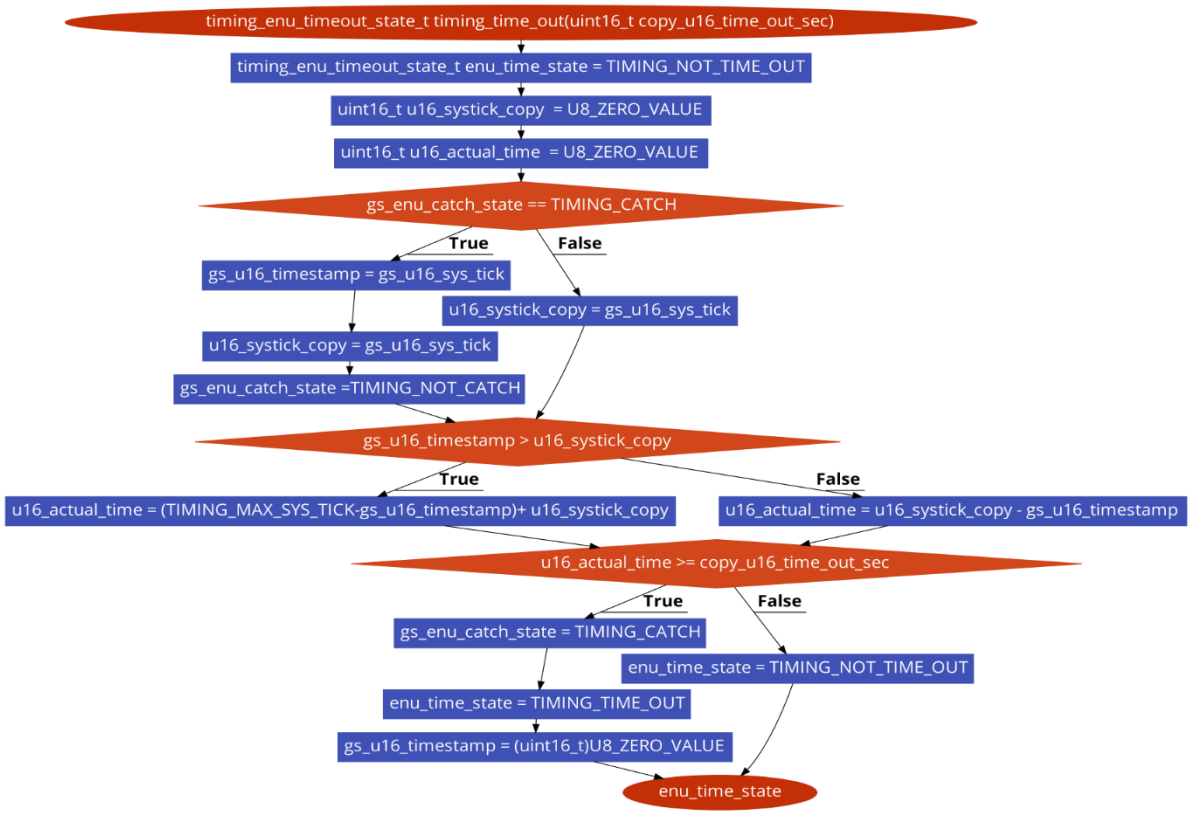

Figure 41 timing_time_out()

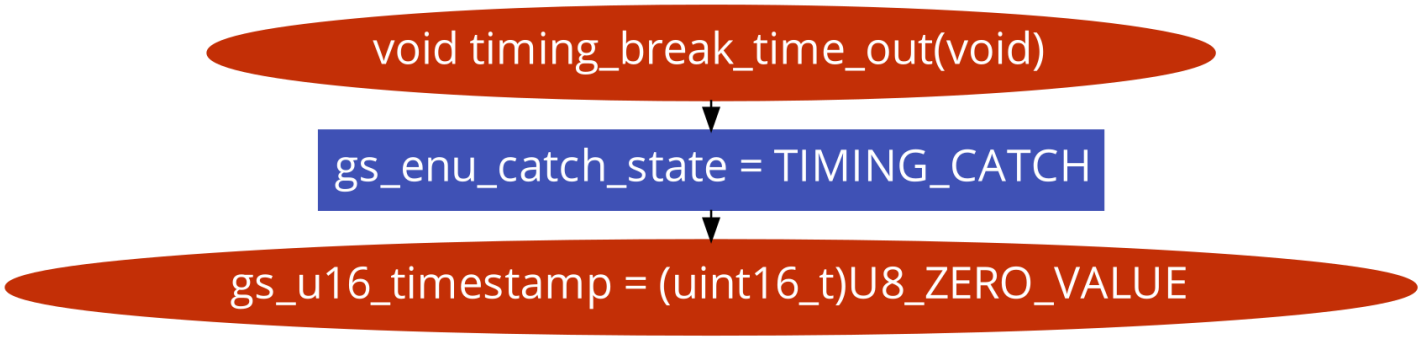

Figure 43 timing_break_time_out()

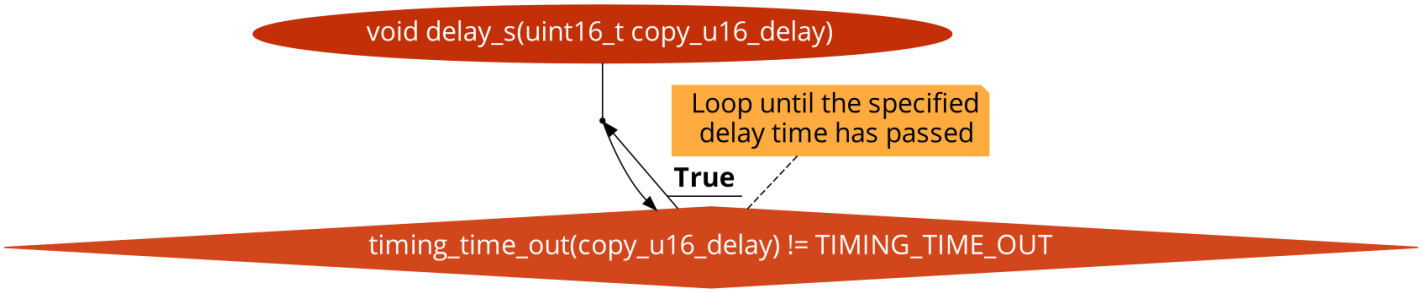

Figure 44 delay_s()

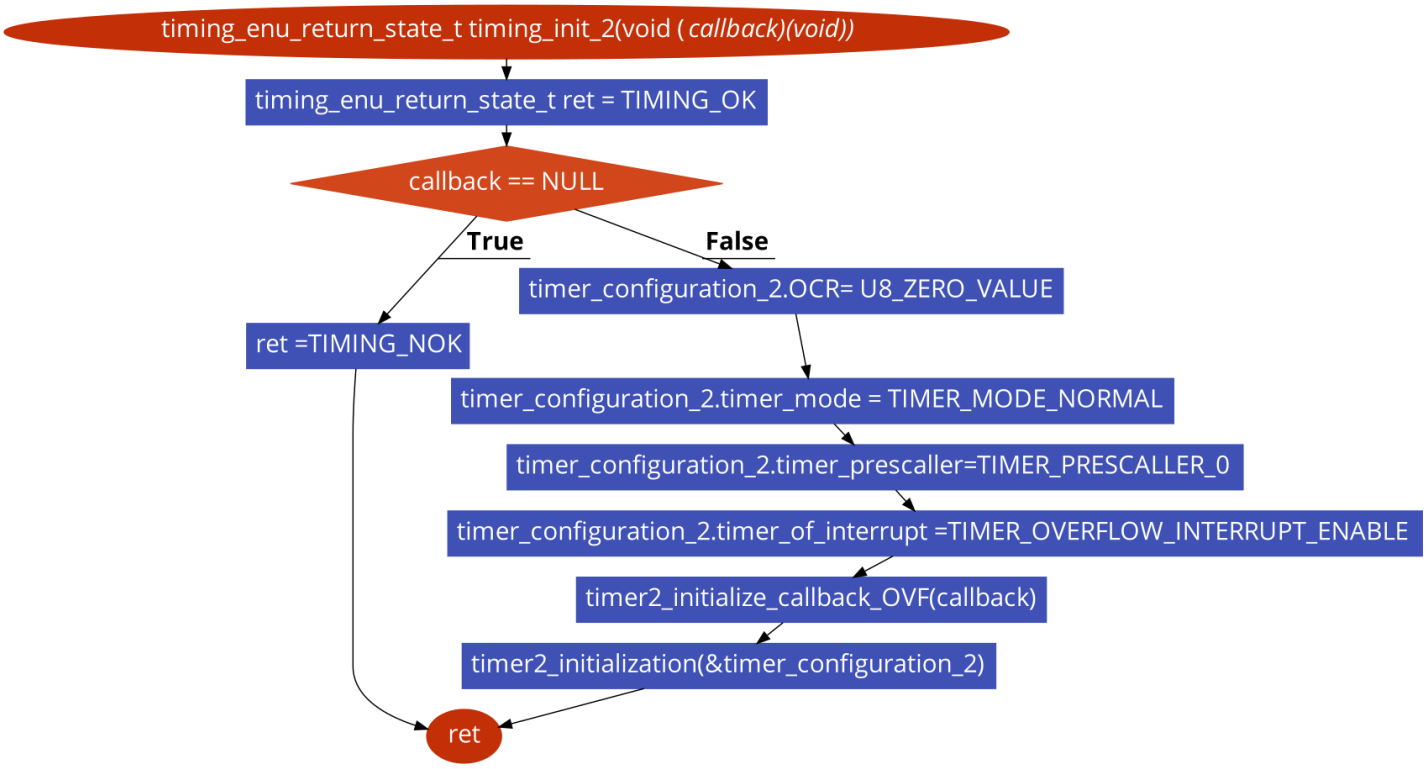

Figure 45 timing_init_2()

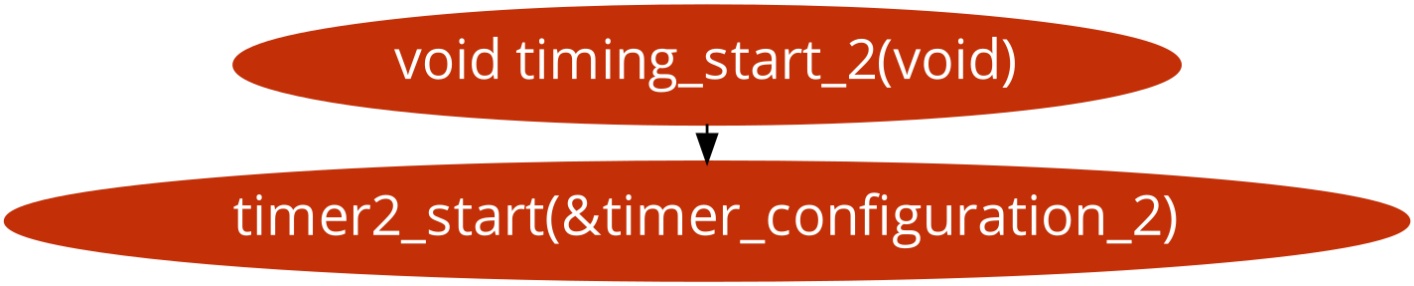

Figure 46 timing_start_2()

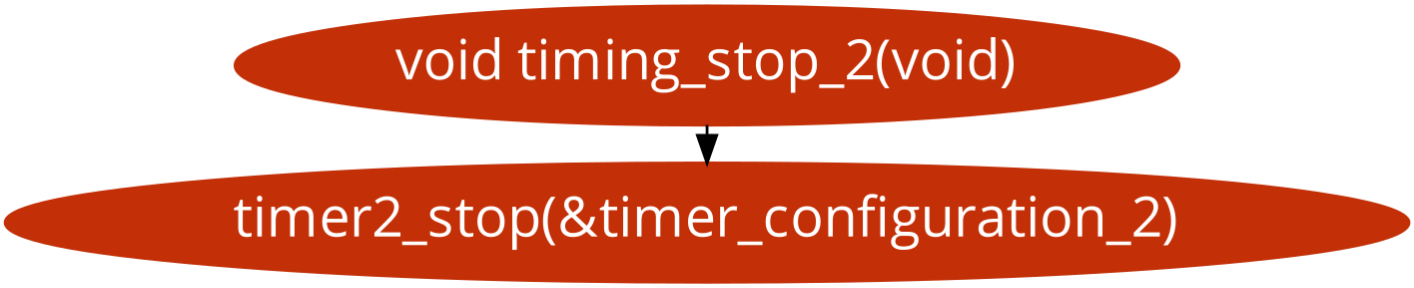

Figure 47 timing_stop_2()

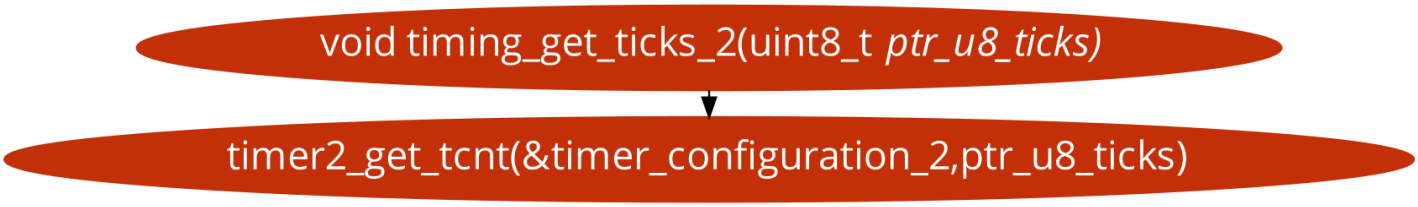

Figure 48 timing_get_ticks_2()

Interrupt Manager API :

Flowcharts:

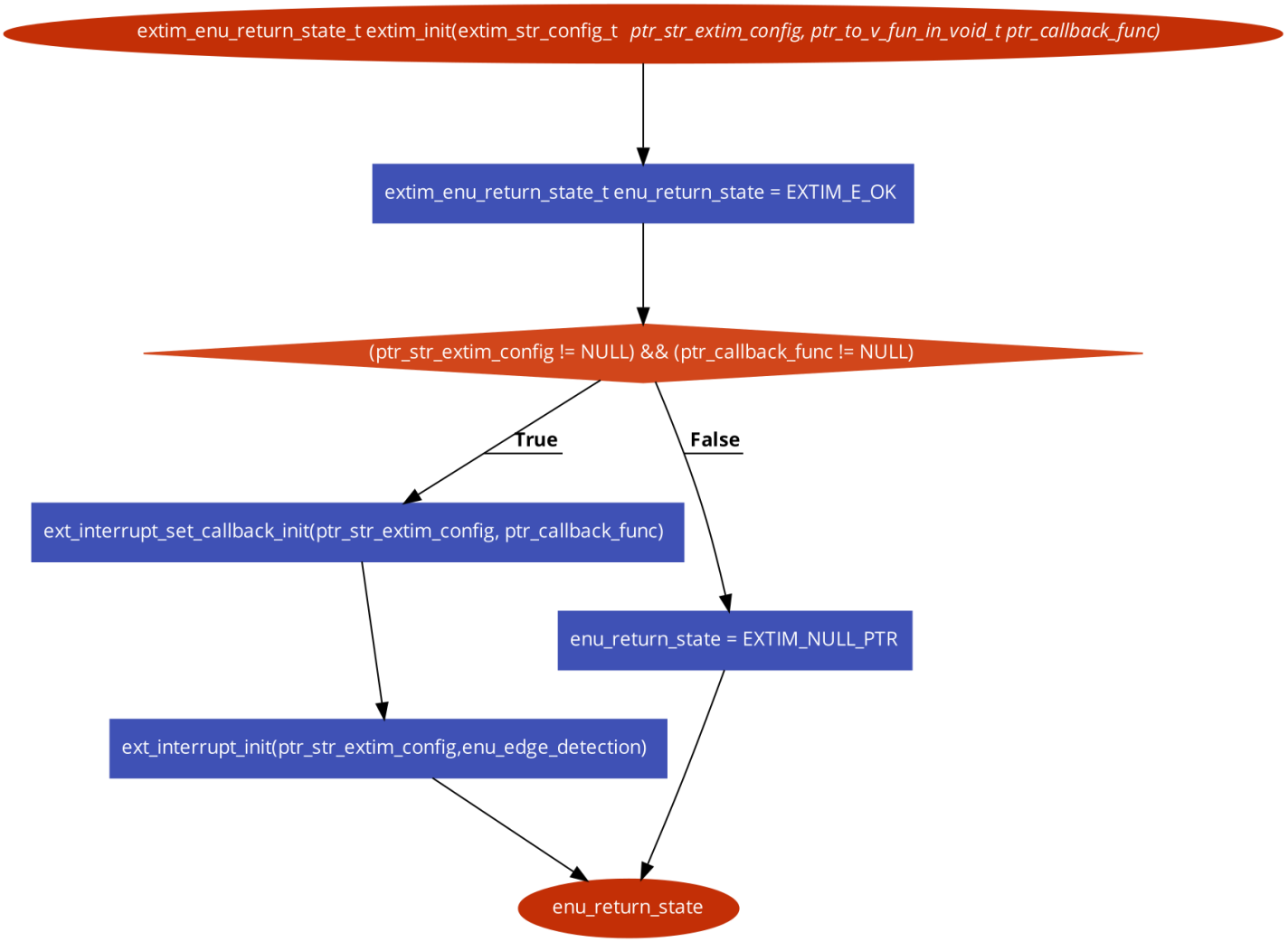

Figure 49 extim_init()

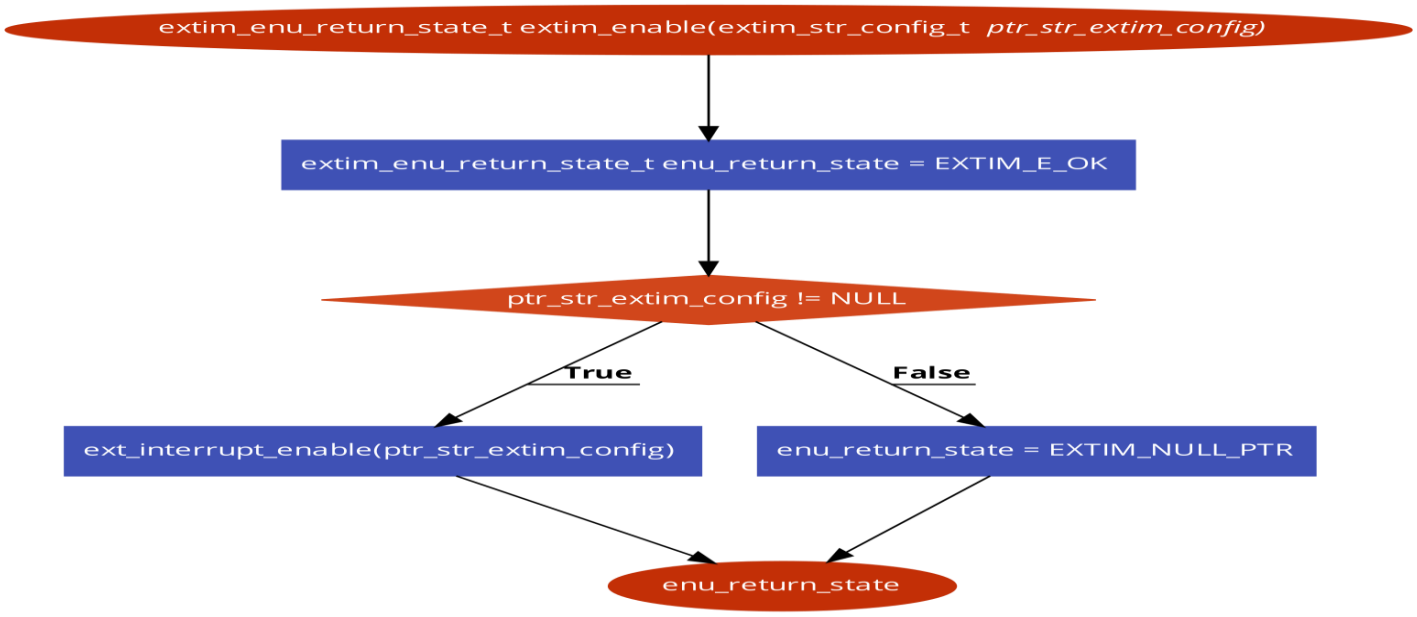

Figure 50 extim_enable()

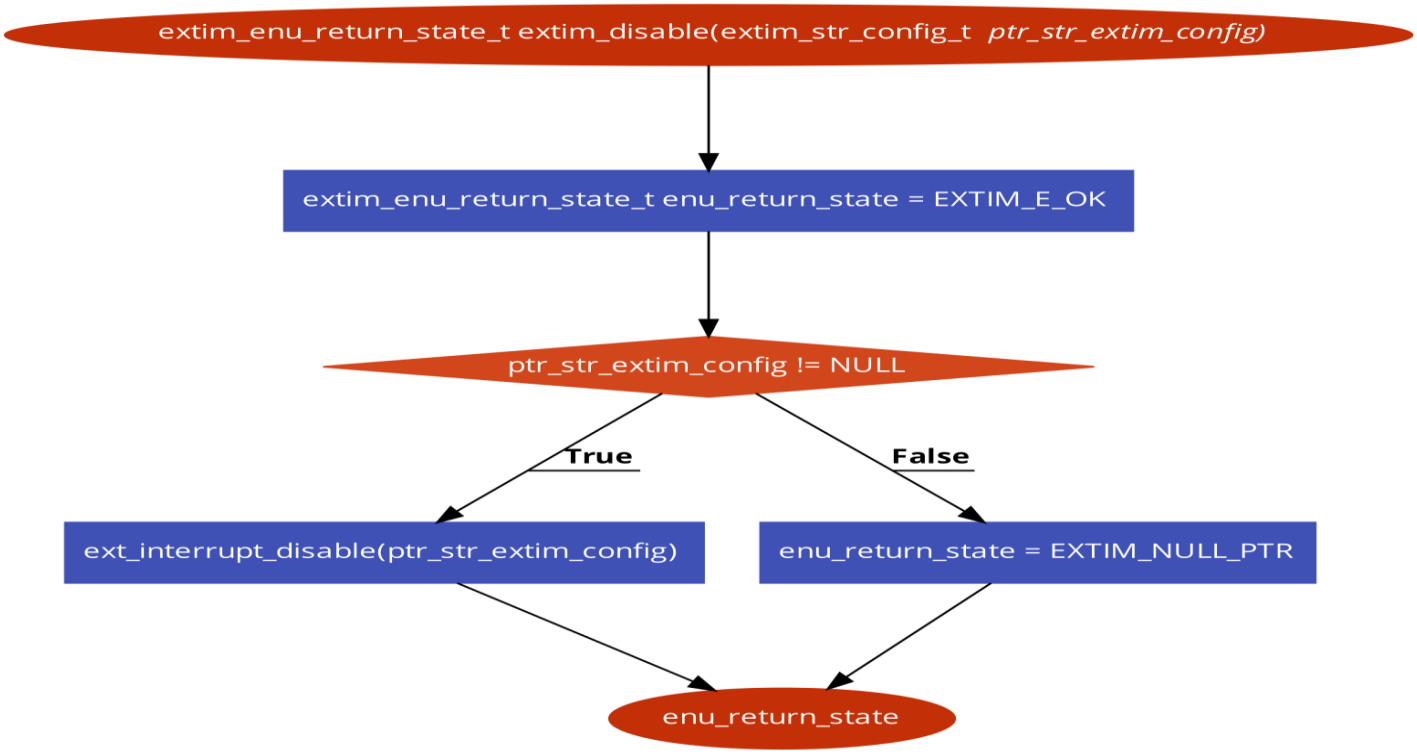

Figure 51 extim_disable()

PWM API:

Flowcharts:

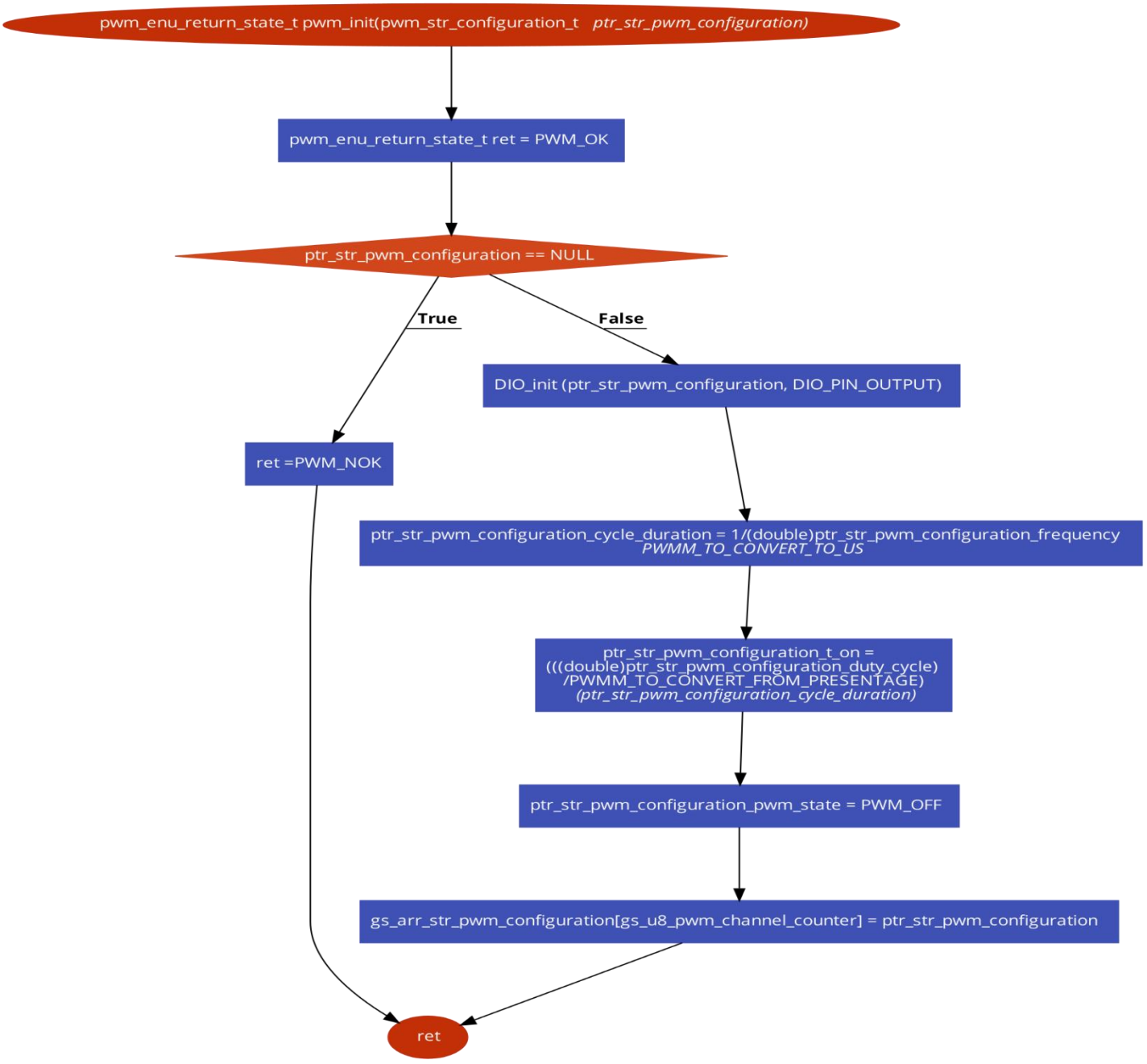

Figure 52 pwm_init

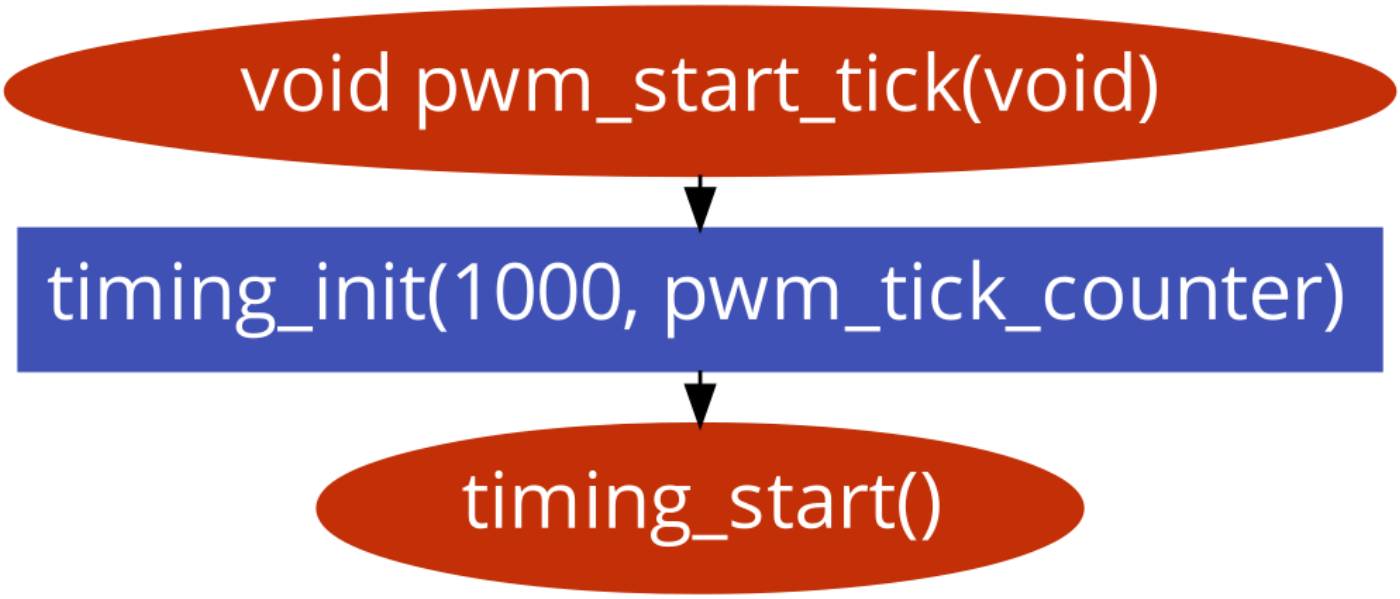

Figure 54 pwm_start_tick()

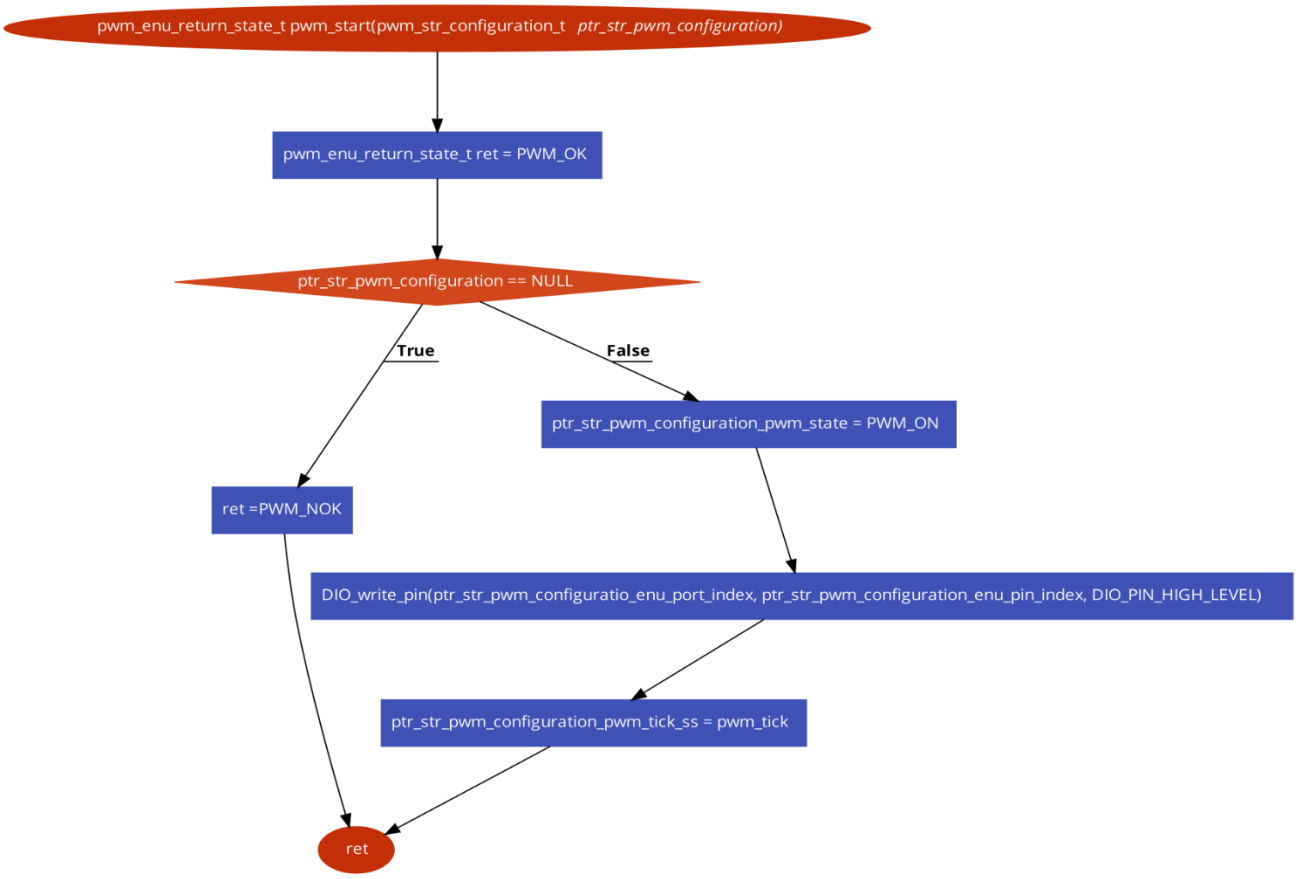

Figure 53 pwm_start()

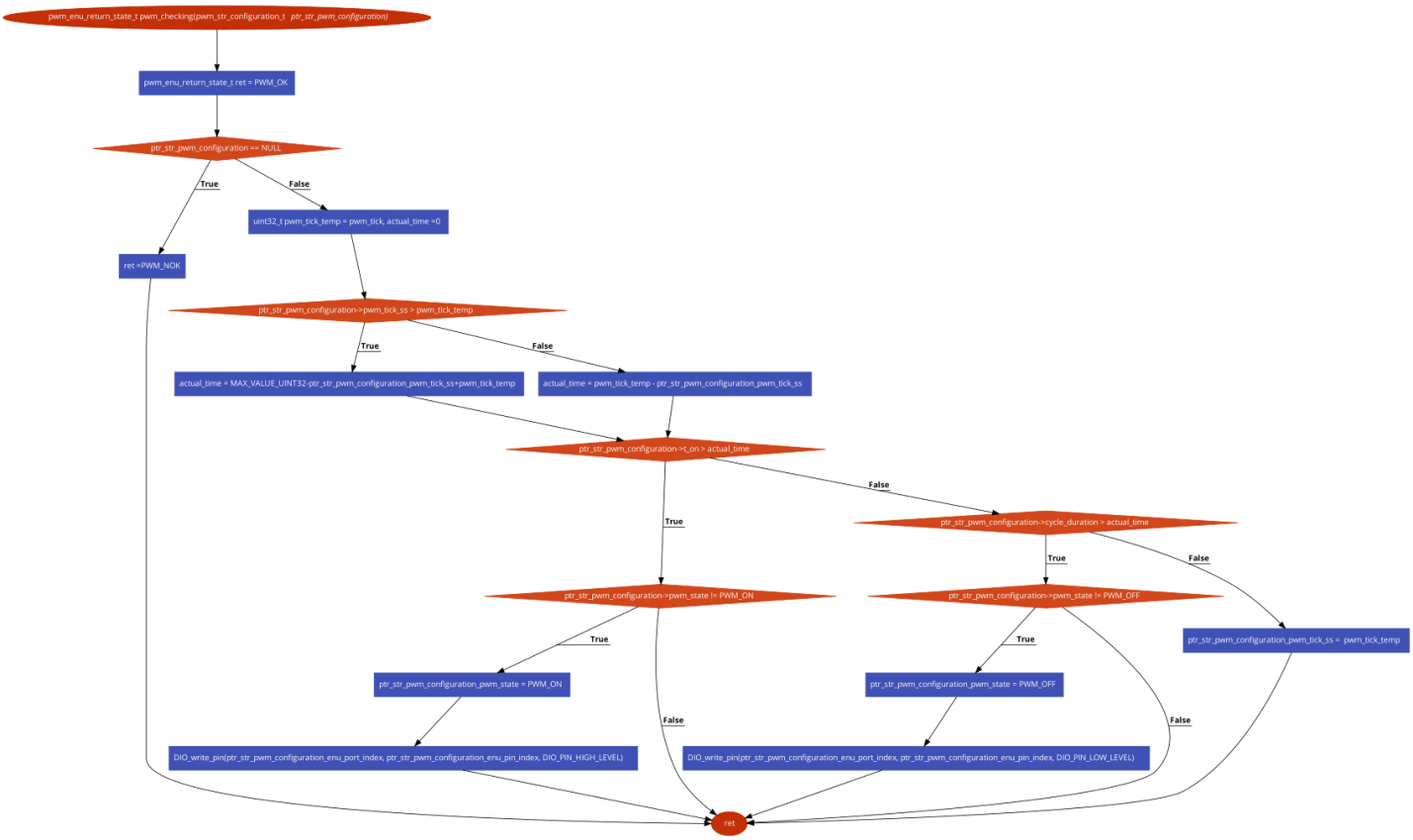

Figure 55 pwm_checking()

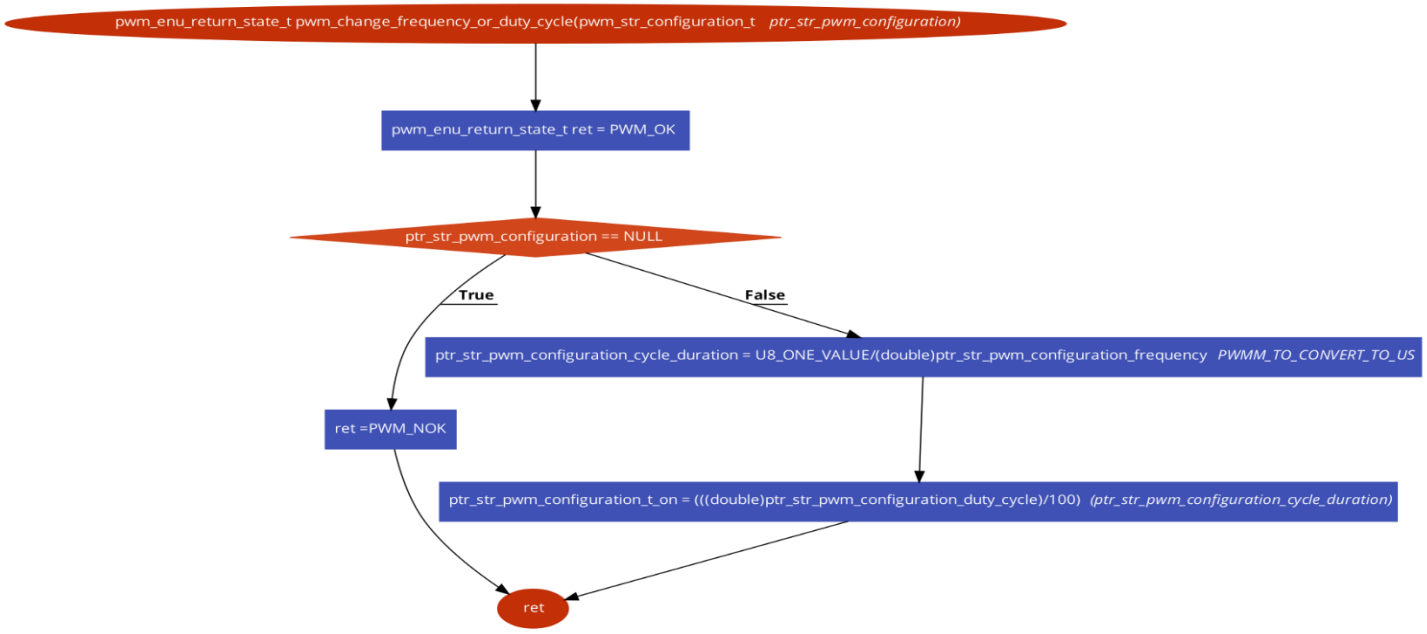

Figure 56 pwm_change_frequency_or_duty_cycle()

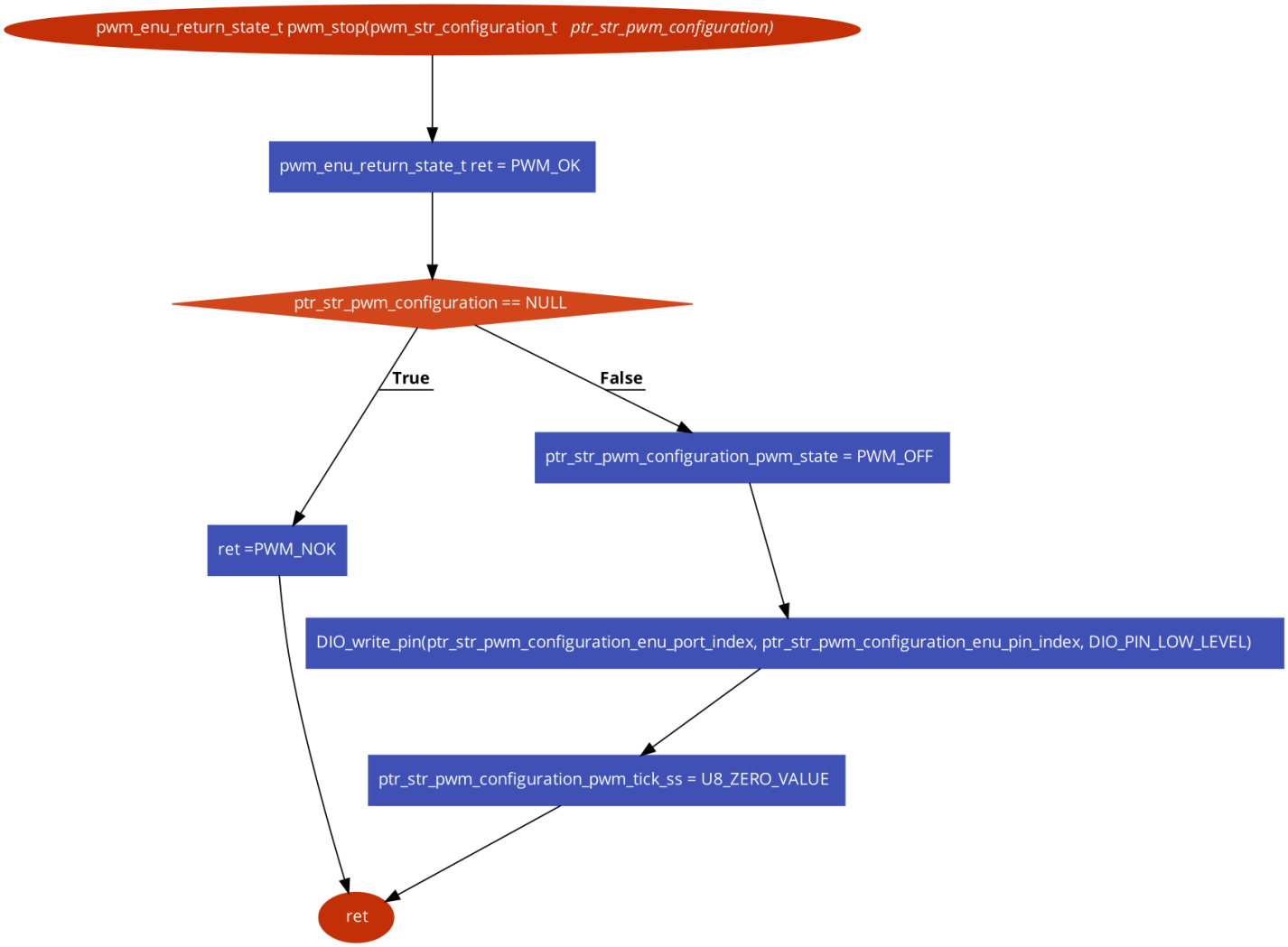

Figure 57 pwm_stop()

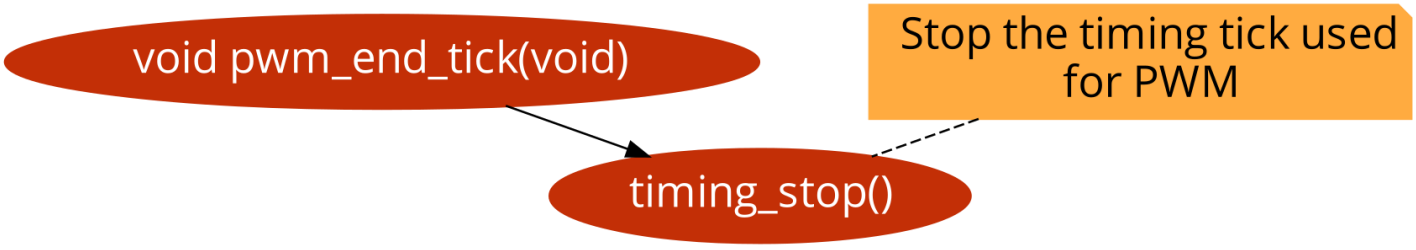

Figure 58 pwm_end_tick()

Button API:

Flowcharts:

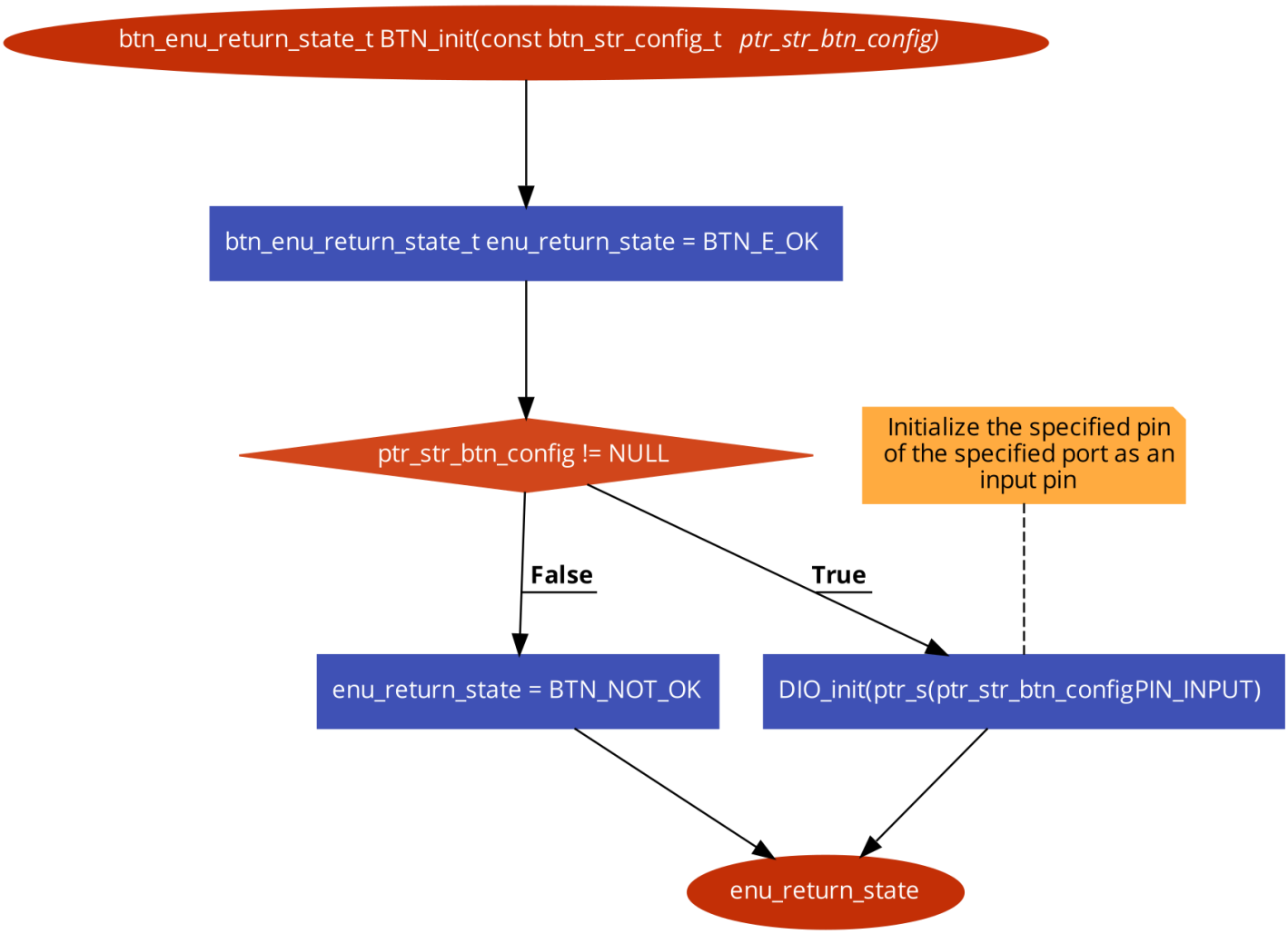

Figure 59 BTN_init()

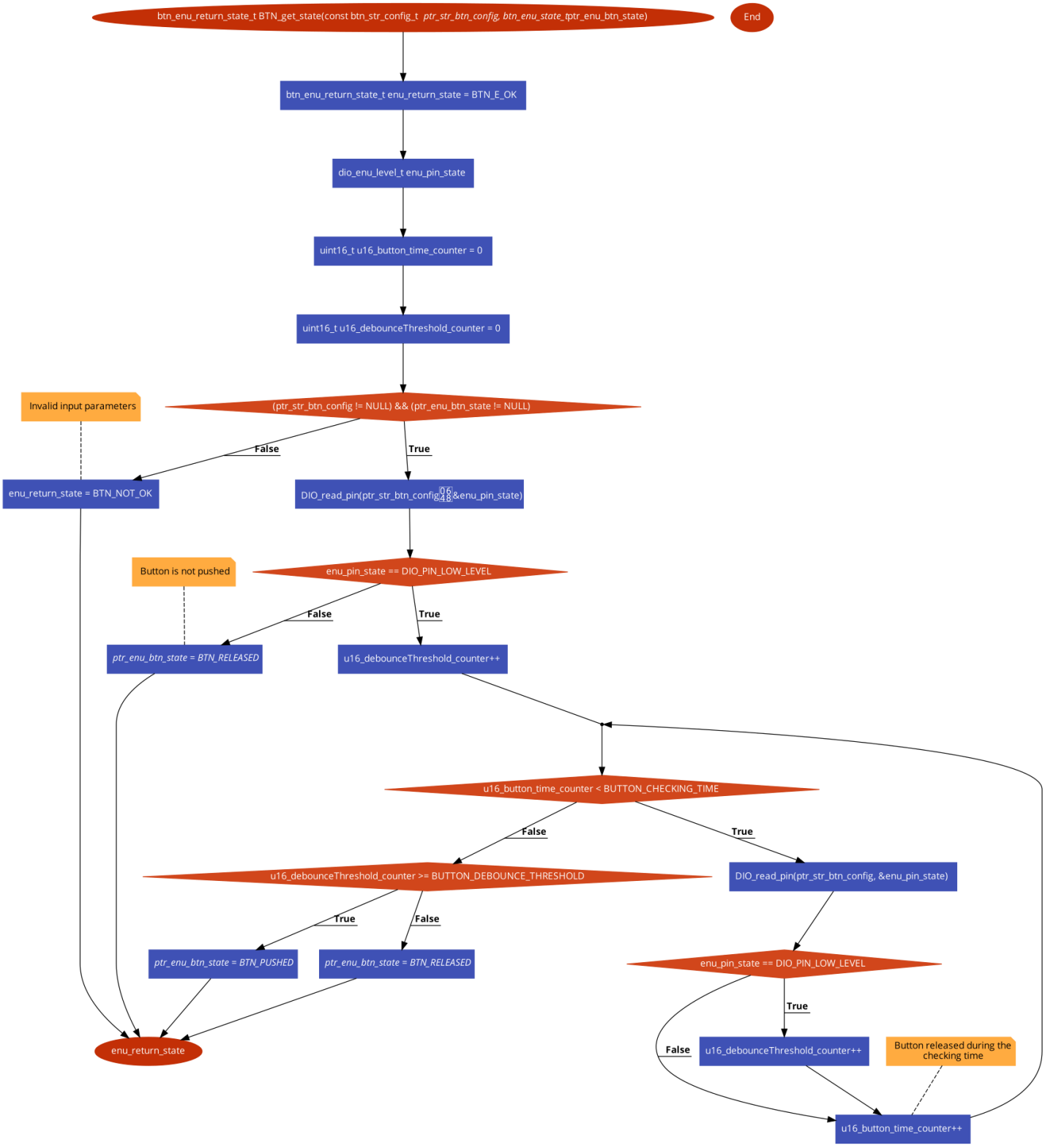

Figure 60 BTN_get_state()

App

App API:

Flowcharts:

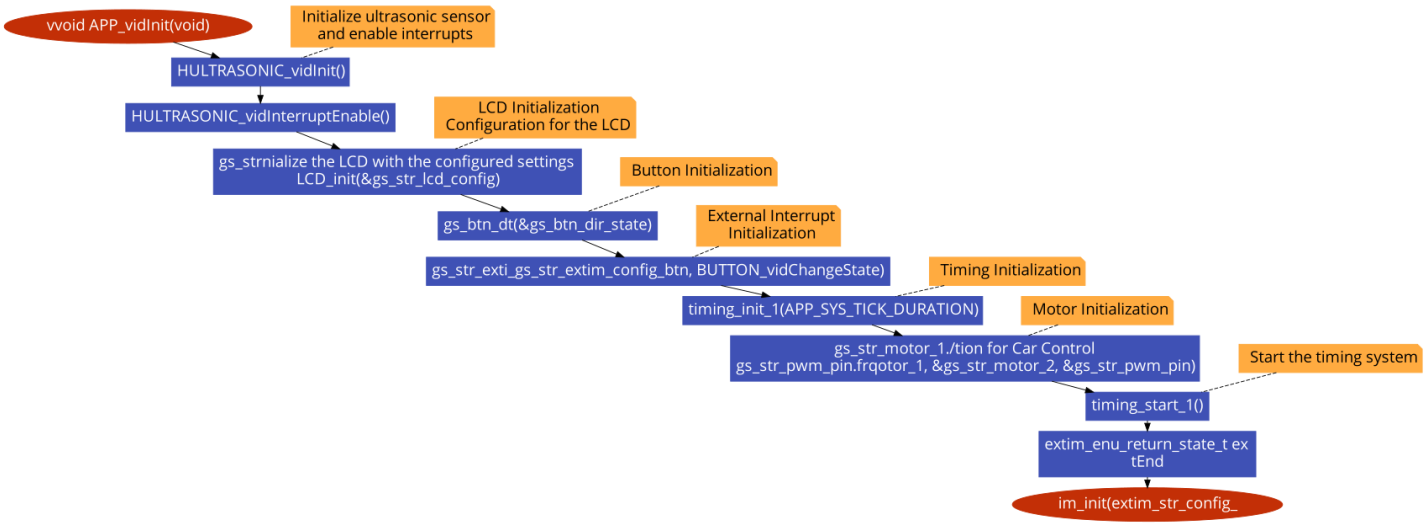

Figure 61 APP_init()

Figure 62 APP_start() added besides the pdf as SVG

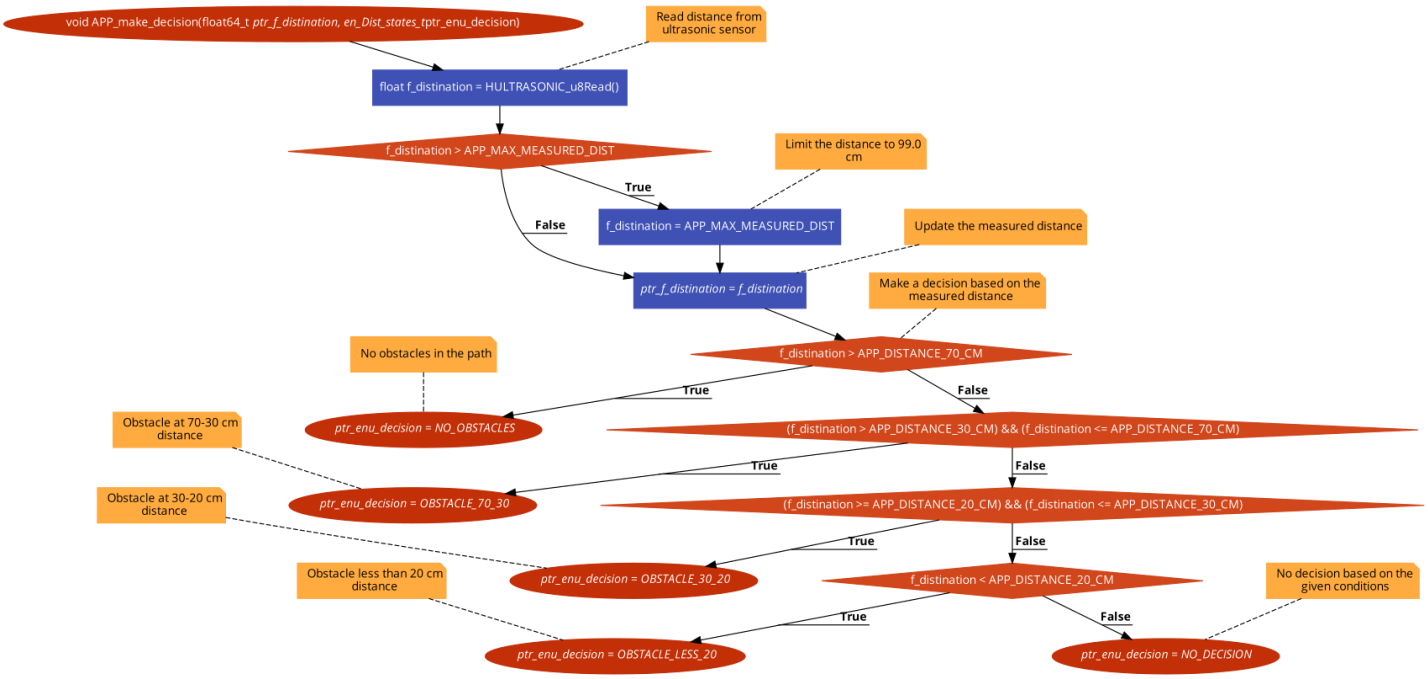

Figure 63 APP_make_decision()

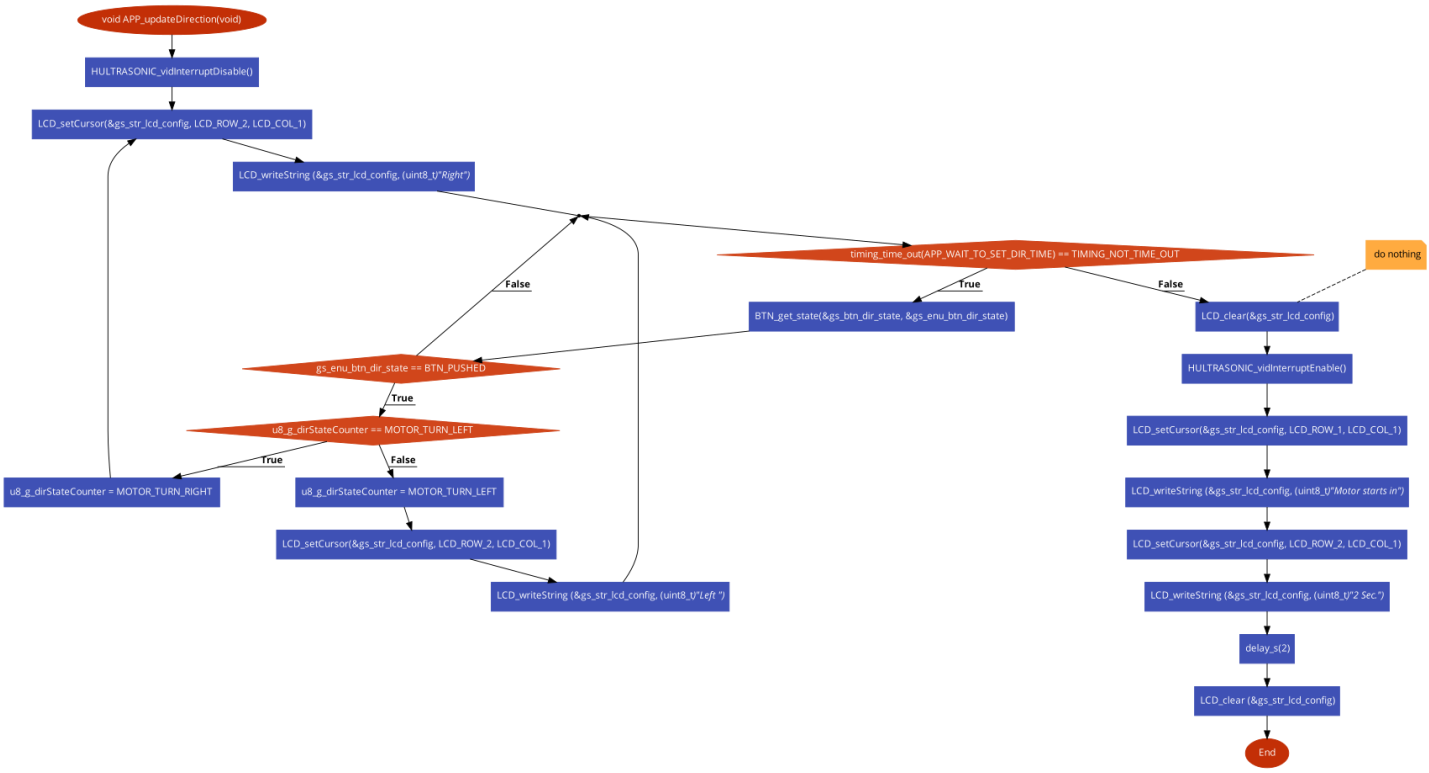

Figure 64 APP_updateDirection()

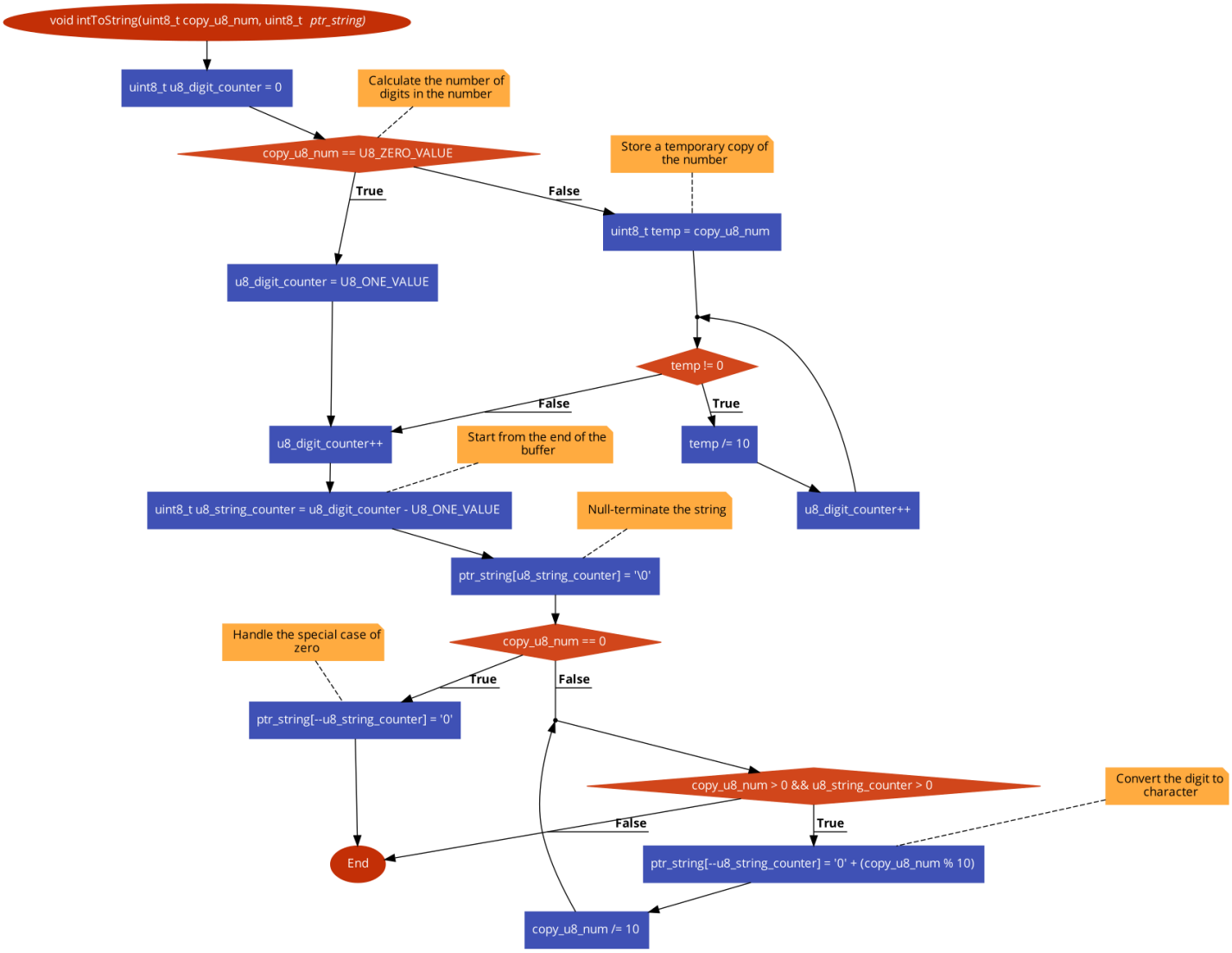

Figure 65 intToString()

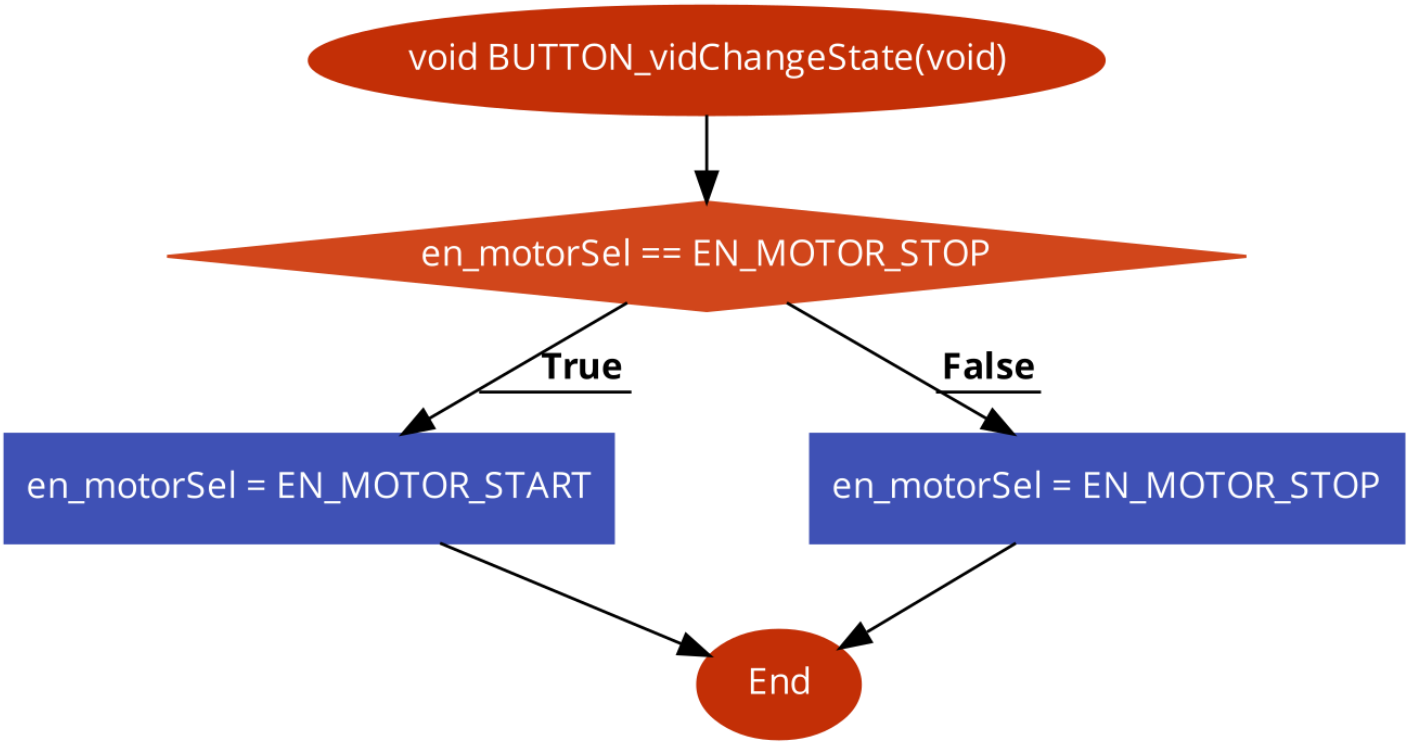

Figure 66 BUTTON_vidChangeState()