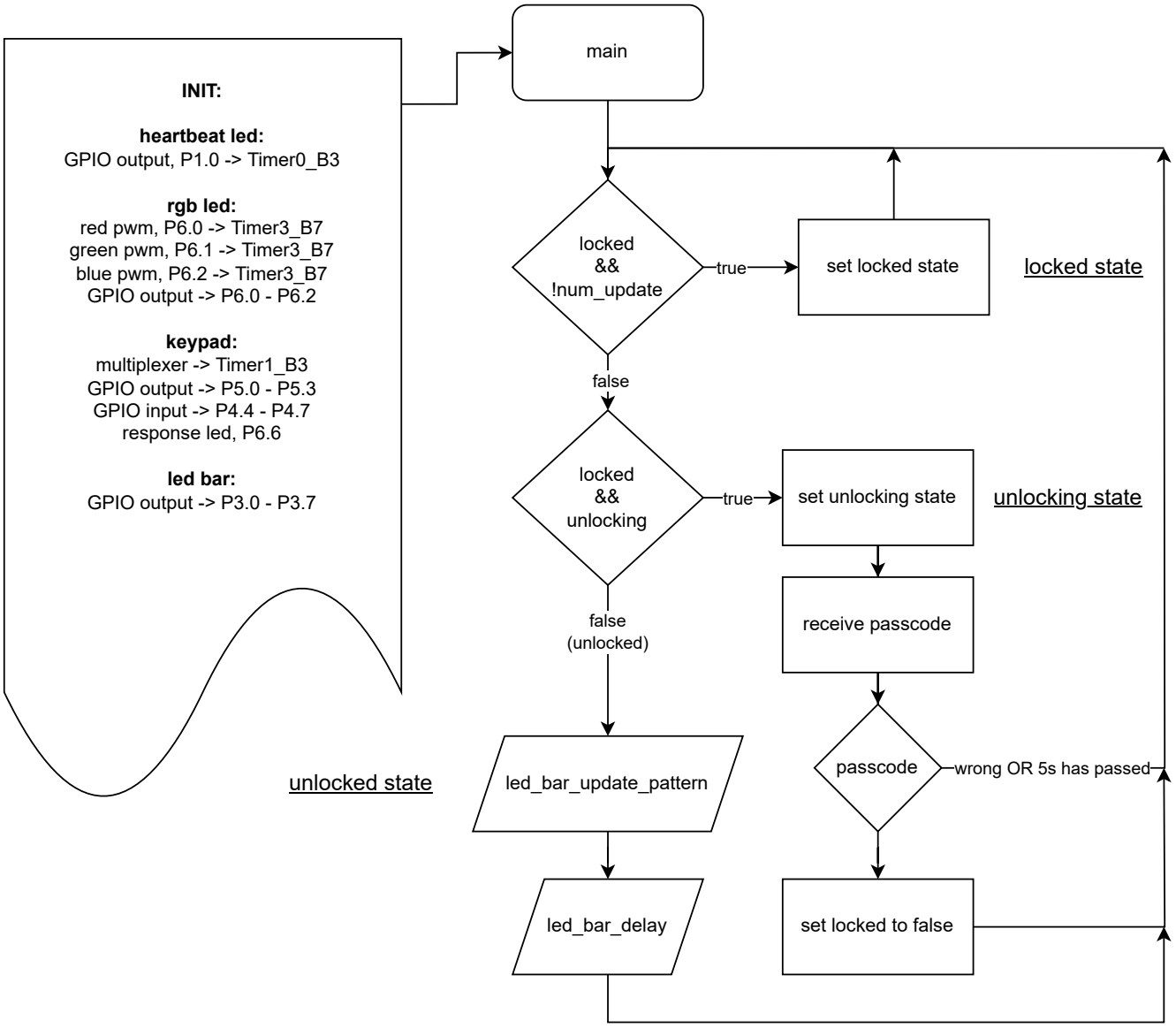
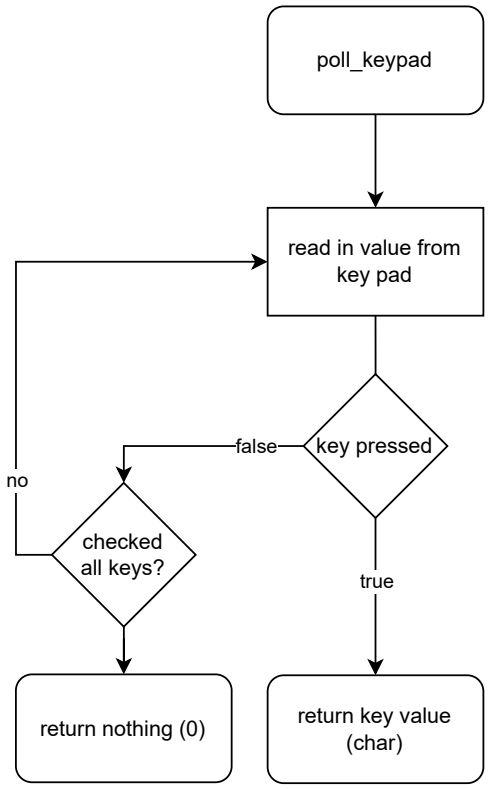
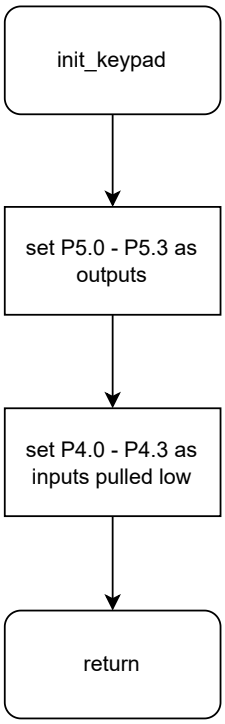
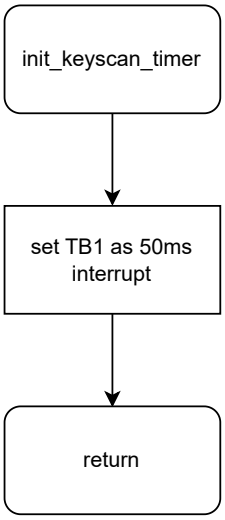
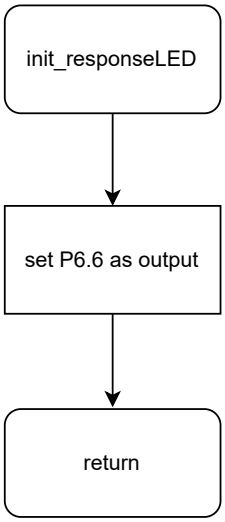
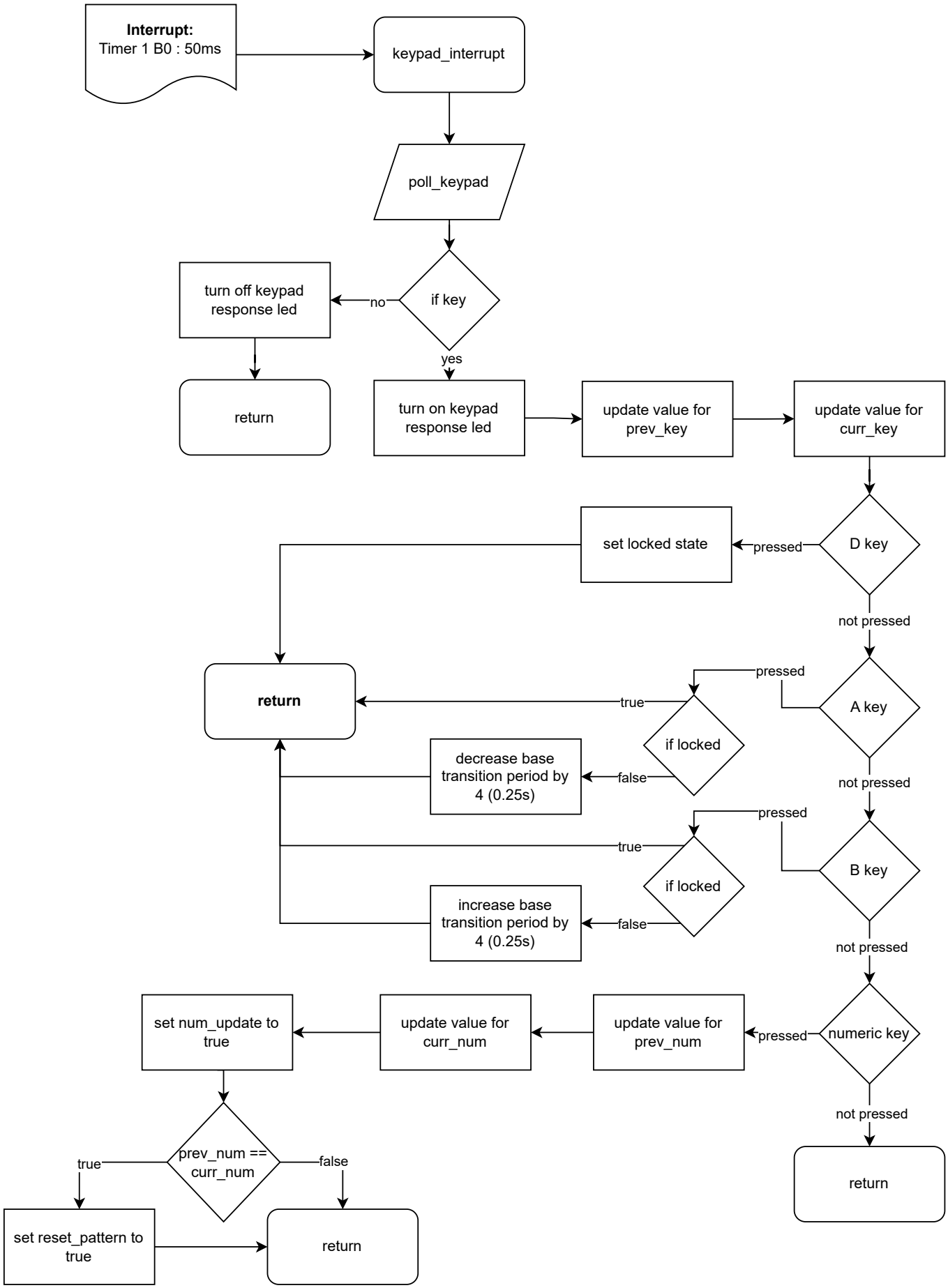

INIT:
heartbeat led:
GPIO output, P1.0 -> Timer0_B3
rgb led:
red pwm, P6.0 -> Timer3_B7
green pwm, P6.1 -> Timer3_B7
blue pwm, P6.2 -> Timer3_B7
GPIO output -> P6.0 - P6.2
keypad:
multiplexer -> Timer1_B3
GPIO output -> P5.0 - P5.3
GPIO input -> P4.4 - P4.7
response led, P6.6
led bar:
GPIO output -> P3.0 - P3.7
unlocked state
main
locked && !num_update
true
set locked state
locked state
false
locked && unlocking
true
set unlocking state
unlocking state
receive passcode
false (unlocked)
led_bar_update_pattern
passcode
wrong OR 5s has passed
led_bar_delay
set locked to false

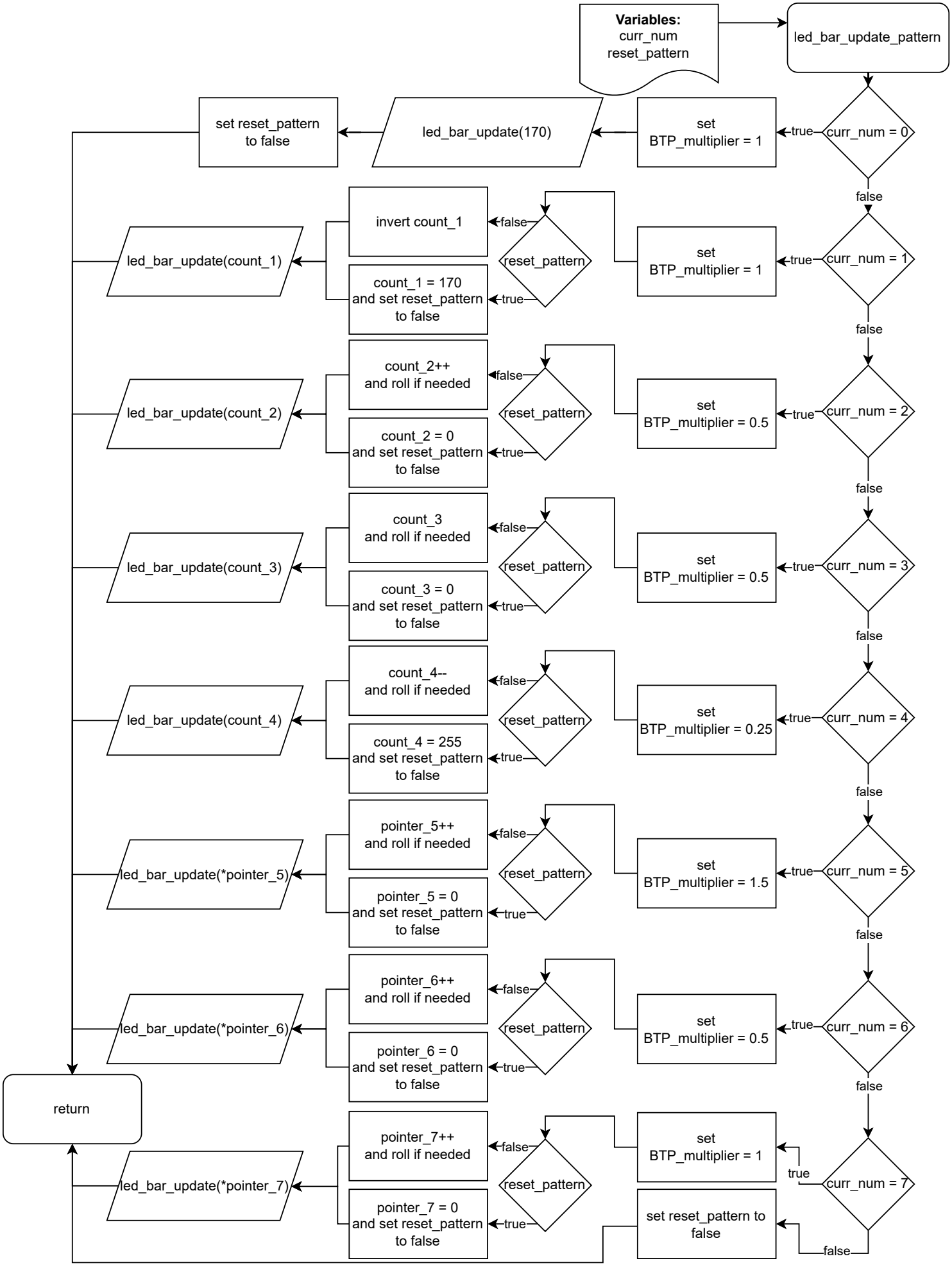

poll_keypad
read in value from key pad
key pressed
false
no
checked all keys?
true
return nothing (0)
return key value (char)

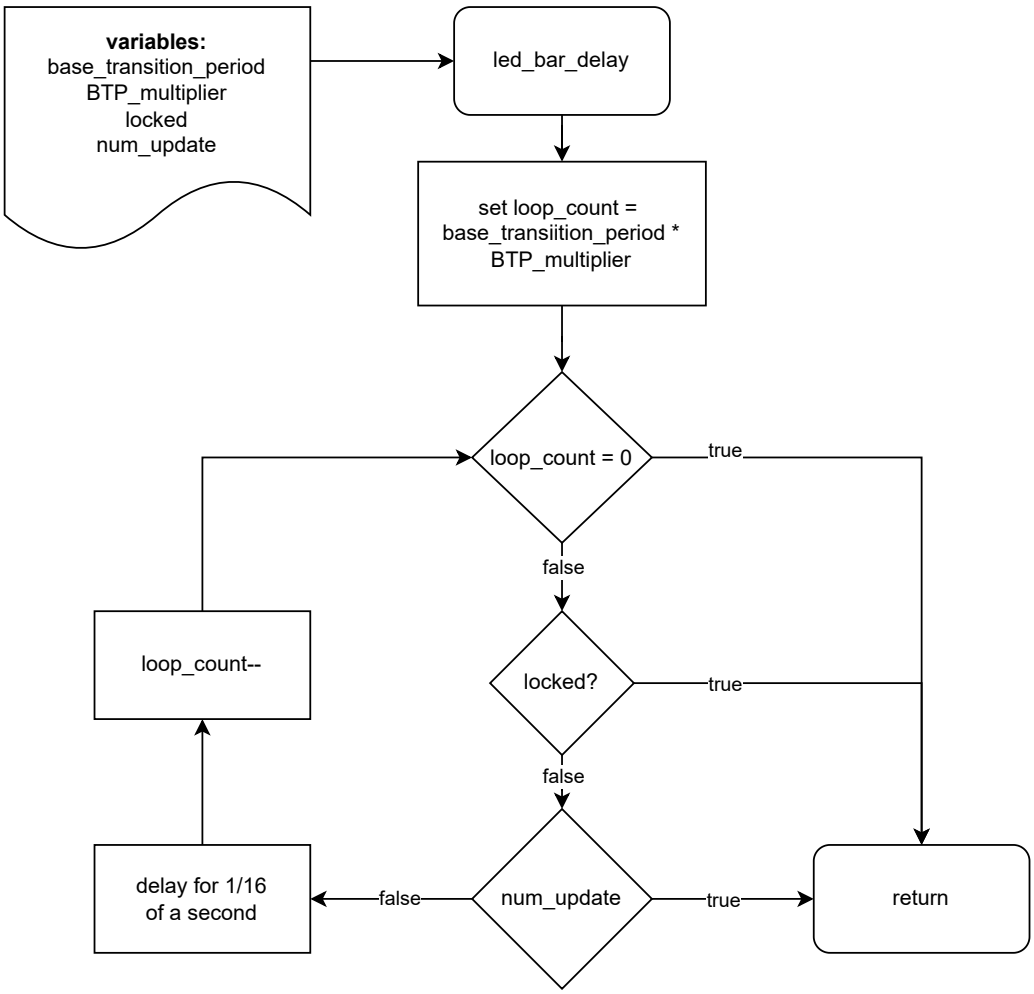

init_keypad
set P5.0 - P5.3 as outputs
set P4.0 - P4.3 as inputs pulled low
return

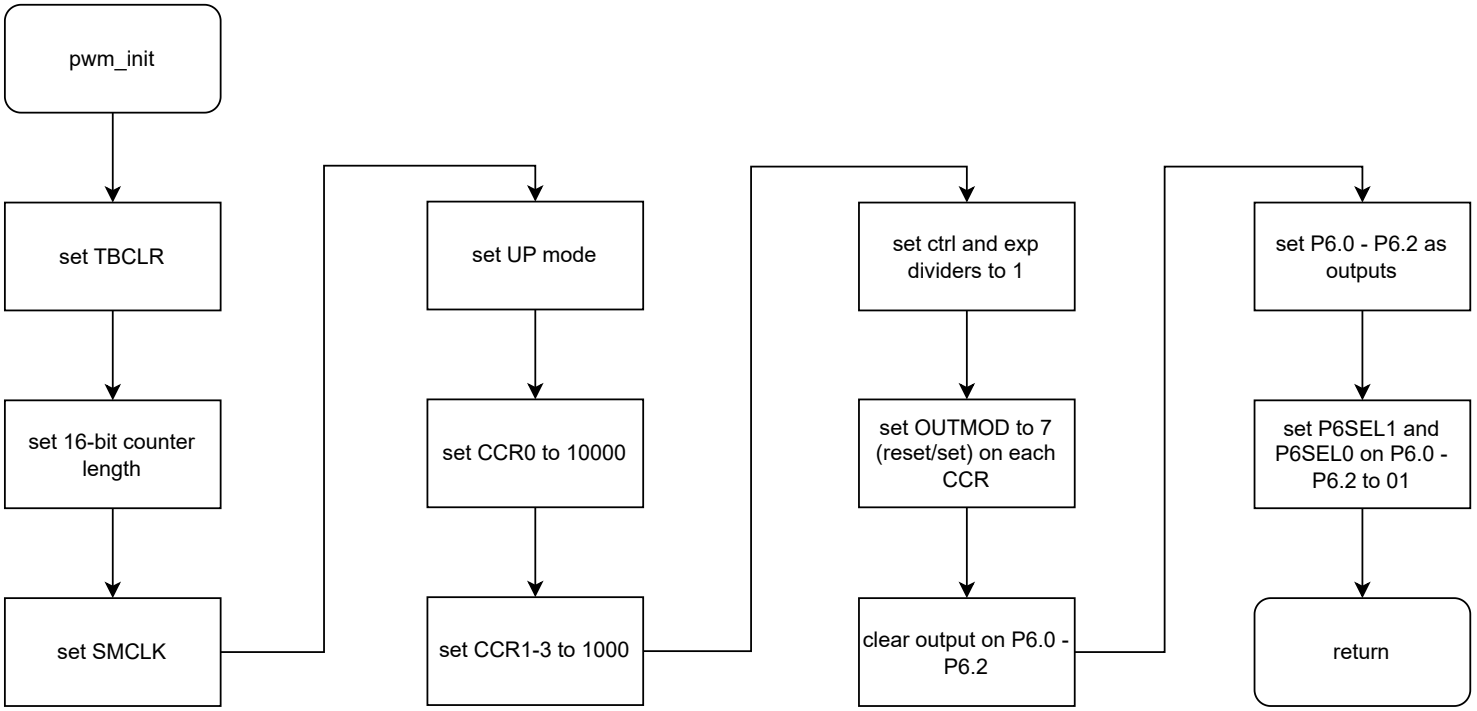

init_keyscan_timer
set TB1 as 50ms interrupt
return


init_responseLED
set P6.6 as output
return

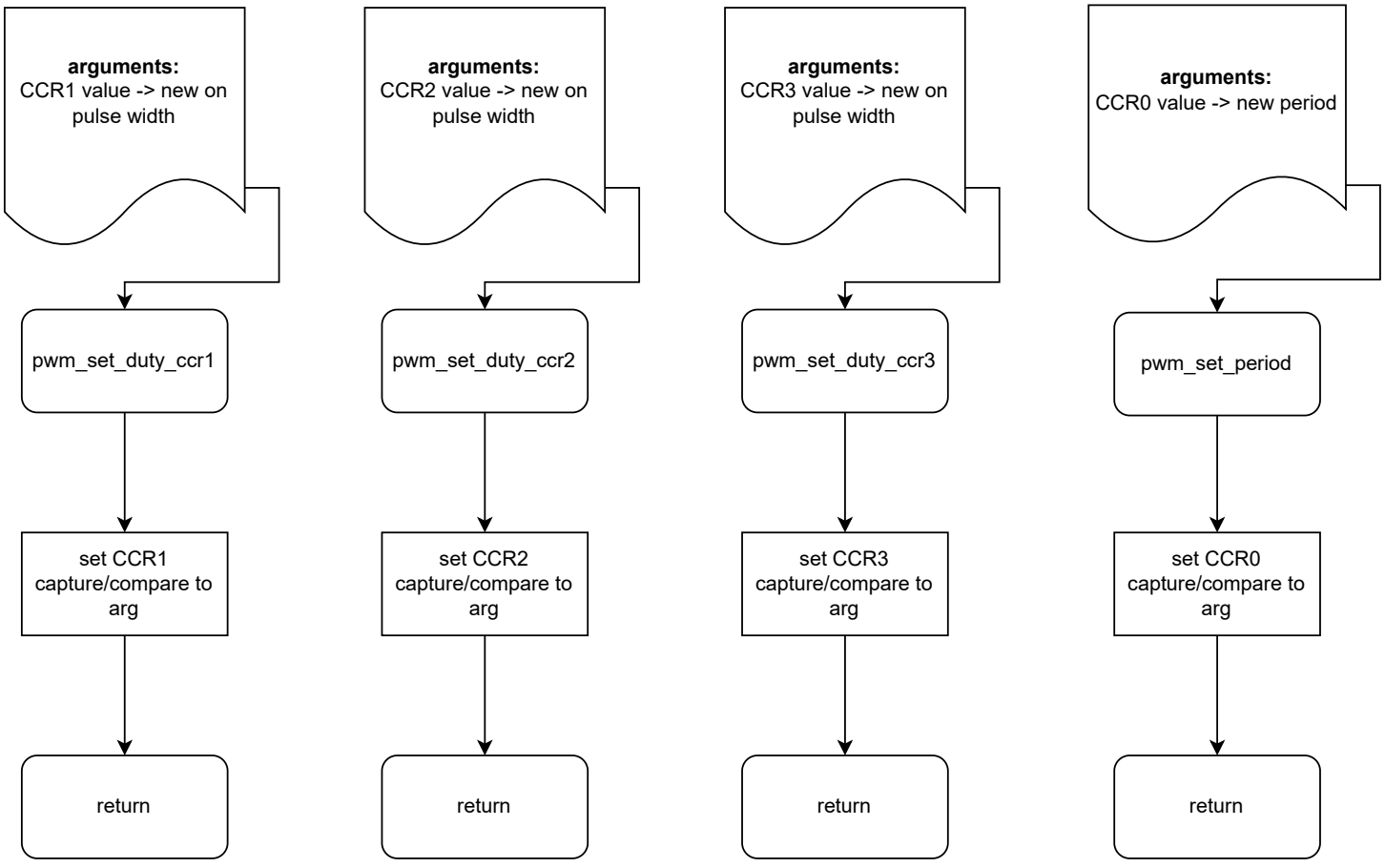

Interrupt:
Timer 1 B0 : 50ms
keypad_interrupt
poll_keypad
if key
no
turn off keypad response led
return
yes
turn on keypad response led
update value for prev_key
update value for curr_key
D key
pressed
set locked state
not pressed
A key
pressed
true
if locked
false
decrease base transition period by 4 (0.25s)
return
not pressed
B key
pressed
true
if locked
false
increase base transition period by 4 (0.25s)
not pressed
numeric key
pressed
update value for prev_num
update value for curr_num
set num_update to true
prev_num == curr_num
true
false
set reset_pattern to true
return
not pressed
return

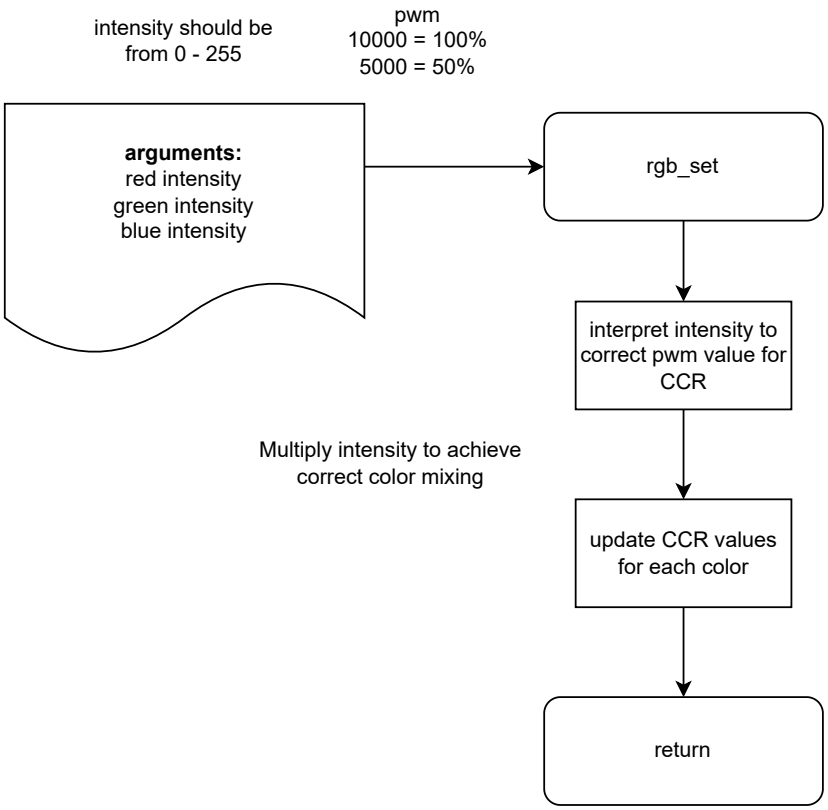

Variables:
curr_num
reset_pattern
led_bar_update_pattern
curr_num = 0
true
set
BTP_multiplier = 1
led_bar_update(170)
set reset_pattern
to false
false
curr_num = 1
true
set
BTP_multiplier = 1
reset_pattern
false
invert count_1
led_bar_update(count_1)
true
count_1 = 170
and set reset_pattern
to false
curr_num = 2
true
set
BTP_multiplier = 0.5
reset_pattern
false
count_2++
and roll if needed
led_bar_update(count_2)
true
count_2 = 0
and set reset_pattern
to false
curr_num = 3
true
set
BTP_multiplier = 0.5
reset_pattern
false
count_3
and roll if needed
led_bar_update(count_3)
true
count_3 = 0
and set reset_pattern
to false
curr_num = 4
true
set
BTP_multiplier = 0.25
reset_pattern
false
count_4--
and roll if needed
led_bar_update(count_4)
true
count_4 = 255
and set reset_pattern
to false
curr_num = 5
true
set
BTP_multiplier = 1.5
reset_pattern
false
pointer_5++
and roll if needed
led_bar_update(*pointer_5)
true
pointer_5 = 0
and set reset_pattern
to false
curr_num = 6
true
set
BTP_multiplier = 0.5
reset_pattern
false
pointer_6++
and roll if needed
led_bar_update(*pointer_6)
true
pointer_6 = 0
and set reset_pattern
to false
curr_num = 7
true
set
BTP_multiplier = 1
reset_pattern
false
pointer_7++
and roll if needed
led_bar_update(*pointer_7)
true
pointer_7 = 0
and set reset_pattern
to false
set reset_pattern to
false
false
return

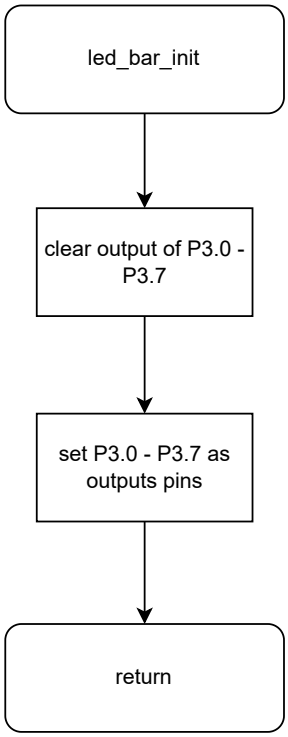

variables:
base_transition_period
BTP_multiplier
locked
num_update
led_bar_delay
set loop_count =
base_transiition_period *
BTP_multiplier
loop_count = 0
true
false
loop_count--
locked?
true
false
delay for 1/16
of a second
false
num_update
true
return



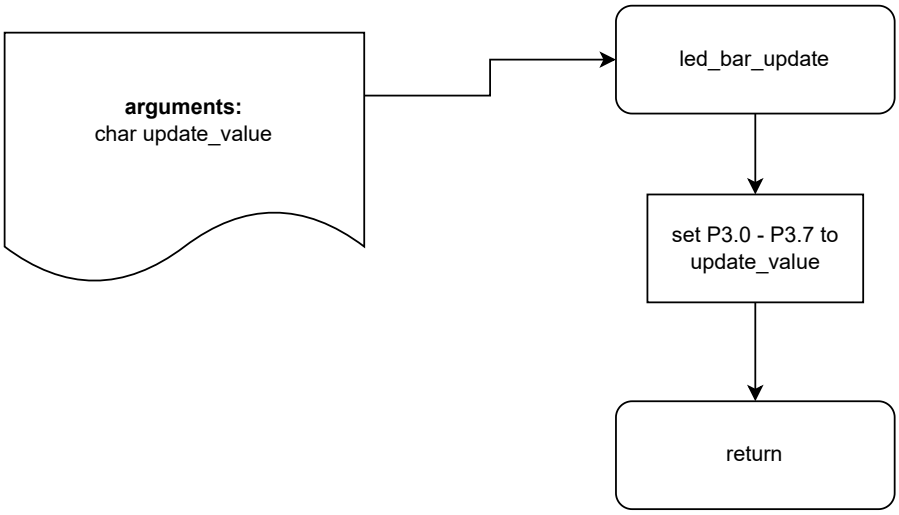
These should be implemented as macros

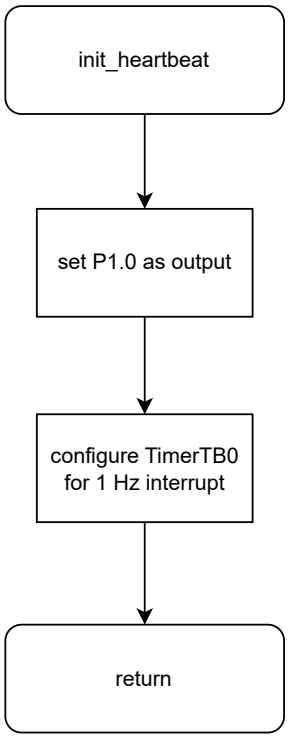


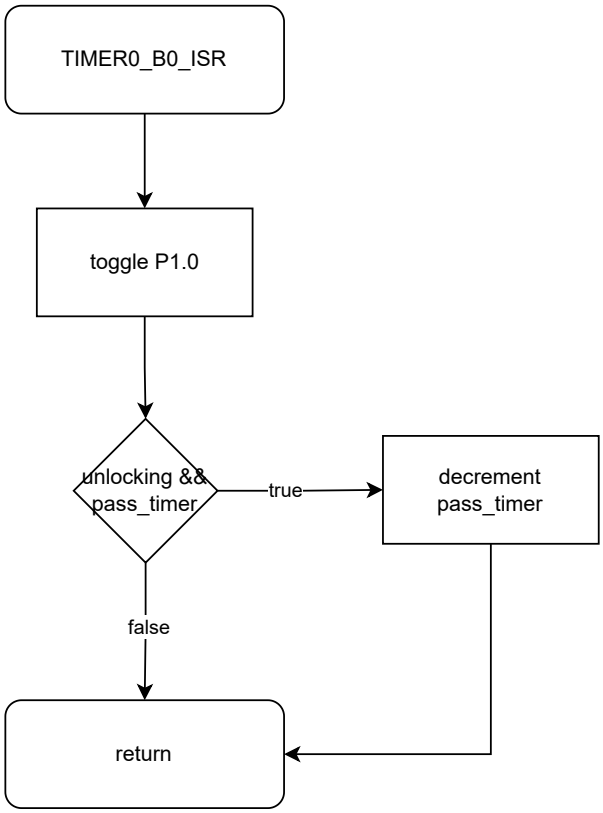

intensity should be
from 0 - 255
pwm
10000 = 100%
5000 = 50%
arguments:
red intensity
green intensity
blue intensity
rgb_set
interpret intensity to
correct pwm value for
CCR
Multiply intensity to achieve
correct color mixing
update CCR values
for each color
return


led_bar_init
clear output of P3.0 - P3.7
set P3.0 - P3.7 as outputs pins
return

Macro


arguments:
char update_value
led_bar_update
set P3.0 - P3.7 to update_value
return


init_heartbeat
set P1.0 as output
configure TimerTB0 for 1 Hz interrupt
return


TIMER0_B0_ISR
toggle P1.0
unlocking && pass_timer
true
decrement pass_timer
false
return