Youseif essam

Led sequene v1

Project description:

1. **The car starts initially from 0 speed**
2. **When PB1 is pressed, the car will move forward after 1 second**
3. **The car will move forward to create the longest side of the rectangle for 3 seconds with 50% of its maximum speed**
4. **After finishing the first longest side the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second**
5. **The car will move to create the short side of the rectangle at 30% of its speed for 2 seconds**
6. **After finishing the shortest side, the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second**
7. **Steps 3 to 6 will be repeated infinitely until you press the stop button (PB2)**
8. **PB2 acts as a sudden break, and it has the highest priority**

Layered architecture:

Application
ECUAL
MCAL
Microcontroller

ECUAL: sensors and actuators independent on target contain(Leds and Buttons)

MCAL : this contains all drivers and APIS (Dio, interrupt)

System modules:

Application			
Leds	Buttons	Motors	PWM
Dio	INTERRUPT	Timer	
Microcontroller			

Timer: will interact with leds and motors to determine the time of on and off

Dio: will interact with leds ,buttons and motors to determine state of both.

PWM : will interact with motors to determine the speed.

APIs:

Dio_API:

dio_init: this function take pin number and port name and directon to determine state of pin

dio_write: this function take pin number to write data on bin

dio_read: this function take pin number to read data on bin

dio_toggle:this function to togglr state of pin

Dio_writeport: this function to write value in all port and determine which pin will be high

Button_API:

Button_init : function to intialize state of button

Button_read: function to take value of button is pressed or not

Leds_API:

led_init: this function take pin number and state of this pin will be output

ledon : this function take pin number and let led to be on

ledof: this function take pin number and let led to be off

Led_toggle: this function to toggle value in pin

INTERRUPT _API:

Sei: this function set global interrupt through set status register

Cli: this function clear global interrupt through clear status register

Isr: this function take interrupt from button and make action

TIMER _API:

init_timer: this function to choose mode of timer and initial value of timer.

set_prescaler: this function to let timer start and if there are prescaler or not.

stop_timer:this function to stop timer to stop of counting.

Delay:this function take seconds which want to delay

MOTOR _API:

MOTOR_init: this function take port name and pin number which motor pin will be added on it.

Motor_move: this function let motors to move.

Motor_shut_down: this function to shut down all motors from working.

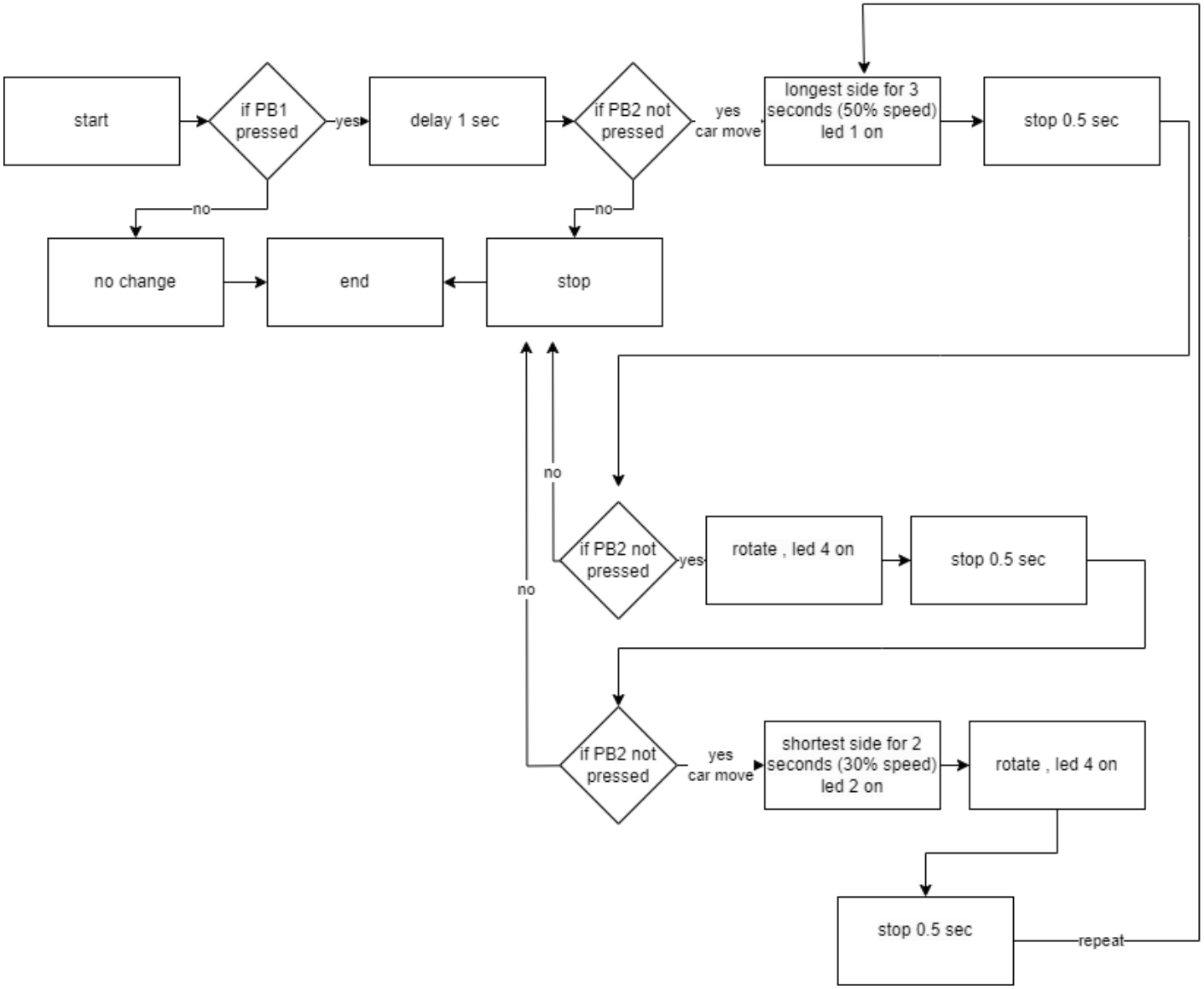
PWM_API:

PWM_init: this function to initialize PWM.

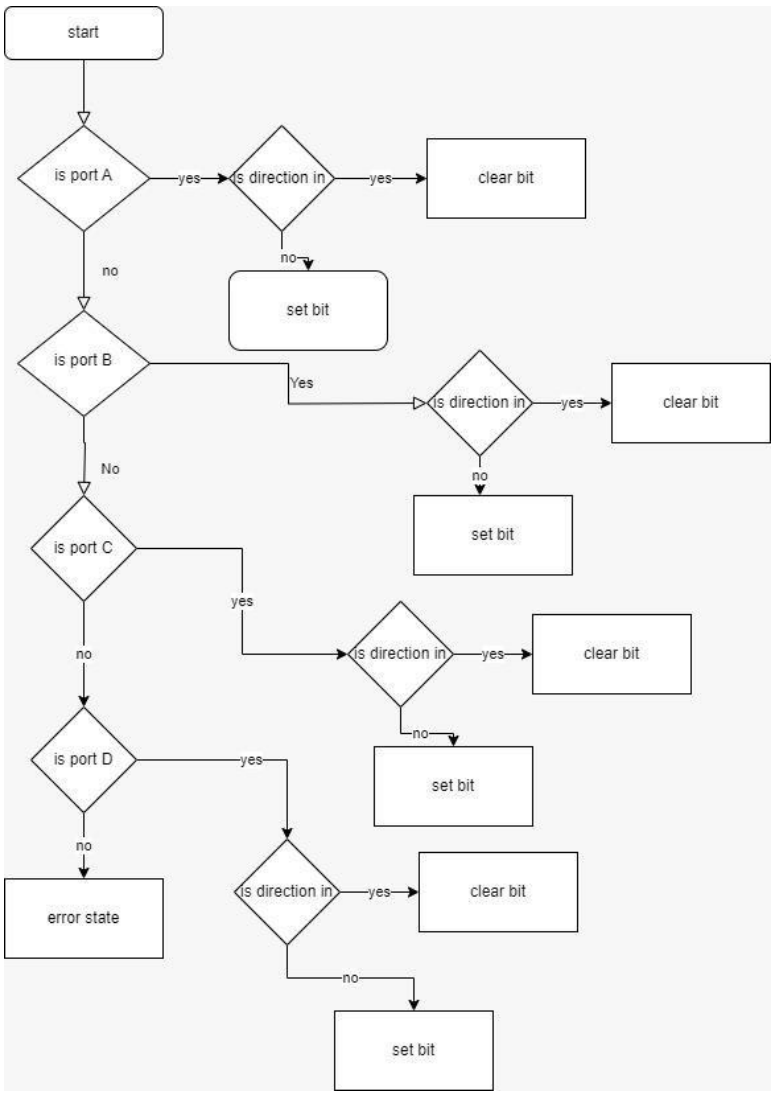
PWM_ocr_value : this function determine percentage of speed.

PWM_ocr_init: this function to initialize ocr pin.

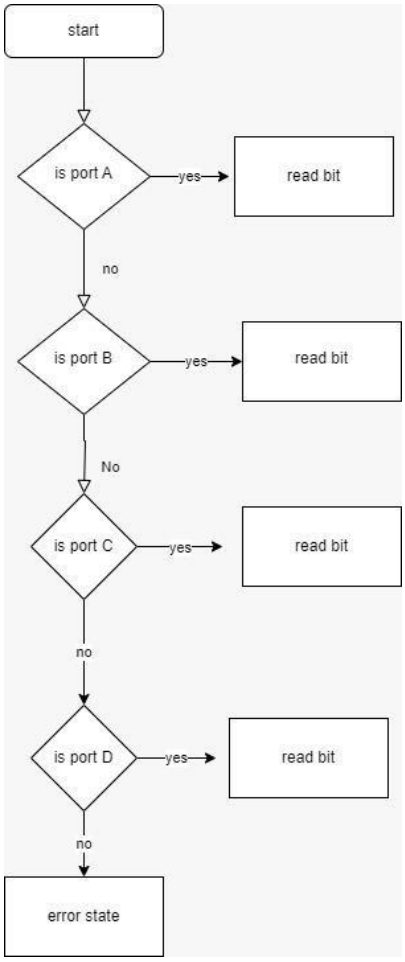
flow chart:


start
if PB1 pressed
yes
delay 1 sec
if PB2 not pressed
yes
car move
longest side for 3 seconds (50% speed) led 1 on
stop 0.5 sec
no
no
no change
end
stop
no
if PB2 not pressed
yes
rotate , led 4 on
stop 0.5 sec
no
if PB2 not pressed
yes
car move
shortest side for 2 seconds (30% speed) led 2 on
rotate , led 4 on
stop 0.5 sec
repeat

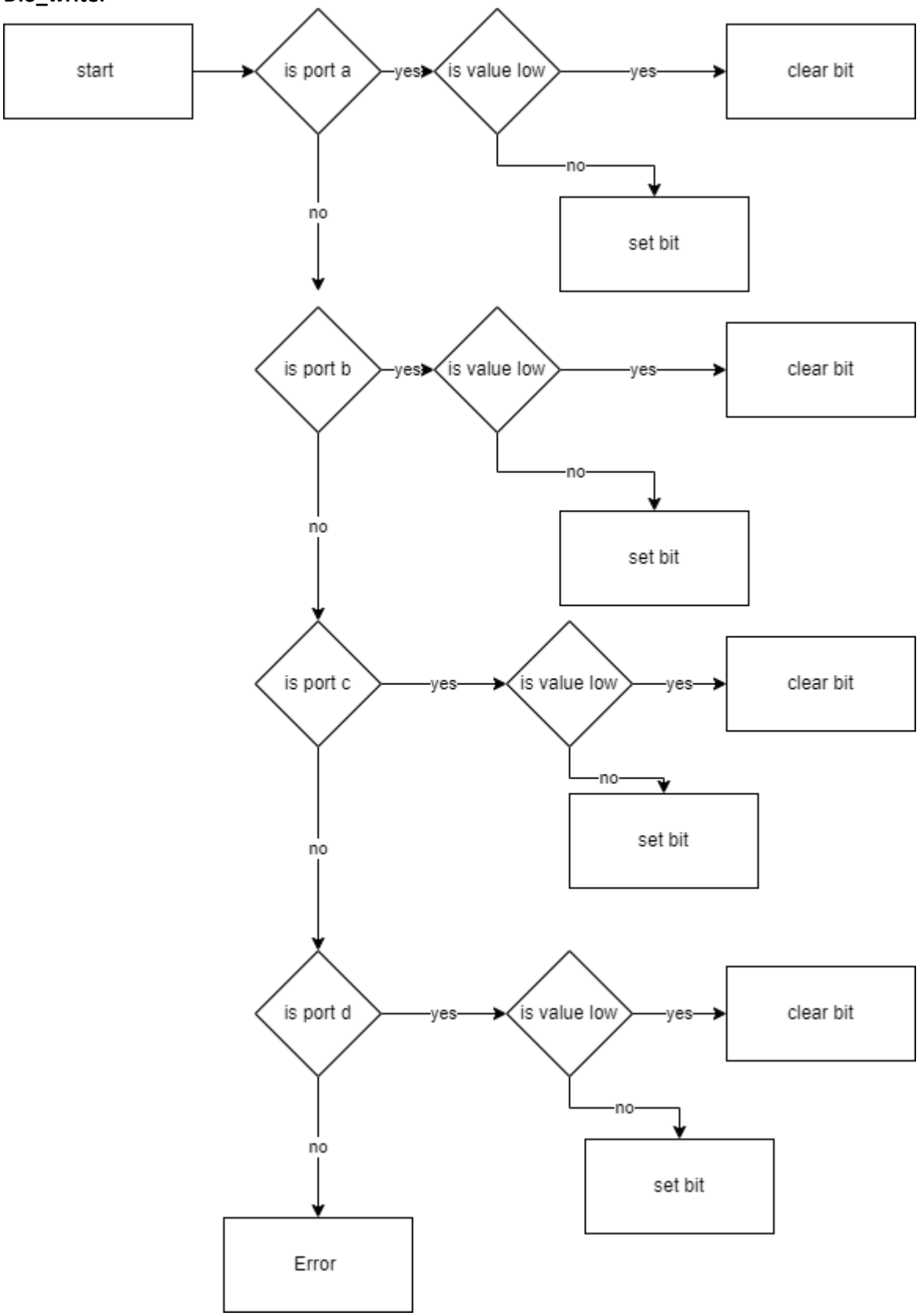
Dio init flow chart:



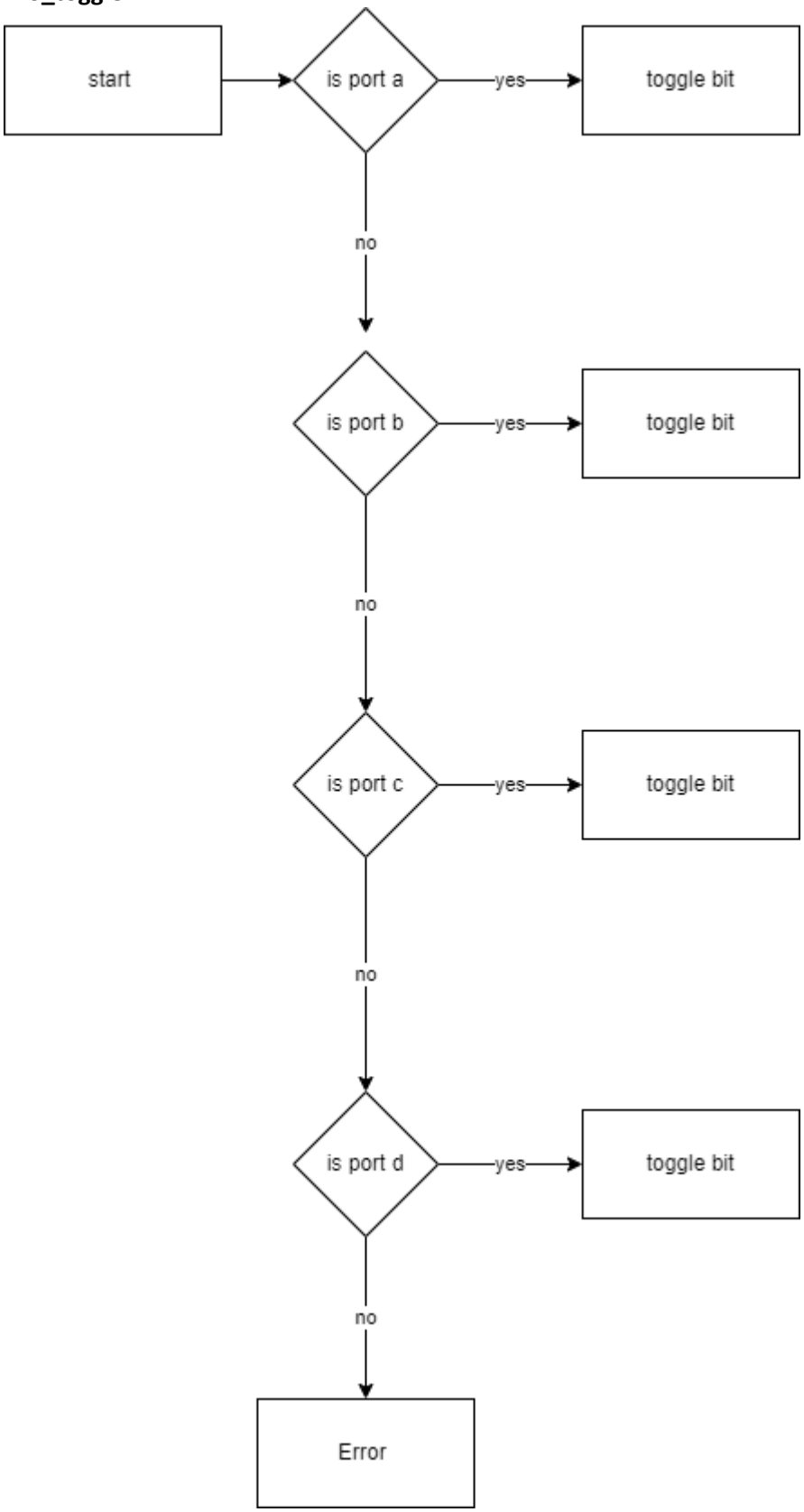
Dio_read flow chart:



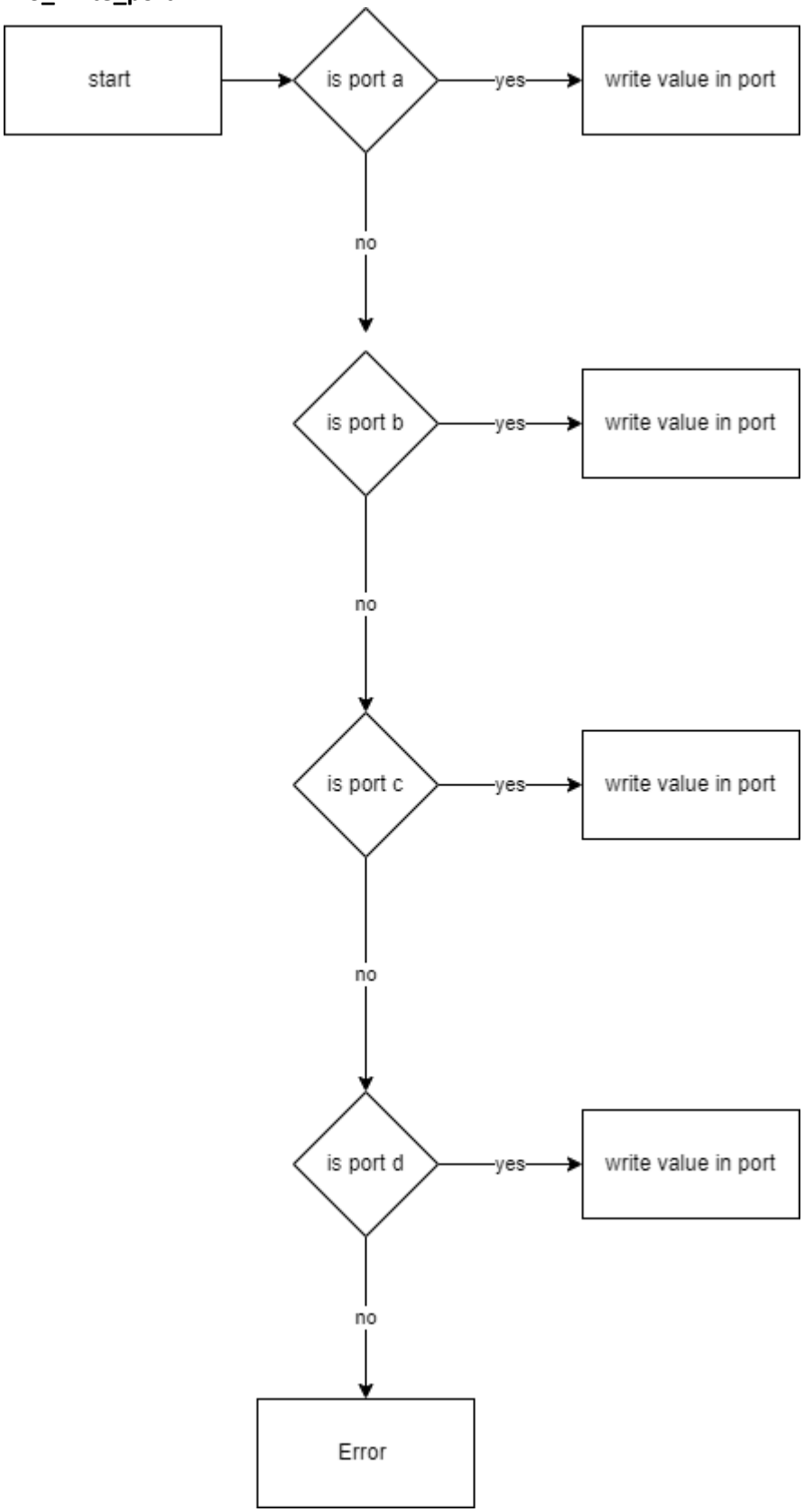
Dio_write:

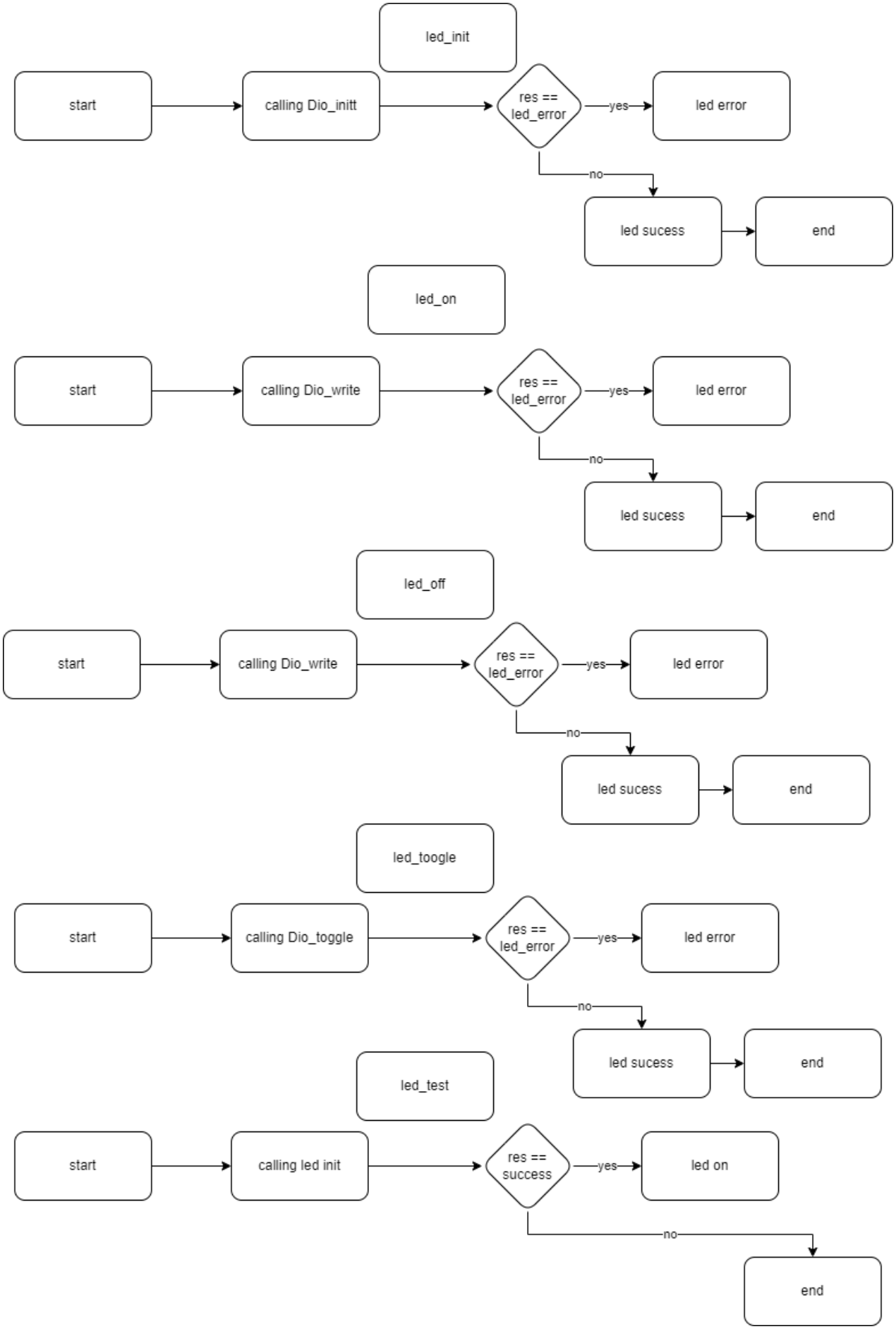

start
is port a
yes
is value low
yes
clear bit
no
set bit
no
is port b
yes
is value low
yes
clear bit
no
set bit
no
is port c
yes
is value low
yes
clear bit
no
set bit
no
is port d
yes
is value low
yes
clear bit
no
set bit
no
Error

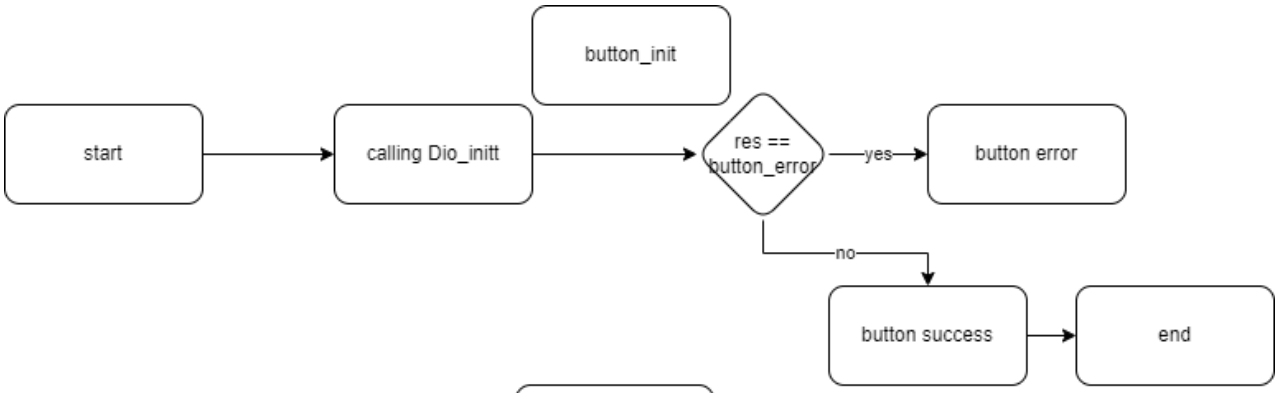
Dio_toggle:

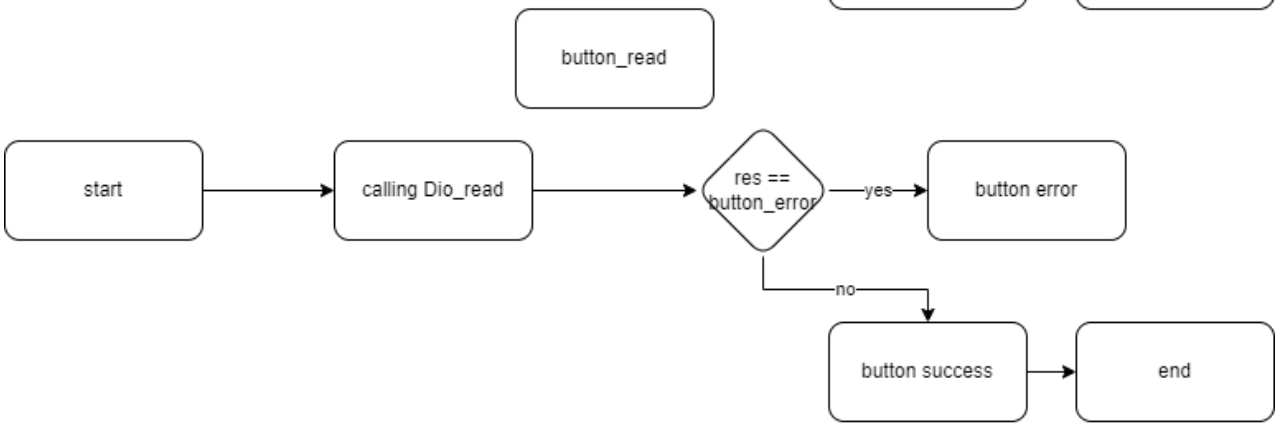

start
is port a
yes
toggle bit
no
is port b
yes
toggle bit
no
is port c
yes
toggle bit
no
is port d
yes
toggle bit
no
Error

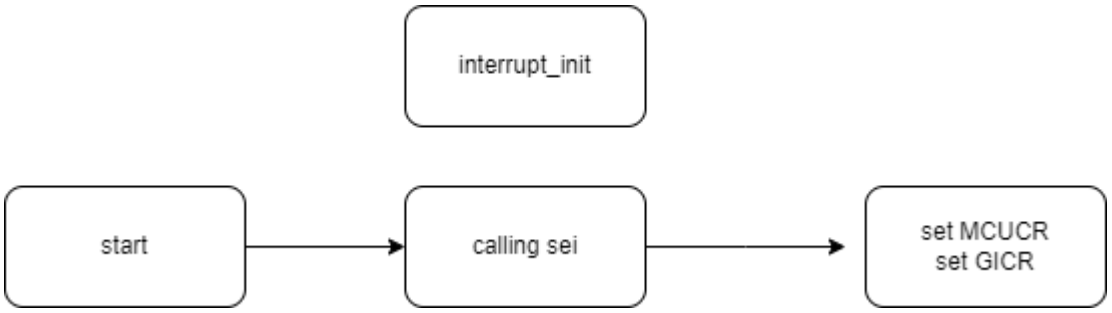
Dio_write_port:

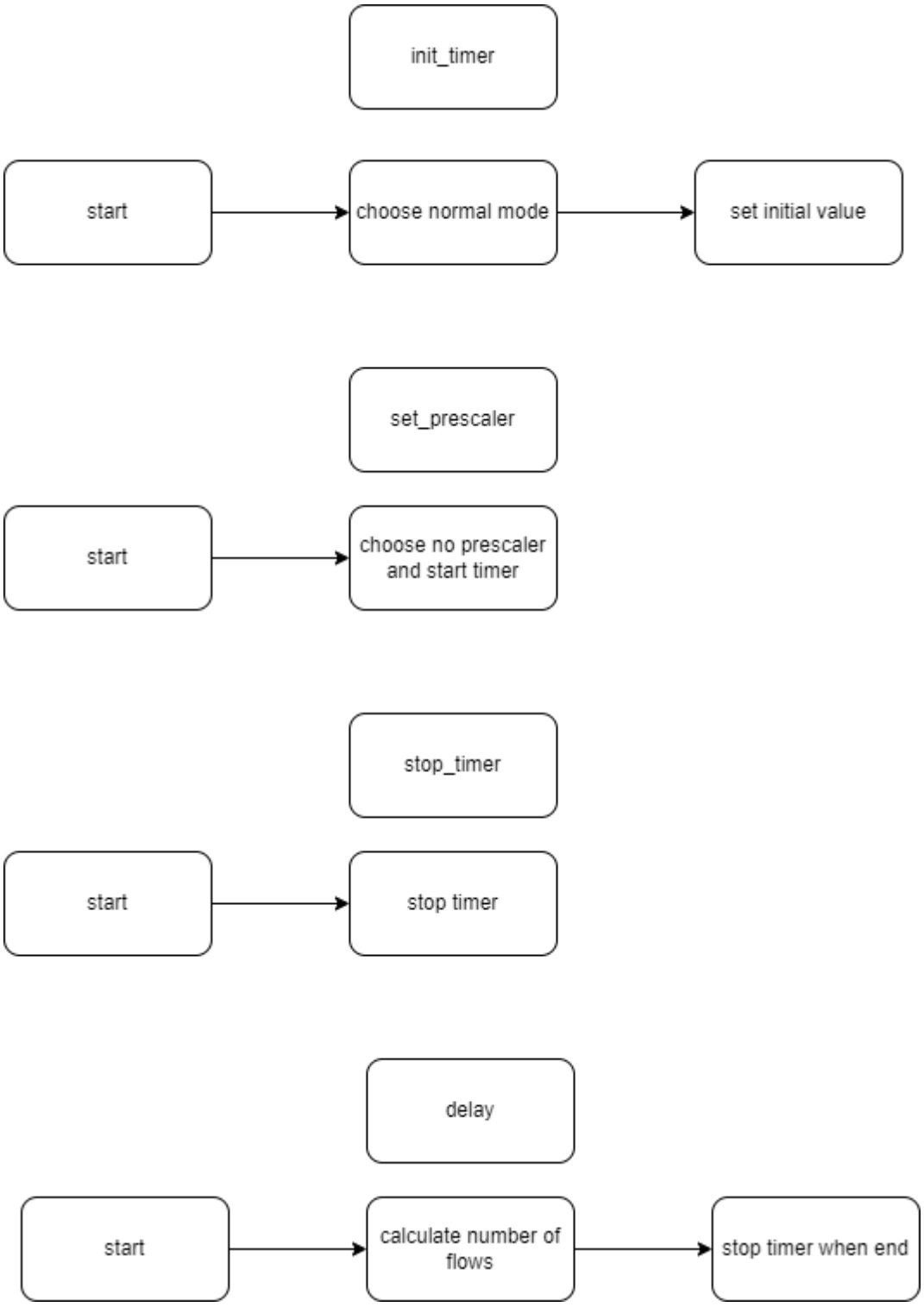


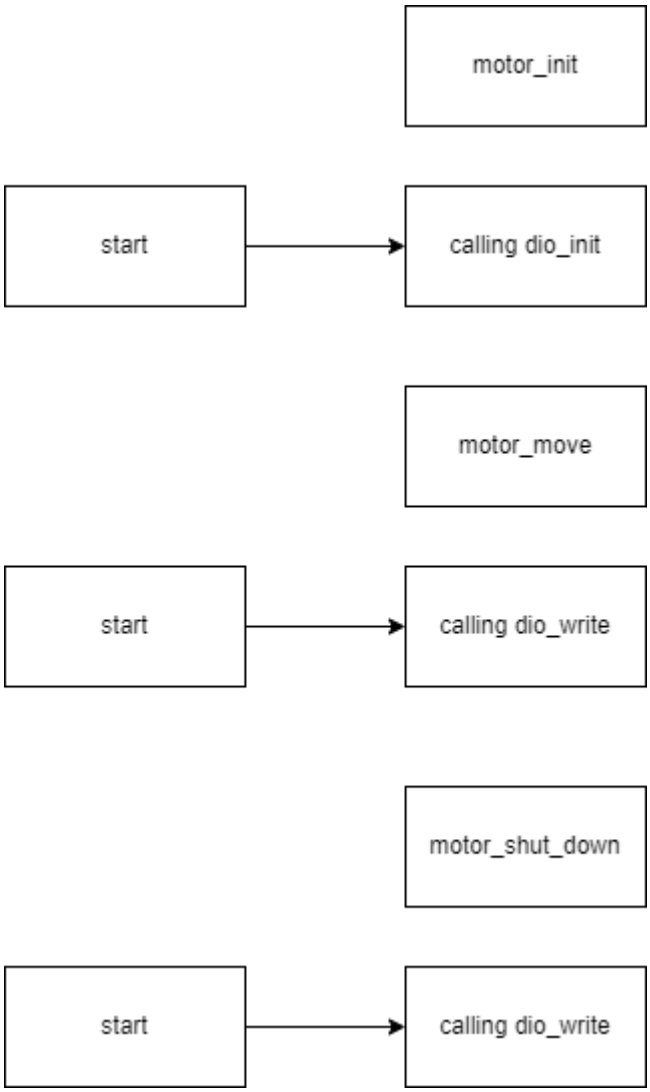

led_init
start
calling Dio_initt
res == led_error
yes
led error
no
led sucess
end
led_on
start
calling Dio_write
res == led_error
yes
led error
no
led sucess
end
led_off
start
calling Dio_write
res == led_error
yes
led error
no
led sucess
end
led_toogle
start
calling Dio_toggle
res == led_error
yes
led error
no
led sucess
end
led_test
start
calling led init
res == success
yes
led on
no
end


button_init
start
calling Dio_initt
res ==
button_error
yes
button error
no
button success
end


button_read
start
calling Dio_read
res ==
button_error
yes
button error
no
button success
end


interrupt_init
start
calling sei
set MCUCR
set GICR


init_timer
start
choose normal mode
set initial value
set_prescaler
start
choose no prescaler and start timer
stop_timer
start
stop timer
delay
start
calculate number of flows
stop timer when end


motor_init
start
calling dio_init
motor_move
start
calling dio_write
motor_shut_down
start
calling dio_write