RZ code description.

The very first time the board is powered on the EEPROM will be filled with FF’s.

The initialization steps in main will execute. This setups GPIO, CLKS and real time counter settings. The PIT is enabled to interrupt every 1024 cycles. Pin interrupts are also enabled. The read state from eeprom will return system state of FF which is invalid so the switch statement for the default case

Set evt button long to 1, mode to off, set reticle setting, set brightness

After this we will enter the while loop

The handle events will set the oldstate to mode\_off which gets written to eeprom state

In the first execution of handle events the evtbtn long was set to 1 by the switch case.

The mode is currently off and button long is set so the else statement runs setting the state to ON and enabling the real time counter and turning on the LED, clearing RTC\_Counts and clear any possible interrupts on the accelerator side.

Then set event btn long to 0;

Now to check the IF statement in the while loop. The IF is only executed if there is no button press, no events no blank or blink flags. So basically if nothing has happened.

The next pass through oldstate gets mode on and is written to eeprom

The wile loop will basically do nothing else until motion or PIT int.

Assuming PIT int-

The PIT is active and will track the motion timeout for MOTAC. For every 1024 clock cycles there will be an int. Going int to the ISR the int flag is cleared RTC count variable increases by 1. The motion time out is set for 90 seconds. This ISR will check at 45 and 90 for motion. If there is motion the RCT count will clear and the accelerometer int is cleared. If there is no motion and we have exceeded a RTC count of 90 seconds then the system events reticle timeout is set to 1.

At the next pass into handle events, the IF event time out statement will execute. If motac is not disabled then put system state to sleep and enable the motac interrupt, disable the PIT set RTC counts to 0, turn LED off.

Old state will now get MODE SLEEP and be written to eeprom

Still in the while loop the If statement will be evaluated, with no motion no flags no button presses and in sleep mode the MCU I will be set into sleep power down mode. Then the sleep mode enabled.

In sleep mode everything is off and no code is executing it is sitting doing nothing but waiting for and interrupt from the accelerometer.

Now we assume a user picks up the optic and this causes the motion to be detected on the port C ISR. Here the int flag is cleared. Event.motion is set to 1. The MCU comes out of sleep and executes the next step after it went to sleep, which is handle events.

In this execution of handle events we set old state to sleep then we fall into the IF statementa for evt.motion

RTC counts set to 0. If state is sleep then we turn state to mode on we enable the PIT, enable the LED. Set events.motion to 0.

Write the new on state to eeprom. Then execute the IF statement in the while loop which sets sleep mode to standby and enables the mode.

At the next PIT int

Clear pit int flag and increment RTC counts check to see if they motion n

Now assume we press a button 1 time before the 90 second timer runs out

During all this there is also a a 100ms periodic timer that is a interrupt that looks for any setting changes for led, reticle etc. or button press to handle debouncing.

The short button press will have debounced and then set evt btn short to 1. In handle events if btn short and mode on then we increase brightness or decrease brightness

Now assume we have a long button press, 3 seconds or so to increment the reticle. When the button is pressed for a long time the 100ms loop for debounce loops many times and each time it is incrementing the debounce counter. If the counter hits the reticle debounce value then sets retmode enabled to 1 and retmode to 1 and clears the debounce counter. The next loop through will then see the button release and button press will be reset.

A short button press will now changed reticle. Once ret timer is timed out goes back to normal mode. Lots of blink and blank here I don’t necessariy understand yet but going to move on for now.