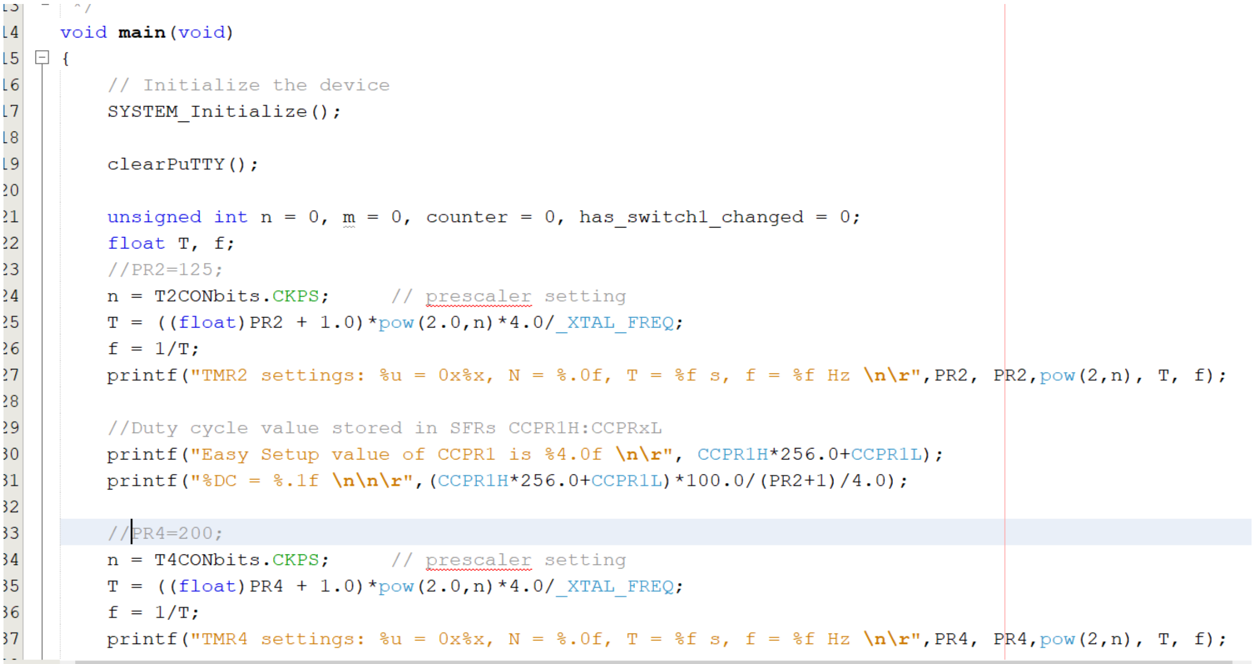
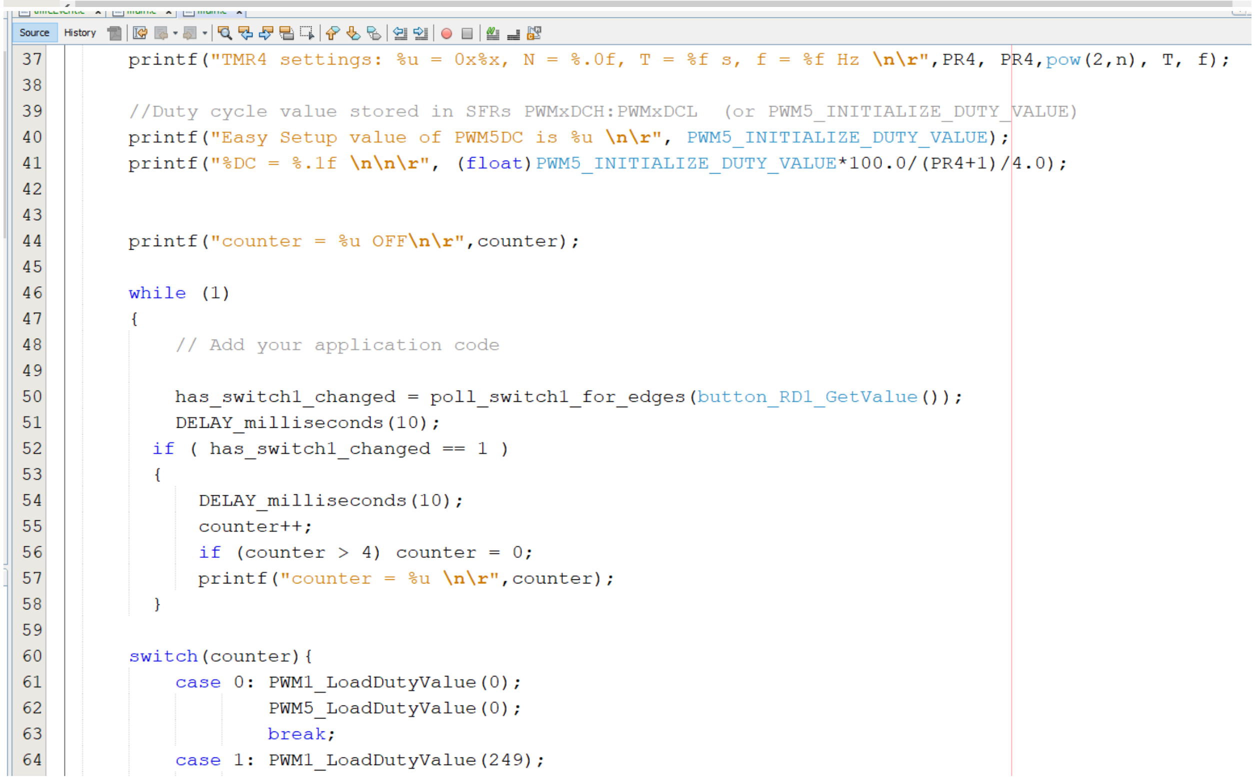
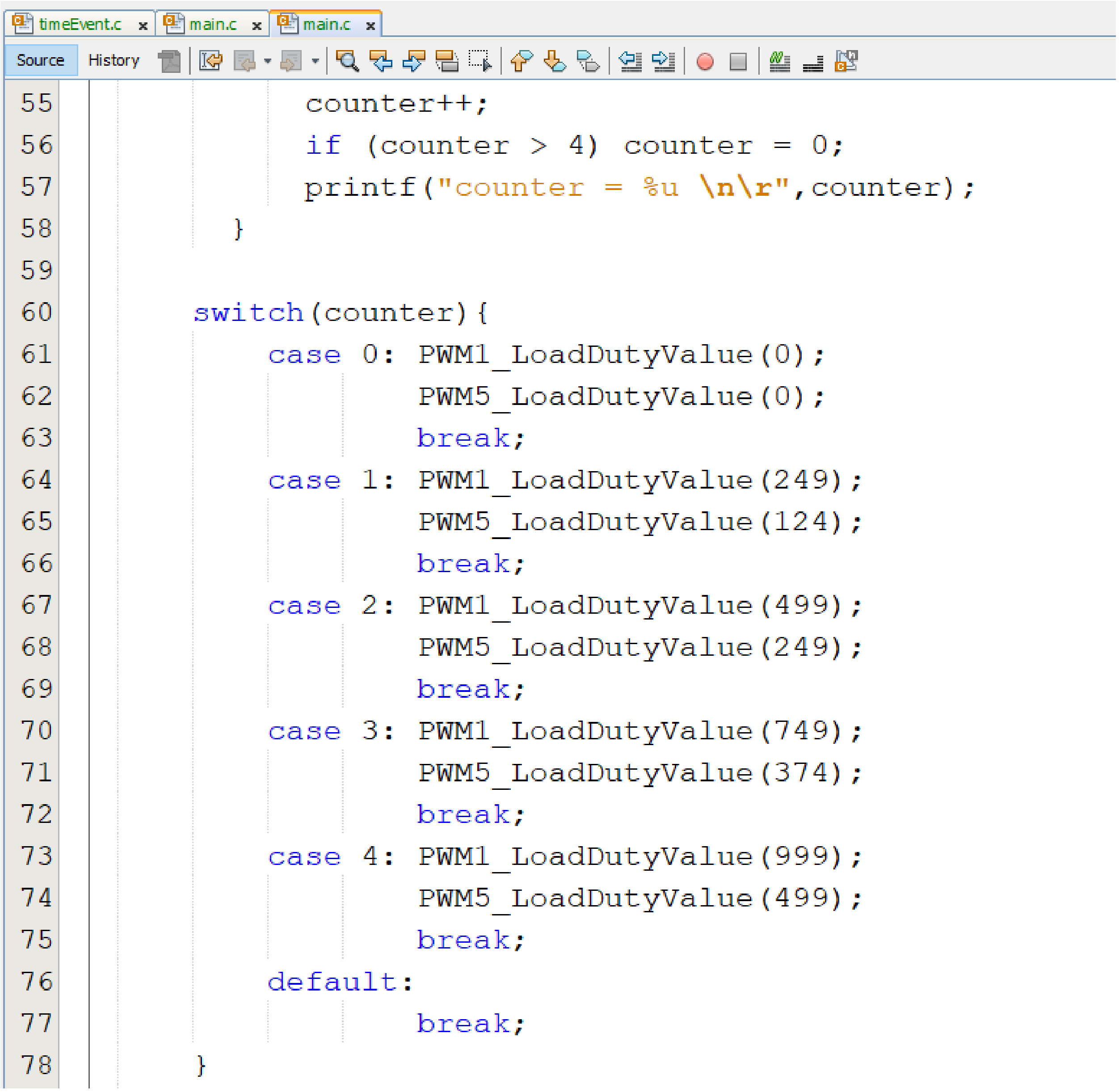


```
*/
void main(void)
{
    // Initialize the device
    SYSTEM_Initialize();

    clearPuTTY();

    unsigned int n = 0, m = 0, counter = 0, has_switch1_changed = 0;
    float T, f;
    //PR2=125;
    n = T2CONbits.CKPS;     // prescaler setting
    T = ((float)PR2 + 1.0)*pow(2.0,n)*4.0/_XTAL_FREQ;
    f = 1/T;
    printf("TMR2 settings: %u = 0x%x, N = %.0f, T = %f s, f = %f Hz \n\r",PR2, PR2,pow(2,n), T, f);

    //Duty cycle value stored in SFRs CCPR1H:CCPRxL
    printf("Easy Setup value of CCPR1 is %4.0f \n\r", CCPR1H*256.0+CCPR1L);
    printf("%DC = %.1f \n\n\r",(CCPR1H*256.0+CCPR1L)*100.0/(PR2+1)/4.0);

    //PR4=200;
    n = T4CONbits.CKPS;     // prescaler setting
    T = ((float)PR4 + 1.0)*pow(2.0,n)*4.0/_XTAL_FREQ;
    f = 1/T;
    printf("TMR4 settings: %u = 0x%x, N = %.0f, T = %f s, f = %f Hz \n\r",PR4, PR4,pow(2,n), T, f);
```

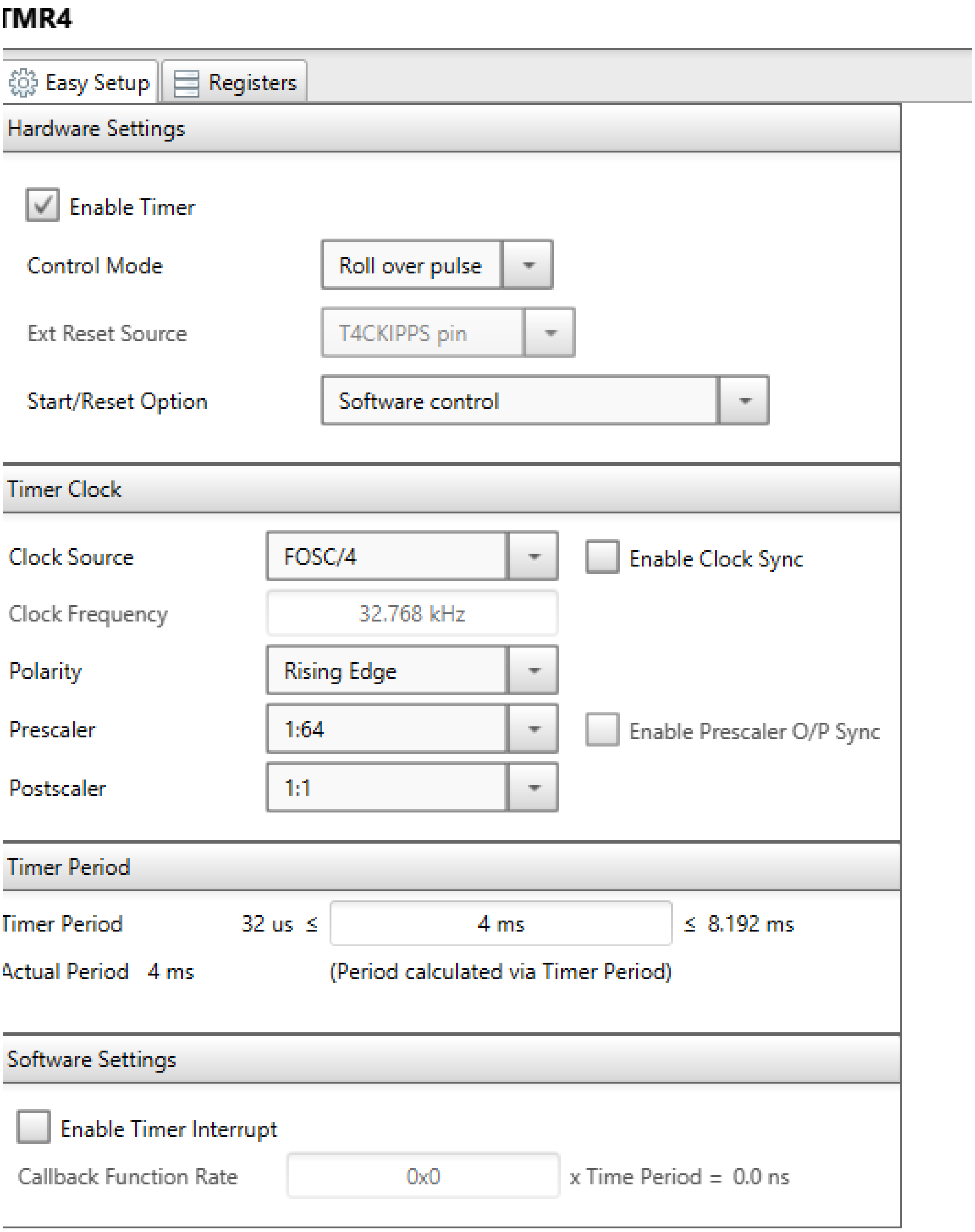


```
    printf("TMR4 settings: %u = 0x%x, N = %.0f, T = %f s, f = %f Hz \n\r",PR4, PR4,pow(2,n), T, f);
```

```

    //Duty cycle value stored in SFRs PWMxDCH:PWMxDCL  (or PWM5_INITIALIZE_DUTY_VALUE)
    printf("Easy Setup value of PWM5DC is %u \n\r", PWM5_INITIALIZE_DUTY_VALUE);
    printf("%DC = %.1f \n\n\r",  (float)PWM5_INITIALIZE_DUTY_VALUE*100.0/(PR4+1)/4.0);


    printf("counter = %u OFF\n\r",counter);

    while (1)
    {
        // Add your application code

        has_switch1_changed = poll_switch1_for_edges(button_RD1_GetValue());
        DELAY_milliseconds(10);
      if ( has_switch1_changed == 1 )
      {
          DELAY_milliseconds(10);
          counter++;
          if (counter > 4) counter = 0;
          printf("counter = %u \n\r",counter);
      }

    switch(counter){
        case 0: PWM1_LoadDutyValue(0);
                PWM5_LoadDutyValue(0);
                break;
        case 1: PWM1_LoadDutyValue(249);
```



```
              counter++;
              if (counter > 4) counter = 0;
              printf("counter = %u \n\r",counter);
          }

        switch(counter){
            case 0: PWM1_LoadDutyValue(0);
                    PWM5_LoadDutyValue(0);
                    break;
            case 1: PWM1_LoadDutyValue(249);
                    PWM5_LoadDutyValue(124);
                    break;
            case 2: PWM1_LoadDutyValue(499);
                    PWM5_LoadDutyValue(249);
                    break;
            case 3: PWM1_LoadDutyValue(749);
                    PWM5_LoadDutyValue(374);
                    break;
            case 4: PWM1_LoadDutyValue(999);
                    PWM5_LoadDutyValue(499);
                    break;
            default:
                    break;
        }
```



TMR4

Hardware Settings

☑ Enable Timer

Control Mode: Roll over pulse

Ext Reset Source: T4CKIPPS pin

Start/Reset Option: Software control

Timer Clock

Clock Source: FOSC/4 □ Enable Clock Sync

Clock Frequency: 32.768 kHz

Polarity: Rising Edge

Prescaler: 1:64 □ Enable Prescaler O/P Sync

Postscaler: 1:1

Timer Period

Timer Period 32 us ≤ 4 ms ≤ 8.192 ms

Actual Period 4 ms (Period calculated via Timer Period)

Software Settings

□ Enable Timer Interrupt

Callback Function Rate 0x0 x Time Period = 0.0 ns