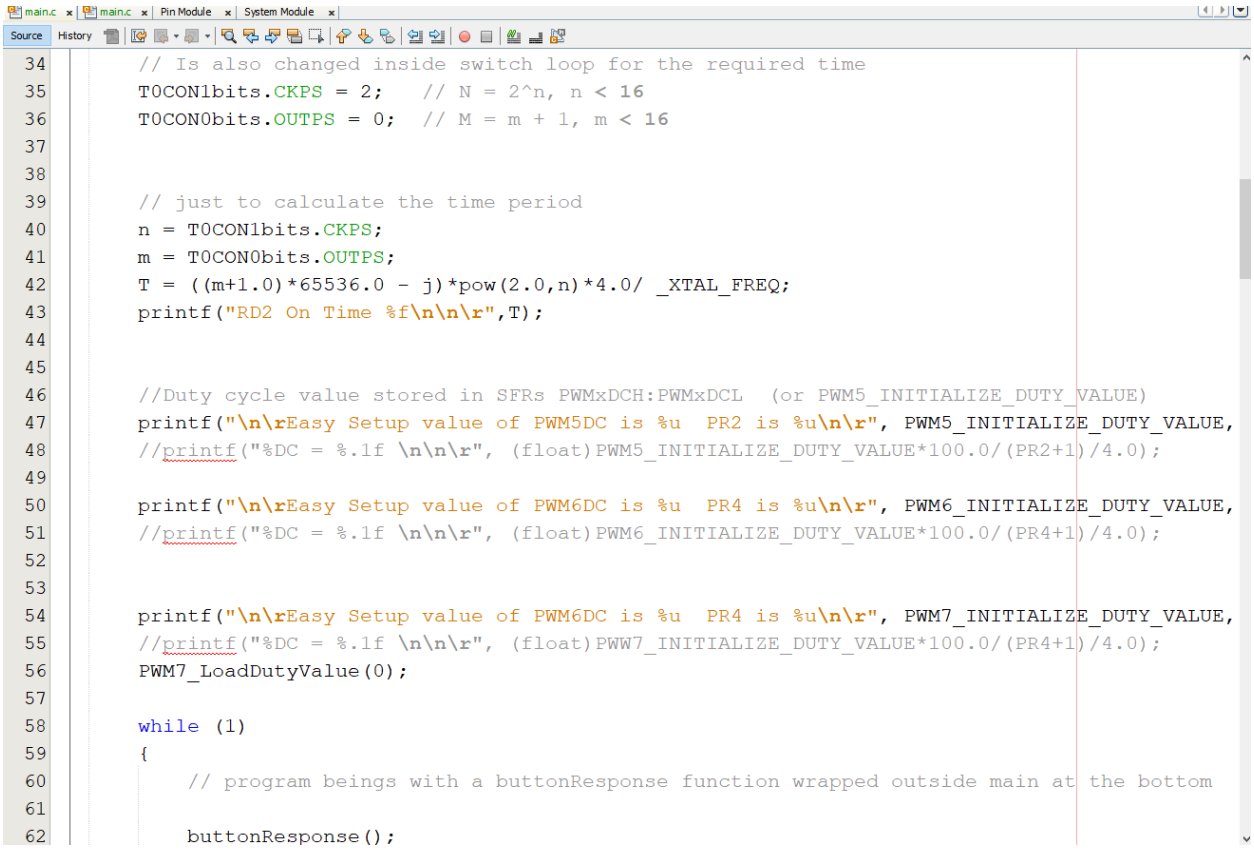
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "putty.h"
#include "myUART.h"
#include <ctype.h>
#include "buttons.h"

#include "mcc_generated_files/mcc.h"

/*
                         Main application
 */
void buttonResponse(void);
unsigned int counter = 0, has_switch1_changed = 0;


void main(void)
{
    // Initialize the device
    SYSTEM_Initialize();
    clearPuTTY();

    unsigned int n = 0, m = 0;
    float T, f;

    // Initial value of Timer for blinking timers
    unsigned int j=45536;
```

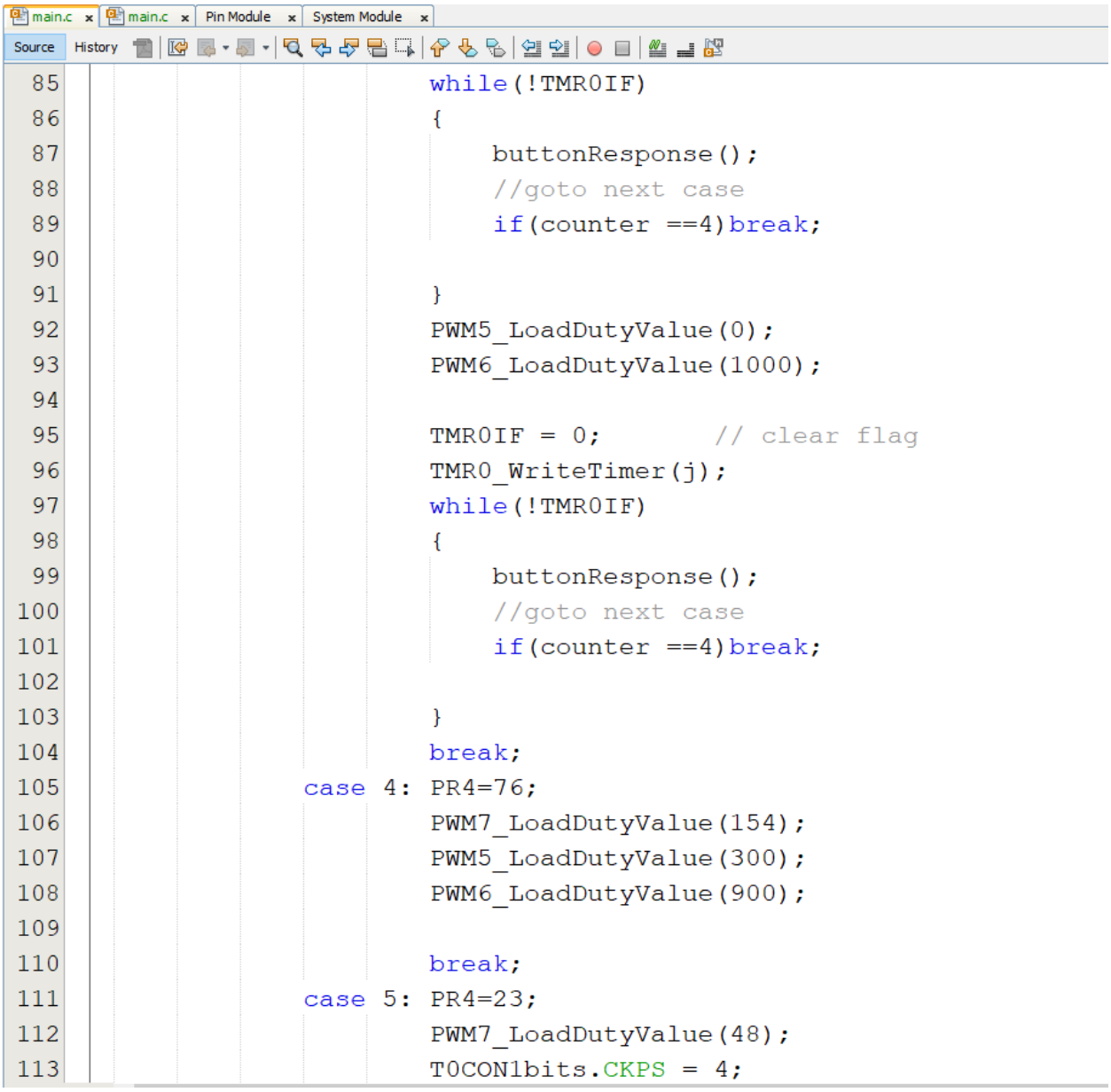


```
34      // Is also changed inside switch loop for the required time
35      T0CON1bits.CKPS = 2;    // N = 2^n, n < 16
36      T0CON0bits.OUTPS = 0;   // M = m + 1, m < 16
37
38
39      // just to calculate the time period
40      n = T0CON1bits.CKPS;
41      m = T0CON0bits.OUTPS;
42      T = ((m+1.0)*65536.0 - j)*pow(2.0,n)*4.0/ _XTAL_FREQ;
43      printf("RD2 On Time %f\n\n\r",T);
44
45
46      //Duty cycle value stored in SFRs PWMxDCH:PWMxDCL  (or PWM5_INITIALIZE_DUTY_VALUE)
47      printf("\n\rEasy Setup value of PWM5DC is %u  PR2 is %u\n\r", PWM5_INITIALIZE_DUTY_VALUE,
48      //printf("%DC = %.1f \n\n\r", (float)PWM5_INITIALIZE_DUTY_VALUE*100.0/(PR2+1)/4.0);
49
50      printf("\n\rEasy Setup value of PWM6DC is %u  PR4 is %u\n\r", PWM6_INITIALIZE_DUTY_VALUE,
51      //printf("%DC = %.1f \n\n\r", (float)PWM6_INITIALIZE_DUTY_VALUE*100.0/(PR4+1)/4.0);
52
53
54      printf("\n\rEasy Setup value of PWM6DC is %u  PR4 is %u\n\r", PWM7_INITIALIZE_DUTY_VALUE,
55      //printf("%DC = %.1f \n\n\r", (float)PWW7_INITIALIZE_DUTY_VALUE*100.0/(PR4+1)/4.0);
56      PWM7_LoadDutyValue(0);
57
58      while (1)
59      {
60          // program beings with a buttonResponse function wrapped outside main at the bottom
61
62          buttonResponse();
```

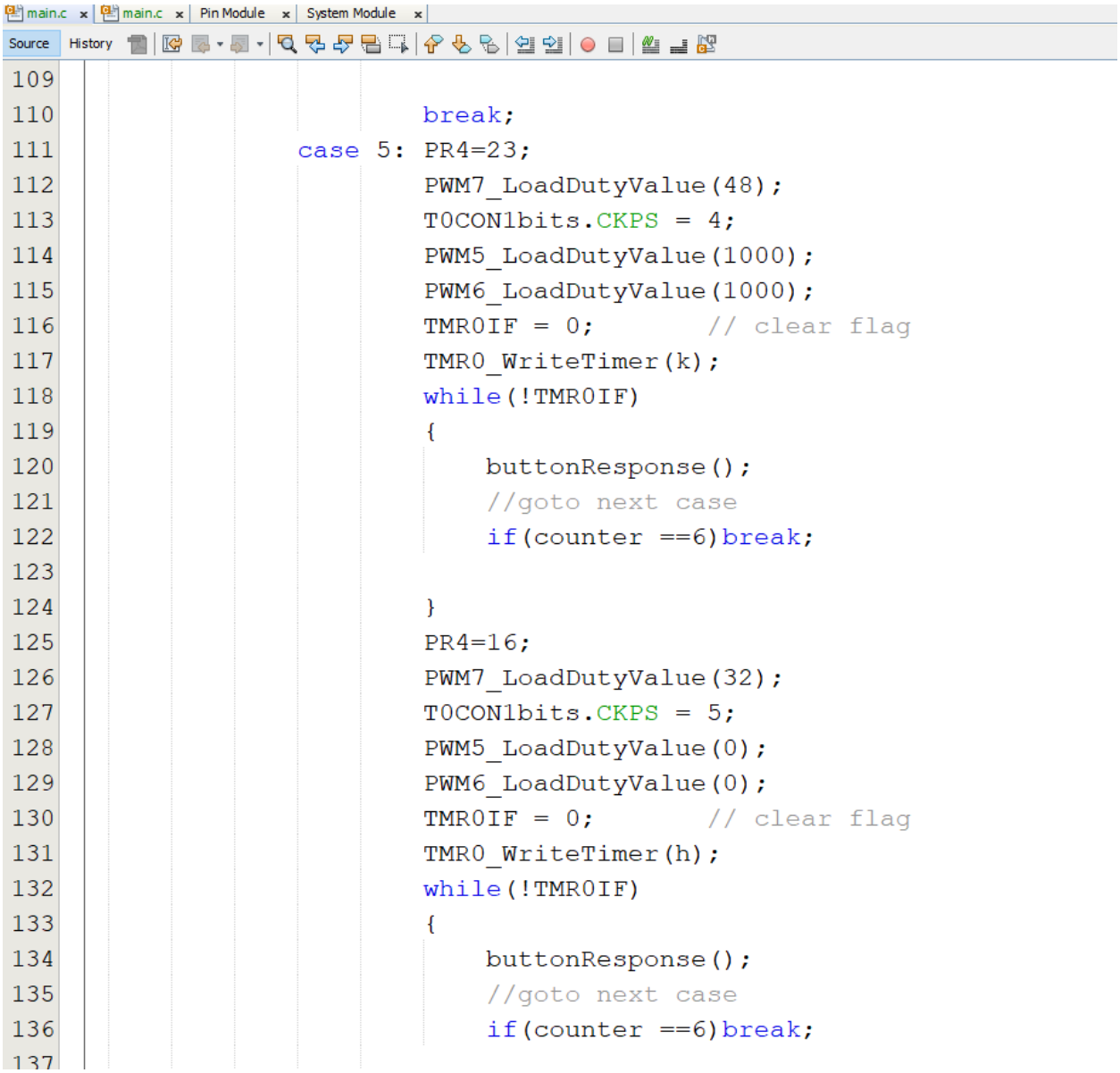
main.c | main.c | Pin Module | System Module

Source | History

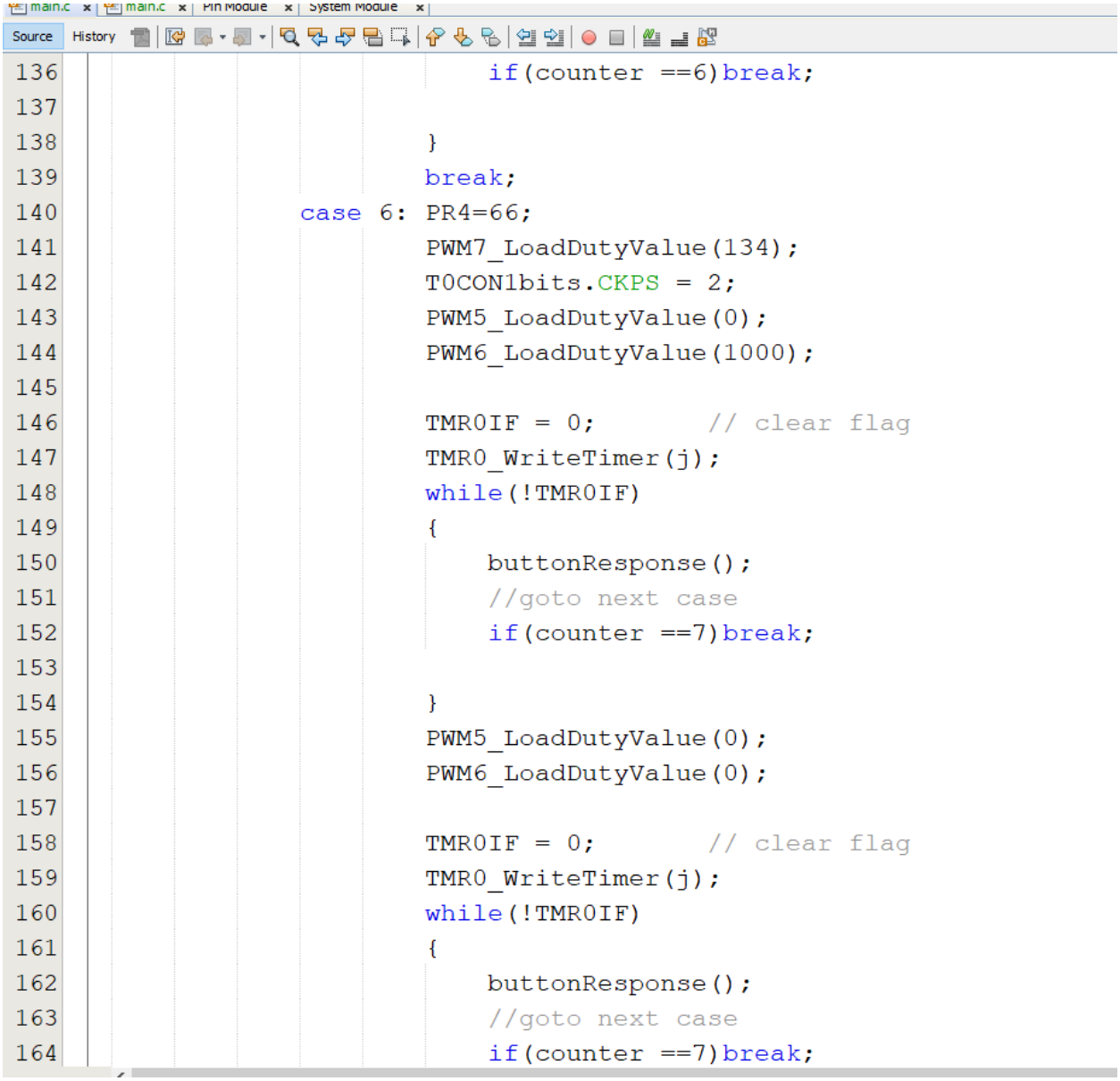
```
61
62          buttonResponse();
63
64          switch(counter)
65          {
66              case 1: PR4=118;
67                      PWM7_LoadDutyValue(238);
68                      PWM5_LoadDutyValue(1000);
69                      PWM6_LoadDutyValue(1000);
70
71                      break;
72              case 2: PR4=102;
73                      PWM7_LoadDutyValue(206);
74                      PWM5_LoadDutyValue(350);
75                      PWM6_LoadDutyValue(1023);
76
77                      break;
78              case 3: PR4=88;
79                      PWM7_LoadDutyValue(178);
80                      PWM5_LoadDutyValue(1000);
81                      PWM6_LoadDutyValue(1000);
82
83                      TMR0IF = 0;       // clear flag
84                      TMR0_WriteTimer(j);
85                      while(!TMR0IF)
86                      {
87                          buttonResponse();
88                          //goto next case
89                          if(counter ==4)break;
```




```
85                  while(!TMR0IF)
86                  {
87                      buttonResponse();
88                      //goto next case
89                      if(counter ==4)break;
90
91                  }
92                  PWM5_LoadDutyValue(0);
93                  PWM6_LoadDutyValue(1000);
94
95                  TMR0IF = 0;         // clear flag
96                  TMR0_WriteTimer(j);
97                  while(!TMR0IF)
98                  {
99                      buttonResponse();
100                     //goto next case
101                     if(counter ==4)break;
102
103                 }
104                 break;
105             case 4: PR4=76;
106                 PWM7_LoadDutyValue(154);
107                 PWM5_LoadDutyValue(300);
108                 PWM6_LoadDutyValue(900);
109
110                 break;
111             case 5: PR4=23;
112                 PWM7_LoadDutyValue(48);
113                 T0CON1bits.CKPS = 4;
```




```
109
110                         break;
111                 case 5: PR4=23;
112                         PWM7_LoadDutyValue(48);
113                         T0CON1bits.CKPS = 4;
114                         PWM5_LoadDutyValue(1000);
115                         PWM6_LoadDutyValue(1000);
116                         TMR0IF = 0;       // clear flag
117                         TMR0_WriteTimer(k);
118                         while(!TMR0IF)
119                         {
120                             buttonResponse();
121                             //goto next case
122                             if(counter ==6)break;
123
124                         }
125                         PR4=16;
126                         PWM7_LoadDutyValue(32);
127                         T0CON1bits.CKPS = 5;
128                         PWM5_LoadDutyValue(0);
129                         PWM6_LoadDutyValue(0);
130                         TMR0IF = 0;       // clear flag
131                         TMR0_WriteTimer(h);
132                         while(!TMR0IF)
133                         {
134                             buttonResponse();
135                             //goto next case
136                             if(counter ==6)break;
137
```


main.c
main.c
Pin Module
System Module
Source
History
136 if(counter ==6)break;
137
138 }
139 break;
140 case 6: PR4=66;
141 PWM7_LoadDutyValue(134);
142 T0CON1bits.CKPS = 2;
143 PWM5_LoadDutyValue(0);
144 PWM6_LoadDutyValue(1000);
145
146 TMR0IF = 0; // clear flag
147 TMR0_WriteTimer(j);
148 while(!TMR0IF)
149 {
150 buttonResponse();
151 //goto next case
152 if(counter ==7)break;
153
154 }
155 PWM5_LoadDutyValue(0);
156 PWM6_LoadDutyValue(0);
157
158 TMR0IF = 0; // clear flag
159 TMR0_WriteTimer(j);
160 while(!TMR0IF)
161 {
162 buttonResponse();
163 //goto next case
164 if(counter ==7)break;

main.c | main.c | Pin Module | System Module

Source | History

```
160                         while(!TMR0IF)
161                         {
162                             buttonResponse();
163                             //goto next case
164                             if(counter ==7)break;
165 
166                         }
167 
168                         break;
169                     case 7: PR4=52;
170                         PWM7_LoadDutyValue(106);
171                         PWM5_LoadDutyValue(0);
172                         PWM6_LoadDutyValue(750);
173 
174                         break;
175                     case 8: PR4=40;
176                         PWM7_LoadDutyValue(82);
177                         PWM5_LoadDutyValue(0);
178                         PWM6_LoadDutyValue(100);
179                         break;
180                     case 9: PWM7_LoadDutyValue(0);
181                         PWM5_LoadDutyValue(0);
182                         PWM6_LoadDutyValue(1000);
183                         break;
184 
185             }
186         }
187 }
```

```
}

void buttonResponse(void)
{
    has_switch1_changed = poll_switch1_for_edges(button_RD1_GetValue());
    DELAY_milliseconds(10);
    if ( has_switch1_changed == 1 )
    {
        DELAY_milliseconds(10);
        counter++;
        printf("State = %u \n\r",counter);
        if(counter > 9)
        {
            counter = 1;
            printf("State = %u \n\r",counter);
        }

    }
}
```

COM4 - PuTTY

```
RD2 On Time 0.040000


Easy Setup value of PWM5DC is 499  PR2 is
 249

Easy Setup value of PWM6DC is 499  PR4 is
 249

Easy Setup value of PWM6DC is 499  PR4 is
 249
State = 1
State = 2
State = 3
State = 4
State = 5
State = 6
State = 7
State = 8
State = 9
```