Could change the reset of the counter to zero but that would require an extra click, thought it would be better this way. I did try it out and it works that way as well.

main.c × | main.c × | inputChar.c × | main.c × | random.c × | main.c × | buttonCounter.c × | Available Resources × | Pin Module × | System Module

Source | History

```c
20          int i;
21          unsigned int newcount=0,  oldcount=0,k=0;
22          unsigned char button;
23          unsigned int has_switch1_changed=0,  counter=0;
24          clearPuTTY();
25
26
27          printf("Ready\n\r>");
28          printf("press button on breadboard\n\r");
29
30          while (1)
31          {
32              // Add your application code
33              button = RA4_GetValue();
34              has_switch1_changed = poll_switch1_for_edges(button);
35              if(has_switch1_changed==1){
36                  counter++;
37                  if(counter<=3)
38                  {
39                  printf(" Count = %u \n\r", counter);
40                  }
41              }
42              if(counter>=4)
43              {
44                  counter=1;
45                  printf(" Count = %u \n\r", counter);
46
47              }
48
49              switch(counter)
50              {
51              case 1: redB_RD2_SetHigh();
52                      greenB_RD1_SetLow();
53                      break;
54              case 2: greenB_RD1_SetHigh();
55                      redB_RD2_SetLow();
56                      break;
57              case 3: redB_RD2_SetLow();
58                      greenB_RD1_SetLow();
59                      break;
```