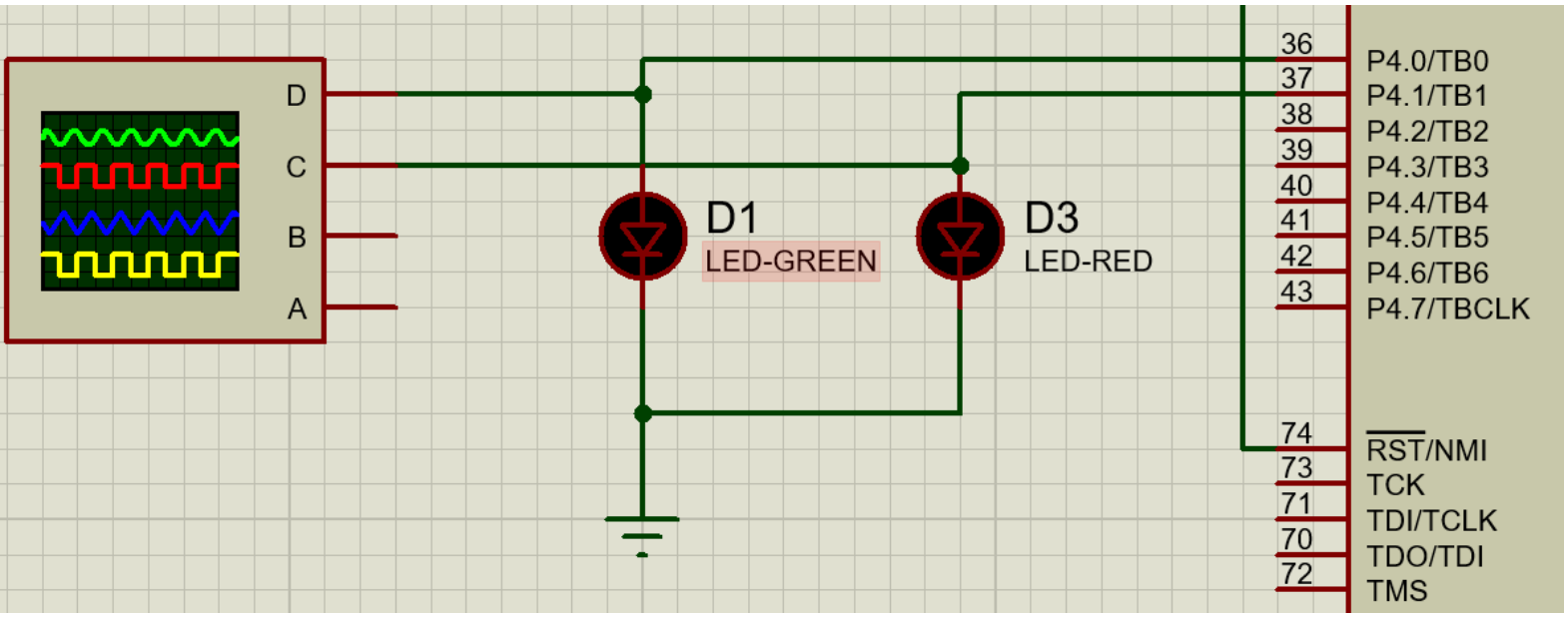
How to use two channels of a single timer for two different tasks

```
#include <msp430.h>
#include <stdint.h>
#define msec_25          25000
#define msec_50          50000
#define red_led          BIT1
#define green_led        BIT0
uint16_t taiv_temp= 0;

int main(void)
{
  WDTCTL = WDTPW + WDTHOLD;
  BCSCTL1 = CALBC1_1MHZ;
  DCOCTL = CALDCO_1MHZ;

  TA0CCR1 = msec_50;             //compare value for first interupt
  TA0CCTL1 = CCIE;                  //enable interupt

  P4DIR = red_led | green_led;
  P4OUT = 0;
  TA0CCR2 = msec_25;
  TA0CCTL2 = CCIE;

  TA0CTL = TASSEL_2|MC_2|ID_0|TACLR|TAIE; //Continuous mode, SMCLK, /1, TAR is cleared

  while(1)
  {
      __bis_SR_register(GIE);
  }
    return 0;
}

#pragma vector =  TIMER0_A1_VECTOR
__interrupt void TIMERA1_ISR (void) // ISR for TACCRn CCIFG and TAIFG
{
   taiv_temp = TA0IV;     // necessary because accessing TAIV resets it
  switch (taiv_temp)     // Efficient switch-implementation
  {

    case TAIV_TACCR1:
      P4OUT ^= green_led;             //Toggle Green Led
      TA0CCR1 += msec_50;        // Run freely in Continous Mode
      break;

    case TAIV_TACCR2:
      P4OUT ^= red_led;                //Toggle Red LED
      TA0CCR2 += msec_25;
      break;
  return;
  }
}
```


D
C
B
A
D1
LED-GREEN
D3
LED-RED
36
37
38
39
40
41
42
43
74
73
71
70
72
P4.0/TB0
P4.1/TB1
P4.2/TB2
P4.3/TB3
P4.4/TB4
P4.5/TB5
P4.6/TB6
P4.7/TBCLK
RST/NMI
TCK
TDI/TCLK
TDO/TDI
TMS