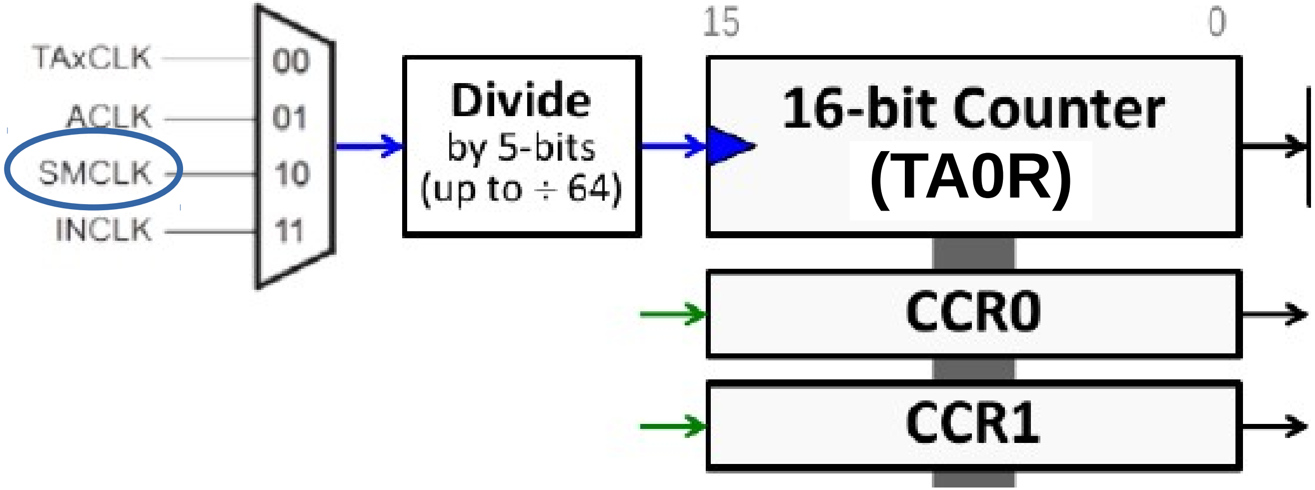
M7 Assignment

Generate PWM signal to fade the GREEN LED

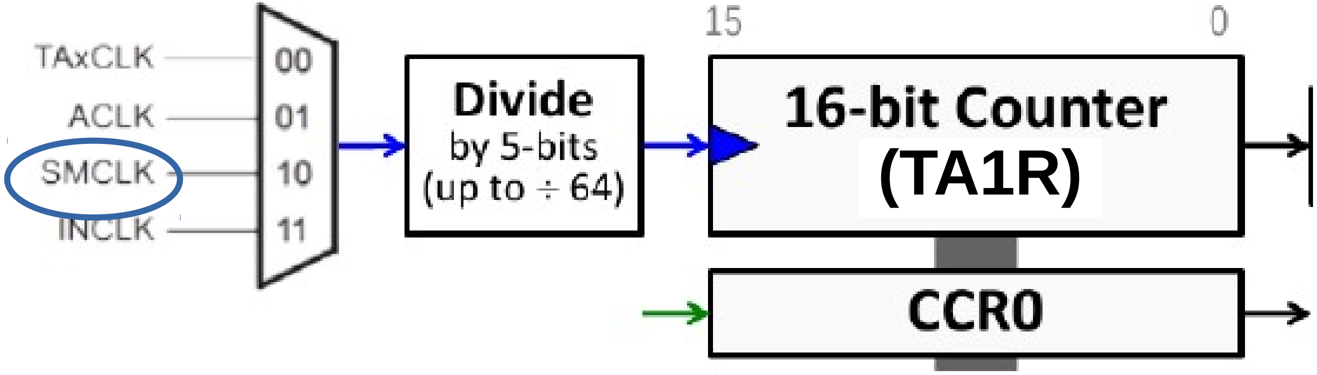
And also use the Serial Plotter to plot the intensity values

PWM



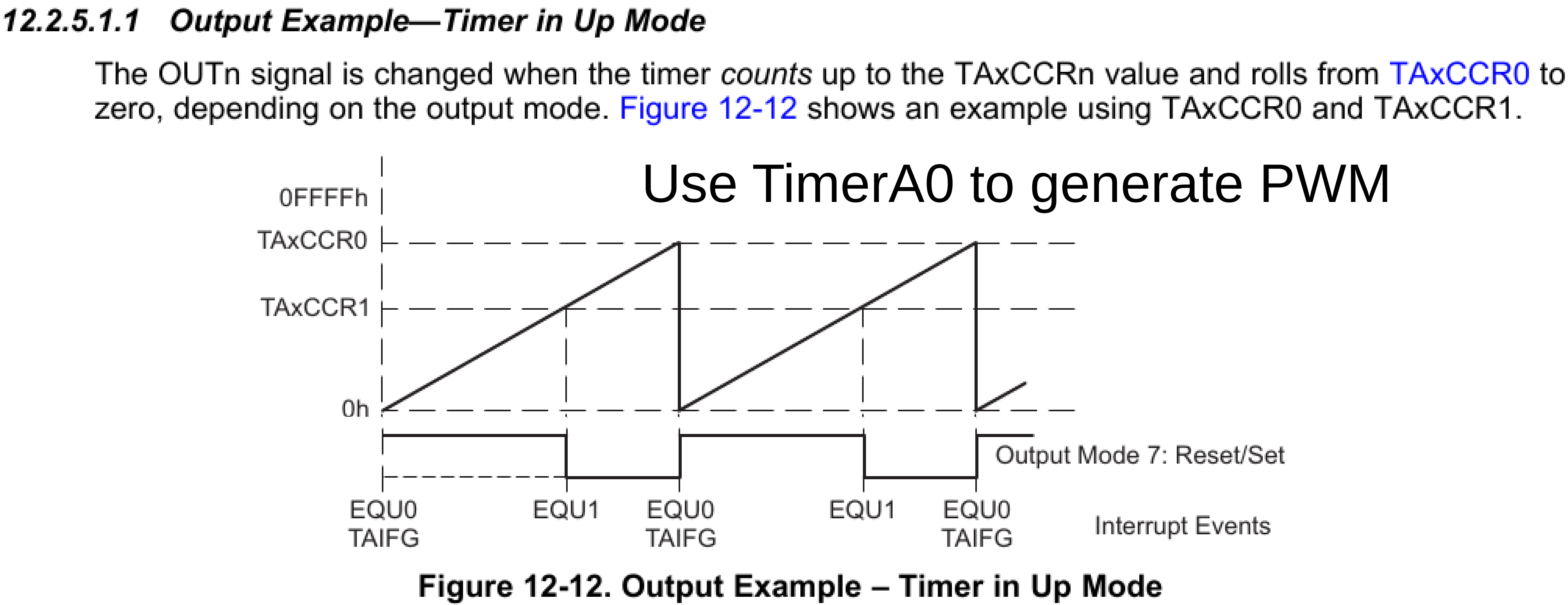
100% Pulse Time

Duty % Pulse Time



Reset & print value for interrupt

Use TimerA0 to generate PWM



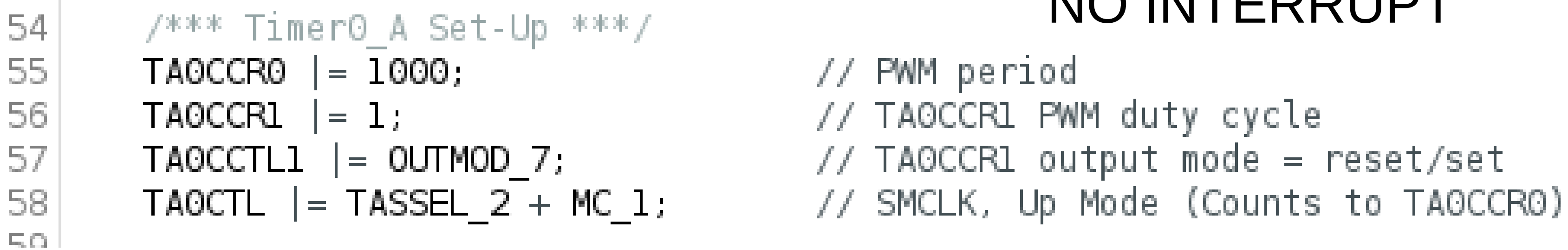
12.2.5.1.1 *Output Example—Timer in Up Mode*

The OUTn signal is changed when the timer *counts* up to the TAxCCRn value and rolls from TAxCCR0 to zero, depending on the output mode. Figure 12-12 shows an example using TAxCCR0 and TAxCCR1.

0FFFFh
TAxCCR0
TAxCCR1
0h
Output Mode 7: Reset/Set
EQU0
TAIFG
EQU1
EQU0
TAIFG
EQU1
EQU0
TAIFG
Interrupt Events

Figure 12-12. Output Example – Timer in Up Mode

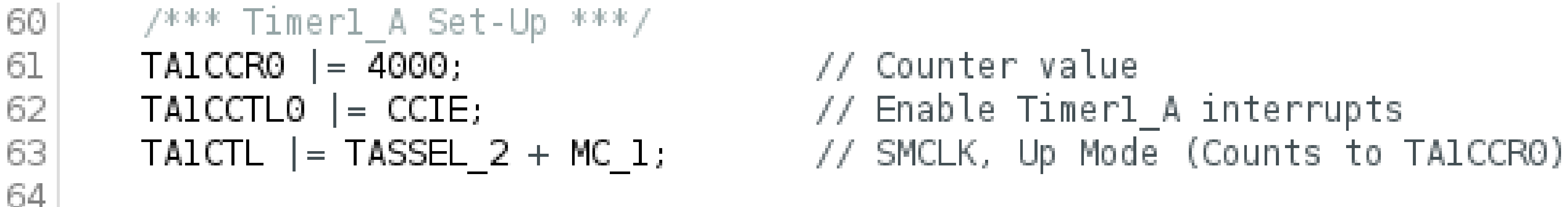
NO INTERRUPT



```
    /*** Timer0_A Set-Up ***/
    TA0CCR0 |= 1000;                    // PWM period
    TA0CCR1 |= 1;                       // TA0CCR1 PWM duty cycle
    TA0CCTL1 |= OUTMOD_7;               // TA0CCR1 output mode = reset/set
    TA0CTL |= TASSEL_2 + MC_1;          // SMCLK, Up Mode (Counts to TA0CCR0)
```

SET TA0CCR1 to a fixed duty cycle – CCR1/CCR0

Use TimerA1 to interrupt every 4000 clock cycles to insert a new fixed duty cycle in TimerA0



```
    /*** Timer1_A Set-Up ***/
    TA1CCR0 |= 4000;                    // Counter value
    TA1CCTL0 |= CCIE;                   // Enable Timer1_A interrupts
    TA1CTL |= TASSEL_2 + MC_1;          // SMCLK, Up Mode (Counts to TA1CCR0)

```

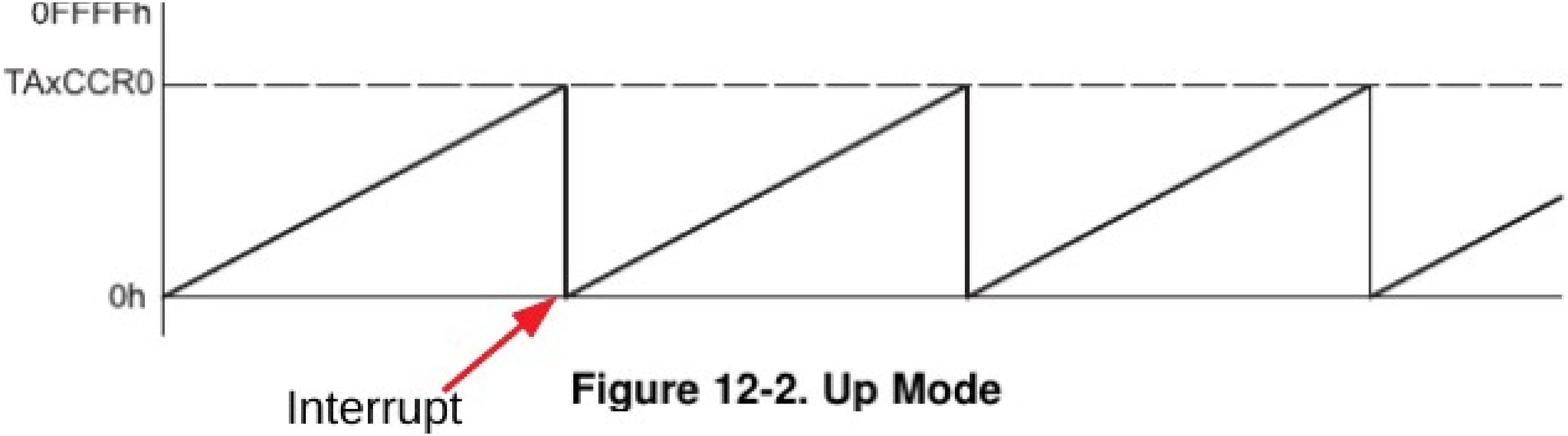
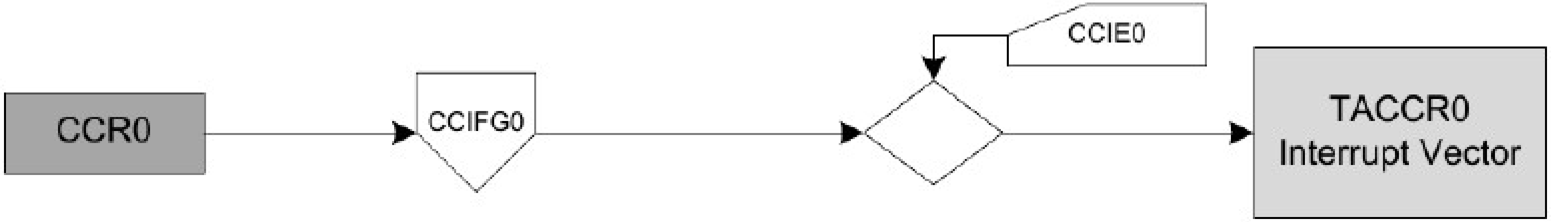
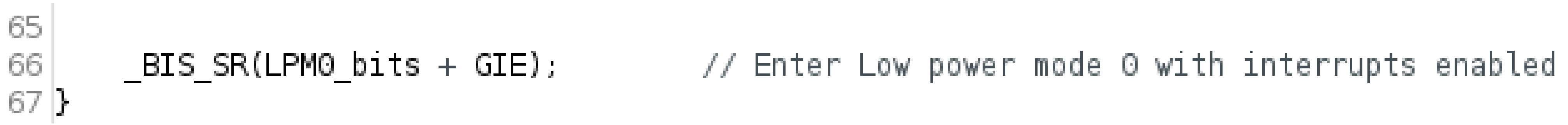


Figure 12-2. Up Mode





```

    _BIS_SR(LPM0_bits + GIE);          // Enter Low power mode 0 with interrupts enabled
}
```

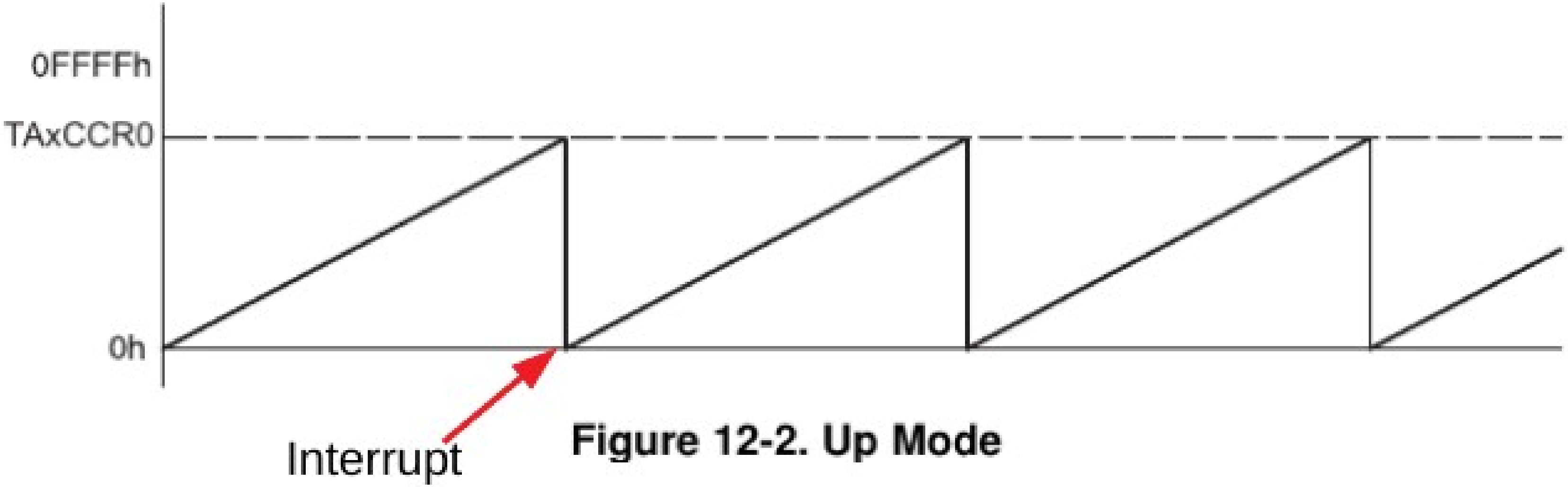
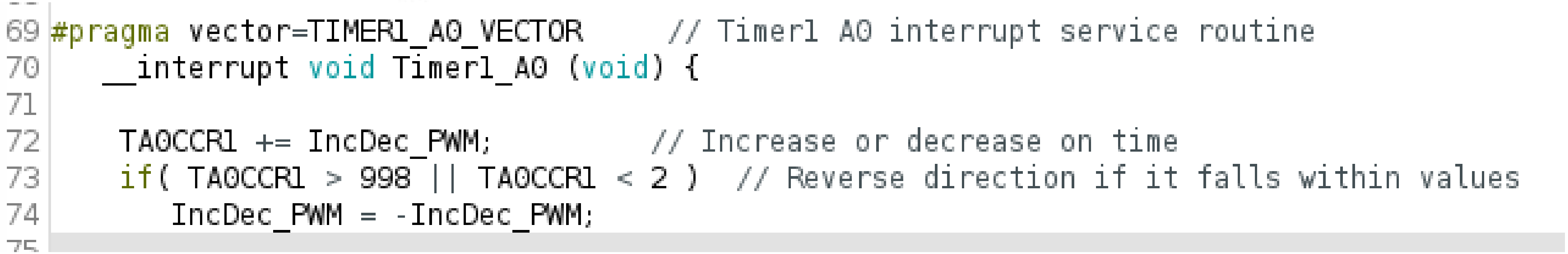
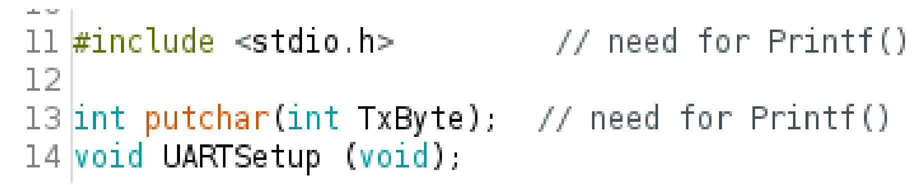


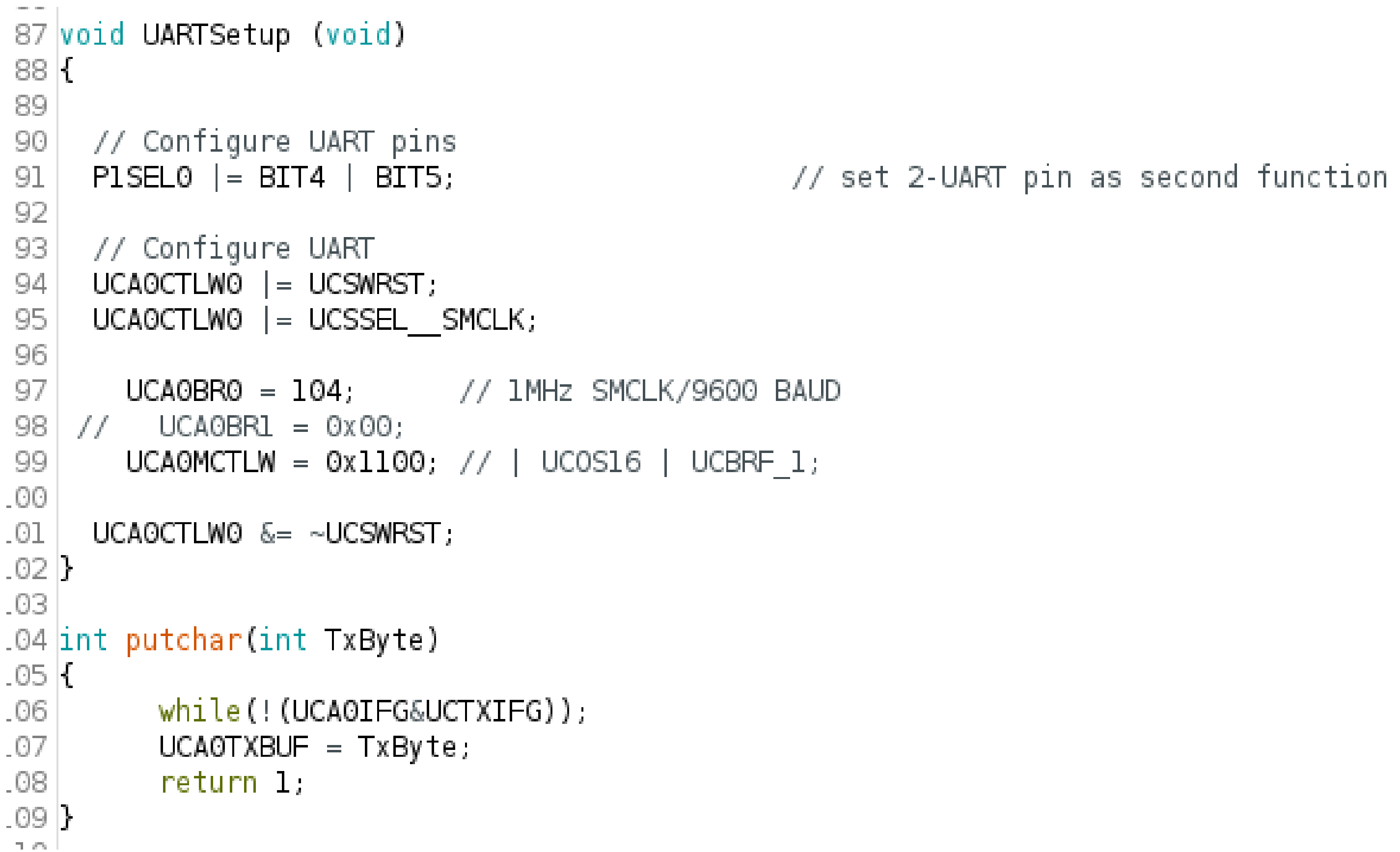
Figure 12-2. Up Mode



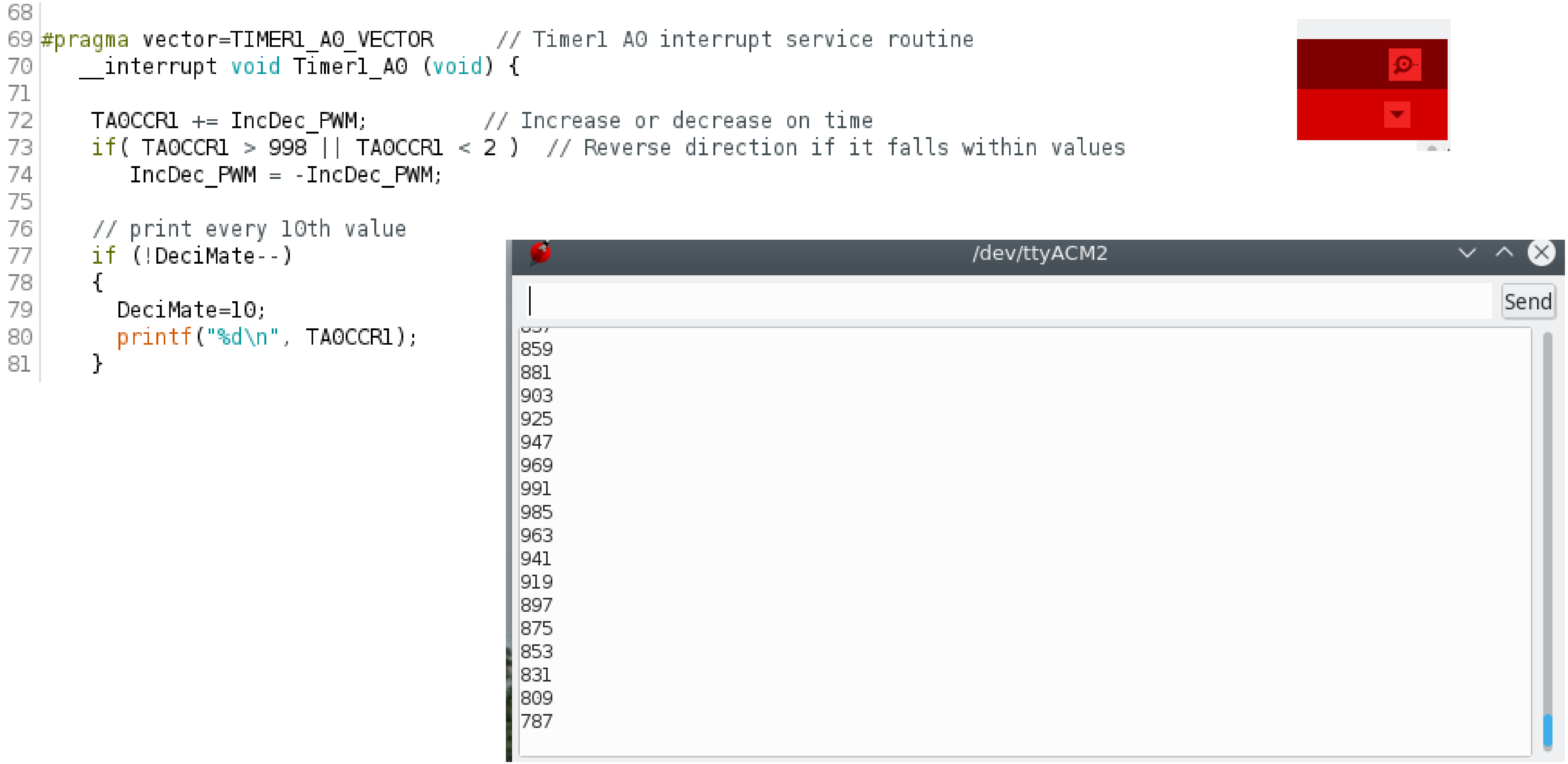
```
#pragma vector=TIMER1_A0_VECTOR      // Timer1 A0 interrupt service routine
   __interrupt void Timer1_A0 (void) {

    TA0CCR1 += IncDec_PWM;          // Increase or decrease on time
    if( TA0CCR1 > 998 || TA0CCR1 < 2 )  // Reverse direction if it falls within values
       IncDec_PWM = -IncDec_PWM;
```



```c
#include <stdio.h>         // need for Printf()

int putchar(int TxByte);  // need for Printf()
void UARTSetup (void);
```

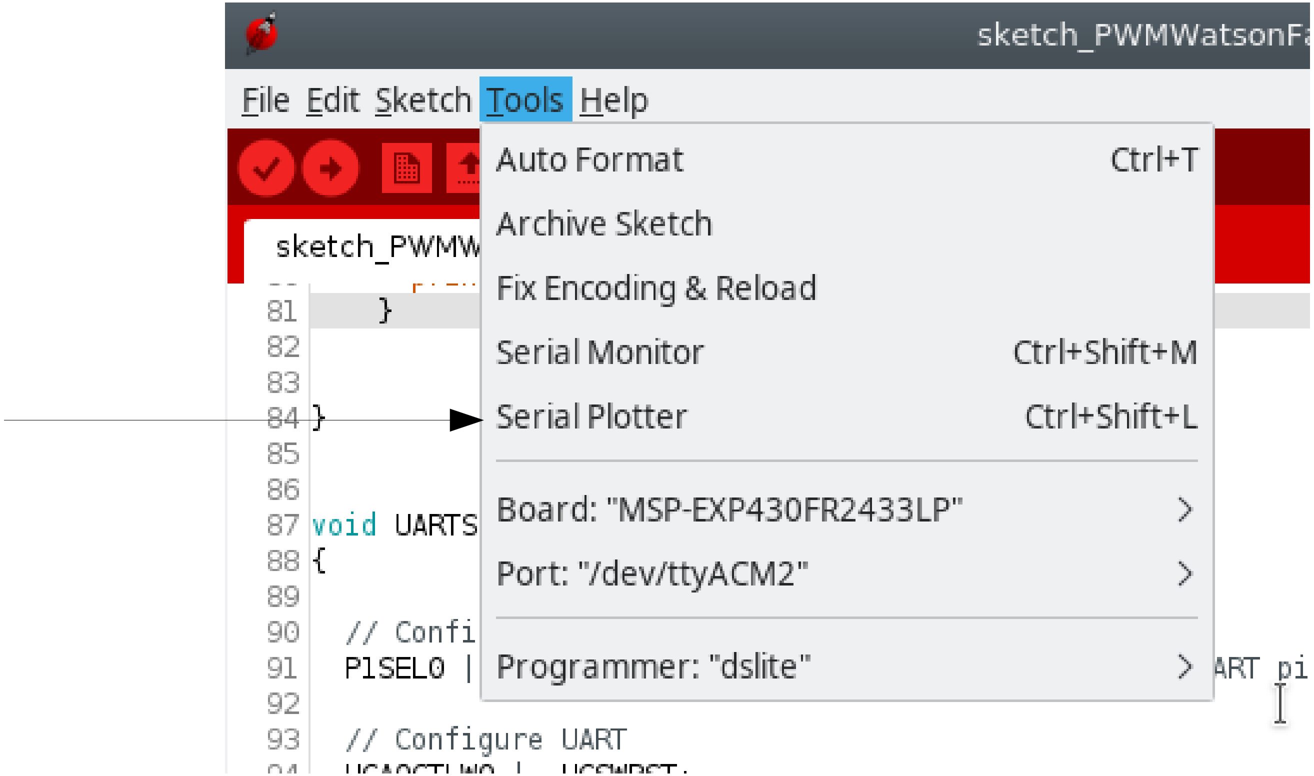


```c
void UARTSetup (void)
{

  // Configure UART pins
  P1SEL0 |= BIT4 | BIT5;                    // set 2-UART pin as second function

  // Configure UART
  UCA0CTLW0 |= UCSWRST;
  UCA0CTLW0 |= UCSSEL__SMCLK;

    UCA0BR0 = 104;      // 1MHz SMCLK/9600 BAUD
 //   UCA0BR1 = 0x00;
    UCA0MCTLW = 0x1100; // | UCOS16 | UCBRF_1;

  UCA0CTLW0 &= ~UCSWRST;
}

int putchar(int TxByte)
{
      while(!(UCA0IFG&UCTXIFG));
      UCA0TXBUF = TxByte;
      return 1;
}
```



```
#pragma vector=TIMER1_A0_VECTOR     // Timer1 A0 interrupt service routine
   __interrupt void Timer1_A0 (void) {

    TA0CCR1 += IncDec_PWM;         // Increase or decrease on time
    if( TA0CCR1 > 998 || TA0CCR1 < 2 )  // Reverse direction if it falls within values
       IncDec_PWM = -IncDec_PWM;

    // print every 10th value
    if (!DeciMate--)
    {
      DeciMate=10;
      printf("%d\n", TA0CCR1);
    }
```

```
859
881
903
925
947
969
991
985
963
941
919
897
875
853
831
809
787
```



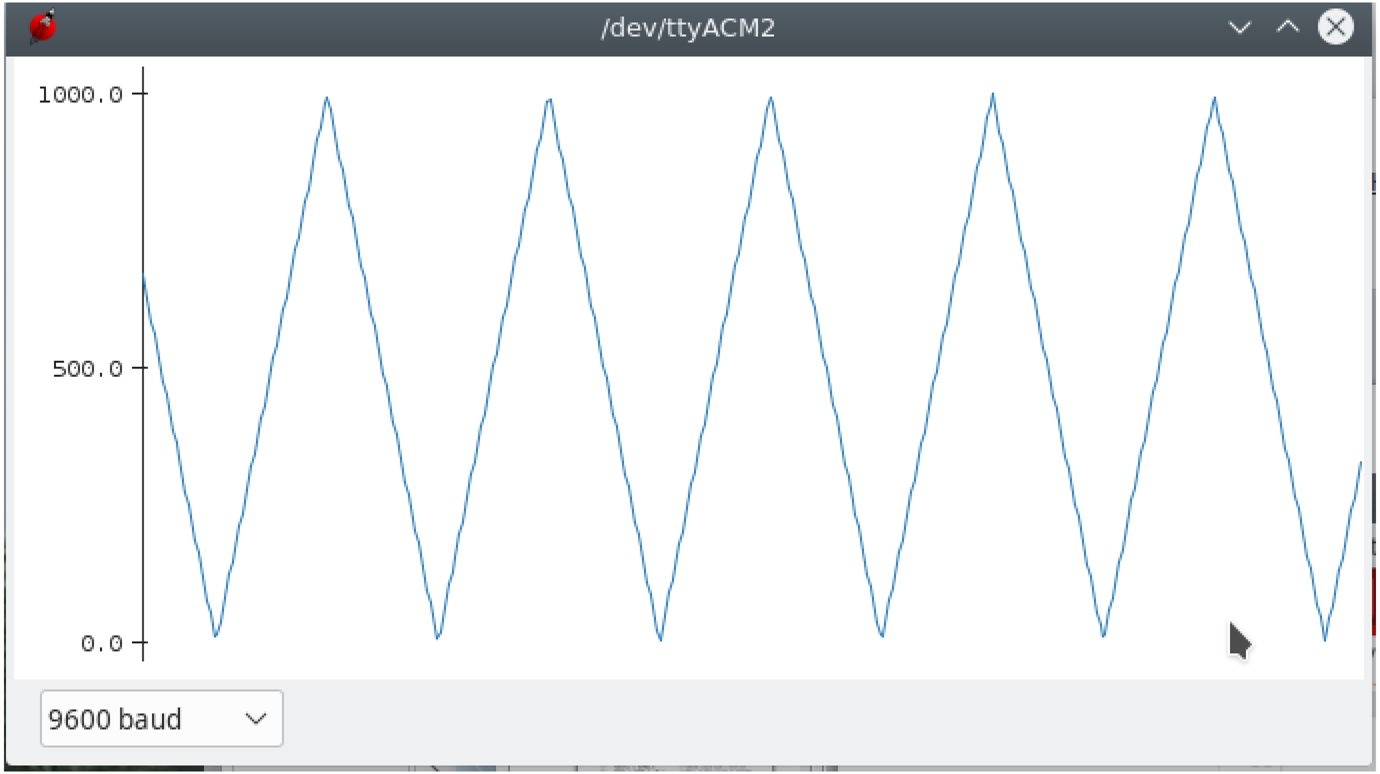
sketch_PWMWatsonF
File Edit Sketch Tools Help
sketch_PWMW
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L

Board: "MSP-EXP430FR2433LP"

Port: "/dev/ttyACM2"

Programmer: "dslite"

```
    }


}


void UARTS
{

  // Confi
  P1SEL0 |                                                    ART pi

  // Configure UART
```



9600 baud