MSP430 Timer in Capture Mode

Praveen Kumar Yadav

September 22, 2025

1 Aim

To program the MSP430 Timer_A module in **capture mode** for measuring the period of an external signal.

2 Theory

Timer_A in capture mode allows the microcontroller to record the timer value whenever an edge is detected on a capture pin. By storing multiple capture values, the time period of an input waveform can be computed.

3 Circuit

4 Source Code

```
#include <msp430.h>
#define NUMBER_TIMER_CAPTURES 20

volatile unsigned int timerAcaptureValues[NUMBER_TIMER_CAPTURES];
unsigned int timerAcapturePointer = 0;

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;      // Stop watchdog timer

    // Configure GPIO
    P1DIR |= BIT0;
    P1OUT |= BIT0;
    P1SEL1 |= BIT2;                // TA0.CCI2A input
    P1REN  |= BIT2;
    P1OUT &= ~BIT2;

    PM5CTL0 &= ~LOCKLPM5;          // Unlock GPIO

    // Timer0_A3 Setup
    TA0CCTL2 |= CM_1 | CCIS_0 | CCIE | CAP | SCS;
```

main.c Thursday, October 20, 2022, 2:40 PM

```
/*    programming the TIMER in CAPTURE MODE
 *
 *    step-1: An external signal from the function generator is to be captured by the mc.
 *
 *            choose CCIS bit---find which pin is multiplexed for capture input signal from datasheet
 *            see the controls need to be written in the SEL and OUT bits from datasheet
 *
 *    step-2: setup timer clk,clr bit enable interrupt, MC from TA0CTL register
 *    step-3: setup capture mode by writing appropriate bits in the TA0CCTL register( CAP,CCIS,SCS,capture on rising/falling edge, CIE)
 *
 *    step-4: In the interrupt vector, under case TA0IV_TACCx, write the interrupt subroutine.
 *    in this program ,Capture a number of periods of the external signal and store them in an array.
//  When the set number of periods is captured the program is trapped and the LED on
//  P1.0 is toggled. At this point halt the program execution read out the values using
//  the debugger.
//  ACLK = REFOCLK = 32kHz, MCLK = SMCLK = default DCODIV = 1MHz.
 *
 */
#include <msp430.h>
#define NUMBER_TIMER_CAPTURES        20

volatile unsigned int timerAcaptureValues[NUMBER_TIMER_CAPTURES];
unsigned int timerAcapturePointer = 0;


/**
 * main.c
 */
int main(void)
{
```


Figure 1: Circuit for Timer Capture Mode

```
    TA0CTL    |= TASSEL_2 | MC_2 | TACLR;

    __bis_SR_register(LPM0_bits | GIE);
    __no_operation();
}
```

```
27
28  // Timer0_A3 ISR
29  #pragma vector = TIMER0_A1_VECTOR
30  __interrupt void TIMER0_A1_ISR(void)
31  {
32      switch(__even_in_range(TA0IV, TA0IV_TAIFG))
33      {
34          case TA0IV_TACCR2:
35              timerAcaptureValues[timerAcapturePointer++] = TA0CCR2
                    ;
36              if (timerAcapturePointer >= 20)
37              {
38                  while (1)
39                  {
40                      P1OUT ^= BIT0;
41                      __delay_cycles(100000);
42                  }
43              }
44              break;
45          default:
46              break;
47      }
48  }
```

Listing 1: Timer Capture Mode Code

5 Observation

When the program runs, the MSP430 captures the timer counts on each rising edge of the input signal. After 20 captures, the LED at P1.0 toggles continuously, indicating completion.

6 Conclusion

The experiment successfully demonstrated the use of the Timer_A peripheral in capture mode for measuring signal periods.