Lab 1.0 - MSP432 I/O Report

Zehao Chen, Xinyu Zhu

I. Problem Statement

The aim of this lab is to program a MSP432 board that a joystick can control servomotor's position. The angle of servomotor should be changed with the joystick's position.

The analog value of the joystick is sampled and converted every 20 ms. And The servomotor should be controlled by a PWM with a width between 1 ms and 2 ms.

II. High-level system description

For the high level design of the system, we used three main peripherals Timer_A0, ADC14(Analog to Digital Converter) and GPIO ports.Flow chart is shown in Figure 1.

• Timer_A0: The Timer_A0 serves two purposes. One is to periodically initiate the sampling and conversion of the analog value The other is to generate the 1-2ms PWM signal. The Timer_A0 can use ACLK signal to control the time of sampling and PWM precisely.

• ADC14: The ADC14 is used to sample the analog value of the joystick in every 20ms. And the converted value will store in MEM[0], which is used for controlling the width of PWM signal.

• GPIO ports: In this system, P4.0 is connected with joystick and used to input analog value. P2.4 is connected with servomotor and used to output the PWM signal.

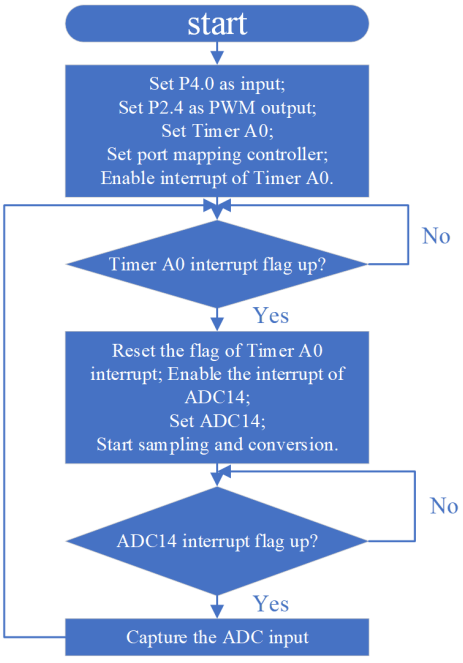

Fig. 1. The flow chart of the code.

III. Lower-level system description

For the lower level design of the two peripherals, the followings are some details:

• Timer_A0: In order to sample the ADC14 analog value in every 20ms. We set up CTL register to Up mode and select ACLK as source signals. And then we set CCR[0] as 32.7 * 20, which means we can control the sampling time as 20ms. By setting CCTL[0] register as TIMER_A_CCTLN_CCIE, the Timer_A0 will enter into the interrupt function in every 20ms. As for generating the PWM, we choose P2.4 as out put port. And we also want to use the value of CCR[1] to control the width of PWM. To realize it, we set CCTL[1] as Toggle/set Method and use the value in MEM[0] to adjust the PWM.The ADC14 setting code is shown in 2.

```
//set TimerA0,  use: up mode, ACLK, output mode 6 and enable interrupt.
TIMER_A0->CTL =  TIMER_A_CTL_MC_1 | TIMER_A_CTL_SSEL__ACLK;
TIMER_A0->CCR[0] = (int)32.768*20;
TIMER_A0->CCR[1] = 0;
TIMER_A0->CCTL[1] = TIMER_A_CCTLN_OUTMOD_6;
TIMER_A0->CCTL[0] = TIMER_A_CCTLN_CCIE;
```

Fig. 2. The short snippet of Timer_A0 setting code.

•ADC14: When Timer_A0 enters into the interrupt function, ADC14 will be active. We map P4.0 to A13 as the input of analog value joystick. To control ADC sampling and conversion, we set ADC14SC, ADC14ENC and ADC14ON to 1 when it starts sampling. And By enabling ADC14_IER0_IE0, the analog value will store in MEM[0]. Whenever there is a value in MEM[0], it will enters into ADC interrupt function. In the function, the value in CCR[1] will be changed according to the analog value in MEM[0]. The ADC14 setting code is shown in 3.

```
//enable ADC14 interrupt
ADC14->IER0 = ADC14_IER0_IE0;
//set ADC14, use A13, start sample and conversion
ADC14->MCTL[0]=ADC14_MCTLN_INCH_13;
ADC14->CTL0 = ADC14_CTL0_ON | ADC14_CTL0_ENC | ADC14_CTL0_SC;
//offset
ADC14->CTL0 &= ~ADC14_CTL0_SC;
```

Fig. 3. The short snippet of ADC14 setting code.

• GPIO ports: P4 is set as input and to use A13 function. P2 is set as output and to use TA0.1 function.

IV. Conclusion

The code and servomotor work as expected. But there are still some precision issues.

Firstly, the period of PWM signal is not accurate. We choose ACLK and set Timer_A0 CCR[0] = (int)32.768*20, so the

period is supposed to be 20 ms. However, as shown in Figure 4 - 6, the period is about 19.5 ms. Secondly, the period of PWM signal is not fixed. As it is seen in the Figure 4 - 6, the periods of PWM signals are 19.514 ms, 19.513 ms and 19.511 ms, respectively. There is a little difference among them. Thirdly, the minimum and maximum of the width of PWM signal is not exactly accurate. The minimum and maximum width should be 1 ms and 2 ms, respectively. But in the experiment, they are 1.034 ms and 1.948 ms. The reason accounts for this may be that: 1) there are noise signals from joystick; 2) it is nearly impossible to achieve the ideal minimum and maximum joystick value by manual control of joystick (The ideal minimum and maximum joystick value is 0 and 0x3FFF).

Fig. 4. Screen shots of logic analyzer measurements when servomotor is being driven at the max value of the joystick. The maximum width and period of PWM signal is 1.948 ms and 19.514 ms, respectively.

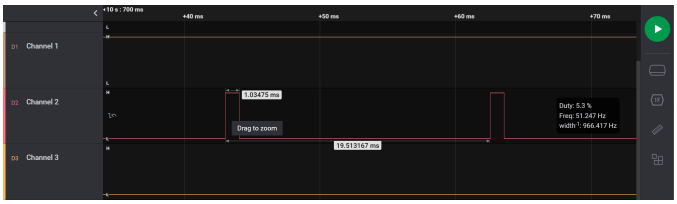

Fig. 5. Screen shots of logic analyzer measurements when servomotor is being driven at the min value of the joystick. The minimum width and period of PWM signal is 1.035 ms and 19.513 ms, respectively.

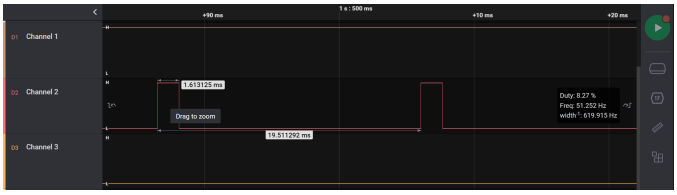

Fig. 6. Screen shots of logic analyzer measurements when servomotor is being driven at the middle value of the joystick. The middle width and period of PWM signal is 1.613 ms and 19.511 ms, respectively.