MCU Lab 2 - Week 8

MCU: Interrupt-based Timer and ADC

[2.5 Marks Total]

This computer lab will introduce the ATmega328P Timer and Analogue to Digital Converter (ADC), and their operations using Interrupt Service Routine (ISR), which a special type of function called upon hardware events (e.g. a time has reached its maximum value and is about to reset to 0). ISRs enable a timely processing of hardware events without delay and they play crucial roles in microcprocesor systems.

1. Objectives

- To set up a timer in a “Clear Timer on Compare match” (CTC) mode to make precise delay functions in a polling mode operation.
- To use the interrupt service routine (ISR) of the timer in a CTC mode to perform a periodic task.
- To use the ISR of an ADC to measure an analogue input voltage from a potentiometer.

2. Marking and Due Date

- Students can form a group (maximum of **two** students) to complete the lab tasks but the demonstration will be assessed **individually**. There is a total of **2.5 marks** allocated to this lab. You need to complete the tasks during the lab session or at latest before the following week lab (in which case, students need to upload a short video to Canvas capturing the operational tasks).

3. Activity #1: Preparation

- Construct a circuit, as shown in Fig. 1, adding two LEDs and a potentiometer to an analog input. Note that the middle pin of the potentiometer is the voltage-divider which varies between 0V and 5V.
- Connect the potentiometer output (pin 2) to one of the ADC input pins. The pin number is selected based on the last digit of your student ID number. Take the % (modulus, or remainder operator) of 6 of the last digit. For example, if your last digit of SID is 8, then 8 % 6 gives 2 (the remainder). Connect the potentiometer to A2 pin (in UNO board).
- Note that the ADC stores the data in a 10-bit register which gives the maximum value of 1023 (2^10-1).

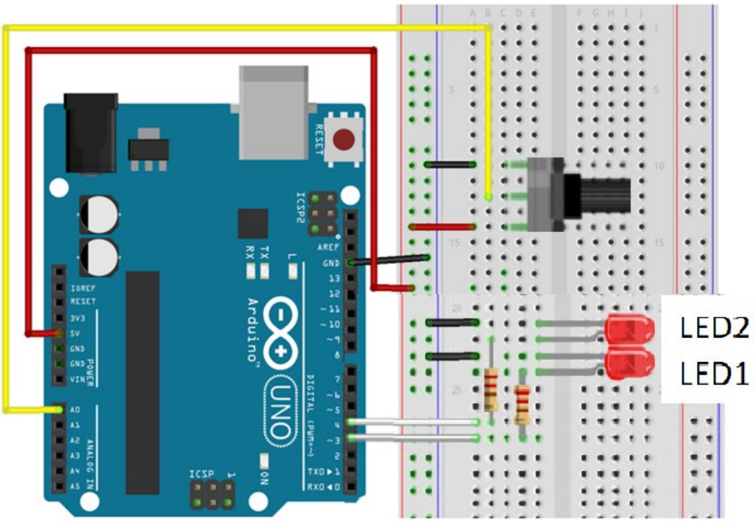


Figure 1. Two LEDs and a potentiometer connected to Atmega328P

- Pin Connection Summary

DIO pins	Setup	Connection
PD3 (PORT D3, or UNO pin-5)	Output mode	An LED1 is connected
PD4 (PORT D4, or UNO pin-4)	Output mode	An LED2 is connected
PC[X], X= [Last digit of SID] % 6	Analog Input	A Potentiometer output (middle pin) is connected

4. Activity #2: Delay function using Timer0 in CTC mode

- Download “***mcu2a.cpp***” from Canvas.
- Configure Timer0 in the CTC (Clear Timer Compare) mode so that it overflows exactly every 1ms (milliseconds).
- To make the Timer0 overflow at every 1ms using the system clock frequency of 16 MHz, you would need to select the pre-scaler value and the TOP value (OCR0A). The OCR0A value is the number of timer ticks minus 1 (as it starts from 0).

Prescaler	Number of timer ticks in 1 sec	Number of timer ticks during 1 ms	Comment
1	16M	16000	> 255 (too big for OCR0A)
8	2M	2000	> 255 (too big for OCR0A)
64	250K	250	Okay
256	62.5K	62.5	Okay but Fractional
1024	15.625K	15.625	Okay but Fractional

- Complete the provided function (my_delay_ms) which takes an input value of time in a millisecond, and polls the timer register (TIFR0) until the OCF0A flag is set (which should happen at every 1 ms if the timer is set up properly). Then it increases a counter value and repeats the process until the counter value reaches the required input delay time. Use my_delay_ms() to blink LED1 at 1Hz.

```
void my_delay_ms (unsigned long time_ms)
{
        // poll/wait the register TIFR0 to check the OCF0A-bit
        // Once OCF0A-bit is set, write '1' to clear the bit
        // increment a millisecond counter
        // repeat until the millisecond counter reaches time_ms
}
```

- Show the results to tutors for marking.

Input	LED1 output	Marking
Polling-based Timer0 in CTC mode.	my_delay_ms() works for blinking LED1	/1 mark

Task 2: Interrupt-based Timer and ADC: Using a potentiometer to control the LED period

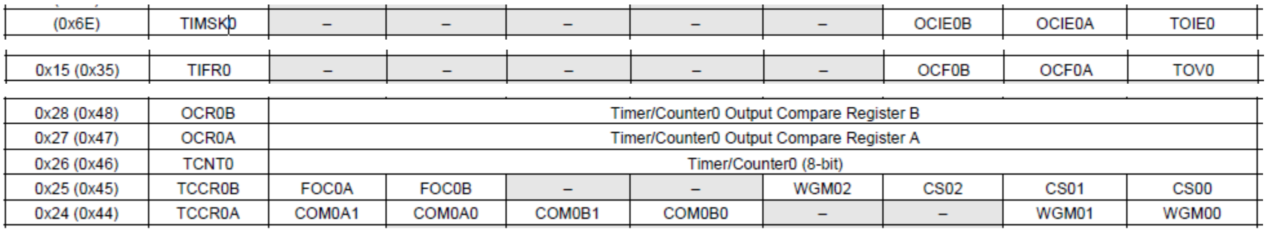
- Download "***mcu2b.cpp***" and test it on the UNO board. LED2 (PD4) should blink at 1Hz, which should be running during the task.
- Set up the Timer0 in a CTC mode as in Task1. This time enables the output compare Interrupt bit (OCIE0A) bit in the TIMSK0 register.
- Program the ISR (TIMER0_COMPA_vect) function so that the number of 1-millisecond events is counted and any tasks such as toggling the LED1 are activated.
- Set up the ADC to read the potentiometer output. Use the potentiometer to vary the toggling-period of LED1. That is, when the potentiometer reading is 0 (0V), the LED1 toggles at every 0ms (always ON), and when potentiometer reading is 1023 (5V), the LED1 toggles at every 1023ms (roughly 0.5Hz), and changes linearly between 0 and 5V.
- Modify the code such that, when the potentiometer reading is 0, LED1 toggles at every 100 ms (100ms ON, and 100ms OFF, giving 5Hz frequency), and when the potentiometer reading is 1023, the LED1 toggles at every 500ms (500ms ON, 500ms OFF, giving 1Hz). In between, the toggling-period should change linearly proportional to the ADC value.

- Show the results to tutors for marking.

Input	LED1 output	Marking
Potentiometer output is	LED1 is ON	/0.5 marks

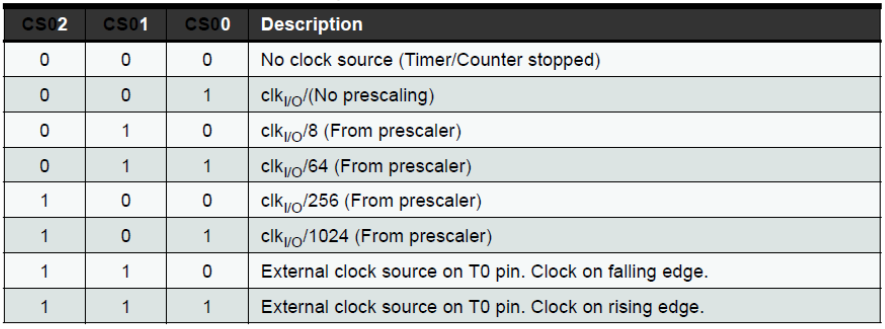
0V (or minimum)		
Potentiometer output is 5V (or maximum)	LED1 blinks at 0.5Hz	/0.5 marks
Potentiometer output is between 0V and 5V	Blinking frequency changes [5-1]Hz	/0.5 marks
Note) LED2 should blink at 1Hz during these operations		

Appendix: Registers related to Timer and some Bits (Clock Select and Mode)



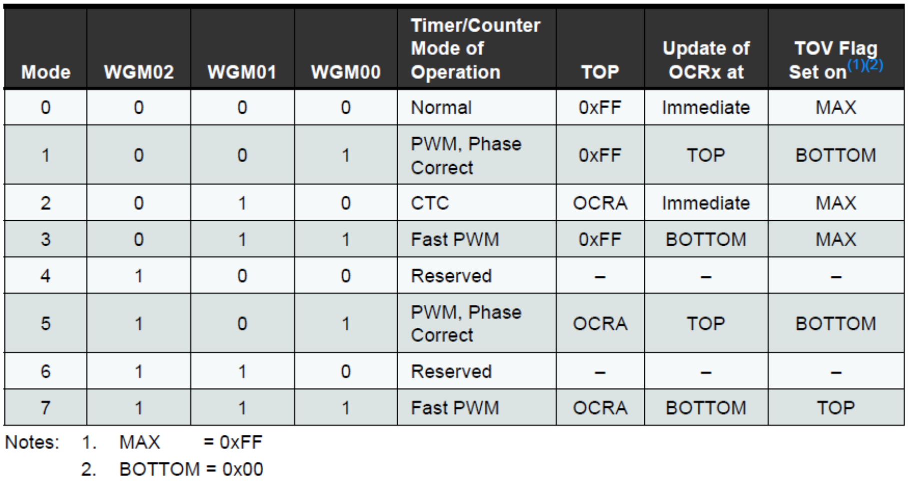
(0x6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B							
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A							
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)							
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00

Table 15-9. Clock Select Bit Description



CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Table 15-8. Waveform Generation Mode Bit Description



Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00