**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Arbab Latif  **Dated:** 18/04/2022

**Semester:** 4th **Section:** BEE 12C

**EE-222: Microprocessor Systems**

Lab 10: Pulse Width Modulation

Group Members

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg. No** | **Report**  **Marks / 10** | **Viva**  **Marks / 5** | **Total**  **Marks / 15** |
| Danial Ahmad | 331388 |  |  |  |
| Muhammad Umer | 345834 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Experiment

## Objectives

1. Solve daily life problems through microcontroller.
2. Develop a voltmeter using ADC.
3. Propose your engineering solution to the problem.

## Equipment

Software

* *Atmel Studio*

Hardware

* ATmega16A Microcontroller Unit
* Universal Programmer
* Seven Segment Display
* Resistance 47Ω
* LEDs (may use from trainer kit)
* USB-TTL Converter.

## Introduction

 Atmel Studio IDE is a free development environment for programming Atmel MCUs, sourced by Microchip Technology Inc. It provides us with the means to simulate assembly language codes on specific Microcontrollers and provides an easy and intuitive way of producing .HEX files, which are what makes burning the code on the hardware possible.

In this specific lab, we familiarize ourselves with the extension of application of timers present within the ATmega16A. We use specific values of Output Compare Register (OCR) to generate a pulse with desired duty cycle. To implement a practical application of PWM, we use a potentiometer (control voltage) to directly change the brightness of an LED.

# Lab Tasks

## Task A

Implement the code in example above, just to get familiar with ATmega16 PWM.

**#define *F\_CPU* 1000000UL**

**#include <avr/io.h>**

**#include <util/delay.h>**

**void PWM\_init()**

**{**

**/\*set fast PWM mode with non-inverted output\*/**

**TCCR0 = (1<<WGM00) | (1<<WGM01) | (1<<COM01) | (1<<CS00);**

**DDRB |= (1<<PORTB3); /\*set OC0 pin as output\*/**

**}**

**int main ()**

**{**

**unsigned char duty;**

**PWM\_init();**

**while (1)**

**{**

**for(duty=0; duty<255; duty++)**

**{**

**OCR0=duty; /\*increase the LED light intensity\*/**

***\_delay\_ms*(8);**

**}**

**for(duty=255; duty>1; duty--)**

**{**

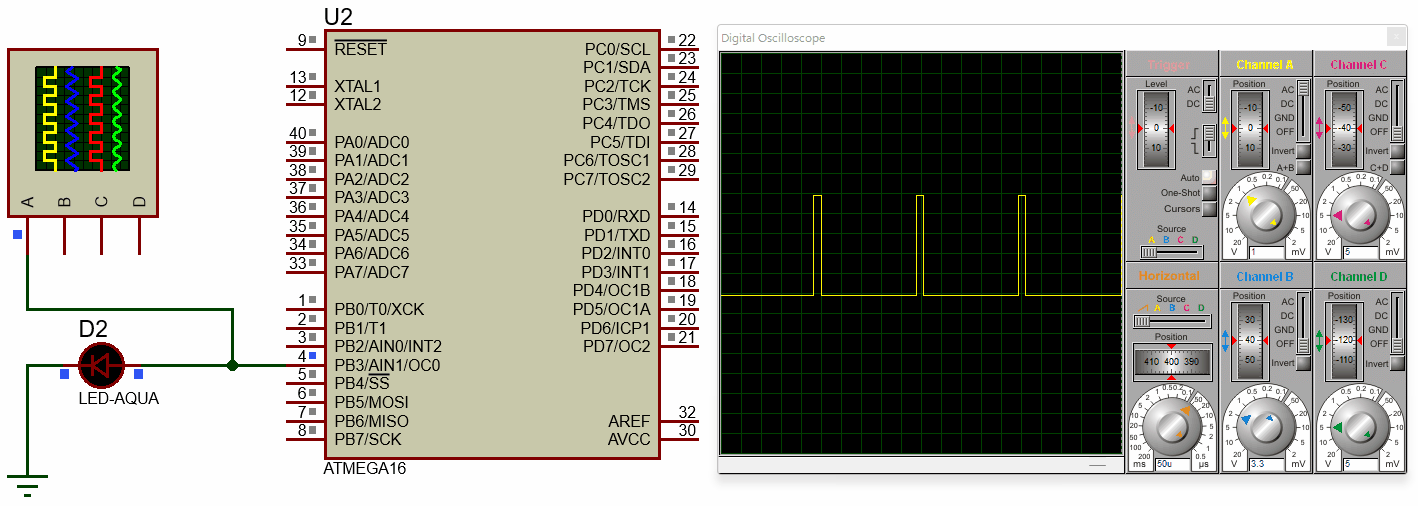
**OCR0=duty; /\*decrease the LED light intensity\*/**

***\_delay\_ms*(8);**

**}**

**}**

**}**



Simulation

## Task B

Use a potentiometer to control the brightness of an LED using PWM in ATmega16. Simulate the circuit on Proteus and patch the hardware. Use Oscilloscope to observe the duty cycle of the generated PWM signal as the knob of potentiometer changes.

As the value of duty cycle changes, print out the duty cycle in percentage on 7 Segment display as well as on the Serial port of PC using UART.

**/\***

**\* Lab10.c**

**\***

**\* Created: 18/04/2022 4:08:04 pm**

**\* Author : Danial and Umer**

**\*/**

**#define *F\_CPU* 1000000UL**

**#include <avr/io.h>**

**#include <math.h>**

**#include <util/delay.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**char SevenSegment(char number);**

**double num;**

**double percentage;**

**void UART\_initializer(){**

**UBRRL = 0x0C ; //set baud rate to 4800**

**UCSRB |= (1<<TXEN); //enable transmitter**

**UCSRC |= (1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1) ; //set data size**

**}**

**// initialize adc**

**void adc\_init()**

**{**

**// AREF = AVcc**

**ADMUX = (1<<REFS0);**

**// ADC Enable and prescaler of 8**

**// 1000000/8 = 125000**

**ADCSRA = (1<<ADEN)|(1<<ADPS1)|(1<<ADPS0);**

**}**

**// read adc value**

***uint16\_t* adc\_read()**

**{**

**// start single conversion**

**// write '1' to ADSC**

**ADCSRA |= (1<<ADSC);**

**while(ADCSRA & (1<<ADSC));**

**return ADC;**

**}**

**//function to transmit data over UART**

**void UART\_transmitter(double a){**

**double temp\_1 = (a / 256) \* 100;**

**char temp\_2;**

**char msg[4];**

**char d = '\n';**

**char f = '\r';**

**temp\_2 = (int)temp\_1;**

***itoa*(temp\_2, msg, 10);**

***strncat*(msg, &d, 1);**

***strncat*(msg, &f, 1);**

**unsigned char i;**

**for (i = 0 ; i < 4 ; i=i +1){**

**while (!(UCSRA & (1<<UDRE)));**

**UDR = msg[i];**

**}**

**}**

**void split\_send()**

**{**

**char temp;**

**percentage = (num / 256) \* 100;**

**PORTC = (int)percentage / 10;**

**temp = (int)percentage % 10;**

**temp = temp << 4;**

**PORTC |= temp;**

**}**

**int main(void)**

**{**

**char temp\_num;**

**UART\_initializer();**

**DDRC = 0xFF;**

**PORTC = 0x00;**

**TCCR0 = (1<<WGM00) | (1<<WGM01) | (1<<COM01) | (1<<CS00);**

**DDRB |= (1<<PORTB3); // set OC0 pin as output**

**adc\_init();**

**while (1)**

**{**

**adc\_read();**

**num = ADC / 4;**

**OCR0 = num;**

**if (temp\_num == num)**

**{**

**split\_send();**

**}**

**else**

**{**

**split\_send();**

**UART\_transmitter(num);**

**}**

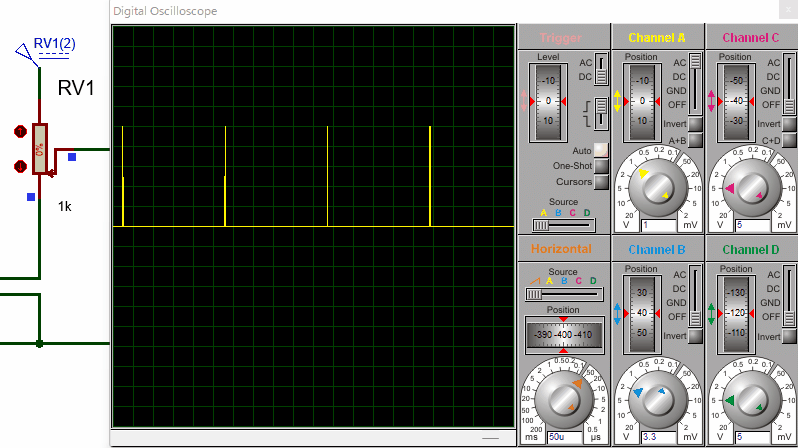
**temp\_num = num;**

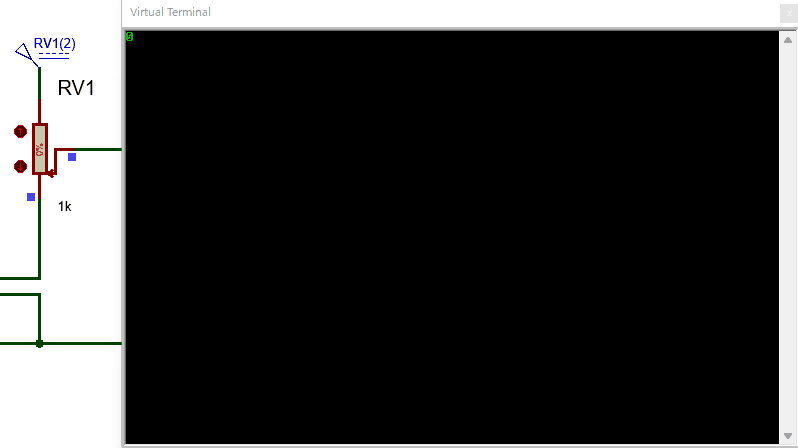
***\_delay\_ms*(50);**

**}**

**}**

## 





## Conclusion

After the conduction of this lab, we have learnt how to initialize and setup the TIMERS present within the ATmega16A to generate a pulse of a specific duty cycle; Pulse Width Modulation. PWM is an essential feature to drive motors and other components. To demonstrate one such application, we changed the brightness of an LED by altering the width of the PULSE, and this change was a direct function of the voltage reading being input to the ADC.