Lab report 07

**Generating a PWM Waveform**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**307L-MBSD LAB**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Shahzada Fahim Jan**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

Date: **Sun, 28th April, 2024**

**Lab 07**

**Generating a PWM Waveform**

**Objective:**

* To Learn how to create exact delays using timers in 8051 micro-controller

**Components needed for this lab:**

* Keil µVision IDE
* Proteus Software
* 8051 Micro-controller

**8051 Micro-controller:**

* 8051 have 40 pins dedicated for various functions such as I/O, -RD, -WR, address, data, and interrupts.
* The 8051 has an on-chip oscillator but requires an external clock to run it. A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18). The quartz crystal oscillator also needs two capacitors of 30 pF value



Fig 01: 8051 Pin Diagram

**Tasks**

**1: Generate a signal of 500Hz with 40% duty cycle**

#include *<reg51.h>*

sbit BIT = P1^3;

void start\_timer()

{

TMOD = 0x01;

IE = 0x82;

}

void timer() interrupt 1

{

**if**(!BIT)*//*

{

TH0=0xFC;

TL0=0xDC;

}

**else**

{

TH0=0xFB;

TL0=0x4F;

}

}

void init()

{

TR0 = 1;

}

void main()

{

init();

start\_timer();

**while**(1)

{

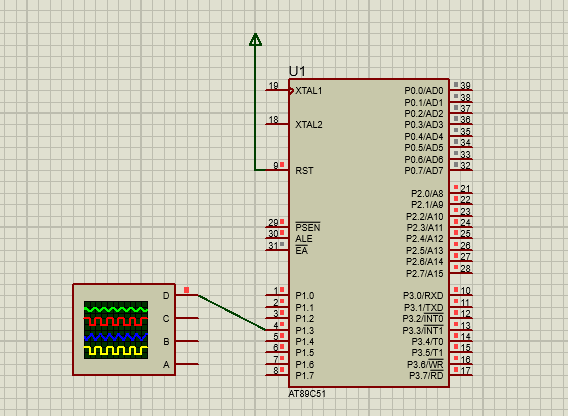
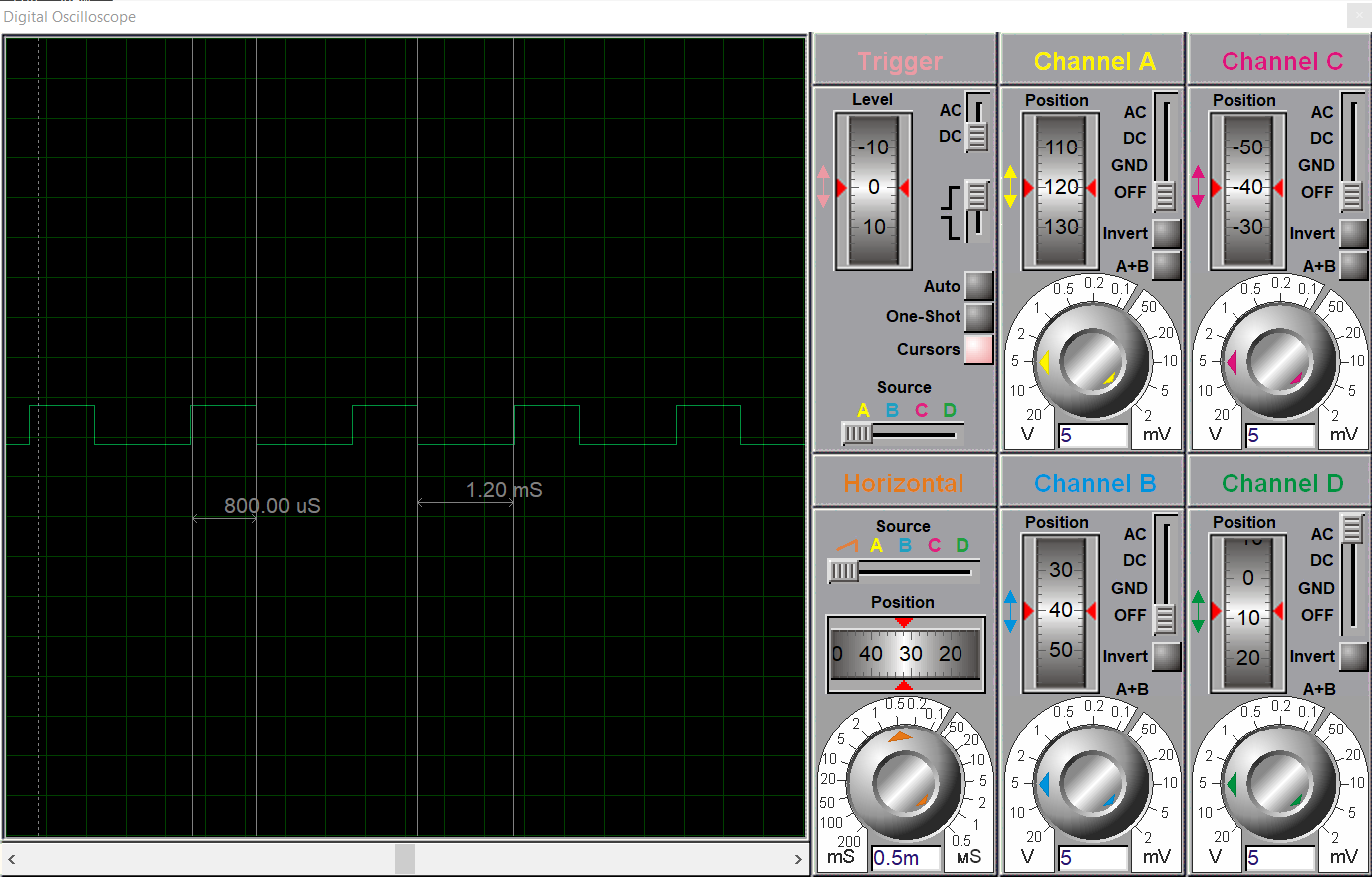
**while**(TF0==0);

BIT = ~BIT;

}

}

8051 C Code to show 15 ms delay using timer0



Oscilloscope is connected to the port 1.3

40% duty cycle on P1.3 for 500Hz

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2: Generate a signal of 600Hz with 60% duty cycle on P1.3**

#include *<reg51.h>*

sbit BIT = P1^3;

void start\_timer()

{

TMOD = 0x01;

IE = 0x82;

}

void timer() interrupt 1

{

**if**(!BIT)*//*

{

TH0=0xFC;

TL0=0x17;

}

**else**

{

TH0=0xFD;

TL0=0x61;

}

}

void init()

{

TR0 = 1;

}

void main()

{

init();

start\_timer();

**while**(1)

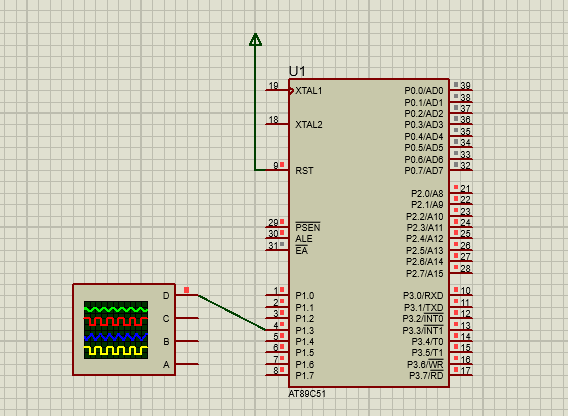
{

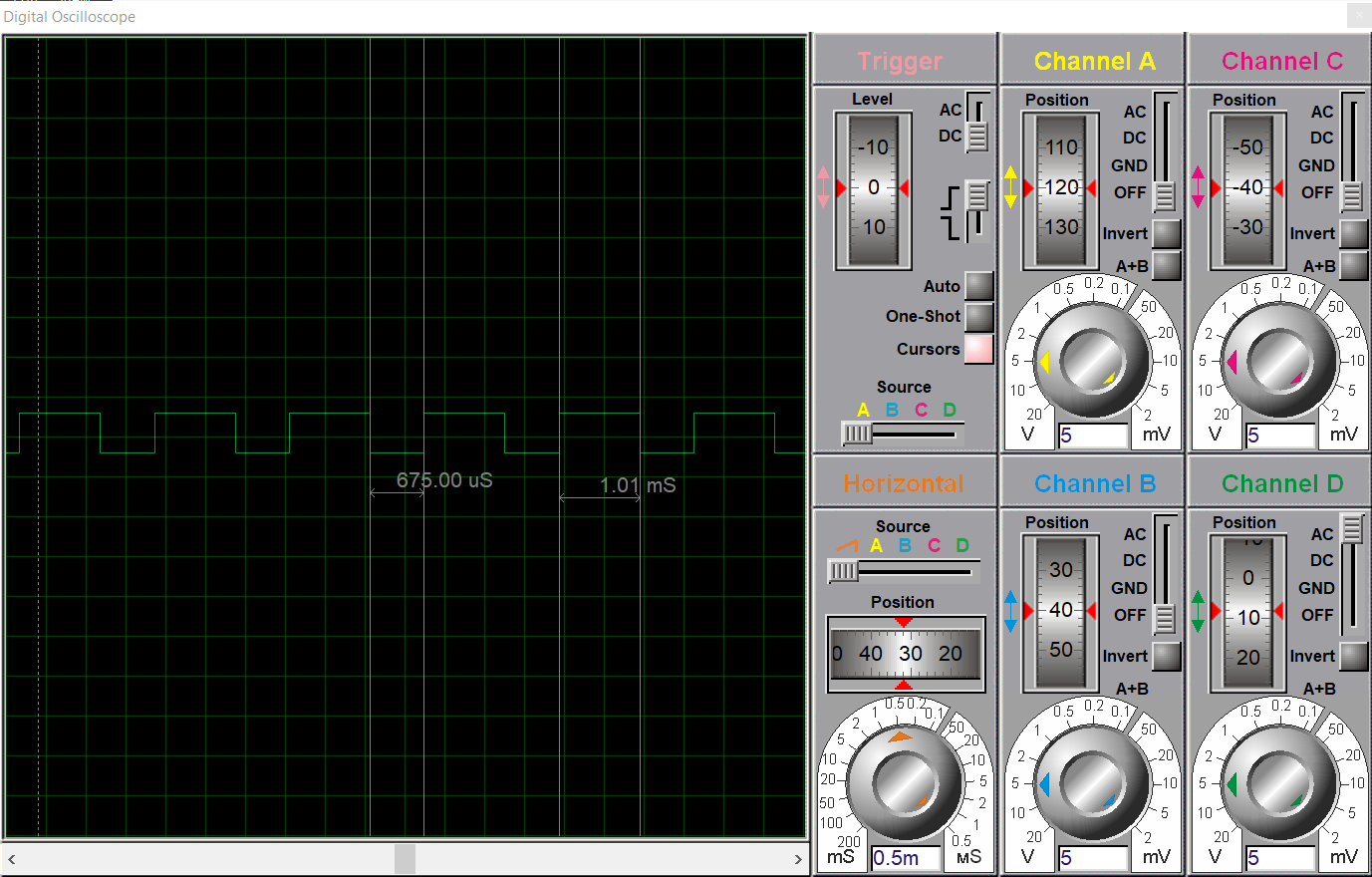
**while**(TF0==0);

BIT = ~BIT;

}

}

 Fig 04: LED and Oscilloscope is connected to the port 1.3



60% duty cycle on P1.3

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3: Generate a signal of 100Hz with 40% duty cycle on P1.2 When a user presses a button at P2.3 the signal change to 300Hz with 60% duty cycle.**

#include *<reg51.h>*

sbit BIT = P1^2;

sbit button = P2^3;

void start\_timer()

{

TMOD = 0x01;

IE = 0x82;

}

void timer() interrupt 1

{

**if**(!button)

{

**if**(!BIT)*//*

{

TH0=0xf0;

TL0=0x5f;

}

**else**

{

TH0=0xe8;

TL0=0x8f;

}

}

**else**

{

**if**(!BIT)*//*

{

TH0=0xf8;

TL0=0x43;

}

**else**

{

TH0=0xfa;

TL0=0x55;

}

}

}

void init()

{

TR0 = 1;

}

void main()

{

init();

start\_timer();

**while**(1)

{

**while**(TF0==0);

BIT = ~BIT;

}

}

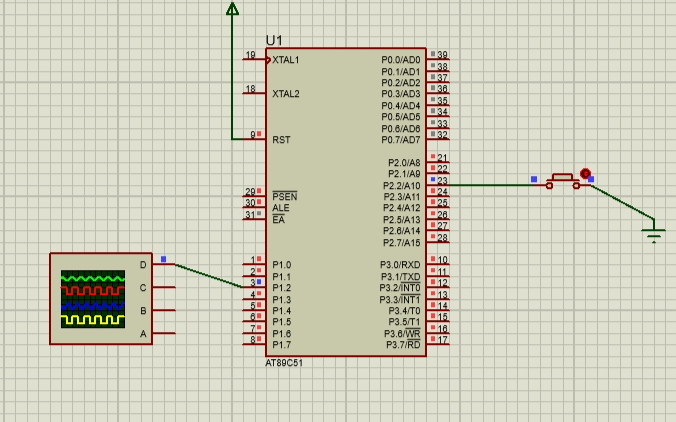
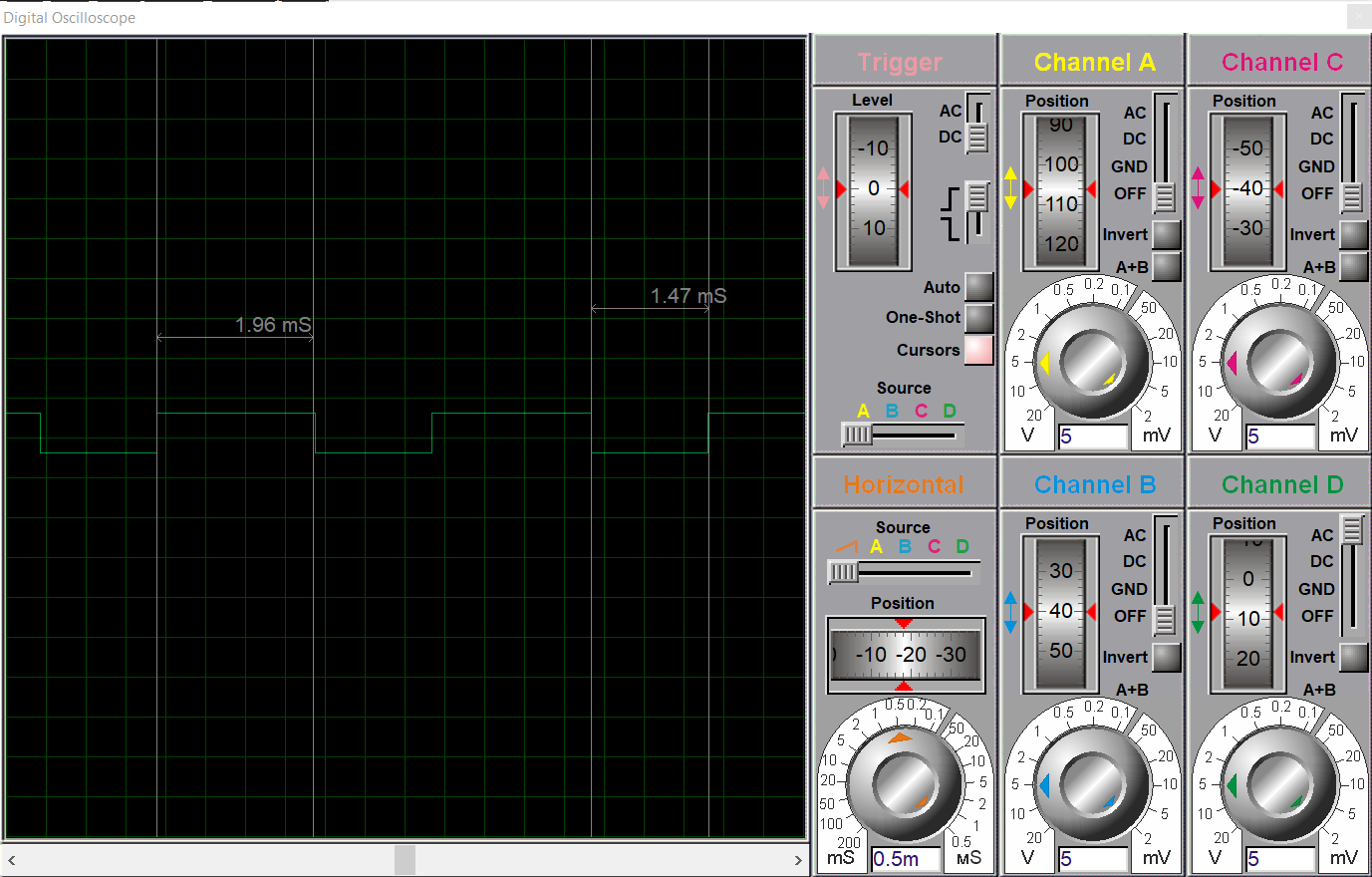
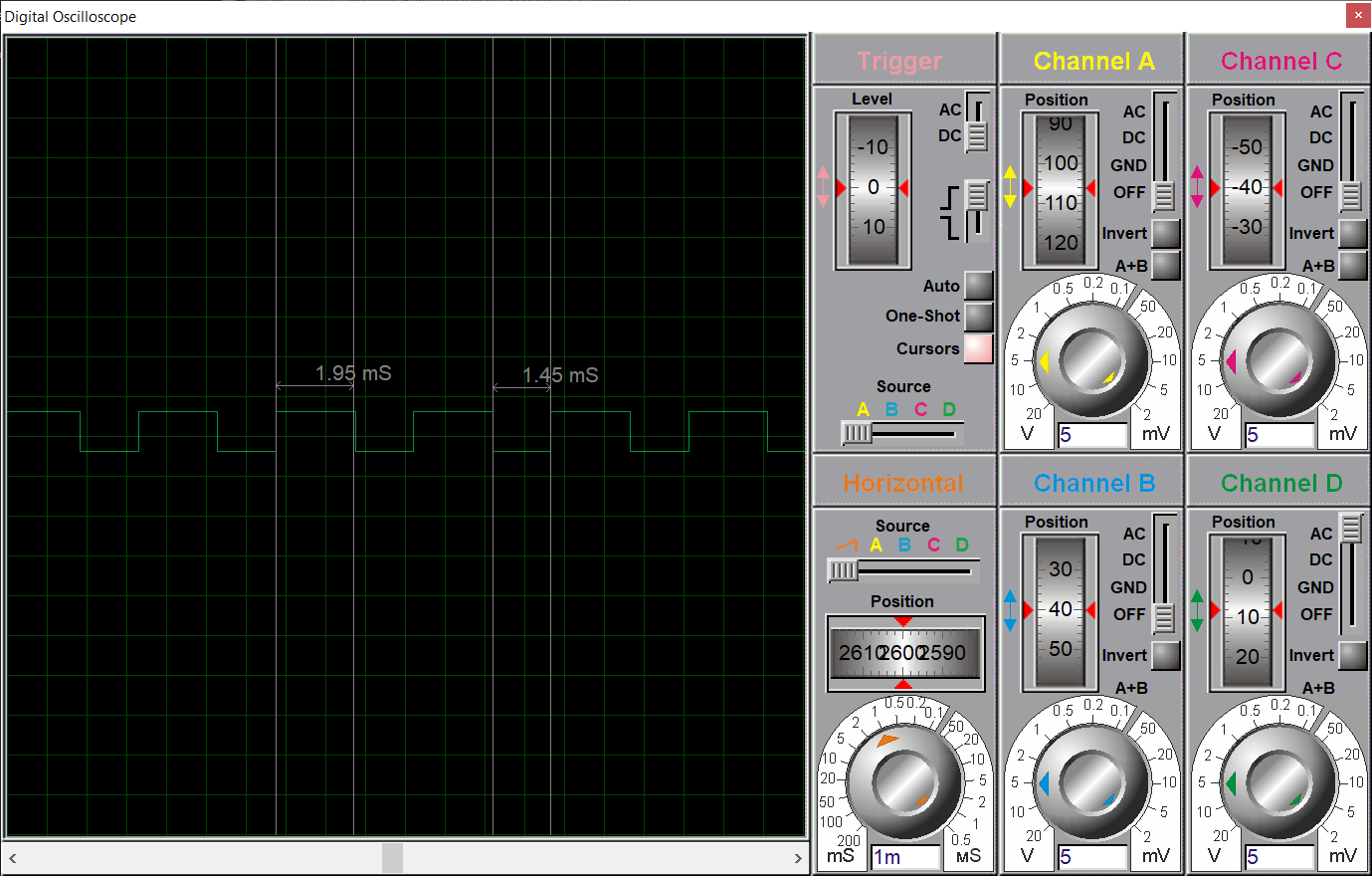


Fig 06: Oscilloscope is connected to the port 1.3



100Hz with 40% duty



300Hz with 60% duty

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion:**

In this Lab we learned about generating pwm waves and duty cycles. We also learned about how generate the form the frequency the duty cycle given.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**