

Ahmedabad
University

ECE302 Embedded Systems Design

Assignment 3

Section 1

Submitted to faculty: Anurag Lakhani

Date of Submission: November 10, 2021

Student Details

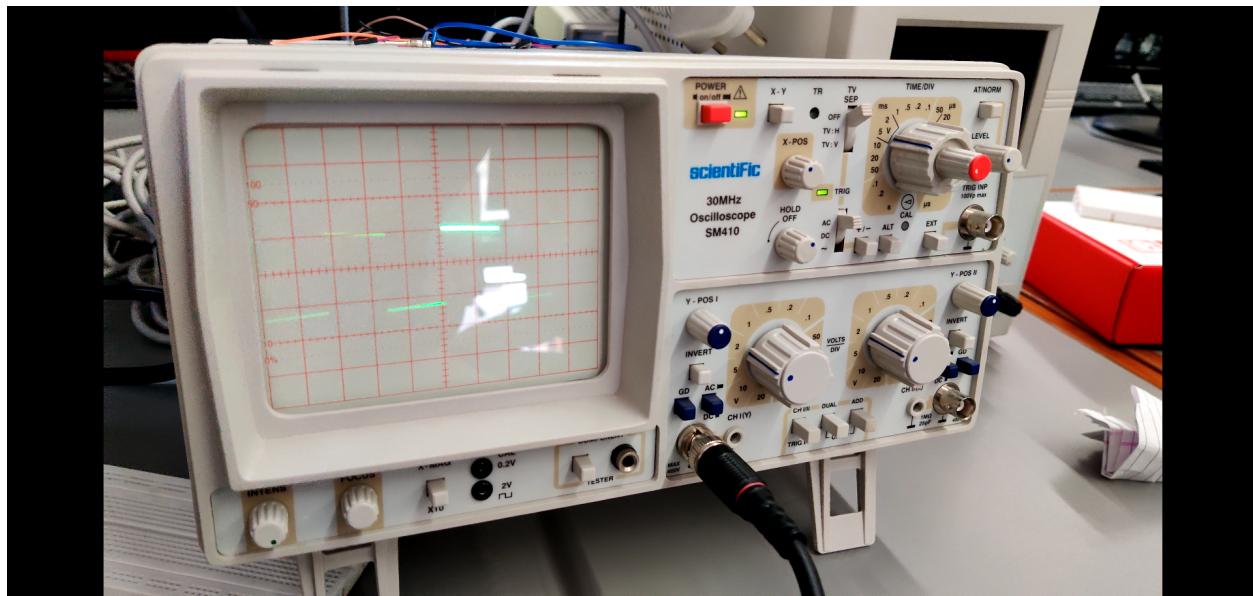
Roll No.	Name of the Student	Name of the Program
AU1940049	Sameep Vani	BTech CSE

2021-2022 (Monsoon Semester)

Note that all circuit diagrams can be accessed [here](#)

Experiment 1 (a)

Experiment Result



Description

In this experiment, we had to generate a square wave using the timer1 circuit and overflow flag. The circuit setup is quite simple, the PORTA.b0 is connected with the red cable of the alligator and the black cable is connected to the ground of the microcontroller. In this particular question, we had to run timer1 with a full cycle (TCNT will count from 0 to 65535). Also note that since we are using the timer1 circuit, it has 16-bit registers, and hence its value is set in two lines according to two 8-bit registers.

Theoretical Frequency - 61Hz

Calculated Frequency - 50Hz (5ms for 4 blocks which makes 20ms)

Video

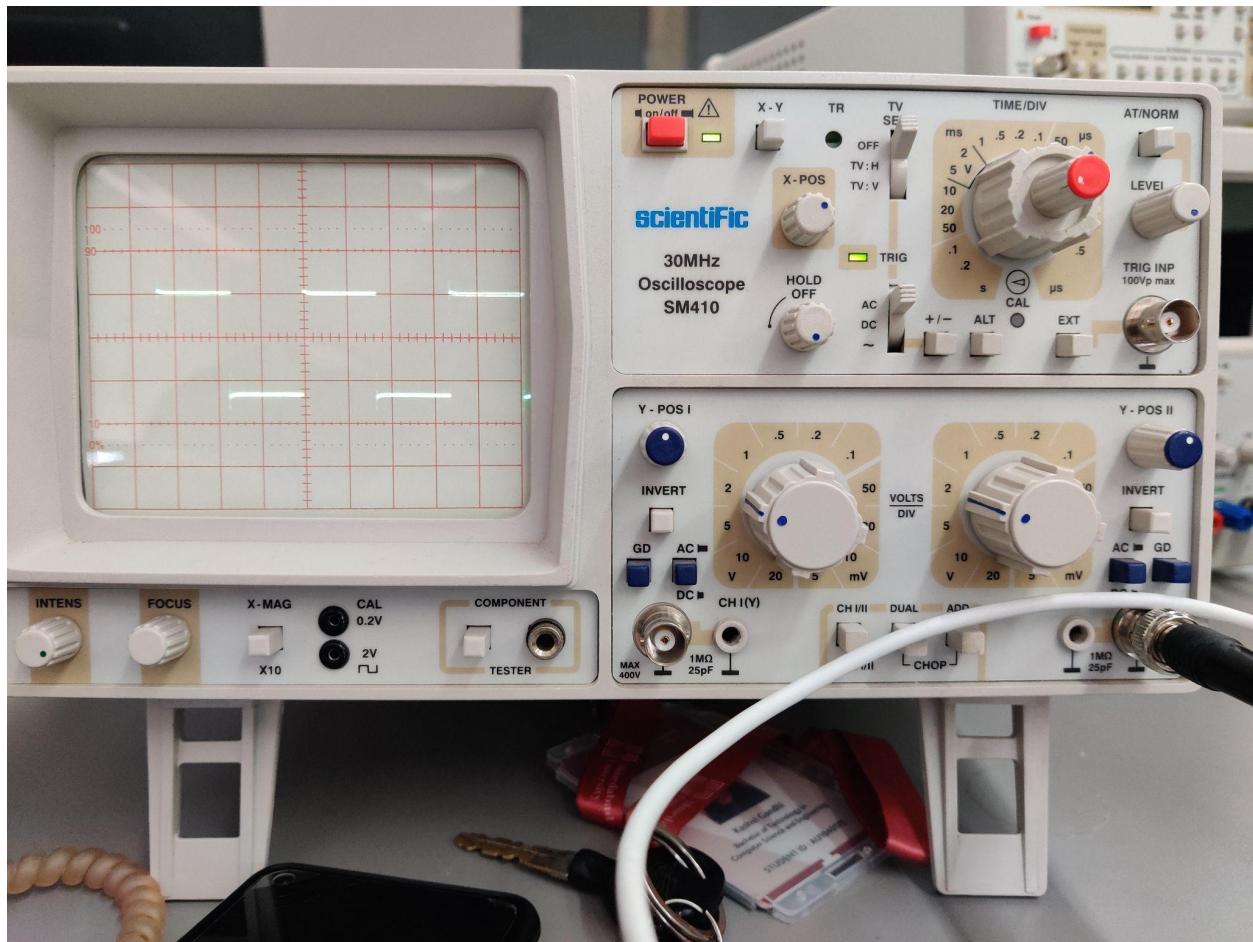
[Link](#)

Code

```
void T1Delay()  
{  
    TCCR1A = 0x00;  
    TCCR1B = 0x01;  
    TCNT1H = 0x00;  
    TCNT1L = 0x00;  
  
    while((TIFR & 0x04) == 0);  
    TIFR = 0x04;  
    TCCR1A = 0x00;  
    TCCR1B = 0x00;  
}  
  
void main() {  
    DDRA = 0x01;  
    while(1)  
    {  
        PORTA ^= 0x01;  
        T1Delay();  
    }  
}
```

Experiment 1 (b)

Experiment Result



Description

This question is similar to the above experiment. The only difference is that the timer1 here runs in CTC mode instead of normal mode. Thus, the value of OCR is calculated and the value of TCCR is set accordingly.

Theoretical Frequency - 17.02 kHz

Calculated Frequency - 14 kHz

Video

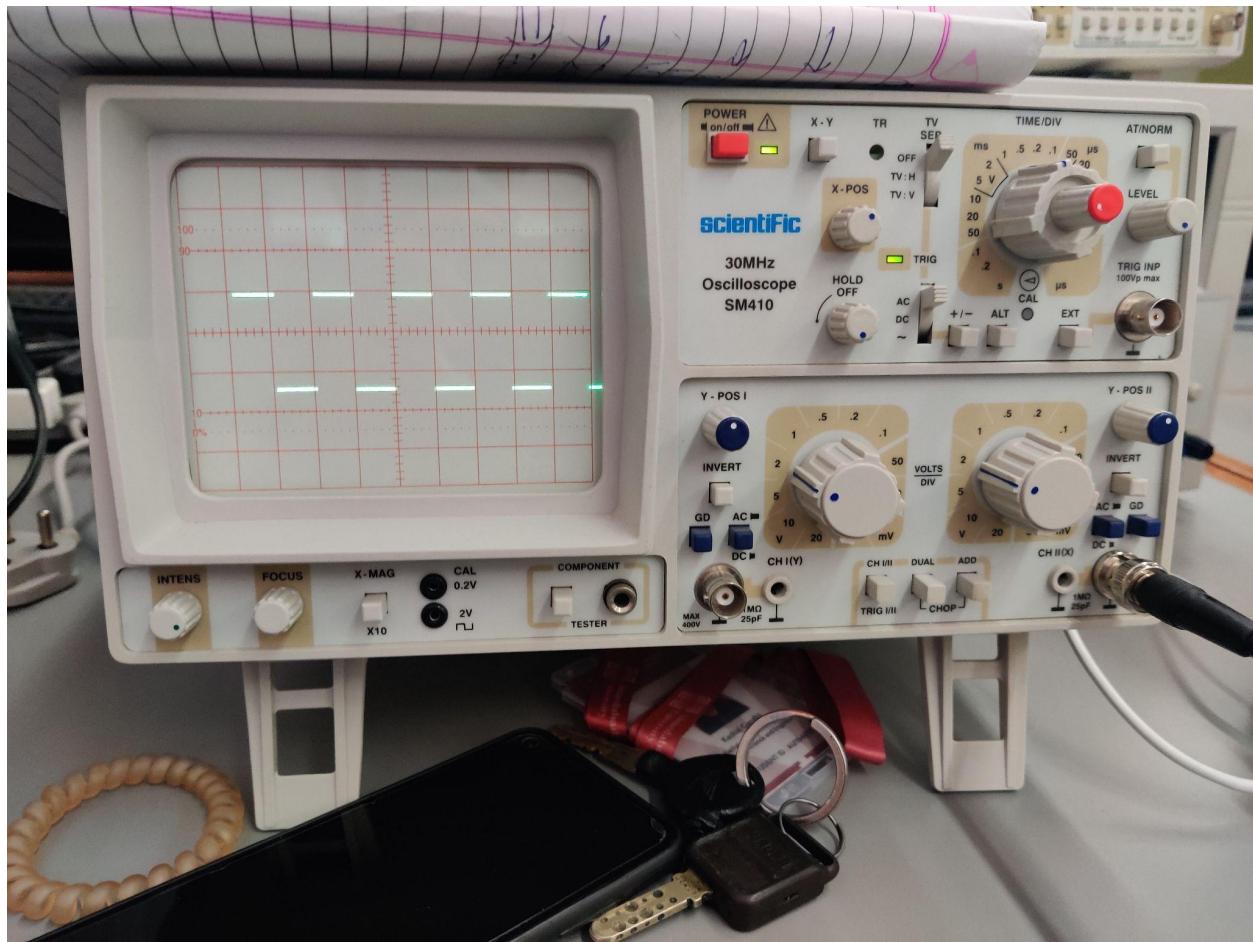
[Link](#)

Code

```
void T1Delay()  
{  
    TCCR1A = 0x00;  
    TCCR1B = 0x09;  
    TCNT1H = 0x00;  
    TCNT1L = 0x00;  
    OCR1AL = 235;  
    OCR1AH = 0x00;  
  
    while((TIFR & 0x08) == 0);  
    TIFR = 0x08;  
    TCCR1A = 0x00;  
    TCCR1B = 0x00;  
}  
  
void main() {  
    DDRA = 0x01;  
    while(1)  
    {  
        PORTA ^= 0x01;  
        T1Delay();  
    }  
}
```

Experiment 2

Experiment Result



Description

In this experiment, we had to generate one square wave using two different prescaler and timer1 circuit. The end result of both the prescaler will remain the same. Furthermore, the time delay for the square wave is 1kHz which means the time delay for 1 full cycle is 1ms. This means that the time delay

for the half-cycle is 0.5ms. This can also be observed from the result obtained above. The circuit is similar to the experiments completed above.

Video

[Link 1](#), [Link 2](#)

Code

Part 1

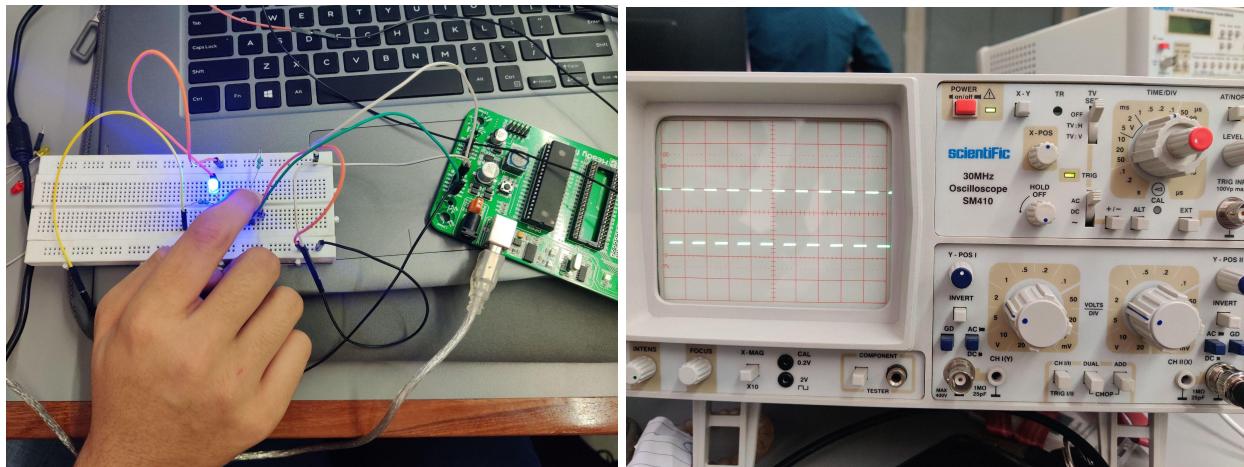
```
T1Delay()  
{  
    TCNT1L = 0x0B;  
    TCNT1H = 0xFE;  
    TCCR1B = 0x02;  
    TCCR1A = 0x00;  
  
    while((TIFR & 0x04) == 0);  
    TIFR = 0x04;  
    TCCR1A = 0;  
    TCCR1B = 0;  
}  
  
void main() {  
    DDRA = 0x01;  
    while(1)  
    {  
        PORTA ^= 0x01;  
        T1Delay();  
    }  
}
```

Part 2

```
T1Delay()  
{  
    TCNT1L = 0xF1;  
    TCNT1H = 0xFF;  
    TCCR1B = 0x04;  
    TCCR1A = 0x00;  
  
    while((TIFR & 0x04) == 0);  
    TIFR = 0x04;  
    TCCR1A = 0;  
    TCCR1B = 0;  
}  
  
void main() {  
    DDRA = 0x01;  
    while(1)  
    {  
        PORTA ^= 0x01;  
        T1Delay();  
    }  
}
```

Experiment 3

Experiment Result



Description

In this experiment, we had to generate a square wave at PORTA.b0 of frequency 2 kHz which means 1ms would accommodate the full cycle. Furthermore, at the same time, we had to take input from PORTB.b0 using a push-button switch and display it using led on PORTB.b1. This can be achieved using an interrupt which is called every time timer1's overflow flag turns on while simultaneously reading a value from PORTB.b0 and sending it to PORTB.b1. Thus, the connections regarding the switch will hold true here. Further, connections similar to experiment 1 (a) should also be there in order to generate the square wave.

Video

[Link](#)

Code

```
void main() {  
    DDRA = 0x01;  
    DDRB = 0x02;  
  
    TCCR0 = 0x02;  
    TCNT0 = 6;  
  
    TIMSK = 0x01; // Local  
    SREG.b7 = 1; // Global  
  
    while(1)  
    {  
        PORTB.b1 = PINB.b0;  
    }  
}  
  
void timer0_ovf_isr(void) org 0x012  
{  
    PORTA ^= 0x01;  
    TCNT0 = 6;  
}
```