Ahmedabad
University

ECE302 Embedded Systems Design

Lab Assignment 2

Section 1

Submitted to faculty: Anurag Lakhlani

Date of Submission: October 27, 2021

Student Details

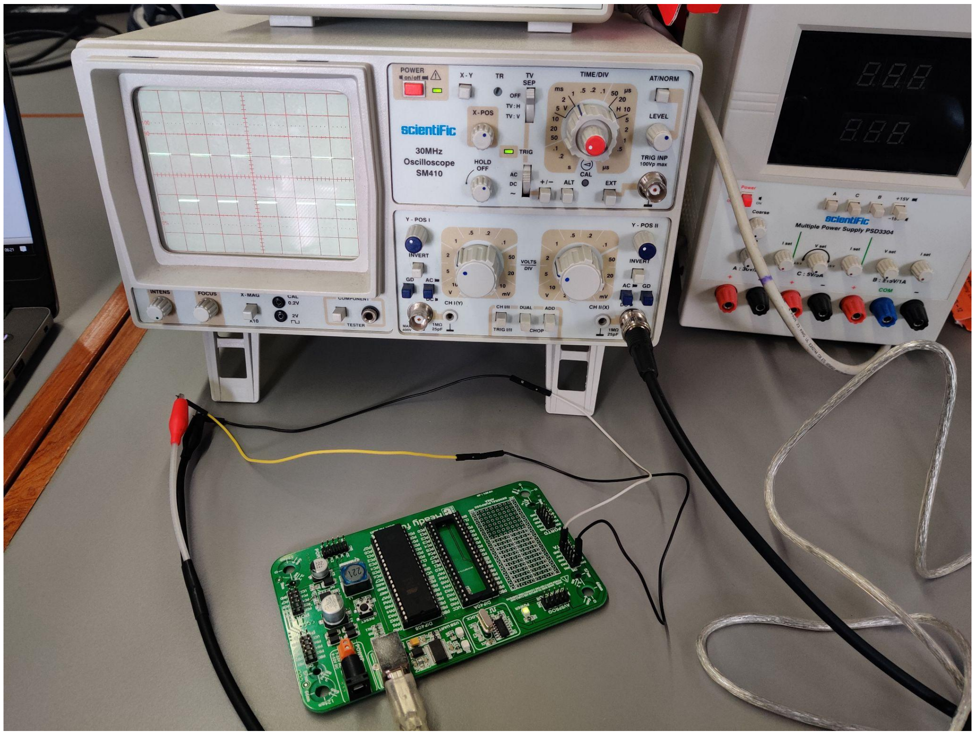
Roll No.	Name of the Student	Name of the Program
AU1940049	Sameep Vani	BTech CSE

2021-2022 (Monsoon Semester)

Note all circuit diagrams can be accessed here.

Experiment 1

Experiment Set-up



Description

In this question, we had to generate a square wave of 500Hz by toggling PORTA.b0 and applying some time delay. As mentioned in the question, the time delay to be applied for the **whole wave** is 2 ms. This means that the time delay for one pulse is 1 ms.

Now, we can see in the image above, we have 1 block and that block has a delay of 1ms. This proves the correctness of my code. Furthermore, the connections to CRO were simple. Connect the 0th pin of PORTA to the red portion of the alligator and the ground with the black portion. The other end of the alligator is connected to the CRO.

Video

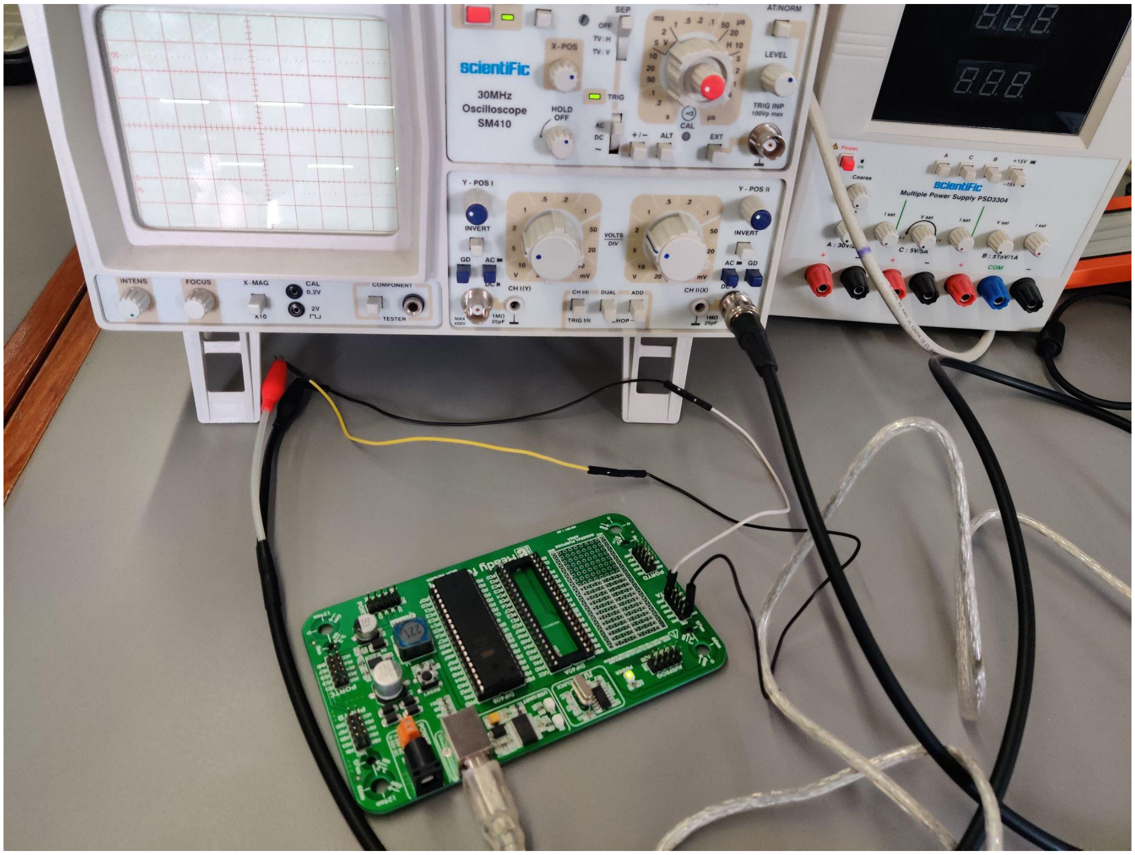
Link

Code

```
void main() {
    DDRA = 0x01;
    while(1)
    {
     PORTA = 0x01;
     Delay_ms(1);
     PORTA = 0x00;
     Delay_ms(1);
    }
}
```

Experiment 2a

Experiment Set-up



Description

Here, we needed to generate a square wave by toggling PORTA.b0 using the timer0 circuit in normal mode. The connections are similar to the previous question. Furthermore, as we can see in the code below, it is evident that the timer counter register runs for a full cycle i.e from 0 to 255, and rolls over at 256. This means that the delay is 64 microseconds. This means that the theoretical frequency is 15.625 kHz.

Now as we can see in the image above, the time delay for **2 blocks** in CRO is approximately 20 microseconds each which means 40 microseconds. This means the time delay for the whole square wave is 80 microseconds. This means that the calculated frequency is 12.5 kHz.

Video

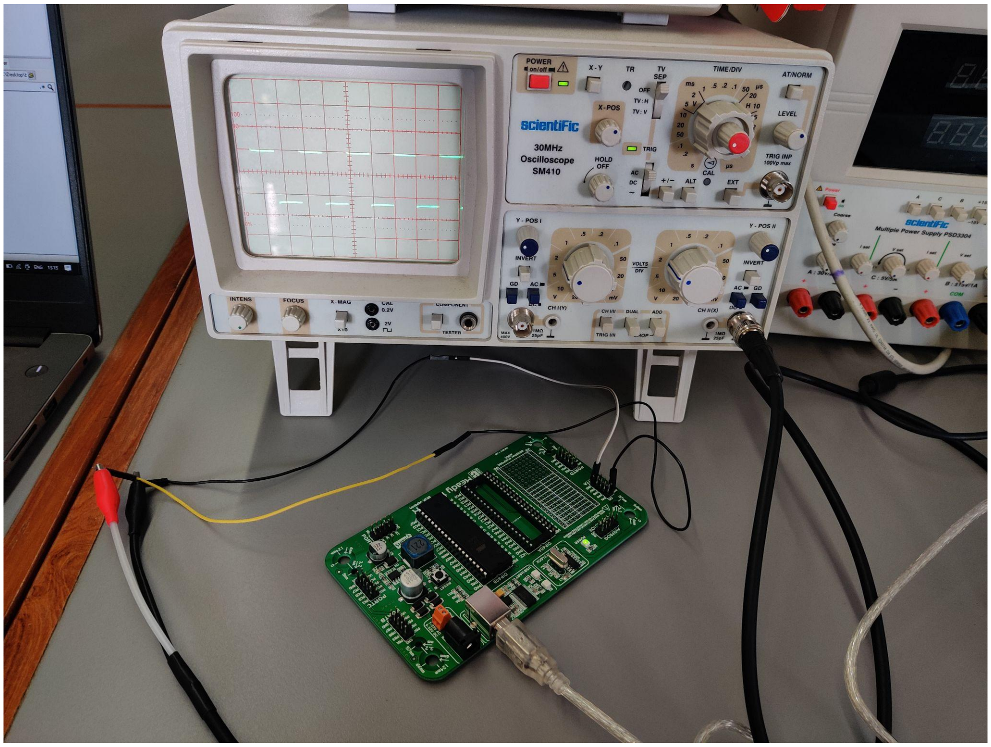
Link

Code

```
void T0Delay() {
     TCNT0 = 0x00;
     TCCR0 = 0x01;
     while((TIFR&0x1) == 0);
     TIFR=0x1;
}
void main() {
    DDRA = 0x01;
    while(1)
    {
       PORTA = 0x01;
       T0Delay();
       PORTA = 0x00;
       T0Delay();
    }
}
```

Experiment 2b

Experiment Set-up



Description

This experiment was similar to the above-performed experiment. Here the timer0 counter runs from 0 to 150 and rolls over. For this, the theoretical time delay is 37.5 microseconds and the theoretical frequency is 26.67 kHz.

As visible in the photo, **1 block** of square wave has a 20 seconds delay and thus the whole wave has a 40 seconds delay. Thus, the calculated frequency becomes 25 kHz.

Video

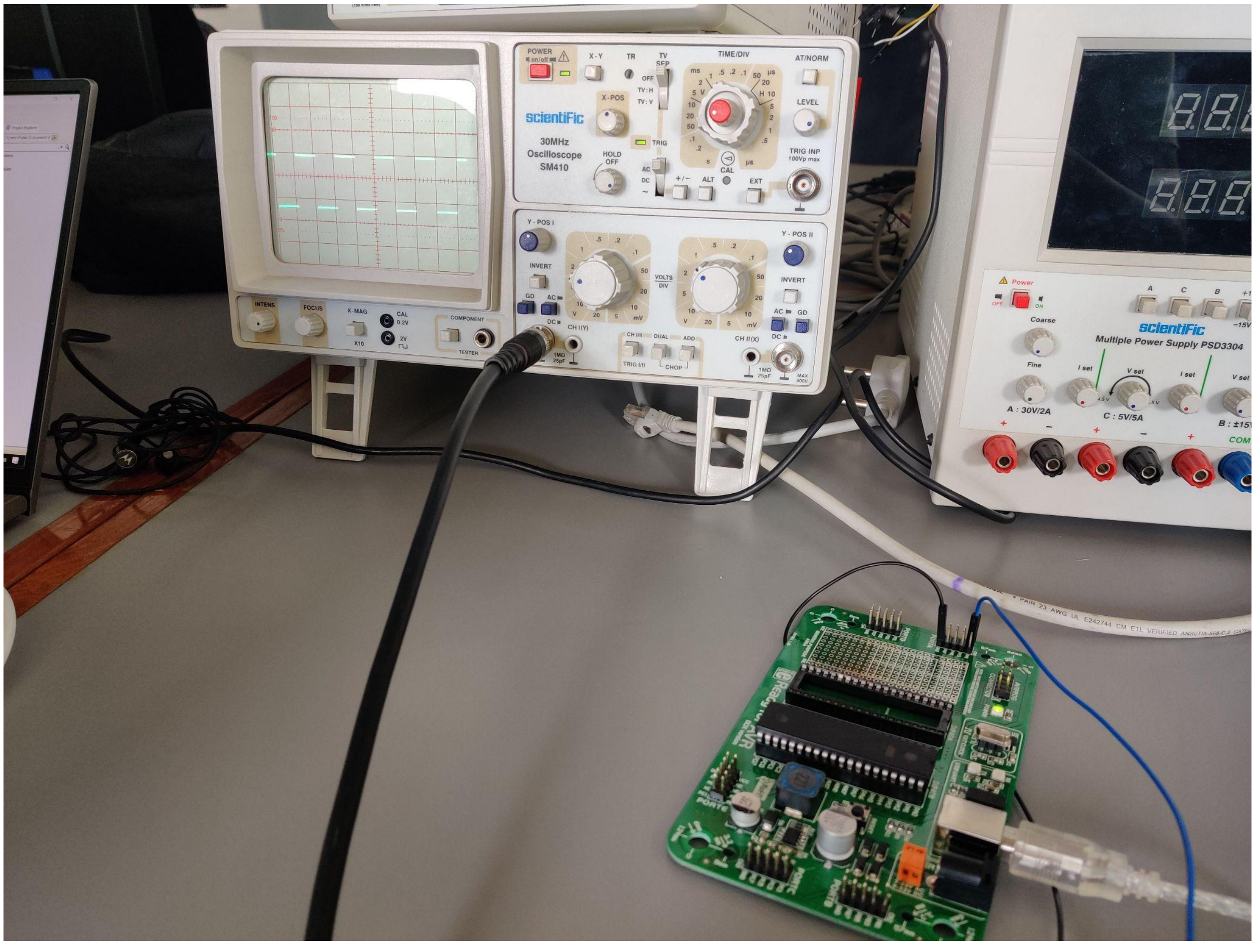
Link

Code

```
void T0Delay() {
     TCNT0 = 0x00;
     OCR0 = 150;
     TCCR0 = 0x09;
     while((TIFR&0x02) == 0);
     TIFR=0x02;
}
void main() {
    DDRA = 0x01;
    while(1)
    {
       PORTA = 0x01;
       T0Delay();
       PORTA = 0x00;
       T0Delay();
    }
}
```

Experiment 3a

Experiment Set-up



Description

In this experiment, we had to use a timer0 circuit with an appropriate prescaler. Creating a square wave of frequency 1Khz lies beyond the range of no prescaler, thus we had to use CLK/64 prescaler. When the program was loaded into the microcontroller, it showed the result stated above. As seen in the image above, the delay for half part of the square wave is 0.5ms and thus the delay for the whole wave becomes 1ms which is the required delay.

Video

Link

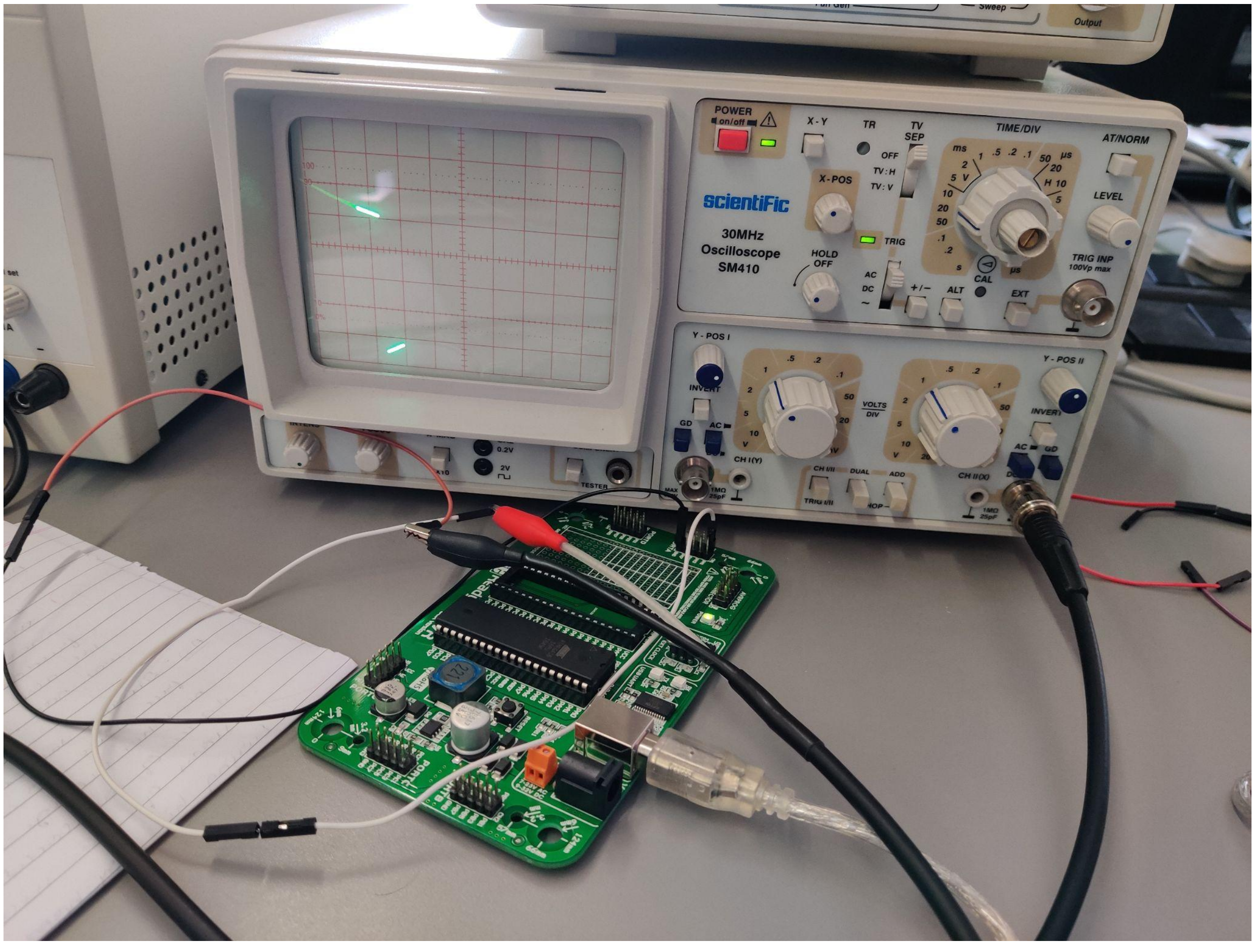
Code

```
void T0Delay()
{
 TCCR0 = 0x03;
 TCNT0 = 194;
 while((TIFR & 0x01) == 0);
 TIFR = 0x01;
 TCCR0 = 0;
}

void main() {
   DDRA = 0x01;
   while(1)
   {
    PORTA ^= 0x01;
    T0Delay();
   }
}
```

Experiment 3b

Experiment Set-up



Description

Here, we are asked to generate a square wave of 10Hz frequency which implies a delay of 0.1 seconds to the whole wave and 0.05 seconds to the half-wave. For this, even if we used the highest prescaler of CLK/1024, it was not possible to obtain the required time delay. Thus, we had to use multiple time delays. So we obtained the time delay of 0.025 seconds using CLK/1024 prescaler and applied it 2 times making the overall delay for half-wave to be 0.05 seconds. Even though the image is not clear we can see that the delay for half-wave is actually 0.5 seconds.

Video

Link

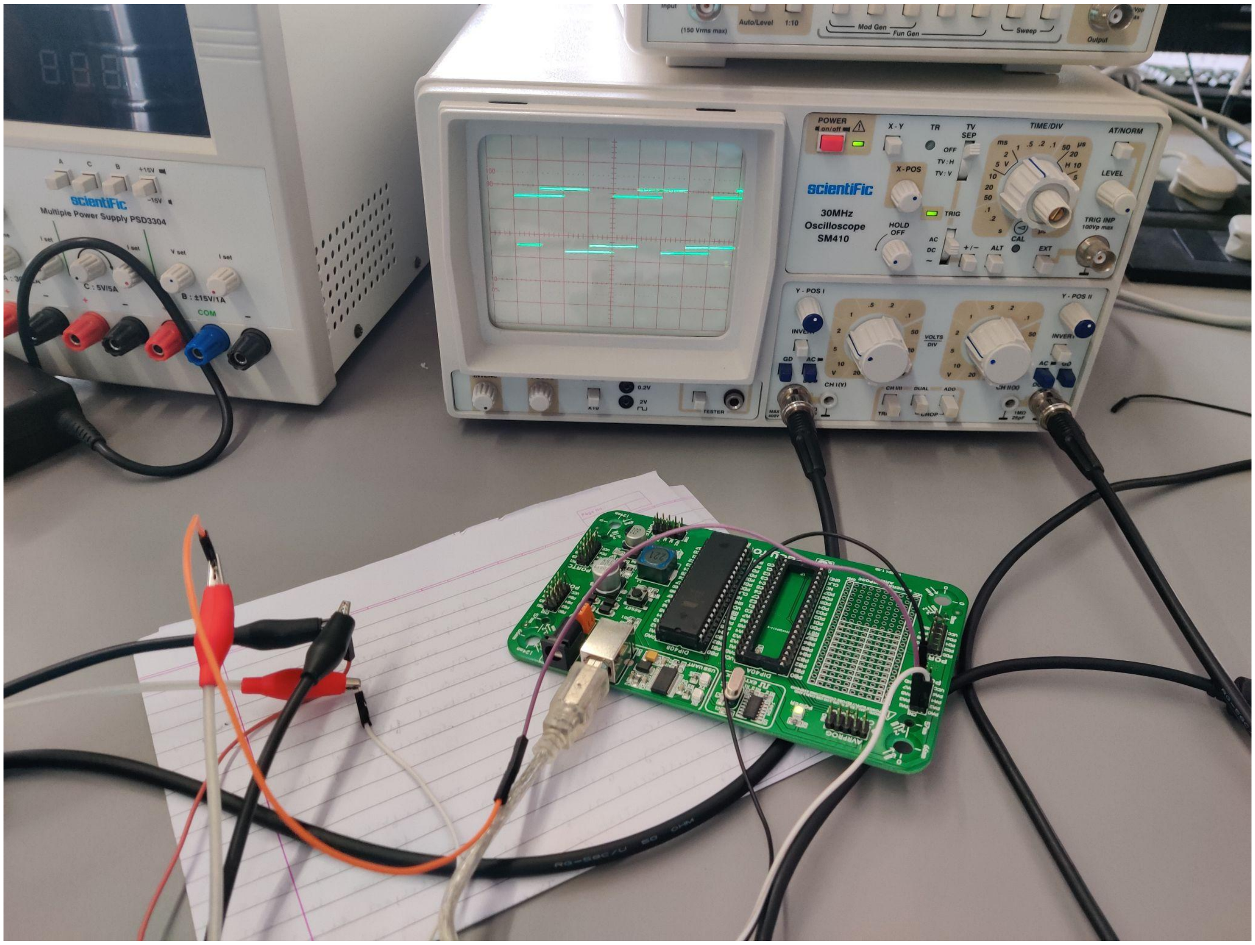
Code

```
void T0Delay()
{
 TCCR0 = 0x05;
 TCNT0 = 0x3D;
 while((TIFR & 0x01) == 0);
 TIFR = 0x01;
 TCCR0=0x00;
}

void main() {
   DDRA = 0x01;
   while(1)
   {
    PORTA ^= 0x01;
    T0Delay();
    T0Delay();
   }
}
```

Experiment 4

Experiment Set-Up



Description

The question required to generate two square waves of frequency 1Khz that are shifted by 90 degrees. For this, we added a delay before toggling the PORTA.b0. Thus, it gave a proper 90 degrees shift. The time delay function remains the same as that in experiment 3a. The connections are quite simple with a modification that we need two CRO cables and operate the CRO in dual mode to get the desired result. As it can be seen in the image above, the 90 degrees phase shifting is clear.

Video

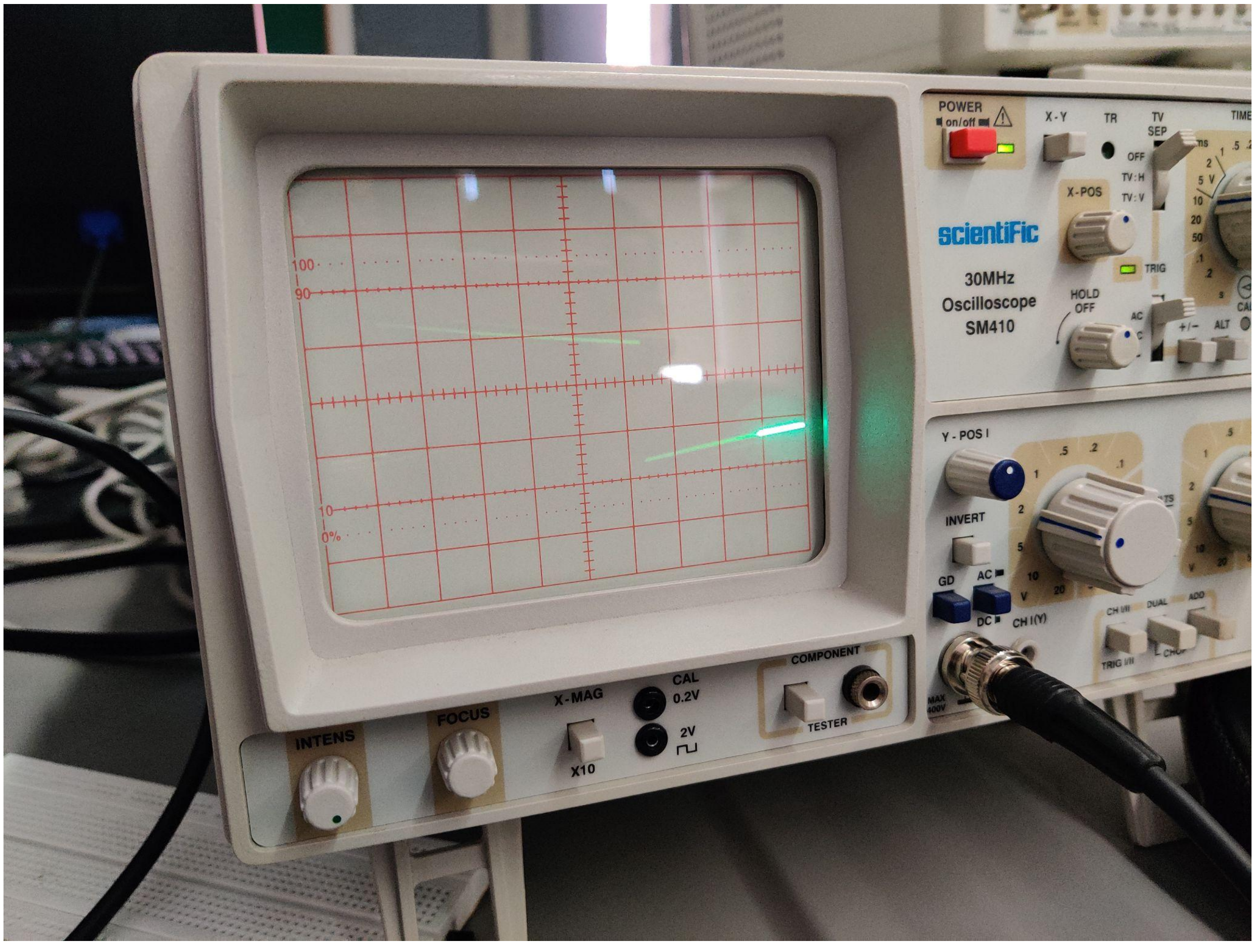
Link

Code

```
void T0delay()
{
 TCNT0 = 194;
 TCCR0 = 0x03;
 while((TIFR & 0x01) == 0);
 TCCR0 = 0;
 TIFR = 0x01;
}
void main() {
    DDRA = 0x03;
    while(1)
    {
     T0Delay();
     PORTA ^= 0x02;
     T0Delay();
     PORTA ^= 0x01;
    }
}
```

Experiment 5

Experiment Set-Up



Description

For the above experiment, we just had to replicate the results that we obtained in question 3 simultaneously at two different port pins. Thus we used PORTA.b0 and PORTA.b1. Furthermore, it was required to use an interrupt because two delays are two be applied using timer0 and timer2. Thus, timer0 is used in interrupt mode and timer2 is used in regular mode. Thus, we obtain the above result which is not clear because the time delay is very large.

Video

[Link]

Code

```
void T2Delay() {
 TCNT2 = 61;
 TCCR2 = 0x07;

 while((TIFR & 0x40) == 0);
 TIFR = 0x40;
 TCCR2 = 0x00;
}

void main() {
    DDRA = 0x03;

    TCCR0 = 0x05;
    TCNT0 = 253;

    TIMSK = 0x01; // Local Interrupt timer0
    SREG.b7 = 1; // Global Interrupt
    while(1)
    {
     PORTA ^= 0x01;
     T2Delay();
     T2Delay();
    }
}

void timer0_ovf_isr(void) org 0x012
{
 PORTA ^= 0x02;
 TCNT0=253;
}
```