

ECE 376 - Homework #9

Timer 0/1/2/3 Interrupts. Due Monday, April 4th

1) Write a C routine using Timer0 interrupts to measure time to 100ns. Using this routine, determine how long the following operations in C take:

a) LCD display routine

```
long int A;  
A = 123456789;  
LCD_Out(A, 10, 7); // time to execute this instruction
```

time = 6.2543ms



b) The time it takes you to press all buttons on PORTB sequentially

```
TRISB = 0xFF;  
while(!RB0); // start  
while(!RB1);  
while(!RB2);  
while(!RB3);  
while(!RB4);  
while(!RB5);  
while(!RB6);  
while(!RB7); // end
```

after three tries, time = 2.3142262 seconds



c) The time it takes you to press and release RB0 10 times

```
TRISB = 0xFF;
for(i=0; i<10; i++) {          // start
    while(!RB0);
    while(RB0);
}                                // end
```

time = 1.3644655 seconds



note: With Timer0, you can measure time to a completely unreasonable number of decimal places.

2) Write a C routine using Timer0 / Timer1 / Timer2 / Timer3 interrupts to play 4 notes at the same time when you press button RB0 (4-string Violin)

| Output Pin | RC0 | RC1 | RC2 | RC3 |
|----------------|-----------|-----------|------------|-----------|
| Note | A1# | C2# | D2# | F2# |
| Frequency (Hz) | 58.27 Hz | 69.30 Hz | 77.78 Hz | 92.50 Hz |
| Interrupt | Timer0 | Timer1 | Timer2 | Timer3 |
| N | 85,807.44 | 72,150.07 | 64,283.877 | 54,054.05 |
| PS | 2 | 2 | A = 16 | 1 |
| Y | 42,904 | 36,075 | B = 251 | 54,054 |
| | | | C = 16 | |

Interrupt Service Routine

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TMR0 = -42904 + 20;
        RC0 = !RC0;
        TMR0IF = 0;
    }
    if (TMR1IF) {
        TMR1 = -36075 + 20;
        RC1 = !RC1;
        TMR1IF = 0;
    }
    if (TMR2IF) {
        RC2 = !RC2;
        TMR2IF = 0;
    }
    if (TMR3IF) {
        TMR3 = -54054 + 40;
        RC3 = !RC3;
        TMR3IF = 0;
    }
}
```

Interrupt Initialization

```
// set up Timer0 for PS = 2
T0CS = 0;
T0CON = 0x80;
TMR0ON = 1;
TMR0IE = 1;
TMR0IP = 1;
PEIE = 1;
// set up Timer1 for PS = 2
TMR1CS = 0;
T1CON = 0x91;
TMR1ON = 1;
TMR1IE = 1;
TMR1IP = 1;
PEIE = 1;
// set up Timer2 for A = 16, B = 251, C = 16
T2CON = 0x7F;
PR2 = 250;
TMR2ON = 1;
TMR2IE = 1;
TMR2IP = 1;
PEIE = 1;
// set up Timer3 for PS = 1
TMR3CS = 0;
```

```

T3CON = 0x81;
TMR3ON = 1;
TMR3IE = 1;
TMR3IP = 1;
PEIE = 1;
// turn on all interrupts
GIE = 1;

```



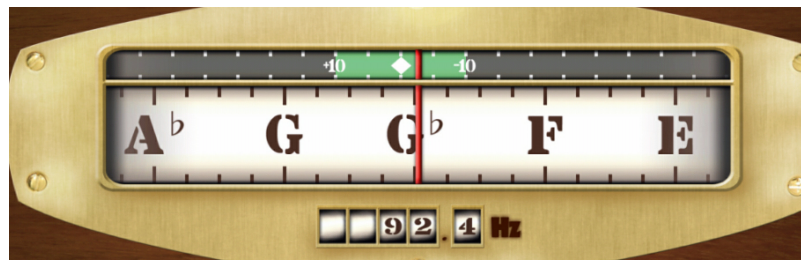
58.27Hz (RC0)



69.30Hz (RC1)



77.78Hz (RC2)



92.50Hz (RC3)

Problem 3-7) Write a C program which uses at least two interrupts Timer0/1/3 interrupts. Some suggestions are

Iambic Paddle

3) Requirements: Explain what the inputs are / what the outputs are / and how they relate. Also explain how each timer interrupt is used in your embedded system.

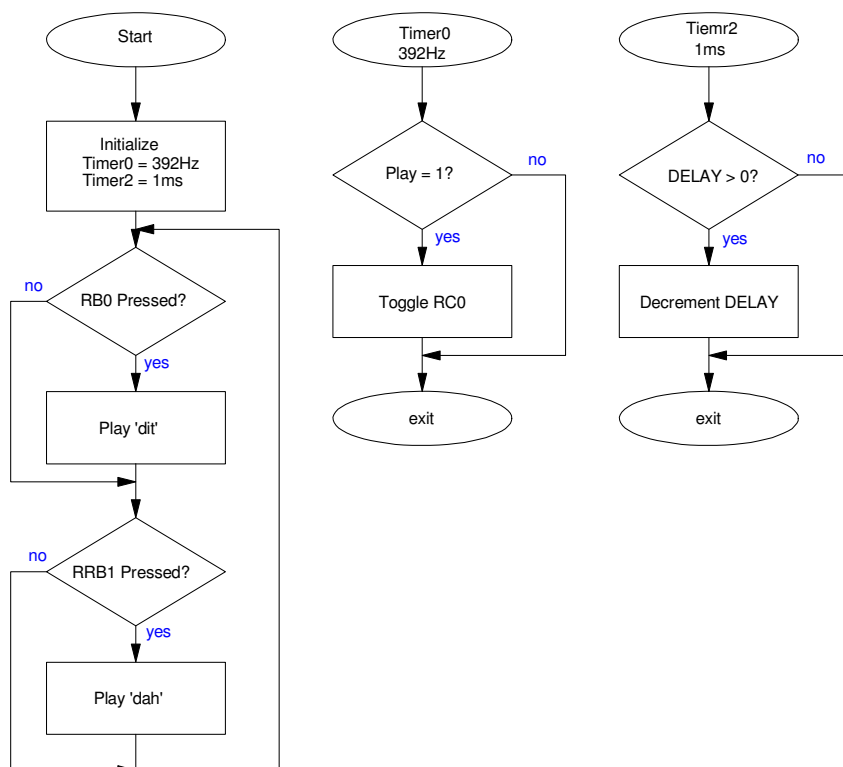
Generate Morse code when you press RB0 and RB1

- RB0: Output a dit: Play 392Hz for 200ms, then pause 200ms
- RB1: Output a dash: Play 392Hz for 600ms, then pause for 200ms
- If you hold down RB0, it plays a series of dits
- If you hold down RB1, it plays a series of dashes
- If you hold both buttons down, it alternates, dit-dah-dit-dah

Timer0: Set the frequency (392Hz)

Timer2: Set the duration (measures time to 1ms)

5) C-Code and flow chart.



Interrupt:

```
// Interrupt Service Routine
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TMR0 = -12705;
        if (PLAY) RC0 = !RC0;
        else RC0 = 0;
        TMR0IF = 0;
    }

    if (TMR2IF) {
        if (DELAY) DELAY -= 1;
        TMR2IF = 0;
    }
}
```

Main Loop:

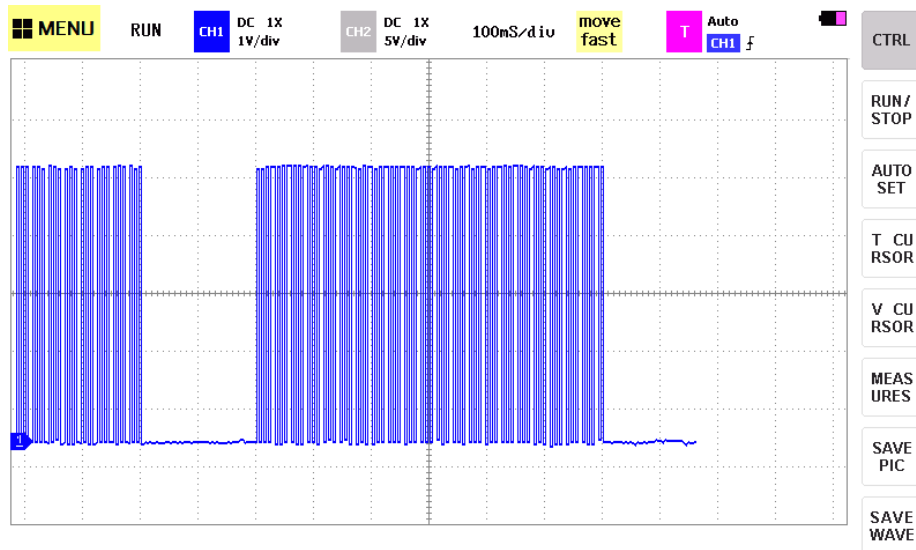
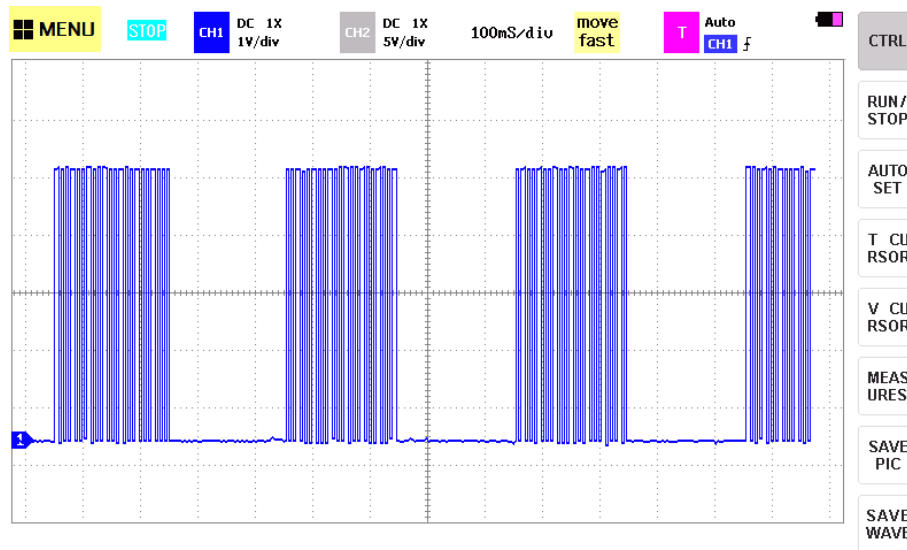
```
while(1) {
    if(RB0) {
        DELAY = 200;
        PLAY = 1;
        while(DELAY);
        DELAY = 200;
        PLAY = 0;
        while(DELAY);
    }
    if(RB1) {
        DELAY = 600;
        PLAY = 1;
        while(DELAY);
        DELAY = 200;
        PLAY = 0;
        while(DELAY);
    }
}
```

6) Test: Collect data in lab to verify that your interrupts are working properly.

- Frequency = 392.9Hz
- 'dit' on time = 200ms, off time = 200ms
- 'dah' on time = 600ms, off time = 200ms
- Timer2 interrupt = 1.00ms

7) Validation: Collect data in lab to verify you met your requirements

- Holding RB0 plays a series of dits
- Holding down RB1 plays a series of dahs
- Holding both down plays dit-dah and repeats



8) Demo (in person during Zoom office hours or in a video)