

# ECE 376 - Homework #8

Timer 0/1/2/3, INT. Due Monday, November 1st

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) Double precision floating point sine

```
#include <math.h>
double A;

A = 1.23456;
A = sin(A);           // time to execute this instruction
```



b) LCD display routine

```
long int A;
A = 3141592653;
LCD_Out(A, 10, 7);    // time to execute this instruction
```



c) The time it takes you to press RB0 ten times (without using INT interrupts)

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



Code: 10 button presses

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TIME = TIME + 0x10000;
        TMR0IF = 0;
    }
}

// Main Routine

while(1) {
    TIME1 = TIME + TMR0;
    for(i=0; i<10; i++) {
        while(!RB0); while(RB0);
    }
    TIME2 = TIME + TMR0;
    LCD_Move(1,0); LCD_Out(TIME2-TIME1, 10, 7);
}
```

2) Write a C routine using INT0 and Timer0 interrupts to measure time to 100ns. Using this routine, determine how long it takes to press RB0 ten times (ten INT0 interrupts)



Code

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TIME = TIME + 0x10000;
        TMR0IF = 0;
    }
    if(INT0IF) {
        N = (N + 1)%10;
        if(N == 0) {
            TIME1 = TIME2;
            TIME2 = TIME + TMR0;
        }
        INT0IF = 0;
    }
}

:
:
:
while(1) {
    LCD_Move(1,0);  LCD_Out (TIME2-TIME1, 10, 7);
}
```

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

Output Pin	RC0	RC1	RC2	RC3
Note	E2	F2	F#2	G2
Frequency (Hz)	82.41 Hz	87.31 Hz	92.50 Hz	98.00 Hz
Interrupt	Timer0	Timer1	Timer2	Timer3
N	60,672	57,267	54,054	51,020
PS	1	1	A = 16 B = 211.149 C = 16	1



## C-Code - Interrupt service Routine

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TMR0 = -60672 + 40;
        RC0 = !RC0;
        TMR0IF = 0;
    }
    if (TMR1IF) {
        TMR1 = -57267 + 40;
        RC1 = !RC1;
        TMR1IF = 0;
    }
    if (TMR2IF) {
        RC2 = !RC2;
        TMR2IF = 0;
    }
    if (TMR3IF) {
        TMR3 = -51020 + 40;
        RC3 = !RC3;
        TMR3IF = 0;
    }
}
```

## Interrupt Initialization

```
// set up Timer0 for PS = 1
T0CS = 0;
T0CON = 0x88;
TMR0ON = 1;
TMR0IE = 1;
TMR0IP = 1;
PEIE = 1;
// set up Timer1 for PS = 1
TMR1CS = 0;
T1CON = 0x81;
TMR1ON = 1;
TMR1IE = 1;
TMR1IP = 1;
PEIE = 1;

// set up Timer2 for A = 16, B = 211, C = 16
T2CON = 0x7F;
PR2 = 210;
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;
```

## Main Routine (doesn't do anything)

```
while(1);
```

Problem 4-8) Write a C program which uses at least two interrupts. Some suggestions are

### **Iambic Paddle**

4) 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.

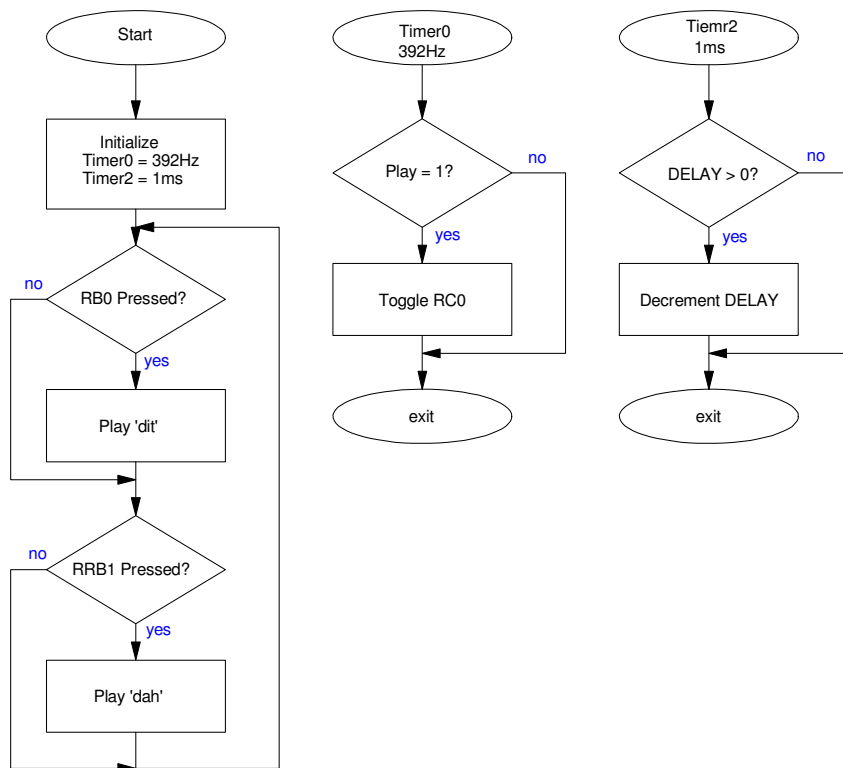
Generate Morse code when you press RB0 and RB1

- 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:

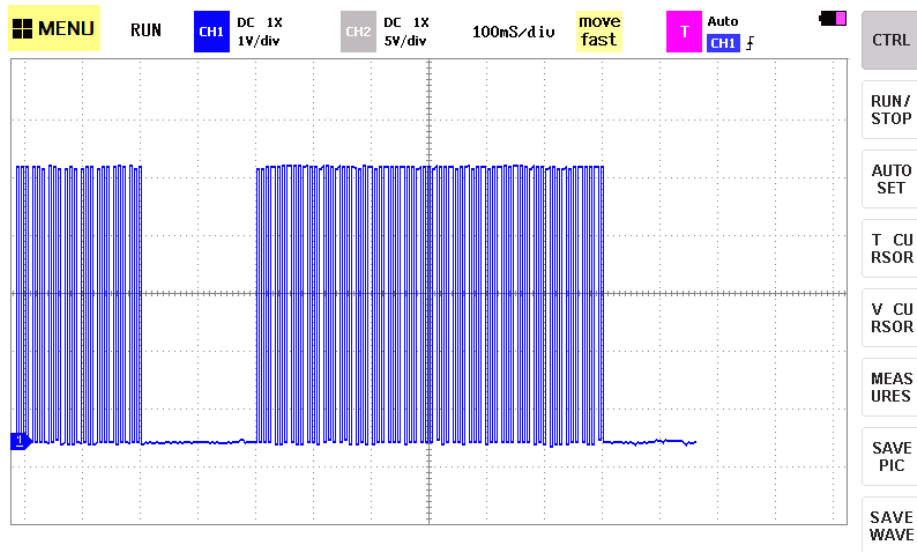
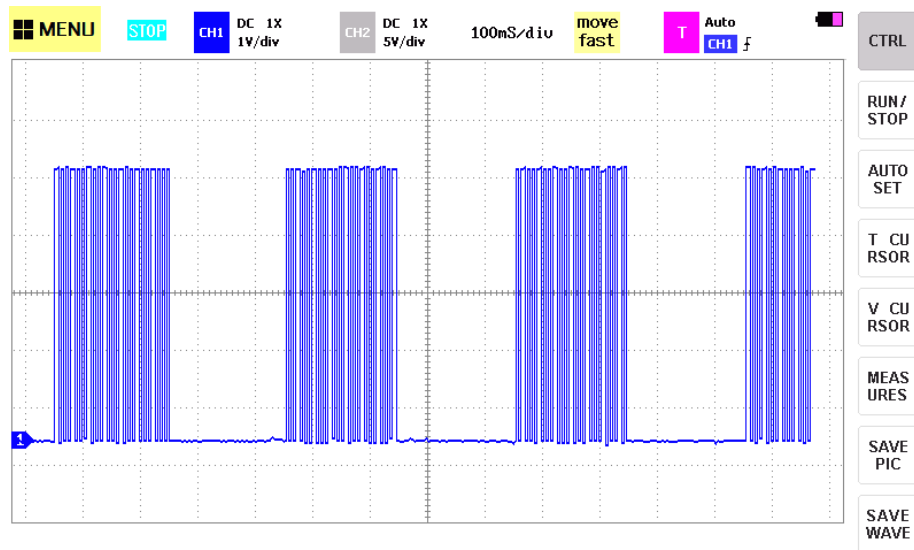
```
while(1) {
    if(RB0) {
        DELAY = 200;
        PLAY = 1;
        while(DELAY);
        DELAY = 200;
        PLAY = 0;
        while(DELAY);
    }
    if(RB1) {
        DELAY = 600;
        PLAY = 1;
        while(DELAY);
        DELAY = 200;
        PLAY = 0;
        while(DELAY);
    }
}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

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)