

ECE 376 - Homework #9

INT, Timer 0/1/2/3 Interrupts - Due Monday, April 7th

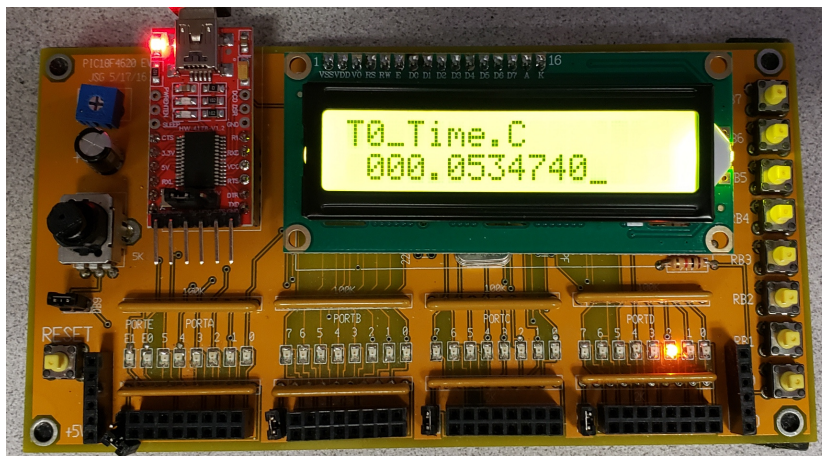
Timer0 Interrupts

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

a) Press and release RB0 one time:

```
TRISB = 0xFF;  
  
while(!RB0);  
// start  
while(RB0);  
// end
```

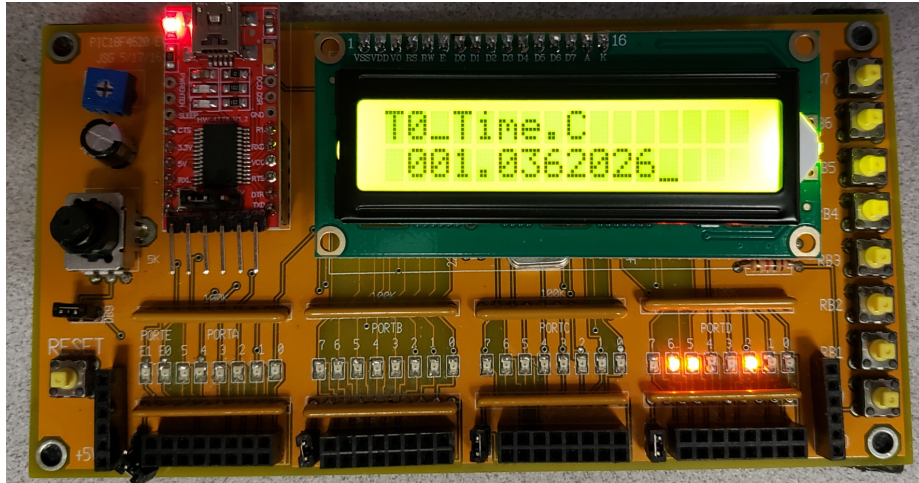
Results: {0.0555712, 0.0459578, 0.0534740}



b) Input code 1234

```
// start
while(!RB1); while(!RB1);
while(!RB2); while(!RB2);
while(!RB3); while(!RB3);
while(!RB4); while(!RB4);
// end
```

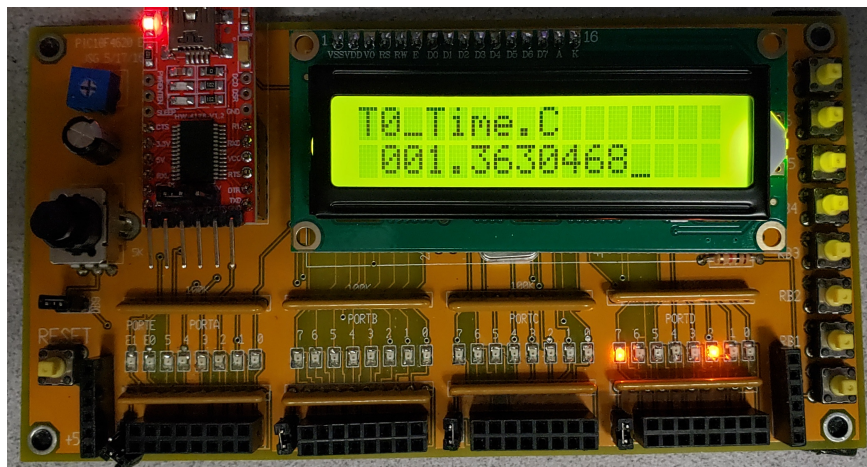
Time = 1.0362026 seconds



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

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

Time = 1.3630468 seconds



note: You can measure time to an absurd number of decimal places. That's one thing that computers are *really* good at: measuring time.

Timer 0/1/2/3 interrupts

2) Write a program which uses interrupts to play four notes at the same time on PORTC

- Output Note Hz
- RC0 D#3 155.563Hz
- RC1 F#3 184.997Hz
- RC2 G#3 207.652Hz
- RC3 A#3 233.082Hz

Give the resulting C code and compiled code size

First, compute N and set up the interrupts

Interrupt	Timer0	Timer1	Timer2	Timer3
Note	155.563 Hz	184.997 Hz	207.652 Hz	233.082 Hz
N	32,141.319	27,027.465	24,078.979	21,451.678
Setup	PS = 1 TMR0 = -32,141	PS = 1 TMR1 = -27,027	A = 7, B = 215 C = 16	PS = 1 TMR3 = -21,451

For Timer2

- T2CON =
- PR2 = 221

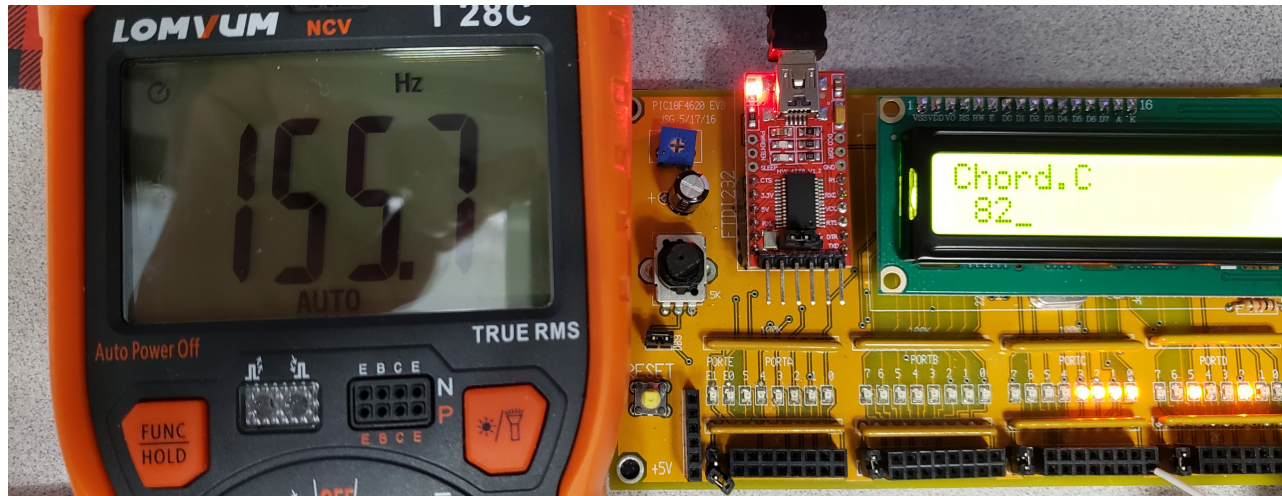
T2CON = 0x37							
7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	1
A = 7					C = 16		

Interrupt Service Routine

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        TMR0 = -32141 + 40;
        RC0 = !RC0;
        TMR0IF = 0;
    }
    if (TMR1IF) {
        TMR1 = -27027 + 45;
        RC1 = !RC1;
        TMR1IF = 0;
    }
    if (TMR2IF) {
        RC2 = !RC2;
        TMR2IF = 0;
    }
    if (TMR3IF) {
        TMR3 = -21451 + 55;
        RC3 = !RC3;
        TMR3IF = 0;
    }
}
```

3) Measure the actual frequencies produced by your program

Interrupt	Timer0	Timer1	Timer2	Timer3
Note	155.563 Hz	184.997 Hz	207.652 Hz	233.082 Hz
Measured Freq	155.7 Hz	185.2 Hz	207.9 Hz	233.4 Hz
Error	+0.088%	+0.110%	+0.119%	+0.136%



Hungry Hungry Hippo!

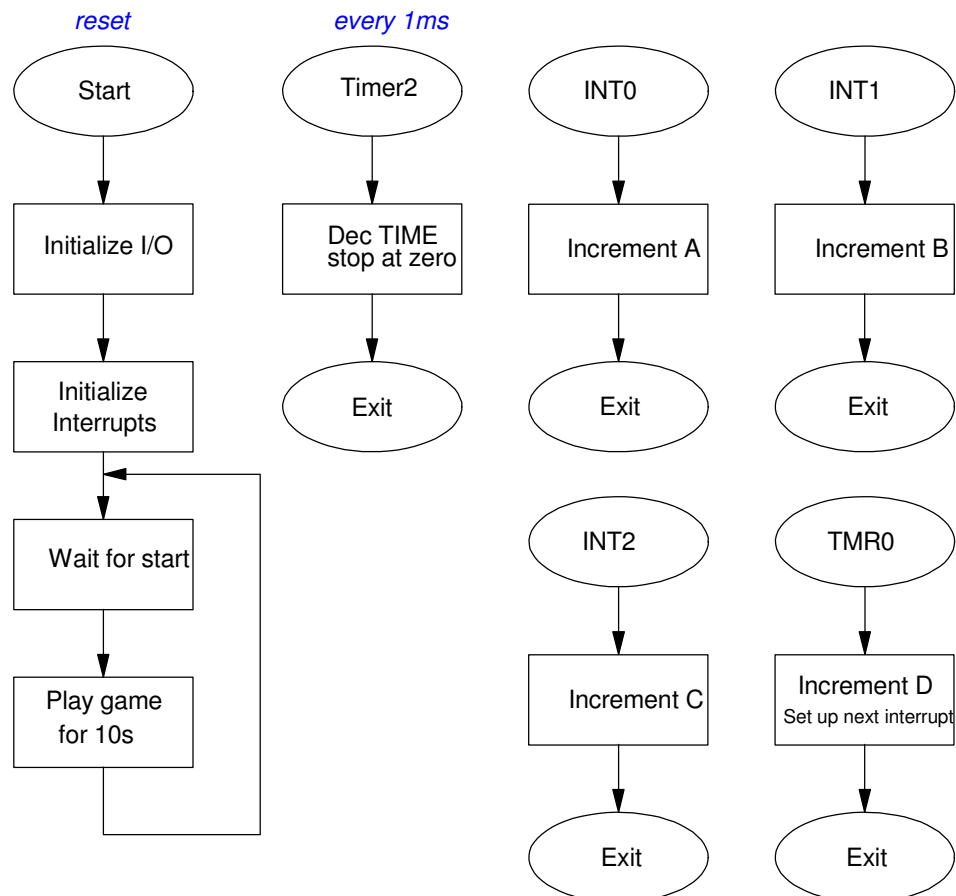
Write C a routine which has a 4-player game of Hungry-Hungry Hippo:

- The main routine constantly updates the score of the four players (A, B, C, D) and the time remaining (0.000 to 10.000 seconds)
- Interrupts update the scores of the four players
- Interrupts update the time remaining in the game with a resolution of 1ms (0.001 second)
- Pressing RB7 starts the game (scores reset to zero, time reset to 10.000 seconds)

4) Specify which interrupts you are going to use and the function of each interrupt.

- INT0: Count player A
- INT1: Count player B
- INT2: Count player C
- TMR0: Count player D
- TMR2: Keep track of time to 1ms

5) Give the flow chart for your program



6) Give the C code and resulting compiled code size

Memory Summary:

Program space	used	D22h (3362)	of 10000h bytes	(5.1%)
Data space	used	3Bh (59)	of F80h bytes	(1.5%)
EEPROM space	used	0h (0)	of 400h bytes	(0.0%)
ID Location space	used	0h (0)	of 8h nibbles	(0.0%)
Configuration bits	used	0h (0)	of 7h words	(0.0%)

Interrupt Service Routines

```
void interrupt IntServe(void)
{
    if (TMR0IF) {
        if (TIME) D += 1;
        TMR0 = -1;
        TMR0IF = 0;
    }
    if (TMR2IF) {
        if (TIME) TIME -= 1;
        TMR2IF = 0;
    }
    if (INT0IF) {
        if (TIME) A += 1;
        INT0IF = 0;
    }
    if (INT1IF) {
        if (TIME) B += 1;
        INT1IF = 0;
    }
    if (INT2IF) {
        if (TIME) C += 1;
        INT2IF = 0;
    }
}
```

Initialization

```
// set up Timer0 for PS = 1
TRISA4 = 1;
T0CON = 0x88;
T0CS = 1;
TMR0ON = 1;
TMR0IE = 1;
TMR0IP = 1;
PEIE = 1;
// set up Timer2 for 1ms
T2CON = 0x4D;
PR2 = 249;
TMR2ON = 1;
TMR2IE = 1;
TMR2IP = 1;
PEIE = 1;
// set up INT0 for rising edges
INT0IE = 0;
INTEDG0 = 1;
// set up INT1 for rising edges
INT1IE = 0;
INTEDG1 = 1;
// set up INT2 for rising edges
INT2IE = 0;
INTEDG2 = 1;
```

```
// turn on all interrupts
GIE = 1;
```

Main Loop

```
while(1) {
    while(!RB0);
    TIME = 10000;
    A = B = C = D = 0;
    TMR0 = -1;
    while(TIME) {
        LCD_Move(1,0); LCD_Out(A, 2, 0);
        LCD_Move(1,3); LCD_Out(B, 2, 0);
        LCD_Move(1,6); LCD_Out(C, 2, 0);
        LCD_Move(1,9); LCD_Out(D, 2, 0);
        LCD_Move(1,12); LCD_Out(TIME/100, 2, 1);
    }
    Wait_ms(1000);
}
```

7) Validate your program

- Reset sets the scores to zero and time to 10.000 seconds
- Time is devremented using interrupts every 1ms
- Each player's score is incremented when his/her button is pressed (edge interrupts)

Results

- Code starts on RB0 press
- Time starts out at 10 seconds
- Time goes to 0.0 in ten seconds (1ms per interrupt)
- Each time you press RB0, A's score increases
- Each time you press RB1, B's score increases
- Each time you press RB2, C's score increases
- Each time you press RB3 (shorted to RA4)., D's score increases
- After 10.0 seconds, button presses are ignored

8) Demo

- In-person or with a video

