

## Examples of Timer2 Interrupts:

Once you can keep track of time, there's lots of things you can do. A short list is:

- Build a better wait routine
- Build a stopwatch that's accurate to 0.001 second (N=10,000)
- Generate musical notes by outputting a square wave of 440Hz to 783Hz (A5 to G5)
- Send data to the PC at 9600 baud

### Better Wait Routine:

Objective:

Write a routine, Wait(unsigned int X), which waits X milliseconds and then returns.

Solution:

1) Set up Timer2 for 1ms (a nice round number)

a) Compute how many clock ticks equals 1ms

- $1\text{ms} * (10,000,000 \text{ clock} / \text{second}) = 10,000 \text{ clocks}$

b) Find A, B, C

- $A * B * C = 10,000$

Let C = 4

- $A * B = 2500$

Let A = 10, B = 250

c) Find PR2 and TMR2CON

- $PR2 = 249$  (B = PR2+1)
- $(A3:A2:A1:A0) = 9 = \text{b}1001$  (A = # + 1)
- $(C1:C0) = \text{b}01$
- $TMR2CON = \text{b} 0100 1101 = 0\text{x}4\text{D}$

d) Initialization routine:

```
PR2 = 249;
TMR2CON = 0x4D;
TMR2IE = 1;
PEIE = 1;
```

2) Set up the interrupt service routine to do stuff, such as decrement a number to zero stopping at zero:

```
// Global Variables
unsigned int DELAY;

void interrupt IntServe(void)
{
    if (TMR2IF) {
        RA0 = !RA0;           // heartbeat for testing code
        if (DELAY) DELAY -= 1;
        TMR2IF = 0;
    }
}
```

Every millisecond, DELAY is decremented until it reaches zero. Once zero, it stays zero.

3) Write a subroutine to wait X milliseconds:

```
void Wait(unsigned int X)
{
    DELAY = X;
    while(DELAY);
}
```

Example: Build a binary clock where PORTC counts in seconds.

```
#include <pic18.h>

// Global Variables
unsigned int DELAY;

// Subroutine Declarations

void interrupt timer2(void)
{
    RA0 = !RA0;
    if (DELAY) DELAY -= 1;
    TMR2IF = 0;
}

void main(void)
{
    TRISA = 0;
    TRISB = 0;
    TRISC = 0;
    TRISD = 0;
    ADCON1 = 15;

    // initialize Timer2

    T2CON = 0x4D;
    PR2 = 249;
    TMR2IE = 1;
    PEIE = 1;
```

```

TMR2ON = 1;
TMR2IP = 1;

// Turn on all interrupts

GIE = 1;

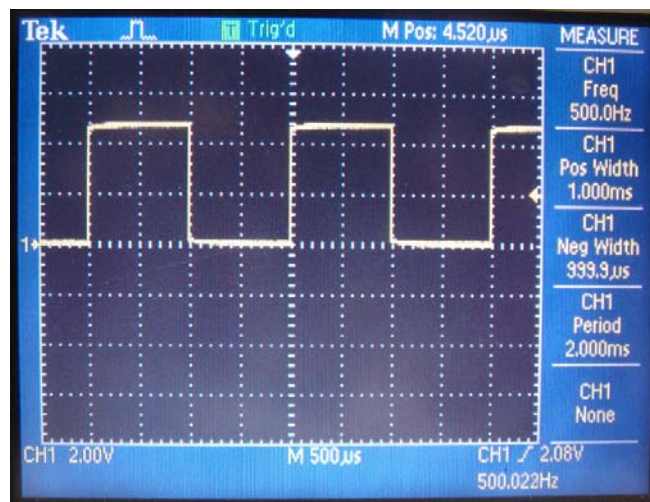
PORTC = 0;

while(1) {
    while(Delay);
    Delay = 1000;

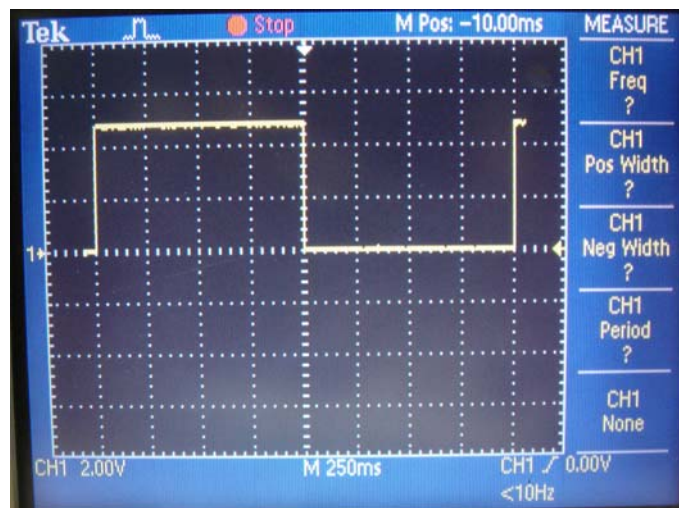
    PORTC = PORTC + 1;
}

```

Pin RA0 toggles every Timer2 interrupt: this lets you check that the interrupts are running correctly:



RA0 toggles every Timer2 interrupt: Timer2 is set up for 1.000 ms



RC0 counts every 1.000 second (1000 interrupts)

## Bulid a Stopwatch, accurate to 0.001 second

Use the same initialization as before to interrupt every 1ms. Each time you interrupt, check the push buttons:

- If RB0 is pressed, start the stopwatch.
- If RB1 is pressed, stop the stopwatch.
- If RB2 is pressed, clear the time.

A global variable, TIME, is used to keep track of the current time (1 count = 1ms).

```
#include <pic18.h>

// Global Variables

unsigned int TIME;
unsigned char RUN;

// Subroutines

#include "LCD_PortD.C"

void interrupt IntServe(void)
{
    if (TMR2IF) {
        RA0 = !RA0;           // Allows you to measure the interrupt time
        if (RB0) RUN = 1;
        if (RB1) RUN = 0;
        if (RB2) TIME = 0;
        if (RUN) TIME += 1;
        TMR2IF = 0;
    }
}

// LCD output time

void LCD_Out(unsigned int DATA, unsigned int N)
{
    unsigned char A[5], i;
    for (i=0; i<5; i++) {
        A[i] = DATA % 10;
        DATA = DATA / 10;
    }
    for (i=5; i>0; i--) {
        if (i == N) LCD_Write('.');
        LCD_Write(A[i-1] + 48);
    }
}

// main routine

void main(void)
{
    TRISA = 0;
    TRISB = 0xFF;
    TRISC = 0;
    TRISD = 0;
    ADCON1 = 15;
    LCD_Init();
}
```

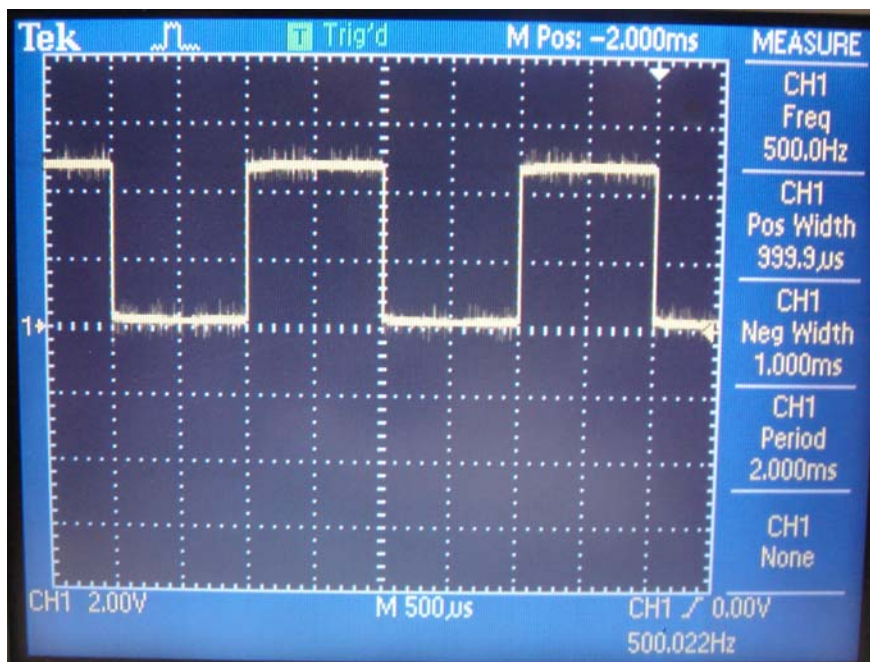
```
// initialize Timer2 for 1ms (10,000 clocks)
PR2 = 249;
T2CON = 0x4D;
TMR2IE = 1;
PEIE = 1;
TMR2IP = 1;

TIME = 0;
RUN = 0;

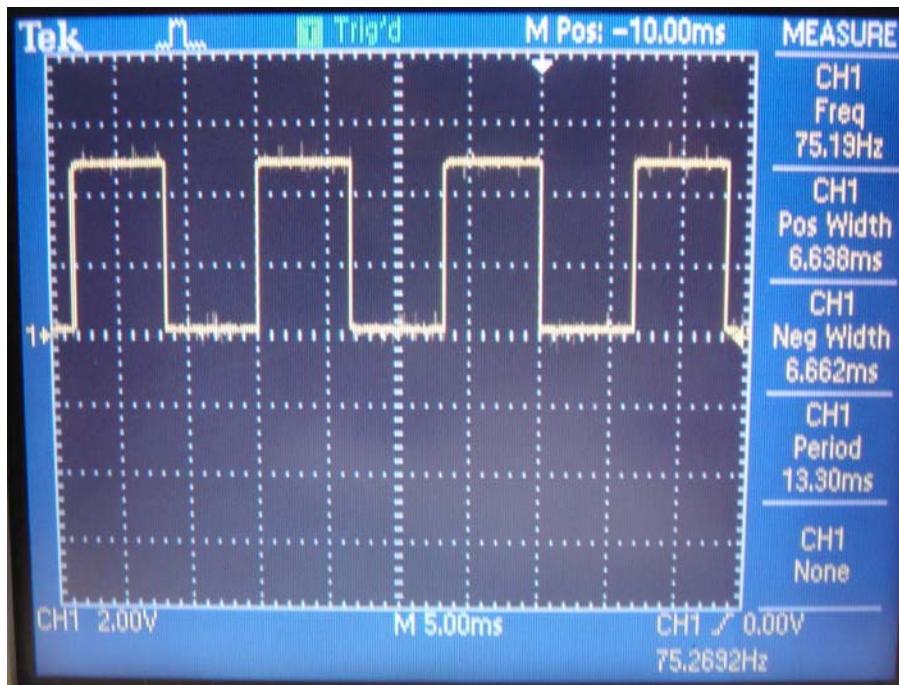
GIE = 1;

while(1) {
    RA1 = !RA1;           // allows you to measure the main loop
    LCD_Move(1,0);
    LCD_Out(TIME, 3);
}
}
```

Note that the main routine only displays whatever the current time is. The interrupts service routine does all the work.



Signal on RA0: Timer2 is being called every 1.000ms (the stopwatch resolution)



Signal on RA1: The main routine loops every 6.38ms (doesn't tell you much other than that's how often the LCD display is updated)



LCD Display of Time, Accurate to 1.000ms (one interrupt)

**Build a piano to play musical notes:**

Note	A4	B4	C5	D5	E5	F5	G5	A5
Hz	440	493.88	523.25	587.33	659.25	698.46	783.99	880
N	11,363.64	10,123.92	9,555.66	8,513.1	7,584.38	7,158.61	6,377.63	5,681.82
A	12	12	12	12	12	12	12	12
B	236.74	210.91	199.08	177.36	158.01	149.14	132.87	118.37
C	4	4	4	4	4	4	4	4

```

// Global Variables
const unsigned int A4 = 236;
const unsigned int B4 = 210;
const unsigned int C5 = 198;
const unsigned int D5 = 176;
const unsigned int E5 = 157;
const unsigned int F5 = 148;
const unsigned int G5 = 132;
const unsigned int A5 = 117;

// Subroutine Declarations
#include <pic18.h>
#include "lcd_portd.c"

void interrupt IntServe(void)
{
    if (TMR2IF) {
        if(PORTB) RA1 = !RA1;
        else RA1 = 0;
        TMR2IF = 0;
    }
}

void Play(unsigned char NOTE)
{
    PR2 = NOTE;
    TMR2ON = 1;
    Wait_ms(450);
    TMR2ON = 0;
    Wait_ms(50);
}

// Main Routine
void main(void)
{
    unsigned char i, j;
    TRISA = 0;
    TRISB = 0xFF;
    TRISC = 0;
    TRISD = 0;
    TRISE = 0;
    ADCON1 = 15;

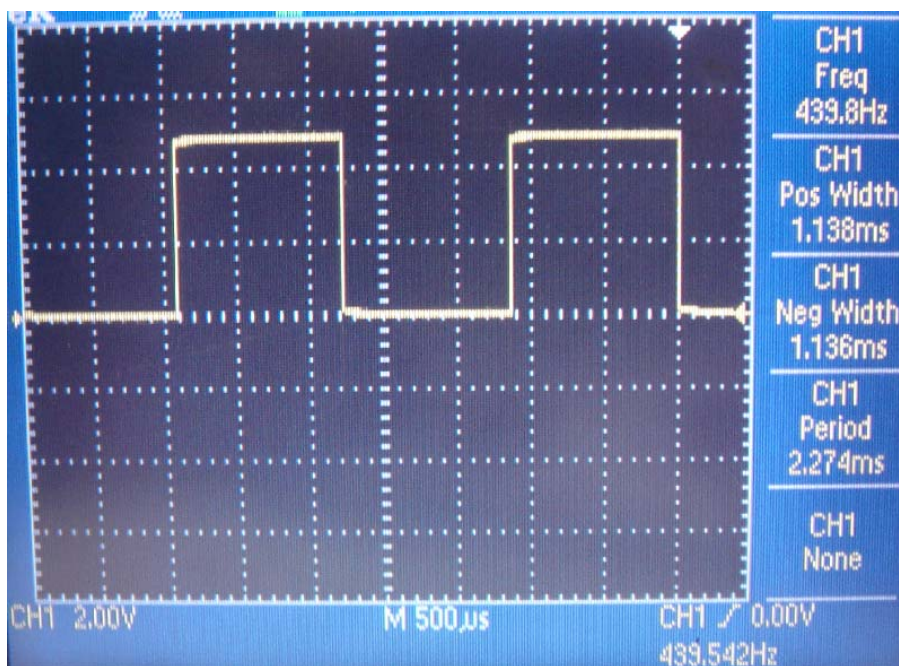
    // Timer2 Initialize
    TMR2ON = 1;
    TMR2IE = 1;
    PEIE = 1;
    T2CON = 0x5D; // A=12, C=4 0 1011 1 01

```

```
PR2 = 49;  
GIE = 1;  
  
while(1) {  
  if (RB0) PR2 = A4;  
  if (RB1) PR2 = B4;  
  if (RB2) PR2 = C5;  
  if (RB3) PR2 = D5;  
  if (RB4) PR2 = E5;  
  if (RB5) PR2 = F5;  
  if (RB6) PR2 = G5;  
  if (RB7) PR2 = A5;  
};  
}
```

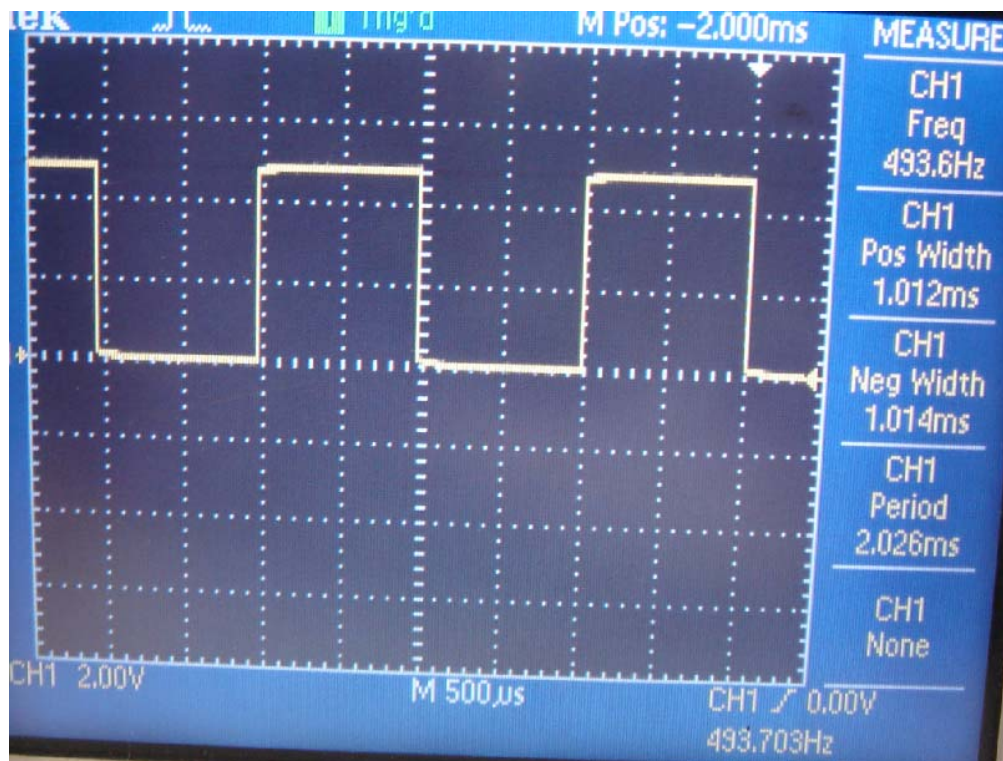
By changing PR2, you change the rate at which the Timer2 interrupt is called. This lets you change the frequency of the square wave output of RA0, creating different notes.

To turn the note of and off, TMR2ON is used to turn on and off TIMER2. When it's off, TIMER2 quits playing. You could also switch RA0 to input to turn off the output as well.



RB0: Music Note A4 (440Hz)





RB1: Music Note B4: 493.88Hz

### Strobe Light:

Output a pulse

- On for 1ms
- Off for 9ms

Code:

```
// Global Variables
unsigned int DELAY;

// Subroutine Declarations
#include <pic18.h>
#include "lcd_portd.c"

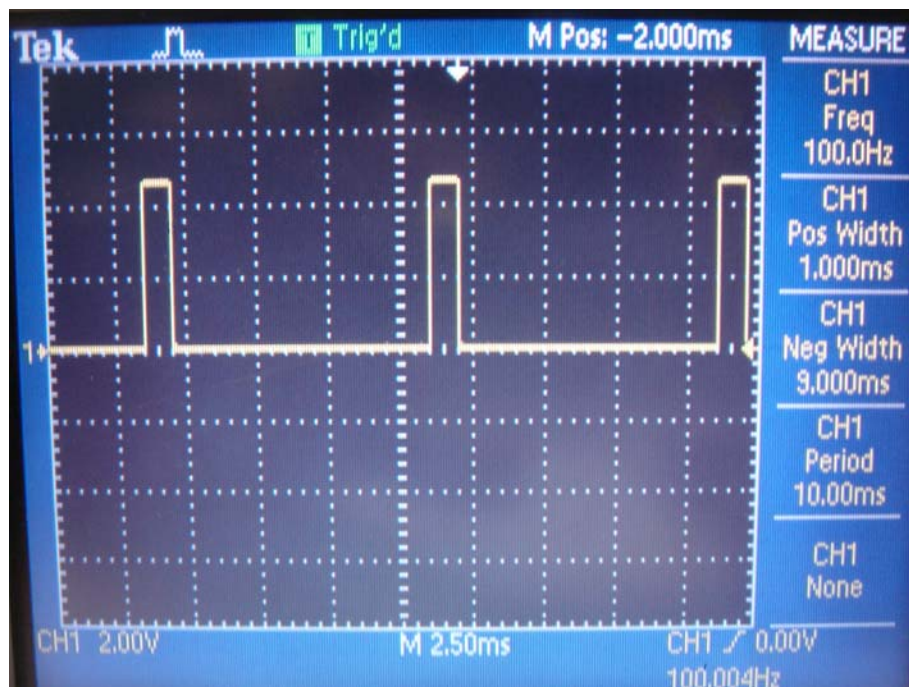
void interrupt IntServe(void)
{
    if (TMR2IF) {
        if (DELAY) DELAY -= 1;
        TMR2IF = 0;
    }
}

// Main Routine
void main(void)
{
```

```
unsigned char i, j;
TRISA = 0;
TRISB = 0;
TRISC = 0;
TRISD = 0;
TRISE = 0;
ADCON1 = 15;

// Timer2 Initialize to 1ms
TMR2ON = 1;
TMR2IE = 1;
PEIE = 1;
T2CON = 0x4D;
PR2 = 249;
GIE = 1;

while(1) {
    PORTC = 0xFF;
    DELAY = 1;
    while(DELAY);
    PORTC = 0x00;
    DELAY = 9;
    while(DELAY);
}
}
```



Strobe Light: RC0 is on for 1ms, off for 9ms