**TIMERS**

**Aayahna M. Herbert**

**October 31, 2017**

**ECE 3720 Section 007**

**Microcontroller Interfacing Laboratory**

**Abstract:** A circuit was designed that sends an output from the microcontroller to the AC-1005G-P speaker. Within the microcontroller, an interrupt was set up for the timer that updates the period and plays music as necessary.

**INTRODUCTION**

The purpose of this lab was to design a circuit that sends an output from the microcontroller to the AC-1005G-P speaker. Within the microcontroller, an interrupt was set up for the timer that updates the period and plays music as necessary. The overall skeleton for the code that contains the notes, their frequencies, and the structure that switches to the next note was given.

**EXPERIMENTAL PROCEDURES**

The equipment used include one PIC32 MC, NI-ELVIS II board, and a AC-1005G-P speaker. The first thing to be done, to avoid forgetting it later, is to set the B ports as outputs. Pin B7 is used as the output from the microcontroller to the speaker. Because we are dealing with one interrupt signal, the registers INT0 should be enabled through Timer A, set to high direction, be interrupted on a high value, and have a priority of one by using TxCON, IECx, IFSx, and IPCx. In addition to the interrupt, it must also be globally enabled by adding INTEnableSystemMultiVectoredInt() and INTEnableInterrupts() to the main function. With the main function comes the ISR functions for the specific register numbers in use, void \_\_ISR(\_TIMER\_2\_VECTOR) interruption(void). Inside of these functions goes the commands for what should happen to the notes as time passes. For this function, we want the count for the time length to increment and PR2 to play the note the while loop has incremented to. As long as the note being played that represented by PR2 is not a rest, the value for pin B7 should invert in order to keep the frequency changing constantly. In terms of wiring, make sure the power the chip is receiving is +5V and the speaker is connected to pin B7 and ground.

**RESULTS**

After the circuit was hooked up and ready to be tested, the program detected no compiling errors when ran so the board could then be tested. While running, the speaker played the correct notes.

**DISCUSSION**

Problems arose initially when the speaker was not playing the correct notes initially. The code and wiring were second-checked by the TA and the wires all came from a new pack recently purchased to avoid the possibility of dead wires being used and affecting the lab results. The source of the error was the incorrect vector was being used in the interruption function. After it was changed to the correct vector, the speaker proceeded to play the right notes.

**CONCLUSIONS**

In general, the take-away of this lab was to learn how to set up and use timers, get a better understanding of how speakers work, and how to use the two together to make music.

**FIGURES AND TABLES**

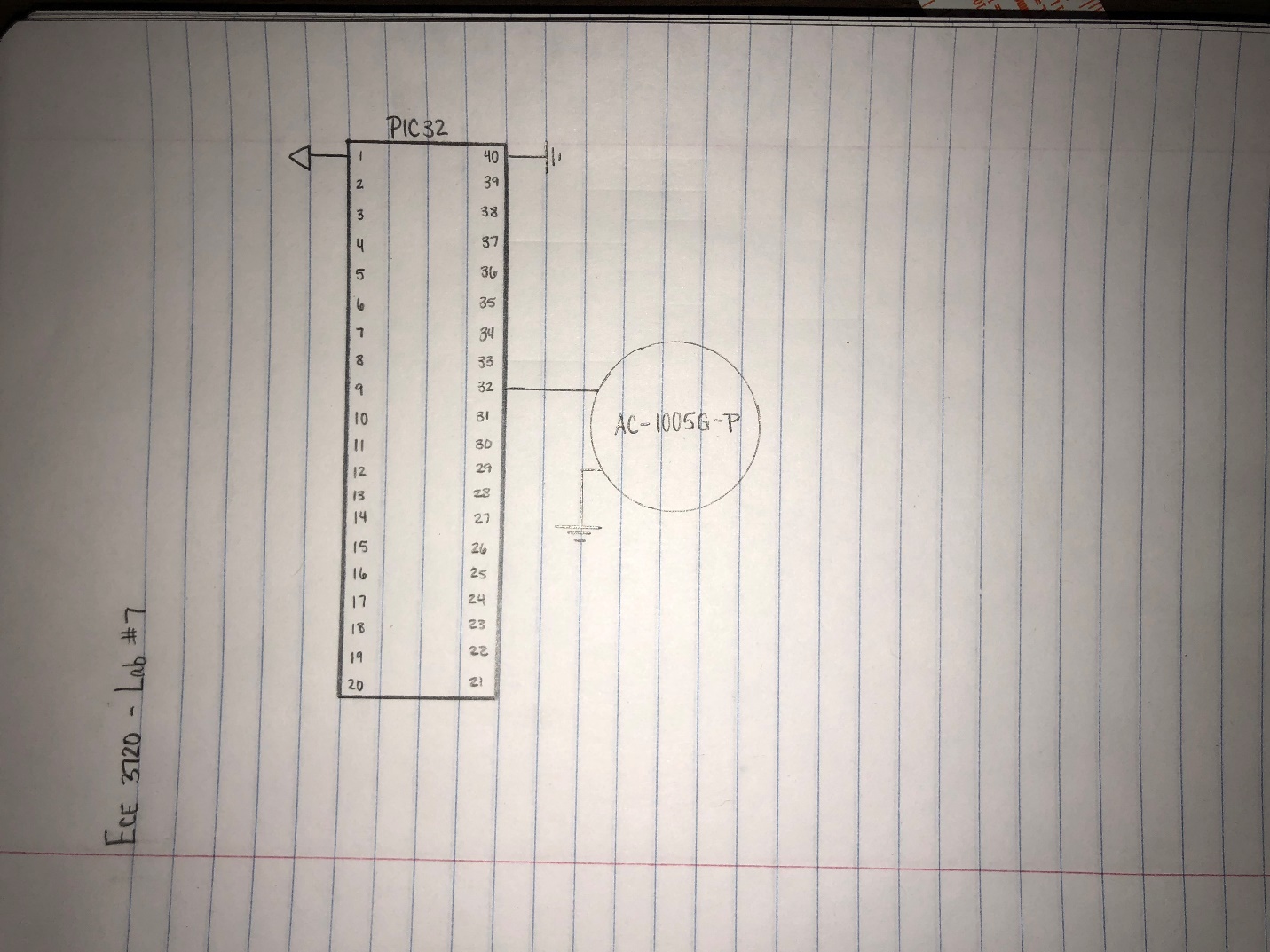


Figure 1: Circuit Schematic

**CODE**

#include<plib.h>

//This is here to make timing calculations easier!

#pragma config FPBDIV = DIV\_2

//r is a Rest. If r, do not toggle! (don't want noise to play)

#define r 3000

#define a 4545

#define b 4050

#define C 3817

#define C\_ 3610

#define D 3402

#define D\_ 3216

#define E 3031

#define F 2866

#define F\_ 2703

#define G 2551

#define G\_ 2410

#define A 2273

#define A\_ 2146

#define B 2025

#define CC 1911

#define q 400

#define qdot q \* 1.5

#define e q/2

#define s e/2

#define t32 s/2

#define sdot s+t32

#define h q\*2

#define hdot q+e

#define edot e+s

#define num\_notes 52

int i,j;

short delay[num\_notes] = {t32,t32,t32,t32,t32,t32,t32,t32,t32,t32,s,sdot,t32,t32,

t32,t32,t32,t32,t32,t32,t32,t32,t32,s,sdot,t32,t32,t32,

t32,t32,t32,t32,t32,t32,t32,t32,s,sdot,t32,t32,t32,t32,

t32,t32,t32,t32,t32,t32,t32,t32,s,e};

short music\_notes[num\_notes] = {b,r,B,r,F\_,r,D\_,r,B,F,r,D,r,C,r,CC,r,G,r,E,r,CC,G,r,

E,r,b,r,B,r,F\_,r,D\_,r,B,F,r,D,r,D,E,D\_,r,F,F\_,G,r,G,G\_,A,r,B};

//Interrupt Function

void \_\_ISR(\_TIMER\_2\_VECTOR) Interruption(void)

{

j++;

PR2 = music\_notes[i];

if(PR2 != r)

LATBbits.LATB7 ^= 1;

IFS0bits.T2IF = 0;

}

main()

{

INTEnableSystemMultiVectoredInt();

INTEnableInterrupts();

//Setup Timers/Interrupts

TRISB = 0;

T2CON = 0;

T2CONbits.TGATE = 0;

T2CONbits.TCS = 0;

T2CONbits.T32 = 0;

T2CONbits.TCKPS = 0;

IEC0bits.T2IE = 1;

IFS0bits.T2IF = 1;

IPC2bits.T2IP = 1;

PR2 = music\_notes[0];

T2CONbits.ON = 1;

LATBbits.LATB7 = 0;

while(1) //Run Continuously

{

for(i=0; i < num\_notes; i++)

{

j = 0;

while(j < delay[i])//for length of time to play note

{

}

}

}

}