**Music Project – CECS 262**

Student’s Name: Len Quach

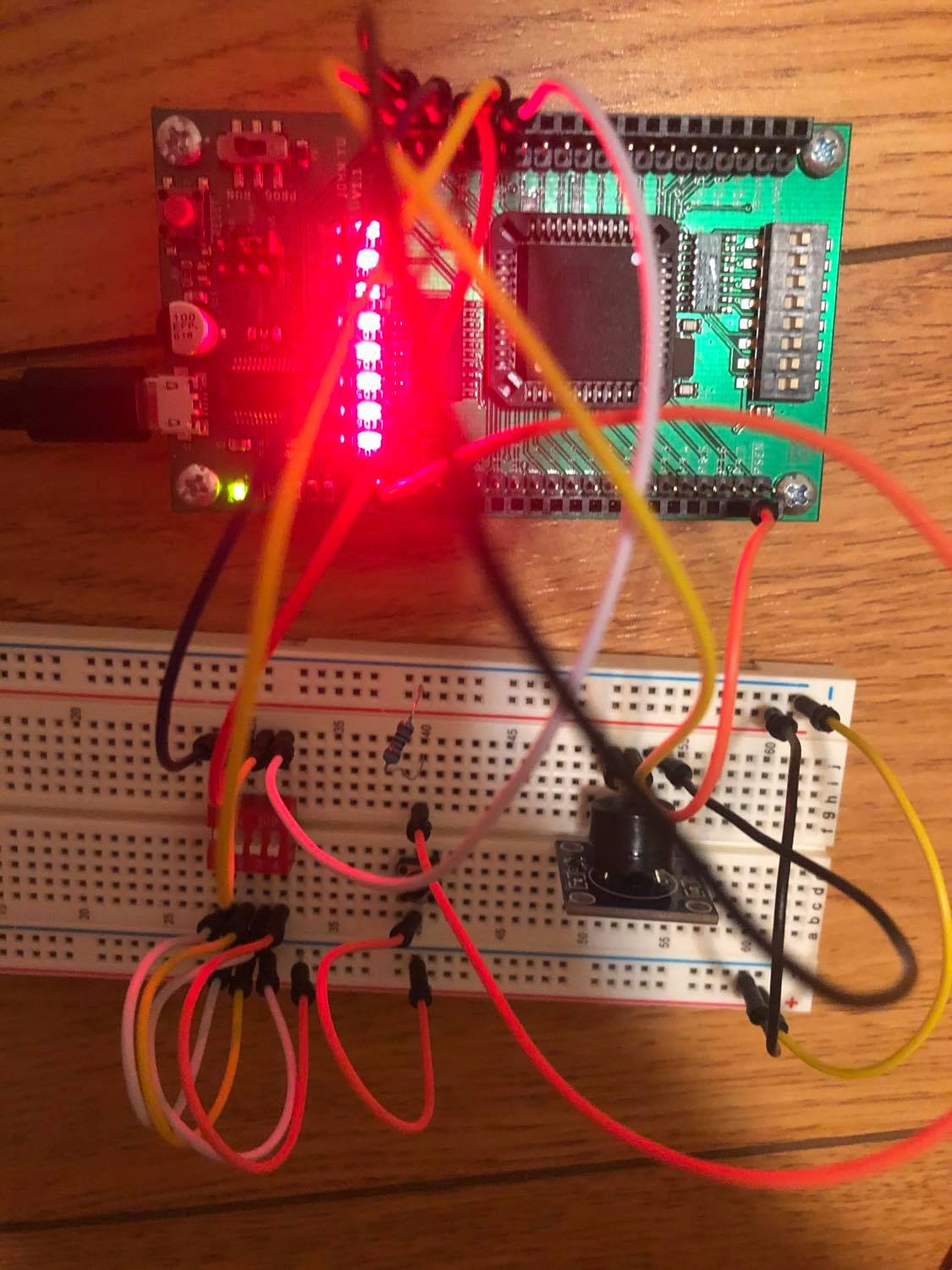
Due Date: 4/24/2020

The project is about how to play simple songs using 8051 timer and external interrupt using a speaker. We design a system with the 8051 board with some components such as a passive buzzer with 4 dip-switches, 1 push button, some resisters as optional and jumper wires.

We will play these four songs including Mary had a little lamb, Twinkle Twinkle little star, Happy Birthday, and Jingle Bell. The dip-switch will control which song to play. If the first pin is on, the first song Mary had a little lamb will be on. If the pin 2, then second song Twinkle Twinkle little star will be played. The third song Happy Birthday will be played as pin 3 on, and Jingle Bell as pin4 is on. However, if multiple pins are on, the pins with the smallest index will take priority to control which song to play.

To play a decent melody, there are three factors are required are note value, note duration and tempo. Tempo is just how fast the melody should be played. We can play any tone by having the note value and note duration table. We produce noise by using the tone function. The note table is given that equates each frequency to a specific musical note type and we use this table to play themes.

We can play the song with the dip switches then press the button to make a change or to generate an external hardware interrupt. So the external change song event will be respond by the system as soon as possible.



Here is the code:

//===============================================================

#include <reg51.h>

#include <intrins.h> //for nop()

#define uchar unsigned char

#define uint unsigned int

#define C4 1+7

#define D4 2+7

#define E4 3+7

#define F4 4+7

#define G4 5+7

#define A4 6+7

#define B4 7+7

#define C5 1+14

//define 4 pins to control which song to play

sbit song1 = P0^0;

sbit song2 = P0^1;

sbit song3 = P0^2;

sbit song4 = P0^3;

sbit buzzer = P3^0;

bit change; //to change song

volatile uchar th0, tl0; //to hold the information of current tone

struct music\_type

{

uchar tone;

uchar delay;

};

uchar th0, tl0; //too hold the initial value of current timer

//intial value table for C scale

uint code tonetab\_C[] = {62018, 62401, 62743, 62903, 63185, 63441, 63670, //c3

63777, 63969, 64140, 64216, 64360, 64489, 64603, //c4

64655, 64751, 64837, 64876, 64948, 65012, 65070}; //c5

//note table for “Mary Had a Little Lamb” song

struct music\_type code maryhadalittlelamb[] = {

E4, 0x04, D4, 0x04, C4, 0x04, D4, 0x04, E4, 0x04, 0, 1 , E4, 0x04, 0, 1, E4, 0x08,

D4, 0x04, 0, 1 , D4, 0x04, 0, 1, D4, 0x08, E4, 0x04, G4, 0x04, 0, 1, G4, 0x08,

E4, 0x04, D4, 0x04, C4, 0x04, D4, 0x04, E4, 0x04, 0 , 1, E4, 0x04, 0, 1, E4, 0x08,

D4, 0x04, 0, 1, D4, 0x04, E4, 0x04, D4, 0x04, C4, 0x012, 0 , 0

};

//note table for “Twinkle Twinkle Little Star” song

struct music\_type code twinkletwinkle[] = {

C4, 0x04, 0, 1, C4, 0x04, G4, 0x04, 0, 1, G4, 0x04, A4, 0x04, 0, 1, A4, 0x04, G4, 0x08,

F4, 0x04, 0, 1, F4, 0x04, E4, 0x04, 0, 1, E4, 0x04, D4, 0x04, 0, 1, D4, 0x04, C4, 0x08,

G4, 0x04, 0, 1, G4, 0x04, F4, 0x04, 0, 1, F4, 0x04, E4, 0x04, 0, 1, E4, 0x04, D4, 0x08,

G4, 0x04, 0, 1, G4, 0x04, F4, 0x04, 0, 1, F4, 0x04, E4, 0x04, 0, 1, E4, 0x04, D4, 0x08,

C4, 0x04, 0, 1, C4, 0x04, G4, 0x04, 0, 1, G4, 0x04, A4, 0x04, 0, 1, A4, 0x04, G4, 0x08,

F4, 0x04, 0, 1, F4, 0x04, E4, 0x04, 0, 1, E4, 0x04, D4, 0x04, 0, 1, D4, 0x04, C4, 0x08,

0, 0

};

//note table for “Happy Birthday” song

struct music\_type code happybirthday[] = {

C4, 0x03, 0, 1, C4, 0x01, D4, 0x04, C4, 0x04, F4, 0x04, E4, 0x08, C4, 0x03, 0, 1,

C4, 0x01, D4, 0x04, C4, 0x04, G4, 0x04, F4, 0x08, C4, 0x03, 0, 1, C4, 0x01,

C5, 0x04, A4, 0x04, F4, 0x04, E4, 0x04, D4, 0x04, B4, 0x03, 0, 1, B4, 0x01,

A4, 0x04, F4, 0x04, G4, 0x04, F4, 0x12, 0, 0

};

//note table for “Jingle Bell”

struct music\_type code jinglebells[] = {

E4, 0x02, 0, 1, E4, 0x02, 0, 1, E4, 0x04, 0, 1, E4, 0x02, 0, 1, E4, 0x02, 0, 1, E4, 0x04, 0, 1,

E4, 0x02, G4, 0x02, C4, 0x02, D4, 0x02, E4, 0x08,

F4, 0x02, 0 , 1, F4, 0x02, 0, 1, F4, 0x04, 0, 1, F4, 0x02, E4, 0x02, 0, 1, E4, 0x04, 0, 1,

E4, 0x02, D4, 0x02, 0, 1, D4, 0x02, E4, 0x02, D4, 0x04, G4, 0x04, 0, 0

};

void delay\_us(uchar n\_usec);

void delay\_ms(uint n\_msec);

void play\_a\_song(struct music\_type notetab[]);

void main(void)

{

TMOD = 0x01; //use timer0 mode 1 to generate tone

IE = 0x83; //enable timer0 overflow interrupt and external interrupt

change = 1;

while (1) {

change = 1;

if (song1)

play\_a\_song(maryhadalittlelamb);

else if (song2)

play\_a\_song(twinkletwinkle);

else if (song3)

play\_a\_song(happybirthday);

else if (song4)

play\_a\_song(jinglebells);

else

TR0 = 0;

}

}

//--------------------------------------------------------------

void play\_a\_song(struct music\_type notetab[])

{

uchar i, j;

while (change)

{

i = 0;

while (notetab[i].delay&&change)

{

if (!notetab[i].tone)

TR0 = 0; // turn off timer0

else

{

th0 = TH0 = tonetab\_C[notetab[i].tone-1] >> 8; //higher 8-bit of the initial value

tl0 = TL0 = tonetab\_C[notetab[i].tone-1]; // lower 8-bit of the initial value

TR0 = 1;

}

//provide duration for each tone

for (j=0; j<notetab[i].delay; j++)

delay\_ms(1000); //minimum delay unit for 1/16

i++;

}

TR0 = 0; //to stop between the songs

delay\_ms(2000);

}

}

//---------------------------------------------------------

void delay\_us(uchar n\_usec)

{

do

{

nop();

nop();

nop();

nop();

nop();

} while (n\_usec);

}

//-------------------------------------------------------------

void delay\_ms(uint n\_msec)

{

do

delay\_us(131);

while (n\_msec);

}

//---------------------------------------------------------------------

void T0\_ISR(void) interrupt 1 //timer0 interrupt service routine

{

TR0 = 0;

buzzer = !buzzer;

TH0 = th0;

TL0 = tl0;

TR0 = 1;

}

//-----------------------------------------------------------------

void change\_song(void) interrupt 0

{

change = 0;

}

//===============================================================

The speaker use electric material to bend a metal diaphragm that makes sound. It changes the shape when we apply electricity to it. By adhering an electric disc to a thin metal plate and then applying electricity, the metal can be bended back and forth which in turn creates the sound. That is how the buzzer makes some sound with the 8051 board.

This music project gave me a lot of interests during working with it, which helped me to learn how to read the music notes and know how a buzzer can be played with songs using 8051 and circuit board.

<https://www.youtube.com/watch?v=0QyKShiU3zU&feature=youtu.be>