## **Laboratory 3: Speakers and Tones**

## *Objectives*

1. Understand the concept of speakers and tones (notes).
2. Practice more complex sketches.

## *Speaker/Music Notes*

Invented by Thomas A. Edison, a speaker is a small device that will convert an electric current into sound. By generating frequency pulses to induce a magnetic field that vibrates a diaphragm, the sound is then created from the resonated diaphragm.

To make a music note, the vibration has to match the frequency of the corresponding note. The standard frequency for each note is presented in the table below. The frequency of higher/lower octave is a double/half of the standard note. For example, in the table below, the frequency of C' is twice of that of C.)

|  |  |
| --- | --- |
| Note | Frequency (Hz) |
| C | 261.63 |
| D | 293.67 |
| E | 329.63 |
| F | 349.23 |
| G | 392.00 |
| A | 440.00 |
| B | 493.89 |
| C' | 523.26 |

## *Arduino Tone functions*

Arduino has provided a programmer with two functions related to tone. They are tone and mute functions. Basically, the tone function will generate a beep at a specified frequency. However, the frequency must be specified in integer. Thus we cannot generate a precise note (according to the note table provided). The naïve solution is to round the number. To stop the square wave, make a call to the mute function. The example code show how to make a note for C for 1 second.

#include <M5Stack.h>

void setup() {

M5.begin();

M5.Speaker.tone(262);

delay(1000);

M5.Speaker.mute();

}

### 'for'

The 'for' is useful for repeating a block of statements. In this example, the system will repeat the println function for 100 times (0 to 99).

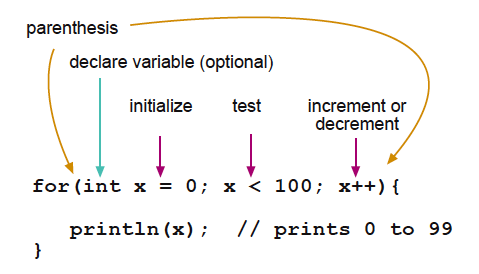


Illustration 11: Description of for loop (picture taken from http://www.arduino.cc)

### 'while'

The 'while' loop will run indefinitely as long as the test expression is true.

Example

int a=0;

while (a<100)

{

a++;

}

while (true)

{

// this is an infinite loop

}

### 'do .. while'

The 'do .. while' loop is similar to the while loop. However, the test expression is tested at the end of the loop. This means the loop will always run at least once.

Example

int a=0;

do

{

a++;

} while (a<100);

## *Lab Exercises*

In this laboratory, you have to:

1. Use the following code to create the short melody.

#include <M5Stack.h>

const int nC4 = 262;

const int nD4 = 294;

const int nE4 = 330;

const int nF4 = 349;

const int nG4 = 392;

const int nA4 = 440;

const int nB4 = 494;

const int nC5 = 524;

const int note[] = {

nC5,nG4,nG4,nA4,nG4,0,nB4,nC5};

const int duration[] = {

4,8,8,4,4,4,4,4 };

void setup() {

M5.begin();

}

void loop() {

for (int i=0;i<8;i++) {

int wait=1000/duration[i];

M5.Speaker.tone(note[i]);

delay(wait);

M5.Speaker.mute();

delay(50);

}

M5.Speaker.mute();

delay(2000);

M5.update();

}

1. Modify the code in the first sketch to the first section of Maha Chulalongkorn song (the music score is given below).

D' C' D' F C' A A A.. 

1. Please complete the given sketch by using the knowledge from previous laboratory to create a sketch that will read a key from the serial input. For keys 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C', 'D', 'E', 'F', 'G', 'A', 'B,' sound the corresponding note C, D, E, F, G, A, B, C', D', E', F', G', A', B' for one second.

#include <M5Stack.h>

void setup() {

M5.begin();

Serial.begin(9600);

}

void loop() {

if (Serial.available()>0) {

int key=Serial.read();

Serial.write(key);

switch (key) {

case 'c' :

M5.Speaker.tone(262);

delay(1000);

M5.Speaker.mute();

break;

case 'C' :

M5.Speaker.tone(523)

delay(1000);

M5.Speaker.mute();

break;

}

delay(1300);

menu();

}

M5.update();

}

void menu() {

Serial.println("\nExercise 3\nPlease enter note [c..a, C..A]");

Serial.print("> ");

}

1. The following is the notes for the Jingle Bells song: E, E, E, E, E, E, E, G, C, D, E, F, F, F, F, F, E, E, E, D, D, E, D, G. Construct a new sketch that will make M5stack plays each subsequent note of the Jingle Bells song for the duration when the A button is pressed and hold. After releasing the button, the song will proceed to the next note when the button is pressed again.
2. Change the song to any song you like and make the speed of the music playing be adjustable by Buttons (i.e. 5 speed level), fast: Button A increase the speed, Button B decrease the speed, button C reset the speed to default speed.

Section: \_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_

**Members**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Student ID: \_\_\_\_\_\_\_\_\_\_\_\_

**2190151 Computer Programming Laboratory.**

**Laboratory 5:**

1. Short Melody

Graded by ….................................................

1. Maha Chulalongkorn

Graded by ….................................................

1. Serial Piano

Graded by ….................................................

1. Jingle Bells

Graded by……………………………………...........

5.Variable speed music box

Graded by……………………………………...........