Table of Contents

[Brief explanation 1](#_Toc5632155)

[FLOW CHART 1](#_Toc5632156)

[FINITE STATE MACHINE 2](#_Toc5632157)

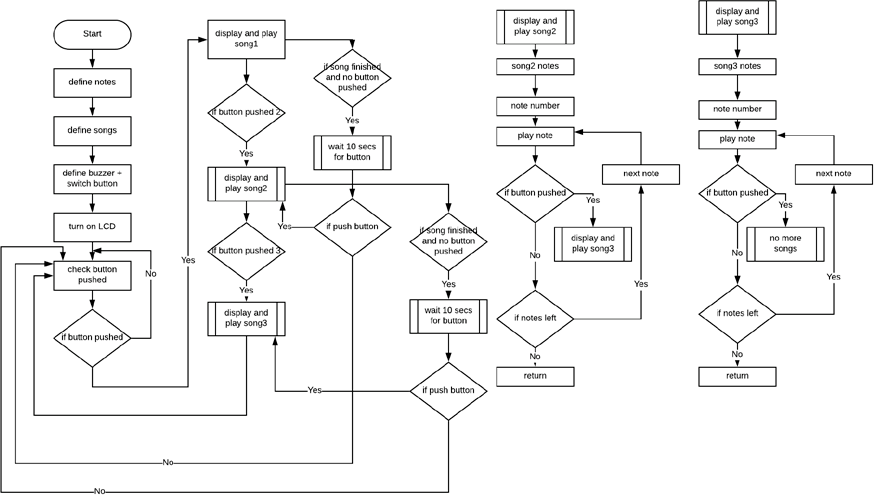
[Calculations 3](#_Toc5632158)

[References 3](#_Toc5632159)

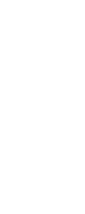
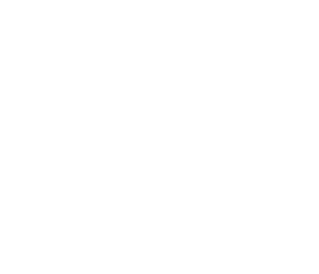
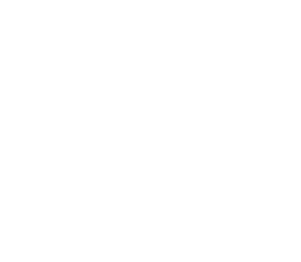
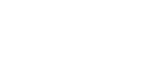
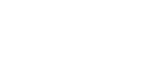
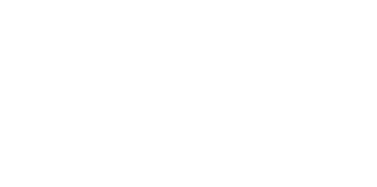
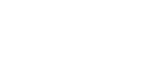
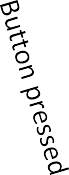
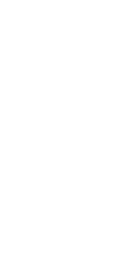
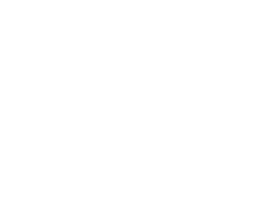
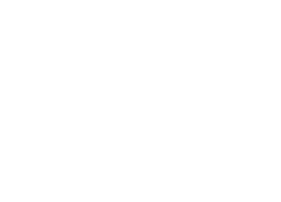
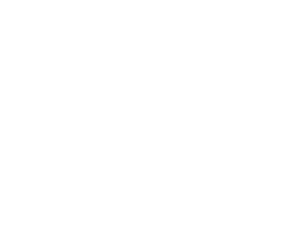
# Brief explanation

The algorithm that I used for this program needs the LiquidCrystal.h library, even though it may come preinstalled with the newer versions of the IDE, it might be missing from older versions. I took advantage of the liquid crystal display to make the program more user-friendly such that when he/she presses a button a notification would be shown on the screen to know what happens. I have implemented the buzzer to make it play a song, and I have used arrays of predefined characters as notes to make the piezoelectric speaker sound like a song is being played, alphabetical letters where associated with notes from real musical pieces such as “Alouette”, “I’m a little teapot” and “Do your ears hang low?” more in depth explanation comes below.

# FLOW CHART



# FINITE STATE MACHINE



Q0

Q3

Q1

Q2

Song finished and

no input detected

Check for

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Play all songs** | **Play 2 songs** | **Play first song** |
| Q0 | Q1 | Q1 | Q1 |
| Q1 | Q2 | Q2 | Q0 |
| Q2 | Q3 | Q0 |  |
| Q3 | Q0 |  |  |

Basic States:

Q0 represents the start or reset state where the program waits for the user to press the button; Q1 is the state in which the first song is played;

Q2 is the following state, song 1 was already played and song 2 is being played;

Q3 illustrates the last state, song 1 and 2 were already played, song 3 is being played;

Events:

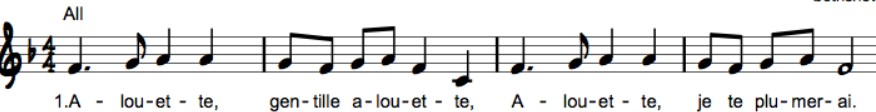
User pushes the button;

Song finished, no button is pressed within 10 seconds after finishing;

# Calculations

Programming the buzzer to make it sound alike a song was played I had to interpret the notes given from the songs since I could not understand exactly how Margolis (“Arduino Cookbook”,2011, pg 329) has used letters with notes, I still made use of several lines from his code, and I came up with my own methodology.

By looking at the musical notes I considered that there were 7 main tunes, I told myself that those were: do-re-mi-fa-sol-la-si, however when I started writing a letter for each of them I noticed there were more than that so whatever was more would get the last letter and whatever was less would get the first, thus I included alphabetical letters from A to G.



e.g.: first four notes are converted to CDEE

# References

Margolis, M., 2011. *Arduino Cookbook*. 2nd ed. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media.