**Simon on Arduino**

# Purpose

This document will describe the development of Simon on the Arduino.

# Table of Contents

Contents

[1 Purpose 1](#_Toc407741395)

[2 Table of Contents 1](#_Toc407741396)

[3 What does the system need to do 1](#_Toc407741397)

[4 Hardware 1](#_Toc407741398)

[5 Software 1](#_Toc407741399)

[5.1 Function Prototypes 1](#_Toc407741400)

[5.2 Function Hierarchy 2](#_Toc407741401)

[5.3 Flow Chart 2](#_Toc407741402)

[6 2](#_Toc407741403)

# What does the system need to do

* Read switches which are push buttons and possibly slide switches.
* Write to four LEDs.
* Make four sounds of a diatonic arpeggio and a fifth sound of a “raspberry”.
* Random number generation.
* Maintain array of the expected sequences of push buttons.
* Maintain array of what buttons were pressed sequentially.

# Hardware

* Arduino Mega 2560
* 4 LEDs of different colors (may need more to make colors).
* Push Button Switches.
* Slide switches SPDT
* Pull up resistor SIP packs for LEDs and Switches.
* 555 Timer with DAC ladder tied to the FM input.
* Speaker
* External 5VDC adaptor 500mA – 1A.
* Ribbon connector and headers 0.1 sp (2.54mm sp) to connect from Arduino to custom IO board.

# Software

## Function Prototypes

* byte read\_inputs(void) – This function returns the following:

Returns 1 – 4 corresponding to each of the 4 buttons.

* void write\_leds(byte) – This function lights the LEDs

<Add definitions – colors TBD> Each LED will be mapped to a bit. This allows multiple LEDs to be on.

* void set\_frequency(byte) – This function sets the frequency when sound is set to on.
  + 0 - Rasberry
  + 1 – Tonic
  + 2 – 3rd
  + 3 – 5th
  + 4 – 8va of Tonic
* void enable\_sound(boolean) – This function turns on the sound if set to true.
* byte get\_random\_button(void) – This function returns a random number from 1 – 4 using a random number generator and modulo arithmetic.

## Function Hierarchy

## Flow Chart

# 