

Team 15 Product Design Specification

Executive Summary

This project is a simple electronic sequence-based memory game. The game works by displaying a randomly generated sequence of lights and tones that correlate to specific buttons on the device. The game player then attempts to input the same sequence as displayed. The sequence starts at a length of 1 and increases by 1 with each successful matched sequence input by the player. When the player inputs an incorrect sequence the game ends and reports the length of the final correct sequence. Each time the game is played a new random sequence will be generated. Multiplayer mode will allow players to take turns against one another until a player inputs the incorrect sequence.

Brief Market Analysis

Our product mainly targets parents who are looking for a game that helps their children develop memorization abilities while having fun. Parents and adults may even find it enjoyable and it may have a nostalgic appeal to people familiar with the similar memory game known as Simon.

Similar classic Simon games and other memory related toys that are already out there on the market these would become our primary competition. We decided to make our product with more add-on features to enhance user's experience with the game compared to classic version. Those features include two extra sides with original design (total of 6 sided compared to 4 sided classic toy), a 7 segment display screen to report player's score, which is useful to make challenges with friends, various difficulty levels, and multiplayer support.

Additional features may be explored during development. The price range for the Simon games currently found on the market are around \$20-30. We estimate the price point for our design to be reasonably estimated at \$20. As we finalize the design and components we will adjust the price accordingly.

Requirements

The device must be assembled on a 2 or more layer printed circuit board, with a microcontroller to accurately record the user's inputs and their score. The device will have 6 buttons that will serve as inputs, as well as 6 LEDs which will correspond to each of the buttons.

The system must have a way to accurately keep track of what buttons the user presses. The device must have a randomized sequence of colored buttons each time the game is played. Upon startup, the device may have a short animation using the light displays.

The device should hold the user's attention for at least one game. The difficulty of the game may be toggleable by the user based on their preference or skill level. A multiplayer mode may be included to expand potential gameplay options.

The system will have a speaker to make a distinct sound for each button that is pressed, and the speaker may notify the user of the number of rounds they have successfully passed. The speaker may also be used to play startup tones, or a tone when the user reaches a certain benchmark.

The device must be portable and untethered, using batteries as the source of power for the system. The batteries should last for at least 100 games before needing to be changed out. The device should have an accessible battery compartment for maintenance, while all of the other parts of the device should be permanently fixed onto the device.

As a whole, the device should be durable and able to withstand minor impacts. The unit will also have a power switch to turn the device on and off. The device should be comfortable to hold and use, with rounded edges and corners, and must not cause harm to the user. The device will not infringe on any patents or copyrights.

System Architecture

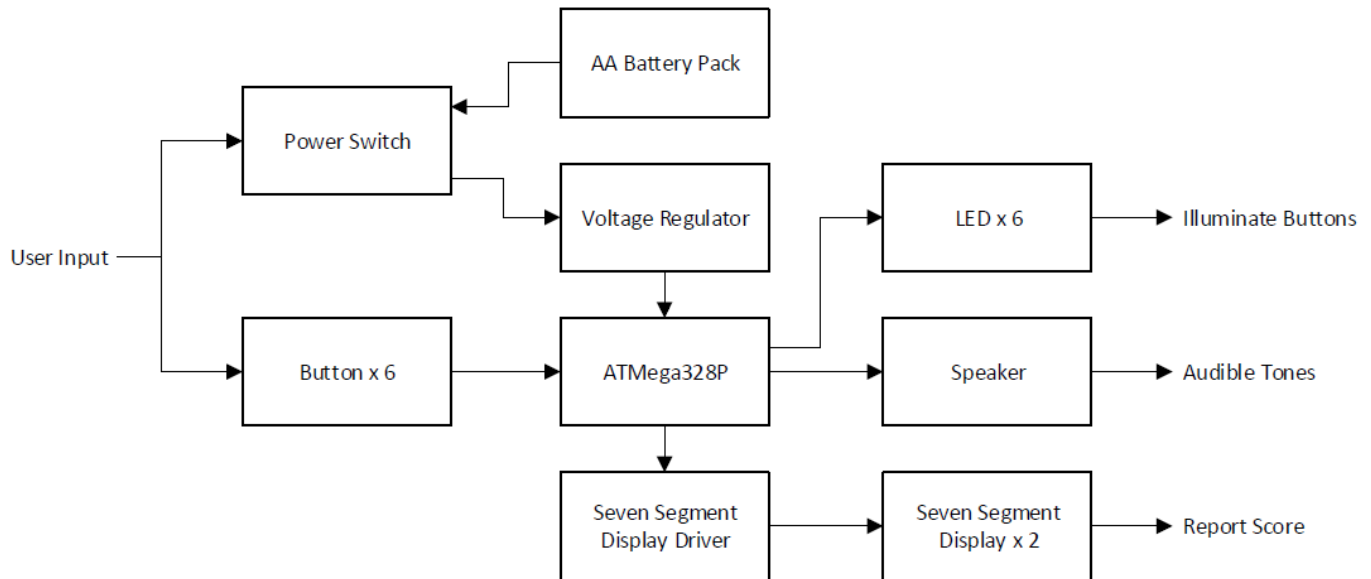


Figure 1: L1 Decomposition

Design Specifications

- **Processor:** Atmel ATmega328P
- **Clock:** 16MHz external crystal oscillator
- **Sensors:** 6 x push buttons, power switch
- **Actuators:** 6 x LEDs, 2 x seven segment displays, speaker
- **Power:** AA Batteries with voltage regulator
- **Firmware:** Embedded C
- **Development Environment:** Atmel Studio 7, Github Repository

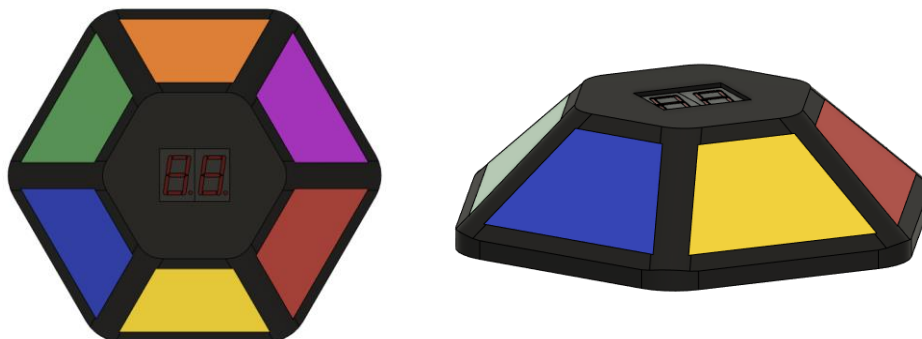


Figure 2: Preliminary Physical Design