Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2022

Senior Design Bois
Metaglass

Long Nguyen
Nathen Paul
Thinh Nguyen
Zane Malacara

Revision History

Revision	Date	Author(s)	Description
0.1	1.01.2016	GH	document creation
0.2	1.05.2016	AT, GH	complete draft
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests
1.2	3.03.2022	LN	Initial draft
1.3	3.11.2022	LN, TN, NP, ZM	Update draft

Contents

List of Figures

List of Tables

1 Introduction

Our objective for MetaGlass is to create the next generation of interactive glass windows. Our goal for our project is create a touch screen glass panel that implements a Simple Simon Says Game in which the user will be presented with window consisting of 6 squares appear. The player must memorize the order of the squares light up and then replicate it by touching each individual square in the correct order. The glass window has a film of ITO (Indium Tin Oxide) covering it, making it conductive and possible to attach wire and copper tape to. We will connect the glass with an Adafruit Feather hardware which will detect capacitive touch when the user touches the panel. All the electronics will be enclosed in wooden frame, giving the user the impression that it is a regular window. Our future vision is to make interactive glass the new standard in the construction of transparent visual displays.

2 System Overview

Our system has three layers: Input, Processing, and Output. The Input layer captures signals using capacitive touches, converts signals to data and send them to the Input controller. The Input controller communicates with the Processing layer by transmitting data to the Central Processing Unit (CPU). After receiving data, the CPU starts communicating with the software (games) by performing some data handling procedures. The CPU will convert data to signals after exchanging data with software before transmitting data to the Output layer. The Output controller in the Output layer receives signals from the CPU and send them to LED to display the software.

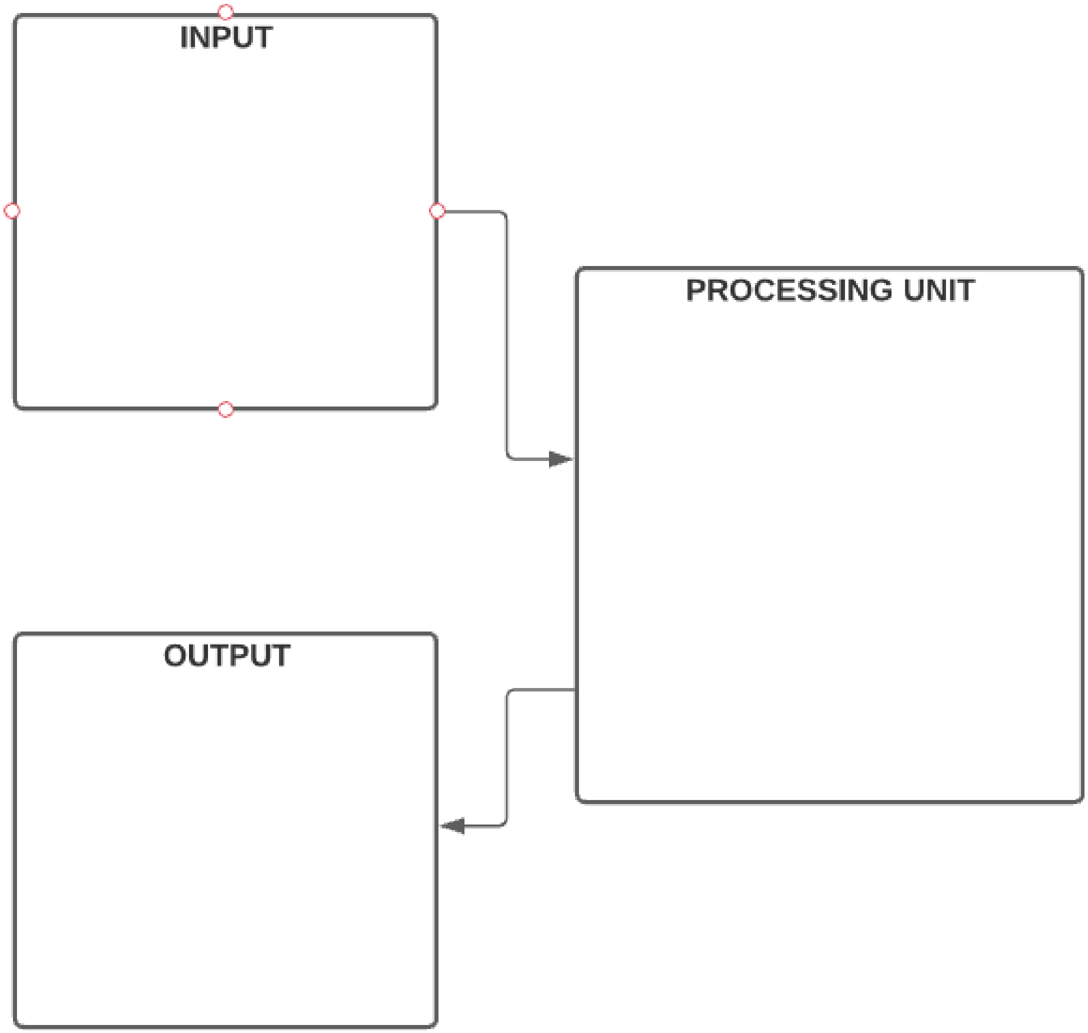

Figure 1: System Architecture Overview

3 Subsystem Definitions & Data Flow

The operation starts by the user touching the LED display. Capacitive touches will capture signals and send them to the Input controller at the Input layer, then the Input controller transmits signals the Processing Unit to process data. After receiving data, the Central Processing Unit (CPU) at the Processing layer communicates with the software (games) to perform some data handling in between, and then the CPU sends data to the Output layer. After getting data, the Output layer converts data into signals and display on the LED.

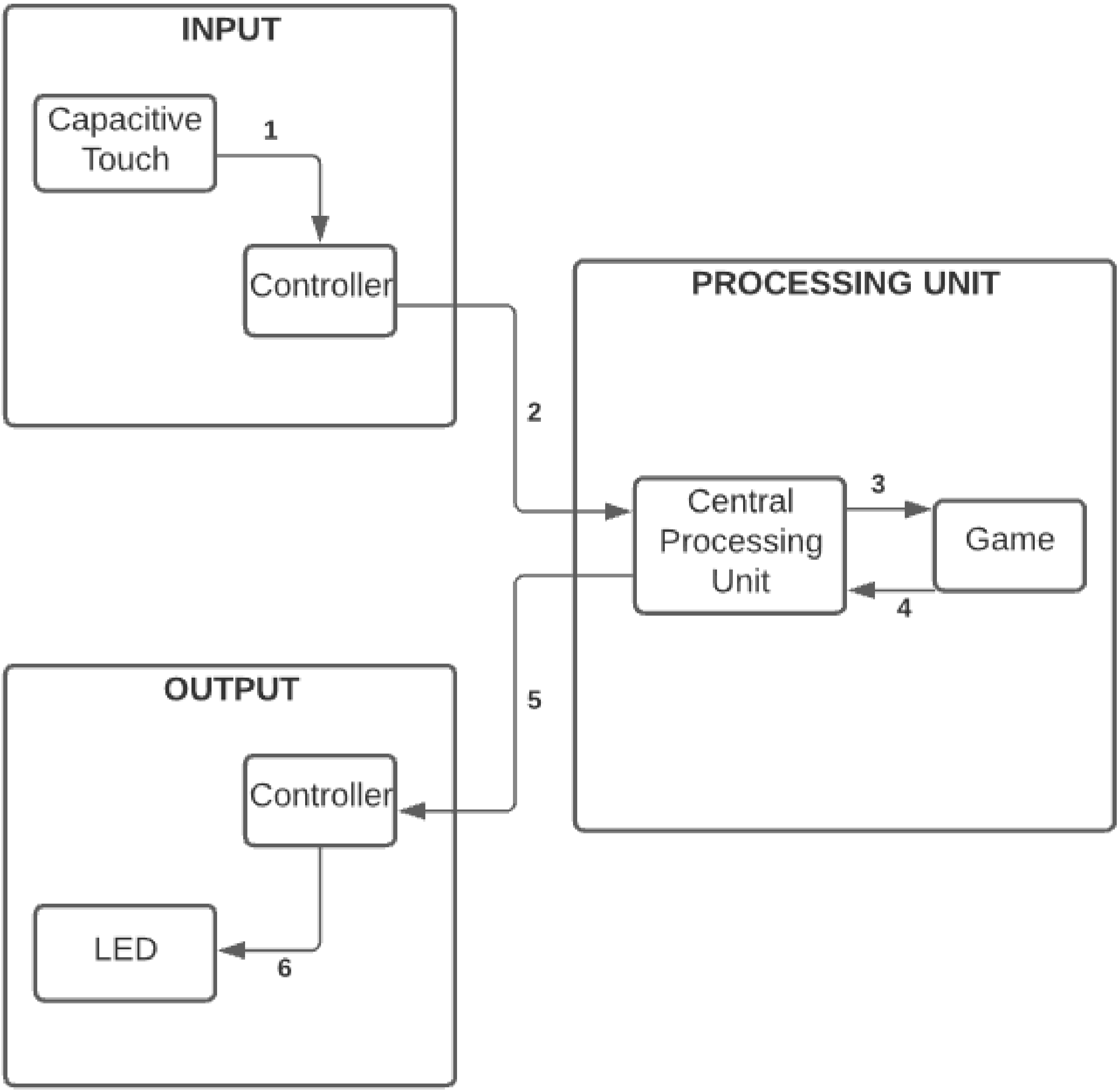

Figure 2: Data flow diagram

4 Input Layer

The input layer manages the physical input of the overall system. It decides what happens when a person touches the glass surface / interface. This layer mainly takes a wire from the Adafruit CAP1188 sensor and attaches it to the glass with copper tape. This allows for input to be sensed from the glass to the sensor. Then the sensor sends this data to the Adafruit Feather microcontroller.

4.1 Layer Hardware

Hardware components include: Feather M0 WiFi - ATSAMD21 + ATWINC1500 Boards

The M0 feather is a programmable micro-controller that is used as the center of the system. It programs the CAP1188 Capacitive Touch Sensor and the NeoPixel LED strip, by sending data via I2C protocol.

4.2 Layer Operating System

There is no OS in the system.

4.3 Layer Software Dependencies

We are using the Arduino SAMD Boards(Version 1.8.12) and Adafruit SAMD Boards(Version 1.7.9) packages, and the Arduino I2C library.

4.4 Subsystem 1

The CAP1188 capacitive touch sensor will be used to take in a user's touch as input for used for an interactive game, showcasing the ability of glass circuits.

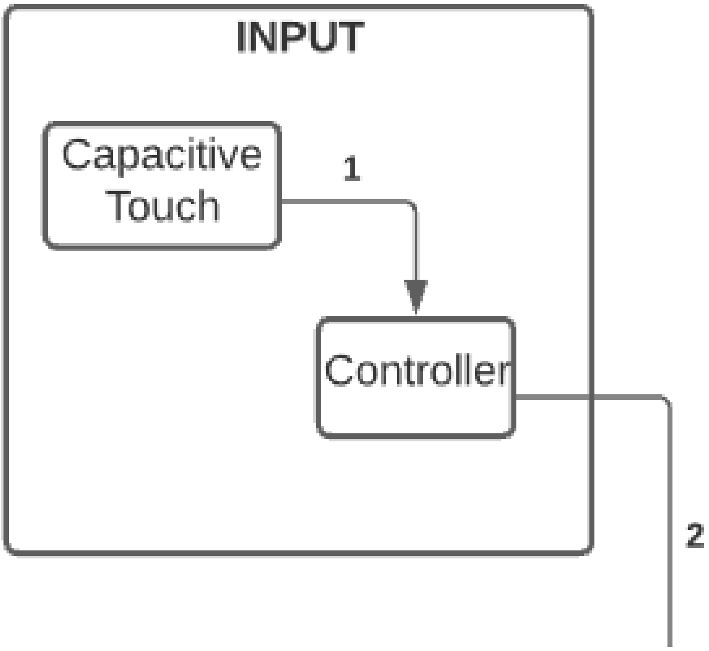

Figure 3: Diagram for Input Layer Subsystem

4.4.1 Subsystem Hardware

Will receive data from micro-controller to configure touch settings, sensitivity settings, etc. via I2C protocol. There will be glass that is coated in an ITO(Indium Tin Oxide) material that will be connected to the touch inputs, and will act as a wire so that when the user touches the glass, it will send that signal back to the sensor.

4.4.2 Subsystem Operating System

There is no OS in this subsystem

4.4.3 Subsystem Software Dependencies

Using the CAP1188 Library (Version 1.1) to program the sensor

4.4.4 Subsystem Programming Languages

Data is being sent from the micro-controller to the CAP1188 using C++11.

4.4.5 Subsystem Data Structures

The data is being sent through I2C, which is a synchronous serial communication bus. There are two wires, a data line, and a clock line. The protocol sends 8 bits of data across the data line at a time. The master device sends the data, and the child devices acknowledge that the data has been sent. I2C is slow due to being configured as an open drain by default and forcing pull-up resistors on the data and clock lines.

4.4.6 Subsystem Data Processing

The sensor processes the data by storing the bits in the registers, which configure the device to be used a specific way.

5 Output Layer

The output layer handles the physical output of the overall system based on the instructions received from the controller from the processing layer, it will decide what to display on our glass. The Adafruit Neopixel receives a wire from the Adafruit Feather microcontroller which sends the Neopixel data. The data consists of which pixel is turned on.

5.1 Layer Hardware

There is no central hardware structure for the subsystems in this layer. Each subsystem is installed in separate sections of the window frame.

5.2 LED

The Adafruit NeoPixel LED Strip is a strip of smart and modular LED lights. The Neopixel LED strip is split into 4 sections which are lit up asynchronously. Each section is connected to the microcontroller. The Neopixel will be outputting colored light onto the glass and inserted into side of the frame.

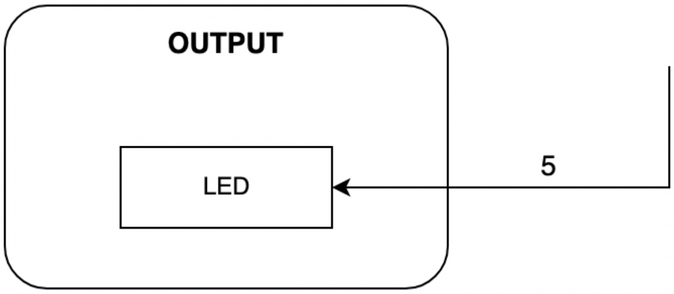

Figure 4: Diagram of Ouput Layer Subsystem

5.2.1 Subsystem Hardware

The outputted light comes from the Neopixel itself.

5.2.2 Subsystem Software Dependencies

The Adafruit_NeoPixel version 1.10.4 C library is used and developed in the Arduino IDE.

5.2.3 Subsystem Programming Languages

We are coding using C++11 in the Arduino IDE.

6 Processing Layer

The central processing unit first initializes the game by setting the LED display to interact with the player. After the game mode is selected, the CPU retrieves the stored game setup and sends it to the output layer. Simultaneously, a copy of the current state of the game is also stored in the memory. At each play, after sending signals to the LED display, the CPU will verify the player's respond to determine whether the player has successfully passes the round. After the player completes the final round successfully, the CPU will erase the memory of the game and return to the beginning/read-to-play state.

6.1 Layer Hardware

Hardware components include: Adafruit Feather M0 WiFi - ATSAMD21 + ATWINC1500 Boards

6.2 Central Processing Unit

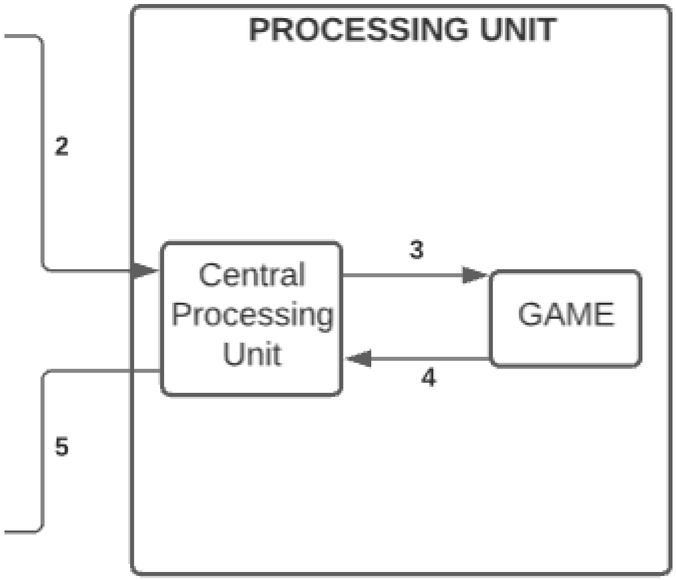

Figure 5: Flow Diagram of Processing Subsystem

6.2.1 Subsystem Hardware

- Wood (for the enclosure containing this subsystem)
- Wires to connect input and output from this subsystem

6.2.2 Subsystem Software Dependencies

- Arduino 1.8.19

6.2.3 Subsystem Programming Languages

- C++11

6.3 Game

6.3.1 Subsystem Hardware

- Adafruit Feather M0 WiFi - ATSAMD21 + ATWINC1500 Boards

6.3.2 Subsystem Software Dependencies

- Arduino 1.8.19

6.3.3 Subsystem Programming Languages

- C++11

6.3.4 Subsystem Data Structures

- Stack

6.3.5 Subsystem Data Processing

All the moves in the game are determined randomly using the random generator. There are a few steps of processing in this game:

1. The program generates a random move, send it to the LED display, and store it to the "game move" stack.
2. The program verifies the player's input with the "game move" stack. If the input is matching with the stack, the program continues the next round.
3. If the user's input does not match the record, the program will erase the stack and prompt the player to whether restart or end the game.
4. After the user successfully completes all 15 rounds, the program will send out the "congrats" message and prompt the play to whether restart or end the game.

Note that messages in this game are represented by animated light colors.

7 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

- Frame CAD design
- Outside wooden frame/enclosure CAD design

REFERENCES