**Smart Home (Simulator)**

Simon Chau

861099861

**Project Description**

What I attempted to build was a smart home with automatic windows, with a smart door, and temperature sensors. However due to some hardware and software difficulties, I was not able to fully create what I had hoped to make. The doors would have a smart lock which would receive input from the Bluetooth. The temperatures would be adjusted also with Bluetooth to control the air conditioning and heater. However, I was only able to create the smart windows.

I built an automatic/manual curtain roller that is determined by the amount of sunlight it detects. For sunlight simulation, I used IR sensors and blocked them to simulate darkness. The curtains/blinds should automatically adjust according to whether the IR sensor has a high value or a low value. If the value is high, that simulates sunlight, and the curtains/blinds should rise to allow light into the house. If the value is low, that simulate darkness, and the curtains/blinds should drop so people do not see into the house easily. If set to become manual, the curtains/blinds will not move automatically and can be manually adjusted via keypad. This project is in hopes of helping to save electricity by automatically providing natural light.

**Complexities**

**Working**

Bluetooth Communication - Windows

Used a Bluetooth Module HC-05. First time configuring with Bluetooth to transmit data. Using USART to transmit and receive data on the Master Microcontroller. Made sure that this part was working in conjunction with IR sensors automatic movement.

IR Sensors

Connected IR sensors to simulate light. The sensors transmit a value that will be determined if above or below the threshold. If below the threshold, then the curtains/blinds should drop and simulate darkness. If above the threshold, then the curtains/blinds should rise and simulate light. This was a vital part for the milestone to show that the curtains/blinds can rise and drop automatically when receiving the IR sensor data.

Stepper Motor

The Stepper Motor will be utilized to control the curtains/blinds, doors. However, I was only able to get this working for the curtains/blinds. The precise rotations are used to ensure the blinds are not over risen or over dropped. I could have simulated a door, but there would not have been any code to act as a smart lock. Due to time constraint and software issues, I could not get the smart part of the door working.

USART and SPI Communication

I am using USART when communicating between the Master Microcontroller and the Bluetooth Module. The Bluetooth sends signals and the Microcontroller will have flags depending on what signals are sent. Using those signals and flags, the Master would then send the signals to the Slaves through SPI. This in turn would control the curtains/blinds to rise or drop if the automatic flag was on, or do nothing and wait for manual input.

**Not Working**

Temperature Sensors

If I was able to produce this part I would have attempted to measure the temperature of a room and have it automatically simulate the air conditioning or heater to turn on. I could not get to work on this feature however because of time constraint and issues with other components of my project. I would have used Bluetooth to control the temperature that these sensors would be outputting.

Bluetooth Communication – Door and Temperature

If the door and temperature were functioning, the Bluetooth communication would control the door and temperature. As mentioned above about the temperature, the Bluetooth communication would allow control over the air conditioning and heater. For the door, it would be an automated lock or unlock after a code was inputted. These could not be gotten to due to time constraints as well as some software/hardware difficulties.

**User Guide**

* Connect Phone to Bluetooth Module using App
* “Auto” Button allows the system to run automatically
* “Manual” Button allows the user to have manual control over the system
* If the IR sensors are exposed to each other while the system is in Auto mode, then the curtains/blinds will rise on their own
* If the IR sensors are blocked from each other while the system is in Auto mode, then the curtains/blinds will drop on their own
* If the system is in Manual mode, blocking or exposing the IR sensors to each other will have no effect
* In manual mode, the User can press C to move the curtains/blinds up or B to move the curtains/blinds down

**(Make sure that the stepper motors have their own separate power source to ensure efficiency)**

**Technologies and Components Used**

* AVR Studio 6.2
  + Used to program the microcontrollers
* ATmega1284
  + The microcontroller used to receive and transmit data
* Bluetooth Module HC-05
  + Sent data from Bluetooth to the microcontroller on USART
* Stepper Motors
  + Control the curtains/blinds
* IR Sensors
  + Acts as a simulator for light detection
* USART/SPI Communication
  + Bluetooth communicated to microcontroller through USART
  + Master microcontroller communicated to other Slave microcontrollers through SPI
* Keypad
  + Used for input when the mode was set in Manual

**Links to Demo:**

<https://www.youtube.com/watch?v=u1esHqUuufk>

* Video Demo

<https://github.com/schau009/CS-122A-Smart-Window/blob/master/122A%20Final%20Project/122A%20Final%20Project%20-%20Master%20Part.c>

* The Master Part of the Code

<https://github.com/schau009/CS-122A-Smart-Window/blob/master/122A%20Final%20Project/122A%20Final%20Project%20-%20Slave%20Part.c>

* The Slave Part of the Code

<https://github.com/schau009/CS-122A-Smart-Window/blob/master/122A%20Final%20Project/keypad.h>

* Keypad for input when the mode is Manual

<https://github.com/schau009/CS-122A-Smart-Window/blob/master/122A%20Final%20Project/scheduler.h>

* Scheduling for the SM’s to work

<https://github.com/schau009/CS-122A-Smart-Window/blob/master/122A%20Final%20Project/usart_ATmega1284.h>

* ATmega1284 Initialization and Communication setup

**Image of Component Connections**



All files or sources used/provided were from the CS 122A Course taught by Jeff McDaniel. Hardware connections and information was provided by IEEE-UCR website.