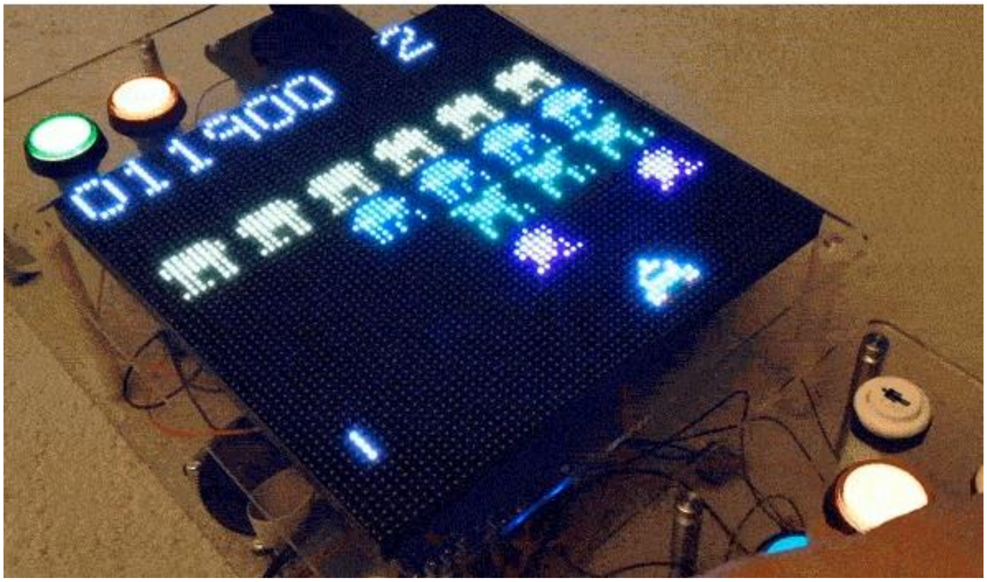
LED Gaming System Design Report



https://learn.adafruit.com/ledgames-beaglebone-black-64x64-led-game

Team Name: “Arcade Vae”

EEL4924C Senior Design Student:
Victor Adoptante Espidol [victor.espidol@gmail.com]

Table of Contents

Project Summary

The goal for this project is to create an LED gaming system with a wireless remote control. It aims to bring back the nostalgic entertainment of the old arcade games into your own home. The application interface allows the user to choose from a variety of games that suit the users need, as well as being able to play at a comfortable distance. Incorporating the use of a wireless remote with not only a joystick and pushbuttons, but an accelerometer as well can allow users to enjoy a multi-faceted way of playing games. There is no limitation to the creativity of what games can be played and that is what makes this, and many other gaming systems, to be continually amusing. I hope the result of this project is a well-functioning and useful gaming system that allows people to continue to enjoy low resolution games.

Project Features/Objectives

Components

- Adafruit (64x32 RGB LED Matrix)
- ~~FPGA (Altera Max 10 10M50DAF484C7G)~~ **NOT USED**
- 2 Microprocessors (ATSAM4SD32C)
- Accelerometer
- Joystick
- 4 Pushbuttons
- HC-05 Bluetooth chip (receiver/transmitter)
- Schmitt trigger chip
- Speaker
- LTC1661 DAC chip
- Transistors
- LM386 Amplifier chip
- 5V and 3.3V Voltage Regulators
- 5V Power supply
- 9V Batteries
- Atmel-ICE Debugger/Programmer

Setup and Interfacing

Physical Configuration:

For the main system, it will consist of the LED Matrix to display the game, the FPGA (Altera Max 10) to drive the LED Matrix, the microprocessor (A) to handle the RTOS software algorithm for the game, the Bluetooth chip to receive the remote-control data, the voltage regulator, the DAC chip, the amplifier circuit, the speaker, the power supply to power the system, and the PCB to contain it all in one place. The entire system will be held in a wooden box with a glass side to display the LED Matrix.

The secondary system is the wireless remote control. It will consist of another PCB that holds another microprocessor (B), joystick, pushbuttons, Schmitt trigger for debouncing, accelerometer, Bluetooth chip to send the data to the other microprocessor, voltage regulator, and the power supply.

System Process:

1) Initialize communication protocols and Display “Home Game” menu upon startup
2) Wait for remote controller to choose what game to play
3) Once game is chosen, there will be an option to play it or view the top scores

4) When in play mode the game starts with a countdown
5) Player moves and attacks with joystick, pushbuttons, or accelerometer from the remote controller
6) Upon winning, if within constraints of top 10 scores, the player will be able to input their name otherwise, the game can be reset or returned to home game menu
7) Upon losing, the player can either play again or return to home game menu
8) The game can be reset whenever the player wants and can choose to either reset the game or return to the home game menu

Main Concerns:

The biggest concern for this project is the communication interface between each of the systems. The FPGA will communicate to the LED Matrix using 13 GPIO pins for the data and control bits. Microprocessor (A) will communicate to the FPGA using an external bus interface (EBI) to send the data to display onto the LED Matrix. Microprocessor (B) will communicate to microprocessor (A) using Bluetooth communication to send the data from the remote control to update the game. The issue with all the inter-communication between the devices is synchronicity in timing with receiving, updating, and sending to have a fluid game.

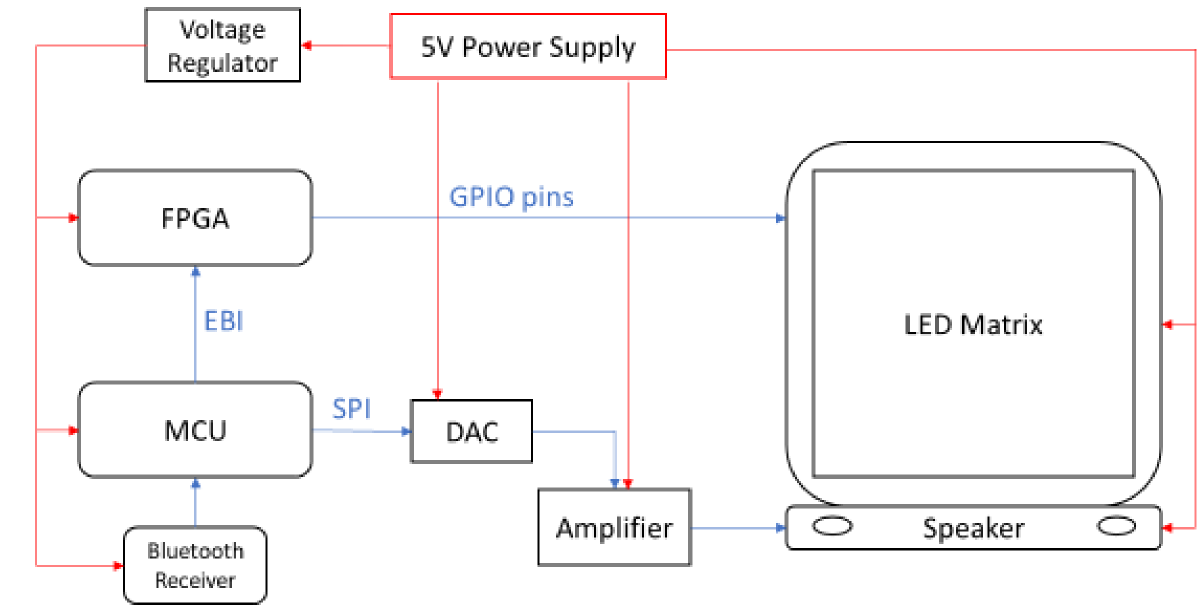
Additional Features

These additional features are stretch goals to be implemented onto the system if and only if, the main functionality of the game has been finished.

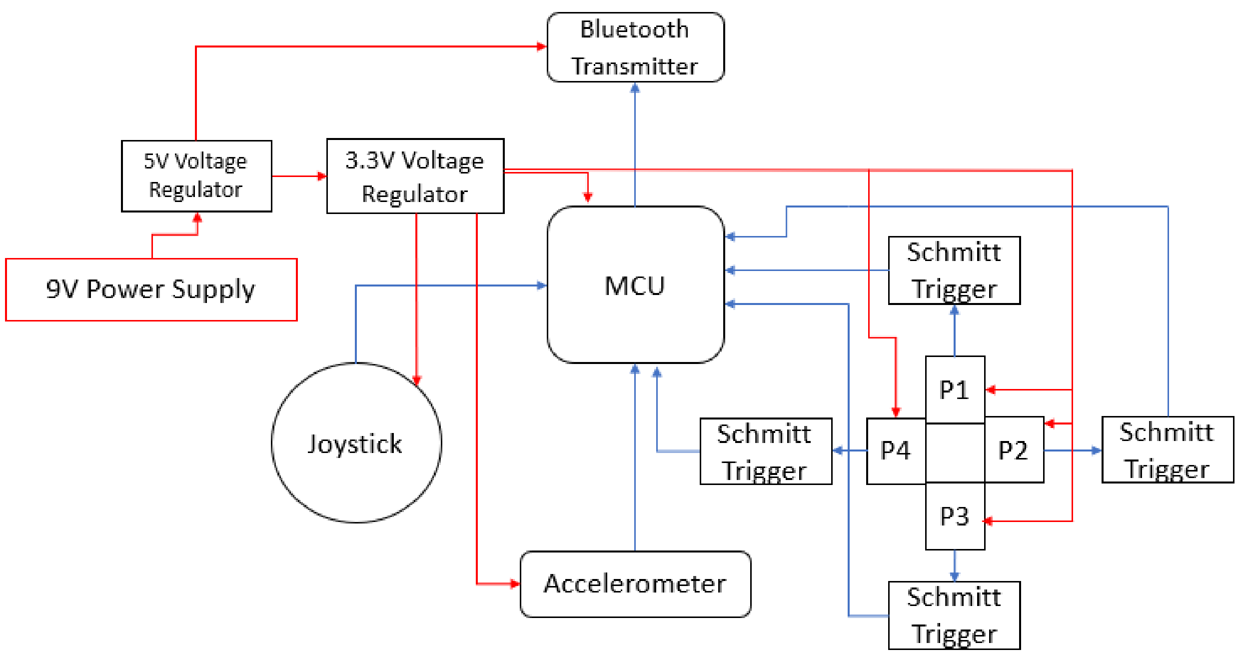
- Speaker to play in-game sounds

Flowcharts & Diagrams

Main System Diagram

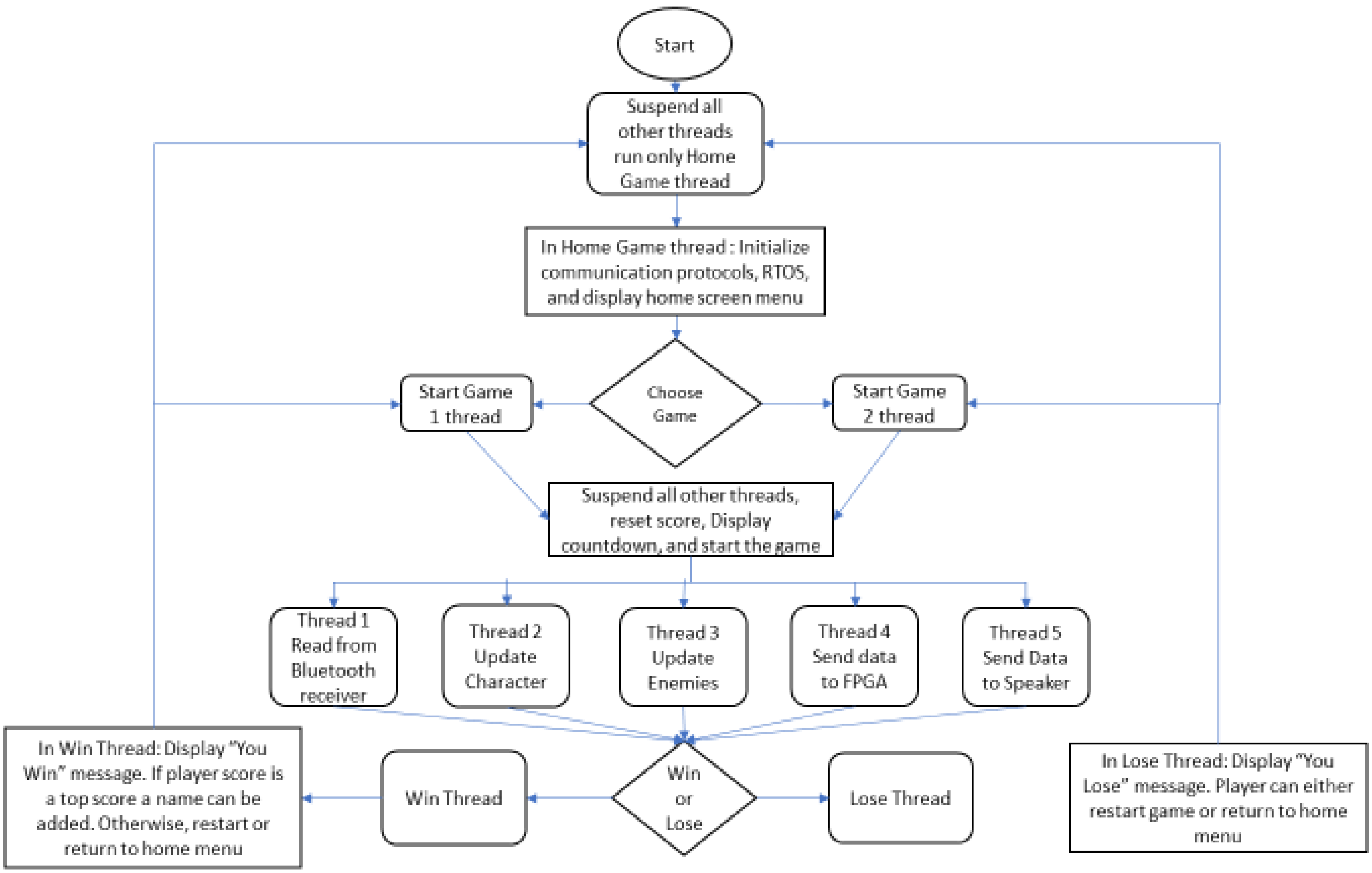


Wireless Remote-Control Diagram

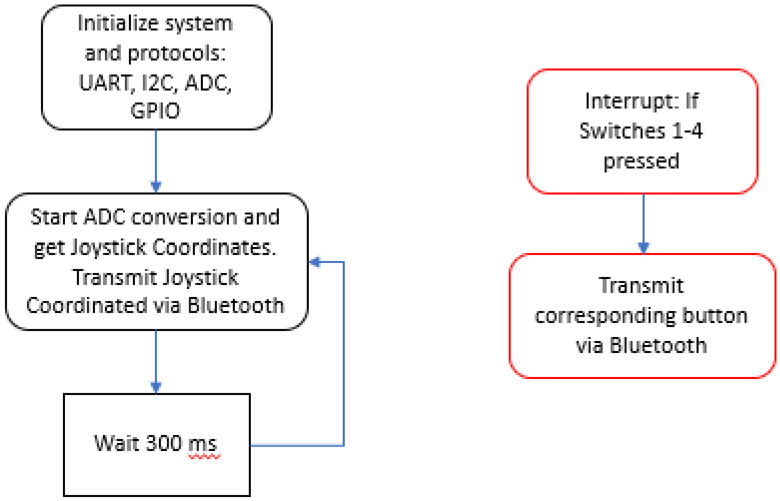


Software Flowchart for RTOS Game

Main System Flowchart:



Remote Control Flowchart



Concept/Technology Selection

FPGA and LED Matrix:

LED Matrix's are visually aesthetic and are at a lower cost when scaled up as opposed to LCD screens. Therefore, displaying a game onto an LED Matrix was more of an optimal choice. The reason for choosing an FPGA to drive the LED Matrix was chosen because of the high processing speed that FPGA's have to offer. Since it requires drawing the image onto the display repeatedly, the speed at which it gets drawn needs to be optimized for the game to be played without any lag. Also, the Adafruit 32x32 RGB LED Matrix was designed to be driven by an FPGA.

Microcontroller to FPGA:

The reason for using EBI instead of programmable I/O (PIO) and advanced high-speed bus (AHB) is because PIO would take away processing time from the microcontroller, and the microcontroller chosen does not have the AHB functionality. Also note, that the speed of EBI interface is faster than PIO.

RTOS Software Algorithm:

Using a real-time operating system (FreeRTOS) to develop the code for the game algorithms was decided because of functionality of using shared resources for a multi-threaded system that games require. This will allow for the microprocessor to handle intercommunication from different devices in a synchronized fashion whilst being able to progress the game. Using an event driven loop with several complicated tasks could prove catastrophic if the resources within the microprocessor are too limited. The real issue lies within how well the scheduler can optimally decide the right sequence of tasks to run for the game to run smoothly.

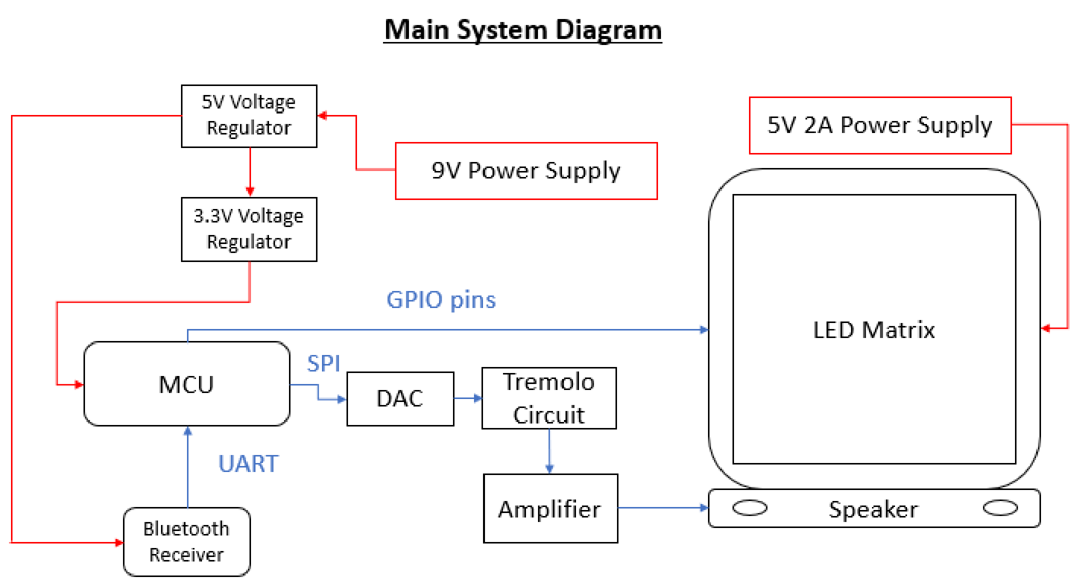
Wireless remote control:

The wireless remote control will transmit the controller data via Bluetooth communication. This was decided, as opposed to using IR communication, because its effective range is wider, and it is not constrained by line-of-sight communication. The reason for a wireless vs wired application was decided for more comfortability playing. The remote control will also have a Schmitt trigger circuit to debounce the pushbuttons for more reliability.

Obstacles Overcome

1) The original idea of this project was to utilize an FPGA to drive the LED matrix to save processing time and resources for the microcontroller. However, when attempting the EBI communication protocol with the FPGA it turned to be more difficult than expected. At first, I attempted to utilize the ASF drivers for the ATSAM4SD32C chip to get the EBI interface working and ended up failing. Next, I tried writing my own EBI driver code to get the communication working and ended up with another failure. I tried debugging with the Digital Analog Discovery 2 to see what was being outputted from the chip, but nothing logical came out. Therefore, to move forward with the project I made the decision to completely drop the FPGA and to utilize GPIO pins from the microcontroller to drive the LED Matrix. This allowed the project to work out smoothly in the end. Also, because the clock speed of the microcontroller was set to 120 MHz the display was still able to work without any noticeable lag.

   Updated Diagram:



2) When trying to get the HC-05 Bluetooth module working in the beginning stages, the two chips did not connect. This led to further investigation which showed an error in the designed PCB, which always had the HC-05 chips in command mode. After, de-soldering and putting the two chips in data mode for communication the chips still did not connect. After, reading the datasheet and several other sources I realized I needed to first program the HC-05 chips in command mode with the correct AT commands, and then set them back to data mode for them to communicate. To do this, I needed to use the Arduino Uno to program the correct AT commands because I was not able to

do so with my microchip. After several different configurations I realized that I needed:

a. Set one HC-05 as a master and the other a slave
b. Set the master to bind to the slave address
c. Change the device class type for both to be 0
d. Set the baud rates of both HC-05 chips to be the same.

This configuration worked and the chips were then put back in data mode to communicate with each other

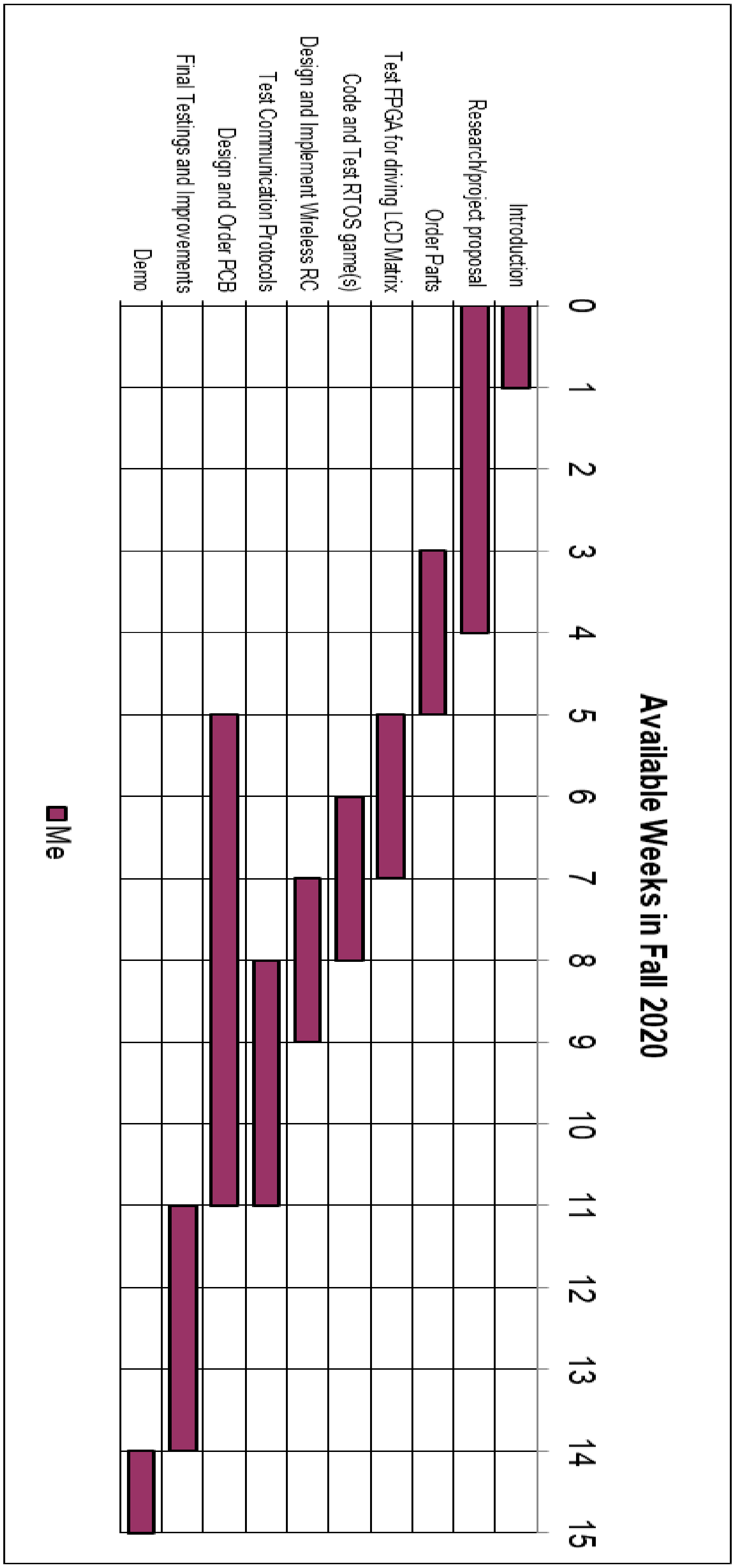
3) The analog circuitry of my design that utilizes the LTC1661 DAC chip, sound effect circuit, the LM386 amplifier, and speaker was ultimately not able to be put onto a PCB/perf board due to lack of time. The reason for this, was because of the amount of time used on trying to get the FPGA working with the ATSAM4SD32C and the LED Matrix. This led to the sound effect circuit being chosen at the very last minute, which ended up being a Tremolo circuit. Therefore, when it came down to the demo everything was on the breadboard.

Bill of Materials (BOM)

Design 1 - Final Project - Bill of Materials				
Description	Quantity	Price/Part	Part Number	Source
ATSAM4SD32C	2	$ 8.69000	ATSAM4SD32CA-AU	Microchip
32x32 RGB LED Matrix Panel - 4mm Pitch	1	$ 49.95000	PID: 607	Adafruit
5V 4A (4000mA) switching power supply - UL Listed	1	$ 14.95000	PID: 1466	Adafruit
Female DC Power adapter - 2.1mm jack to screw terminal block	1	$ 2.00000	PID: 368	Adafruit
Voltage Regulator - 5V	2	$ 0.95000	L7805	Sparkfun
Thumb Joystick	1	$ 3.95000	Thumb Joystick	Sparkfun
Schmitt Trigger	1	$ 0.50000	CD40106B	Sparkfun
9V Snap Connector	1	$ 1.25000	Snap Connector	Sparkfun
9V Battery Holder	1	$ 2.95000	9V Battery Holder	Sparkfun
SparkFun Triple Axis Accelerometer Breakout - MMA8452Q	1	$ 9.95000	MMA8452Q	Sparkfun
HC-05 Wireless Bluetooth RF Transceiver	2	$ 6.29500	HC-05	Amazon
120pcs Breadboard Jumper Wires	1	$ 5.79000	Jumper Wires	Amazon
20PCS 2.54mm 40Pin Male and Female Header	1	$ 7.49000	Headers	Amazon
IC DAC 10BIT V-OUT 8DIP	1	$ 4.00000	LTC1661CN8#PBF-ND	Digikey
Capacitor- 47uF -Electrolytic Capacitor	1	$ 0.99000	2721027	Digikey
CAP ALUM 4.7UF 20% 50V RADIAL	2	$ 0.03740	P10392TB-ND	Digikey
CAP ALUM 10UF 20% 25V RADIAL	11	$ 0.03570	ECA-1EM100I-ND	Digikey
IC REG LINEAR 2.85V 1.5A TO220-3	2	$ 2.98	LT1086CT-2.85#PBF-ND	Digikey
RES 10 OHM 2W 5% AXIAL	1	$ 0.12880	FW20A10R0JATB-ND	Digikey

CAP CER 0.1UF 50V X7R RADIAL	33	$ 0.03240	SR215C104KAT-ND	Digikey
CAP ALUM 0.47UF 20% 50V RADIAL	1	$ 0.04874	493-5923-3-ND	Digikey
POT 10K OHM 1/4W PLASTIC LINEAR	2	$ 2.90000	3310Y-001-103L-ND	Digikey
LED GREEN DIFFUSED T-1 3/4 T/H	2	$ 0.36000	160-1130-ND	Digikey
LED RED DIFFUSED T-1 3/4 T/H	2	$ 0.36000	160-1132-ND	Digikey
SWITCH PUSHBUTTON	4	$ 0.28661	PS12F91AF3.3NS-ND	Digikey
SPEAKER 8OHM 250MW TOP PORT 86DB	1	$ 1.26000	458-1130-ND	Digikey
IC AMP AUDIO PWR .325W MONO 8DIP	1	$ 1.17000	296-44414-5-ND	Digikey
RES 10K OHM 1/4W 5% AXIAL	4	$ 0.00475	CF14JT10K0TR-ND	Digikey
RES 470 OHM 1/4W 5% AXIAL	1	$ 0.004750	CF14JT470RTR-ND	Digikey
RES 1K OHM 1/4W 5% AXIAL	2	$ 0.00475	CF14JT1K00TR-ND	Digikey
2N2222	4	$ 0.45000	2N2222-ND	Digikey
	Total:	$ 155.96		

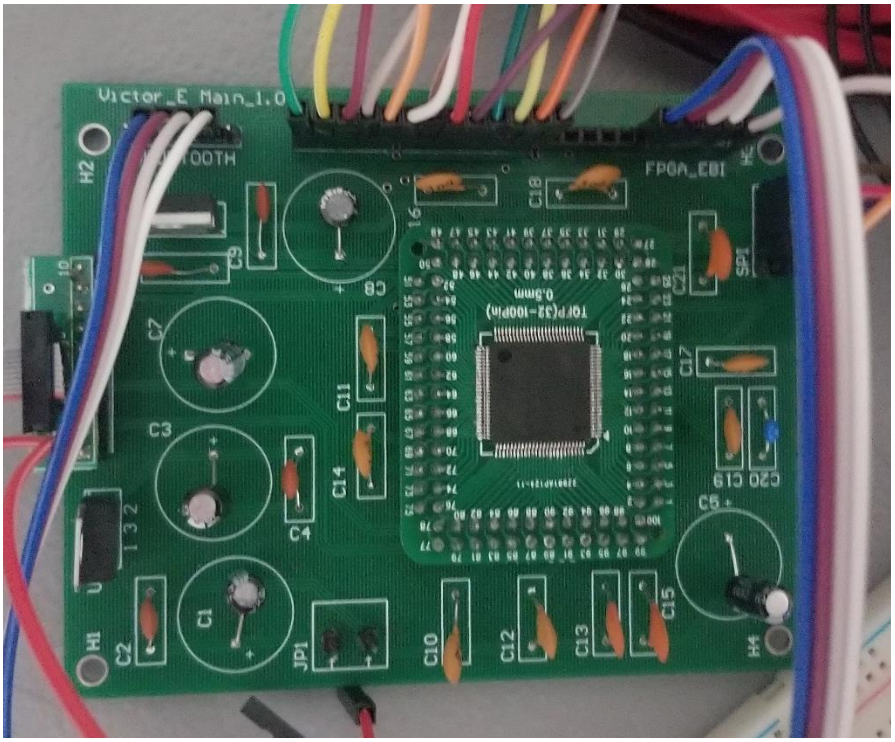
Gantt Chart


Available Weeks in Fall 2020
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Introduction
Research/project proposal
Order Parts
Test FPGA for driving LCD Matrix
Code and Test RTOS game(s)
Design and Implement Wireless RC
Test Communication Protocols
Design and Order PCB
Final Testings and Improvements
Demo
Me

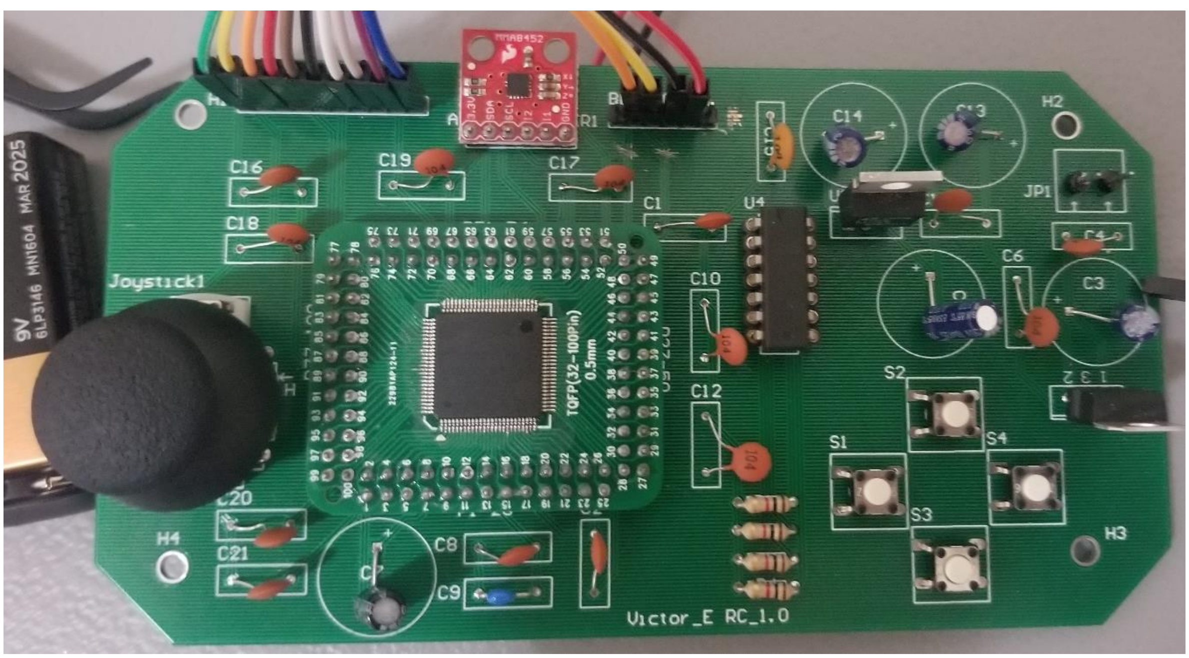
Appendix

Final Project Images:

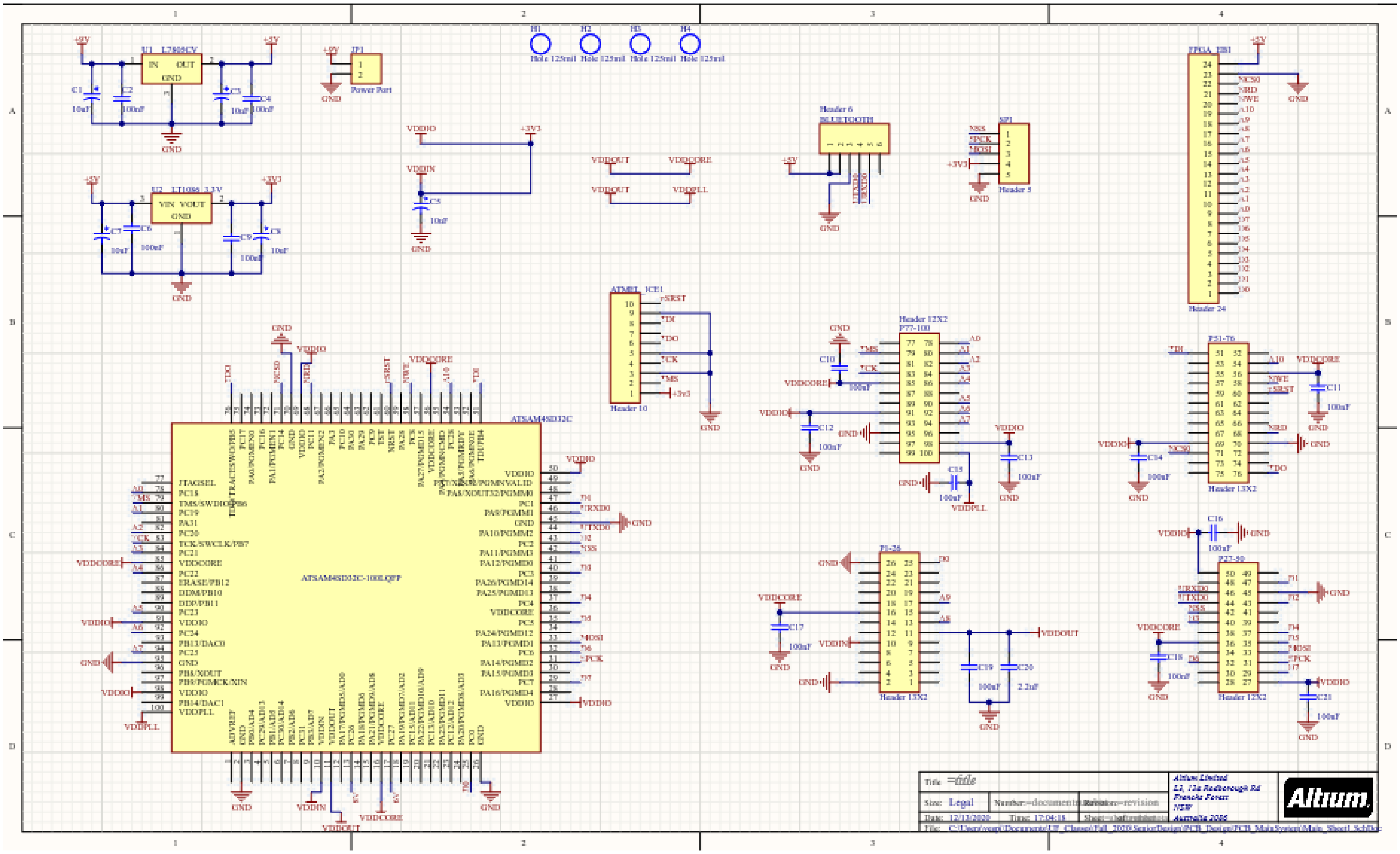
Main System PCB



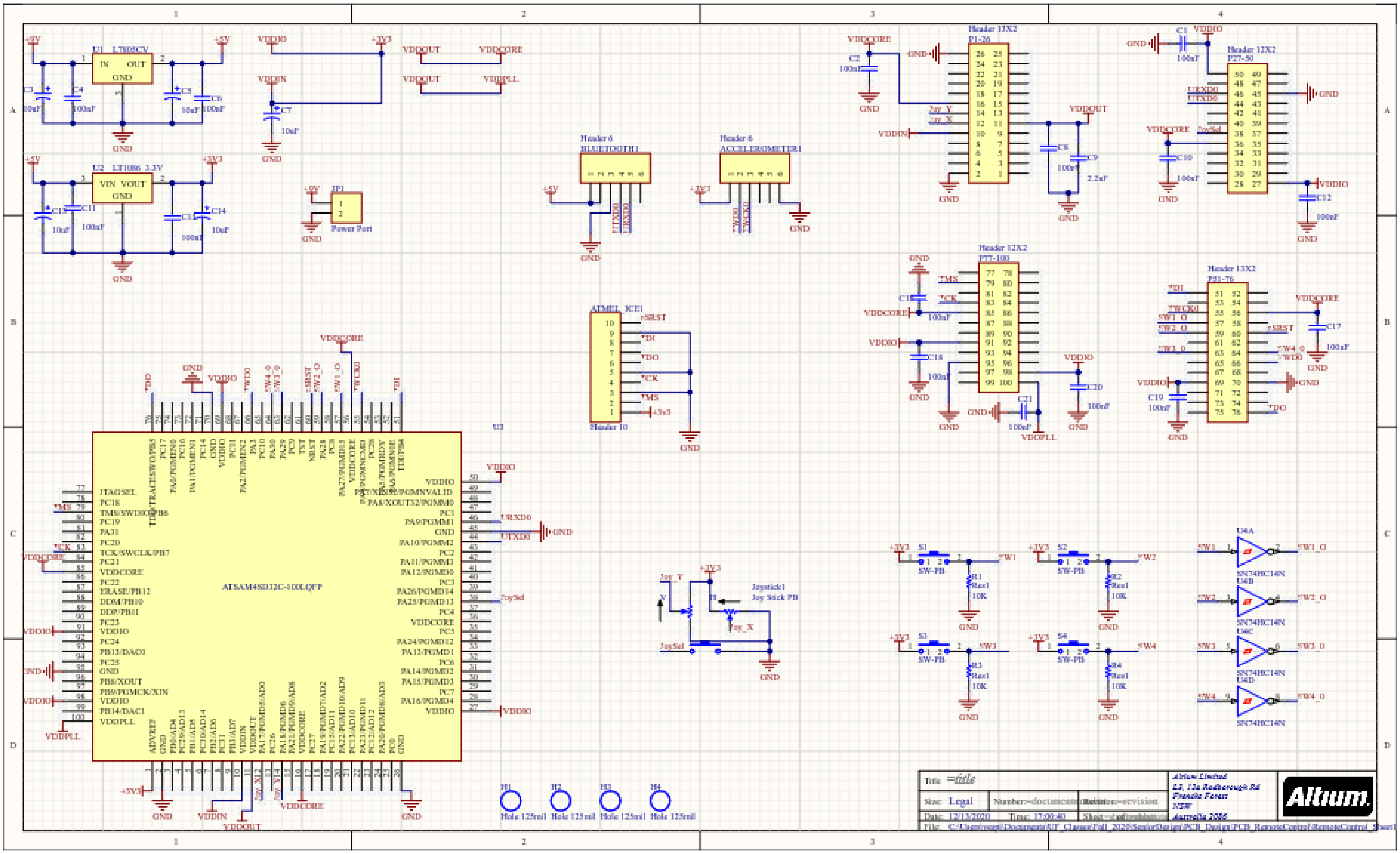
Remote Control PCB



Main System Schematic



Remote Control Schematic



Analog Board Schematic