Team 35 Senior Design I: Initial Document

Alejandro Torroella   Christopher Hubbard
Tyler Drack   Leonardo Achutegui

December 4, 2014

Contents

1 Executive Summary

The Visually Entertaining Smart Prism, or VESP for short, is a revolutionary device that can function as both a personal assistant as well as an entertainment device for all to enjoy. This project aims to give the now classic 3D LED light arrangement project a modern, high-tech spin. Instead of an arrangement of LED lights, the VESP is composed of four LCD screens forming its sides, while the top and bottom consist of plastic casings housing various sensors, inputs and power connectors. Going a step further, the VESP is a portable device and is paired with a charging station equipped with stereo speakers.

The VESP will be an interactive device intended for consumer use. Out of the four LCD screens making up it enclosure, one of the screen will be touch-enabled in order for the consumer to interact with buttons displayed on the screen. In addition, the incorporation of a gyroscope sensor will allow the consumer to further interact with the device using tilt gestures. The VESP also aims to be a "smart" device, having many autonomous conveniences that promote efficiency. Namely, one of these "smart" features is the device ability to automatically adjusting LCD screen brightness to fit the ambient lighting of the room. Another is the devices ability to sense whether there is someone present nearby to observe the device, which will be used to automatically reduce internal processing activity when no one is present to use or view the device.

The VESP project was chosen by a group of four engineers, three being electrical engineers and the other a computer engineer. The project was chosen not only because it posed to be a challenging and unique project, but also because it matched the team's composition of majors. Having mostly electrical engineers on the team, the VESP project will largely be a hardware design project, with some software components to enrich the user experience. The hardware design problems the VESP team will be faced with will include designing a custom made LCD controller that will allow the main processing unit to output to four separate screen by only using one video output. Other hardware challenges include combining various sensors into a singular subsystem, designing a power supply, powering all internal components with a battery, and designing a stereo speaker system. The main software design problems the VESP team will be faced with are designing a custom UI that is easy and intuitive to use, and designing a mix of both utility and entertainment applications.

The VESP device has the potential of becoming a real consumer product in the future. After the completion of the VESP project, the team will consider adding new features and improving upon the design of the device in order to make it a competitive product.

2 Project Description

The following section will be dedicated to explaining the idea behind the VESP, along with outlining the features that that VESP will have and why. In addition, this section will contain the teams motivation and goals for the VESP, as well as the teams desired specifications for the VESP.

2.1 Project Motivation

The initial concept for the VESP began with a vague concept for a new-age lava lamp. The idea was proposed to an audience and three engineers stepped up to the task. Early on, it was the desire of the team to work on a project that everyone had some kind of input on. The team's first couple of meetings consisted purely of brainstorming sessions to come up with ideas for features that the team thought would make a truly unique device. Our team's interests consisted of: computer graphics, sensors, FPGA, Linux, ARM processors, human-computer interaction (HCI), audio, batteries, and wireless communications. From these interests, many great ideas were brought up. The team also looked to previous senior design projects and current consumer products to see what was lacking and what did and didn't work well. From the many features proposed, the team chose the most significant and useful to add to our initial concept, and the VESP was born.

The project itself promised to be fun and interesting, but also quite challenging. A similar project would have been the traditional LED cube project, but the team felt that that project would be too simple and overdone. The team wanted to be proactive during senior design by choosing an ambitious project like the VESP. Unlike the traditional LED project, the VESP will be composed of four LCD screens, instead of the 3D array of LED lights. This greatly increases the complexity of the project and makes choosing the correct hardware all the more crucial. On top of that, the VESP also had considerably more potential than the traditional project. With more processing capabilities, the VESP can render more complex, detailed visuals without sacrificing the user's overall experience. The VESP's unique shape and touch interface introduces the potential for developing creative applications and games. In addition, the VESP's compact and portability factor gives it the opportunity to become a real-life consumer product in the future. It's for these reasons that the team opted to go forth with the VESP project, instead of the traditional LED cube project.

2.2 Project Goals and Objectives

The main goal of this project is to create a working prototype of the VESP, along with a compatible dock charging station. The VESP is to be a portable, user-friendly, entertaining, utility device that is intended to assist its user with day-to-day functioning. The VESP will be composed of four LCD screens, forming the outside of the device. The VESP will be large enough to provide plenty of display area and internal space for components, while also small enough to be easy to carry. The casing of the VESP shall be integrated as seamlessly as possible to the LCD screens, to give it a slick, modern look and feel. The VESP itself will house all of the components necessary for it's own operation, the docking station will only serve as a charging station and an audio speaker accessory.

The VESP must be interactive, intuitive, visually stunning and seamless. The VESP will be touch-enabled and will be the primary method of how the user will input information. To avoid excessive user input or unintentional interaction with the device, not all of the LCD panels will be touch-enabled. In addition, the user must be able to interact with the VESP by changing its orientation, information and graphics displayed on the VESP's screens will adjust according to the device's current orientation. The VESP also must be able to display different information and visuals on each screen independently. On top of all of this, the VESP must be able to run with as little hiccups as possible to not have the user experience suffer. By combining these four qualities, the team will be able to create truly unique user experience for the VESP.

The VESP aims to be a "smart" device, constantly updating its current state based on conditions and displaying useful information to observers when present. The VESP must be able to detect whether a person is in the general area, this way the VESP can know when it is appropriate to display information. The VESP must also be able to automatically adjust the screen brightness of all four LCD screens depending on the amount of light present. In general, the VESP must be aware of current environment conditions and respond accordingly.

The docking station of the VESP must be able to fully charge the VESP's battery in a reasonable amount of time. The docking station itself will not be portable, as it will need to be plugged into an AC outlet. In addition, the docking station will be able to serve as the VESP's speakers for music playback. The hardware volume controls for the speakers must be located on the docking station as well. The charging socket for the VESP will be as tight a fit as possible, in order to the assure proper alignment of any I/O connections.

By the end of senior design, the team would like for the VESP to have all of the above mentioned capabilities. If all our goals for the VESP are properly executed, the device might have the potential of becoming a real consumer product. Beyond that, by completing this project, the team would have gained invaluable hands-on experience with design, development, integration and testing.

2.2.1 Crucial Features

The crucial features listed below must be implemented in order to create the VESP described previously. The remainder of this report will describe how these features will be implemented onto the VESP.

- All of the components of the VESP must fit inside the enclosure of the device and must be kept cool enough to be able to function.

- The VESP will be composed of four LCD screens, one being touch-enabled in order to allow touch input from the user. The other three LCD screens will simply display information, this is to prevent excessive or unintentional touch input from the user.

- The VESP will include an LCD controller which will handle the video output from the FPGA and distribute it to the four LCD screens

- The FPGA for the VESP must be powerful enough to drive the applications to be run on the device, on top of handling sensor data from the MCU and Wi-Fi and Bluetooth data. The applications might include rendering complex 3D objects in real time and so the FPGA must also include a dedicated processor.

- The FPGA for the VESP must support modern operating systems and APIs.

- The applications that will run on the VESP will be a clock app, a scheduling/calendar app, a music app with visualizer, an email app, a reminder/to-do list app, and a phone sync app.

- The I/O connections on the VESP will include a 3.5mm audio (female) connector and a micro-USB (female) connector. The VESP will output sound to the 3.5mm audio jack and will receive power from the micro-USB.

- The VESP must implement a gyroscope and accelerometer in order for the device to utilize its current orientation and change thereof.

- The VESP must implement an ambient light sensor in order for the device to measure the amount of light in the room and adjust the LCD panels' brightness settings accordingly.

- The VESP must implement an IR sensor in order to detect whether there is a person nearby to observe the device. The VESP should go to sleep mode if no person is present for an extended amount of time and wake-up from sleep mode if a person approves the device.

- The VESP must be able to run on its internal battery for an extended amount of time. Ideally, the team would like the VESP to be able to run between five (5) and eight (8) hours on battery.

- The VESP will have both Wi-Fi and Bluetooth wireless communication capabilities. Wi-Fi communication will be used keep the VESP connected to the internet, allowing it to access various information. Bluetooth communication will be mainly used to connect to the user's smartphone device, allowing the VESP to access data from the device.

- The VESP must not weigh more than 4 lbs in order to keep the VESP as portable and comfortable to hold as possible.
- The docking station for the VESP will handle charging the device and will be connected to an A/C outlet.
- The docking station for the VESP must include speakers that the VESP will be able to utilize while attached to the station.

2.2.2 Additional Features

The following features could be added to the VESP at a later revision and are included in this report in order to show the potential of the VESP. These features will not be implemented into the VESP prototype and so will not be discussed in detail in this report.

- An interactive, 3D rendering application. The user will be able to interact with rendered complex 3D models using the VESP's touch and gyroscope capabilities.
- A video camera. The VESP will be able to take pictures and video to keep on the device. Alternatively, the VESP would implement image processing algorithms to be able to detect certain things, such as humans or specific faces.
- Various social media applications. The VESP will be able to update the user on social media updates from various social media sites including Facebook, Twitter, and Instagram.
- Home integration with multiple user profiles. The VESP will be able to control home amenities such as air conditioning, alarm system, video surveillance, and other autonomous home functions. In addition, the VESP will be able to support multiple user accounts in order to serve as a daily assistant to the whole family.

2.3 Project Specifications

In order to fulfill the goals and objectives of the project, the VESP must meet certain hardware and software specifications. These specifications will make the VESP operate as desired and will satisfy the team's goals and objectives mentioned in the previous section. The hardware specifications are outlined by Table 1, while Table 2 outlines the software specifications.

Display	4 x 5-7" LCD Screens
User Interaction	Touch, Tilt, Acceleration
Audio Output	3.5mm Audio Jack (device), Stereo Speakers (dock)
Housing Material	Plastic or Acrylic
Wireless Communications	Wi-Fi and Bluetooth Connectivity
Peripherals	IR sensor, Ambient Light sensor, Accelerometer, Gyroscope
I/O	Micro-USB
Dimensions (LxWxH, mm)	90-120 x 90-120 x 150-180 (device), 100-130 x 100-130 x 40-60 (dock)
Input Voltage	120V AC 60Hz
Power	Lithium Polymer Battery

Table 1: VESP Project Hardware Specifications
Permission here

Operating System	Open Source OS
Supported Music Formats	.mp4, .flac, .mp3, .avi, .wav, and others
Supported External Devices	Android Devices
Applications	Music app, clock/alarm app, weather app, calendar app, notification app and settings app
Programming Languages	C++ or Java

Table 2: VESP Project Software Specifications

If all of these specifications are met and all of the crucial features mentioned in the previous section are implemented, the VESP will be a successful project and will satisfy the requirements for the Senior Design course.

3 Project Research

This section will be dedicated to any relevant research done for the purpose of learning how to construct and design the VESP. The section will start with a discussion on related projects, followed by research on potential components to be used for the VESP, leading up to the final decisions on which components meet all of the requirements and specifications for the VESP.

3.1 Related Projects

The following projects are in some way related to the VESP. Due to the unique and novel nature of the VESP, the projects to be discussed are loosely related. These projects are all taken from senior design courses of previous semesters and all have individual features and functionality which the VESP tries to not just combine, but also improve upon.

3.1.1 Dynamic Animation Cube II

The Dynamic Animation Cube II, made by Group 5 of the Fall 2012/Spring 2013 senior design period, was a traditional 3D LED cube project that aimed to be user interactive. The LED arrangement was 16x16x16, making the device extremely large and impractical to carry around. The user could interact with the device by using a repurposed Wii Nunchuck controller to shifts between games, a visualizer and a main menu.

The VESP aims to improve upon the Dynamic Animation Cube II by making the device more portable, useful, and user friendly. Also, the visuals produced by the VESP vastly outclass the visuals that the Dynamic Animation Cube II is capable of. This is possible due to the design decision to use LCD panels instead of LED lights. The shift from LED to LCD allows for more complex visuals to be rendered, and sharper, crisper, more colorful images can be displayed. Being made of only four LCD panels, instead of the huge arrangement of LEDs, gives the VESP a smaller form-factor advantage over the Dynamic Animation Cube II. By being smaller, it allows for the user to more easily interact with the device, and promotes user interaction.

Other than the Dynamic Animation Cube II's purpose as a purely visually entertainment device, the VESP is a completely different project and requires an entirely different approach to be realized. Where as the Dynamic Animation Cube II used LED drivers to control all the individual LEDs, the VESP will use an LCD controller to drive its four LCD displays using a single video data signal from the FPGA.

3.1.2 GameQube

The GameQube was made by Group 33 of the Fall 2013/Spring 2014 senior design period. The project was another traditional 3D LED cube project, but it separated itself from the rest by providing a better user experience and functionality to the LED cube.The GameQube, unlike other 3D LED cube projects, was able to run simple games, like Snake, and display them in a three-dimensional space, thereby creating a unique user experience. The GameQube also featured Bluetooth connectivity to a game controller for the user interaction aspect. The LED arrangement was 10x10x10, making it smaller than the Dynamic Animation Cube II, but still impractical to carry around.

Like with the Dynamic Animation Cube II, the VESP aims to improve on the features and uses of the GameQube. The VESP will use a single touch-enabled LCD screen to control the device, along with motion gestures, courtesy of the gyroscope/accelerometer combo inside the VESP. The shift to a touch/motion gesture interface makes the VESP more interactive and generally more interesting to use than the GameQube controller. In addition, all the previously mentioned improvements over the Dynamic Animation Cube II also apply here.

3.1.3 Smart Mirror

The Smart Mirror was made by Group 10 of the Fall 2013/Spring 2014 senior design period. This project is particularly interesting and is probably the most similar to the VESP in terms of intentional use and function. The Smart Mirror is a device that's integrated into a standard bathroom mirror that can display various kinds of useful information to the user. The Smart Mirror is capable of showing the user weather updates, news, a to-do list, a calendar scheduler, and a clock, as well as music playback. The Smart Mirror is controlled through gestures using an infrared/camera combo controller called "Leap Motion" allowing for the user to not have to touch the mirror to interact with it. On top of the gesture control, the Smart Mirror can also take in voice commands as user input. Like the VESP the Smart Mirror aims to assist the user by being integrated in the user's normal at-home patterns and rituals. In addition, both the Smart Mirror and the VESP are intended to be kept at home, to be used when the occasion arrises.

However, unlike the Smart Mirror, the VESP is able to be easily carried around the house if the need arises, or stay in one location. The VESP is both small enough to act as a portable device, and big enough to pass as a useful adornment for the home. In addition, the Smart Mirror must take into consideration the humidity level of the bathroom it's installed in otherwise run the risk of damaging the internal hardware components, the VESP doesn't have to concern itself with this. Another difference between the two projects are the software applications and expandability. The VESP will have a social media app, a calendar app, an email app, an interactive renderer app, and a music playback app, with the ability to install additional apps. The Smart Mirror on the other hand didn't have this design decision in mind and so the apps that are on the Smart Mirror are the apps you're stuck with.

3.2 Component Research

The following section will be dedicated to researching specific, available devices that will potentially be used to build the VESP. The requirements and specifications outlined earlier in the this report will considered when choosing the final components of the VESP. Other factors that will be considered in this section will be cost, energy consumption, and performance.

3.2.1 Power adapter

From the first moment, the group needed to build an AC to DC power supply because we are going to use high voltage AC which is the main electricity and step down to low voltage for the electronic circuits that will operate the device. The group found a

Power Supply System suitable for our device. We are going to use a step down power supply that will convert 230V AC to a regulated 12V DC. The whole power system will include a transformer, a rectifier, a capacitor, and a regulator.

The transformer will help step down from the main high voltage AC to low voltage AC. Transformers only work with AC and this is the reason why the step down output is in AC as well. The transformer has two parallel coils, with the left one being the input coil which is called the 'primary' and the right one which is the output coil called the 'secondary' as seen in Figure 1. There is no connection between both coils, instead they are linked by an alternating magnetic field that is located in the soft-iron core of the transformer. Also, the 'turn ratio' which is the ratio of the number on each coil determines the ratio of the voltage. For example, the step down voltage transformer will have a large number of turns on the input coil which is connected to the 230V AC, and a small number of turns on the output coil which will give a low voltage of 6V AC.

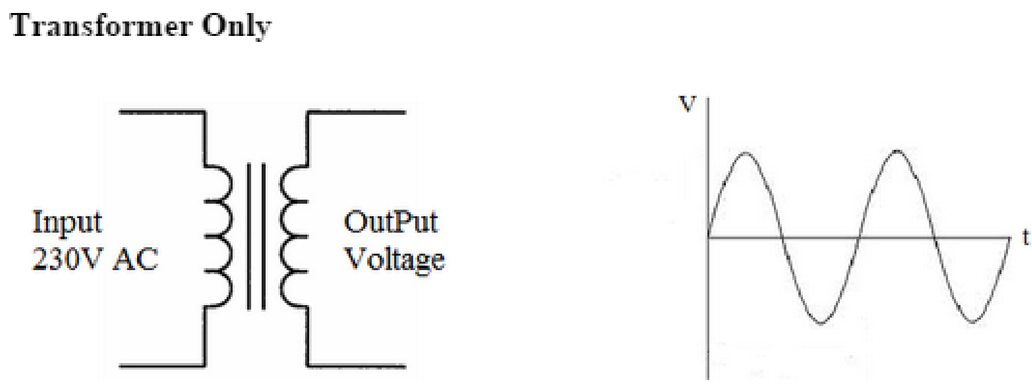


Figure 1: Transformer symbol and its output signal - stepping down voltage AC

The rectifier will be use to convert the alternating current (AC) to direct current (DC). The reason why is because most of the electronics and batteries that will be used during this project run through direct current. Rectifier circuits are usually single phase which are commonly used for power supplies for domestic equipment, or three phase which are used mostly for industrial and high power applications. In this project I will use a single phase rectifier, but there are two famous types: half wave and full wave rectification.

A half wave rectifier usually uses one diode connected to either the positive or negative side of transformer output. That means that the diode will be forward biased in the positive half cycle and reversed biased in the negative half cycle. The output will be available during the positive or negative depending on which side you connect the diode, while the other side will be blocked because only one half of the input waveform reaches the output. I will not use this type of rectifier because is very inefficient when is used for power transferring.

On the other hand, we will be using a full wave rectifier in this project. The reason we are using this rectifier is because it converts the whole input waveform to one constant polarity at its output, which is either positive or negative as shown in Figure 2. This means that it converts both polarities of the input waveform into direct current, which is more efficient when it comes to power transferring. The bridge rectifier, which is a full wave rectifier uses four individual diodes in which alternate pair of diodes conduct, changing over the connections so the alternating directions of AC are converted to DC. Also, bridge rectifiers are rated by the maximum current they can pass and withstand at least three times the supply RMS voltage, meaning it can withstand peak voltages.

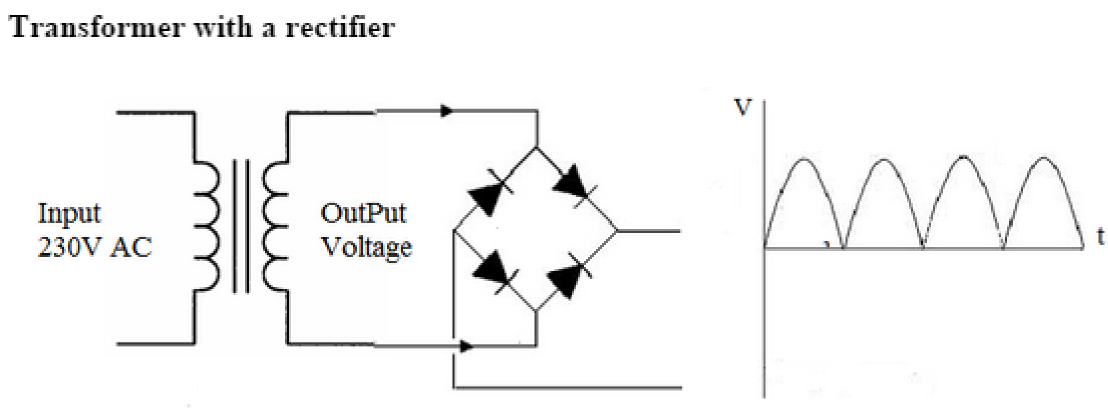


Figure 2: Rectifier converts AC to DC, while DC still in wave form.

The capacitor as you can see in Figure 3, also known as the reservoir capacitor will be used to smooth or even out the fluctuations in the signal. The capacitor will be connected across the DC supply as a reservoir, providing current to the output when the DC voltage from the rectifier is falling. The capacitor is not perfect due to its voltage falling when it discharges, giving the shape of small waves. This is called ripple voltage. In this case, the larger the capacitor the smaller the ripple it will give.

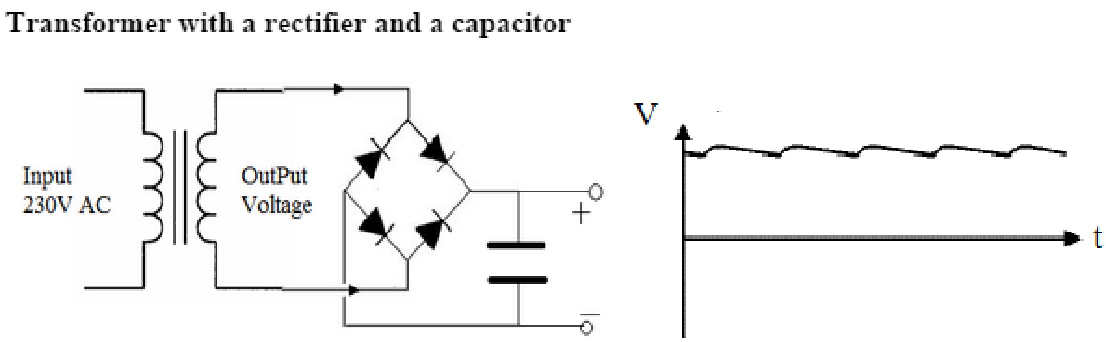


Figure 3: Reservoir capacitor smooth's the DC by converting into smalls ripples cause by charging and discharging of the capacitor.

In order to make the output DC very smooth with no ripple and suitable for our electronic circuits, I will be adding a regulator. A voltage regulator as seen in Figure 4 is designed to keep a steady voltage level regardless of changes are made to its input or load conditions. There are two types of regulators: linear and switching regulators. In this project the group will be using a linear regulator due to the fact that it is more efficient than the switching regulator when it comes to small load currents and also no ripple at the output, which is why the group will be using during this project.

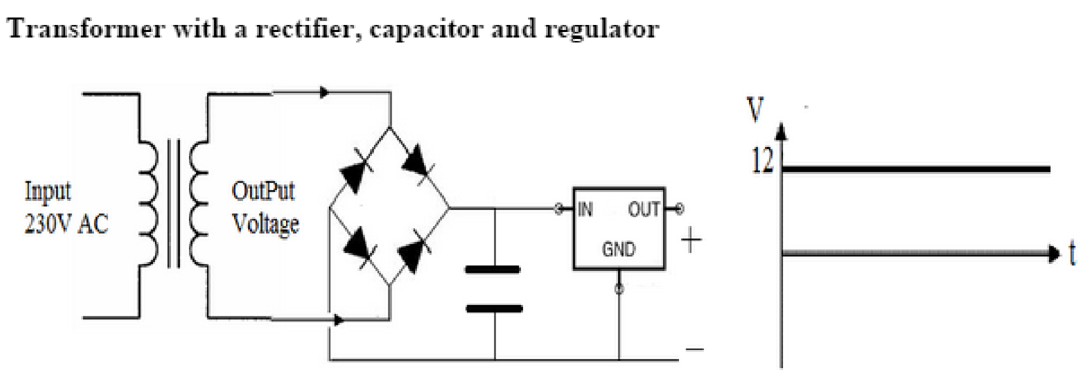


Figure 4: The regulator removes the ripple making a fixed voltage DC output

Resistors, Inductors, Capacitors, and Diode

VESP will definitely not be able to be designed without some basic components in the electric circuitry industry. Resistor will be used to keep control of high currents that need to be step down for a specific device. Inductors are electronic filters that separates signals of different frequencies and are going to help in the conversion of currents and signals. Capacitors is another electrical filter that will be used to decouple some of the integrated circuits (IC) from each other. Diodes is a two terminal electronic component that has low resistant in one side while having high resistant in the other. Diodes will be used in the design of the rectifier.

Decoupling is the prevention of undesired transfer of energy from one IC to another. We use the capacitor to decouple because the power supply will definitely be slow in the transition to provide constant voltage. Due to the change of output of an IC and its power requirements, the capacitor will be located at the power input of the chip until it's discharged and it will act as a temporary power source while the power supply reaches its demand..

3.2.2 LCD screens

One of the most important components for the project is the LCD screens. On top of their functionality, they determine the volume of the "cube" as they make up four walls of the housing. If the screens are too small, the internal hardware may be unable to fit. The decision for which LCD screens to use will consider the cost, the size, and the functionality. The first option to consider would be a 5.0" 40-PIN 800x480 TFT display from adafruit.com. Without touchscreen capabilities, the price will only be $29.95 per screen. Since three out of the four screens will not need touchscreen capabilities, this is an excellent choice as far as cost goes. The choice for the final screen would be the touchscreen version of this LCD screen which is only an additional $10. This keeps the dimensions of the screens uniform, which is very important since they decide the dimensions of the "cube".

This LCD screen has an 800x480 pixel display with 24-bit color capability and an LED backlight. It is connected to a processor via a 40 pin connection. The display is supposed to be constantly refreshed at 60Hz and will require dedicated hardware or a powerful processor to support it, as a typical small microcontroller cannot handle all the raw data. The backlight will require a constant-current mode boost converter than can go as high as 24V. The screen itself operates on a voltage supply of 3V-3.6V. The 5" refers to the diagonal length, but dimensions of the screen are 121mm x 76mm for the touchscreen version and 120mm x 75mm for the non-touchscreen version which are more important for determining the estimated size and volume that will be available internally in the "cube".

However, 5 inch displays making up the four walls may not provide enough room inside the "cube" for all the other components. Adafruit.com also sells a 7.0" version of the screen with all the same capabilities. The non-touchscreen version sells for $69.95 and also includes a mini driver that includes a USB cable connection for power supply, an HDMI input connection, and a wired PCB that has buttons for the option of brightness, contrast, and color controls. This attached driver will help ease the task of connecting all four screens to a single processor. To run the display is as simple as connecting the USB power cable to a 5V power supply that can provide 500mA and connecting an HDMI cable to the input. The range of its voltage supply is 5V-24V, which starts to be a little high for the scope of the project. The touchscreen version of the 7” screen will cost an addition $14.95. Although not as strong of an option price-wise, it greatly increases to dimensions of the “cube” allowing a lot more room for the internal components. The dimensions of the 7” display are 154mm x 86mm.

Another option for a 7” LCD screen was found on Olimex.com (A13-LCD7). Again, these LCD screens have a 480x800 pixel display, a DC-DC connection backlight, and 24-bit color capability. These screens are also connected to a processor via a 40 pin connector. Other notable features include an anti-glare surface and test pads for easy diagnostics. The non-touchscreen version is $57.42 and the touchscreen version is $70.17. This screen runs on a voltage supply range of 9.3V-10.5V. The dimensions of these screens are 165mm x 100mm. The Olimex.com LCD screens are not only cheaper than the Adafruit.com versions, but their dimensions are slightly bigger which will provide extra internal room in the “cube”. Additionally, Olimex.com provides several options for processors directly compatible with the LCD screens, potentially simplifying future roadblocks of components that are difficult to integrate together.
Buydisplay.com offers a very inexpensive 7” LCD option for only $30.84. This screen is touch, 800x480 display with 24 bit color and a backlight. The total dimensions are 164.9mm x 100mm which provides basically the same amount of volume as the Olimex.com option. This screen also comes in a package purchase with lot of additional components, such as a video+VGA driver board, a power cable, a video cable, a VGA cable, and a keyboard cable all of which have corresponding terminals on the driver. The touch screen component is an additional $7.54. The downfall of this option is it operates on a supply voltage of 12V, which starts to get a little high for the scope of the project.

Another option from buydisplay.com (part number ER-TFT070-4) which has requires a supply voltage of only 3.3V, is another 7” display with optional touch screen panel. This component is just the LCD (no on board MCU) so the cost is significantly cheaper than any other option at $17.56, and the voltage supply is far more ideal for the project compared to the previous option from this website. This screen is also an 800 x 480 pixel display with 24 bit color and a backlight. Its overall dimensions are 164.9mm x 100mm with the visual area being 156.9mm x 89mm. It is connected using an FPC (Flexible Printed Circuit) Connector which will be helpful for the flexibility necessary for all the components connecting together internally in the VESP. Additionally, the capacitive touch screen expansion is only $7.54. Buydisplay.com provides a series of demo codes for each of the interfacing options which will ease the process of integrating the LCD screens to the rest of the project. This LCD screen option may be the strongest choice for the project; it meets the low voltage preference, it provides a high amount of internal volume in the “cube” and is significantly cheaper than all the previous options.

Table 3 will be used to quickly analyze the specs of each LCD screen in comparison. The Potential Volume column refers to the dimensions of the “cube” constructed by using the screens as the the four walls, and the corresponding internal volume that will be provided as space for all the internal components.

LCD Screen	Connection Type	Voltage Supply	Screen Dimensions (in mm)	Potential Volume (mm^3)
Adafruit 5"	40-PIN	3V-3.6V	120x75	675K
Adafruit 7"	HDMI	5V-24V	154x86	1,138K
Olimex 7"	40-PIN	9.3V-10.5V	165x100	1,650K
Buydisplay 1	Video/VGA	12V	164.9x100	1,649K
Buydisplay 2	FPC connector	3.3V	164.9x100	1,649K

Table 3: LCD Screen Specification Data Sheet

The next figure, Table 4 will be used to reference the cost comparison of all the units. The Total Cost column refers to the cost of buying three non-touchscreen LCD screens and one touchscreen LCD screen. If the non-touchscreen column is mark N/A, then the total cost refers to the cost of buying four touchscreen LCD screens.

LCD Screen	Non-Touch Cost	Touchscreen Cost	Total Cost
Adafruit 5"	$29.95	$39.95	$129.80
Adafruit 7"	$69.95	$84.90	$294.75
Olimex 7"	$57.42	$70.17	$242.43
Buydisplay 1	$30.84	$38.38	$130.9
Buydisplay 2	$17.56	$25.10	$77.78

Table 4: LCD Screen Cost Data Sheet

In conclusion, if further research determines that the internal components will not need a lot of room, the 5" Adafruit.com version may be the strongest choice, as the cost is significantly cheaper. If more room is necessary, the processor choice may help decide between the several 7" options. As far as cost and size go, the buydisplay.com LCD screens are preferable, however, the choice in processor may make the Olimex.com 7" version more viable as they also sell compatible processors. The research into processors will help in the decision of which screen to choose.

3.2.3 Microcontroller

The microcontroller board is an essential component for the project. It will be communicating with the sensors, the FPGA, and possibly the Wi-Fi module and the Bluetooth module, depending on whether or not it is decided to connect the two modules to the FPGA. The microcontroller must have compatible inputs for each of the components connecting to it. This will likely be one ambient light sensor, at least one proximity/motion sensor, one gyroscope/accelerometer, the Wi-Fi module, the Bluetooth module, and the connection to the FPGA for a total of six possible connections (unless the Wi-Fi and Bluetooth are connected to the FPGA).

The first microcontroller to be researched will be the MSP430 from Texas Instruments. The project group has already acquired an MSP-EXP430G2 Launchpad prior to the designation of the project, so this choice would save on costs. This Launchpad comes with both an MSP430G2452 and an MSP430G2553. Some key features of the microcontroller are it has 16Mhz speed, 16KB of flash memory, providing plenty of room for writing the code to interface with all the sensors, 512B of RAM, an 8 channel 10-bit ADC, which will be useful for any sensors that create an analog output instead of a digital one, a comparator, two 16-bit timers, up to 1 I2C, 2 SPI, and 1 UART interfacing buses, which can be used to connect to all the different components, and has low power operation which makes it great for battery-operated applications such as when the "cube" is removed from the dock. This Launchpad comes with a 20 pin DIP socket for easy bread-boarding/prototyping and also has an option for a 20 pin boost pack. TI also offers free downloadable software to easily program the device. The Launchpad runs on a voltage supply maximum of 3.6V, which falls in the same range as a lot of the sensors that have been researched. The dimensions of this board are 68mm x 51mm x 13mm which must be considered as the space inside the "cube" will be limited. The Launchpad is a strong choice for the project as it is cost efficient, easily programmable, and should be able to support all the necessary connections to sensors and other components.

The next microcontroller board to be considered is the Tiva C Series TM4C LaunchPad (part number EK-TM4C123GXL) sold on TI.com for $12.99. This Launchpad's key features include 80 MHz speed, 256KB flash memory, which is certainly enough for handling all the programming and information for the sensors and modules connected, 2 12 channel 12-bit ADCs which are more than enough pins for receiving any analog sensor information, 3 analog comparators, which may be used to combine sensor outputs to determine when a user is near the device, four SPI/SSI, four I2C, and eight UART buses for interfacing options, more than enough to connect and control all the possible sensors, and a low-power hibernation mode which could be used to help increase the battery life. This board has 43 GPIO (general purpose I/O) pins and is compatible with a 40-pin booster pack, which adds a lot of extra support for the sensors and wireless modules. TI.com offers an array free software solutions for programming the board. This board runs on a supply voltage range of 4.75V to 5.25V powered through a Micro USB cable that comes with the kit. The dimensions are 50mm x 57.15mm x 10.795 mm (L x W x H) which must be kept in mind as this component must fit inside the "cube" portion of the VESP whose dimension will be determined by the LCD screens selected. This Launchpad is a good choice for the project and may actually overshoot the capabilities it is needed for, however it is relatively low priced so overkill may not be a problem if the "cube" has the spare room and the price is within the budget.

The next option to be considered is the Arduino Uno. This board has 12 digital I/O pins, 6 analog inputs, a 16 MHz resonator, a USB connection, a power jack, and ICSP header for programming and interfacing, and a reset button. This board operates at a voltage supply of 5V, but has a supply range of 7V-12V. The board has 32

KB flash memory which is, again, plenty of room to program the board to integrate with the sensors and wireless modules. The 6 analog pins and 14 digital pins should fit all the possible sensor connections to read their outputs, and the board supports SPI and serial communication to interface. Arduino offers software downloads for programming the microcontroller and has demo code which the project group can utilize to quickly learn how to write to the board. The dimensions are 68.58mm x 53.34mm which also must be considered to when determining how much room will be available inside the "cube". This board should be able to handle all the components that need to be controlled by it, however, it costs $24.99 which makes doesn't make it as viable of a choice as the cheaper microcontrollers as they are just as capable for the scope of the project.

Key features and specifications can be easily referenced and compared using the following figure (Table 5). Dimensions and costs can be quickly looked up and compared using Figure 6. Although the Arduino Uno may seem too expensive compared to the other options, its size may make it a stronger choice; further research will determine this.

MSP430	Tiva	Arduino
16MHz	80MHz	16MHz
16KB Flash	256KB Flash	32KB Flash
512B RAM	32KB RAM	2KB SRAM
8ch 10-bit ADC	2-KB EEPROM	1-KB EEPROM
Comparator	On-chip ROM with drivers and boot loaders	14 Digital I/O Pins
2 16-bit Timers	2x 12ch 12-bit ADCs (1 MSPS)	6 Analog Pins
Up to 1 I2C, 2 SPI, 1 UART	16x Motion PWM channels	ISCP Header
20 PIN DIP	24x Timer/Capture/Compare/PWMs	
	3x Analog comparators	
	4x SPI/SSI, 4x I2C, 8x UART	
	USB Host/Device/OTG	
	2x CAN	
	Low-power hibernation mode	
	43x GPIO pins	

Table 5: Microcontroller Hardware Specification Comparison Table

Microcontroller	Voltage Supply	Dimensions (mm)	Cost
MSP430	3.6V	68 x 51 x 13	Already Acquired
Tiva	4.75V-5.25V	50 x 57.15 x 10.795	$12.99
Arduino	7V-12V (power jack)/ 5V (USB)	68.58 x 53.34	$24.99

Table 6: Microcontroller Voltage/Cost/Dimension Comparison Table

3.2.4 Wireless Communication

Like many Senior Design projects seen over past years the VESP will contain wireless communications as a specification. To be more specific the group wanted the VESP to be able to have connectivity to the internet as well as communicate with a Smartphone app for many reasons, the most notable being the transfer of music data. In choosing to use wireless communications in the VESP the group looked at several different types of wireless communication methods and compared them to see which of these communication methods would serve as the best for the VESP.

Wi-Fi

The first standard of wireless communication that the group looked at is Wi-Fi. Wi-Fi is very commonly used everywhere and enables almost any device capable of integrating with a Wi-Fi module to access the internet. Wi-Fi was looked at for the VESP for the main purpose that it was created, as a means to connect our small VESP to the internet. By connecting a wireless adapter such as a modem into a network multiple devices can connect to the same wireless network. Wi-Fi is capable of various throughput data rates depending on whatever wireless standard is used in the device. The Wi-Fi "802.11a" standard is capable of a data rate of up to 54 Mbps and has less interference due to the method that incoming radio waves are handled to the device. Other standards include the more cost efficient "802.11b" which is slower (11 Mbps) and operates at a different operating frequency (2.4 GHz opposed to 5 GHz), the "802.11g" standard which combines the speed and efficiency of the "802.11a" standard with the frequency and ease of use of the "802.11b" standard. The last commonly seen standard which was looked at for the VESP was the "802.11n" which is the widest used standard and as it is backwards compatible with "802.11a, b and g" standards which helps drive its popularity. The Wi-Fi component for this project looked at several modules using the "802.11n" standard to encompass our need for high data rate and efficiency. In short, the positive of Wi-Fi is that it is widely used in many standards and is the easiest form of communication for internet connection in the VESP.

Bluetooth

The second standard of wireless communication that the group looked at is the wireless standard Bluetooth, or IEEE Standard 802.15. Bluetooth is meant as a secure, low-power alternative for wiring and cables and can be found in almost any device today ranging from speakers, to Smartphones to medical and automotive equipment. Bluetooth connects the devices that previously would require a cable connection and does so through the use of 2.4 GHz radio waves making the process wireless. Bluetooth has a range of around 10 meters maximum based on the Bluetooth core standards and is able to network itself with other devices that contain Bluetooth communication abilities in a close range, wireless communication network. Bluetooth power consumption varies depending on which particular model of Bluetooth (1.0, 2.0 etc.) is used however, all present Bluetooth standards run at around 2.5 mW of power consumption with some standards bringing that much lower. This close range communication and data transfer ability found in the Bluetooth standard makes it a good choice for a project or product that seeks to have communication over the span of space such as a bedroom, which is why it is looked at by the group for the VESP.

ZigBee

The final wireless communication standard that was looked at for the VESP is the IEEE 802.15.4 standard, that is known as ZigBee. As can be seen in ZigBee's IEEE Standard, it has many similarities to the Bluetooth standard, close range, security and low power consumption. ZigBee, however, operates as several different frequencies: 2.4 GHz, the same as Bluetooth, as well as 868 and 915 MHz which means there is no interference with Bluetooth in those frequencies. ZigBee finds its uses in home security, home automation as well as specific items such as smoke detection lighting control. ZigBee has a much lower data rate, defined by the IEEE standard as 250 kbps, when compared to Wi-Fi or Bluetooth and is suited more for sensor reading and other slower data transmissions. ZigBee draws many similarities and differences when compared to Bluetooth which is why the group looked at the two standards separately, to determine which if either would be better fit for the VESP.

Comparison and Selection

As a result of the group's research into wireless communications as well as the specifications and requirements that were laid out in past group meetings when coming up with the VESP, the group decided to implement two different standards of wireless communication. The first of these will be the Wi-Fi standard, Wi-Fi was appealing to us for its ability to connection to the internet at a fast speed, allow the VESP to perform the same as an internet browser as something like a home PC or a modern-day Smartphone. As Wi-Fi is widespread in today's world it will be trivial to locate and find information and assistance in the implementation of Wi-Fi into the VESP, making it an easier task but still one with great importance to the project. The second standard that the group chose was made after considering the major pros and cons of Bluetooth and ZigBee, as seen below in Table 7 In the end, the group decided to implement Bluetooth as our second wireless communication standard. The main factor in the group's decision was that despite its lower wireless communication reach, it is still sufficient for the VESP and at a much higher data rate when compared to ZigBee and that Bluetooth is able to do many of the tasks that are expected of the VESP that ZigBee is unable to perform.

Bluetooth	ZigBee
Pros	
More Widespread	Extended Range
The common tasks of Bluetooth align more with what our group wishes to do.	Different operating frequencies
More data rate	Lower power consumption
Cons	
Significantly less range	Significantly slower data rate
Sharing frequency with Wi-fi may cause interference unintentionally	More specific applications that aren't a part of the VESP

Table 7: ZigBee and Bluetooth Pros and Cons.

Possible Ramifications

As the VESP is a small electronic device that incorporates both Bluetooth and Wi-Fi communications the group has the potential to run into problems as both wireless communication methods run using the 2.4 GHz frequency band. This creates potential interference between the two separate wireless networks and may impede or completely disable one function of the VESP's wireless communications while allowing the other to function as normal. Both wireless standards have to be functioning enough for wireless communication within the VESP is seamless. Bluetooth manages this potential interference using a method called adaptive frequency hopping, where the Bluetooth communication would jump on one of 79 different 1 MHz bandwidth allowing Bluetooth to avoid potential conflicts of frequency bands. Wi-Fi communication on the other hand, uses almost one third of the 2.4 GHz spectrum meaning that Bluetooth jumps around the Wi-Fi and sometimes directly on the same frequency. This conflict of frequencies causes issues as Wi-Fi receivers are unable to receive data from the transmitter as the information has been blocked and must with until the Bluetooth messages are sent before transmitting its data. The solution for us is to either trust that the adaptive frequency hopping in the Bluetooth module is able to keep both communication protocols operating as much as it needs to. A second solution is to remove the frequency conflicts by using a Wi-Fi that uses the "802.11a" protocol which uses the 5 GHz frequency band, there is no longer any frequency conflict; however, the resulting Wi-Fi communications will become much slower as a result of the much slower data rate of the "802.11a" protocol. Alternatively another option is to program the Bluetooth and Wi-Fi to operate at alternating intervals so that both communication methods are not active at the same time. Regardless of how small or how large the possible interference is, this is still an issue that must be addressed by the group as we continue to develop the Wireless communication abilities on the VESP.

3.3 Wi-Fi

Wi-Fi has become a modern day certainty. In almost every device today, Wi-Fi capabilities can be found: inside your phone, your car, gaming systems, tablets and of course home computers and laptops. The basic purpose of it is the same everywhere, wireless connection and communication to the internet for boundless information. Naturally, the VESP project will also incorporate wireless communications. The group has looked at several different Wi-Fi modules from Texas Instruments in particular, this was done to allow easier compatibility with the MSP430 MCU and to allow for a simpler attachment for the two devices. In the following paragraphs are some of the Wi-Fi modules that were researched which will lead you into our group's ultimate decision for the Wi-Fi module for the VESP.

The first Wi-Fi module that was looked into was the CC110L Air Boosterpack from Texas Instruments. This module looked appealing for one initial reason, price. Thanks to the TI workshops held earlier in Fall 2014 one of the group members received two free Boosterpacks alongside a TIVA MCU, having the Wi-Fi module given to us for free allowed us to have forty dollars of overhead budget and gave us the opportunity to begin programming it now. Other benefits include a wide voltage range needed for operation, from -0.3 to 3.9 V making it easier to pull the power from the main power rail without worrying about getting a correct specific voltage. However, there are problems; the main problems come from its data rate, which is relatively slow at around 600 Kbps, when other dedicated Wi-Fi modules can run at over 12 Mbps. Another issue came in regarding the CC110L is that it doesn't allow true Wi-Fi internet connection, it acts as a transceiver meaning two of them would need to be in place in order to have communications established. As a result of the speed and the limited functionality in wireless communications the group decided to browse through TI's other wireless communication solutions.

The second Wi-Fi module that the group looked at for the needs of the VESP was the CC3000 SimpleLink Wi-Fi module. The 3000 series of wireless network processing from TI all share the common traits of a minimum memory intrusion on their host MCU as well as the integrated protocols needed to connect to the internet, meaning they all will allow true wireless connectivity to the internet which fits the specifications of the project. Within the CC3000, power management is handled for the group meaning all the group has to do is apply the required voltage to operate the processor, integration with the MSP430 host is done through SPI pins and the module itself can be stacked onto the pins allowing the group to save space in our already constrained space budget. The CC3000 compared to the CC110L is meant more for internet connection which is why it is preferable to the group over the CC110L; however, as we couldn't pick one particular network processor without looking at newer and more powerful ones such as the CC3100 and CC3200 series of TI wireless network processors.

The third Wi-Fi module was the CC3100BOOST, an improved version of the CC3000 with faster data rates and the same power management subsystems contained inside of it. The CC3100 maintains the industry standard wireless protocol stack that is needed for internet and wireless connectivity and it keeps the same simple integration to the MSP430 that the CC110L possesses. Similar to the CC3000 the host MSP430 MCU can communicate with the CC3100 through SPI pins; however, communication between the two devices can also be done through the UART terminal onboard the MCU. The CC3100 is readily available through TI's website and only costs 20 dollars or more which would give us an MSP430F5xx series Launchpad to test with. Given many of the positive traits that the CC3100 carried with it from the CC3000 and CC110L modules as well as its generous pricing we decided to check one last entry in TI's line of wireless network processors before deciding which would be the best for the VESP.

The final Wi-Fi module that our group looked at for use within the VESP is the CC3200 module. Compared to the CC3100 module the CC3200 has an onboard MCU, which means that the Wi-Fi component of the VESP can all be handled by a single IC without needing to connect directly to the main MSP430 board. This saves us effort in interfacing the Wi-Fi module of the VESP to the MCU, but adds in more interfacing in order to connect CC3200 MCU to the FPGA, this is an example of a trade-off that the group has to decide upon, smaller space on the chip size but more space needed for integration or more chip space but less space for integration and interfacing. Compared to the CC3100 there are few differences with the exception of the onboard Cortex MCU meaning if the CC3100 or CC3200 were chosen they would behave very similarly thanks to the almost identical processor architecture. In addition owing to the addition of the MCU the CC3200 is pricier at 30 dollars, still below the budget projection labeled below.

Since we had a total of four separate Wi-Fi capable components to chose from our decision was based on the power input that we needed to operate the device, the price of the device and the method of integration with the projected use of the MSP430 MCU as well as what is naturally expected from an internet enabled device, more speed and security. This information is listed in Table **??** below.

Option	Voltage Needed	Data Rate	Price	Integration
CC110L	-0.3V - 3.9V	600 Kbps	Free (Already own)	SPI, can be attached directly to MCU
CC3000	2.1V - 3.6V or regulated 1.85V	Up to 11 Mbps	Unknown	SPI, can be attached directly to MCU
CC3100	2.1V - 3.6V or regulated 1.85V	Up to 16 Mbps	$20.00 for just module	SPI and UART, can be attached directly to MCU
CC3200	2.1V - 3.6V or regulated 1.85V	Up to 16 Mbps	$30.00 for just module	Independent board, no need for MCU interfacing

Table 8: List of Wi-Fi module candidates

3.3.1 Bluetooth

In today's world wireless communication is handled by many different wireless standards. One is 802.11, commonly called Wi-Fi and the other is 802.15.1 which is the standard for Bluetooth, which uses the 2.4 GHz frequency to communicate. The detachable cube part of the VESP contains a Bluetooth module inside of it in order for short range wireless communication, in particular with a Smartphone and a possible Smartphone app.

In choosing an appropriate module for use in our project we looked at several variables, including size, cost and voltage needed to operate. Some factors such as operating temperature are of no concern as we will most likely never exceed temperatures of -40°C or 85°C as is common on many Bluetooth modules. In addition the data rate of the module is only needed to be fast enough to transmit data from the app, which includes music and may include pre-programmed light shows and email. Table 9 below lists several Bluetooth modules that were looked at as options for our project, included in our list are several products from Texas Instruments, which was selected as a possible Bluetooth vendor due to the group's decision to use the TI MSP430 microcontroller as our MCU for the VESP, as well as other companies that offered their own Bluetooth radio solutions. Some of the different types of Bluetooth solutions that were looked at during the research phase of the project include Bluetooth controllers and Bluetooth systems on a chip.

Option	CC2564MODN - Bluetooth HCI Module	CC2541-SimpleLink Bluetooth Smart and Proprietary Wireless MCU	Unistone PBA 31308	Bluegigas BLE113 Bluetooth Smart Module
Manufacturer	Texas Instruments	Texas Instruments	Intel	Bluegigas
Unit Price	6.65	5.44	10.20	13.95
Data Rate	4000 kbps	2000 kbps	3000 kbps	2000 kbps
Protocol	Bluetooth 4.0 Low Energy	Bluetooth 4.0 Low Energy	Bluetooth 2.1 + EDR	Bluetooth 4.0
Supply Voltage	2.2-4.8 V	2.0-3.6 V	2.9-4.1 V	2.0-3.6 V
Data Interface	UART	UART	UART	UART
Size	7x7 mm^2	6x6 mm^2	11.6 x 8.7 x 1.8 mm^3	9.15x15.75x2.1 mm^3

Table 9: Candidate Bluetooth modules.

The first Bluetooth solution that the group looked into for the VESP was the Texas Instruments CC2564MODN Host Controller Interface (HCI) module. It is priced at $6.65 per unit and is able to support a data rate of up to 4 Mbps and a network of up to seven devices while I operating temperature range of -20°C to 70°C , though in practice the number of devices that the VESP needs is much lower than seven. The CC2564 finds uses in applications including toys, mobile devices, audio solutions among others making the Bluetooth requirement of phone communication and music playing for the VESP possible with one chip. The CC2564 is able to use any MSP430 or ARM Cortex-M3 or Cortex-M4 MCU to perform its operations making it more appealing in the event the group decides on the MSP430 MCU. This device operates using the Bluetooth 4.1 standard, which is the most modern and up-to-date version of Bluetooth and manages its internal power using the modules onboard power management subsystems, minimizing the cost to us in doing extra design and saving space as we won't need to fit another piece of silicon into the internal cavity of the VESP. As a final thought for the group this module provides extended range, high speeds, lower energy and power consumption and an easier integration towards a TI MCU which is why the group considered this MCU in the first place.

The second Bluetooth solution that was researched as a possible candidate for the VESP is the CC2541 Bluetooth Smart and Wireless MCU chip. This chip contains the coding preloaded into the memory banks of the onboard proprietary TI MCU meaning the group will spend less time coding the MCU t act as a Bluetooth transceiver and more time integrating the chip on its circuit card and getting it to work reliably. The CC2541 supports variable data rates from 250 kbps to 2 kbps and sees use in many industry products including wireless monitors and gateway access controls making it more suited towards home automation; however, as the methods of Bluetooth communication do not change from device to device the CC2541 can be made to fit the Bluetooth needs of the VESP. The MCU on board the CC2541 chip contains all of the Bluetooth protocols needed to be considered a low-energy Bluetooth solution, helping us keep power consumption down and holds a total of 128 kb of flash ROM and 8 kb of flash RAM, which is enough for the group to program Bluetooth communications from. From an electrical standpoint, the CC2541 pulls a total of between 2 to 3.6 V from the power rail and operates as intended from temperatures ranging from -40 to 75°C which should be well beyond anything that the VESP undergoes in use. The onboard MCU and decreased voltage pull make the CC2541 an appalling option for a Bluetooth transceiver for the VESP; however, the group decided to search for additional solutions for Bluetooth communication.

The third Bluetooth solution looked at was the Intel Unistone PBA 31308 Bluetooth Module. The PBA 31308 runs using the Bluetooth 2.1 protocol with an Enhanced Data Rate (EDR); however, it can be configured to operate using the Bluetooth 2.0 + EDR or 1.2 protocols. The module features internal power and voltage regulators and only requires an external supply voltage of 2.9-4.1 V to power the module, and operates inside of an internal temperature range of -40 to 85°C . The module incorporates internally the RAM and ROM needed to store protocols and keep the Bluetooth module communicating; in fact many all components within the PBA 31308 are internal except for the antenna. The PBA 31308 is capable of running at data rates of 3.23 MBaud through HCI UART and holds additional interfaces for audio solutions and WLAN coexistence. The module features several Bluetooth specifications, including: the ability to operate with seven slave units, enhanced audio performances, power control and adaptive frequency hopping. For security the PBA 31308 has the ability to handle authentication, pairing, encryption and secure simple pairing. The total package size of the module is 11.6 mm by 8.7 mm by 1.8 mm, making it extremely small and easy to fit into the space within the VESP.

The fourth and final Bluetooth solution that the group looked into is the BLE113 Bluetooth Module by Bluegigas. Using the Bluetooth 4.0 protocol it contains the Bluetooth radio, software stacks and profiles needed to be considered a Bluetooth Smart device. From an electrical standpoint, the BLE113 operates at a voltage range of 2.0 to 3.6V and draws 0.4 uA of current in sleep mode and around 20 mA in transmit/receive mode. The module is able to operate within a wide range of operating temperatures, from -40 to 85°C . This module presently has seen use in medical and fitness sensors, mobile phone appliances and home automation. From a Bluetooth feature standpoint the BLE113 supports operation as both master and slave, as well as a network of up to 8 different devices and is capable of host interfacing via SPI and UART. This module has a small size profile with dimensions of 9.15 by 15.75 by 2.1 mm, giving it a very small volume size relative to the total internal cavity of the

VESP. The data rate of the BLE113 can be configured via the baud rate connection to up to 2 Mbps. This module draws heavily in design from the earlier mentioned Texas Instruments CC2541 Bluetooth module and like it includes an MCU to handle the software stacks needed for Bluetooth operation and communications.

3.3.2 FPGA

The FPGA chosen for the VESP project must be powerful enough to generate dynamically changing 3D objects on four separate screens in real time, while also running various operations in the background. This amount of processing power is easily achievable by today's average desktop computer, but is much more difficult for embedded systems. Fortunately, today's FPGA's have some pretty impressive hardware, which makes our design goal a realistic one. For the VESP project, the team took a look at various FPGA's that could possibly be sufficient for our requirements. Table **??** shows an overview of the specifications for the FPGAs that will be discussed.

FPGA	**ODROID-XU3**	**ODROID-U3**	**BeagleBone Black**
CPU	Quad-core Cortex-A15/Quad-core Cortex-A7 Combo	Quad-core Cortex-A9	Single-core Cortex-A8
GPU	Mali-T628	Mali-400	SGX520 3D
RAM	2GB LPDDR3	2GB LPDRR2	512MB DDR3L
Onboard Memory	-	-	4GB
PMIC	Samsung S2MPS11	MAXIM MAX77686	TPS65217C
Ethernet	10/100Mbps RJ-45	10/100Mbps RJ-45	10/100Mbps RJ-45
Storage Slot	MicroSD, eMMC	MicroSD, eMMC	MicroSD
I/O Ports	USB 3.0 Host x 1, USB 2.0 Host x 4, USB 2.0 OTG x 1, 30-Pin GPIO/IRQ/SPI/ADC	USB 2.0 Host x 3, USB 2.0 Device x 1, 8-Pin I2C/UART/GPIO, 4-Pin SPI	USB 2.0 Host x1, 69-Pin GPIO/I2C/SPI
Video Output	Micro-HDMI 1.4a, DisplayPort 1.1	Micro-HDMI	Micro-HDMI
Audio Output	Headphone Jack, via HDMI	Headphone Jack, via HDMI	via HDMI
API Support	OpenCL 1.1 Full Profile, OpenGL ES 1.1, 2.0, and 3.0	OpenGL ES 1.0 and 2.0	-
PCB Size	94 x 70 mm	83 x 48 mm	86 x 53 mm
Power	5V @ 4A	5V @ 2A	5V @ 0.45 A

ODROID-XU3

The ODROID-XU3 is powered by a Samsung Exynos 5422 SoC, which contains both a quad-core Cortex-A15 processor @ 2.0 Ghz and a quad-core Cortex-A7 processor @ 1.4 Ghz. Having both of these processor gives the Samsung Exynos 5422 the ability to migrate processes between processors, giving less demanding processes to the A7 and more demanding processes to the A15. This ability to dynamically allocate the appropriate amount of computing power to complete any given task is desirable for the VESP project since the device will be battery powered and so energy efficiency is very important.
On top of the super fast CPU, the Samsung Exynos 5422 also has a powerful Mali-T628 MP6 GPU with eight shader cores. The Mali-T628 is compatible with OpenGL ES (versions 1.1, 2.0, 3.0, and 3.1), OpenCL 1.1, and DirectX 11, giving the ODROID-XU3 to ability to render some highly complex 3D objects without having to create a custom API. With its high-end specifications, the Mali-T628 would be more than enough to drive the desired visuals the team wants to display on the VESP.

Other useful details for the ODROID-XU3 include: 2GB of LPDDR3 RAM, HDMI and DisplayPort connectivity, eMMC and MicroSD slot storage options, USB 2.0/3.0, Ethernet port, on-board audio codecs, and 30 pins for I/O (GPIO,IRQ,SPI, and ADC). All of these features are fully contained on a PCB that measures 94 x 70 x 18 mm, making the ODROID-XU3 a small and compact FPGA. Overall, the ODROID-XU3 is an amazing piece of hardware, which would be more than capable to drive the VESP with minimal lag. Priced at $179.99, the ODROID-XU3 is also an expensive piece of hardware, but even still, it's justifiable just by looking at the specification sheet.

ODROID-U3

The ODROID-U3 is powered by a Samsung Exynos 4412 Prime SoC, which contains a quad-core Cortex-A9 processor @ 1.7 Ghz and does not have any built-in processor optimizations like the Samsung Exynos 5422. The CPU is accompanied by a Mali-400 GPU with four shader cores, and is only compatible with OpenGL ES 1.1 and 2.0. Other specifications include: 2GB of LPDDR2 RAM, micro HDMI connectivity, eMMC and MicroSD slot storage options, USB 2.0, Ethernet port, on-board audio codecs, and only a total of 12 pins for I/O (GPIO, UART, I2C, and SPI). The device is priced at $65.00 and measures 83 x 48 x 18 mm, making it significantly more affordable and compact than the ODROID-XU3. Overall, the ODROID-U3 is a decently capable device, that's inexpensive and spacialy efficient, which might be sufficient for the VESP to run adequately.

BeagleBone Black

The BeagleBone Black is powered by a TI Sitara AM3358BZCZ100, which contains an ARM Cortex-A8 processor @ 1.0 Ghz and a SGX530 3D graphics accelerator. Unlike, the previously discussed FPGA's the BeagleBone Black does not support

OpenGL ES in any capacity. Other specifications include: 512MB DDR3L RAM, HDMI connectivity, 4GB of on-board flash memory with the option of expansion using the MicroSD port, Ethernet, USB 2.0, and two 46-pin headers for I/O (GPIO, SPI, I2C, and GPMC). The device runs for $55.00 and measures 86 x 53 x 18 mm. The BeagleBone Black would be our cheapest option, with plenty of I/O for our purposes, however, the device would be unable to render complex 3D objects without having to build custom 3D graphics libraries.

3.3.3 Battery

Batteries are composed of three different parts, an anode which is the negative side, a cathode which is the positive side, and the electrolyte which is the part that divides the cathode and the anode. As all the electrons want to jump from the anode to the cathode, the electrolyte is the chemical that keeps the electrons from jumping from one side to the other. Once a circuit is connected, the electrons travel from the anode out the circuit into the cathode, but this electrochemical process changes the chemical between the cathode and anode which is how the battery runs out after many cycles. On the other hand, to recharge the battery, the process of electrochemical happens in reverse where the anode and cathode are restore to their original full power state using any power source.

There are two types of battery connections, serial and parallel connections. Serial connections means when the battery pack has two or more cells connected in series in which the whole voltage is added but the current remains the same. A higher voltage has the advantage of using a small size conductor. The disadvantage of the batteries being connected in serial is that if one of the cells fails for any reason, the whole battery pack needs to be replaced because the other cells are unable to deliver the same amount of energy causing to reach the end of discharge sooner and also because it's hard to find the same capacity of the battery cell when trying to replace the defective battery cell. On the other hand, the parallel connections means when the battery cells are connected in parallel. This type of connection increases the current and runtime of the battery while the voltage is kept the same. A defective battery cell will only reduce the total capacity of the battery pack. Finally, the serial/parallel connection allows us to design the specific voltage and current needed for any project. This type of connection is usually use for laptop batteries with a protection circuit to monitor each cell individually

The capacity of the battery which is measure in Amp-Hours is really important when deciding the amount of time wanted for the VESP to be power without being charged in the dock. In order to make the VESP work without being connected to the dock, the group needed to choose a battery that will power every element inside the 4 LCDs screens. There were multiples of batteries to choose, but we needed a battery that recharges as well. The group research the top four rechargeable batteries that could be used for VESP and compare which one will be best suitable for the whole project. A Lead battery acid, Nickel Metal Hydride, Lithium ion, and a Lithium-ion polymer battery.

Lead acid batteries are mostly design for automotive engines, and is the oldest rechargeable battery in existence. Some good advantages about the lead-acid batteries I that in price it provides the best amount of power and energy per kilowatt-hour. Ninety-seven percent of the lead in the battery is recycled and used again in the new batteries. The disadvantage of the lead-acid batteries is that it's one of the heaviest battery in the industry, and also their size is bigger than other types of batteries. This are reasons why they are only used for any type of vehicle or house.

Nickel-Metal Hydride (NiMH) is another type of rechargeable battery that is a non-toxic battery that has a higher efficiency compare to the other nickel batteries and it's one of the lower cost rechargeable batteries for portable devices. The advantages of the NiMH battery is that it has a thirty to forty percent higher capacity and less prone to memory that any other nickel batteries. The disadvantages of the NiMH battery is that is limited in their battery life, meaning that deep discharges reduces its service life. Also, it does not work well when it's overcharge in which the charge has to be kept low and generates heat during fast charging and high load discharge. Another big disadvantage is that it should not be stored in high temperatures and at about forty percent of charge.

Lithium Ions battery is one of the new batteries that has a high energy density. This batteries are in almost all cellular phones and laptops due to its higher capacity. The advantages of this battery it's definitely its high energy density, also its low self-discharge which is less than half of the nickel batteries, and it needs low maintenance because it has no memory which it does not need any periodic discharge. The disadvantages of the lithium-ion battery is that its relative expensive in terms of cost to energy ratio and compared to the nickel batteries, about 40 percent higher. Also, it is fragile which it requires a protection circuit to maintain its regular operations.

Lithium Ion Polymer batteries are batteries similar to lithium-ion but are its added a gelled electrolyte which makes the battery a flexible material and enhance ion conductivity. The advantages of this battery is that its light weighted due to the gelled rather than the liquid electrolyte, and its more resistance to overcharge which is less chance of electrolyte leakage. The disadvantages of the lithium ion polymer battery is that compared to the lithium-ion its capacity and energy density is lower. Also its cost is higher that the lithium ion batteries

3.3.4 Sensors

There are several different types of sensors intended to be used for the project. The device needs to be able to sense when someone is nearby so it knows when to leave its power saving mode and be ready to interact with its user. To determine this, a proximity/motion sensor and a light sensor will be combined; the light sensor will also communicate with an auto-brightness control. The device must also be able to sense when it has been picked up off of the dock, and know its new orientation when held in a user's hands to know which screen if facing upward. For this, a gyroscope will be used.

Proximity/Motion Sensor

The first component to be looked at is the IR Mid-Range Proximity Sensor (TSSP4P38) found on Vishay.com. This part is used for proximity sensing for up to 2 meters or approximately 6.5 feet, which is a little on the shorter end of the range ideal for the project. The supply voltage required is 2.5 V to 5.5 V, with a low supply current. It receives 38 kHz modulated signals with a peak sensitivity of 940 nm and is resistant to supply voltage ripple and noise, which is useful for the project since the power source will be splitting between so many components, some of which may be turned off during use (for example, particular LCD screens) which may cause voltage ripples. The output is analog which isn't ideal for the project but a microcontroller can still easily analyze and respond to the data. This component sells for $1.28. Although the range is on the lower end, this component is still viable for the project. The next option for is a motion sensor from Panasonic.biz (part number EKMC1601111). This sensor has a range of 5 meters or 16.4 feet which is very ideal for the project as the VESP will need to sense the area of a room to know when a user has entered. The area of sensing is also spread out across 82 degrees. It operates at a supply voltage of 3V-6V which keeps it within a low range necessary for the project. It has a digital output (Figure 5) with is preferable for the project so it can directly give information to a microcontroller. This sensor sells for $10.26 per unit and although one unit may be enough for the specifications of only sensing motion in on general cone-like direction (as seen in Figure 6), to get about a 360 degree sensing area would require four units for a total of $41.04. This sensor makes a far stronger choice for the project as it will more efficiently detect when a user enters the proximity, however, the cost must be taken in to consideration.

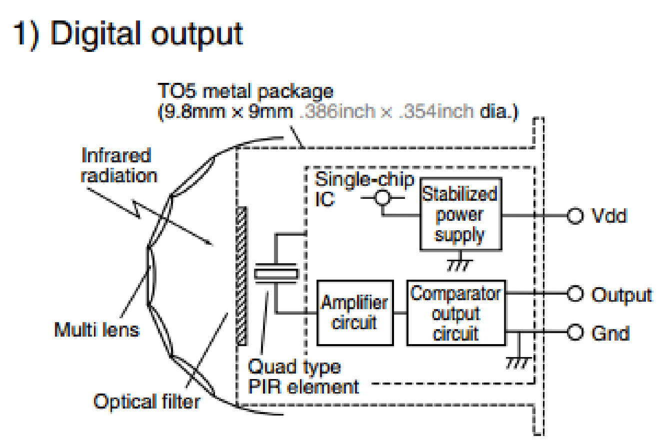


Figure 5: EKMC1601111 Diagram
Image Use Courtesy of Panasonic

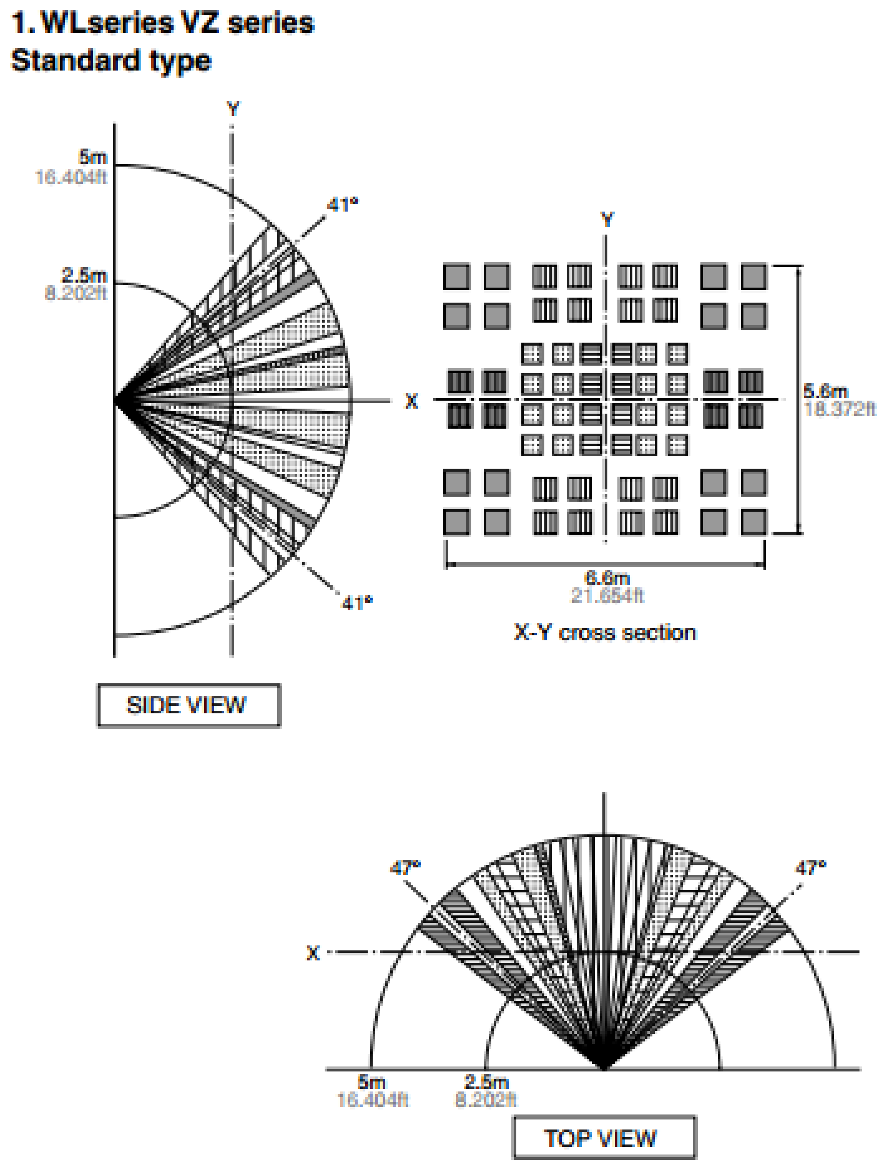


Figure 6: IR Sensor Area Coverage
Image Use Courtesy of Panasonic

Ambient Light Sensor

The ambient light sensor must accurately sense the lighting for both determining if a user has entered the room and to adjust the brightness of the LCD screens when lighting changes to help save power. The first ambient light sensor to be looked at is from Texas Instruments (part number OPT3001). This component has precision optical filtering to "match the human eye" which makes it very accurate. It has a supply voltage range of 1.6V to 3.6V and a very low operating current of 1.8 uA. The low supply voltage will help increase the duration of the battery life when the device is not on the dock. Its measurements can have the option to either be continuous or single-shot and the control and interrupt system can be autonomous so the processor can enter a sleep mode, which is ideal for the project. That feature can be used to help determine when a user enters a room, and the device saves power while waiting for a wake-up event. The output is digital and can communicate directly with the processor via an I2C or SMBus 2 wire serial interface. This sensor is a strong choice for the project. The next option for a light sensor is found on adafruit.com (part number TSL2561). This sensor can detect ranges from .1 up to 40,000 Lux and has three different options for spectrum measurement. It can be configured to separately measure infrared, full-spectrum, or human-visible light, which can be applied to the project to both accurately determine when a user may have entered the room and to communicate with an auto brightness feature. This sensor communicates digitally through an I2C bus interface and is compatible with any microcontroller. It operates with a supply voltage range of 2.7V-3.6V and has a current draw of 0.5 mA when actively sensing and less than 15 uA in power-down mode, which is helpful for keeping the battery life long while the “cube” is off the dock. This sensor is another good choice for the project and costs $5.95.

Accelerometer/Gyroscope

The gyroscope is important for the project because it determines the orientation of the “cube” when it has been taken off the dock and is in a user’s hands. The first gyroscope to be looked at comes from st.com (part number LPR410AL) which sells for $6.59. This component operates on a voltage supply range of 2.7V to 3.6V and a current supply of 6.8mA, and has a low power consumption, which is preferable for the project. It also has a sleep mode function which is useful as it does not need to be active while the “cube” is on its dock and not in a user’s hands. Its output is analog and has two separate outputs for each axis. Although an analog output is not preferred for the project, this is still a viable choice and is relatively cheap. The first accelerometer to be considered is from Kionix Inc (part number KXCJ9-1008). This accelerometer is 3mm x 3mm x .9mm and operates from a DC voltage supply range of 1.8V 3.6V. Other key features include an internal voltage regulator, low current consumption, and most importantly, a user-configurable motion wake-up function, which will be important for the project because the accelerometer only needs to be active when it’s held in the user’s hands. It communicates digitally via an I2C 2-wire serial bus, which eliminates any analog-to-digital converter requirements and allows for easy integration into the project. The device can be communicated with directly by a microcontroller. This accelerometer makes a strong choice for the project, especially with the “wake-up” feature, and costs $2.65. The next component to be considered is a combination component of both a gyroscope and an accelerometer. This part (MPU-3050) comes from InvenSense.com and sells for $9.09. This sensor has X-, Y-, and Z-Axis angular rate sensors on one integrated circuit and a digital output through an I2C bus which connects to a digital 3-axis third-party accelerometer. The component also produces a digital output through an I2C bus and 16-bit ADCs for digitizing sensor outputs. These connections can be seen in Figure 7. It requires a supply voltage between 2.1V and 3.6V which keeps it in the lower voltage range required by the project. Its max power consumption (when all 3 axis gyros are active) is 6.1 mA, and its sleep mode requires only 5 uA. The sleep mode is useful for the project as the gyro will not need to be active when the "cube" is on its dock. This component makes a very strong choice for the project as it is a combination of a gyroscope and an accelerometer and provides a lot of possibilities with applications of the information these sensors provide.

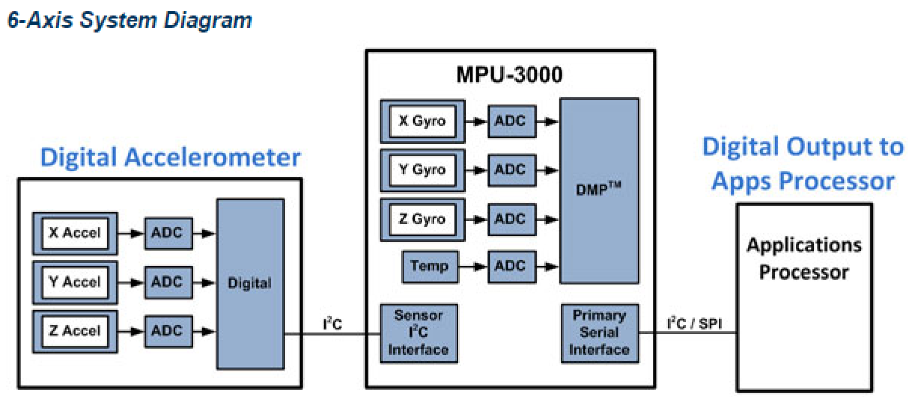


Figure 7: MPU-3000 Block Diagram
Image Use Courtesy of InvenSense

Table 10 can be used to analyze and compare the features and costs of the sensors.

Table 10: Sensor Comparison Sheet

3.3.5 Speaker System

There were few limitations to find a quality and efficient sound system that would fit in the dock. First the group decided to find out on the amplifier for the speakers. An amplifier modifies the output from the power input and makes it a stronger signal output. Basically the amplifier takes the energy from the power supply and controls its output to match the input signal but with a bigger amplitude. There are different types of amplifier with different efficiencies but the group is just going to specify in three types of amplifier: Class A, Class B, Class AB, and Class D amplifiers.

Class A amplifier are the most common amplifier and one of the best class amplifier because of their low signal distortion level and also because their sound output is great quality. This class of amplifier have a single output that conduct a full 360 degree of the output waveform and the output transistor always ON because they have a constant current flowing through the output of the transistor even when there is no incoming audio which represents a continuous loss of power in the amplifier.

Also, it has the lowest distortion of any type of amplifier but it's really inefficient, wasting about 70 percent of its power heat with an efficiency of less than 30 percent.

Class B amplifier was created due to the inefficiency and heading problems of class A amplifier. This type of amplifier uses two transistors instead of one that was use in class A meaning each transistor conducts 180 degrees of output waveform before both are combine. One of the transistor is for the positive portion of the waveform and the other is for the negative part of the waveform. There is an ON and OFF switch in the transistors when conducting positive and negative signals, this causes nonlinear distortions. Unlike class A amplifier, class B does not give a current flow signal through the transistors if there no audio input signal and their efficiency is at about 50 percent.

Class AB amplifier is a combination of class A and class B amplifier. The conduction angle is somewhere between 180 to 360 degrees with an efficiency of 50 to 60 percent. This type of amplifier uses two transistors as well, having one output transistor stay a little bit longer, while the other transistor takes over amplifying the other half of the audio waveform. The advantages of this type of amplifier is the diodes and resistor in series that makes a small current output to the speakers at all times which takes out the switching distortion of the class B design.

Class D amplifier is a switching amplifier or also same as a Pulse-Width Modulation (PWM) supply. This class of amplifier is composed of a pulse-width modulator, two output MOSFETS, and an external low pass filter to recover the amplified audio signal. When the top MOSFET is ON and the bottom is OFF, the square wave goes high. When the bottom one goes ON and the top goes OFF, the square wave goes low. Because only one MOSFET is ON at any given time, the resistance is low, meaning less heat which makes this type of amplifier really efficient; around 90 percent efficient. Finally, the amplified audio signal is send to a lowpass filter.

3.3.6 Housing

The Housing of the VESP will be divided into two sections, one will be the portable section which is where all the drivers and components that are powering the LCD screens. The LCD housing will only have two additional material that will close each end of the cube. The material will be made out of plastic because we want to limit the weight of the LCD screens and its components. Besides that, there will be four rods in each corner inside the LCD screen. This will help with glueing all LCD screen, the support of the components inside, to screw together the two plates that will be in the bottom and top.

For the bottom housing part of the project, the group was thinking of getting something rigid that will support the LCDs screens and its components. Metallic material was the first options that the group was looking at due to its rigidness, but it will have more weight than expected and the possibility of making an interference with the circuits. Also, the possibility of using was wood was an option as well. One, because it will reduce the distortion of the speakers and it will have add or subtract the tone significantly depending on how the sound system is build. On the other hand, plastic is the third option just because it will be cheaper and because it's a common used material for most types housing in the electronics industry. At the same time, it will be easier if the group decided to use a 3D printer for the design and to manufacture.

3.3.7 Operating System

With such an ambitious project as the VESP, the team wanted to cut as much as the work as possible and so made the decision to depend on an already available open-source operating system to bridge the hardware and the software. Now, there are really only two major options when choosing a free-to-use, stable operating system and they are Android and Linux. Ultimately, however, only one can be chosen and so the team sought to find the best fit.

Android

Android is an operating system developed by Google and is specifically catered to mobile devices. At first glance, Android seemed like the optimum choice when picking the VESP's operating system, but a closer look revealed some interesting points. The Android operating system is actually driven by a Linux kernel that talks to the actual hardware. Android implements a special virtual machine that translate code written for Android into Linux interpretable code, and then that code is compiled and run on the hardware. The amount of layers between the top layer of applications and software and the bottom layer of hardware is an optimization issue. With the VESP and its hardware components, the team wants to utilize as much of the processing power to drive the actual applications and visuals, and not have to deal with so much overhead.

Linux

Linux is one of the most stable and most efficient operating systems on the planet and comes in many different flavors with a multitude of different distributions or distros. When considering Linux as the main operating system for the VESP, the team must also consider which distro of Linux is most appropriate. The Linux distros that the team inVESPigated are as follows: Ubuntu, ArchLinux, and Debian.

Ubuntu

Ubuntu is the second most widely used Linux distros out on the net. In turn, Ubuntu is also one of the most supported Linux distros with very active forums and constant updates. With so many Ubuntu users actively developing software, it makes the distro very appealing since more than likely the team will have questions about specific functionality and the active community would likely be quick to respond. Additionally, Ubuntu is a highly mature distro with a very slick UI, optimized operation algorithms (task scheduler, resource manager, etc), and a lot of API support.

Another reason to choose Ubuntu, is the possibility of going a step further and installing the Phone version of Ubuntu onto the VESP. The Phone version of Ubuntu is still in its early stages, having very limited platform support and potentially ridden with bugs, but it also offers a more touch friendly UI off the bat. If the Phone version of Ubuntu were to be able to be installed on the VESP, then it would remove the need to program a custom UI for the user to use to navigate the device.

Arch Linux

Arch Linux is a unique distro of Linux in that it comes with very little functionality out of the box. With Arch Linux, the user is given the ability to specify exactly which software packages are installed on the system in order to remove as much overhead as possible. This unique aspect to Arch Linux would make the VESP more optimized and would run faster while also using less power. On the other hand, implementing Arch Linux would take considerably more time compared to another distro since researching and installing the essential packages the team will be using would be required.

Debian

Debian is the most widely used Linux distro on the net and also the oldest. Unlike Ubuntu, Debian is not developed by a private company and instead is purely driven by the community. One would think that this would make Ubuntu more fast and stable, but on the contrary, Debian is even more stable and quicker than Ubuntu, though arguably less so than Arch Linux. The drawback of choosing Debian would be that it's not nearly as user-friendly as Ubuntu and so there would be a more significant learning curve, though, again, arguably less so than Arch Linux.

3.3.8 API

Without good software, the VESP would just be an expensive, sleek looking prism and good software take a lot of time to develop. Thankfully, there are readily available APIs that cut a lot of the work of developing quality software. For the VESP project, the team used a variety of APIs in order to create a friendly and interactive user experience, as well as various useful and fun applications.

Qt

The Qt API is comprised of a multitude of modules, making it an extremely powerful and useful tool for software development. Qt streamlines coding up GUIs, which is arguably the most time consuming part of creating an application. Qt even has specific tools for building GUIs for embedded systems, which is another giant advantage. On top of that it includes the "Qt OpenGL" module which will make interfacing OpenGL code into a Qt application without too much difficulty. Qt also has both C++ and Javascript libraries, but for the sake of speed and efficiency, the team decided to only use the C++ libraries. Overall, the Qt API will be an essential asset to creating quality software for the VESP.

OpenGL

OpenGL is one of the most advanced and supported 3D Graphics API available, plus it's open source. By using OpenGL, the team can create the beautifully complex visuals that the VESP will be able to render. Unlike Qt, however, OpenGL can be pretty difficult to understand since a lot of steps and processes go into making a rendered 3D object. Thankfully, the team has previous experience with OpenGL, which will greatly reduce the learning curve.

OpenCL

OpenCL is another open source API, just like OpenGL, however OpenCL is used to make fast, efficient, parallel software. This is due to OpenCL's ability to tap into the GPU in order to execute simple, redundant calculations, thereby taking away some of the workload off the CPU. One of the goals of the VESP is to create a smooth and streamlined user interface with minimal hiccups, and implementing OpenCL is one way to achieve that. However, unlike OpenGL, Qt doesn't have a module to support OpenCL, so trying to implement OpenCL might prove to be too time consuming for the team to include in the VESP. The team will consider using OpenCL if time allows.

3.4 Component Selection

The component selection for the VESP centered around our project goals and objectives. We want to the VESP to run relatively cool, and not have to worry about the internal hardware components overheating. Due to the VESP's unique form factor and needing all the components encased within the four LCD screens, the components selected for this project must be either able to be properly cooled or be able to withstand the temperatures which it'll have to operate in. Additionally, we want the VESP to be as energy efficient as possible in order to be as independent from the charging station as possible. For this to be possible, the energy requirement and built-in energy saving features must be considered when choosing the components. Cost is also another factor that must be considered when choosing the components for this project, as the team must work with a strict budget which ideally should not be exceeded. Finally, the last thing that must be considered is the size of the components; if the many components making up the VESP are to fit within the four LCD screens, they must be relatively small.

3.4.1 Power Adapter

In order to build the power adapter for the power supply, the group decided to research on power transformers. Due to our inexperience in making a transformer and also to avoid any possible fire hazards in the future, the group decided to buy one. The group was looking for a transformer that wasn't too big but needed to have enough power to make every component work. Finally after many research, it was narrowed down to two power transformers that were founded in the Jameco Electronics website. As you can see in the Table 11, both transformers have the same primary voltage and frequency and also the same secondary voltage. Transformer 2 have double of the secondary current which makes it an output of double the power. Besides that, due to double the power, transformer 2 need to be bigger in size and weight.

	Transformer 1	Transformer 2
Primary Voltage	115/230 VAC	115/230 VAC
Primary Frequency	50-60 Hz	50-60 Hz
Secondary Voltage	12 VAC	12 VAC
Secondary Current	2 A	4 A
Power Rating	24 VA	48 VA
Length	1.89"	2.25"
Width	2.75"	4.00"
Height	1.99"	2.63"
Weight	1.3 lbs	2.3 lbs

Table 11: Transformer specifications

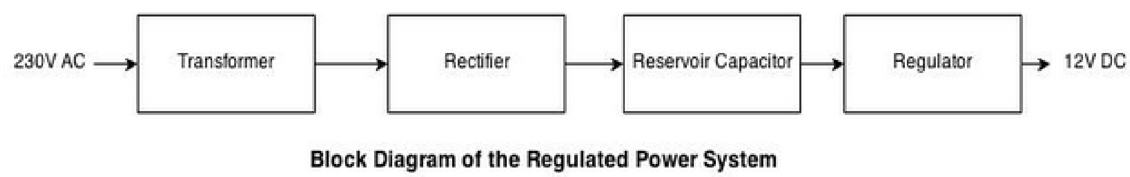


Figure 8: AC voltage Regulated

When choosing which transformer the group will be using, we needed to figure out the amount of power it was needed. We will only be using power to charge the battery and power for the speakers. Based on that result, with a 12V output and a 2A current will be enough for both components as you can see in the block diagram of Figure 8 above. Besides that, the primary side of the transformer will be connected to a 3 prong cord that will be used for connection to the wall. After making sure that the secondary side is a sinusoidal of 12V, four diodes will be connected properly making a bridge rectifier delivering direct current of double the frequency. In this case it would be 120Hz DC. We will add a capacitor to smooth the signal, but we got to make sure the the capacitor and he diodes are rated for the voltage the we will be working with. Even though it may seemed like we created a direct current, its just an unregulated direct current. In this case we will be using a 5V linear voltage regulator.

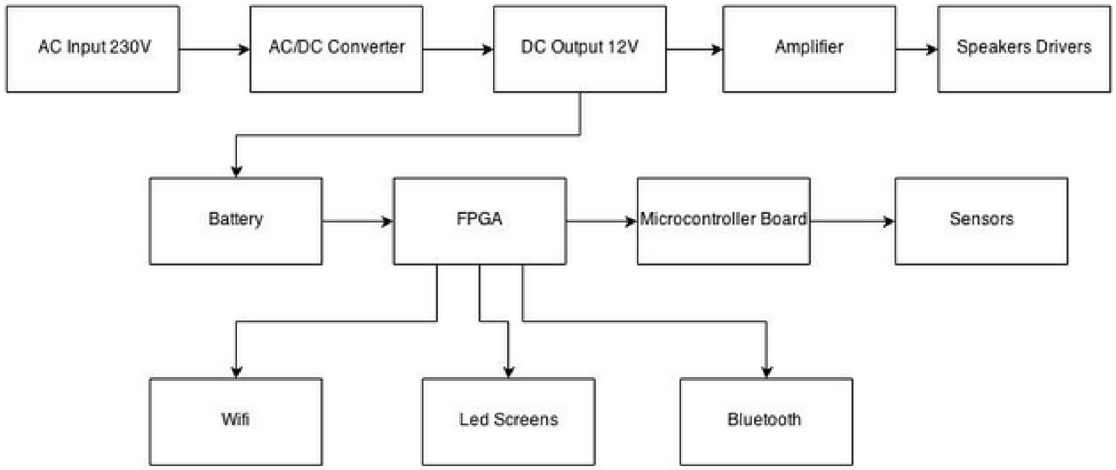


Figure 9: The whole Power System block diagram for the VESP

3.4.2 LCD Screens

The selection of which LCD screens to use for the project is one of the most essential decisions for the design. Not only do the screens have to meet the performance specifications, they also determine the dimensions of the "cube" portion of the VESP. The "cube" dimensions are very important because most of the hardware must fit inside, such as the microcontroller board, the FPGA, the LCD drivers, any additional PCBs and wires, etcetera. The screens must also be compatible with the processor and any LCD drivers. Another very important requirement is there must be at least one touchscreen LCD, otherwise there would be no way to interact with the Therefore, the selection of the LCD must meet a number of broad specifications, these specification are outlines by Table 12 below.

LCD Selection Specifications
• Compatible with Processor
• Creates a relatively large volume for internal hardware
• Meets performance specifications (colored screen, backlight, resolution, voltage supply level)
• Affordable (does not exceed LCD budget)
• Touchscreen option available

Table 12: LCD Selection Specifications

Of all the LCD screens researched, the best choice appears to be the second 7" LCD screen (part number ER-TFT070-4) option from BuyDisplay.com. This LCD display meets all of the general specifications mentioned in Table 12 and does so outstandingly. This 7 inch display, which is the ideal size for the scope of the project, has much greater dimensions than any of the other options that were researched, other 7" displays included. The dimensions of a single screen is 164.9mm x 100.0mm, so the combination of four screens making the walls of the "cube" should provide plenty of room for the internal hardware. The voltage supply range is relatively low, at 3.3V which is great for the project as there will be four screens running at this voltage, and the battery life when the VESP isn't in its dock needs to be significant. The other performance specifications this option meets are the display specs. It has an 800 x 480 pixel display with 24 bit color and a backlight. The cost of this screen is where it shines the most. Because this screen doesn't have an on-board MCU (which research showed isn't necessary for the project) this screen sells for $17.04 per unit versus the other 7" options which had a range of about $65-$75 per screen. On top of that outstanding cost, the capacitive touch screen upgrade is only an addition $7.54 which runs on a voltage supply of 2.8V-3.3V, whereas the other options had about a $10-$15 difference between a touchscreen version and a non-touchscreen version.

This screen option also is preferable to use for the project due to its simple connection option. Since there will be four screens total being integrated into the VESP, the simpler the interfacing, the better. Using the TFT070-4's FPC-Connector to link all four displays into a single LCD controller both simplifies the project and saves on internal room (versus having a driver for each screen). A brief summary of this LCD screen choice can be reviewed in Figure 13 below:

Connection Type	FPC-Connector
Display Voltage Supply	3.3V
Touchscreen Voltage Supply	2.8V-3.3V
Display	800x480, 24-Bit Color
Screen Dimensions (in mm)	164.9x100
Potential Volume (mm^3)	1,649K
Per Unit Cost	\$17.04
Touch Screen Cost	\$7.54
Total Cost	\$77.78

Table 13: ER-TFTM070-5 Specifications

3.4.3 Microcontroller

The microcontroller board is an important part of the project and research has come up with a few great options. The microcontroller board must be able to communicate with all the sensors, the Wi-Fi module, and the Bluetooth modules, and send information to the FPGA (Figure 10 briefly shows the block diagram of these connections). This means its processing power must be strong enough for all this information, and it has to have enough physical hardware room to connect to all these components. Ideally it should also run on a relatively low voltage because of the projects battery-powered mode, although all the researched microcontroller boards didn't really exceed the voltage range so that won't be a problem. Additionally, the size of the board must be kept in mind as it will have to fit side the "cube" portion and still leave enough room for all the other internal components.

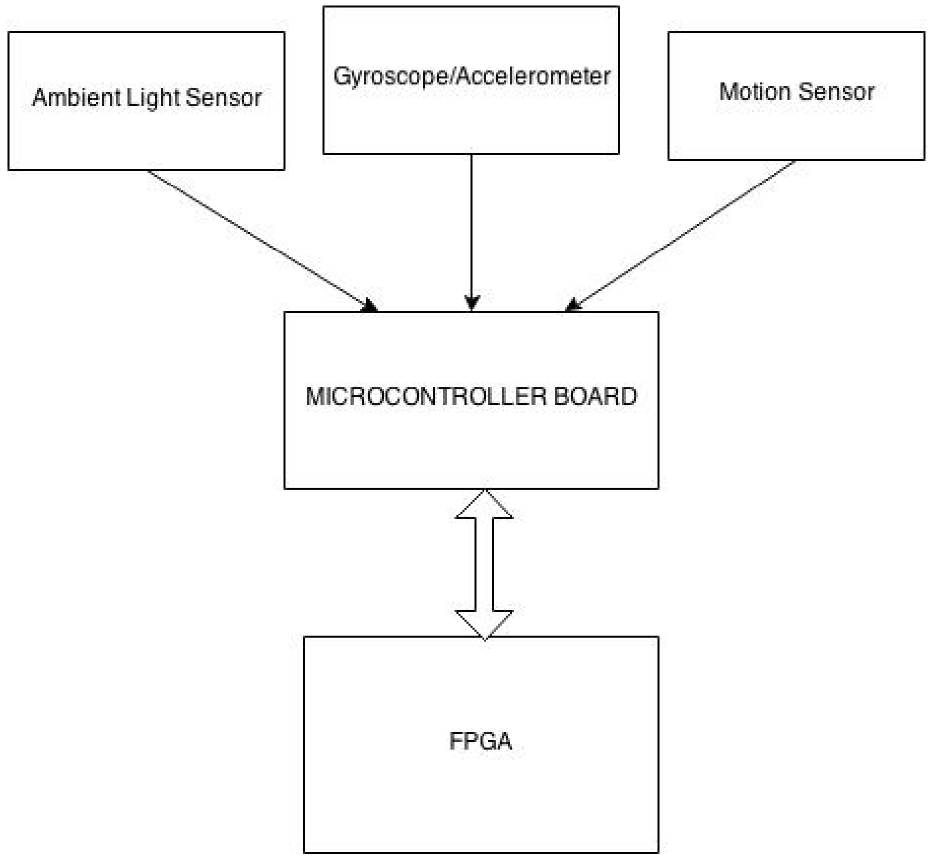


Figure 10: Microcontroller Block Diagram

All of the researched microcontroller boards appear to be strong, capable choices. Since they all seem like they could handle the functions required by the project (processing ability and hardware space) and are relatively the same size, the cost will be one of the main deciding factors. The project group has already acquired an MSP-EXP40G2 Launchpad prior to the assignment of the project, therefore, none of the budget would have to be spent towards this selection. This board should have the processing power and the hardware room required for the project making it a great choice for the microcontroller.

The MSP-EXP430G2 from Texas Instruments comes with two different microcontroller chips: the MSP430G2452 and the MSP430G2553. The board itself has 16 MHz speed, 16 KB of flash memory, 512B of RAM, a 32 KHz crystal, an 8 channel 10-bit ADC (analog to digital converter), a comparator, two 16-bit timers, and several interfacing options. Additionally, it has a 20 pin DIP socket and an option for a 20 pin booster pack. The interfacing options are 1 I2C, 2 SPI, and 1 UART. All these provide plenty of room to connect all the sensors and modules. The Wi-Fi and Bluetooth modules can communicate through any of the compatible serial communication options and the sensors can connect to the necessary pins (whether they need the ADC, the DIP sockets, or one of the extra serial communications). This board also operates at a supply voltage of 3.6V which is ideally low, important for not only being efficient when the device is operating off the battery, but for not producing too much heat since it is inside the "cube" region of the VESP. Table 14 can be used to quickly reference the specs of the microcontroller board.

MSP430
• 3.6V
• 68mmx51mmx13mm dimensions
• 16MHz
• 16KB Flash
• 512B RAM
• 8ch 10-bit ADC
• Comparator
• 2 16-bit Timers
• Up to 1 I2C, 2 SPI, 1 UART
• 20 PIN DIP, 20 PIN Expansion available

Table 14: MSP430 Hardware Specification Data Sheet

3.4.4 Wi-Fi

After looking through the datasheets and researching the uses of the viable candidates our group has decided to use the Texas Instrument CC3100 as the Wi-Fi module for the VESP. The primary factors included its lower price compared to the more space efficient CC3200 but a far superior data rate which means a faster connection when compared to the CC110L and CC3000. Additionally the group hopes that by not using an additional integrated circuit in the form of the CC3200 we can better manage any latent heat formed within the VESP by allowing more open space for the heat to disperse. The factor of space also and interfacing also played a role in the group's decision as the addition of the CC3200 IC would fill up more of our already restricted space and force us to use more space for wiring to integrate the CC3200 with the FPGA inside the VESP. The ease of integrating the Wi-Fi module will hopefully allow us to ensure that the communication capabilities onboard the VESP are strong and will allow for an optimum user wireless communication experience.

Inside the VESP wireless communication will be used to allow for time and date readings, weather forecasts and naturally checking email and the internet. It is for those reasons that we planned to incorporate wireless communication via Wi-Fi as well as short range communication via Bluetooth. It is hoped that the VESP will be able to function similarly to a Smartphone in terms of internet connectivity and

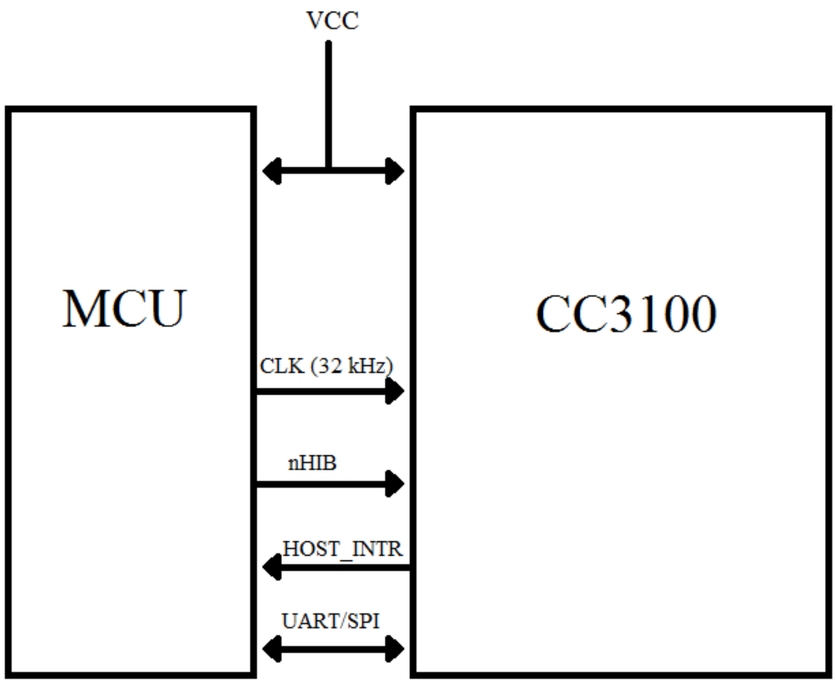


Figure 11: Functional Block Diagram of the CC3100
Courtesy of Texas Instruments

wireless capabilities, so it can be used for a home internet browsing within a unique prism package. With the group's decision to use the CC3100 module the group now had to figure out what features were important to the VESP and what modes would the CC3100 be operated under to ensure that the VESP does not lose its wireless communications to the internet.

Texas Instrument CC3100 Overview

The Texas Instrument CC3100 SimpleLink Wi-Fi is a small booster to TI's line of microcontrollers including the MSP430 and the TIVA. Since it is a booster it can connect directly to the microcontroller and establish a data connection via the UART terminal on the CC3100, allowing easier management and connection of the two separate parts and it allows us to fit more into our already limited building space. Included within the code of the CC3100 are the protocols, such as IP and TCP and the stacks required to have secure, reliable connectivity to the internet, allowing the microcontroller to save more coding space to be used for more important tasks. The CC3100 is capable of wireless audio and as a use for an internet gateway, and thanks to an included simple security program it can also be used in small security projects such as home security. The top level block diagram of the CC3100 and its MCU connections is shown below in Figure 11.

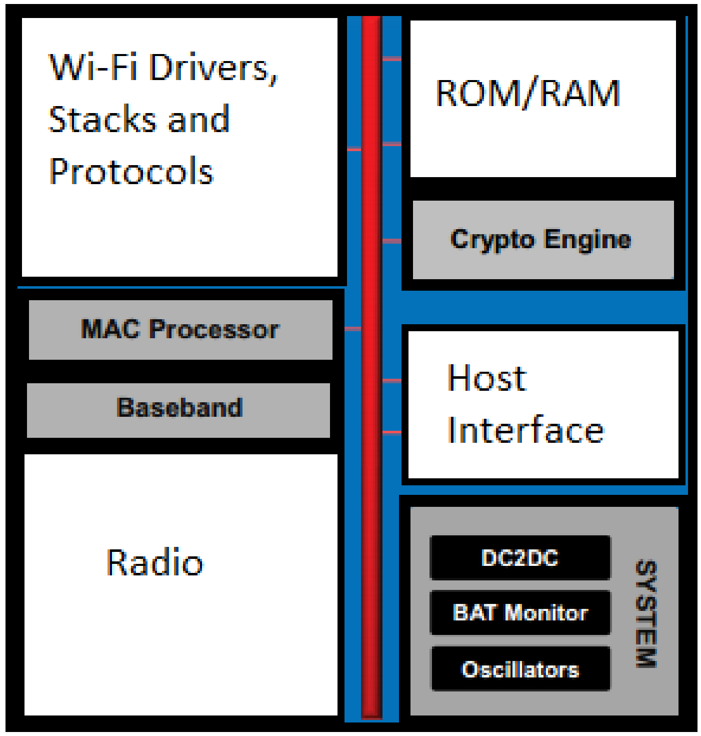


Figure 12: Hardware Block Diagram of the CC3100
Courtesy of Texas Instruments

CC3100 Hardware

For every many components within the VESP there are two distinct sections, the first is hardware which handle the electric characteristics of the component device and the second is software which handle what the device will do and for wireless communication how it will behave and connect to wireless networks. As shown in Figure 12 there are several hardware components within the CC3100, most important to our project is the radio, the host interface, the onboard ROM and RAM, the power systems and all components needed for wireless communication including the Wi-Fi drivers, stacks, protocols and the MAC processor. All of these components will be discussed in the forthcoming subsections.

Wireless Features

Wireless communication within the CC3100 is composed of 3 primary components, the first is WLAN, the second is the network stack and the third is host interfacing and drivers. WLAN on the CC3100 is achieved by an integrated radio capable of using the 802.11 b/g/n standards of wireless communication, in addition it is very simple to connect to a home network using user-configurable profiles that are stored within the memory of the CC3100 while still having the ability to keep important or confidential data secure using memory stored security encryptions. The network stack on the CC3100 features an integrated IPv4 TCP/IP stack allowing for simple

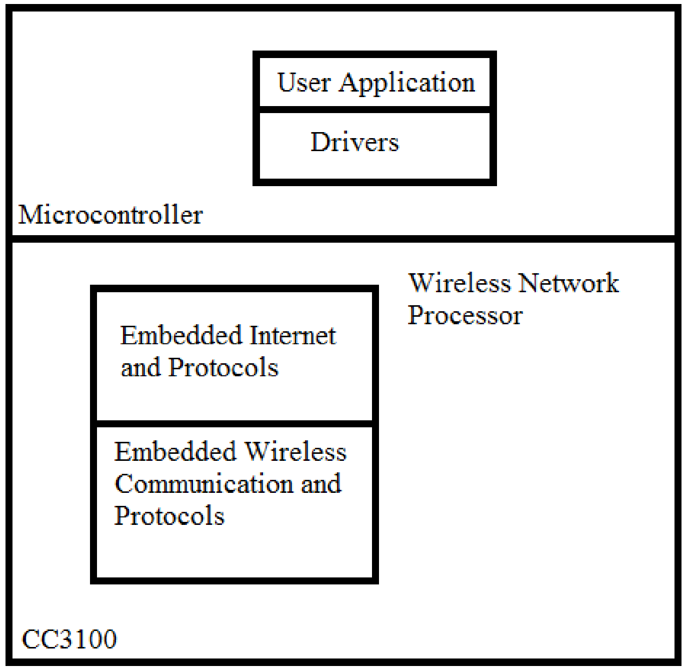


Figure 13: Software Block Diagram of the CC3100
Courtesy of Texas Instruments

internet connection using any microcontroller or small scale circuit such as an ASIC or microprocessor. Lastly, the host interfacing on the CC3100 is achieved by either an SPI running at a clock rate of up to 20 MHz or a UART terminal with a maximum baud rate of 3 Mbps, both methods are easily performed thanks the ease of connection of the CC3100 with a TI made microcontroller such as the MSP430. The block diagram of the microcontroller and CC3100 software interfaces and components is shown below in Figure 13. The coding that will program the CC3100's wireless communications will be covered in the later section on wireless communication coding.

Power Management

The CC3100 as any integrated circuit does needs a supply of voltage in order for it to operate as it needs to. The CC3100 contains DC to DC converters internally to accommodate two different forms of power and voltage supply. The first of which is for covering a wide voltage supply, from 2.1 V to 3.6 V, the input DC voltage is stepped down internally by DC-DC converters until the needed voltage to operate the CC3100 is achieved. The second and more direct method of powering the CC3100 is to simply apply from a DC voltage supply a pre-regulated 1.85 V directly to several of the input pins on the CC3100, specifically pins 10, 25, 33, 36, 37, 39, 44, 48 and 54 (See Table 15 in Section 3.4.4). For our project we will operate the CC3100 in its wide-voltage connection mode, which allows for a greater range of voltages to be input from our power supply as opposed to one particular voltage.

Modes of Operation

As with laptops and computers today the CC3100 will operate in modes other then active when under certain conditions in order to preserve the life of the battery, this section will describe the two most common modes: Low-Power Deep Sleep and Hibernate. The low power deep sleep mode is activated based off of inactivity within the CC3100, which is based off of internal power optimization. Under low power deep sleep the battery drain current decreases to about 115 uA, in addition as the certain configuration data is preserved in the CC3100 under this mode no handshaking from the CC3100 or microcontroller is needed to awake or enter the CC3100 deep sleep mode and typical wake up is faster than the hibernate mode, around 3 ms. In hibernate mode only the digital logic is power gated and only a small amount of that logic is retained, and that logic is powered by the main power supply to the CC3100. Waking the CC3100 out of hibernate mode requires an input command from the host driver and the wake up time is increased, around 50 ms. For the sake of our project, the CC3100 will be kept in a mostly active state; however, during periods inactivity we will keep the CC3100 in low-power deep sleep mode so the VESP can begin using wireless Wi-Fi connections immediately after activity is monitored as opposed to small but noticeable delay in wireless connectivity.

External Memory

Onboard the CC3100 is a proprietary file system stored as SFLASH memory. This file system allows for user allocation of file size though the API although the start of the memory that is used is always set at the start of the SFLASH. Communications with the microcontroller first go through the CC3100 file systems before interfacing with the SFLASH memory, which could delete or corrupt system files or user files stored in the SFLASH. Files that are held within the CC3100 have memory size allocated based on the order of download, meaning that file locations of user files or configuration files will be separate on two separate CC3100s. Files are recovered from the CC3100 in plain text so that users can understand the file, as opposed to the ID number of the specific file of interest. For safety any encrypted files or files undergoing decryption or encryption are hidden from the user and encrypted files can only be tampered with by directly going to the file system. The onboard memory of the CC3100 allows for a maximum space size of 16MB in an external memory device for an encrypted and security backed file, many of the files in the domain of our project will be smaller user configuration, service pack, web page and certificate files which are much smaller at around 2Mb assuming we don't use fail safe support, in which case the file sizes would double approximately. As a result of this we will require a small external memory drive to manage the files needed in communications with the CC3100. Thankfully, included on the CC3100 Booster Pack is a 8 pin flash memory card (Part number M25PX80-VMN6TP by Micron Technology Inc.). The attached memory chip has a total of 8 Mb of storage, which is more than sufficient to handle the wireless communication needs of our project so long as they remain small scale.

Pin Diagram

Shown below in Table 15 is the pin functions of the CC3100 Wi-Fi Module 64-Pin package. It measures a total of 81 mm^2 on the integrated circuit and has 16 pins per side with a grounding pad located on the chip itself.

Pin Number	Name	I/O Type	Function
1, 16, 17, 18, 19, 20, 26, 27, 28, 53, 63, 64	NC	N/A	Not Connected
2	nHIB	I	Hibernate Signal Connected to MCU
3, 29, 30	Reserved	N/A	Reserved for Future Use
4	FORCE_AP	I	Used to force access point mode, otherwise pull 100k resistor to ground
5	HOST_SPI_CLOCK	I	Host Interface SPI Clock
6	HOST_SPI_MOSI	I	Host Interface Data In
7	HOST_SPI_MISO	O	Host Interface SPI Data Out
8	HOST_SPI_CS	I	Host SPI Chip Select
9	VDD_DIG1	Power	Digital Supply Voltage (1.2 V)
10	VIN_IO1	Power	I/O Supply Voltage
11	FLASH_SPI_CLK	O	Flash SPI Clock
12	FLASH_SPI_MOSI	O	Flash SPI Data Out
13	FLASH_SPI_MISO	I	Flash SPI Data In
14	FLASH_SPI_CS	O	Flash SPI Chip Select
15	HOST_INTR	O	Output Interrupt
21	SOP2/TCXO_EN	I	Enable signal for external TCXO. 10k pull-up resistor to ground
22	WLAN_XTAL_N	Analog	WLAN 40 MHz XTAL
23	WLAN_XTAL_P	Analog	WLAN 40 MHz XTAL
24	VDD_PLL	Power	Internal PLL voltage (1.4 V)
25	LDO_IN2	Power	Internal LDO in-

As the wiring and construction surrounding the chip have been pre-fabricated by Texas Instruments, the most important things that the group is able to control are the input, output and the power pins in order to ensure the Wi-Fi module maintains the communication uptime that the project needs. The following table (Table 16) is a list of recommendations that the CC3100 should be operated under in order for the pins on the device to have full functionality.

Parameter	**Value**
Operating Voltage	2.1V - 3.6V with 3.3V typical
Operating Current	5 nA
Operating Temperature	-40-85°C
Output Voltage	2.4V high, 0.4V low
Temperature Slew	20°C per minute

Table 16: CC3100 Recommended Operating Conditions

Electrical Characteristics

Starting with the table below (Table 17) are several tables that show the electric characteristics of the CC3100 including, maximum ratings, recommended ratings, power consumption and current draw, UART characteristics and RF parameters.

Parameter	**Min**	**Max**
VBAT/VIO	-0.5 V	3.8 V
VBAT and VIO Differential		0 V
Digital Input	-0.5 V	0.5 V
RF and Analog Pins	-0.5 V	2.1 V
Operation Temperature	-40°C	80°C

Table 17: CC3100 Maximum Ratings

The CC3100 has a lifetime of up to ten years provided that the device is operated on a 20% duty cycle. It is safe to store in temperatures that exceed the maximum ratings; however, these temperatures will never be met during the course of the building, prototyping and demo of the VESP. The recommended operation values for voltage, current and temperature are located above in Table 16. The drain on power and the current ratings for the CC3100 are shown below in Table 18

Parameter	Typical Current Flow
Transmitting	Varies from 160 to 272 mA
Recieving	53 mA
Idle	690 uA
Hibernate	4 uA
Calibration	Varies from 450 to 700 mA

Table 18: Current Consumption from the CC3100

In Table 19 shown below are the total electrical characteristics of the CC3100 module including internal capacitances as well as output voltages.

Parameter	Min	Typical	Max
Capacitance of pins		4 pf	
Input Voltage High	0.65 x VDD		VDD + 0.5 V
Input Voltage Low	-0.5 V		0.35 x VDD
Input Current High		5 nA	
Input Current Low		5 nA	
Output Voltage High	2.4 V		
Output Voltage Low			0.4 V
Output Current High	6 mA		
Output Current High with Pull-Up	5 uA		
Output Current Low	6 mA		
Output Current Low with Pull-Up	5 uA		

Table 19: Total Electrical Characteristics

The UART connection on the CC3100 module has its settings set upon activation, requiring that the baud rate be adjusted to match what the group believes is the most effective baud rate for the VESP, a brief list of the initial UART settings are as follow.

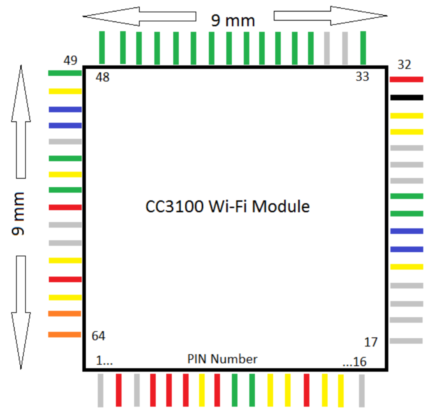


Figure 14: UART Connection Pins of the CC3100
Courtesy of Texas Instruments

- 115,200 Baud Rate, which can be configured up to 3 Mbps
- 8 Data bits, including one stop bit with no parity bits
- Clear to Send and Ready to Send flow control
- Little Endian formatting with the LSB sent first.

Reconfiguring the baud rate for the CC3100 is done using a programming command in the code for the device.

The physical connection for the CC3100 device to the host MCU can be performed using three different modes. All modes of operation use the pins shown in Figure 14 below.

The first of the three operation modes for the UART is the 5-Wire mode, which is the typically used choice for projects that implement the CC3100 UART. The transmit line of one UART is connected to the receive pins of the other device. The Clear to Send line of each device is connected to the Ready to Send pin on the other. The H_IRQ line of the CC3100 flows straight to the H_IRQ line of the MCU. The second mode of operation is the 4-Wire UART mode, the only difference between the 4-Wire mode and the 5-Wire mode is the H-IRQ line between the two devices is removed. The third and final UART connection mode that the CC3100 can operate using is the 3-Wire mode, in this mode both the H_IRQ is removed from each device and the Ready to Send line beginning from the MCU connecting to the Clear to Send line on the CC3100 is removed as well, forming the minimum number of pin connections needed for communications between the two devices. The group will operate the physical connection of the two devices in the 5-Wire mode, allowing for reliable communications between the MCU and the CC3100.

3.4.5 Bluetooth

After looking through the many options of Bluetooth solutions available on the present market the group decided to use the TI CC2564MODN Bluetooth module as the Bluetooth capability provider on the VESP. The price allows us a lot of room in the budget while still providing a high speed data rate that eclipses the other options that were looked into. The wide voltage range present will allow for easier voltage supplying from the group voltage and power supply while t uses a simple UART connection to allow straightforward connection to the MCU. As the CC2564 is classified as a low energy Bluetooth module it also saves us on power and heating within the detachable cube part of the VESP. Being from Texas Instruments is also a benefit as both it and the Wi-Fi module should be coded in a similar programming language helping to save a little time in the rush of time that is the build and prototype phase of the project. The only potential disadvantage to the CC2564 is a possible need for an additional PCB in order to manage the Bluetooth capabilities which will add more stuff into an already cramped working environment; however, TI has several reference designs and the materials needed to create them in the event that the group decides a slightly modified Bluetooth module is the best method to achieve our end goals for Bluetooth. Regardless the group has decided on the Bluetooth module we feel is the most suitable in terms of pricing, power and performance to utilize within our VESP and look forward to making it work.

Bluetooth Hardware

As with the Wi-Fi module located on the VESP, the Bluetooth module on the VESP requires knowledge of both the hardware capabilities of the device as well as the software and programming side of the device. In order to use Bluetooth to connect to other Bluetooth enabled devices, the group will have to work to ensure both sides are working well with the other components located on the VESP, such as the FPGA, MCU and Wi-Fi. From a software perspective the group has to ensure that Bluetooth, Wi-Fi, the MCU and the FPGA all are able to communicate fast and efficient with one another. Figure 15 below shows the top level block diagram, encompassing both hardware and software.

The main hardware points of the CC2564 are the two voltage inputs and the external antenna. While the main software points are the UART and PCM connections to the host MCU and the shutdown enable line. The hardware aspects and concepts found within the CC2564 will be discussed in the following subsections while the software sections of the CC2564 will be discussed in a later section.

CC2564 Pin Functions

The CC2564MODN Bluetooth module is a 33 pin, 7 mm by 7 mm chip, this chip is the heart of the VESP's Bluetooth communication abilities and without understanding how the chip itself works, the group will be unable to create any board needed to utilize the CC2564. Shown below in Table 20 is the pin functions of the CC2564.

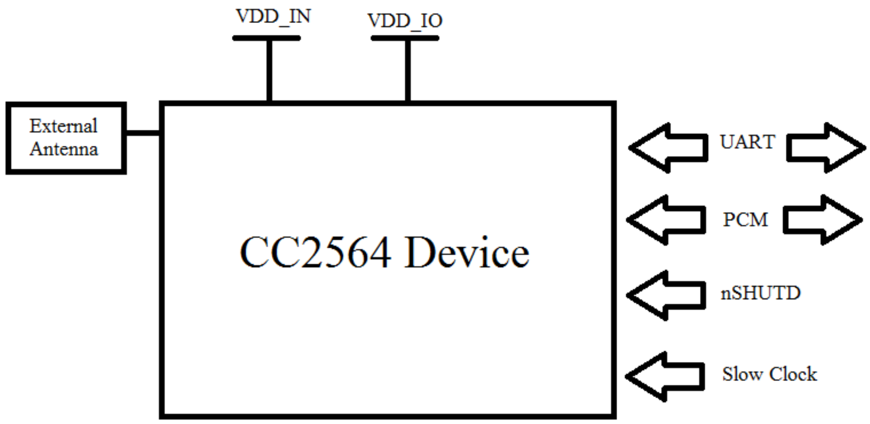


Figure 15: Functional Block Diagram of the CC2564 /newline Courtesy of Texas Instruments

Pin Number	Name	I/O Type	Function
1	HCI_CTS	I	UART Clear to Send signal (Send when low)
2	HCI_TX	O	UART Transmit
3	HCI_RX	I	UART Receive
4	HCI_RTS	O	UART Request to Send (Host sends when low)
5, 7, 9, 13, 15, 17	GND	N/A	Ground
6, 10, 11, 23	NC	N/A	Not Connected
8	SLOW_CLK_IN	I	32,768 kHz clock input
12	VDD_IN	I	Main power supply
14	BT_ANT	I/O	Bluetooth Antenna RF Input and Output
16	nSHUTD	I	Shutdown input when low
18	VDD_IO	I	I/O power supply (1.8 V)
19	AUD_IN	I	PCM Data in
20	AUD_OUT	O	PCM Data out
21	AUD_CLK	I/O	PCM Clock
22	AUD_FSYNC	I/O	Frame Sync
24	TX_DBG	IO	Debug messages and test point
25, 26, 27	GNDPAD	N/A	Ground Pad

Electrical Characteristics and Ratings

As with any and all electrical components there is a set of standard voltage and current inputs and outputs that will be seen from the device in use. This information helps the group design and build the power management systems within the VESP to ensure that the correct amount of power is going where it needs to go, the voltage and current drawn to the CC2564 are no exceptions in the following tables the ratings of the voltages and currents that drive the CC2564 is shown.

Parameter	**Value**
VDD_IN	-0.5 to 4.8 V
VDD_IO	-0.5 to 2.145 V
Analog Pin Input	-0.5 to 2.1 V
Other Pin Input	-0.5 to (VDD_IO + 0.5)

Table 21: Maximum Input and Pin Voltage Ratings

The numbers listed above are the maximum ratings, which are values the group hopes not to need to drive through the device. In addition to the maximum voltage ratings the CC2564 has maximum operating temperature range of 20 to 70°C , well beyond what the group sees as the possible temperature generated internally by the VESP. The CC2564 also is able to withstand 10 dBm of Bluetooth RF inputs before the stress on the device may cause damage to the device. If the CC2564 is exposed to a duty cycle of 25 percent uptime and 75 percent downtime, it is rated to have a life expectancy of 7 years assuming a 70°C ambient temperature. While the maximum rating of the CC2564 are listed above, the recommended operating conditions stated in the CC2564 datasheet as well as the settings that the group will most likely operate the CC2564 under are located below in Table 22

Parameter	Min	Max
Supply Voltage	2.2 V	4.8 V
I/O Supply Voltage	1.62 V	1.92 V
Input High-Level Voltage	0.65 x I/O Supply Voltage	I/O Supply Voltage + 0.5 V
Input Low-Level Voltage	0 V	0.35 x I/O Supply Voltage + 0.5 V
Input Asynch Rise Fall Time	1 ns	10 ns
Input PCM Rise Fall Time	1 ns	2.5 ns
Ambient Operating Temperature	-20°C	70°C

Table 22: CC2564 Recommended Operating Settings

As with all electric components located within the VESP the CC2564 also has power consumption due to current input and output from the device. A list of static current draws is located below in Table 23 These number indicate a far greater current draw coming from the Wi-Fi module when compared to the Bluetooth module.

Parameter	Min	Max
Shutdown	1 uA	7 uA
Deep Sleep	40 uA	105 uA
I/O Active Current Consumption		1 mA + 0.5 V
EDR Transmission		112.5 mA + 0.5 V

Table 23: Static Current Characteristics

As with the Wi-Fi module there are timing characteristics and UART speeds that must be considered in the VESP. A truncated list of these timings and speeds are listed below in Table 24.

Parameter	Min	Max
Baud Rate	37.5 kbps	4000 kbps
Baud byte accuracy	-2.50%	1.50%
Baud bit accuracy	-12.50%	12.50%
Propagation Time (Assuming 40 pf load)	0 ns	10 ns
Cycle Time	64 kHz	4.096 MHz

Table 24: UART and Timing Characteristics

The final table presented shows an overall look at the electrical characteristics including both input and output voltages, resistance and input capacitance. This table is located below as Table 25

Parameter	Conditions	Min	Max
Output Voltage High	Current of 2, 4 or 8 mA	0.8 x I/O Supply Voltage	I/O Supply Voltage
	Current of 0.1 mA	I/O Supply Voltage – 0.2	I/O Supply Voltage
Output Voltage Low	Current of 2, 4 or 8 mA	0 V	0.2 x I/O Supply Voltage
	Current of 0.1 mA	0 V	0.2 V
I/O Impedance	Resistance	1 MOhm	
	Capacitance		5 pf

Table 25: Output Voltages and Impedance of CC2564

Bluetooth Hardware Characteristics and Features

As a fully functional Bluetooth solution there are several features and modes that are supported within the CC2564, some of which will be utilized in order to enable Bluetooth communication abilities within the VESP. The CC2564 is always able to utilize its enhanced data rate speed which supports 7 devices and 3 piconets. Internal detection allows for the CC2564 to manage its temperature in order to ensure minimum variation within the RF parameters of the CC2564. The CC2564 also has built-in support for all voice coding and has several parameters that can be modified to adjust the pulse code modulation and inter-IC-sound interface. When utilized in the LE mode the CC2564 is optimized for sports environment use. Data rate is unaffected due to independent channel buffering allowing multiple connections to maintain a constant speed. Ten connections are supported through LE mode and power consumption is reduced through tightly coupled sniffs.

The CC2564 Module features several transport layers that interact with each other as follows. The RF link controller has a two way data interface with the Link manager and HCI data handler while receiving control commands and giving event commands from and to the Link manager. The HCI data handler transmits data to and from the host controller interface, which is a part of the UART transport layer. Control and Event commands from the Link manager pass through the HCI command handler, control commands are sent down from the Host Controller Interface through the command handler and link manager to the Link controller. While Event commands travel up from the link controller through the link manager to the HCI command handler and ultimately to the Host Controller Interface itself.

CC2564 UART Connection

The CC2564 HCI module contains one UART module devoted to act as the HCI transport controller and handles the event and control commands as well as the ACL of the host controller using sent data packets. These UART data packets compose the maximum 4 kbps that the CC2564 is capable of releasing; however, it is set to 115200 bps as a default and must be programmed in the HCI to allow for a higher Baud rate. However, the command that changes the baud rate from the default 115.2 kbps to its new value is an event command that will run at the 115.2 kbps speed. The total default settings for the UART module of the HCI of the CC2564 is that the baud rate is set to, obviously, 115.2 kbps with a 1 byte data length, 1 stop bit with no parity bits for error checking. The transport layer of the UART module has two distinct modes of operation: the first is a four signal UART transport layer, while the second is a three signal 3-Wire UART transport layer. The difference between the two is found in the hardware connections. In both cases there exists a clear to send, a ready to send, transmit and receive pin on both the UART module and the host controller. In the case of the normal four signal layer the receive pin of one device is connected to the transmit pin of the other, while the clear to send is connected to the ready to send of the other device. In the case of the three wire method, the two devices share a common ground while transmit and receive are still connected as they are in the four signal approach. The clear to send and ready to send pins have no impact and are not used in the case of the 3-Wire method of connection. A general image of what the used pins are for both the CC2564 and the Host device is shown below in Figure 16.

This figure shows that the connections between the CC2564 and the Host controller and the connection between the CC3100 and the MCU are both very similar in what signals are needed to allow communication between the two devices.

CC2564 Digital Codec

The CC2564 is equipped with an onboard coder-decoder, known as a codec, which helps the Bluetooth module integrate with other PCM or I2S devices and is an in-

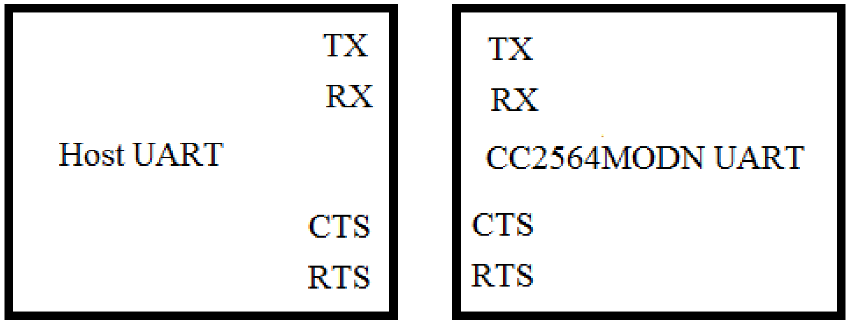


Figure 16: UART Connection Pins of the CC2564 /newline Courtesy of Texas Instruments

tegral part to managing audio and voice communication between the CC2564 and other devices. The digital codec allows the CC2564 to operate two voice channels as either master device or slave device and supports as Bluetooth voice communication standards today including linear, A-law and u-law. The codec is located as a fully programmable port on the CC2564 allowing it to be changed as needed through coding. The codec had a four signal interfacing with the device that it is communicating with; two of these signals, the Clock and Frame_Sync signals determines the CC2564 is operating as master or slave device, it is master when the CC2564 generates these signals and slave when it is receiving them. The other two hardware interfacing signals are simply the signal for data input and data output. Even though the CC2564 inside of the VESP will handle audio, the group does not believe that the codec in the CC2564 will be used as we will not use the VESP for voice communication making a voice codec kind of pointless; however, the group still looked into it as it may become needed in the VESP later for managing music and audio output for the device.

3.4.6 FPGA

After much consideration, the VESP team decided to work with the ODROID-XU3. We considered many factors, namely: cost, size of PCB, power efficiency, processing power, 3D graphics API support, I/O, and display interface. The team found that the ODROID-XU3, while being the most expensive option, was the most ideal as well. Not only does the ODROID-XU3 have the most powerful CPU/GPU configuration amongst the candidates, but it also has built-in performance optimization algorithms to maximize energy efficiency. Also, the ODROID-XU3 has the most support for 3D graphics APIs, including OpenGL 1.1, 2.0, 3.0, 3.1 and DirectX 11.

Overall, the ODROID-XU3 is a step-up from the other candidates with more display connectivity, more than sufficient I/O support, all in a reasonably sized PCB. Our second choice, the ODROID-U3, would have also been a good choice, however the VESP team felt that its processing power might not be enough to run the VESP smoothly and instead went for a safer bet with the ODROID-XU3. Finally, using the BeagleBone Black for the VESP was simply out of the question; it lacked 3D API support and nowhere near the processing power to sufficiently drive the VESP at desired speeds.

3.4.7 Battery

When selecting the type of battery to use for VESP, the group decided to get a battery that has a high capacity, comes in small and different shapes and also with small weight. Based on the description of each battery in Table **??** , the group decided to use the lithium ion polymer battery. Even though its price is the highest one, the group wanted a lightweight battery. The group didn't want for the user to carry a heavy device when it was not connected to the dock.

	NiMH	Lead Acid	Li-Ion	Li-ion Polymer
Energy Density	60-120	30-50	110-160	100-130
Cycle Life	300 to 500	200 to 300	500 to 1000	300 to 500
Fast Charge Time	2-4h	8-16h	2-4h	2-4h
Overcharge Tolerance	Low	High	Very Low	Low
Operating Temperature (C^0)	-20 to 60	-20 to 60	-20 to 60	0 to 60
Maintenance Requirement	60 to 90 days	3 to 6 months	Not req.	Not req.
Cost (7.2V)	$60	$25	$100	$100
Cost Per Cycle	$0.12	$0.10	$0.14	$0.29
Year Used Since	1990	1970	1991	1999

Table 26: Characteristics of NiMH, Lead Acid, Li-Ion, and Li-Ion Polymer rechargeable batteries

3.4.8 Battery Charger

Now that the group decided which battery will be used to power the components of the VESP when is not connected, the battery charger was the next step to find. The group found out that the charging depended on the size and the type of battery that it was being used. We could not use the same battery charger as the lead acid of the nickel-metal hydride because they are compose of different materials which means that those batteries need different type of charge rate and also some have a high tolerance for overcharging which is not case for the lithium-ion polymer battery that we chose.

When deciding how fast the group wanted to the battery to charge, the group needed to understand what how did the charge rate work or often known as ‘C’ or ‘C-rate’. For example, a 4.0Ah battery can have a charge rate of 2C which means that its charge rate is 8.0A, meaning it would take half hour to charge the battery. If we take that same battery example and we use the charge rate of 0.5C which means that its charge rate is 2A, meaning it would take two hours to charge the whole battery. This is of course when accounting that the battery is one hundred percent efficient.

The group also found out that lithium ion and lithium ion polymer batteries have the same aspects when it comes to charging and discharging. Due to lack of information of how lithium polymer ion batteries are charged, the group is going to explain base on how lithium ion batteries are charged. First off, one of the big challenges with lithium batteries are overcharging or over discharging because it can result as a physical damage to the battery, and also it decreases its capacity and potential causing a fire hazard to the device. Besides that, the group found out that in order to safely charge the lithium battery, it needed to go through four stages. The first stage is considered the fast stage where the voltage increases maintaining a constant current. Also, taking in consideration that at this stage it should not exceed one third of the overall battery capacity. For example, if the battery has a capacity of 1200mAh, then it should not be pushed to pass 400mA. The second stage is known as the saturation charge which is the stage that takes the longest. At the beginning of this stage, the voltage threshold and it’s about eighty-five percent of the battery. At this point the current decreases and it terminates when the current is less than three percent of the rated current. After this, the stage three starts is when there is no current and the charge terminates, but the voltage begins to drop. The stage 4 is basically called the ‘standby mode’ which means there is a small current going on to occasionally top charge when the voltage starts dropping as you can see below in Figure 17. This continuous top-off can reduce the battery life, which there is integrated chips in the charging system that prevents from overcharging, overheating. This four stages occur when the battery is not fully discharged. When the battery is fully discharged, it must be trickled charge at first with very small current before proceeding to the first stage.

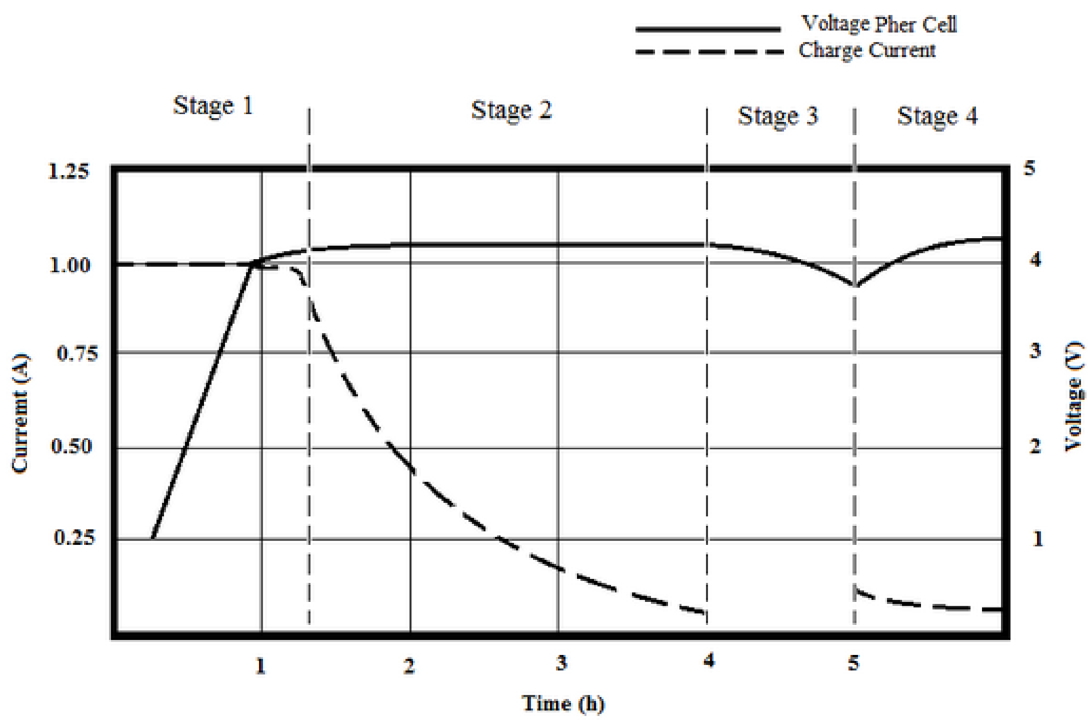


Figure 17: Charging states of lithium-ion

3.4.9 Sensors

The project design for the VESP is going to require it to have several sensors. These sensors will be utilized by the VESP to determine when a user is present, how bright to make the screens to increase power efficiency, and what the current orientation of the "cube" portion is when a user takes the device off the dock. Selection of these sensors must regard functionality, power usage, and cost among other factors. The sensors must also be able to connect with the microcontroller so their connection type and interfacing methods will be taken into consideration.

Proximity/Motion Sensor Selection

The first sensor selection to be made will be for the motion sensor. This sensor will be the primary method used for determining the presence of a user. For example, when a user enters the room the VESP is located in, this sensor must be able to recognize the event. This means that the range of the sensor should be able to cover the distance of an average household room. Of the researched options, the best fit for the project will be the motion sensor for Panasonic.biz (part number EKMC1601111). One of the main specifications that makes this selection a better choice is the range of sensing it reaches. This sensor can sense movement of up to 5 meters away (16.4 feet) which is far greater and far more ideal for the project compared to the other researched option (which could only sense 2 meters away). However, this sensor can only sense in one direction, in sort of a cone shape in the way it's facing at about 82 degrees (which can be seen in Figure 6). A solution to this would be to purchase several of these sensors and face them in multiple directions to approach a combine total of 360 degrees of sensing. This component operates on a supply voltage of 3V-6V which keeps it within the scope of the projects voltage preferences. This sensor sells for $10.26 per unit so if the project group decides that using four units to approach a 360 degree of sensing is the necessary solution (Figure 18), the cost would be $41.04 and must be taken into consideration because that would exceed the current budget for the motion sensor portion. Another factor that makes this selection a strong choice is the digital output it provides helps to easily integrate the component with the microcontroller. The sensor has three terminals (as seen in Figure 5): Gnd, Vdd, and Output. The output will connect to one of the pins of the microcontroller, leaving plenty of hardware room for other sensors and components to connect to the microcontroller. If four sensors are used, they would either require 4 pins total or perhaps first go through some intermediate OR gate configuration and the final output would connect to the microcontroller.

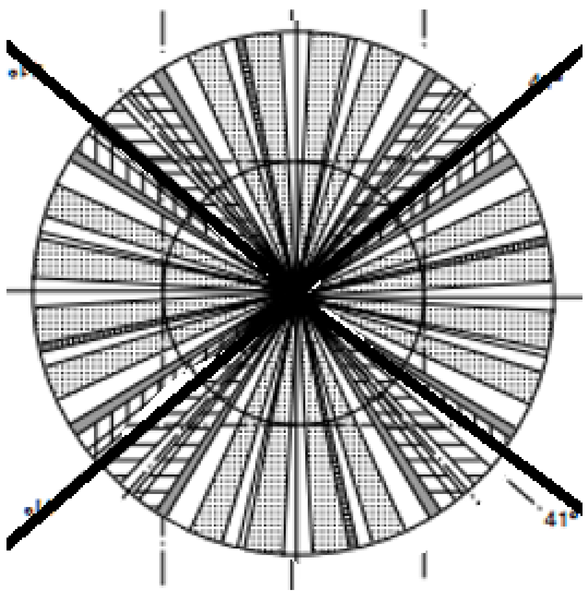


Figure 18: Four IR Sensors Combined to Create Full Surrounding Coverage

Ambient Light Sensor Selection

The ambient light sensor will be applied to the VESP for multiple tasks. The first of which would be to assist in determining when a user enters the room. This would be implemented by having the sensor detect a sudden and drastic change in lighting, such as a user entering a dark room the VESP is located in and turning on a light switch, thus notifying the device that a user is near. The second implementation of the sensor will be to determine the current lighting of its surroundings during use, and then to adjust the brightness of the LCD screens accordingly. If the VESP is being used in a well-lit space, the screens would stay bright enough for the user to easily see the them; if the user were to move to a darker area, the sensor would let the processor know to turn down the brightness of the screens to help preserve power and therefore prolong the life of the battery when the device is not on the dock. The selection for the ambient light sensor must consider these responsibilities for the component, while also remaining in the allotted budget amount. Additionally, the selected sensor will need to run on a relatively lower supply voltage (also to help preserve battery life as well as keep heat generation of the device low).

The ambient light sensor selected by the project group will be the sensor found on adafruit.com (part number TSL2561). This sensor is more than capable of filling the role of detecting a sudden and drastic change in the surrounding lighting. It can detect ranges from .1 to 40,000 Lux which includes all forms of light visible to the human eye. This sensor can be configured to measure either infrared, full-spectrum, or human-visible light, however, the project will likely only utilize the human-visible light configuration as that would be helpfully precise in determining the appropriate brightness for the LCD screens for the user to easily see. The interfacing method of this sensor will be digitally through an I2C bus which will be compatible with the selected microcontroller. The voltage supply range of this sensor is also within the specifications of the project at a supply range of 2.7V-3.6V, keeping it relatively low and not too draining on the battery life. The current draw of this component is .5 mA when actively sensing and smaller than 15 uA when in power-down mode (which, if utilized for the project, will help to further increase the battery life). This sensor sells for $5.95 which is relatively low and does not greatly impact the current budget for sensors. Table 27 can be used to quickly reference the primary information.

Part Number	TSL2561
Sensor Type	Ambient Light
Voltage Supply	2.7V-3.6V
Output Type	Digital
Interfacing	I2C
Cost	$5.95

Table 27: TSL2561 Data Sheet

Gyroscope/Accelerometer Selection

The specifications for the selection of the gyroscope have to meet a few requirements to be implemented into the project. The first, and most important specification, the gyroscope must be able to accurately communicate the orientation of the VESP when it is off the dock portion. As the user holds the VESP they may switch around which screen is facing upwards at them and, if they power saving mode is activated, all the away-facing screens would be turned off. The front facing screen would be determined by the information given by the gyroscope. Another specification the selected sensor should me is that it must operate at a relatively low voltage to help preserve battery life as well as reduce heat output created internally in the VESP. Additionally, the sensor must remain within the budget for sensors. Accelerometer capabilities could open a lot of possibilities for the project, however, the current specifications for the project do not require them. These capabilities would just be additional perks that could be implemented in a possible future, improved version of the VESP. The research showed that gyroscopes and accelerometers can be sold as separate sensors, but typically they are both in the same component without a noticeable difference in price.

The selected component will be from InvenSense.com (part number MPU-3050). This decision was easy because this sensor acts as both a gyroscope and an accelerometer. As mentioned before, the project does not require the capabilities of an accelerometer, but this additional feature of the sensor is included and the price difference between this unit and the other research gyroscope is less than $3.00 at a price of $9.09. The component has X, Y, and Z-axis angular rate sensors on one integrated circuit which then connects to a digital accelerometer through one I2C bus and connects to the microcontroller through a second I2C or an SPI connection (as seen in Figure 7). This sensor is powered by a voltage supply of 2.1V-3.6V, keeping it in the ideal range for the project and its max power consumption does not exceed 6.1 mA. Additionally, its sleep mode only requires 5 uA which the project will utilize as the sensor only needs to be active when the VESP is off its dock. This gyroscope is low cost, runs on a low voltage supply, and the included accelerometer opens a lot of options for expansion of the project making this sensor the best choice out of the researched components. Table 28 can be used to quickly reference the primary information of this component.

Part Number	MPU-3050
Sensor Type	Accelerometer/Gyroscope
Voltage Supply (V)	2.1-3.6
Output Type	Digital
Interfacing	I2C/SPI
Cost	$9.09

Table 28: MPU-3050 Data Sheet
Image Use Courtesy of InvenSense

3.4.10 Speakers

When choosing for the right speaker, the group had limitations because the base of VESP had a width of three inches. Besides that, the group was looking for a good quality speaker with the diameter of 2 inches to 3 inches. Unfortunately, there were few types of speaker with this qualifications but the two types that stood out was the AuraSound from Parts Express website and the Mid-Range speaker from the MCM Electronics. Table 29 shows the specifications of each speaker.

	Mid-Range	**AuraSound**
Frequency Response	150Hz - 20kHz	200HZ - 20kHz
Impedance	8 ohms	4 ohms
Size	2”	2”
Power, RMS	5W	7W
Power, max	8W	15W
Sound Level	84dB	84dB
Price	$ 7.20	$9.19

Table 29: Speakers Specifications

Both speakers almost have the same description but with the difference that the AuraSound has a larger maximum power and RMS due to its low resistance. At the same time both produce the same sound level of 84dB. Because of the price and because it is hard to find an amplifier with that will give a power of 5W with 8 ohms, the group decided to pick the AuraSound speaker.

Now, we learned about the types of amplifiers at the beginning of this project and we found out that the most efficient and cheapest amplifier are the Class D amplifiers. Because of that, the group is gonna use a PAM8610 2x10W Class-D Audio Amplifier Board from the Parts Express website. This type of amplifier has a efficiency of 90 percent with a frequency response of 20Hz to 50kHz. It has power requirements of 7 to 15VDC and a quiescent current of 20mA. Also, it has a audio control for a higher sound.

3.4.11 Housing

When deciding the type of material that we will be using for the Base Housing of the project, the group decided to choose wood for many reasons. One because its rigid enough that will support the four LCDs screen and its components and at the same time it will not bend. Two because it will be easier to make holes for the circuits and because wood does not conduct current. Finally, three because wood will give a nice finish to the project. A good design that will contrast the LCD screens on the top.

3.4.12 Operating System

In the end, the team decided to go with Ubuntu as the operating system for the VESP. Choosing Android over a Linux distro for the operating system would have introduced tou much overhead and would have made the VESP considerably less energy-efficient and could have potentially affected the user experience. Both Arch Linux and Debian would have been more energy-efficient than Ubuntu, but the team felt that the drawback of a larger learning curve outweighed that advantage. With Ubuntu, installing and developing apps would be far simpler and take considerably less time with its user-friendly nature. In addition, the possibility of being able to implement the Phone version of Ubuntu into the VESP is very appealing and would be an easy transition from regular Ubuntu.

3.4.13 API

Out of all the APIs discussed, Qt and OpenGL will be used to supplement the VESP's software development. For the sake of integration and time, the team decided to not use OpenCL. This is due to the fact that it's not absolutely essential to the VESP's operation, and currently Qt does not support OpenCL, which could potentially cost the team precious time. Qt will be used as the framework for all of the VESP's software, while OpenGL will be used to create any 3D animations that the VESP's software might include.

4 Project Hardware and Software Design Details

The following section will be dedicated to the design details of the VESP. The overall design approach will be discussed, followed by an high-level overview of how the various components of the VESP will interact with each other, along with the various software components. Each major aspect of the VESP will then be discussed in detail.

4.1 Design Paradigms

When designing the VESP, the team wanted to prioritize efficiency and scalability. By doing so, the VESP will be able to run smoothly, and the addition of future features won't require a complete or significant redesign of the system. The VESP described in this document is intended to be a first step, and so a scalable design paradigm would allow the VESP project to expand and become even more useful and entertaining.

4.1.1 Hardware

When integrating all the various sensors, the MCUs, the FPGA, the communication modules and the LCDs, the team wanted to prioritize modularity, while also taking into consideration space. One reason for this is that we wanted to be able to expand on the hardware, as previously stated. Another reason is that during the development process, a lot of issues can be easily narrowed down, and troubleshooting in general is significantly easier when the hardware is split up into individual parts that interface with each other. And finally, due to the necessity of all hardware components having to fit within a tight and enclosed space, the team cannot have too many separate hardware components.

For these reasons, the team decided to have one MCU handle the raw sensor data from all the different sensors and convert it to a format that the FPGA can take as input. This design decision balances the modularity requirement and the space limitation, as well as takes some of the load from the FPGA, reducing overhead. The team also decided to have the FPGA handle all of the wireless communication modules (Wi-Fi and Bluetooth); this way the team saves valuable space by not having another MCU handle the wireless data. Another important design decision is to have a separate IC module handle the DisplayPort output from the FPGA to distribute amongst the four LCD screens. The integration of the four LCD screen will arguably be the most challenging aspect of the VESP project, so separating that piece will be advantageous during development.

4.1.2 Software

When designing the software side of the VESP, again, the team wanted to prioritize modularity, but this time, the user experience was also to be prioritized. Software for the VESP must be easy to use and intuitive in order to make it user-friendly, and so when developing the applications and UI, a number of design paradigm must be followed. These paradigm are as follows: touch buttons must be easily accessible, but also not in the way to avoid unintentional input, for utility apps screen space must be used to display as much information as possible, for entertainment apps screen space must be used to display as many visuals as possible, device settings must always be accessible, and finally transition between apps should be as seamless as possible. If all of these design paradigms are followed, the VESP will be a very user-friendly device.

4.2 Project Block Diagrams

Figure 19 below shows a block diagram of the highest level view of the entire VESP system and shows how the VESP device will interact when connected to the docking station. The docking station will contain the power supply, powered by an AC outlet, as well as the audio system for playing audio data coming from the VESP device. The power supply in the docking station will charge the battery found in the VESP device, the battery itself will power all of the hardware components (sensors, FPGA, LCDs, MCUs, etc.).

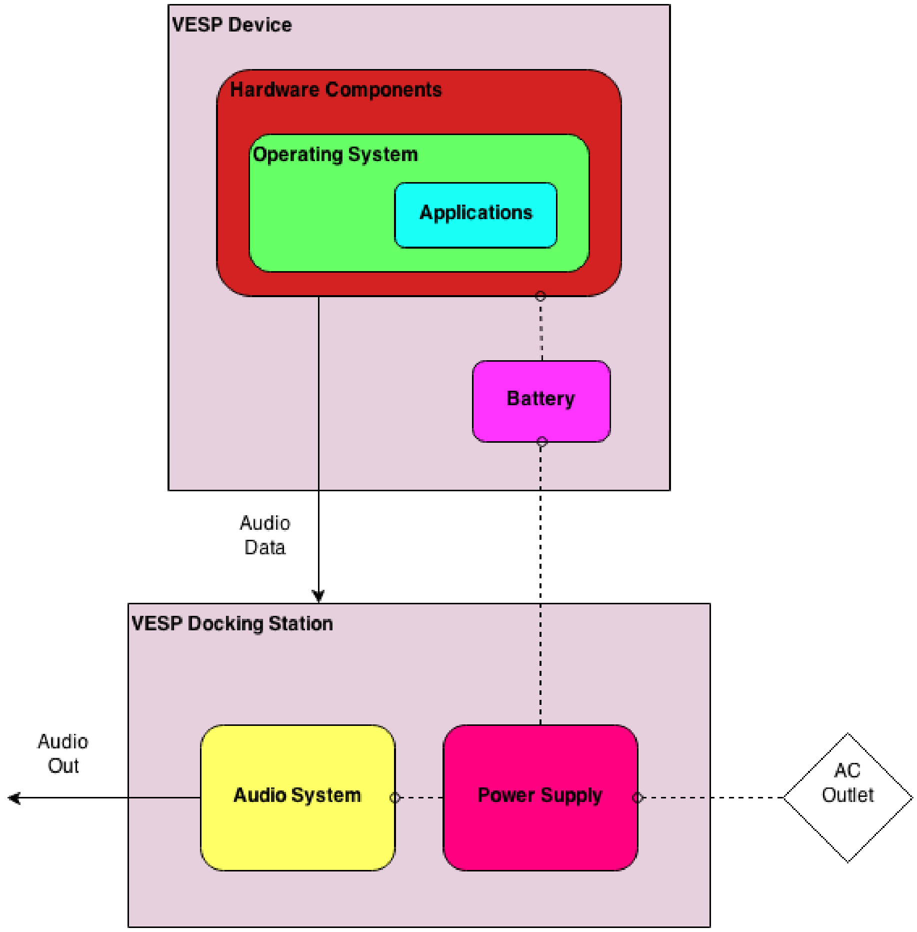


Figure 19: Entire System Block Diagram of the VESP.

4.2.1 VESP Block Diagram

Figure 20 below shows the hardware block diagram of the VESP. The four main components of the VESP are the FPGA, the MCU, the Wi-Fi and Bluetooth module, and the LCD Controller. The gyroscope/accelerometer, ambient light, and IR sensors will all be interfaced with the MCU. The MCU will interpret all the raw sensor data and then transfer the appropriate data to the FPGA or the LCD controller. The FPGA will handle the bulk of the computations necessary to drive the VESP including sensor control signals, Wi-Fi and Bluetooth data, touch control signals, and audio/video outputs. The LCD Controller will handle the video data coming from the FPGA, as well as the brightness control signal from the MCU and the raw touch sensor data. The Wi-Fi and Bluetooth module will handle all incoming raw wireless data coming from sources, as well as outgoing requests from the FPGA.

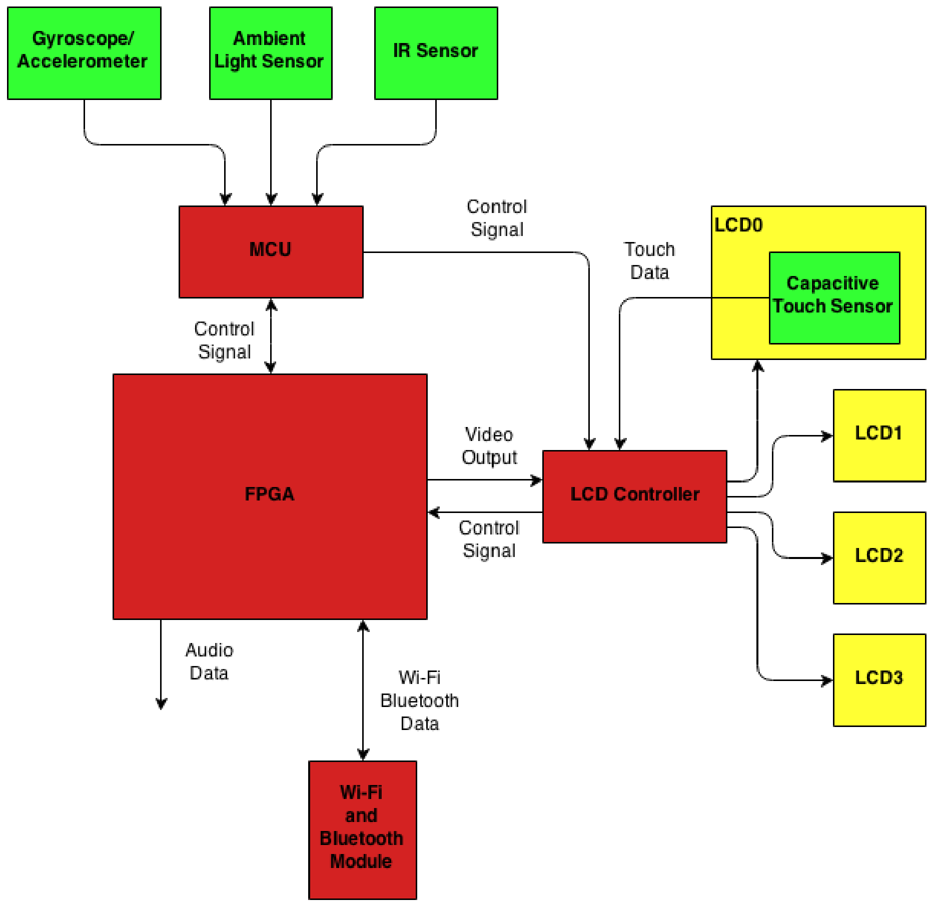


Figure 20: Hardware Block Diagram of the VESP device.

4.2.2 Software Block Diagram

Figure 21 below shows the software block diagram of the VESP. Starting with the MCU, the raw sensor data will be taken as input and processed to be interpretable by the FPGA. Processed data from the gyroscope\accelerometer will be used be used by the FPGA to keep track of the devices current orientation and change thereof, which will be used as user input. Processed data from the ambient light sensor will be used by the LCD Controller to control the brightness of the LCD screens. Processed data from the IR proximity sensor will be used as control signals for the Linux operating system to enter and exit to\from sleep mode. The LCD Controller will handle both the video data coming from the FPGA and the user touch sensor data from the capacitive touch panel. The video data will be partitioned in order to be distributed to the four LCD screens, and the user touch sensor data will be sent to the Linux operating system. Wi-Fi and Bluetooth data will be sent to the Linux operating systems to be used by the various applications of the VESP. Within the Linux operating system, the Qt API will used as a wrapper for the various applications.

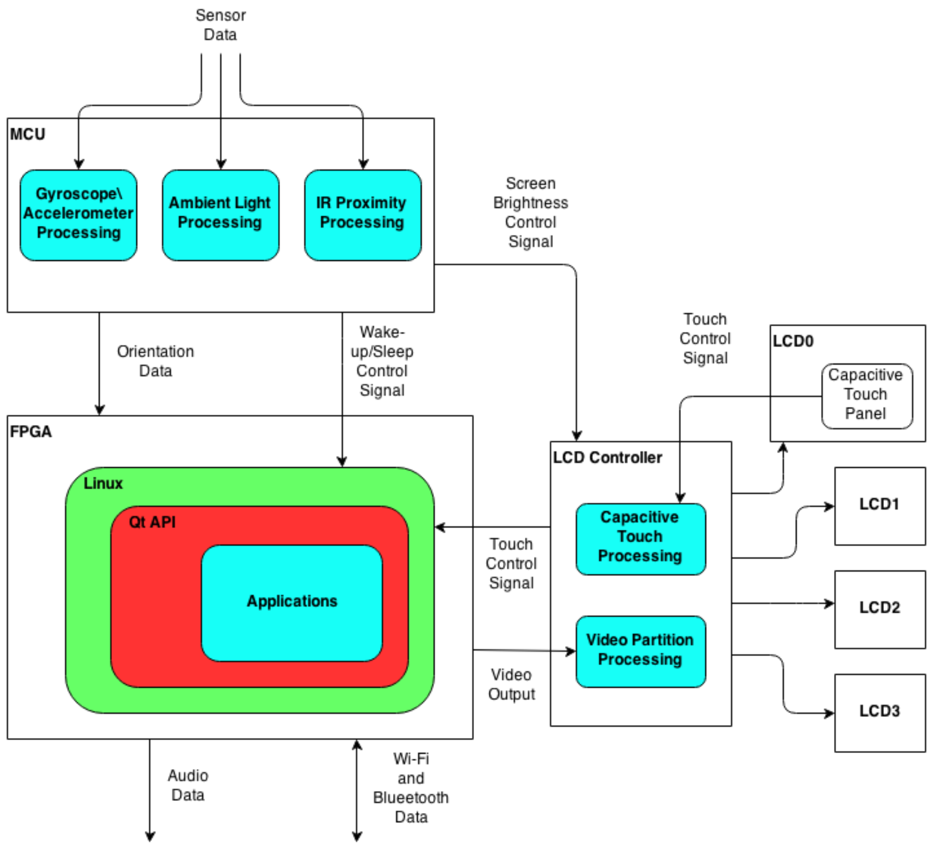


Figure 21: Software Block Diagram of the VESP device

4.3 LCD Panel Subsystem

Figure 22 will be used to reference all of the different dimensions for the LCD panels, most importantly the perimeter dimensions which are used to determine the size of the "cube" portion of the VESP. Figure 23 shows the dimensions of the capacitive touch panel that will lay over one of the LCD screens. It should be noted that in both figures, the "front" view is shown, as the corner which has the connection must be noted for design of the VESP.

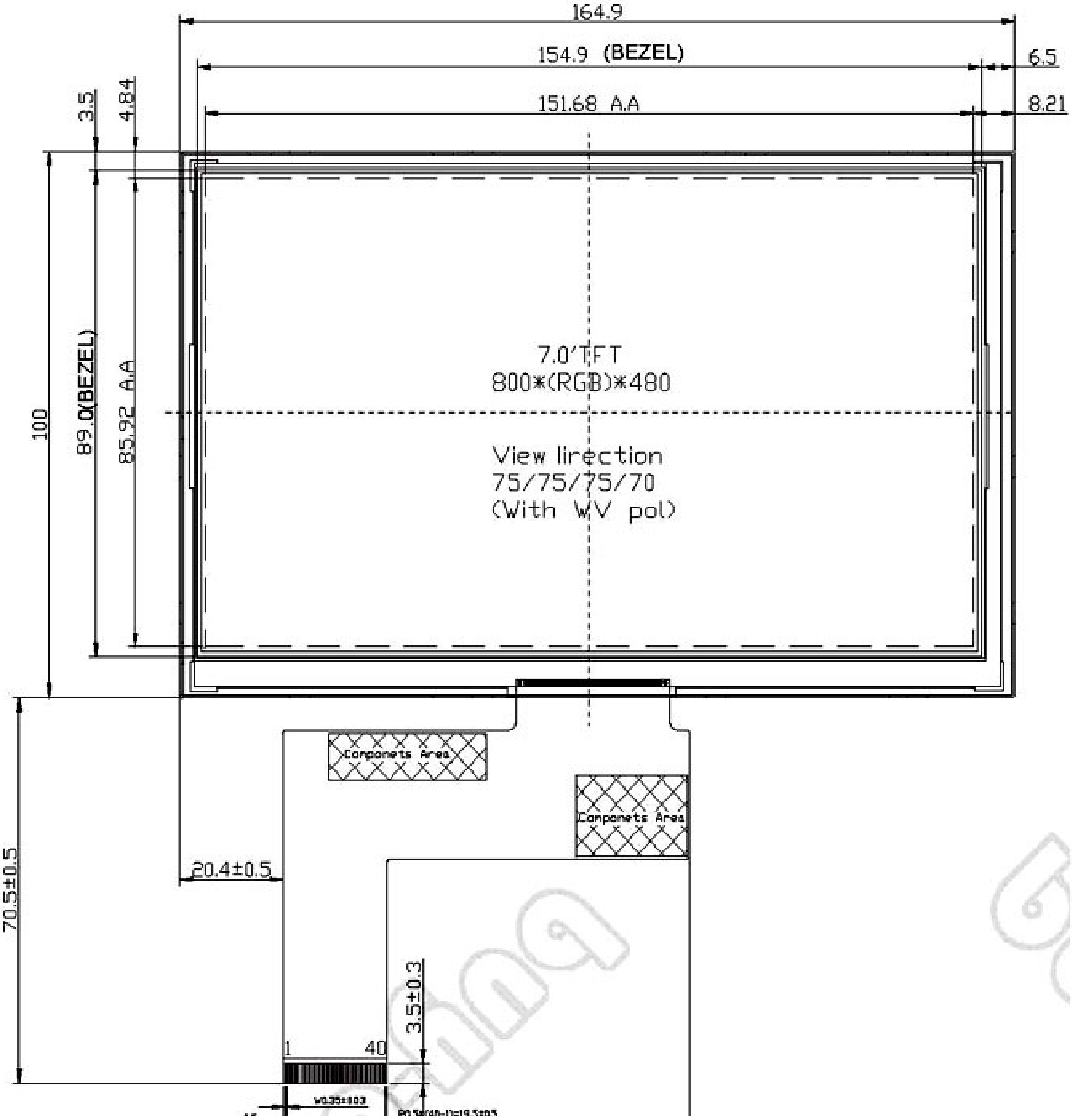


Figure 22: LCD screen dimensions.
Image use courtesy of BuyDisplay.

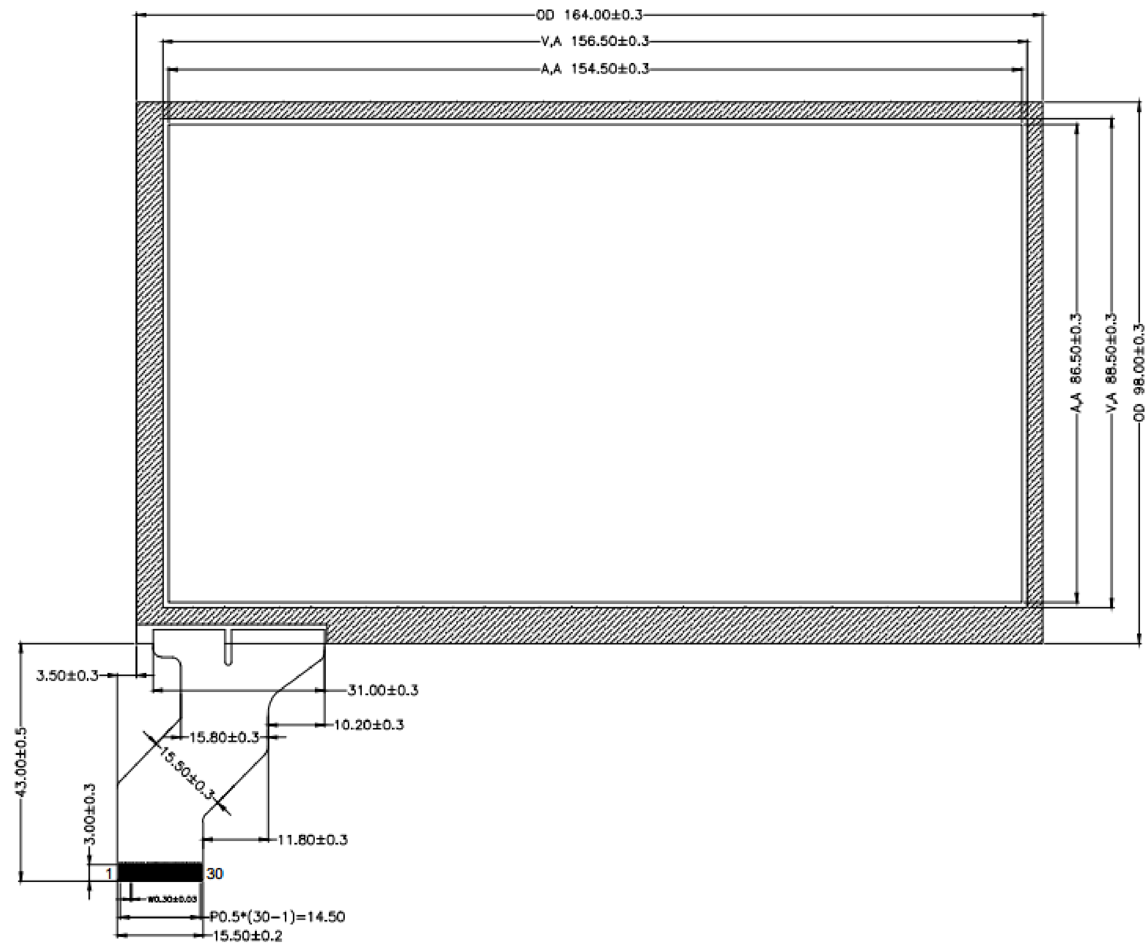


Figure 23: Capacitive touch panel dimensions.
Image use courtesy of BuyDisplay.

4.4 LCD Controller Subsystem

4.4.1 Overview

One of the main aspects for the VESP is that it's composed of four LCD screens. In order for the VESP to function as the team wishes, the VESP must be able to output to each LCD screen independently. This requires that the video output coming from the FPGA be partitioned appropriately. Also, one of the LCD screens must be touch enabled in order for the user to be able to interact with the device. In addition, the brightness setting of the LCD screens must also be controlled by control signals from the MCU. All of these tasks are to be handled by the LCD Controller. Designing the LCD Controller is probably one of the most challenging parts of the entire VESP project, as there is no available hardware that does all of the tasks that were outlined previously and so it must be custom made.

Figure 24 shows a high-level view of how the LCD Controller will ultimately work. The main component of the LCD Controller is the 1:4 Serial Link De-Aggregator that takes in the video output data from the FPGA and de-muxes the data into four separate signals. The split signals are then decoded into a format that the LCD driver can interpret and finally display on to each screen. The other component of the LCD Controller is the capacitive touch controller that takes in the raw touch sensor data from the capacitive touch panel, processes the data, and sends it to the FPGA. Finally, the backlight brightness control signal coming from the MCU is sent to each LCD driver.

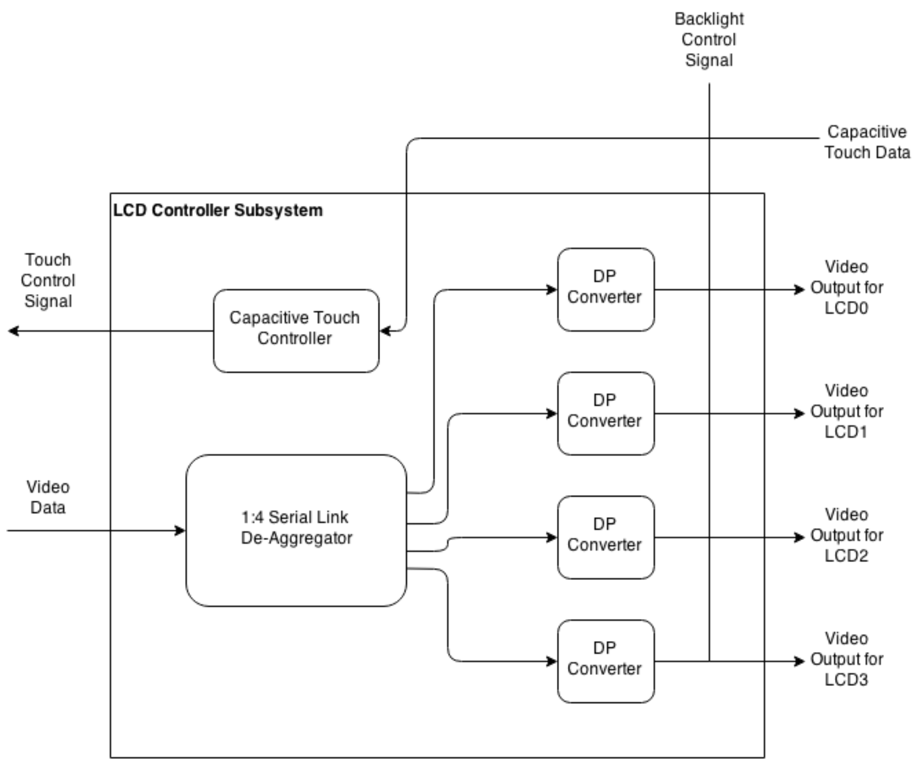


Figure 24: LCD Controller Block Diagram.

4.4.2 Components

For the purpose of de-aggregating the video output from the FPGA, the team opted for the TLK10022 from TI. The TLK10022 is a bi-directional dual-channel multi-rate serial link aggregator, it is able to combine up to four lower bandwidth signals into one high bandwidth signal, as well as the opposite, which is the desirable function. The idea is for the TLK10022 to de-aggregate the high-resolution (1920x1080) video into four low-resolution (800x480) video streams. Table 30 below shows the specifications for the TLK10022.

	TLK10022
Supply Voltage	1-1.8V
Input Compatibility	CML
Package	144FCBGA
# of Slow Speed Serial Channels Per Port (TX/RX)	4
Operating Temperature Range (°C)	-40 - 85
Output Compatibility	CML
# of High Speed Serial Ports (TX/RX)	2

Table 30: TI TLK10022 Specifications.

From the table, the TLK10022 has input and output compatibility for CML, the physical layer for the DisplayPort interface, which will be the interface of choice for the VESP. This makes the TLK10022 ideal for what the team wants to achieve. In addition, the operating temperature range for the TLK10022 is superb, eliminating the concern that the IC might overheat.

Once the original video signal from the FPGA is split up into four separate DisplayPort signals, they must be converted into RGB format in order to be interpreted by the LCD driver. The team opted for the STDP4020 by STMicroelectronics for this purpose. Table 31 below shows the specifications for the STDP4020.

	STDP4020
Supply Voltage	3.3V I/O; 1.2V core
Bandwidth Per Lane (Gbps)	1.62 - 3.24
Output Compatibility	LVTTL 60/48b wide and QLVDS 8/10b per color
Color Format Support	RGB and YCC
Supported Interfaces	I2C
Package	164LFBGA
Power	900 mW Max, 20mW on standby

Table 31: STDP4020 DisplayPort Converter Specifications [**?**]

The STDP4020 is a powerful DisplayPort converter that is capable of decoding DisplayPort signals into RGB format, which is what the team needs in order to interface with the display driver. Each STDP4020 on the IC will only need to decode DisplayPort signals for a 480p resolution display at 60 Hz, which translates to about

0.184 Gbps, making its performance specification overkill for its purpose. However, this is not an issue, the STDP4020 can operate at low pixel rates and removes the concern of the STDP4020 being overworked. Also, for the purpose of displaying four different images, it is desirable to be able to decode the four DisplayPort signals in parallel to avoid syncing issues.

As it was mentioned before, one of the LCD screens of the VESP will be touch-enabled, in order to achieve this, a touch sensor controller must be implemented to handle the analog signal from the touch panel. For this purpose, the team decided on the STMT07 capacitive touchscreen controller from STMicroelectronics. Table 32 below shows the specifications for the STMT07.

	STMT07
Power Supply Scheme	2.8 - 5V Analog & 1.8V Digital
Supported Screen Size	Up to 10 inches
Power Consumption	5 uW (sleep) & 13 mW (active)
Report Rate	>150 Hz
Response Time	<3 ms
Supported Interfaces	I2C and SPI
Supported Sensor Types	Glass and Plastic
Package	LGA84

Table 32: STMT07 Touchscreen Controller Specifications

The team chose to go with capacitive touch technology in particular because it offers a better user experience compared to resistive with it's multi-touch support, higher response time and sensitivity. The team opted for the STMT07 for it's overall high performance, power efficiency and compatibility with the VESP's LCD screen size and capacitive touch panel. The STMT07 is also flexible in that it can interface with an FPGA either via I2C or SPI, both of which are available on the ODROID-XU3.

4.4.3 Display Function

The following section will explain in detail how the components described in the previous section will be connected in order to correctly display on all four LCD screens independently. As mentioned previously in this paper, the FPGA of the VESP will transmit video data through a DisplayPort interface to the LCD Controller, and in turn the LCD Controller will split this data to the four LCD screens. Table 33 below shows the pinout for the DisplayPort connector that the LCD Controller will use to get video data from the FPGA.

Pin #	Signal
1	ML_Lane0 +
2	GND
3	ML_Lane0 -
4	ML_Lane1 +
5	GND
6	ML_Lane1 -
7	ML_Lane2 +
8	GND
9	ML_Lane2 -
10	ML_Lane3 +
11	GND
12	ML_Lane3 -
13	Config1
14	Config2
15	AUX CH +
16	GND
17	AUX CH -
18	Hot Plug
19	Return
20	DP_PWR

Table 33: Pinout for DisplayPort.

The DisplayPort interface is able to transmit various types of data through one, two, or four of its Main Link data pairs (ie: ML_Lane0 +/-), each with a raw bit rate of up to 8.1 Gbps. The TLK10022 is able to de-aggregate one high-speed data pair signal into four lower-speed data pair signals, this limitation requires the team to only use one of the Main Link data pairs from the DisplayPort interface for video data transmission. Thankfully, this is an acceptable restriction since each of the Main Link data pairs has a relatively high throughput that can support resolutions of 1080p running at 60 Hz, plenty for the purpose of the project.

Figure 25 below shows the schematic for the display function part of the LCD Controller. All of the inputs and outputs coming to and from the TLK10022 are to be AC coupled by 0.1 uF capacitors, as by the manufacturer's instructions. From the figure you'll see that only two data pair lanes from the DisplayPort connector are being used, with all the others set to ground or used for powering the connection. The first pair will be used to transmit the actual data to the high-speed receiver ports of the TLK10022. The other pair will be used as the reference clock signal for the TLK10022. The TLK10022 will output the four de-aggregated video data pairs, as well as a clock signal pair. Each video data pair is sent to its respective STDP4020 IC, along with the clock signal pair from the TLK10022. Each STDP4020 then converts the incoming DisplayPort signal to RGB format and sends the data to the 40-pin connector corresponding to an LCD panel. Unfortunately, the team was not able to find the pinout information for the STDP4020, so it's abstracted in the schematic out of necessity.

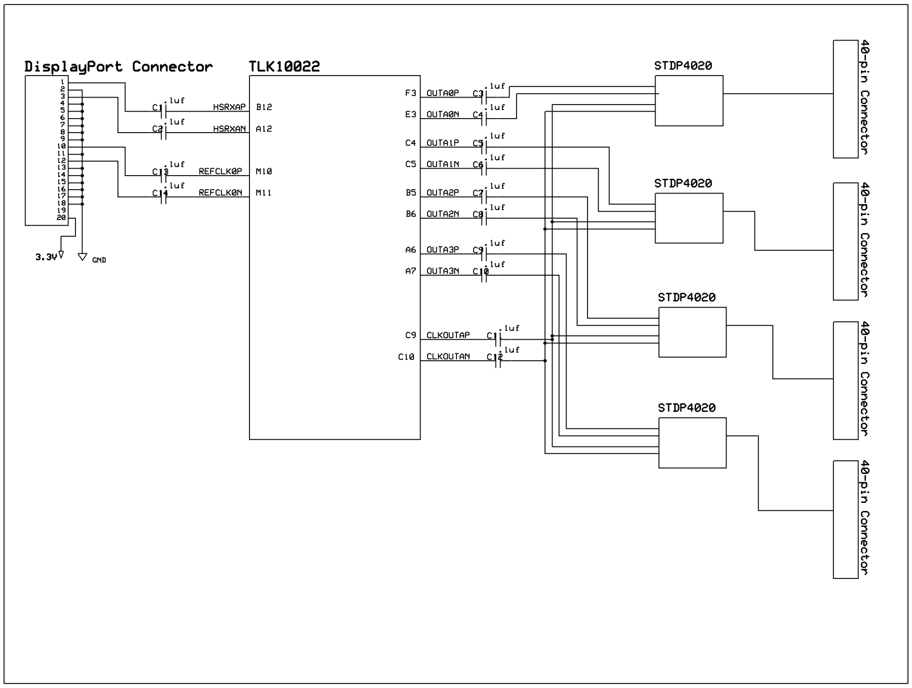


Figure 25: Partial Schematic of the LCD Controller (Display Part).

4.4.4 Capacitive Touch Function

The following section will describe the capacitive touch input handling portion of the LCD controller. As mentioned before, the capacitive touch controller that will be implemented will be the STMT07 by STMicroelectronics. Unfortunately, STMicroelectronics did not produce the full datasheet for the IC when requested, and so the team cannot go into detail on the pin layout of the STMT07 and how it will be connected to the other components. However going by the specifications at the beginning of the section, the team knows that the STMT07 can interface with other devices using either the SPI or I2C interface, which the ODROID-XU3 handle without a problem. As for how the physical touch panel will connect to the LCD controller, a 30-pin FFC connector will be used for this purpose. The 30-pin FFC connector is a common connection for capacitive touch panels, so there is no doubt that the STMT07 will be able to be integrated.

4.5 MCU Subsystem

Figure 26 shows the PCB Launchpad layout and Figure 27 describes the hardware and more importantly the PIN locations and layout.

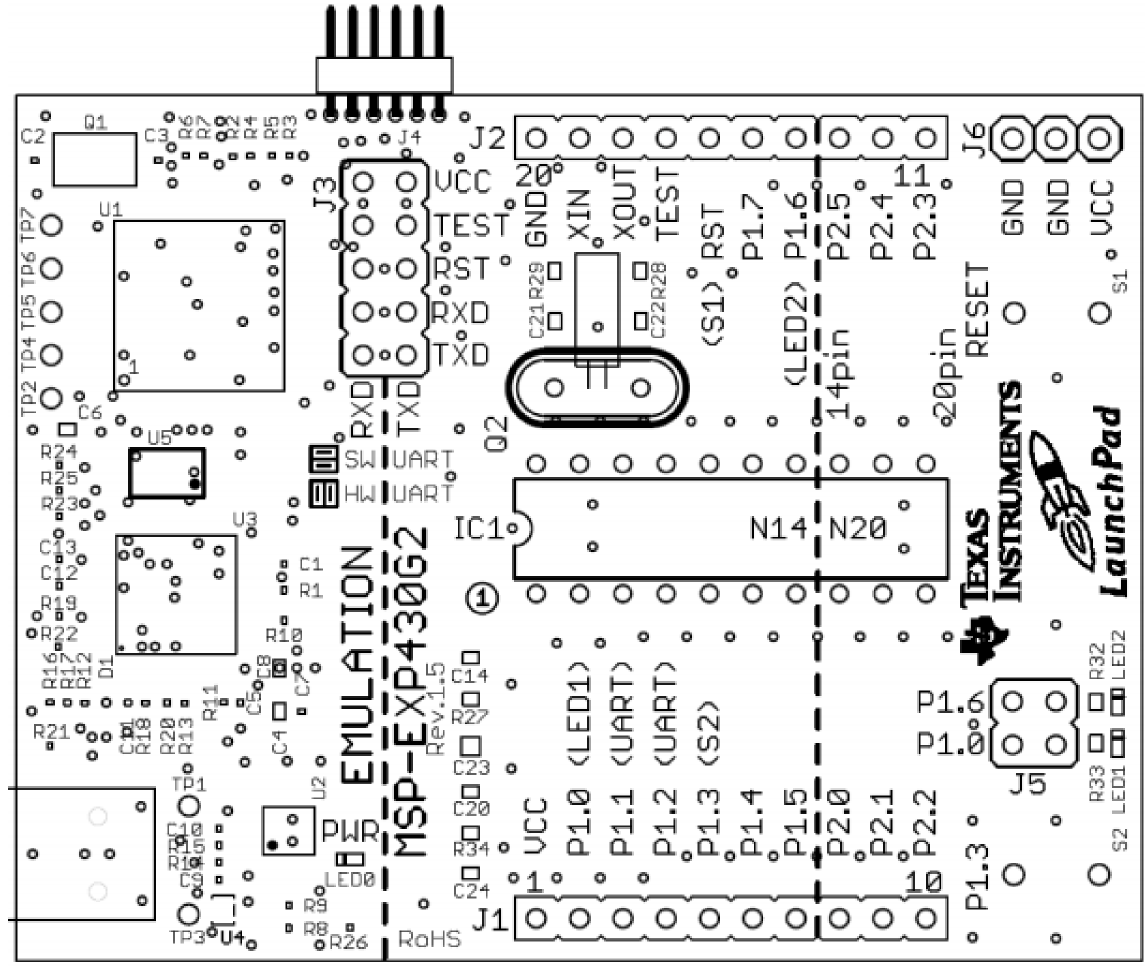


Figure 26: MSP430 PCB layout.
Image use courtesy of Texas Instruments.

Pos.	Ref Name	Number per Board	Description
1	C2, C3	2	16pF 0402 (33 pF on Rev 1.3)
2	C9, C10	2	22pF 0402
3	C1	1	10nF 0402
4	C5, C7, C11, C12, C13	5	100nF 0402
5	C4, C6, C8	3	1µF, 6.3V 0604
6	D1	1	1N4148 MicroMELF
7	EZ_USB	1	Mini-USB connector
8	Q1	1	SMD oscillator 12 MHz
9	R1, R2, R3, R16, R17	3	47k 0402 (R16, R17 is not populated)
10	R8	1	61k5 0402 (6k8 in Rev 1.3 and prior)
11	R19, R22	2	3k3 0402
12	R9	1	30k 0402 (3k3 in Rev 1.3 and prior)
13	R12, R21	2	33k 0402
14	R4, R5, R6, R7, R23	5	100R 0402
15	R14, R15	2	33R 0402
16	R18, R20	2	100k 0402
17	R13, R24, R25	3	1k5 0402
18	R10	1	10k 0402
19	R11	1	15k 0402
20	U1	1	MSP430F1612IPMR
21	U4	1	TPD2E001DRLR
22	U3	1	TUSB3410VF
23	U2	1	TPS77301DGKR
24	U5	1	I2C EEPROM 128k (AT24C128-10TU-2.7)
25	TP1, TP2, TP3, TP4, TP5, TP6, TP7		
26	C14	1	1nF, SMD 0603
27	C21, C22		12.5pF, SMD 0603 (not populated)
28	C23	1	10µF, 10 V, SMD 0805
29	C20, C24	1	100nF, SMD 0603 (C24 is not populated)
30	LED0, LED1	2	Green DIODE 0603
31	LED2	1	Red DIODE 0603
32	R34, R27	1	47k SMD 0603 (R34 is not populated)
33	R32, R26	2	270R SMD 0603
34	R33	1	470R SMD 0603
35	R28, R29	2	0R SMD 0603
36	IC1	1	DIP20 socket
37	Q2		Clock crystal 32kHz (Micro Crystal MS3V-T1R 32.768kHz CL:12.5pF ±20ppm included)
38	J1, J2,	2	10-pin header, TH, 2.54mm male (female header included)
39	J3	1	2X05 pin header male
40	J4		6 pin header male 1.28mm
41	J5	1	2x02 pin header male
42	J6	2	3-pin header, male, TH
43	S1, S2	2	Push button

Figure 27: MSP430 pin table.
Image use courtesy of Texas Instruments

Figure 28 show the block diagram of the data flow to and from the microcontroller board and how it integrates with the project. The sensor outputs are sent to the board and it determines if a user is present, if the brightness of the LCD screens needs adjustments, or if the orientation of the VESP has changed and then sends that data to the FPGA.

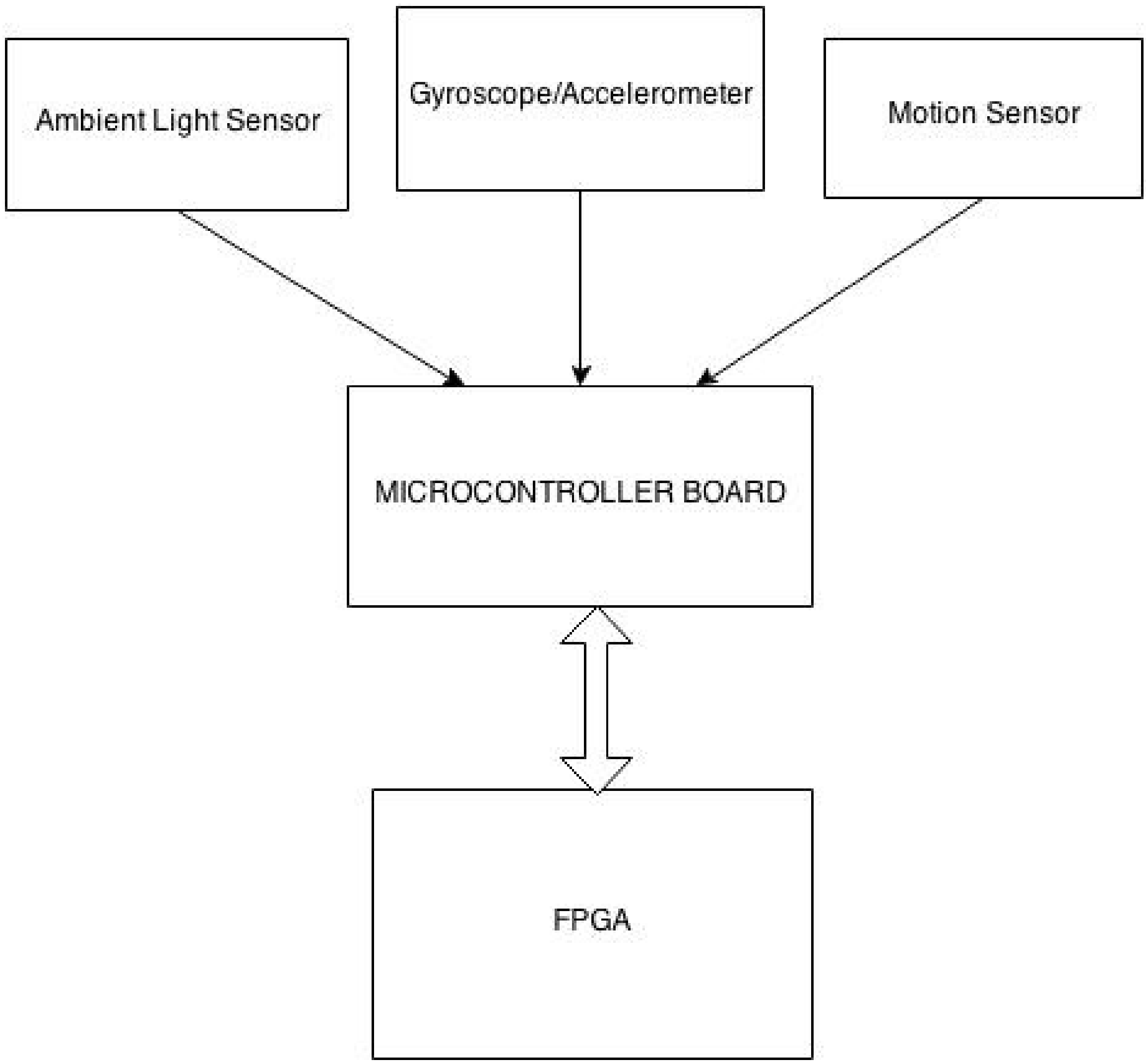


Figure 28: Microcontroller block diagram.

4.6 Sensor Subsystem

The sensors section will refer to the three different types of sensors: the motion sensor, the ambient light sensor, and the gyroscope. Figure 29 shows how the data flow of all three sensors are integrated into the project. The ambient light sensors data will both help determine if a user is present and also determine the LCD screen brightness adjustment, the motion sensor will detect when a user is present, the gyroscope will determine the orientation of the VESP while it is off the dock.

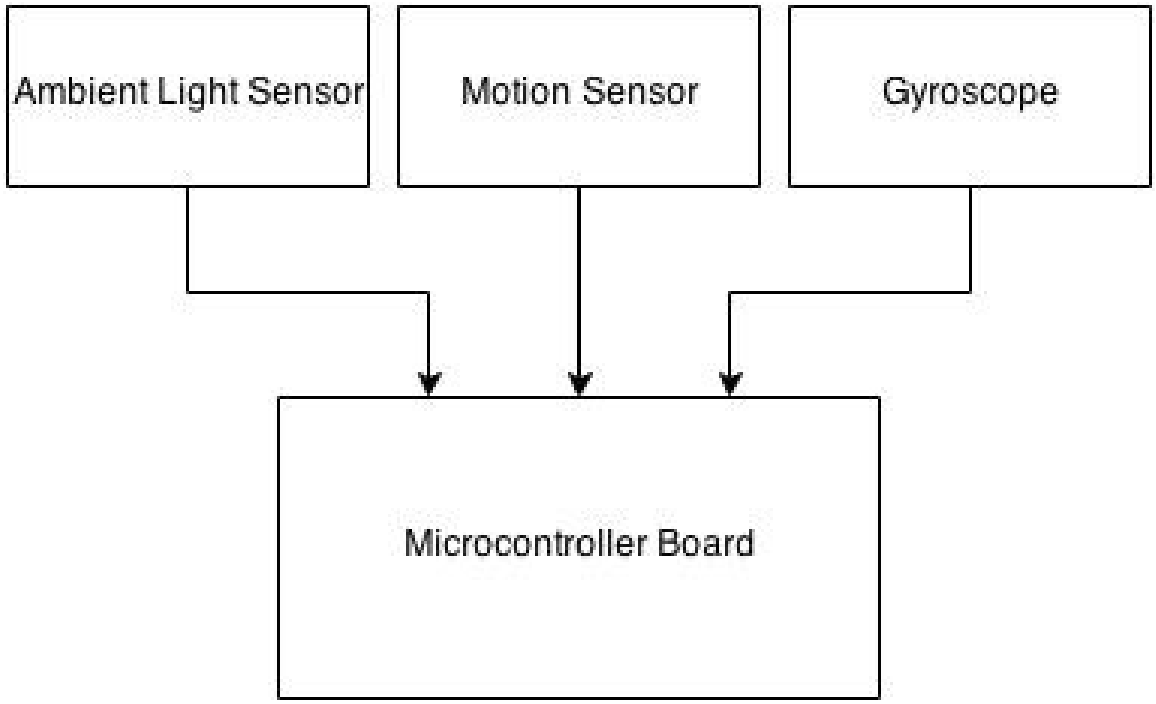


Figure 29: Sensor subsystem block diagram.

4.6.1 Gyroscope

The gyroscope sensor is used to determine the orientation of the VESP when it is off the charging dock. This sensor only needs to be active while off the dock, and when it is being used, it determines the orientation of the VESP and determines whether the LCD screen images should be landscape or portrait, and which way is "up" for the screens so they do not produce upside-down images (relative to the user). It will also be used to determine which LCD screen is facing the user, so if a power-saving mode is activated, it can turn off the away facing LCD screens. The data gathered by the gyroscope will be processed by the microcontroller board and if a new action needs to be taken (for example, the device needs to switch its display from portrait to landscape), the microcontroller will then communicate with the FPGA to take action. This data flow can be seen in Figure 30.

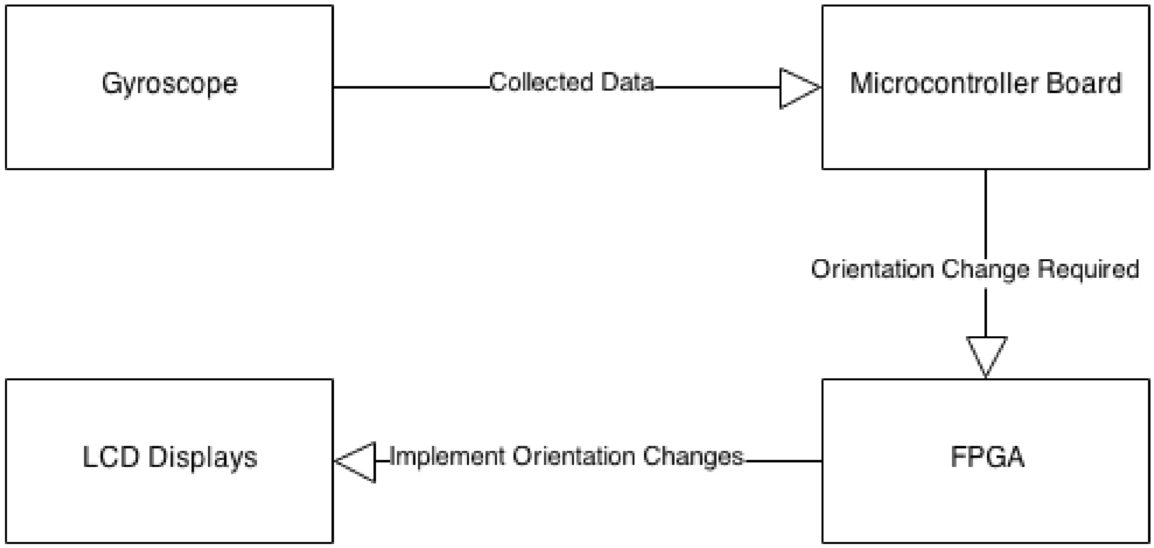


Figure 30: Data flow for the gyroscope.

Additionally, this sensor has a digital accelerometer. Although there is no current application of an accelerometer in the project design, it leaves a lot of room for expansion and improvement of the project. Both the accelerometer and the gyroscope interface through an I2C bus (Figure 31).

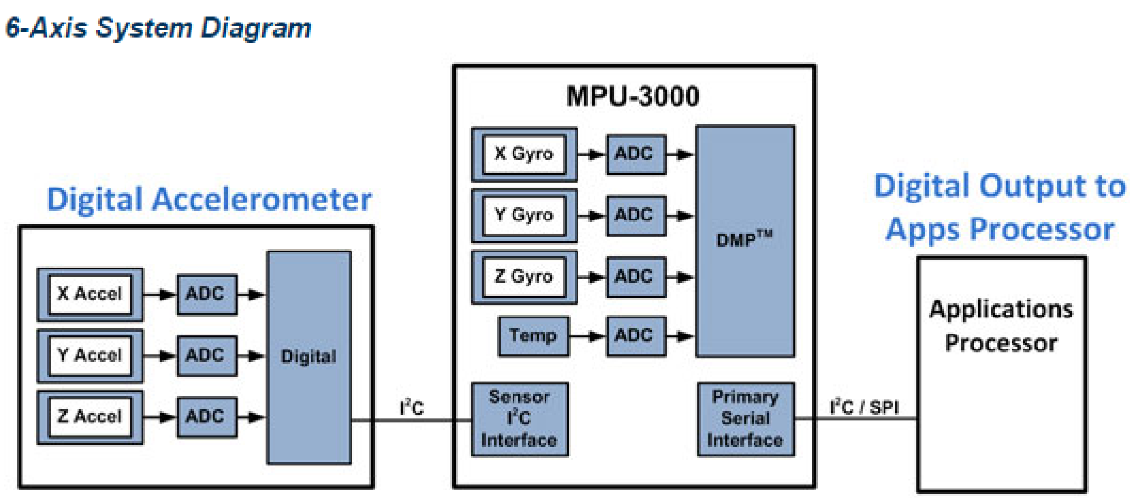


Figure 31: MPU-30X0 block diagram.
Image use courtesy of InvenSense.

Figure 32 will be used to reference the pin layout of the sensor as well as their description.

Pin Number	MPU-3000	MPU-3050	Pin Name	Pin Description
1	Y	Y	CLKIN	External reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	Interface to a 3rd party accelerometer, SDA pin. Logic levels are set to be either VDD or VLOGIC. See Section 6 for more details.
7	Y	Y	AUX_CL	Interface to a 3rd party accelerometer, SCL pin. Logic levels are set to be either VDD or VLOGIC. See Section 6 for more details.
8	Y		/CS	SPI chip select (0=SPI mode, 1= I^{2}C mode)
8		Y	VLOGIC	Digital I/O supply voltage. VLOGIC must be ≤ VDD at all times.
9	Y		AD0 / SDO	I^{2}C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I^{2}C Slave Address LSB
10	Y	Y	REGOUT	Regulator filter capacitor connection
11	Y	Y	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	Y	Y	INT	Interrupt digital output (totem pole or open-drain)
13	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19	Y	Y	RESV	Reserved. Do not connect.
20	Y	Y	CPOUT	Charge pump capacitor connection
21	Y	Y	RESV	Reserved. Do not connect.
22	Y	Y	CLKOUT	1MHz clock output for third-party accelerometer synchronization
23	Y		SCL / SCLK	I^{2}C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I^{2}C serial clock
24	Y		SDA / SDI	I^{2}C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I^{2}C serial data
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.

Figure 32: Pin layout for the MPU-30X0.
Image use courtesy of InvenSense.

Figure 33 shows typical schematics for this sensor to integrate into a system. The project group will use this figure as a reference when designing the PCB that connects all the components of the VESP.

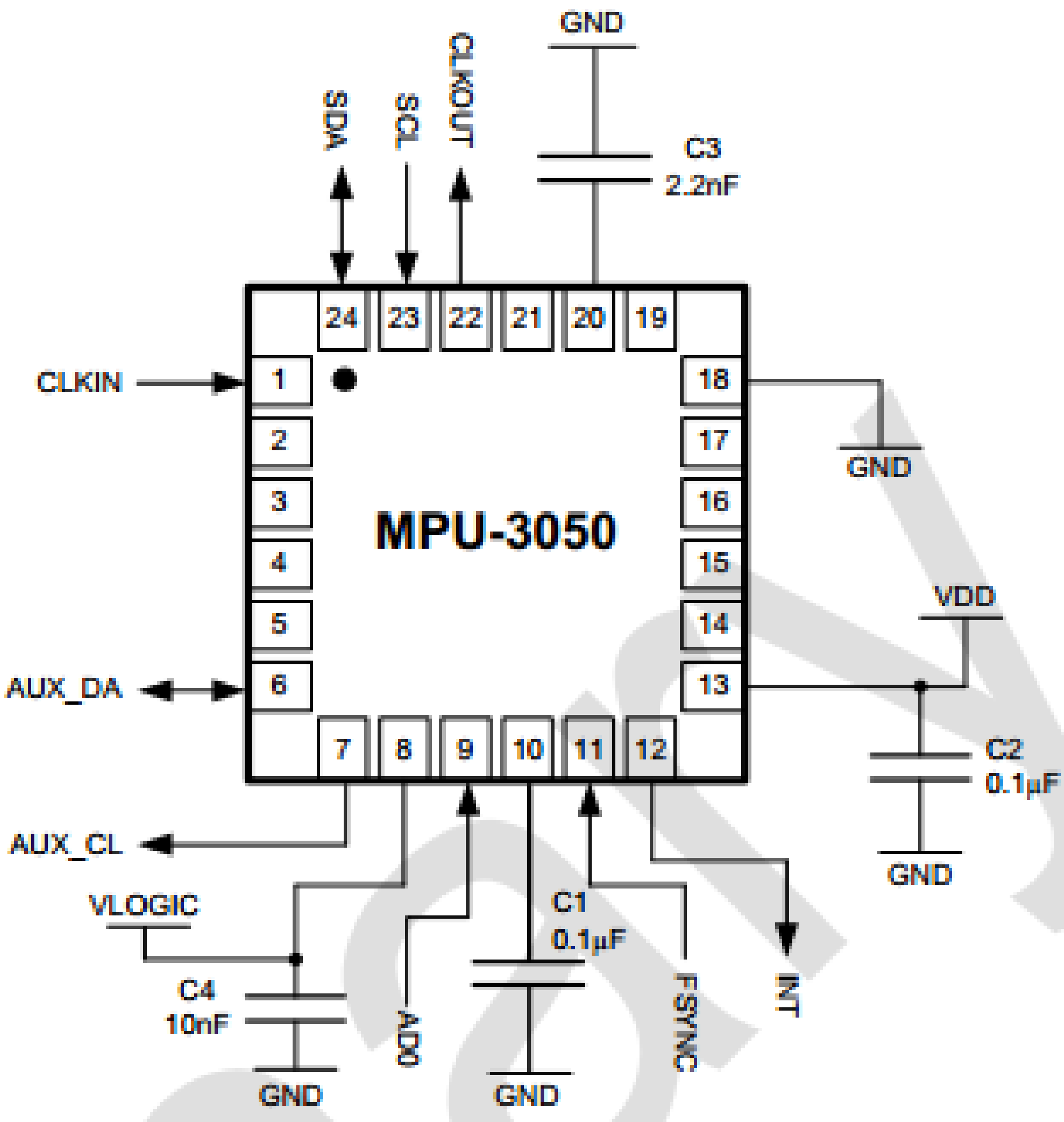


Figure 33: Gyroscope schematic diagram.
Image use courtesy of InvenSense.

4.6.2 Ambient Light Sensor

This sensor will be used for two functions: first, it will sense a sudden change in the lighting to help determine if a user has entered the room the VESP is in (for example, a user enters and turns on a light switch, with the drastic change in lighting alerting the sensor). Secondly, it will be used to determine the LCD screen brightness adjustment necessary for the screens to still be visible to the user, but also save on power (most important when off the dock and being powered by the battery). Figure 34is the block diagram describing how this component functions as well as the interfacing and output methods.

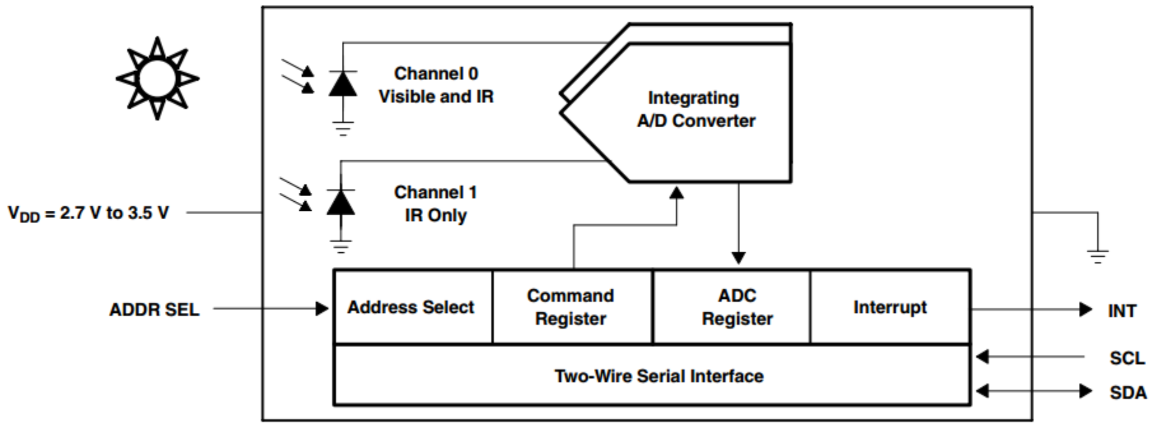


Figure 34: Ambient light sensor block diagram.
Image use courtesy of Adafruit.

Figure 35 shows the bus pull up resistors and decoupling capacitor that must be applied when integrating this sensor to the VESP. The project group will use this as a reference when designing the PCB that combines all the components of the VESP.

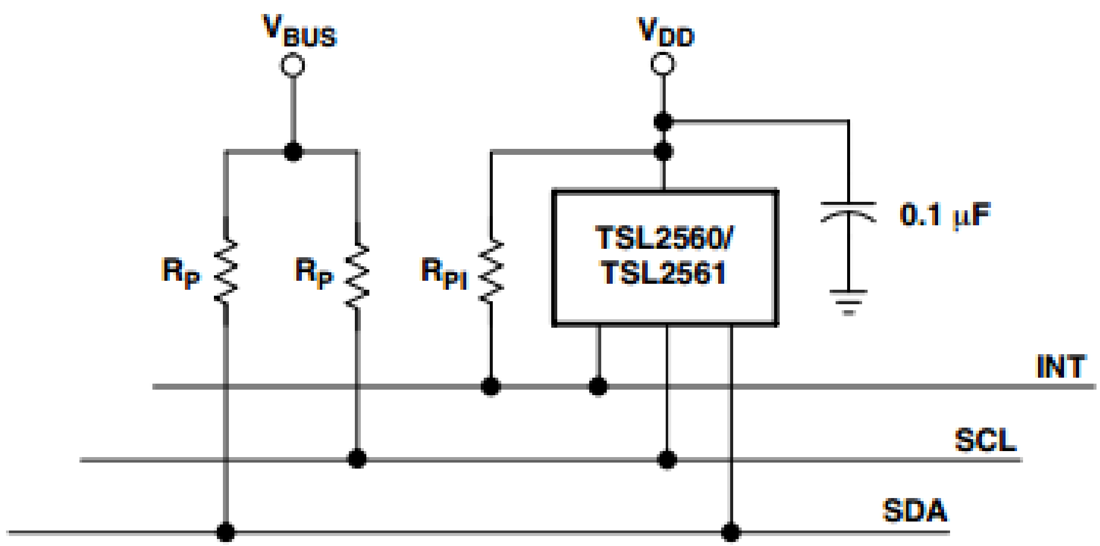


Figure 35: Ambient light circuit configuration.
Image use courtesy of Adafruit.

4.6.3 Motion Sensor

The motion sensor will be the primary sensor for determining if a user is present. It does so by sensing motion from up to 5 meters away in a cone-shaped manner (as seen in Figure 36). A maximum of 4 may be used to approach a 360 degree sensing area (Figure 37).

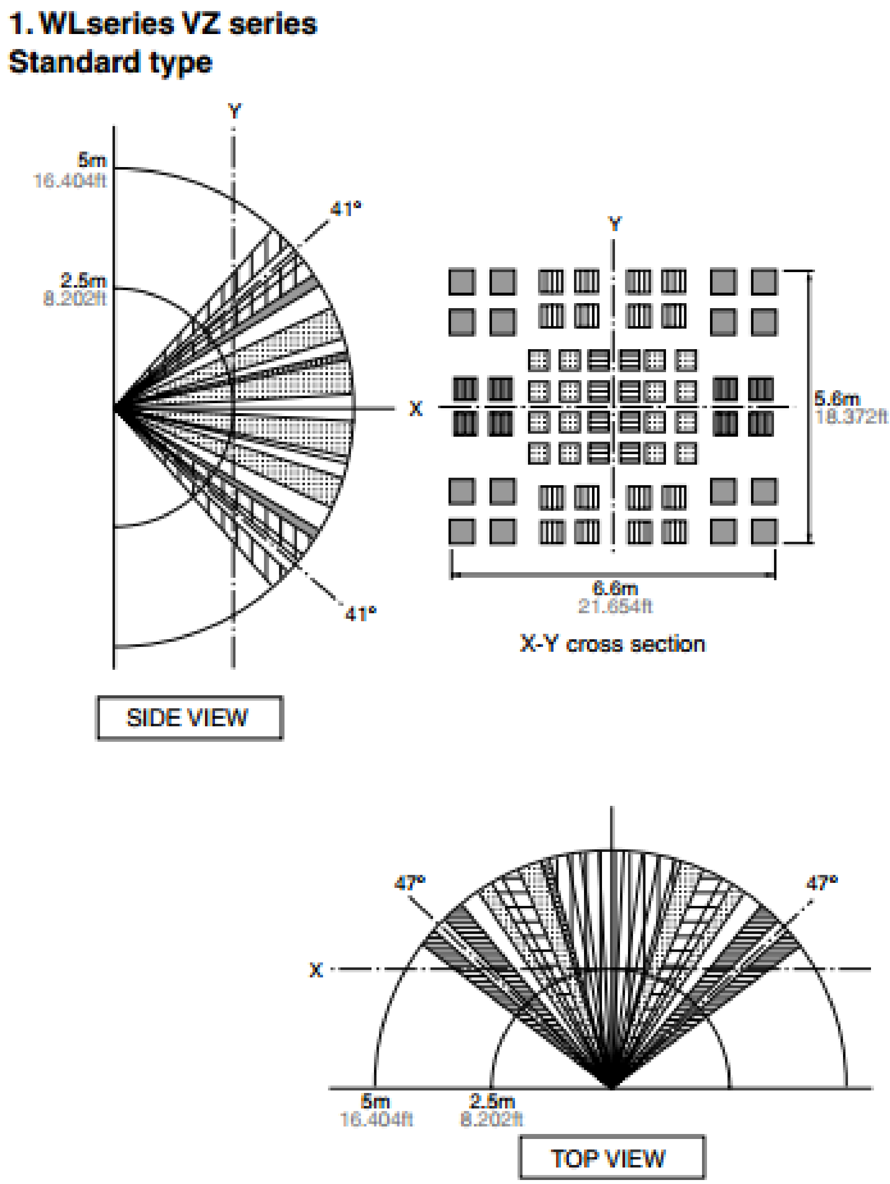


Figure 36: IR motion sensor area of detection.
Image user courtesy of Panasonic.

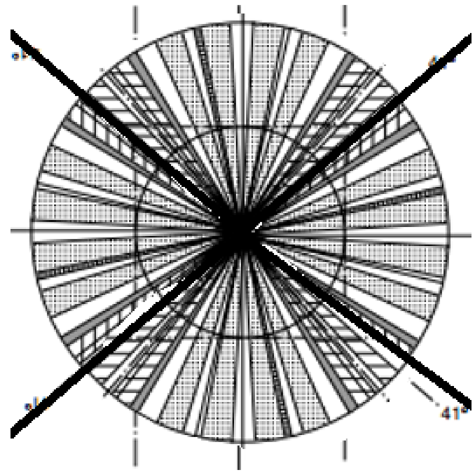


Figure 37: IR motion sensor area of detection if four IR sensors were implemented.

This sensor has a three pin connection (Figure 38), one of which produces a digital output that will be used to integrate with the microcontroller. This connection should only take up a single pin on the microcontroller board or perhaps be connected to the comparator as suggested by the next figure. Figure 39 shows the schematics of how to apply the sensor to a load (the microcontroller board) which will be referenced when designing the PCB for the VESP.

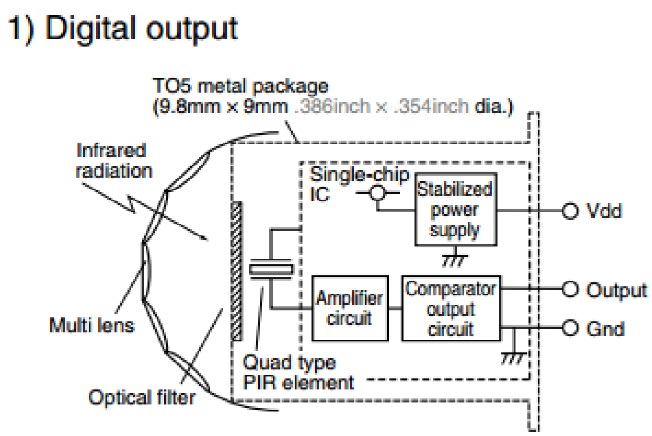


Figure 38: IR motion sensor diagram.
Image use courtesy of Panasonic.

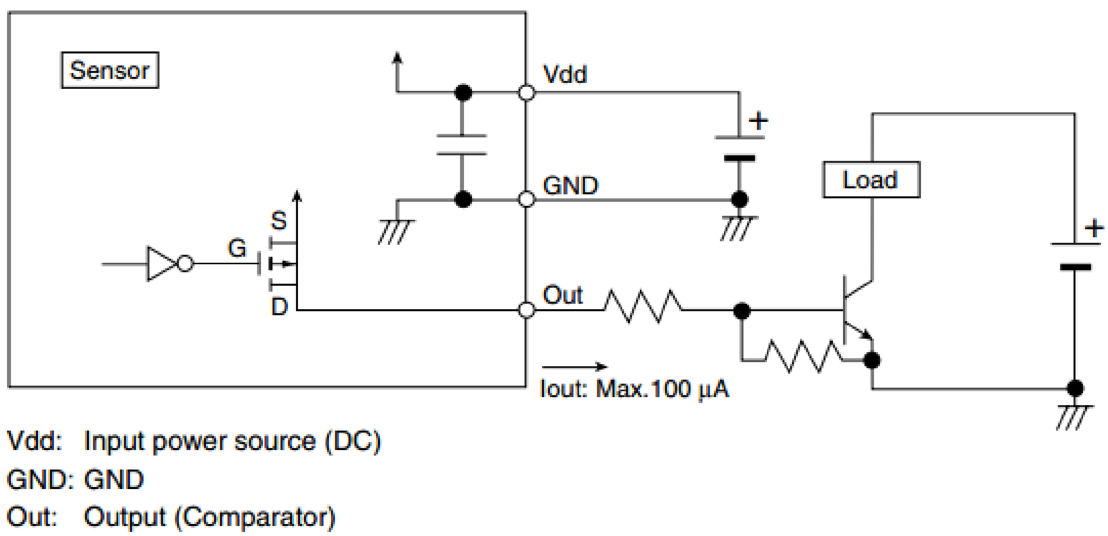


Figure 39: Schematic for the IR motion sensor.
Image use courtesy of Panasonic.

4.7 Control Software Subsystem

4.7.1 Overview

One of the goals that the team set for the VESP was for the device to be enjoyable to use, and the team will achieve this goal through well-designed, useful, and visually stunning software. The team will be using Ubuntu, along with Qt and OpenGL APIs to design the software. The VESP will implement both a lock screen, which will simply display useful data, and a home screen, which will be used by the user to interact with and run applications from. Figure 40 below shows the state diagram of how the VESP will switch to and from its home and lock screens, and applications. From the figure, the VESP will turn on to the lock screen and will wait for the user's touch input. Once the user does a simple gesture (say, swipe up or down), the device will "unlock" and go to the home screen. From the home screen, the user is able to do a variety of things including: launch applications, access and change settings, and access status information (such as battery life remaining, Wi-Fi network information, Bluetooth device connected, etc.). When an application is launched, it will take the place of the home screen and, depending on the application, may display on more than one LCD screen. From any given application, the user has the option of going back to the home screen and starting over. The device will go back to the lock screen after sufficient time has passed with no activity from the user, the only exception being if the device is currently playing music. The home screen will also include a touch button that will allow for immediate transition to the lock screen.

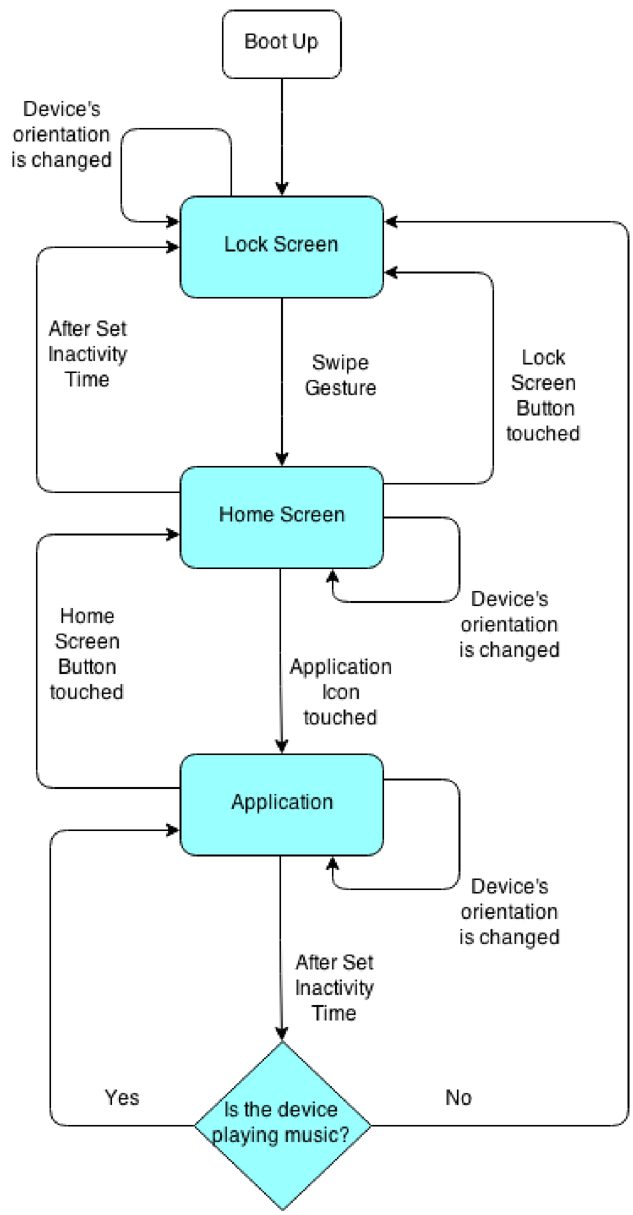


Figure 40: VESP State Diagram

4.7.2 User Interface Layout

Arguably, the most important part of any user interface is the design of its layout. A user interface that has a bad design layout can be very annoying to interact with and discourages users from using a given device. The team aims to design a good layout for the UI of the VESP to effectively display important data, buttons and visuals in order to create an enjoyable user experience.

Lock Screen

The lock screen will be of a relatively simple design. After all, all it will function as is as a standby screen of sorts and won't have any interactive elements (ie: buttons). The lock screen itself will only display on the touch-enabled LCD panel of the VESP, while all of the other screens will display notifications and passive application data. Figure 41 below describes the vertical layout of the lock screen, along with an example of how the other LCD screens might display application data, the following figure, Figure 42 shows the horizontal layout example. A large sized clock (in the figure it is an analog clock, but this can be changed via a setting) will display the time clearly and elegantly. Underneath the clock will be the current battery level for the user to take into consideration. Under the battery display, the VESP will display an image of the user, along with a greeting for aesthetic purposes. The image of the user to be displayed is contingent on the Bluetooth paired Android device, if there is one available. If there is no paired Bluetooth device, a default anonymous image will be used. The Qt Bluetooth API will be used to retrieve the user's image from the paired Android device.

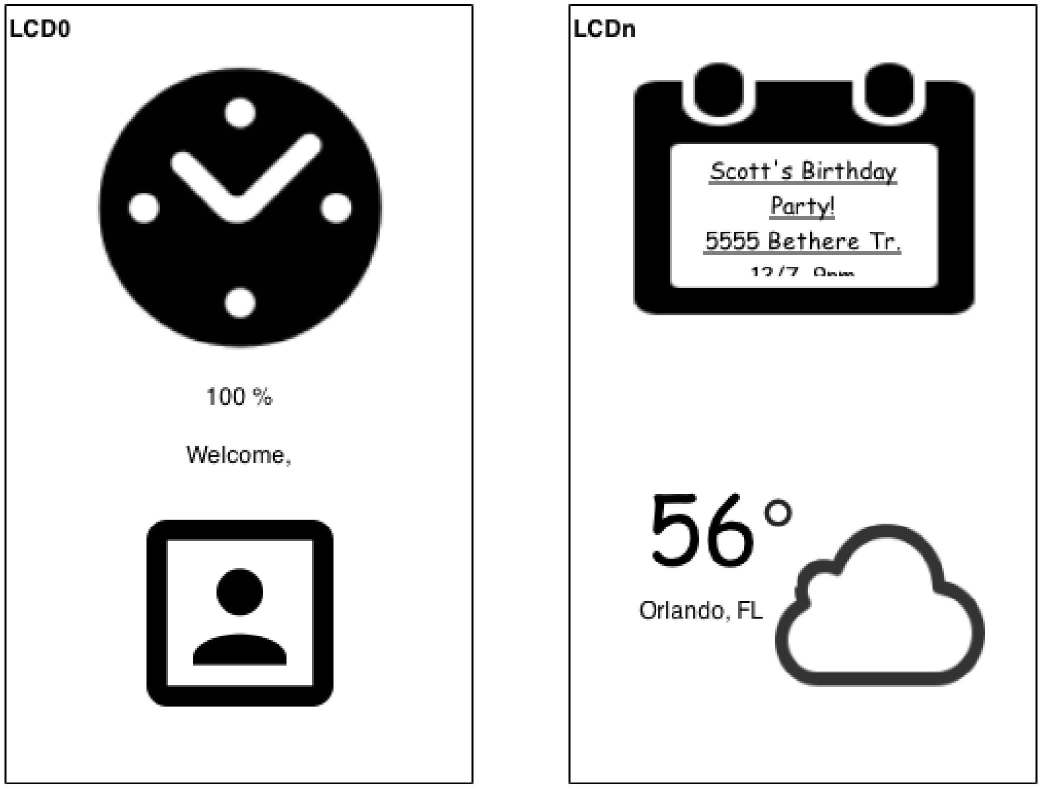


Figure 41: (left) VESP Lock Screen UI Vertical Layout. (right) Example of potential calendar event notification (from the calendar app) and current weather forecast (from the weather app).

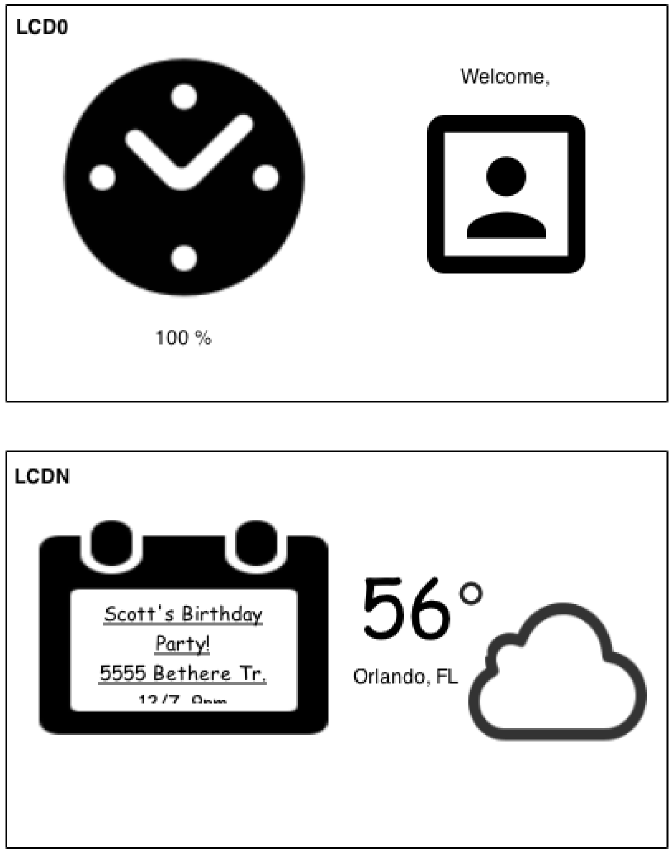


Figure 42: (top) VESP Lock Screen UI Horizontal Layout. (bottom) Example of potential calendar event notification (from the calendar app) and current weather forecast (from the weather app).

Home Screen

The home screen is considerably different from the lock screen in that it contains actual buttons and interactive elements. Again, the actual home screen will only display on the touch-enabled LCD panel of the VESP. However, unlike the lock screen, the other LCD panels will be turned off while the user interacts with the home screen. Figure 43 below describes the vertical layout for the home screen, with all interactive elements highlighted in orange to differentiate them from the rest of the layout. The following figure, Figure 44 shows the horizontal layout of the home screen. On the top of the home screen, the current time can be found in the middle, while the two small buttons on either side are (from left to right) the "User Profile" button and the "Lock" button. The "User Profile" button will launch a device status application (discussed later in this report), while the "Lock" button will intuitively bring back the lock screen and thereby "lock" the device. The bigger cloud shaped buttons represent the icons for the applications of the VESP, when these are touched, the corresponding application will launch. Lastly, the long orange arrow running along the middle of the home screen signifies that the user will be able to do an upwards (or right for horizontal) swipe in order to scroll through more applications.

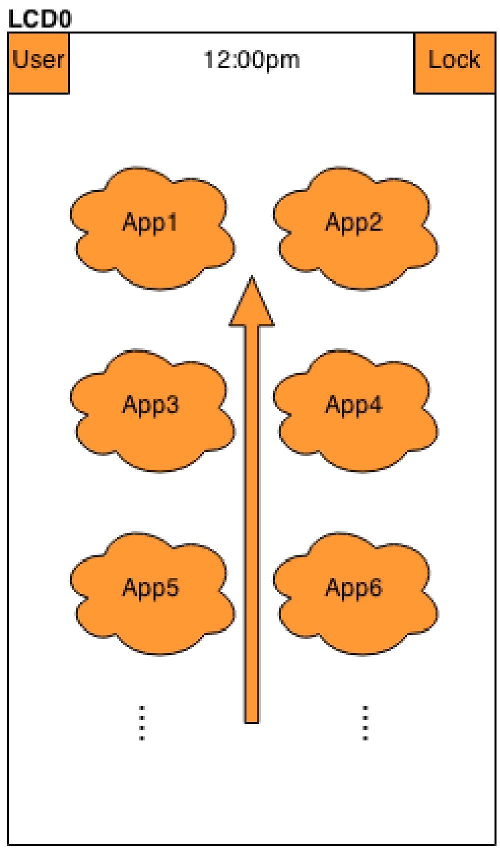


Figure 43: VESP Home Screen UI Vertical Layout

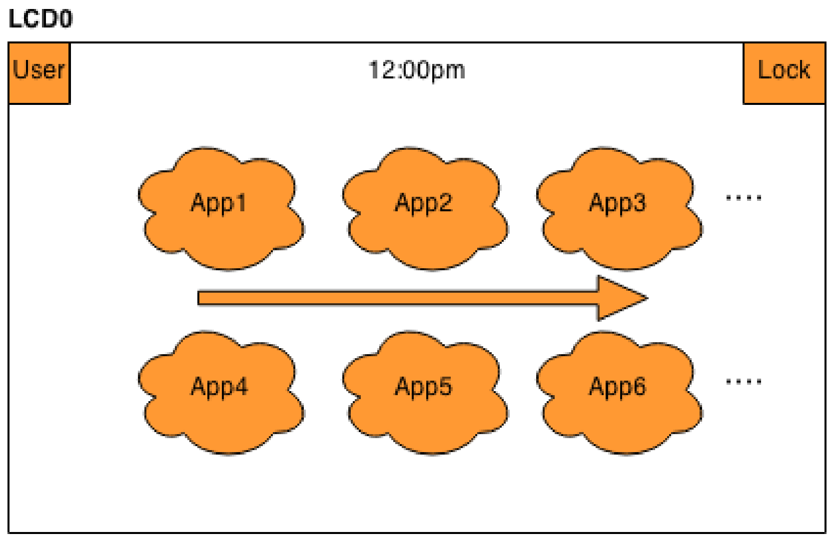


Figure 44: VESP Home Screen UI Horizontal Layout

4.7.3 Applications

The applications available on the VESP are intended to give the VESP a mix of utility and entertainment, with the option of adding more applications in the future. For the purpose of this senior design project, the team opted to design six applications. These six applications will include a music app, a calendar app, a clock app, a weather app, a settings app, and a device status app. Figure 45 below shows a general vertical layout that all of the applications that will follow, Figure 46 following shows the general horizontal layout. Things to notice are that each application will have a "home" button on the top right of the screen that will allow the user to go back to the home screen, and that the application's specific layout will re-arrange accordingly to the orientation of the device. Each application will be described individually in this section.

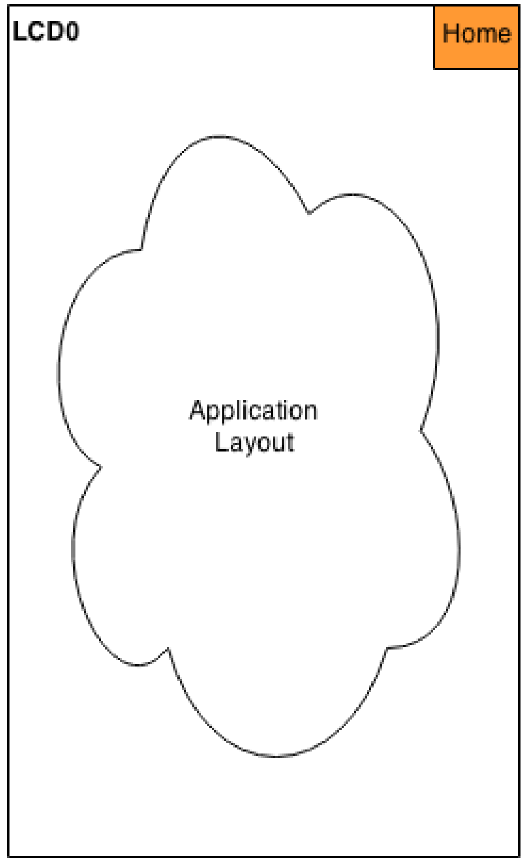


Figure 45: General VESP Application Vertical Layout.

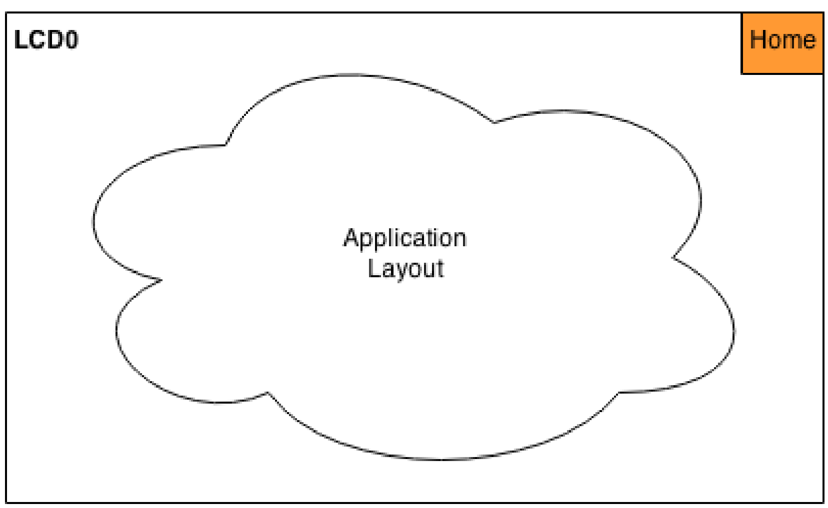


Figure 46: General VESP Application Horizontal Layout.

Music

The music application for the VESP is perhaps the most unique app included in the VESP. Not only will the music be able to stream music from a Bluetooth paired Android device, but the music app will also render a visualizer animation that takes in the music as input. Once the music application is launched, the device will wait for a music stream from the paired Android device. This music stream will be cached and by using a visualization algorithm, a visual representation of the music is created and displayed. The music app will use all of the LCD screens of the VESP to display this visualization. Music playback controls such as volume, play/pause, stop, next, back will be available on the touch-enabled display. After 5 seconds of inactivity while playing music, the playback controls and the home button will disappear from the touch-enabled display to not obstruct the visualizer. To bring back the playback controls and home button, the user need only touch the display. This application will use the Qt Bluetooth API to retrieve music data available on the paired Android device and will rely on OpenGL to create the visuals.

Calendar

The calendar app for the VESP will provide event notifications to the user. Like the music app, all of the applications useful data will be retrieved from a Bluetooth paired Android device using the Qt Bluetooth API. The calendar app will not support entering new events and reminders, as there is no way of entering English characters into the VESP. The calendar will app will utilize three of the four LCD screens, the touch-enabled LCD panel along with its two adjacent panels. On the touch-enabled panel, the current day's schedule of events will be displayed, with the left and right panels showing the previous day and next day, respectively. The calendar app will support swiping left or right to get to the next or previous day, respectively. By touching the day label on the touch-enabled panel, the app will transition to display the current week, with the current day highlighted (if it coincides). In turn, the adjacent LCD panels will display the previous week and the next week, respectively. By touching the week label on the touch-panel, the app will again transition, this time to the current month with the current week highlighted (again, if it coincides). Again, in turn, the adjacent LCD panels will display the previous month and next month, respectively. Touching the month label again will bring the user back to the current day and the adjacent LCD panels will be brought back to the adjacent days as well.

Clock

The clock app for the VESP will display the current time to the user, as well as provide the option to set an alarm for a specified time. The clock app will only utilize the touch-enabled LCD panel. The clock app will not rely on an external device in order to function and will create all of its associated data internally. The user will be able to set a new alarm by using a graphical interface that resembles an analog clock, the user need only touch the desired hour, followed by the desired minute. Once the time is specified, the user can specify AM or PM, weekday and whether to repeat the alarm indefinitely until turned off or deleted. Each new alarm that the user creates will be stored in a list, which the user will be able to edit, turn off/on, or delete any given alarm at their discretion.

Weather

The weather app for the VESP will provide the user current weather information relevant to the their location. Location information will be retrieved using the Qt Positioning API, which utilizes the Wi-Fi in order to calculate the information. The weather app will only utilize the touch-enabled LCD panel. The app will start with displaying the current day's weather information such as temperature highs and lows, humidity level, and expected conditions (rain, fog, sun, etc.) for various parts of the day (morning, mid-day, evening, etc.). By touching the current day's label, the app will transition to display the week's expected forecast, another touch, this time on the current week's label, will bring back the current day's weather information. Weather information for the app will be retrieved from the National Weather Service server. The app will query the XML from the server and use the Qt XML API to parse the data to usable variables.

Settings

The settings app for the VESP will provide the user with various settings pertaining to the function of the VESP device. These settings include, but are not limited to: turn Wi-Fi on/off, turn Bluetooth on/off, lock screen wait time, volume, and whether to display a digital or analog clock on the lock screen. The settings app will only take advantage of the touch-enabled LCD panel. The app's design will be relatively simple, with each setting being represented as part of a list. Most settings will be a binary choice, so a simple touch on the setting will toggle it. Non-binary settings will have increment or decrement buttons to change them.

Device Status

The device status app for the VESP will provide the user with various information pertaining to the operating conditions of the VESP. This information includes, but is not limited to: battery life remaining, whether Wi-Fi/Bluetooth are on, which Wi-Fi network the VESP is currently connected to, and which Android device the VESP is currently paired with. The device status app will only utilize the touch-enabled LCD panel. The device status app won't provide any other function other than display useful information pertaining to the VESP's operating conditions and the only touch functionality it will have is the "home" button that all apps share.

4.7.4 Data Flow

The VESP will be constantly handling lots of background data during normal operation. In order to minimize lag during normal operation, the team must implement an efficient data flow structure. Parallelism plays a vital role in this, and so the team will focus on making as many processes run in parallel as possible. Figure 47 below shows a sequence diagram detailing the flow of data going on within the VESP device. From the figure, the FPGA will load the GUI elements and send the data to be displayed to the display portion of the LCD Controller (via DisplayPort). The touch portion of the LCD Controller will always be waiting for touch input from the user and once it's received, it will be sent to the FPGA to be interpreted. Similarly, the MCU will be constantly checking its sensors for any changes on conditions. When there is a change in the ambient lighting, the MCU will send this data to the LCD Controller to adjust LCD brightness. When there is a change in orientation (includes acceleration) or IR signal, the MCU will send this data to the FPGA. Proximity data from the MCU will be used to determine whether to turn on or off the LCD Controller and the ambient light, gyroscope, and accelerometer sensors on the MCU. The LCD Controller will be told to update its displays when there is either new touch input or if there is a significant change in the VESP's orientation. The touch, orientation, and proximity data will be handled asynchronously by the FPGA in order to achieve a sufficient level of parallelism.

4.8 Docking Station Subsystem

4.8.1 Power Supply

The power supply will be using 120 VAC to 12 VAC transformer. The Primary side will be connected to the main electricity which is the your regular household wall socket and the secondary side will connected to bridge rectifier that will help the change the alternating current into a direct current. Finally to get a linear regulated 5V voltage that will charge the battery, the group will be using high voltage capacitor and a voltage regulator.

4.8.2 Battery Charging

While looking for the best lithium polymer battery that will power the LCDs screens and all its components shown in Table 34, the group needed a high capacity battery. The most convenient batteries were founded at the Sparkfun website. Compare to others websites, the price was right. In Table 35 you will find the top three choices that the group narrowed down to: 1000mAh, 2000mAh, and 6000mAh.

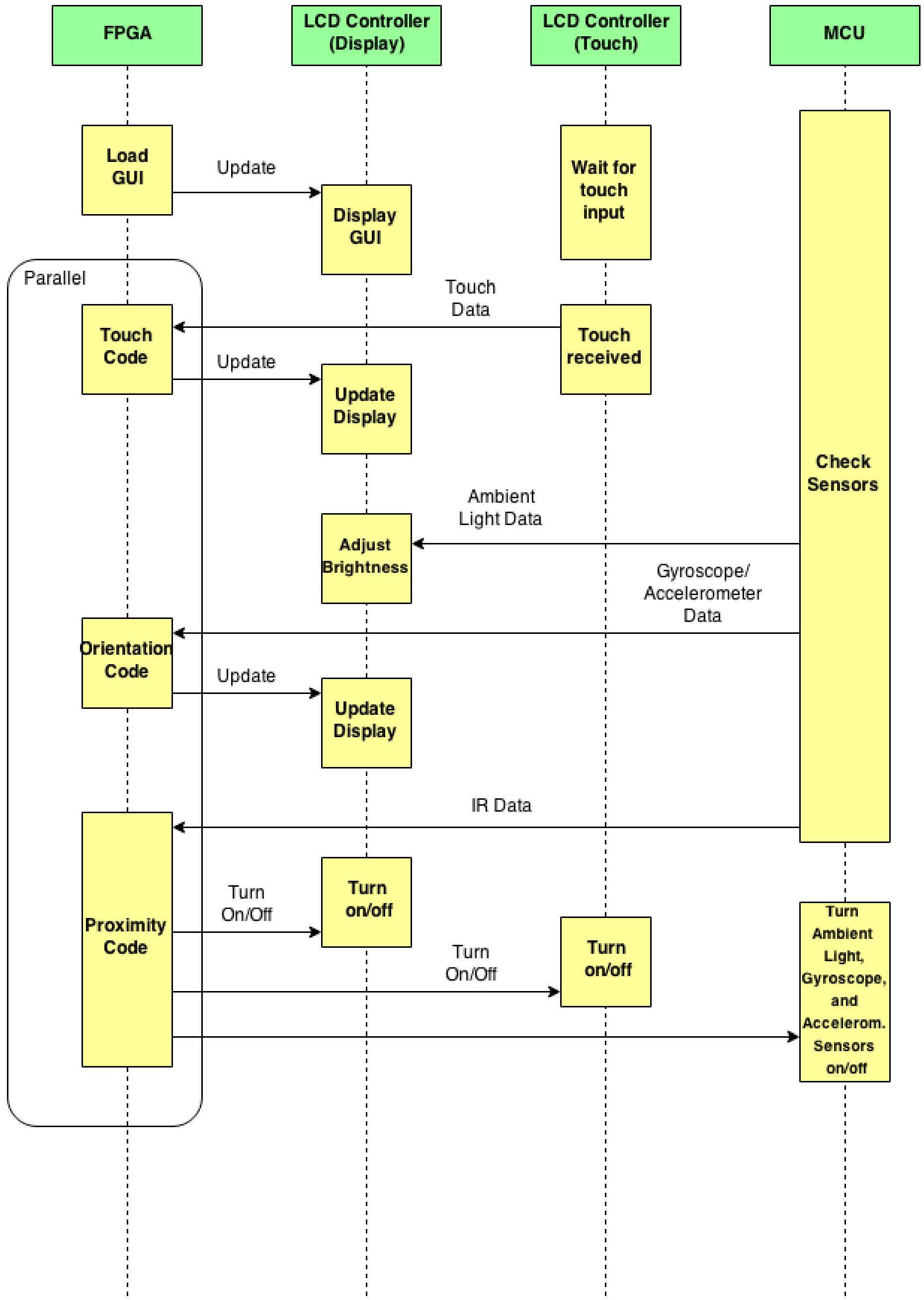


Figure 47: Data Sequence Diagram for the VESP

Component	Power Requirement	Power Supply Line
Wi-Fi CC3100BOOST	2.1V – 3.6V	3.3V
Bluetooth CC2564MODN	2.2V – 4.8V	3.7V
Capacitive Touch Panel	2.8V – 3.3V	3.3V
Microcontroller Board	3.6V	3.3V
Motion Sensor	3.0V – 6.0V	3.7V
Ambient Light Sensor	2.7V – 3.6V	3.3V
Gyroscope	2.1V – 3.6V, max 6.1mA	3.3V
FPGA ODROID-UX3	5.0V, 4A	3.3V
LDC Controllers		
1x TLK10022	1.0V – 1.8V	3.3V
4x TFP401	3.0V – 3.6V	3.3V
1x STMT07	2.8-5V AV_dd , 1.8V DV_dd	3.3V

Table 34: Power requirements for every component using the battery

	1000mAh	2000mAh	6000mAh
Nominal Voltage	3.7V	3.7V	3.7V
Continuous Discharge	2C	2C	2C
Working Temperature	-25 to 60 C	-25 to 60 C	-25 to 60 C
Length	2"	2.4"	2.4"
Width	1.32"	2.1"	2.1"
Height	0.23"	0.25"	0.75"
Weight	22g	36g	110g

Table 35: Battery Specifications

After looking at the specifications of the lithium batteries, the group notice that the voltage and the rate charging was the same. Meaning the only things that were taking in consideration was the capacity and the size of the battery. Due to the low weight of the lithium polymer battery, the group decided to get the 6Ah battery. Despite the size and weight of the battery, the group though it was necessary to have a battery that will last a proper time so the user won't have to recharge the battery all the time.

Lithium Ion Polymer batteries need a specific charger in order to avoid any damage to the batteries. The group found the charger suitable for the 6Ah battery in the same website that the battery was founded. The Figure 48 shows the diagram of the LiPo USB Charger

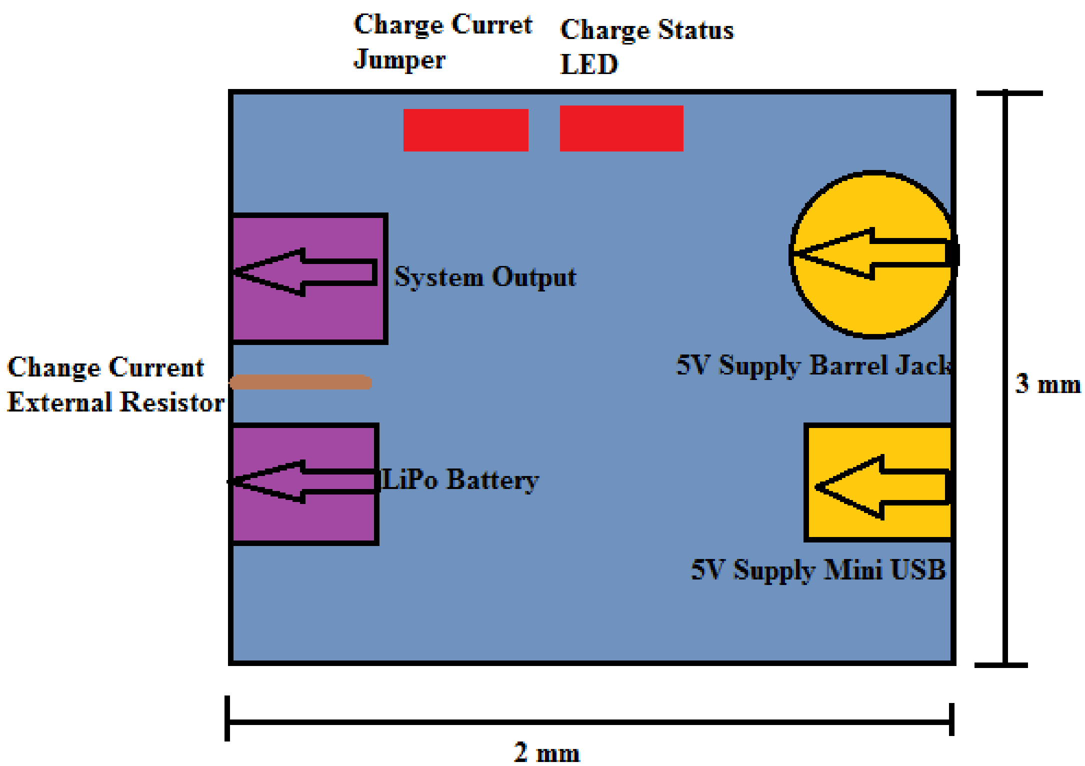


Figure 48: LiPo USB charger Block Diagram

This charger has the ability to charge from a mini-B USB cable and also from a 5V wall wart supply, which makes it kind of convenient for the user isn't at home and would like to charge the cube with just the plug. Also, the default charge current of the board is set to 500mA which is the amount of current the power supply has to give out. The board comes with a battery JST connector for the lithium battery that should have a nominal voltage output of about 3.7V, and 4.2 when the battery is at full charge.

The system output has another JST connector for the FPGA board. That means that a voltage of 3.6V to 4.2V will power the FPGA and then distribute among the other components. The board also has a charge status LED to show when the battery is charging or not. When there is no battery, no power input to the board, or when the charge is complete, the LED is OFF. The only time the LED will be ON is when the battery is charging.

Finally, the board includes an over-current protection which keeps the battery safe and from blowing the whole supply with excessive current. Beside that, It has the ability to configure the delivered current from its default which is 500mA to 100mA. To make this change, the small trace between the pads of the 100mA located in the 'Charge Current Jumper' will have to be cut. Another way to get a specific current is to disconnect both jumpers and add a resistor in the 'External Resistor' located between the system output and battery charger connector.

4.8.3 Audio Speakers

The speakers is one the important source of entertainment that the user will like to see working. In order to get the best audio quality, the group decided to get a high efficient amplifier which is a Class D amplifier. Due to its 90 percent efficient amplifier, the output audio will have a fine tone. besides that, the amplifier will be outputting two 7 watts speakers what will have an outside volume control for the users preference. In order for the speakers to play the music from our portable LCDs screens, the group designed an male audio jack output through the top of the base to connect with the LCDs.

4.9 Housing Schematics

The housing design for VESP is divided into 2 sections: one being the base and the second one for the portable electronics. The base housing will include the power supply and the audio system. The portable electronics housing will be composed of the LCDs screens, Wi-Fi, Bluetooth, Sensors, Microcontroller, and FPGA.

Below are the schematics for the base housing in Figure 49 shows the inside of the box, Figure50shows the front view of the box, and Figure 51 shows the top view of the box when its closed. The total measurements of the outside box should be 6" x 6" x 3". The base will be a square prism made with wood. The back, top, left and right side will completely solid and fixed. The bottom will have a cover plate screwed in each corner. The plate will have a square placement for the four LCDs screens and a charger input with the male audio jack as seen in Figure 51. The front will have two holes of 2 inches of diameter each so that the two speaker can fit on it as seen in Figure 50. Also the speakers will have a cover for protection purposes.

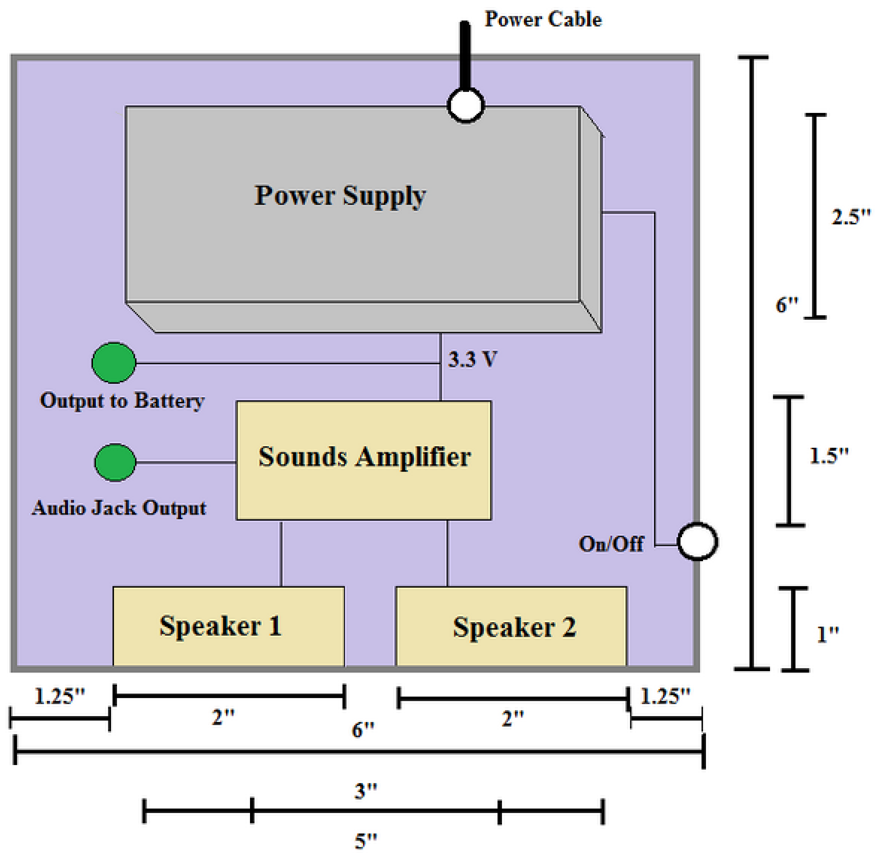


Figure 49: Block Diagram of the Base Housing

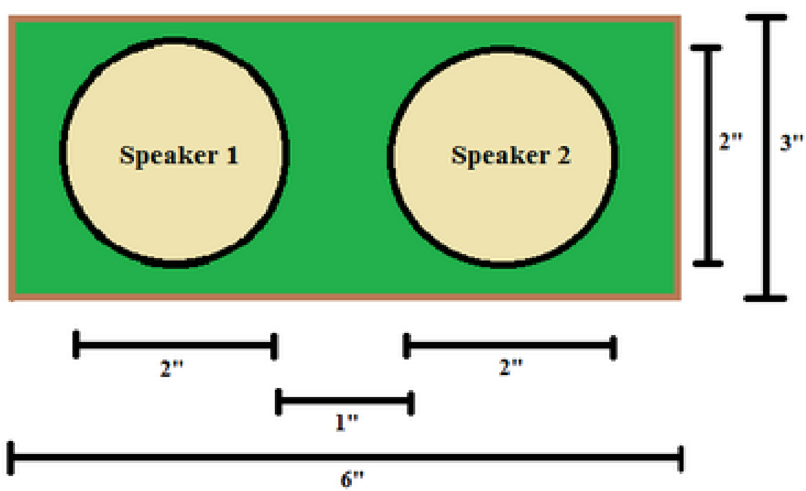


Figure 50: Front side of the Base Housing

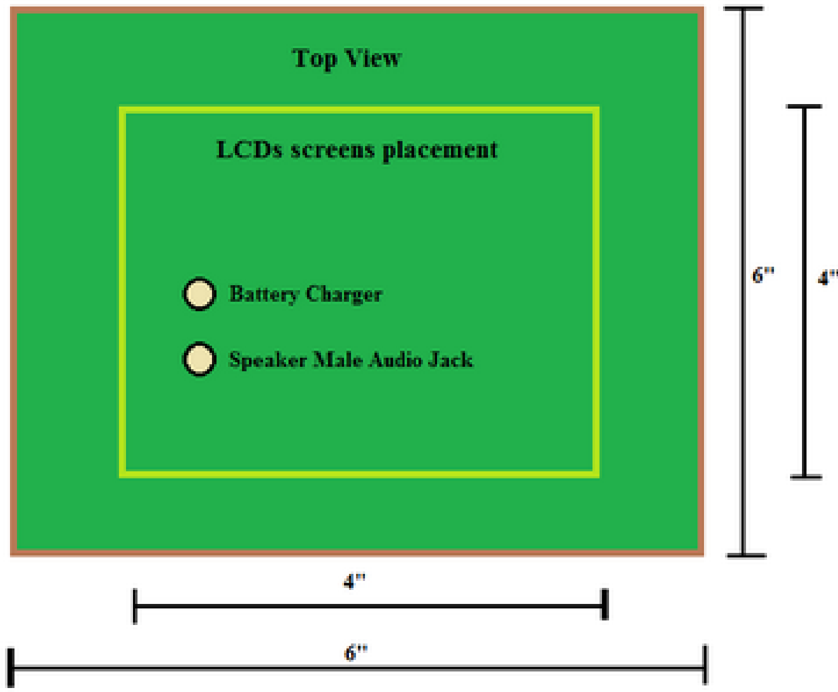


Figure 51: Top side of the Base Housing

The second part of the housing is where most of the components are located. Figure 52 shows the the outside part of the LCDs screen housing. The four LCDs screen will be glue with an adhesive that will keep all the screens together. Besides that, the top and bottom part of the screens will be covered with two plastic plates that will used to support the components inside the housing. In Figure 53, you could see the top and bottom plate of the housing. The top plate will have the motion sensor on it, and the bottom plate will have the battery charger connection with the female jack audio.

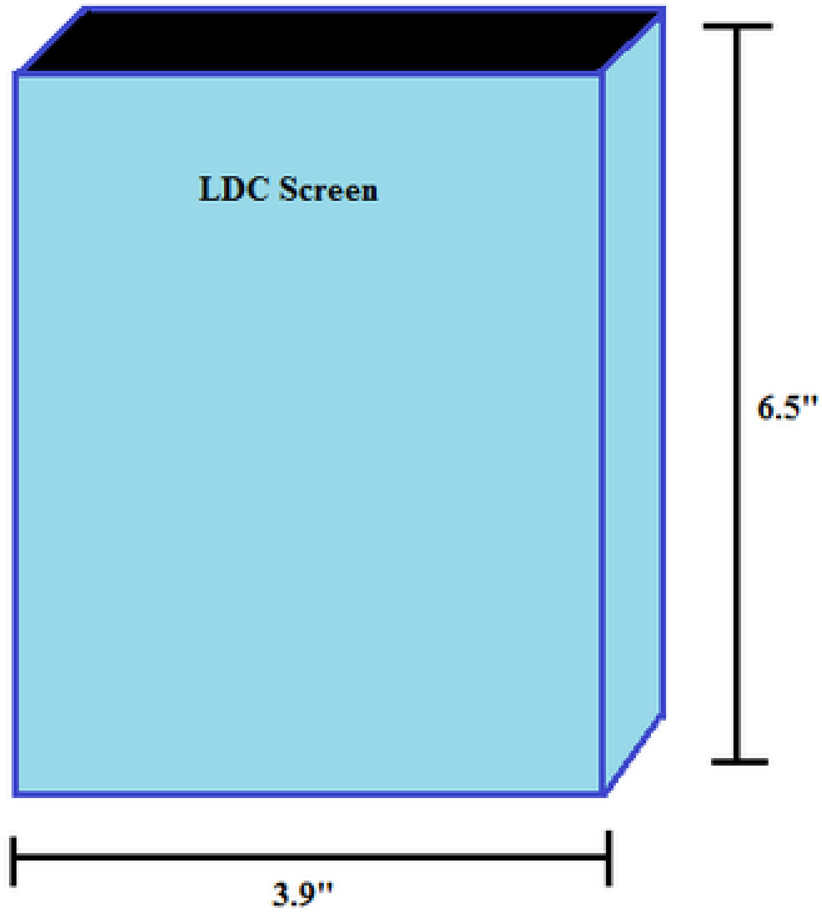


Figure 52: Outside of the LCDs Screen Housing

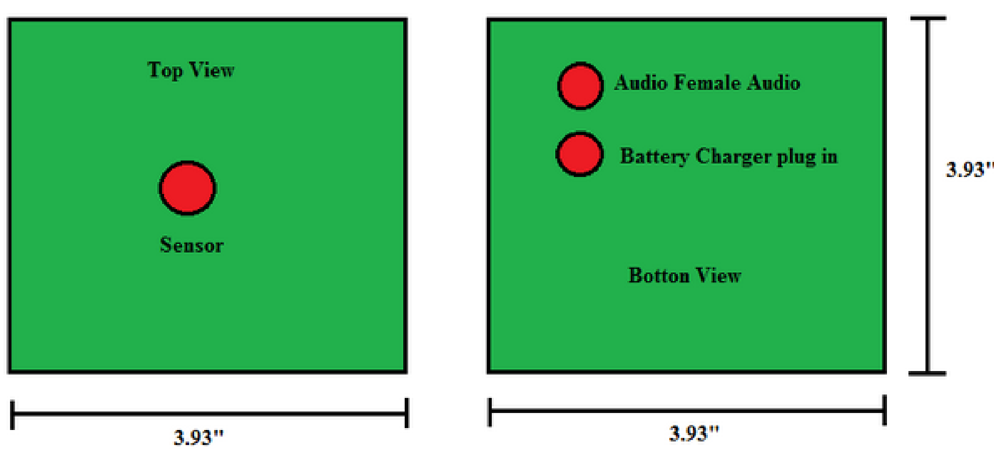


Figure 53: Top and bottom side of the LCD Housing

Finally, in Figure 54 shows the block diagram of each component and its connections. Most of the components will placed in a small rod that will go from the top the bottom. Starting from the bottom place, the battery will be place on top of the place connected to the charger. The charger output will be connected to its main source which will the FPGA in this case. The FPGA will distribute the amount of voltage to the Bluetooth, Microcontroller, LCD controller, and Wifi. The Microcontroller will power the motion sensors and the gyroscope. According to the dimensions of every component, everything should fit almost like in Figure 53. Besides that, there is a probability that the group will be using more than one motion sensor. This will be determine at the beginning implementation of the project.

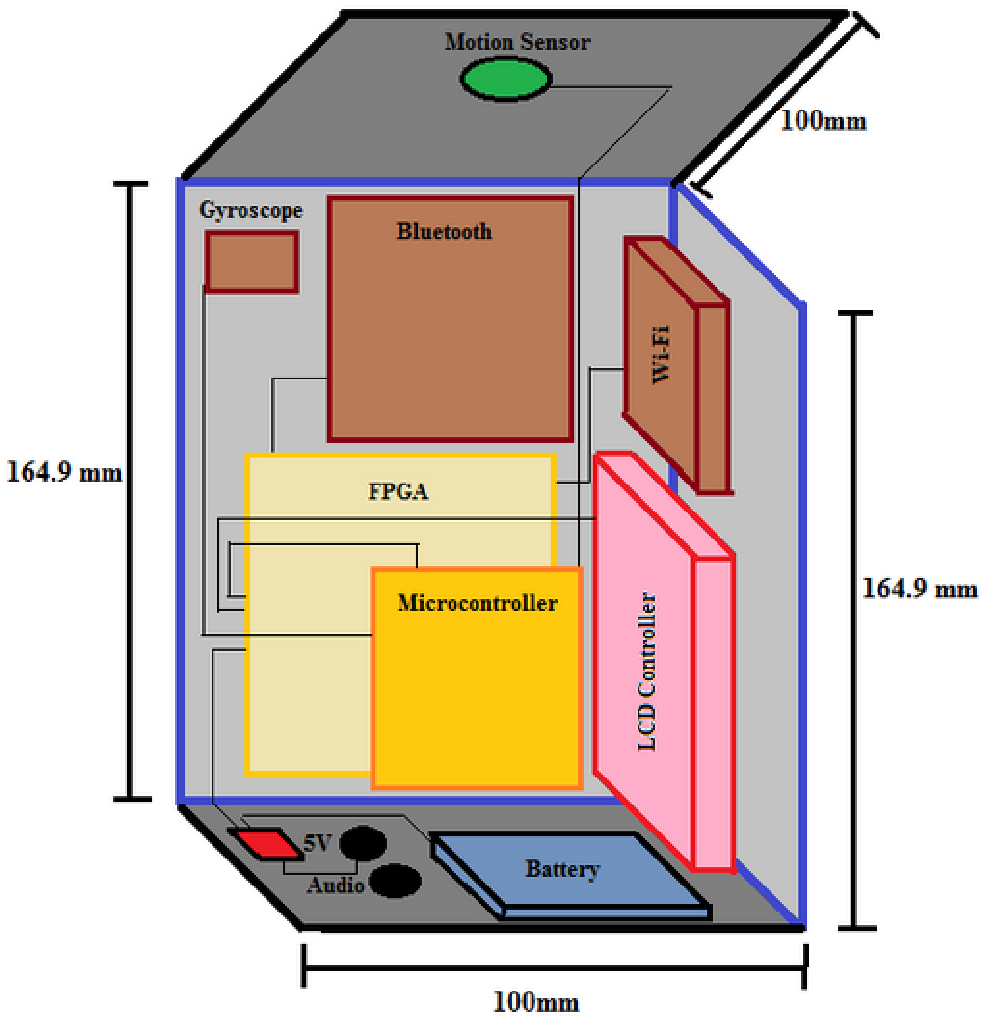


Figure 54: Diagram inside the LCD Housing

5 Project Design Summary

5.1 Hardware

The VESP project incorporates a variety of different hardware subsystems in its design. Each subsystem is essential to the VESP project. The sensor subsystem gives the VESP the ability to: sense the lumen levels of its current to adjust LCD screen brightness, use infrared light to detect whether there is someone in the room, and sense the VESP device's current orientation and acceleration to interact with the device. The LCD controller subsystem gives the VESP device the ability to display to its four LCD screens, as well sense touch input given by the user. The dock station subsystem gives the VESP project additional audio output options, as well as a means of recharging the VESP device's battery all at the same time. These subsystems are combined in order to fulfill the goals and objectives for the VESP project.

5.1.1 Parts List

The parts to be used in the construction for the final device are listed in Table 36 below. It is estimated that the team will need about 10 feet worth of wires to connect the components together.

Part Name	**#**	**Description**	**Purpose**
	1	Battery	Battery to power the VESP.
Hardkernel ODROIDD-XU3	1	FPGA	Main processing unit for the VESP.
TI MSP430G2553	1	Microcontroller	Main sensor handler.
TI CC3100	1	Wi-Fi module	Provide Wi-Fi wireless communication.
TI CC2564MODN	1	Bluetooth module	Provide Bluetooth wireless communication.
Panasonic EKMC1601111	1	IR sensor	Sensor to detect nearby motion.
Adafruit TSL2561	1	Ambient light sensor	Sensor to measure the ambient light in a room.
InvenSense MPU-3050	1	Gyroscope/Accelerometer combo	Sensor to measure the device's orientation and acceleration
TI TLK10022	1	1:4 Serial Link Aggregator	Split DisplayPort video signal into four separate signals
STMicroelectronics STDP4020	4	DisplayPort to RGB converter	Convert DisplayPort signal to usable RGB format
STMicroelectronics STMT07	1	Capacitive touch controller	Converts raw touch signals into usable digital signal
BuyDisplay ER-TFT070-4	4	LCD panel	LCD displays for the VESP.
BuyDisplay ER-TPC070-3	1	Capacitive touch panel	Capacitive touch panel for the VESP
DisplayPort Connector	1	DisplayPort Connector	PCB connector for DisplayPort
40-pin FPC-Connector	4	40-pin FPC-Connector	PCB connector for the LCD panel
30-pin FPC-Connector	1	30-pin FPC-Connector	PCB connector for the capacitive touch panel
USB 2.0 Connector	1	USB 2.0 Connector	PCB connector for USB 2.0
DisplayPort Cable	1	DisplayPort cable	Cable connecting LCD controller to FPGA.
40-pin FFC Cable	4	40-pin FFC cable	Cable connecting LCD controller to LCD panels.
30-pin FFC Cable	1	30-pin FFC cable	Cable connecting LCD controller to capacitive touch

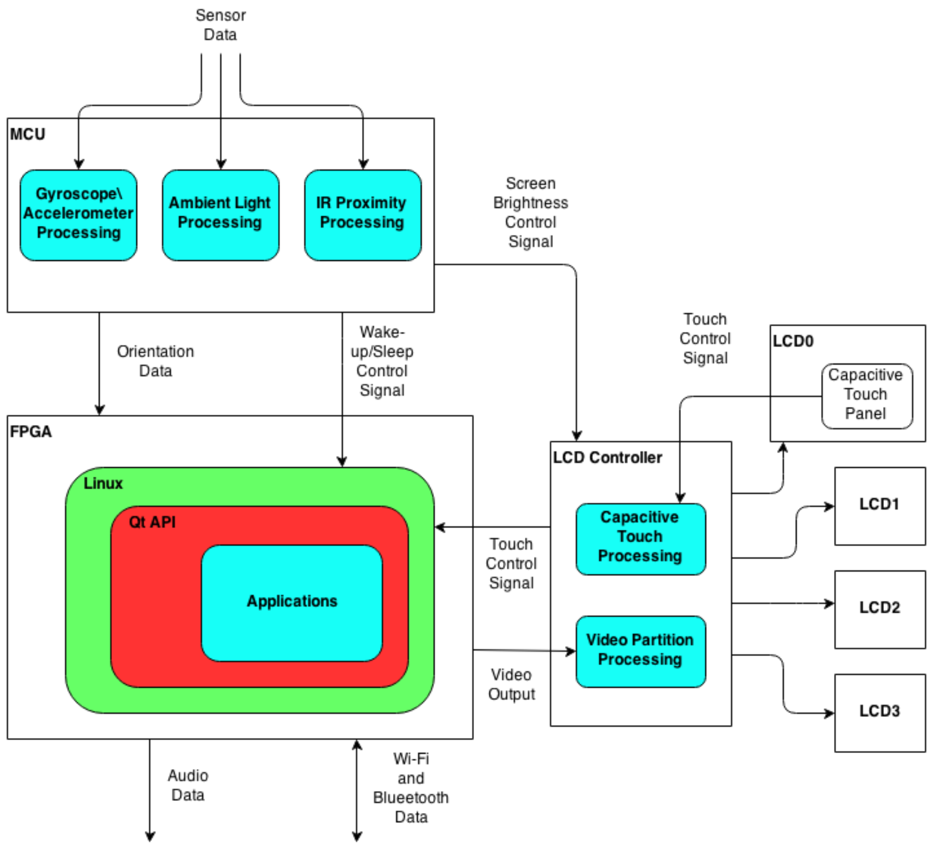
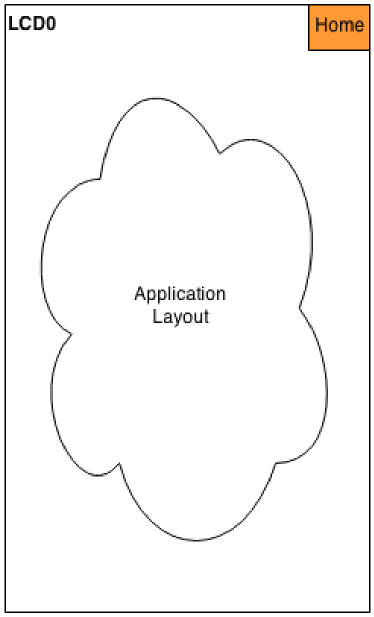


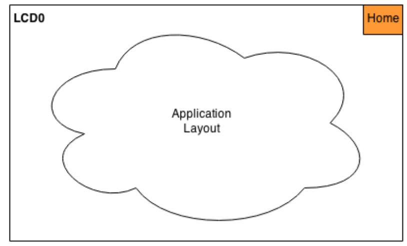
Figure 55: VESP High-Level Software Block Diagram.

5.2 Software

The VESP software will be housed in a Linux environment, more specifically within a Ubuntu environment. The Qt framework API will be used extensively throughout the development of the of the software for the VESP. This can be seen below in Figure 55, which shows again the high-level software block diagram for the VESP. The data flow within the VESP mainly goes to the main processing unit, the FPGA, to be handled and used to drive the software for the VESP. Figures 56, 57, and 58 below shows the vertical and horizontal UI Layouts to be used for the various software states of the VESP (home, lock, and application). The UI layouts are designed to show as much relevant information to the user as possible in a visually stunning way. These layouts will also allow the user to transition from state to state effectively, without sacrificing too much screen real-estate.

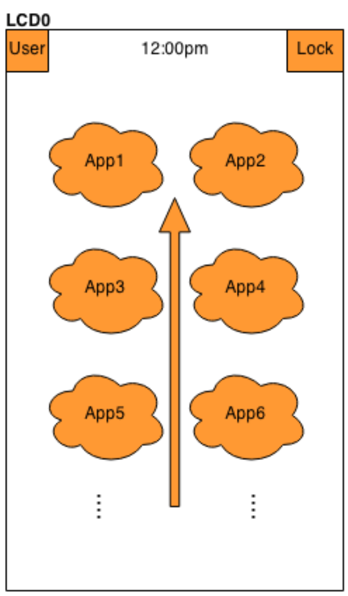


(a) Vertical Layout.

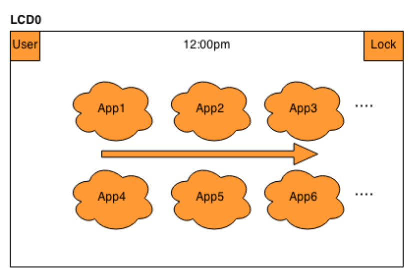


(b) Horizontal Layout.

Figure 56: VESP General UI Layouts

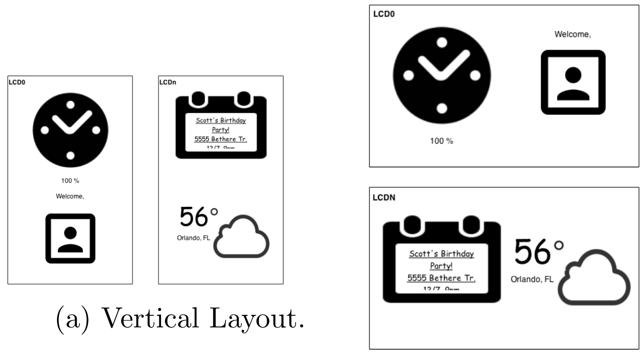


(a) Vertical Layout



(b) Horizontal Layout

Figure 57: VESP Home Screen UI Layouts



(a) Vertical Layout.

(b) Horizontal Layout.

Figure 58: VESP Lock Screen UI Layouts.

The applications that are included in the VESP are a music playback app, a weather app, a clock/alarm app, a calendar app, a settings app and a device status app. These apps will be developed using the Qt API in a C++ environment.

5.3 Housing

The Housing for VESP was divided into two sections: The LCD housing and the Base housing. The LCD housing is composed of the 4 LCD screens together with two plastics cover plates to cover the top and bottom of the open and also to cover the components and drivers. The cover plates will be the same size as the width of the LCD screens which is 100 mm. Aside from that, the top plate will have a hole for the motion sensor and the bottom plate will have two holes; one for the audio jack and the other for the battery charger. The Base section of VESP will be made out of wood with the dimension of 6" x 6" x 3". The front part will have two holes of 2" x 3" for the two speakers, and the bottom part of base will be open for an change that the group will like to make. It will be sealed and screwed when finished.

6 Project Construction and Coding

6.1 Building Strategy

The building strategy for the project is for each member of the team to be responsible for the assembly of a particular subsystem. Each subsystem is independent on the other subsystems, so this is feasible. This strategy is also desirable since each subsystem can be built at the same time as other subsystems. However, care must be taken when delegating subsystems, as each team member must make their assigned subsystem compatible with the other subsystems, otherwise face integration issues during the assembly of the final device.

The entire team is aware of this risk, and so another strategy when building the device components is to make each subsystem as modular as possible. Each subsystem should be independent of the other subsystems, and should resemble a "black box" (as in; inputs are received and outputs are transmitted, the inner workings are unknown and irrelevant) as much as possible.

6.2 Main Device Construction

For the construction of the main device, each subsystem previously discussed must first be assembled. In addition, the housing of the device must be constructed. Once all of the subsystems and housing are assembled/constructed, they must be tested individually to confirm that they function. After testing is done, integration is the next step. The integrated system must then be tested again to ensure that all of the subsystems integrated properly and that the device meets the requirements and specifications outlined previously in this report. Details on the tests to be performed on the different subsystems and software are discussed in a later section. In this section, the team will discuss the parts used in the construction of the device, the integration of these parts and the assembly of the final device.

6.2.1 Subsystem Assembly

Each subsystem of the VESP will be constructed using the parts listed in section 5.1.1. To recap, the subsystems that make up the main device are as follows:

- Sensor subsystem
- LCD controller subsystem
- FPGA subsystem

For the sensor subsystem, all of the sensors are to be connected and interfaced with the MSP430. The ambient light sensor and the gyroscope/accelerometer sensors will be connected to the MSP430 via the I2C interface. The IR sensor has a simple digital binary output, which will detect whether or not someone is present. A simple wire to one of the MSP430's GPIO pins, as well as a pull down resistor connected in series to the power source will suffice in connecting the sensor. Once all of this is done, the sensor subsystem is assembled.

For the LCD controller subsystem, the ICs and connectors must be connected to a single PCB and provided with power. The ICs must be connected correctly in order to achieve the desired function of splitting the video signal from the FPGA. On the display side of things, the DisplayPort connector pinouts must be connected to the input video signal pins of the TLK10022 IC. The four separate output video signal pins from the TLK10022 must then be connected to the input video signal pins of their associated STDP4020 IC. Finally, the STDP4020 output video signal pins must be connected to their associated 40-pin FFC connector which will drive the connected LCD screen. For the capacitive touch side, the 30-pin FFC connector must be connected to the capacitive touch panel that's attached to one of the LCD screens.

The connector pinouts must then be connected to the input pins of the STMT07 IC. Once that's done, the output pins of the STMT07 are then connected to the pins of the USB. After all of pins are connected and power is properly distributed to all of the components, the LCD controller is assembled.

Lastly, we have the FPGA subsystem. The FPGA itself is largely already assembled, all that is needed is to connect the WiFi and Bluetooth modules. For the Wi-Fi module, the CC3100 will be connected to the ODROID-XU3's available SPI interface. For the Bluetooth module, the CC2564MODN will be connected to the ODROID-XU3's available UART interface. Once the Wi-Fi and Bluetooth modules are connected, the FPGA subsystem is assembled.

6.2.2 System Integration

When all of the subsystems for the VESP device are assembled, the next step is to integrate them all together into one system. This step is key when constructing the device, since all of the separate subsystems must be able to communicate and transfer data in order for the VESP to accomplish all of its functions. The heart of the VESP is the FPGA, and so all of the subsystems will be integrated one way or another to it. For the sensor subsystem, the MSP430's remaining available GPIO ports will be connected to the ODROID-XU3's GPIO ports using wires. For the display side of the LCD controller, the video data will be sent to the ODROID-XU3 via a DisplayPort cable. As for the capacitive touch side of the LCD controller, the touch data will be sent to the ODROID-XU3 via the I2C interface.

6.3 Coding

6.3.1 MSP430 Coding

The MSP430 must be programmed in order to interpret the digital signals coming from the sensors, as well as send this interpreted data to the FPGA to be taken as input. The MSP430's I2C interface will be programmed to be an input port, allowing it to accept data from the ambient light sensor and the gyroscope/accelerometer sensor. As for the GPIO interface, certain ports will be programmed to be input ports in order to accommodate the IR sensor. Most of the GPIO ports, however, will be programmed to be output ports in order to send data to the FPGA. For the purpose of actually programming the MSP430, Code Composer Studio will be used as the IDE, since it can easy interface with the MSP430s USB interface.

6.3.2 Wireless Module Coding

As with the FPGA and microcontroller, the CC3100 Wi-Fi module will need to have code written in order to carry out wireless communication and networking. In our project we will need to program the CC3100 using several different software blocks that are apart of networking. The software blocks that will need to be coded into the CC3100 include the Wi-Fi subsystem initiation, which wakes the processor from hibernation mode; configuration of the Wi-Fi processor, this block is needed if the MAC address or the type of wireless network used in the processor change; directly connecting the wireless network and obtaining an IP address for the CC3100; creation of the TCP/IP sockets needed for connection to the network; the exchange and transfer of data to and from the network and finally the disconnection of the established sockets; a small snippet of programming will also place the Wi-Fi module into hibernation mode in the case that Wi-Fi is not connected to any network for an extended period of time. Shown below in Figure 59 is the state diagram that the CC3100 uses in a simple networking application, which is similar to the purpose and function that we will use for our project.

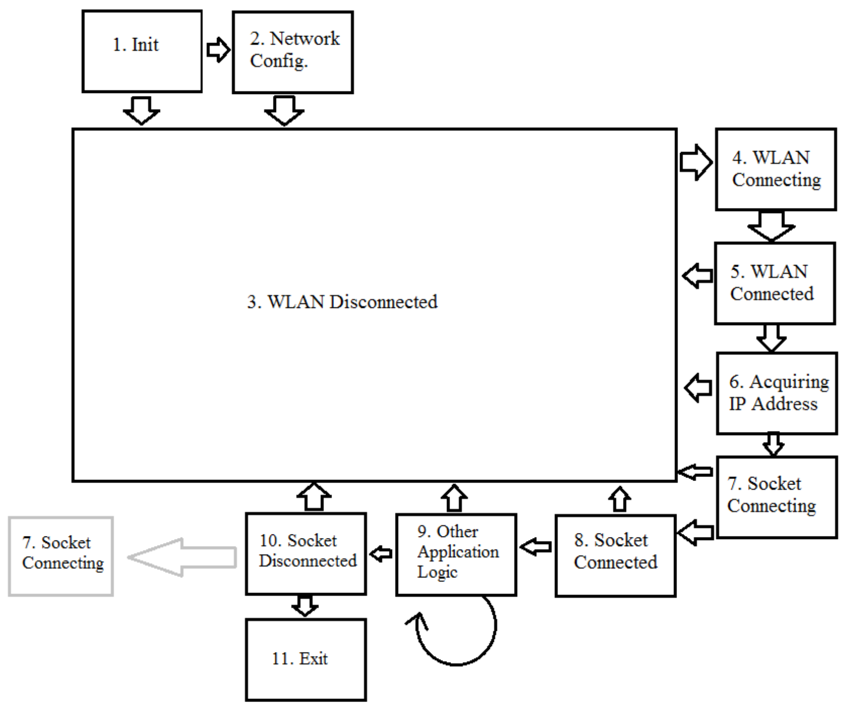


Figure 59: CC3100 State Diagram /newline Courtesy of Texas Instruments

The CC3100 follows an 11 state system with a mostly linear path, with many alternate states returning to state 3: WLAN Disconnected. In the case of state 1, state 2 is bypassed if the network configuration is already known. In the cases of states 5, 6, 7, 8, 9 and 10 a return to state 3 occurs if a disconnect event case is reached within the code. At state 11 the CC3100 closes all remaining open applications including: disconnecting from WLAN, stopping the SimpleLink and the running application. For the purpose of our project and our desire for it to have Wi-Fi communication we will need the WLAN to remain active under different networks meaning state 11 will loop back into step 1 upon leaving the area of wireless connectivity and signal.

Based on the state diagram shown in Figure 59 the first step in the operation the wireless communications of the CC3100 is the initialization of the device parameters that will be used in initial operations. Within the CC3100 the basic call for initialization to begin is the C command "sl_Start()" which enables the Wi-Fi subsystems, as well as the UART interface of the CC3100. This step within the program also sends a message to the Wi-Fi processor from the host that will be pinged back to the host driver to signal the completion of the initialization phase of the program. The initialization stage of the program takes a relatively long time to complete, tens of milliseconds. Lastly, the host driver is able to use one of two options when using the "sl_Start" command, the first blocks the application that is calling the Wi-Fi subsystem until the subsystem is fully initialized, in the second case a pointer is given that is called when initialization is completed, this can be used to handle multiple events by continuing to point new events back to be the initialization stage of the program.

The second step seen within the state machine is the configuration of the Wi-Fi subsystem to meet the network that it is connecting to. This is done through the device configuration settings of the CC3100, which ranges from basic information such as time and date to more important information such as device status, UART configuration and WLAN device mode. One of the most important configuration settings is how the Wi-Fi subsystem network will behave, as a station, an access point or a peer to peer connection. For the VESP the group will most likely be using the WLAN as a station, the purpose of the wireless communications on the VESP means that the group has no reason to configure the WLAN as an access point.

From a software development standpoint, the CC3100 and its networking capabilities are relatively straightforward to code. The host driver for the Wi-Fi subsystem is coded in the ANSI C programming language and supports any standard C compiler; however, for ease of use the group will use a Texas Instrument based software development kit (SDK), such as Code Composer Studio or the TI recommended CC3100SDK which includes sample applications to help with understanding the coding style and commands that will be implemented within the CC3100. The host driver of the CC3100 is able to support a single command at a time, requiring serial programming as opposed to parallel programming, such as that that can be used on an FPGA. The Wi-Fi subsystem host driver is able to handle asynchronous events, which is beneficial ascertain wireless communication tasks, such as establishing a connection, can be time consuming. The host driver operates on multiple bus sizes from the MCU, most notably 8, 16 and 32 bits, is capable of running based on any clock speed and supports both little and big endian memory formatting. The communication port from the driver to the network processor can be done through either serial peripheral interface (SPI), with a clock rate up to 20 Mbps as well as through the UART, with a baud rate of 115200, 8 bits, no parity with one start/stop bit. Lastly, the host driver for the CC3100 wireless network processor supports the use or non-use of an operating system, making it more flexible in the case the group decides against the use of an OS on board the FPGA or MCU.

CC3100 APIs

A common program that fulfills a network application onboard the CC3100 is composed of a total six different application programming interfaces (APIs), which are the Device API, which manages hardware functionality within the CC3100. The second is the WLAN API, which manages wireless protocols and connection profiles and policies. The third is the Socket API, which is set for user applications. The fourth is the NetApp API which assists with networking services including HTTP, DHCP and MDNS servers. The fifth is the NetCfg API which as its truncated name implies helps with many of the network configurations, such as IP and MAC addresses that are needed for a wireless network. The final API that is needed within the host driver of the CC3100 is the File System API which manages read and write operations performed while networking. All six APIs are crucial to the operation of the host driver as a wireless network processor and all six of these API modules will be needed in the VESP programming to ensure a solid and reliable internet connection.

API modules within the host driver operate from several different commands found in the program, a truncated list of these commands is seen in the tables below for each API module of the CC3100, as well as their functionality, and many of these commands will see use within a networking application, such as the one that will be included on the VESP. Following the tables will be an example of the coding that may be seen in the development of the CC3100 in the VESP (Figure 60).

Command	**Description**
Sl_DevGet	Enables the reading of device params
sl_EventMaskSet	Masks asynchronous events
sl_EventMaskGet	Return event bit mask
sl_UartSetMode	Used to set UART configuration (Baud, flow and com port)

Table 37: Device API commands

Command	Description
sl_WlanSetMode	Set the mode of the WLAN network (Station, AP, P2P)
sl_WlanGet	Enables WLAN parameters to be configured
sl_WlanPolicySet	Manages WLAN policies including power management, connection and role in a P2P connection
sl_WlanConnect	Manually connect to an existing network
sl_WlanDisconnect	Disconnect from the current network

Table 38: WLAN API commands

Command	Description
sl_Socket	Create a new socket and define its type and resource allocation
sl_Close	Close the socket and release allocated resources. Connection terminated if TCP
sl_Recv	TCP socket data read
sl_Send	TCP socket data write
sl_RecvFrom	UDP socket data read
sl_SentTo	UDP socket data write

Table 39: Socket API commands

Command	Description
sl_NetAppStart	Enables networking services (HTTP/DHCP/MDNS)
sl_NetAppStop	Disable networking services
sl_NetAppDnsGetHostByService	Returns server services (IP address, ports and text) based on service name.
sl_NetAppGetServiceList	Obtains peer sevice list in various structures
sl_NetAppPingStart	Ping network hosts

Table 40: NetApp API commands

Command	Description
sl_NetCfgSet	Sets configuration of networking functionalities (MAC address, IP address (Static or dynamic)
sl_NetCfgGet	Reads back network configurations set using above command

Table 41: NetCfg API commands

Command	Description
sl_FsOpen	Open a file in Sflash
sl_FsClose	Close file in Sflash
sl_FsRead	File data read
sl_FsWrite	File data write
sl_FsDel	Delete a file or all files of a particular format from Sflash
sl_FsGetInfo	Get information about the file including its size and allocation

Table 42: File System API commands

```
Example WLAN connection code

void SimpleLinkWlanEventHandler(void *pWlanEvents)
{
SlWlanEvent_t *pWlan = (SlWlanEvent_t *)pWlanEvents;
switch(pWlan->Event)
{
case SL_WLAN_CONNECT_EVENT:
g_Event |= EVENT_CONNECTED;
memcpy(g_AP_Name, pWlan->EventData.STAandP2PModeWlanConnected.ssid_name, pWlan-
>EventData.STAandP2PModeWlanConnected.ssid_len);
break;
case SL_WLAN_DISCONNECT_EVENT:
g_DisconnectionCntr++;
g_Event |= EVENT_DISCONNECTED;
g_DisconnectionReason = pWlan->EventData.STAandP2PModeDisconnected.reason_code;
memcpy(g_AP_Name, pWlan->EventData.STAandP2PModeWlanConnected.ssid_name, pWlan-
>EventData.STAandP2PModeWlanConnected.ssid_len);
break;
default:
break;
}
}
```

Figure 60: Example Software Code for the CC3100
Courtesy of Texas Instruments

Wireless Communication Software

In order to create the programming and coding needed to operate the wireless connectivity and communications of the CC3100 and CC2564 modules we need a software suite in order to develop. test, debug and finally complete the programs. In that regard, the group has decided to use TI's Code Composer studio in order to meet our software development needs, with selected downloads from the dedicated software development kits for both the CC3100 and CC2564. Code Composer studio is a program that the group already has experience using from previous electrical engineering coursework and should still be located on the computers of those students. The only requirements that needed to be met for the software development of the Bluetooth and Wi-Fi components of the VESP is that it must be able to compile C code, which CCs does wonderfully. The dedicated SDK's for both the CC3100 and CC2564 will be repeatedly referenced by the group as we develop the programming for the purpose of studying the code of the demonstrations and using that to help us code and debug our own developed code.

Even though the group has elected to use CCS as our programming platform for the wireless communications on the VESP the group will use demonstration codes and samples from the dedicated software development kit for both the CC3100 and the CC2564. The SDK for the CC3100 is conveniently called the CC3100SDK and is available as a free download from TI's website. The SDK contains the drivers and many sample application codes ranging from sample email codes, sample weather code and sample http server code among many others. This particular SDK works with other MCUs in the MSP line, notably the MSP-EXP430FR5969 and the MSP-EXP430F5529; however, as the group has decided to use an MSP430 board as opposed to an MSP-EXP430 board the group is unable to operate this SDK with it; however, it will still be downloaded so that the group has access to many sample codes to guide the conception and creation of the Wi-Fi capabilities of the VESP.

The CC2564 also has its own dedicated SDK in the TI CC256XMSPBTBLESW contains in it sample applications ranging from phone alert status and heart rate monitors. The Bluetooth software stack contained in the SDK supports the Bluetooth 2.1 with DR protocol as well as Bluetooth 4.0. The software stack allows for the active enabling and disabling of user profiles and protocols and contains a fully documented API interface. As with the CC3100SDK, the CC2564 SDK is available for the group free to download and mess with from TI's website online. The codes and sample that are received from the software download will then be moved to the CCS suite in order for the group to program and flash the CC2564 to communicate with other Bluetooth devices.

Wireless Communication Reference Designs

As the CC3100 Wi-Fi module is a pre-made module with emphasis on the group working the software and programming aspects of the device the module is already made for us and no design work went in on the groups end. The PCB that makes the CC3100 is a 4 layer board that is fitted on the surface layer with several micro-electronic components including the resistors, capacitors, connectors, inductors, RF antenna and crystals required to make the CC3100 self-contained and easy to implement. The BOM provided within the information package available on TI's website includes the manufacturer of the part, the part number from the manufacturer and the value and case size of the part in question if that information is needed. Located below in Figure 61 is the Reference Placement and Design of the CC3100 of the top layer of the four layer board.

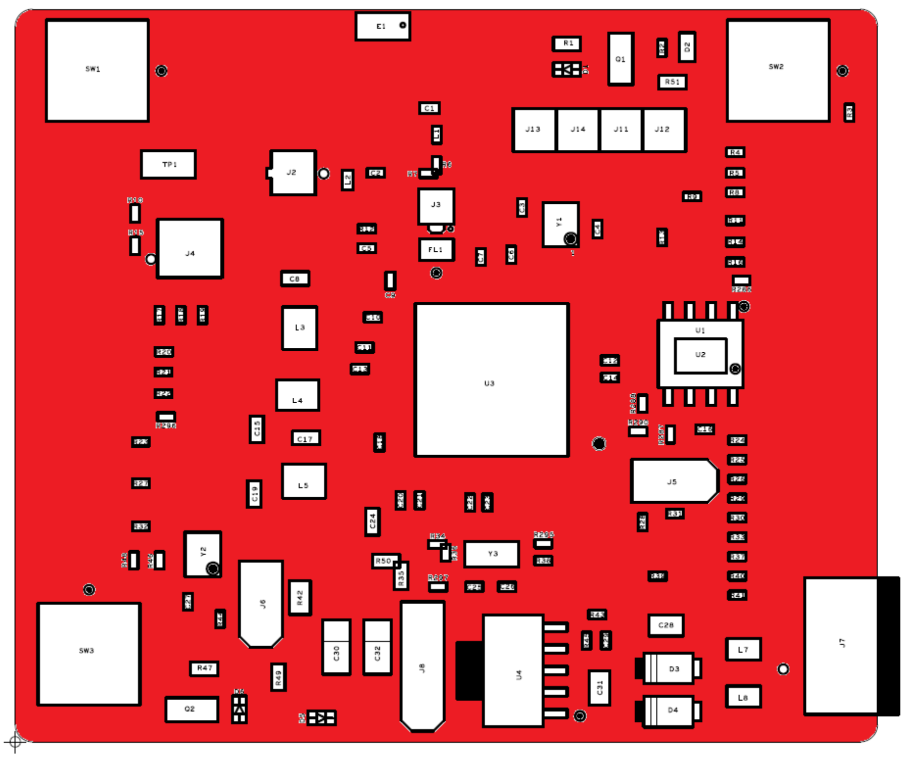


Figure 61: The top level PCB of the CC3100 /newline Courtesy of Texas Instruments

The individual components needed to create the module are located on the BOM list provided by TI and is a tad too long to include here. The CC3100 will be purchased direct from TI pre-fabricated so the group can quickly move into software programming and implementation of Wi-Fi on the VESP.

7 Project Prototype Testing

7.1 Hardware Tests

7.1.1 Sensors

Each of the sensors must be tested not only to make sure they work, but to test if they meet the specifications and requirements they will be used for in the project. The project group has devised a number of ways to test each sensor to ensure they were the correct choice for the project.

Motion Sensor

The first sensor to be discussed is the motion sensor. The first part of testing this sensor will be connecting it to a power supply to make sure it operates. The next step will be to connect it to the MCU which will be connected to a computer via USB (which allows for powering and interfacing with the Launchpad). The computer will have Code Composer Studio open to interface with the microcontroller board and use a small segment of code to read the input of the motion sensor, and configure the Launchpad light up an LED when motion is being sensed. Once that stage is completed, the range of the sensor will be test. One of the project group members will walk past the sensor at least five times per distance interval, from different directions. Because the motion sensor has maximum range of up to 16.4 feet, the tested distances will begin directly at the sensor, and then be incremented by two feet up to 18 feet. After “passing-by” motion has been tested, the next step will be to test the sensor by walking towards it and away from it at slightly different angles. Once the project group has determined its max range and sensitivity to human-size movement through these test, one final test will be conducted to test its sensitivity to size. One of the project group members will stand at different increments of distance away and stand still until the sensor no longer reads that there is movement, then begin making small movements until the sensor finally picks up on it. It’s assumed that users of the VESP may want to activate their device by a simple wave of a hand, so this sensitivity test will determine if that would be possible.

Ambient Light Sensor

The testing for the ambient light sensor is likely to be far more complex than the motion sensor due to the amount of data that must be analyzed. This testing will have to come after the testing of the LCD displays, because the project group must learn how to program the LCDs before the light sensor can be used to determine the brightness adjustment necessary for the LCDs. The setup for the hardware to test this sensor will be as shown in Figure 62.

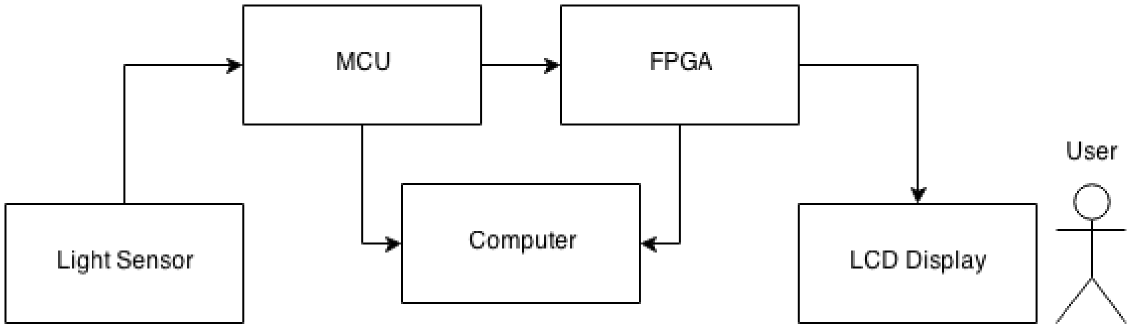


Figure 62: Ambient Light Sensor Test System

In the figure, both the MCU and the FPGA are connected to a computer so they can both have their interfacing software open for programming and data monitoring. The first test will only involve the light sensor and the MCU/computer. The sensor will be tested to see if it can recognize drastic, sudden changes in lighting. This test is important not only to determine if the sensor is functioning properly, but because this will be directly applied to the project as sudden, drastic changes in lighting will be a way for the VESP to recognize that a user has entered the room (e.g. turning on a light switch). The next will prove to be more difficult. The MCU will be programmed to analyze the sensor data and determine if a brightness adjustment is necessary. It will tell the FPGA to perform this adjustment and the LCD display will respond accordingly. However, the calibration of this needs to be very precise, ideally meeting visibility as closely as it can with optimal power saving. This will require multiple trials in a number of lighting changes whether it's bright in a room then the lights are turned off or a simulation of using the VESP by a window as the sun sets. These tests must also be conducted by multiple users to make sure the calibration is more precise. Programming will be tweaked and edited between each test to make sure the screen can be seen in any lighting setting.

Gyroscope

The function of the gyroscope will be to tell the orientation of the VESP so that it will switch the LCD displays between landscape and portrait mode when necessary and to know which screen is facing the most upward for use in a power saving mode. The tests conduct will simulate these events. As with the other sensors, the first test the gyroscope will undergo is connecting it to a power source to see if it works. The next test will be for determining whether the LCDs need to change their display between portrait and landscape. This test will be conducted using only one LCD. First, the project group will construct a box (possible out of cardboard) to simulate the "cube" portion of the VESP. The gyroscope will then be attached to this box to simulate the actual position in the VESP it will take. This is important because the sensor's measurements must be calibrated from where it will be relative to the LCD displays. From here, the gyroscope will then be connected to the rest of the test system as seen in Figure 63. The LCD will not be attached to the test box but its corresponding face will be noted. The box will be shifted around to simulate a user interacting with the VESP while the LCD is calibrated via programming the MCU to determine when to switch its display's orientation. The next test will be conducted to calibrate the VESP to determine which LCD is facing upward (as if the user is holding it and looking at the topmost screen). Again, the test box will be moved around simulating a user's actions while the screen is calibrated through programming the MCU. This test will be done four different times, the LCD corresponding to a different side of the test box each run, until the programming for all four sides is complete.

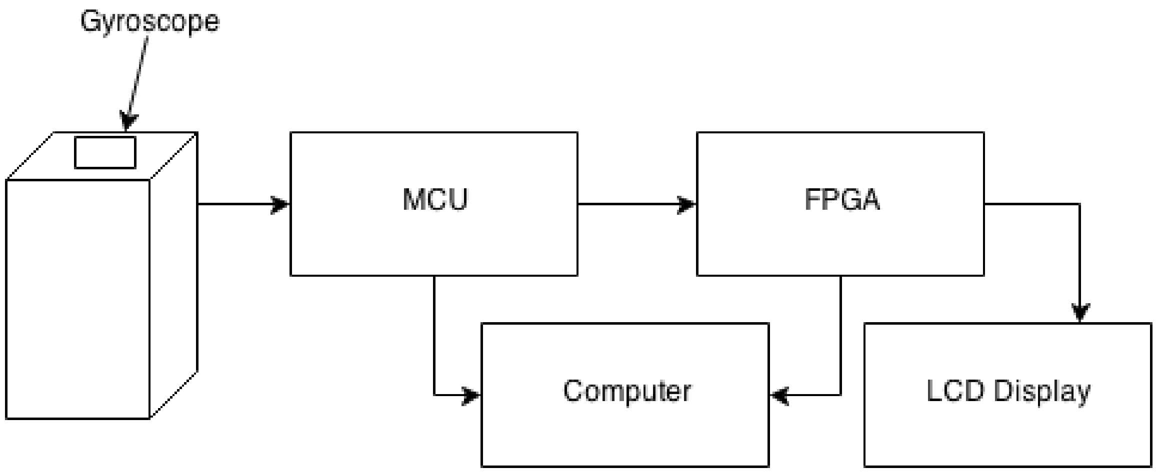


Figure 63: Gyroscope Test System.

7.1.2 LCD Screens and Controller

Making sure that the LCD Screens of the VESP work correctly is paramount to the VESP's normal function. In order to correctly test the entire LCD system (LCD screens, touch panel, and the LCD Controller) the team must do the following:

- Check each individual LCD panel to confirm that they are all functional.
- Check each component on the LCD Controller to ensure that the right components are connected.
- Connect the LCD panels to the LCD Controller and check if all four LCD panels are displaying the expected image.
- Connect the touch panel to the LCD Controller to confirm that it can register input from the user.

The first thing to do is test each individual LCD panel. Each LCD panel will be tested to ensure that they can all correctly display any image. In order to do this, the team will use a breadboard to breakout the pin connectors of the LCD panel and send artificial data to the LCD panels. This artificial data will be used to visually determine whether the each LCD panel is able to correctly display red, blue, and green color channels, as well as adjust the backlight brightness. team will test each individual LCD screen to make sure that each panel can correctly display an image.

The second item on the list is to ensure that the components that are meant to be connected are actually connected and components that are not meant to be connected are actually not. To do this, the team will test for continuity on each connection on the PCB. Continuity can be measured using a multimeter set to measure from impedance. If there is very low impedance, then the two end-points are connected, if the impedance is a tad larger (in the realm of a few Ohms) then the two end-points are not connected. Once both the LCD panels and the LCD Controller are tested, they can be connected and be tested together. The team will test the integration of the LCD panel and the LCD Controller based on the image displayed on the LCD panels. The FPGA will be used in this test to send data to the input DisplayPort on the LCD Controller. The data will be a simple image, if the output image on each LCD panel is as expected, then the two components work and interact as they should.
Finally, the touch panel will be tested. The I/O port on the LCD Controller used for sending touch data will be connected to the FPGA as it would during the VESP's normal operation. The touch data will be measured and processed into a 2-dimensional coordinate, this coordinate will be roughly compared to where the user gave the input. If the coordinate position sufficiently supports that the touch panel was touched where it was, then the touch panel is working correctly. Table 43 shows a more detailed summary of the different tests that will be conducted on the LCD Controller subsystem.

Component	Test Description	Expected Result
LCD Panels	Each LCD panel will be hooked up to a breadboard in order to feed the panel test data. The data will include simulated RGB data, as well as backlight brightness data. The RGB data will fluctuate starting with red being the only channel present, then the green or blue channel will be incorporated, and so on and so forth until all possible color combinations are achieved. The analog backlight brightness signal will change within the LCD panel's operating range, from completely off to the its brightest setting.	Colors should display on the panels as expected. Also, no dead pixels should be present. Backlight brightness should be able to change from completely off to maximum brightness.
LCD Controller	Each connection between component will be tested for continuity using a standard multimeter.	Components that are intended to be connected will show an impedance of <1 Ohm. Components that are not intended to be connected will show an impedance of >3 Ohms.
LCD Controller with LCD Panels	Image data will be sent to the LCD Controller via the FPGA's DisplayPort.	The image correctly displays on all four LCD panels without cutting off part of image. In addition, the portion of the image intended for a certain display panel must display on its intended display.
LCD Controller with Touch Panel	Touch data will be recorded from the user via the FPGA and LCD Controller I/O ports. The touch data will be translated into a 2-D coordinate.	The 2-D coordinate should match the region of where the user touched the display.

Table 43: Detailed Test Summary for the LCD Controller Subsystem.

7.1.3 Wireless Communication

As the wireless communication modules are both an electrical circuit as well as a data transceiver they must be tested in two separate ways to ensure that both the modules work correctly and as the group needs them to. In order to test the CC3100BOOST Wi-Fi module we will first apply the recommended operating voltages to the devices input pins and then check to ensure that the voltage and current output from the device are as the group expects from the datasheet provided by TI. The recommended values of the voltage inputs and expected outputs and currents is located below in Table 44. Testing the software aspect of the device is slightly more tedious and time consuming. First we must develop a simple program that is able to send and receive packets of data, or in the case that we find demo software code that does this for us download it and flash the code to the CC3100. After the program code has been flashed we run the code and check that it both connects wirelessly to the internet and check that the data rate of the packets is what the group programmed into the module. The testing for software will utilize an MSP430 Launchpad in order to test the software side anywhere we have access to a computer with the ability to flash code to the CC3100.

Parameter	Pin	Testing Value	Expected Return
Input Supply Voltage	10, 37, 39, 44, 54	3.0 V	
Output High Voltage			2.4 V
Output Low Voltage			0.4 V Supply Voltage
Output High Current			6 mA
Output Sink Current			6 mA

Table 44: Wi-Fi Module Testing Voltages and expected Returns

The CC3100 has dedicated hard points and connectors on the board to test various traits such as RF characteristics and voltage and current draws. Current testing of the CC3100 is performed by removing a jumper on the board and replacing it with an ammeter allowing the current to be measured. TI recommends using an ammeter measurement while the board is operating in a low current mode (hibernate, low power deep sleep), for active currents it is recommended that a small resistor is placed in the place of the jumper and the voltage differential is measured. Using Ohm's law in conjunction with the known resistor value and the discovered voltage differential the current across the resistor can be found. For RF testing there are two connectors, one to connect to an external antenna in the event one is used and a second to connect to an RF testing device. The group does not anticipate much RF testing over the course of the construction and prototyping of the VESP and most hardware testing of the CC3100 will focus on maintaining the correct voltage, current and power consumption values as found in the datasheet of the CC3100.

Testing the Bluetooth module is a similar task. First we apply the needed input voltages to the input supply power pins on the Bluetooth module and then we probe the output pins to check that the correct voltage and current is being received from the output. We proceed once the device is correctly operating from an electrical perspective; we turn to the software and programming aspect of the device. We build or download a snippet of coding that will send data packets at a controlled and known rate, we then check this data rate using another Bluetooth enabled device that is able to track its incoming data packets. If the data packets and data rate are the same than the group is comfortable enough to say that the Bluetooth module is working both from a hardware and software perspective. Lastly, once both Wi-Fi and Bluetooth have been tested independently the group will interface the two devices in close range simultaneously in order to see if the Wi-Fi module or Bluetooth module suffer from radio interference brought about by the other device in testing, if they cause problems during testing then these problems will only magnify as the group moves to produce the prototype VESP. The table below, Table 45 will list the operating voltages and the output voltage and currents that will be used and expected from the electrical testing of the CC2564 module. As the software testing of both the Bluetooth and Wi-Fi components of the VESP do not require access to any important lab bench machines, it will be easier for the group to test each component individually wherever a computer and code compiler such as CCS is located.

Parameter	**Pin**	**Testing Value**
Input Supply Voltage	12	3.0 V
I/O Power Supply	18	1.8 V

Table 45: Testing Voltages for the CC2564

7.1.4 Power Supply

When building the power supply, the group has to make sure that the we won't create any fire hazard that will create damage to our components and extent the time to our projects implementation. To avoid any problem, the group decided to test each components voltages and currents. First we gotta measure that the secondary side of the transformer outputs 12V AC. The second test will be on a breadboard after the rectifier and capacitors are connected and its outputting a direct current. The group will be using a breadboard in all testing before we design and print on PCB board so we don't waste money and time if the the board catches on fire or other components are needed.

Another test that we will have to take in consideration is before connecting the voltage regulator due to the possibility of overheating. We need see the efficiency of the voltage regulator and avoid high input voltages because he higher the input voltage, the lower the efficiency and the higher probabilities to heat sink. For all voltage regulators, there is a dropout voltage which is the amount of voltage you need before the voltage regulator starts losing regulation. Most voltage regulators are need to have an input voltages have to be one or two voltages above the output regulated voltage. We will definitely will be using a 5V regulator, because that the amount of voltage to charge the battery and the speakers

7.1.5 Audio

The voltage going to the amplifier has to be tested before being connected. We cannot go over the maximum voltage for the board. Also, the group has to double check that the resistance of the amplifier is the same resistance of the speakers because by any change if we get a higher impedance, we can seriously damage the speaker and the amplifier. Finally the last testing of the audio system will be making sure that the volume works accordingly to the users input and that when the LCDs screen is placed in the base, the audio will be working without any problem.

7.1.6 Battery Charging

When it comes to charging the battery we just have to make sure that the current from the coming from the power supply that connects to the micro-B USB should not be higher that 500mA because we can damage the whole board. Also, because the charger board has a current protection, we don't have to worry about the battery being damage and because the current use to charge the battery does not exceed 2000A which is the maximum current about to charge the battery. Current Protection was one of the reasons why the group decided to chose this battery charger.

7.2 Software Tests

7.2.1 Applications

Each application must be tested in order to confirm that they function as intended. In this section, a variety of tests shall be outlined for each application available on the VESP.

Music

The music app tests are pretty straightforward. The tests will help confirm that the music app can do the following:

- Stream music from the Bluetooth paired Android device.
- Use the music data to dynamically render a visualizer on the VESP displays.
- Automatically hide the music playback controls and "home" button after five seconds of inactivity.
- Automatically display the music playback controls and "home" button once a touch input has been registered.
- When the device is tilted from vertical position to horizontal, the playback controls should follow suit and vise versa.

The first and second item on the list are the music app's essential functions, while the other two items are required for controlling the app. The music app's music streaming capability will be tested using a Bluetooth paired Android device, that will attempt to send an arbitrary audio file over the air to be played through the VESP's audio output port. If the audio file is both received and played correctly, the test was successful. For the visualizer function, the app must be able to dynamically render a unique the visualizer based on the music data being streamed in. In turn, there will be two tests for this function; one will test the app's ability to render the visualizer on all four LCD screens, while the other test the app's ability to render two unique visualizer sequences given two unique audio files. The test for the other three functions of the music app are straightforward and are described in Table 46 below along with a detailed summary of the other tests.

Function	Test Description	Expected Result
Music Playback	An Android device will be paired to the VESP via Bluetooth. The Android device will play an arbitrary audio file, having Bluetooth as the audio output.	The audio file should play correctly, without lag or stutter.
Visualizer Rendering	This test will be split into two: 1. The Bluetooth paired Android device will send an arbitrary audio file to the VESP for rendering 2. The Bluetooth paired Android device will send two unique arbitrary audio files to the VESP rendering	Each test will have an expected result: 1. The rendered visualizer should display on the all four LCD screens of the VESP. 2. The two rendered visualizer should be unique enough to the able to be differentiated from each other.
Auto-hide User Controls	While the user controls are present, a timer will measure the amount of time without user touch input.	After roughly five seconds, the user controls should disappear from the display.
Auto-show User Controls	While the user controls are not present, touch input will be given to the VESP	The user controls should appear only after given the touch input.
Auto-tilt User Controls	While the user controls are present, change the orientation of the device from vertical to horizontal, and horizontal to vertical.	The user controls should reorient themselves according to the change in orientation executed.

Table 46: Detailed Test Summary for the Music Application.

Weather

The following functions of the weather app will be tested:

- Accurately display weather information retrieved from the National Weather Service server on the touch-enabled panel.

- Accurately (to a certain degree) calculate the current location of the VESP device using Wi-Fi.
- Correctly transition from the current day's weather information to a seven day forecast, and vice versa.
- Correctly orient the UI to match the orientation of the device.
- While in the lock screen state, display weather information on the adjacent LCD panels

The first function will be tested by comparing the displayed information to the information available on the National Weather Service server. The second function will be tested by comparing the geolocation that the app is displaying information for with the actual location of the VESP, a certain degree of error is expected. The third function will be tested by touching the appropriate touch button on the app and observing the output. The fourth function will be tested by changing the orientation of the device. Finally, while the VESP is on the lock screen, relevant weather information should be displayed on the adjacent LCD panels. Table 47 below shows a detailed summary of the tests that were previously mentioned.

Function	Test Description	Expected Result
Weather Information Consistency	Information that is being displayed for the current day and seven day forecast will be compared with the information available on the National Weather Service server.	Information on the device and the server should be consistent.
Geolocation Calculation	Calculated geolocation will be compared to the device's current, actual geolocation	Calculated geolocation should be at least within a two mile radius from the actual geolocation.
Transition to Seven Day Forecast	While the app is displaying the current day's weather information, the current day's label will be touched by the user.	The application should transition to display the seven day forecast.
Transition to Current Day Forecast	While the app is displaying the seven day weather information, the seven day label will be touched by the user.	The application should transition to display the current day forecast.
Rearrange UI Elements According to Orientation	While using the app, either in current day or seven day view, change the orientation of the device from vertical to horizontal and vice versa.	The UI elements of the application should re-arrange themselves according to the current orientation of the device.
Display Weather Information While Locked	While the VESP is on the lock screen state, wait three seconds	The current day's weather information should display on their the right or left adjacent LCD panels.

Table 47: Detailed Test Summary for the Weather Application.

Calendar

The following functions of the calendar app will be tested:

- Accurately display events and reminders that are retrieved from the Bluetooth paired Android device on three of the VESP's display.

- Transition to the next or previous day/week/month by swiping right or left on the touch-enabled panel
- Transition to display events in day view to week view, as well as week view to month view , and month view to the current day.
- Transition from month view to a particular week, and from week view to a particular day.
- Correctly orient the UI to match the orientation of the device.
- While in the lock screen state, upcoming events within a week's time will be displayed on the adjacent LCD panels.

The calendar events displayed on the app are dependent on the Bluetooth paired Android device, so naturally, the information should be compared to ensure consistency. Transitioning from the different calendar views and navigating between adjacent days/weeks/months will be tested by giving touch input. The calendar app will utilize all of the LCD screens but the backfacing panel, this aspect will also be tested. The calendar app UI elements, on all LCD screen should orient themselves according to the device's current orientation. Lastly, while in the lock screen state, relevant event reminders will be displayed on the adjacent LCD panels. Table **??** below shows a detailed summary of these tests.

Function	Test Description	Expected Result
Calendar Information Consistency	Calendar events being displayed on the app will be compared to the calendar events being displayed on the Bluetooth paired Android device.	Information on the app and the Android device should be consistency.
Adjacent View Transition	Swipe right and left gestures will be done on the touch-enable panel if oriented vertically, and swipe up and down gestures when oriented horizontally.	When a swipe right/down is executed, the previous day/week/month will be moved to the touch-enabled panel. When a swipe left/up is executed, the next day/week/month will be moved to the touch touch-enabled.
Transition Between Views	The current view's label will be touched.	A touch on the day view label should transition to the week view. A touch on the week view label should transition to the month view. A touch on the month label should transition to the current day view.
Particular View Transition	From month view, a particular week will be touched. From the week view a particular day will be touched.	When a particular week is touched from the month view, the app should transition to the selected week. When a particular day is touched from the week view, the app should transition to the selected day.
Rearrange UI Elements according to Orientation	While using the app in month, week and day view, change the orientation of the device from vertical to horizontal and vice versa.	The UI elements of the application should re-arrange themselves according to the current orientation of the device. This should apply to all of the LCD panels.
Display Events While Locked	While on the lock screen state, wait three seconds.	Upcoming event notifications should display on either the right of left adjacent LCD panels. Upcoming events should be within a week's time from the current day.

Clock

The following functions of the clock app will be tested:

- Display the correct time to the user.
- Correctly orient the UI to match the orientation of the device.
- Create, edit, and delete alarms.

The clock app is one of the more simplel applications on the VESP and doesn't require much testing. In addition, there is no reliance on external sources for its operation. The first function is very straightforward, the displayed time need only be compared to the known time. The second function will be tested by orienting the device both from vertical to horizontal and vice versa.The last function will be tested by creating new alarms, editing and deleted them, and making sure that the expected result is achieved. Table 49

Function	Test Description	Expected Result
Display Correct Time	The time displayed by the app will be compared to a time that is known to the accurate (cellular phone, online resource, etc.)	The displayed time should match the known time and should remain accurate after a day's time.
Rearrange UI Elements According to Orientation	While using the app, change the orientation of the device from vertical to horizontal and vice versa.	The UI elements of the application should re-arrange themselves according to the current orientation of the device.
Create New Alarm	A new alarm will be created.	Once the time that the alarm was set to is reached, the VESP should output an alarm noise.
Edit Existing Alarm	An existing alarm will be edited to a different time.	Once the time that the edited alarm was set to is reached, the VESP should output an alarm noise. In addition, the original time of the alarm should be ignored and no noise should play when that time is reached.
hline Delete Existing Alarm	An existing alarm will be deleted.	Once the time that the deleted alarm was set to is reached, the VESP should do nothing.

Table 49: Detailed Test Summary for the Clock Application.

Settings

The only unique function of the settings app is to change settings that the VESP and its apps take in as parameters during operation. The testing for this app is dependent on the amount of settings available on the final device, as each setting must be tested to ensure that it had the desired effect. For example, when the Wi-Fi setting is set to "off", applications such as the weather app will cease to correctly function and the device status app should also show this change. Another example would be when the clock setting is set to "Analog", the lock screen and the clock app should display an analog clock, instead of a digital clock. In addition, like other apps previously discussed, the UI elements of the clock app should re-arrange themselves according to the change in orientation of the device (ie: vertical to horizontal, and vise versa)

Device Status

Like the settings app, the device status app only has one function. The device status app's only function is to display information pertaining to the device current operating conditions. Again, the testing for this app is dependent on the final device, however, the device status app will at least include information on the device's current battery capacity, whether Wi-Fi or Bluetooth are on or off, the name of the Wi-Fi network currently connected to (if any), and the name of the paired Android device (if any). In addition, like other apps previously discussed, the UI elements of the clock app should re-arrange themselves according to the change in orientation of the device. These tests will not be described further as they are self-explanatory.

7.2.2 User Interface

The UI of the VESP must be extensively tested to ensure that the experience of the user is as smooth and error free as possible. In this section, the home screen state and the lock screen state will be the main focus of the testing, with a small portion on the general application UI.

Home Screen UI

The interactive UI elements of the home screen, along with their function and location are outlined in Table 50 below.

UI Element	Function	Location
User Profile Button	This button will launch the device status app.	Top left corner of the home screen.
Lock Button	This button will override the wait time for the VESP to enter the lock screen state and will immediately make the device enter the lock screen state	Top right corner of the home screen.
Application Buttons	These buttons will launch their respective applications.	Middle, center aligned, in a two-button wide arrangement.
Application Field	This field will be used to scroll through the application buttons, and will be the container for the application buttons.	Middle, center aligned, in the background of the application buttons.

Table 50: Home Screen Interactive UI Elements and their Function and Location.

In order for the team to confirm that these UI elements work as intended, they must be thoroughly tested. Table 51 below outlines the tests to be performed on these UI elements to ensure they are ready for user interaction.

UI Element	Test Description	Expected Result
User Profile Button	Touch input will be done on the button.	The device status application should launch and operate as normal.
Lock Button	Touch input will be done on the button.	The device should immediately transition to the lock screen state.
Application Buttons	Touch input will be done on each application button.	After each touch input, the respective application should launch and operate as normal..
Application Field	Touch input will be done on the field. Depending on the type of touch input, a different result will be expected.	Single touch or hold: Nothing should happen. Drag up or down: The application buttons should move up or down depending on the direction of the drag. Drag left or right: Nothing should happen.

Table 51: Home Screen UI Element Tests

It is important to note that the home screen will also respond to the a change in orientation of the device and this must also be tested. This test is pretty straightforward: while in the home screen state, change the orientation of the device from vertical to horizontal and vise versa. The home screen should re-arrange the its UI elements accordingly.

Lock Screen UI

The lock screen UI is much simpler than the home screen UI, and in turn, so is its testing. As previously mentioned in this report, the lock screen does not have any actual interactive elements, and simply displays information. The only interactive function that the lock screen has is to hold its state until touch input is given by the user. This function is simple, but crucial and must be tested appropriately. The test is outlined by Table **??** below.

UI Element	Test Description	Expected Result
Entire Lock Screen	Touch input will be done on the screen.	The device should immediately "unlock" and transition to the home screen.

Table 52: Lock Screen UI Test

Again, it is worth noting that the lock screen will respond to the current orientation of the device, just like the home screen. This function will be tested the same way that the home screen was tested for the same function and will not be explained further.

General App UI

Each application of the VESP shares the general app UI layout. This layout is defined by a "home" button located on the top right corner of the screen, all other UI elements are dependent on the application itself. The general app UI ensures that the user doesn't get stuck on any particular app and can always go back to the home screen if an application was accidentally launched. Table 53 below outlines the test to be done on the general app UI to ensure proper navigation of the VESP.

UI Element	Test Description	Expected Result
Home Button	Touch input will be done on the home button.	The application should immediately close and the VESP should transition back to the home screen.

Table 53: General App UI Test.

Again, it is worth noting that the general app UI will respond to the current orientation of the device, just like the home and lock screens. This function will be tested the same way that the home and lock screens were tested for the same function and will not be explained further.

7.3 System Test

The entire system of the VESP (dock and device) must be tested once completely assembled to ensure that the device operates as intended by the team. This section will focus on testing the different interactions between hardware and software components, as well as other essential functions. Table 54 below shows a detailed outline of the various tests that will be conducted on the VESP to confirm correct system function.

System Function/Interaction	Test Description	Expected Result
Startup	While the VESP is off, the power button will be pressed.	The VESP device should power up all of its hardware components, and then display the lock screen.
Shutdown	While the VESP is on, the power button will be pressed briefly, then also pressed and held for three seconds.	When the power button is pressed briefly, the device should remain on. When the power button is pressed and held for 3 seconds, the device should shutdown.
Automatic Backlight Control	The VESP will be placed in a dimly lit room and then moved to a brightly lit room, and vise versa.	When transitioning from the dimly lit room to the brightly lit room, the backlights of the LCD panel of the VESP should dim. Inversely, when transitioning from the brightly lit room to the dimly lit room, the backlights of the LCD panels of the VESP should brighten.
Launch and Close Applications	Each application of the VESP will be launched from the home screen and then closed.	The VESP should launch and close each application in succession successfully without errors or bugs surfacing.
Orientation Response	While running each application, and while in each state. The device's orientation will be change from vertical to horizontal and vise versa.	The UI of each application and state should correctly render after each change in orientation.
Charge Battery	The VESP will be placed on the dock station to charge.	The VESP should show that the battery is currently being charged.
Audio Output	While connected to the dock or some other headphone/speaker device, the VESP will play a song through its music application and also sound an alarm through its clock alarm.	The VESP should correctly output the song and alarm sound through the connected audio device.

Table 54: VESP System Tests

8 Administrative Content

8.1 Project Milestones and Task Distribution

The project's implementation, design and build will be separated in different phases. All these phases will have a period of time to get the work done, as well as deadlines. Also, these phases are separated in different categories, and each category will have an assigned group member that will be responsible for meeting each deadline as seen in Tables 55 through 61 . There will be eight phases throughout the development of his project which are: Research, Parts Acquisition and Initial Design, Initial Prototyping and Simulation, Final Prototyping and Development, Final Design and Development, Building and Debugging, Testing, and Additional Delays.

The VESP project was divided up as much as possible to ensure that no single team was doing a great portion of the work. Also, since the entire Senior Design course is intended to provide the student an opportunity to get hands-on experience, the collective team did not want to cheat its individual members of the learning experience that comes with being actively involved in the project by not dividing the work up evenly. The personnel for each task was either chosen by the individual themselves, or assigned to them by the group based on their strength.

Research **Week 1-6: 10/21/14 – 12/02/14**	
Research	**Group Member**
▪ Software Components: Microcontroller, FPGA	Alejandro
▪ Hardware Components: LCD screens, Sensors	Chris
▪ Hardware Components: Power, Audio System	Leonardo
▪ Hardware Components: Bluetooth, Wi-Fi	Tyler

Table 55: Research Phase

Part Acquisition and Initial Design **Week 7-9: 12/21/14 – 01/11/15**	
Design	**Group Member**
▪ Purchase all Parts	All members
▪ Software Components: Microcontroller, FPGA	Alejandro
▪ Hardware Components: LCD screens, Sensors	Chris
▪ Hardware Components: Power, Audio System	Leonardo
▪ Hardware Components: Bluetooth, Wi-Fi	Tyler

Table 56: Part Acquisition and Initial Design Phase

Initial Prototyping and Simulation **Week 10-11: 01/12/15 – 01/25/15**	
Prototyping	**Group Member**
▪ Software Setup: Microcontroller, FPGA	Alejandro
▪ Hardware Setup: LCD screens, Sensors	Chris
▪ Hardware Setup: Power, Audio System	Leonardo
▪ Hardware Setup: Bluetooth, Wi-Fi	Tyler

Table 57: Initial Prototyping and Simulation Phase

Final Prototyping and Development **Week 12-14: 01/26/15 – 02/15/15**	
Prototyping	**Group Member**
▪ Software: Microcontroller, FPGA	Alejandro
▪ Application for the Cube	Alejandro
▪ Hardware: Sensors	Chris
▪ Connection of LCDs and its components	Chris
▪ Hardware: Speakers and amplifier	Leonardo
▪ AC to DC convertor connection with battery	Leonardo
▪ Hardware: Bluetooth and Wi-Fi connections	Tyler
Developments	
▪ Hardware: Initial cube construction will all components	Tyler and Chris
▪ Hardware: Initial dock construction	Leonardo
▪ Software: Bluetooth communication	Alejandro
▪ Software: Sensor programing	Tyler

Table 58: Final Prototyping and Development Phase

Final Design and Development Week 14-16: 02/16/15 – 03/08/15	
Design	**Group Member**
▪ Software: Microcontroller, FPGA and App for Phone	Alejandro
▪ Hardware: Sensors and LCDs screens	Chris
▪ Hardware: Speakers and amplifier	Leonardo
▪ Battery Charger connection	Leonardo
▪ Hardware: Bluetooth and Wi-Fi final connections	Tyler
Developments	
▪ Hardware: Initial cube construction will all components	Tyler and Chris
▪ Hardware: Initial dock construction	Leonardo
▪ Software: Bluetooth communication	Alejandro
▪ Software: Sensor programing	Tyler

Table 59: Final FInal Design and Development Phase

Construction and Debugging Week 16-18: 03/09/15 – 03/29/15	
Construction	**Group Member**
▪ Hardware: Sensors, LCDs screens,	Chris
▪ Hardware: Speakers and Battery	Leonardo
▪ Hardware: Bluetooth and Wi-Fi	Tyler
▪ Housing for power and speaker	Leonardo
▪ Housing and casing for LDCS Screens, Sensors, Wi-Fi, and Bluetooth	Chris and Tyler
Debugging	
▪ Hardware	Tyler, Chris, Leonardo
▪ Software: Phone Application	Alejandro
▪ Software: Bluetooth communication	Alejandro
▪ Software: Sensor programing	Alejandro

Table 60: Construction and Debugging Phase

Testing Week 10-11: 03/30/15 – 04/19/15	
Testing	**Group Member**
▪ Microcontroller, FPGA and phone application	Alejandro
▪ LCD screens and Sensors	Chris
▪ Power and Audio System	Leonardo
▪ Bluetooth and Wi-Fi	Tyler

Table 61: Testing Phase

Delays Week 10-11: 04/20/15 – 04/26/15	
Delays	**Group Member**
▪ Any additional delays or documents	All members

Table 62: Delay Phase

8.2 Project Budget

As with all Senior Design projects one of the team's interests was in keeping our costs reasonable. At the start of the project period, The VESP team submitted a budget proposal to Boeing, asking for $600.00 dollars worth of funding for the VESP project. Listed in Table 63 below is the list of general hardware components and estimated costs that are included in our project.

Hardware	**Price (USD)**
Wi-Fi Module	$40.00
Bluetooth Module	$40.00
3x LCD Screens	$150.00
Touch-Enabled LCD Display	$60.00
FPGA	$179.00
Ambient Light Sensor	$10.00
Infrared Sensor	$15.00
Accelerometer	$15.00
Polymer Lithium Battery	$30.00
Speakers	$30.00
MSP430 MCU	$20.00
PCB, Wiring, and Housing	$11.00
Grand Total	$600.00

Table 63: VESP Project Budget

This is our projected budget for both semesters of the project; if the team goes over the proposed budget, the remaining expenses will be paid for out of pocket with no reimbursement. The estimates proposed were purposefully made to be a tad higher than actual in order to give the team a buffer to work with in case other factors or necessary components come up during development.

9 Conclusion

In conclusion, Senior Design I has been a very eye-opening experience for the team as a whole. It showed the team members just how much it goes into planning, designing, testing, and building a real world device. At the start of the project period, the team was unsure as to where to even start, being faced with such a daunting task to build a standalone, functional device with a clear and concise purpose. Now, at the end of Senior Design I, the team has a much better idea as to what still needs to be done, and how to go about meetings these requirements.

Throughout the semester, the VESP's features were changed, as well as its components. This was especially true for the LCD controller subsystem, which was a big unknown for a good portion of the course, and went through many revisions. Another aspect of the VESP that was difficult to solidify were its applications; there were initial, ambitious ideas for apps that after close inspection were cast aside for the sake of meeting requirements under strict deadlines. Reflecting back on the semester, the project could have been better managed at the start, and if it had been, this document would be of better quality and some of the initial ideas that were thrown out could have been included. However, that being said, after experiencing Senior Design I first hand the team now has a better view of the task at hand. The team hopes that its matured perspective will cause Senior Design II to go a lot more smoothly.

A Copyright Permissions

Hi Christopher,

Thank you for contacting TI Semiconductor Support. My name is Dylan and I will be assisting you with this inquiry.

Please refer to the below link for Texas Instruments Copyright information:

http://www.ti.com/corp/docs/legal/copyright.shtml

Of most interest to you is the section regarding educational institutions:

> ***TI further grants permission to non-profit, educational institutions (specifically K-12, universities and community colleges) to download, reproduce, display and distribute the information on these pages solely for use in the classroom. This permission is conditioned on not modifying the information, retaining all copyright notices and including on all reproduced information the following credit line: "Courtesy of Texas Instruments".*** *Please send us a note describing your use of this information under the permission granted in this paragraph. Send the note and describe the use according to the request for permission explained below.*
>
> ***Please send requests for permission, beyond the preceding two grants, to make copies of the information on these pages to:***

- *copyrightcounsel at list dot ti dot com (substitute special characters as required), or*
- *Texas Instruments Incorporated*
  Attn: Copyright Counsel
  P.O. Box 655474, MS 3999
  Dallas, Texas 75265

> ***Please identify:***
>
> - *Your company or organization name,*
> - *The information you want to copy,*
> - *The number of copies you plan on making, and*
> - *The intended use of the information.*
>
> *These permissions terminate if the use of this information under these limited permissions adversely affects the reputation of TI, as determined solely by TI. These permissions also terminate if you breach any of these terms and conditions. Upon termination you agree to destroy all copies of information downloaded from this site.*

With care,

Dylan Levan
TI Customer Support
Americas Customer Support Center
512-434-1560

Marini, Luis <luis.marini@us.panasonic.com>
Mon 11/10/2014 3:35 PM
Inbox

To: ChristopherHubbard;

Good afternoon Christopher,

Thank you for emailing us. Yes, you do have permission to use the EKMC1601111 product image in your report.
Good luck with the paper!

Best Regards,

Luis A. Marini
Web Business Partner- Interactive Marketing
Panasonic Industrial Devices Sales Company of America,
a Division of Panasonic Corporation of North America

E: luis.marini@us.panasonic.com
T: (201) 271-3519
C: (862) 368-9789
W: http://na.industrial.panasonic.com

Panasonic

From: Luis_Marini/PNA/AM/PanasonicGroup@ca.panasonic.com [mailto:Luis_Marini/PNA/AM/PanasonicGroup@ca.panasonic.com]

David Almoslino <DAlmoslino@invensense.com>
Mon 11/10/2014 6:58 PM
Inbox

To: ChristopherHubbard;

Christopher,

Yes, as long as you correctly identify the imagery as InvenSense, I do not have a problem with you using and publishing the imagery from our Corporate website.

Thank you,
Dave

David A. Almoslino | Sr. Director of Marketing and Communications | InvenSense, Inc. | M +1 408.478.5799 | T +1 408.501.2278 | E dalmoslino@invensense.com Our Vision:To Become the Leader in Sensor System on Chip http://www.invensense.com

To: ☐ Christopher Hubbard <christopherhubbard@knights.ucf.edu>;

- Flag for follow up.

Dear Christopher

Thanks for your email.

Have you ever ordered ER-TFTM070-4V2.1 display from us before?

I think it's ok to use our images and PDF files, Please go ahead.

Best Regards
Mr.Sajay Peng
Sales Engineer
EastRising Technology Co., Ltd.
Floor 6,Building A1,Zhujiang Square
Zhongxin Cheng,Longgang District,Shenzhen,China
Tel: 755 3315 1029
Mobile Phone: +86 137 6034 5037
Web: http://www.buydisplay.com

To: ☐ Christopher M Hubbard <christopherhubbard@knights.ucf.edu>;

• Flag for follow up.

totally OK

On Sun, Nov 30, 2014 at 7:31 PM, Christopher M Hubbard <christopherhubbard@knights.ucf.edu> wrote:

> contactname : Christopher M Hubbard
> email address : christopherhubbard@knights.ucf.edu
> contact us 2 section : press
> useragent string : Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.71 Safari/537.36
> message text : Hello,
>
> I am an engineering student at the University of Central Florida working on my senior design project. I intend to use the TSL2561 ambient light sensor for my project. However, for my report, I require written permission (email) to use images from adafruit.com related to this component. Please let me know if I am granted permission to include images from adafruit.com and additional provided PDFs for the TSL2561 or please direct me to the correct person to contact for this permission.
>
> Thank You!
> Christopher Hubbard
> Client IP: 97.102.62.25

References

[1] http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/CD00238572.pdf

[2] http://www.mouser.com/catalog/specsheets/Bluegiga_Technologies_BLE113_Datasheet.pdf

[3] http://www.bluetooth.com/Pages/Basics.aspx

[4] http://www.ti.com/lit/ds/symlink/cc1101.pdf

[5] http://www.ti.com/lit/ds/symlink/cc2560.pdf

[6] http://www.ti.com/lit/ds/symlink/cc2564modn.pdf

[7] http://www.ti.com/lit/ds/symlink/cc3000.pdf

[8] http://www.ti.com/lit/ds/symlink/cc3100.pdf

[9] http://www.ti.com/lit/ug/swru368/swru368.pdf

[10] http://www.ti.com/lit/ug/swru371a/swru371a.pdf

[11] http://www.ti.com/lit/ds/symlink/cc3200.pdf

[12] http://www.adafruit.com/products/1928

[13] http://www.adafruit.com/products/1933

[14] https://www.olimex.com/Products/OLinuXino/A13/A13-LCD7/open-source-hardware

[15] https://www.olimex.com/Products/OLinuXino/A13/A13-LCD7-TS/open-source-hardware

[16] http://www.buydisplay.com/default/7-lcd-display-tft-module-in-800x480-w-vga-v

[17] http://www.buydisplay.com/default/7-tft-lcd-touch-screen-display-module-800x4

[18] http://www.buydisplay.com/default/7-capacitive-touch-screen-panel

[19] http://www.ti.com/tool/msp-exp430g2?keyMatch=MSP-EXP430G2&tisearch=Search-EN

[20] http://www.ti.com/tool/ek-tm4c123gxl?keyMatch=EK-TM4C123GXL&tisearch=Search-EN

[21] http://arduino.cc/en/Main/arduinoBoardUno

[22] http://www.vishay.com/product?docid=82474

[23] http://www3.panasonic.biz/ac/e/search_num/index.jsp?c=detail&part_no=EKMC1601111

[24] http://www.ti.com/product/OPT3001?keyMatch=OPT3001&tisearch=Search-EN

[25] http://www.adafruit.com/product/439

[26] http://www.st.com/web/en/catalog/sense_power/FM89/SC1288/PF248621?s_searchtype=keyword

[27] http://www.kionix.com/accelerometers/kxcj9

[28] http://www.invensense.com/mems/gyro/mpu3050.html

[29] http://www.mouser.com/ds/2/612/Intel_BMU_PBA31308_V2%2001_PO_Rev1%202-298244.pdf

[30] http://www.wi-fiplanet.com/tutorials/article.php/1379911

[31] http://computer.howstuffworks.com/wireless-network1.htm

[32] http://www.ehow.com/how-does_5006425_zigbee-work.html

[33] http://developer.android.com/images/system-architecture.jpg

[34] http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127&tab_idx=1

[35] http://www.hardkernel.com/main/products/prdt_info.php?g_code=G138745696275&tab_idx=1

[36] http://elinux.org/Beagleboard:BeagleBoneBlack#Revision_C_.28Production_Version.29

[37] https://learn.sparkfun.com/tutorials/how-to-use-a-multimeter/continuity

[38] http://qt-project.org/doc/qt-5/qtxml-index.html

[39] http://qt-project.org/doc/qt-5/qtgui-index.html

[40] http://qt-project.org/doc/qt-5/qtmultimedia-index.html

[41] http://qt-project.org/doc/qt-5/qtbluetooth-overview.html

[42] http://www.ti.com/lit/df/tidrbu7/tidrbu7.pdf

[43] http://www.buydisplay.com/download/ic/OTA7001A.pdf

[44] http://www.ti.com/lit/ds/symlink/tlk10022.pdf

[45] http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/DM00031386.pdf

[46] http://en.wikipedia.org/wiki/DisplayPort

[47] https://developer.motorolasolutions.com/docs/DOC-1878