Visually Entertaining Smart Prism (V.E.S.P)

Alejandro Torroella, Leonardo Achutegui, Chris Hubbard, Tyler Drack

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Visually Entertaining Smart Prism, or VESP for short, is intended to be a luxury entertainment and utility device that works in conjunction with readily available internet services and smartphone devices. The project was chosen for its broad range of hardware and software specifications, demonstrating the use of LCD display drivers, sensors, ARM, Arduino, Linux, Wi-Fi and Bluetooth wireless communication, battery powering, charge controllers, power distribution, and audio processing. The goal of the project was to have a working prototype with as much functionality as the team could implement during the period of Senior Design, along with the potential of including more features in the future.**

***Index Terms* — VESP, LCD Display, ARM, Arduino, HDMI, RGB, Wi-Fi, Bluetooth, Battery power, Smart device, Sensors, OpenGL, Embedded Linux, Qt.**

I. Introduction

The world of technology is progressing in such a way that gizmos and gadgets are popping up everywhere for a variety of purposes, all to help mankind's daily routine be more efficient and enjoyable. Most notable of these gadgets are tablets, smartphones, gaming consoles, and smart TVs. This is the motivation behind the idea for the Visually Entertaining Smart Prism, or VESP. The goal was to make a fun, autonomous notification center that could easily find its place on any desk or table, in either a home or workplace environment. This technology would help assist the user stay notified of upcoming calendar events, current weather conditions, or simply the time. It would also provide entertainment for the user through 3D animation imagery and music playback from the user's phone. In addition, the ability to operate the device portably was desired, and so the VESP was split into two portions, the VESP device, and the dock. The following figure (Fig. 1) illustrates this feature.

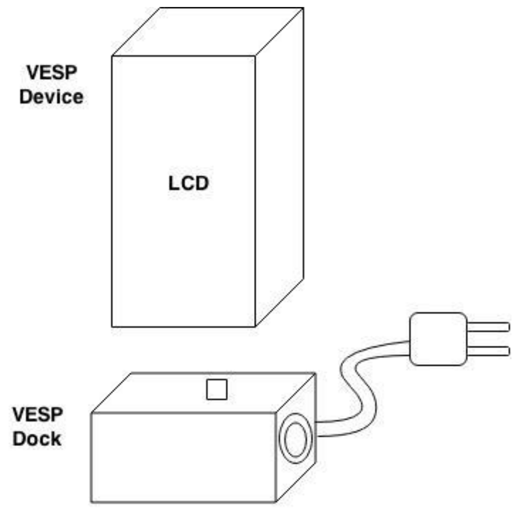


Fig. 1. VESP device and dock high-level diagram

Working on this project offered the chance for the project engineers to apply their undergraduate coursework and get experience in areas such as PCB design, wireless communications, power distribution design, computer graphics, sensor, and Linux/ARM development, just to name a few.. It also helped to develop other professionally applicable skills such as technical research, data analysis, team work, project management, and working under strict deadlines.

II. System Components

The VESP is composed of various hardware components, purchased or designed, that work in unison to give the VESP its essential functions. Each of the components will be briefly described in this section.

A. Main Processing Unit

At the heart of the VESP lies the Main Processing Unit (MPU). The VESP MPU is what runs all of the software for the project. In addition, the MPU takes in incoming orientation data from the MCU, as well as touch data from the touchscreen controller. It was anticipated that the VESP would have to be able to handle all of this processing, while also minimizing the amount of lag that the user might experience. With that in mind, the ODROID XU3 from HardKernel was chosen for the MPU. The device was within our size constraints and is a very high performance ARM development platform that could more than satisfy all of our current and potential future requirements.

B. LCD Displays

The LCD displays are an essential part to the VESP's aesthetic and form factor. For that reason, the selection of which LCD displays to use for the project was crucial. Not only did the screens have to meet the performance specifications, special consideration had to be given on the dimensions as well. This is due to the project requirement that all of the internal components for the device (MCU, MPU, PCBs, battery, data/power cables, etc.) had to fit within the form factor. Too big of displays would have made the final result too bulky to carry by hand, while too small of displays would mean even stricter size constraints.

C. LCD Controller

The LCD controller is responsible for taking in the HDMI output from the MPU and distribute it to the four LCD displays. In addition, the LCD controller is responsible for handling the raw touchscreen data coming from the capacitive touch screen and processes it to be then used by the MPU.

D. Microcontroller

The VESP's MCU is used to control and process the sensor subsystem. Having an MCU added additional hardware connection options and helped take some of the processing load off the MPU. The MCU had to be connected to the motion sensor, the gyroscope, the ambient light sensor, the MPU, and the power distribution PCB.

E. Sensors

Various sensors were incorporated into the VESP device in order to give it its "Smart" acronym, these included a motion sensor, an ambient light sensor, and a gyroscope.

The motion sensor's purpose for the project was to assist in power saving. It would sense when a user was nearby or present, so that the VESP would "know" whether to auto lock the device software and power off the LCD display backlights, which on any portable device is the main perpetrator for battery power consumption. This meant that the sensor's range should encompass the length of a typical household room.

The purpose of the light sensor is to measure the surrounding lighting and detect changes, giving this information to the MCU. That data would then be used to adjust the brightness of the LCDs accordingly. This is to further help make the VESP as power efficient as possible which is especially important when it is off the power dock.

The purpose of the gyroscope will be to sense when the VESP is off the power dock and to determine the orientation of the VESP as the user moves it around in their hands. The data it gathers will be processed by the MCU which will then send commands to the MPU to change the orientation of the displays accordingly.

F. Wi-Fi

Every relevant electronic device on the market has some sort of connection to the internet, the VESP is no different. For the purpose of getting Wi-Fi on the VESP, a proprietary Wi-Fi module was purchased from HardKernel. The module is sleek and small enough to fit within the enclosure without issue, can be powered via one of the MPU's USB ports and provides more than sufficient bandwidth for the purposes of the project.

G. Power Distribution

In order to make the VESP device portable, the use of a battery and battery charge controller was needed for the project. For the battery, a lithium-ion polymer battery was used. This type of battery is lightweight, small and can output sufficient power to the drive all of the internal components. For the battery charge controller, the LTC4020 from Linear Technology was used. The charge controller charges the battery at a constant-voltage/constant-current, and is designed to withstand a maximum charge of 2 Amps. Also, the LTC4020 is able to pull the current necessary to power all the devices at maximum usage.

H. Dock

The dock for the VESP houses all of the components that for one reason or another, weren't necessary to include inside the VESP device. This included audio speakers, an audio amplifier, a Bluetooth audio receiver, and an AC-DC converter circuit. For the components pertaining to audio, a TA2024 Class-D audio amplifier was used. This would take in audio signals coming from the Bluetooth audio receiver and output them to the audio speakers. For the purpose the project, two 2"x3" extended range speakers were used. For the AC-DC converter, an AC-DC transformer was used in conjunction with a 12V regulator, which was used to power both the audio amplifier, as well as the battery charge controller within the VESP device.

III. SYSTEM OVERVIEW

This section will briefly describe the organization of the hardware, and software components, as well as the go into more detail on the interactions between the hardware, and software components. This is will give an overview of how the VESP works, before a low-level description is given.

A. Hardware

Figure 2 in the next page shows the hardware block diagram for the VESP device. Each of the blocks represent

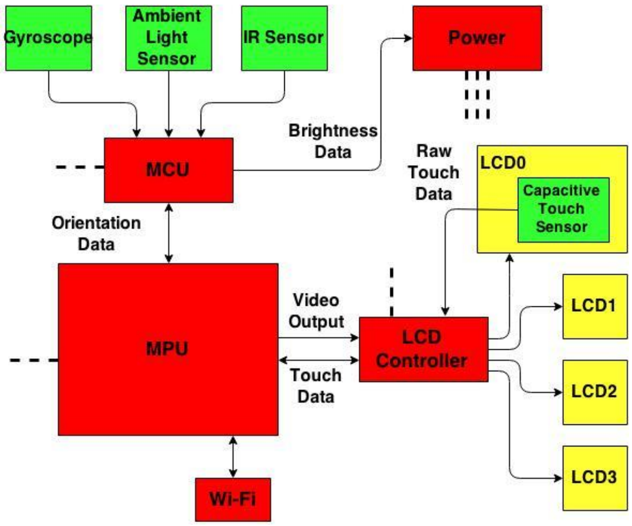


Fig. 2. VESP Hardware Block Diagram

a major component in the systems; the arrows connecting the blocks represent an interaction between the components. Starting from the upper-left side, the MCU is connected to the gyroscope, ambient light, and motion sensors and handles all of the raw sensor data. The orientation and motion data is passed to the MPU to be used as input for the software. The MCU is also connected to the Power Distribution board in order to control the backlight power going to the LCD displays. The MPU is connected to the all of the other major hardware components. The MPU sends its video data to the LCD controller, which then distributes it to the four LCD displays. The MPU is connected to the internet via the connected Wi-Fi module. Finally, the LCD controller sends its processed touch data to the MPU.

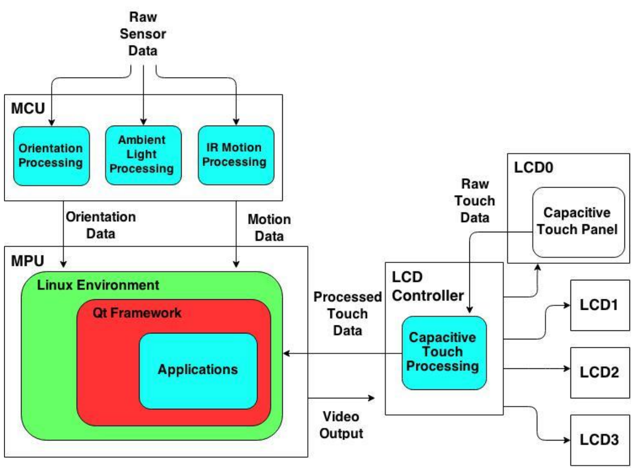


Fig. 3. VESP Software Block Diagram

B. Software

Figure 3 below shows the software block diagram for the VESP. Each of the colored blocks represent major software components, while arrows represent relevant interactions between them. Starting from the upper-left, the MCU takes in the raw sensor data coming into the various sensors connected to it, and processes the data to an expected format for the rest of the VESP to understand. The orientation data and motion data is sent to the MPU which is accepted by the Linux environment running with it. The Linux environment also accepts the processed touch data coming from the LCD controller. The accepted data going into the Linux environment is then used by the Qt Framework, which drives all of the VESP's applications.

B. Dock

Figure 4 below shows the hardware block diagram describing the dock for the VESP. From the diagram, the power supply is powering both the audio amplifier and the Bluetooth receiver. The audio amplifier and the Bluetooth are connected to provide audio data transfer. Finally, the amplifier is connected to the audio speakers to output sound.

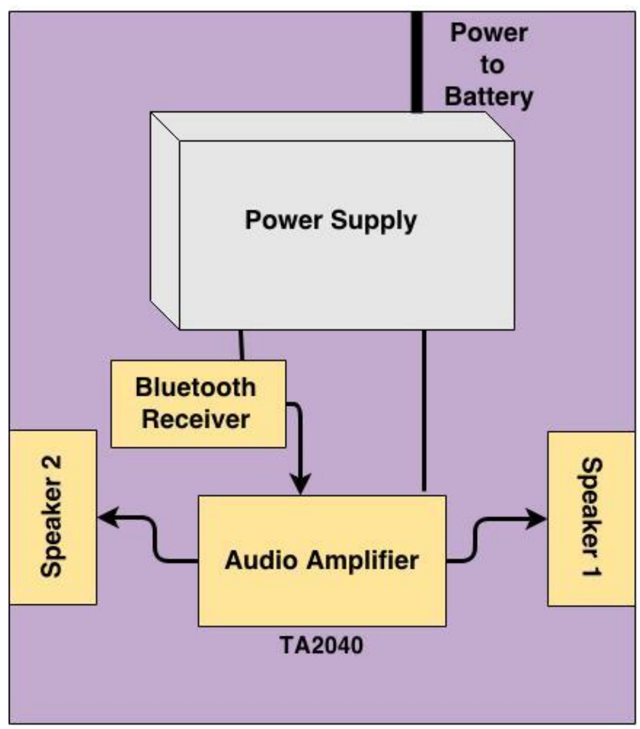


Fig. 4. Dock Hardware Block Diagram

IV. Hardware Detail

This section will give a lower-level view of the hardware components previously mentioned.

A. MPU

As previously stated, the MPU chosen for the project was the ODROID XU3 by HardKernel. As you can see from the Table 1 below, the XU3 has a fast, dual quad-core CPU ARM architecture, giving the project more than enough processing power to work with when developing applications, without needing to take to prioritize coding efficiency. Along with its speedy processor, the XU3 has 2GB of DDR3 RAM, giving it the VESP the ability to run more than one applications in the background. Another major feature of the XU3 is its graphics chip, the Mali-T628 MP6, allowing for the development of 3D hardware accelerated graphics. Other useful features of the XU3 are its 30-pins for I/O (GPIO/UART/SPI/I2C), four USB 2.0 ports, and HDMI output.

Model	ODROID-XU3 Lite
CPU	Quad-core Cortex-A15/ Quad-core Cortex-A7 Combo
GPU	Mali-T628
RAM	2GB DDR3
Memory	16GB
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
Video Output	Micro-HDMI 1.4a
API Support	OpenGL ES 1.1, 2.0 and 3.0
Power	5V @ 4 A

Table. 1. MPU Specification and Features

B. LCD Displays

The LCD displays chosen for the project were the ER_TFT070_4 from BuyDisplay.com. The displays measure ~7" diagonally, have a 800x480 resolution, and a color depth of 24 bits. The 7" option was ultimately chosen to be the ideal size for the project (compared to similarly priced and capable 5" displays), providing plenty of space for internal components, without making the VESP excessively bulky. The voltage required to power each screen was 3.3V is similar to many of the other components in the project, which assisted in simplifying the power supply design. In addition, the ER_TFT070_4 came with an optional configuration which included a mounted capacitive touch screen, thereby illuminating the need to research and purchase a separate sensor and having to mount the sensor on the display manually.

C. MCU

The selection for the MCU was the ATmega328 on the Arduino Uno 3. This option was a great fit for the project, Table 2 below shows the MCU's specifications. It offers a sufficiently fast clock, enough to handle all of the sensors and transmitting data to the MPU. Plenty of digital I/O pins that include UART, SPI, I2C, and PWM. The PWM is particularly important as it will be the way the MCU will control the backlight brightness of the displays. A USB connection is available to both power the device and transmit data to the MPU. This board operates on a voltage range of 7-12V which was within the scope of the VESP's power range. Additionally, two of the sensors could be powered directly from the board's 3.3V pin and 5V pin, which helped to simplify the power supply design. Software-wise, there were many different libraries and examples to reference online, which helped to ease the development of the project's sensor software.

MCU	ATmega328
Clock Speed	16 MHz
Input Voltage	7-12V
Operating Voltage	5V
Digital I/O	14
Analog I/O	6
Flash Memory	32KB
I/O Voltage	3.3V & 5V

Table. 2. MCU Specification and Features

D. Motion Sensor

The selection for the motion sensor was a PIR sensor found on Adafruit.com. This sensor could sense up to 20 feet away, was a reasonable size, and runs on a voltage range of 5V-16V, although the project runs it on 5V as that was more compatible with the power supply design. The sensors design also offered screw-holes, which helped with mounting it on the top of the VESP.

E. Ambient Light Sensor

The light sensor selected for the project was the TSL2561 Digital Luminosity Sensor. This sensor had a vast detection range of 0.1 to 40,000 lux with the option to measure IR, visible, or full-spectrum light. It runs on a supply voltage of 3.3V and communicated digitally via I2C which made it a great match for the MCU.

F. Gyroscope

The gyroscope selected for the project was the MPU-3050 Triple Axis Gyroscope from Invensense.com.

Additionally, this gyroscope will be used on the OSEPP Gyroscope Sensor Module. This component runs on a 5V power supply so it was easily integrated into the power supply design. The gyroscope interfaces with the MCU through I2C. The issue that arose from both the light sensor and the gyroscope requiring the MCU's single I2C connection was overcome through the use of the OSEPP module, as it provided a pass-through connector, allowing these two sensors to be chained together without interrupting each other's data flow.

G. LCD Controller

The LCD controller is one of the hardware components that was designed for the project. The LCD controller satisfies two functions: distribute the video signal to the four LCD panels, process incoming touch data from the capacitive touch panel. For the first function, the LCD controller first takes the HDMI output from the MPU through its female HDMI connector and sends it to the HDMI decoder, which then converts the TMDS signals to 24-bit parallel RGB signals that the LCD displays can interpret. The HDMI decoder chosen for the project was the TI TFP401A, it is able to decode video resolutions up to 1080p @ 60 Hz, plenty for the purpose of the project.

For the second function, the LCD controller includes the FocalTech FT5406 capacitive touch panel controller, which takes in the raw touch data from the capacitive touch panel and processes it to then send to the MPU for use. The FT5406 uses the I2C protocol for transmitting its touch data to the MPU. Figure 5 shows the PCB board layout for the LCD controller.

H. Charge Controller

A charge controller was necessary to safely charge the lithium-ion polymer battery. The LTC4020 was configured to provide a constant-current/constant-voltage charge profile to avoid any damages on the battery. It includes a 4-switch buck/boost dc/dc power converter to optimize the battery charging and allowed us to accept input voltages up to 55V and produce voltages that are lower, higher or the same as the input voltage. This was essential when the device was plugged to the dock because it allowed us to charge the battery and at the same time be able to power the components in the device. Also the charger was configured with a timer to sense the end of cycle of battery charging and stop charging after 22 minutes while the device was still connected to a power supply. The LTC4020 is able to manage the power distribution between the input voltage, the backup battery, and the output in response to the load variations, battery charge requirements and input power limitations.

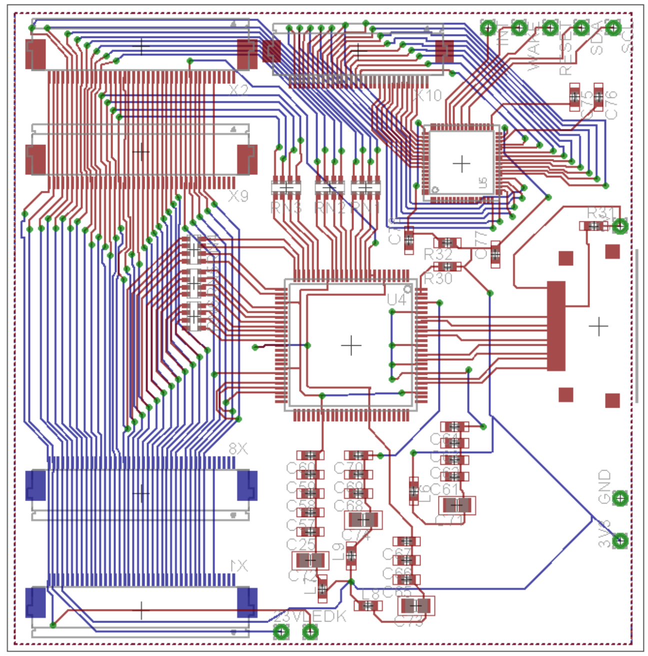


Fig. 5. PCB layout for the LCD Controller.

Aside from the LTC4020, there were other IC components that allowed us to power all our small devices in VEST. The TPS54618 step down voltage device, outputted a voltage of 3.3V with a max current of 6 amps. The LMR62421 is a step up voltage regulator that powered the devices that needed a 12 volts input like the motion sensor and the Arduino. The LM3478 is a switching regulator to power the MPU and the FAN5333A is high current serial LED driver that was used to power the LCD backlight and adjust its frequency.

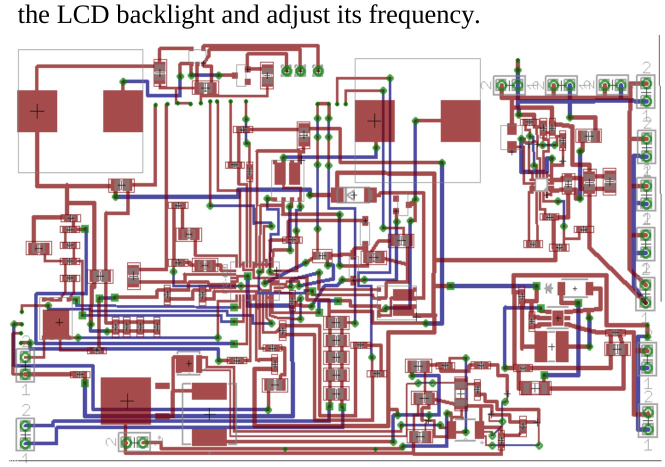


Fig. 6. PCB layout for the charge controller

V. Software Detail

This section will describe in detail the software running in the VESP. This will include the architecture, the overall GUI, and each of the applications available in the VESP.

A. Software Architecture

For developing the software for the VESP project, the Qt Creator IDE was used. The software all was written using C++, for the main logic, in conjunction with QML (Qt Meta Language), for the user interface. QML is a declarative language developed by the Qt Company, and allows for developing fast, easy, and beautiful user interfaces. In addition, other APIs and SDKs were used in creating the VESP software. The OpenWeather API is used to fetch weather data based on the positioning data obtained from Wi-Fi, to then display to the user. The Google Calendar API is used to fetch calendar event data from the user's Google account to remind the user of any events going on that day. Finally, the Mali OpenGL ES SDK, in conjunction with the OpenGL ES 2.0 API was used is used to render visually appealing, hardware accelerated 3D graphics to the displays.

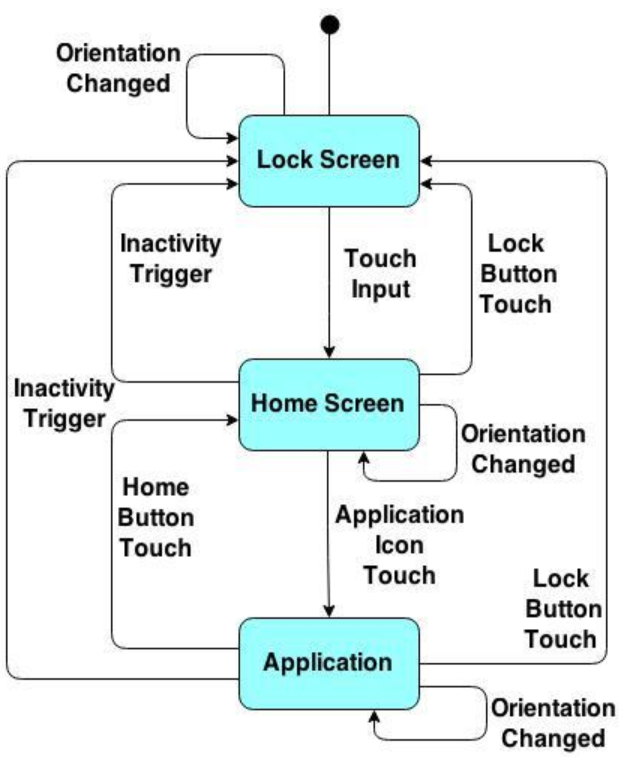


Fig. 7. VESP GUI State Diagram

B. Graphical User Interface

The GUI of the VESP is intended to be both visually appealing, simple and easy to use. Figure 7 on the left shows the state diagram for the general UI flow. The GUI is split up between two major elements: the Lock screen and the Home screen. The Lock screen is intended to be the first thing the user sees when starting the software; showing the current time in analog format, as well as the current weather of the VESP's current location. A simple touch of the screen will transition the GUI to the Home screen. The Home screen's purpose is to show all of the VESP's applications, and is where all of the VESP's current (and future) applications are (will be) launched from. The Home screen also has a lock button on the bottom which will transition the UI back to the Lock screen. A touch on any of the application icons will launch the respective application. Every application of the VESP has a home button on the bottom of the screen that will transition the UI back to the Home screen. The UI will also transition back to the Lock screen either after a set amount of time has passed without any user input, or if the device detects that no one is present, essentially "locking" the device. In addition, every state of the UI (lock, home, and application) have an portrait and landscape layout, a sufficient change in orientation of the device will transition the device to/from either layout.

C. Applications

The applications on the VESP are the Animation apps, the calendar app, and the weather app. Of the three, the Animation is the only app which has to be launched in order to use. The other two apps are passive, run in the background, and require no user interaction to use.

The Animation applications all display a particular 3D graphic on the screen. Some applications take in touch input from the user and change accordingly, while others simply loop over time. Their purpose is purely aesthetic and give the VESP its first two letters of the acronym (Visually Entertaining). The weather app is a passive app which runs on the Lock screen, displaying the current temperature and weather condition of the location obtained from the positioning data coming from the Wi-Fi connection. Finally, the calendar is the other passive app that runs on the Lock screen, it displays calendar event reminders to the user. Calendar event reminders will be stored in a queue data structure, when a user has dismissed the currently displayed reminder (via touch), the next reminder will be displayed.

VI. HOUSING

The housing for the VESP is split into two separate components, one for the VESP device and the other for the dock. For both of the housings, a clear plastic acrylic was used to hold the components together. For the VESP device housing, exact measurements were made on the LCD displays to make the windows of the housing as tight as fit as possible. The windows are joined using a combination of a jigsaw pattern intersections and acrylic glue. In addition, for the sake of making sure that all of the internal components fit within the enclosure, the housing windows were made to be a couple of inches larger than initially planned.

For the dock, the housing was made to be wider than the VESP device, with a significant depression on the top the size of the device in order to ensure a secure fit between the dock and the device. On the opposite sides of the dock, a hole was made in order to include the speakers. Other holes were made for the wall outlet connector, and the battery charging connector as well. Figure 7 below shows a prototype of the housing design with the four LCD displays placed in the windows.

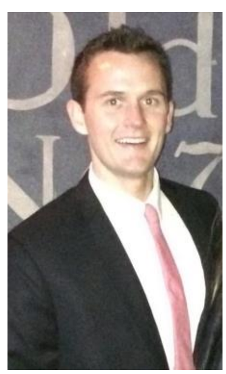
Fig. 7. VESP device housing prototype

VII. CONCLUSION

The world is filled with a plethora of entertainment and utilitarian electronic devices. Every day, more and more of such devices arrive to the market. The VESP is an attempt to present a novel idea for a new device, with an interesting, (albeit unorthodox) form factor. The VESP prototype described in this documentation is only the first step. The internal hardware contained within the device is powerful enough to handle new and exciting features, and with a powerful framework such as the one provided by Qt, the possibilities are near infinite. With that said, in the future, the team would like to improve on the design of the VESP, as well improve its current functionalities. In conclusion, the project gave the team invaluable learning experience in the many disciplines of Electrical and Computer Engineering.

ENGINEERS

Tyler Drack will graduate from UCF in May 2015 with a B.S. in Electrical Engineering. Tyler has accepted an offer from Harris Corporation in Palm Bay, Florida as an electrical engineer within their Microelectronics Department. In his free time Tyler enjoys playing video games, writing, jigsaw puzzles and reading and watching several topics including meteorology, astronomy, geology and in-depth articles about mechanics in video games. In the future Tyler would like to live in rural New Hampshire, preferably near the town of New London.



Christopher Hubbard will graduate from UCF in May 2015 with a B.S. in Electrical Engineering. Chris has accepted an offer for a position at Northrop Grumman in Melbourne, Florida as a Systems Integration and Test Engineer. Throughout his undergraduate career, some of his favorite courses were Electronics, HDL in Digital System Design, DSP Fundamentals, Intro to Modern & Robust Control, and Digital Systems. Outside academia, he enjoyed UCF’s gymnasium, leisure pool, football game days, and skateboarding around campus. He hopes to become a great asset at Northrop Grumman, and spend his days surfing, working, and enjoying life by the beach.

Leonardo Achutegui will be graduating in May 2015 with a B.S. in Electrical Engineering. Leonardo's interests for the future are Power Systems Design and Project Management. He aspires to work in a Power Plant or Utility Company and at the same time work in his Masters degree in Power Systems. During his undergraduate time in UCF, Leonardo acquired many skills in electronics and design that has led him to choose the path of power systems. Aside from his engineering career, Leonardo enjoys outdoor activities and traveling.

Alejandro Torroella will graduate from UCF in May 2015 with a B.S. in Computer Engineering. Alejandro has accepted an offer from Harris Corporation in Melbourne, Florida as a software engineer in the Image Processing team. During his time at UCF, Alejandro has done computer vision research and co-authored a paper published in the ECCV 2014. His other topics of interest include computer graphics, HCI, computer architecture, and embedded systems. In the future, he looks forward to getting his Ph. D., while doing research in any of his fields of interest

Acknowledgement

The authors wish to acknowledge the assistance and support of the UCF EECS faculty, most notably, the TI Innovation Lab staff, Nathan Bodnar, David Douglas, and Dr. Samuel Richie. Without their assistance and guidance, the project team would not have accomplished their goals. Additionally, we would like to thank the members of the faculty review committee for taking the time out of their schedules to participate in our project's presentation.