Pet Connect

Graham Goerg, Joy Weaver, Ryan Flynn, and Michael Choi

University of Central Florida, Department of Computer and Electrical Engineering, Orlando, Florida, 32816

***Abstract* — Pet Connect is a home system that allows pet owners to let their pet outside when they are away from their house. It consists of an Android mobile application, a rod-like linear actuator, and a multiple sensors unit. The multiple sensors recognize when the pet is at the door waiting to go outside and will then alert the owner through the mobile app. The owner then has the ability to open the door from the app resulting in the linear actuator pulling the door open. This paper presents the design methods we used to create the Pet Connect system.**

***Index Terms* — Microcontrollers, home automation, mobile applications, wireless communication, databases.**

I. Introduction

Pet Connect gives pet owners a convenient way to let their pets out when they are not at home. It allows people to experience the joy of having a pet even if they are working all day without worrying about the pet "going" in the house and being restless. According to the American Pet Products Associations (APPA), there are about 8.5 million families that own a pet. Many American pet family owners had to leave the home for work and were forced to decide either to find someone to look after their beloved pet or send them to pet care. Hiring someone to look after one's pet tends to be expensive and undesired as it requires inviting a stranger into your household or commuting to the location of the pet care. Especially during the COVID-19 pandemic, families will be leaning towards a user-friendly, self-sufficient, and affordable option to take care of the pet's needs.

Our prototype is a remotely operated pet door that will allow the owner to open their standard sliding glass door to let their pet out even when they are not at home. When the pet gets close to the door an alert will be sent to the owner's phone which will then give them the option to unlock and open the door. The pet will be wearing a small device that can be attached to a collar, which allows the sensors on the doorway to recognize the pet is close. For security reasons, a camera will show the user their doorway to make sure that their pet is trying to get out and an intruder is not trying to get in. Once the pet has come back inside the door will be closed and locked by the owner. Our design also includes a speaker that can be used by the owner to either call the pet inside or call the pet to the door if the owner desires to let the pet out without the pet initiating the system. With an easy to use mobile app, the user would also be able to open and close their door, look at the live video feed, play a message on the speaker, or change the current mode that they have their system in.

II. System Overview

The project design includes a housing that sits inside of a sliding glass door, in the track where the movable door would slide. The housing will have an extendable and retractable arm (linear actuator) that will connect to the moveable glass door. It will also hold the single board computer and PCBs. The outside of the housing will be the user interface display, consisting of an LCD screen as well as LEDs, all controlled by an MCU. The housing itself will assist in acting as a door jam or lock for the sliding door when the actuator arm is fully extended. Externally connected from the housing will be a webcam and speaker for audio and visual security measures that will be connected to the housing via USB cabling. The webcam is to be installed above the housing angled towards the sliding door opening. There will also be external sensors, an RFID and an ultrasonic sensor, that will extend from the housing using wiring. These sensors are to be attached to the sliding door where sensing would be optimized. The last piece of hardware is a wearable, the RFID tag, which will be attached to the pets collar.

The system has three modes of operation that the user can change with the mobile application. The closed mode will keep the door shut and will not trigger any notification if the pet is at the door. The home mode will open when the pet goes to the door without alerting the pet owner. Finally the away mode alerts the owner when their pet is attempting to go outside. The door will then open with the user allowing the door to open through the app. The flowchart below shows an overview of how the system will be used by the owner.

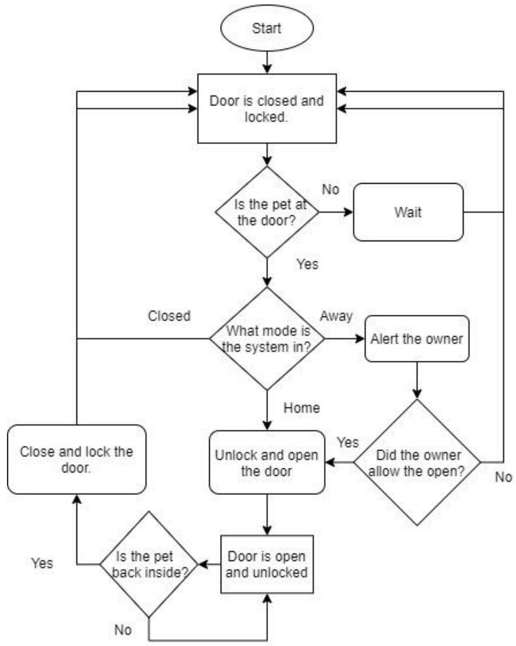


Figure1. System Overview Flowchart

III. Hardware

Our design is based around many different pieces all communicating and working together to perform the task of opening and closing a door. This section will lay out all relevant technical information related to each hardware piece or portion of our project.

A. Single Board Computer

The Raspberry Pi 4B with 4GB RAM controls all operations and handles each peripheral of the design. The Pi was chosen for its versatility and ability to handle audio/visual constraints. The design uses two of the four USB connections and close to half of the forty GPIO pins. The Pi has its own Linux based OS variant, which allows for the ease of use and flexibility of programming languages. Th Pi also comes with a WiFi adapter built into it, which is utilized for our mobile application integration.

B. Microcontroller

The task of controlling all user interface operations is to be handled by a MSP430G2553 microcontroller. This MCU has less than 100mA power usage and plenty of GPIO outputs to communicate with the LCD screen, associated LEDs, and the Raspberry Pi. The processing power this MCU has is enough to implement our design but a multicore unit would be needed to implement a more complex UI.

C. RFID

The RC522 is a 13.56MHz RFID module that supports I2C, SPI and UART. It has an operating voltage of 3.3 volts at about 20mA (10uA when in powered down mode) and a max data transfer rate of 10MBps. The RC522 has a read range of 5cm. This module comes with a single RFID card that has 1KB of memory. The RC522 acts as both a reader and writer. The design comes with the associated RFID tag number(s) hard coded into our program to avoid having the user having to enter them.

D. Ultrasonic Sensor

The HC-SR04 has a theoretical measuring distance of up to around 450cm, with a practical measuring distance around 100cm, and an accuracy close to 3mm. The HC-SR04 also measures an area within a 15° angle from sensor facing. It operates at 5 volts using less than 15mA of current and at a frequency of 40Hz. The ultrasonic sensor is programmed to alert the system when an entity is detected within 25 cm.

E. Linear Actuator

The PA-14-18-50 Mini Linear Actuator from Progressive Automations is a DC brush motor with a power requirement of 60 W and a maximum of 50 lbs of force at full load. The linear actuator has 18 inch stroke and can extend at 0.83 inches per sec at 50 lb load. The end-to-end dimension with the stroke fully retracted is 22.13 inches and 40.13 inches when the stroke is completely extended. The linear actuator provides linear motion in contact to the standard DC stepper motor.

F. Home User Interface

The user interface consists of an LCD screen and six LEDs. The Sunfounder LCD 1602 [SUNF] module consists of a blue background with white display text. The option that we are considering has 16x2 possible characters available for us to show all messages and operations for the whole system. This LCD module runs on a working voltage of 5 V and we have the option to adjust the brightness with a 50k potentiometer. Each of the LEDs in the user interface has a max current of 20 mA and 1.8 – 2.0 forward voltage for the red, yellow, and green colors, while the blue has a forward voltage of 3.0 – 3.2 V. The multiple LED colors give the UI design a better look and easier readability. The figure below illustrates the UI implementation.

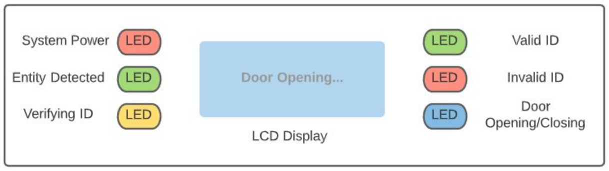


Figure 2. Home User Interface

G. Audio / Visual

The Logitech C270 is a USB widescreen HD webcam that comes with an internal microphone. The camera has a max resolution of 720p at 30 frames per second, a fixed focus lens, automatic light correction, and a field of view at around 60°. The built in microphone is noise reducing and can pick up sound up to 3 meters. The C270 also comes with a 5 foot USB 2.0 cable. The webcam will be installed above the housing near the corner of the non-movable glass door.The USB Mini Speaker is connected via USB 2.0 and consists of two 0.5W stereo speakers. It also comes with a 47 inch cable. The speaker has an operating voltage of 5v and can be installed on the housing or nearby where the user desires.

H. DC Motor Driver

The Drok dual DC motor driver module utilizes an electrical circuit with four power mosfets in a H-bridge configuration to switch the polarity of the voltage applied to the linear actuator. The Drok motor module can drive up to 7A current each H-bridge circuit and input voltage between 6.5 to 27 V. The Drok DC motor driver can support a maximum of 160 W. The Raspberry Pi can communicate with the motor driver and signal relay to control the linear actuator. With a pulse-width modulation signal ranging from 0 to 10 kHz, the Raspberry Pi may control the speed of the linear actuator by controlling the duty cycle through the motor driver.

I. Signal Relay

In combination with the motor driver, the signal relay can control the linear actuator through the Raspberry Pi's general purpose interface pin. The signal relay is a double pole double throw relay that has a minimum operating voltage of 2.4 V and has contacts capable of uphold a maximum 2 A current load.

J. 96 W Power Supply Adapter

The task to convert the AC Wall power to DC voltage is handled by the off-the-shelf power adapter. This power adapter can support input ranging from 100 to 240 V alternating current at 50 to 60 Hz in order to output a constant DC voltage of 12 V at 8 amps. The following power adapter provides over-voltage, over-current, high thermal, and short-circuit protection. The power supply delivers the necessary power for the project.

IV. Software

The software design was very important to the project because we needed to meet certain criteria to have it be deemed as successful. We needed to design a mobile application that would allow the user to control all of the necessary components while also being easy to use. With the data from both the Raspberry Pi and the Android app being passed back and forth, we used Google Firebase to store all of the necessary data. During the early stages of the project we decided to dedicate a portion of the research and the design to how the different components communicate and stay up to date with each other.

A. Android Mobile Application

The mobile app is used as our control panel which users will interact to send commands to update the system. We decided to build a native Android application to run on an Android device. Native Android apps can take full advantage of software and the OS's features. Moreover, native apps are faster in execution since we prefer to use real-time databases which ultimately result in a better user experience. Android Studio is the platform chosen to develop the mobile app. Java is the language that commonly uses and supports Android Studio. We used the builtin emulators and an actual Android phone to test the app. A typical Android app contains multiple components including activities, fragments, services, content providers, and broadcast receivers. In addition, Android Studio has built-in support for integrating Google Cloud services and Firebase which is the database that we chose to implement in our software development.

B. Database

We implemented Google Firebase Realtime Database as our Database Management System [2]. Although the software team was not familiar with this going into the project, Firebase gave us the ability to keep all of the necessary devices updated with the most recent information within seconds. Google Firebase is non-relational and uses a JSON-like tree structure to organize data nodes. We organized our data into three branches that can be seen in the image below.

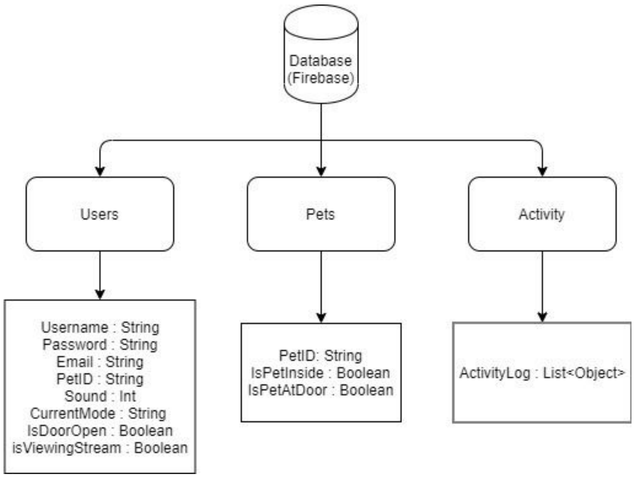


Figure 3. Database Structure Diagram

Since each branch needs a key value, we selected the username to be the key for the 'Users' and 'Activity' branch while the petid is the key for the 'Pets' branch.

We utilized boolean flags that are updated by the Raspberry Pi and the app in order to update the system. Event listeners are used from the Google Firebase Application Program Interface (API) and implemented in Python and Java. These event listeners are set to the nodes of the boolean flags so the program can react correctly based on the change that was made. When writing to the database from the app we had trouble with writing different data types to Firebase. We solved this problem by writing everything as a String with "0" and "1" being used for the boolean values. The reason for the different data types in figure 3 is because the diagram is showing how we interpret the data for the system.

For example, when the Raspberry Pi recognizes the pet is at the door it will set IsPetAtDoor to "1" representing true and the pet being at the door. The event listener from the Android app will see this data change and will alert the user. On the other hand, when the user opens the door from the app, IsDoorOpen will be set to "1" representing true and the door being open. The python event listener will then tell the linear actuator to open the door from the program on the Raspberry Pi.

C. Wireless Communications

In this section we look at the different techniques we used to transfer data between the wireless components of Pet Connect. The diagram below illustrates each of the methods used between each of the components.

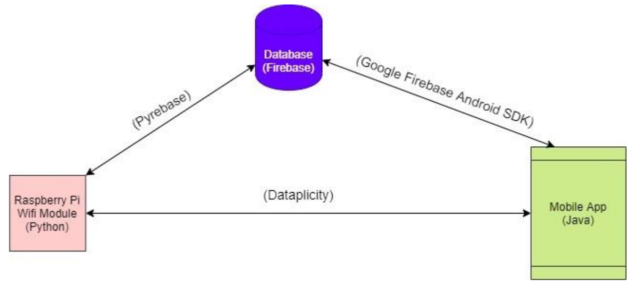


Figure 4. Wireless Communication methods between each of the components.

Looking at the Android app and Firebase communication, the Google Firebase Android SDK is used to make all of these data transfers. This is implemented in Java and has all of the methods that we need to read, write and set event listeners in the database.

On the other side of the Firebase communication we have the Raspberry Pi. Google Firebase does not currently have a native API for Python. However they do endorse a few third party libraries that act as a Python wrapper around the Firebase API. We implemented Pyrebase by James Childs-Maidment. It is very similar to the Android SDK and again provided us with all of the necessary functions to transfer the data correctly with a slightly different terminology.

The final communication path handles the mobile app displaying the live video stream from the Logitech Webcam. We used Dataplicity to achieve this which provides numerous software tools such as Hawkeye and Wormhole to access a Raspberry Pi remotely [1]. Hawkeye locally hosts the video stream from the Raspberry Pi while Wormhole generates a link to the webpage that will display the videostream. This link can then be accessed remotely through an Android WebView that the software team has programmed as part of the Android app user interface. This is a secure option due to the fact that the URL is specific to each user and is hosted at a secure web address. An alternative method was to Port Forward the Raspberry Pi's IP address but that would open up the user's home network to the public.

D. Application Function

When users first launch the application, they have the ability to create an account to login and use the application. If the user has already made an account, then they can login to start the application. When the user first opens the app they see the control screen and have the option to navigate to the other two screens. All three screens require initial reads from the database to display the appropriate data. The Profile screen also gives the users the ability to edit their information which is then updated in the database. From the control screen the user can interact with the speaker and camera peripherals, change their current mode, open or close the door directly, or simply just view the data. Figure 5 shows a flowchart for each UI and its function.

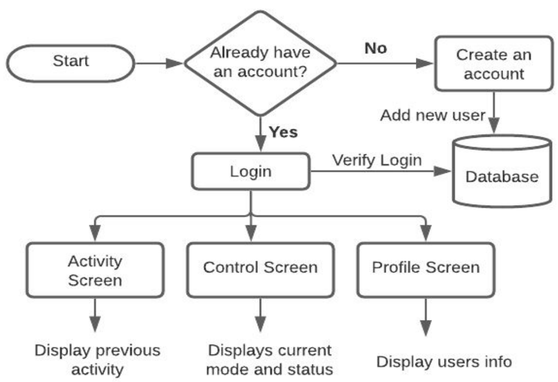


Figure 5. User Interface flowchart and purpose

As mentioned in the System Overview section, the application consists of three different modes. Closed mode is used to keep the door closed at all times. Home mode sets the door to open automatically whenever the RFID sensor detects a pet approaching the door. Away mode sends a notification to users when the RFID triggers and updates the database. Users can access live video and open or close the door from mobile user interfaces. Figure 4 shown system workflow from the user's action.

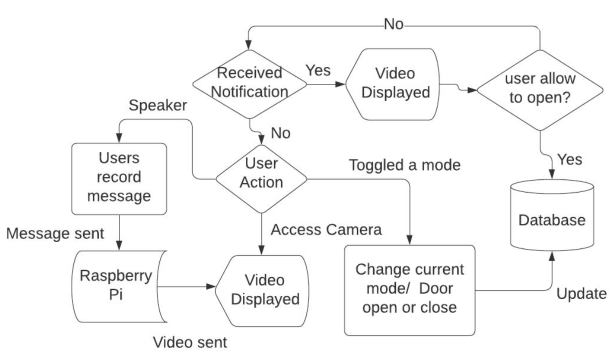


Figure 6. Mobile Application Diagram.

The other path for interacting with the mobile app would be when the notification is received. When opened, the user would be shown the live video from the camera and be asked if they want to allow the door to open, letting their pet outside. If they do not allow it then the flow of the program just goes back to the 'Notification Received?' option. If they do allow it then the database is updated to reflect the decision and the door will open.

E. Notification Service and System Update

It is crucial for the system performance for the Firebase database to get updated correctly from the mobile application and Raspberry Pi. We created three different boolean attributes in the database to communicate with the user interface and Raspberry Pi. The IspetInside attribute is used to update pet status. The Raspberry Pi will update this variable whenever it receives a new signal of 0 or 1. The mobile application dashboard showing the pet status will be updated accordingly. The IsDoorOpen attribute determines the status of the door. This value will change based on the user's command. The Raspberry Pi will be updated according to this value. The IspetAtdoor attribute plays an important part in the flow of the system due to its use in the Away mode. This variable generates the notification to mobile applications through the event listeners described in the database section. The Raspberry Pi will update this variable when the RFID sensor detects an object that has a valid RFID. The application will detect a change in the variable then notify a user. Figure 5 shows the system update between the mobile application, Raspberry Pi, and Firebase.

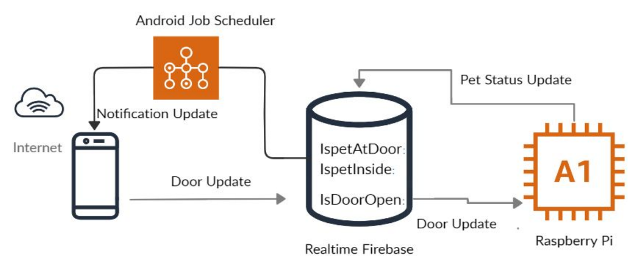


Figure 7. System update workflow between mobile application, Firebase, and Raspberry Pi.

When a user opens the application, a job scheduler initiates and monitors the IspetAtDoor variable. When the Raspberry Pi sends a signal to the database, this variable will change its value from 0 to 1. The event will trigger the application to generate the notification message. Job schedulers work even when the application is closed.

F. Hardware Programming

Along with the programming of the mobile app we also needed to create software for the Raspberry Pi and the MSP430G2553 microcontroller. As stated in previous sections the MSP is used for controlling the home user interface consisting of the LCD screen and the multiple LEDs. To achieve this the C programming language is used to control the MSP430's GPIO pins. The code initializes the LCD for the "home screen", enables the power LED, then sits in a loop, waiting for the operation variable to change. The MSP waits for interrupts from the

Raspberry Pi, which sends a UART message that determines what operation needs to be displayed to the user interface. After each operation the variable is reset and the program returns to the loop.

The Raspberry Pi acts as a main control unit and makes sure all of the components are communicating correctly. We used Python3 to implement this code as this version was needed for the Pyrebase library. Other libraries included the RPi.GPIO for using the GPIO pins, Serial for the UART transmissions to the MCU, and Time for having the program wait in various scenarios. The code consists of an infinite while loop that continuously checks the distance that the Ultrasonic Sensor is reading. If the range read is less than 25 cm the RFID sensors are used to see if the pet was the object detected. With a valid ID the door sequence is then activated by first checking to see what mode the system is in. In Closed mode the code will return to the main loop. In the Away mode the database will be updated accordingly and will then wait in a while loop to give the user a chance to view the video stream. In Home mode the door is opened to let the pet, closed behind it, and then waits for the RFID to recognize the pet trying to get back in. The door is opened once again, is closed and then returns back to the main while loop. Throughout the process the MSP430 is being alerted of the changing states to be displayed on the home user interface.

VI. Hardware Detail

A full understanding in aspects of a hardware design is key to a successful project. The success of the project depends on the ability of the electrical subsystems coming together.

A. *System Thermal Dissipation*

Supplying current and voltage into electrical components generate heat. Thermal analysis is necessary to ensure the life expectancy of the circuit and therefore the project can meet its requirements and specifications. It is to be expected to have an increased thermal footprint as the system reaches full load.

By analyzing the components with a large expected current draw and voltage drop, it would provide an accurate approximation to the heat dissipation of the circuit. Despite the large power draw, the linear actuator has internal components regulated by the speed of the DC brush motor to ensure no overheating issue. In the system's environment, all electrical components would be contacted in free air, creating a natural way for heat dissipation.

The TPS565208 from the 5 V step down circuit is expected to increase to 77 °C junction temperature. The junction-to-ambient thermal resistance of 60 °C/W with expected power dissipation of 784.8 mW. The TPS562207 from 3 V step down circuit is expected to increase temperature to 51.37 °C from ambient temperature. The 3.3V buck regulator has junction-to-ambient thermal resistance of 75 °C/W with expected power dissipation of 285 mW. Both IC chips have operating junction temperatures of 125°C. Therefore, these IC chips do not require additional heat sink or thermal pad to increase heat dissipation.

B. *Motor Control Subsystem*

To support the communication between linear actuator and Raspberry Pi, one must use a motor driver circuit. The linear actuator is a DC brush motor that produces a lot of noise and pulse width modulation signal to operate.

A motor driver must satisfy the following requirements to fit within the electrical system.

1. It must have a minimum of 60 watt power output capability,
2. It must be able to switch the polarity of the output voltage,
3. It must have a low power consumption and low cost,
4. It must be able to be operated by Raspberry Pi's 3.3V GPIO pins,
5. It must provide noise isolation.

The Drok 160W DC Motor Driver Module PWM Speed Regulation Optocoupler Isolation Motor Controller satisfies most of these conditions. The Drok motor driver utilizes four power mosfets in H-Bridge circuit configuration to Upon further inspection and testing, the control signals of the Drok motor driver are configured in active low logic and reference at 4.5 volts. The 3.3V logic signal from Raspberry Pi will not be compatible with this module alone.

A single pole double throw (SPDT) relay is used to foster the digital communication between the Raspberry Pi and motor driver. A double pole double throw (DPDT) relay may be used instead of an SPDT relay since a DPDT relay is equivalent to two SPDT relays. Figure 8. reflects the motor controller configuration.

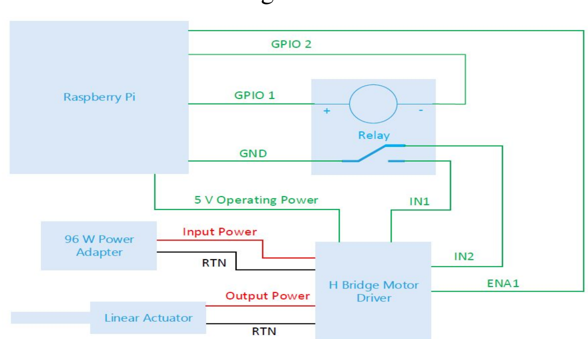


Figure 8. Motor control circuit

VI. Power Design

Power distribution safely and effectively is an important aspect in this project. The project is supplied through a wall outlet that provides 120 volts AC at 60 Hz. The power schematic will provide an AC-DC conversion and then DC-DC step down conversions to supply three voltage rails.

A. *Power Topologies Research*

During the research and development phase, a full wave bridge rectifier with decoupling capacitor and zener diode in parallel, and an inductor in series was investigated as a viable AC-DC converter. However, this power topology wasn't a viable option due to its 80% efficiency and it's need for a large step-down isolation transformer.

An interesting option was a switching mode power supply topology called flyback converter. It is roughly 90% efficient or higher depending on the design. As a modern concept, it showed a great deal of promise. The drawbacks were its circuit complexity and the need for a custom multiple winding transformer, which wouldn't fit the target project budget.

B. *Final Power Design*

Due to the time and budget restraint, a simpler design was needed. A off-the-shelf wall power adapter simplified the design by handling AC-DC conversion to a constant DC voltage. Through power requirements restricted by the other components in the system, three separate power rails with the appropriate voltage levels were needed.

A buck or step down regulator was the best low cost DC- DC converter with minimal external components outside the integrated circuit. Figure X. shows the general circuitry for both 5 V and 3.3 V power rails with the difference in passive components and IC chip. In this power topology, there's decoupling input capacitors to reduce any high-frequency noise into the buck regulator and output capacitor to minimize any output voltage ripple. The resistor divider is used to provide feedback to the buck regulator for output regulation. The inductor is in series and is used to compensate for any sudden change in desired output current. Figure 9 below reflects the general buck regulator configuration for both 5 V and 3.3V power rail.

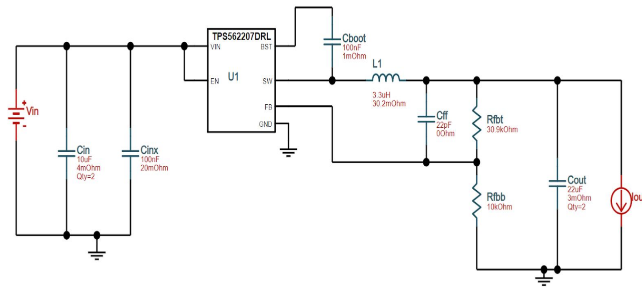


Figure 9. DC-DC Step down Converter Schematic [3]

C. *Overall Power Summary*

The power requirement for the project is required to validate the hardware design. A power rollup analysis generates an approximation of the power requirement through looking at the major power consumption units of the system.

The Raspberry Pi requires 15 watts to operate with its internal components and support all peripherals. With a maximum 50 lb load, the linear actuator draws 60 watts of power. In the Home User Interface, the LCD draws approximately 5 miliwatts. Figure 10, shown below, divides up the significant power consumption based on three different rails: 12 volts, 5 volts, and 3.3 volts respectively.

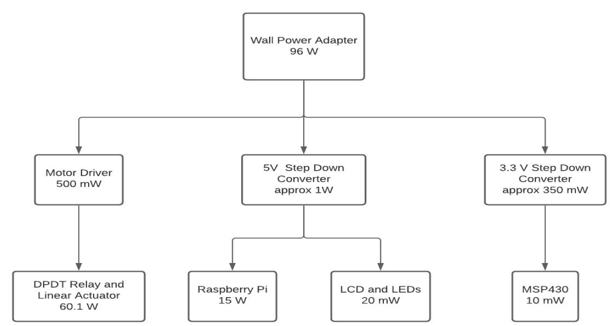


Fig 10. System overall power rollup diagram based on power rails

The 5 V and 3.3 V power rails will draw approximately 16 watts and 1 watts respectively due to its circuit configuration. With a 96 W wall power adapter, AC-DC power supply can provide approximately 79 watts of power for the 12 V power rail, which is sufficient enough for the linear actuator and its supporting components. In conclusion, the power design supplies enough power to support the project.

VII. Board Design

With the main computational unit of the project, Raspberry Pi, as its self sufficient system, this allows for a straightforward circuit design to be implemented on a two layer printed circuit board. The PCB is tasked to step down two voltage rails from a 12V power adapter to the three largest power consumed components:Raspberry Pi, microcontroller, and linear actuator. The printed circuit board will house the MSP430 microcontroller and the home user interface. As a cost effective and polished method, the combination of the home user interface and DC-DC buck regulators reduce the need of two separate housing methods or PCBs. Isolation of the Raspberry Pi, the wifi module, and the linear actuator eliminates the noise concerns to the PCB. Selections of integrated circuits reduce the number of external components and the large spacing between components will support the thermal dissipation.

VIII. Conclusion

At the beginning of our project we set out to make a system that would help both the busy pet owner and their pets. Throughout the senior design process we continuously improved our initial design through research, design, implementation and testing phases. Although we have faced many obstacles and challenges along the way we believe we have met the necessary requirements to call this project a success. We know that with any product or system there is always room for improvement and Pet Connect is not exempt from this. The senior design process has taught us that we must work together as a group to continuously make progress towards the bigger goal and get past the challenges that are going to come along with any project. This skill and experience was very beneficial to all of us as we look forward to beginning careers as engineers.

Acknowledgement

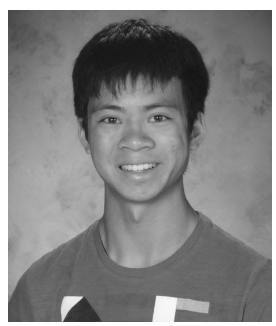
The authors wish to acknowledge the assistance and support of Dr. Samuel Richie and the University of Central Florida CECS department.

The Engineers

Graham Goerg is a 35 year old USMC veteran and a graduating computer engineering student. He hopes to pursue a career as a software engineer or profession of the like, for a company where he can use his prior clearance level, such as Lockeed Martin. Graham is also interested in game design and development.

Joy Weaver is a second-degree student majoring in computer engineering. She received her bachelor's degree in industrial chemistry. She is currently interning with Lockheed Martin and accepted a full-time position as a software engineer starting in December 2020. Joy plans to pursue a Master's degree in Engineering Management.

Ryan Flynn is 23 years old graduating with a major in computer engineering and a minor in psychology. After graduation he hopes to begin his career in software engineering. He has specific interest in embedded systems and mobile app development.



Michael Choi is a 23 year old electrical engineering student. He is currently interning with Lockheed Martin and accepted a full-time position as an entry level electrical engineer, specializing in Analog Design, starting in January 2021. Michael plans to pursue a Master's degree in Electrical Engineering.

References

[1] dataplicity. Stream live video from your Pi. URL: https://docs.dataplicity.com/docs/stream-live-video-from-your-pi

[2] Firebase Realtime Database. Structure Your Database. URL: https://firebase.google.com/docs/database/android/structure-data

[3] Texas Instrument. (n.d.). Power Designer. Texas Instrument Webench. Retrieved November 18, 2020, from https://webench.ti.com/power-designer/switching-regulator/customize/29