Robo-dog: Smart Robot

Jonathan Hernandez, Adrian Lopez, Anthony Coniglio, Esteban Arguello

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

***Abstract* — In recent years, home automation, smart devices, and assistive technology have greatly increased in popularity. Once considered inaccessible or luxurious, smart assistive devices such as Amazon's Alexa, Apple's HomePod, or Google's Assistant have become common household devices. Now, the inaccessible devices are mobile and multi-input solutions which can interact with the users in more ways than one. When it comes to devices that have music playback capabilities, controlling smart home networks connection, or just adding items to your grocery list through command. All these devices lack the physical interactivity and personality present in future AI. This shortcoming alone leaves a space for a new type of product, the smart, interactable, and personable, robodog. Our project seeks to create a prototype of a smart robot dog with multiple levels of interactivity, responding to audio, visual, and kinesthetic cues. This greater aim and level of interaction and responsivity is to fill another role for which a need has arisen, the growing awareness of the problems caused by loneliness and cognitive slump.**

***Index Terms* – Artificial intelligence, computer vision, robot motion, robot programming, smart devices**

I. Introduction

There is an increasing number of people, old and young, suffering from loneliness and anxiety at home, with not many alternatives to traditional cats or dogs. Emotional support animals require time and money to take care of. This can become overwhelming or unattainable for many people. Furthermore, some people in need of these animals have allergies or a family member with allergies that do not allow obtaining an emotional support animal. Therefore, an interactive robot that mimics the basic behavior of an emotional support animal, tied with the functionality of a virtual assistant such as Amazon's Alexa™, could provide a sense of companionship without the need for continuous expenses. Moreover, the integration of the virtual assistant will allow this companion to not only provide emotional support, but utility in a smart, connected household.

When brainstorming multiple ideas for potential projects, there was one clear aspect that the team desired: utility. The team wanted to create a project that would be helpful to its user. Nowadays, the majority of electronic devices are connected to each other via the Internet of Things (IoT). This interconnection permits one potential device to control several devices. With this idea in mind, the team decided on creating a device that could control the IoT within a household. After some brainstorming, it became clear that the best and easiest solution was a virtual assistant. As will be shown further on, virtual assistants are slowly taking over as the main control hubs for multiple IoT devices. If we could create a project with virtual assistant functionality, then our device would be capable of controlling multiple devices with ease. Moreover, companies like Amazon and Google have made integrating their own virtual assistant onto simple projects, like our own, possible. This allows our project to not only control these multiple devices, but it will also be capable of having all the advanced features created by these advanced technology companies. However, this would mean that we are not creating our own virtual assistant, rather integrating it onto our project. This is where the next motivation for our project ties everything together.

As mentioned previously, another motivation for creating this project is to help people in need of companionship. The recent pandemic, mental disorders and the current society have created a lot of issues when it comes to people receiving their desired companionship. The team viewed this project as an opportunity to use our skills to create a robot that could provide the companionship desired by many. From experience, there are few better companions than a pet. Pets are extremely loyal and also loving towards their owners. The team decided that they would then create a robot that would act like a pet. A robot that would be loyal to the user while also never showing negative feelings. From several research papers and studies, it is evident that infusing emotion onto a robot is very complex and beyond the knowledge of the team. As will be further explained later, the robot only moves and shows basic emotions to act like a dog. The team hopes that our project can be used as a prototype for a much more complicated robot, capable of imitating emotion with higher fidelity. It is when we combine both motivations that we obtain the overall idea of our project; a robot dog with the capability of connecting and controlling IoT devices using a virtual assistant.

The overall goal of the project is to have a portable, easy to use autonomous robot that can be interacted with to perform both the functions of a virtual assistant, and the functions of a dog. The project should be light weight (easy for the average adult to pick up and carry), not prohibitively expensive, interactive, reliable, and with a reasonable use time. The size of the robot is around the size of a small dog. Its size allows the robot to be easily transportable while also making it cheap to produce. Most of the robot's costs will be in the Raspberry Pi due to the fact this robot requires a recent, powerful Raspberry Pi 4 due to its level of complexity. Maintenance is cheap as it is rechargeable and lasts about 4 hours of use. One of the goals is to make the robot easy to use. The commands are not so complicated that the user must remember specific esoteric sentences. Rather, the actions of the robot are straightforward phrases commanded by the user. An easy-to-use smart phone app was also developed to provide another method of controlling the robot.

II. Overview of Requirements

A. Movement Capability and Resemblance to a Dog

There are three major requirements necessary for the robot to accomplish its defined goals and objectives. The first requirement is movement capability and resemblance to a dog. The robot is capable of walking like a dog and posing in a few positions that will make it resemble a dog. Specifically, these movements are fluid and do not take the robot more than 20 second to accomplish. When walking, the robot moves its legs in a manner that resembles the walking of a dog. This means moving two legs first and then the other two. Furthermore, the joints of the legs resemble those used by a regular dog. There are two motors for each leg.

Fig. 1. Image showing example of leg mounted onto the lower body of the robot.

To satisfy this requirement, the motors are strong enough to withstand the weight of the entire dog. Likewise, the batteries are powerful enough to manage all eight motors and other electronic components while also have limited weight so the motors can control the movement efficiently. Good balance is also necessary to satisfy this requirement. As the robot is capable of moving to different positions, it must know at which position the motors currently are at and at what position it must get to.

B. Virtual Assistant Integration

Another important requirement for this project is the virtual assistant integration. This requirement means the robot is capable of internet access. Furthermore, the robot uses the virtual assistant for its mandated purpose. New commands can be coded into the virtual assistant to allow the robot to do more actions and tasks, but in this prototype stage only basic commands are used. The virtual assistant integration will make this robot more than just a robotic animal. The virtual assistant capability will make the robot dog useful on a day-to-day basis. Of course, for this to work, the robot is also capable of voice recognition. The robot is capable of listening to commands in order to execute them. The voice recognition is accurate enough so that it understands the user most of the time, without annoying said user. In other words, the robot has a microphone and a speaker that can understand the users' commands and demonstrates that the commands have been understood. Specifically, the microphone is able0 to understand when the robot is being called from a distance of at least two meters with almost total accuracy. Likewise, the speaker is loud enough so the user can hear the response from this distance.

C. Camera and Artificial Intelligence (AI) Integration

Since the robot is representing a dog, it is natural for the robot to also have some perception through multiple "senses". The robot is capable of listening, speaking, "feeling" (responding to) touch, and roughly observing its surroundings. The final major requirement is the robot being capable of detecting its surrounding. Specifically, Robo-dog is able to distinguish colors put in front of its camera and identify which color is being presented. Initially, the goal was for the robot to be able to identify its owner through the camera and use of AI integration. Unfortunately, the AI program only works with the Python 2 programming language. The team needed to use Python 3 for the rest of the functions being handled by the Raspberry Pi and thus the original AI software had to be discarded. Until the software used for the AI integration gets updated, the team cannot use it for this project. However, the robot still has smart vision capabilities. The camera will recognize different colors and identify them accordingly as previously described.

III. Part Description

A. Changes in Design

Many different designs and parts were selected for this project. In the early designs, the robot would have a neck and head piece, and a tail. Both the neck and the tail would include separate motors and would move with the robot.

These were discarded since head movement would increase the complexity of the robot and the team did not feel comfortable with the increased difficulty. Another design considered was the use of wheels for movement. This design allows for simple programming and creation of movement for the robot. It did, however, go against one of the main goals of the project which involves the robot resembling and acting like a dog. The wheels would make the robot look more like a vehicle than a pet, so the idea was discarded. This decision created a large difficulty spike when it came to the movement and balance of the dog. The dog is designed to move like a dog using all two servo motors on each leg.

B. Current Design

The design chosen for the robot will have no neck, tail, or head piece. It does, in simple terms, include four legs with two articulations each and a main body that hosts most of the electronics on the inside. The battery and the speaker will be placed on the bottom of the robot on the outside of the casing since they do not fit on the inside. The team decided on not expanding on the original design since the dimension of the robot chassis were created with a specific width of the body in mind. The touch sensor was placed on the top of the robot where it is easily accessible by the user. The motors are embedded within the legs of the robot and the cables are moved inside the legs of the robot, so they remain hidden and help the robo-dog keep a more dog-like appearance. All PCBs and microcontrollers are located on the inside of the body and hidden from the user. The body contains multiple ports that will allow the used to interact with the PCBs within.

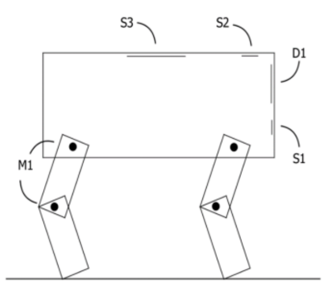

Fig. 2. Sketch diagram of robot body and articulations.

The body is 3D-printed using white plastic. This plastic is not only cheap but also easy to come by. Moreover, by 3D-printing the body, the body can be customized to adapt to the multiple specifications the robot requires to function properly while also maintaining a pet image.

The original architecture of the robot was not created by the team, it was created by the Daniel Hingston [1] and is used accordingly to the with the CC license provided. As no one from the team were mechanical engineers and had little experience with designing mechanical parts, the team decided on taking the template provided from Hingston and adapting it to satisfy the needs of the team. Change in the design include increasing the width to fit a larger PCB, more holes for different sockets required to connect different components, and other small changes to help better satisfy the needs of this project.

C. Microcontroller Unit (MCU)

The microcontroller part selection was an integral part of the design because the MCU will coordinate the motor control and have several components attached to it through its general-purpose pins.

MCU	**Atmega328P**	MSP430G2553
Frequency	**20 MHz**	16 MHz
Architecture	**8-bit**	16-bit
Pins	**28**	24

Table 1. Comparison table between two considerations. Final choice is bolded.

D. Servo Motors

Analyzing the different movement options available and deciding on a walking quadruped robot, servo motors became the perfect motors to be used for this robot. The servo motors chosen were the MGS90 servo motors. These motors include their own encoders and gearbox, simplifying the circuitry required to use them. Furthermore, they have metal gears which help with precision and rigidness of the robot. The main advantage of using servo motors over stepper motors is the feedback system. With feedback, the team was capable of observing the motors' current position and making adjustments as required. This was especially helpful for balancing the robot and assuring that each pose is done correctly.

E. Voltage Regulator

One of the most important selections for this project was the voltage regulator which was used at the center of the power supply circuit in this project. It is important that it is flexible, low-cost and has a small a footprint as possible while maintaining those other key factors. With a large assortment to choose from, with at least hundreds of relevant ICs available, the selection process needed to be efficient to avoid unnecessary delay. The selection was narrowed greatly by the constraints placed on the design process due to current market and manufacturing conditions.

Specification	L7805CV	**LD1085V50**
Max voltage in	30 V	**30 V**
Max amperage	1.5 A	**3 A**
Voltage output	5 V	**5 V**

Table 1. Comparison between two possible voltage regulators, final choice was the LD1085V50 (in bold).

The design seen in Figure 7, centered around the LD1085 5V, 3A voltage regulator, was quickly pivoted to and implemented in the final PCB power supply design to provide power for the MCU and servo motors.

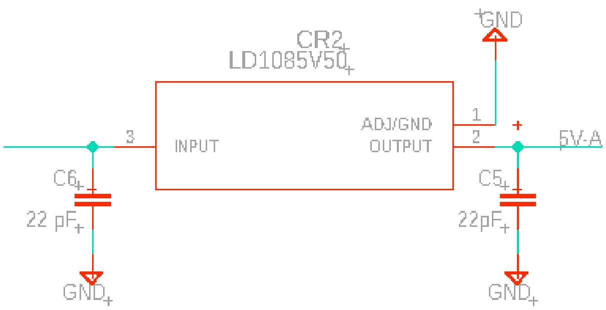

Fig. 3. LD1085 voltage regulator integrated circuit used for power the motors and MCU of the robot.

The LD1085 voltage regulator has several benefits. It requires very few parts to work in conjunction with it, thus reducing failure points and is simple to implement. For a project on a short time scale such as this one, that is also very important given that it should theoretically be quickly deployable and one less system to troubleshoot once deployed. Another factor in the selection of this part which is not a technical parameter but is vital in the current environment is that the part is in stock and ready to order through consumer markets. Many of the voltage regulators and DC-to-DC convertor circuits considered were found to be out of stock or had parts required for the implementation be out of stock. Again, given the time frame of this project and current market plus manufacturing conditions, making sure that every part selected is ready-to-order is of the utmost priority. The lack of ready availability of voltage regulators is one area where these shortages are apparent and greatly influence possible selections. Other, more power efficient designs had to be discarded due to these supply constraints.

F. Force Sensing Resistor (FSR) Sensor

For the touch sensing capability of the robot, the team decided on using an FSR sensor. These sensors work by changing resistance based on the pressure applied to the sensor. A pulldown resistor and software are used to adapt the sensitivity of the sensor. The objective of this sensor is to detect when the robot is “petted.” Upon detection, the dog acts accordingly, either by pronouncing an auditory cue or performing a specific action. The sensor does not provide multi-touch capability, but this is not necessary for the simple requirements of the touch capability of the robot. Moreover, the sensor is light and thin enough that it can be placed on top of the robot casing and not be very noticeable. Ideally, the sensor would have gone under the plastic casing of the robot, but after testing the casing proved to too thick and this idea unfeasible.

IV. Hardware Detail

The design of the robot dog needs to be separated into multiple subsystems so that it can be easily tested and created. The first subsystem involves the power control of the entire robot. This subsystem explains how the robot will be powered and the circuits used to do so. The second subsystem deals with all the components connected to the ATmega328p. The third subsystem deals will all the components related to the Raspberry Pi and the AI. These subsystems are designed in an independent manner; they expect the subsystem they depend on to work. For example, the second and third subsystems are created with little regard to their need in power. The power subsystem is designed knowing the requirements the second and third subsystems need, yet not knowing how the subsystem will distribute the power. This division in subsystems is necessary to allow for individual problems to be fixed efficiently.

The potential problems of each subsystem were different and required different methods to solve them. Problems in the first subsystem included incorrect voltage readings and unusual draws of current. Problems in the second subsystem included robot balancing and coordination between the motors. Problems in the third subsystem were mostly software related. To reiterate, this division helped to fix the multiple problems. Issues with current and voltage originated from the first subsystem, while issues with movement had to do with the second subsystem. There were scenarios where this problem division was not necessarily true, and the problem was actually in another subsystem, but these scenarios were minor, not every significant, and easily resolved

A. Power Subsystem

The first subsystem is assigned to all the power controls of the robot. The system is capable of powering all the devices and components used in the robot dog. This subsystem contains several different technical engineering applications and its ability to work shows a clear understanding on different power-related circuits.

To accomplish powering the robot, voltage regulators will be required to provide a safe power source to all the power supplies. Furthermore, a DC-to-DC Buck converter is necessary to convert the batteries voltage into 5 volts DC to be used by Raspberry Pi computer of the robot.

Components are smartly chosen to have the same voltage requirement so less complication had to be done in the power subsystem, that is, a 2S lithium polymer battery was used which outputs at 7.4 V nominal, and all voltage regulators and buck convertors can use this input at output a consistent 5 V which is used by all components.

Part	Input voltage	Min draw	Max. amp draw
Motors	5 V	100 mA	800 mA
MCU	5 V	15 mA	40 mA
RPi	5 V	500 mA	900 mA
Total		**615 mA**	**1740 mA**

Table 2. Summary of expected power draw.

Since the motors can draw a large amount of current, it is better to have this part of the system powered separately from the rest of the components. The next section to power is the ATmega328p integrated circuit and the touch sensor attached to it. Powering this part of the robot is simple and the voltage needs to simply be stepped down and regulated to 5V, as both components work well with this value. The next part this subsystem needs to power is the Raspberry Pi and all the sensors it will include. This subsystem is expected to take a fair amount of current for processing the camera, virtual assistant and AI. This system also uses 5 V to function, so a DC-to-DC buck convertor was used. This part proved to be most difficult of all to power as the internal circuity of the Pi required a very consistent voltage, refusing to power on at all if its input voltage drops below 4.7 V. Previous power system designs did not accomplish this, and thus an aftermarket design had to be adapted for use with our power circuitry. A similar PCB was designed which was more fitting for the extremely steady voltage and high current draw required by the Pi, but time, budget, and supply constraints made it so ordering this new PCB design was not feasible for integration by the project deadline. All components were selected to make sure powering and using them would be straightforward, and the rest of the PCB was custom designed.

This system is also designed with rechargeable batteries in mind. The battery needs to survive at least eight hours of normal functionality and around two hours at maximum strain. The batteries are rechargeable so the robot can be used constantly and recharged ideally once per day. Furthermore, this leads to less waste and more environment-friendly design. Realistically, there should rarely be a situation in which the robot will be used at maximum power consumption. A 5500 mAh 2S li-po battery was used, which is in the form of a standard “shorty” remote-controlled (RC) car battery. This adds convenience for the user as it ensures that a commonplace battery issue or simply battery degradation over time will not leave their robo-dog unusable. The same is true for battery charging, as many cheap aftermarket chargers are available for these types of batteries.

To regulate and step down the voltage, voltage regulators and buck convertors were used. For the robot, linear voltage regulators are used to power the atmega328p and smaller components attached to it, as well as the servo motors. These components will have not very strict power requirements so the use of voltage regulators will have little impact on overall power consumption. Hence, using them will make this subsystem simpler to design and test.

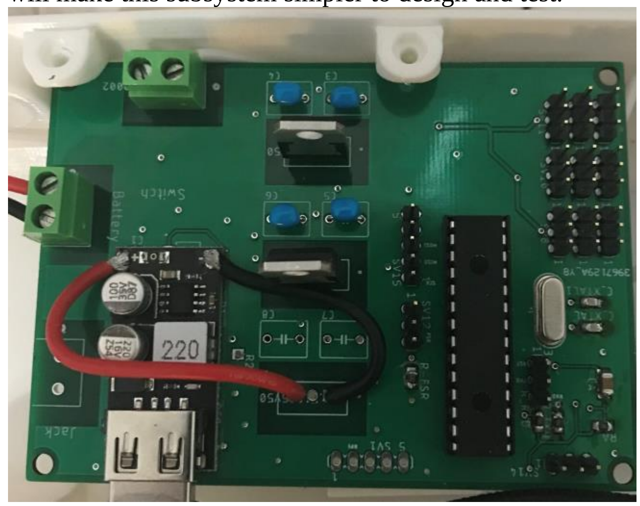

Fig. 4. PCB containing power supply subsystem and Atmega328p Subsystem connections.

The input and output of these circuits are the most important parts and can damage the other electronics components if they are not designed properly. It is extremely important that these circuits provide accurate results and that they are also stable. Varying voltages could lead to damages in the electronic components of the robot. These necessities were met and the problems leading to such issues were dealt with.

B. Atmega328p Subsystem

The Atmega328p [2] is the microcontroller chip of choice for controlling the mobility and touch sensor of the robot. The chip is the same one used in many Arduino products like the Arduino Uno. Using this microcontroller, the team was capable of programming the motors with Arduino software and libraries without using the entire PCB of an Arduino. The custom PCB is small and only requires connections for the motors, touch sensor, and connection to the other subsystems of the robot.

The Atmega328p has 13 digital pins which can be used for controlling the servo motors used for movement. As mentioned, the Atmega328p is the same microcontroller that is used with many Arduino boards. In fact, the Arduino Uno is the device that was used to prototype the robot software and connections. The Arduino is a microcontroller board that everyone in the team has previous knowledge working on. It is simple to use and has many openly available libraries. Furthermore, creating a PCB that contained this IC was easy given its compact size. It can be easily integrated into any custom PCB design we use for this project. To control all eight servo motors, the microcontroller required a functioning clock quicker that the one built in. Therefore, a 16 MHz crystal oscillator was used to create this clock. The final PCB design containing both subsystems is shown in Figure 5. Another subsystem, the battery charger, had to ultimately be abandoned due to consistency issues.

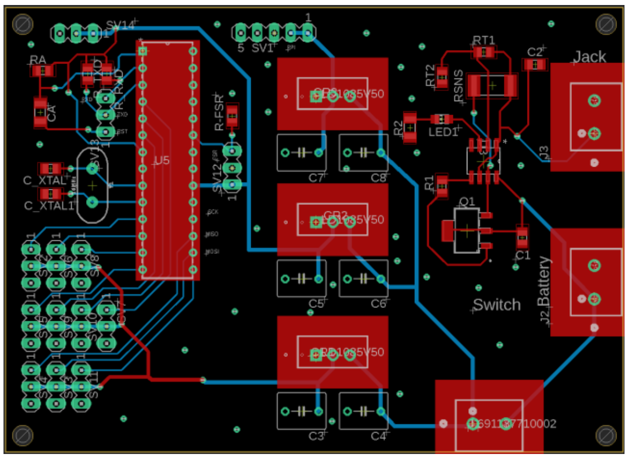

Fig.5. PCB design containing power supply subsystem and Atmega328p Subsystem connections.

The overall printed circuit board design includes the MCU subsystem on the left side, with the motor pins tied to digital pins on the MCU at the bottom left, the crystal oscillator in the middle, and the serial communication pins near the top left. To the right of the MCU is the FSR connection. The middle part contains the power subsystem, and the right part is where the terminal blocks to connect to the battery and an on/off switch are located.

C. Raspberry Pi Subsystem

The microcontroller chosen to control the virtual assistant, Atmega328p, camera, microphone, speaker, visual AI software (TensorFlow lite and OpenCV), and Apache server all run on the Raspberry Pi 4. The Raspberry Pi 4, like all other components of the Robo-dog, runs off of 5V DC. Furthermore, its Quad core 1.5 GHz chip with 8GB of RAM is more than enough for running the AI, Alexa and motor control software. This microcontroller is strong enough to handle all the heavy software load required for the project [3]. The Pi subsystem includes the camera, speaker, and microphone which will all be used by the robot to interact with the user. The biggest focus of this subsystem comes with in the software implementation. The hardware subsystem for this project is handled mostly by the Atmega328p chip and communicates with the Pi through serial connection. Leaving the Pi with minimum hardware system use, it mainly serves to run all the software the microcontroller will be reacting to.

IV. SOFTWARE DETAIL

This robot has large amount of software required to work properly. The software portion of our robo-dog was implemented on an Atmega328P MCU and Raspberry Pi 4 computer. Both electronics have their own separate tasks that needed to come together to make everything work properly. Summarization of main tasks include processing camera object/face inputs, voice inputs, generating pulse width modulation signals to control the pulses to the servos, receive and interpret touch data from the touchpad sensors, and more. The software had to be designed in a way where it can interface with all the components, while also having each component sufficiently separated for easy debugging and user control.

A. Direction and Speed Controls

The direction buttons send commands to GPIO pins 8 & 11 for motor 1 and GPIO pins 14 & 15 for motor 2.
The speed slider sends a value between 0-100 (in increments of 10) to server. This value is used to generate PWM on pins 20 & 21 simultaneously, resulting in speed control of motors.

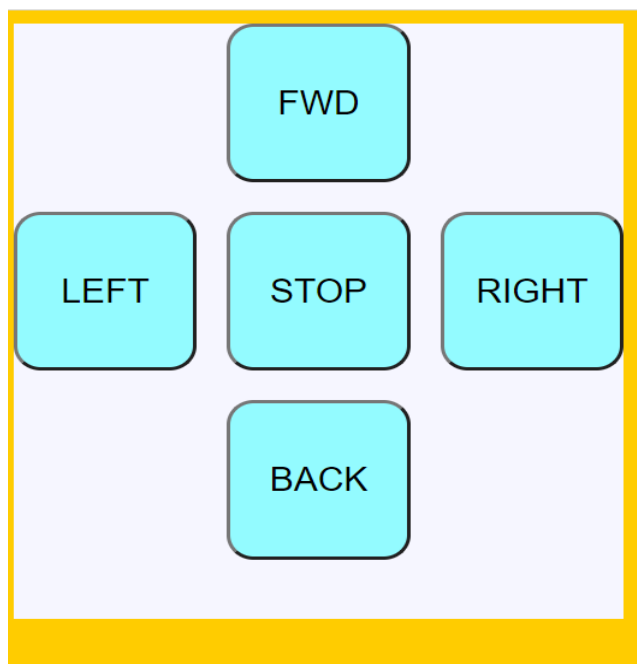

Fig. 6. Webapp controls for motor direction and speed

B. Raspberry Pi Camera

Below is an image of the camera which gives our robot the vision it needs to recognize objects. With its 1080p30 resolution this camera is more than capable enough for our robot to recognize objects and colors.

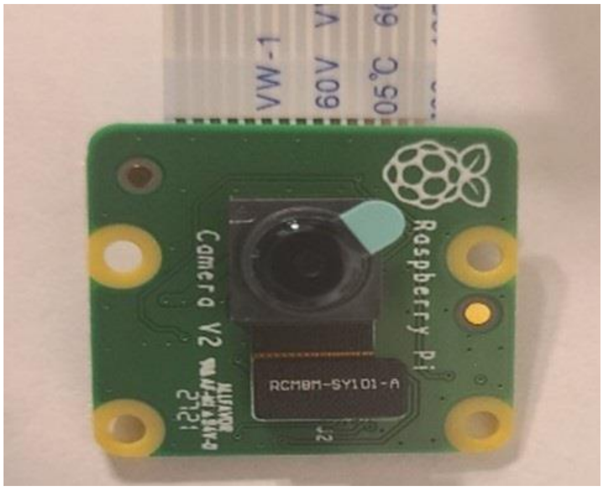

Fig. 7. Raspberry Pi camera module V

This camera will be powered through a dedicated connection on the Raspberry Pi and uses TensorFlow with Python to give the robot the vision it needs. When it comes to an end result, our camera can tell the different between colors and disregarding the integration issues, it was also able to detect it owners and recognize unfamiliar members and categorize them. With Python 3 integration, a new revision of the robot could have user specific actions.

Fig. 8. Webapp controls for camera enable/disable

When you press camera 'ON' button, a python script '/cam_server.py' is launched in the background and starts streaming the camera video.

C. TensorFlow Object Detection

TensorFlow is a free and open source software capable of powerful image recogniction and customizable levels of abstraction allowing us to customize the software to our needs. Additionally, TensorFlow is easy to use and has many resources available on the web for interfacing with a Raspberry Pi.

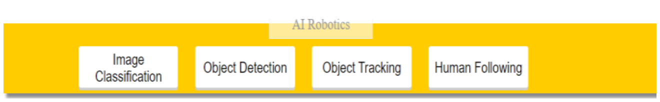

Fig. 9. Webapp controls for TensorFlow systems image classification, object detection, object tracking, human following

Pressing the button in the image above each run a different python script. 'Image Classification' runs a python script named 'image_recog_cv2.py' which will try to predict what object the camera is looking at. 'Object Detection' runs 'object_detection.py' which will show a web UI through which you can select objects of interest to track. 'Object Tracking' runs 'object_tracking.py' which will try to track an object's location. Finally, 'Human Following' runs 'human_follower.py' which will launch the camera view and track a person.

Fig. 10. Webapp portrayal of object detection using TensorFlow

D. Alexa AI Assistant

Artificial Intelligence (AI) assistants combine a group of AI technologies to facilitate the use of devices. Some of these technologies include a Wake Word (WW) detector, Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Dialogue Manager (DM), and Text to Speech (TTS). Combining all of these, assistants can perform multiple tasks by simple voice commands.

Unfortunately, at the current time the two wake word engines supported by the Amazon Alexa SDK are Sensory and KITT.AI both repositories being deprecated/un-updated making having a wake word engine in a Pi based Alexa product impossible. As a result, to activate the Alexa's listening state a button which manually enters "t" to the terminal will be used. Pictured below is our webapp's implementation of a listening state enable/disable.

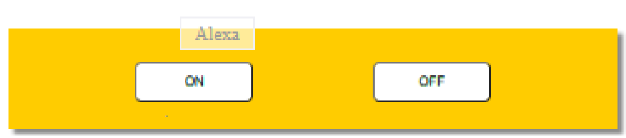

Fig. 11. Webapp controls for enable/disable of Alexa listening state.

VII. Conclusion

This project seeks to cater to users with an alternative to live animal emotional support companionship. The number of people regardless of age, old and young, are suffering from anxiety, depression, loneliness, and stress in the home, with not many alternatives to traditional cats or dogs. Besides the emotional benefactor of our project a robo-dog is also a monetary proposition. Emotional support animals require time and money to take care of. This financial burden can become overwhelming or render the possibility of attaining or caring for the dog unattainable for many people. Additionally, some people in need of these animals have allergies or a family member with allergies that do not allow obtaining an emotional support animal. Therefore, an interactive robot that mimics the basic behavior of an emotional support animal, tied with the functionality of a virtual assistant such as Amazon's Alexa™, could provide a sense of companionship without the need for continuous expenses.

This robot will hopefully, not only bring companionship, but several other qualities that will make it unique. The robot was designed with the idea of being more than just a robotic dog. The robot has many features like touch sensing, hearing, speaking, and many other features than make the robo-dog feel real. The dog will listen to custom commands, interact when it is petted, etc. It also has different poses and expresses different emotions. The robot is made with the purpose of being a realistic dog, while also providing new features. With a virtual assistant, the robot can communicate and interacting with several smart devices. It can also do anything any other smart device can do.

As we developed and designed the robo-dog we approached and overcame many obstacles in its hardware and in its software. The creation of a robo-dog is not unique and is a project that has been executed many times to varying degrees. We wished to create not just something emotional but also useful bringing the robot assistant to the modern era with onboard speech recognition and Alexa/Assistance support. To create a companion a family can, find utility and joy with.

Acknowledgement

The authors wish to acknowledge the assistance and advice from the multiple online communities which provided free resources and recommendations on how to proceed with different issues encountered by the authors. Additional acknowledgement and thanks go to project advisors Dr. Lei Wei and Dr. Samuel Richie, as well as project reviewers Mr. Sundar Sinnappan, Dr. Mahdi Assefzadeh, and Dr. Chinwendu Enioyha.

References

[1] Hingston, D. (2020). GoodBoy - 3D Printed Arduino Robot Dog. Instructables Workshop. Retrieved 2021, from https://www.instructables.com/GoodBoy-3D-Printed-Arduino-Robot-Dog/

[2] Arduino Uno R3. (2021, June 12). Retrieved from Arduino: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[3] Raspberry Pi 4 Model B. (2019, June). Retrieved from Raspberry Pi: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

[4] Functional Requirements for AVS Products. (n.d.). Retrieved from Alexa Developer: https://developer.amazon.com/en-US/docs/alexa/alexa-voice-service/functional-requirements.html

[5] NNAMC1580PC01. (n.d.). Retrieved from Digikey: https://www.digikey.com/en/products/detail/neonode-inc/NNAMC1580PC01/7776715

[6] ROBOMECH Journal. (2020, June 25). Retrieved from Springer Open: https://robomechjournal.springeropen.com/articles/10.1186/s40648-020-00174-1

[7] Square Force-Sensitive Resistor. (n.d.). Retrieved from Adafruit: https://www.adafruit.com/product/1075#technical-details