COEN315 Final Project
Smart Cat Feeder

Ivy Wu
Department of Computer Science
Santa Clara University
San Jose, 95128, USA
Lwul@gmail.com

Chang Liu
Department of Computer Science
Santa Clara University
San Jose, 95050, USA
Cliu6@scu.edu

***Abstract*:** Our final project for COEN 315 is to build an automatic cat feeder, which is capable of automatically feeding cats at the correct time with remote control. This final report describes our analysis, design, and implementation of the final project assignment for COEN 315, Web Architecture & Protocols. The term project is built upon the material covered in class. The ultimate goal of the project is to enable students to have hands-on experience with IoT project that involves micro-controller-based sensor modules, firmware for the modules, software components, RESTful middleware server, as well as user-facing web and mobile application.

Keyword: ESP8266, ESP32, Arduino, ArduCAM, Microcontroller, IoT, Angular5, Firebase, JSON.

I. Introduction

Nowadays, people live in the cities all lead busy lives, and our tight schedules sometimes prevent us from feeding our cats in an routine manner. One of the top health concerns of cats are overeating and obesity [4]. At younger age, kittens usually can eat much is given to them. So, without a healthy feeding schedule, our cats can easily get overweighed. Thus, the main problem is the difficulty to guarantee that our furry friends get their meals at the same time everyday. To solve this problem, our group, Orange, implemented a smart cat feeder, which is capable of automatically and remotely dispense the correct amount of food on time. Pet keeping is a time consuming responsibility, with our project, owners can feed their cats easily and smartly.

II. Project Objective

A. Purpose

The main purpose of this project is to implement a smart cat feeder that helps all the pet owners to feed their pets remotely and smartly with the following functionalities:

- dispenses correct amount of food at preset time
- displays the current food amount by capturing pictures of the bowl
- dispense food instantly by a button click
- create user-friendly web and mobile interface that is easy to navigate and use

B. Background

With the development of Internet, everything in today's world is connected regardless of time or location. The Internet of things (IoT), a global network infrastructure, links physical and virtual objects through the exploitation of data capture and communication capabilities. It creates a new convenient, quantifiable, and measurable world. IoT devices can be classified into three domains:

- wearable
- smart home devices
- M2M devices

With the development of Internet, wireless network, and cellular network, mobile smart devices and web are changing the way we connect with the world. The use of mobile data has gown exponentially in the last 5 years. According to the Flurry Analytics research, round 90% of our daily time are spend on apps within mobile devices [7]. Web and Mobile app programming provide perfect scope to access IoT devices. They are not just website or simple mobile apps; they are about interfaces across many devices working together to create seamless experience [9]. Many applications such as healthcare, travel, retail and more are using mobile and web connectives to access IoT ecosystems. This integrated experience creates huge potential of mobile, web, and the IoT to be more transformative for our lives. In our project, we implemented both web and mobile user interface to connect with the automatic cat feeder. Users are able to monitor and control the devices remotely anywhere at anytime.

III. Design

A. Our solution

The block diagram (Figure1) illustrates the high level communication between system components. Sensor modules, including camera and servo, communicate with the

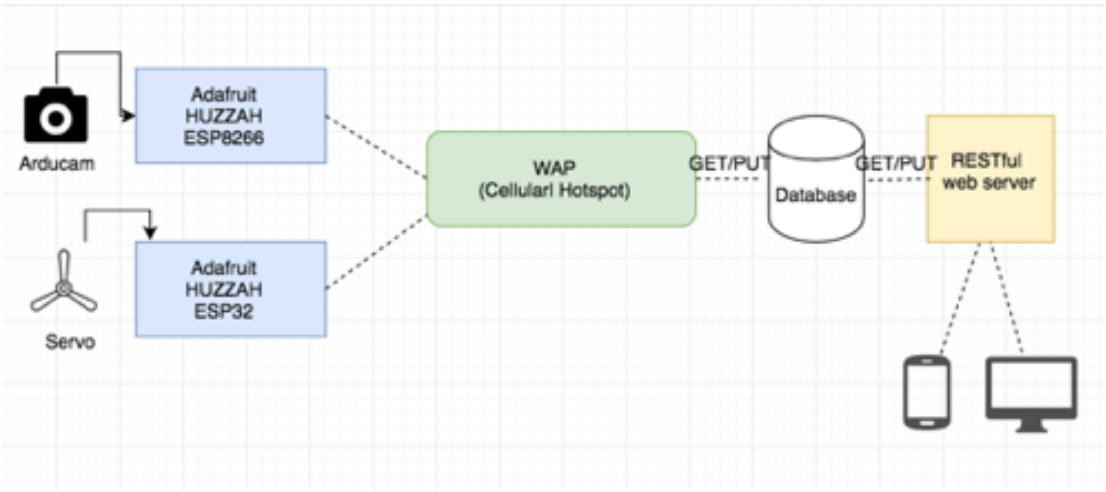

Figure1. Block Diagram

database by constantly GET users updates and PUT self information via a secure Wi-Fi connection. Our web server also connects to the same database to constantly GET the updated information of the device and PUT data when user interacts with the front-end interface.

B. Hardware

In our design, we used two micro-controllers: Adafruit Feather HUZZAH ESP8266 (Figure2) and Adafruit HUZZAH ESP32.Our design approach uses one ESP8266 micro-controller as the heart of Arduino camera and one ESP32 micro-controller to control the servo.

ESP8266

Adafruit Feather HUZZAH ESP8266 (Figure3) is a bite-sized Wi-Fi micro-controller. It is the new development board from Adafruit. ESP8266 is at 80MHz with 3.3V logic/power. It has 9 GPIO pins. We chose this controller because of its superior ease of use and "all-in-one" ESP8266 Wi-Fi development board with built in USB and battery charging [4]. It is affordable for the scope of project with all the extras we need.

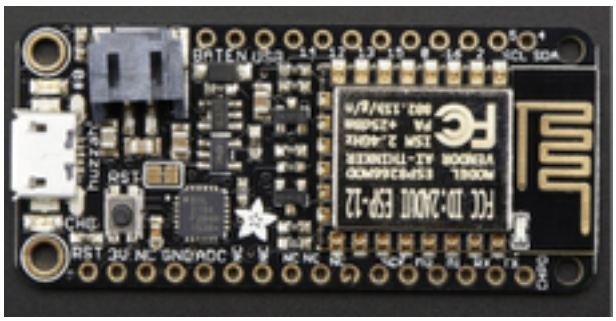

Figure2. ESP8266

ArduCAM Mini 2MP (OV2640)

We used ArduCAM Mini 2MP in our project to capture the food image. This camera is a high definition 2MP SPI camera that reduce the complexity of the camera control interface [5]. We chose this camera because the price is more affordable than other ones in the ArduCAM series. Also, it has easy-to-use hardware interface and many open-source libraries online to refer to.

Figure3. ArduCAM mini OV2640

Hardware Set up

As seen from Figure, the micro-controller board is mounted to a half-sized breadboard. It is powered by connecting to the USB port of a computer. The board can also be powered by connecting a USB cable to a power bank or connect to a 3.7V Lithium polymer battery from the controller's V_{bat} and GNV pins on the top edge of the breadboard. The ArduCAM is connected to the 3.3v regulated power rail on the bottom edge of the breadboard.

Serial Peripheral Interface (SPI) is an interface that enables the serial (one bit at a time) exchange of data between two devices [11]. As mention above, our camera is a 2MP SPI camera. ESP8266 can be used to control SPI devices and it is fast enough to match Arduino level speeds. Table1 concludes the complete pin-out data for the camera and its connection with the ESP8266 controller.

Pin	Type	Description	
CS	Input	SPI slave chip select input	GPIO16
MOSI	Input	SPI master output slave input	GPIO13
MISO	Output	SPI master input slave output	GPIO14
SCK	Input	SPI serial clock	GPIO12
GND	Ground	Power ground	GND
VCC	Power	3.3v - 5v power supply	3V
SDA	Bi-directional	Two-wire serial interface data I/O	GPIO4
SCL	Input	Two-wire serial interface clock	GPIO5

Table 1. Pin-out data of ArduCAM (connection with ESP8266)

ESP32

The ESP32 is a perfect upgrade from the ESP8266 that has been so popular (Figure 4). In comparison, the ESP32 has way more GPIO, plenty of analog inputs, two analog outputs, multiple extra peripherals (like a spare UART), two cores so we don't have to yield to the Wi-Fi manager, has much higher speed processor. It contains a dual-core ESP32 chip, 4 MB of SPI Flash, tuned antenna. The ESP32 has both Wi-Fi and Bluetooth Classic/LE support. That means it's perfect for Internet-connected cat feeder [2].

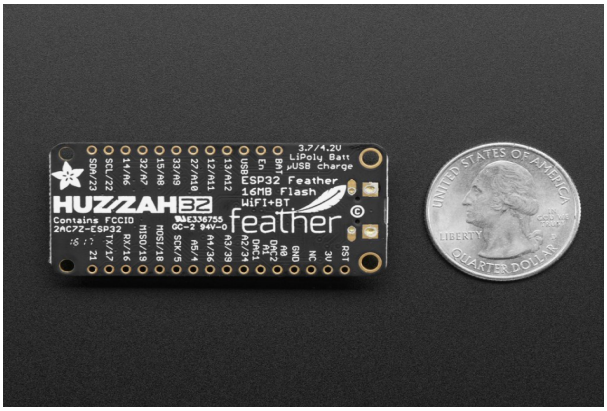

Figure4. HUZZAH ESP32

Servo

This servo is a linear actuator that allows for precise control of angular or linear position, velocity, and acceleration (Figure5). It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are controlled by sending an electrical pulse of variable width or pulse width modulation (PWM), through the control wire. There is a minimum pulse with 1m width, a maximum pulse with 2.0m width, and a repetition rate.

Figure5. Digital Servo

A servo motor cans usually only turn 90 degrees in either direction for a total of 180 degrees' movement. Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

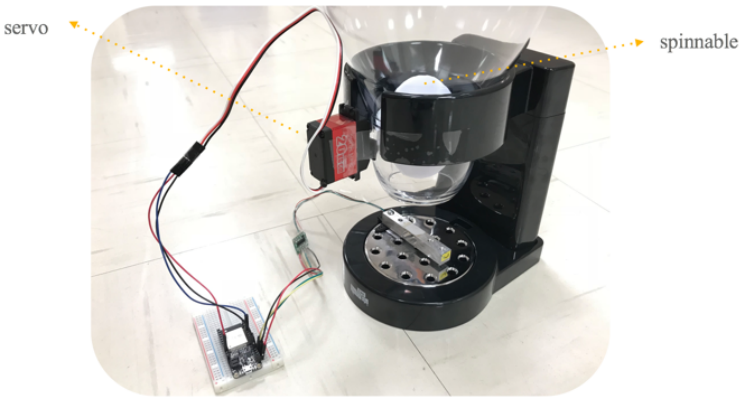

Figure 6. Servo assemble

We used a servo to spin the switch and connect the servo to a rotatable switch (Figure6) so we can control food dispenser. This servo has a full metal gear and it is better for accuracy. The aluminum case enhances the heat dissipation and ensures that servo motor can work well. Although the controllable angle rage if just from 0 to 180 degrees with linear change, it already meets our needs.

Hardware Set up

We connect three jumper wires to the servo in Figure 7:

- Feather GND to the brown or black servo wire
- Feather USB to the red servo wire
- Feather Pin 2 to the yellow servo wire

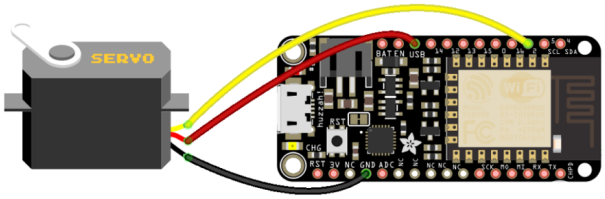

Figure7. How servo connected with ESP32

Table2 shows which pins are connected to the digital servo in our program.

Pin	Description
GND	Breadboard ground rail
USB	Power for servo
21	Pin for servo DOUT

Table2. Servo pin-outs table

C. Software

Arduino 1.8.5

Arduino IDE can be used with any Arduino board, including the Arduino Yún and Arduino DUE [9]. We use C++ to program our servo, weight sensor and camera, and use HTTPclient to connect to WiFi and use the get and put methods to transfer data with Firebase.

WebStorm

WebStorm is a great full-fledged IDE for those who wants a full development experience when working with JavaScript [10]. We used this IDE and created our website design with the Angular CLI.

AngularFire Lite is the very first and the only current library that supports server-side rendering out of the box for both Firestone and the Real-time Database. It has other features over the Angularfire2 library like Storage, Observable based Transactions & Batched Writes, and Cloud Messaging. So we use the Angular 5 framework for our project.

Firebase Real-time Database

Firebase can store and sync data with our NoSQL cloud database. The Firebase Real-time Database is a cloud-hosted database [6]. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Real-time Database instance and automatically receive updates with the newest data.

We connected our database with website and also connect database with microcontrollers. So both website and two microcontrollers can transfer data through the database (Figure 8). The real-time database can be used as a backend for our web application. Our website is built on the Angular 5 and Firebase is a perfect match for Angular 5 application.

To integrate Firebase services with Angular 5 application, we wrote our own service in Angular 5 and injected it into our code as a library, so we can use the RESTful method for data communication between chips and websites. With the speed and performance Angular 5 delivers for the front-end as well as the real-time data sync feature, we are able to further improve the user experience.

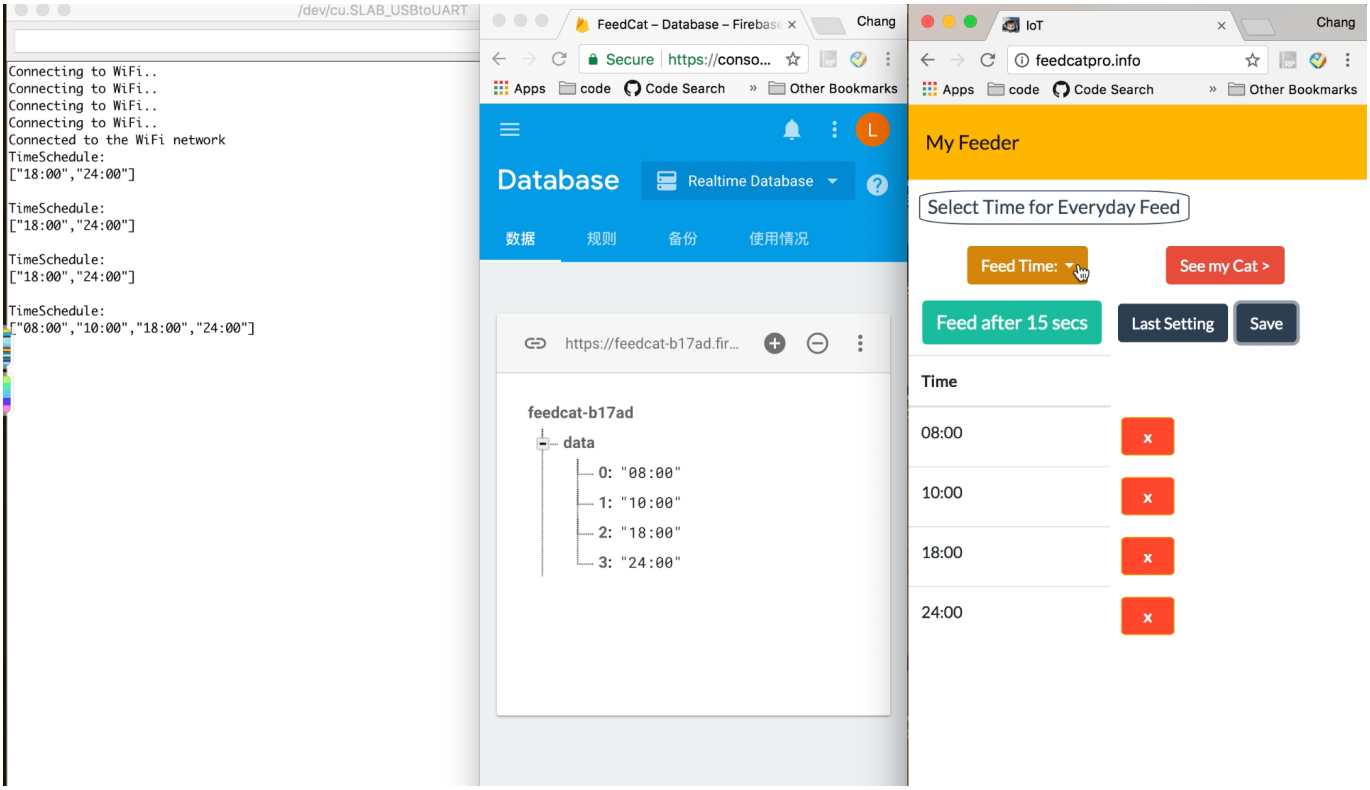

Figure8. Real-time data transfer between controller and website

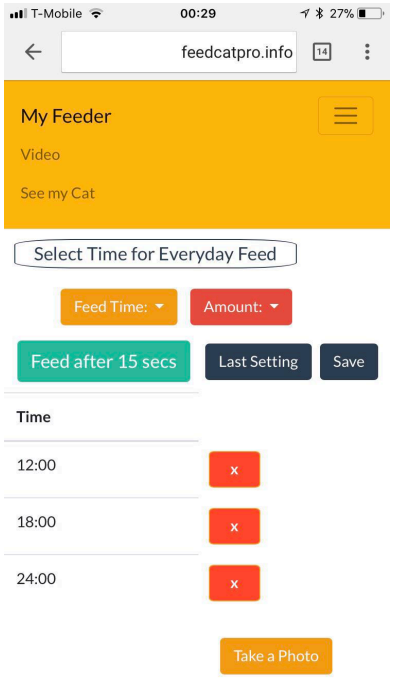

Figure9. User Interface (Mobile)

D. User interface

As shown in Figure9, the whole theme of our UI uses the orange color which matches with our team name. Our user interface is a responsive web design in Angular 5 using Bootstrap 4 and media queries of CSS3, so it can be viewed in any size of desktops, tablets, and phones. The client part uses HTML5 and CSS3, mixed with JQuery and TypeScript for UI design and data transfer, and the navigation bar with animation including popping and scrolling uses JQuery libs.

E. User testing

Since the cat feeder is a user-centered interaction design. We need to involve users in our testing procedures. After we performed the integration testing, the cat feeder had all the functionalities ready for the final user test. There are two users involved: cat owner that controls the feeder and a cat. We obtained the permission to test the product on our friend's cat named JJ. Ivy set a quick timer for 15 seconds and food was successfully dispensed and dropped in the bowl. Fortunately, JJ was appealed by the new design and was willing to test it. However, problems always occur when the final user is involved. According to the Figure 10, we found the gap between the dispenser and the bowl turned out to be too small for the head of a grownup cat. It might work for a kitten but certainly not cats like JJ. Beside this issue, other functionalities all worked as expected.

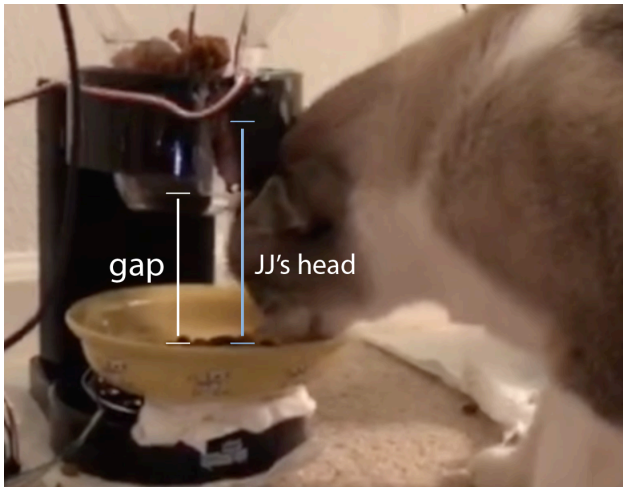

Figure10. User testing. JJ is trying to eat food but the gap between the food dispenser and bowl is too narrow for his head. This is a design issue that need to be fixed

.

IV. RESULT

We used the website as a remote controller for both servo and camera, and defined different parameters and time-schedule JSON as a controller of servo and camera. During the demo, we set feed time with different feed amount and also made an instant feed by click button on the website. We also show our capture function which is implemented by the nodejs web server. When the button was clicked, the corresponding parameter will change on the website, transfer the parameter to microcontrollers by its constant GET request, and the corresponding camera capturing function was called and worked.

However, our weight sensor and our load cell did not work stably. sometimes the result is accurate while other times it would jump to a very large number. Therefore, we decided not to use the weight sensor during the demo. It will be implement in our future scope.

V. CONCLUSION

A. Findings

This project helped us to learn each detailed steps of making an interesting IoT project, We learned how to program ESP8266 and ESP32 with Arduino IDE, and how to control devices using digital and analog outputs. We also learned a lot of new software skills such as

- building web app features and adding user interactivity to our project using Angular 5 and Firebase
- creating web applications that run on mobile and the web with the latest technologies like NgRx which is a Redux-inspired state management solution that can greatly enhance our Angular app.

We try many new things in Angular 5 like Datepickers, Sidenavs or the Material Data Table (including sorting, filtering and live updating). A real-time database connection powered by Firebase using Firestore and Angularfire makes our project more robust and fast. We also had a better understanding of RxJS observables and state-of-the-art state management with the help of NgRx.

We explored on both the hardware and the software side of things, and was able to connect to the internet and create spectacular devices and solutions. This project also improveed our cooperation skills with teammates including problem analyzing, features prioritizing, group meetings and discussion, work distributions, and so on. In conclusion, we really enjoyed this project and saw our potentials in adapting new technologies.

B. Future scope

After our visit to JJ's house. His owner thought our project could really benefit his life and would like us to continue to complete the product so he can use it in the future. We were very glad and want to add the following additional features to our cat feeder.

1) Add weight sensor to monitor the daily consumptions for food.

As mentioned before, our group attempted to add the load cell sensor to the IoT project. However, due to the wrong assembly of the load cell, we were not able to get stable weighing data from the sensor each time we tried. According to the suggestion from Professor, we found there are other kinds of pressure sensors with high precision and easy to install. With the weight sensing enabled, the system will record the weight reduction of the bowl on daily basis, save the data to the database, and display the food consumption on current day by calculating the weight decrement.

2) Enable video stream and video recording

With our current design, the user can check the food amount by capture an instant photo through a button click. We want to replace the capture by video streaming so the user can directly see the food amount without any clicking event. Also, the system will be able to record videos when the cat is eating, which can be detected by the weight changes. The video will then upload to the cloud and save as an mp4 file. So the cat owner will be able to
monitor his/her cat and its eating habits.

VI. LESSON LEARNED

This Web Infrastructure course introduces the infrastructure and various web-related technologies. Different from the typical lecturing and testing class, this course aims to provide students with broad understanding of different components that form what we know as the internet, IoT and WoT by providing many hands-on experiences in related technologies including embedded (controllable physical environment), mobile (mobile/web to interact with the user), and cloud (for saving data). We have learned so much from this project in

many aspects.

A. Learned the IoT building steps

By doing this group project, we have gained more knowledge and practice of how to build an IoT project from scratch, such as registering domain, setting up Amazon AWS servers, understanding microcontrollers and different sensors, programming in Arduino IDE, setting up access point and web proxy, connecting software with hardware to form a complete working piece, and working as a group. Each step was all challenging for us because these are new concepts to us, but all this knowledge greatly prepared us to have necessary initial needs in joining the industry.

B. Prioritize features

Before building the project, our group came up with a number of ideas and features to the cat feeder including the motion sensor, video streaming, voice transmission, etc. Nevertheless, we found things were not as easy as we thought because we never played with the hardware, it ended up taking us a period of time to just make one sensor working. We learned it is necessary to make a priority list of features we want to implement and finish the most important one first.

C. Get user involved

As we know, during the product development process, engineers need the constant connection with the customer because we need to know their preference as well as how they use this product. The user of our cat feeder involves both pet owners and the cat. We paid too much attention to the owner-related functionalities and neglect the cat's experience while using the automatic feeder. As seen from our user testing section, our group didn't keep enough distance between the dispenser and the bowl so it would be difficult for big cats, like JJ, to squeeze their heads between the gap. We are glad we get the cat involved in the design testing so we know what to improve in the next step.

D. Ask when you need help

Since we did not have prior knowledge about hardware and C++ language, it took us a long time to do the online researching, which can sometimes become inefficient. Luckily, we have a lot of good resources around us. For example, we got help from the professor regarding hardware punches options. One classmate helped us with C++ languages and another friend in Mechanical Engineering department helped us to assemble the load cell with screws. It really saved a lot of time by asking people who are experts in the area we are not familiar with. It is a smart way to improve the project efficiency.

E. Appreciate every technology around us

Nowadays, high-tech inventions seem pretty much sunk in our daily lives. For instance, we use Google to find answers to all the problem, connect with people from thousand miles away, use GPS to guide us to anywhere, enjoy digital entertainments, and so much more. It seems that people live in the 21st century take these great life experience for granted. Before doing the project, we would not be surprised for an automatic pet feeder or a fingerprint door lock because these inventions are so ubiquitous in our daily life. We would have never experienced the challenges of building even a small IoT project if we did not implement this project. Johannes Parvus, a 12th-century theologian and author, used a version of phrase in a treatise on logic called Metalogicon, written in Latin in 1159. "We are like dwarfs sitting on the shoulders of giants. We see more, and things that are more distant, than they did, not because our sight is superior or because we are taller than they, but because all they raise us up, and by their great stature add to ours"[8]. The message is to appreciate any discoveries, insights, or progress due to previous work of great minds that have come before. We are standing on the shoulders of predecessors who spent lifetime testing and inventing the technologies that benefit our current life. So be grateful for and treasure every beautiful invention in our lives, and work hard to create a better environment for our descendants.

VII. References

[1] Adafruit HUZZAH32 - ESP32 Feather. (n.d.). Retrieved March 23, 2018, from https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/overview

[2] Adafruit Industries. (n.d.). Adafruit Feather HUZZAH with ESP8266. Retrieved March 23, 2018, from https://www.adafruit.com/product/2821

[3] Arduino IDE. (n.d.). Retrieved March 23, 2018, from https://www.arduino.cc/en/Main/OldSoftwareReleases

[4] Benefits of an Automatic Pet Feeder. (n.d.). Retrieved March 23, 2018, from https://www.petsafe.net/learn/benefits-of-an-automatic-pet-feeder

[5] Craine, J. (2008). Virtualizing Los Angeles: Pierre Levy, The Shield, and http://theshieldrap.proboards45.com/. GeoJournal, 74(3), 235-243. doi:10.1007/s10708-008-9225-x

[6] Firebase. (n.d.). Retrieved March 23, 2018, from https://firebase.google.com

[7] Kunz, P. (n.d.). Subscription. Retrieved March 23, 2018, from https://ercim-news.ercim.eu/en101/keynote/1605-the-internet-of-things-will-change-our-world

[8] Martin, G. (n.d.). 'Standing on the shoulders of giants' - the meaning and origin of this phrase. Retrieved March 23, 2018, from https://www.phrases.org.uk/meanings/268025.html

[9] Mobile Apps & IoT. (n.d.). Retrieved March 23, 2018, from https://www.devmynd.com/custom-development/mobile-apps-and-iot/

[10] WebStorm IDE. (n.d.). Retrieved March 23, 2018, from https://www.jetbrains.com/webstorm/

[11] What is serial peripheral interface (SPI)? - Definition from WhatIs.com. (n.d.). Retrieved March 23, 2018, from http://whatis.techtarget.com/definition/serial-peripheral-interface-SPI