Shuyao Li

Professor Shirley Chiang

First Year Seminar-Join the Maker Revolution

June 14th, 2018

Remote Controlled Cat Feeder

1. **Background**

Cats are cute, but feeding them can be annoying sometimes, especially when you have to go back home to feed them and miss a review session. In order to feed the cat anytime in anywhere with internet connection, me and my partner, Ruize Tao, planned to make a cat feeder that can be controlled remotely. We found similar projects on hackster.in (Arduino\_Genuino, 2017) and also on YouTube (Applied Science, 2011), which helped us a lot.

1. **Overview**

**Basic Functions**

The ideal final project should be able to

1. receive signal through WiFi;
2. play music to attract the cat;
3. dispense a certain portion of cat food with a servo.

**Main Hardware components**

1. A cereal dispenser (bought on Amazon) and miscellaneous connecting parts (3D printed at home);
2. A ESP8266 NODEMCU board;
3. A 3.3V 9g micro servo;
4. A buzzer.

1. **Detailed Process**

We divided the project into parts by functions and started from the basic ones.

**Simplified working logic**

Loop{

if\_signal(ture){

music();

moveServo();

}

}

1. **Choosing a board**

Our project is supposed to be controlled remotely. We considered several methods including infrared, bluetooth, and WiFi. The first two can only cover a small area ,while the last one can receive signal from anywhere, although it requires additional knowledge in connecting to local WiFi, using API of Telegram (an instant messaging and voice service) , if we want to imitate what “Pavlov’s cat” did (Arduino\_Genuino, 2017), and reading signals. As the class provided ESP266 WiFi boards, which has built-in WiFi module and is smaller and lighter than the Arduino Uno board we have, we chose it without hesitation. Then we tested it with a built-in program *blink.ino* and made sure that the board was in good condition.

1. **Dispensing cat food**

The most important goal is to feed the cat, so we started with building a servo motor that rotates certain degree each time (thus equal portions of food will be dispensed). Because the ESP266 board has 3.3V as the biggest voltage, and the 9V stepper motor did not work with a gadget that adjusting the voltage (at that time we did not know GPIO numbers are the actual pin numbers), we continued the project with a 3.3V micro servo motor. We copied the code from “Pavlov’s cat”(Arduino\_Genuino, 2017) , cut out the if statement since we only need one portion option and also changed the rotation instruction by changing the digree. Unfortunately, it didn’t work either.

Weirdly, after we reconnected the servo motor to our Arduino Uno board and repeated the test,the motor functioned well. So we kept the Uno board to continued compiling other parts. Since there wasn’t built-in WiFi, I suggested using delay() in loop to control the time interval of feeding. Firstly we tried delay(32400000)//9 hours, but the servo motor did not work at all while the hardware were good. Then we reduced the time interval to 8 hours (delay(28800000)), which somehow worked.

Late till we receive the email that points out the correct pin number we should use for ESP8266 we finally switched back to our original board.

1. **WiFi control**

We used the code “ESP8266\_HTTPGetRequests.ino” provided on canvas and connected to the WiFi specified in email.

This part is mainly conducted by my partner Ruize. He spent a long time trying to figure out while the TelegramBot code provided by “Pavlov’s cat”(Arduino\_Genuino, 2017) doesn’t work in our project since some header files required are missing. Finally we went with another method provided by “Pavlov’s cat”, which is using a HTTP server. This makes the controlling process as simple as opening a website on the phone.

1. **Playing some music**

While my partner was struggling with WiFi control, I got to play with the buzzer. I have two buzzers, one is self drive and the other is external drive. The self drive one starts beeping once it is connected to the circuit so I have to use the external drive one.

The code controlling buzzer was adapted from “Song1.ino”(Chiang, 2018). In the original code, it takes multiple steps to change a melody since the range of frequency, length of beats and notes needs to be counted manually and updated in according for loops. If the numbers are not consistent, the buzzer will keep beeping and never goes into the next step (move servo).

To solve the problem, I cut off this process by using sizeof(frequencies)/

sizeof(int) to represent size of the arrays. Thus it becomes easier to add more notes from pitch.h (Chiang, 2018) and to change melody freely.

1. **Power source**

I have a GP 9V battery buckle with DC power plug so we tried if we can power the machine with a 9V battery. However, we found that we did not have enough knowledge to determine what resistors we should use. So we go with the easy way --- using a USB adapter.

1. **Assembly**

Ruize has a small 3D printer at home and he has learned to draw 3D model before. The connecting parts were all his work. A box-like part with a groove on it was used to connect the servo motor’s arm and the handle of cereal dispenser. Other miscellaneous parts made sure that the main body of the motor is fixed by attaching it to the base of dispenser. The arduino board was loosely sticked to the inner wall of the foundation, which makes it possible to be taken out and to make some adjustment.

1. **Final Result and Problems**

In class representation we have shown that all the parts worked well. However, there is a practical problem remains that our micro servo motor is not powerful enough to rotate the dispense handle once the dispenser is filled over ¼ . Ruize suggested that a more powerful motor is needed. And I think that decreasing the moment of inertia of the handle mechanically is also viable.

The bad news is that our cat ran away from home during the final week and I am, personally, too sad to continue the project.

After all, I learned a lot from this project, from compiling to assembling physical model. Most importantly, I gained confidence from finishing a project and getting to see my idea becomes real machine. I interested to make more arduino projects by my own in the future.

References

Applied Science. (2011, July). Automatic cat feeder with stepper motor and ATMega

AVR. *YouTube.*Retrieved from https://www.youtube.com/watch?v=xrDOFbLw8

pU&feature=youtu.be

Arduino\_Genuino. (2017, December). Pavlov’s Cat. *hackster.io.* Retrieved from https://www.

hackster.io/arduino/pavlov-s-cat-7e6577

Chiang, S. (2018, May). ESP8266\_HTTPGetRequests.ino. Retrieved from https://canvas.uc

davis.edu/courses/223399/files/3617345/

Chiang, S. (2018, January). pitch.h. Retrieved from https://canvas.ucdavis.edu/files/3245627/

Chiang, S. (2018, January). Song1.ino. Retrieved from https://canvas.ucdavis.edu/courses/

223399/files/3245644/