Pet Feeder

CS 426 IoT Project Report

Selin Eylul Bilen
Binghamton University
sbilen1@binghamton.edu

ABSTRACT

IoT integration in any field can bring many benefits to our daily life. Controlling the gadgets remotely and automatically can ease human's life and helps saving time and effort. However, these benefits come with a price, IoT enabled gadgets are more expensive when they are compared to their simple versions. Most of the time it is cheaper to get the necessary devices and integrate them by yourself, with this way people can convert devices into an IoT devices with functionality.

KEYWORDS

IoT, Pet Feeder, Hardware, Software

I. INTRODUCTION

Financially DIY IoT integrations are affordable compared to buying one, with this way so many people can benefit from this technology.

What I did with using IoT technology is I designed a system for feed a pet and get the information of the percentage left in the container. The user can use this system from wherever they are, they don't need to be on the same WI-FI.

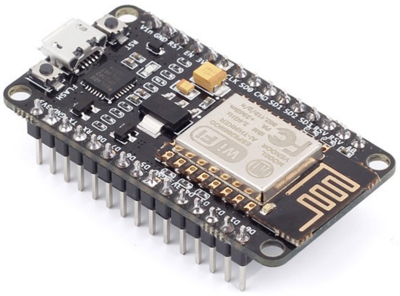
II. MOTIVATION

My main motivation when I was creating this project is to creating something useful and required for great amount of human. Nowadays most of the people wants a furry friend thus adopt a pet and nowadays most of the people spent their lives outside of their houses. In the meantime, their pets are staying alone at home, and sometimes they can't predict how long they going to stay outside thus these pets might waiting hungry. My project helps users to feed their pet whenever they want and wherever they are. Also, they can check if the container is empty or there is necessary amount of food.

III. HARDWARE

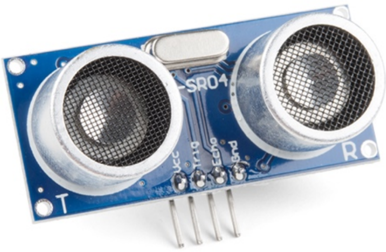
Besides from some software components I used such as Arduino IDE, Blynk, IFTTT, and Google Assistant. I used four hardware components;

1. Jumper Wires: I used jumper wires to connect servo motor to ESP8266 and ultrasonic sensor to ESP8266.
2. ESP8266 Nodemcu: A microcontroller that will connect to IFTTT servers via WI-FI.



3. Servo Motor: A simple motor that will open the gate for food to pour.

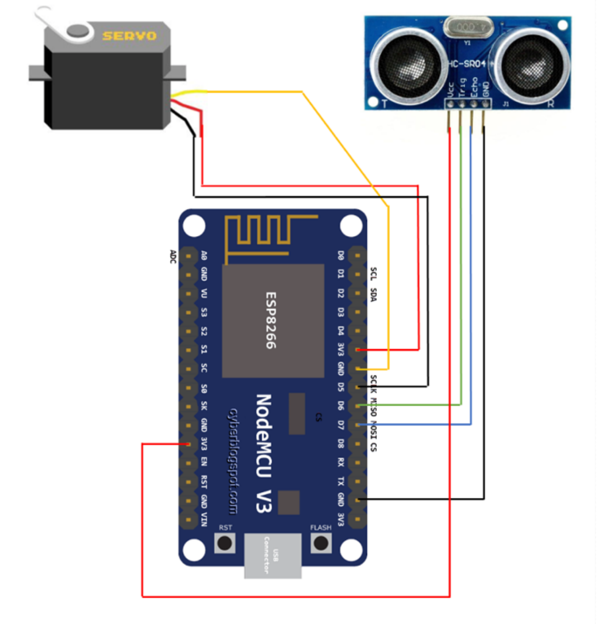
4. HCSR04 Ultrasonic Sensor: A sensor which calculates the food amount in the container.



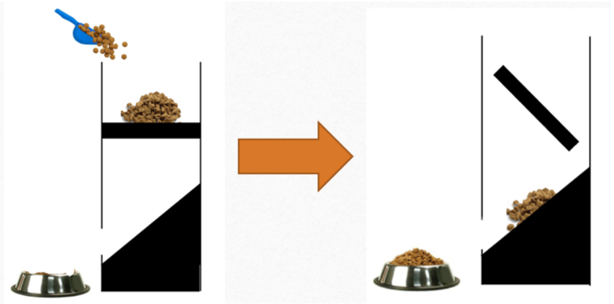
ESP8266 is acting as a main board thus sending and receiving data from the IFTTT server and Google Assistant to the Blynk App and controlling the servo motor and ultrasonic sensor. The ultrasonic sensor is responsible for calculating the percentage of the food in the container.

IV. Hardware Connection

I made a simple diagram that demonstrates my hardware connection. The ESP8266 is connected to the power supply via micro-USB cable. I connected my servo motor to the 3V3, GND and D5 ports of ESP8266 and connected my sensor to the 3V3, D6, D7 and GND ports of ESP8266. After connected the wires, I made a setup for my project.

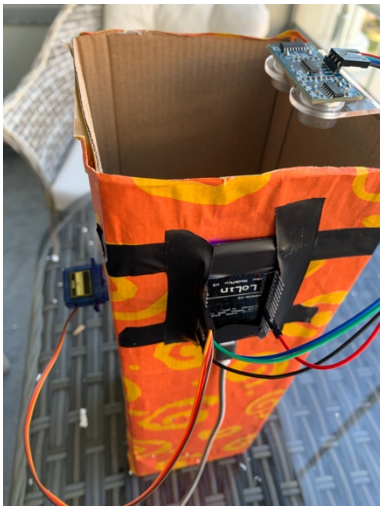

SERVO
HC-SR04
Vcc
Trig
Echo
GND
ADC
A0 GND VU S3 S2 S1 SC S0 SK GND 3V3 EN RST GND VIN
D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
SCL SDA SCLK MISO MOSI CS
ESP8266
NodeMCU V3
cyberblogspot.com
RST
FLASH
USB Connector

V. Design



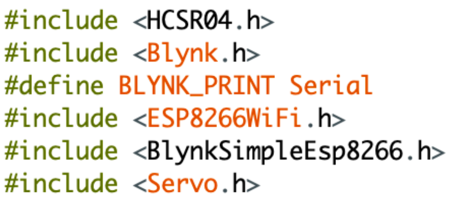
Before I build my setup, I designed setup which shown above. I divided the container with a one block in the middle, the block is controlled by the servo motor, when the servo motor spin the block opens up for 50-degrees and create enough space to food to pour.

Since the user will add the food from the top, I placed the ultrasonic sensor to the top of the setup. Therefore, the ultrasonic sensor will calculate the food percentage starting from the block in the middle to the top of the setup. I taped the ESP8266 to the back of the setup so the wires were reach to the necessary points and ESP8266 were not visible. The bottom I open a gate for the food to pour to the pet's food bowl.



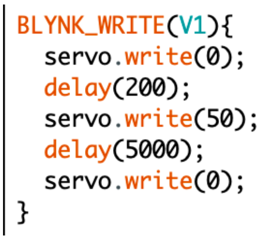
VI. Implementation

I used Arduino IDE for the code implementation. During the development I include some libraries shown below. These libraries are used to provide for hardware devices I used for my project also, it provides some libraries for Blynk App.



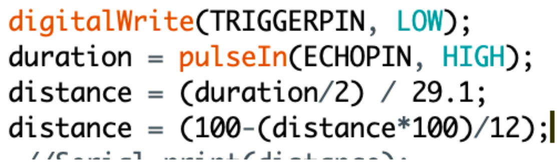
```
#include <HCSR04.h>
#include <Blynk.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
```

After I included the libraries, I set up Nodemcu, I created 8 virtual pins and attached them to the servo motor and ultrasonic sensor. I defined a function for servo motor to slide 50-degree angle and wait there for 5 seconds before close. The button which calls this function is attached to the V1 pin.



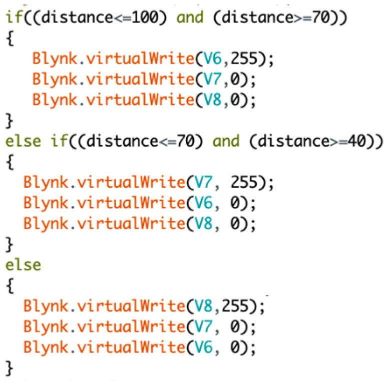
```
BLYNK_WRITE(V1){
  servo.write(0);
  delay(200);
  servo.write(50);
  delay(5000);
  servo.write(0);
}
```

I have a slider for servo motor which gives user to ability to setting delay time and degree manually. For the ultrasonic sensor I calculated the length of my container which is 12cm starting from the block to the top. So, I made a basic calculation for converting distance to the percentage.



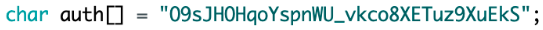
```
digitalWrite(TRIGGERPIN, LOW);
duration = pulseIn(ECHOPIN, HIGH);
distance = (duration/2) / 29.1;
distance = (100-(distance*100)/12);
```

I put 3 led lights in the Blynk app, led lights colors are red, orange and green. When the percentage is between 100 to 70 the green led light up, the percentage is between 70 to 40 the orange led lights up and the percentage is between 40 to 0 the red led lights up.

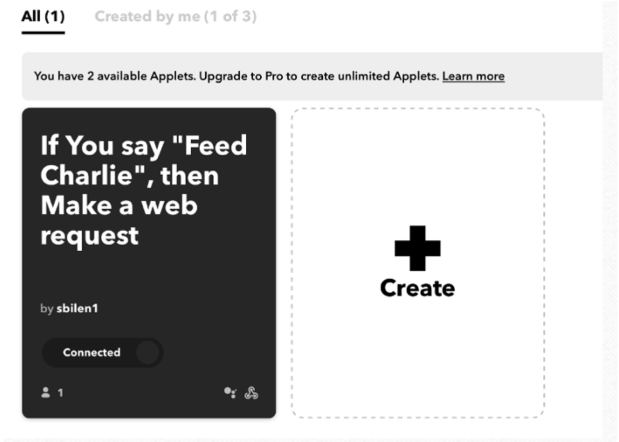


```
if((distance<=100) and (distance>=70))
{
   Blynk.virtualWrite(V6,255);
   Blynk.virtualWrite(V7,0);
   Blynk.virtualWrite(V8,0);
}
else if((distance<=70) and (distance>=40))
{
  Blynk.virtualWrite(V7, 255);
  Blynk.virtualWrite(V6, 0);
  Blynk.virtualWrite(V8, 0);
}
else
{
  Blynk.virtualWrite(V8,255);
  Blynk.virtualWrite(V7, 0);
  Blynk.virtualWrite(V6, 0);
}
```

To connect the IFTTT and Google Assistant I used an authentication code given by IFTTT.



```
char auth[] = "O9sJHOHqoYspnWU_vkco8XETuz9XuEkS";
```

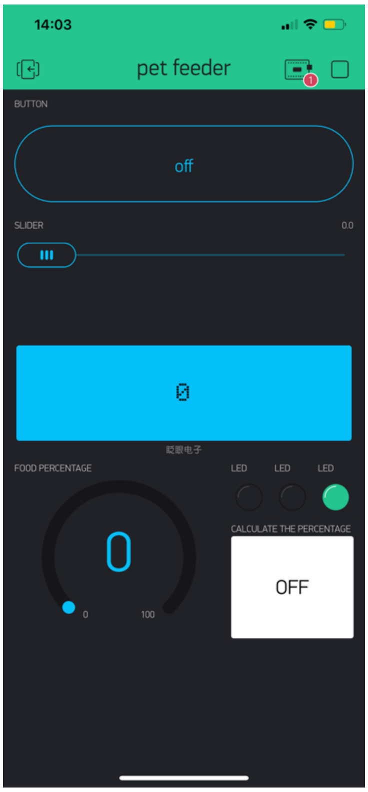


You have 2 available Applets. Upgrade to Pro to create unlimited Applets. Learn more

If You say "Feed Charlie", then Make a web request

by sbilen1

After setting up the hardware connections and completing the code I started working on the Blynk App. My Blynk application interface shown in the image. The button on the top is for servo motor to slide 50-degree angle and wait there for 5 seconds before close. The slider is for servo motor which gives user to ability to setting delay time and degree manually. The LCD shows the percentage of the food in the container and the graph below also shows the same value. There is a one button for starting ultrasonic sensor and there are three leds for lights up due to percentage of the food in the container.



VII. Usage

The usage scenario;

1. Open the Google Assistant and say the command for the request, or open the Blynk application and press the button.
2. Check to see if the food level is enough by looking at the LCD monitor or graph.

VIII.Limitation

One of the limitations of this project was to create a useful setup, it was hard to attach the servo motor and hard to close the gate when there are so many food loads. I used a tiny stick to hold it still but if the servo motor is not able to close the gate there are no ways to notify user.

The other limitation is to refill the food in the container, even though the app tells you the remaining food level, there is no way of refilling the food automatically. This means the application needs some form of human interaction and is not autonomous.

IX. Future Work

I completed everything that I planned for this project but I want to add more additional features to this project.

1. Do the same setup for water.
2. Calculate the food percentage in pet's food bowl.
3. When the food bowl is running out of food the servo motor will automatically runs and pours food.
4. Set a timer for servo motor to run in specific time zones.

CONCLUSION

This report shows how to connect Google Assistant, IFTTT, ESP8266, servo motor and ultrasonic sensor to create a DIY IoT enabled pet feeder. The main objective of this project is to show the practicality and usability of IoT solutions in a human's life. No need to spend lots of money to buy a gadget with IoT Technology. With the necessary equipment's and enough time everyone can make their own IoT enabled devices at home.

REFERENCES

[1] https://support.meshprj.com/hc/en-us/articles/213717818-How-to-create-an-IFTTT-recipe

[2] https://roboindia.com/tutorials/nodemcu-connecting-internet-arduino/

[3] https://www.instructables.com/Interfacing-Servo-Motor-With-NodeMCU/

[4] https://www.instructables.com/Ultrasonic-Sensor-With-Nodemcu/