AOUTOMATIC PET FEEDER

ICT 305 2.0

AS2019945

B.S.H.Perera

December 09, 2022

Background

Keeping pets takes many commitments. Mainly feeding them on time and in the correct way. However, not everyone is a pet expert, taking care of your pet's diet can be hard and time consuming. One of the top health concerns of pets are overeating and obesity. Especially at younger age, they are usually satisfied with whatever much is given to them. Another problem of feeding pets is that owners might not always be home regularly. Being occupied by personal plans knowing that they still have a starving little fellow at home to be taken care of is always a concern that bothers owners. Therefore, we want to take care of owners' concern of feeding by building a phone-controlled automatic pet feeder that can dispense the correct amount of food on time, based on the type of animal that's demanding it

Overview

Mainly this system helps you to feed your pet at any-time anywhere. It also can feed your pet at specific time according to their diet schedule. Also we can do all these in seconds using google assistant. Furthermore If your pet is hungry that he comes near the pet feeder it will notify you that your pet is hungry.

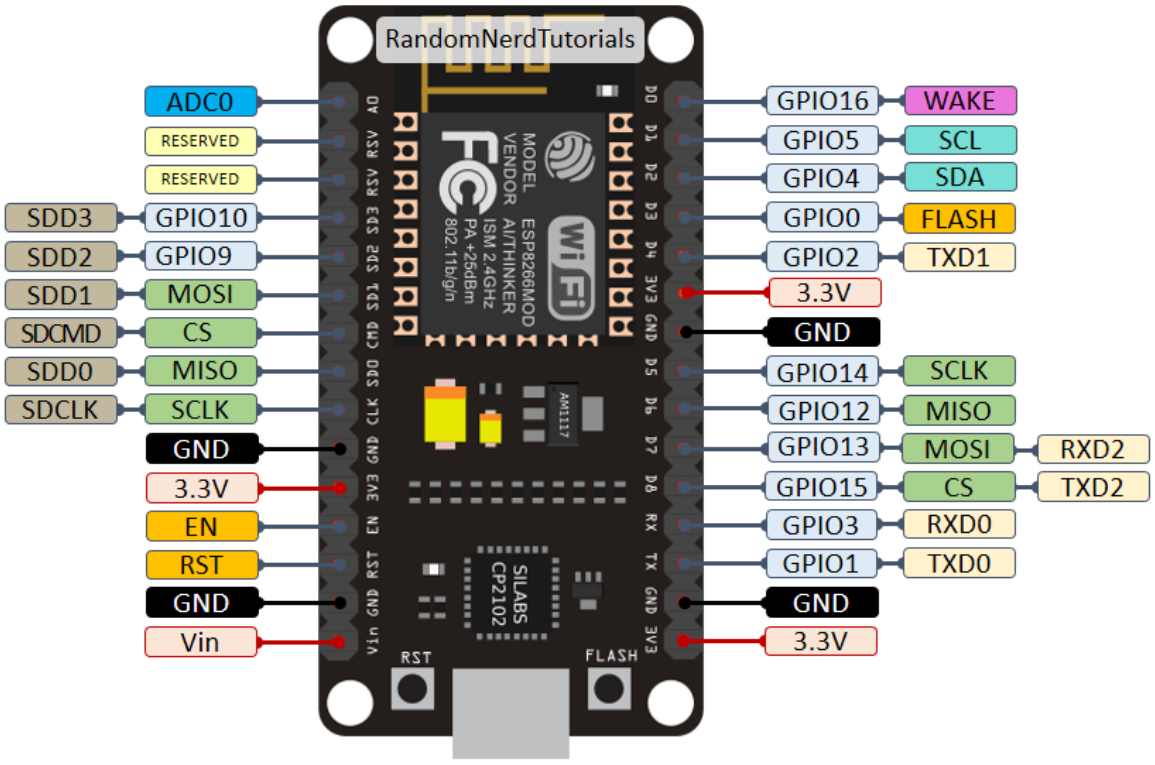
Problems Considered

- Over eating and obesity is a big health concern of pets
- Owners may not be regularly home to feed their pets and sometimes they forget

Components Needed

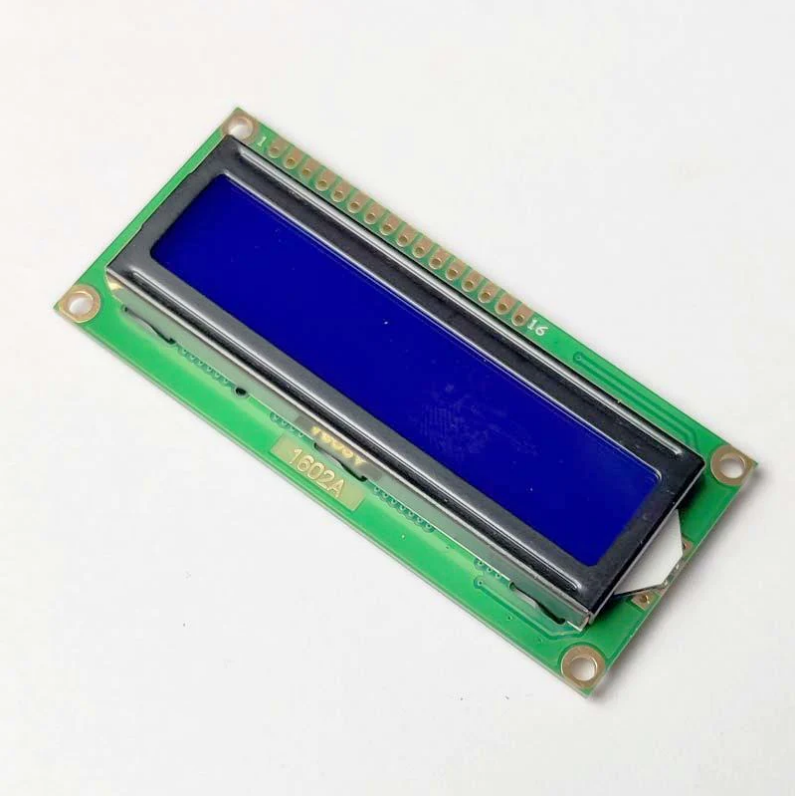
1.NodeMCU esp8266

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.



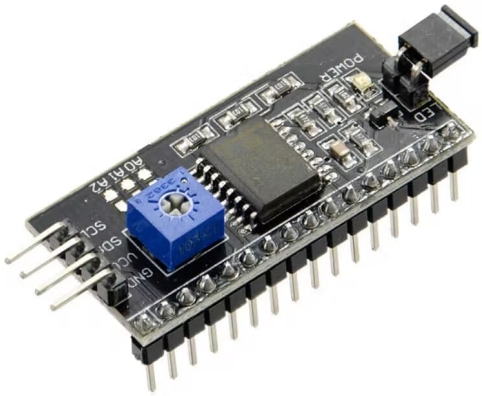
2. 16x2 LCD Module

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.



3. LCD I2C Module

I2C Module is a parallel to serial converter compatible with LCD2004 and LCD1602. By using this module, LCD can be interfaced with using only 2 wires.

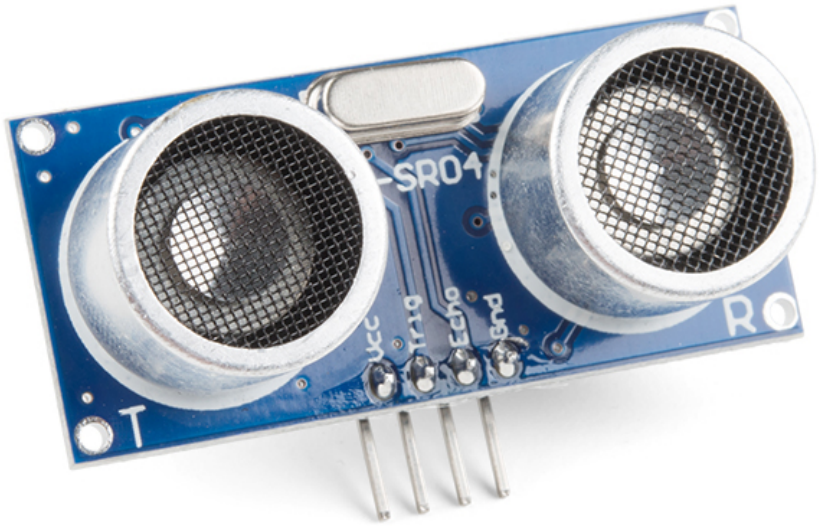


4. Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.

5. Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

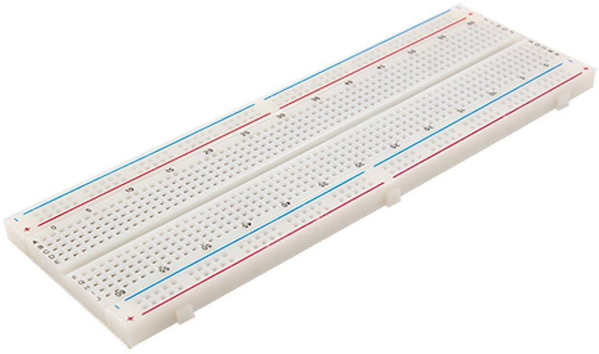


6. Jumper Wires

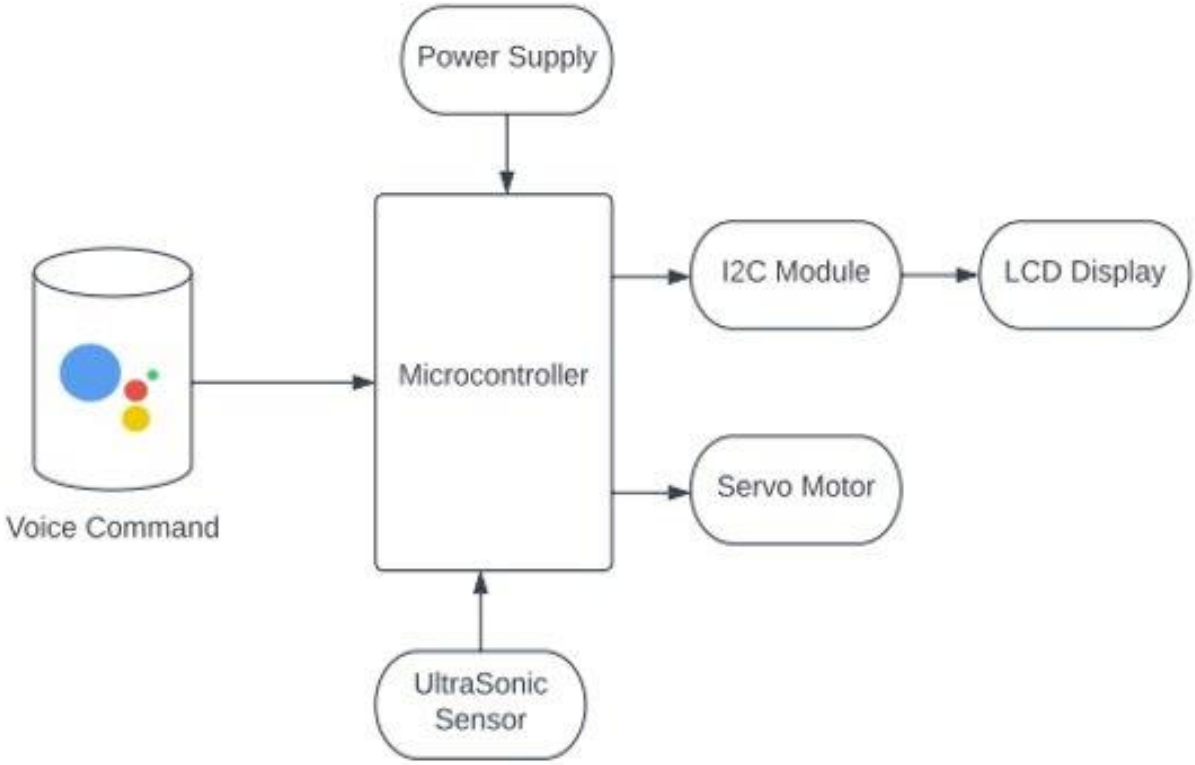
Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering.

7. Breadboard

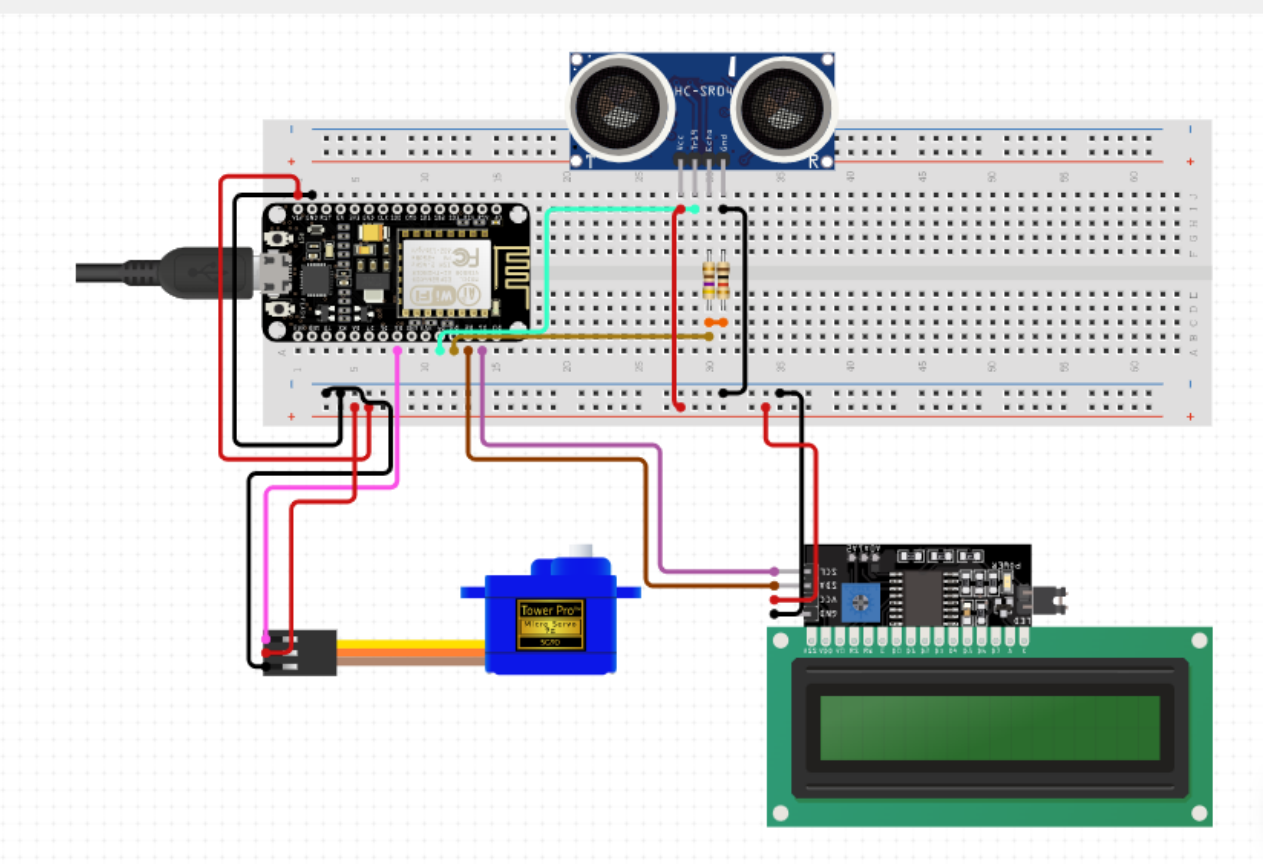
A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits.



Block Diagram



Circuit Diagram



Code

```
#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"

#include<Servo.h>
```

```
#include <NTPClient.h>

#include <WiFiUdp.h>

#include <LiquidCrystal_I2C.h>

#include <Wire.h>


WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800,60000);


Servo servo;

LiquidCrystal_I2C lcd(0x27 , 16, 2);


#define WIFI_SSID "Galaxy A10s1329"

#define WIFI_PASS "sandy123"

#define MQTT_SERV "io.adafruit.com"

#define MQTT_PORT 1883

#define MQTT_NAME "SandaminiPerera"

#define MQTT_PASS "aio_HkvB12psmF2Tp9jZ0dnLmrF20Vee"


int CLOSE_ANGLE = 0;  // The closing angle of the servo motor arm

int OPEN_ANGLE = 90;  // The opening angle of the servo motor arm

int  hh, mm, ss;
```

```
int feed_hour = 0;

int feed_minute = 0;


//Set up MQTT and WiFi clients

WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT, MQTT_NAME, MQTT_PASS);


//Set up the feed you're subscribing to

Adafruit_MQTT_Subscribe onoff = Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME
"/f/IOTPetFeeder");

boolean feed = true; // condition for alarm


const char* host = "maker.ifttt.com";

const int trigPin = 12;

const int echoPin = 14;

//define sound velocity in cm/uS

#define SOUND_VELOCITY 0.034

#define CM_TO_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;
```

```
void setup()
{
  Serial.begin(9600);
  timeClient.begin();
  Wire.begin(D2, D1);
  lcd.begin();

  //Connect to WiFi
  Serial.print("\n\nConnecting Wifi... ");
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
  }
  Serial.println("OK!");

  //Subscribe to the onoff feed
  mqtt.subscribe(&onoff);
  servo.attach(D3);
  servo.write(CLOSE_ANGLE);
```

```
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}

void loop() {
   MQTT_connect();
   timeClient.update();
   hh = timeClient.getHours();
   mm = timeClient.getMinutes();
   ss = timeClient.getSeconds();
   lcd.setCursor(0,0);
   lcd.print("Time:");
    if(hh>12)
  {
    hh=hh-12;
    lcd.print(hh);
    lcd.print(":");
    lcd.print(mm);
    lcd.print(":");
    lcd.print(ss);
    lcd.println(" PM  ");
```

```
}
 else
 {
  lcd.print(hh);
  lcd.print(":");
  lcd.print(mm);
  lcd.print(":");
  lcd.print(ss);
  lcd.println(" AM  ");
}
 lcd.setCursor(0,1);
 lcd.print("Feed Time:");
 lcd.print(feed_hour);
 lcd.print(':');
 lcd.print(feed_minute   );

Adafruit_MQTT_Subscribe * subscription;
while ((subscription = mqtt.readSubscription(5000)))
{
  if (subscription == &onoff)
  {
```

```
//Print the new value to the serial monitor
Serial.println((char*) onoff.lastread);
if (!strcmp((char*) onoff.lastread, "ON"))
{
 open_door();
  delay(1000);
  close_door();
 }
if (!strcmp((char*) onoff.lastread, "Morning"))
{
  feed_hour = 10;
  feed_minute = 30;
}
if (!strcmp((char*) onoff.lastread, "Afternoon"))
{
  feed_hour = 1;
  feed_minute = 15;
}
if (!strcmp((char*) onoff.lastread, "Evening"))
{
  feed_hour = 6;
```

```
      feed_minute = 30;
    }
   }
  }
  if( hh == feed_hour && mm == feed_minute &&feed==true) //Comparing the current time with the Feed time
  {
    open_door();
    delay(1000);
    close_door();
    feed= false;
    }

WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort))
    {
      Serial.println("connection failed");
      return;
    }
```

```
{       // Clears the trigPin
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
      // Sets the trigPin on HIGH state for 10 micro seconds
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);


 // Reads the echoPin, returns the sound wave travel time in microseconds
 duration = pulseIn(echoPin, HIGH);


 // Calculate the distance
 distanceCm = duration * SOUND_VELOCITY/2;


 // Convert to inches
 distanceInch = distanceCm * CM_TO_INCH;


 // Prints the distance on the Serial Monitor
 Serial.print("Distance (cm): ");
 Serial.println(distanceCm);
 Serial.print("Distance (inch): ");
```

```
 Serial.println(distanceInch);
 delay(1000);


      if (distanceInch <= 2.00 && distanceInch > 0.00)
     {
        String url = "/trigger/note/json/with/key/04ka1TR_re2AVEkxUMa05";
        Serial.print("Requesting URL: ");
        Serial.println(url);
        client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection:
close\r\n\r\n");
        delay(50000);
      }
       else
       {
        Serial.println("---------------------------------------------");
       }
}
}


void MQTT_connect()
{
 int8_t ret;
```

```
 // Stop if already connected.
 if (mqtt.connected())
 {
   return;
 }

 uint8_t retries = 3;
 while ((ret = mqtt.connect()) != 0) // connect will return 0 for connected
 {
     mqtt.disconnect();
     delay(5000);  // wait 5 seconds
     retries--;
     if (retries == 0)
     {
       // basically die and wait for WDT to reset me
       while (1);
     }
 }
}
```

```
void open_door(){
  servo.write(OPEN_ANGLE);   // Send the command to the servo motor to open the trap door
}
void close_door(){
  servo.write(CLOSE_ANGLE);   // Send te command to the servo motor to close the trap door
}
```

How it Works

The current time and Feed time will be displayed on the 16*2 LCD. Initially, it will show 0:0 in feed time as we didn't enter any feed time. Now Say "Okay Google. Activate pet Feeder" to your Google Assistant. Google Assistant will recognize the phrase and respond with "Activating pet Feeder." After that it rotates the servo motor from its initial position 60^{0} and after a delay, it returns to its initial position. You can also feed your pet at a specific time. For that we have used three strings i.e., "Morning", "Afternoon" and "Evening". These strings are assigned with different times, so when you say "Feed My Pet Today Morning" it will take the time assigned to "Morning" string as feeding time and when it matches with current time it rotates the servo motor. Also when your pet near the feeder UltraSonic sensor will detect it and sends notification to your mobile phone

Budget

Item	Price
NodeMCU esp8266	1490
16x2 LCD Module	1250
LCD I2C Module	470
Servo Motor	1000
Ultrasonic Sensor	450
Jumper wires	200
Breadboard	100
Total	4960

To Be Developed

This system can be further developed as we can change the door open angle according to the type of food (Solid and Liquid) and the animal (Dog , Cat , Bird ect...) to change the potion size.

References

https://www.youtube.com/

https://create.arduino.cc/projecthub