Title: Cat Feeder: An Automated Feeding System Utilizing Sonar and Weight Sensor Technologies

Course No: CSE 3104

Course Title: Peripherals and Interfacing Laboratory

Submitted by:

Jobair Al Nahian

Roll : 1907071

Kawsar Ahmed Kazol

Roll: 1907063

Submitted To:

Md Repon Islam	Md Badiuzzaman Shuvo
Lecturer	Lecturer
Department of Computer Science and Engineering	Department of Computer Science and Engineering
Khulna University of Engineering and Technology	Khulna University of Engineering and Technology

Khulna University of Engineering & Technology , Khulna

Objectives:

- This lab report describes the design, implementation, and evaluation of a cat feeder project utilizing sonar and weight sensor technologies.
- The objective of the project was to create an automated feeding system that can detect the presence of a cat using sonar, measure its weight using a weight sensor, and dispense an appropriate amount of food into a feeding pot using a servo motor.
- The project employed an Arduino microcontroller to control and coordinate the various components.

Project Intorduction:

Automatic pet feeders are becoming increasingly popular as they offer convenience to pet owners and help maintain a regular feeding schedule for pets.

This project aims to develop an automatic pet feeder using Arduino, servo motor, weight sensor, and sonar sensor. The servo motor controls the food dispensing mechanism, the weight sensor measures the weight of the cat to identify the cat, and the sonar sensor detects the presence of the pet near the feeder.

The structure is made for stable for the cat to use and we used card board for platform and wood for the main structure.

Theory:

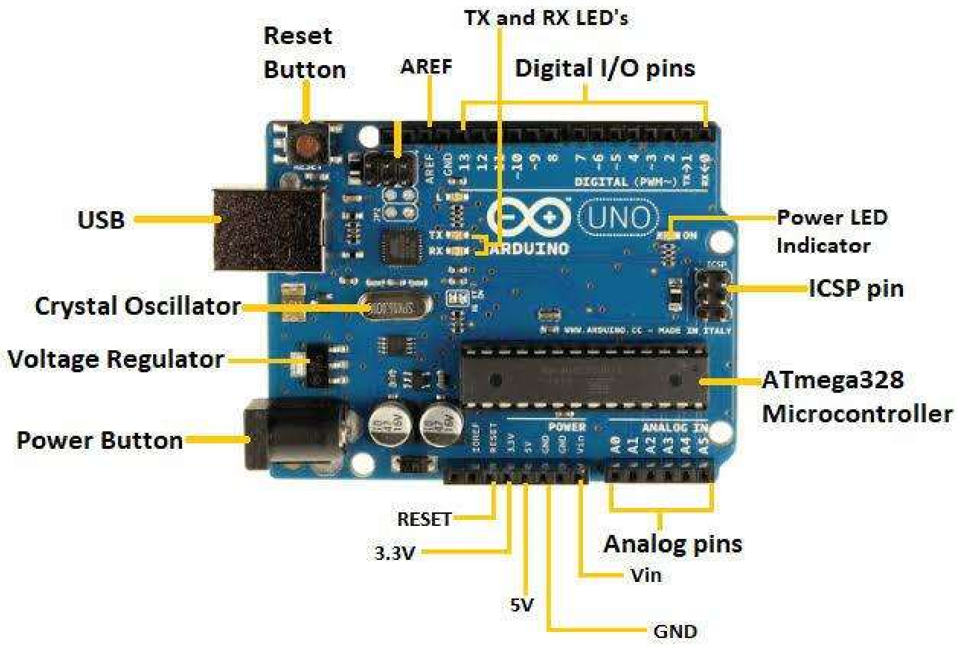


Figure 1.1 Arduino Uno

Arduino Uno:

Arduino Uno is based on the ATmega328P microcontroller and offers a wide range of input/output pins and features, making it suitable for beginners as well as experienced enthusiasts.
The Arduino Uno board comes with 14 digital input/output pins, where 6 can be used as
PWM (Pulse Width Modulation) outputs. It also has 6 analog input pins, a USB connection for programming and communication with a computer, and a power jack for external power supply. Additionally, it has a reset button for restarting the program execution.
One of the key advantages of Arduino Uno is its simplicity and ease of use. It has a user-friendly programming environment that allows you to write and upload code to the board using the Arduino Integrated Development Environment (IDE). The IDE supports a C/C++-based programming

language, making it accessible to beginners and experienced programmers alike.

Ultrasonic Sensor:

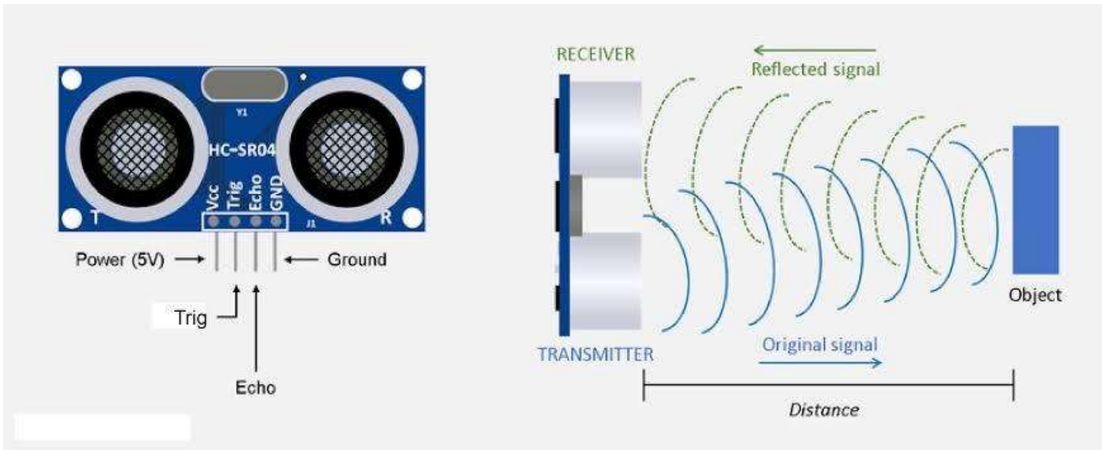


Figure 1.2 Ultrasonic Sensor

An ultrasonic sensor is a device that uses sound waves of high frequency (typically above 20 kHz) to detect and measure distances or detect the presence of objects. It emits ultrasonic waves and then listens for the echo produced when the waves bounce back after hitting an object. By measuring the time it takes for the echo to return, the sensor can calculate the distance to the object.

Ultrasonic sensors consist of a transmitter and a receiver. The transmitter emits ultrasonic waves, while the receiver detects the reflected waves. The sensor operates based on the principle of time-of-flight, where the time taken for the sound wave to travel to the object and back is measured.

If we need to measure the specific distance from your sensor, this can be calculated based on this formula:

Distance = ½ T x C

(T = Time and C = the speed of sound)

At 20°C (68°F), the speed of sound is 343 meters/second (1125 feet/second), but this varies depending on temperature and humidity.

Servo Motor:

Figure 1.3: Servo Motor

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor, a control circuit, and a feedback mechanism. Servo motors are commonly used in various applications that require accurate and controlled movement, such as robotics, industrial automation, CNC machines, and remote-controlled vehicles.

Weight Sensor:

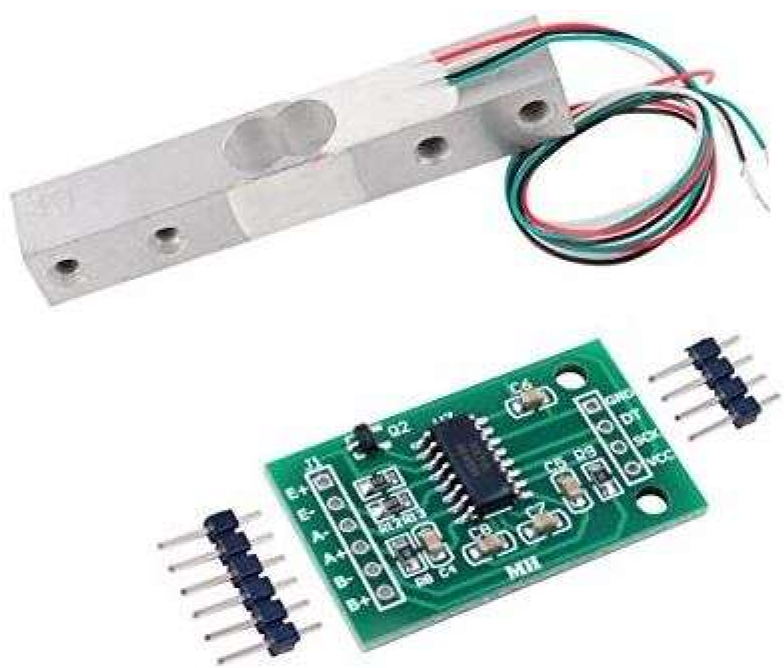


Figure 1.4: Weight sensor and module

Here are some key aspects of weight sensors and modules:

1. Load Cell: The load cell is the primary component of a weight sensor. It is typically made of metal and contains strain gauges or piezoelectric elements that change their electrical resistance or generate electrical signals in response to applied force or weight. The load cell's design depends on the specific application and the expected weight range to be measured.

2. Types of Load Cells: There are different types of load cells available, including:

   - Strain Gauge Load Cells: These load cells use strain gauges attached to the load cell body. The strain gauges change their resistance when subjected to force, and this change is measured to determine the weight.

   - Piezoelectric Load Cells: These load cells utilize piezoelectric materials that generate electrical signals when subjected to mechanical stress or force. The generated electrical signals are proportional to the applied weight.

   - Capacitive Load Cells: Capacitive load cells use changes in capacitance due to applied force to measure weight. They typically consist of two parallel plates separated by a small distance, and the applied force changes the distance between the plates, affecting the capacitance.

3. Weight Sensor Module: A weight sensor module typically includes the load cell and additional electronic components required to amplify and process the electrical signal from the load cell. The module may also include analog-to-digital converters (ADCs) to convert the analog signal into a digital format for further processing or interfacing with microcontrollers or other electronic systems.

4. Calibration: Weight sensors need to be calibrated to ensure accurate measurements. Calibration involves applying known weights to the sensor and adjusting the output signal to match the expected values. Calibration compensates for any variations in the load cell's performance and ensures accurate and reliable measurements.

Methods:

1. Hardware Setup:

- Connect the servo motor to the Arduino board using appropriate jumper wires.
- Connect the weight sensor to the Arduino board, ensuring proper wiring and calibration.
- Connect the sonar sensor to the Arduino board, following the manufacturer's instructions.

2. Software Implementation:

- Install the Arduino IDE and relevant libraries for servo motor control, weight sensor, and sonar sensor.
- Write the Arduino code to control the servo motor, read data from the weight sensor, and monitor the sonar sensor.
- Define appropriate thresholds for weight and sonar sensor readings to trigger feeding actions.
- Implement a feed+ing routine that dispenses a pre-defined amount of food when triggered.

3. Testing and Calibration:

- Verify that the servo motor moves correctly in response to feeding commands.
- Test the weight sensor's accuracy by measuring different amounts of food and comparing the readings.
- Experiment with the sonar sensor to ensure it accurately detects the presence of the pet.
- Fine-tune the threshold values based on the test results to optimize the system's performance.

4. Evaluation:

- Assess the accuracy of the automatic feeding process by comparing the dispensed food amount with the expected value.
- Evaluate the system's responsiveness to the pet's presence by observing its reaction to the sonar sensor.
- Assess the reliability and stability of the overall system during extended operation.

Circuit diagram:

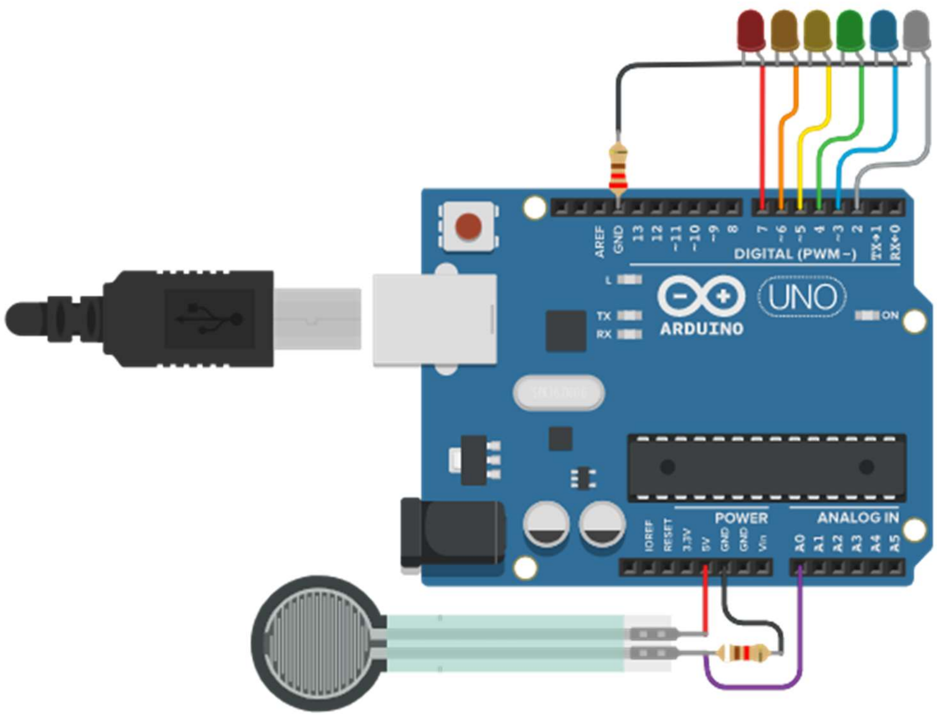


Figure 1.5: circuit diagram of weight sensor

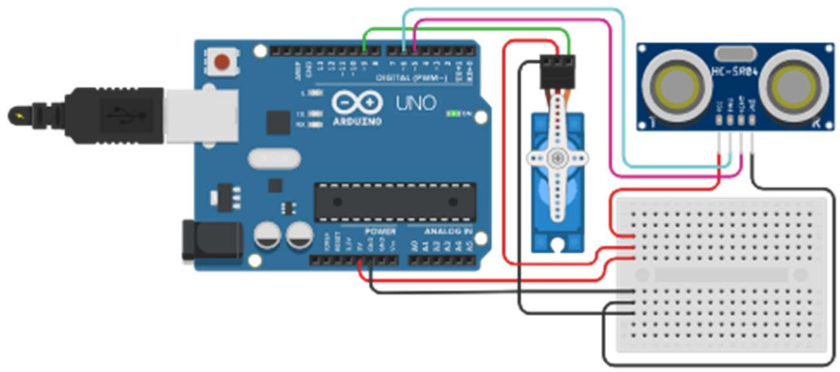


Figure 1.6: circuit diagram for sonar & servo motor

Project Code Implementation:

```
#include <Servo.h>
#include <HX711_ADC.h>
#include <EEPROM.h>
Servo myservo;
int trigPin1 = 5;
int echoPin1 = 6;
long duration1;
int distance1;
const int HX711_dout = 3;  //mcu > HX711 dout pin
const int HX711_sck = 2;   //mcu > HX711 sck pin
HX711_ADC LoadCell(HX711_dout, HX711_sck);
 void pingpong(){
   myservo.write(0);
```

```
digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = duration1 * 0.034 / 2;
Serial.print(distance1);
Serial.println(" cm ");
// if distance from either sensor is less than or equal to 5 cm
if (distance1 <= 20) {
  myservo.write(90);
  Serial.println("kkkk");  // rotate servo motor to 90 degrees
  delay(800);        // wait for the servo to move
  myservo.write(0);
  delay(100);
  while (distance1 <= 20) {  // while object is still in range of either sensor
    // read distance from first sensor
    digitalWrite(trigPin1, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1, LOW);
    duration1 = pulseIn(echoPin1, HIGH);
    distance1 = duration1 * 0.034 / 2;
```

```
      Serial.print(distance1);
      Serial.println(" cm ");
      delay(100);
   }
    // myservo.write(0);    // rotate servo motor back to original position
    delay(100);  // wait for the servo to move
  }
 }
void setup() {
  myservo.attach(12);  // attach servo motor to pin 10
  myservo.write(0);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
float calibrationValue = 96.06;
  Serial.begin(9600);
  LoadCell.begin(); // start connection to HX711
  LoadCell.start(2000); // load cells gets 2000ms of time to stabilize
  EEPROM.get(0, calibrationValue);
  LoadCell.setCalFactor(calibrationValue);
}
void loop() {
  // read distance from first sensor
  LoadCell.update(); // retrieves data from the load cell
  float i = LoadCell.getData(); // get output value
  Serial.print(i);
```

```
Serial.println(" g ");
  if(i>400.0){
  pingpong();
  }
}
```

Project Images:

Figure 1.7: Front View

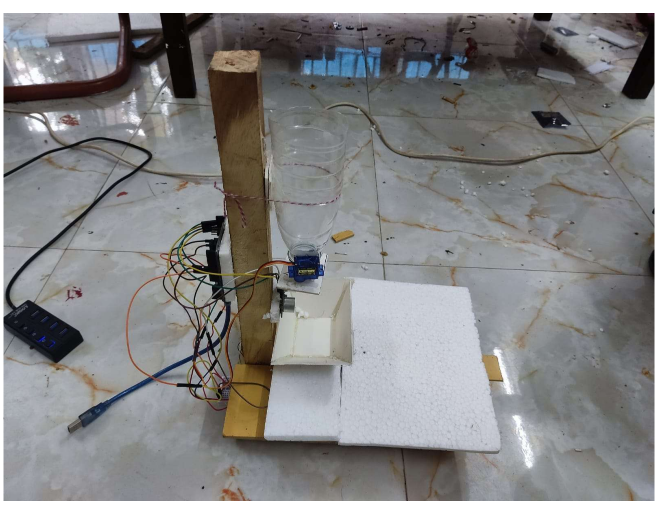
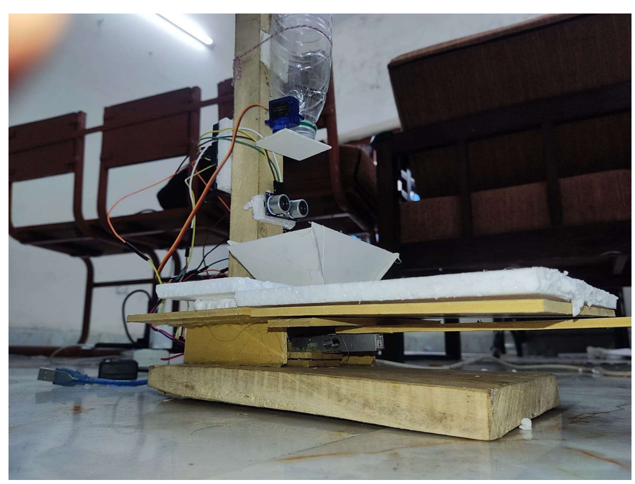


Figure 1.8 : Side View



2 Figure 1.9 : view with weight sensor

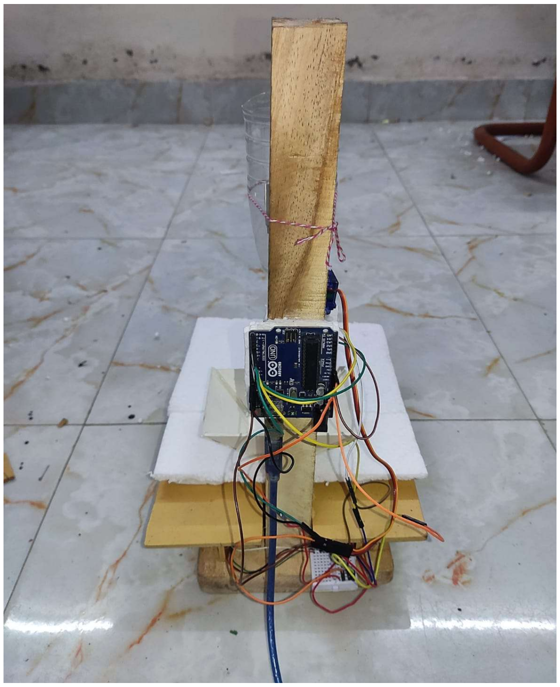


Figure 1.10 : Back View

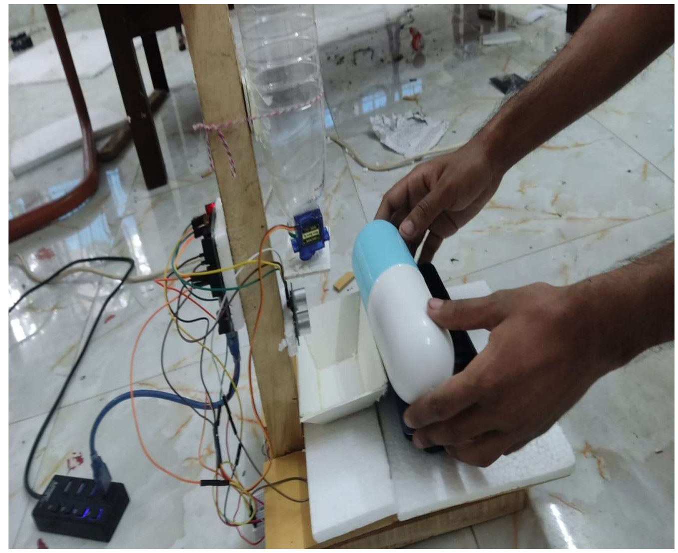


Figure 1.11 : with weight and sonar detecting object distance

Results:

The automatic pet feeder successfully dispenses food according to the predefined routine. The servo motor accurately controls the food dispensing mechanism, and the weight sensor provides precise measurements of the food in the bowl. The sonar sensor reliably detects the presence of the pet, triggering the feeding process as intended. The system operates reliably over an extended period, ensuring the pet is fed consistently and in the right amounts.

Application:

1. Automated Feeding: The Cat Feeder project automates the feeding process for cats, eliminating the need for manual feeding by pet owners.
2. Cat Detection: The sonar sensor used in the project detects the presence of a cat near the feeding pot, ensuring that meals are dispensed only when the cat is present.

3. Portion Control: The weight sensor integrated into the project measures the cat's weight accurately, allowing for precise portion control and preventing overeating or underfeeding.
4. Convenience for Pet Owners: The automated system offers convenience to pet owners by ensuring regular and timely feeding, even in their absence or busy schedules.
5. Pet Health and Well-being: By providing accurate portion control, the Cat Feeder system promotes a healthy weight and prevents obesity, enhancing the overall health and well-being of the cat.

Discussion:

The automatic pet feeder demonstrated reliable and accurate feeding functionality. The integration of the servo motor, weight sensor, and sonar sensor allowed for precise control over the feeding process. The system effectively addressed the challenges associated with automatic pet feeding, such as controlling food portions and detecting the pet's presence.

However, there are some limitations to consider. The accuracy of the weight sensor may be affected by external factors such as vibrations or the pet's movement near the bowl. Additionally, the sonar sensor's effectiveness may be hindered if the pet is positioned too close to the feeder, resulting in false triggers.

Possible improvements include implementing additional sensors for more precise detection of the pet's presence, incorporating a feedback mechanism to prevent overfeeding, and integrating wireless connectivity to enable remote monitoring and control.

Conclusion: The automatic pet feeder developed using Arduino, servo motor, weight sensor, and sonar sensor successfully automates.