CLASS 8

CONTENTS:

1. ARDIUNO- Servo motor (Theory)
2. ARDIUNO- Servo motor (Coding)
3. 3D- 123D Introduction
4. 3D- Chair Design
5. AI-Mad Libs Generator(Explanation)
6. AI-Mad Libs Generator(Practical)
7. ARDIUNO- Ultrasonic Sensor(Theory)
8. ARDIUNO- Ultrasonic Sensor(Coding)
9. 3D- Gear Design
10. 3D- Gear(Printing)
11. AI- TextBased Adventure Game(Explanation)
12. AI-4. TextBased Adventure Game(Practical)
13. ARDIUNO- Moisture Sensor(Theory)
14. ARDIUNO- Moisture Sensor(Coding)
15. 3D- Key Ring Design
16. 3D- Key Ring Print
17. AI- Guess the word(Explanation)
18. AI- Guess the word(Practical)

19. ARDIUNO- Temperature Sensor(Theory)
20. ARDIUNO- Temperature Sensor(Coding)
21. 3D- Nut Bolt Design
22. 3D- Nut Bolt Print
23. AI- Wheel of Python(Explanation)
24. AI- Wheel of Python(Practical)
25. ARDIUNO- Heart Beat Sensor(Theory)
26. ARDIUNO-Heart Beat Sensor(Coding)
27. 3D- House Design
28. 3D- House Print

Class 1 & 2: Arduino- Servo Motor

You can connect small servo motors directly to an Arduino to control the shaft position very precisely. Because servo motors use feedback to determine the position of the shaft, you can control that position very precisely. As a result, servo motors are used to control the position of objects, rotate objects, move legs, arms or hands of robots, move sensors etc. with high precision. Servo motors are small in size, and because they have built-in circuitry to control their movement, they can be connected directly to an Arduino.
Most servo motors have the following three connections:
•Black/Brown ground wire.

•Red power wire (around 5V).

•Yellow or White PWM wire.

In this experiment, we will connect the power and ground pins directly to the Arduino 5V and GND pins. The PWM input will be connected to one of the Arduino's digital output pins.

Hardware Required

•1 x TowerPro SG90 servo motor

•1 x Arduino Mega2560

•3 x jumper wires

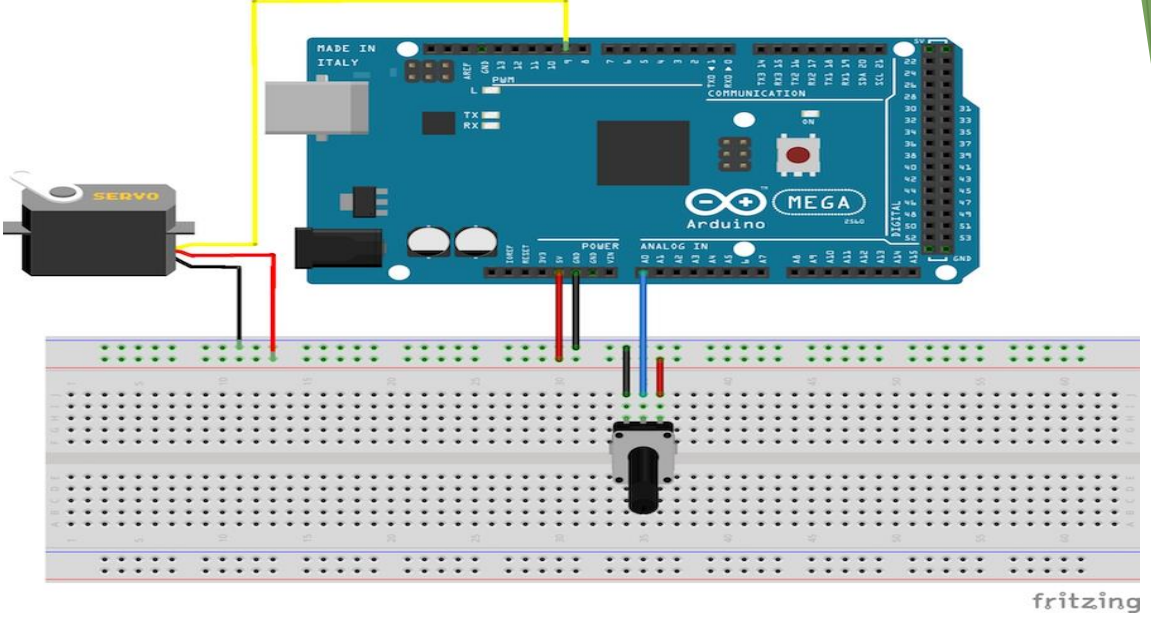
Wiring Diagram

The best thing about a servo motor is that it can be connected directly to an Arduino. Connect to the motor to the Arduino as shown in the table below:

•Servo red wire – 5V pin Arduino

•Servo brown wire – Ground pin Arduino

•Servo yellow wire – PWM(9) pin Arduino



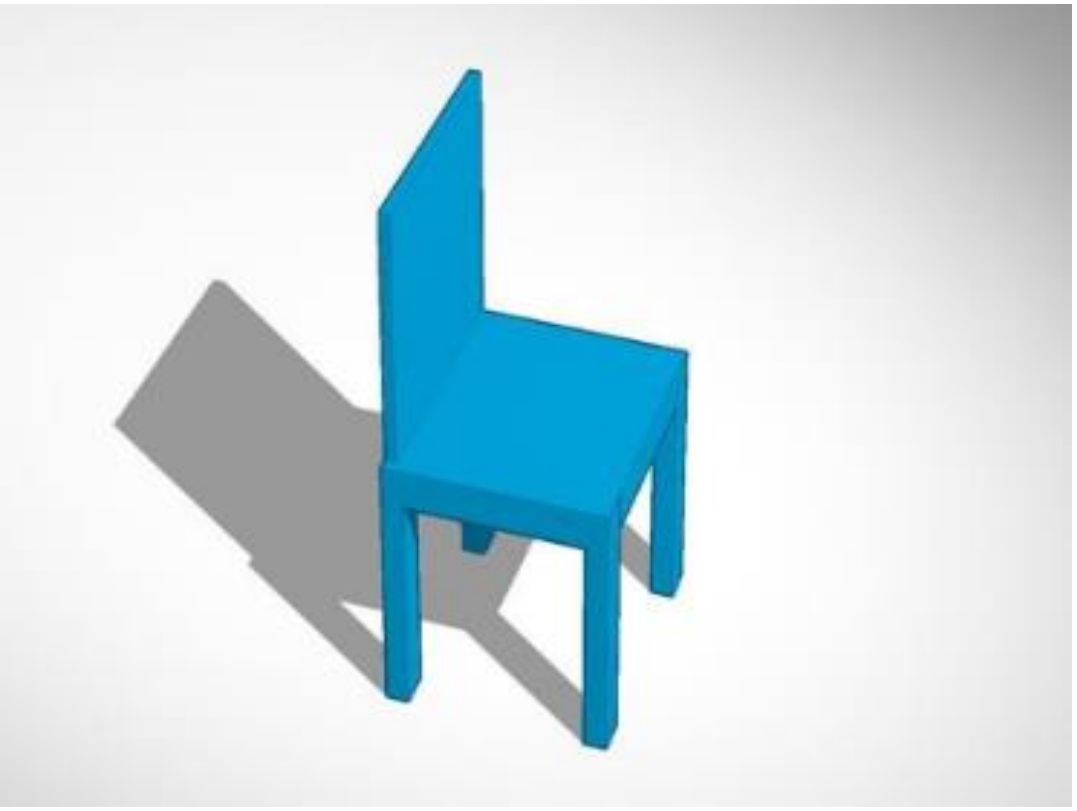

```
#include <Servo.h>

Servo servo_test;
int angle = 0;
void setup()
{
  servo_test.attach(9);
}
void loop()
{
  for(angle = 0; angle < 180; angle += 1)
  {
    servo_test.write(angle);
    delay(15);
  }
  delay(1000);
  for(angle = 180; angle>=1; angle-=5)
  {
    servo_test.write(angle);
    delay(5);
  }
    delay(1000);
}
```

Class 3: 3D- Introduction to 123D

Autodesk 123D was a suite of hobbyist CAD and 3D modelling tools created by Autodesk. It is similar in scope to Trimble SketchUp and is based on Autodesk Inventor. As well as the more basic drawing and modelling capabilities it also has assembly and constraint support and STL export. Available for the software is also a library of ready-made blocks and objects. Autodesk is also working in collaboration with three companies (Ponoko, Techshop and 3D Systems) to enable users of 123D to create physical objects from their designs using 3D printing technology.

All 123D apps were discontinued by Autodesk beginning November 2016 and completing March 2017. The tools in the 123D group were replaced by Tinkercad, Fusion 360, and ReMake. Only Autodesk 123D Circuits survived.

Class 4: 3D- Chair Design



Tools Used:

1.Different 3D Shapes
2.Rotation
3.Translation
4.Scale

Learning Outcome:

1. Understanding of 3D shapes, i.e. box, cylinder etc.
2. Understanding of proper measurement.
3. Understanding rotation and translation in 3D space in 123D design.

Class 5 & 6 : AI- Mad Libs Generator

The Goal: Inspired by Summer Son's Mad Libs project with Javascript. The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You'll need prompts for user input, and to then print out the full story at the end with the input included.

Concepts to keep in mind:

- Strings
- Variables
- Concatenation
- Print

A pretty fun beginning project that gets you thinking about how to manipulate userinputted data. Compared to the prior projects, this project focuses far more on strings and concatenating. Have some fun coming up with some wacky stories for this!

Example:

```
color = input("Enter a color: ")
pluralNoun = input("Enter a plural noun: ")
celebrity = input("Enter a celebrity: ")

print("Roses are", color)
print(pluralNoun + " are blue")
print("I love", celebrity)
```

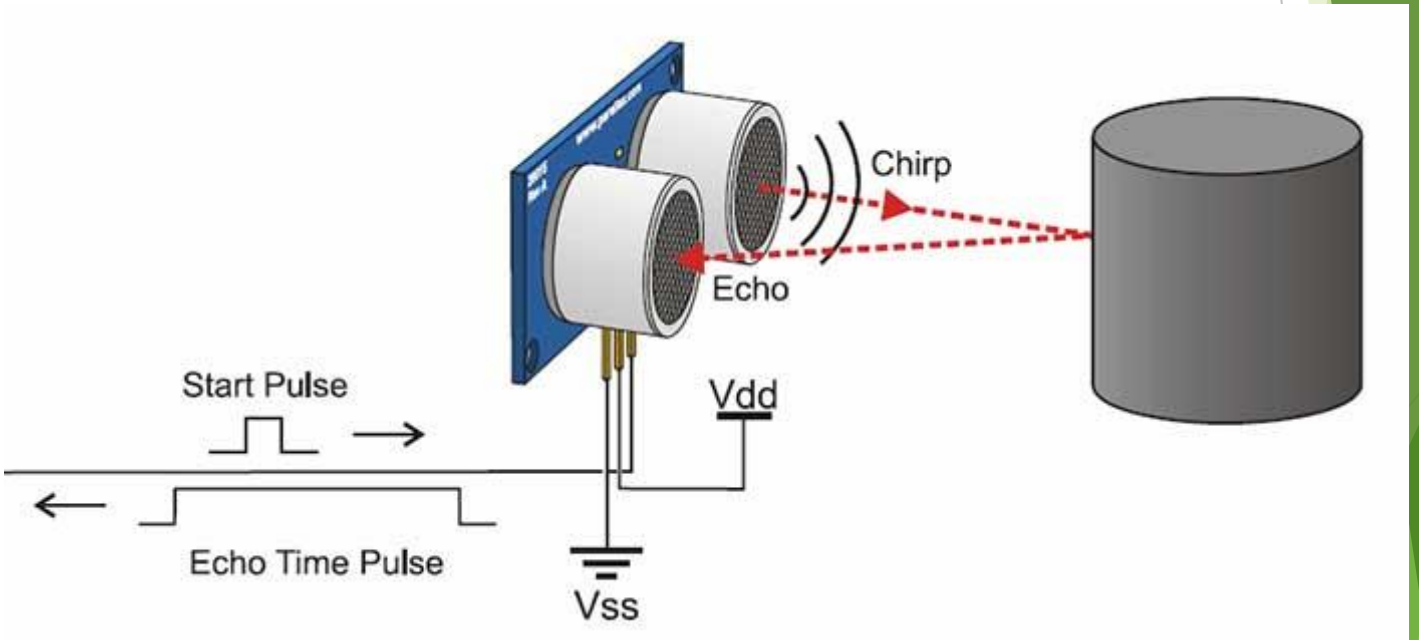
Class 7 & 8: Arduino- Ultrasonic Sensor

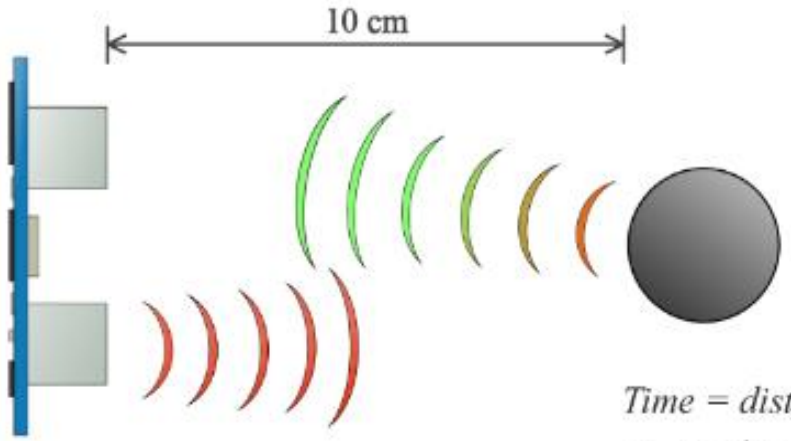
How It Works – Ultrasonic Sensor

It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.

In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.





speed of sound:

$v = 340\ m/s$

$v = 0{,}034\ cm/\mu s$

Time = distance / speed:

$t = s / v = 10 / 0{,}034 = 294\ \mu s$

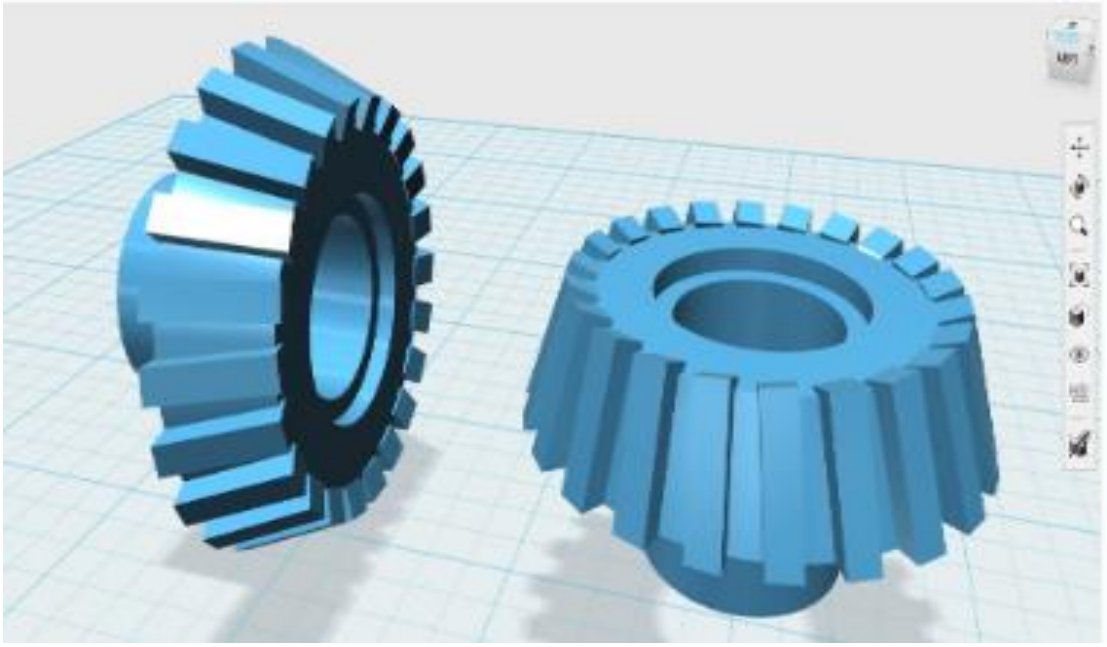
Distance:

$s = t \cdot 0{,}034 / 2$

First you have to define the Trig and Echo pins. In this case they are the pins number 9 and 10 on the Arduino Board and they are named trigPin and echoPin. Then you need a Long variable, named “duration” for the travel time that you will get from the sensor and an integer variable for the distance.

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;
Serial.print("Distance: ");
Serial.println(distance);
}
```

Class 9 & 10: Gear(Designing & printing)



Tools Used :

1. Different 3D shapes
2. Extrude
3. Circular Pattern tool
4. Subtract

Learning Outcome:

1. Getting to know various 3D shapes

2. Familiar variables and their use in 123D design

3. Using circular pattern tool, Subtract in 12D design

Class 11 & 12: AI- 4. TextBased Adventure Game (Explanation and practical)

The Goal: Remember *Adventure*? Well, we're going to build a more basic version of that. A complete text game, the program will let users move through rooms based on user input and get descriptions of each room. To create this, you'll need to establish the directions in which the user can move, a way to track how far the user has moved (and therefore which room he/she is in), and to print out a description. You'll also need to set limits for how far the user can move. In other words, create "walls" around the rooms that tell the user, "You can't move further in this direction."

Concepts to keep in mind:

- Strings
- Variables
- Input/Output
- If/Else Statements
- Print
- List
- Integers

The tricky parts here will involve setting up the directions and keeping track of just how far the user has “walked” in the game. This project also continues to build on using userinputted data. It can be a relatively basic game, but if you want to build this into a vast, complex word, the coding will get substantially harder, especially if you want your user to start interacting with actual objects within the game. That complexity could be great, if you’d like to make this into a longterm project.

Class 13 & 14: Arduino- Moisture Sensor

- **Operation**

The Soil Moisture Sensor measures soil moisture grace to the changes in electrical conductivity of the earth (soil resistance increases with drought).

The electrical resistance is measured between the two electrodes of the sensor.
A comparator activates a digital output when a adjustable threshold is exceeded.

3V --> VCC
GND --> GND
A0 --> A0

Code-

```
int sensorPin = A0;
int sensorValue;
int limit = 300;

void setup() {
 Serial.begin(9600);
 pinMode(13, OUTPUT);
}

void loop() {

 sensorValue =
analogRead(sensorPin);
 Serial.println("Analog Value : ");
 Serial.println(sensorValue);

 if (sensorValue<limit) {
 digitalWrite(13, HIGH);
 }
 else {
 digitalWrite(13, LOW);
 }

 delay(1000);
}
```

Class 15 & 16: 3D- Key Ring(Design and Print)

Tools Used :

1. Fillet
2. Sketch
3. Extrude
4. Subtract

Class 17: AI- Guess The Word(Explanation and Practical)

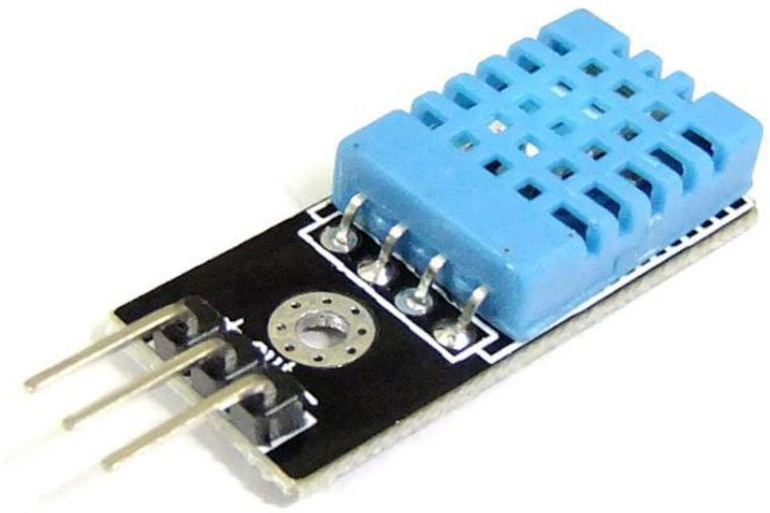
The Goal: The main goal here is to create a sort of “guess the word” game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you’ll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

Concepts to keep in mind:

- Random
- Variables
- Boolean
- Input and Output
- Integer
- Char
- String
- Length
- Print

Likely the most complex project on this list (well, depending on just how intense you went with the adventure text game), the Hangman project compiles the prior concepts and takes them a step further. Here, outcomes are not only determined based on user-inputted data, that data needs to be parsed through, compared, and then either accepted or rejected.

Class 19 & 20: Arduino-Temperature Sensor



Technical Specifications

- Calibrated directly in Celsius (Centigrade)
- Linear + 10-mV/°C scale factor
- 0.5°C ensured accuracy (at 25°C)
- Rated for full −55°C to 150°C range
- Suitable for remote applications

The Temperature Sensor LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature.
The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range.

Code to Note

LM35 sensor has three terminals - V_s, V_{out} and GND. We will connect the sensor as follows −

•Connect the $+V_s$ to +5v on your Arduino board.
•Connect V_{out} to Analog0 or A0 on Arduino board.
•Connect GND with GND on Arduino.

The Analog to Digital Converter (ADC) converts analog values into a digital approximation based on the formula ADC Value = sample * 1024 / reference voltage (+5v). So with a +5 volt reference, the digital approximation will be equal to input voltage * 205.

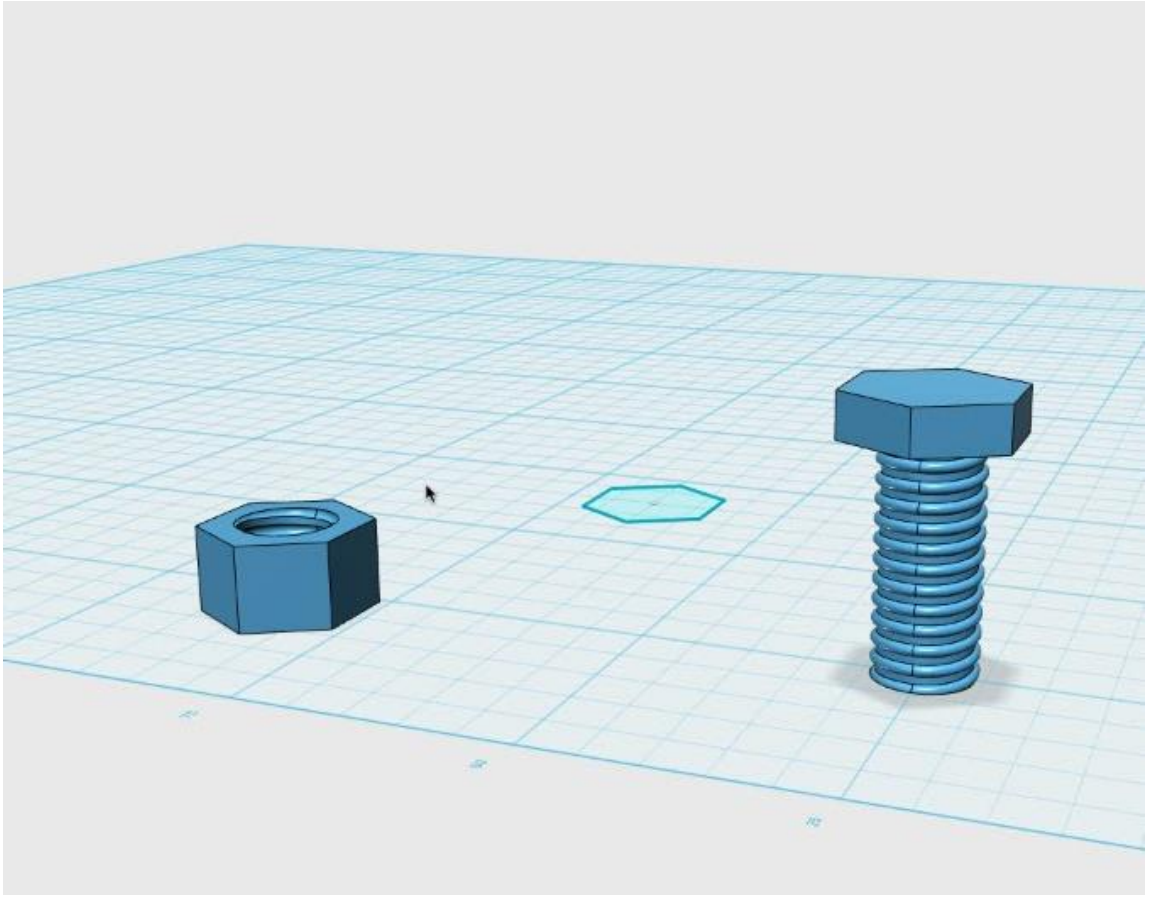
Result

You will see the temperature display on the serial port monitor which is updated every second.

Code:

```
float temp;
int tempPin = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  temp = analogRead(tempPin);

  temp = temp * 0.48828125;

  Serial.print("TEMPERATURE =
");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

Class 21 & 22: 3D- Nut Bolt(Design and Print)



Tools Used :

1. 2D Shapes
2. Extrude
3. Circular Pattern
4. Loft
5. Fillet

Learning Outcome:

1. Learning the concept of threads dimension in Nut bolts
2. Using Loft and Circular

Class 23& 24: AI- Wheel Of Python

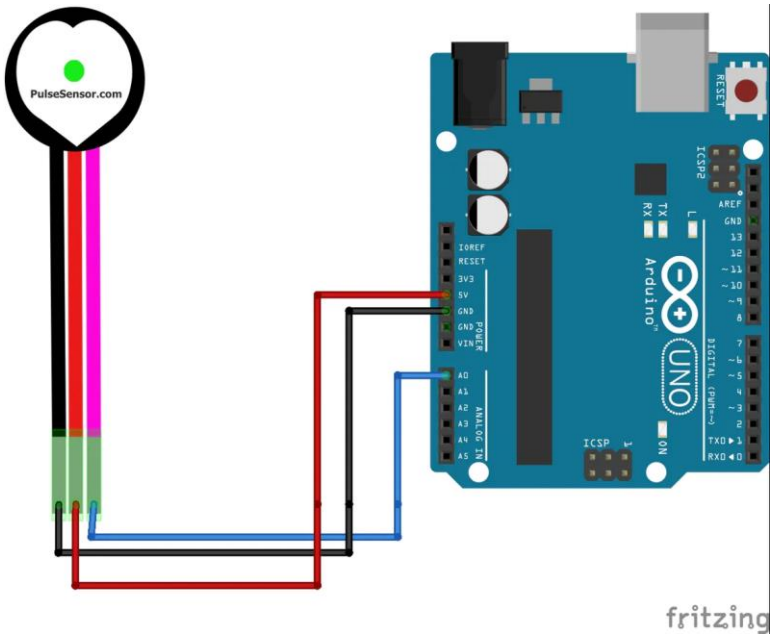
This project will take you through the process of implementing a simplified version of the game **Wheel of Fortune**. Here are the rules of our game:

- **There are num_human human players and num_computer computer players.**
  - Every player has *some* amount of money ($0 at the start of the game)
  - Every player has a set of prizes (none at the start of the game)
- **The goal is to guess a phrase within a given category. For example:**
  - Category: **Artist & Song**
  - Phrase: **----------**
- **Players see the category and an obscured version of the phrase where every alphanumeric character in the phrase starts out as hidden (using underscores: _):**
  - Category: **Artist & Song**
  - Phrase: _____ _____'_ _ _____ _____ ___

During their turn, every player spins the wheel to determine a prize amount and:

- **If the wheel lands on a cash square, players may do one of three actions:**
    - **Guess any letter that hasn't been guessed by typing a letter (a-z)**
        - Vowels (a, e, i, o, u) cost $250 to guess and can't be guessed if the player doesn't have enough money. All other letters are "free" to guess
        - The player can guess any letter that hasn't been guessed and gets that cash amount for *every time* that letter appears in the phrase
        - If there is a prize, the user also gets that prize (in addition to any prizes they already had)
        - **Example: The user lands on $500 and guesses 'W'**
        - There are three W's in the phrase, so the player wins $1500
    - **Guess the complete phrase by typing a phrase (anything over one character that isn't 'pass')**
        - If they are correct, they win the game
        - If they are incorrect, it is the next player's turn
    - **Pass** their turn by entering 'pass'

Class 25 & 26: Arduino- Heart Rate Sensor



Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Monitoring body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy.

In order to measure the body temperature, we use thermometers and a sphygmomanometer to monitor the Arterial Pressure or Blood Pressure. Heart Rate can be monitored in two ways: one way is to manually check the pulse either at wrists or neck and the other way is to use a Heartbeat Sensor.

In this project, we have designed a Heart Rate Monitor System using Arduino and Heartbeat Sensor. You can find the Principle of Heartbeat Sensor, working of the Heartbeat Sensor and Arduino based Heart Rate Monitoring System using a practical heartbeat Sensor.

Introduction to Heartbeat Sensor

Monitoring heart rate is very important for athletes, patients as it determines the condition of the heart (just heart rate). There are many ways to measure heart rate and the most precise one is using an Electrocardiography

But the more easy way to monitor the heart rate is to use a Heartbeat Sensor. It comes in different shapes and sizes and allows an instant way to measure the heartbeat.

Heartbeat Sensors are available in Wrist Watches (Smart Watches), Smart Phones, chest straps, etc. The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute.

Principle of Heartbeat Sensor

The principle behind the working of the Heartbeat Sensor is Photoplethysmograph. According to this principle, the changes in the volume of blood in an organ is measured by the changes in the intensity of the light passing through that organ.

Usually, the source of light in a heartbeat sensor would be an IR LED and the detector would be any Photo Detector like a Photo Diode, an LDR (Light Dependent Resistor) or a Photo Transistor.

With these two i.e. a light source and a detector, we can arrange them in two ways: A Transmissive Sensor and a Reflective Sensor.

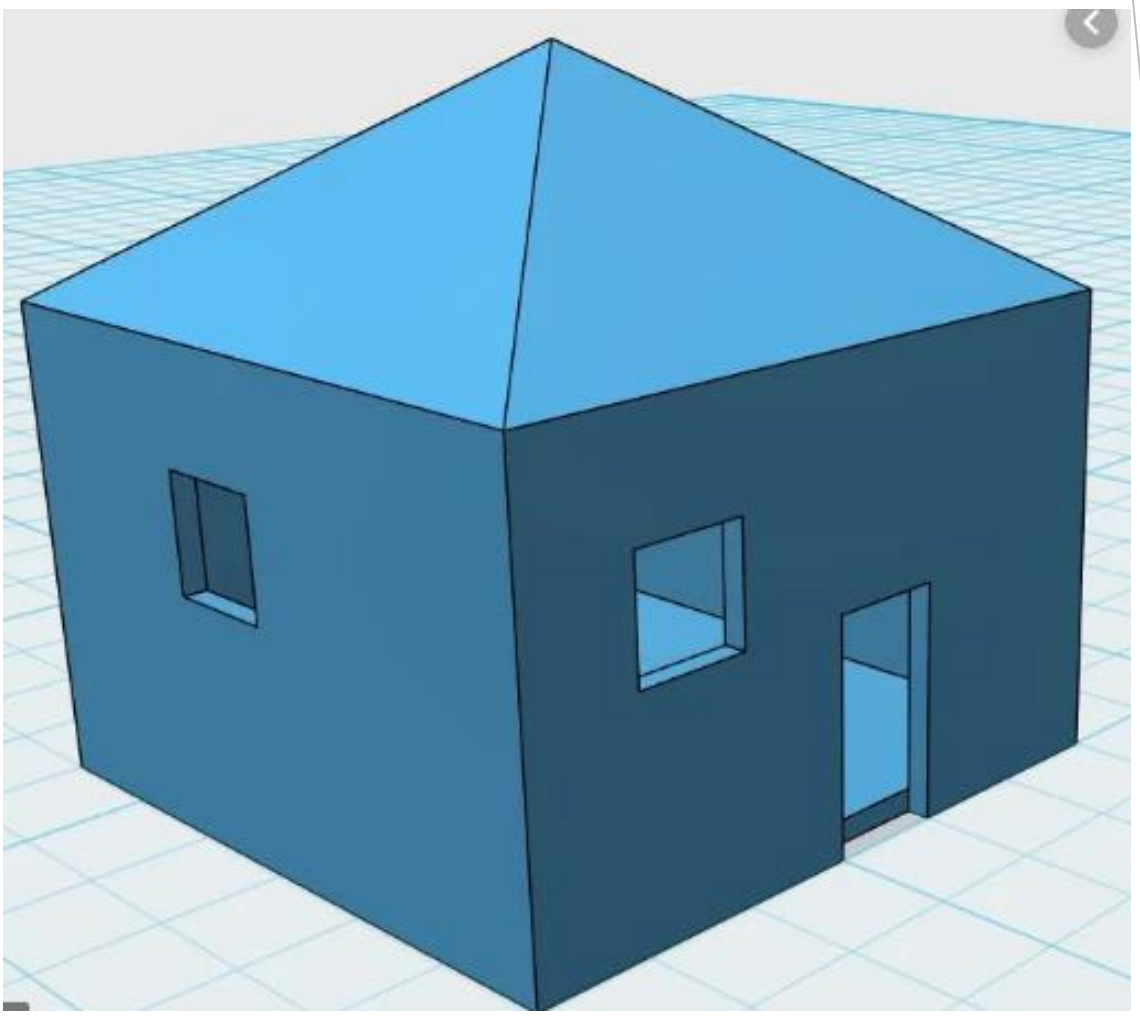
In a Transmissive Sensor, the light source and the detector are place facing each other and the finger of the person must be placed in between the transmitter and receiver.

Reflective Sensor, on the other hand, has the light source and the detector adjacent to each other and the finger of the person must be placed in front of the sensor.

Components Required

•Arduino UNO
•16 x 2 LCD Display
•10KΩ Potentiometer
•330Ω Resistor (Optional – for LCD backlight)
•Push Button
•Heartbeat Sensor Module with Probe (finger based)
•Mini Breadboard
•Connecting Wires

Class 27 & 28: 3d- House



Tools Used :

1. 2D Shapes
2. Extrude
3. Rectangular Pattern
4. Loft
5. Fillet

Learning Outcome:

1. Learning the concept of changing 2d into 3d shapes.
2. Using Loft and Rectangular **pattern**