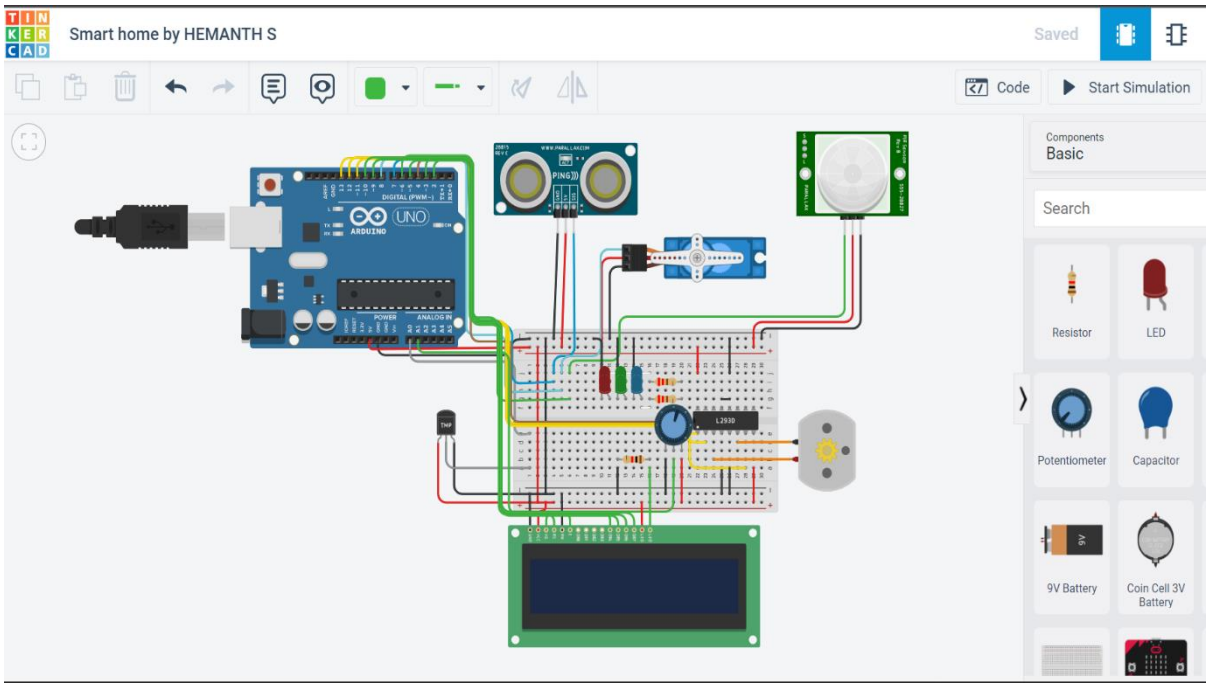
SMART HOME ASSIGNMENT 1



CODE

```
#include<Servo.h>
#include<LiquidCrystal.h>
LiquidCrystal lcd(A1,10,9,6,5,3);
float value;
int tmp = A0;
const int pingPin = 7;
int servoPin = 8;

Servo servo1;
void setup()
{
  Serial.begin(9600);
  servo1.attach(servoPin);
```

```
  lcd.begin(16, 2);
  pinMode(2,INPUT);
  pinMode(4,OUTPUT);
  pinMode(11,OUTPUT);
  //pinMode(10,INPUT);
  //pinMode(2,OUTPUT);
  //pinMode(8,OUTPUT);
  //pinMode(9,output);
  //pinMode(11,OUTPUT);
  //pinMode(13,OUTPUT);
  //pinMode(14,OUTPUT);

  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(A0,INPUT);
  digitalWrite(2,LOW);
  digitalWrite(11,HIGH);
  //digitalWrite(5,OUTPUT);
  digitalWrite(3,OUTPUT);
  digitalWrite(7,OUTPUT);
  digitalWrite(11,OUTPUT);
  digitalWrite(13,OUTPUT);
  //digitalWrite(A0,OUTPUT);
}

void loop()
{

  long duration, inches, cm;
```

```
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);


pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);


inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

servo1.write(0);

if(cm < 40)
{
  servo1.write(90);
  lcd.setCursor(0,1);
  lcd.print("Door:OPEN");

}
else
{
  servo1.write(0);
  lcd.setCursor(0,1);
  lcd.print("Door:CLOSED");
```

```
}


int pir = digitalRead(2);

if(pir == HIGH)
{
  digitalWrite(4,HIGH);
  lcd.setCursor(10,0);
  lcd.print("LED:ON");
 // delay(500);
}
else if(pir == LOW)
   lcd.setCursor(12,0);
  lcd.print("OFF");
{
  digitalWrite(4,LOW);
}

value = analogRead(tmp)*0.004882814;
 value = (value - 0.5) * 100.0;
 lcd.setCursor(0,0);
          lcd.print("Tmp:");
          lcd.print(value);
          delay(1000);


 Serial.println("temperature");
 Serial.println(value);
```

```
  if(value > 20)
  {
    digitalWrite(12,HIGH);
    digitalWrite(13,LOW);
  }
  else
  {
    digitalWrite(12,LOW);
    digitalWrite(13,LOW);
  }
  lcd.clear();
}

long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```