**Department of Information Science & Engineering**

**BE**

**IS45 Microprocessor**

**Smart Surveillance Module**

**Submitted By: 1MS17IS125-UJWAL K C 1MS17IS123-TARUN R**

**Faculty In-Charge: George Philip C**

Associate Professor

**RAMAIAH INSTITUTE OF TECHNOLOGY**

(AUTONOMOUS INSTITUTE, AFFILIATED TO VTU)

**Bangalore – 560054**

**2018-20**

**Abstract**

Burglary is one of the main problem faced in most of the over populated countries like India, Punishing the culprit would only lead to depression and the mindset of the culprit will probably try to find a different way than the solve the problem present at hand ,Since prevention is better than cure its better to avoid burglary from happening, the best way to do it will be to implement a surveillance system. Since the range of the most commonly used cameras are always lesser than 180 degrees it would require 2- 3 cameras if the field to be covered is bigger So in order to cut down costs and make it more efficient and reliable we thought of coming up with a smart surveillance system, our module uses just one camera and uses a sound sensing module to detect the direction of the sound and rotate the camera in that direction.

**Aim:** The objective of our project is to build a module which detects sound waves with the help ofelectret microphones and finds out the direction of the source of sound.

**Devices Used:**

Arduino Uno

28BYJ-48 DC 5V 4 Phase 5 Wire Stepper Motor

l298-n dual bridge motor driver

KY-038 sound sensor module

Jumper wires

**Procedure:**

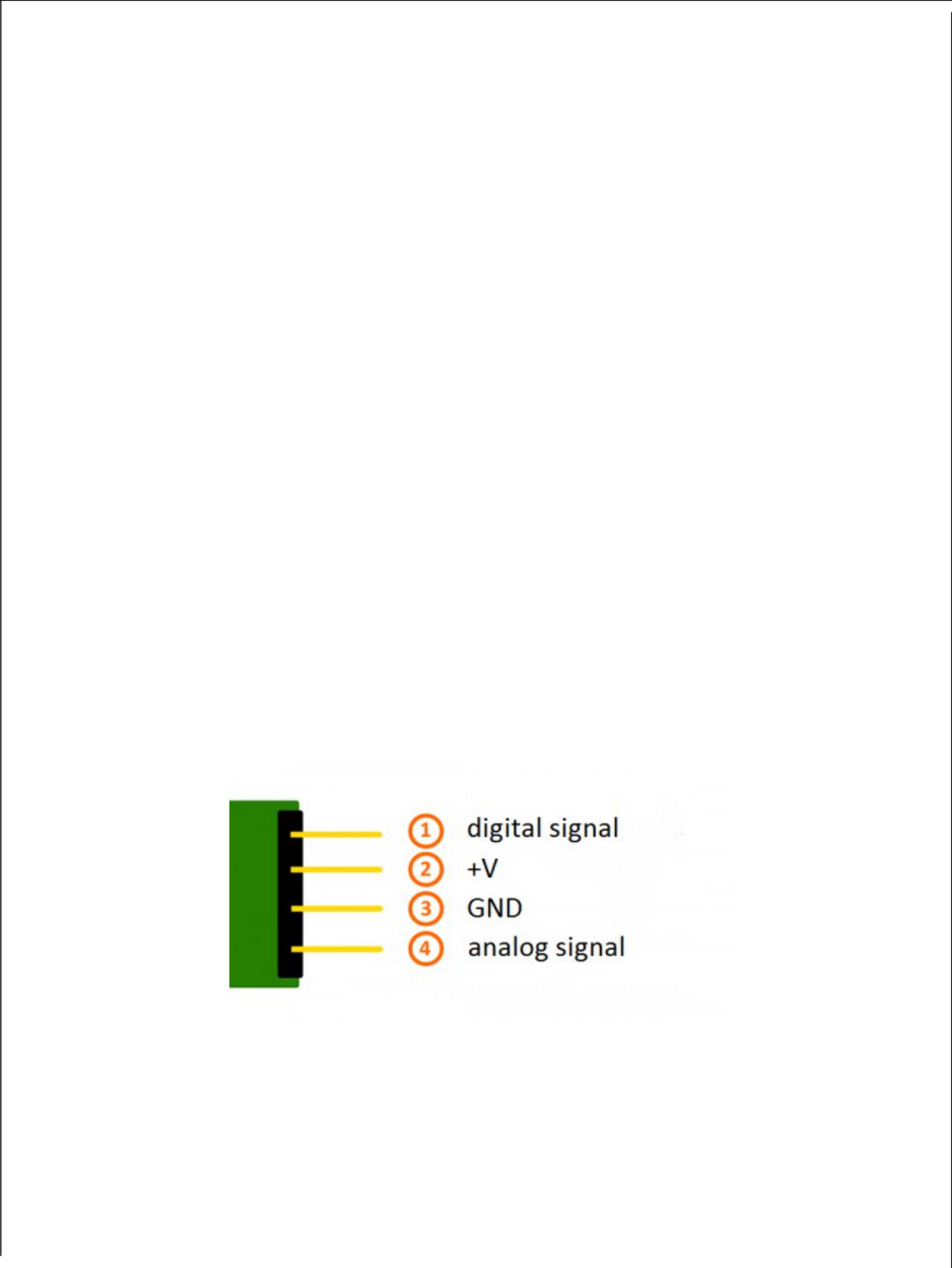
We arranged the three sound sensing modules KY-038 with an angle of 90 degree between them,KY-038 has 3 main components on its circuit board. First, the

Electret microphone unit at the front of the module which measures the intensity of the sound and sends an analog signal to the second unit, the amplifier. The amplifier amplifies the signal, according to the resistant value of the potentiometer, and sends the signal to the analog output of the module.

The third component is a comparator which switches the digital out and the LED if the signal falls under a specific value.

Pin Description

**AO, Analog Output** - real-timeoutput voltage signal of the microphone



**DO, Digital Output**- when the sound intensity reaches a certain threshold, theoutput high and low signal (Our project wont be using this)

The threshold-sensitivity can be adjusted via potentiometer on the sensor

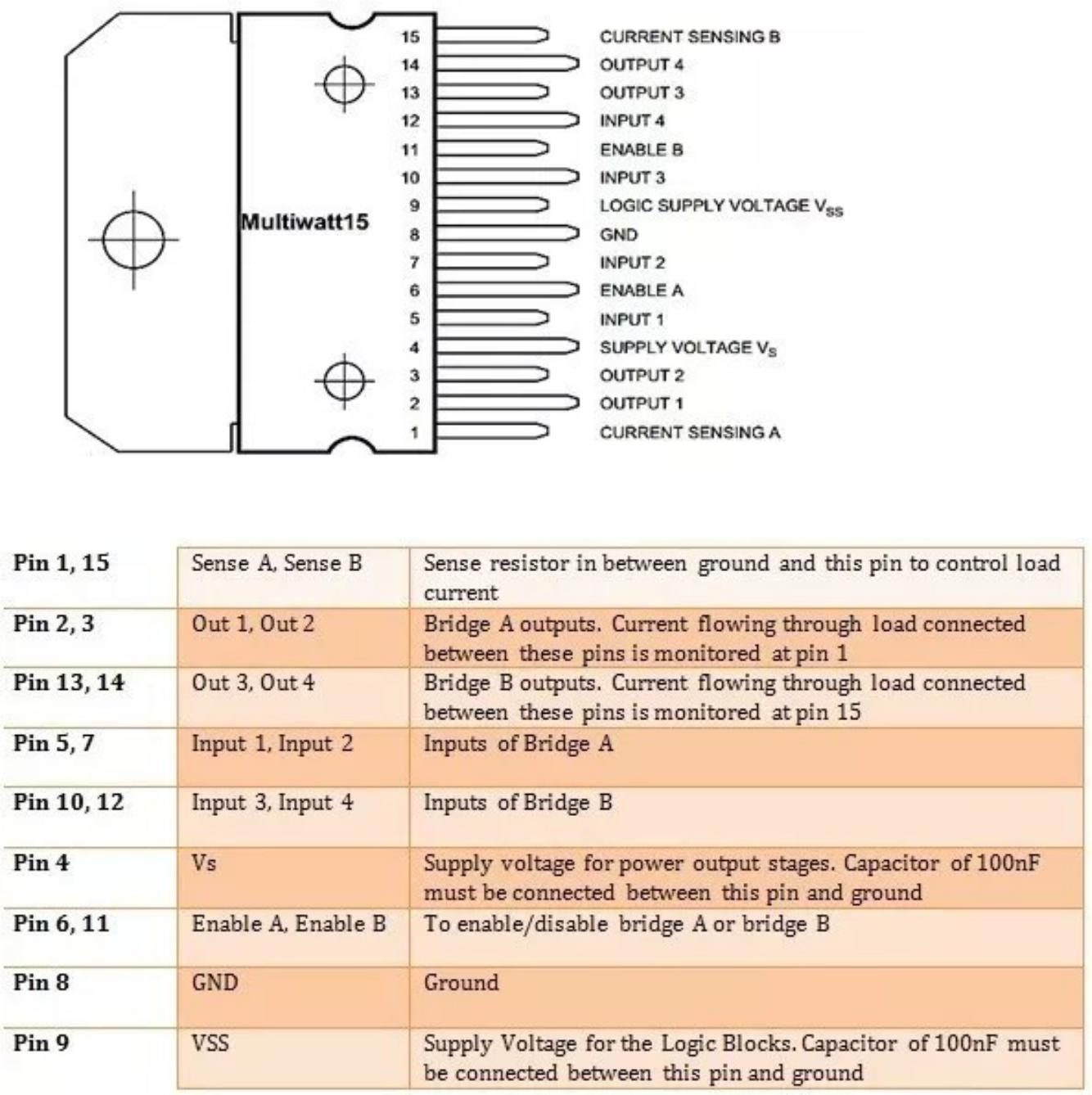


Connecting to the arduino

* Pin +5 of each microphone to +5 of Arduino
* G of each microphone to GND of arduino AO of first microphone to A4 of arduino AO of second microphone to A5 of arduino AO of third microphone to A6 of Arduino

**L298N motor driver:** Most of the microcontrollers on very low voltage (5v)and current while the motor require higher voltage and current So, the microcontrollers cannot provide them such higher current. For this purpose we use motor driver ICs. Motor driver is a little current amplifier. It takes a low current signal and gives out a high current signal which can drive a motor. It can also control the direction of motor.

**Pin description**



L298N consists of four independent power amplifiers. Two of them form H-bridge A while other two form H-bridge B. One H Bridge is used to switch the polarity in controlling the direction of DC Motor Pair of H Bridge is used to control a bi-polar stepper motor.

Amp A1 and A2 => H Bridge A

Amp B1 and B2 => H Bridge B

Basically L298N is used to drive inductive or magnetic loads, so there can come voltage spikes in output. To avoid that voltage spikes there should be some internal parasitic or Flywheel diodes. But it lacks them. We use externally these flywheel diodes. They can be 1N5819 schottky diodes or 1N4001 rectifier diodes.

Each bridge is provided with enable pins (ENA, ENB) and current sense pins (CSA, CSB) . Current sense pins can be tied to ground but we can also insert low value resistor and its voltage reading is proportional to current. Both enable pins can be used at the same time which makes all for outputs active at the same time.All the four inputs and Enable pins work on 5v TTL logic which makes the connection easy with microcontrollers.

ENA=5v, High logic (Amplifier A1 and A2 on)

ENA=0v, Low logic (Amplifier A1 and A2 off)

ENB=5v, High logic (Amplifier B1 and B2 on)

ENB=0v, Low logic (Amplifier B1 and B2 ff)

**L298N motor driver working**

We will be using H bridge motor diver IC L298N and one Stepper motor. This IC is used to control these motors. What we want to do is to change the polarity of motors so they can run in either direction depending upon logic.

Enable bits are used to select specific amplifier. ENA can select two amplifiers

A1, A2 and similarly ENB can select two amplifiers B1, B2. While using as a bridge circuit, ENA Selects bridge A and ENB selects bridge B. To drive both the motors by using H bridges, both enable bits are set high.

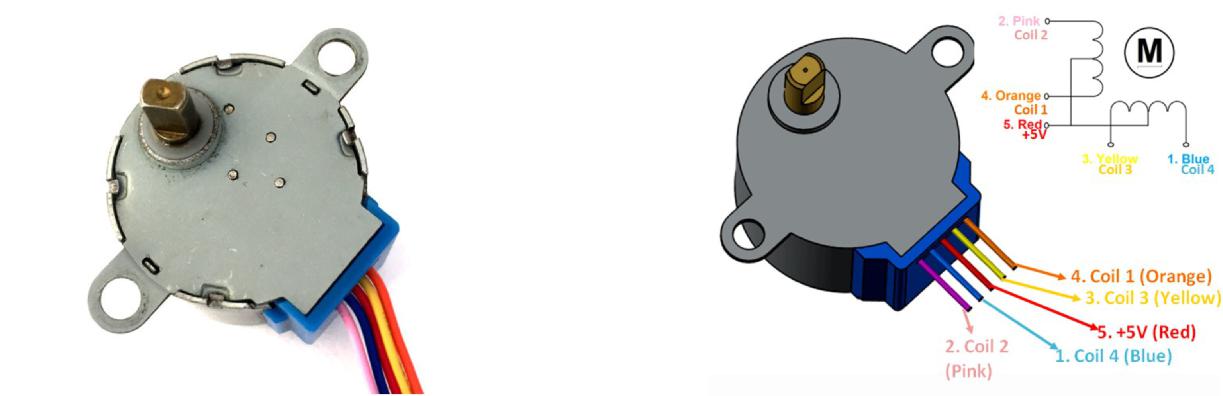
OUTPUTS: There are four outputs. The output for motor A is obtained from out1 out2 pins and similarly for motor B output is obtained from out3 out4 pins. L298N does not have built in protection diodes we used external diodes to prevent the IC from getting damaged.

This IC is using two different voltages. On input side, 5v is given to the pin 9 (Vss), push buttons and enable bits. On output side, pin 4 (Vs) supplies the motors and it can be upto 46 volts. Here we are not using the current sensing scheme, so we have grounded those pins 1 & 15. Motors speed will be lower if low voltages are on output side.

**Connecting to the arduino**

1. Digital pin 6 of arduino - input 4 of Motor Driver
2. Digital pin 7 of arduino - input 3 of Motor Driver
3. Digital pin 8 of arduino - input 2 of Motor Driver
4. Digital pin 9 of arduino - input 1 of Motor Driver
5. 5V pin of arduino - Enable A and Enable B

**28BYJ-48 DC 5V 4 Phase 5 Wire Stepper Motor**



As we can see there are four coils in the motor and one end of all the coil is tied to +5V (Red) and the other ends (Orange, Pink, Yellow and Blue) are taken out as wires. The Red wire is always provided with a constant +5V supply and this +5V will be across (energize) the coil only if the other end of the coil is grounded. A stepper motor can be made to rotate only if the coils are energized (grounded) in a logical sequence. This logical sequence can be programmed using a microcontroller or by designing a digital circuit. The sequence in which each coil should be triggered is shown in the table below. Here “1” represent the coil is held at +5V, since both the ends of coil is at +5V (red and other end) the coil will not be energised. Similarly “0” represents the coil is held to ground, now one end will be +5V and the other one is grounded so the coil will be energised.

Connections of Stepper Motor to Motor Driver

|  |  |
| --- | --- |
| L298N Motor Controller | 28BYJ-48 Stepper Motor |
|  |  |
| OUT1 | 1. Blue |
|  |  |
| OUT2 | 3. Yellow |
|  |  |
| OUT3 | 4. Pink |
|  |  |
| OUT4 | 5. Orange |
|  |  |

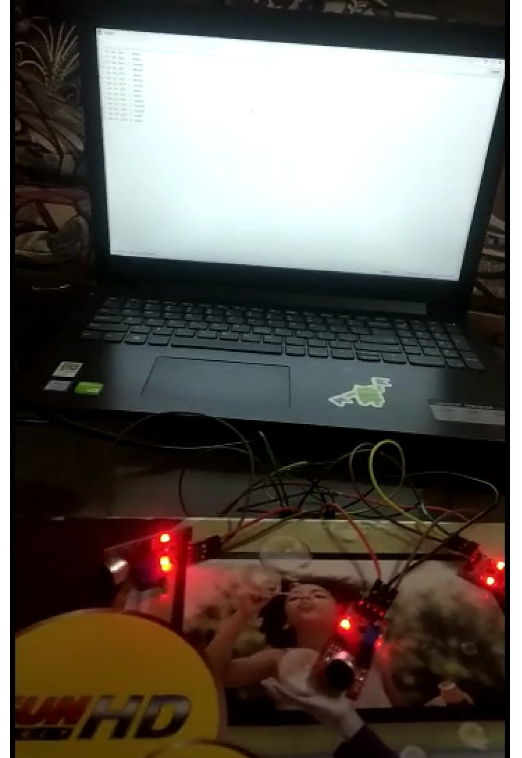
The three microphones are arranged such that one microphone points to North direction one to east direction and the last one to South direction.

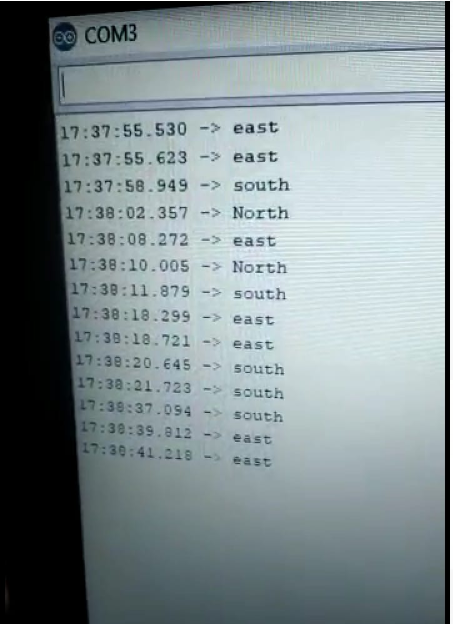
When a sound is emitted from a source, The capacitors inside the microphonevibrate and they send the analog signal to the amplifier present inside the sound sensing module KY- 038 ,the amplified signal is taken out of AO of each microphone module and fed to A4 A5 A6 analog pins of the arduino using south1[i]=analogRead(A5); east1[i]=analogRead(A4); north1[i]=analogRead(A3); For increased efficiency The mean value of ten inputs of the microphone is taken for calculations, for this purpose an array of size 10 is declared and inputs are taken using a loop which runs 10 times.

Since there will be noise in the environment and because of the hardware we need to set a threshold value for each microphone,This we found out by trail and error method(north=65 south=87 east=77) When a value above this threshold value is read by at least one of the microphones then the code evaluation starts if the sound intensity in one microphone is greater than both of the other microphones then the direction to which is the source of the sound This direction is printed on to the console screen and the motor rotates such that the camera points to that direction The Stepper motor controlled by using the inbuilt library stepper.h a stepper object myStepper is created with function Stepper(steps, pin1, pin2, pin3, pin4) and the rotation is controlled using the function Stepper. Step(steps)

with trail and error method we found out that for 180 degree rotation the steps needed is 1100 and for 90 degrees it is 550 we also made use of a variable(d) to know the present direction to which the camera is pointing to, so that if the camera is pointing to the north direction and if the sound is coming from the same direction the camera need not rotate.

RESULT





**Appendix**

**The Working Code**

**#include <Stepper.h>**

**Stepper myStepper(400, 8, 9, 10, 11);**

**int south=0;**

**int north=0;**

**int east=0;**

**int d=0;**

**int in1 = 9;**

**int in2 = 8;**

**int in3 = 7;**

**int in4 = 6;**

**int south1[10];**

**int north1[10];**

**int east1[10];**

**int i;**

**void setup() {**

**myStepper.setSpeed(60);**

**pinMode(in1, OUTPUT);**

**pinMode(in2, OUTPUT);**

**pinMode(in3, OUTPUT);**

**pinMode(in4, OUTPUT);**

**d=0;**

**}**

**void loop() {**

**Serial.begin(9600);**

**readmics();**

**}**

**void readmics(){**

**for(i=0;i<10;i++){**

**south1[i]=analogRead(A5);** //Microphone 1 input

**south=south+south1[i];** //Microphone 2 input

**east1[i]=analogRead(A4);** //Microphone 3 input

**east=east+east1[i];**

**north1[i]=analogRead(A3);**

**north=north+north1[i];**

**}**

**south=south/10; //Taking average**

**north=north/10;**

**east=east/10;**

**if(north>65||south>87||east>77)** //Threshold values of microphones

**{**

**Serial.println(south);**

**Serial.println(east);**

**Serial.println(north);**

**Serial.println();**

**Serial.println();**

**if(north+22>south&&north+10>east){**

**if(d==-1)**

**myStepper.step(-1100);** //Rotating stepper motors

**if(d==0)**

**myStepper.step(-550);**

**d=1;**

**Serial.println("North");**

**}**

**if(south>north+22&&south>east+10){**

**if(d==1)**

**myStepper.step(1100);**

**if(d==0)**

**myStepper.step(550);**

**Serial.println("south");**

**d=-1;**

**}**

**if(east+10>south&&east>north+10){**

**if(d==-1)**

**myStepper.step(-550);**

**if(d==1)**

**myStepper.step(550);**

**d=0;**

**Serial.println("east");**

**}**

**}**

**delay(100);**

**}**

