DEVELOPMENT OF IN HOUSE NON MECHANICAL ANEMOMETER

Contents

1 Introduction **4**

2 Problem definition and background **5**
2.1 Literature review 5
2.2 Reference solution 5

3 Design **6**
3.1 Sensor specification 6
3.1.1 Ultrasonic Transducer 6

4 Block Diagram **7**
4.1 Ultrasonic transducer Setup 7

5 Implementation **8**

6 Hardware **8**
6.1 Arduino Nano Ble 8
6.2 Arduino mega 2560 9
6.3 Signal Processing Unit 10
6.4 Amplifier 11
6.5 Band pass filter 11

7 Wiring Diagram **13**
7.1 First phase circuit diagram 13
7.2 Final circuit diagram 14

8 How it Works **14**

9 Firmware **18**
9.1 Testing 18

10 Results **19**
10.1 Test 1 19
10.2 Test 2 21
10.3 Test 3 Comparison Ultrasonic transducers 22
10.4 Multicomp Pro 23
10.5 Multicomp 24
10.6 Transducers salvaged from HCSR04 Module 25
10.7 Transducers salvaged from US100 26
10.7.1 Comparison Results 28
10.8 Test 4 28
10.8.1 Amplifier Circuit Design 28
10.9 Test 5 30
10.9.1 Bandpass Filter Simulation Result 30
10.10Test 6 32

11 Conclusions **33**

12 Bibilography **34**

13 APPENDIX **35**
13.1 Ultrasonic anemometer code 35
13.2 Machine Learning code 40

List of Figures

1 Block diagram 7
2 Enclosure 8
3 Anemometer inside. 8
4 Arduino nano ble 8
5 Arduino Mega 2560 9
6 IC 74HC4052AP 10
7 IC LM386 11
8 TL082 11
9 Wiring Diagram 13
10 Wiring Diagram 14
11 Dataset 16
12 Non mechanical anemometer with digital anemometer calibration 16
13 Datalogging 16
14 Data 16
15 Multiplexer Function 17
16 Amplifier 18
17 Test1 Result 19
18 Machine Learning Results 20
19 Result 20
20 Transmitter to receiver 21
21 Receiver to transmitter 21
22 pcb for multiplexer Front view 22
23 Bottom view 22
24 Switching funtionality using multiplexer Dot board 22
25 Multicomp pro 23
26 Specifications 23
27 multicomp pro with distance 0cm 23
28 specifications 23
29 multicomp pro with distance 20cm 24
30 specifications 24
31 Multicomp 24
32 Specifications 24
33 multicomp with distance 0cm 24
34 specifications 24
35 multicomp with distance 20cm 25

36 specifications . . . 25
37 HCSR04 with distance 0cm . . . 25
38 specifications . . . 25
39 HCSR04 with distance 20cm . . . 25
40 specifications . . . 25
41 HCSR04 with distance 0cm . . . 26
42 specifications . . . 26
43 HCSR04 with distance 20cm . . . 26
44 specifications . . . 26
45 US100 with distance 0cm . . . 26
46 specifications . . . 26
47 US100 with distance 20cm . . . 27
48 specifications . . . 27
49 US100 with distance 0cm . . . 27
50 specifications . . . 27
51 US100 with distance 20cm . . . 27
52 specifications . . . 27
53 Comparison of transducers . . . 28
54 Comparison graph . . . 28
55 Amplifier circuit ith multiplexer . . . 29
56 Front view . . . 29
57 Bottom view . . . 29
58 Transmitter to receiver signal strength with amplifier . . . 29
59 Transmitter to receiver signal strength with and without amplifier. 29
60 Receiver to transmitter signal strength . . . 30
61 Simulation . . . 31
62 Simulation . . . 31
63 Hardware setup . . . 32
64 Observation . . . 32
65 Processed Data . . . 32
66 Observation . . . 33
67 Output . . . 33

1 Introduction

Anemometer is any device that is used for measuring wind speed and direction. Commonly used mechanical anemometers consist of cup shaped vanes that allow them to rotate at a speed proportional to the wind speed. Due to moving parts, such type of anemometers is more susceptible to wear and tear and thus loss in accuracy. Another type of wind speed measuring device is "Ultrasonic Anemometer" which uses the time of flight (TOF) concept to measure the wind speed and direction. Compared with traditional cup type mechanical anemometers, ultrasonic anemometers have high accuracy and no abrasion.

An ultrasonic anemometer works on the principle of measuring the time it takes for ultrasonic sound waves to travel between transducers in different directions. The three methods mentioned—time difference, phase difference, and Doppler are all techniques used to calculate wind speed based on this principle.

In time difference method, measures the time it takes for an ultrasonic pulse to travel from one transducer to another in both the upstream and downstream directions. By knowing the distance between the transducers and the time it takes for the pulses to travel, the speed of sound in the air can be determined. With this information, the difference in time between the upstream and downstream pulses can be used to calculate wind speed.

In phase difference Method, the phase difference between transmitted and received ultrasonic signals is measured. As wind affects the propagation speed of the ultrasonic waves, the phase shift between them can be used to calculate wind speed. In Doppler method: The Doppler effect occurs when there is a change in frequency of a wave due to the motion of the source or the observer. In the case of an ultrasonic anemometer, the frequency shift of the ultrasonic waves due to the motion of air particles caused by wind is measured. This frequency shift can be used to calculate wind speed.

While all three methods have their advantages and disadvantages, the time difference method is often preferred for its simplicity and ease of application. It directly measures the time it takes for sound waves to travel, making it relatively straightforward to calculate wind speed based on the Time of Flight (ToF) value. However, the choice of method may also depend on factors such as accuracy requirements, environmental conditions, and specific application needs.

The ultrasonic anemometer designed in this research uses time difference method to measure wind speed and direction.Ultrasonic anemometer consists of using ultrasonic transducers which salvaged from HCSR04 modules and DHT sensor to measure temperature and humidity.

Transducers arranged orthogonally. Make the ultrasonic anemometer and takes the duration and temperature & humidity values. Then compared with a digital anemometer. Then the datasets are used for machine learning purpose. Our aim is to find a solution for smart weather station with no mechanical moving parts for measuring all weather conditions with a focus on wind based on low power ML at the edge that can be deployed locally with low cost, low power, reliable, accurate, easy to install and maintain.

To avoid the temperature & humidity dependencies, use a signal routing unit.

Signal routing is used to control the sensor operation and to reduce the hardware complexity by reusing the transmitter and receiver circuit. It used as roll switching.

2 Problem definition and background

Measuring wind speed and direction with a weather station is an important part of understanding and predicting weather conditions. Wind plays a significant role in many weather patterns and phenomena, such as storms, hurricanes, and tornadoes.
The application of anemometer is in the field of weather forecasting, marine operation, Aviation, bridge building, environmental monitoring, security and defence and scientific research but the wind power generation industry is the biggest consumer of anemometer worldwide.

2.1 Literature review

Mechanical sensors used to measure wind speed can be described as having three hemispheric cups attached to arms on a vertical axis turn in the wind at a speed proportional to wind speed. Wind direction is measured by another type of mechanical sensor a vane anemometer that combines a small horizontally oriented propeller and tail to turn the axis perpendicular to the wind direction and the propeller blades transmit wind direction to the turbine. These two types comprise of moving parts such as ball bearings that are subject to damage at high wind speeds, icing incidents, and typical mechanical ageing due to wear and tear found operating under various weather conditions.

2.2 Reference solution

So, moved to non mechanical anemometers. Ultrasonic anemometers without hydraulic or mechanical parts are not prone to rust, scale, or other buildup that may impede proper operation. Some ultrasonic sensors are also equipped with heating components that control the internal temperature in icing events, which may occur at hub height. It is imperative that the sensors are kept at a temperature that allows it to operate accurately and maximizes power output of the turbine. Furthermore, ultrasonic anemometers may have both digital or analog outputs directly from the sensor, which can provide direct communication with the turbine's Programmable Logic Controller (PLC), that monitors and operates each turbine's electromechanical processes. This direct communication is important for minimizing power loss, reducing replacement costs, identifying turbine maintenance needs, reducing dead band.

3 Design

Ultrasonic anemometer is used to measure wind velocity and direction for weather stations and other industrial applications. It consists of ultrasonic transducers, signal processing unit, amplifier, bandpass filter and also a sonic anemometer is used. Transducers are used to generate 40KHz ultrasonic sound waves.Signal routing is used to control the sensor operation and to reduce the hardware complexity by reusing the transmitter and receiver circuit. Amplifier is used to amplify the receiver.The received signal first passes from the amplifier stage to increase the strength of the signal after that it passes through the band pass filter which is having a center frequency of 40 kHz and bandwidth of 10kHz, to remove the unwanted frequency component.Bandpass filter designed for this purpose. .

3.1 Sensor specification

3.1.1 Ultrasonic Transducer

Four multicomp MCUSD16A40S12RO ultrasonic ceramic transducers (one for each direction) are used for receiving and transmitting ultrasonic waves which is highly reliable, water proof, light weight, high sensitivity and less power consumption. (Transducers salvaged from HCSR04 Module). Ultrasonic ranging module HCSR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm.

- Working Voltage :- DC 5V
- Working Current:- 15mA
- Measuring Angle:- 15 degree

4 Block Diagram

4.1 Ultrasonic transducer Setup

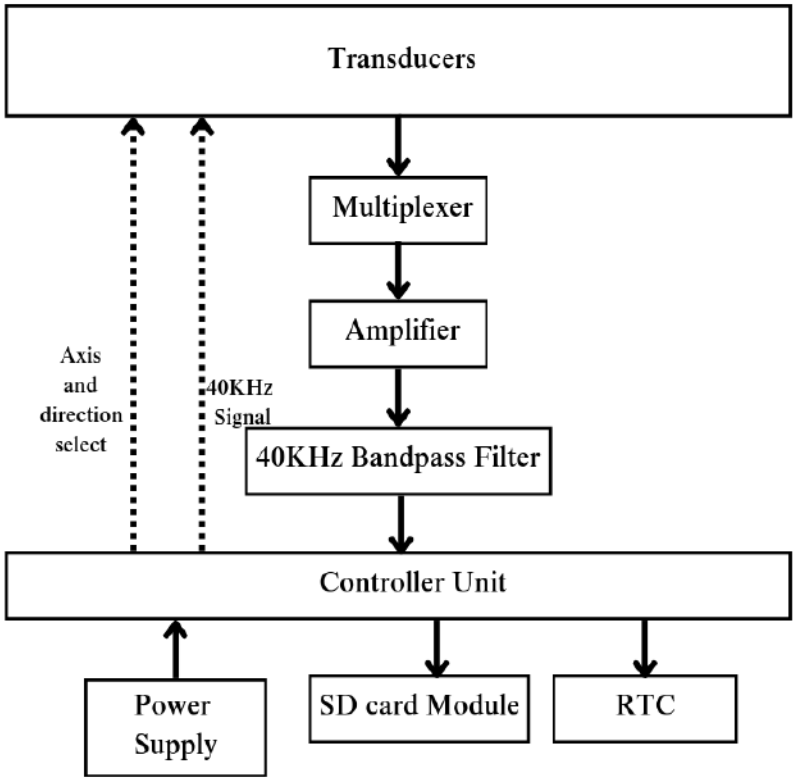


Figure 1: Block diagram

For prototyping Arduino Nano 33 BLE Sense microcontroller based development board and Arduino IDE is used, architecture for the prototype is as shown in figure. Microcontroller is used to, generate the square wave of 40 kHz which is used to excite the ultrasonic sensor to generate the ultrasonic signal, control the signal routing unit, calculate the delay using internal timer, and to interface the storing unit. Data is send to serial port so that user can access the data at serial port of the connected terminal. Also planned to collect the data through SD Card module in .csv format. RTC module is used to store the timing information.

5 Implementation

Connect the circuit as per the block diagram and take the duration between 2 transducers ,temperature and humidity values from DHT sensor.

Figure 2: Enclosure

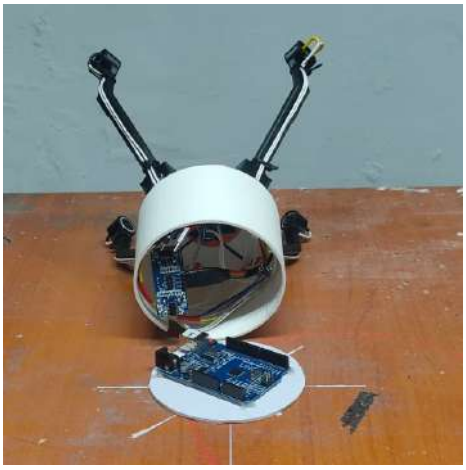


Figure 3: Anemometer inside.

6 Hardware

6.1 Arduino Nano Ble

The Arduino Nano 33 BLE Sense is a compact and feature-rich microcontroller board based on the nRF52840 microcontroller from Nordic Semiconductor. It is designed for projects that require Bluetooth Low Energy (BLE) connectivity and a host of various sensors while maintaining a small form factor. The Nano 33 BLE Sense is equipped with various onboard sensors including those for temperature, humidity, pressure, light, color, 3D motion, and sound, making it exceptionally well-suited for IoT (Internet of Things) projects, wearable devices, and environmental sensing applications.

Figure 4: Arduino nano ble

- Microcontroller: Nordic nRF52840
- Clock Speed: 64 MHz
- Supply Voltage: The board is designed to be powered through the USB connection at 5V. However, it also supports a Vin (voltage input) range of 4.5V to 21V. Internally, it operates at 3.3V logic.
- Flash Memory: 1 MB
- Connectivity: BLE (Bluetooth Low Energy) and NFC
- Additional Features: Embedded IMU (Inertial Measurement Unit) for motion sensing. Temperature, pressure, humidity, light, and color sensors along with a gesture, proximity, and a microphone for audio sensing, making it an ideal board for IoT projects and applications requiring multiple sensor inputs. A USB interface for programming and power supply.

6.2 Arduino mega 2560

For prototyping ATmega 2560 microcontroller based development board and Arduino IDE is used. ATmega 2560 is used to, generate the square wave of 40 kHz which is used to excite the ultrasonic sensor to generate the ultrasonic signal, control the signal routing unit and calculate the delay using internal timer.

Figure 5: Arduino Mega 2560

- Operating Voltage:- 5V
- DC Current per I/O Pin :- 20 mA
- DC Current for 3.3V Pin :- 50 mA
- Clock Speed :-16 MHz

6.3 Signal Processing Unit

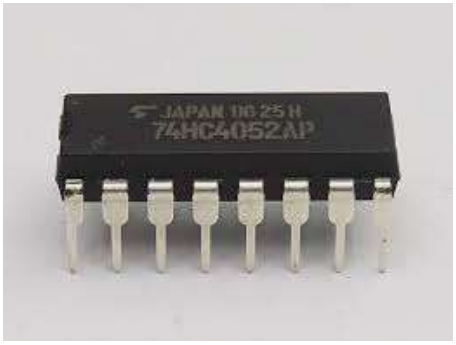


Figure 6: IC 74HC4052AP

For signal routing TC74HC4052AP is used.It is a high speed CMOS analog multiplexer/ demultiplexer fabricated with silicon gate C2MOS technology. It acheive the high speed operation similar to equivalent LSTTL while maintaining the CMOS low power dissipation. It has a 4 channel* 2 configuration. The digital signal to the control terminal turns "ON" the corresponding switch of each channel a large amplitude signal (VCC-VEE) can then be switched by the small logical amplitude (VCC-GND) control signal.

There are 2-axis, N-S and E-W each axis having two directions (in case of N-S,N→S and S→N)delay for both the direction and axis has to be calculated one after another for wind velocity calculation. TN-S and T S-N are the delay for N-S axis for N→S and S→N direction respectively and T E-W and T W-E are the delay for the E- W axis for E→W and W→E respectively. When TN-S is calculated, North sensor is working as a transmitter and South sensor is working as a receiver and 40 kHz signal generated from the control unit is routed to North sensor and the received signal from the south sensor is routed to receiver circuit and in case of T S-N calculation South sensor is working as a Transmitter and North sensor is working as receiver and 40 kHz signal routed to South sensor and received signal is routed from North sensor to receiver circuit. So this switching from transmitter to receiver or receiver to transmitter and signal routing from control unit to Tx and Rx to receiver circuit is done by signal routing unit along with control unit. For signal routing 4052B IC (Dual 4-channel Analog Multiplexer/Demultiplexer) with common channel select logic (S0 and S1) is used. Each multiplexer/demultiplexer has four independent inputs/outputs (nY0 to nY3) and a common input/output (nZ). Here S1 is used to select the axis (i.e, N-S,E-W) and S0 is used to select the direction.

- Analog input voltage range from -5 V to +5 V
- Temperature range from -40 °C to +85 °C .

6.4 Amplifier

Figure 7: IC LM386

The output signal from the ultrasonic transceivers has to be amplified & conditioned in order for efficient signal processing by the micro controllers. Hence, the sinusoidal waves generated by the transceivers has to be transformed to square waves. We developed an amplifier circuit in order to accomplish this; by utilizing LM386.

LM386: This is a power amplifier designed for use in low voltage consumer applications. The gain is set internally to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.

- Supply Voltage Range: 4 V–12 V or 5 V–18 V
- Low Quiescent Current Drain: 4 mA
- Voltage Gains from 20 to 200

6.5 Band pass filter

Figure 8: TL082

The received signal first passes from the amplifier stage to increase the strength of the signal after that it passes through the multiple feedback band pass filter which is having a center frequency of 40 kHz and bandwidth of 10 kHz, to

remove the unwanted frequency component.We developed a bandpass filter in order to accomplish this, by utilizing TL082. TL082: The TL082 is a high-speed, low-cost, wide bandwidth and dual JFET input operational amplifier. This component is available with an internally trimmed offset voltage. It comes with a fast slew rate and low supply current.This JFET input device extends low offset and input bias current. The TL082 is electrically compatible with LM1558 and is used to enhance the overall performance of the LM1558 device. With high input impedance and low total harmonic distortion, this device features low noise and offset voltage drift. This amplifier is widely used in audio pre-amplification, sample and hold amplifiers, peak detectors and active filters.

7 Wiring Diagram

7.1 First phase circuit diagram

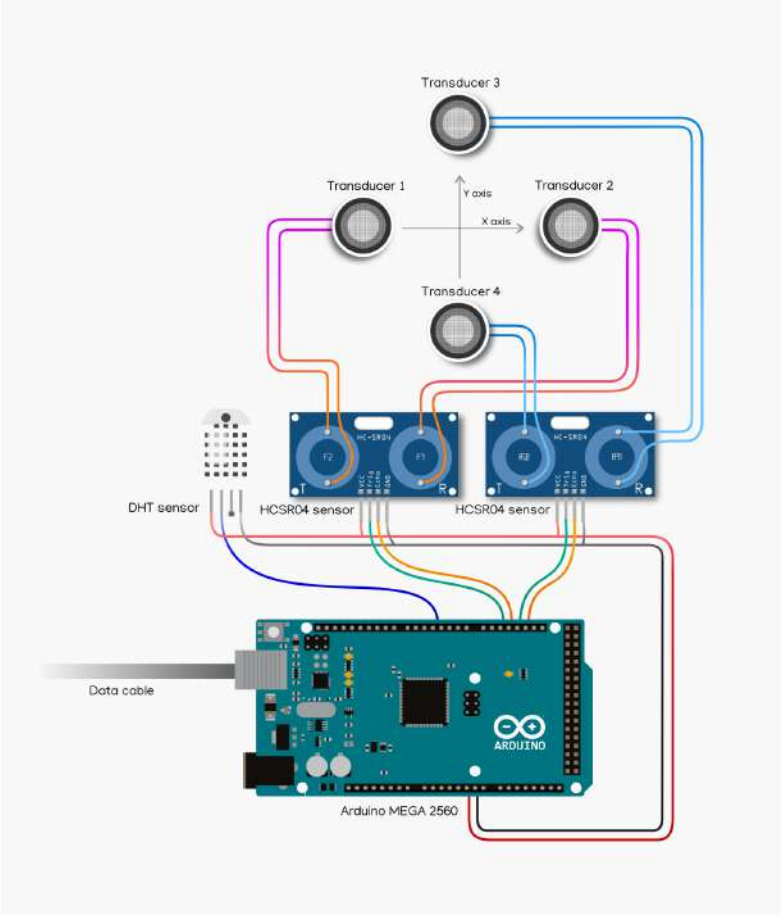


Figure 9: Wiring Diagram

The development board for the ATmega 2560 microcontroller prototype, designed using Arduino IDE, follows the architecture depicted in the figure. The ATmega 2560 serves multiple functions including generating a 40 kHz square wave to excite the ultrasonic sensor for signal generation and calculating the DHT sensor output.Additionally, data is transmitted to the serial port for user access via a connected terminal. Wind velocity depends on the temperature and humidity values. Here, used the salvaged transducers from HCSR04 or US100. This is the first phase of the project. After, for getting higher efficiency we use arduino nano ble 33 sense board.

7.2 Final circuit diagram

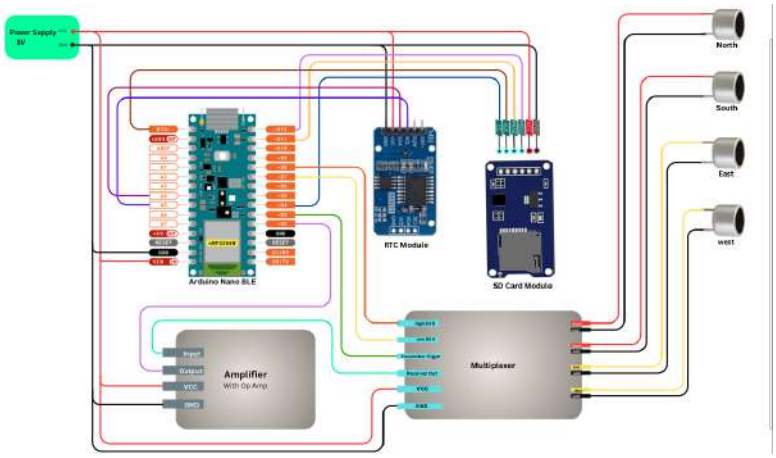
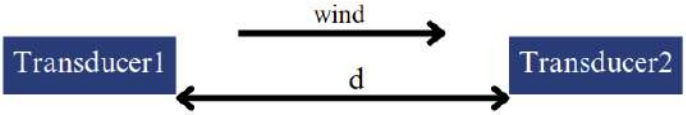


Figure 10: Wiring Diagram

The development board for the arduino nano ble 33 sense prototype, designed using Arduino IDE, follows the architecture depicted in the figure. The ATmega 2560 serves multiple functions including generating a 40 kHz square wave to excite the ultrasonic sensor for signal generation,controlling the signal routing unit, calculating delays using internal timers, and interfacing with the storage and timer units. An SD card module is incorporated to store data in .csv or .xlsx formats, while an RTC module is utilized for timekeeping. Additionally, data is transmitted to the serial port for user access via a connected terminal.

8 How it Works

The working principle of the ultrasonic anemometer is to use the ultrasonic time difference method to measure wind speed and direction. Because the speed of sound in the air will be superimposed with the speed of airflow in the wind. If the propagation direction of the ultrasonic wave is the same as the wind direction, then its speed will increase. Conversely, if the propagation direction of the ultrasonic wave is opposite to the wind direction, then its speed will slow down. Therefore, under fixed detection conditions, the speed of ultrasonic propagation in the air can correspond to the wind speed function. The precise wind speed and direction can be obtained by calculation. As the sound wave propagates in the air, its speed is greatly affected by temperature; the wind speed sensor detects two opposite directions on the two channels, so the effect of temperature on the sound wave speed can be ignored.



t21 is the delay when transducer2 and transducer1 is working as a Transmitter (Tx) and Receiver (Rx) respectively

$$t_{12} = \frac{d}{V + U} \tag{1}$$

$$t_{21} = \frac{d}{V - U} \tag{2}$$

Where, V is speed of wind and U is speed of ultrasonic sound wave which is generated by the ultrasonic transducer. The value of the t12 is calculated by the internal timer of the micro-controller. So from (4.1) or (4.2) alone we can find the speed of the wind. But (4.1) and (4.2) contain the term U which is very critical term. The value of U is dependent on many factor like temperature, humidity etc.

$$V = \frac{d}{2}\left(\frac{1}{t_{12}} - \frac{1}{t_{21}}\right) \tag{3}$$

For the measurement of the wind speed and direction we have to place 4 ultrasonic transceivers in a horizontal X-Y plane one in each direction. We can find two wind velocity vector i.e. V x and V y which can be calculated as

$$V_x = \frac{d}{2}\left(\frac{1}{t_{E-W}} - \frac{1}{t_{W-E}}\right) \tag{4}$$

$$V_y = \frac{d}{2}\left(\frac{1}{t_{S-N}} - \frac{1}{t_{N-S}}\right) \tag{5}$$

From this, find wind speed

$$V_{wind} = \sqrt{V_x^2 + V_y^2} \tag{6}$$

From this, find wind direction

$$\phi = \arctan(\frac{V_y}{V_x}) \tag{7}$$

On the first phase,from the circuit, get the duration 1 and duration 2 , temperature and humidity values. Speed of sound depends on temperature and humidity.
Speed of Sound (m/s) = 331.5 + 0.6 * Temperature (°C) + 0.0124 * Humidity (%)
it depends on the temperature , humidity.

duration1	duration2	temperature	humidity
591	591	27.7	61.2
567	560	27.7	61.2
568	558	27.7	61
573	557	27.7	61
573	557	27.7	61.2
573	564	27.7	61.2

Figure 11: Dataset

We take a large amount of dataset at different speed levels using pedestrial fan and move the distance between the non mechanical anemometer setup and the fan. Also compared the wind speed of non mechancial anemometer with digital anemometer.

Figure 12: Non mechanical anemometer with digital anemometer calibration

Figure 13: Datalogging

duration1	duration2	temperature	humidity	speedlevel
591	591	27.7	61.2	13.85
567	560	27.7	61.2	13.85
568	558	27.7	61	13.85
573	557	27.7	61	13.85
573	557	27.7	61.2	13.85
573	564	27.7	61.2	13.85
574	563	27.7	61.3	13.85
567	564	27.7	61.3	13.85
567	564	27.7	61.3	13.85
567	563	27.7	61.3	13.85
567	588	27.7	61.3	13.85
568	561	27.7	61.3	13.85
591	585	27.7	61.2	13.85

Figure 14: Data

Datasets are collected for 2 purpose

- For Wind Velocity and direction using the Equations shows above

- For Machine Learning

For eliminating temperature and humidity values, role switching circuit was designed.(Signal processing unit mentioned above).

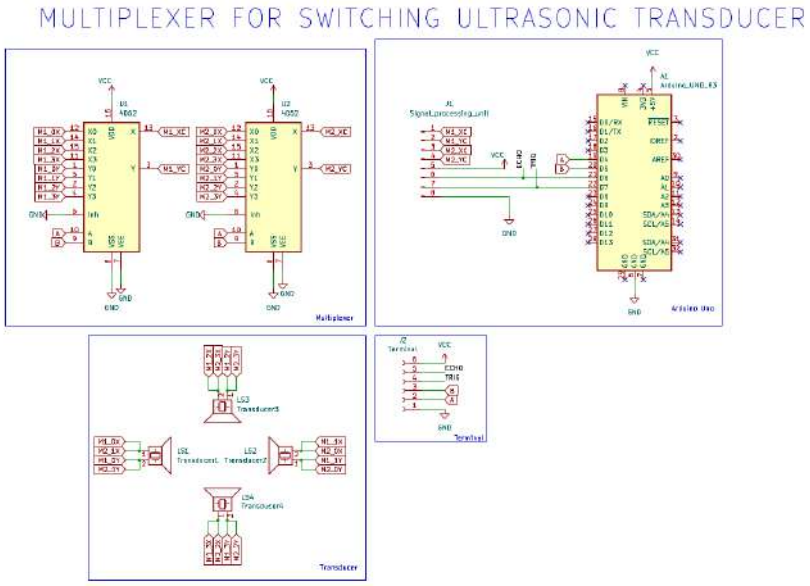


Figure 15: Multiplexer Function

4 ultrasonic transducers are connected orthogonal to each other with a distance 20cm. A multiplexer is used for switching the transducers. The output signal from the ultrasonic transceivers has to be amplified & conditioned in order for efficient signal processing by the micro controllers. Hence, the sinusoidal waves generated by the transceivers has to be transformed to square waves. We developed an amplifier circuit in order to accomplish this; by utilizing LM386.

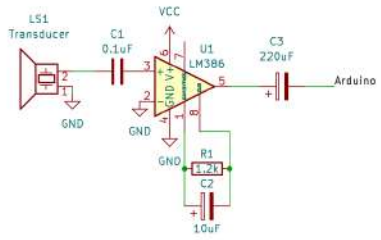


Figure 16: Amplifier

Capacitors are used essentially for coupling and filtering.

- C1 (0.1 µF): This is an input decoupling capacitor, acting as a DC blocking capacitor that allows the AC signal from the transducer to pass into the amplifier while blocking any DC voltage. Its main function is to prevent any DC offset from the preceding stage from affecting this stage.
- C3 (220 µF): This is an output decoupling capacitor, often called a coupling capacitor as well. It performs a similar role to C1, allowing the AC audio signal to pass onwards to the Arduino, while blocking any DC voltage from heading into the Arduino's analog input. Its purpose is to prevent the DC operating point (the quiescent current) of the amplifier from affecting subsequent stages or devices.

9 Firmware

9.1 Testing

The main and the conclusive part of the project is testing. Initially, all the components are tested and verified that they are working properly and identified the current consumption's while using sleep mode. After all the individual tests the components are integrated and checked that all the connections are proper. The second phase is software testing and the code is developed and dumped into the controller. Next, testing the device performance under the lab condition and continuously monitor the output of the device and check if the device outputs are valid to our need. The last phase is to deploy the device for the field test, whether the device is performing under certain weather conditions. All the outputs are noted and compared with other similar devices. Finally, deploy the device for the implementation.

10 Results

For the testing of the device, several procedures are taken for accurate results. The project was designed to analyze the wind velocity ,speed and monitor the data for future references.

10.1 Test 1

On the first phase take the duration 1, duration2, temperature, humidity and corresponding digital anemometer values. and analyzed the data. We take a large amount of dataset at different speed levels using pedestrial fan and move the distance between the non mechanical anemometer setup and the fan. Also compared the wind speed of non mechancial anemometer with digital anemometer.

duration1	duration2	temperature	humidity	speedlevel
591	591	27.7	61.2	13.85
567	560	27.7	61.2	13.85
568	558	27.7	61	13.85
573	557	27.7	61	13.85
573	557	27.7	61.2	13.85
573	564	27.7	61.2	13.85
574	563	27.7	61.3	13.85
567	564	27.7	61.3	13.85
567	564	27.7	61.3	13.85
567	563	27.7	61.3	13.85
567	588	27.7	61.3	13.85
568	561	27.7	61.3	13.85
591	585	27.7	61.2	13.85

Figure 17: Test1 Result

Large Dataset was created and evaluated through model creation. During the initial stages of the project, our team had to have self reliance on the domain of machine learning, so we were in need of relevant data at our disposal. Prototype-I was used for creation of the fore mentioned dataset. During our learning phase we planned for the creation of two seperate models for wind velocity & wind direction. Using the Prototype-I, we collected 18000 datapoints for creation of wind velocity model, which yielded an R2 score of 98 using Decision Tree Regression model. Most commonly used AI algorithms were tested during this phase including Simple linear regression, Polynomial regression, Support Vector Machine & Random Forest regression. Out of these models decision tree model gave highest R2 score. The dataset for wind direction had 1.5 lakh datapoints. 1000 duration readings were recorded for each 15 degree rotation of the primary axis of Prototype-I with respect to the wind flow direction, yielding 25000 data points per fan speed and distance configuration. The wind direction model also yielded R2 score of 98 using Decision Tree model. Roughly 70 different AI Models were created & evaluated .

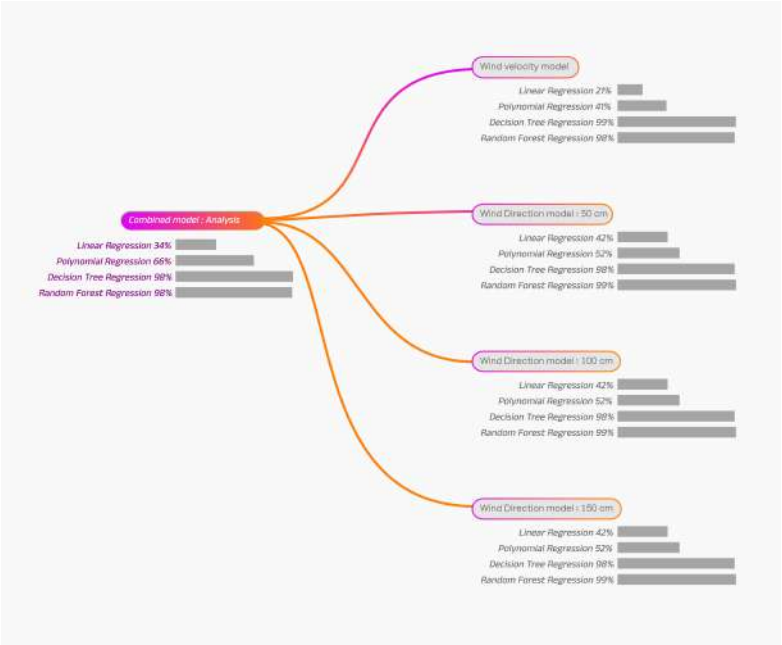


Figure 18: Machine Learning Results

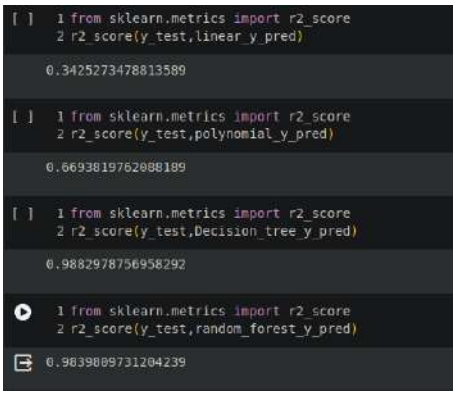


Figure 19: Result

After that the code converted into cpp code. The code which used as .h file. Then the code inserted to Arduino nano ble 33 sense board.The result is 8.23(Wind velocity). There is no change in the output. Our conclusion is that , we only checked the data in indoor. So the Fan has some limitations. it will affect the data output, dataset etc. So, we need an outdoor test experiment. For more details will get from the data.

10.2 Test 2

During the experiment found that found that transducers salvaged from HCSR04 module cannot be acts as transievers. Functionality testing of the roll-switching circuit was conducted using ultrasonic transducers salvaged from HCSR04 and US100 modules. The received signal was monitored using a DSO (Digital Storage Oscilloscope). Selector pins were assigned different values corresponding to the X and Y axes. It was observed that the transducers are not designed for roll switching, as the signals were found to be weaker. Utilizing the DSO, it was determined that the peak-to-peak voltage of the received signal drops to one-third. The peak-to-peak voltage obtained from the transmitter to the receiver was 3.3V. However, the rolls were reversed, i.e., the receiver was transmitting, and the transmitter was receiving; the voltage of the signal dropped to 1V as Vpp. The quality of the signal was also affected. The results are attached. The switching functionality of the multiplexer-based circuit was tested, and it was found to be working. To obtain accurate results, we need transducers that can function as both transmitters and receivers.

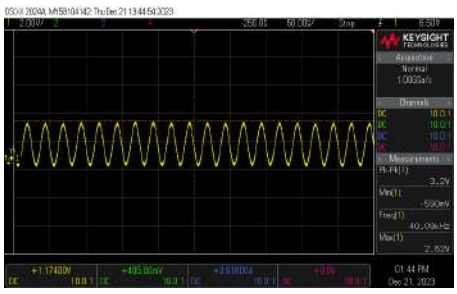


Figure 20: Transmitter to receiver

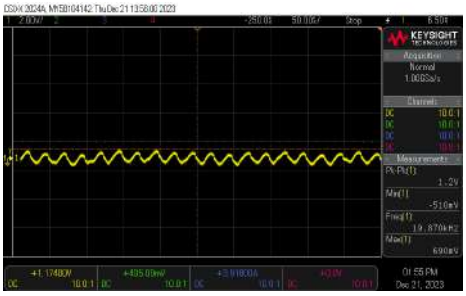


Figure 21: Receiver to transmitter

Schematic diagram of Multiplexer for switching circuit attached here.The track design for switching 4 ultrasonic transducers using a single signal generation & processing circuit and 2 units of dual 4 channel multiplexer is attached herewith.

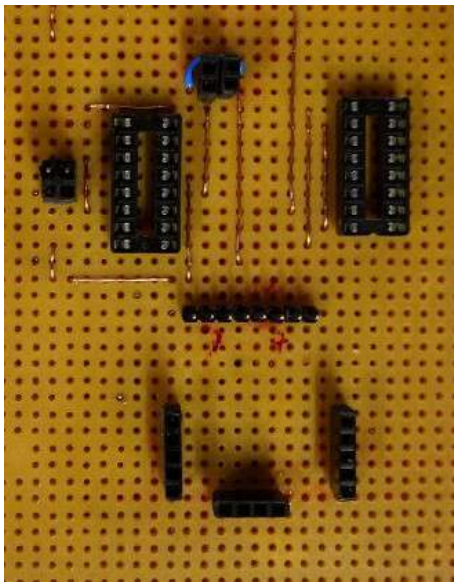


Figure 22: pcb for multiplexer Front view

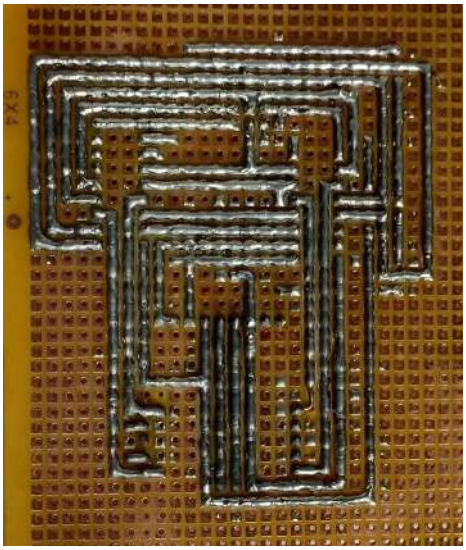


Figure 23: Bottom view

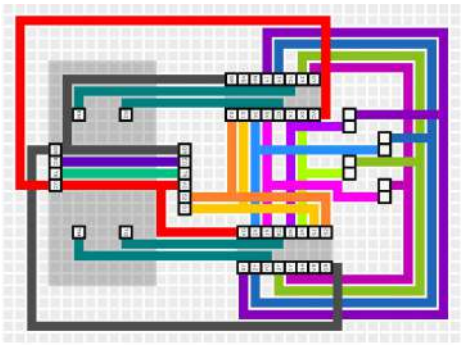


Figure 24: Switching funtionality using multiplexer Dot board

10.3 Test 3 Comparison Ultrasonic transducers

Purpose of the task was to find transducers, which better for the research. For this purpose, 4 ultrasonic transducers are used for the testing. i.e, Multicomp Pro ultrasonic transducer Multicompo ultrasonic transducer Transducer salvaged from HC SR04 Transducer salvaged from US100. From the transievers,

- One connected to Arduino mega 2560 board. and the other transiever connected to DSO.

- Connect signal pin to D10 of the arduino and ground it.
- Program

```
void setup()
{
  pinMode(10,OUTPUT); /*for testing signal pin from the transiever
                        connected to D10 pin of the arduino*/
  tone(10,40000);

}

void loop()
{

}
```

10.4 Multicomp Pro

Figure 25: Multicomp pro

Item	Specification
Part Number	MCUSD16P40B12RO
Construction	Open struct
Using Method	Dual Use
Frequency	40 ± 1kKz
Transmitting Sound Pressure Level	min. 112dB (30cm/10V rms Sine Wave)
Receive Sensitive	min.-74dB/V/ µbar
-6dB Directivity	50°
Aftershock Time	max. 1.5ms
Sensitivity Distance	0.6m to 16m
Capacitance	3,000pF 20% at1kHz
Soldering conditions	350°C to 400°C
Operating Temperature. Range	-30°C to 85°C
Allowable Input Voltage	120Vp-p (40kHz)
Housing Material	Plastic

Figure 26: Specifications

Transducers placed in 0 cm and 20 cm distance and takes the output from DSO.

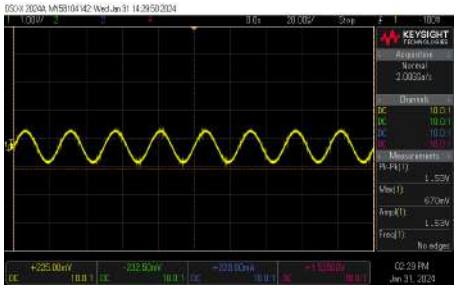


Figure 27: multicomp pro with distance 0cm

Figure 28: specifications

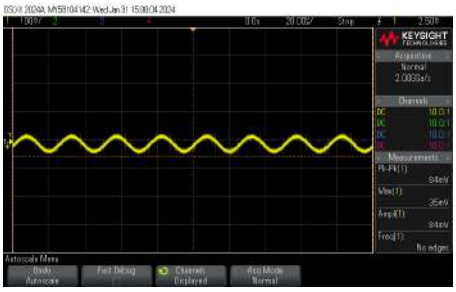


Figure 29: multicomp pro with distance 20cm

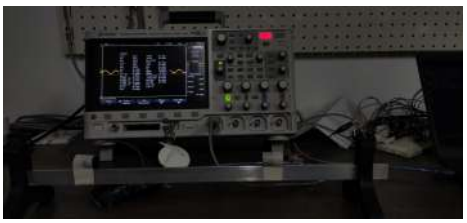


Figure 30: specifications

10.5 Multicomp

Figure 31: Multicomp

Item	Specification
Construction	Water Proof
Using Method	Dual Use
Centre Frequency	58 ± 1kKz
Sound Pressure Level (dB)	≥85 (30cm/10V rms)
Sensitivity (dB)	≥-90dB/V/μbar
Beam pattern	110 × 50°
Capacitance (pF)	2,000 ± 25% at1kHz
Operating Temperature Range	-40°C to 85°C
Storage Temperature Range	-40°C to 85°C
Allowable Input Voltage (Vp-p)	160Vp-p
Housing Material	Aluminium

Figure 32: Specifications

Transducers placed in 0 cm and 20 cm distance and takes the output from DSO.

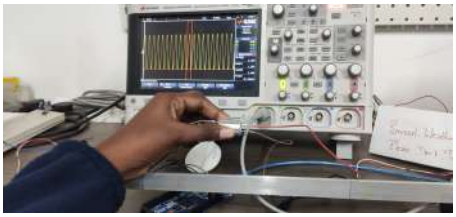


Figure 33: multicomp with distance 0cm

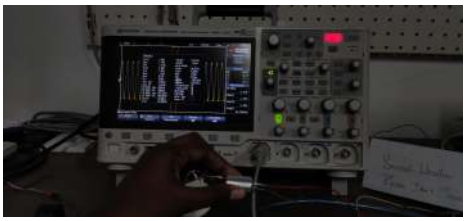


Figure 34: specifications

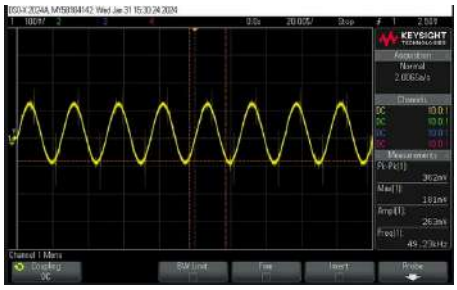


Figure 35: multicomp with distance 20cm

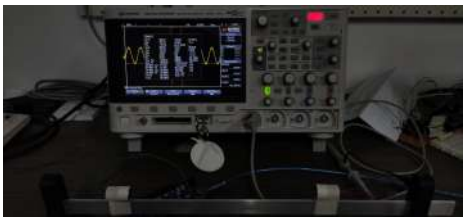


Figure 36: specifications

10.6 Transducers salvaged from HCSR04 Module

The transducers are salvaged from HCSR04 module.It has transmitter and receiver section. During the previous experiments, found that the transmitter and receiver has different peak to peak voltage and other parameters too. Here, check the transmitter to receiver side transducer test and receiver to transmitter test at 0 cm and 20cm.

- Transmitter to receiver

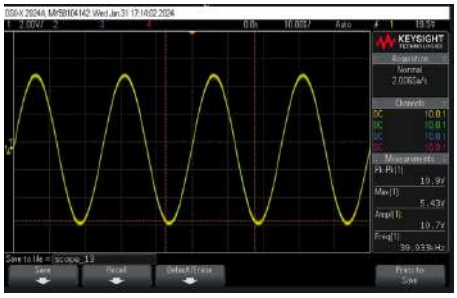


Figure 37: HCSR04 with distance 0cm

Figure 38: specifications

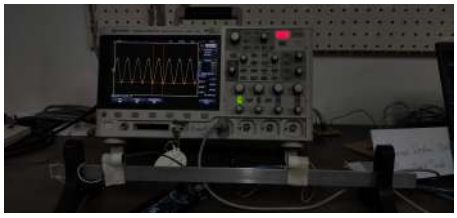


Figure 39: HCSR04 with distance 20cm

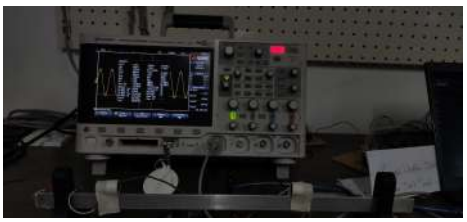


Figure 40: specifications

The transducer modules were found to be marked as R & T respectively by the manufacturers, while in operation, T to R (from HCSR04) arrangements shown here . All measurements are represented by the fig. The waveform generated from the DSO shows in Fig.

- Receiver to transmitter

Figure 41: HCSR04 with distance 0cm

Figure 42: specifications

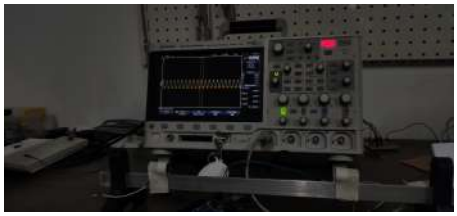


Figure 43: HCSR04 with distance 20cm

Figure 44: specifications

10.7 Transducers salvaged from US100

The transducers are salvaged from US100 module.It has transmitter and receiver section. During the previous experiments, found that the transmitter and receiver has different peak to peak voltage and other parameters too. Here, check the transmitter to receiver side transducer test and receiver to transmitter test at 0 cm and 20cm.

- Transmitter to receiver

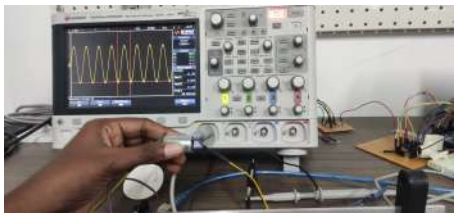


Figure 45: US100 with distance 0cm

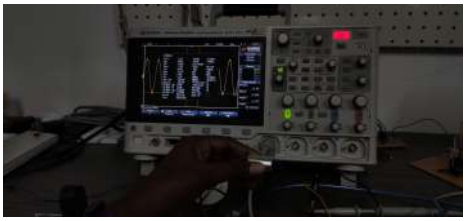


Figure 46: specifications

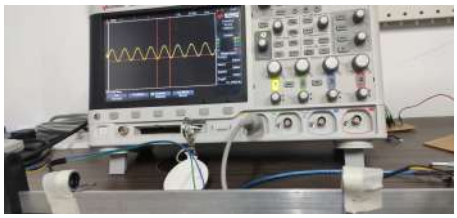


Figure 47: US100 with distance 20cm

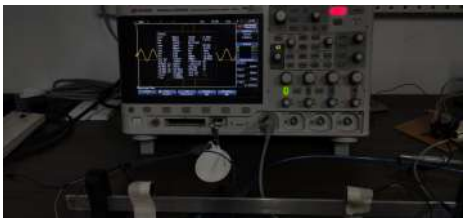


Figure 48: specifications

The transducer modules were found to be marked as R & T respectively by the manufacturers, while in operation, T to R (from US100) arrangements shown above . All measurements are represented by the fig. The waveform generated from the DSO

- Receiver to transmitter

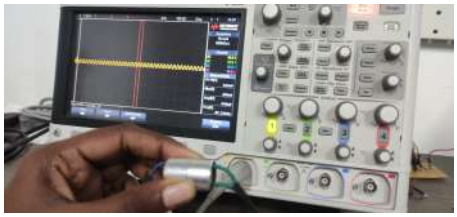


Figure 49: US100 with distance 0cm

Figure 50: specifications

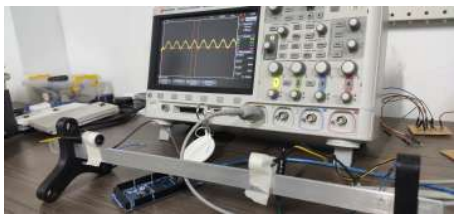


Figure 51: US100 with distance 20cm

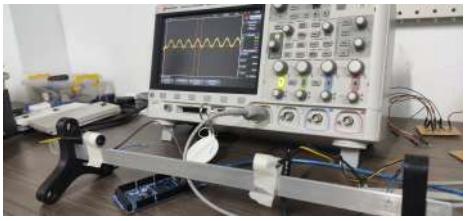


Figure 52: specifications

US100 transducer are arranged very closely.The transducer modules were found to be marked as R & T respectively by the manufacturers, while in operation, R to T (from US100) arrangements shown above . All measurements are represented by the fig. The waveform generated from the DSO shows in Fig

10.7.1 Comparison Results

	Ultrasonic Transducers at distance 0 cm					
(V)	Multicomp Pro	Multicomp	HCSR04 (T to R)	HCSR04 (R to T)	US100(T to R)	US100(R to T
Vpp	1.53	2.49	11.3	2.13	11.3	0.81
Vm	0.67	1.25	5.5	1.02	5.64	0.3
Amplitude	1.53	2.49	11.3	2.13	11	0.64
	Ultrasonic Transducers at distance 20 cm					
	Multicomp Pro	Multicomp	HCSR04 (T to R)	HCSR04 (R to T)	US100(T to R)	US100(R to T)
Vpp	0.084	0.362	1.02	0.416	0.8	0.149
Vm	0.035	0.181	0.46	0.197	0.366	0.094
Amplitude	0.084	0.263	0.9	0.189	0.72	0.067

Figure 53: Comparison of transducers

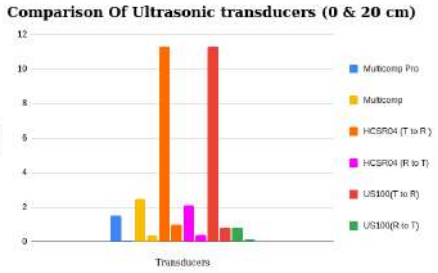


Figure 54: Comparison graph

From the data mentioned above, it can be concluded that the multicomp sensor is apt for our purpose. Out of the above cases it could be concluded that performance of multicomp & multicomp pro transievers are much lesser than the salvaged transducers (HCSR04). The peak to peak voltage of HCSR04 T to R is 1.02V and R to T is 0.416V at 20cm distance.

10.8 Test 4

10.8.1 Amplifier Circuit Design

Implementation of ultrasonic anemometer using 4 ultrasonic transducer. here 4 ultrasonic transducers are connected orthogonal to each other with a distance 20cm. A multiplexer is used for switching the transducers.

The output signal from the ultrasonic transceivers has to be amplified & conditioned in order for efficient signal processing by the micro controllers. Hence, the sinusoidal waves generated by the transceivers has to be transformed to square waves. We developed an amplifier circuit in order to accomplish this; by utilizing LM386.

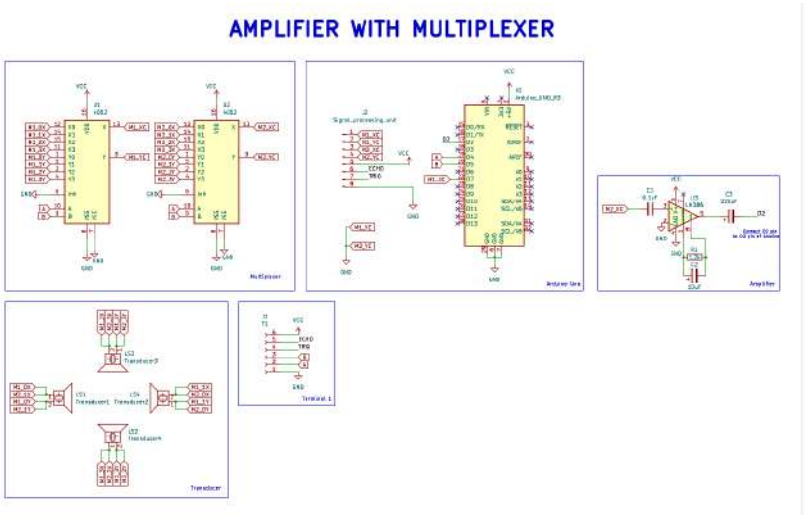


Figure 55: Amplifier circuit ith multiplexer

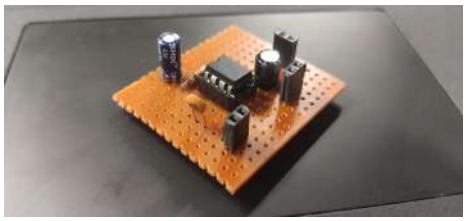


Figure 56: Front view

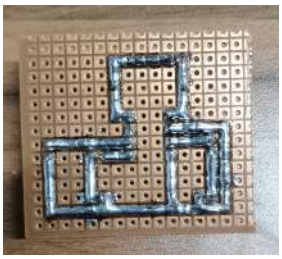


Figure 57: Bottom view

The output from DSO shown below.

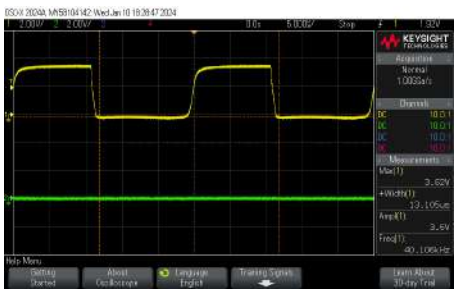


Figure 58: Transmitter to receiver signal strength with amplifier

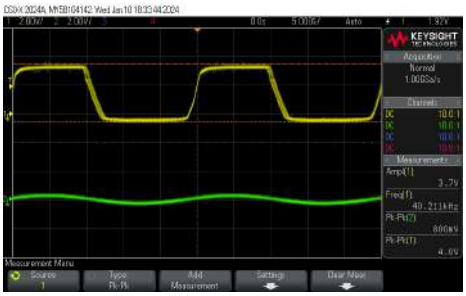


Figure 59: Transmitter to receiver signal strength with and without amplifier.

Here the Peak to Peak voltage of receiver using amplifier circuit is approxi-

mately 3.9V. Also the Amplitude is 3.6V. Without using the amplifier, got the peak to peak voltage 800mV and amplitude as 330mV. Based on the previous experiment, receiver to transmitter section has less peak to peak voltage.Now,

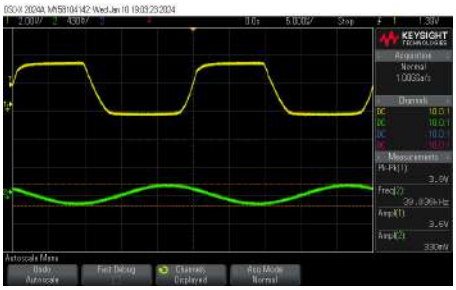


Figure 60: Receiver to transmitter signal strength

10.9 Test 5

10.9.1 Bandpass Filter Simulation Result

A bandpass filter is an electronic circuit that allows a specific range of frequencies to pass through while attenuating frequencies outside that range. It is commonly used in audio and radio frequency applications to selectively filter out unwanted frequencies and isolate the desired signal. The received signal first passes from the amplifier stage to increase the strength of the signal after that it passes through the multiple feedback band pass filter which is having a center frequency of 40 kHz and bandwidth of 10 kHz, to remove the unwanted frequency component. Opamp based multiple feedback band pass filter is designed and simulated on LTspice. First we are planned to design and setup a narrow band pass filter by using one operational amplifier. A narrow band pass filter is nothing but a band pass filter with which the pass band bandwidth is very narrow.
Creating a 40kHz Bandpass Filter with the TL082 IC involves selecting appropriate values for resistors and capacitors based on desired filter parameters such as center frequency(Fc) 40kHz and bandwidth BW =±5kHz).

$$Q_factor[Q] = \frac{CentreFrequency}{Bandwidth} = 4kHz$$

Design Formulas

- C1=C2=C
- R3 = RF
- Gain, Af less than $2Q^2$

$$R1 = \frac{Q}{2\pi F_c C A_f} \tag{8}$$

$$R2 = \frac{Q}{2\pi F_c C(2Q^2 - A_f)} \tag{9}$$

$$R2 = \frac{Q}{\pi F_c C} \tag{10}$$

Take,

- C1 = C2=C = 0.1uf
- Af =1
- Q = 10
- Fc = 40KHz

By applying these values to Eqs, we get the values -

- R1 = 400 ohms
- R2 = 2 ohms
- R3 = Rf = 800 ohms

These values are theoretical we didn't get the Fc of 40Kz during the simulation, so so changes are made in these values to get correct frequency response.

Circuit Diagram

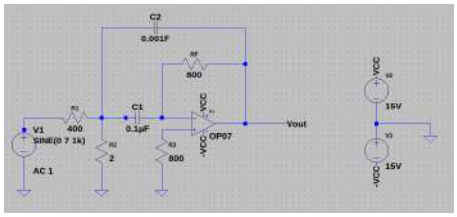


Figure 61: Simulation

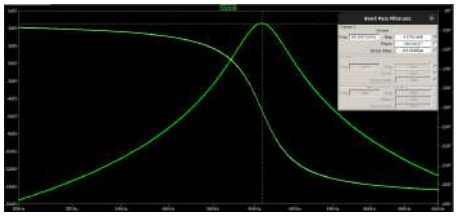


Figure 62: Simulation

10.10 Test 6

Integrating components to the microcontroller. The arduino code using for the research attached in appendix. Data collected in 4 phases. We use pedestrial fan. It has 3 speed level. Fan speed measured using digital anemometer. Tested the ultrasonic anemometer in the wind tunnel test setup and collected the wind data with different fan speed using the digital anemometer and ultrasonic anemometer simultaneously . The dataset and the calculations are stored as csv file Measuring wind speed; 0 wind, speed1, speed2, speed 3.Fan positioned in north to south direction. From theoretical calculations, at N-S direction, there is no speed that will be affected by wind. Wind velocity at N-S =0. But, in practice we get a wind speed in the East to west direction. It becomes 1.9m/s. The final output is;

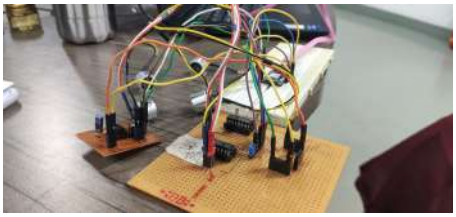


Figure 63: Hardware setup

Fan Speed	Digital Anemometer		Ultrasonic Anemometer	
	North - South	East - west	All Direction	North - South
0	0 m/s	0 m/s	8 m/s to 13 m/s	1 ms to 4 m/s
1	4.5 m/s	1.9 m/s	6 m/s to 13 m/s	2 m/s to 7 m/s
2	5 m/s	1.8 m/s	6 m/s to 12 m/s	2 m/s to 7 m/s
3	5.2 m/s	2.2 m/s	6 m/s to 12 m/s	2 m/s to 7 m/s

Figure 64: Observation

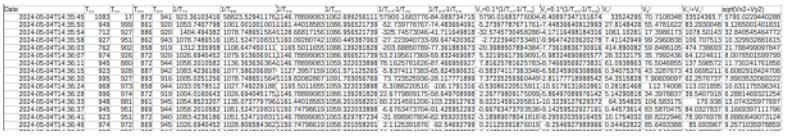


Figure 65: Processed Data

Then the hardware placed inside the wind tunnel setup and the dataset for

each speed level ,filtered the values as per the wind velocity equations. From the dataset, we found that

Fan speed level	Digital anemometer (m/s)		Ultrasonic anemometer(m/s)	
	N-S	E-W	All direction	N-S
0	0	0	8 -12	1 -4
1	4.5	1.9	6 -13	2 -7
2	5	1.8	6 -12	2 -7
3	5.2	2.2	6 -11	2 -7

Figure 66: Observation

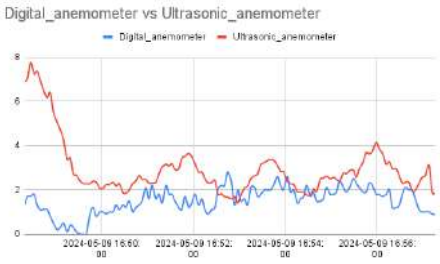


Figure 67: Output

11 Conclusions

The objective of the project was to design and implement a non-mechanical anemometer. Through our efforts, we successfully developed an anemometer capable of detecting both wind velocity and direction. However, in order to meet higher standards of precision, we identified areas for improvement. Moving forward, our focus lies on enhancing the accuracy and efficiency of the product.

12 Bibilography

1. Yadav, V. P., Sinha, A., & Khosla, A. (2017). Design and implementation of ultrasonic anemometer. 2017 4th International Conference on Power, Control & Embedded Systems (ICPCES). doi:10.1109/icpces.2017.8117645

2. Del Valle, M. P., Castelan, J. A. U., Matsumoto, Y., & Mateos, R. C. (2007). Low Cost Ultrasonic Anemometer. 2007 4th International Conference on Electrical and Electronics Engineering. doi:10.1109/iceee.2007.4345008

3. Fan, H. B., Liu, J. L., & Sun, G. Z. (2018). Design of Low-power Ultrasonic Anemometer Based on STM32L476. IOP Conference Series: Materials Science and Engineering, 408, 012036. doi:10.1088/1757-899x/408/1/012036

4. Du, H. Y., & Qin, M. (2014). Design and Experiment of Ultrasonic Anemometer Using TDC-GP2 Time-to-Digital Converter. Key Engineering Materials, 609-610, 1002–1007. doi:10.4028/www.scientific.net/kem.609-610.1002

5. Fernandes, D., Gomes, L., & Costa, A. (2017). Wind speed and direction measurement based on time of flight ultrasonic anemometer. 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). doi:10.1109/isie.2017.8001453

6. Lopes, G. M. G., da Silva Junior, D. P., de Franca, J. A., de Morais Franca, M. B., de Souza Ribeiro, L., Moreira, M., & Elias, P. (2017). Development of 3-D Ultrasonic Anemometer With Nonorthogonal Geometry for the Determination of High-Intensity Winds. IEEE Transactions on Instrumentation and Measurement, 66(11), 2836–2844. doi:10.1109/tim.2017.2714438

7. Han, D., Kim, S., & Park, S. (2008). Two-dimensional ultrasonic anemometer using the directivity angle of an ultrasonic sensor. Microelectronics Journal, 39(10), 1195–1199. doi:10.1016/j.mejo.2008.01.090

8. Arduino official page (https://www.arduino.cc).

13 APPENDIX

13.1 Ultrasonic anemometer code

```
#include <NRF52_MBED_TimerInterrupt.h>
#include "NRF52_MBED_ISR_Timer.h"
#include "nRF52_MBED_PWM.h"

#include "RTClib.h"

#include <SPI.h>
#include <SD.h>

#define TIMER_INTERRUPT_DEBUG         0
#define _TIMERINTERRUPT_LOGLEVEL_     1

#define HW_TIMER_INTERVAL_MS          1
#define TIMER_INTERVAL_1S             1075L



#define sensor_select1                7
#define sensor_select2                8

#define Receiver_pin                  2
#define transimitter_pin              3
#define frequency                     40000

#define max_buffer_size               4
uint8_t buffer_index = 0;

uint16_t time_of_travel[max_buffer_size] = {0};


float dutyCycle = 50.0f;

volatile bool timerFlag = false;
volatile bool status_flag = true;
volatile uint16_t count = 0;
volatile uint64_t signal_start_time = 0;
volatile uint64_t signal_receive_time = 0;

uint16_t pulse = 0;
// chip select
uint8_t A = LOW, B = LOW;
```

```
String Time;

NRF52_MBED_Timer ITimer(NRF_TIMER_2);
NRF52_MBED_ISRTimer ISR_Timer;

mbed::PwmOut* pwm    = NULL;

RTC_DS1307 rtc;

File myFile;

void setup() {
  Serial.begin(115200);
  while (!Serial);
  Serial.println("starting program");
  pinMode(transimitter_pin, OUTPUT);

  pinMode(Receiver_pin, INPUT_PULLUP);

  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);

  digitalWrite(7, A);
  digitalWrite(8, B);

  Serial.print("Initializing RTC...");
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    Serial.flush();
    while (1);
  }
  /*Serial.println("RTC initialization done.");

  Serial.print("Initializing SD card...");

  if (!SD.begin(4)) {
    Serial.println("SD card initialization failed!");
    while (1);
  }
  Serial.println("SD card initialization done.");
*/
  attachInterrupt(digitalPinToInterrupt(Receiver_pin), wind_count, FALLING);

  ITimer.attachInterruptInterval(HW_TIMER_INTERVAL_MS * 1000, TimerHandler);
  signal_start_time = micros();
```

```
  setPWM(pwm, transimitter_pin, frequency, dutyCycle);

  ISR_Timer.setInterval(TIMER_INTERVAL_1S,  ISR_Timer_Interrupt);
}



void loop() {
  if (timerFlag)
  {
    DateTime time = rtc.now();
    Time = time.timestamp(DateTime::TIMESTAMP_FULL);
    timerFlag = false;
    if (signal_receive_time < signal_start_time)
    {
      for (uint8_t i = 0; i < max_buffer_size; i++)
      {
        time_of_travel[i] = 0;
      }
      buffer_index = max_buffer_size;
      A = LOW; B = LOW;
      digitalWrite(7, A);
      digitalWrite(8, B);
    }
    else
    {
      time_of_travel[buffer_index] = signal_receive_time - signal_start_time;
      buffer_index = buffer_index + 1;
      chip_select();
    }
    if (buffer_index == max_buffer_size)

    {
      for (uint8_t i = 0; i < max_buffer_size; i++)
      {
        Serial.print(String(time_of_travel[i]) + ",");
      }
      Serial.print(Time);
      Serial.println("");
      buffer_index = 0;
    }
    status_flag = true;
    signal_start_time = micros();
    setPWM(pwm, transimitter_pin, frequency, dutyCycle);
    ITimer.restartTimer();
  }
```

```
}



void wind_count()
{
  if (status_flag == true)
  {
    signal_receive_time = micros();
    status_flag = false;
  }
  count++;
}

void TimerHandler()
{
  ISR_Timer.run();

}

void ISR_Timer_Interrupt()
{
  stopPWM(pwm, transimitter_pin);
  ITimer.stopTimer();
  pulse = count;
  count = 0;
  timerFlag = true;

}
void chip_select() {
  //selecting transceivers for next transmission and reception
  //need to examine the if else ladder.need clarification of its working and che
  // CHECK POSSIBILITY of using comma separator
  if (A == LOW && B == LOW) {
    A = LOW;
    B = HIGH;
  }
  else if (A == LOW && B == HIGH) {
    A = HIGH;
    B = LOW;
  }
  else if (A == HIGH && B == LOW) {
    A = HIGH;
    B = HIGH;
  }
```

```
  else {
    A = LOW;
    B = LOW;
  }
  digitalWrite(7, A);
  digitalWrite(8, B);
}
```

13.2 Machine Learning code

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')

data = pd.read_csv('/content/drive/MyDrive/AI_models_wind_direction
/combined_data_direction_velocity.csv')

X=data.iloc[:, :-2].values
print(X)



y=data.iloc[:,[4,5]].values
print(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.2,
random_state=1)

from sklearn.linear_model import LinearRegression
linear_regressor=LinearRegression()
linear_regressor.fit(X_train, y_train)

from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_train, y_train)
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=6)
X_poly=poly_reg.fit_transform(X)
lin_reg_2=LinearRegression()
lin_reg_2.fit(X_poly, y)

from sklearn.tree import DecisionTreeRegressor
Decision_tree_regressor=DecisionTreeRegressor(random_state=0)
Decision_tree_regressor.fit(X,y)

from sklearn.ensemble import RandomForestRegressor
random_forest_regressor=RandomForestRegressor(n_estimators=100,random_state=0)
random_forest_regressor.fit(X,y)

linear_y_pred=linear_regressor.predict(X_test)
np.set_printoptions(precision=6)
```

```
precision=np.concatenate((X_test,linear_y_pred,y_test),1)
result1=','.join(map(str,precision))
print(result1)

polynomial_y_pred=lin_reg_2.predict(poly_reg.fit_transform(X_test))
np.set_printoptions(precision=2)
#precision=np.concatenate((X_test,y_pred.reshape(len(y_pred),1),
y_test.reshape(len(y_test),1)),1)#
precision=np.concatenate((X_test,y_test,polynomial_y_pred),1)
result2=','.join(map(str,precision))
print(result2)

Decision_tree_y_pred=Decision_tree_regressor.predict(X_test)
np.set_printoptions(precision=6)
precision=np.concatenate((X_test,Decision_tree_y_pred,y_test),1)
result3=','.join(map(str,precision))
print(result3)

random_forest_y_pred=random_forest_regressor.predict(X_test)
np.set_printoptions(precision=6)
precision=np.concatenate((X_test,random_forest_y_pred,y_test),1)
result4=','.join(map(str,precision))
print(result4)

from sklearn.metrics import r2_score
r2_score(y_test,linear_y_pred)

from sklearn.metrics import r2_score
r2_score(y_test,polynomial_y_pred)

from sklearn.metrics import r2_score
r2_score(y_test,Decision_tree_y_pred)

from sklearn.metrics import r2_score
r2_score(y_test,random_forest_y_pred)

#df=pd.DataFrame(precision)
#df.to_csv('/content/drive/My Drive/
combined_data_Polynomial_Regression.csv', index=False)

#df_verify = pd.read_csv('/content/drive/My Drive/combined_data_Polynomial_Regre
#print(df_verify)

Decision_tree_regressor.predict([[565,590,27.5,57.5]])
```