A
Dissertation
On

Design and development of Data Acquisition Console for Naval Ships using a Microcontroller

Submitted in

partial fulfilment of the requirement

for the award of the degree of

Master of Technology

in

VLSI Design and Embedded System

Submitted by

Naveen Mavi
University Roll No. 2K13/VLS/13

Under the Guidance of

Prof. Jeebananda Panda
Associate Professor,
Department of Electronics and Communication Engineering
Delhi Technological University

Department of Electronics and Communication Engineering
Delhi Technological University
2013-2015

Electronics and Communication Engineering Department
Delhi Technological University
Delhi - 110042
www.dce.edu

CERTIFICATE

This is to certify that the dissertation titled **"Design and Development of Data Acquisition Console (DAC) for Naval Ships using Microcontroller"** is a bonafide record of work done by **Naveen Mavi, Roll No. 2K13/VLS/13** at **Delhi Technological University** for partial fulfilment of the requirements for the degree of Master of Technology in VLSI and Embedded System Design. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Date: ________

(Prof. Jeebananda Panda)
Associate Professor
Department of Electronics and Communication Engineering
Delhi Technological University

ACKNOWLEDGEMENTS

I would like to express my deep sense of respect and gratitude to my project supervisor **Prof. Jeebananda Panda,** Associate Professor, Electronics and Communication Engineering Department, DTU for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

I am also grateful to **Prof. P R Chaddha**, HOD, Electronics and Communication Engineering Department, DTU for his immense support.

I would also like to acknowledge Delhi Technological University for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my colleagues for constantly encouraging me during the completion of work.

Naveen Mavi

University Roll no: 2K13/VLS/13

M.Tech (VLSI and Embedded System Design)

Department of Electronics & Communication Engineering

Delhi Technological University

Delhi – 110042

ABSTRACT

1. The Indian Navy presently operates more than 150 Ship/ Submarines of different class. These platforms are fitted with large no of state of the art systems required for there smooth operation. It is pertinent to mention that any average frigate/ destroyer of Indian navy carries more 500 tones of live ordinance / ammunition onboard. Further the said ordinances is stored in specialized compartments called magazines and the same are required to be monitored on real time basis for various hazards like Fire and flooding.

2. In addition to the above mentioned hazards, the magazine compartments are also required to be monitored for various parameters like temperature. Humidity, gas content etc. to avoid any damage to the ammunition. The present project is an attempt to design a microcontroller based real time magazine compartment monitoring system for naval platforms.

3. In this project a prototype, Data Acquisition Console has been designed using a COTs (commercial off the shelf) microcontroller board like Arduino/ Raspberry pi/ Beaglebone. The DAC (data acquisition console) will be able to pick up data from various environment sensors fitted inside the magazine compartments and transfer the data on real time basis to the (COP) Computer Observation Posts (on a GUI) in the various compartments of the ship on UART/ Ethernet protocol. Further provisions has also been made to set alarms, whenever a particular sensor reading crosses a pre-designated safe limit.

4. In addition to the above mentioned features the following has also be incorporated in the project: -

(a) Acquire and display roll/ pitch/ yaw data of the ship.

(b) Use the acquired roll data to stabilize the radar antennae fitted onboard.

TABLE OF CONTENTS

Certificate	**i**
Acknowledgements	**ii**
Abstract	**iii**
Table of Contents	**iv**
List of Figures	**vii**
List of Tables	**ix**
Abbreviations	**x**
1. Introduction	**1**
1.1Brief about the project	1
1.2Scope of the project	
2. Data Acquisition Console Design	**6**
2.1Ethernet Network onboard Ship	6
2.2Data Acquisition Console	
3. The Microcontroller Board	**9**
3.1 Choice of Evaluation Board	9
3.2 About Arduino Uno R3	9
3.3 Details of Power Pin	10
3.4 Details of Specialized Pins	11
3.5 Details of analog pins	13
3.6 Communication	13
4. Temperature and Humidity Data Acquisition	**14**
4.1 Introduction	14
4.2 About DHT 11	14
4.3 Pin configuration of DHT 11	16

4.4 Hardware Design – Temperature and Humidity. 16

4.5 Software Design – Temperature and Humidity. 19

5. Fire and Smoke Detection in Magazine Compartment 20

5.1 Introduction 20

5.2 About MQ-2 Sensor 21

5.3 Pin configuration of MQ 2 22

5.4 Hardware Design – Gas Reading. 23

5.5 Software Design – Gas Reading 25

6. Collision detection between ordinances 26

6.1 Introduction 26

6.2 About HC SR 04 Sensor 26

6.3 Pin configuration of HC SR 04 28

6.4 Hardware Design – Collision detection Reading 29

6.5 Software Design – Collision detection Reading 31

7. Acquiring Roll/Pitch/Yaw Data Of The Ship And Stabilizing The Radar Antenna 32

7.1 Introduction 32

7.2 About GY 521 33

7.3 Pin configuration of GY 521 35

7.4 About SG 90 36

7.5 Hardware Design 38

7.6 Software design 39

8. Final Design Of The Data Acquisition Console 40

8.1 Introduction 40

8.2 Hardware design of DAC 40
8.3 Reading data on Serial Window 44
8.4 Data Acquisition on Graph 45
8.5 Software for Data Acquisition 47
8.6 Software of Data Display in Graphical Form 47

9. Future Scope 48
References xi

LIST OF FIGURES

Sl.	Fig. No.	Details of the Figure	Page no.
1.	1.1	Magazine Compartment onboard a Submarine	01
2.	1.2	Magazine Compartment Onboard a Ship	03
3.	2.1	Distribution of Ethernet network on a typical Naval ship	06
4.	2.2	Block Diagram of Data Acquisition console	07
5.	3.1	Arduino UNO R3 board	09
6.	3.2	STM32F4 Discovery board	09
7.	3.3	Detailed schematic of Arduino UNO R3	10
8.	4.1	DHT 11 sensor	14
9.	4.2	Schematic depiction of the circuit connections	17
10.	4.3	Hardware Implementation for the Temperature and Humidity data acquisition	18
11.	4.4	Temperature and Humidity Data Read on the Serial port	18
12.	5.1	Smoke sensor MQ -2.	20
13.	5.2	Schematic representation for the Smoke detection	23
14.	5.3	Hardware Implementation for the Smoke detection	24
15.	5.4	Software implementation for the Smoke detection	24
16.	6.1	Ultrasonic Sensor HC-SR04	26
17.	6.2	Schematic representation for Collision detection data	29
18.	6.3	Hardware Implementation for Collision detection data	30
19.	6.4	Software implementation for Collision detection data	30
20.	7.1	Three dimensional motion of the ship	33
21.	7.2	GY 521 sensor	34
22.	7.3	SG 90 servo	37
23.	7.4	Schematic representation for Radar stabilization	39

24.	8.1	Schematic representation of the DAC	42
25.	8.2	Hardware Implementation of the DAC	43
26.	8.3	Software Implementation of the DAC	45
27.	8.4	GUI for graphical display of sensor data	46

LIST OF TABLES

Sl.	Table No.	Details	Page no.
1.	1.1	Scope of project	04
2.	3.1	Power Pins of Arduino Uno board	11
3.	3.2	Details of specialized pins of Arduino Uno board	12
4.	4.1	Specifications of DHT 11 sensor	15
5.	4.2	Pin configuration DHT 11 sensor	16
6.	5.1	Specifications of DHT 11 sensor	21
7.	5.2	Pin configuration MQ 2 sensor	22
8.	6.1	Specifications of HC SR- 04 sensor	28
9.	6.2	Pin configuration HC SR- 04 sensor	28
10.	7.1	Pin configuration GY 521 sensor	35
11.	8.1	Connection pin Details - DAC	40
12.	8.2	Alarm limits of sensor data	44
13.	8.3	Details of GUI graph	46

ABBREVIATIONS

SAM	Surface to Air Missile
SSM	Surface to Surface Missile
ASR	Air Surveillance Radar
DAC	Data Acquisition Console
COTS	Commercial of the shelf
COP	Computer Observation Post
LAN	Local Area Network
LED	Light Emitting Diode
IDE	Integrated Development environment
GUI	Graphic user Interface
IN	Indian Navy
IMU	Inertial Measurement Unit
DMP	Digital Motion Processor

CHAPTER 1
INTRODUCTION

1.1 Brief about the Project.

The Indian navy presently operates more than 150 ships/ submarines of different class. These ships are fitted with large no of state of the art weapon systems required for the undertaking various combat roles. The weapon systems primarily comprise of the following type of ordinances: -

(a) SAM (surface to air missiles).
(b) SSM (surface to surface missiles)
(c) Anti submarine rockets (ASR)
(d) Torpedoes.
(e) 70-80 mm shells for the anti aircraft guns.

Fig 1.1 – Magazine Compartment onboard a Submarine

It is pertinent to mention that any average frigate/ destroyer of Indian navy carries more **500 tones of live ordinance / ammunition** onboard. Further the said ordinances is stored in specialized compartments called magazines and the same are required to be monitored on real time basis for various hazards like Fire and flooding.

In addition to the above mentioned hazards the magazine compartments are also required to be monitored for various parameters like temperature. Humidity, gas content etc to avoid any damage to the ammunition. The present project is an attempt to design a microcontroller based real time magazine compartment monitoring system for naval platforms. In the said project an attempt has been made to understand the practical problem of monitoring the proper stowage of live ordinances and ammunitions onboard naval ships.

In this project a prototype, Data Acquisition Console has been designed using a COTs (commercial off the shelf) microcontroller board. The DAC (data acquisition console) will be able to pick up data from various environment sensors fitted inside the magazine compartments and transfer the data on real time basis to the (COP) Computer Observation Posts in the various compartments of the ship on Ethernet based ship LAN.

The DAC will also set alarms (LEDs) when the environment parameters will cross predefined values. Further the Ethernet/CAN based Data acquisition console board can act as a platform for communication between various industrial interfaces over a common protocol. This project involves interfacing various sensors for acquiring Temperature, distance, acceleration, etc on real time basis.

Fig 1.2 – Magazine Compartment Onboard a Ship

A complete chain of data acquisition and display will be shown using open source softwares. Arduino IDE has been used for programming micro-controllers and Processing software has been used for developing GUI for graphical representation of processed data over USB protocol.

The above mentioned project can be further be developed as a single-ended hardware for wide variety of applications onboard naval ships. Various engineering systems used onboard such as Fire-main pumps, Air conditioning and Refrigeration plants, Diesel Alternators etc require to run 24X7 and hence need to be switched between Primary and Secondary systems after specified time or in case of any eventuality such as overheating or drawing excessive current. Presently, a constant monitoring by watch-keeper is required to maintain these systems. Identifying some critical conditions like those stated above and adding them to microcontroller's interrupt service can allow human free monitoring of the various machinery systems onboard a ship.

Further, the same board can be also used as a low cost repeater for GPS, EM LOG and Gyroscope systems Also, a ship's Roll, Pitch indicator for precise landing of aircrafts onboard ship can be designed using same hardware by interfacing accelerometer. Similarly, many more such applications can be designed over this interface board.

1.2 Scope of the Project.

The scope of this project is quite wide and requires extensive testing before being actually implemented for use onboard military ships. Hence, the project has been developed in stages and more complexity and functions have been added as we proceed further. The following is the break up of the scope of this project: -

SL.	Project name	Scope of project
(a)	Minor project 2	Understanding the Arduino UNO 3 board, which includes understanding pin configuration, data formats supported by the board etc. Implementing basis hardware functions like controlling LED, reading data from potentiometer etc. Reading data from environment sensors like LM 35, gas sensor MQ2, ultrasonic sensor HC-SR04, humidity sensor etc. in solo

(b)	Major project 1	Reading data from environment sensors like LM 35, gas sensor MQ2, ultrasonic sensor HC-SR04, humidity sensor etc Transferring the data in UART format to the PC displaying the received data packets in graphical form on real time basis
(c)	Major project 2	Reading data from additional sensors like GY 521 ICs like roll, pitch and yaw Using the roll, pitch and yaw data to stabilize a ship fitted radar antenna

Table 1.1 – Scope of Project

CHAPTER 2
DATA ACQUISITION CONSOLE DESIGN

2.1 Ethernet Network onboard ship

Every Naval vessel is installed with a military grade LAN to facilitate fast and efficient flow of data between the different Offices/ compartments onboard the ship. The typical distribution of Ethernet network onboard a ship is as shown in figure below. The aim of project is to utilize this existing Ethernet network for data acquisition from various locations across the ship. This would save an extensive task of cabling across decks and bulkheads and thereby preserve the watertight and gas tight integrity of the vessel.

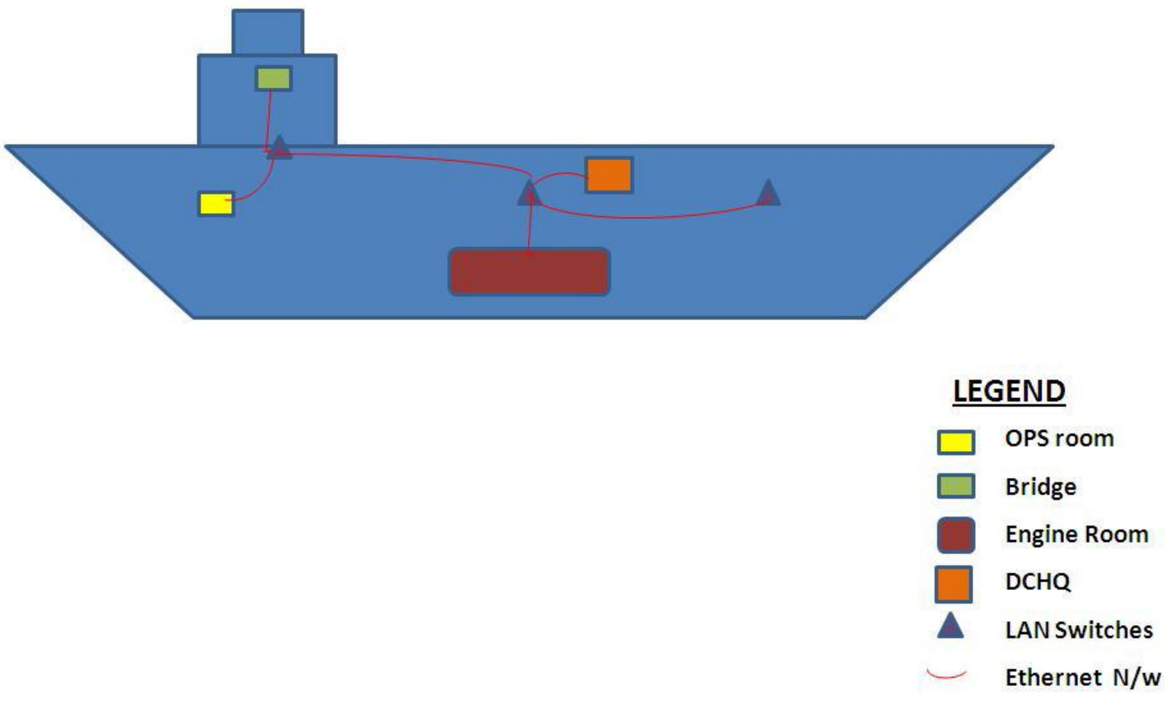


Figure 2.1 Distribution of Ethernet network on a typical Naval ship

The Ethernet network is distributed all across the ship with LAN switches placed at all major locations. This network is very lightly loaded with less than 100 nodes connected over ship's network. Hence, there lies a great scope to utilize this network for data acquisition from distant locations and porting data over some server from where it can be accessed at various ships locations.

2.2 Data Acquisition Console

The project aims at interfacing various sensors and devices over different communication protocols on a single board and transmitting the collated data over USB protocol which can be further transmitted over Ethernet/CAN protocol to be available all across the ship.

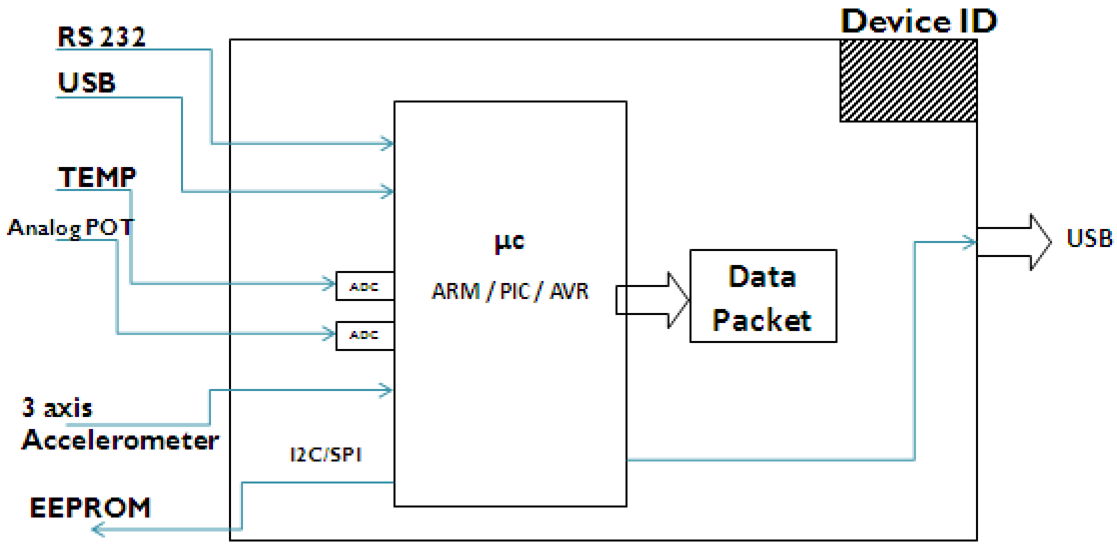


Figure 2.2 Block Diagram of Data Acquisition console

There are various systems spread onboard ship that transmit data in different formats and over different protocols. Few important data interfaces are listed below:

(a) GPS data over RS 232 / USB interface in NMEA format
(b) Gyro Compass Data over RS 232 interface in NMEA format
(c) Temperature & Pressure data interfaced to 10 bit ADC
(d) Fire fighting and Flooding control data from all compartments

(e) Engine / Alternator Parameters from engine control panel

(f) Various video outputs (from ECDIS, RADAR display, SONAR Display, MCR panel) through VGA port

(g) Camera interfaced to the console to visually monitor a ship’s compartment

CHAPTER 3
THE MICROCONTROLLER BOARD

3.1 **Choice of evaluation board**

There are various evaluation boards available nowadays that can be programmed using USB interface. Some of these evaluation boards are shown below:

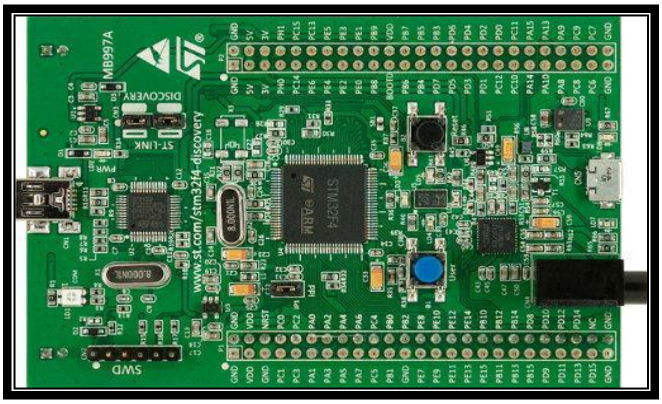


Figure 3.1 Arduino UNO R3 board

Figure 3.2 STM32F4 Discovery board

We are using Arduino UNO R3 board for implementation of the project since the board is use friendly and cheap.

3.2 **About the ARDUINO UNO R3 board.**

The Arduino Uno is a ATmega328 microcontroller based board. It has Fourteen digital IP/OP pins (of which Six can be used as PWM outputs), Six analog I/P, One 16 MHz crystal oscillator, a USB connector, a power supply jack, an ICSP header, and a reset switch. It contains almost everything to support the microcontroller. we can simply connect it to a computer using a USB cable or power it on with a AC-DC adapter or battery to get started. The Uno board differs from all previous version boards since it does not have the FTDI USB-to-serial driver chip. Instead of that it is fitted with the Atmega8U2 programmed as a USB-to-serial converter. Detailed schematic of the board is shown in figure 3.3

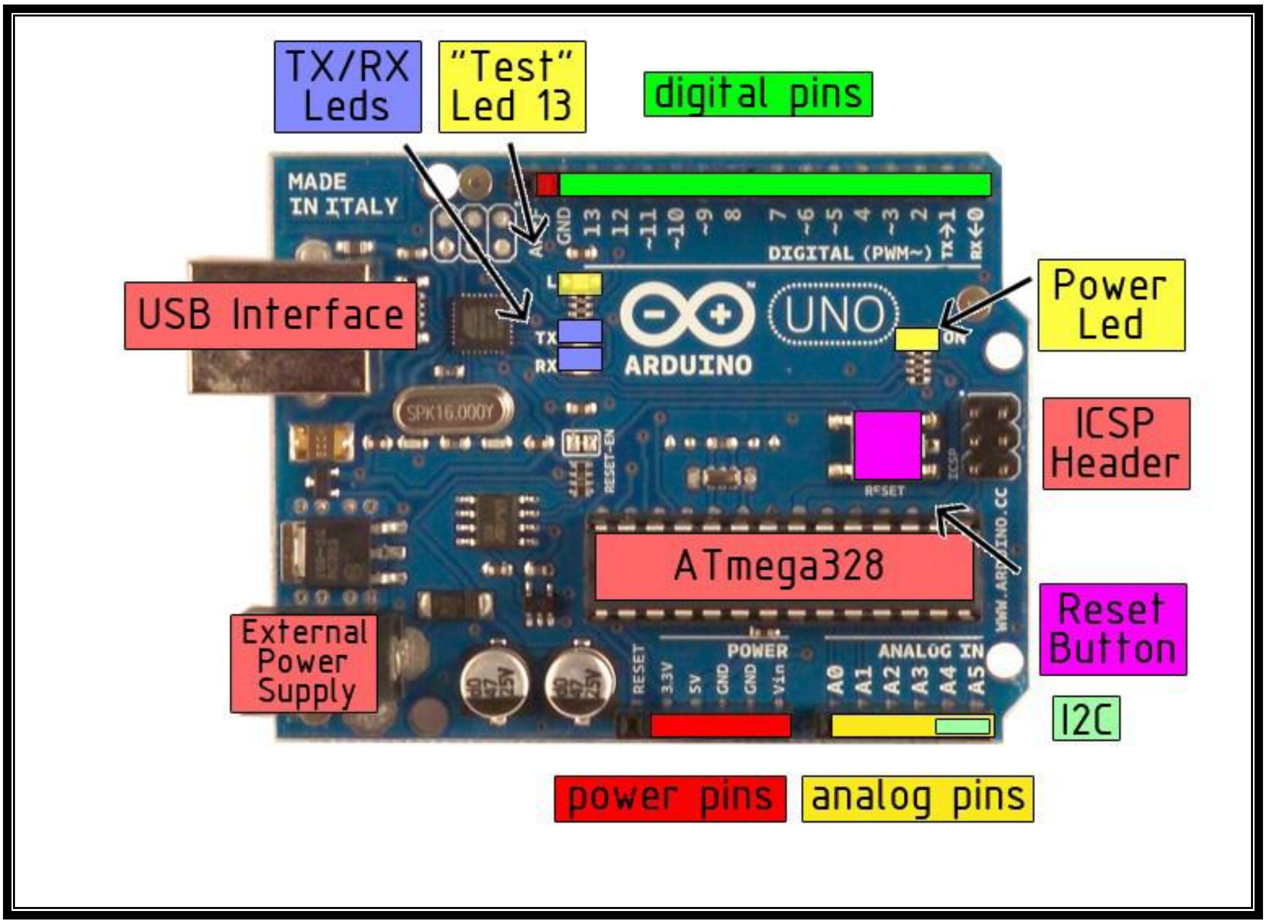


Fig 3.3 – Detailed schematic of Arduino UNO R3

3.3 **Details of the Power Pins**. The Arduino Uno can be powered on through the USB port or using an external power supply source. The power supply source is selected automatically by the board. External power can be taken either from an AC-DC adapter or from a battery. The adapter can be connected by connecting a center positive plug into the board's power socket. Supply lines from the battery can be inserted into *Gnd* (ground) and *Vin* (Voltage in) pin heads of the Power connector. The board can function on an external supply of about 06-20 volt. However if supplied with less than 7V, the 5V pin may supply less than five volts and the board may become unstable. If using a voltage more than 12V, the voltage regulator may get over heated and may further damage the board. Therefore the recommended range of supply is 07-12 volts. The power pins of the board are as follows: -

SL.	**Name of Pin**	**Details**
(a)	**VIN**	The input voltage to the Arduino board when it uses an external power source (and not 05 volt from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage through the power jack, access the same through this pin.
(b)	**5V**	This is a regulated power supply used to power the microcontroller and other components available on board. This either comes from VIN pin via the on-board regulator, or can be supplied from USB or any another regulated 5 Volt supply.
(c)	**3V3**	This is a 3.3 volt power supply generated by the on-board fitted regulator. Maximum current drawn is about 50 mille Amp.
(d)	**GND**	Ground pins (0 volt).

Table 3.1 - Details of the Power Pins of Arduino UNO Board

3.4 **Details of the specialized Pins**. The Atmega328 has an inbuilt 32 Kilobyte of flash memory for storing the code (of which 0.5 Kilobyte is used by boot loader). It has also 2 Kilobyte of Static RAM and 1 Kilobyte of EEPROM. All the 14 digital pins on the board can be used as an IP/OP, using functions like pinMode(), digitalWrite(), and digitalRead(). All the pins operate at 5 volt. Each of the pin can send/ receive a maximum of 40 mA current and are fitted with an internal pull-up resistor (by default disconnected) of 20-50 kilo Ohm. In addition, the following pins have the specialized functions: -

SL.	Name of Pin	Details
(a)	**Serial: Pin no '0'/(RX) and '1'/(TX)**	These pins are used to receive (RX) and transmit (TX) TTL serial data. Further These pins are soldered to the corresponding pins of ATmega8U2 USB-to-TTL Serial chip.
(b)	**External Interrupts: Pin no. 2 and 3**	These pins can be configured to trigger/initiate an interrupt on a low value, rising/ falling edge, or on a change in value.
(c)	**PWM: 3, 5, 6, 9, 10, and 11**	Provide 8-bit Pulse Width Modulation O/P using the analogWrite() function.
(d)	**SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)**	These pin support SPI communication. Although the protocol is supported by the underlying hardware, but is not currently part of/included in the Arduino language.
(e)	**LED: 13**	This is an in-built LED connected to pin no 13. When the pin is set to HIGH value, the LED glows and when the pin is set LOW, it is switched off
(f)	**I2C: 4 (SDA) and 5 (SCL)**	These pin Support I2C (TWI) communication with the help of Wire library.
(g)	**AREF**	This pin gives Ref voltage for the analog I/P. the pin is Used with analogReference()

		function.
(h)	**Reset**	The microcontroller can be reset by pressing this button. Also used to add reset button to shields which block the one available on board.

Table 3.2 - Details of the Specialized Pins of Arduino UNO Board

3.5 **Details of the Analog Pins**. The board has six analog I/P, each of which can provide Ten bits of resolution i.e. a total of 1024 different value). By default the Analog pins can measure voltages of 0-5V. However provision is there to change the upper end of their range with the AREF pin and analogReference() function.

3.6 **Communications.** The Board has a number of facilities to communicate with a computer/ another Arduino/ other Microcontrollers. The ATmega328 microcontroller provides UART TTL (5V) serial communication mode and the same is available on digital pins 0 (Receiver) and 1 (Transmitter). An ATmega8U2 on the board converts this serial communication over USB and appears as a virtual com port to the software running on computer. The '8U2 firmware on the board uses standard USB COM drivers, and therefore no external driver is required. However, for on Windows platforms, an *.inf file is required. The Arduino IDE software includes a serial monitor facility which allows simple textual data to be sent/received to the Arduino board. The RX/TX LEDs on the board will be flashed whenever the data is being transmitted via the USB-to-serial chip and USB connection to the computer.

CHAPTER 4

TEMPRATURE AND HUMIDITY DATA ACQUISITION

4.1 Introduction.

All the ordinances stored in the magazine compartment onboard a naval ships are required to be kept in environment with temperature's below ***40-45 degree Celsius*.** Any further increase in the temperature might render the explosives unstable leading to explosion. Further the ordinances are also required to be maintained below a humidity level of ***60 – 65 %*** or the ordinance will be rendered defective and will not explode when required to be triggered. The acquisition of temperature and data has been implemented using low cost DHT 11 sensor which is readily available in the market. Further an alarm check in the form of LED indication has been incorporated in the design so that when ever the temperature and humidity crosses the above mentioned limits the ships crew will be informed to undertake corrective necessary action.

4.2 **About DHT 11**.

Fig. 4.1 - DHT 11 sensor.

The DHT 11 Sensor contains a Temperature - Humidity sensor complex which gives a calibrated digital O/P signal. it ensures high reliability and excellent stability by using an exclusive digital signal acquisition technique and temperature & humidity sensing technology. This sensor contains an resistive-type humidity measurement component and an NTC temperature measurement component. it connects to a high performance 8-bit microcontroller and offers excellent quality, fast response, interference proof ability and cost-effectiveness. The specifications of the sensor are as follows: -

Sl.	**Parameters**	**Conditions**	**Minimum**	**Typical**	**Maximum**
Humidity					
1.	Resolution	---	1%RH	1%RH	1%RH
2.	Repeatability	---	---	±1%RH	---
3.	Accuracy	25 deg	---	±4%RH	---
4.		0-50 deg	---		±5%RH
5.	Interchangeability	Fully interchangeable			
6.	Measurement Range	0 deg	30%	---	90%
		25 deg	20%	---	80%
		50 deg	20%	---	80%
7.	Response Time (Seconds)	1/e(63%)25 ℃, 1m/s Air	6 s	10 s	15 s
8.	Hysteresis	---	---	±1%RH	---
9.	Long-Term	Stability Typical	---	±1%RH/ year	---
Temperature					
10.	Resolution	---	1 deg celsius	1 deg celsius	1 deg celsius
		---	8 Bit	8 Bit	8 Bit

11.	Repeatability	---	---	±1 deg celsius	---
12.	Accuracy	---	±2 deg celsius	---	±1℃
13.	Measurement range	---	0 deg	---	50 deg
14.	Response Time (sec)	---	6 s	---	30 s

Table 4.1 - Specifications of DHT 11.

4.3 Pin Configuration - DHT 11.

In order to identify the pins of this sensor, turn the back side of the sensor (the one shown in figure above towards yourself) and start numbering the pins from left to right. The pin configuration will be as follows: -

Sl.	**Pin no**	**Function**
(a)	1	Vcc supply to be applied (5v)
(b)	2	Data out (to be given to Arduino)
(c)	3	Not connected
(d)	4	Ground

Table 4.2 – pin configuration of DHT 11

4.4 Hardware Design – Temperature and Humidity.

The hardware design for acquiring temperature and humidity data using DHT 11 sensor has been enumerated in Fig. 4.2. Further photographs of the actual realization of the same undertaken by the student has been depicted in Figure 4.3 and 4.4.

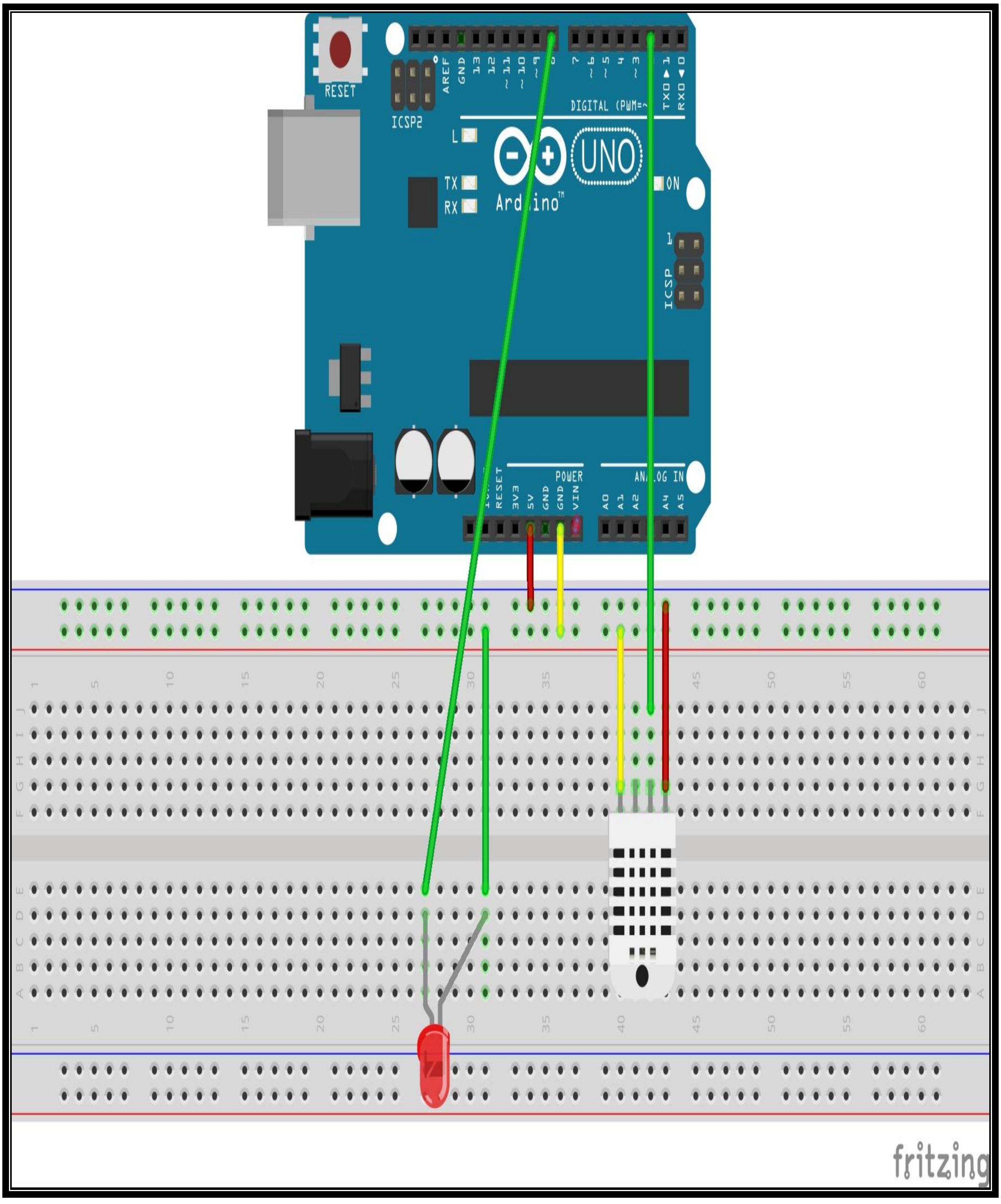


Fig. 4.2 Schematic depiction of the circuit connections

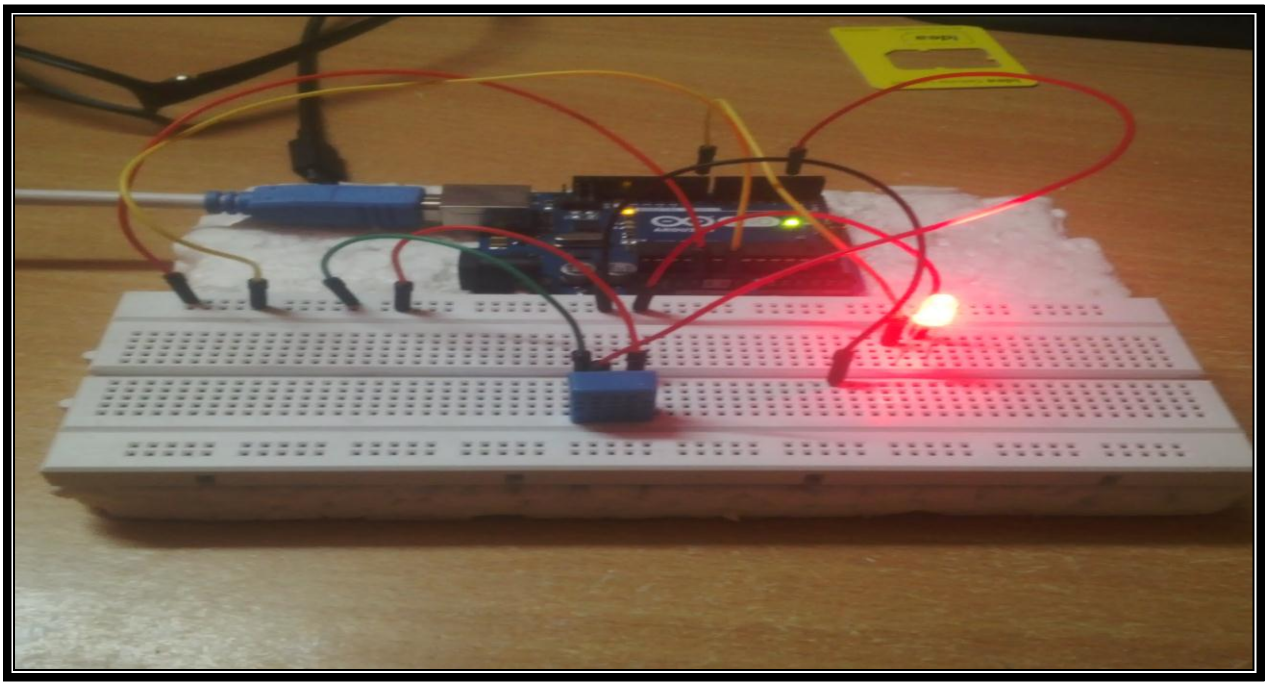


Fig 4.3 - Hardware Implementation for the Temperature and Humidity data acquisition

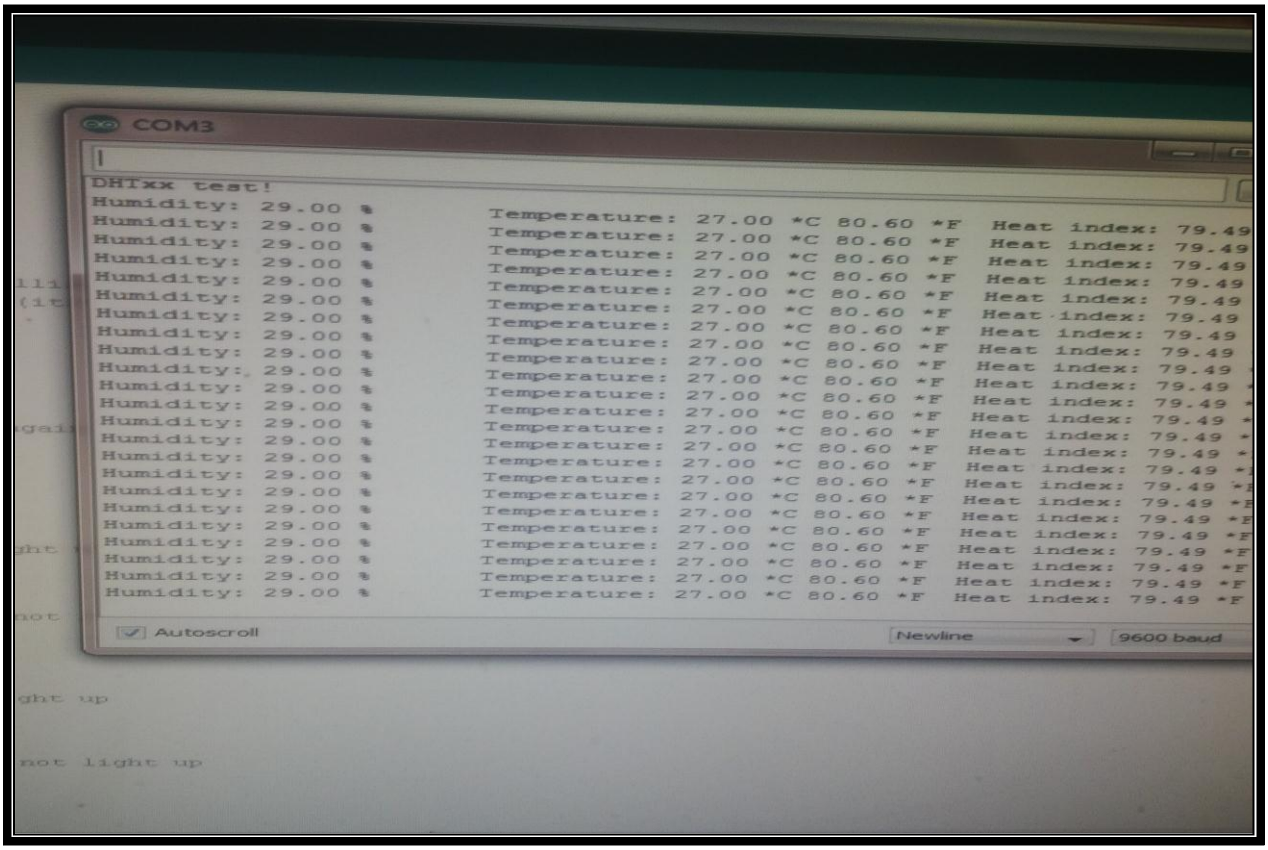


Fig 4.4 –Temperature and Humidity Data Read on the Serial port

4.5 Software – Temperature and Humidity

The code for the above mentioned design has been implemented using C language in Arduino IDE. Further the detailed code for the implementation has been provided separately on a Compact Disk.

CHAPTER 5
FIRE/ SMOKE DETECTION IN MAGAZINE COMPARTMENT

5.1 Introduction.

Most of the Ships used across the world are diesel propelled and hence are required to carry large amount of diesel for long deployment at the sea. Further It is believed that the average quantity carried by any frigate/ destroyer varies from 1000-3000 tons. It is pertinent to note that this fuel is required to be stored in various tanks onboard and further required to be pumped through pipelines across the ship. Hence it is an accepted fact that there is always a perennial risk of fire onboard.

In view of the above mentioned risk there is requirement for continuous monitoring of the magazine compartments for smoke. Further the concentration of smoke particles are required to be maintained below ***380 ppm***. Any further increase in the concentration can be understood as a fire in the compartment. Further an alarm check in the form of LED indication has been incorporated in the design so that when ever the smoke concentration crosses the above mentioned limits the ships crew will be informed to undertake necessary action.

Fig 5.1 - smoke sensor MQ -2.

5.2. About MQ 2 sensor.

MQ2 is a flammable gas/ smoke sensor which detects the concentration of combustible gases and smoke in the air and provides its O/P as analog voltage. The sensor can measure gas concentrations in the range of 300 to 10,000 ppm. The sensor can operate in a temperatures range of -20 to 50°C and consumes less than 150 mA of current at 5 V input supply.

We are required to connect a 5 volt supply across the heating (H) pins of the sensor to keep it hot enough to function correctly. Connecting five volts at either of the A or B pins will make the sensor generate an analog voltage at the other pins. A resistive load put between the O/P pins and ground is used to set the sensitivity of the detector. The resistive load should be calibrated for your particular application using the equations in the datasheet, but a good starting value for the resistor is 20 kΩ. The specifications of the sensor are as follows: -

Sl.	Symbol	Parameter name	Technical condition	Remarks
Standard operation Conditions.				
1.	Vc	Circuit voltage	5 Volt±0.1	AC/ DC
2.	Vh	Circuit voltage	5 Volt±0.1	AC/ DC
3.	RL	Load resistance	can be adjusted	
4.	RH	Heater resistance	33 ohm±5%	Room Temperature
5.	PH	Heating consumption (power)	less than 800 Milliewatt	---
Environment Conditions.				
6.	Tao	Using Temperature	-20℃ to 50℃	
7.	Tas	Storage Temperature	-20℃ to 70℃	
8.	RH	Relateive humidity	less than 95% Rh	
9.	O2	Oxygen concentration	21% (standard condition) Oxygen	minimum value is over 3%

<table>
<tr><td></td><td></td><td></td><td>concentration can affect the sensitivity</td><td></td></tr>
<tr><td colspan="5">Sensitivity Characteristics.</td></tr>
<tr><td>10.</td><td>Rs</td><td>Sensing Resistance</td><td>3KΩ to 30KΩ (1000ppm iso-butane)</td><td rowspan="4">Detecting concentration scope : 200-5000ppm for LPG and propane

300-5000ppm for butane

5000-20000ppm for methane
300-5000ppm for H2

100-2000ppm for Alcohol</td></tr>
<tr><td>11.</td><td>α (3000/1000) isobutane</td><td>Concentration Slope rate</td><td>Less than/equal to 0.6</td></tr>
<tr><td>12.</td><td>Standard Detecting Condition</td><td colspan="2">Temprature - 20℃±2℃
Vc- 5V±0.1
Humidity- 65%±5%
Vh - 5V±0.1</td></tr>
<tr><td>13.</td><td>Preheat time</td><td colspan="2">More than 24 hour</td></tr>
</table>

Table 5.1 – Specification of MQ 2 sensor

5.3 Pin Configuration – MQ 2.

The pin configuration will be as follows: -

Sl.	Pin no	Function
(a)	1	Analog data (to be given to Arduino analog pin)
(b)	2	Digital data out (to be given to Arduino digital pin)
(c)	3	Ground (to be connected to 0v)
(d)	4	Vcc (to be connected to 5v)

Table 5.2 – pin configuration of MQ 2

5.4 **Hardware Design.** The hardware design for acquiring gas sensor data using MQ 2 sensor has been enumerated in Fig. 5.2. Further photographs of the actual realization of the same undertaken by the student has been depicted in Figure 5.3 and 5.4.

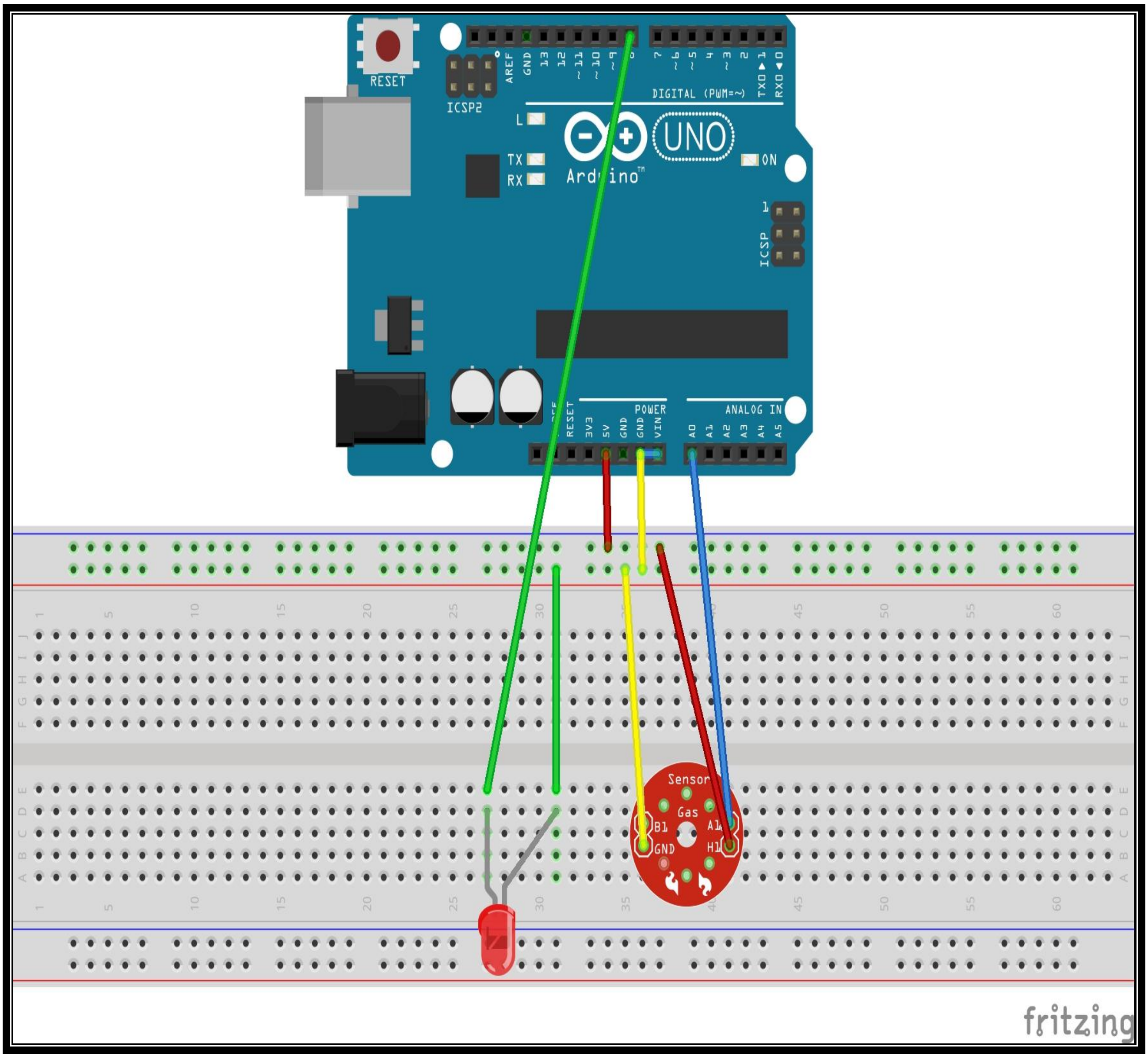


Fig 5.2 – Schematic representation for the Smoke detection

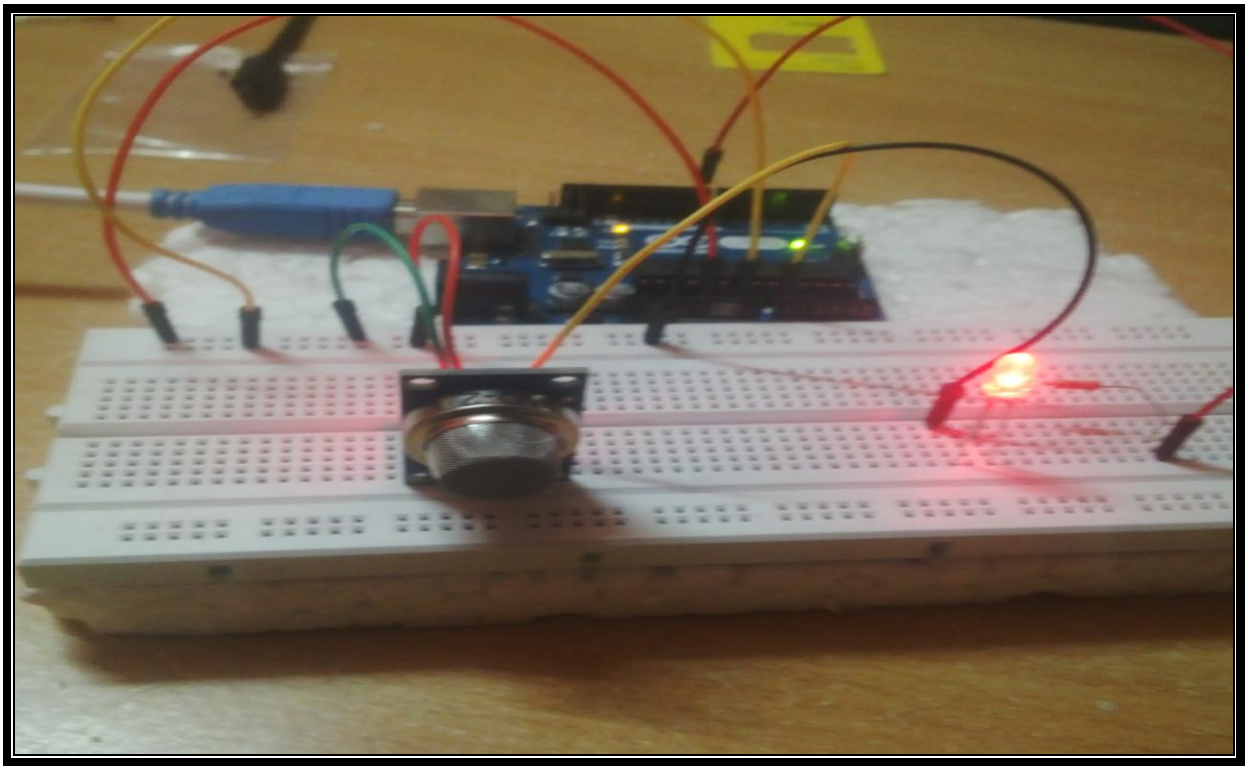
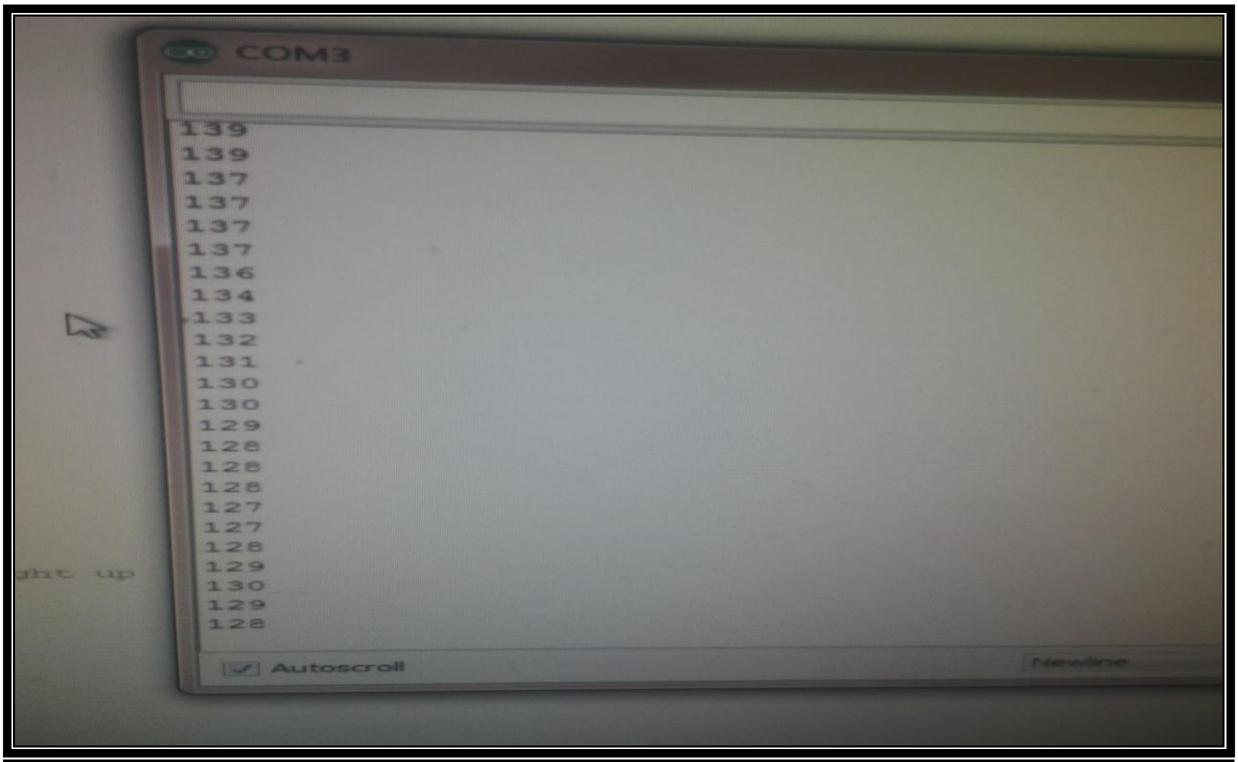


Fig 5.3 - <u>Hardware Implementation for the Smoke detection</u>



<u>Fig 5.4 Smoke sensor MQ-2 data Read on the Serial port</u>

5.5 **Software – Gas data**

The code for the above mentioned design has been implemented using C language in Arduino IDE. Further the detailed code for the implementation has been provided separately on a Compact Disk.

CHAPTER 6
COLLISION DETECTION BETWEEN ORDANACES

6.1 Introduction.

In addition to the application of magazine compartment monitoring All the ordinances stored in the magazine compartment are secured properly with latches to avoid any king of movement during rough weather. Further the ordinance is required to be secured properly to avoid any kind of collision among the same. A spacing of ***4-6 inches*** between the ordnances is considered to be safe and healthy. The above mentioned feature of collision detection has been also incorporated in the DAC design using a HC – SR 04 sensor (ultrasonic sensor)

Figure 6.1 - Ultrasonic Sensor HC-SR04

6.2 About the sensor.

Ultrasonic sensors work on the principle of a sonar in which distance of a target is evaluated by interpreting the echoes of the ultrasonic sound waves. The Ultrasonic transmitter emits an ultrasonic wave in a particular direction, and starts measuring the time when the wave is launched. Ultrasonic waves would spread in the air, and then would return back post

collision with an obstacles on its way. Finally, the ultrasonic receiver will stop timing when it receives the reflected wave. The velocity of Ultrasonic wave in the air is 340m / s. Based on the time recorded (**t)**, we can calculate the distance (s) between the obstacle and the transmitter, i.e
s = 340 X t / 2, which is also called the time difference distance measurement principle.

The ultrasonic distance measurement uses the already-known air velocity of the wave, measures the time taken by the wave to travel back post collision with the encountered obstacle, and then calculates the distance between the transmitter and the obstacle. Thus, the principle of ultrasonic distance measurement is similar to that of radar and sonar systems. Distance Measurement formula can be written as: L = C X T. Where, L is the measured distance, C is the ultrasonic wave velocity in air and T represents time (T is half of the time taken from transmission to reception of the wave).

This ultrasonic module measures accurate distances in a range of 0 – 400 cm with a gross error of about 3 cm. Its compact size and higher range makes it a handy sensor for collision distance measurement. The module can easily be interfaced with different micro controller based boards where the triggering and measurement operations can be executed using two different pin. The sensor transmits an ultrasonic wave and produces an O/P pulse which corresponds to the time required for the burst echo to be received at the sensor. The distance between the sensor and the target can be easily calculated by measuring the echo pulse width,

Distance = (high level time × velocity of sound) / 2

The specifications of the sensor are as follows: -

Sl.	**Parameter**	**Value**
(a)	Working Voltage	5 Volt DC
(b)	Working Current	15 Millie Ampere
(c)	Working Frequency	40 Hertz
(d)	Max Range	4 meter
(e)	Min Range	2 centimeter
(f)	Measuring Angle	15 degree
(g)	Trigger Input Signal	10uS (TTL pulse)
(h)	Echo Output Signal	I/P TTL lever signal and the ranges in respective proportions
(j)	Dimension	45X20X15 mm

Table 6.1 specification of HC SR-04

6.3 **Pin Configuration – HC-SR04.**

The pin configuration will be as follows: -

Sl.	**Pin no**	**Function**
(a)	1	Vcc (to be connected to 5v)
(b)	2	Trigger pulse
(c)	3	Echo pulse (sent to Arduino)
(d)	4	Ground (connected to 0v)

Table 6.2 – pin configuration of HC-SR04

6.4 **Hardware Design.** The hardware design for acquiring collision sensor data using HC SR 04 sensor has been enumerated in Fig. 6.2.

Further photographs of the actual realization of the same undertaken by the student has been depicted in Figure 6.3 and 6.4.

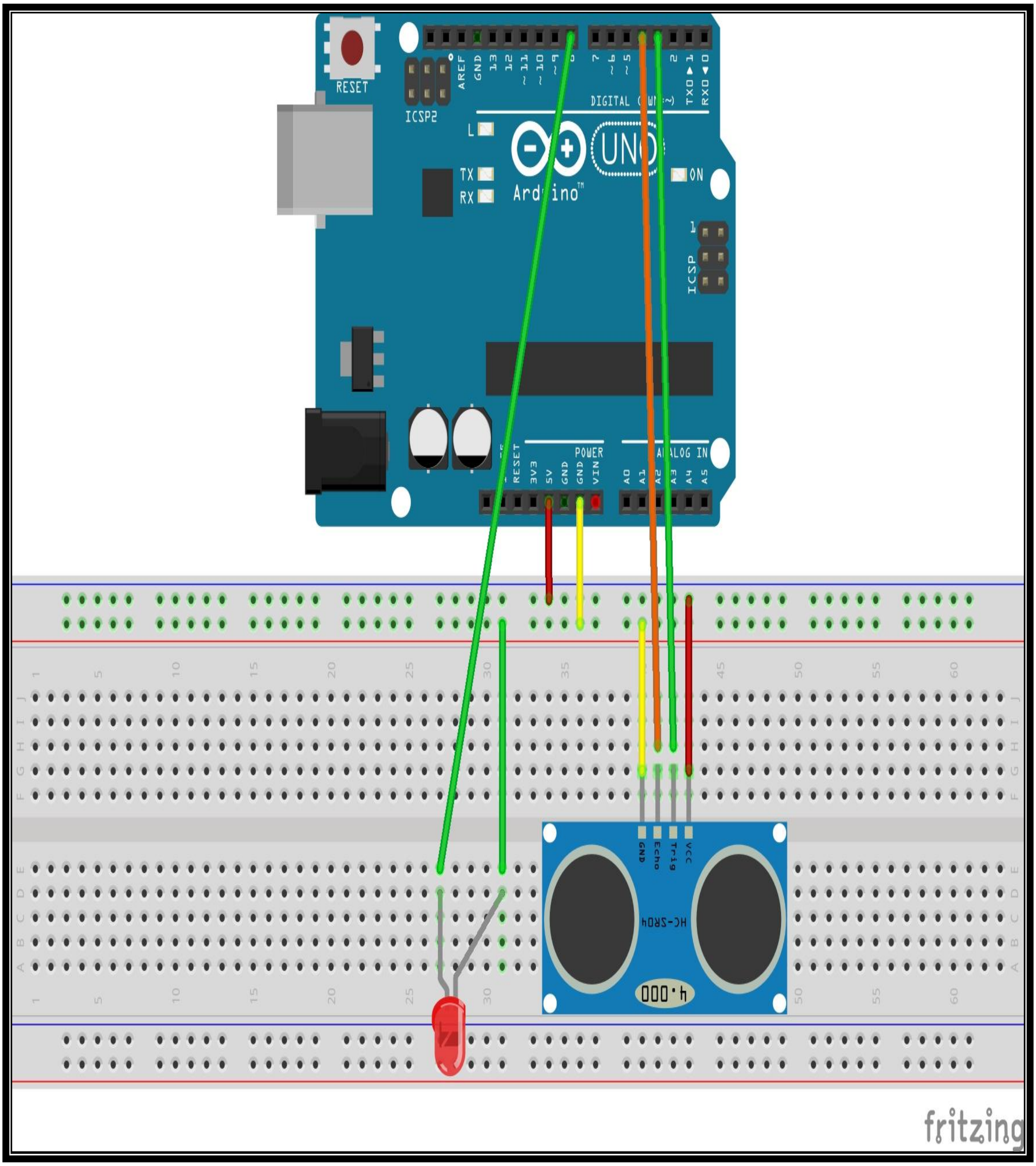


Figure 6.2 – Schematic representation for Collision detection data

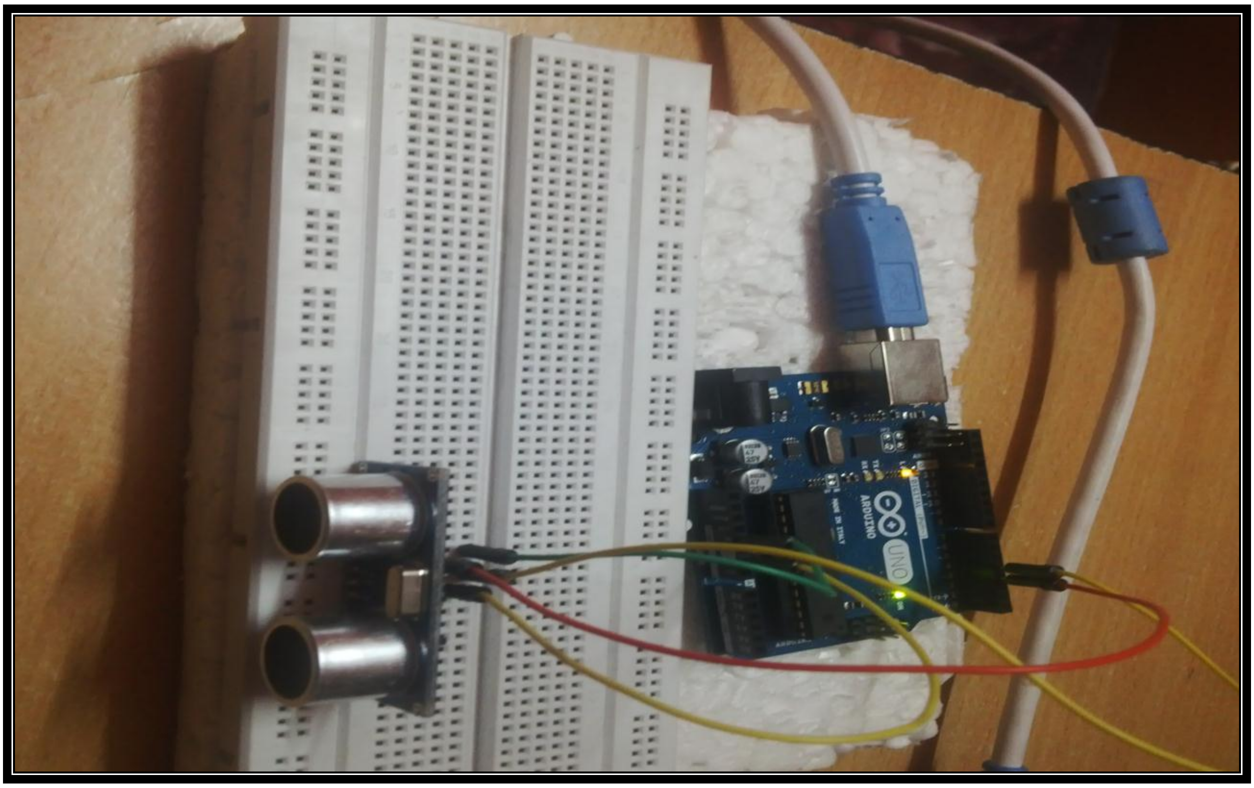


Fig 6.3 - Hardware Implementation for the ultrasonic sensor

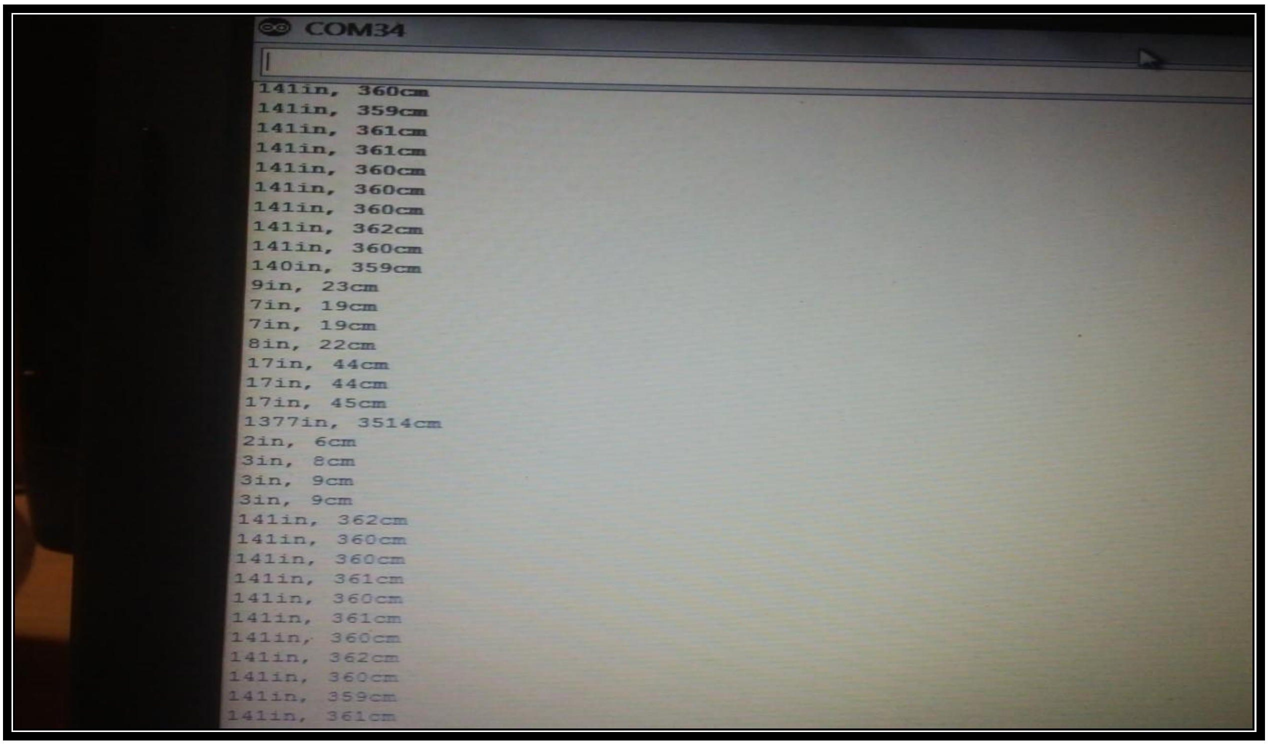


Fig 6.4 ultrasonic sensor HC – SR 04 data Read on the Serial port

6.5 **Software**. The code for the above mentioned design has been implemented using C language in Arduino IDE. Further the detailed code for the implementation has been provided separately on a compact disk

CHAPTER 7
ACQUIRING THE ROLL/PITCH/YAW DATA OF THE SHIP AND STABLISING THE RADAR ANTENNA

7.1 **Introduction**.

All IN platforms are installed with a minimum of 2-3 radars. These radars are primarily used for the following purpose: -

(a) Navigation and direction
(b) Search and surveillance
(c) Tracking of targets.

The search radars mentioned above are used to look out for hostile surface and air borne targets. However on occasions when the sea is rough, it has always been observed that targets painted on screen of the radar systems are erroneous (the bearing readings). This problem has been attributed to the two dimensional movement of the ship i.e roll and pitch. Further these movements also get passed on to the antennae of ship and cause the above mentioned erroneous readings.

In order to tackle the above mentioned problem, stabilization of antenna against the roll movement has been proposed as the solution. The antenna will be stabilized using a servo mechanism which will rotate the antenna in direction counter to the roll movement.

The above mentioned application has been demonstrated here using a three axis motion sensor IC called GY 521 and a servo SG 90. The GY 521 sensor is used to read the three dimensional movement of the ship and then roll data from the same is used to control the servo motor, on which the antenna will be mounted.

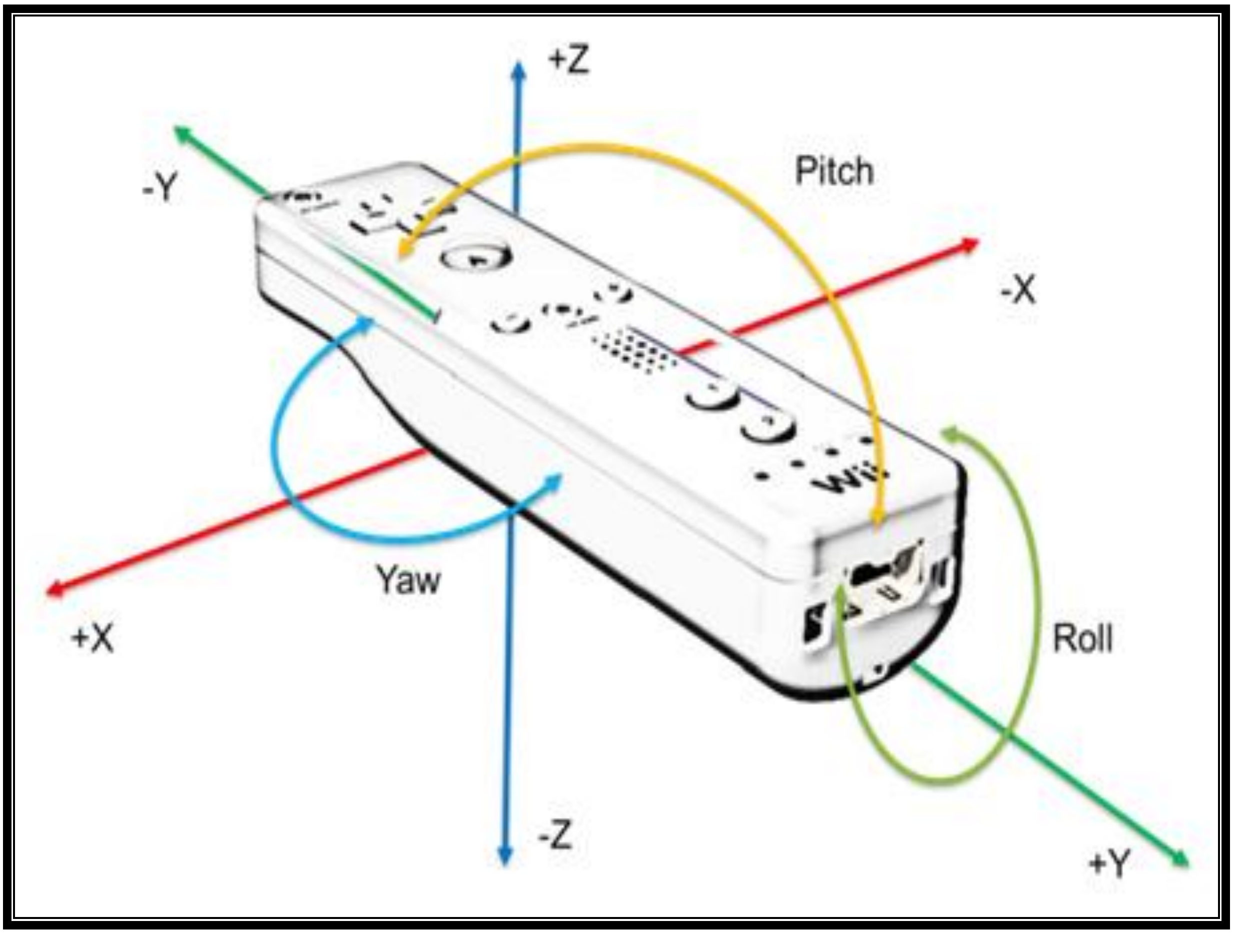


Fig 7.1 – Three dimensional motion of the ship

7.2 **About GY 521.**

This chip is fitted with a 3-axis gyroscope and a 3-axis accelerometer. This makes it a "**Six degrees of freedom inertial measurement unit (IMU)**" or 6DOF IMU, for short. Other features of the sensor includes a in-built 16-bit ADC conversion on each channel and a single Digital Motion Processor (DMP).

Function of the DMP is to combines the raw sensor data and perform some complex calculations onboard to minimize the errors of each sensor. when used on their own, Accelerometers and gyros have different inherent limitations and errors. However when the data from the two types of sensors are combined and used along with a process called sensor fusion, we can apparently get a much accurate and robust estimation of the heading. The DMP on the MPU6050 reduces the above mentioned error and returns the result in "quaternions". This quaternions data can then be converted to roll/pitch/yaw, or to Euler angles format for us to read and understand.

The main function of the DMP is to eliminate the need to perform complex and resource intensive calculations on the Arduino board. The main flipside is that it not much information has been provided by the manufacturer on the inner workings of the DMP view the same being proprietary in nature. Nevertheless, smart and creative folks have figured out how to use its main features, and the same have been shared with the rest of us.

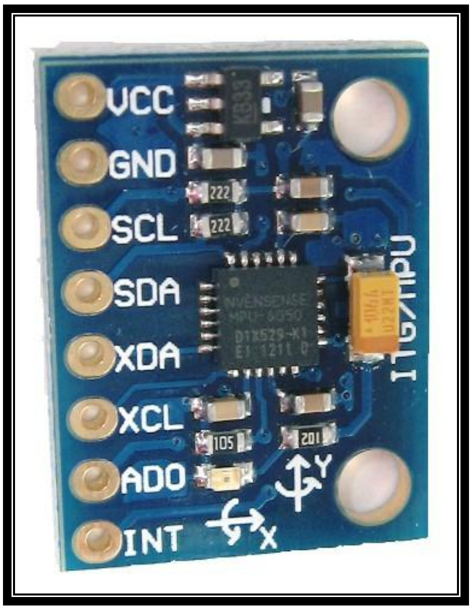


Fig 7.2 – GY 521 sensor

Communication between the MPU-6050 and the microcontroller can be undertaken on the I2C protocol interface. The sensor also has an built in additional I2C controller, which allows it perform the act of a master on a second I2C bus. The intention of providing this additional feature is to facilitate the IMU to read data from say, an external magnetometer (hooked up using the XDA / XCL pins, seen on the board) and send the same to DMP for processing.

Finally, the MPU-6050 has a in-built FIFO buffer and a built-in interrupt signal. It can be directed to place the sensor data in the buffer and the interrupt pin will then instruct the Arduino, when the data is ready to be read/accessed.

7.3 **Pin connection details**

The pin connection details are as follows: -

Sl.	**Pin details**	**Function**
(a)	Vcc	to be connected to 5v
(b)	GND	connected to 0v
(c)	SCL	Connected to pin A4 and % pf the Arduino to facilitate I2C communication
(d)	SDA	
(e)	XDA	Not connected
(f)	XCL	
(g)	ADO	
(h)	INT	Connected to digital pin 1

Table 7.1 – pin configuration of GY 521

7.4 **About SG 90 servo**.

Servos are small rectangular electro-mechanical devices widely used in Model cars and airplanes. Following are the sub parts: -

(a) a DC motor

(b) electronics to control the motor from a signal,

(c) a gear system to produce slow/strong output to a shaft

(d) a position feedback potentiometer.

Servos usually come with several different arms, which can be used as per requirement of the design. Usually a Servo is used to position things like steering direction, arm/leg movement etc. The position angle of the servo is controlled by the length of the pulse in milliseconds (about 1.00-2.00ms duration). 1.50ms is usually taken as Center.

All servos have a three wire connector. One wire supplies positive 5 to 6 DC volts. The second wire is for providing the ground (0v), and the third wire is used to provide the signal. The receiver can talk to the servo through this particular wire by using a simple on/off pulsed signal.

Pin Connections*:* (wire colors may vary!)

(a) Servo Black or Brown – connected to Gnd.
(b) Servo Red or Orange (Center wire) – connected to +5V
(c) Servo White or Yellow - connected to Signal (Pin 9 in present case)

The analog RC servo can control speed of any motor by applying on/ off voltages or pulses. This voltage is to be kept constant

This PWM (pulse Width Modulation) technique is happens within the little servo itself and is not related to the PWM capability of the Arduino. 50 cycles per second is the standard internal PWM frequency of the servo.

The longer the duration of the ON pulse is, the faster the motor will turn and similarly more torque will be produced. This is similar to how the speed of most of the motors is controlled. For instance, if we have a fan (ceiling or exhaust) in our house which is controlled using a variable rotary dial speed switch, the fan motor is not given lower/ higher voltage supply to control the speed. However the speed switch cycles the 120 v to the fan motor on/ off many times a second. The longer the duration of each ON pulse is, the faster the fan will rotate. This is similar to how the electronic speed controllers for electric RC helicopters, planes, cars, and boats function.

Fig 7.3 – SG 90 servo

Explanation. When the motor is at rest, there is no voltage going to the motor. However if a small transmitter command is given/ some external pressure is applied to the servo horn forcing it off its neutral position, a short duration voltage pulse will be generated and sent to the motor. The larger the stick/ potentiometer movement, the longer will be the "ON" pulse duration in order to move the servo quickly to the required/ desired position. These voltage pulses are sent 50 TPS (times per second). This means that in one second, there are 50 windows that will last for 20 ms each i.e 50x20 = 1000 ms = 1 second. The longer each ON voltage pulse duration is in each of these fifty 20 ms windows, the faster the servo motor will turn and the larger more torque will be generated.

When small amounts of stick movement/ external forces are applied moving the servo off its neutral position, only a short duration voltage pulse will be generated and sent to the servo motor every 20 ms. With large stick movements, a long voltage pulse will be sent every 20 milliseconds to the servo motor.

7.5 **Hardware Design.** The hardware design for acquiring Roll/Pitch/Yaw data and controlling the servo using GY 521 sensor has been enumerated in Fig. 7.4. Further photographs of the actual realization of the same undertaken by the student has been depicted in Figure 7.5 and 7.6.

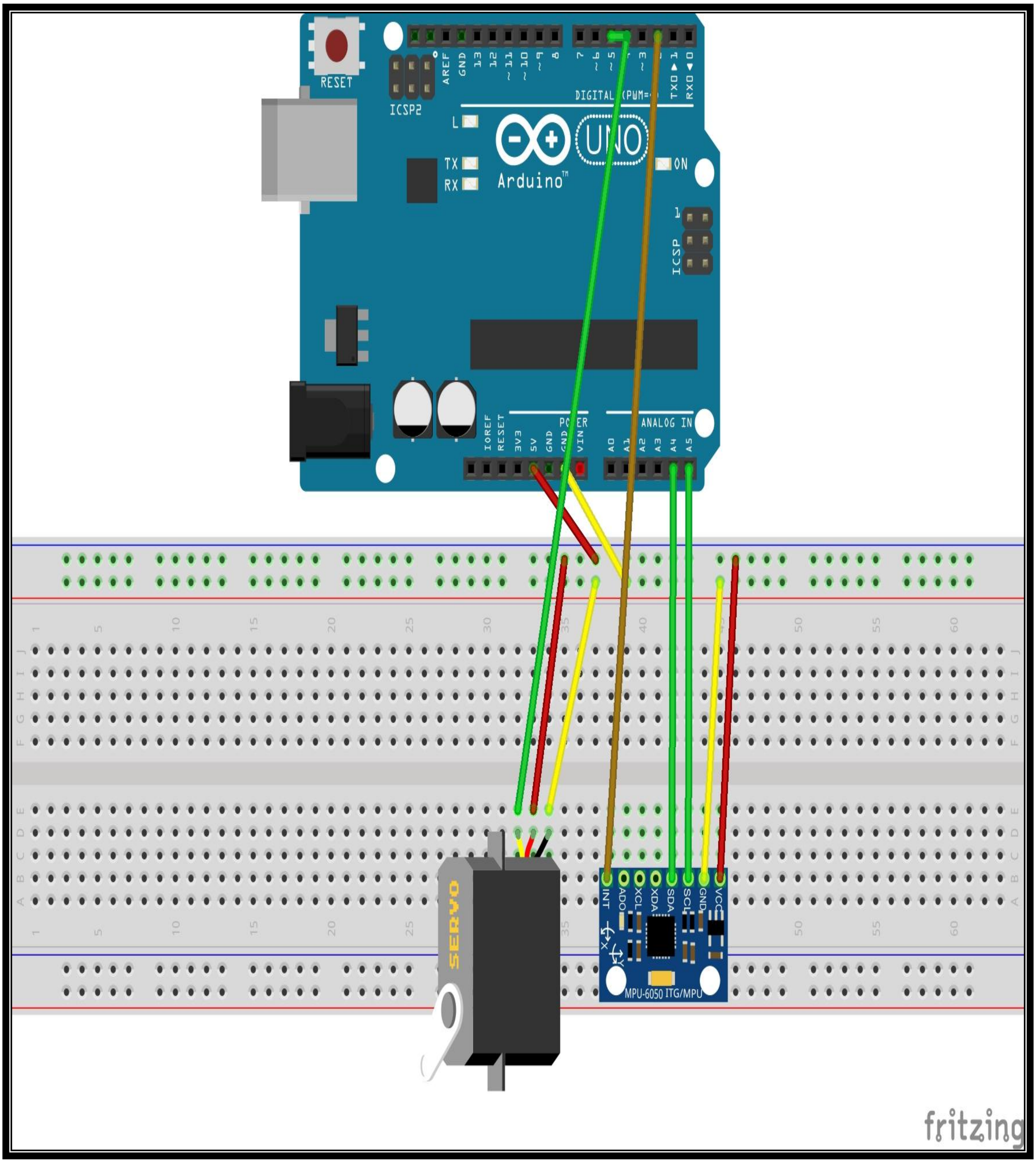


Fig 7.4 - Schematic representation for Radar stabilization

7.6 **Software**. The code for the above mentioned design has been implemented using C language in Arduino IDE. Further the detailed code for the implementation has been provided separately on a compact disk

CHAPTER 8
FINAL DESIGN OF THE DATA ACQUISITION CONSOLE
(COMBIED ACQUISITIN OF DATA FROM ALL SENSORS)

8.1 **Introduction.**

As we can see that in preceding chapters, the acquisition of temperature, humidity, distance and gas data in solo mode has been completed. Now there is a requirement to read the data from the sensors in a single sting. Further, since the data received from the sensors is required to be monitored on real time basis. Hence there is also a requirement to design a GUI (Graphic user interface) to facilitate the real time monitoring of magazine parameters by operators onboard ships.

The issues mentioned vide para above have been implemented in this chapter and the details of the same are enumerated in succeeding paragraphs.

8.2 **Hardware Design of DAC**.

The detailed schematic design for acquisition of data from all the four sensors concurrently has been implemented as shown in Fig 8.1. Further the actual implementation of the same has been shown in figure 8.2. The detailed connections are as follows: -

Sl.	**From**		**To**		**Remarks**
(a)	**Item**	**Pin**	**Item**	**Pin**	
(b)	Sensor HC-SR04	Vcc	Breadboard	5V	Provided from Arduino
(d)		GND		0V	
(e)		TRIG	Arduino	Pin 3	Arduino board sends a trigger to the sensor
(f)		ECHO		Pin 5	Echo received from the

					sensor send to Arduino
(g)	Sensor MQ 2	Vcc	Breadboard	5V	Provided from Arduino
(h)		GND		0V	
(j)		D out	---	---	Not connected
(k)		A out	Arduino	Pin A0	Sends reading of sensor to Arduino in analog format
(l)	Sensor DHT 11	1 (Vcc)	Breadboard	5V	Provided from Arduino
(m)		2	Arduino	Pin 7	Sends temperature and humidity readings to the Arduino in digital format
(n)		3	---	---	Not connected
(o)		4 (Gnd)	Breadboard	0V	Provided from Arduino
(p)	Sensor GY 521	Vcc	Breadboard	5V	
(q)		Gnd		0V	
(r)		SCL		Pin A4	Communication between the Arduino and the sensor in I2C format
(s)		SDA		Pin A5	
(t)		XDA	---	---	Not connected
(u)		XCL	---	---	
(v)		ADO	---	---	
(u)		INT		Pin 2	
(v)	SERVO SG 90	VCC	Breadboard	5V	Provided from Arduino
(w)		GND		0V	
(x)		SIGNAL	Arduino	Pin 5	Arduino sends signal to the servo to control its rotation

Table 8.1- Connection Pin Details

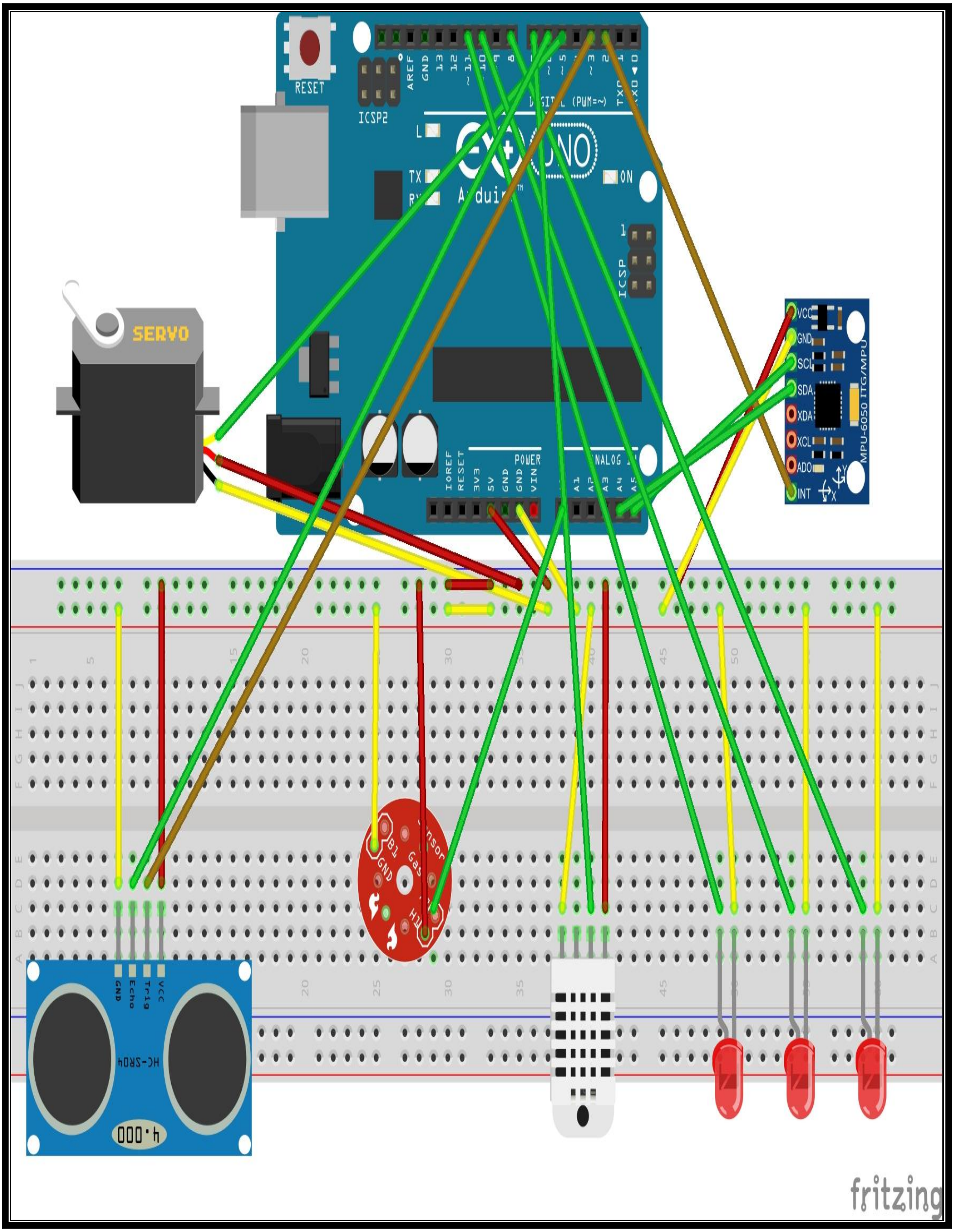


Fig 8.1 - Schematic representation of the DAC

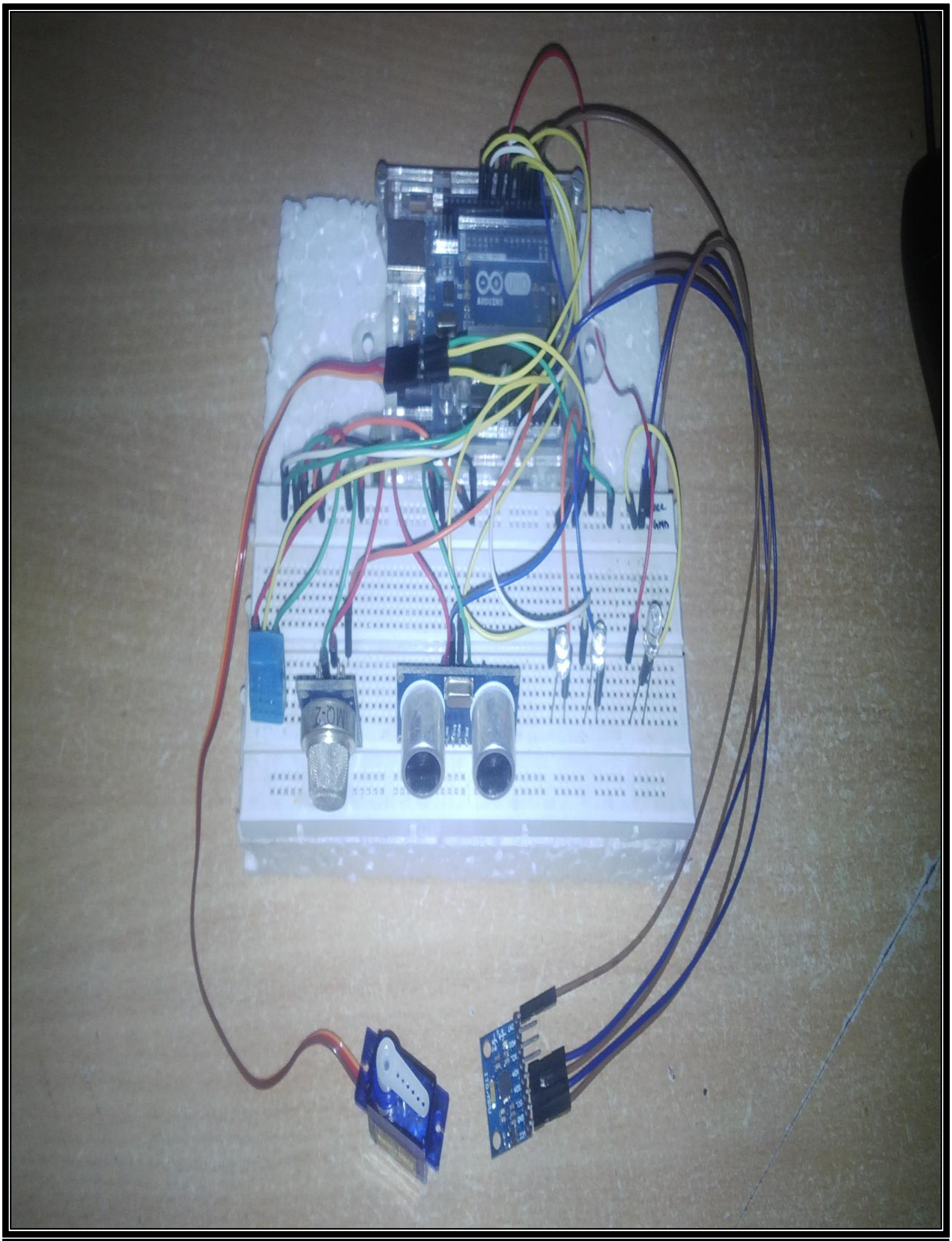


Fig 8.2 - Hardware Implementation of the DAC

8.3 **<u>Reading data on Serial data Window</u>**. The data read by the above design can be read on the serial port of the Arduino IDE. Further details of the same are as follows (Figure 8.3 relevant): -

Sl.	**Data Displayed**	**Sensor reading the value**	**Alarm Limit set**
(i)	Humidity	DHT 11	Not to exceed **45%**
(ii)	Temperature (Celsius)		Not to exceed **60 degrees**
(iii)	Temperature (Fahrenheit)		No limit set
(iv)	Temperature Index		No limit set
(v)	Distance (Inch)	HC-SR04	Not to be less than **4 Inches**
(vi)	Distance (Cm)		No limit set
(vii)	Gas sensor value	MQ 2	Not to exceed **380**

<u>Table 8.2 - Alarm limits for the DAC</u>

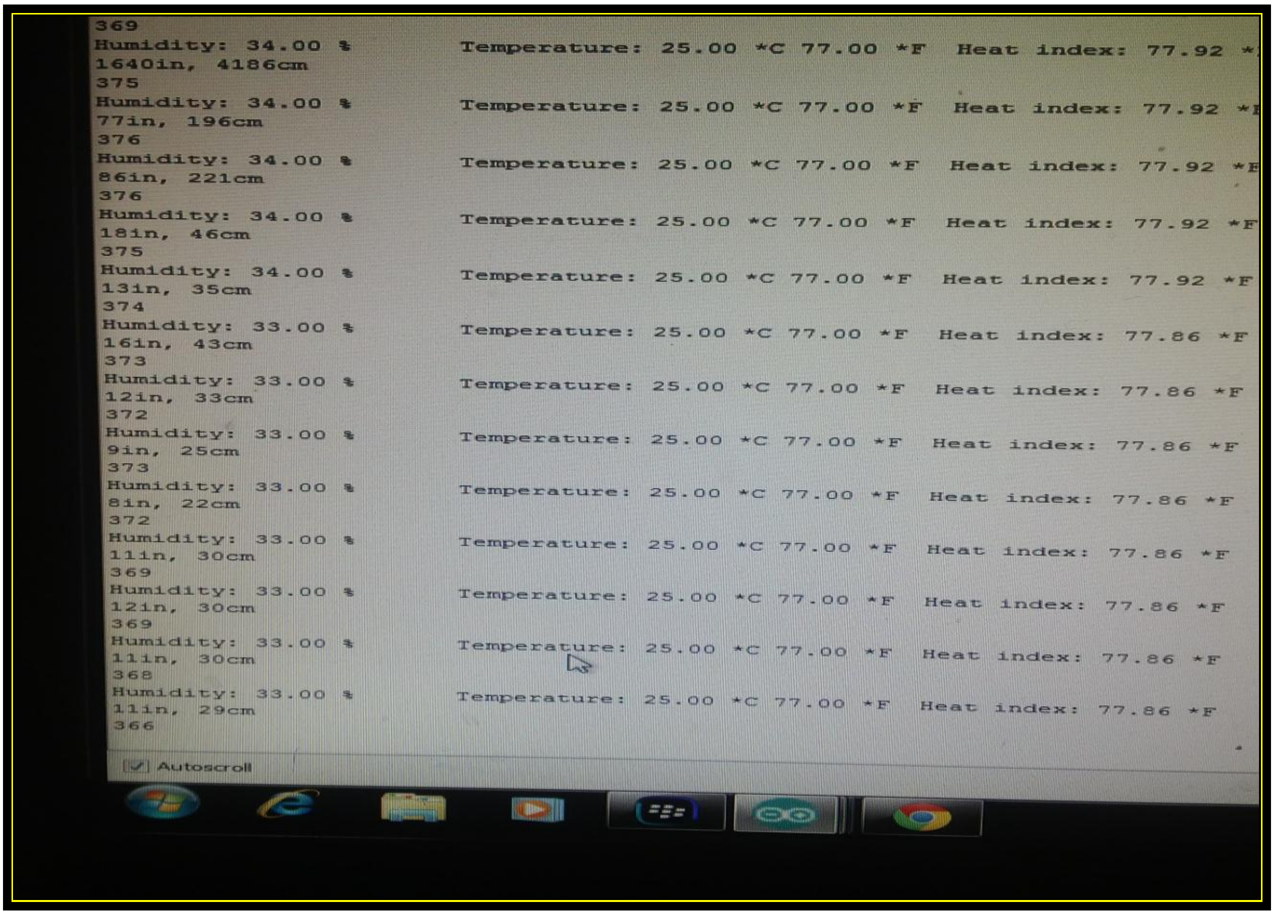


Fig. 8.3 – sensor data displayed on the serial window3

8.4 Data display on graph. In order to monitor the data on real time a user specific GUI has been designed using freeware processing (in C language). Where in the following data will be displayed in a graphical form (Figure 8.4 relevant): -

Sl.	**Data Displayed**	**Quadrant**	**Stroke Colour**	**Alarm Limit set**
(i)	Humidity	III	Blue	20%
(ii)	Temperature (Celsius)	II	Red	30 degrees
(iii)	Distance (Inches)	IV	Green	4 Inches
(iv)	Gas sensor value	I	Black	380

Table 8.3 - Details of the graph

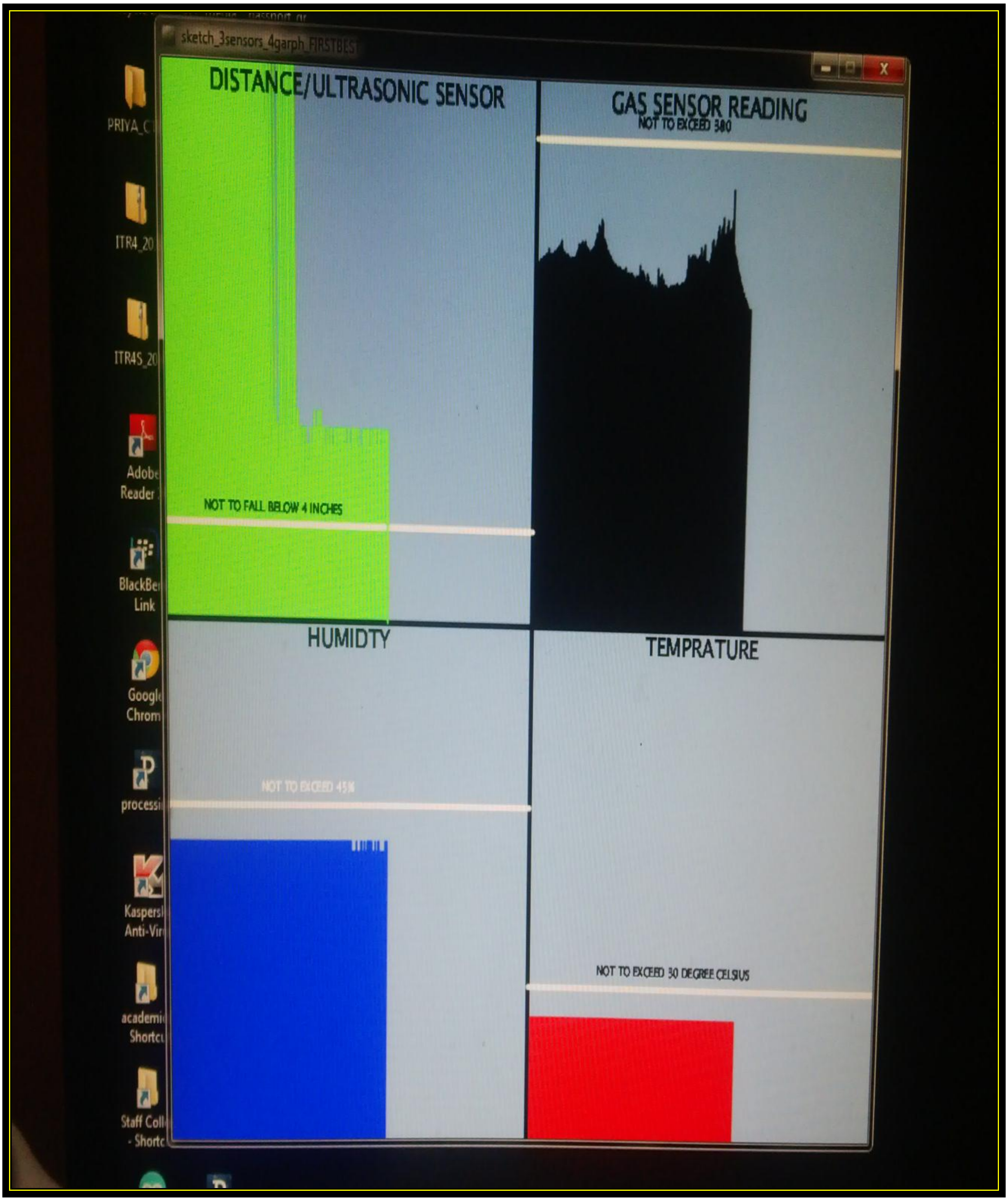


Fig 8.4 GUI for graphical display of sensor data

8.5 **<u>Coding for Data Acquisition</u>** has been implemented in freeware Arduino IDE. The detailed code for the same has been provided separately on a compact disk.

8.6 Coding for Data display in graphical form.

The GUI for the display of the sensor data has been implemented on freeware called processing. The language used is C. The detailed code for the same has been provided separately on a compact disk.

CHAPTER 9
CONCLUSION AND FUTURE SCOPE

9.1 The following activities can be undertaken in the future: -

(a) **Data acquisition from COTS GPS**. it is pertinent to note that almost every system fitted onboard a ship requires the feed of GPS to calculate data like range of targets. There fore a provision can be made in data acquisition console to read data from a serial out GPS and feed the same to ship fitted systems.

(b) **Interfacing the DAC data on ship fitted LAN**. Every naval platform is installed with a military grade LAN which is also interfaced with the ship fitted systems. Therefore an endeavor will be made to convert the data generated at the DAC into Ethernet format and dump the same on the Ships LAN so that the same can be made available to all ship fitted systems.

REFERENCES.

(a) Taghi Mohamadi, "**Designing an Embedded System for Interfacing with Networks Based on ARM**", Design & Test Symposium (EWDTS), 2011 9th East-West Digital Object Identifier

(b) Li Wei ,Cheng LiangLiang,Yu Min,Chen BingBing, "**Design of Temperature Acquisition system based on Ethernet**", Electrical and Control Engineering (ICECE), 2011 International Conference on Digital Object Identifier

(c) Mahboob Imran Shaik, "**Design & Implementation of ARM Based Data Acquisition System**", Electronics, Communication and Computing Technologies (ICECCT), 2011 International Conference on Digital Object Identifier

(d) ARM Limited, "**ARM Cortex M4 Generic user guide**"

(e) Arduino website, URL: http://www.arduino.cc

(f) Processing software website, URL: http://www.processing.org