PROJECT REPORT

ON

"Intelligent Helmet for Coal Miners "

SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY

FOR THE DEGREE OF

M.Sc.

(ELECTRONIC SCIENCE)

BY

"Mr. Shelar Bhagirath Vishnu"

(211457)

DEPARTMENT OF ELECTRONIC SCIENCE

FERGUSSON COLLEGE (Autonomous)

PUNE - 411004

May – 2022

Deccan Education Society’s

Fergusson College (Autonomous), Pune-411004.

Department of Electronic Science

CERTIFICATE

This is to certify that Mr. Shelar Bhagirath Vishnu. student of M.Sc. (Electronic Science) has satisfactorily completed the project work towards the partial fulfillment of his **M.Sc. (Electronic Science) Semester-IV** during the academic year 2021-2022.

INTELLIGENT HELMET FOR COAL MINERS

He has carried out his work under the guidance of ***Prof. P.B.Kamble*** at Fergusson College, Pune.

Internal Guide External Guide

Mrs. Panchsheela Kambale

(Dr. N. M. Kulkarni)

Head,

Dept. of Electronic Science

Examiners:

Internal Examiner External

ACKNOWLEDGMENT

I wish to thank the Principal ***Dr.R.G.Pardeshi*** of my college for permitting me to use all the facilities available in the institution for my project work. I would also like to thank the Head of the Electronics Science department ***Dr.M.S.Zambare*** and all the teaching faculty and non-teaching staff of my department for their support in completion of work successfully.

I am grateful to my Guide ***Ms.Panchsheela Kamble*** for her encouragement, guidance and supervision of my project work during the year. My classmates have been of great help to me during the project work. My ideas were shaped and refined progressively through my discussions with them from time to time. I cannot miss to thank them.

It is obligatory duty to express my whole thanks to ***Ms. Yogita Khandge***, ***Ms. Ranjana Ubale*** and ***Dr. Kalpana Kulkarni*** of their constant encouragement and useful suggestion in improve the quality work.

Last but not least I would like to thanks all friends or their enthusiasm and great ideas. I am also thankful to teaching & non teaching staff for their help which I got.

Thanking you !

Mr. Shelar Bhagirath Vishnu

Date: / /2022

Index

Chapter	Details	Page no.
Chapter 1.	**Introduction**	1
1.1	Importance of Intelligent Helmet for Coal Miners	2
1.2	Problem Definition (purpose)	3
1.3	Aim and Objective	4
Chapter 2.	**Fundamentals of ------**	5
2.1	Literature Survey	5
2.2	Basic Theory	6
Chapter 3.	**Project Proposal**	7
3.1	System Specifications	8
3.2	System Block Digram	9
3.3	Method of Implementation	10
3.4	Task Vs Time Schedule	15
Chapter 4.	**Planning Resources**	16
4.1	Hardware	17
4.2	Software	76
4.3	BOM	98
Chapter 5.	**Experimentation and Results**	99
5.1	Experimentation	99
5.2	Results	104
Chapter 6.	**Summary and Future Scope**	108
Chapter 7.	**Bibliography**	
	References	106
	Weblinks	107
	List of Figures	108
	List of Tables	111
	List of Codes	112

Chapter 1

Introduction

In mining industry worker safety is very important issue. Every year, thousands of miners die in accidents and many more get injured, especially in the processes of coal mining and hard rock mining. The main reasons of accidents are gas or dust explosions, gas intoxications, improper use of explosives, electrical burn, fires, collapsing of mine structures, rock falls from roofs and side walls, flooding, workers stumbling/slipping/falling, or errors from malfunctioning or improperly used mining equipment. In coal mine use of personal protective equipment like helmet, shoes etc. are not proper and proper arrangements were not there to check if the person is wearing personal protective equipment or not. The proper supervision for worker wear the protective equipment is very important factor for consideration. Underground mines are very dark so any miners are fall unconscious because of suffocation or falling of structure, supervisor don‟t known about her health condition and proper treatment is not provided her in time. The main reason for miner death is harmful gases explosions. In coal mines carbon monoxide, methane, LPG gases are existing and they are very harmful for human body.

The persons who are working in the coal mining has to face various environmental parameters in their mining. They have the danger from the methane, carbon monoxide, and temperature. So we need to provide a strong security for the people who are working in the coal mining. The purpose of this project is to provide a solution to mining by wireless communication and safety monitoring. The person must use the helmet while working in the coal mining. Here we have to arrange our total circuit within the kit to provide safety to the person who is working in coal mining. In recent days coal mining has been very dangerous activity that can result in a number of adverse effects on the environment for example during mining operations methane, a known greenhouse gas, may be released into the air. Underground mining hazards include suffocation, gas poisoning, roof collapse and gas explosions. Keeping all these aspects in mind we designed a system, i.e. smart helmet using zigbee technology for monitoring the hazardous gases, abnormal temperature conditions and the humidity levels in the air.

The improved safety features in our system dramatically increased life expectancy of the coal miners by alerting them about the hazards. In our system, the helmet is having the circuit with three sensors i.e. temperature, humidity and gas to monitor the conditions in coal mine. If there is any hazardous situation in the mine the helmet gives the information to the control station through the zigbee transmitter and the control station will alert the coal miner using the zigbee receiver by making the buzzer active which is positioned in the helmet so that a miner can have a chance to rescue his life from the hazards occurred in coal mines.

1.1 : Importance of Smart Helmet for Coal Miners

This project focuses on a mine supervising system which is based on the cost effective IOT (ZigBee) system. Our project aims at developing a sensor networks, realized real-time surveillance with early-warning intelligence on harmful gases, heartbeats, location, temperature, humidity in mining area and used ZigBee communication to reduce potential safety problems in coal production using a ZigBee technology. All these three parameters are detected continuously by temperature sensor, gas sensor, humidity sensor and if they cross the pre-defined limit, then the user gets information about all three sensors and it displays on thing speaks site as the graphs and it will automatically updates the values to this site. With a Zigbee positioning devices the system might be easily extended. The values of different sensors are continuously transmitted by ZigBee transmitter to the remote monitoring unit which are received by Thingspeak site through IP address.

In the importance of GSM and GPS is discussed. The GSM (Global System for Mobile) used to send the position of the miner as a data messege the Thingspeak cloud. And also the position of the miner can be obtained. With this equipment, we can keep track of the miner by periodically sending sensors data and the position of the miner is sent by the GSM modem as a messege to the thingspeak. The GPS modem will continuously give the data i.e. the latitude and longitude indicating the position of the the miner. This aims on the supervised system which is based on cost effective Zigbee technology. It is used for developing a wireless sensor networks, realized real time surveillance with early-warning intelligence on harmful gases, temperature, heartbeats, obstacles, humidity in mining area and used Zigbee communication to reduce potential safety problems in coal production using a Zigee wireless technology. With a zigbee Wireless positioning devices the system might be easily extended. The Zigbee transmitter acts as a monitoring unit where random values are received by Zigbee receiver module. A major portion of the fatalities occur because the person was either not wearing a helmet, or his accident was not reported in time, and he could not be saved because of the delayed admittance to a hospital. We propose mechanisms that can detect if one is wearing the helmet, detect helmet. The values of different sensors are continuously transmitted by RF transmitter to the remote monitoring unit which is received by RF receiver module. It gives the detailed description about the smart helmet and the hardware requirements for the development. This invention relates to safety helmet device, especially a coal miners with a safety helmet. This helmet is a type of protective headgear used by miners. More particularly, the present invention relates to a safety helmet device which shall check the presence or absence of helmet on miner's head and miner has not dunked by sensors.

1.2 : Problem Definition

The major problem identified is that there are numerous accidents occuring in the coal mines due to improper maintanance and inadequate monitoring of the mining activities. These led to numerous life losses and immeasurable resourse loss. There is no proper early detection of the uncertainity in the coal mines. Coal mining has been a very dangerous activity. The principal hazards are mine wall failures and vehicle collisions; underground mining hazards include suffocation, gas poisoning, roof collapse and gas explosions. Chronic lung diseases, such as pneumoconiosis (black lung) were once common in miners, leading to reduced life expectancy. In some mining countries black lung is still common. However, in lesser developed countries and some developing countries, many miners continue to die annually, either through direct accidents in coal mines or through adverse health consequences from working under poor conditions.

The proper supervision and proper communication is very important requirement of mining industry. The smart helmet provides a real time monitoring of harmful gases, person fall detection and miner wear the helmet or not. The harmful gases like carbon monoxide, LPG, Methane and also temperature are monitor using this system.

Firstly to identify the worker, each worker will be having different tag. Once the tag is identified, person's data will be sent to the PC through ZIGBEE. In order to check whether the worker has been using the helmet or not, IR sensors are used to check the helmet presence. The surrounding hazardous gases will be detected by the gas sensor present in the helmet. When gas is detected voice notification will be given through speaker. By the use of MEMS sensor, the head injuries occurrence will be identified.

1.3 : Aim and Objective

Aim

The present work is an attempt to analyze the safety scenario of a mine. The idea is an improvisation on previous related works where WSN is used to detect the toxic gases present in the mine. The real time monitoring which is the requirement now a days is designed for the purpose. The system presents all the related technologies such as ZigBee and embedded designing etc.

This project focuses on a mine supervising system which is based on the cost effective IOT system. Our project aims at developing a sensor networks, realized real-time surveillance with early-warning intelligence on harmful gases and be easily available for the help of miner in panic situation. parameters are detected continuously gas sensor and if they cross the predefined limit, then the user gets information about all the sensors and it displays in control room and it will automatically updates the values.

Objective

Mining environment often has hidden dangers within such as toxic gases, which may present severe health exposures to the people working within mining. These gases need to be detected at times and informed the dangerous situation in right time for the safety of miners. Wired network monitoring systems have assisted the mine safety significantly, but it is not idea for all types of mining environment. A real-time monitoring systems may assist in monitoring and control over the mining environment. Zigbee technology offers its most of the advantages ideal for the real-time monitoring system. Thus, the primary objective of this project is decided to design an efficient real-time monitoring system so that various leaked mine gases could be identified at times and preventive measures could be devised accordingly. The research investigations to be carried out with the following objectives:

1. Detection of different environmental parameters in their mining.
2. Communication establishment between sensors and Zigbee.
3. Establishment of Wireless Sensor Network .
4. Design of a real-time monitoring system.

Chapter 2

Fundamentals of Project

2.1 : Literature Survey

1st..Zigbee technology is implemented for providing safety for the people working in mining industry and also change their way of working system which controls the numerous changes in the mining area environment.

2nd. For detecting three types of hazardous such as gases, helmet removal and collision or impact.

3rd. Gas, MEMS and IR sensors are capable of monitoring and reporting the conditions prevailing in the mining area. Is there any hazardous situation, then it detects using the smart helmet and pass the message to control room which was monitored frequently via zigbee transmitter, once message received then control station will immediately alert the people working in that area by zigbee receiver, which rings the alarm.

4th. Helmet detect hazardous event , monitor environmental conditions and information will be updating such as GPS location and sensor data to the central for tracking purpose.

5th. The parameters will be displaying on the current stat on personal computer and if any harmful gases like CO2 exceed their limit then the corresponding person fall detection & alert to remove the helmet for the mining industry workers.

6th. The real time data where the sensors sense the environmental conditions in the mining area is displayed on the Light Emitting Diode, which is also to be updated on the respective website server with the aid of Thing Speak using IoT.

7th. It provides reliable communication between mining workers using zigbee module.

8th To recognize unsafe situations in the mining area such as carbon monoxide gas accumulation, removal of helmet and detection of crash.

9th. To provide flexible and complete solution for building private LoRa network, design and implementation using hardware and software were done.

10th. The aim of this study about the most recent research papers analysis using LoRa technology.

11th.The literature survey includes the propagation models performance, energy efficiency, MAC layer and channel access challenges of LoRaWAN.

12th LoRaWAN is compared with other wireless communication devices such as Bluetooth, Wi-fi, zigbee and it has huge scope for implementation in smart city applications.

13th The energy consumption of LoRaWAN end device transmitting data have been modelled.

14th The required power of LoRaWAN in class A and C type devices were investigated which have the high efficiency.

15th-16th LoRaWAN is an open standard development to prevent consumption and moderate the network effectively. This new technology is able to concentrate on network management, optimization of high dense,etc.

2.2 Basic Theory

The aim of the project is to design a wireless helmet using Zigbee era. The machine is a value powerful ZigBee-based wireless mine supervising machine. Software adopted ZigBee technology to build wireless sensor networks, determined out real-time surveillance with early-warning intelligence on temperature, leakage of gasoline in mining place, and alerting the manage station the use of wireless zigbee technology. The device is used to reduce capacity protection troubles in coal manufacturing. Zigbee is a WPAN era primarily based on the IEEE 802.15.4 popular. Unlike Bluetooth or wireless USB gadgets, ZigBee devices have the capability to shape a mesh community among nodes. This technique allows the quick range of an person node to be accelerated and improved, covering a far larger region. This machine has transmitter phase interfaced to arduino to the helmet. The gadget has sensors the usage of Zigbee generation. The machine also monitors the gas leakages the usage of gas detection sensor, fireplace the use of temperature sensor, in the mines and if it exceeds the threshold degree, it alerts via alarm the usage of buzzer and also displays on pc to the person. This sensors output is given to the Arduino for similarly processing to ship the alerting message to the station the use of Zigbee module. The Arduino is programmed the usage of C language.

The Arduino microcontroller is used to detect and monitor variables in a coal mine. Live readings are provided by the temperature & humidity sensor, IR sensor, and gas sensor, Pulse sensor, and GPS. A Arduino and a transceiver are connected to all of these sensors. The data is sent to the microcontroller, and communication between the gateway and the specific node is done via Xbee WPAN IEEE 802.15.4. As mentioned previously, the data is sent to the control room via the XBee protocol. In an abnormal situation, an alert indication is sent to the system, which is also displayed on an LCD screen connected to Arduino at the coalfield's entrance. A buzzer is also programmed and controlled with the help of Arduino, which activates at any abnormal reading detected by the above sensors.

Chapter 3

Project Proposal

The proposed system consists of the sensor modules that senses all the data around the coal mine environment and logs the data onto the cloud controlled server page using IOT module. The logged data is processed into the average values for each entry on an interval basis. These values are automatically processed using a predefined values maintained by the server page. Ehen there is any arbitrary change in the values of the sensed data an alert is send by the zigbee to the base station and again base station to the IOT MODULE and the concerned authorities. The IOT module detects the alert signal and glows the inbuilt alarm system and alert message to the authorities may take precaution steps. The main advantage of this project is that Iot detects the uncertainty in the environment in beforehand using data analysis reports the situation to the concerned authority and the miners. The system also considers the emergency situations in hand to alert the miners quickly as possible. This project serves the aspect of “Prevention is better than Cure”.

The proper supervision and proper communication is very important requirement of mining industries. The smart helmet provides a real time monitoring of harmful gases, proper light intensity for work, humidity and miner is wearing the helmet or not. The harmful gases like carbon monoxide, LPG, Methane and also temperature are monitor using this system. The wired communication network is not so effective because when natural calamity or a roof fall occurred, wired network is damages, so it is very difficult and costly to reinstall the entire system. The effective solution for communication from base station to underground mine is IOT based wireless network. The smart helmet for mining industry consists of various sensors which are fixed on the helmet. The sensors used are Gas Sensor, Temperature Sensor, Humidity Sensor, Pulse sensor, GPS, and IR sensor to detect weather the miner wearing helmet or not and obstacle detection. By using IOT module the mine information like environmental parameters in mine or wearing helmet or not can be seen anywhere anytime by using internet. So the proper action can take within time to rescue the miner.

3.1 System Specifications

The system is developed in two main sections. Where the first section is helmet section used for Sensor networks connected with RF transceiver modules, Temperature and humidity, pulse rate, gps locator, IR for obstacle detection, gas sensor to sense the environmental events and vib, buzzer & LED indicator to display the danger status of the minor. The second section is the control room section which consists of Arduino NANO, GSM, RF module, IOT and GUI system. The wireless communication used is ZigBee due to its robustness in closed area. Clearly, WiFi or Bluetooth protocols will not be a sufficient solution forunder ground systems. The overall block diagram as shown in Fig.3.2.1 consists of 6 sensors(Temperature and humidity, IR, Gas sensors, pulse sensor, gps,) that will be collecting data from the surrounding environment of the helmet and will transfer each data to Arduino Nano which is connected with XBee Reciever. However, GSM also connected and they send the data to the Thingspeak server. Arduino is communicating with XBee Transmitter in UART ports where Tx is continuously sending the information to the Rx through Digi mesh topology. Thingspeak and control room computer are used to specify the status of the miner condition weather he is safe, in dangeror needs rescue team. On the output side a GSM module is used to send the message.

The MQ135 is used to monitor the level of Methane, LPG, Hydrogen, Carbon monoxide, Alcohol, and Smoke. The DHT11 is used for monitoring temperature. The DHT11 is used to monitor the humidity. Pulse sensor count the pulses. To detect weather the miner is wearing helmet or not and the obstacles the IR sensor is used. Also GPS locates the Real Time Location. These are semiconductor type sensor. The sensor senses the various environmental parameters like temperature, humidity, level of gases, pulses and transmitted real time data to the controller, the controller receive the data processing on it and transmitted to the base station using IOT module. At the same time if gases level is increases above threshold, alert signal will be send to the base station. The alerting device, buzzer is used for alerting to the supervisor or the person at the base station if any of the sensor data exceed than rated or threshold values. displays the massage according to sensor values, like if gas detected (any poisonous gas) then level increases and LCD will display “GAS detected”. If helmet is removed by miner then, “Helmet Removed” will be displayed on display. All the data is received from helmet and transferred to the base station through IOT and display on PC. By using IOT module we will able to get mine’s update anywhere anytime through internet. and send this information to the remote monitoring unit. Low rate Zigbee is used for data transmission. When the control center detects the parameters are sends remote control area which is Thing speak sites through IP address. A temperature sensor (DHT11) shows the present temperature every 2 sec. it plot the graph with respect the values in Thing speak site. Similarly remaining sensors sense respective values and post controlling area i.e. Thing speak site trough IP address. All values shows in the form graph representation.

3.2 : System Block Diagram

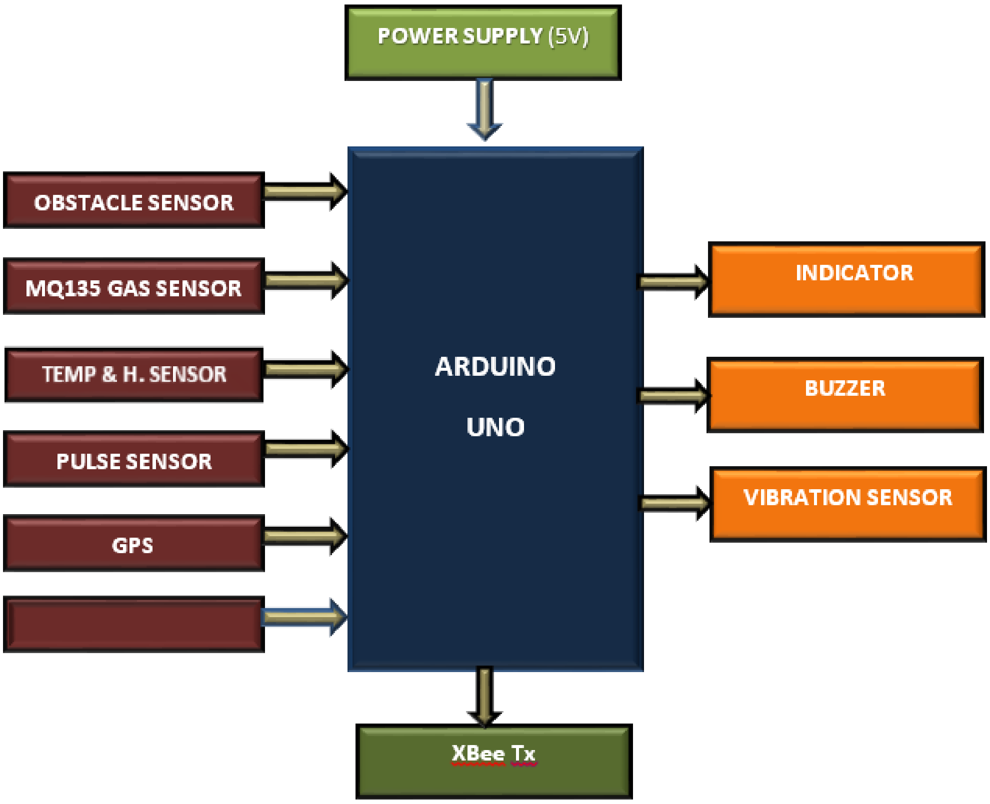


Fig. 3.2.1 : Transmitter Section

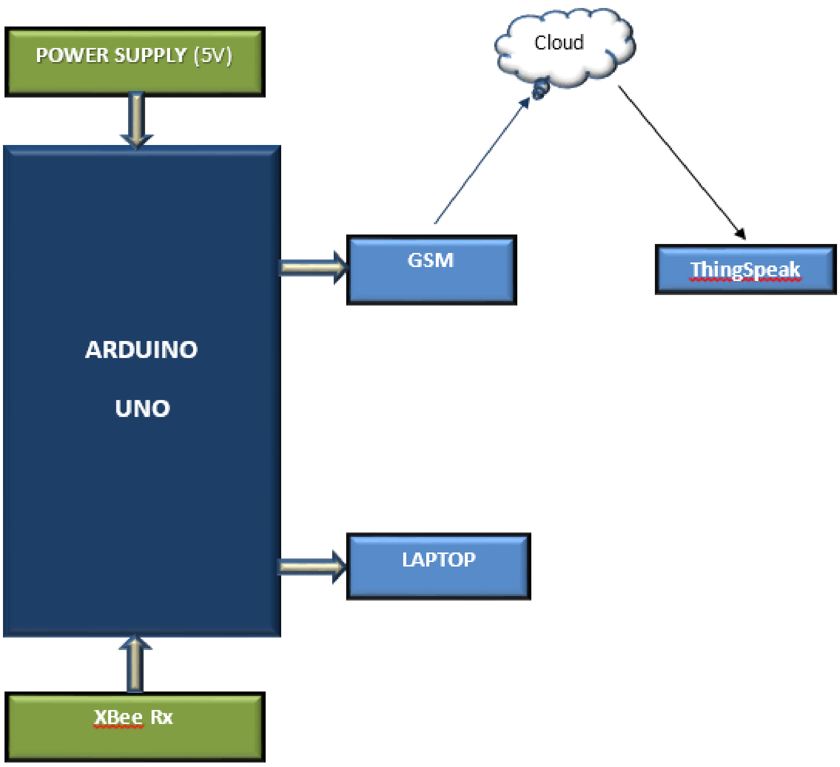


Fig. 3.2.2 : Receiver Section

3.3 : Method of Implementation

In the present work, the programming of Arduino Uno & Nano has been done in C language. Firstly, a miner should wear the proposed intelligent helmet when on the mine field. Then, the proposed intelligent helmet will start the sensors. Following figures represents the prototypes of Helmet.

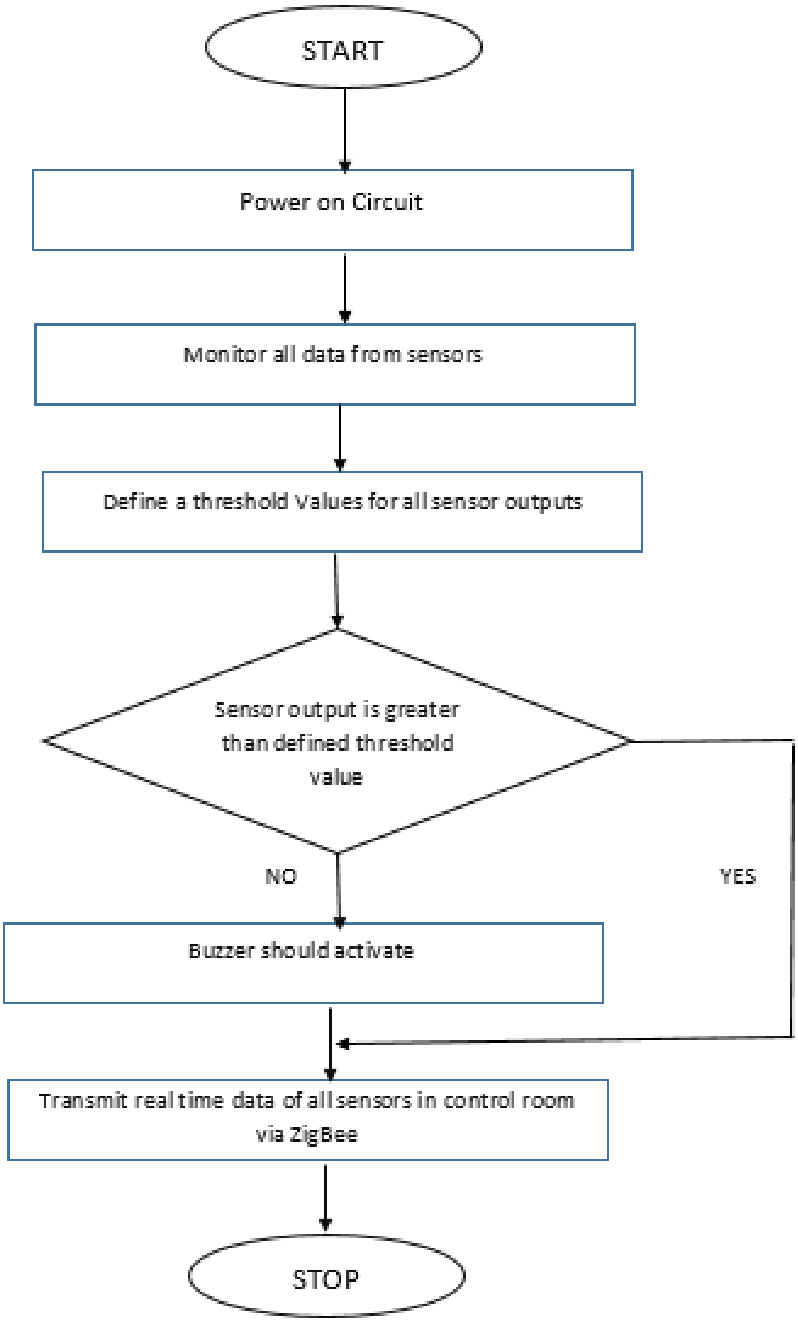


Fig 3.3.1 : Prototypes of Helmet

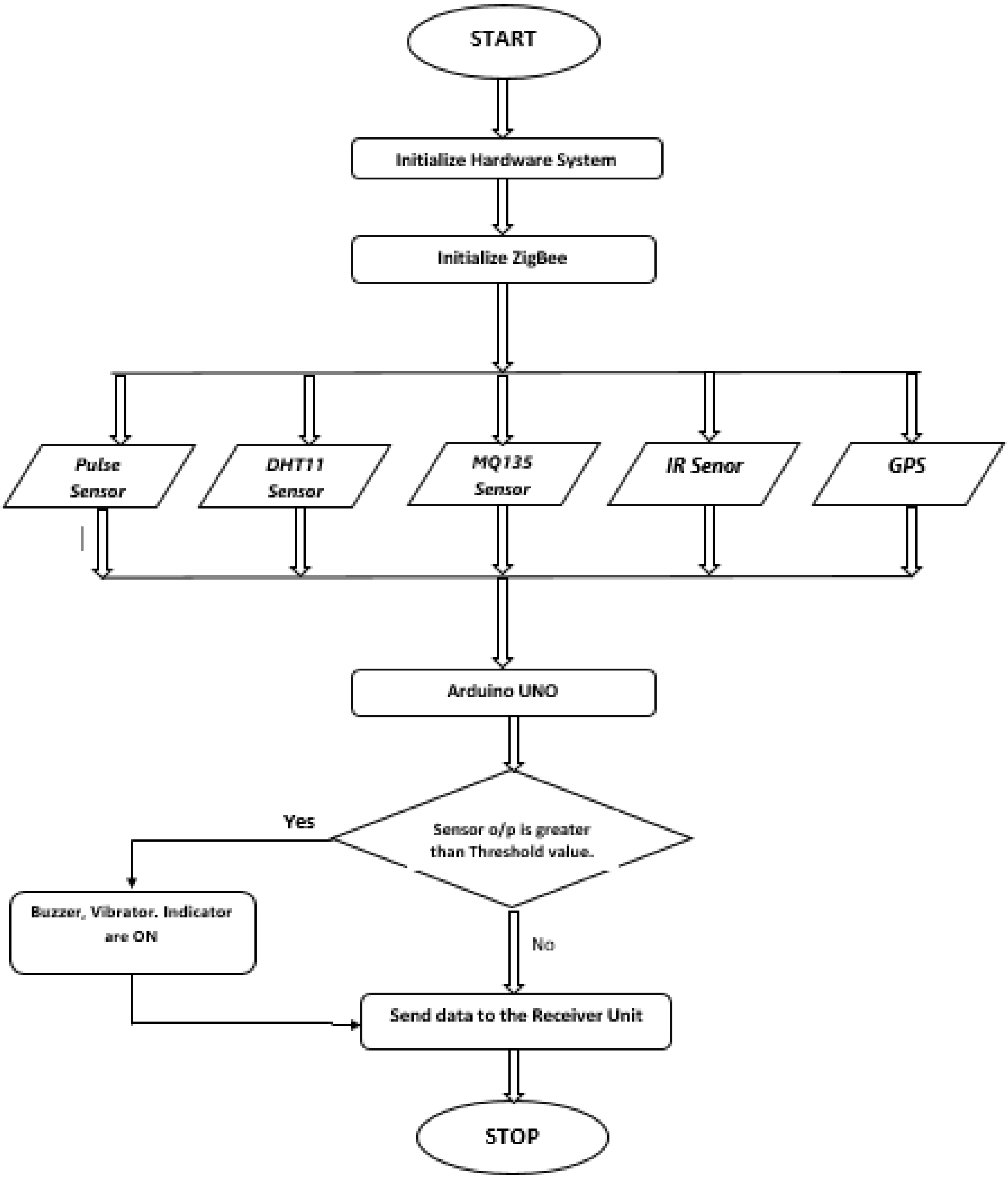


Fig. 3.3.2: Flow chart of the monitoring System for Sensor Unit

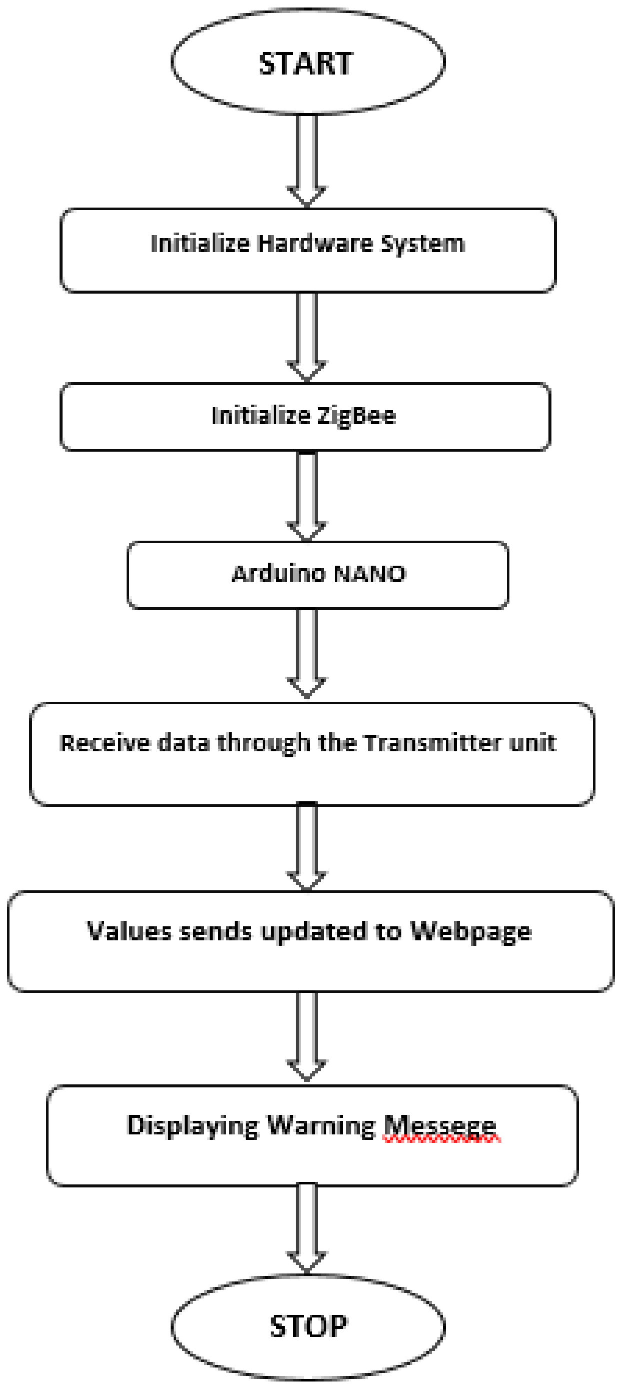


Fig 3.3.3 : Flow chart of the monitoring System for Monitoring Unit

To test the designed real time monitoring system using wireless sensor network, an artificial mining environment is simulated inside the laboratory. As a first implementation, we designed the complete system on a breadboard which is presented in Figure.

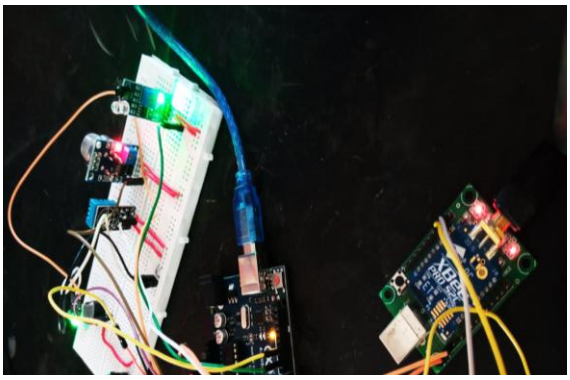
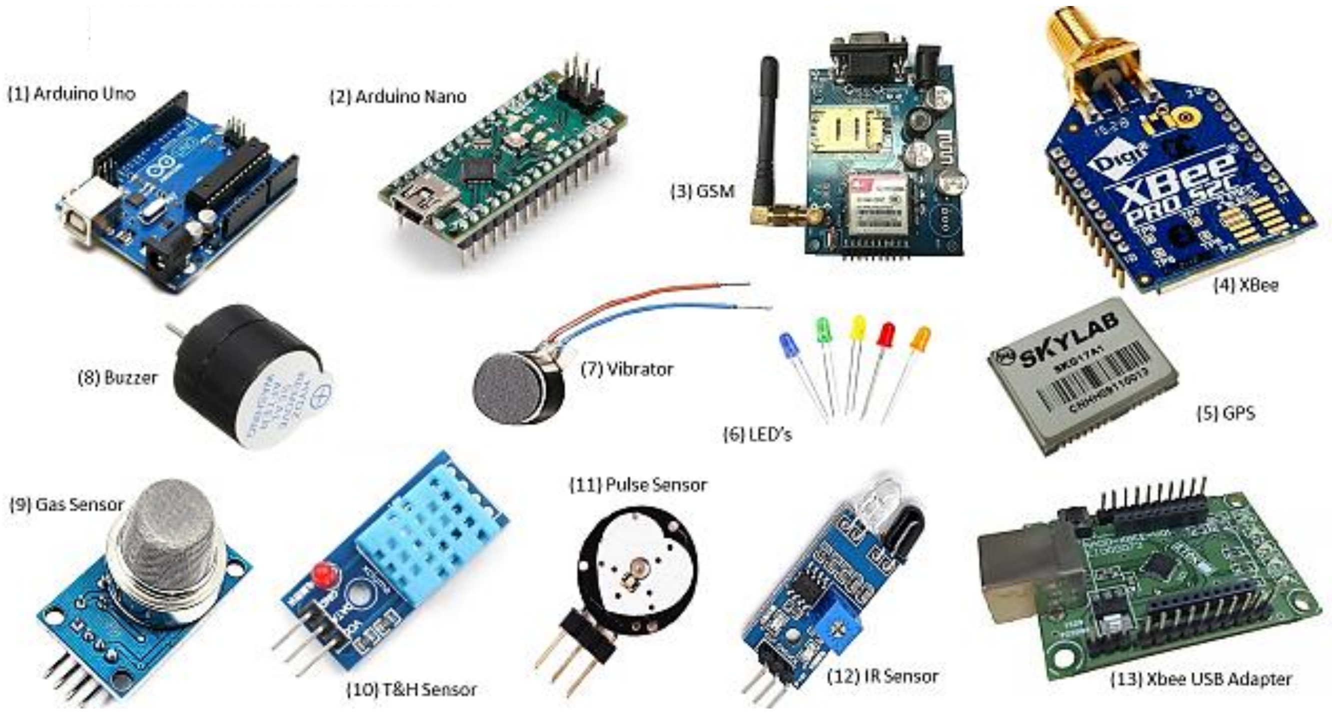


Fig 3.3.3 : Breadboard Connection

The system consists of following components:

- Arduino Board – Model Arduino UNO & NANO
- Xbee – Model Xbee Pro S2C
- Air Quality Gas Sensor – MQ135
- Temperature and Humidity Sensor – DHT-11
- Heart Beat Sensor – XD-58C
- IR Sensor
- GSM SIM900A
- GPS
- Vibration Sensor
- Buzzer
- Indicators LEDs



The final design of the sensor unit consists of the all above components.

By using above circuit, we designed complete real time monitoring system, which is shown in Figure.

Fig 3.3.4 : Complete System

3.4 : Task Vs Time Schedule.

Table 3.4.1 Task vs Time Schedule

Intelligent Helmet for Coal Miners								
Done Under :- Department of Electronic Science, Fergusson College, Pune								
Task	TIME							
	Feb-22		Mar-22		Apr-22		May-22	
	FH	SH	FH	SH	FH	SH	FH	SH
Reporting								
Research								
Coding								
Testing								
Documentation								
Implementation								

	Poor
	Good
	Extreme
	Completed
	Everything Disrupted
FH	First Half
SH	Second Half

Description :-

The time vs task schedule graph is all about how the project kept on and the research was started with finding useful concept visualisation, component selection and interfacing options that how to make the concept possible, afterwards the sensors were taken and the interfacing & testing each sepratley. Then the coding was started. With coding the sensors were tested along with the flow of the go.

Chapter 4

Planning Resources

This chapter consists of the design of the system. This chapter mainly contains the details of required hardware and software. The appropriate working environment is setup with all required components to develop the system. After developing the system, it tested in the particular environment. This chapter explains the step-by-step development of hardware system followed by software development and its implementation.

This monitoring system contains several components like boards (Arduino boards UNO & NANO, Xbee modules and Zigbee USB interfacing board), different sensors and other small electronic components. This chapter gives a detailed review of each of this part along with its working principle.

This chapter deals with the hardware implemented for the real time monitoring system. The details of each components used were described briefly based on its functionality and specifications. The flow chart and block diagram shows the organization and working of the system. The above mentioned hardware and design plan has been described in the subsequent chapter which explains the implementation part**.**

- First of all we gather the information from various resources.
- After this checked the specification of components as per our requirements and then buy the components.
- Write code for each sensor & test separatley in Arduino IDE with interfacing.
- Configure the both XBee modules and Write code for Transmitter & Receiver part separately in Arduino IDE.
- Then build the hardware connection on breadboard and perform test on subject.

4.1 Hardware :

4.1.1 : Arduino UNO

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits. The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.

Fig. 4.1.1 : Arduino UNO

What is Arduino Uno R3 ?

The Arduino UNO R3 is frequently used microcontroller board in the family of an Arduino. This is the latest third version of an Arduino board and released in the year 2011. The main advantage of this board is if we make a mistake we can change the microcontroller on the board. The main features of this board mainly include, it is available in DIP (dual-inline-package), detachable and ATmega328 microcontroller. The programming of this board can easily be loaded by using an Arduino computer program. This board has huge support from the Arduino community, which will make a very simple way to start working in embedded electronics, and many more applications.

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means "one" in the language of "Italian" and was selected for marking the release of Arduino's IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. The Uno-board is the primary in a sequence of USB-**Arduino boards**, & the reference model designed for the Arduino platform.

Technical Specifications :

The Arduino Uno R3 board includes the following specifications.

- It is an ATmega328P based Microcontroller
- The Operating Voltage of the Arduino is 5V
- The recommended input voltage ranges from 7V to 12V
- The i/p voltage (limit) is 6V to 20V
- Digital input and output pins-14
- Digital input & output pins (PWM)-6
- Analog i/p pins are 6
- DC Current for each I/O Pin is 20 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory -32 KB, and 0.5 KB memory is used by the boot loader
- SRAM is 2 KB
- EEPROM is 1 KB
- The speed of the CLK is 16 MHz
- In Built LED
- Length and width of the Arduino are 68.6 mm X 53.4 mm
- The weight of the Arduino board is 25 g

Arduino UNO R3 Pin Diagram :

The Arduino Uno R3 pin diagram is shown below. It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, a USB connection, resonator-16Mhz, a power jack, an ICSP header an RST button.

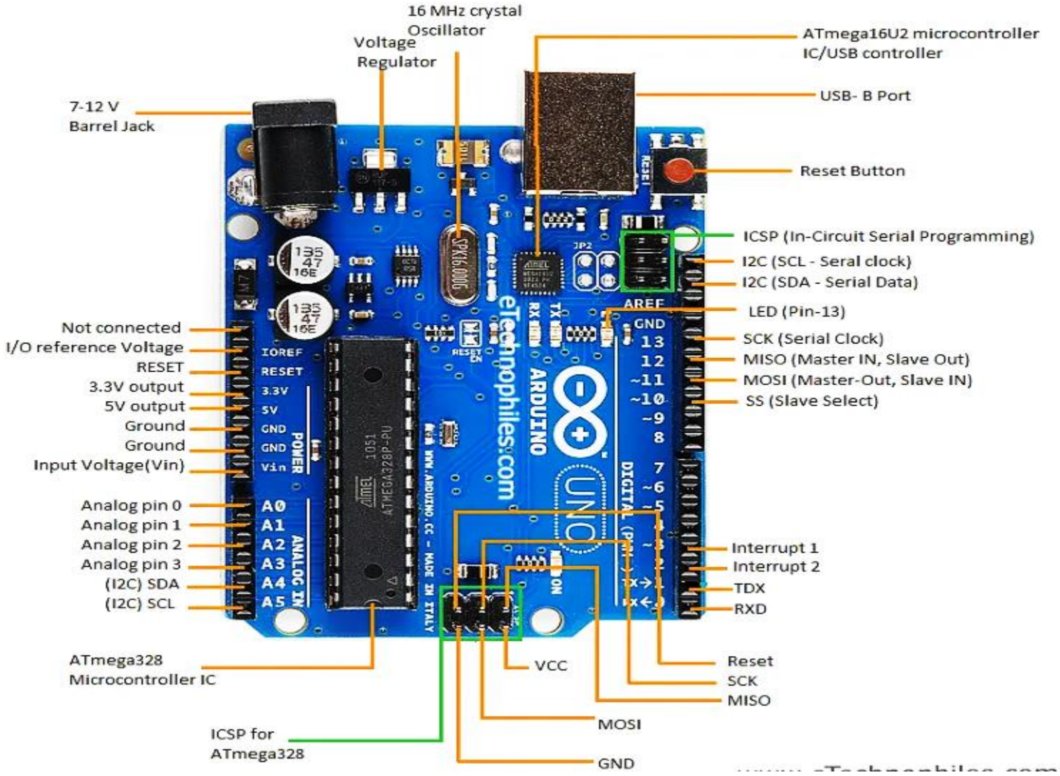


Fig. 4.1.2 : Arduino UNO Pin Diagram

Power Supply

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) mainly includes a battery or an AC to DC adapter. The connection of an adapter can be done by plugging a center-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of Vin as well as GND. The power pins of an Arduino board include the following.

- **Vin:** The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.
- **5Volts:** The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.
- **3V3:** A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.
- **GND:** GND (ground) pins

Communication

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication.

- **UART**
  An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.
- **I2C**
  An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.
- **SPI Pins**
  The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the SPI communication with the help of the SPI library.
  MOSI (Pin11)
  This is the master out slave in the pin, used to transmit the data to the devices
  MISO (Pin12)
  This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.
  SCK (Pin13)
  The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming ATmega microcontroller directly with the boot loader.

Memory

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB

Input and Output

We know that an arguing Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write(). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

Serial Pins
The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

External Interrupt Pins
The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

PWM Pins
The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

LED Pin
An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

TWI (2-Wire Interface) Pins
The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

AREF (Analog Reference) Pin
An analog reference pin is the reference voltage to the inputs of an analog i/ps using the function like analog Reference().

Reset (RST) Pin
This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

ICSP pin - The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.

Power LED Indicator- The ON status of LED shows the power is activated. When the power is OFF, the LED will not light up.

TX and RX LED's- The successful flow of data is represented by the lighting of these LED's.

USB- It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.

Crystal Oscillator- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a powerful board.

Voltage Regulator- The voltage regulator converts the input voltage to 5V.

Connector Pinouts :

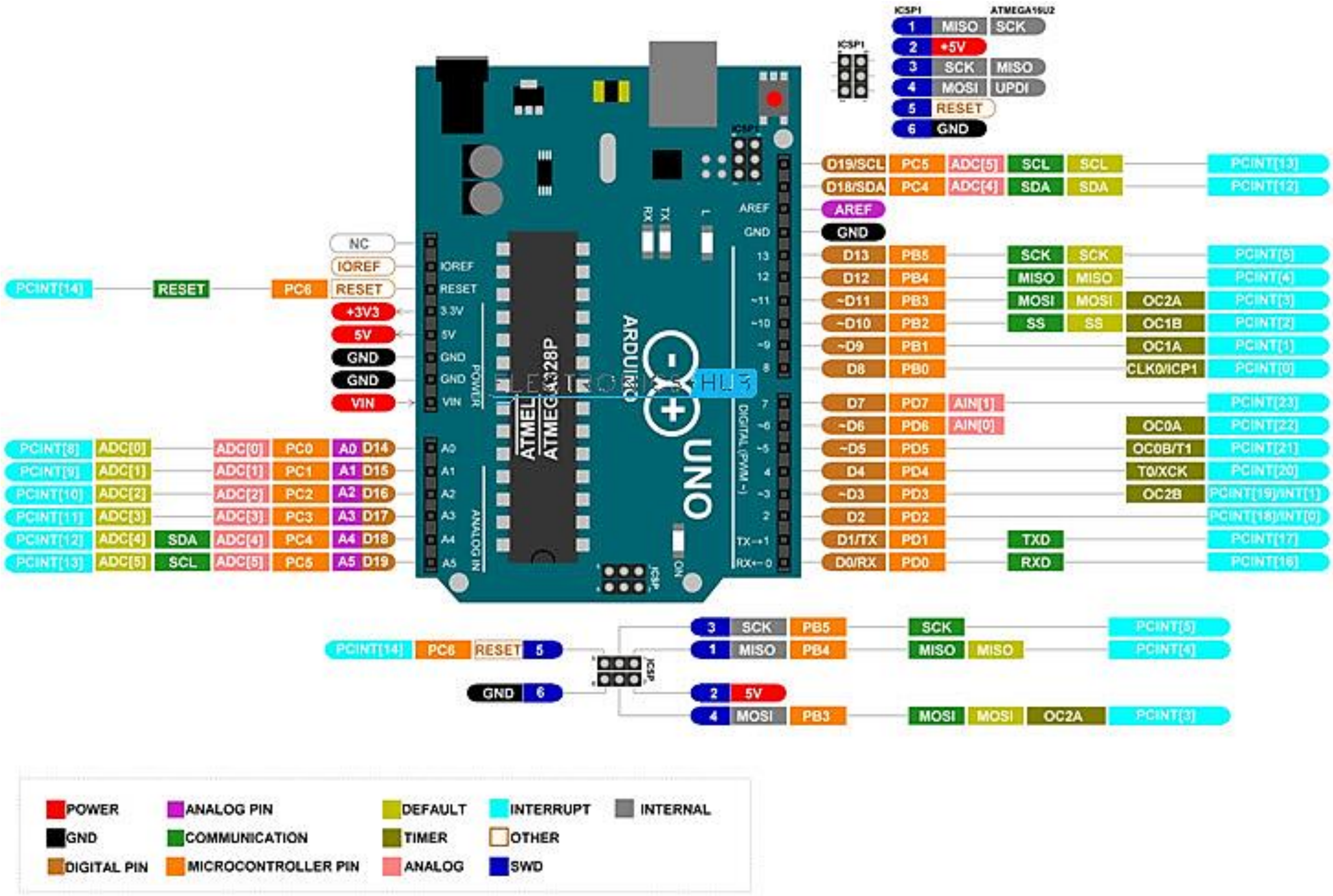


Fig. 4.1.3 : Arduino UNO Connector Pinouts

JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

Table. 4.1.1 : Arduino UNO Analog Pins

JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Table 4.1.2 : Arduino UNO Digital Pins

4.1.2 : Arduino Nano

The Arduino Nano development board was first released in 2008 by Arduino and is one of the most popular Arduino boards. It is based on the ATmega328 8-bit microcontroller by Atmel (Microchip Technology).

This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in packaging. It doesn’t have any DC jack so that the power supply can be given using a small USB port otherwise straightly connected to the pins like VCC & GND. This board can be supplied with 6 to 20volts using a mini USB port on the board.

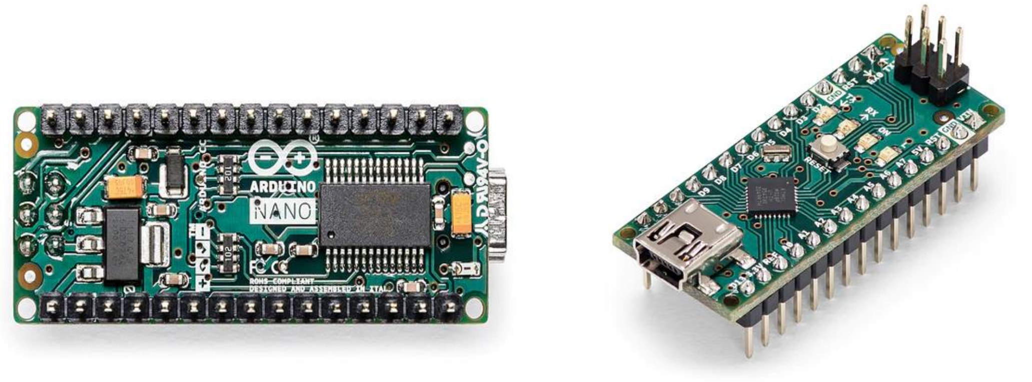


Fig. 4.2.1: Arduino NANO

What is an Arduino Nano Board ?

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila/Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper.

Nano’s got the breadboard-ability of the Boarduino and the Mini+USB with smaller footprint than either, so users have more breadboard space. It’s got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It is two layers. That make it easier to hack and more affordable.

Technical Specifications :

The Arduino Uno R3 board includes the following specifications.

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5V
- Input voltage (Vin) is 7V to 12V
- Input/Output Pins are 22
- Analog i/p pins are 6 from A0 to A5
- Digital pins are 14
- Power consumption is 19 mA
- I/O pins DC Current is 40 mA
- Flash memory is 32 KB
- SRAM is 2 KB, EEPROM is 1 KB
- CLK speed is 16 MHz
- Size of the printed circuit board is 18 X 45mm & Weight-7g
- Supports three communications like SPI, IIC, & USART

Arduino Nano Pin Diagram :

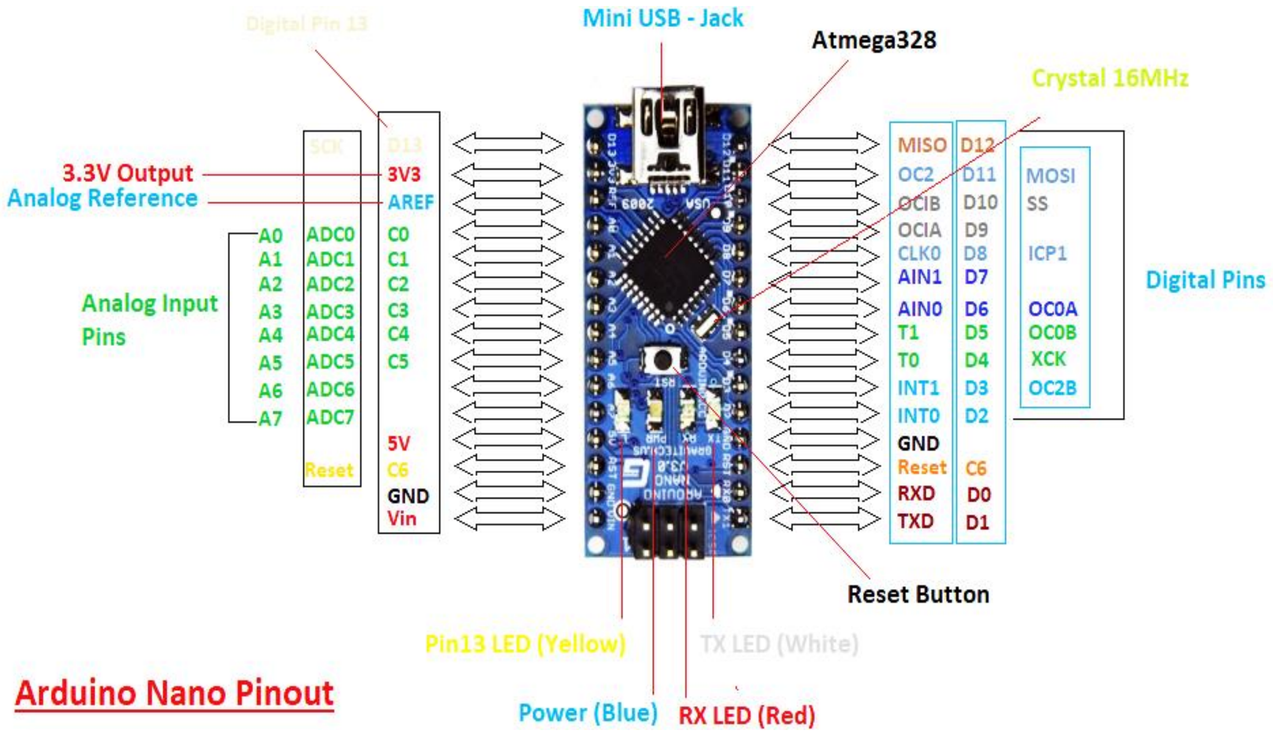


Fig. 4.2.2 : Arduino NANO Pi Diagram.

Arduino Nano has a total of 36 pins. Out of these 8 are analog input pins and 14 digital input/output pins (of which 6 can be used as PWM outputs). Nano has a 16 MHz SMD crystal resonator, a mini USB-B port, an ICSP header, 3 RESET pins, and, a RESET button.

Power Supply

Mini USB : The Mini USB is smaller than the standard USB but thicker than the micro USB. The Nano board is powered through this port. And it also allows us to connect the board to the computer for programming purposes.

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

- **Vin:** It is input power supply voltage to the board when using an external power source of 7 to 12 V.
- **5V:** It is a regulated power supply voltage of the board that is used to power the controller and other components placed on the board.
- **3V3:** This is a minimum voltage generated by the voltage regulator on the nano board
- **GND Pin:** These are the ground pins on the board.
  There are multiple ground pins on the board that can be interfaced accordingly when more than one ground pin is required.

Memory

- Flash memory is 16KB or 32KB that all depends on the Atmega board i.e Atmega168 comes with 16KB of flash memory while Atmega328 comes with a flash memory of 32KB. Flash memory is used for storing code. The 2KB of memory out of total flash memory is used for a bootloader.
- The SRAM memory of 2KB is present in Arduino Nano.
- Arduino Nano has an EEPROM memory of 1KB.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions.

- **Analog Pins:** There are 8 analog pins on the board marked as **A0 – A7**.
  These pins are used to measure the analog voltage ranging between **0 to 5V**.

- **Digital Pins**: Arduino Nano has 14 digital pins starting from D0 to D13.
  These digital pins are used for interfacing third-party digital sensors and modules with Nano board.
- **PWM Pins:** Arduino Nano has 6 PWM pins, which are Pin#3, 5, 6, 9, 10 and 11. (All are digital pins). These pins are used to generate an 8-bit PWM (Pulse Width Modulation) signal.

- **External Interrupts:** Pin 2 and 3 are used for generating external interrupts normally used in case of emergency, when we need to stop the main program and call important instructions.
  The main program resumes once interrupt instruction is called and executed.

- **AREF:** This pin is used to give reference voltage to the input voltage

- **RST Pin(Reset):** This pin is used to reset the microcontroller.

Communication :

- **Serial Pins (UART) :** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

These pins are used for serial communication where:

1. Pin#0 is RX used for receiving serial data.
2. Pin#1 is Tx used for transmitting serial data.

- **SPI Protocol:** Four pins 10(SS->Slave Select), 11(MOSI -> Master Out Slave In), 12(MISO -> Master In Slave Out) and 13(SCK -> Serial Clock) are used for SPI (Serial Peripheral Interface) Protocol.
  SPI is an interface bus and is mainly used to transfer data between microcontrollers and other peripherals like sensors, registers, and SD cards.
  SCK-It stands for Serial Clock. These are the clock pulses, that are used to synchronize the transfer of data.

MISO-It stands for Master Input/ Slave Output. This data line in the MISO pin is used to receive the data from the Slave.

MOSI-It stands for Master Output/ Slave Input. This line is used for sending data to the peripherals.
SS-It stands for Slave Select. This line is used by the master. It acts as the enable line. When a device's Slave Select pin value is LOW, it can communicate with the master. When its value HIGH, it ignores the master. This allows us to have multiple SPI peripheral devices sharing the same MISO, MOSI, and CLK lines.

- **I2C Protocol:** I2C communication is developed using A4 and A5 pins
  SCL-It stands for Serial Clock. It is defined as the line that transfers the clock data. It is used to synchronize the shift of data between the two devices. The Serial Clock is generated by the master device.
  SDA-It stands for Serial Data. It is defined as the line used by the slave and master to send and receive the data. That's why it is called a data line**,** while SCL is called a clock line.

LED Indicators on Arduino Nano:

Arduino Nano board consist of 4 LED indicators:

- **Transmitting Data Indicator LED (White):** When this LED ON, the Arduino Nano is transmitting data to the computer.

- **Receiving Data Indicator LED (Red):** When this LED lights up, the board is receiving data from the computer.

- **Power Indicator**: It indicates the status of the battery. It can also display the voltage of the battery onto the LCD connected to the Arduino board.

- **Pin 13 LED Indicator (Blue):** In the board, there is a built-in LED connected to digital pin 13. When this pin is set HIGH or 1, the LED turns ON. When the pin is set LOW or 0, the LED turns OFF.

SMD Crystal: The Surface Mount crystals have better stability than other crystals and can be easily soldered onto the PCB board.

ICSP pins:

The ICSP header consists of 6 pins:

It stands for In-Circuit Serial Programming. We can use these pins to program the Arduino board's firmware. The firmware with the new functionalities is uploaded to the microcontroller with the help of the ICSP header

Connector Pinouts :

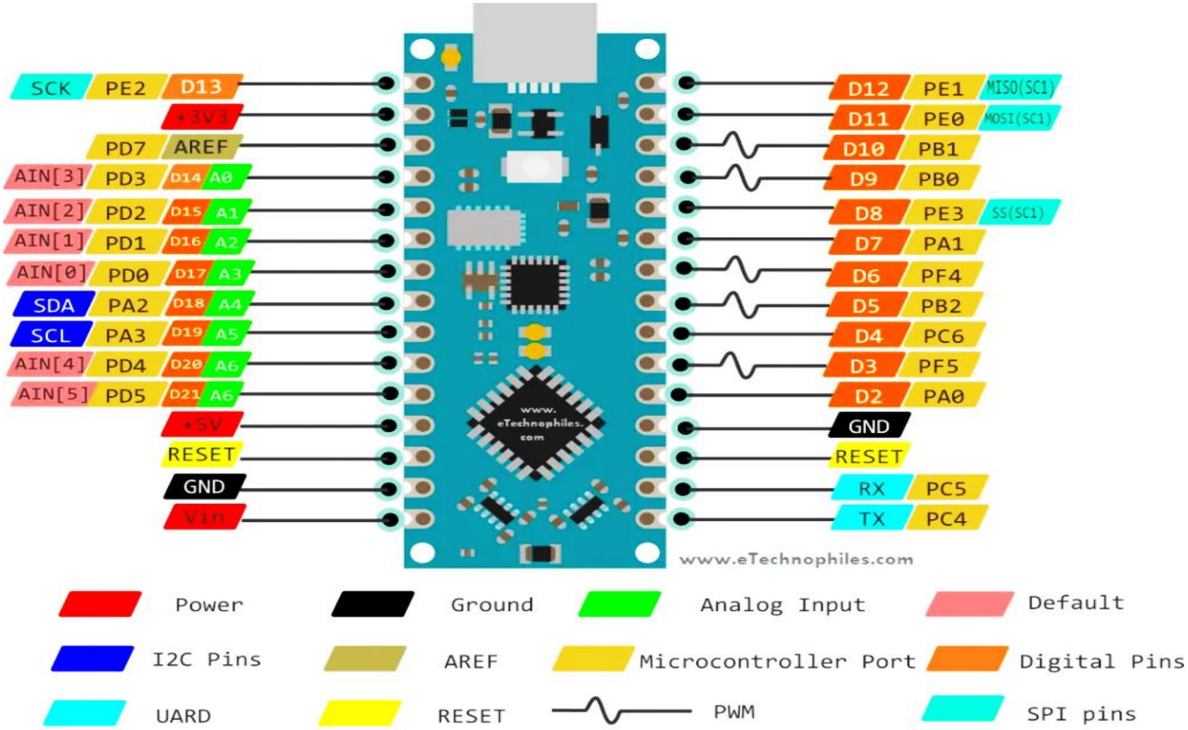


Fig. 4.2.3 : Arduino NANO Connector Pinouts

Table 4.2.1 : Arduino NANO JMicro USB/Power Pins

Pin	Function	Type	Description
1	VBUS	Power	5V USB Power
2	D-	Differential	USB differential data -
3	D+	Differential	USB differential data +
4	ID	Digital	Unused
5	GND	Power	Ground

Table 4.2.2 : Arduino NANO Digi Pins

Pin	Function	Type	Description
1	TX1	Digital	UART TX / Digital Pin 1
2	RX0	Digital	UART RX / Digital Pin 0
3	RST	Digital	Reset
4	GND	Power	Ground
5	D2	Digital	Digital Pin 2
6	D3	Digital	Digital Pin 3
7	D4	Digital	Digital Pin 4
8	D5	Digital	Digital Pin 5
9	D6	Digital	Digital Pin 6
10	D7	Digital	Digital Pin 7
11	D8	Digital	Digital Pin 8
12	D9	Digital	Digital Pin 9
13	D10	Digital	Digital Pin 10
14	D11	Digital	Digital Pin 11
15	D12	Digital	Digital Pin 12

Table 4.2.3 : Arduino NANO Analog Pins

Pin	Function	Type	Description
1	D13	Digital	Digital Pin 13
2	3.3V	Power	3.3V Power
3	REF	Analog	NC
4	A0	Analog	Analog Pin 0
5	A1	Analog	Analog Pin 1
6	A2	Analog	Analog Pin 2
7	A3	Analog	Analog Pin 3
8	A4	Analog	Analog Pin 4
9	A5	Analog	Analog Pin 5
10	A6	Analog	Analog Pin 6
11	A7	Analog	Analog Pin 7
12	VUSB	Power	USB Input Voltage
13	REC	Digital	BOOTSEL
14	GND	Power	Ground
15	VIN	Power	Voltage Input

4.1.3 : Gas Sensor : MQ135

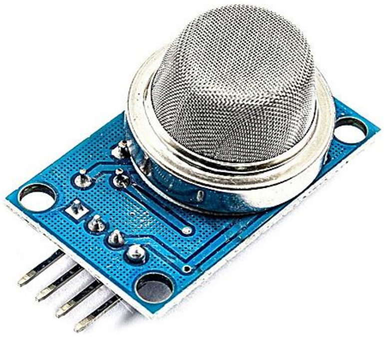


Fig. 4.3.1 : Gas Sensor (MQ135)

Feature :

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Simple drive circuit
- Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer
- Preheat duration : 20 seconds
- Operating Voltage : +5V
- Analog output voltage : 0V to 5V
- Digital output voltage : 0V or 5V (TTL Logic)
- Power consumption : 150mA

Technical Parameters : (Table 4.3.1)

Mode			MQ135
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite, Metal cap
Target Gas			ammonia gas, sulfide, benzene series steam
Detection range			10～1000ppm(ammonia gas, toluene, hydrogen, smoke)
Standard Circuit Conditions	Loop Voltage	Vc	≤24V DC
	Heater Voltage	VH	5.0V±0.1V AC or DC
	Load Resistance	RL	Adjustable
Sensor character under standard test conditions	Heater Resistance	RH	29Ω±3Ω (room tem.)
	Heater consumption	PH	≤950mW
	Sensitivity	S	Rs(in air)/Rs(in 400ppm H2)≥5
	Output Voltage V	Vs	2.0V～4.0V（in 400ppm H2）
	Concentration Slope α	α	≤0.6(R400ppm/R100ppm H2)
Standard test conditions	Tem. Humidity		20℃±2℃；55%±5%RH
	Standard test circuit		Vc:5.0V±0.1V VH: 5.0V±0.1V
	Preheat time		Over 48 hours

Specifications : (Table 4.3.2)

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	AC OR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33Ω±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10℃-45℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21% (standard condition) Oxygen concentration can affect the sensitivity.	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical condition	Remark 2
Rs	Sensing Resistance	30KΩ-200KΩ (100ppm NH3)	Detecting concentration scope ： 10ppm-300ppm NH3 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH3	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20℃±2℃ Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

Sr No	Parts	Material
1	Gas sensing layer	SnO2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al2O3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

Pin Configuration :

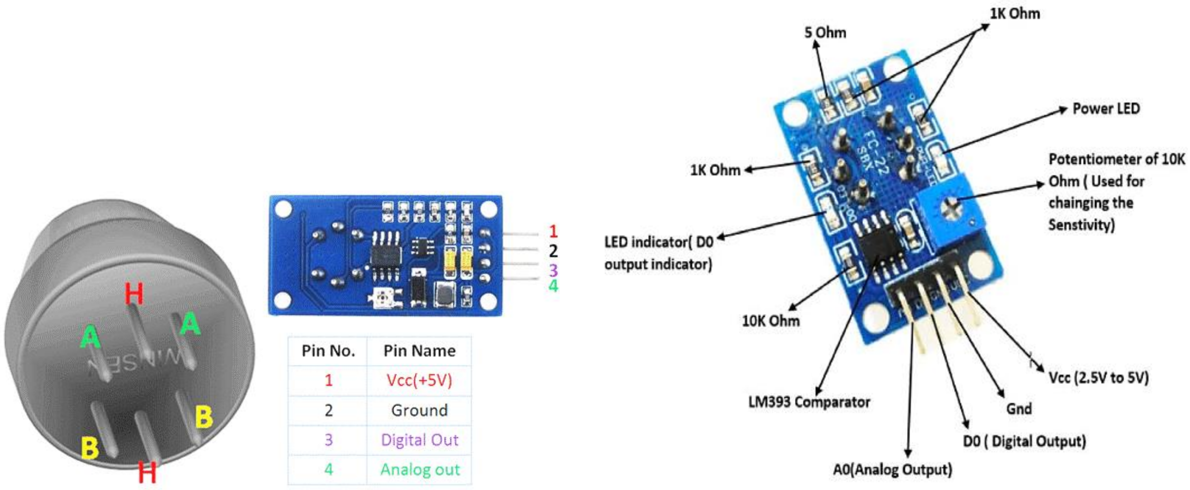


Fig. 4.3.2 : Pin Configuration

Pin No	Pin Name	Description
For Module		
1	Vcc	Used to power the sensor, Generally the operating voltage is +5V.
2	Ground	Used to connect the module to system ground.
3	Digital Out	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
4	Analog Out	This pin outputs 0-5V analog voltage based on the intensity of the gas.
For Sensor		
1	H -Pins	Out of the two H pins, one pin is connected to supply and the other to ground
2	A-Pins	The A pins and B pins are interchangeable. These pins will be tied to the Supply voltage.
3	B-Pins	A pins and B pins are interchangeable. One pin will act as output while the other will be pulled to ground.

(Table 4.3.3 : Pin Configuration)

Gas sensors (also known as gas detectors) are electronic devices that detect and identify different types of gasses. They are commonly used to detect toxic or explosive gasses and measure gas concentration. Gas sensor detect the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated.

When it comes to measuring or detecting a particular Gas, the MQ series Gas **sensors** are the most inexpensive and commonly used ones. MQ135 is available as a module or as just the sensor alone. If you are trying to only detect (not measuring PPM) the presence of a gas, then you can buy it as a module since it comes with an op-amp comparator and a digital output pin. But if you planning to measure the PPM of a gas it is recommend buying the sensor alone without module.

The MQ-135 Gas sensor can detect gases like Ammonia (NH_3), sulfur (S), Benzene (C_6H_6), CO_2, and other harmful gases and smoke. Similar to other MQ series gas sensor, this sensor also has a digital and analog output pin. When the level of these gases go beyond a threshold limit in the air the digital pin goes high. This threshold value can be set by using the on-board potentiometer. The analog output pin, outputs an analog voltage which can be used to approximate the level of these gases in the atmosphere.

The MQ135 air quality sensor module operates at 5V and consumes around 150mA. It requires some pre-heating before it could actually give accurate results.

Sensitive material of MQ135 gas sensor is SnO_2, which with lower conductivity in clean air. When target pollution gas exists, the sensor's conductivity gets higher along with the gas concentration rising. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit. MQ135 gas sensor has high sensitivity to ammonia gas, sulfide, benzene series steam, also can monitor smoke and other toxic gases well. It can detect kinds of toxic gases and is a kind of low-cost sensor for kinds of applications.

How to use MQ-135 Sensors to detect gases

You can either use the digital pin or the analog pin to do this. Simply power the module with 5V and you should notice the power LED on the module to glow and when no gas it detected the output LED will remain turned off meaning the digital output pin will be 0V. Remember that these sensors have to be kept on for pre-heating time (mentioned in features above) before you can actually work with it. Now, introduce the sensor to the gas you want to detect and you should see the output LED to go high along with the digital pin, if not use the potentiometer until the output gets high. Now every time your sensor gets introduced to this gas at this particular concentration the digital pin will go high (5V) else will remain low (0V).

You can also use the analog pin to achieve the same thing. Read the analog values (0-5V) using a microcontroller, this value will be directly proportional to the concentration of the gas to which the sensor detects.

Detect Harmful Gases using Digital Pin:

The digital output pin of the sensor can be used to detect harmful gases in the environment. The sensitivity of the digital pin can be controlled by using the 10k potentiometer. If the gas is detected the indicator LED D0 will turn on and the digital pin will go from logic high to logic low (0V). The LM393 Op-Amp Comparator IC is used to compare the actual gas value with the value set using the potentiometer. If the actual gas value increases than the set value then the digital output pin gets low.

Because of the onboard LM393 comparator IC the MQ135 Gas sensor module can also be used without the need of an external microcontroller. Simply power up the module and set the sensitivity of the digital pin using the potentiometer, then when the module detects the gas the digital pin will go low. This digital pin can directly be used to drive a buzzer or LED with the help of simple transistors.

How to use MQ-135 sensor to measure PPM

The Analog output pin of the sensor can be used to measure the PPM value of the required gas. To do this we need to use an external microcontroller like Arduino. The microcontroller will measure the value of analog voltage and perform some calculations to find the value of Rs/Ro where Rs is the sensor resistance when gas is present and Ro is sensor resistance at clean air. Once we find this ratio of Rs/Ro we can use it to calculate the PPM value of required gas using the graph below which is taken from the datasheet of MQ135 Sensor

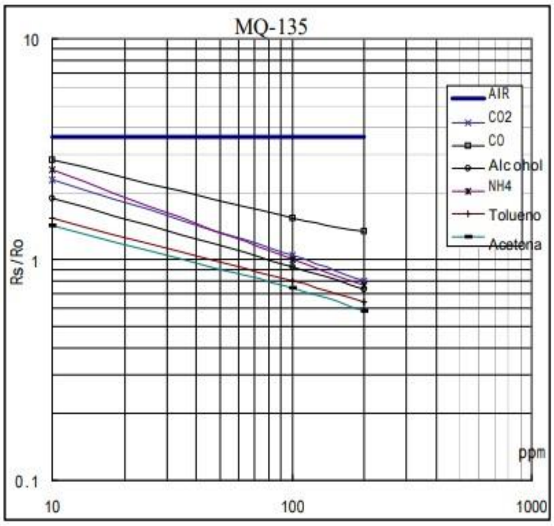


Fig. 4.3.3 : PPM Values Graph

MQ-135 gas sensor applies SnO2 which has a higher resistance in the clear air as a gas-sensing material. When there is an increase in polluting gases, the resistance of the gas sensor decreases along with that. To measure PPM using MQ-135 sensor we need to look into the (Rs/Ro) v/s PPM grap.

The above figure shows shows the typical sensitivity characteristics of the MQ-135 for several gases. in their:

Temp: 20, Humidity: 65%, O2 concentration 21%, RL=20kΩ,

Ro: sensor resistance at 100ppm of NH3 in the clean air.

Rs:sensor resistance at various concentrations of gases.

The value of Ro is the value of resistance in fresh air (or the air with we are comparing) and the value of Rs is the value of resistance in Gas concentration. First you should calibrate the sensor by finding the values of Ro in fresh air and then use that value to find Rs using the below formula:

Resistance of sensor(Rs): Rs=(Vc/VRL-1)×RL

Once we calculate Rs and Ro we can find the ratio and then using the graph shown above we can calculate the equivalent value of PPM for that particular gas.

Interfacing of MQ-135 gas sensor with Arduino :

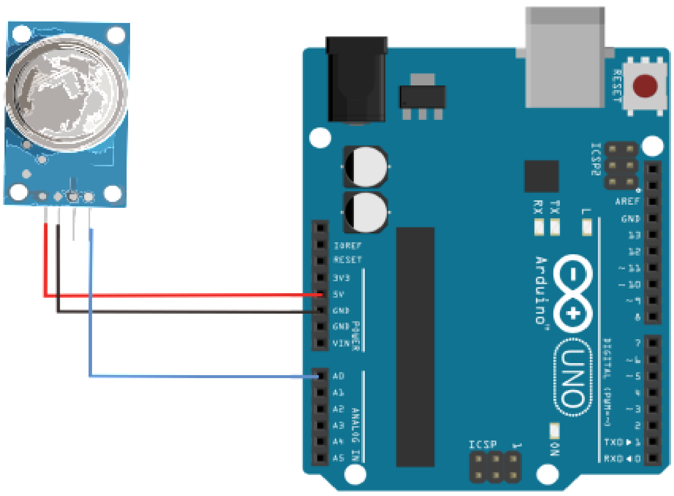


Fig. 4.3.4 : Interfacing of MQ-135 gas sensor with Arduino :

Wire the circuit as follows:

Arduino A0 pin	with	Sensor A0
Arduino D0 pin	with	Sensor D0
Arduino 5Vpin	with	Sensor Vcc
Arduino GND pin	with	Sensor GND

When no gas digital output is 1 and analog output gives 1023 max value. When gas is present digital output is 0 and analogue output is much less than 1023. Using potentiometer on chip we can control the turning OFF point of digital pin at some value of analog pin. The sensor needs a loadresistor at the output to ground. Its value could be from 2kOhm to 47kOhm. The lower the value, the less sensitive is the sensor. The higher the value, the less accurate is sensor for higher concentrations of gas. If only one specific gas is measured, the load-resistor can be calibrated by applying a known concentration of that gas. If the sensor is used to measure any gas (like in a air quality detector) the load-resistor could be set for a value of about 1V output with clean air. Choosing a good value for the load-resistor is only valid after the burn-in time.

Testing the Circuit :

1. After hardware connection, insert the sample sketch into the Arduino IDE.
2. Using a USB cable, connect the ports from the Arduino to the computer.
3. Upload the program.
4. See the results in the serial monitor.

Applications:

- Used to detect leakage/excess of gases like Ammonia, nitrogen oxide, alcohols, aromatic compounds, sulfide and smoke.
- It is widely used in domestic gas alarm, industrial gas alarm and portable gas detector.
- Air quality monitors.
- Domestic air pollution detector.
- Industrial air pollution detector.
- Portable air pollution detector.

4.1.4 : Temperature And Humidity Sensor : DHT11

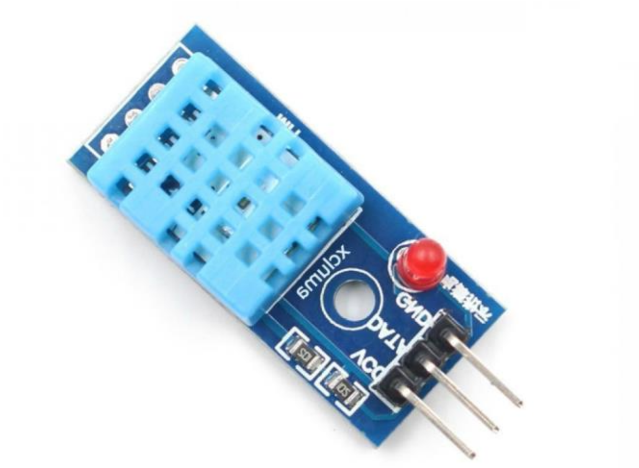


Fig. 4.4.1 : Temperature And Humidity Sensor : DHT11

Feature :

- Good precision .
- Resistive type .
- Full range temperature compensated .
- Relative humidity and temperature measurement .
- Calibrated digital signal.
- Outstanding long-termstability .
- Extra components not needed .
- Long transmission distance, up to 100meters.
- Low power consumption .
- pins packaged and fully interchangeable.

Technical Parameters : (Table. 4.4.1 : Technicle Parameters**)**

Model	DHT11
Power supply	3-5.5V DC
Output signal	digital signal via single-bus
Sensing element	Polymer resistor
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius
Accuracy	humidity +-4%RH (Max +-5%RH); temperature +-2.0Celsius
Resolution or sensitivity	humidity 1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-1Celsius
Humidity hysteresis	+-1%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	size 12*15.5*5.5mm

Detailed Specifications : (Table. 4.4.2 : Detailed Specification**)**

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

Pin Configuration :

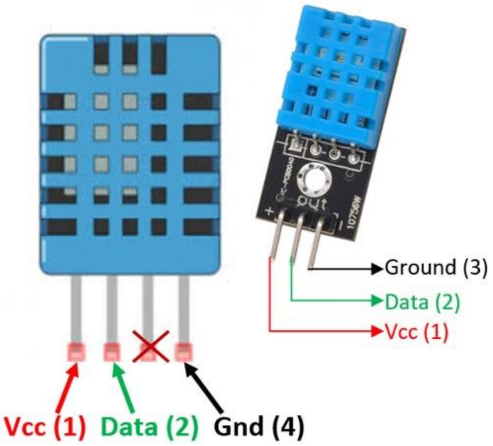


Fig 4.4.2 : Pin Configuration

(Table. 4.4.3 : Pin Configuration)

Pin No	Pin Name	Description
For DHT11 Sensor		
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit
For DHT11 Sensor module		
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

DHT11 Temperature & Humidity Sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a highperformance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request. This module makes it easy to connect the DHT11 sensor to an Arduino or microcontroller as includes the pull-up resistor required to use the sensor. Only three connections are required to be made to use the sensor - Vcc, Gnd, and Output.

What is Temperature & Humidity sensor ?

Temperature and humidity sensor (or rh temp sensor) is devices that can convert temperature and humidity into electrical signals that can easily measure temperature and humidity.

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously.

DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

Working Principle of DHT11 Sensor :

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.
For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA. DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

How to use DHT11 Sensor

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.

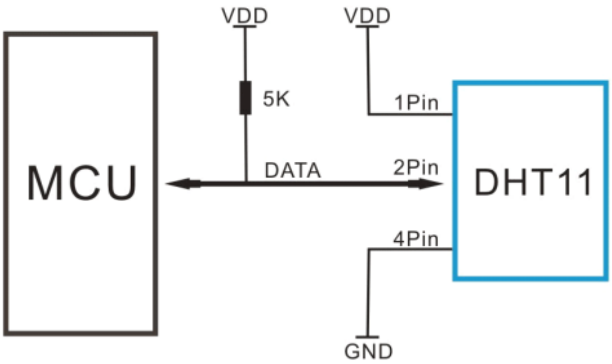
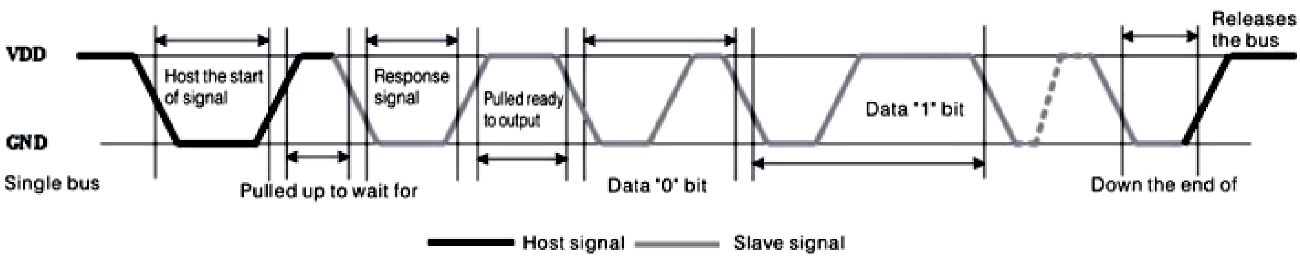


Fig. 4.4.3 : Connection Diagram : DHT11

As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interfaceDHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

If you are trying to interface it with some other MCU, then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below.



What is Relative Humidity ?

The DHT11 measures relative humidity. The relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in the air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew.

The saturation point changes with air temperature. Cold air can hold less water vapor before it becomes saturated, and hot air can hold more water vapor before it becomes saturated.

The formula to calculate relative humidity is:

$$RH = \left(\frac{\rho_w}{\rho_s}\right)\ x\ 100\%$$

RH : *Relative Humidity*

ρ_w : *Density of water vapor*

ρ_s : *Density of water vapor at saturation*

The relative humidity is expressed as a percentage. At 100% RH, condensation occurs, and at 0% RH, the air is completely dry.

How DHT11 Measures Temperature and Humidity :

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The DHT11 measures temperature with a surface mounted NTC temperature sensor (thermistor) built into the unit. With the plastic housing removed, you can see the electrodes applied to the substrate:

Front view with cover removed

An IC mounted on the back of the unit converts the resistance measurement to relative humidity. It also stores the calibration coefficients, and controls the data signal transmission between the DHT11 and the Arduino:

Rear view with cover removed

The DHT11 uses just one signal wire to transmit data to the Arduino. Power comes from separate 5V and ground wires. A 10K Ohm pull-up resistor is needed between the signal line and 5V line to make sure the signal level stays high by default (see the datasheet for more info).

There are two different versions of the DHT11 you might come across. One type has four pins, and the other type has three pins and is mounted to a small PCB. The PCB mounted version is nice because it includes a surface mounted 10K Ohm pull up resistor for the signal line. Here are the pinouts for both versions:

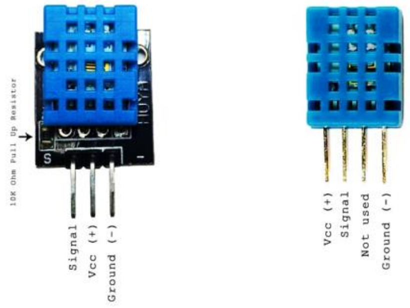


Fig. 4.4.4 : DHT11 Pinouts

DHT11 Sensor Interfacing with Arduino UNO :

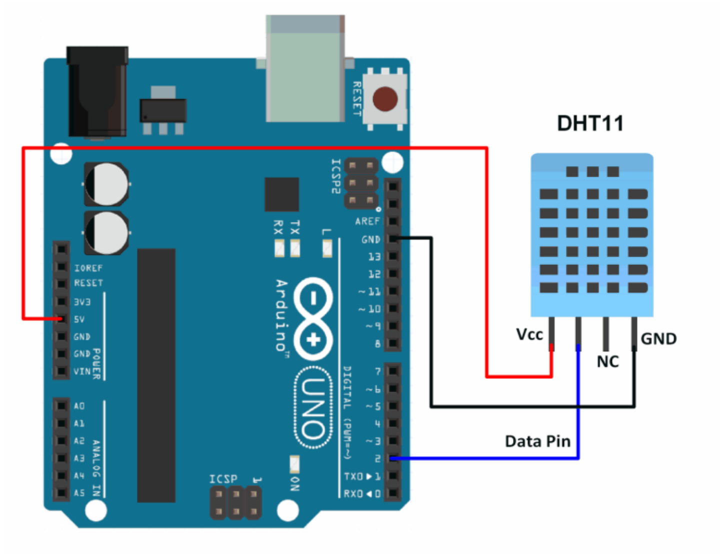


Fig. 4.4.5 : DHT11 Sensor Interfacing with Arduino UNO

Wire the circuit as follows:

Arduino D2 pin	with	Sensor Data pin
Arduino 5Vpin	with	Sensor Vcc
Arduino GND pin	with	Sensor GND

To read sensor data from the DHT11 sensor, Arduino must first send it the start signal. To do so, the pin that DHT11's DATA pin is interfaced with must be set to a digital output. A digital pulse of 18 milliseconds must be passed to the DATA pin, followed by a rising edge. Immediately afterward, the Arduino pin must be set to a digital input with an internal pull-up. Now, read the response signal from the DHT11 at the Arduino pin. If a falling edge is detected within 90 microseconds, this means that DHT11 has successfully sent a response pulse.

The data received from the DHT11 sensor can be sampled by polling the logical level of the digital pulse while tracking the pulse width. It's possible to track the pulse width by measuring the time elapsed from a particular instant of time, while polling for a logical HIGH or LOW. There are millis() and micros() functions available that track the time elapsed since Arduino boots. The millis() function provides the time from the boot in milliseconds and the micros() function provides the time from the boot in microseconds.

Testing the Circuit :

1. After hardware connection, insert the sample sketch into the Arduino IDE.

2. Using a USB cable, connect the ports from the Arduino to the computer.

3. Upload the program.

4. See the results in the serial monitor.

Applications :

- DHT11 Relative Humidity and Temperature Sensor can be used in many applications like:
- HVAC (Heating, Ventilation and Air Conditioning) Systems.
- Weather Stations.
- Medical Equipment for measuring humidity.
- Home Automation Systems.
- Automotive and other weather control applications .

4.1.5 : Heartbeat/Pulse Sensor : XD58c

Fig 4.5.1 : Heartbeat/Pulse Sensor : XD58c

Features :

- This is a heart rate/biometric pulse rate detecting sensor
- This is a plug & play type sensor
- Consumption of current is 4mA
- Length is 0.625”
- Width is 0.125
- Integral Amplification
- Circuit for cancellation of noise

Specifications :

- The maximum current is 100mA
- Heartbeat deduction output LED
- VCC is +5v DC through high-quality regulation
- The operating voltage of this sensor is +5V/+3.3V
- Light source – 660nm super Red LED
- Output data level is 5V TTL

Technical Parameters : (Table 4.5.1;Technicle Paarameters)

Maximum Ratings	VCC	3.0 – 5.5V
	IMax (Maximum Current Draw)	< 4mA
	VOut (Output Voltage Range)	0.3V to Vcc
Wavelength	LED Output	565nm
	Sensor Input	525nm
Dimensions	L x W (PCB)	15.8mm (0.625”)
	Lead Length	20cm (7.8”)

Pin Configuration :

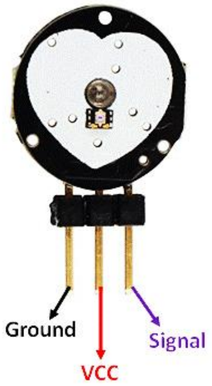


Fig 4.5.2 : XD58c Pin Configuration

Pin Number	Pin Name	Wire Colour	Description
1	Ground	Black	Connected to the ground of the system
2	Vcc	Red	Connect to +5V or +3.3V supply voltage
3	Signal	Purple	Pulsating output signal.

Table 4.5.2;Pin Configuration

XD58C is a low-cost optical heart-rate sensor designed and made by World Famous Electronics. This is an open-source hardware project by Joel Murphy and Yury Gitman. You can learn more about this open-source hardware project at pulsesensor.com.

The sensor is made of an ambient light sensor APDS-9008 from Avago, and a green reverse mounted LED AM2520ZGC09 from Kingbright. The front of the sensor is indicated by a heart shape figure, where you can place your finger on Kingbright's reverse-mounted green LED. On the reverse side of the sensor is Microchip's MCP6001 operational amplifier with the rest of the circuit that makes a RC filter network. The new version of the Pulse sensor has diode protection on the power line, so if the power leads are reversed by mistake, the sensor does not damage.

The Heartbeat rate information knowing is very useful while doing exercise, studying, etc. But, the heartbeat rate can be complicated to calculate. To overcome this problem, the pulse sensor or heartbeat sensor is used. This is a plug & play sensor mainly designed for Arduino board which can be used by makers, students, developers, artists who can utilize the heartbeat information into their projects. This sensor uses an easy optical pulse sensor along with amplification & cancellation of noise to make a circuit. By using this circuit, we can get fast and reliable heartbeat readings. This circuit can be operated with 4mA current and 5V voltage to use in mobile applications.

What is a Pulse sensor ?

A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume change is called a pulse sensor.

Here, a pulse signal is a variation within the blood level that happens when the heart forces the blood & a detector monitors the change in the blood volume. There are four methods to determine heart rate like a photoelectric pulse wave, electrocardiogram, phonocardiography & BP measurement but the pulse sensor uses the photoelectric technique.

Types of Pulse Sensor

Pulse sensors can be classified into two types based on the measurement technique of photoelectric pulse waves like transmission & reflection.

Transmission

Transmission type pulse sensor measures pulse signals by generating IR or red light from the surface of the human body. The signals can be generated by detecting the change in the flow of blood throughout heartbeats. This technique is restricted to some regions wherever light can simply go through like the earlobe or fingertip.

Reflection

The reflection-type pulse sensor which is used at present is ROHM. This kind of sensor is used to generate infrared, red color, or green color light toward the human body & measures the light which is reflected through a phototransistor or photodiode.

In the arteries of the human body, Oxygenated hemoglobin has the feature of absorbing light. So by detecting the rate of blood flow that changes heart tightening on time so that we can able to compute the pulse signal. The measurement of pulse wave signal using IR or red color light can be effected through IR rays limited in sunlight for avoiding stable operation. Due to this reason, semi indoors or indoors is suggested.

For measurement of the pulse wave outdoors, a green color light source includes a high rate of absorption within hemoglobin of blood. So ROHM type uses green color LEDs like a light source for transmission.

How Pulse Sensor Works

The pulse sensor working principle is very simple. This sensor has two surfaces, on the first surface, the light-emitting diode & ambient light sensor is connected. Similarly, on the second surface, the circuit is connected which is accountable for the noise cancellation& amplification.

If we talk about heartbeat rate, then heartbeat rate is the ratio of time between two consecutive heartbeats. Similarly, when the human blood is circulated in human body then this blood is squeezed in capillary tissues. As a result, the volume of capillary tissues is increased but this volume is decreased after each heartbeat. This change in volume of capillary tissues affects the LED light of heart rate pulse sensor, which transmits light after each heartbeat. This change in light is very small but this can be measured by connecting any controller with this pulse sensor. This means the LED light which has every pulse sensor helps for measuring pulse rate. The LED is located above a vein in a human body like ear tip or fingertip, however, it must be located on top of a layer directly. Once the LED is located on the vein, then the LED starts emitting light. Once the heart is pumping, then there will be a flow of blood within the veins. So if we check the blood flow, then we can check the heart rates also.

The working of this sensor could be checked by placing a human finger in front of this pulse sensor. When a finger is placed in front of this pulse sensor then the reflection of LED light is changed based on the volume of blood change inside capillary vessels. This means during the heartbeat the volume of blood in capillary vessels will be high and then will be low after each heartbeat. So, by changing this volume the LED light is changed. This change in of LED light measures the heartbeat rate of a finger. This phenomenon is known as “Photoplethysmogram.” If the blood flow is sensed then the ambient light sensor will receive more light as they will be reproduced by the flow of blood. This small change within obtained light can be examined over time to decide our pulse rates.

Hardware Overview

The front of the sensor is the side with the heart logo. This is where you place your finger. On the front side you will see a small round hole, from where the Kingbright’s reverse mounted green LED shines.

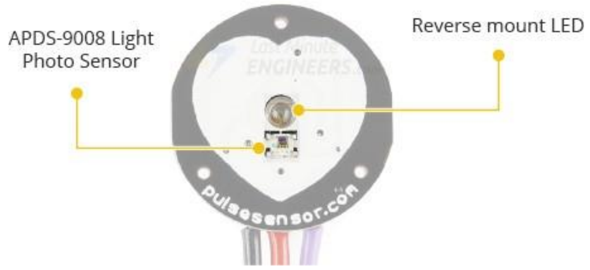


Fig 4.5.3 : Hardware Overview

Just below the LED is a small ambient light photo sensor – APDS-9008 from Avago, similar to that used in cell phones, tablets and laptops, to adjust the screen brightness in different light conditions.

On the back of the module you will find the rest of the components including a microchip’s MCP6001 Op-Amp and a bunch of resistors and capacitors that make up the R/C filter network. There is also a reverse protection diode to prevent damage if the power leads are accidentally reversed.

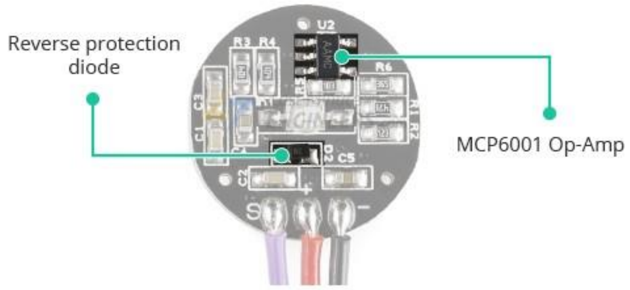


Fig 4.5.4 : Hardware Overview

The module operates from a 3.3 to 5V DC Voltage supply with a operating current of < 4mA

Interfacing Pulse Sensor with Arduino :

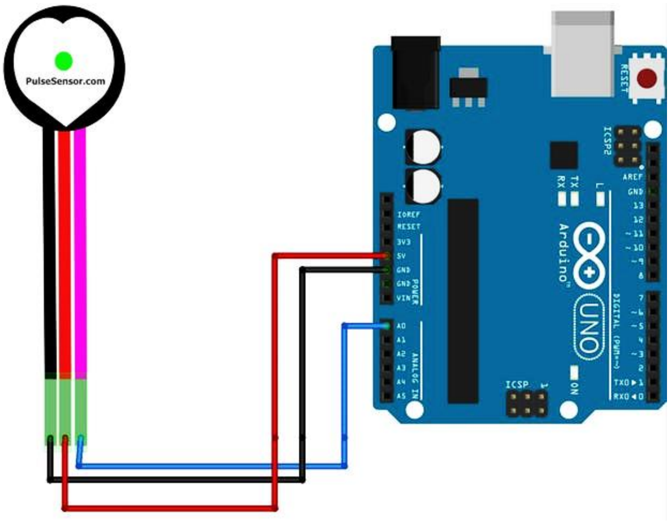


Fig 4.5.5 : Interfacing Pulse Sensor with Arduino

Connect the pulse sensor with Arduino as follows

- GND pin of pulse sensor to GND of Arduino
- VCC of pulse sensor to 5V of Arduino
- A0 of pulse sensor to A0 of Arduino

Testing the Circuit :

1. After hardware connection, insert the sample sketch into the Arduino IDE.

2. Using a USB cable, connect the ports from the Arduino to the computer.

3. Upload the program.

4. See the results in the serial monitor.

Applications :

The **pulse sensor applications** include the following.

- Sleep Tracking
- Monitoring of anxiety
- Alarm system
- Remote patient monitoring
- Health bands
- Advanced gaming consol

4.1.6 : IR Sensor/Infrared Sensor

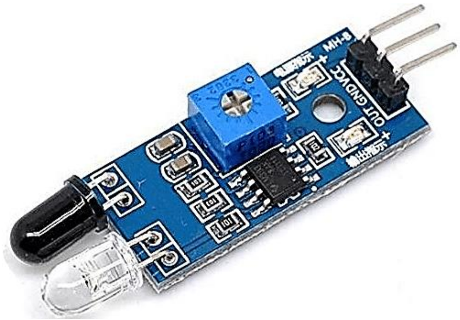


Fig. 4.6.1 : IR Sensor/Infrared Sensor

Features :

- 2 to 30 cm adjustable sensing range (on-board potentiometer).
- 35 degree detection angle.
- LM393 Comparator selection output.
- Easy to use.
- Sensitivity varies depending on the reflection surface applied.
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- Mounting hole

Specifications :

- Detection distance: 2-10cm
- Power supply: 3.3 to 5VDC
- 20mA supply current
- 3-wire output interface
- Digital output
- Signal: Active low
- Sensitivity (Adjustable)
- Size: 30mm x 14mm x 5mm
- FREE 3 pieces of jumper wires
- Range: Up to 20cm

Technical Parameters : (Table 4,6.1 : Technicle Parameters)

Attribute	Values
Sensor type	IR Infrared Sensor
Technology	Analog
Chip	LM393
Interface type	I2C
Mesurment Range Options	2 to 30 cm
Sensitivity	-
Max. Operating Supply Vol.	5 VDC
Min. Operating Supply Vol.	3.3 VDC
Package Type	Module
Dimension	32×14 mm
Max. Operating Temp.	+85°C
Min. Operating Temp.	-40 °C

Pin Configuration :

Fig. 4.6.2 : Pin Configuration

Pin Number	Pin Name	Description
1	Ground	Power Supply Ground
2	Vcc	Power Supply Input
3	OUT	Active High Output

Table. 4.6.2 : Pin Configuration

What is an IR Sensor/Infrared Sensor ?

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation.

These types of radiations are invisible to our eyes, which can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode that is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

Working Principle :

The working principle of an infrared sensor is similar to the object detection sensor. This sensor includes an IR LED & an IR Photodiode, so by combining these two can be formed as a photo-coupler otherwise optocoupler. The physics laws used in this sensor are planks radiation, Stephan Boltzmann & weins displacement.

IR LED is one kind of transmitter that emits IR radiations. This LED looks similar to a standard LED and the radiation which is generated by this is not visible to the human eye. Infrared receivers mainly detect the radiation using an infrared transmitter. These infrared receivers are available in photodiodes form. IR Photodiodes are dissimilar as compared with usual photodiodes because they detect simply IR radiation. Different kinds of infrared receivers mainly exist depending on the voltage, wavelength, package, etc.

Once it is used as the combination of an IR transmitter & receiver, then the receiver's wavelength must equal the transmitter. Here, the transmitter is IR LED whereas the receiver is IR photodiode. The infrared photodiode is responsive to the infrared light that is generated through an infrared LED. The resistance of photo-diode & the change in output voltage is in proportion to the infrared light obtained. This is the IR sensor's fundamental working principle.

Once the infrared transmitter generates emission, then it arrives at the object & some of the emission will reflect back toward the infrared receiver. The sensor output can be decided by the IR receiver depending on the intensity of the response.

Brief about IR Sensor Module

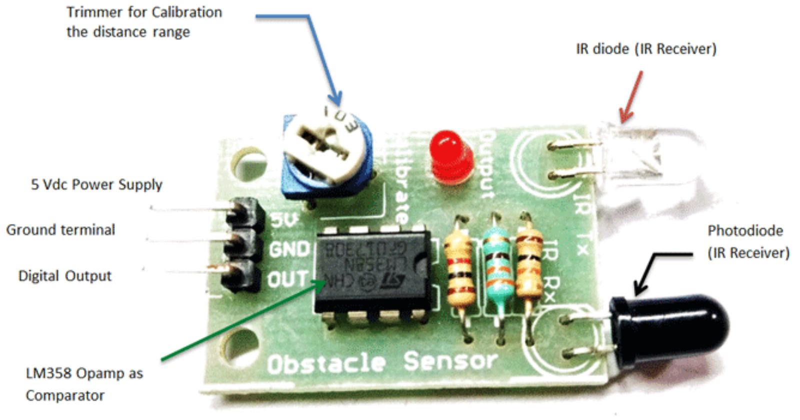


Fig. 4.6.3 : IR sensor Module Inside

The IR sensor module consists mainly of the IR Transmitter and Receiver, Op-amp, Variable Resistor (Trimmer pot), output LED along with few resistors.

IR LED Transmitter

IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimeters to several feets, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometers. IR LED white or transparent in colour, so it can give out amount of maximum light.

Photodiode Receiver

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black colour coating on its outer side, Black colour absorbs the highest amount of light.

LM358 Opamp

LM358 is an Operational Amplifier (Op-Amp) is used as voltage comparator in the IR sensor. the comparator will compare the threshold voltage set using the preset (pin2) and the photodiode's series resistor voltage (pin3).

Photodiode's series resistor voltage drop > Threshold voltage = Opamp output is High

Photodiode's series resistor voltage drop < Threshold voltage = Opamp output is Low

When Opamp's output **is** high the LED at the Opamp output terminal turns ON **(**Indicating the detection of Object).

Variable Resistor

The variable resistor used here is a preset. It is used to calibrate the distance range at which object should be detected.

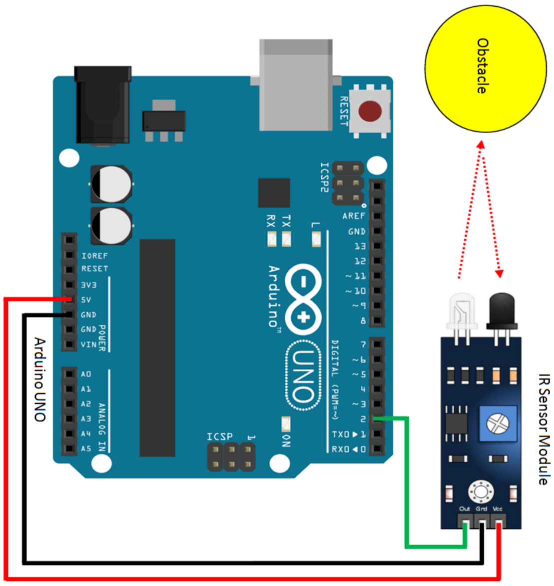


Fig. 4.6.4 : Interfacing IR Sensor with Arduino

Connecting the IR sensor to Arduino using Jumper wise as following:

- Vcc pin on the sensor to 5V pin on Arduino
- Gnd pin on the sensor to GND pin on Arduino
- Out pin on the sensor to any digital pin on Arduino UNO

Connecting the IR sensor to microcontroller is really simple. As we know this sensor outputs a digital signal and processing this signal is very easy. There exist two methods to do so first, you can always check the port in an infinite loop to see when the port changes its state from high to low, or the other way is to do it with an interrupt if you are making a complicated project the interrupt method is recommended. Power the IR with 5V or 3.3V and connect ground to ground. Then connect the output to a digital pin D2. We have just used a Male to Female Jumper wire to connect the IR sensor module with Arduino board as shown above.

Testing the Circuit :

1. After hardware connection, insert the sample sketch into the Arduino IDE.

2. Using a USB cable, connect the ports from the Arduino to the computer.

3. Upload the program.

4. See the results in the serial monitor.

Applications

- Obstacle Detection
- Industrial safety devices
- Wheel encode

4.1.7 : GSM – SIM900A

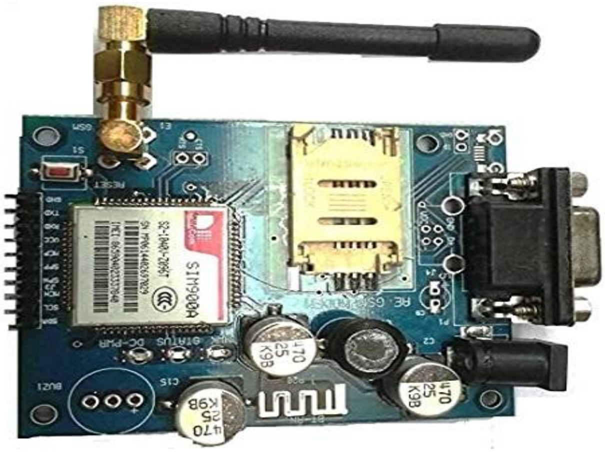


Fig. 4.7.1 : GSM – SIM900A

Features :

- Dual-Band GSM/GPRS 900/ 1800 MHz
- RS232 interface for direct communication with computer or MCU kit
- Configurable baud rate.
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enable with MIC and SPeaker socket.
- With slid in SIM card tray.
- With Stub antenna and SMA connector.
- Input Voltage: 12V DC.

Technical Parameters : Table 4.7.1. Technical Parameters

Parameters	Implementation
Power supply	Single supply voltage 3.4V – 4.5V
Power saving	Typical power consumption in SLEEP mode is 1.5mA (BS-PA-MFRMS=5)
Frequency Bands	• SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the 2 frequency bands automatically. The frequency bands also can be set by AT command. • Compliant to GSM Phase 2/2+
GSM class	Small MS
Transmitting power	• Class 4 (2W) at EGSM 900 • Class 1 (1W) at DCS 1800
GPRS connectivity	• GPRS multi-slot class 10 （default） • GPRS multi-slot class 8 (option) • GPRS mobile station class B
Temperature range	• Normal operation: -30°C to +80°C • Restricted operation: -40°C to -30°C and +80 °C to +85°C(1) • Storage temperature -45°C to +90°C
DATA GPRS:	• GPRS data downlink transfer: max. 85.6 kbps • GPRS data uplink transfer: max. 42.8 kbps

CSD:	• Coding scheme: CS-1, CS-2, CS-3 and CS-4 • Supports the protocols PAP (Password Authentication Protocol) usually used for PPP connections. • Integrates the TCP/IP protocol. • Support Packet Switched Broadcast Control Channel (PBCCH) • CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps, non-transparent • Unstructured Supplementary Services Data (USSD) support
SMS	• MT, MO, CB, Text and PDU mode • SMS storage: SIM card
FAX	Group 3 Class 1
SIM interface	Support SIM card: 1.8V, 3V
External antenna	Antenna pad
Audio features	Speech codec modes: • Half Rate (ETS 06.20) • Full Rate (ETS 06.10) • Enhanced Full Rate (ETS 06.50 / 06.60 / 06.80) • Adaptive multi rate (AMR) • Echo Cancellation • Noise Suppression
Serial port and Debug port	Serial Port: • 8-wire modem interface with status and control lines, unbalanced, asynchronous. • 1.2kbps to 115.2kbps. • Serial Port can be used for AT commands or data stream. • Supports RTS/CTS hardware handshake and software ON/OFF flow control. • Multiplex ability according to GSM 07.10 Multiplexer Protocol. • Autobauding supports baud rate from 1200 bps to 115200bps. Debug port: • 2-wire null modem interface DBG_TXD and DBG_RXD. • Can be used for debugging and upgrading firmware.
Phonebook management	Support phonebook types: SM, FD, LD, RC, ON, MC.
SIM Application Toolkit	Support SAT class 3, GSM 11.14 Release 99
Real time clock	Implemented
Timer function	Programmable via AT command
Physical characteristics	Size: 24mm x 24mm x 3mm Weight: 3.4g
Firmware upgrade	Firmware upgrade by debug port.

SIM900A Modem is built with Dual Band GSM based SIM900A modem from SIMCOM. It works on frequencies 900MHz. SIM900A can search these two bands automatically. The frequency bands can also be set by AT Commands. The baud rate is configurable from 1200-115200 through AT command. SIM900A is an ultra compact and wireless module. The Modem is coming interface, which allows you connect PC as well as microcontroller with RS232 Chip(MAX232). It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The onboard Regulated Power supply allows you to connect wide range unregulated power supply. Using this modem, you can make audio calls, SMS, Read SMS, attend the incoming calls and ect. through simple AT commands. This is a complete GSM module in a SMT type and made with a very powerful single-chip, allowing you to benefit from small dimensions. SIM 900A GSM Modem with serial and TTL outputs.

GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with RS232 interface, which allows you connect PC as well as microcontroller with RS232 Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The onboard Regulated Power supply allows you to connect wide range unregulated power supply . Using this modem, you can make audio calls, SMS, Read SMS, attend the incoming calls and internet through simple AT commands

The GPRS/GSM Shield provides you a way to use the GSM cell phone network to receive data from a remote location. The shield allows you to achieve this via any of the three methods: • Short Message Service • Audio • GPRS Service The GPRS Shield is compatible with all boards which have the same form factor (and pinout) as a standard Arduino Board. The GPRS Shield is configured and controlled via its UART using simple AT commands. Based on the SIM900 module from SIMCOM, the GPRS Shield is like a cell phone. Besides the communications features, the GPRS Shield has 12 GPIOs, 2 PWMs and an ADC.

SIM900A Functional Diagram :

The following figure shows a functional diagram of the SIM900A and illustrates the mainly functional part :

- The GSM baseband engine.
- Flash and SRAM.
- The GSM radio frequency part.
- The antenna interface.
- The other interfaces.

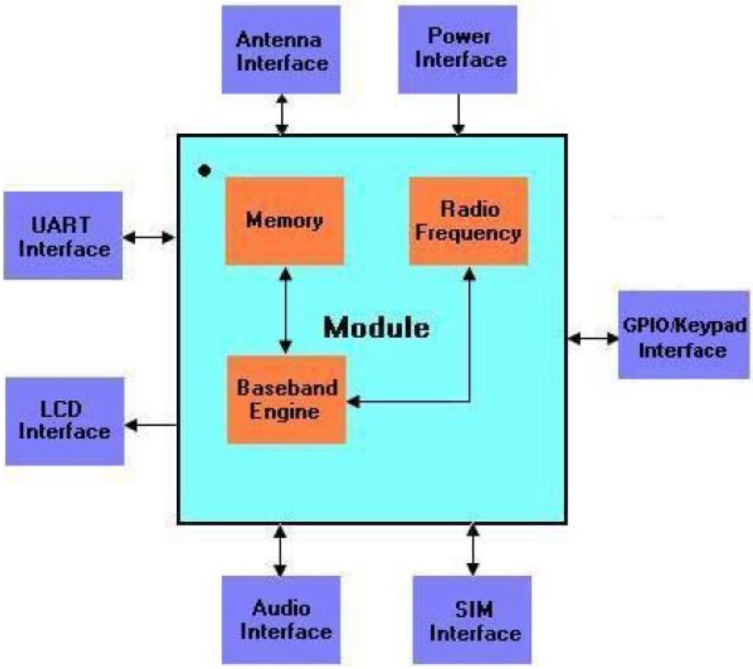


Fig. 4.7.2 : SIM900A Functional Diagram

Hardware overview :

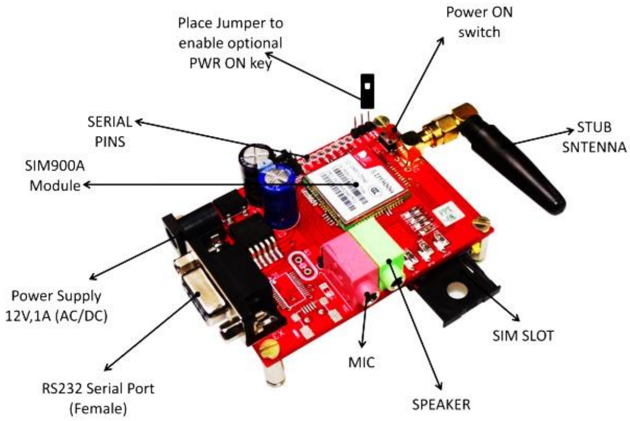


Fig. 4.7.3 : SIM900A Module

The indicator LEDs

The GSM Shield has three indicator LEDs for the GSM Shield power, SIM900 power and net status.

GSM Shield power(P):

This LED is used to indicate the power status of the GSM Shield.If the external power supply is connected to the arduino board ,then the GSM Shield will get power,this LED will light on.

SIM900 power(S):

This LED is used to indicate the power status of the SIM900.After the SIM900 is power on,the status LED will light on.

Net Status(N):

This LED is used to indicate the net status.The LED will blink slowly or quickly according to different states.

Supplying Power for SIM900 Shield

One of the most important parts of getting the SIM900 shield working is supplying it with enough power. Depending on which state it's in, the SIM900 can be a relatively power-hungry device. The maximum current draw of the chip is around 2A during transmission burst. It usually won't pull that much, but may require around 216mA during phone calls or 80mA during network transmissions. This chart from the datasheet summarizes what you may expect:

Table 4.7.2 Power Supplying Modes

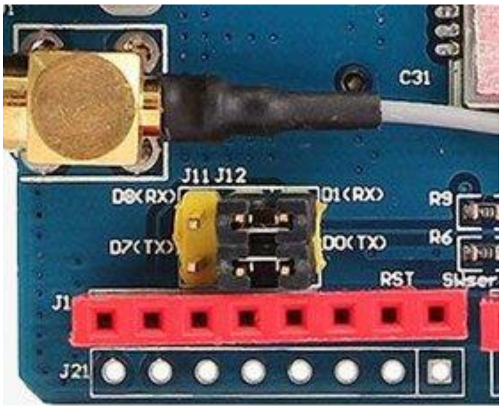
Modes	Frequency	Current Consumption
Power down		60 uA
Sleep mode		1 mA
Stand by		18 mA
Call	GSM850 EGSM900 DCS1800 PCS1900	199 mA 216 mA 146 mA 131 mA
GPRS		453 mA
Transmission burst		2 A

The operating voltage of SIM900 chip is from 3.4V to 4.4V. To keep supply voltage safe at 4.1V, the shield comes with a high current, high accuracy, low-dropout voltage regulator MIC29302WU from Micrel – capable of handling load currents up to 3A.

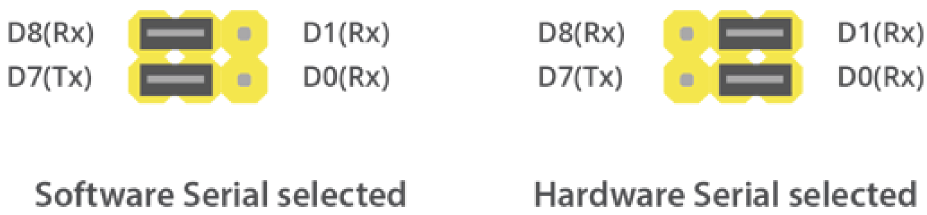
You can add an external power supply to the shield with the 5.5mm DC jack, to which you can connect any 5V-9V DC wall adapter you have. Next to the DC jack, is a Slide Switch to select the power source labeled EXTERN. To use external power source, move the slider as shown above.

UART Communication

The SIM900 GSM/GPRS shield uses UART protocol to communicate with an Arduino. The chip supports baud rate from 1200bps to 115200bps with Auto-Baud detection.

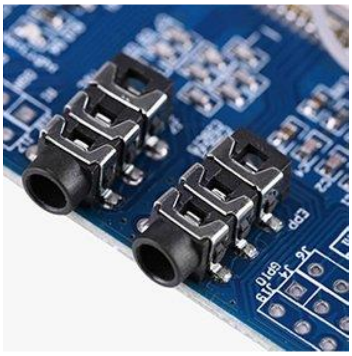


With the help of jumpers you can connect (RX,TX) of the shield to either Software Serial(D8,D7) or Hardware Serial(D1,D0) of the Arduino.



Speaker & Microphone

The shield comes with two standard 3.5mm jacks. One for stereo earphone and other for mono microphone. It allows you to use SIM900's audio interface to make and receive voice calls and listen FM radio.



Mic*:* You can connect an external electret microphone to this jack.

Earphone: You can connect earphones to this jack. Any ‘iPhone’ or ‘Android’ compatible earphones should work.

Antenna

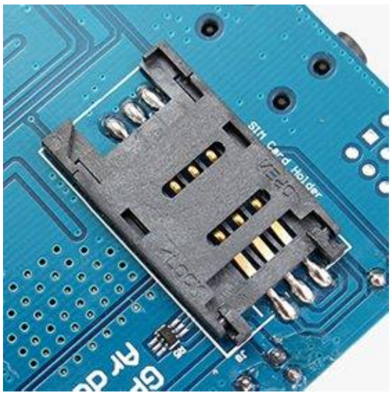
An antenna is required to use the SIM900 for any kind of voice or data communications as well as some SIM commands.

The shield has two interfaces for connecting antenna viz. a U.FL connector and a SMA connector. They are connected through a patch cord.

The shield usually comes with a 3dBi GSM antenna and allows you to put the shield inside a metal case(as long the antenna is outside).

SIM Socket

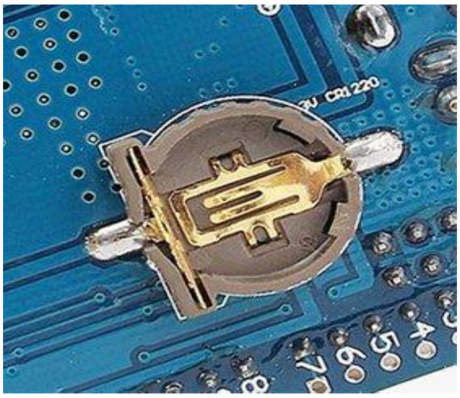
There’s a SIM socket on the back. Any activated, 2G full-size SIM card would work perfectly.



The workings of the SIM card socket can take some getting used to. To unlock the latch, push the top part of the assembly, and then lift it up. Place the SIM card into the bottom part of the socket. Then fold the arm back into the body of the socket, and gently push it forward towards the LOCK position.

RTC(Real Time Clock)

The SIM900 shield can be configured to keep time. So there is no need for any separate RTC. This will keep the time even when the power is OFF.



If you want to use internal RTC, you need to install CR1220 battery at the back side of the shield.

Interfacing Uno And Gsm Shield :

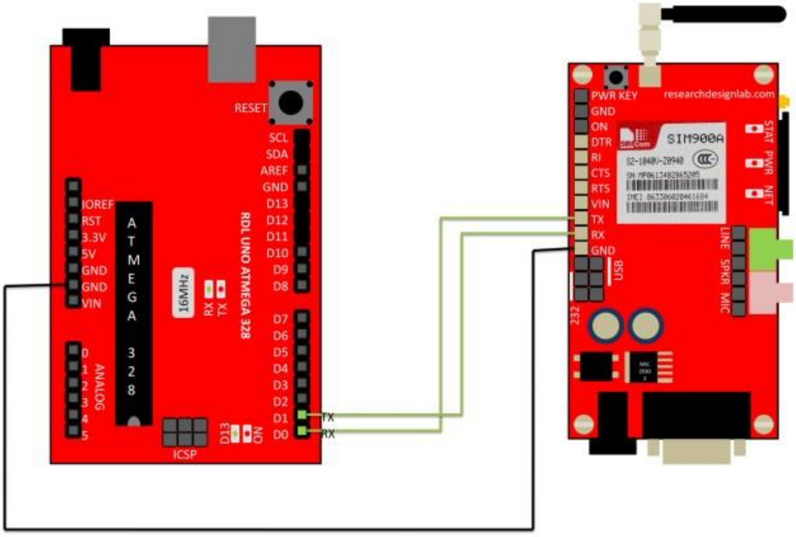


Fig. 4.7.5 : Interfacing Uno And Gsm Shield

Connecting the GSM to Arduino using Jumper wise as following:

- Connect TX pin of GSM Module to RX pin of Arduino Uno.
- Connect RX pin of GSM Module to TX pin of Arduino Uno.
- Connect GND pin of GSM Module to GND pin of Arduino Uno.

There are two ways of connecting GSM module to Arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. You read it right? GSM Tx –> Arduino Rx and GSM Rx –> Arduino Tx. Now connect the ground pin of Arduino to ground pin of gsm module! So that's all! You made 3 connections and the wiring is over! Now you can load different programs to communicate with gsm module and make it work.

Note:- The problem with this connection is that, while programming Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will not be loaded successfully to Arduino. So you have to disconnect wiring in Rx and Tx each time you burn the program to Arduino. Once the program is loaded successfully, you can reconnect these pins and have the system working!

To avoid this difficulty, I am using an alternate method in which two digital pins of Arduino are used for serial communication. We need to select two PWM enabled pins of Arduino for this method. So I choose pins 9 and 10 (which are PWM enabled pins). This method is made possible with the SoftwareSerial Library of Arduino. Software Serial is a library of Arduino which enables serial data communication through other digital pins of Arduino. The library replicates hardware functions and handles the task of serial communication.

Steps to boot GSM module

1. Power ON the GSM module by providing 5V and GND.
2. Insert the SIM card to GSM module and lock it.
3. Initially blinking rate of network LED will be high. After sometime observe the blinking rate of 'network LED' (GSM module will take some time to establish connection with mobile network)
4. Once the connection is established successfully, the network LED will blink continuously for every 3 seconds.
5. Even we can check the connection establishment of GSM module with mobile by making a call to the number of the SIM. If we hear a ring back, the GSM module has successfully established network connection.

Applications

- Cellular Communication
- Home Automations
- Security based projects
- Sensors monitoring
- Robotics
- Mobile Phone Accessories
- Servers
- Computer Peripherals
- Automobile
- USB Dongles

4.1.8 : GPS : SKG17A1

Fig 4.8.1 : GPS : SKG17A1

Features :

- Ultra high sensitivity: -160dBm
- Extremely fast TTFF at low signal level
- Built in high gain LNA
- Low power consumption: Typical 40mA@3.0V
- NMEA-0183 compliant protocol or custom protocol
- Operating voltage: 2.8V to 3.3V
- Operating voltage: 2.8V to 3.3V
- SMD type with stamp holes
- Small form factor: 22.9x17.0x2.8mm
- RoHS compliant (Lead free)

Specification : Table 4.8.1 : Specification

Parameter	Specification	
Receiver Type	L1 frequency band, C/A code, 20-channels	
Sensitivity	Tracking Acquisition	-160dBm -144dBm
Accuracy	Position Velocity Timing (PPS)	3.0m CEP50 without SA(Typical Open Sky) 0.1m/s without SA 60ns RMS
Acquisition Time	Cold Start Warm Start Hot Start Re-Acquisition	36s (Typical Open Sky) 30s 2s <1s
Power Consumption	Tracking Acquisition	35mA @3V Vcc (Typical) 35mA
Navigation Data Update Rate	1Hz	
Operational Limits	Altitude Velocity Acceleration	Max 18,000m Max 515m/s Less than 4g

Technical Parameters : Table 4.8.2 : Technical Parameters

Parameter	Symbol	Condition	Min	Typ	Max	Units
Power Supply Voltage	Vcc		2.7	3.0	3.6	V
Power Supply voltage ripple	Vcc_PP	Vcc = 3.0V			30	mV
Consumption current	Icc	Vcc = 3.0V		40	45	mA
Input Higgh Voltage	V_{IH}		0.7 × Vcc		Vcc	V
Input Low Voltage	V_{IL}		-0.3		0.3 × Vcc	V
Output High Voltage	V_{OH}		0.8 × Vcc		Vcc	V
Output Low Voltage	V_{OL}		0		0.3 × Vcc	V
Operating Temperature	Topr		-40		85	°C

Electrical Characteristics : Table 4.8.3 : Electrical Characteristics

Parameter	Symbol	Min	Max	Units
Power Supply				
Power Supply Volt	VCC	-0.3	3.6	V
Input Pins				
Input Pin Voltage I/O	Reset	-0.3	3.6	V
Input Pin Voltage I/O	RxD0,RxD1	-0.3	3.6	V
Input Pin Voltage I/O	BOOT	-0.3	3.6	V
Antenna Bias DC Vol	V_ANT	-0.3	5.0	V
Backup Battery	V_BCKP	2.0	3.6	V
Environment				
Storage temp.	Tstg	-40	125	°C
Peak Reflow Soldering Temp.	Tpeak		260	°C
Humidity			95	%

Pin Configuration :

- GND is the Ground Pin and needs to be connected to GND pin on the Arduino.
- TxD (Transmitter) pin is used for serial communication.
- RxD (Receiver) pin is used for serial communication.
- VCC supplies power for the module. You can directly connect it to the 5V pin on the Arduino.

What is GPS :

Global **P**ositioning **S**ystem (GPS) is a satellite-based system that uses satellites and ground stations to measure and compute its position on Earth.

GPS is also known as Navigation System with Time and Ranging (NAVSTAR) GPS.

GPS receiver needs to receive data from at least 4 satellites for accuracy purpose. GPS receiver does not transmit any information to the satellites. This GPS receiver is used in many applications like smartphones, Cabs, Fleet management etc.

How GPS works :

GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it.

To calculate your 2-D position (latitude and longitude) and track movement, a GPS receiver must be locked on to the signal of at least 3 satellites. With 4 or more satellites in view, the receiver can determine your 3-D position (latitude, longitude and altitude). Generally, a GPS receiver will track 8 or more satellites, but that depends on the time of day and where you are on the earth.

Once your position has been determined, the GPS unit can calculate other information, such as:

- Speed
- Bearing
- Track
- Trip dist
- Distance to destination

Overview of GPS Module

The Skylab SKG17A1 is a complete GPS module that features super sensitivity, ultra low power and small form factor. The GPS signal is applied to the antenna input of module, and a complete serial data message with position, velocity and time information is presented at the serial interface with NMEA protocol or customprotocol.

Its 160dBm tracking sensitivity extends positioning coverage into place like urban canyons and dense foliage environment where the GPS was not possible before. The small form factor and low power consumption make the module easy to integrate into portable device like PNDs, mobile phones, cameras and vehicle navigation systems

Interfaces Configuration :

Power Supply : Regulated power for the SKG17A1 is required. The input voltage Vcc should be 3.0V ±10%, maximum, current is no less than 100mA. Suitable decoupling must be provided by external decoupling circuitry. It can reduce the Noise from power supply and increase power stability.

Antenna : The SKG17A1 GPS receiver is designed for supporting the active antenna or passive antenna connected with pin RF_IN. The gain of active antenna should be no more than 20dB. The maximum noise figure should be no more than 1.5dB and output impedance is at 50 Ohm.

UART Ports : The module supports two full duplex serial channels UART0 and UART1. All serial connections are at 3V CMOS logic levels, if need different voltage levels, use appropriate level shifters. The baud rate of both serial ports are fully programmable, the data format is however fixed: X, N, 8, 1, i.e. X baud rate, no parity, eight data bits and one stop bit, no other data formats are supported, LSB is sent first. The modules default baud rate is set up 4800bps, however, he user can change the default baud rate to any value from 4800 bps to 115kbps. UART0 is used e.g. for booting and NMEA interface. UART1 can be utilized for UBP protocol.

Boot Mode Select : The pin Bootis used to set the boot mode of the SKG17A1 GPS Receiver. By default the receiver will boot in normal GPS mode. If there are corrupted data in FLASH, it may be necessary to boot the receiver in test mode by pulling Boot pinhigh during a power cycle or hardware reset to update the firmware**.**

Backup Battery Power : In case of a power failure on pin Vcc, real-time clock and backup RAM are supplied through pin V_BAT. This enables the SKG17A1 GPS Receiver to recover from power failure with either a hot start or a warm start (depending on the duration of Vcc outage). If no Backup Battery is connected, the receiver performs a cold start upon powered up.

Interfacing GPS Sensor with Arduino :

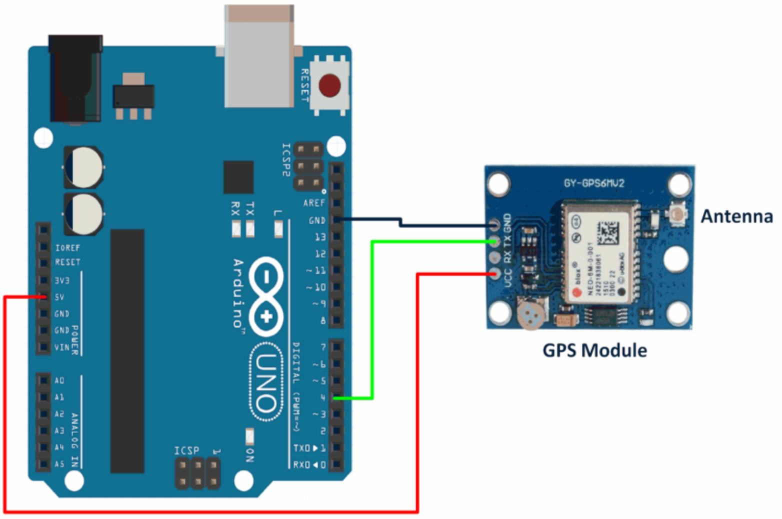


Fig 4.8.2: Interfacing GPS Sensor with

Connecting the GPS sensor to Arduino using Jumper wise as following:

- Vcc pin on the sensor to 5V pin on Arduino
- Gnd pin on the sensor to GND pin on Arduino
- Rx pin on the sensor to any digital pin on Arduino UNO

Testing the Circuit :

1. After hardware connection, insert the sample sketch into the Arduino IDE.
2. Using a USB cable, connect the ports from the Arduino to the computer.
3. Upload the program.
4. See the results in the serial monitor.

Applications :

- LBS (Location Based Service)
- PND (Portable Navigation Device)
- Vehicle navigation system.

4.1.9 : XBee s2c

Fig. 4.9.1 : XBee Pro S2C

Performance : Table 4.9.1

TRANSCEIVER CHIPSET	Silicon Labs EM357 SoC
DATA RATE	RF 250 Kbps, Serial up to 1 Mbps
INDOOR/URBAN RANGE	300 ft (90 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	2 miles (3200 m)
TRANSMIT POWER	63 mW (+18 dBm)
RECEIVER SENSITIVITY (1% PER)	-101 dBm

Features : Table 4.9.2

SERIAL DATA INTERFACE	UART, SPI
CONFIGURATION METHOD	API or AT commands, local or over-the-air (OTA
FREQUENCY BAND	ISM 2.4 GHz
FORM FACTOR	Through-Hole, Surface Mount
HARDWARE	S2C
ADC INPUTS	(4) 10-bit ADC inputs
DIGITAL I/O	15
ANTENNA OPTIONS	Through-Hole: PCB Antenna, U.FL Connector, RPSMA Connector, or Integrated Wire SMT: RF Pad, PCB Antenna, or U.FL Connector
OPERATING TEMPERATURE -	-40º C to +85º C
DIMENSIONS (L X W X H) AND WEIGHT	Through-Hole: 0.960 x 1.297 in (2.438 x 3.294 cm) SMT: 0.866 x 1.33 x 0.120 in (2.199 x 3.4 x 0.305 cm)

Networking and Security : Table 4.9.3

PROTOCOL	XBee 802.15.4 (Proprietary 802.15.4)
UPDATABLE TO DIGIMESH PROTOCOL	Yes
UPDATABLE TO ZIGBEE PROTOCOL	Yes
INTERFERENCE IMMUNITY ENCRYPTION	DSSS (Direct Sequence Spread Spectrum)
ENCRYPTION	128-bit AES
RELIABLE PACKET DELIVERY	Retries/Acknowledgements
IDS	PAN ID and addresses, cluster IDs and endpoints (optional)
CHANNELS	16 channels

Power Requirements : Table 4.9.4

SUPPLY VOLTAGE	2.7 to 3.6V
TRANSMIT CURRENT	120 mA @ 3.3 VDC
RECEIVE CURRENT	31 mA @ 3.3 VDC
POWER-DOWN CURRENT	<1μA@25°C

Regulatory Approvals : Table 4.9.5

FCC, IC (NORTH AMERICA) Y	Yes
ETSI (EUROPE)	No
RCM (AUSTRALIA AND NEW ZEALAND)	No (Coming soon)
TELEC (JAPAN)	No (Coming soon)

XBee Module Pin Description :

Fig. 4.9.2 : XBee Pin Configuration

As shown in above figure of XBee module, it has 20 pins. Each pin function is described in table given below.

Table 4.9.6 Pin Functions

Pin No.	Name	Direction	Description
1	VCC	-	Power Supply
2	DOUT	Output	UART Data out
3	DIN/CONFIG	Input	UART Data in
4	DO8	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse >=200nS)
6	PWM0/RSSI	Output	PWM Output 0/ Received Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	Reserved	-	Do not connect
9	DTR/SLEEP_RQ/DI8	Input	Sleep Control or Digital Input 8
10	GND	-	Ground
11	AD4/DIO4	Input / Output	Analog Input 4/ Digital I/O 4
12	CTS/DIO7	Input / Output	Clear-To-Send Flow Control or Digital I/O 7
13	ON/SLEEP	Output	Module Status Indicator, High = ON, Low = SLEEP
14	VREF	Input	Reference Voltage for ADC
15	ASSOCIATE/AD5/DIO5	Input / Output	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS/AD6/DIO6	Input / Output	Request-To-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3/DIO3	Input / Output	Analog Input 3 or Digital I/O 3
18	AD2/DIO2	Input / Output	Analog Input 2 or Digital I/O 2
19	AD1/DIO1	Input / Output	Analog Input 1 or Digital I/O 1
20	AD0/DIO0	Input / Output	Analog Input 0 or Digital I/O 0

In this present communication world, there are numerous high data rate communication standards that are available, but none of these meet the sensors' and control devices' communication standards. These high-data-rate communication standards require low-latency and low-energy consumption even at lower bandwidths. The available proprietary wireless systems' Zigbee technology is low-cost and low-power consumption and its excellent and superb characteristics make this communication best suited for several embedded applications, industrial control, and home automation, and so on. The Zigbee technology range for transmission distances mainly ranges from 10 – 100 meters based on the output of power as well as environmental characteristics.

What is Zigbee Technology ?

Zigbee communication is specially built for control and sensor networks on IEEE 802.15.4 standard for wireless personal area networks (WPANs), and it is the product from Zigbee alliance. This communication standard defines physical and Media Access Control (MAC) layers to handle many devices at low-data rates. These Zigbee's WPANs operate at 868 MHz, 902-928MHz, and 2.4 GHz frequencies. The data rate of 250 kbps is best suited for periodic as well as intermediate two-way transmission of data between sensors and controllers.

Zigbee is a low-cost and low-powered mesh network widely deployed for controlling and monitoring applications where it covers 10-100 meters within the range. This communication system is less expensive and simpler than the other proprietary short-range wireless sensor networks as Bluetooth and Wi-Fi.

Zigbee supports different network configurations for the master to master or master to slave communications. And also, it can be operated in different modes as a result the battery power is conserved. Zigbee networks are extendable with the use of routers and allow many nodes to interconnect with each other for building a wider area network.

History of Zigbee Technology

In the year 1990, the digital radio networks with self-organizing ad hoc were implemented. The Zigbee specification like IEEE 802.15.4-2003 was approved in the year 2004, on December 14. The Specification 1.0 was announced by Zigbee Alliance in the year 2005, on June 13, called the Specification of ZigBee 2004.

Cluster Library

In the year 2006, September, the Specification of Zigbee 2006 was announced by replacing the 2004 stack. So this specification mainly replaces the pair structure of key-value as well as message utilized within the 2004 stack through a cluster library.

A library includes a set of consistent commands, planned beneath groups called clusters with names like Home Automation, Smart Energy & Light Link of ZigBee. In the year 2017, the library was renamed with Dotdot by Zigbee Alliance and announced as a new protocol. So, this Dotdot has worked for approximately all Zigbee devices as the default application layer.

Zigbee Pro

In the year 2007, Zigbee Pro like Zigbee 2007 was finalized. It is one kind of device which operates on a legacy Zigbee network. Because of the disparities within the options of routing, these devices should turn into non-routing ZEDs or Zigbee end devices (ZEDs) on a legacy Zigbee network. The legacy Zigbee devices have to turn into Zigbee end devices on a network of Zigbee Pro. It functions through the 2.4 GHz ISM band as well as includes a sub-GHz band.

How does Zigbee Technology Work?

Zigbee technology works with digital radios by allowing different devices to converse through one another. The devices used in this network are a router, coordinator as well as end devices. The main function of these devices is to deliver the instructions and messages from the coordinator to the single end devices such as a light bulb.

I An this network, the coordinator is the most essential device which is placed at the origin of the system. For each network, there is simply one coordinator, used to perform different tasks. They choose a suitable channel to scan a channel as well as to find the most appropriate one through the minimum of interference, allocate an exclusive ID as well as an address to every device within the network so that messages otherwise instructions can be transferred in the network.

Routers are arranged among the coordinator as well as end devices which are accountable for messages routing among the various nodes. Routers get messages from the coordinator and stored them until their end devices are in a situation to get them. These can also permit other end devices as well as routers to connect the network;

In this network, the small information can be controlled by end devices by communicating with the parent node like a router or the coordinator based on the Zigbee network type. End devices don’t converse directly through each other. First, all traffic can be routed toward the parent node like the router, which holds this data until the device’s receiving end is in a situation to get it through being aware. End devices are used to request any messages that are waiting from the parent.

Zigbee Operating Modes and Its Topologies

Zigbee two-way data is transferred in two modes: Non-beacon mode and Beacon mode. In a beacon mode, the coordinators and routers continuously monitor the active state of incoming data hence more power is consumed. In this mode, the routers and coordinators do not sleep because at any time any node can wake up and communicate.

However, it requires more power supply and its overall power consumption is low because most of the devices are in an inactive state for over long periods in the network. In a beacon mode, when there is no data communication from end devices, then the routers and coordinators enter into a sleep state. Periodically this coordinator wakes up and transmits the beacons to the routers in the network.

These beacon networks are work for time slots which means, they operate when the communication needed results in lower duty cycles and longer battery usage. These beacon and non-beacon modes of Zigbee can manage periodic (sensors data), intermittent (Light switches), and repetitive data types.

Zigbee Networks Topologies :

There are three main types of Zigbee network topologies; star, mesh, and tree. A topology simply refers to the layout and arrangement of different elements within a communication network.

Star Topology

The star topology is both the simplest and least expensive type of Zigbee to implement. They don't consist of any routers and all of the end devices communicate directly with the coordinator.

The biggest problem with this setup is that if the coordinator fails, the whole network will come crashing down and none of the end devices will work given that there is no one there to give them their instructions.

Star networks are also limited by the range of the coordinator itself and therefore really only suitable for the smallest of networks that consist of just a couple of devices.

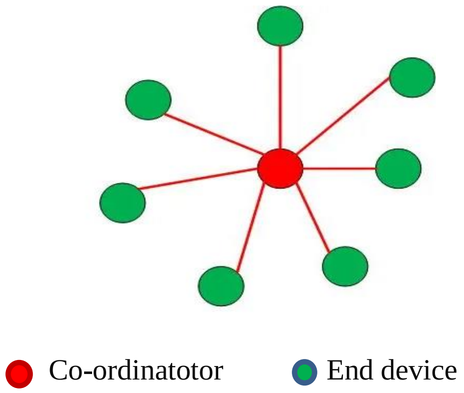


Fig. 4.9.3 : Star Topology

Mesh Topology

In the mesh topology, each node is connected to a neighboring node, except the end devices who are only connected with their parent, a router.

The coordinator connects to the nearby routers who then connect to the other routers closest to them. So, some routers may connect to the coordinator and one other router, whereas others may be connected to three or four different routers depending on the size of the network.

When one device wants to communicate with another, the message hops from one device to another, taking the shortest possible route, until it reaches its destination.

Should a particular node fail, the network will continue to function as the message can simply be re-routed by following a different path. This is referred to as the "self-healing process".

The mesh topology is well suited for smart homes as they will usually have more devices than a star topology is really suitable for.

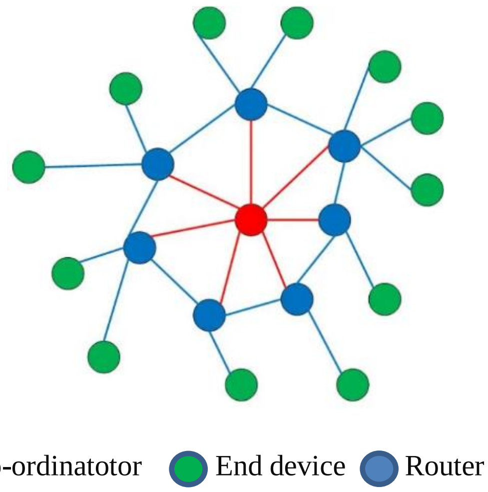


Fig. 4.9.4 : Mesh Topology

Tree Topology

The tree topology is very similar to the mesh topology with the only real difference being that the routers are not interconnected.

The coordinator connects to all of the nearby routers and the routers connect with the nearby end devices, but that is the extent of the connections. The routers don't associate themselves with each other and only communicate with the coordinator and the end devices.

There is another type of setup that is referred to as a cluster tree network that does allow the routers to connect with each other which expands on the range of the network.

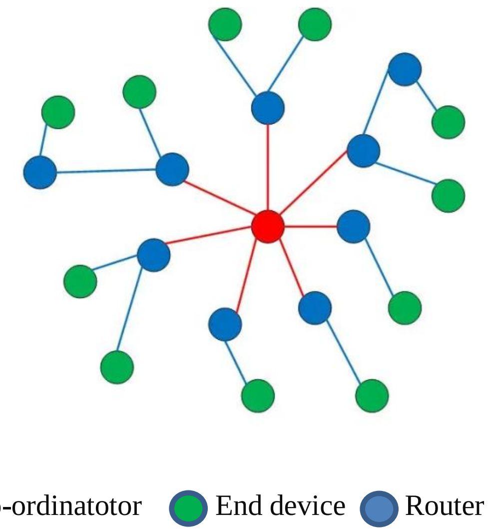


Fig. 4.9.5 : Tree Topology

Zigbee Architecture :

Zigbee system structure consists of three different types of devices as Zigbee Coordinator, Router, and End device.

1. **ZigBee Co-ordinator (ZC)**: This is the most important device as it forms the root of the network tree and helps to bridge to other networks. This means that you will find one ZigBee coordinator in each network as this is the device responsible for the start of the network. This device contains all the information of the network and functions as a Trust Center & repository for the security keys.

– Always first coordinator need to be installed for establishing zigbee network service, it starts a new PAN (Personal Area Network), once started other zigbee components viz. router(R) and End devices(E) can join the network(PAN).

– It is responsible for selecting the channel and PAN ID.

– It can assist in routing the data through the mesh network and allows join request from R and E.

– It is mains powered (AC) and support child devices.

– It will not go to sleep mode.

2. **ZigBee Router (ZR):** In addition to running an application function, its is used to route the data from other devices and help it reach the destination.

   – First router needs to join the network then it can allow other R & E to join the PAN.

   – It is mains powered (AC) and support child devices.

   – It will not go to sleep mode.

3. **ZigBee End-Device (ZED):** The end device contains just enough functionality to talk to either the co-ordinator or the router. Note that it cannot rely data from other devices. This causes the node to stay asleep for a significant time thereby increasing battery life to a considerable extent. A ZED device requires least amount of energy as compared to the ZC or ZR.

   – This may sleep hence battery consumption can be minimized to great extent. There are two topologies, star and mesh, as mentioned Zigbee supports mesh routing. PAN ID is used to communicate between zigbee devices, it is 16 bit number. Coordinator will have PAN ID set to zero always and all other devices will receive a 16 bit address when they join PAN. There are two main steps in completing Zigbee Network Installation. Forming the network by Coordinator and joining the network by Routers and End devices.

Zigbee protocol architecture consists of a stack of various layers where IEEE 802.15.4 is defined by physical and MAC layers while this protocol is completed by accumulating Zigbee's own network and application layers.

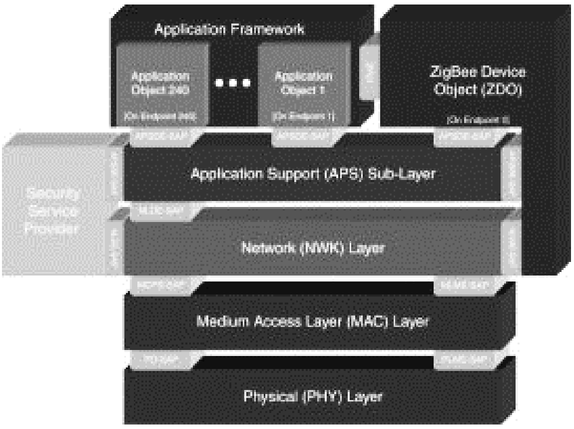


Fig. 4.9.6 :Natework Layers

Physical Layer: This layer does modulation and demodulation operations upon transmitting and receiving signals respectively. This layer's frequency, data rate, and a number of channels are given below.

MAC Layer: This layer is responsible for reliable transmission of data by accessing different networks with the carrier sense multiple access collision avoidances (CSMA). This also transmits the beacon frames for synchronizing communication.

Network Layer: This layer takes care of all network-related operations such as network setup, end device connection, and disconnection to network, routing, device configurations, etc.

Application Support Sub-Layer: This layer enables the services necessary for Zigbee device objects and application objects to interface with the network layers for data managing services. This layer is responsible for matching two devices according to their services and needs.

Application Framework: It provides two types of data services as key-value pair and generic message services. The generic message is a developer-defined structure, whereas the key-value pair is used for getting attributes within the application objects. ZDO provides an interface between application objects and the APS layer in Zigbee devices. It is responsible for detecting, initiating, and binding other devices to the network.

Interfacing XBee with Arduino :

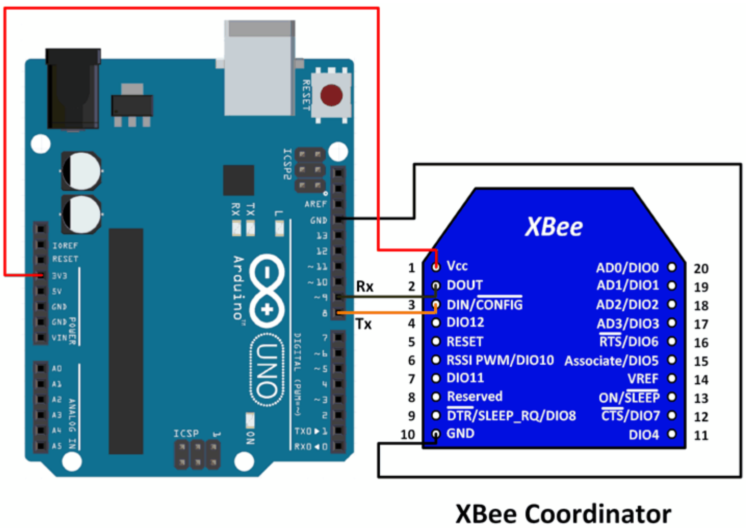


Fig. 4.9.7: Interfacing XBee with Arduino

Connections for interfacing ZigBee module with Arduino are shown in the circuit diagram.

1. Tx (pin2)of XBee -> Tx of Arduino board
2. Rx(pin3) of Xbee -> Rx of Arduino board
3. Gnd(pin10) of Xbee -> GND of Arduino board
4. Vcc (Pin1) of Xbee -> 3.3v of Arduino board

Zigbee Technology Advantages and Disadvantages

The **advantages** of Zigbee include the following.

- Battery life is good.
- Power consumption is less
- Very simple to fix.
- It supports approximately 6500 nodes.
- Less cost.
- It is self-healing as well as more reliable.
- Network setting is very easy as well as simple.
- Loads are evenly distributed across the network because it doesn't include a central controller
- Home appliances monitoring as well controlling is extremely simple using remote
- The network is scalable and it is easy to add/remote ZigBee end device to the network.

The **disadvantages** of Zigbee include the following.

- It needs the system information to control Zigbee based devices for the owner.
- As compared with WiFi, it is not secure.
- The high replacement cost once any issue happens within Zigbee based home appliances
- The transmission rate of the Zigbee is less
- It does not include several end devices.
- It is so highly risky to be used for official private information.
- It is not used as an outdoor wireless communication system because it has less coverage limit.
- Similar to other types of wireless systems, this ZigBee communication system is prone to bother from unauthorized people.

Applications of Zigbee Technology

The applications of ZigBee technology include the following.

- **Industrial Automation:** In manufacturing and production industries, a communication link continually monitors various parameters and critical equipment. Hence Zigbee considerably reduces this communication cost as well as optimizes the control process for greater reliability.
- **Home Automation:** Zigbee is perfectly suited for controlling home appliances remotely as a lighting system control, appliance control, heating, and cooling system control, safety equipment operations and control, surveillance, and so on.
- **Smart Metering:** Zigbee remote operations in smart metering include energy consumption response, pricing support, security over power theft, etc.
- **Smart Grid monitoring:** Zigbee operations in this smart grid involve remote temperature monitoring, fault locating, reactive power management, and so on.

4.1.10 : Vibration Motor (vibrating motor)

Fig. 4.10.1 : Vibration Motor

These tiny circular motors have offset weights that make them vibrate when they spin. They're normally called "pager motors" or coin vibrator because they're the type found in pagers and cell phones that have a "vibrate" feature.

Vibration motors are one of the prominent motors used for machine feedback systems. Its functions as register feedbacks for any user inputs or a source for output. This Disc shaped motor designed specifically for products with small form factors. This motor used in a wide range of devices and gadgets where vibration required. And also widely used in many electronic projects such as Bristlebots, Brushbots, remote control aircraft, robot beam projects, miniature trains, Radio control remote, Camera, Toothbrush, Electric Shaver, Mobile devices, etc

This Mini Disc Vibrator Motor designed as a tiny motor with offset weights that make them vibrate. This motor basically used to generate vibratory alerts. This shaft-less vibration motor comes in fully enclosed with no exposed moving parts. It comes in a small size of 4mm diameter and 8mm height. Here shaftless design indicates that it can mount it on a PCB or even place it in a pocket to add quiet, easily.Two wires are used to control/power the vibe. Simply provide power from a battery or microcontroller pin (red is positive, blue is negative) and it will buzz away. The rated voltage is 2.5 to 3.8V and for many projects, we found it vibrates from 2V up to 5V, higher voltages result in more current draw but also a stronger vibration.

These 0.8V-3V Micro Vibrators with a maximum speed of 1100 RPM perfect for lightweight applications or where space is a constraint. This is a tiny motor consists of double-sided tape on the bottom side used to attach this motor to any flat surface. When power supplied, the vibration motor causes the attached surface to vibrate. The support leads coming out from the motor body used to solder the motor to a PCB quite easily. In this motor, Polarity is not important; the motor can run CW or CCW.

Applications

The perfect thing for making Bristlebots, vibrobots, BEAM bots, and other tiny robots!

These motors can be driven with 3 V coin cells like the CR2032. Each one comes in a removable rubber boot that has one flat side for easy mounting.

Specification

- Nominal voltage: 3 V
- Operating voltage: 2.0 ~ 3.5 V
- Rated current: 85 mA
- Nominal speed: ~12000 RPM
- Size:10x2.5mm

4.1.11 : Buzzer

Fig. 4.11.1 : Buzzer

Features

- Black in colour
- With internal drive circuit
- Sealed structure
- Wave solderable and washable
- Housing material: Noryl

Applications

- Computer and peripherals
- Communications equipment
- Portable equipment
- Automobile electronics
- POS system
- Electronic cash register

Specifications:

- Rated Voltage : 6V DC
- Operating Voltage : 4 to 8V DC
- Rated Current* : ≤30mA
- Sound Output at 10cm* : ≥85dB
- Resonant Frequency : 2300 ±300Hz
- Tone : Continuous
- Operating Temperature : -25°C to +80°C
- Storage Temperature : -30°C to +85°C
- Weight : 2g
- *Value applying at rated voltage (DC)

4.1.12: Zigbee USB Interfacing Board

ZigBee (Xbee) USB Interfacing Board is used to interface Xbee wireless module with computer systems. This Board is used to connect ZigBee modules to make communication between PC to PC or laptop, PC to Mechanical Assembly or robot, PC to embedded and microcontroller based Circuits. As ZigBee communicates through Serial Communication so other end of USB which is connected to a PC, treated as COM port for Serial Communication. It is provided with indication LEDs for ease.

Fig. 4.12.1 : Zigbee USB Interfacing Board

It supports both AT and API mode. Its baud ranges from 2400 bps to 115200 bps. On this interfacing board, CP2102 IC is used for converting TTL logic to USB logic

4.2 Software :

4.2.1 : Arduino IDE

What is the Arduino IDE ?

As we know we need a text/code editor to write the code, a compiler to convert that code to machine code or binary files so that the microcontroller can understand, and also programming software to load these firmware files onto the microcontroller. When we combine all these with some additional features like debugging support, console support, etc, that's what we call an IDE (Integrated Development Environment) or in simple terms the **Arduino Software**. Arduino IDE, as the name states, is a development IDE for the Arduino boards. It consists of a feature-rich code editor, compiler, programmer, serial console, serial plotter, and many other features. It is simple and easy to use.

Arduino IDE is cross-platform, and it can run on operating systems from Microsoft, Linux, and Windows. Furthermore, you can program the boards using the Arduino IDE and Arduino Language, which is a derivative of C/C++.

How to Install Arduino IDE ?

Installing Arduino IDE is pretty straightforward. Go to Arduino IDE to download the recent version of Arduino IDE. There are multiple versions available for different operating systems such as Windows, Mac, and Linux. Not only that, nowadays Arduino IDE comes in two variants Arduino IDE 1.x and Arduino IDE 2.x. For this tutorial, we will be talking about the Classic 1.X variant. Basically, both have almost same functionality with a different GUI plus some additional features such as auto code completion.

- Download the installer for your operating system from the above-given link.
- Once the download is done open the .exe file.
- Agree to the licence agreement and select if the IDE should be installed to all users or not and click on “Next” to continue.
- Select the location in which you want the IDE to install if you want to change the location or keep it default and click on “Install”
- Wait for the installer to finish installation, and click on “Close”.

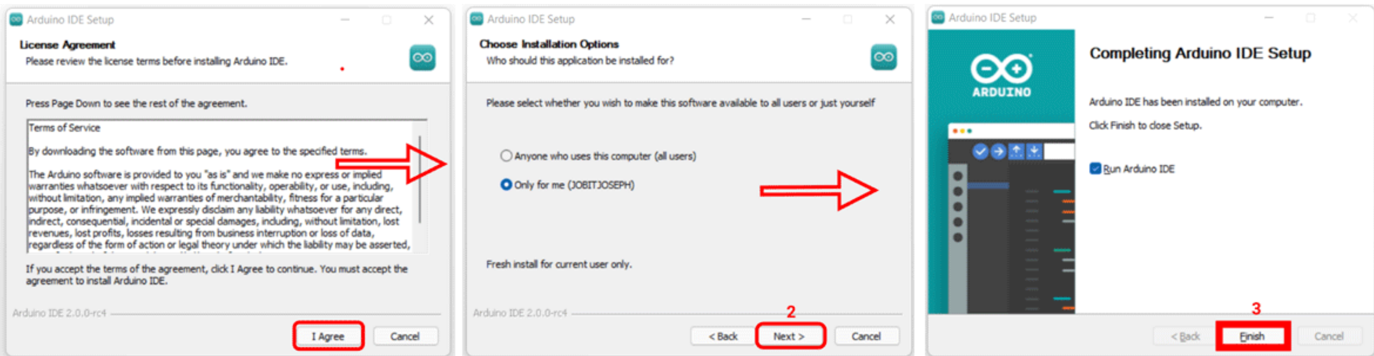


Fig 4.1.1 installation Process

Connecting Arduino Board to the Computer

To connect the Arduino board to the computer, simply connect the appropriate cable to the Arduino board and connect the other end to the USB port of your PC. The power LED will glow indicating the board is powered. The system will automatically install the driver for the board.

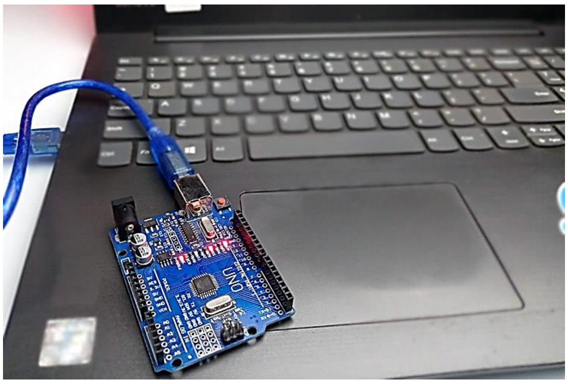


Fig 4.1.2 Arduino Connection

Arduino IDE – Basics

After the Arduino IDE installation, you can launch the Arduino IDE by double-clicking the Arduino IDE shortcut on the Desktop or from the Start Menu. Now the Arduino IDE will be opened. The below image shows the **Arduino IDE interface**.

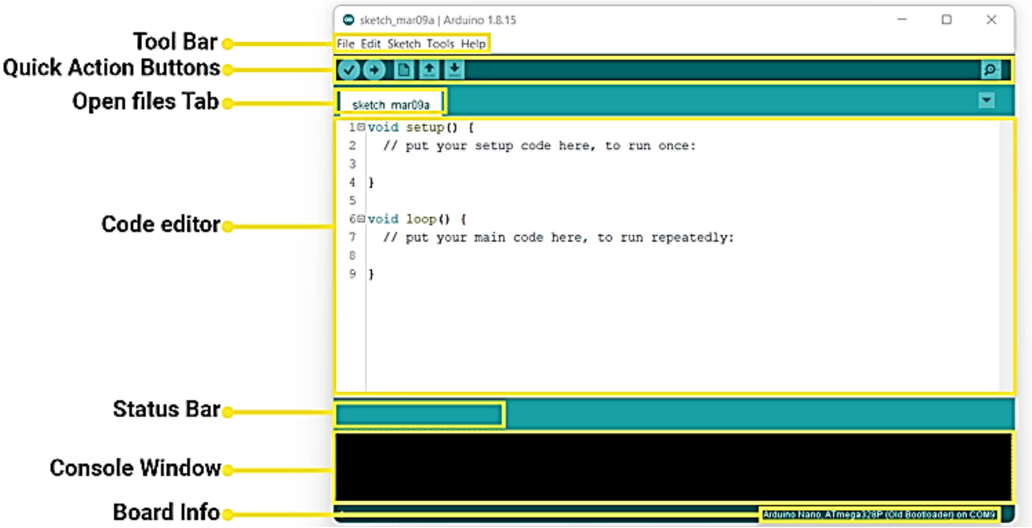


Fig 4.1.3 : Arduino IDE

Selecting the right Board on Arduino

Now let's select the proper board. It is very important to select the proper board before compilation because the compiler will use this in the compile directives. To do that Click on the **"Tools" -> "Board" -> "Arduino AVR Boards"** and select your board from it.

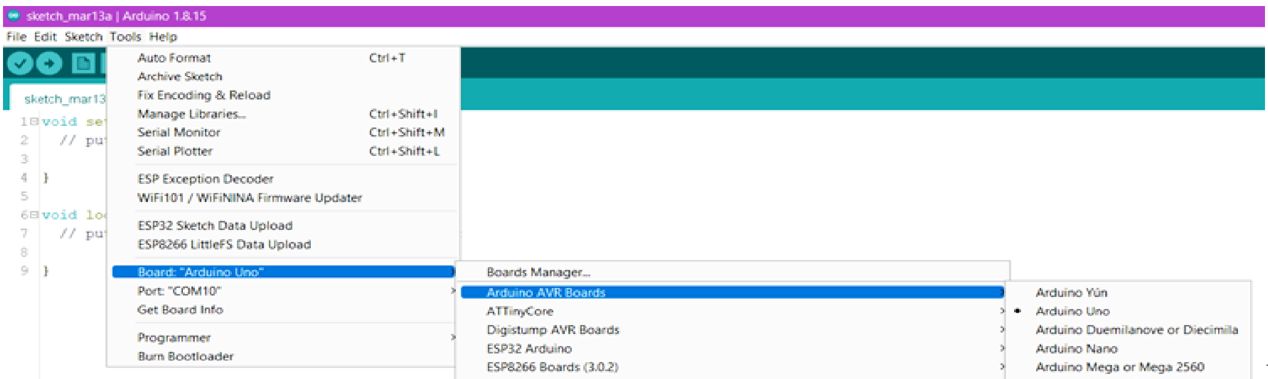


Fig 4.1.4 : Selection of Board

Select the Arduino Serial Port

It's also important to select the proper serial port to which the Arduino board is connected. Otherwise, you won't be able to upload the code to the board. To do that click on **"Tools" -> "Port"** And select the correct COM port. If there are multiple COM ports and you are not sure which one to select, disconnect the Arduino board from the USB port and reopen the menu. The COM port entry that disappeared will be the correct COM port. Reconnect the Board to the same USB port and select that port.

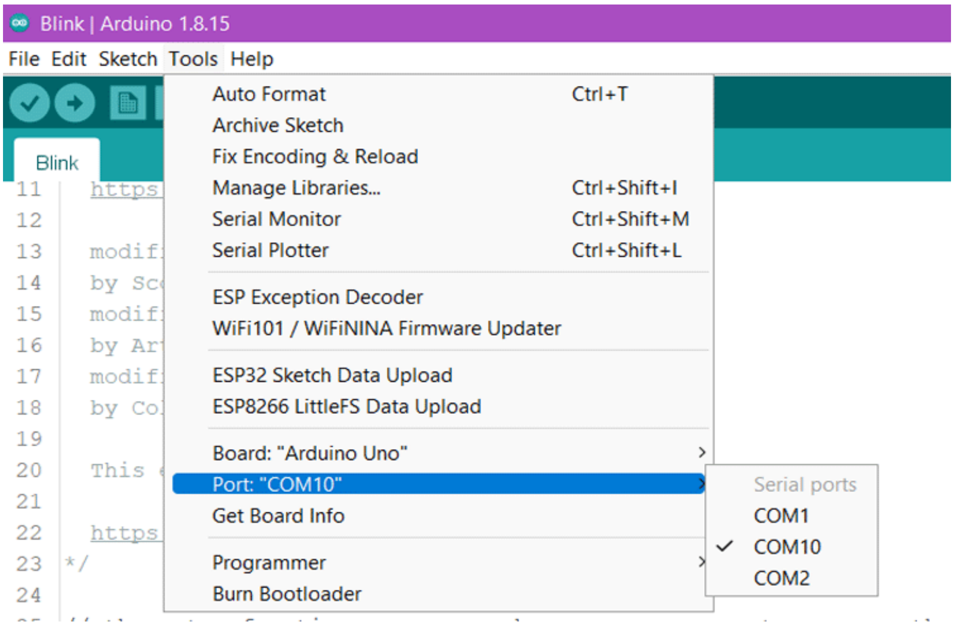


Fig 4.1.5: Selection of Serial Port

Arduino Example Codes

To start with we will be using the LED Blink example that is already provided with the Arduino IDE. To open the Blink example, go to **Files -> Examples -> Basics -> Blink.**

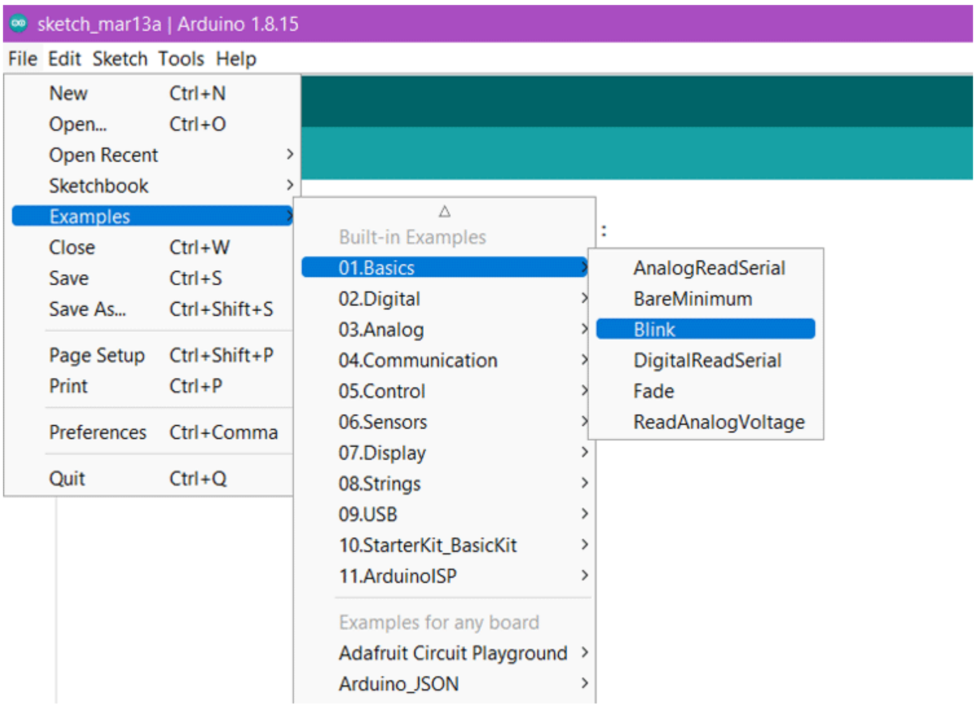


Fig 4.1.6: Arduino Example Codes

Example Blink Program for Arduino

Let's open an example code to understand it further. For that click on **File -> Examples -> Basics -> Blink.**

A new window with the example blink code will be open. In the code area, you can see two functions, **void setup()** and **void loop()**. The void setup() function is the first function that will run when the Arduino is powered on. Normally we will add any initialization codes in this function. The void loop() will run in a loop, and we will add any code that needed to be repeated to this function.

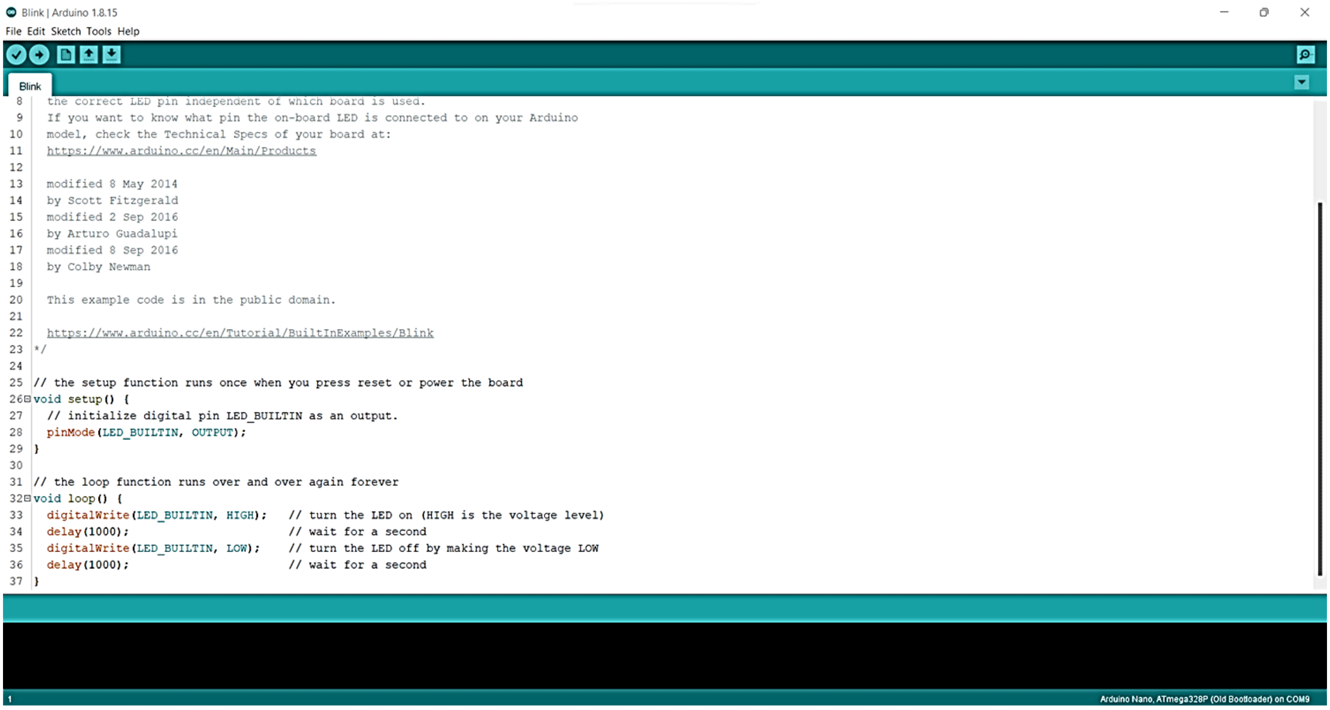


Fig 4.1.7 : Example Code

As mentioned earlier, in the **void setup()** function there is an initialization code.

```
pinMode(LED_BUILTIN, OUTPUT);
```

In this line with **pinMode(),** we set the pin which is connected to the built-in LED as an output. **pinMode()** is used to set a specific pin as INPUT or OUTPUT. The syntax is **pinMode (pin, mode)**. In which **pin** is the pin number, and the **mode** is INPUT or OUTPUT. In the example we are setting the built-in LED pin, i.e., is pin number 13, which is defined in the Arduino board definition file, to an output.

```
Void loop () {
digitalWrite (LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
delay(1000);                        //wait for a second
digitalWrite(LED_BUILTIN, LOW);     //turn the LED off by making the voltage Low
delay(1000);                        // wait for a second
}
```

If we look at the Loop function, we are calling two functions repeatedly **digitalWrite()** and **delay()**.

The **digitalWrite()** function is used to change the state of a specific pin, i.e. we change the pin's state to high or low. The syntax is **digitalWrite(pin, value)** in which the **pin** is the corresponding pin number and the **value** is the state or value we want to set to the pin, i.e., **HIGH** or **LOW**.

The **delay()** function is a blocking function to pause a program from doing a task during the specified duration in milliseconds. For example **delay(1000)** will pause the program for 1000millisecods.

In the blink example, after the pin is set as an output, we set the pin as HIGH and wait for 1000ms. After the 1000ms, we change the pin state to LOW and again wait for 1000ms or 1 second. After the 1Second pause, the program will change the pin state back to HIGH and this loop will repeat continuously. As a result, we will get a square wave out from that pin. When we connect an LED to this pin, the LED will blink. The rate of blinking can be changed by changing the delay, by changing the values in the **delay()** function.

Uploading Code on Arduino

To upload the code to confirm that the Arduino is plugged in and the correct board and port are selected, either click the upload button in the quick action bar or click on **Sketch -> Upload.** Or you can also you the keyboard shortcut **Ctrl+U.**

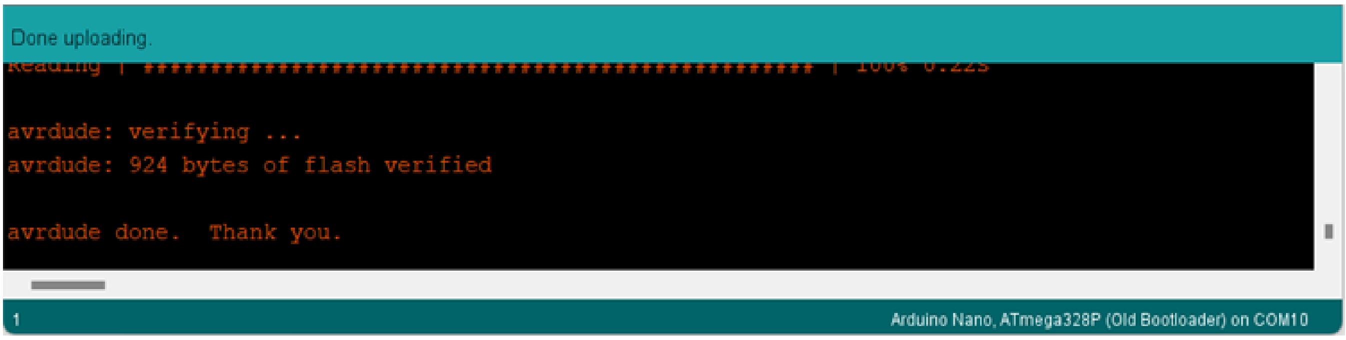


Fig 4.1.8: code uploading done

If everything went correctly you will see a Done uploading message in the status bar and the LED will start to blink.

4.2.2 : XCTU

The Xbee devices communicate with each other wirelessly over the air. They do not have any microcontroller or processor in themselves, so they cannot manage the received or sent data. They can simply transfer the information what they receive. But they can be interfaced with other microcontrollers and processors like Arduino, Raspberry Pi or PC via serial Interface.

So, basically, Xbee modules are capable of two types of communication – wireless communication and serial communication. The wireless communication takes place between Xbee devices so that the devices act as radio frequency (RF) devices. For data to transmit and receive from one Xbee module to another, both devices should be on same network. The data between two devices is transmitted wirelessly. By serial communication (UART), the Xbee modules can communicate with microcontrollers and processors.

A microcontroller, processor or PC can send data through the serial interface to the Xbee module (transmitter) and the Xbee module wirelessly transmits the data to other Xbee module (Receiver). The receiver Xbee module transmits the data through the serial interface to controller, processor or PC to which it is interfaced. The controller interfaced to the Xbee module processes the information received by the Xbee devices. This way, controllers can monitor and control remote devices by sending messages through the local Xbee modules.
The Xbee modules communicate with each other in two modes – Transparent mode and API (Application Peripheral Interface) mode. This project is based on Transparent mode. In transparent mode, Xbee modules act as serial line replacement. All data received through serial input is immediately transmitted over the air. When other Xbee module receives data wirelessly, it sends that exactly as it receives through the serial interface and vice versa. Contrary to this, in the API mode, the data is transmitted with some additional information.

Hardware Requirements for Establishing Zigbee Communication

- 1 x Arduino Nano
- 1 x NodeMCU
- 2 x XBee Pro S2C modules
- 1 x XBee explorer board (to program XBee)
- USB cables
- LED (along with a 220-ohm resistor)
- Push Button

Configuring XBee Modules using XCTU Software

Basically, the XBee module can be configured as Coordinator, Router, or End device. To make it work as we desire, first, we have to configure them using XCTU software. You can download and install the XCTU software using the given link.

Use USB to serial converter or an explorer board to connect the XBee module with a PC or Laptop. Connect the XBee module to the Explorer board and plugin using a USB cable. We will get into the details on how exactly we can use this software and configure the module, but before we do that, let's get our hardware ready.

The XBee modules (transmitter and receiver) need to be configured using the X-CTU Software. It can be downloaded from here. This software is provided by DigiKey, and they have given a detailed configuration guide. Therefore, there is no point of me reinventing the wheel here. You can find the guide here.

There’s another one by Sparkfun that is adapted to the newer version of the X-CTU software.

Here’s another brief one by Instructables.

Please note that the two XBee modules that intend to communicate with each other should belong to the same series.

Here are a few things to note about the configuration −

- You will need a breakout board or an Explorer with a USB to UART converter for this configuration.
- The PAN ID (Personal Area Network ID) has to be the same for the devices that want to communicate with each other.
- One module needs to be set as the transmitter and the other as the receiver (this is determined by the CE field).
- Note the baud rate that you set. This will be used in the Arduino code, when configuring the Serial communication with XBee.

Downloading and Installing XCTU Software

To set up, configure and test your XBee devices. You need XCTU software. It's an easy-to-use, free, multi-platform application for RF XBee modules. Download the XCTU software here and it'll guide you to install it as well. After that, open the application and make sure your XBee module is connected properly. Check the COM port of the explorer board in the device manager.

X-CTU is a tool that configures ZigBee modules with whatever settings required. First step is to install X-CTU software. After this, ZigBee modules configuration begins. A point-to-point network is required to be constructed, which means that the two devices will communicate with each other. For this purpose, one ZigBee should be configured as a Coordinator (or Master), because it is mandatory to have one coordinator in each network and the other should be configured as a Router (or end device). So a Coordinator is created that will manage the network of ZigBee, here a network consists of one more ZigBee module. To begin, we need to take our ZigBee module (xbee radio) and insert it onto the XBee Explorer device and connect it to computer through RS232 serial port.

Lets start with the configuration :

1. Plug the S2C modules on to the USB adapters & connect to USB ports of your PC.
2. Open the new XCTU Software S2 can be configured with this new XCTU. The classic old XCTU does not support the S2C module.
3. Click on the SEARCH icon on top to detect the USB ports.
4. A list of active USB COM ports will be displayed.

5. Select the COM ports where you've connected the USB adapters. To confirm you can verify your DEVICE MANAGER for the proper COM ports. In my case one of the USB adapter is allotted COM3 & the other one COM31.

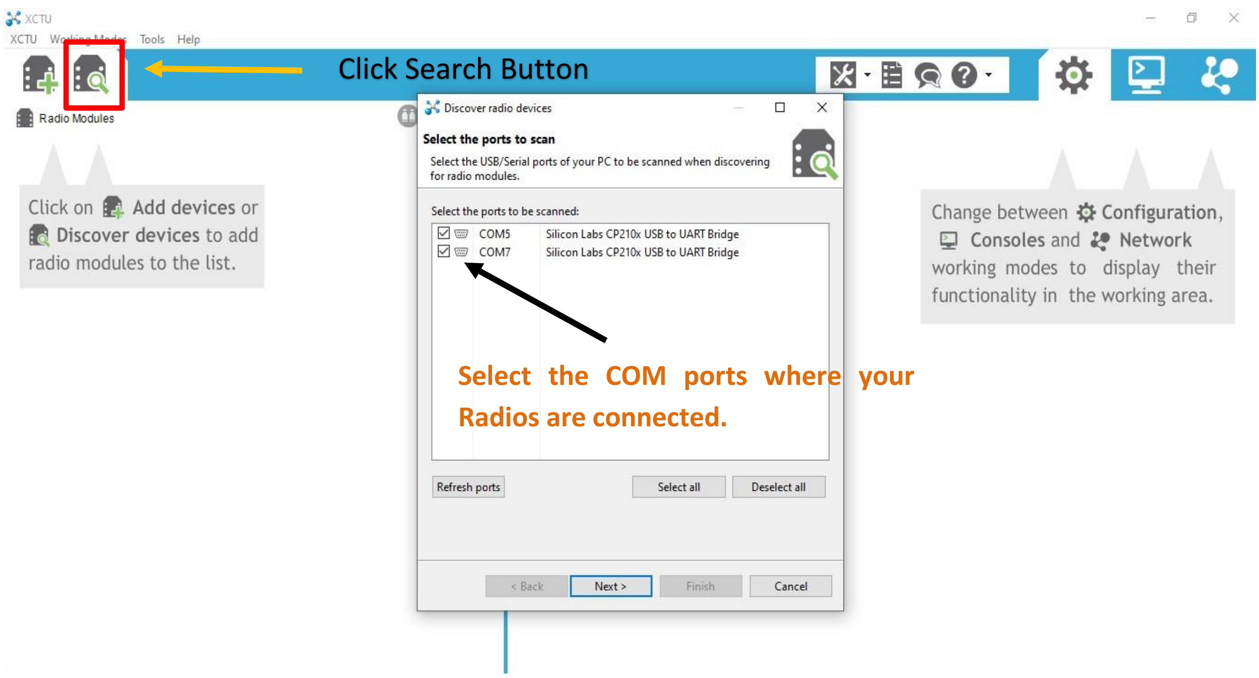


Fig. 4.2.1: Select COM ports

6. Click on NEXT & accept the default PORT PARAMETERS. 96008N1 is the default. 9600 is the BAUD RATE, 8 Data Bits, No Parity & 1 Stop bit.

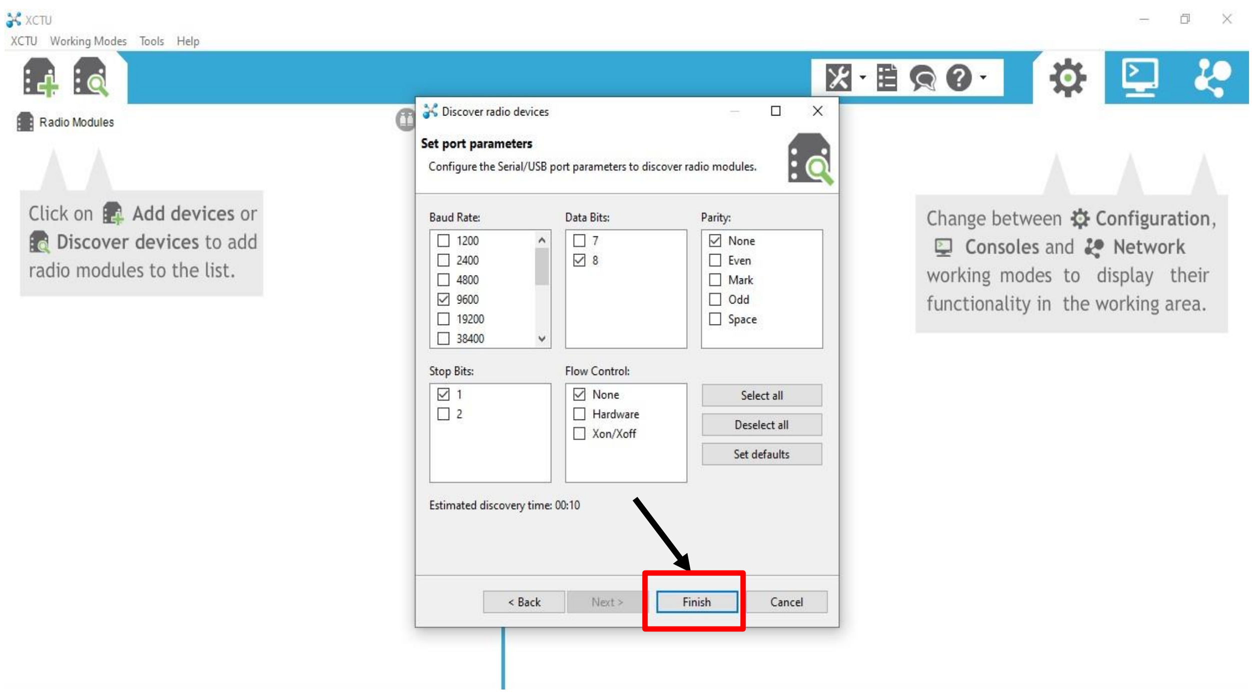


Fig. 4.2.2: Set Port Parameters

7. Click on FINISH. The XCTU scans the USB ports selected & lists the RADIOs found with their unique 64 bit address.

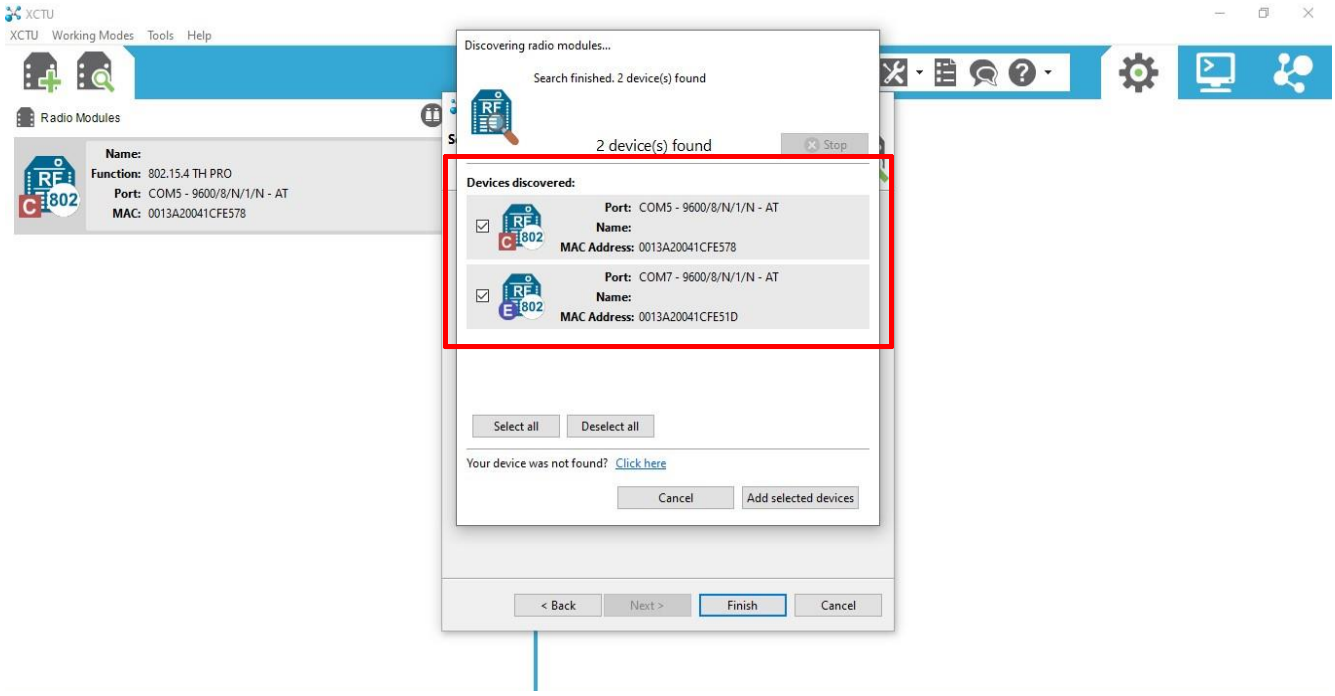


Fig. 4.2.3: Discovering Radio Modules

8. Select both the devices & click ADD SELECTED DEVICES. Now both the Radios appear in the left pane.

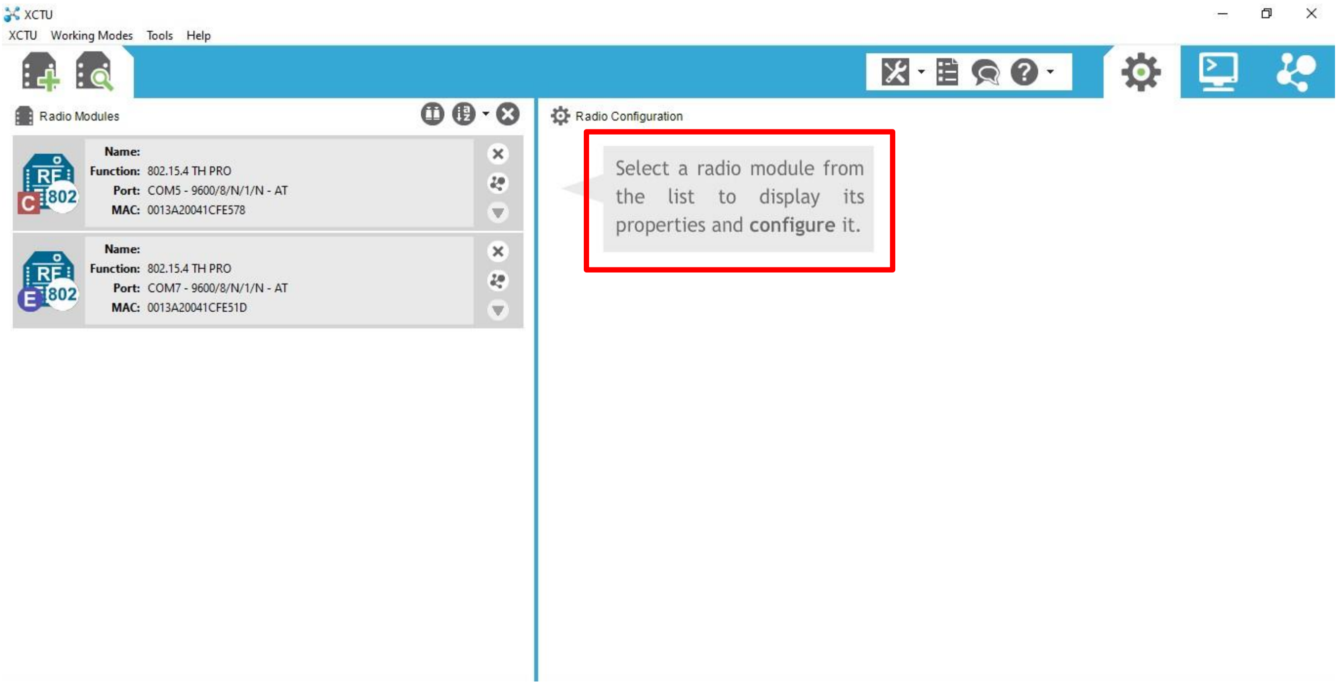


Fig. 4.2.4 : Add Selected devices

Let us configure the RADIO at COM5 as COORDINATOR first.

For this click on the COM5 RADIO to load the module settings.

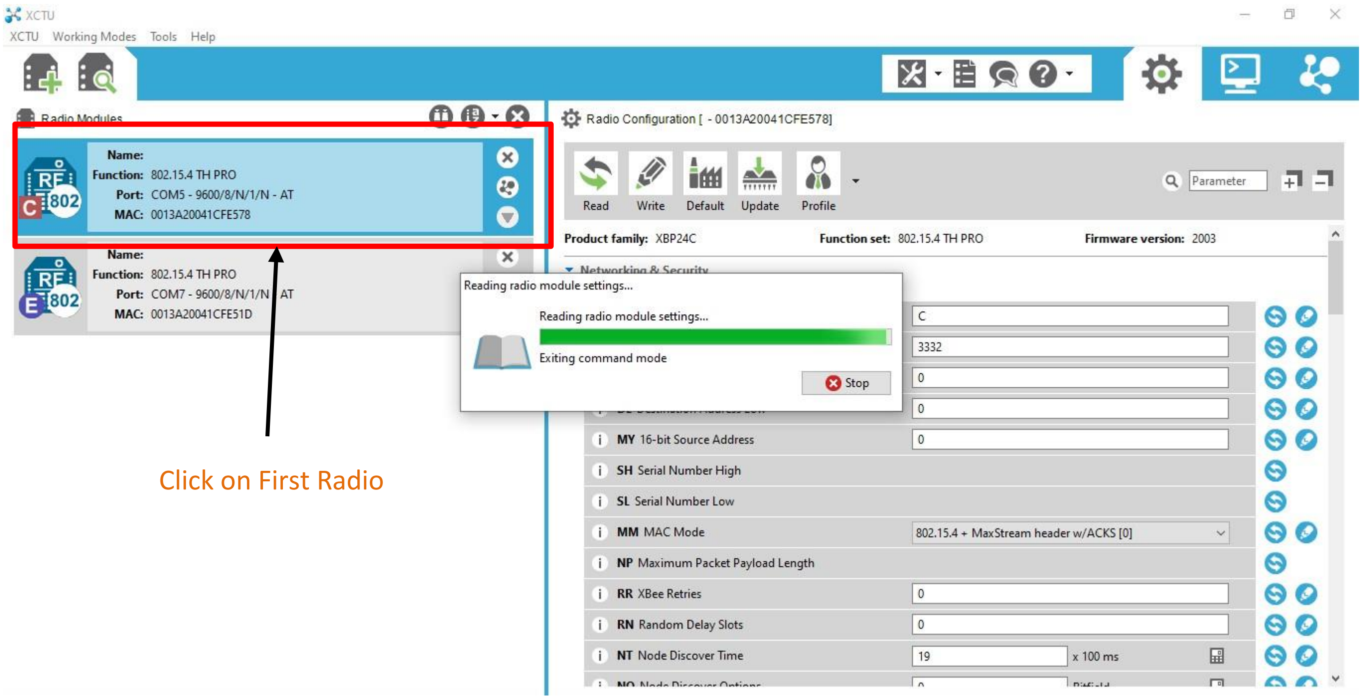


Fig. 4.2.5 : Configuration of Coordinator

Once the parameter settings are loaded you can see that the product family is **XBP24C** (in case of old S2 it is XB24-ZB & of $1 is 802.15.4)

The function set of Firmware is **802.15.4 TH PRO** , These modules use the IEEE **802.15.4** networking protocol for fast point-to-multipoint or peer-to-peer networking. . TH stands for THROUGH HOLE & not SMD.

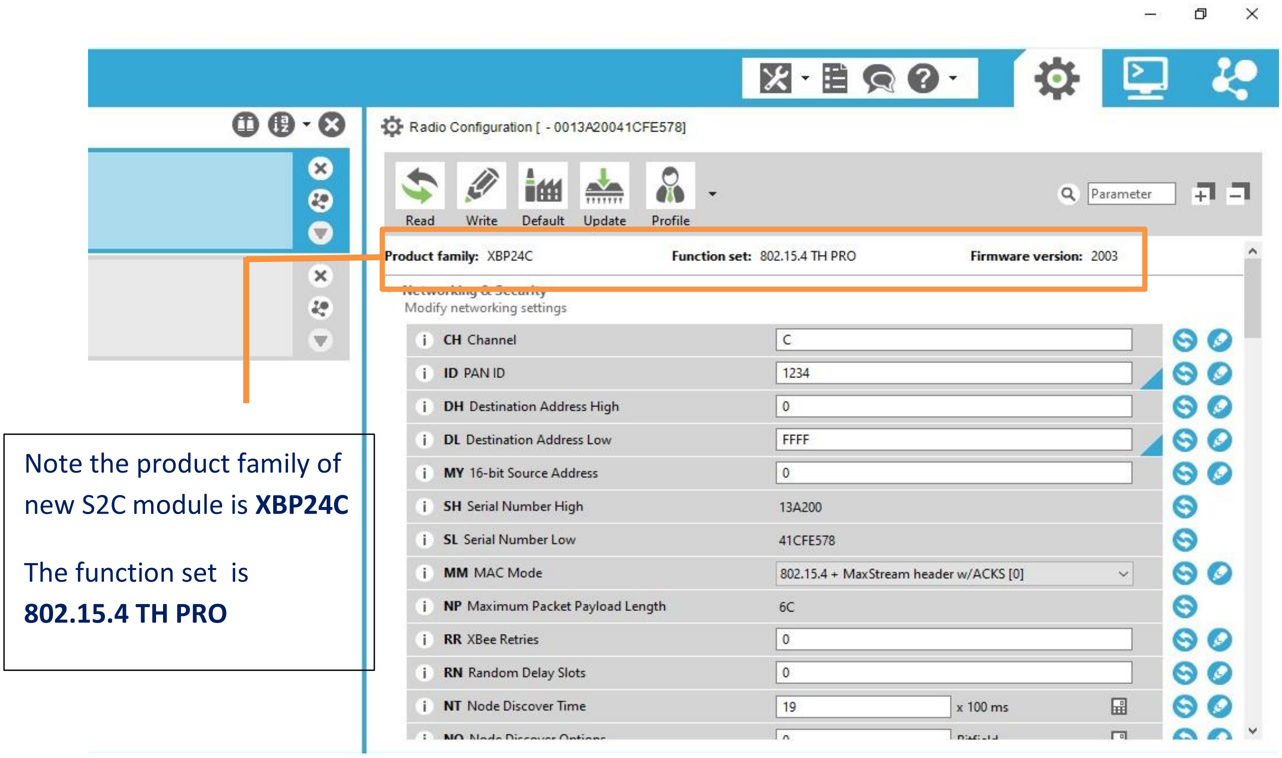


Fig. 4.2.6 : Parameter Setting

First thing is to set the PAN ID of the network. This can be from 0 to FFFF hex. In my case I am setting it to 1234. The other Radios also to be set in the same PAN ID.

Scroll down further & Enable the CE (Coordinator Enable)

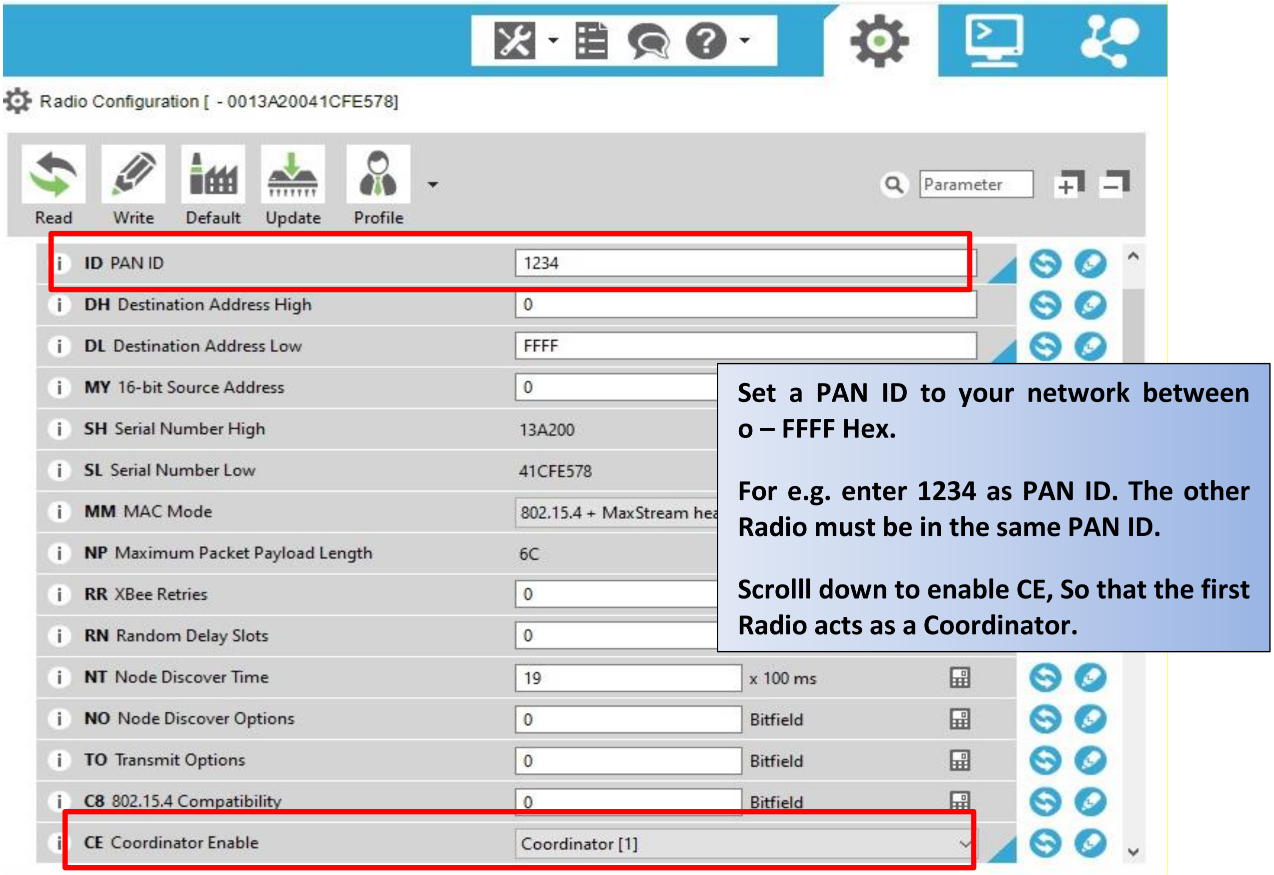


Fig. 4.2.7 : Radio Configuration – Networking 1

The Destination address DH is left to default 0.

The Destination Address DL is set to hex FFFF which makes the Radio work on BROADCAST mode, so that it can communicate with all Radios in the same PANID

The Node Identifier can be given any name like "Coordinator" This naming is optional.

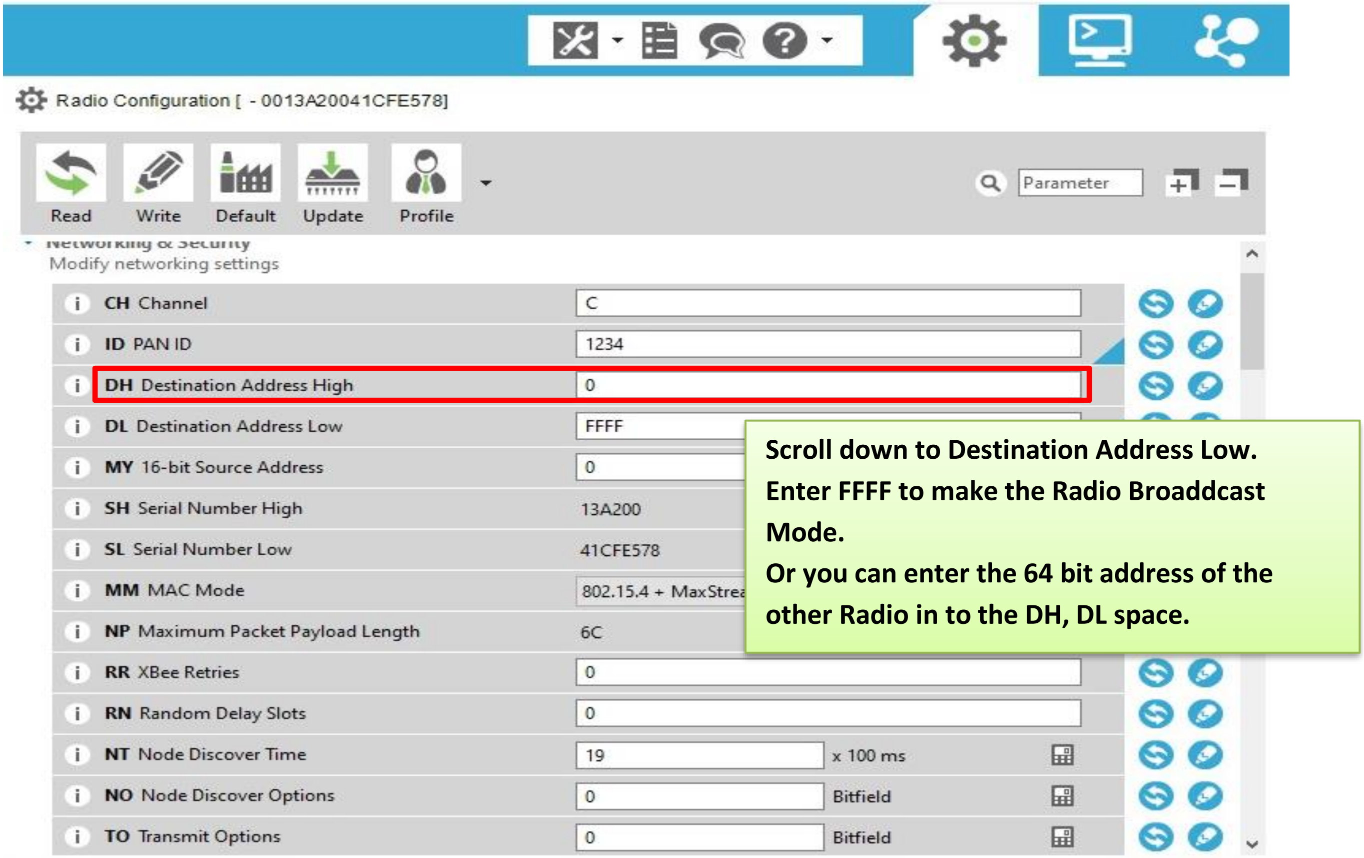


Fig. 4.2.8 : Radio Configuration – Networking 2

Click on the PENCIL icon on top to WRITE the changes made.

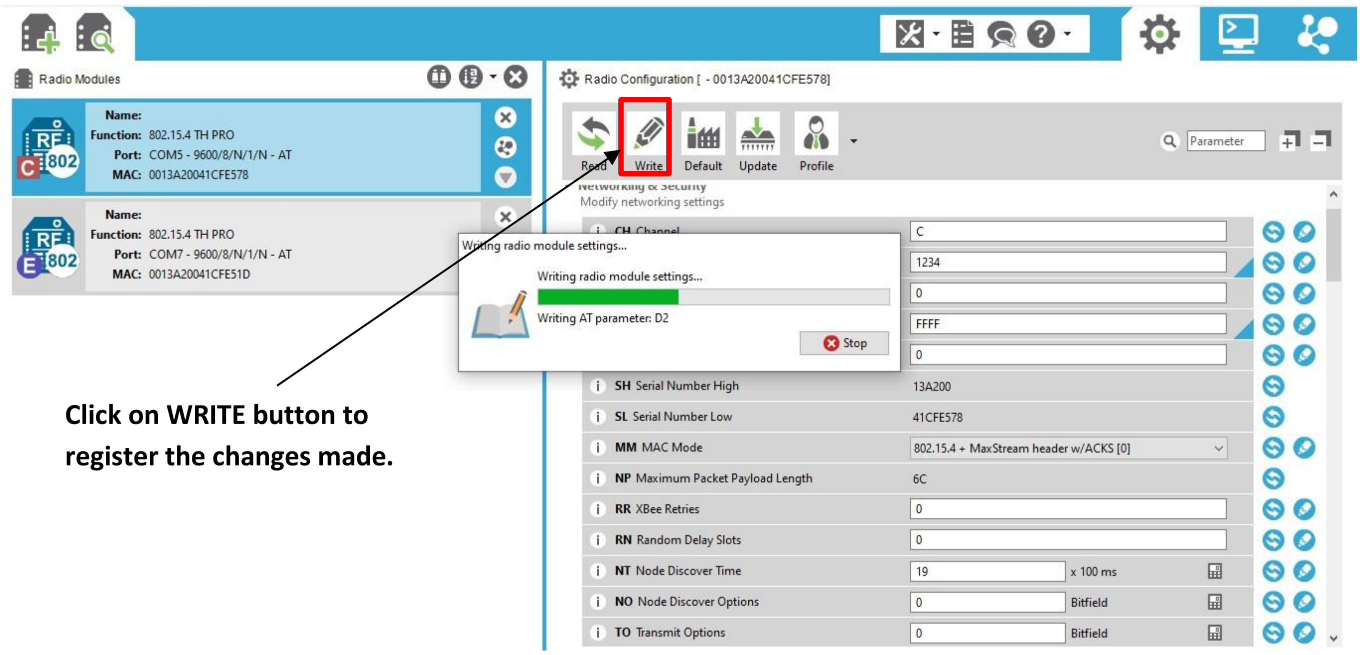


Fig. 4.2.9 : Writing Radio Module Setting

Now let us configure the second Radio as ROUTER.

Click on the second Radio on the left pane to load the settings.

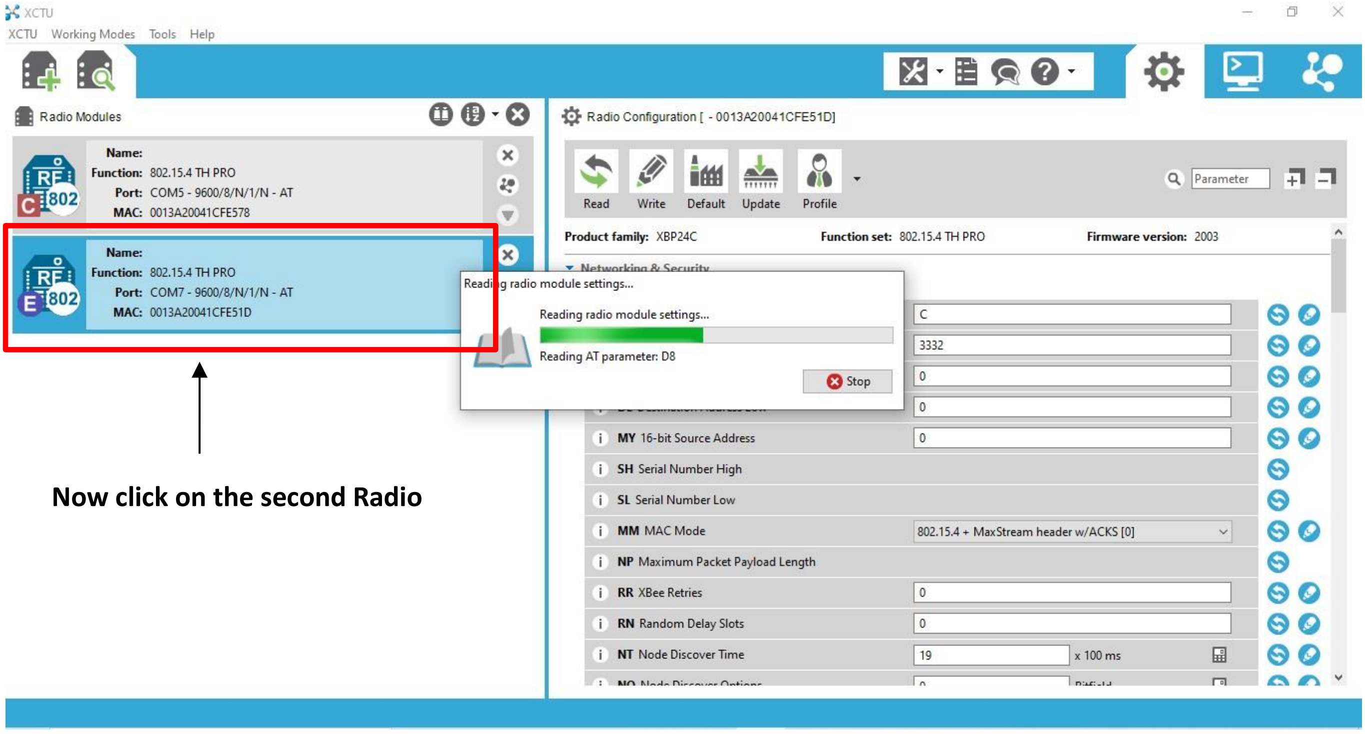


Fig. 4.2.10 : Configuration of End Device

The Router setting is quite simple.

Enter the PAN ID as 1234, same as that of Coordinator.

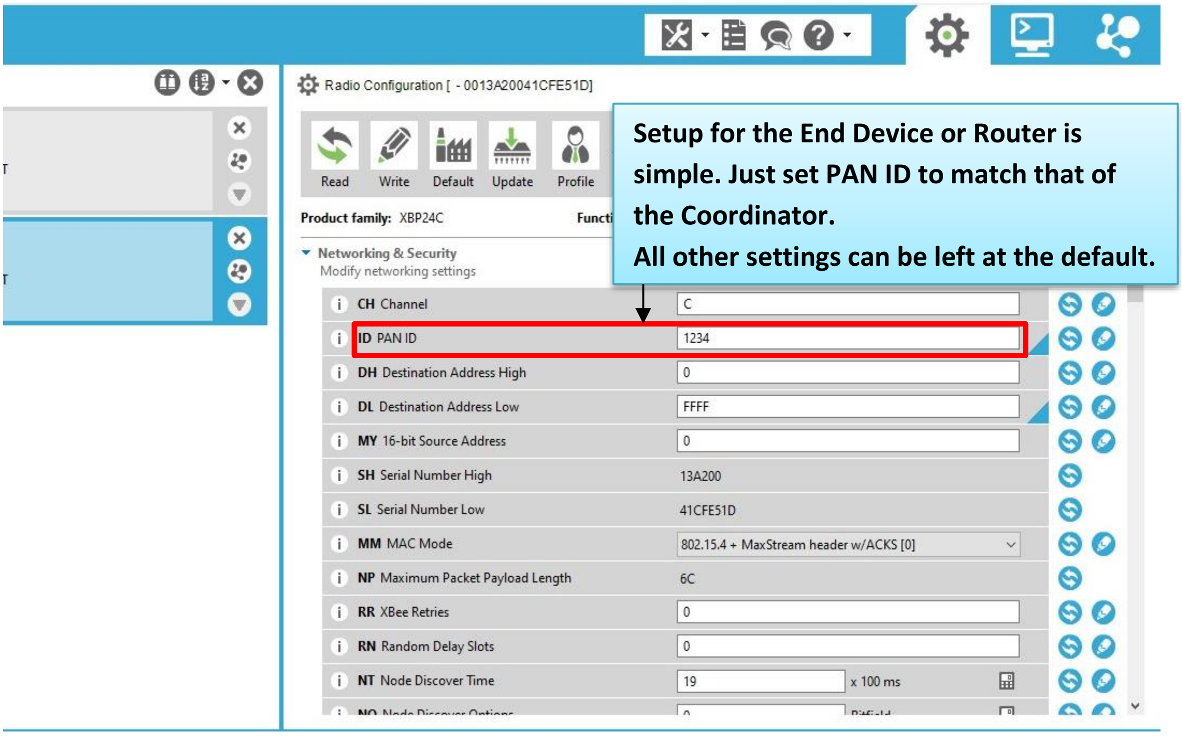


Fig. 4.2.11 : Radio Configuration – Networking 1

JV CHANNEL VERIFICATION is Enabled

CE Coordinator is DISABLED

Destination Address DL is left to default 0. (0 is the default address of Coordinator)

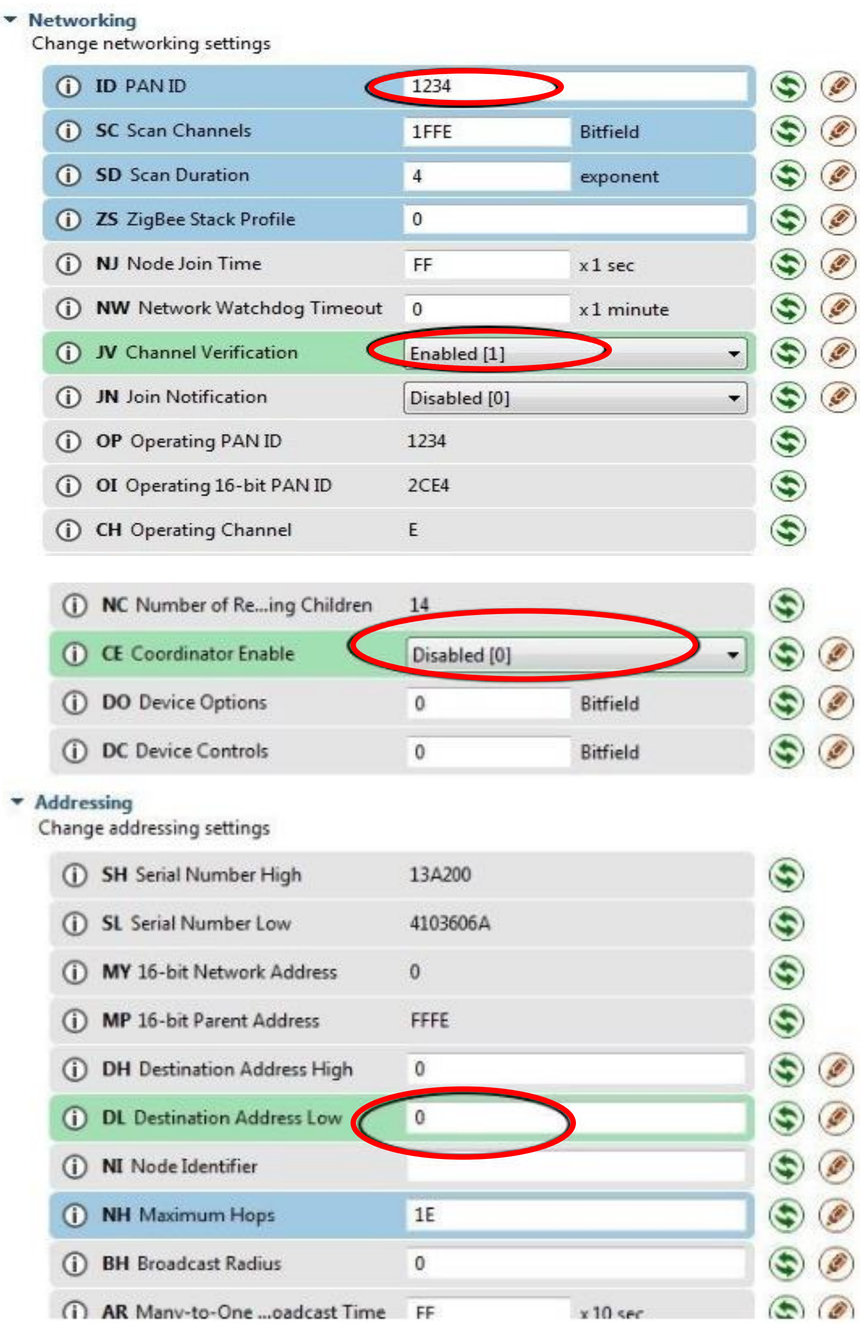


Fig. 4.2.12 : Radio Configuration Setting

Click on WRITE button to save the changes made.

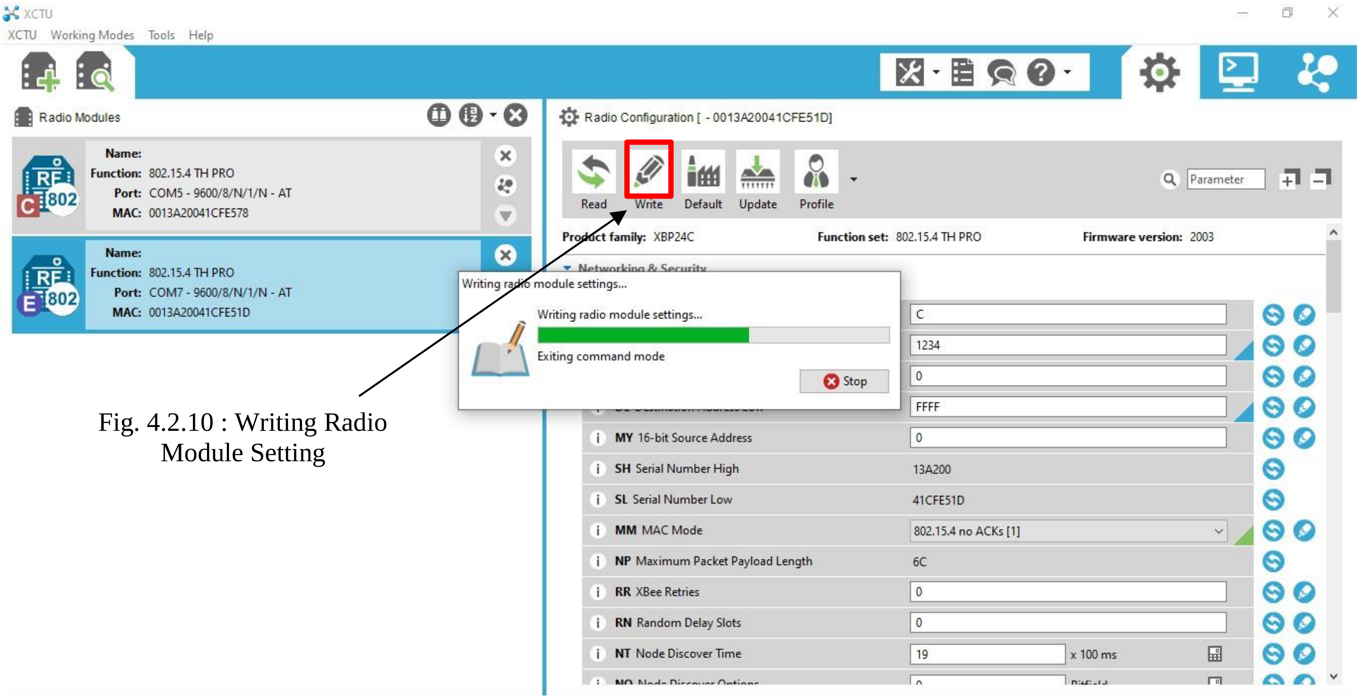


Fig. 4.2.13 : Writing Radio Module Setting

The modules are paired & ready for communication.

Now let us test the communication.

On the XCTU window delete the second Radio.Click on the first Radio to load the settings.

Leaving the XCTU window open, start another instance of XCTU & Openposition it to the left of the previous window.

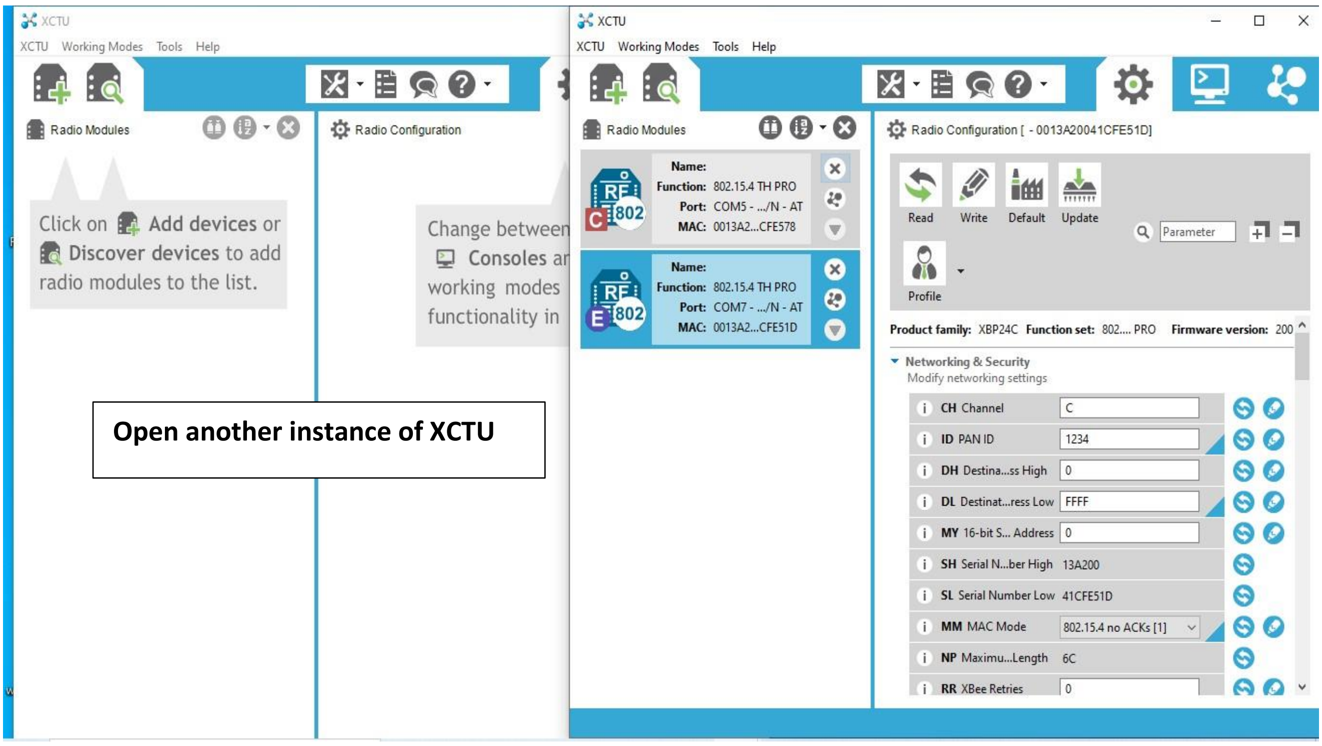


Fig. 4.2.14 : Another XCTU

Click on the Search button on the new instance of XCTU & select the second Radio .

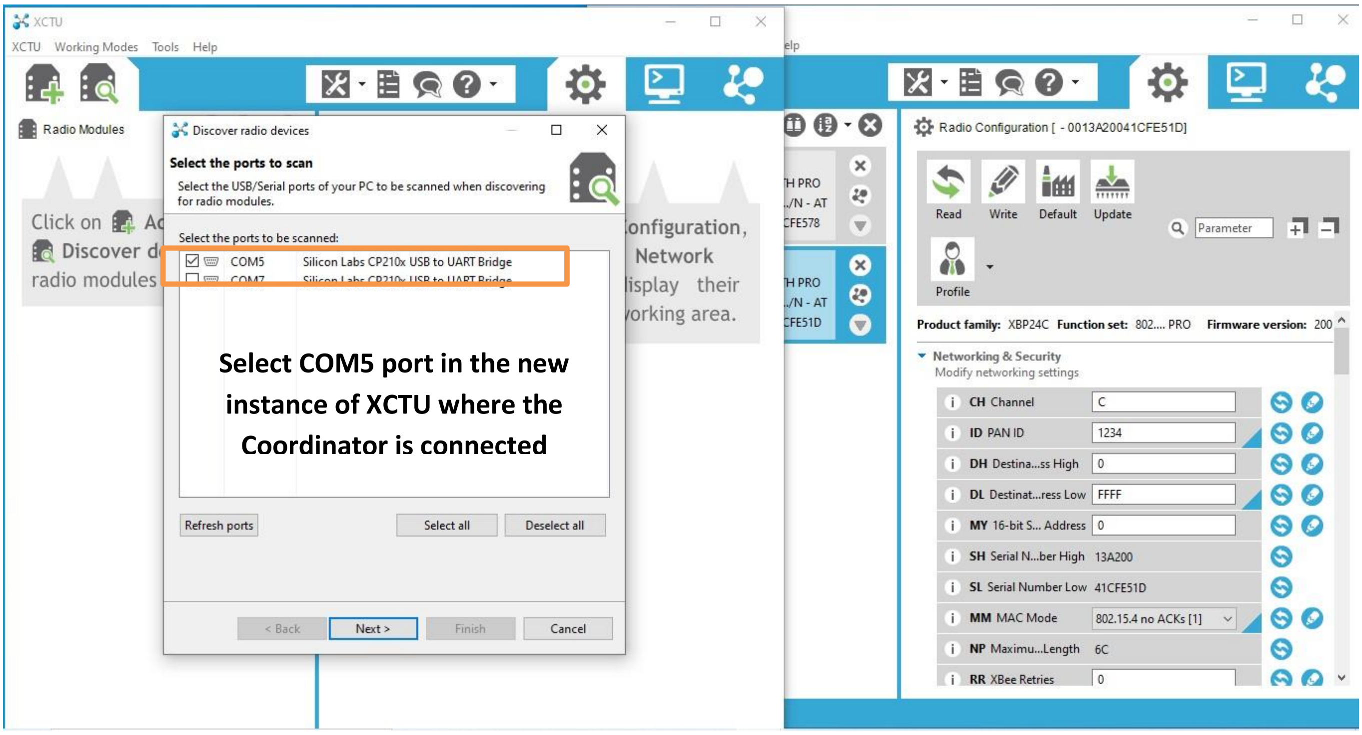


Fig. 4.2.15 : Discover Radio Devices

Click on the Radio selected to load the settings

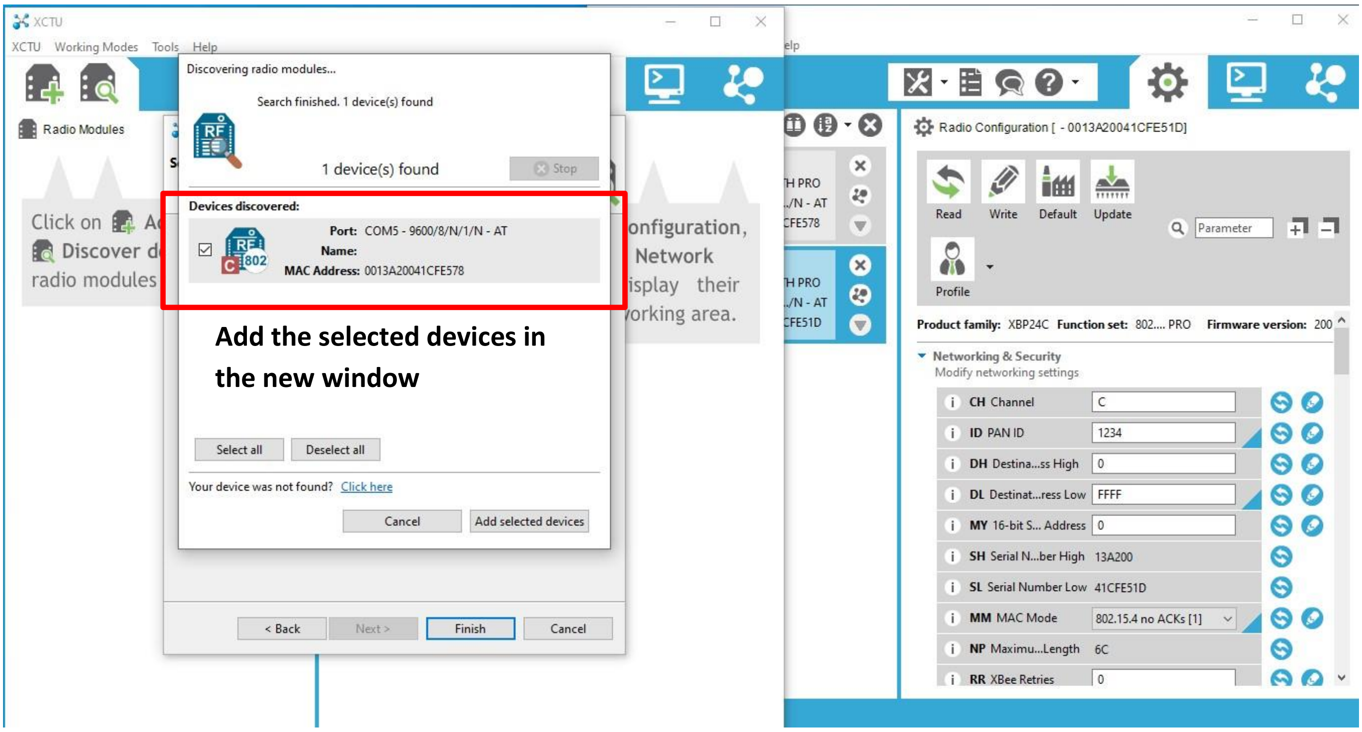


Fig. 4.2.16 : Discover Radio Modules

Now the ROUTER Radio is on the left side & the COORDINATOR Radio to the right.

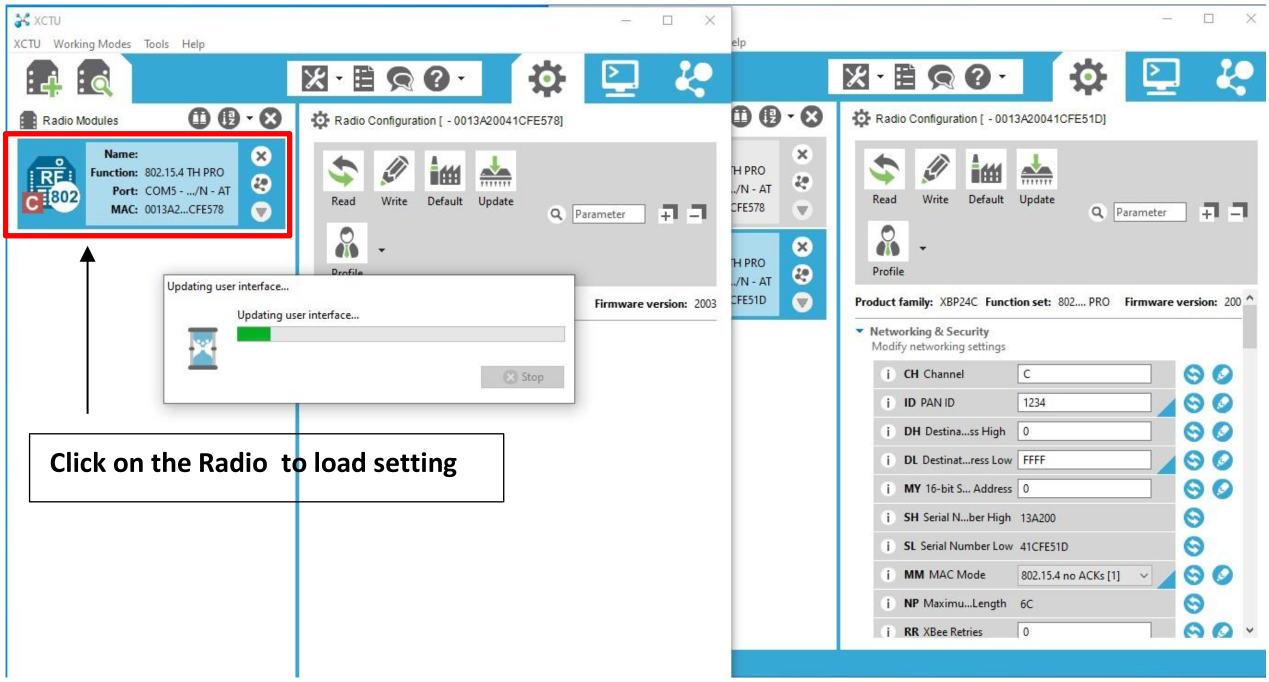


Fig. 4.2.17 : Configuration of device

Click the TERMINAL icon on both the windows to enter Terminal mode.

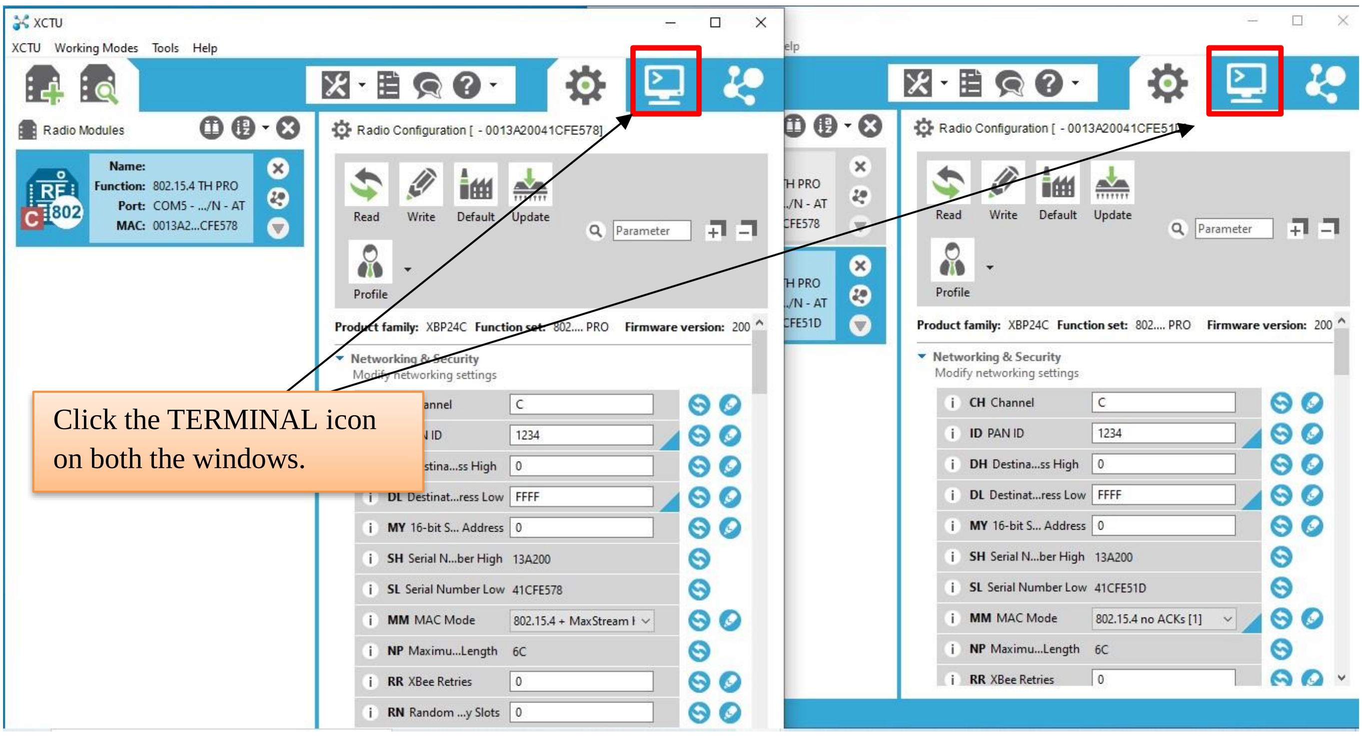


Fig. 4.2.18 : Open Terminal

Click on the serial connection icon on both the windows to enter the serial connection mode.

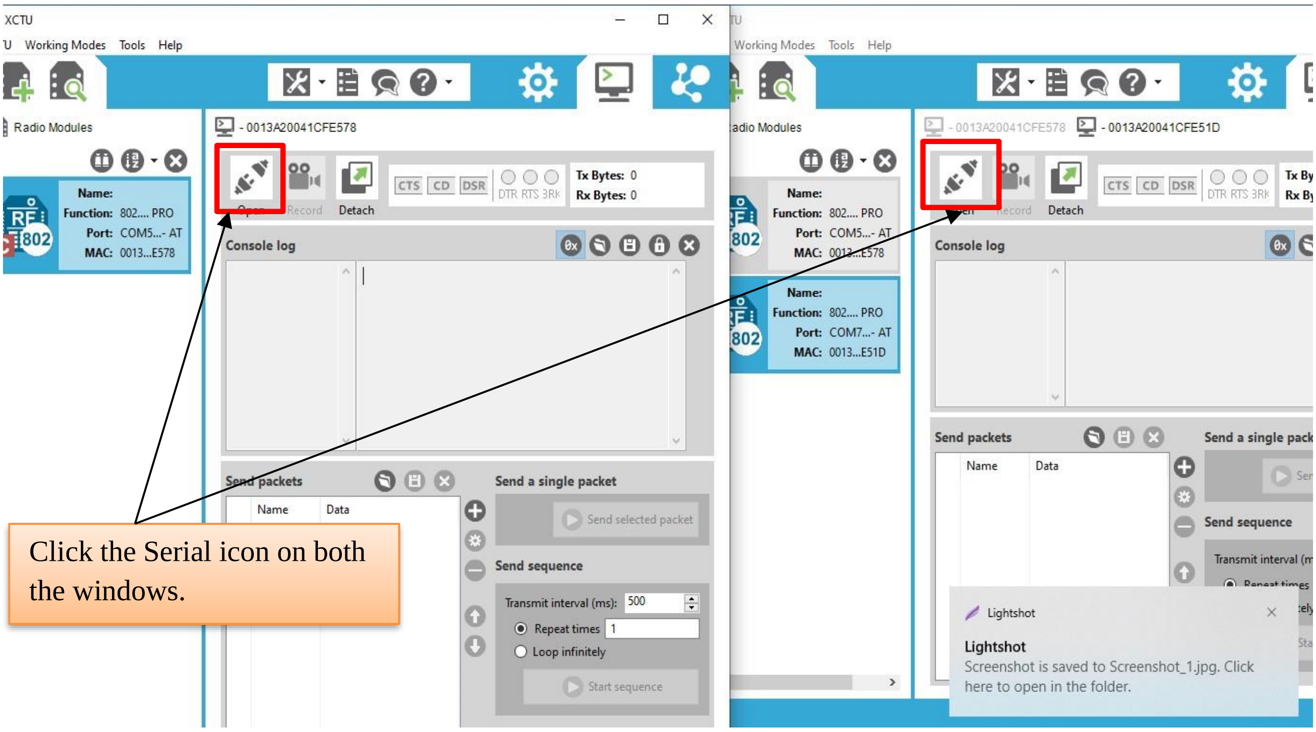


Fig. 4.2.19 : Open Serial Connection

You can see the SERIAL Icon in LOCK mode & the AT CONSOLE Status changes to CONNECTED.

Now you can type any message inside console log window & see that received on the other Radio. The transmit message is in BLUE & received message in RED.

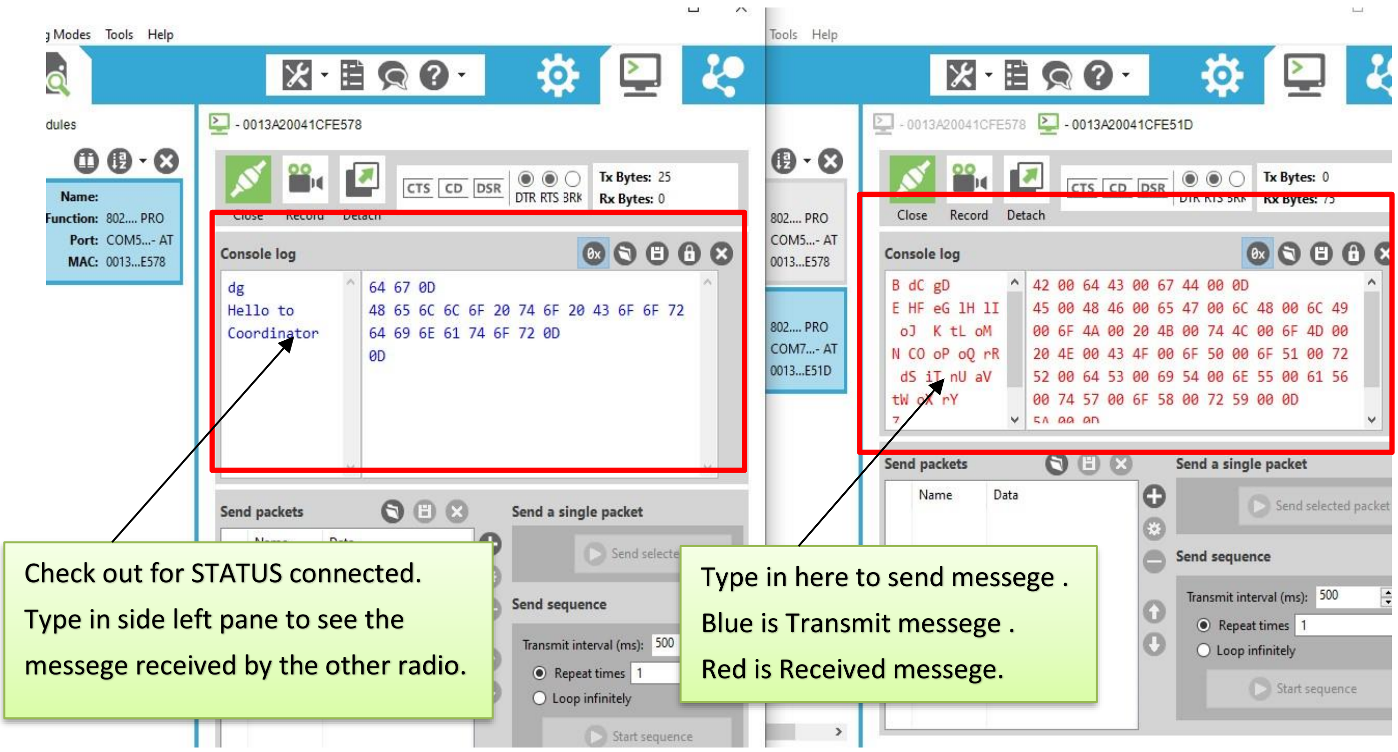


Fig. 4.2.20 : XBee Communication

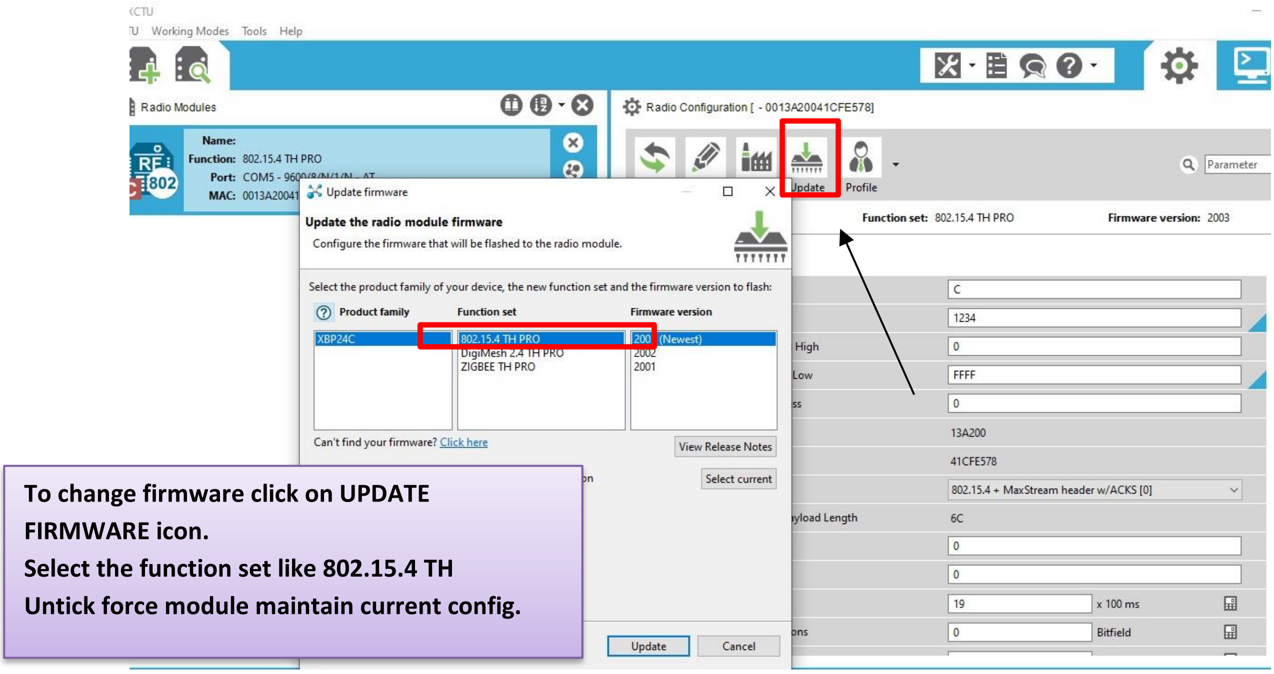


Fig. 4.2.21 : Firmware Updation

Once the Function set is selected , click on FINISH to load the firmware.

802.15.4 firmware is like that of our classic $1 module.But setting is slightly different.

CH channel can be left to default C

PANID to be selected , say 1234 (the other Radio to be on same ID)

DL address is FFFF

CE coordinator enable for first Radio

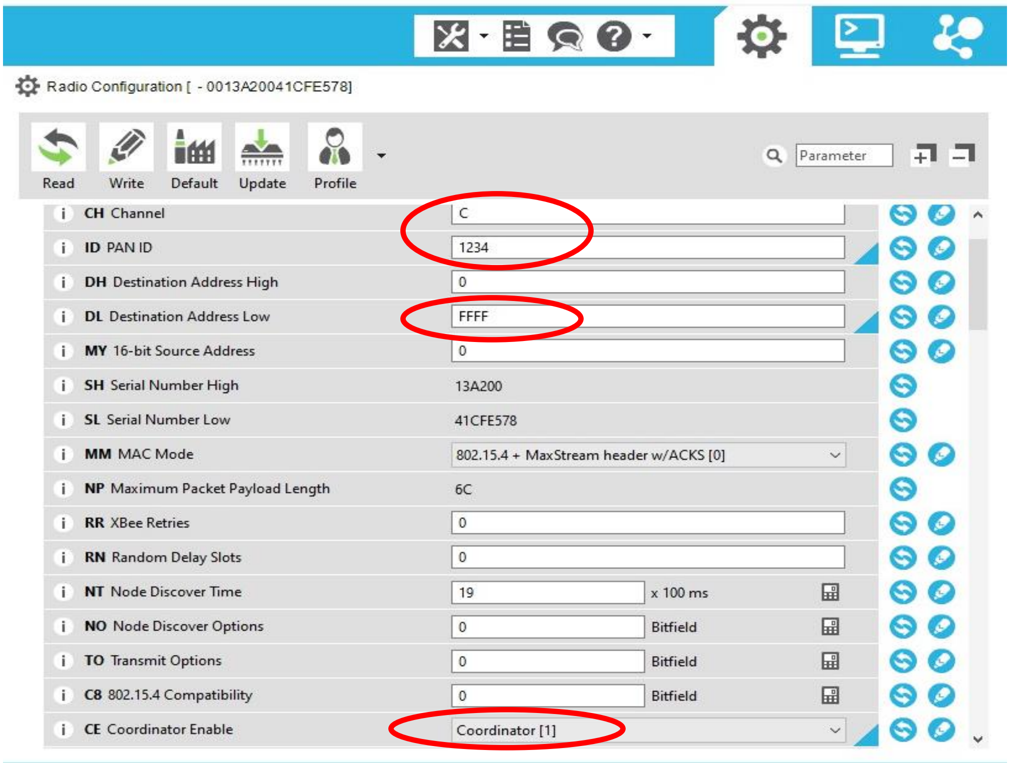


Fig. 4.2.22 : Radio Config 1

For the second Radio , to be set as END device

CH Channel C

PAN ID 1234 , same as Coordinator

MY address is 1 (the coordinator MY is 0)

CE is set to END device

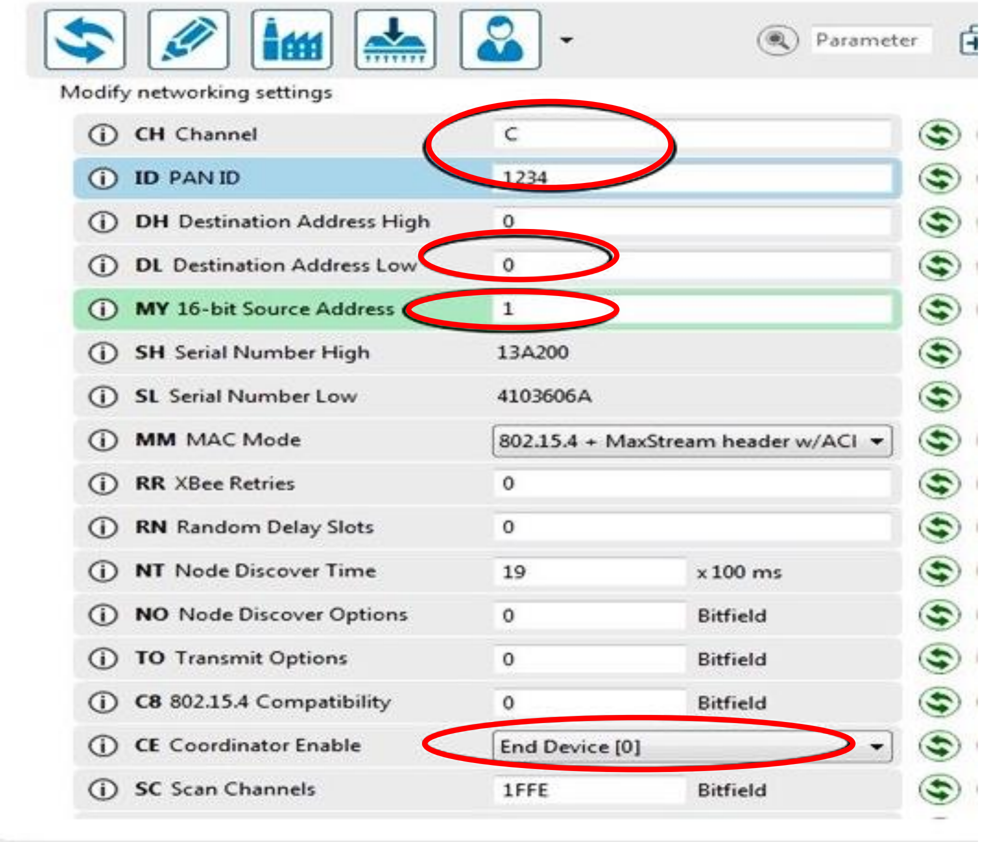


Fig. 4.2.23 : Radio Config 2

Click on WRITE button & test the modules for communication.

In this 802.15.4 Firmware POINT TO POINT communication only is possible & NO MESH Networking.

In a nutshell, the new X2C module is powerful & can work as S2 or the old S1 module.

4.2.3 : ThingSpeak

ThingSpeak is IoT platform for user to gather real-time data; for instance, climate information, location data and other device data. In different channels in ThingSpeak, you can summarize information and visualize data online in charts and analyze useful information.

ThingSpeak can integrate IoT:bit (micro:bit) and other software/ hardware platforms. Through IoT:bit, you can upload sensors data to ThingSpeak (e.g. temperature, humidity, light intensity, noise, motion, raindrop, distance and other device information).

Thingspeak Configuration

Goal :

we need to create the thingspeak channel and get the key

Step 1

Go to https://thingspeak.com/, register an account and login to the platform

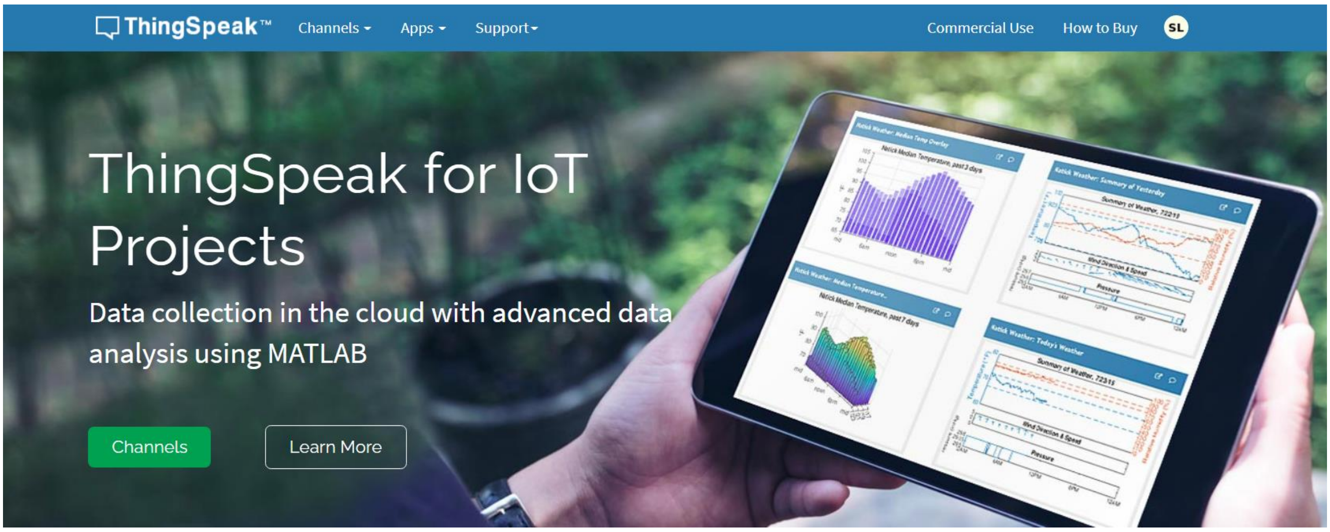
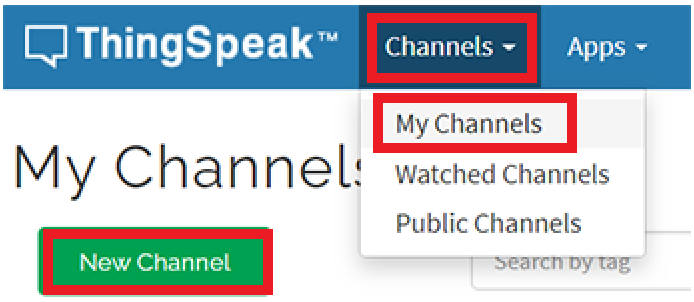


Fig. 4.3.1 : ThingSpeak Configuration

Step 2

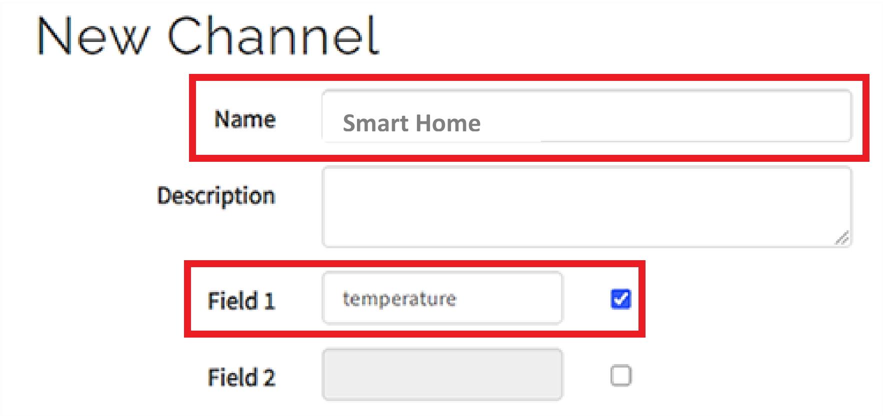
Choose Channels -> My Channels -> New Channel



Step 3

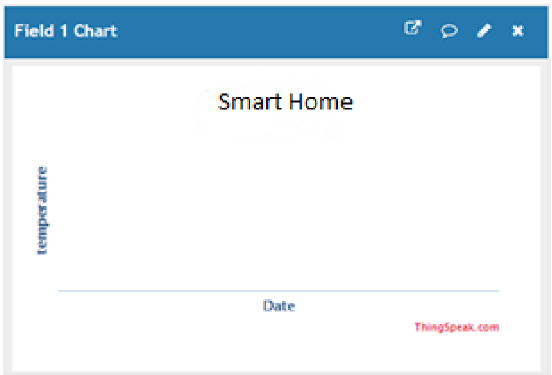
Input Channel name, Field1 , then click “Save Channel”

- Channel name: Smart Helmet
- Field 1: Temperature



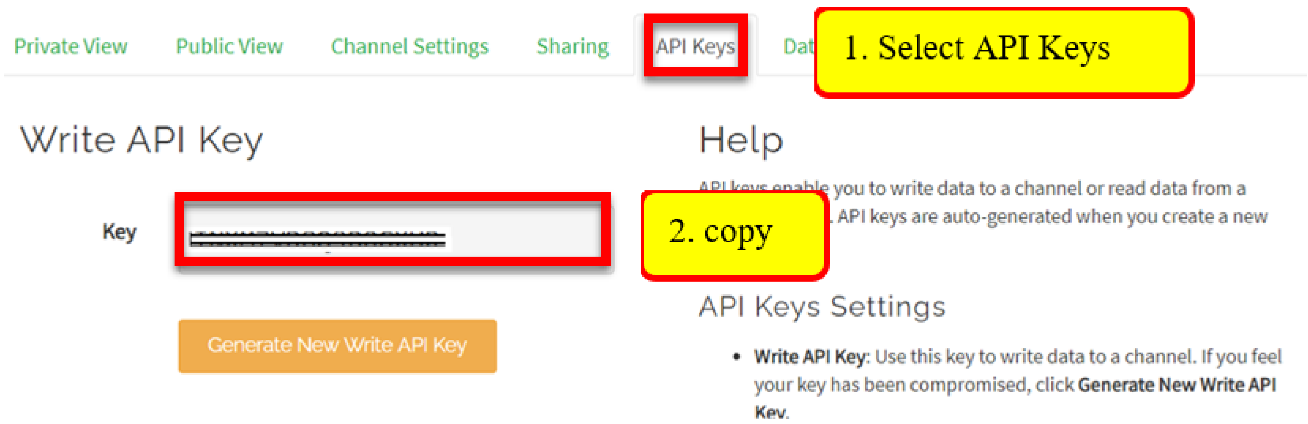
Step 4

You will see a chat for data field1



Step 5

Open your web browser, go to https://thingspeak.com , select your channel > “API Keys” , copy the API key as follows:



4.4 : BOM

Table 4.4.1 : BOM

Sr. No.	Components	Part Number	Quantity	Unit Cost	Total Cost
1	Arduino UNO	-	1	874 Rs.	874 Rs.
2	Arduino Nano	-	1	610 Rs.	610 Rs.
3	ZigBee module	XBee S2C Pro	2	2759 Rs	5518 Rs.
4	XBee USB adapter	CP2102	2	325 Rs.	650 Rs.
5	Male Omni Antenna	-	2	210 Rs.	420 Rs.
6	GSM	SIM900A	1	775 Rs.	775 Rs
7	GPS	SKG17A1	1	976 Rs.	976 Rs.
8	GPS Antenna Magnet	-	1	386 Rs.	386 Rs.
9	Temperature & Humidity Sensor	DHT-11	1	119 Rs.	119 Rs.
10	Air Quality Gas Sensor	MQ-135	1	145 Rs.	145 Rs.
11	Pulse Sensor	XD58c	1	161 Rs.	161 Rs.
12	IR Proximity Sensor	-	1	33 Rs.	33 Rs.
13	Vibartion Sensor	-	1	40 Rs.	40 Rs.
14	Buzzser	-	1	15 Rs.	15 Rs.
15	LED’s	-	6	3 Rs.	18 Rs.
16	UNO Cable	-	1	45 Rs.	45 Rs.
17	Nano Cable	-	1	45 Rs.	45 Rs.
18	Jumper Wires	-	30	90 Rs.	90 Rs.
19	Helmet	-	1	195 Rs.	195 Rs.
20	Breadboard		2	80 Rs	160 Rs.
21	Wires		1	100 Rs.	100 Rs.
Total					11,375 Rs.

Chapter 5

Experimentation and Results

5.1 Experimentation

The intelligent security system consists of a helmet, which is mounted with the sensor circuits. The transmitter section has a microcontroller which receives input from various sections like gas sensor, temperature sensor, humidity sensor, alert buttons and RF receiver. RF (Radio Frequency) receiver receives RF signals produced by RF transmitters fixed on various spots in the coal mine to identify the location of miner. These RF transmitters will emit RF signals containing corresponding location information. These signals are received by the RF receiver placed at the base station and it will decode this location information and fed into the arduino. For the simplicity to demonstrate, here in this project. On receiving a signal corresponding to a particular location, the arduino will be able to identify the position of the coal miner. The humidity, temperature, Pulse, GPS and gas sensors will sense the corresponding parameters. All the sensed parameters are displayed in an Screen which is at base station. By this arrangement, the coal miner itself will be able to know the environmental variations by the alerts and in case of any emergency, he can send alert messages to the outside world using the alert buttons. The sensed parameters or alerts along with the location information are communicated to the outside world through a ZigBee module. The receiver of the ZigBee is connected at the Arduino Nano terminal of the computer of the control room. The information send by the helmet are received by the control room at a fixed base station and displayed in a computer using Thingspeak software. Thingspeak is a open cloud sourcs. This software will detect the data sequence received through the XBee terminal and display it in the monitor. The computer will displays the values of temperature, humidity, gas and also the location of the miner. The Alert message is also displayed on the screen in case of emergency.

The below Fig5.1. shows the hardware kit to implement the smart helmet for coal miner's safety purpose The fig. 5.1 shows the main module which is the control unit of the system. Arduino UNO is used as the control unit. And Fig 5.2 shows the receiver side hardware kit.

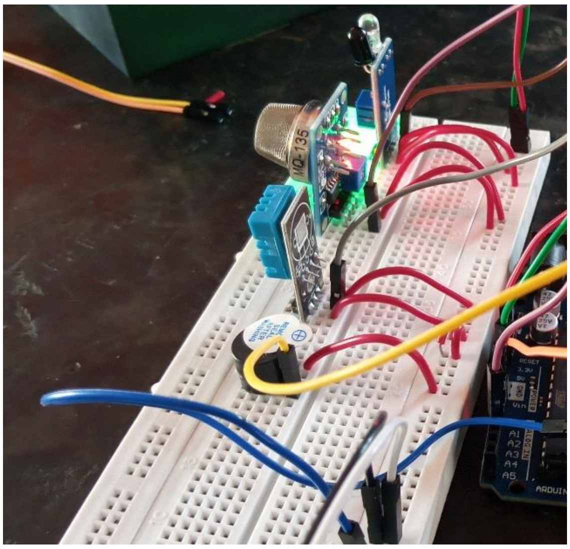


Fig 5.1 : Transmitter Section

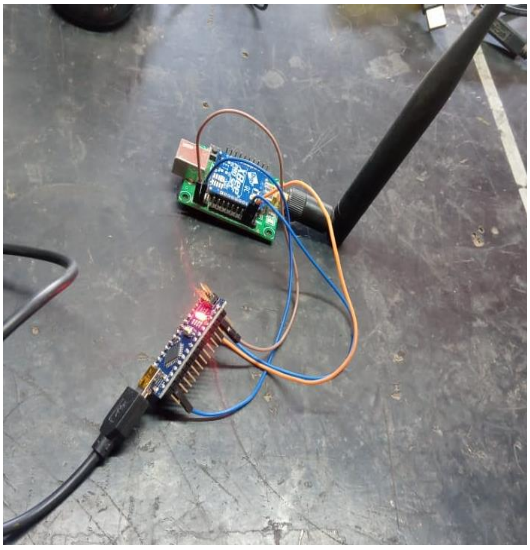


Fig 5.2. Receiver Section

The above kit consists of arduino board , pulse sensor, GPS, IR sensor, Buzzer, Vibration Sensor, the temperature & humidity sensor (DHT11), and the gas sensor (MQ-135), and the Zigbee transmitter. After implementation of code successfully, we see the results in readings in serial monitor and their respective values in form of Fig.5.3 in serial plotter.

Fig 5.3. Readings in serial monitor.

Here, we see the 6 parameters like temperature variations, humidity levels, smoke detection, reading pulses, obstacle detection and helmet remove or not. In the code, we set the temperature to 45 degrees. so, upto that temperature reading comes normally if they exceeds that 45 degrees they readings are change, and they give alert message like temperature is high and greater than 45°C at the same time in helmet also the buzzer, vibrator & indication led will on to alert the coal miner the temperature is high. As per the temperature increases, the humidity level also increases. if the smoke is not detected they show smoke is not detected, otherwise if ppm exceeds threshold they show smoke detected and they give alert to miner. If the any obstacle, roof collapse detectecd they show obstacle ahed . if the helmet is remove they show to person who work in base station in their respective PC. As well as the variations in humidity and increase level of harmful gases. they shows the graphs as per the extend the limit. Above Fig.5.3 shows the parameters variations in serial plotter, according the person going under the earth, the values are changes constantly as per the variations occur in earth.

Fig.5.4. Graph in Serial Plotter

All the other modules except RF transmitter and ZigBee receiver are connected to their boards. The results are viewed by an control room screen which is connected to the receiver section. The RF transmitter will send RF signals with a frequency of 434 MHz contains the location information from various part of the coal mine. By receiving this signal, the position of the miner can be located. A GPS is connected to its in order to provide the location information Manual emergency alerts for undesirable humidity, temperature, gas and accident alert are connected to the helmet.

Fig. 5.5 : System on Helmet

All the modules were tested separately and mounted on the helmet frame as shown in fig. 5.1 & 5.2 Communication up to a length of 8 to 9 meters was observed in the final test. The predetermined values and the obtained values are shown in Table 5.1. Temperature is expressed in degree Celsius and humidity and gas are expressed as percentages in air. The result obtained via the monitoring system has been analyzed and represented in Table 5.1 and Table 5.2, respectively, whereas results obtained are shown in fig 5.3. Whereas Table 5.1 describes the output of the monitoring system when conditions are under control, compares the output from all the sensors with their respective thresholds set and shows the hardware changes and their current state. Whereas table 5.2 describes output in an emergency situation and what hardware changes will take place in that situation. Fig.5.3. shows the actual readings obtained from the monitoring system prototype. Arduino UNO is programmed so that the readings of all the sensors are displayed on the user interface at an interval of 5 seconds.When the threshold value has crossed, warnings like 'high gas concentration', 'high humidity' and 'high temperature' are displayed.

Table 5.1 : Result analysis of monitoring system-a

Sr.No	**Parameters**	**Threshold Value Set**	**Sensor Readings**	**Hardware Changes**	**Remark**
1	Temperature ($°C$)	42	36	Buzzer, Vibrator, LED Off	Safe Condition
2	Humidity (% RH)	50	33	Buzzer, Vibrator, LED Off	Safe Condition
3	Gas (PPM)	600	518	Buzzer, Vibrator, LED Off	Safe Condition
4	IR Sensor (digi. output)	0	0	Buzzer, Vibrator, LED Off	Safe Condition

Table5.2 : Result analysis of monitoring system-b

Sr.No	**Parameters**	**Threshold Value Set**	**Sensor Readings**	**Hardware Changes**	**Remark**
1	Temperature ($°C$)	42	43.05	Buzzer, Vibrator, LED ON	Unsafe Condition
2	Humidity (% RH)	50	53.3	Buzzer, Vibrator, LED ON	Unsafe Condition
3	Gas (PPM)	600	694.1	Buzzer, Vibrator, LED ON	Unsafe Condition
4	IR Sensor (digi. output)	1	1	Buzzer, Vibrator, LED ON	Unsafe Condition

The serially sent data through ZigBee transmitter fixed in helmet was received by the ZigBee receiver in the control room. Results are displayed in a computer using Thingspeak software as shown in. Fig : 5.6 ThingSpeak Result

Fig. shows the operational flow of the monitoring system. Begin by connecting the prototype circuit to a power source. Now we'll have to keep track of all of the sensors' values attached to the Arduino UNO. Set a threshold value for each sensor after evaluating the readings and determining when environmental circumstances become aberrant. The condition will be true if the sensor output is greater than the defined threshold value, and a buzzer will sound and vibration & LED is on for indication, readings will be displayed on the control room screen with an alert message, and data will be transferred through Zigbee protocol to the control room. The Buzzer will not turn on if the sensor value is less than the defined threshold value, but the sensor readings will be shown on the screen and relayed to the control room. This is when the flow of the procedure comes to a stop.

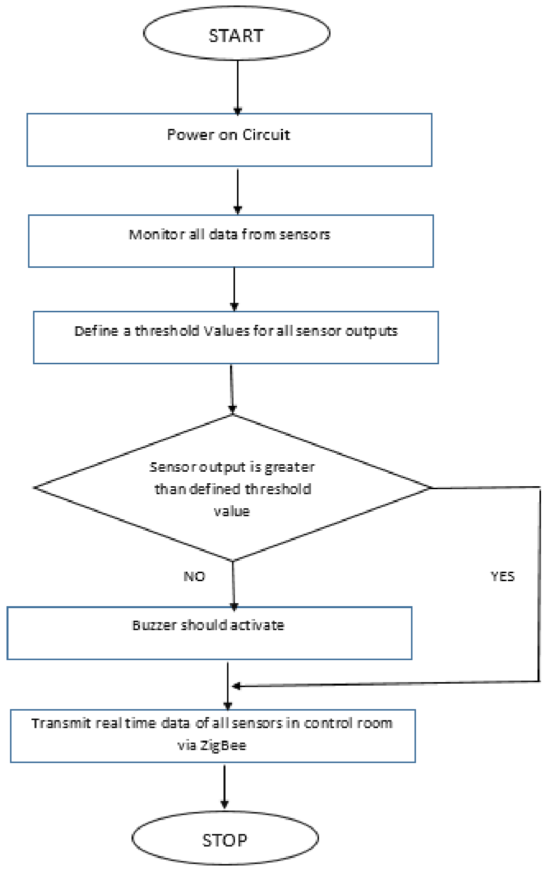
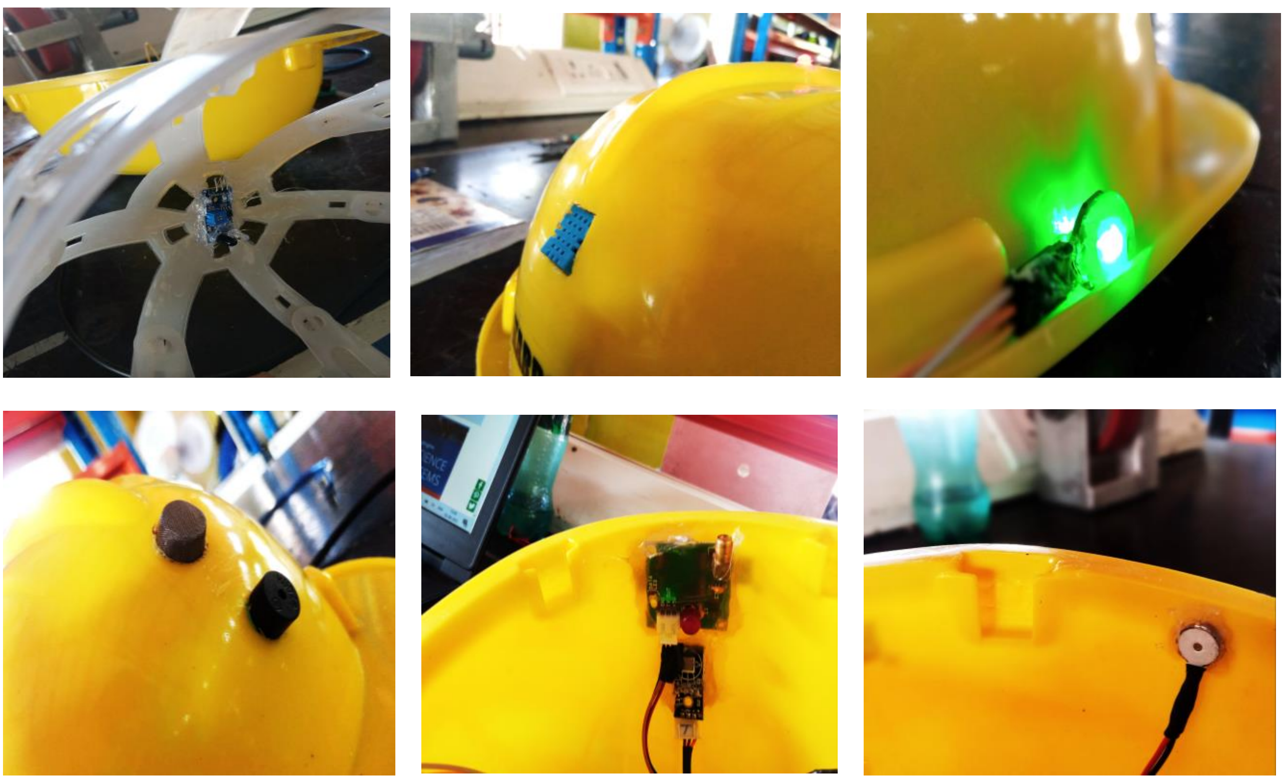


Fig. 5.7 : Flow Chart For Controller Programming

The following Fig. 3, shows the experimental setup of the proposed system. Here the Gas sensor (MQ-135), Humidity and temperature sensor (DHT11), pulse sensor, IR sensors and GPS are connected as an input to the Arduino and vibrator and buzzer as output.



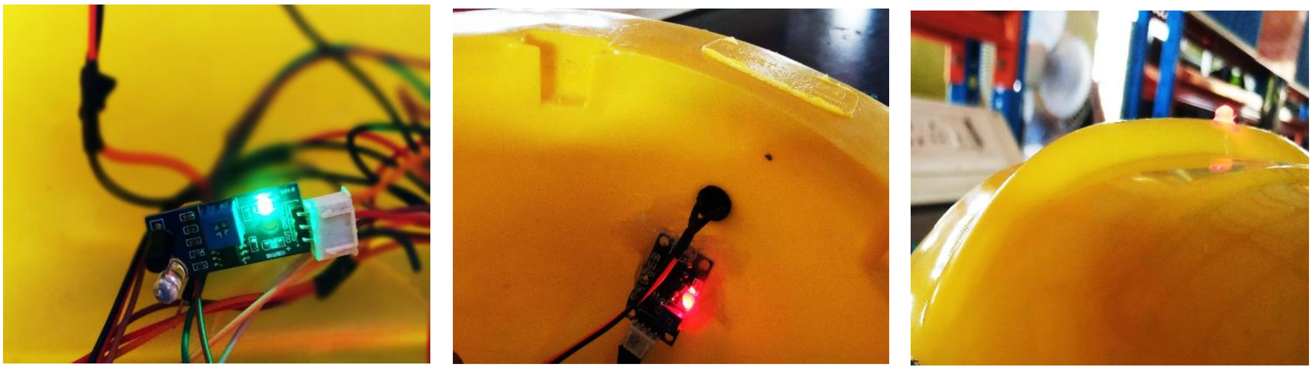
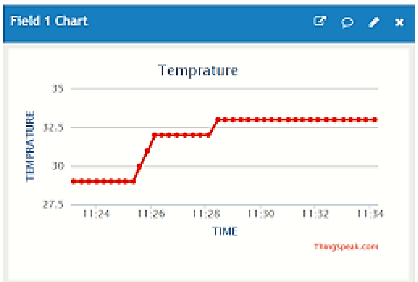
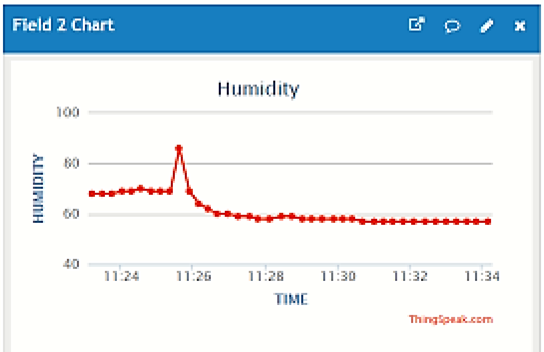


Fig. 3. Experimental setup

Web page showing temperature around miner The following Fig. 4, will show the temperature around the miner working underground in mine. The threshold value of the temperature is set at 50 degree Celsius. In the following chart the X-axis represent the time temperature in degree Celsius and Y-axis represents the temperature in degree Celsius.



Web page showing humidity around miner The following Fig. 5 will show the relative humidity around the miner working in underground mine. The threshold value of the humidity is set at is set at 90% RH. The X-axis represents the time and Y-axis represents the related humidity in percentage.



Web page showing gas leakages around miner The gas sensor i.e. MQ-2 sensor gives the digital output. In the following figure 6 when no gas detected then output will be 1. When gas detected then it will give the output 0. That is for normal condition the output of the sensor must be 1. In the following chart X-axis represent the time and Y-axis represent the gas value

5.2 Result

This section discusses the results of the proposed system. The sensors sense the environmental conditions around the miner working in underground mining. All the real time data is display on screen, and also updated on the web by using IoT with the help of Thing speak. If any of the environmental parameters exceeds its standard value the miner, co-miners, supervisor and the control station get notify by buzzer. If any hazardous event occurred in the mine in such case the control station will be able to provide the rescue team as early as possible.

All the components are assembled and tested successfully. The circuit is designed in such a manner that Set a threshold value for each sensor after evaluating the readings and determining when environmental circumstances become aberrant. The smart helmet alarms the miner if there is high range of toxic gases present in the atmosphere by buzzing an alarm with vibrator. if the smoke is not detected they show smoke is not detected, otherwise if ppm exceeds threshold they show smoke detected and they give alert to miner. If the any obstacle, roof collapse detectecd they show obstacle ahed . The condition will be true if the sensor output is greater than the defined threshold value, and a buzzer will sound and vibration & LED is on for indication, readings will be displayed on the control room screen with an alert message, and data will be transferred through Zigbee protocol to the control room. The Buzzer will not turn on if the sensor value is less than the defined threshold value,

Fig : Warnings and Indications

Thus, the experimental phase was completed. The study was conducted in a controlled manner. Thus, there is no pressing need for further experimentation in real life conditions but before full time deployment more simulations need to be performed.

Here are some of the advantages.

- Safety and improved monitoring in coal mines.
- Providing wireless connection security.

Chapter 6

Summary and Future Scope

6.1 : Summary

The final design and implementation of system gives 100% result in safety and to detect and monitoring sense of the undesirable environmental changes like Temperature & Humidity, Pulses, Location, Pressure and Methane gas and also to generate the range value by the IR sensor for obstacle avoidance inside the underground mining. The use of wireless zigbee technology improves the performance and efficiency for the data transmission to the control panel for the communication and alerts the security system. In this project, we were able to store the data that parameter in the PC, This stored data can will help to detect the hazards has they will happen before itself. This will help us to save the life of mining workers as much before, where the recuse team people staying at the control unit comes to save them.

6.2 : Scope for future work

A few aspects of the system can be improved. Adding an external antenna would extend the range or improve the signal strength in order to allow for more human interference. The distance might still want to be limited as it would be impractical to warn miners that are too far away to find the miner who is experiencing a hazardous event. The processing speed of the system can be improved to allow for more accurate measurement. The IR sensor can be improved to work within the helmet by not triggering because of reflections. Node hopping can be implemented to allow transmissions to the supervisor or even a central control station. This can be done by adding stationary nodes that are programmed to only bounce any signal that is received. The system can be improved by adding more measuring devices to check the miner's blood pressure and other. Gas concentrations can be measured as well. In future, it could also be considered if such modules can also be used for secondary services, such as localization of workers relative to each other. In future using additional sensors all possible safety issues could be monitored such as dust, vibration, water leakage etc. Also we can use number of salve and improve the data transmission distance. The control can be governed from the surface itself as the system provides easy access

References :

[1] Bolla Sravani,K Rambabu, A Smart and Secured Helmet for mining workers, International Journal of Advanced Research Trends in Engineering and Technology(IJARTET),Vol 4,Issue 3,March 2017.

[2] C.J.Behr, A.Kumar, G.P.Hancke,A Smart Helmet for Air Quality and Hazardous Event Detection for the Mining Industry,IEEE,2016.

[3] G.Ravi Kumar, B.Keerthi Reddy, Internet of Things Based an Intelligent Helmet for Wireless Sensor Network, International Journal of Engineering Sciences & Research Technology(IJESRT),June 2018.

[4] A.Dhanalakshmi, P.Lathapriya, K.Divya, A Smart Helmet for Improving Safety in Mining Industry, International Journal of Innovation Science and research Technology(IJISRT),Volume 2, Issue 3,March 2017.

[5] S.R.Deokar, V.M.Kulkarni, J.S.Wakode, Smart Helmet for Coal Mines Safety Monitoring and Alerting, International Journal of Advanced Research in Computer and Communication Engineering(IJARCCE),ISO 3297:2007 Certified, Vol 6, Issue 7,July 2017.

[6] Shruthi P.Borkar,V.B.Baru,IoT Based Smart Helmet for Underground Mines, International Journal of Research in Engineering, Science and Management(IJESM),Volume 1,Issue 9,Sept 2018.

[7] Punam S.Tajane, Shruthika B.Shelke, Sonal B.Sadgir, Archana N.Shelke, IoT Mining Tracking & Worker Safety Helmet, International Research Journal of Engineering and Technology(IRJET), Volume 7,Issue 4,Apr 2020.

[8] Pradeepkumar G, Usha S, Effective watermarking algorithm to protect Electronic Patient Record using image transform, 2013 International Conference on Information Communication and Embedded Systems (ICICES) Retracted , 2013, pp.1030-1034. doi: 10.1109/ICICES.2013.6508251.

[9] Pradeepkumar G, Chandra prasad V, and Rathanasabhapathy G,: Effective watermarking algorithm to protect Electronic Patient Record using DCT, International Journal of Software & Hardware Research in Engineering, Vol. 3 No. 11, 2015, pp. 16-19.

[10] Pradeepkumar G, et.al 2020 Investigation on Watermarking Algorithm for Secure Transaction of Electronic Patient Record by Hybrid Transform. Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 379-383, doi: 10.1109/I-SMAC49090.2020.9243411.

[11] M.Saari,A.Muzaffao bin Baharudin,P.Sillberg,S.Hyrynsalmi,W.Yan,LoRa-A Survey of Recent Research Trends,MIPRO 2018.

[12] Lone Kolobe,Boyce Sigweni,Caspar K Lebekwe,Systematic Literature Survey:Applications of LoRa Communication,International Journal of Electrical and Computer Engineering(IJECE),Vol 10,June 2020.

[13] D. Devikanniga, A. Ramu, and A. Haldorai, Efficient Diagnosis of Liver Disease using Support Vector Machine Optimized with Crows Search Algorithm, EAI Endorsed Transactions on Energy Web, p. 164177, Jul. 2018. doi:10.4108/eai.13-7-2018.164177

[14] H. Anandakumar and K. Umamaheswari, Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers, Cluster Computing, vol. 20, no. 2, pp. 1505–1515, Mar. 2017.

[15] Lluis Casals,Bernat Mir,Rafael Vidal,Carles Gomez, Modeling the Energy Performance of LoRaWAN, Sensors,Oct 2017.

[16] Phui San Cheong,Johan Bergs,Chris Hawinkel,Jereon Famaey, Comparison of LoRaWAN classes and their Power Consumption ,IEEE,2017. Retra

Weblinks :

https://docs.arduino.cc/hardware/uno-rev3

https://docs.arduino.cc/hardware/nano

https://www.digi.com/resources/documentation/digidocs/pdfs/90001500.pdf

https://robu.in/wp-content/uploads/2017/06/XBee-s2c-pro-User-Manual.pdf

file:///C:/Users/Admin/Desktop/sb/New%20folder/SIM900A_SIMCom.pdf

https://simcom.ee/documents/SIM900/SIM900_Hardware%20Design_V2.05.pdf

https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/

https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf

https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf

https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

https://components101.com/sensors/ir-sensor-module

https://manualzz.com/doc/33104912/skynav-skg17a1-ultra-high-sensitivity-and-low-power-gps

https://create.arduino.cc/projecthub/amalmathewtech/arduino-gps-module-destination-notifier-288a55

https://components101.com/sensors/pulse-sensor

List of Figures

Fig. 3.2.1 : Transmitter Section……………………………………………………………………………………..9

Fig. 3.2.2 : Receiver Section……………………………………………………………………………………..9

Fig 3.3.1 : Prototypes of Helmet…………………………………………………………………………………...10

Fig. 3.3.2: Flow chart of the monitoring System for Sensor Unit…………………………………………...……….11

Fig 3.3.3 : Flow chart of the monitoring System for Monitoring Unit………………………………………….…….12

Fig 3.3.3 : Breadboard Connection……………………………………………………………………………….....13

Fig 3.3.4 : Complete System………………………………………………………………………………….14

Fig. 4.1.1 : Arduino UNO……………………………………………………………………………………………..17

Fig. 4.1.2 : Arduino UNO Pin Diagram……………………………………………………………………………….18

Fig. 4.1.3 : Arduino UNO Connector Pinouts…………………………………………………………………………22

Fig. 4.2.1: Arduino NANO…………………………………………………………………………………...…….23

Fig. 4.2.2 : Arduino NANO Pi Diagram………………………………………………………………………………24

Fig. 4.2.3 : Arduino NANO Connector Pinouts……………………………………………………………...…….27

Fig. 4.3.1 : Gas Sensor (MQ135)……………………………………………………………………………...……29

Fig. 4.3.2 : Pin Configuration……………………………………………………………………………………....31

Fig. 4.3.3 : PPM Values Graph……………………………………………………………………………………..…33

Fig. 4.3.4 : Interfacing of MQ-135 gas sensor with Arduino :………………………………………………….……34

Fig. 4.4.1 : Temperature And Humidity Sensor : DHT11…………………………………………………….…...36

Fig 4.4.2 : Pin Configuration…………………………………………………………………………………….…...37

Fig. 4.4.3 : Connection Diagram : DHT11…………………………………………………………………………….37

Fig. 4.4.4 : DHT11 Pinouts……………………………………………………………………………………………41

Fig. 4.4.5 : DHT11 Sensor Interfacing with Arduino UNO…………………………………………………..………41

Fig 4.5.1 : Heartbeat/Pulse Sensor : XD58c……………………………………………………………..……………42

Fig 4.5.2 : XD58c Pin Configuratio……………………………………………………………...……………………43

Fig 4.5.3 : Hardware Overview……………………………………………………………………………………. 44

Fig 4.5.4 : Hardware Overview ……………………………………………………………………………………….45

Fig 4.5.5 : Interfacing Pulse Sensor with Arduino ……………………………………………………………………46

Fig. 4.6.1 : IR Sensor/Infrared Sensor ……………………………………………………………………………..47

Fig. 4.6.2 : Pin Configuration………………………………………………………………………………………… 48

Fig. 4.6.3 : IR sensor Module Inside ………………………………………………………………………………….49

Fig. 4.6.4 : Interfacing IR Sensor with Arduino……………………………………………………………………… 50

Fig. 4.7.1 : GSM – SIM900A …………………………………………………………………………………………51

Fig. 4.7.2 : SIM900A Functional Diagram…………………………………………………………………………… 53

Fig. 4.7.3 : SIM900A Module ………………………………….………………………………………………...54
Fig. 4.7.4 : Interfacing Uno And Gsm Shield …………………………………………………………………………60
Fig 4.8.1 : GPS : SKG17A1……………………………………………………………………………………….. 60
Fig 4.8.2: Interfacing GPS Sensor with ………………………………………………………………………………62
Fig. 4.9.1 : XBee Pro S2C ……………………………………………………………………………………………63
Fig. 4.9.2 : XBee Pin ConfigurationFig…………………………………………………………………………… 64
Fig. 4.9.3 : Star Topology ……………………………………………………………………………………………68
Fig. 4.9.4 : Mesh Topology …………………………………………………………………………………………...69
Fig. 4.9.5 : Tree Topology ……………………………………………………………………………………………69
Fig. 4.9.6 :Natework Layers………………………………………………………………………………………. 71
Fig. 4.9.7: Interfacing XBee with Arduino …………………………………………………………………………71
Fig. 4.10.1 : Vibration Motor ………………………………………………………………………………………73
Fig. 4.11.1 : Buzzer ……………………………………………………………………………………………...…74
Fig. 4.12.1 : Zigbee USB Interfacing Board …………………………………………………………………………75
Fig 4.1.1 installation Process ………………………………………………………………………………………76
Fig 4.1.2 Arduino Connection ……………………………………………………………………………………..77
Fig 4.1.3 : Arduino IDE ……………………………………………………………………………………………77
Fig 4.1.4 : Selection of Board ………………………………………………………………………………………77
Fig 4.1.5: Selection of Serial Port ……………………………………………………………………….……78
Fig 4.1.6: Arduino Example Codes …………………………………………………………………………….....78
Fig 4.1.7 : Example Code ……………………………………………………………………………………………..79
Fig 4.1.8: code uploading done ……………………………………………………………………………………80
Fig. 4.2.1: Select COM ports ………………………………………………………………………………………83
Fig. 4.2.2: Set Port Parameters ……………………………………………………………………………………..83
Fig. 4.2.3: Discovering Radio Modules ……………………………………………………………………………84
Fig. 4.2.4 : Add Selected devices …………………………………………………………………………………..84
Fig. 4.2.5 : Configuration of Coordinator …………………………………………………………………………84
Fig. 4.2.6 : Parameter Setting ………………………………………………………………………………………85
Fig. 4.2.7 : Radio Configuration – Networking 1 ……………………………………………………………………86
Fig. 4.2.8 : Radio Configuration – Networking 2 ……………………………………………………………………87
Fig. 4.2.9 : Writing Radio Module Setting …………………………………………………………………………87
Fig. 4.2.10 : Configuration of End Device …………………………………………………………………………88
Fig. 4.2.11 : Radio Configuration – Networking……………………………………………………………….. 88
Fig. 4.2.12 : Radio Configuration Setting…………………………………………………………………………. 89
Fig. 4.2.13 : Writing Radio Module Setting …………………………………………………………………………90
Fig. 4.2.14 : Another XCTU …………………………………………………………………………………………91

Fig. 4.2.15 : Discover Radio Devices ……………………………………………….………………......................91

Fig. 4.2.16 : Discover Radio Modules ……………………………………………………………………………….91

Fig. 4.2.17 : Configuration of device ………………………………………………………………………………92

Fig. 4.2.18 : Open Terminal ………………………………………………………………………………………….92

Fig. 4.2.19 : Open Serial Connection ……………………………………………………………………………...93

Fig. 4.2.20 : XBee Communication ……………………………………………………………………………...93

Fig. 4.2.21 : Firmware Updation ……………………………………………………………………………………94

Fig. 4.2.22 : Radio Config 1 …………………………………………………………………………………………94

Fig. 4.2.23 : Radio Config 2 …………………………………………………………………………………………95

Fig. 4.3.1 : ThingSpeak Configuration ……………………………………………………………………………100

Table 4.4.1 : BOM ……………………………………………………………………………………….………..96

Fig 5.1 : Transmitter Section ………………………………………………………………………….…….99

Fig 5.2. Receiver Section ……………………………………………………………………………….….……99

Fig 5.3. Readings in serial monitor. …………………………………………………………………….….……….100

Fig.5.4. Graph in Serial Plotter………………………………………………………………………….………..102

Fig 5.5 : System on Helmet ………………………………………………………………………….…………..103

Fig 5.6 : ThingSpeak Result……………………………………………………………….………………...102

Fig. 5.7 : Flow Chart For Controller Programming ………………………………………………………….…..102

List of Tables

Table 3.4.1 : Task vs Time Schedule…………………………………………………………………………….14

Table. 4.1.1 : Arduino UNO Analog Pins………………………………………………………………………..21

Table 4.1.2 : Arduino UNO Digital Pins…………………………………………………………………….....22

Table 4.2.1 : Arduino NANO JMicro USB/Power Pins…………………………………………………………….27

Table 4.2.2 : Arduino NANO Digi Pins……………………………………………………………………………….28

Table 4.2.3 : Arduino NANO Analog Pins…………………………………………………………………………….28

Table 4.3.1 : Technicle Parameters……………………………………………………………………...….29

Table 4.3.2 : Specification………………………………………………………………………………………..30

Table 4.3.3 : Pin Configuration……………………………………………………………………….……31

Table. 4.4.1 : Technicle Parameters……………………………………………………………………………...35

Table. 4.4.2 : Detailed Specification……………………………………………………………………………..36

Table. 4.4.3 : Pin Configuration……………………………………………………………………………….37

Table 4.5.1 ; Technicle Paarameters ………………………………………………………………………….42

Table 4.5.2 ; Pin Configuration ……………………………………………………………………………43

Table 4,6.1 : Technicle Parameters …………………………………………………………………………47

Table 4,6.2 : Pin Configuration…………………………………………………….……………… 48

Table 4.7.1 : Technical Parameters ……………………………………………………….………………..51

Table 4.7.2 : Power Supplying Modes ……………………………………………………….………………….54

Table 4.8.1 : Specification ……………………………………………………….…………………………59

Table 4.8.2 : Technical Parameters………………………………………………..……………………….. 60

Table 4.8.3 : Electrical Characteristics ……………………………………….……………………………60

Table 4.9.1 : Zigbee Performance……………………………………….…………………………………… 63

Table 4.9.2 : Features……………………………………………………………………………………… 63

Table 4.9.3 : Networkingg & Security……………………………………………………………………….. 64

Table 4.9.4 : Power Requirements ……………………………………………………………………………64

Table 4.9.5 : Regultory Approvals ……………………………………………………………………………64

Table 4.9.6 : Pin Functions ……………………………………………………………………………………65

Table 5.1 : Result analysis of monitoring system-a ……………………………………………………………101

Table5.2 : Result analysis of monitoring system-b……………………………………………………………..101

Sensors Interfacing Codes

Code for interfacing of MQ-135 gas sensor with Arduino :

```
#include <MQ135.h>
#define PIN_MQ135 A2

MQ135 mq135_sensor(PIN_MQ135);
float temperature = 21.0; //Assume current temperature. Recommended to measure with DHT22
float humidity = 25.0; //Assume current humidity. Recommended to measure with DHT22

void setup()  //the setup routine runs over and over again forever.
{
  Serial.begin(9600); //initialize serial communication at 9600 bits per second
}
void loop()  //the loop routine runs over and over again forever.
{
  float rzero = mq135_sensor.getRZero();
  float correctedRZero = mq135_sensor.getCorrectedRZero(temperature, humidity);
  float resistance = mq135_sensor.getResistance();
  float ppm = mq135_sensor.getPPM();
  float correctedPPM = mq135_sensor.getCorrectedPPM(temperature, humidity);

  Serial.print("MQ135 RZero: ");
  Serial.print(rzero);
  Serial.print("\t Corrected RZero: ");
  Serial.print(correctedRZero);
  Serial.print("\t Resistance: ");
  Serial.print(resistance);
  Serial.print("\t PPM: ");
  Serial.print(ppm);
  Serial.print("\t Corrected PPM: ");
  Serial.print(correctedPPM);
  Serial.println("ppm");

  delay(1000);  // wait 1000ms for next reading
}
```

Code for interfacing of DHT11 sensor with Arduino :

```
#include "DHT.h"
DHT dht;

void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");

  dht.setup(2); // Digital pin connected to the DHT sensor
}

void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); //Read temperature
  float temperature = dht.getTemperature(); //Read humidity


  Serial.print(dht.getStatusString()); //Print status of communication
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature),1); //Convert temperature
to Fahrenheit

  delay(3000); // wait three seconds
}
```

Code of GPS :

```
#include  <TinyGPS++.h>
#include  <SoftwareSerial.h>
 static const int RXPin = 4, TXPin = 3;
 static const uint32_t GPSBaud = 9600;
 TinyGPSPlus gps;     // The TinyGPS++ object
 SoftwareSerial ss(RXPin, TXPin); // The serial connection to the GPS device


void setup(){
 Serial.begin(9600);
 ss.begin(GPSBaud);
}
void loop(){ // This sketch displays information every time a new sentence
is correctly encoded.
 while (ss.available() > 0){
   gps.encode(ss.read());
   if (gps.location.isUpdated()){
     Serial.print("Latitude= ");
     Serial.print(gps.location.lat(), 6);
     Serial.print(" Longitude= ");
     Serial.println(gps.location.lng(), 6);
   }
 }
}
```

Code for interfacing of Pulse sensor with Arduino :

```
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.

#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0

const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.

int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.

                               // Otherwise leave the default "550" value.

PulseSensorPlayground pulseSensor;  // Creates an instance of the PulseSensorPlayground object called "pulseSensor"

void setup() {

  Serial.begin(9600);          // For Serial Monitor

  // Configure the PulseSensor object, by assigning our variables to it.

  pulseSensor.analogInput(PulseWire);

  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino's LED with heartbeat.

  pulseSensor.setThreshold(Threshold);

  // Double-check the "pulseSensor" object was created and "began" seeing a signal.

   if (pulseSensor.begin()) {

    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.

  }

}

void loop() {

 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".

                                               // "myBPM" hold this BPM value now.

if (pulseSensor.sawStartOfBeat()){ // Constantly test to see if "a beat happened".

 Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".

 Serial.print("BPM: ");                        // Print phrase "BPM: "

 Serial.println(myBPM);                        // Print the value inside of myBPM.

}

  delay(20);                    // considered best practice in a simple sketch.

}
```