VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590014, Karnataka, INDIA

PROJECT REPORT
On

"IOT FOR INDUSTRIAL APPLICATION"

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering
In

ELECTRONICS AND COMMUNICATION ENGINEERING

For the Academic Year
2015-2016

BY

MIDHUN.T.S	**1PE12EC087**
OGETY SAI VAMSI	**1PE12EC103**
CHAITRA.A.S	**1PE13EC401**
AMIT KUMAR	**1PE12EC009**

UNDER THE GUIDANCE OF
Mr. K. PATTABHI RAMAN
Associate Professor
Dept. of ECE, PESIT (BSC).

Department of Electronics and Communication Engineering
PESIT - Bangalore South Campus
Hosur Road, Bangalore-560100

PESIT - Bangalore South Campus

Hosur Road, Bangalore-560100

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project work entitled "IOT FOR INDUSTRIAL APPLICATION" carried out by MIDHUN.T.S, OGETY SAI VAMSI, CHAITRA.A.S, AMIT KUMAR, bearing USNs 1PE12EC087, 1PE12EC103, 1PE13EC401 1PE12EC009, respectively in partial fulfillment for the award of Degree of Bachelors (Bachelors of Engineering) in Electronics and communication Engineering of Visvesvaraya Technological University, Belgaum during the year 2015-2016.

It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the Report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for said degree.

____________________	____________________	____________________
Signature of guide	**Signature of HOD**	**Signature of the Principal**
Mr. K.Pattabhi Raman	**Dr. Subhash Kulkarni**	**Dr. J Surya Prasad**
Associate Professor	**Head Of Dept.**	**Principal / Director**
Dept. of ECE	**Dept. of ECE**	**PESIT- BSC**

External Viva

Name of the Examiners	**Signature with date**
1	____________________
2.	____________________

ABSTRACT

The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services.

The driving philosophy behind IOT in industry is that smart machines are better than humans at accuracy, consistently capturing and communicating data. This data can enable companies to pick up on inefficiencies and problems sooner, saving time and money and supporting business intelligence efforts. In manufacturing specifically, IOT holds great potential for quality control, sustainable and green practices, supply chain traceability and overall supply chain efficiency.

Our Project is designed to have two stations one is Remote station and the other is Base station Remote station is the one which will be present in industrial environment. And Base station is the one which will be inside control room which will receive the data from different Remote stations and sends that data to Cloud. Both Base station and Remote station uses wireless communication between them. Our Mobile application allows the user to remotely monitor and Control industrial equipments.

The main objective is to implement the concept of IOT in industry and further improvements can be made depending on the application

ACKNOWLEGEMENT

We would like express our sincere gratitude to all the lecturers and staff of the department of Electronics and Communication Engineering for extending their help and guidance towards our project.

We would like to thank the college management and express our sincere gratitude to **Dr. J. Surya Prasad,** Director/Principal of PESIT(BSC) have given me the opportunity for the completion of this project.

We would like to thank **Dr. Subhash Kulkarni,** Head of Department Electronics and Communication, PESIT(BSC) for giving us the support and encouragement that was necessary for the completion of this report.

We convey our thanks to the project coordinators **Mr. Ramachandrula Sastry** Assistant Professor, **Rashmi R Kulkarni** Assistant Professor, **Radha.K.N** Assistant Professor and **Mr. K.Pattabhi Raman**,Associate Professor for providing us all the needful information and facilities which was of a great help to complete this project successfully.

We would like to thank our project guide **Mr. K.Pattabhi Raman ,**Associate Professor for providing us required assistance, encouragement and constant support which helped us a lot.

Last but not least the project would not have been a success without the support of our parents and friends.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Chapter 1

PREAMBLE

CHAPTER 1

PREAMBLE

1.1 INTRODUCTION

The **Internet of Things** (IOT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. The Internet of Things is increasing the connectedness of people and things on a scale that once was unimaginable. Connected devices outnumber the world's population by 1.5 to 1. The pace of IOT market adoption is accelerating because of:

- Growth in analytics and cloud computing
- Increasing interconnectivity of machines and personal smart devices
- The proliferation of applications connecting supply chains, partners, and customers.

The IOT vision is a massively instrumented world of intelligent sensors and actuators improving performance and efficiency. IOT will streamline, collapse, and create system architectures that are more affordable, responsive, and effective. IOT eliminates the need for expensive and difficult to maintain middle-level automation software.

1.2 NEED FOR IOT IN INDUSTRIES

The high level goal is to improve manufacturing processes in a number of dimensions, including efficiency, responsiveness, and the ability to meet individual customer needs in a timely manner. Mass production means the larger the quantity produced, the cheaper the run, assuming you can sell the products. One of the reasons mass production has lasted so long is the technology has not been available to create responsive make-to-order manufacturing. Some companies tried in the 1980s with the idea of "digital manufacturing" and computer-integrated manufacturing concepts. Unfortunately, the concepts could not be accomplished with the technology available at the time. Today the technology is becoming available and the economics are making it feasible to accomplish mass customization. Future manufacturing systems will be able to produce inexpensive, personalized, quantity-one orders.

Making smart phones that users can customize with apps is far simpler than making other more complex products. Complex products will require the creation of flexible automation, flexible machines, and manufacturing systems integrated with business systems and the consumers. The first applications of IOT and analytics in manufacturing are overlays to existing automation systems and manufacturing machines. The next step to fully exploit flexible make-to-order manufacturing is a much bigger undertaking that will require the retrofit or replacement of machines and equipment. These changes are fundamental and expensive but may be required for companies to remain competitive. This type of change will radically change automation systems and manufacturing plant floors.

1.3 BASIC BLOCK DIAGRAM

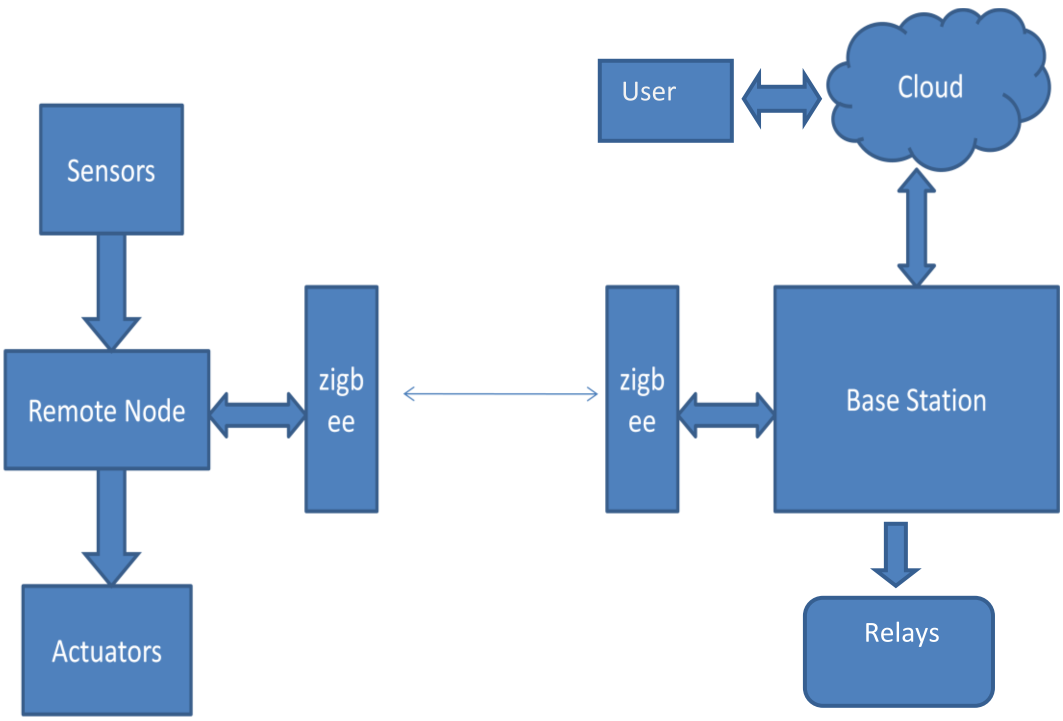


Figure 1.3.Basic Block Diagram

The basic block diagram consist of the main blocks such as

- Remote Node
- Base station
- Cloud
- User interface

The **Remote node** contains the sensors and actuators which are wirelessly connected to the base station.

The **Base station** enables the internet connectivity. The base station is wirelessly connected remote station wirelessly. The base station receives the data from the remote node and sends it to the cloud service.

The **Cloud service** allows the storage of data that is send by the base station and helps in the analysis of the data sent.

The **User** can view his data on the cloud through his pc or mobile phone and control the devices present with the help of mobile app.

1.4 APPLICATIONS OF IOT IN INDUSTRIES

- Smart manufacturing
- Predictive maintenance
- Employee safety
- Flow optimization
- Asset tracking
- Data monitoring

Chapter 2

PROJECT PLANNING

CHAPTER 2

PROJECT PLANNING

The project planning is an important aspect of a project, only with which the smooth running of project can be guaranteed. The work is divided between all the team members and coordination is achieved. The work division and project schedule is explained with the Gantt chart.

2.1 ACTIVITIES AND GANTT CHART

The detailed project planning was done using the Gantt chart by using the software called

“GANTT project”.

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.

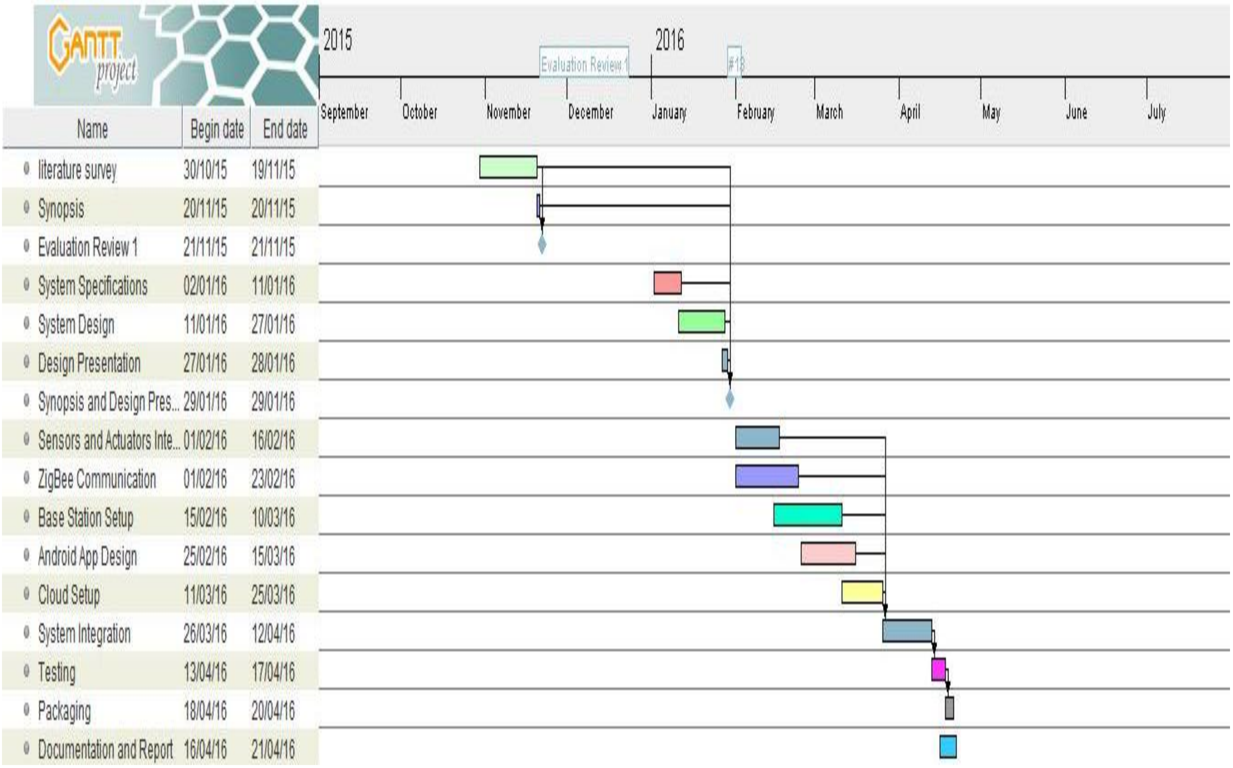


Figure 2.1.GANTT Chart

For our project we have divided the project into multiple tasks that has to be completed in the specified time. The different tasks were assigned to different members of our team so that the project can be finished efficiently in the given time and also learn team work among the group members.

2.2 WORK DIVISION

SL.NO	WORK	MEMBERS
1.	Literature Survey • Study on products in the market • Availability of sensors and actuators • A search on available platforms for IOT.	Amit , Vamsi , Midhun, Chaitra
2.	• Synopsis	Amit, Vamsi, Midhun
3.	• System Specification and system Design	Vamsi, Amit, Midhun, Chaitra
4.	• Sensors interfacing	Chaitra, Amit
5.	• Zigbee communication	Vamsi, Midhun
6.	• Base station setup	Vamsi, Midhun,
7.	• Android App Design	Amit, Chaitra
8.	• Cloud Setup	Midhun, Vamsi, Chaitra
9.	• System Integration	Vamsi, Amit, Midhun, Chaitra
10.	• Testing	Amit, Chaitra
11.	• Packaging	Chaitra, Amit
12.	• Documentation and Report	Amit, Midhun, Chaitra, Vamsi

Chapter 3

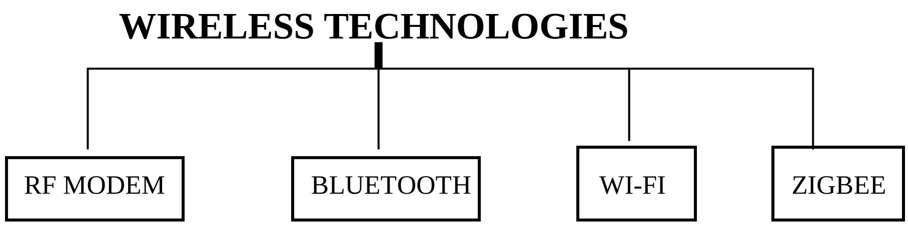
LITERATURE SURVEY

CHAPTER 3

LITERATURE SURVEY

3.1 DIFFERENT WIRELESS TECHNOLOGIES

Wireless technologies have been available for decades, and in the last several years wireless technologies have been making their way to the factory floor. This technology has been allowed for transferring data efficiently over long distances. When this wireless technology is utilized, a new world of applications is available to designers. Sensors and actuators can be added to many industrial, commercial, and entertainment automation applications.



3.1.1 RF MODULE

An RF module (radio frequency module) is a small electronic circuit used to transmit and receive radio signals on one of a number of carrier frequencies. RF modules are widely used in electronic and designing radio circuitry. RF modules are most often applied in medium and low volume products depending on consumer applications such as wireless alarm systems, industrial remote controls, smart sensor applications, and wireless home automation systems. Commercially available RF module frequencies are 433.92 MHz, 315 MHz, 868 MHz and 915MHz. These frequencies are used because of national and international regulations governing the use of radio for communication.

3.1.2 BLUETOOTH

For industrial applications, a wireless technology has to work well in a noisy environment. Bluetooth operates based on the features of Adaptive Frequency Hopping

(AFH) and Forward Error Correction (FEC). It provides a universal short range wireless capability. It operates in the 2.4 GHz frequency band and the devices within 10m of each other can share the data up to 720Kbps of capacity. This technology is also an authenticated one by sending the acknowledgement from the receiver to the transmitter before making the connection between devices. But its limitation is up to eight devices can communicate in a single network and it asks the confirmation about receiving the each data at every time and also it limits the packet size.

3.1.3 Wi-Fi

Wi-Fi stands for Wireless Fidelity, which refers to wireless technology that allows devices to communicate over a wireless signal. This network is based on the IEEE standard 802.11; including 802.11a, 802.11b, 802.11g and 802.11n, by using the centralized router devices can share the Wi-Fi signal. Wi-Fi networking technology that uses waves to allow high speed data transfer over short distances. In indoor environment, this technology causing problem called multipath interference due to reflection of signals from the walls, furniture and other obstacles. Wi-Fi allows local area networks (LANs) to operate without cable and wiring. It is popular for the home and business networks. Generally, it can be used to provide the wireless broadband internet access for many modern devices such as laptops, smart phones, tablet and computers with authentication. By increasing the number of devices in a single Wi-Fi connection, the strength of the signal provides to each device becomes weak.

3.1.4 ZIGBEE

The communication layer of Zigbee is at level 3 and upper layer in the OSI model. Zigbee provides a network topology to let a network of devices communicate between them and to set extra communication features such as authentication, encryption, and the association and in the upper layer application services. A reactive Adhoc protocol has been implemented to perform the data routing and forwarding process to any node in the network. The main application of Zigbee is clustering. Zigbee has a lot to offer in industrial applications such as low cost deployment and redeployment, mesh networking to cover entire industrial plants and factories, an open standard with multiple vendors, battery operation.

3.2 INDUSTRIAL WIRELESS - SELECTING A WIRELESS TECHNOLOGY

Industrial environments are uniquely different from office and home environments. High temperatures, excessive airborne particulates, multiple obstacles and long distances separating equipment and systems, are special challenges that make it difficult to place and reach sensors, transmitters, and other data communication devices. These - and a thousand other factors - create a very unique, complex, and costly challenge for establishing data communication channels that will be reliable, long lasting,. and cost effective.For example, a primary difficulty faced by many companies, is the need to connect remote equipment sensors to central monitoring systems. Inside a steel mill, the environment is extremely intense: excessive heat, heavy machinery, large distances, and high levels of EMI significantly shorten the lifespan of wires and network equipment.

3.2.1 HISTORICAL CHALLENGES WITH INDUSTRIAL WIRELESS

- Signal Echo
- Noise
- Channel Sharing and Interference
- Industrial Protocols Not Supported
- Distance

3.2.2 CERTAIN NEW METHODS AND TECHNOLOGIES SOLVE WIRELESS ISSUES

Today, there is a wide array of data communication solutions that could resolve challenges facing industrial environments. Several transmission and modulation schemes have been developed to counter the effects of echo, noise, and channel sharing. Such as

- **FHSS** (Frequency Hopping Spread Spectrum) - Data is transmitted on a single channel at a time, but the channel is rapidly and constantly changing or "hopping".

- **DSSS** (Direct Sequence Spread Spectrum) - Data is transmitted simultaneously over every available channel, making it a bit more reliable in "noisy" environments, but is also bandwidth intensive.

3.3 IOT DEVICES AVAILABLE IN MARKET

- BROADCOM'S WICED Sense

Figure 3.3.1 Broadcom's WICED Sense

Includes multiple low-power MEMS sensors (gyroscope, accelerometer, eCompass, pressure, humidity and temperature) and Bluetooth Smart radio, powered by a single coin cell battery.

No flexibility for adding extra sensors to meet intended applications.

- INTEL EDISON

Figure 3.3.2 Intel Edison board

This kit includes an Arduino breakout board, to give your Intel Edison compute module the ability to interface with Arduino shields or any board with the Arduino footprint.

- TEXAS INSTRUMENTS single chip MCU with built in WIFI Module and launch board

Figure 3.3.3 Texas Instruments board

CC3200 SimpleLink Wi-Fi—Consists of Applications Microcontroller, Wi-Fi Network Processor, and Power-Management Subsystems.

Applications Microcontroller Subsystem-ARM® Cortex®-M4 Core at 80 MHz

- RASPBERRY PI 2

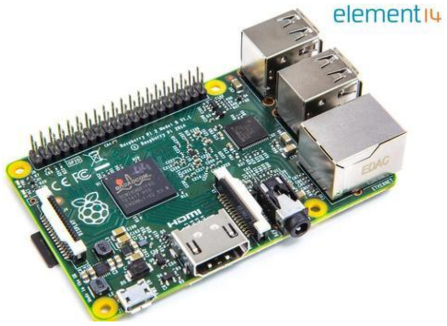


Figure 3.3.4 Raspberry PI 2 board

Microsoft and electronic firm Adafruit Industries have teamed up to deliver a Windows IoT Core Starter Kit based on the popular Raspberry Pi 2 single board computer, making it easier for users to get started building projects.

- ARDUINO YUN

Figure 3.3.5 Arduino Yun

The Arduino Yún is a microcontroller board based on the ATmega32u4 and the Atheros AR9331. The Atheros processor supports a Linux distribution based on OpenWrt named OpenWrt-Yun.

Chapter 4

ZIGBEE WIRELESS STANDARD

CHAPTER 4

ZIGBEE WIRELESS STANDARD

ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. The ZigBee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz. The 802.15.4 specification upon which the ZigBee stack operates gained ratification by the Institute of Electrical and Electronics Engineers (IEEE) in 2003. The specification is a packet-based radio protocol intended for low-cost, battery-operated devices. The protocol allows devices to communicate in a variety of network topologies and can have battery life lasting several years.

MESH NETWORKS

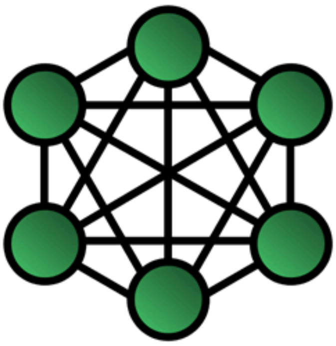


Figure 4 Mesh Network

A key component of the ZigBee protocol is the ability to support mesh networking. In a mesh network, nodes are interconnected with other nodes so that multiple pathways connect each node. Connections between nodes are dynamically updated and optimized through sophisticated, built-in mesh routing table. Mesh networks are decentralized in nature; each node is capable of self-discovery on the network. Also, as nodes leave the network, the mesh topology allows the nodes to reconfigure routing paths based on the new network structure. The characteristics of mesh topology and ad-hoc routing provide greater stability in changing conditions or failure at single nodes.

ZigBee Nodes

The ZigBee protocol defines three types of nodes: Coordinators, Routers, and End Devices, with a requirement of one Coordinator per network. While all nodes can send and receive data, there are differences in the specific roles they play.

Coordinators

Coordinators are the most capable of the three node types. There is exactly one coordinator in each network and it is the device that establishes the network originally. A coordinator store information about the network, including security keys, but it cannot be battery-powered (that is, it cannot sleep) because it must store the packets for end device nodes.

Routers

Routers act as intermediate nodes, relaying data from other devices. Routers can connect to other devices and with the coordinator, but, as with the coordinator, they cannot sleep because they must store the packets for end devices. Routers are commonly used to extend the network.

End devices

End Devices, usually sensors, can be low-power / battery-powered devices. They have sufficient functionality to talk to their parent devices, either the coordinator or a router, and cannot relay data from other devices. This reduced functionality allows for the potential to reduce cost.

Example ZigBee nodes

The following figure displays a topology example of a ZigBee network.

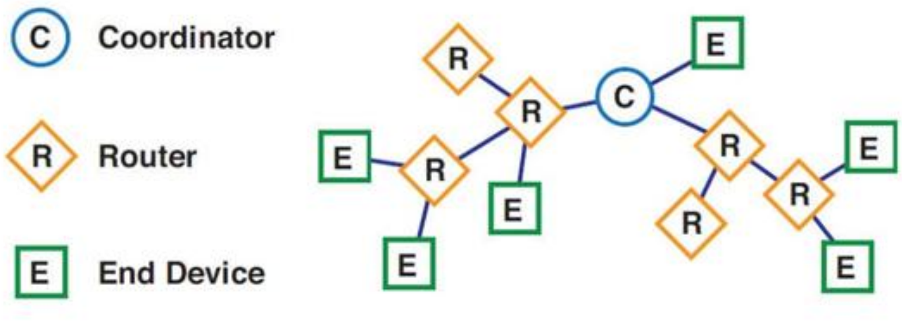


Figure 4.1 Zigbee Nodes

Radio module operating modes

The operating mode of an XBee radio module establishes the way a user or any microcontroller attached to the XBee communicates with the module through the Universal Asynchronous Receiver/Transmitter (UART) or serial interface. Depending on the firmware and its configuration, the radio modules can work in different operating modes:

- AT (transparent) operating mode
- API operating mode

When operating in transparent mode, the modules act as a serial line replacement. That is, all data received through the serial input is immediately transmitted over the air, and when the XBee receives wireless data, it is sent out through the serial interface exactly at it is received.

API (Application Programming Interface) mode is a frame-based method for sending and receiving data to and from a radio's serial UART. The API is an alternative to the default transparent mode.

The ZigBee Advantages

The ZigBee protocol is designed to communicate data through hostile RF environments that are common in commercial and industrial applications.

ZigBee protocol features include:

- Support for multiple network topologies such as point-to-point, point-to-multipoint and mesh networks
- Low duty cycle – provides long battery life
- Low latency
- Direct Sequence Spread Spectrum (DSSS)
- Up to 65,000 nodes per network
- 128-bit AES encryption for secure data connections
- Collision avoidance, retries and acknowledgement

Chapter 5

HARDWARE

CHAPTER 5

HARDWARE

5.1 BOARDS

ARDUINO

Arduino is an open-source physical computing platform based on a simple I/O board and an IDE integrated development environment that implements the Processing/wiring language. Arduino can be used as stand-alone interactive objects and installation or can be connected to software on computer.

5.1.1 ARDUINO UNO

The Uno is a microcontroller board based on the ATmega328P .It has 14 digital input/output pins of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip.Instead,it features the Atmega16U2 programmed as a USB-to- Serial converter.

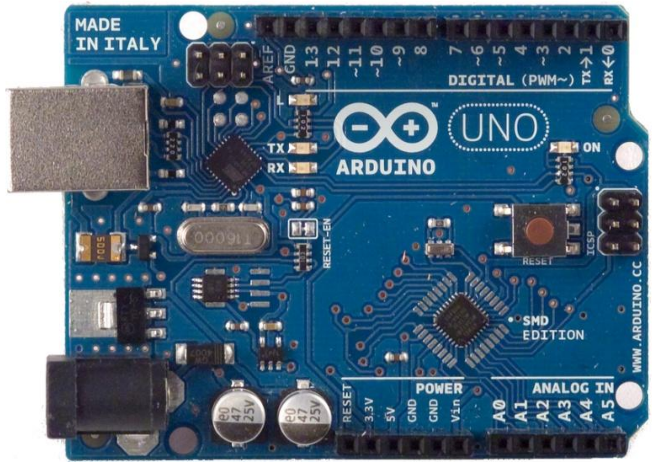


Figure .5. Arduino Uno

Specification:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

5.1.2 ARDUINO MEGA

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

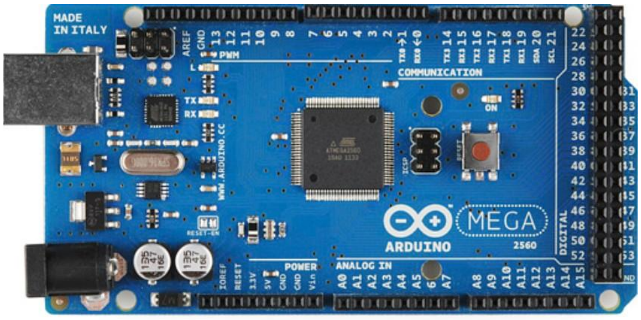


Figure 5.1.2 Arduino Mega

Specification:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

5.2 SENSORS

5.2.1 BMP180 (PRESSURE and TEMPERATURE SENSOR)

Figure 5.2.1 BMP180 Pressure and Temperature Sensor

The BMP180 is the new digital barometric pressure sensor of Bosch Sensortec, with a very high performance, which enables applications in advanced mobile devices, such as smart phones, tablet PCs and sports devices. It follows the BMP085 and brings many improvements, like the smaller size and the expansion of digital interfaces.

The ultra-low power consumption down to 3 μA makes the BMP180 the leader in power saving for your mobile devices. BMP180 is also distinguished by its very stable behavior (performance) with regard to the independency of the supply voltage.

5.2.2 IR SENSOR

Figure 5.2.2 IR 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 measures only infrared radiation, rather than emitting it that is called as a passive IR sensor.

Specification:

- Adjustable Range using preset (Using Potentiometer On board),
- Operating Voltage : 5V DC,
- Digital Output : logic one (+3.5V DC) logic zero (0V DC),
- Mounting Hole of 2.5mm diameter for Easy Mounting.,
- This can in turn be used for detecting white/black lines (in line follower ROBOTs) or bright/dark objects (in object identification ROBOTs).
- Useful for various Robotic Applications, Room Visitor Counter Systems, etc.

5.2.3 MQ2 GAS SENSOR

Figure 5.2.3 MQ2 Gas Sensor

The MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature. The output is an analog signal and can be read with an analog input of the Arduino.

The MQ-2 Gas Sensor module is useful for gas leakage detecting in home and industry. It can detect LPG, i-butane, propane, methane ,alcohol, hydrogen and smoke.

Specifications:

- Sensitive for Methane, Butane, LPG, smoke.
- This sensor is sensitive for flammable and combustible gasses.
- The heater uses 5V.

5.3 Other peripherals

5.3.1 MOTOR DRIVER IC L293D

Figure 5.3.1 Motor Driver IC L293D

This is a motor driver IC that can drive two motor simultaneously. L293D IC is a dual H-bridge motor driver IC. One H-bridge is capable to drive a dc motor in bidirectional. L293D IC is a current enhancing IC as the output from the sensor is not able to drive motors itself so L293D is used for this purpose. L293D is a 16 pin IC having two enables pins which should always be remain high to enable both the H-bridges.

5.3.2 DC MOTOR

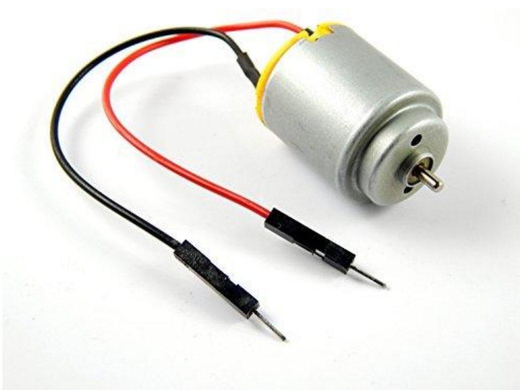


Figure 5.3.2 DC Motor

A DC motor is any of a class of electrical machines that converts direct current electrical power into mechanical power. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor and produce rotary motion.

5.3.3 5V RELAY SRD-05VDC

Figure 5.3.3 5V Relay SRD-05VDC

Features:

- Single SRD-05VDC-SL-C Relay
- Relay UL/CUL Rating: 10A @ 125V AC, 28V DC
- Relay CCC/TUV Rating: 10A @ 250V AC, 30V DC
- Control high-power devices up to 10A with a simple high/low signal
- Provides isolation between the microcontroller and the device being controlled
- Screw terminals for relay connections
- 3-pin servo-style header for power/signal interface
- LED indicator provides visual feedback
- Voltage requirements: 5V DC (Relay Power), 3.3V to 5V DC (Input Signal)
- Current requirements: ~85 mA (Relay Power)
- Communication: Logic High/Low (3.3V to 5V DC)
- Dimensions: 1.57 × 1.06 × 0.71 inches (4.0 × 2.7 × 1.8 cm)
- Operating temperature: -13 to +158 °F (-25 to +70 °C)

5.4 MODULES

5.4.1 XBEE MODULE-SERIES 2

Zigbee is a specification based on the IEEE 802.15.4 standard for wireless personal area networks. It is targeted at applications that require secure networking as well as high flexibility for network expansion anytime new nodes are to be added. It is widely used in the industrial control field, in hospitals, labs and in building automation.

Figure 5.4.1 Xbee Module

Features:

- 3.3V @ 40mA
- 250kbps Max data rate
- 2mW output (+3dBm)
- 400ft (120m) range
- Built-in antenna
- 128-bit encryption
- Local or over-air configuration
- AT or API command set

5.4.2 WiFi MODULE - ESP8266

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. . Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers .The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

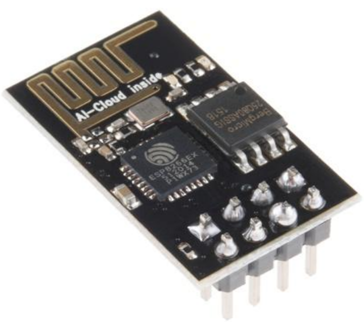


Figure 5.4.2 ESP8266 WiFi Module

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP

- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

Chapter 6

SOFTWARE

CHAPTER 6

SOFTWARE

6.1 ARDUINO IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

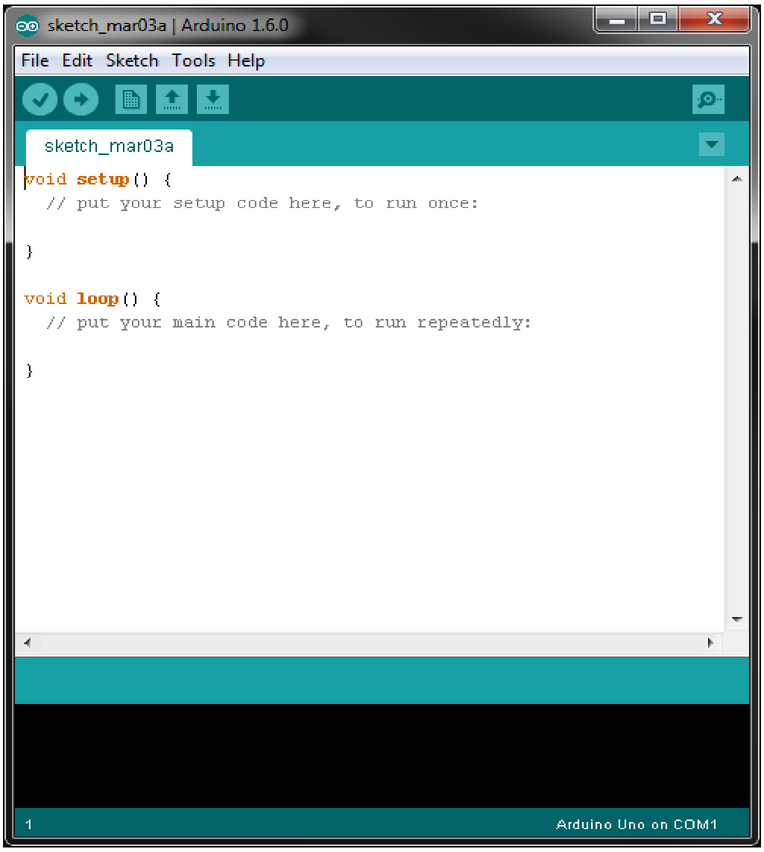


Figure 6.1 Arduino IDE Programming Environment

6.2 XCTU

XCTU is a free multi-platform application designed to enable developers to interact with Digi RF modules through a simple-to-use graphical interface. It includes new tools that make it easy to set-up, configure and test XBee RF modules.XCTU includes all of the tools a developer needs to quickly get up and running with XBee. Unique features like graphical network view, which graphically represents the XBee network along with the signal strength of each connection, and the XBee API frame builder, which intuitively helps to build and interpret API frames for XBees being used in API mode, combine to make development on the XBee platform easier than ever.

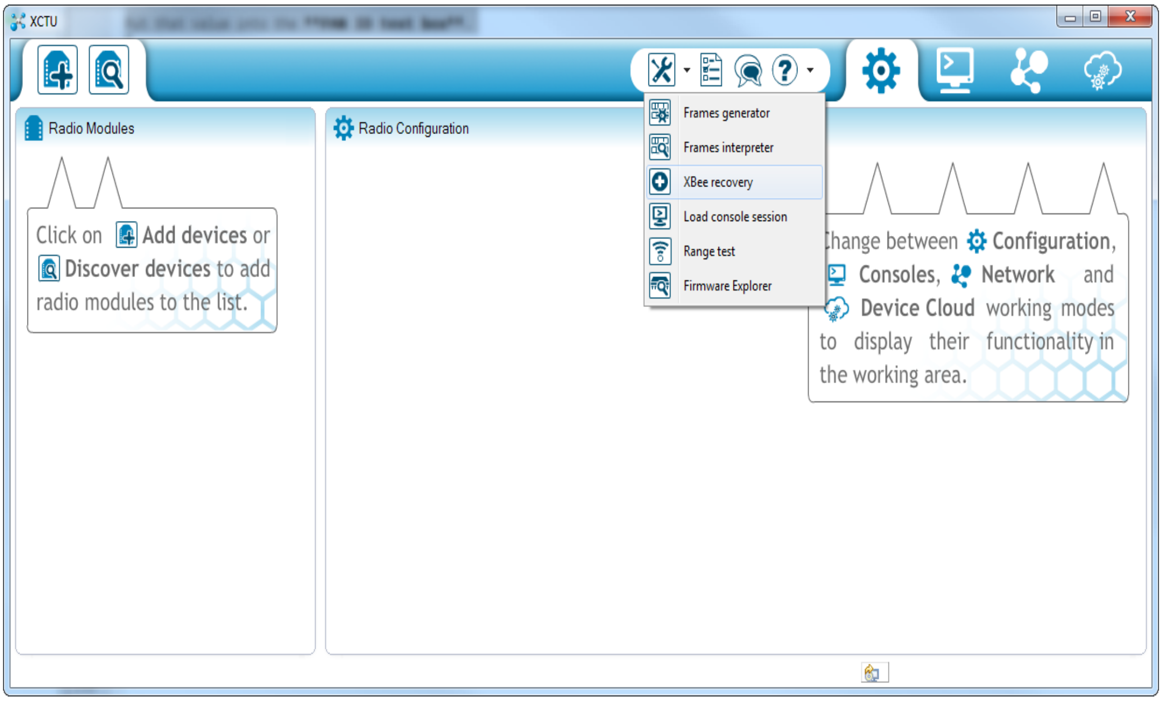


Figure 6.2 XCTU Application

6.3 MIT APP INVENTER

MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less.

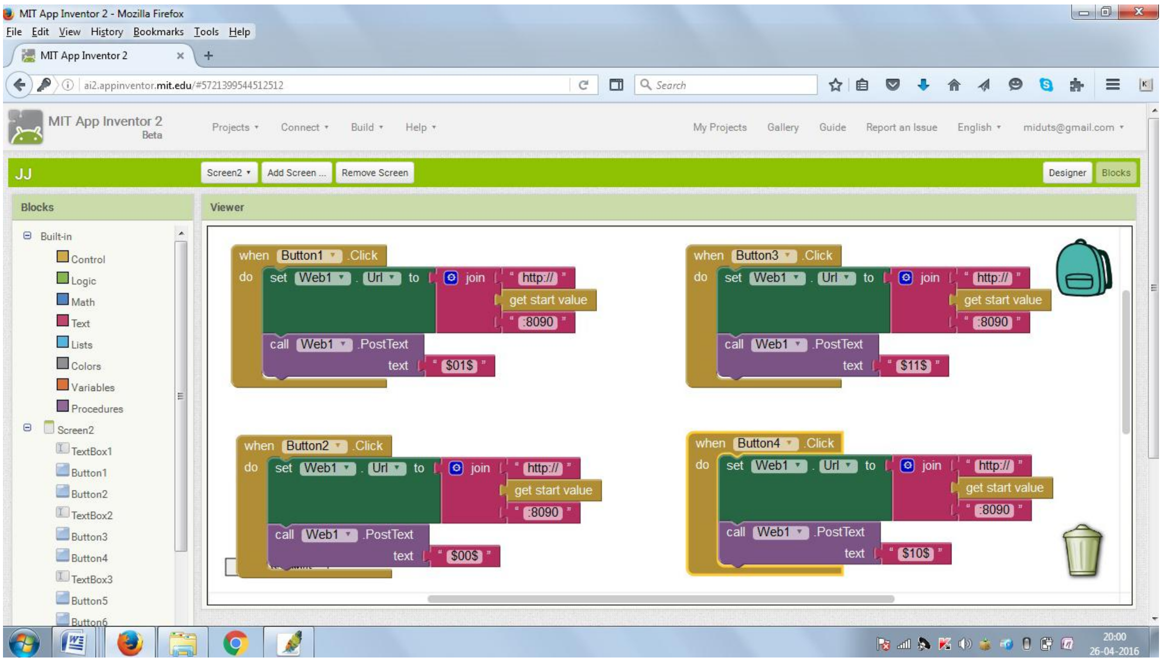


Figure 6.3 MIT APP Inventor 2

Chapter 7

DESIGN AND IMPLEMENTATION

CHAPTER 7

DESIGN AND IMPLEMENTATION

The project is designed to have two stations, Remote node and the Base station.

7.1 REMOTE NODE

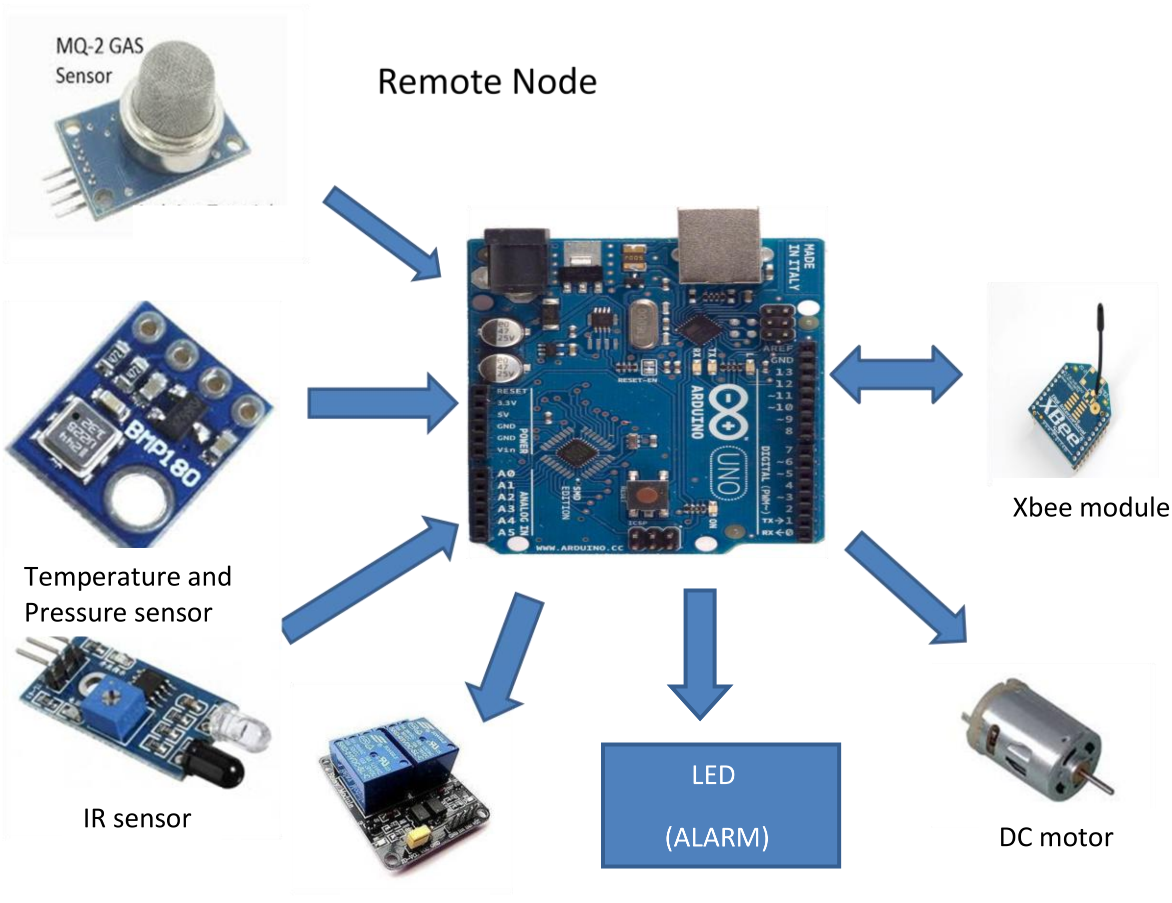


Figure 7.1 Remote Node

The Remote Node consist of sensors such as

- Temperature and Pressure sensor
- IR object sensor
- Gas sensor

The Remote Node actuators include

- Relay
- Led
- Dc motor

The remote station uses the Arduino Uno board as the controller.

All the sensors and actuators are connected to the Arduino Uno Controller board. The Xbee module is also connected to the Uno board to achieve wireless transmission.

7.1.1 Working

The Arduino uno controller runs the code for interfacing the sensors, actuators and modules. The data is collected from the sensors. The collected data is then sent to the Base station wirelessly using the Xbee module.

When the temperature increases above the defined threshold then the relay is turned ON (for example, illustrating the turning ON of a coolant valve to cool down an engine when the temperature is above its limit).

When the Gas sensor detects any combustible gas (illustration of Safety), when this condition occurs the dc motor(turning ON of exhaust fan) and the LED (an alarm) is turned ON(this helps in alerting the workers about a dangerous situation and taking real-time action keep the situation in control).

7.2 BASE STATION

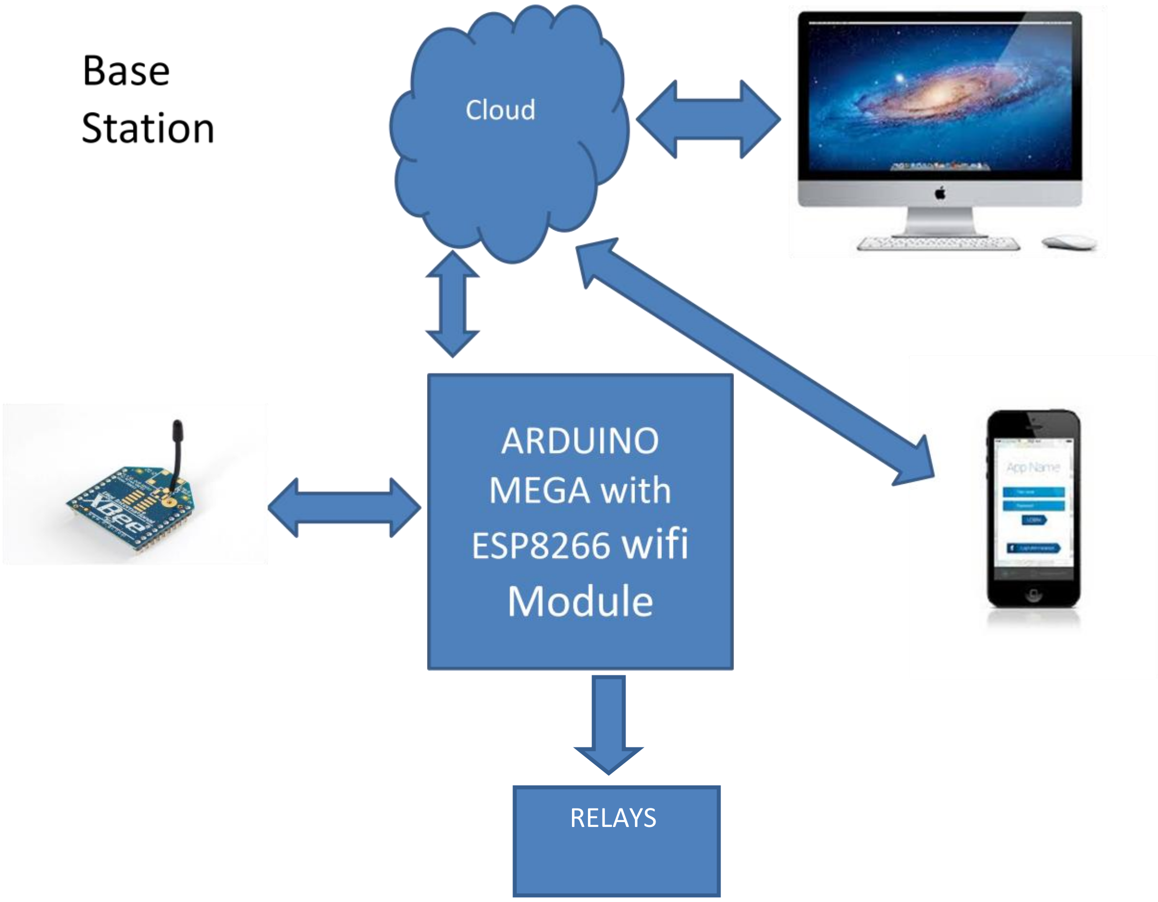


Figure 7.2 Base Station

The base station consist of the Arduino Mega board used as the controller.

The base station consist of

- Xbee module
- ESP8266 Wi-Fi module
- Relays

The base station allows the cloud connectivity using the wi-fi module. This allows the user to view the data collected and to control the equipments using the mobile app . The Base Station (setup in the Control Room in the industries) also allows the manual control of the equipments besides the app control. Our project allows the flexibility of installation in the current industrial setup.

7.2.1 Working

The esp8266 module allows internet connectivity to mega board. The data received is then sent to the cloud service(thingspeak.com).The data sent is stored in the cloud service and the data is also analysed. The user can view this data through his mobile app and also control the relays (represent the equipments that are present in the industries). The base station (represent Control Room in the industries which has the control of all equipments) also allows the flexibility of controlling the relays.

7.3 Cloud service

ThingSpeak is a free web service that lets you collect and store sensor data in the cloud and develop Internet of Things applications. The ThingSpeak web service provides apps that let you analyze and visualize your data in MATLAB, and then act on the data. Sensor data can be sent to ThingSpeak from Arduino, Raspberry Pi™, BeagleBone Black other hardware.

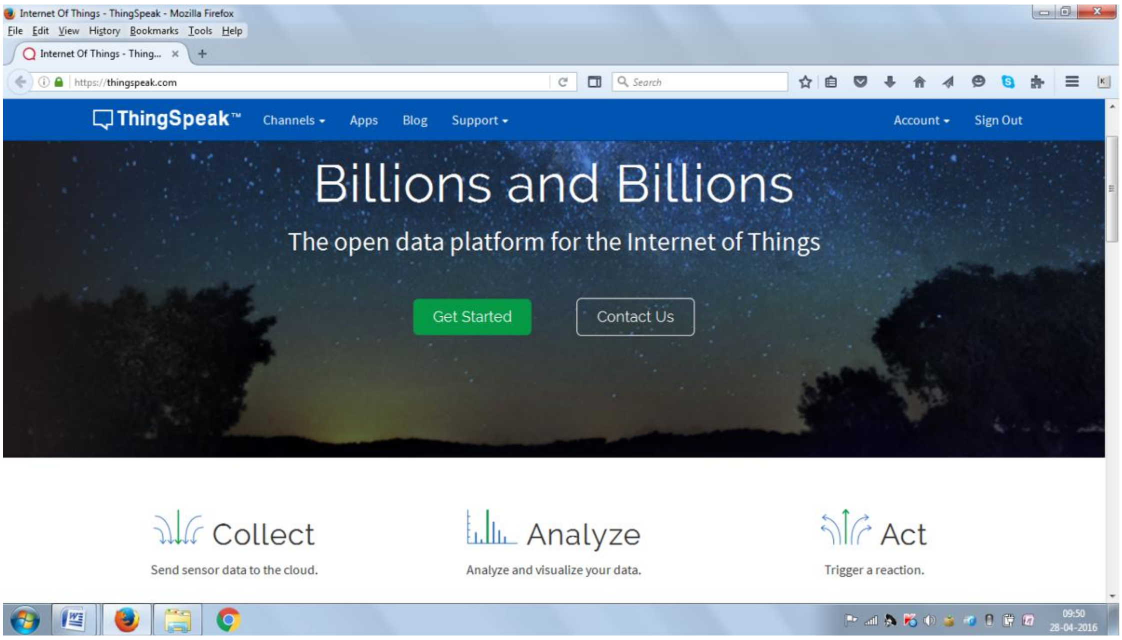


Figure 7.4 Thingspeak Website

7.3.1 ESP8266(Wi-fi module) to Thingspeak(Cloud Service)

At this point, it is assumed that the ESP8266 module is connected to the access point.

Then we set it up to allow multiple connections with CIPMUX:

```
AT+CIPMUX=1
```

Then, start a TCP connection to the ThingSpeak server (184.106.153.149), using http port 80.

```
AT+CIPSTART=4,"TCP","184.106.153.149",80
```

The ESP8286 should respond with:

```
OK
```

Next, We use the CIPSEND to send our data. The first value is the channel id , followed by the size of the string

```
AT+CIPSEND=ch id,size of string
```

Then the module should respond with

>
to show it is ready to receive the data.

The data is sent

```
GET /update?key=API Key&field1=data
```

Finally, close the TCP connection at the end with

```
AT+CIPCLOSE
```

7.4 Mobile App

App Inventor is a cloud-based tool, which means you can build apps right in your web browser. This website offers all the support you'll need to learn how to build your own apps. The App Inventor software, or "service" is at ai2.appinventor.mit.edu**.** You can get there by clicking the orange "Create Apps!" button from any page on this website.

Palette: Find components and drag them to the viewer To add them to our app

Designer Button: Click from any tab to go to the designer tab

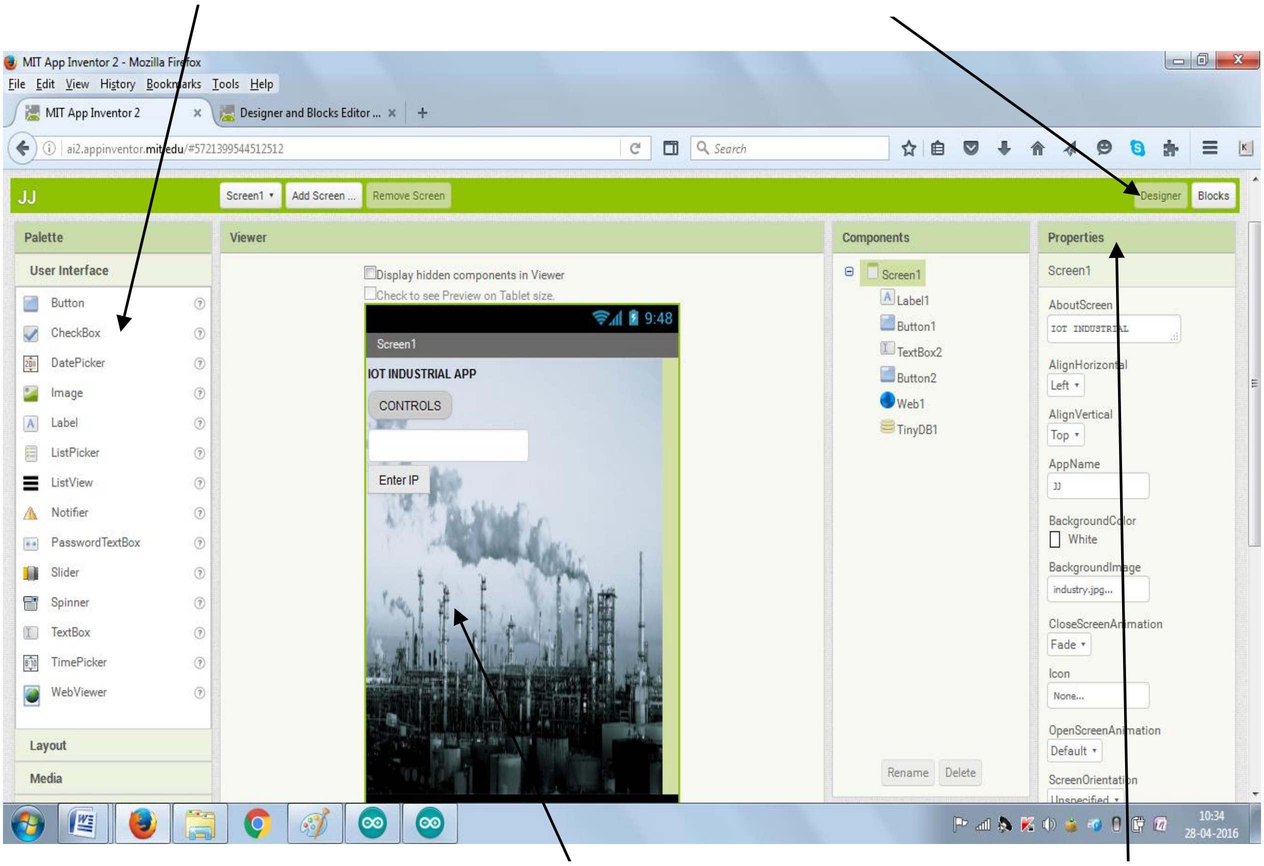


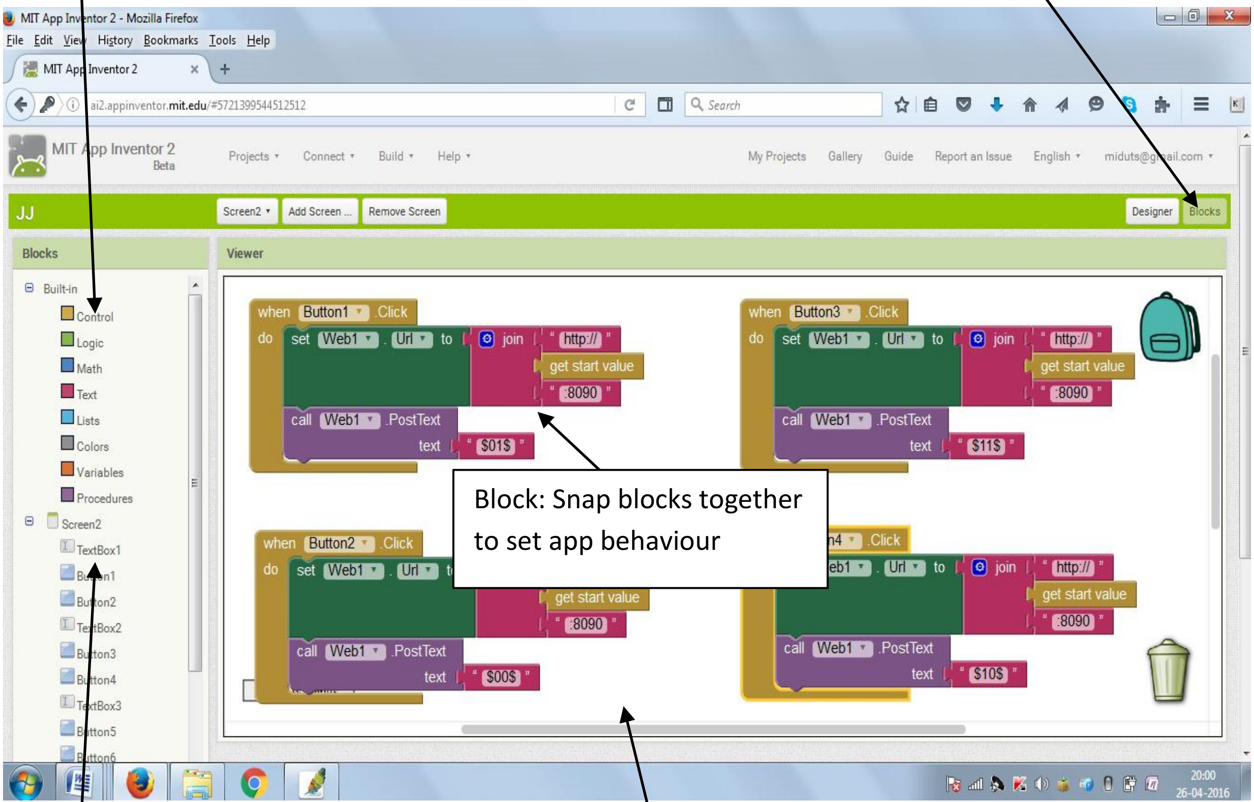
Figure 7.4.1. App inventer View 1

Viewer: Drag components from the palette to the viewer to see what our app will look like

Properties: Select a Component in the component list to change its properties

Built-In drawers: Find blocks for general behaviour that we may want to add to our app and drag them to blocks viewer

Block Button: Click from any to go to blocks tab



Component specific Drawers: Find blocks for behaviours for specific components and drag them to blocks viewer.

Viewer: Drag blocks from the drawers to the blocks viewer to build relationships and behaviour.

Figure 7.4.2. App inventer View 2

7.5 Functioning of the App

The industrial iot app is used to control the equipments in an industry and to view the collected and analysed data. The function of this app is given in this section.

Screen 1:

The Screen 1(Home screen) shows the control button ,which takes the user to the control screen. It has a text field where the user has to enter the IP address of the device to which the app should connect to(here it is the wi-fi module).

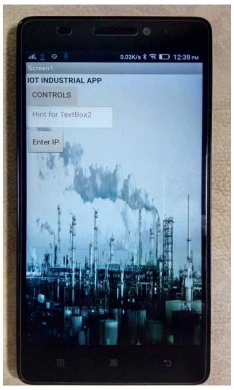


Figure 7.5.1.1 App Screen 1

Screen 2:

Screen 2(control screen), this screen provides control buttons(the number of buttons depends on the number of equipments to be controlled i.e two in our case).The user can control the equipments using the ON and OFF buttons. This screen also provides the button to view the data and to see the status of the equipments.

Figure 7.5.1.2 App Screen2

A unique number array is assigned to each of the control button to differentiate between the different equipments. In our case

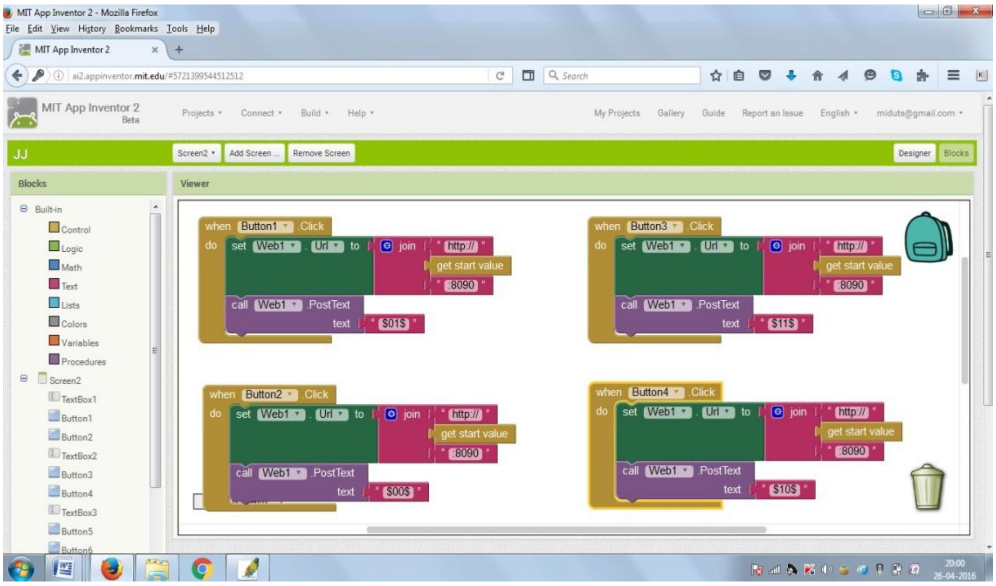


Figure 7.5.1.3 Screen 2 blocks view

for two equipments(01,00,10,11 is used) 01 stands for equipment1-ON and 00 stands for equipment 2-OFF similarly for others. When the control button is pressed the array is sent to the assigned IP address and is decoded to perform the necessary action.

Screen 3:

Screen 3 provides the buttons to view the stored data. Different buttons are provided to view the different data fields. The buttons are linked to the cloud service and connects to the url of the cloud service where the data is stored. The analysis button takes the user to the screen where the user can view the analysed data.

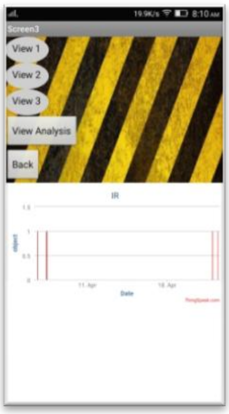


Figure 7.5.1.4 App Screen 3

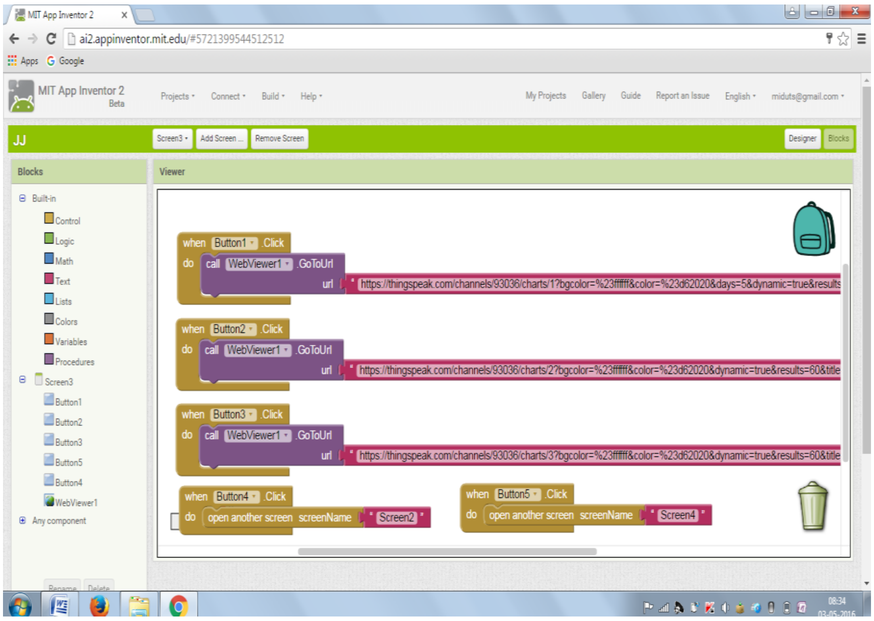


Figure 7.5.1.5 Screen 3 blocks view

Screen 4:

The screen 4 allows the user to view the different analysed data.

The buttons are linked to the cloud service url where the analysed data is present.

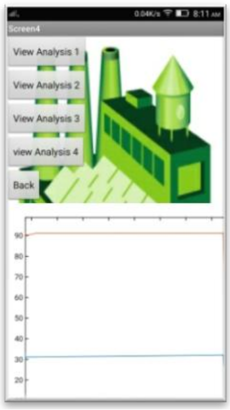


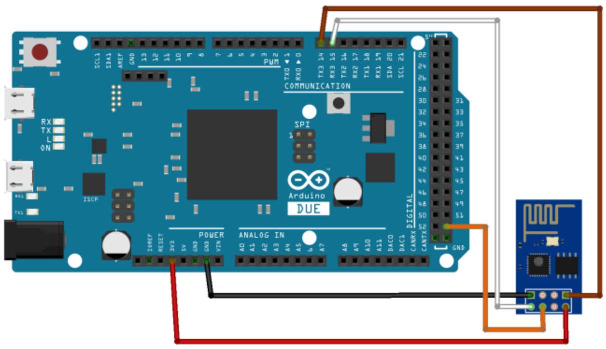
Figure 7.5.1.6 App Screen 4

7.6 INTERFACING

7.6.1 INTERFACING ARDUINO AND WI-FI MODULE

The interfacing of the Arduino Mega to the Wi-Fi module (ESP8266) is achieved through the use of serial interface.

The Tx and Rx pin of both the Arduino and Wi-Fi module is connected along with 3.3v supply and ground pins.



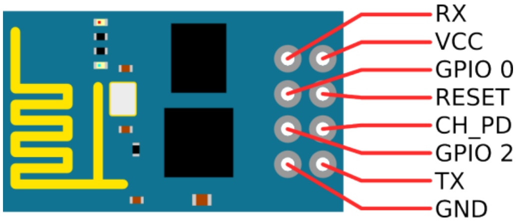


Figure 7.6.1 Arduino to esp8266

Arduino Mega	Esp8266
3.3V	Vcc
Gnd	Gnd
D18(Tx1)	Rx
D19(Rx1)	Tx
3.3V	CH_PD
3.3V	RST
	GPIO 0,GPIO2(not connected)

Table 7.6.1 Arduino to esp8266

7.6.2 INTERFACING ARDUINO AND XBEE MODULE

The Xbee module and the arduino interfacing is achieved through the arduino serial interface. The Rx and the Tx pins of the Xbee module is connected to the Tx and Rx of the arduino board.

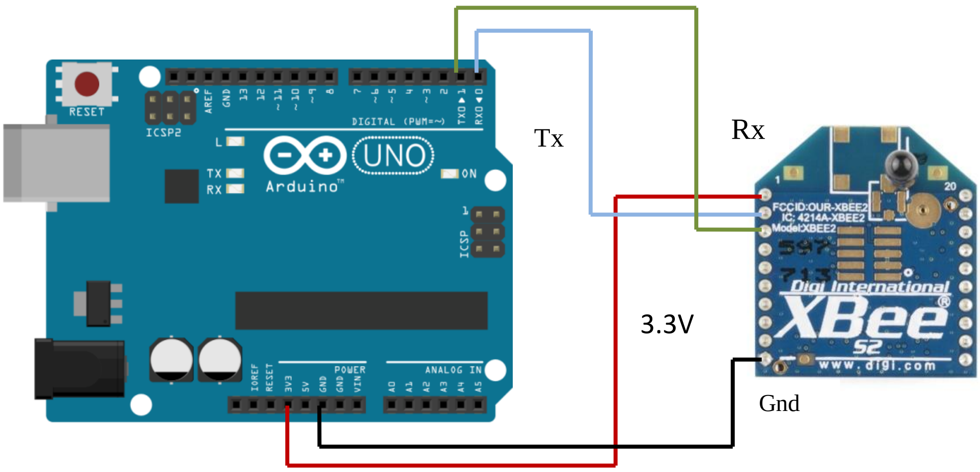


Figure 7.6.2 Arduino to xbee

Arduino Uno	Xbee Series 2 module
Pin 0 (Rx0)	Tx
Pin 1 (Tx0)	Rx
3.3V	3.3V
Gnd	Gnd

Table 7.6.2 Arduino to Xbee

7.6.3 INTERFACING BMP180 AND ARDUINO

Connection between bmp180 and arduino are as shown below

Arduino Uno	Bmp180
5V	Vin
Gnd	Gnd
A5	SCL
A4	SDA

Table 7.6.3 Arduino to Bmp180

7.6.4 INTERFACING IR SENSOR AND ARDUINO

Connection between IR sensor and arduino are shown below

Arduino Uno	Ir sensor
5V	5V
Gnd	Gnd
Pin 7	AOUT

Table 7.6.4 Arduino to Ir sensor

7.6.5 INTERFACING MQ2 SENSOR AND ARDUINO

Connection between MQ2 and arduino are shown below

Arduino Uno	MQ2 sensor
5V	Vcc
Gnd	Gnd
A0	A0

Table 7.6.5 Arduino to mq2

7.6.6 INTERFACING RELAY AND ARDUINO

Connections between arduino and relay are shown below

Arduino Uno	Relay
Pin 9	IN
5V	Vcc
Gnd	Gnd

Table 7.6.6 Arduino to relay

7.6.7 INTERFACING DC MOTOR AND ARDUINO

Connection between arduino and L293d motor driver is as shown

Arduino Uno	L293d motor driver
Pin 6	IN2
Pin 5	IN1
Gnd	Gnd
5v	5v

7.6.7 Arduino to dc motor

Chapter 8

WORKING OF THE SYSTEM

All the sensors, actuators and the communication modules are interfaced as shown in the previous chapter.

At the **remote station** the sensors such as temperature, pressure, ir sensor and gas sensor are used. The actuators used are dc motor(exhaust fan) and relay.The sensors and actuators are interfaced to arduino uno, which is the controlling board at the remote station.

Bmp180 (pressure and temperature sensor)

The Bmp180 sensor(pressure and temperature sensor) is connected to the analog pin of the arduino. The output of the sensor is available in the I2C format. Thus the sensor data can be read by using arduino as it has I2C interface.

The sequence to be followed for taking pressure or temperature reading is

- Read calibration constants from BMP180 EEPROM
- Send start sequence to start pressure ot temperature measurement
- After converting time, the result value can be read via the I2C interface
- For calculating temperature in *C and pressure in Pa, the calibration data has to be used

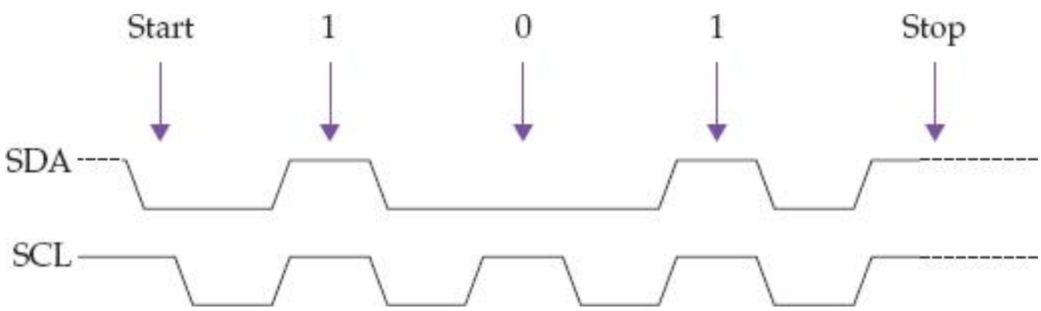


Figure 8.1 Timing diagram

IR sensor

The Ir sensor is connected to the digital pin of the arduino uno. The IR sensors consist of two elements: infrared source and the infrared detector. Infrared detectors include photodiodes or phototransistors. The energy emitted by the infrared source is reflected by an object and falls on the detector. The arduino pin is read. If the object is detected then a HIGH signal is read on the arduino pin.

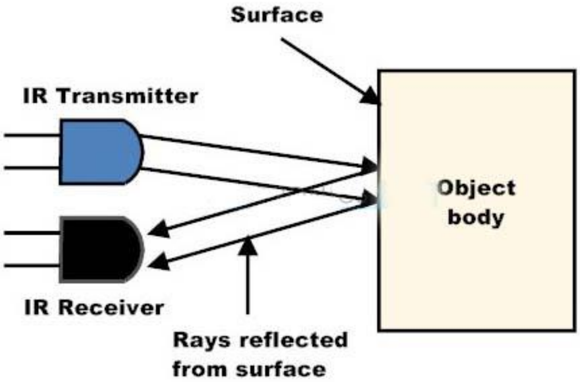


Figure 8.2 IR sensor working principle

Gas sensor

The gas sensor is connected to the analog pin of the arduino uno. The protection resistor and the adjustable resistor which are in series and forms the load resistance RL. The output voltage on the signal pin could be read by arduino. Resistance Rs could be derived from power supply voltage,output voltage and given value of RL. Ro is the intial resistance which is derived from Rs. The concentration of gas can be calculated by using the Rs/Ro ratio as the input.

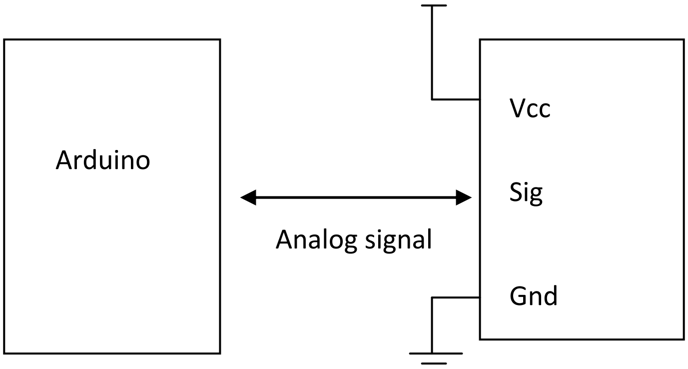


Figure 8.3 Gas sensor to arduino

The procedures that are mentioned above are used for reading the sensors data is implemented. The thresholds are set and if the data exceeds these thresholds then the necessary action are performed. If the temperature data meets the threshold then the relay, which is connected to the arduino digital pin is activated as shown in the below diagram. If any combustible gas is detected then the dc motor is turned ON and the alarm is activated as shown in the below diagram.

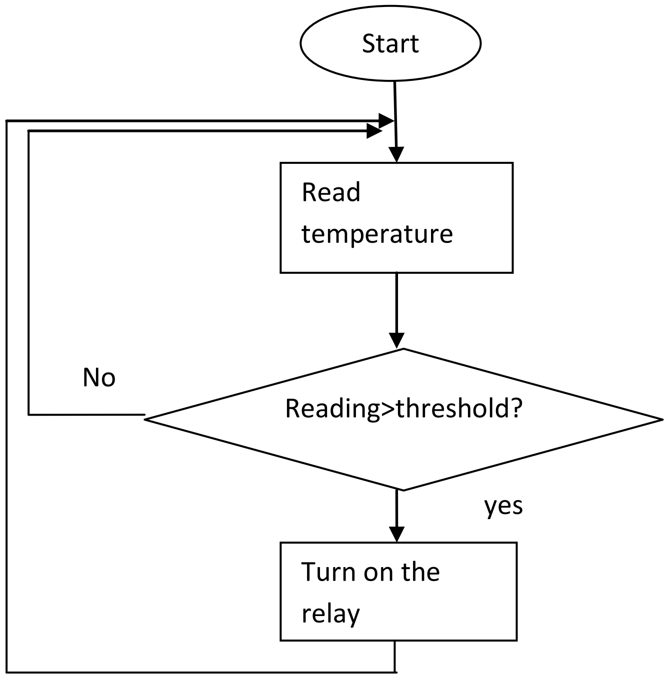


Figure 8.4 Flow chart of temperature sensor based actuation

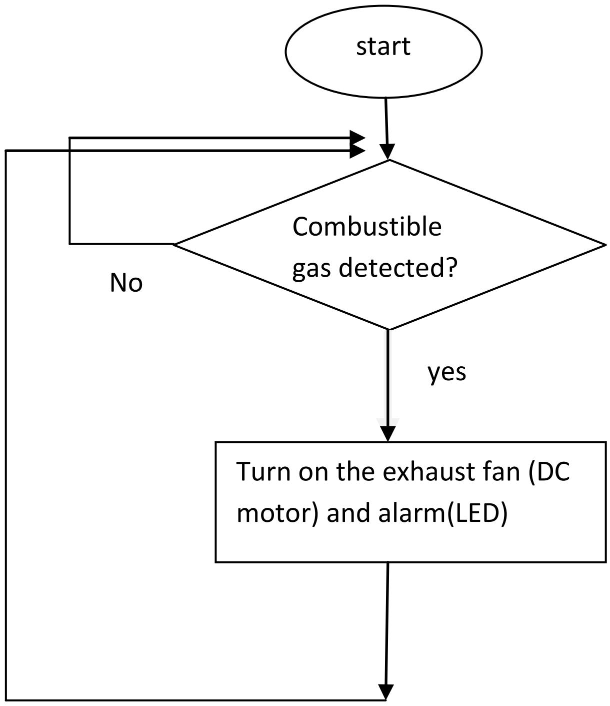


Figure 8.5 Flow chart of gas sensor based actuation

The data read is grouped together and sent to the base station via the xbee module in the form of data frames.

Communication between the remote node and base station

The sensor data is sent in the form of frames by the xbee module between the remote node and base station. They are sent and received through the serial interface of the XBee and contain the wireless message itself as well as some extra information such as the destination/source of the data or the signal quality. The Xbee modules are configured in the API mode.

An API frame has the following structure:

Start Delimiter	Length		Frame Data								Checksum
1	2	3	4	5	6	7	8	9	...	n	n + 1
0x7E	MSB	LSB	API-specific structure								Single byte

Any data received through the serial interface prior to the start delimiter is silently discarded by the XBee. If the frame is not received correctly or if the checksum fails, the data is also discarded and the module will indicate the nature of the failure by replying with another frame.

Start delimiter

The first byte of a frame consisting of a special sequence of bits which indicate the beginning of a data frame. Its value is always 0x7E. This allows for easy detection of a new incoming frame.

Length

The length field specifies the total number of bytes included in the frame data field. Its two-byte value excludes the start delimiter, the length, and the checksum.

Frame data

This field contains the information received or to be transmitted. This includes the sensor data that was collected.

Checksum

Checksum is the last byte of the frame and helps test data integrity. It is calculated by taking the hash sum of all the API frame bytes that came before it, excluding the first three bytes (start delimiter and length).

At the **Base station** the various modules available are esp8266(wifi module), xbee and relays interfaced to arduino mega.

The Base station has two modes of operation: The manual control mode and the App control mode.

Manual control mode: The user can control the equipments in the industry (in our case relay) manually using switches provided at the base station(Master control room in industry).

App control mode: The user can control the equipments in the industry(in our case relays) using the IOT industrial app.

The **ESP8266 module** uses AT commands to get connected to the hotspot(Wi-Fi dongle).The various AT commands used are :

Command	Response	Function
AT+CWMODE=3	OK	Set AP +station mode
AT+CWLAP	AT+CWLAP: ecn,ssid,rssi,mac OK	Lists available Access Points
AT+CWJAP=ssid,pwd	OK	Commands esp8266 to connect a ssid with supplied password

Table 8.6 AT commands

The zigbee frames are received from the zigbee present at the remote station by the zigbee at the base station then the payload is separated from its data frame and stored in an array variable.

Now we have to check whether the base station is in manual control mode or in app control mode. If it is in manual control mode then manual switch control is allowed and the user can now control the equipments manually or if it is in app control mode then the user can use the IOT industrial app to control the equipments.

ESP8266 to cloud(Thingspeak.com):

Now we assume that the esp8266 module is connected to the access point .

We start a TCP connection to the ThingSpeak server (184.106.153.149), using http port 80.

`AT+CIPSTART=4,"TCP","184.106.153.149",80` and the esp responds with an `OK` .

We use the CIPSEND to send our data.

`AT+CIPSEND=ch id,size of string`

The data that is stored in array variables are sent using

`GET /update?key=API Key&field1=data` and the TCP connection is closed.
(For more details refer section 7.4.1)
The data is updated every 15 seconds on to the thingspeak cloud service.

Communication from app to base station:

As shown in the previous section (7.5.1) the IOT industrial app consist of screen with respective control buttons. Every control button is assigned with a two byte array i.e “00”,”01”,”10”,”11”.The first byte is used to differentiate between the equipments and second byte is used to tell if the equipment needs to be switched ON/OFF.

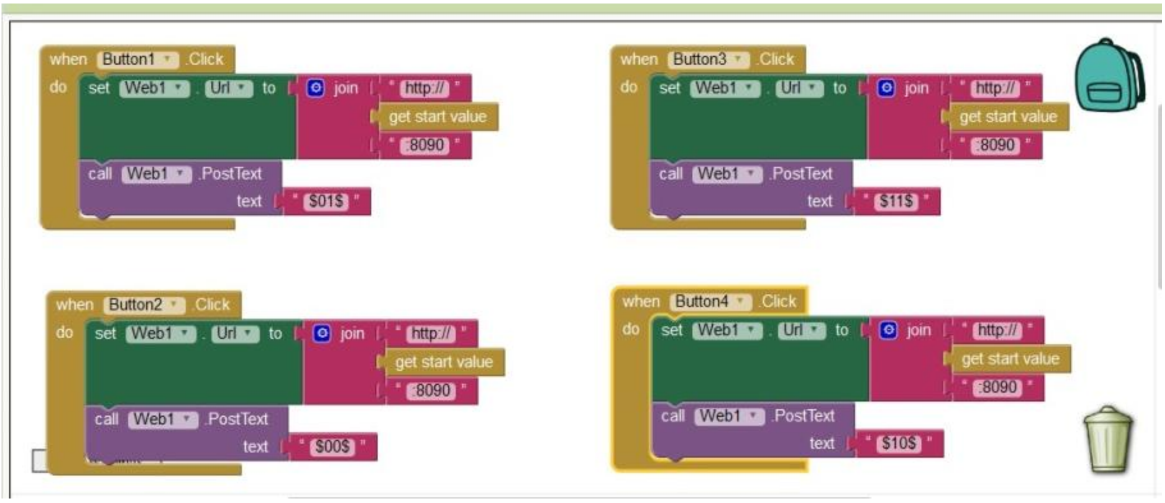


Figure 8.6 The blocks view of the Buttons in the App

Now when the user press the button then that corresponding array is sent to the esp module present at the base station by opening a TCP connection through port 8090 .This array is decoded according to the logic that has been told in previous paragraph and then the corresponding action of controlling the relay takes place.

Monitoring the data in the app: The data stored in the cloud can be monitored using the Industrial IOT app through HTTP requests. When the view button in the app is pressed the button is linked to the URL of the field in the cloud service where the data is stored. Similarly the user can view the analysed data that is present in the cloud.

Chapter 9

TESTING

CHAPTER 9

TESTING

This chapter of the report specifies how each component has to be tested.

Testing of Aduino

The arduino boards has pin 13 as the default on board LED. This is used to check the boards working. A simple code is uploaded into to controller with pin 13 has high. If the LED does not glow then the board is malfunctioning.

Testing of IR sensor

The IR sensor is connected to one of the digital pins of the arduino uno board. A simple code is uploaded to read the digital pin status. The IR sensor returns a HIGH (5V)value when an object is detected.

Testing of BMP180(Temperature and Pressure sensor)

The BMP180 sensor is connected to the two analog pins of the arduino board. A code to read the temperature and pressure is uploaded into the arduino board. The arduino serial monitor window is opened. The temperature and pressure values that are read is shown.

Testing of Gas sensor

The Gas sensor is connected to the analog pin of the arduino board. A simple code to read the analog values(analogRead(analog pin)) and to print that on the serial monitor is uploaded. A lighter is lit near to the gas sensor mesh. The serial monitor window is opened to see the reading variations when the gas is detected.

Testing of the Relay

The relay is connected to the arduino digital pin. A code is uploaded to the arduino board to make the digital pin HIGH. The led on the relay glows when the digital pin is made HIGH. From this it can be known that the relays are in working condition.

Testing of the Xbee module

The Xbee is linked to the PC using a xbee adapter board. The XCTU software is opened and the Xbee modules are added to it. The Xbee is configured in the AT mode. The console window of the software is opened. In the console window”+++” is entered to test the Xbee. If the output displays “OK” then it verifies that the Xbee module is working.

Testing of Wi-Fi module(ESP8266)

All the pins of the esp8266 module is connected to 3.3v except the Tx and Rx pin, which are connected to the Tx and Rx pin of the arduino board. The serial baudrate is set to 115200 bps. A blank code is uploaded into the arduino board. The serial monitor window is opened. In the serial monitor “AT” is entered and sent and the module replies with an”OK”. This verifies that the esp8266 module is working.

Chapter 10

SNAPSHOTS

CHAPTER 10

SNAPSHOTS

10.1 Complete setup

Figure 10.1 Complete Setup

10.2 3D MODEL OF INDUSTRIAL SETUP

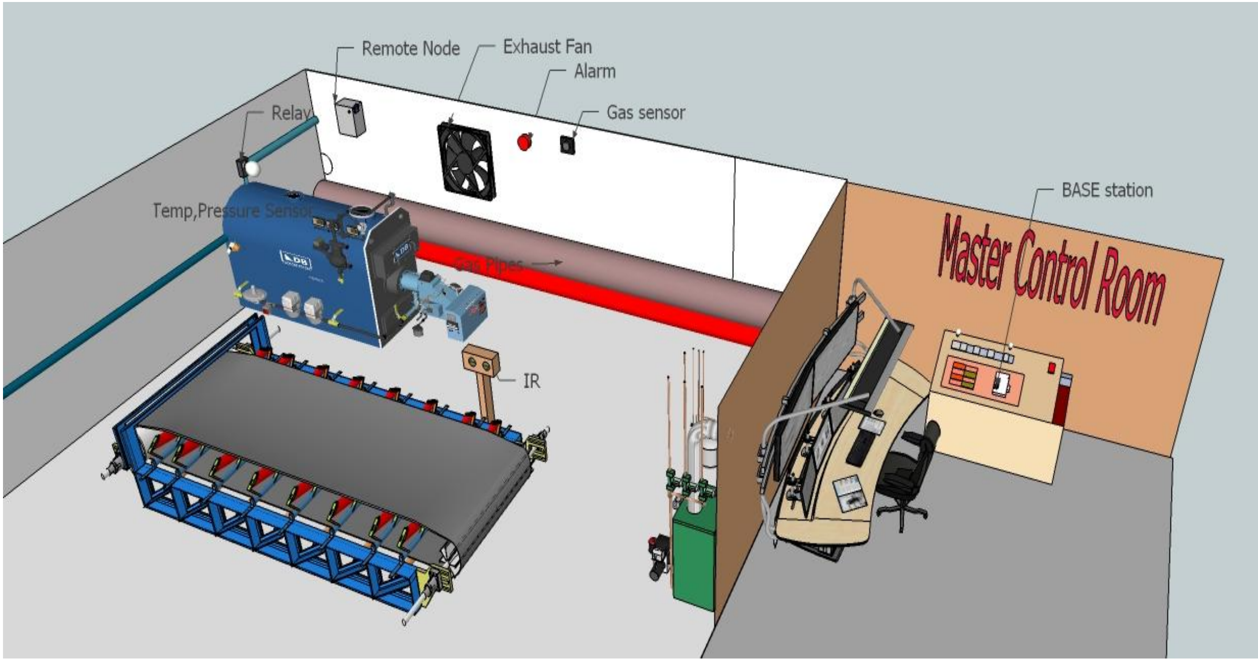


Figure 10.2 Model of Industrial Setup

10.3 REMOTE NODE

Figure 10.3.1 Remote Station outer view

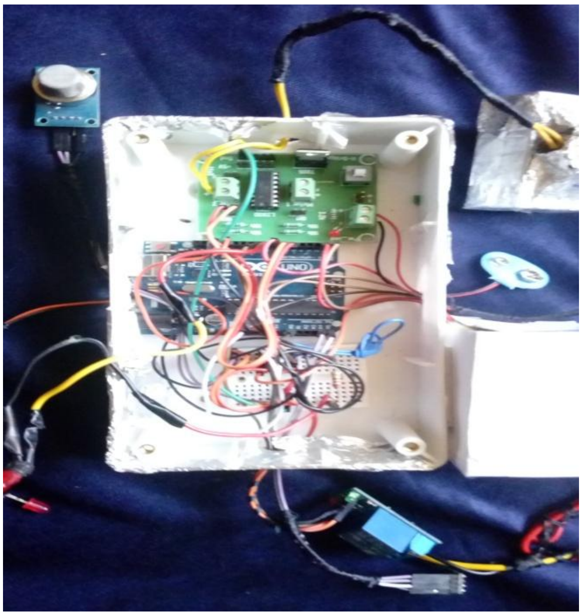


Figure 10.3.2 Remote Station inner view

10.4 BASE STATION

Figure 10.4.1 Base Station outer view

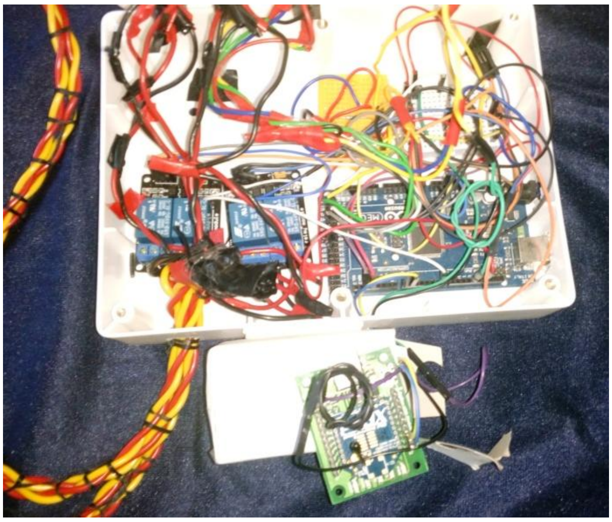


Figure 10.4.1 Base Station inner view

10.5 MOBILE APP

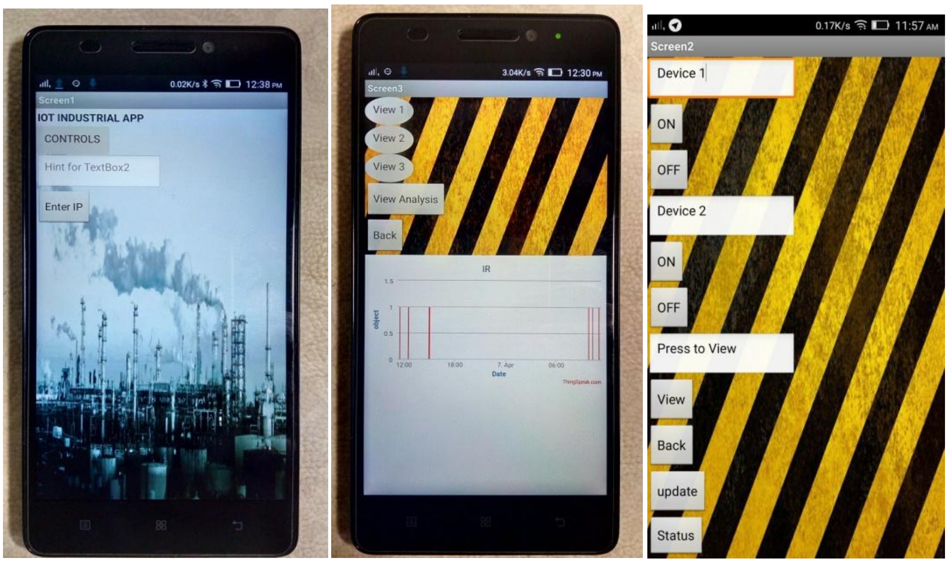


Figure 10.5 Mobile App

10.6 CLOUD SERVICE

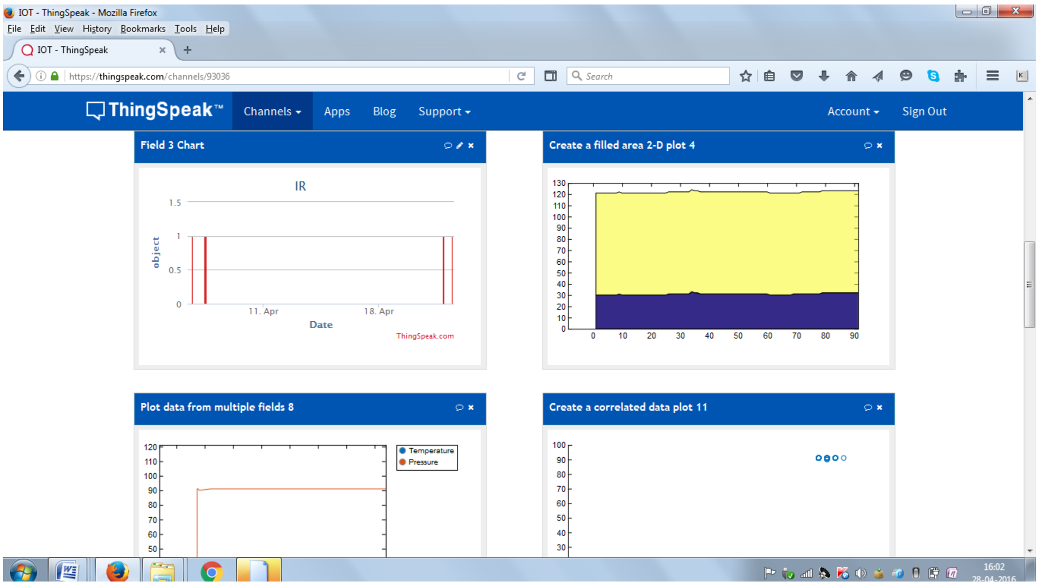


Figure 10.6 Cloud Service

Chapter 11

CONCLUSION

CHAPTER 11

CONCLUSION

With our project we could successfully implement the internet of things setup for the industries. We could demonstrate a reliable zigbee communication between the remote station and the base station . The vital part of our project was connecting the base station to the internet to store the data and using the mobile app to view this data and control the equipments, which we could achieve as we had initially planned. We as a project team could achieve our targets in the specified time as per the gantt chart we made during the beginning stage of our project.

In the course of our project there were many reviews and feedback sessions which helped us in improvising our project. This project helps us in learning about new technologies, getting updated to the current trends in the tech field. With this project we learnt many new concepts and was a good platform to demonstrate our skills that we acquired throughout these years. We also learnt about team work , time management and managerial skills, which would be of great use in future.

11.1 Future Enhancement

There is always a scope for further improvements in any projects.

In our case ,the future improvements include the adding of multiple remote stations that can be placed at different sectors in the industries. The industrial environments is prone to high electromagnetic interferences , inorder to guarantee the efficient operation of our system we need to implement RF hardening techniques. The sleep modes in the zigbee modules can be implemented which will reduce the power usage. The sensors and actuators can be changed according to different industrial applications and environments.

BIBLIOGRAPHY

This section of the project report includes the details of different references made.

Papers referred:

Case Study on Comparison of Wireless Technologies in Industrial Application,V.Abinayaa, Anagha Jayan, International Journal of Scientific and Research Publications, Volume 4, Issue 2, February 2014 ,ISSN 2250-3153.

Zigbee: A low power wireless technology for industrial application,nisha ashok somani and yash patel,international journal of control theory and computer modelling(IJCTCM) Vol 2,no.3,may 2012.

A Survey of the internet of things, De-Li Yang Feng Liu Yi-Duo Liang, The 2010 International Conference on E-Business Intelligence.

Links referred:

1. www.mcuoneclipse.com
2. www.esp8266.com
3. www.digikey.com
4. www.arduino.cc
5. www.thingspeak.com
6. https://youtu.be/odekkumB3WQ
7. www.instructables.com
8. www.mitappinventor.com
9. www.digi-xctu.software.informer.com

Books/PDF'S referred:

1. ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf
2. arduino_notebook_v1-1.pdf
3. MITAppInventorDevelopmentOverview.pdf

APPENDIX

FlowCharts

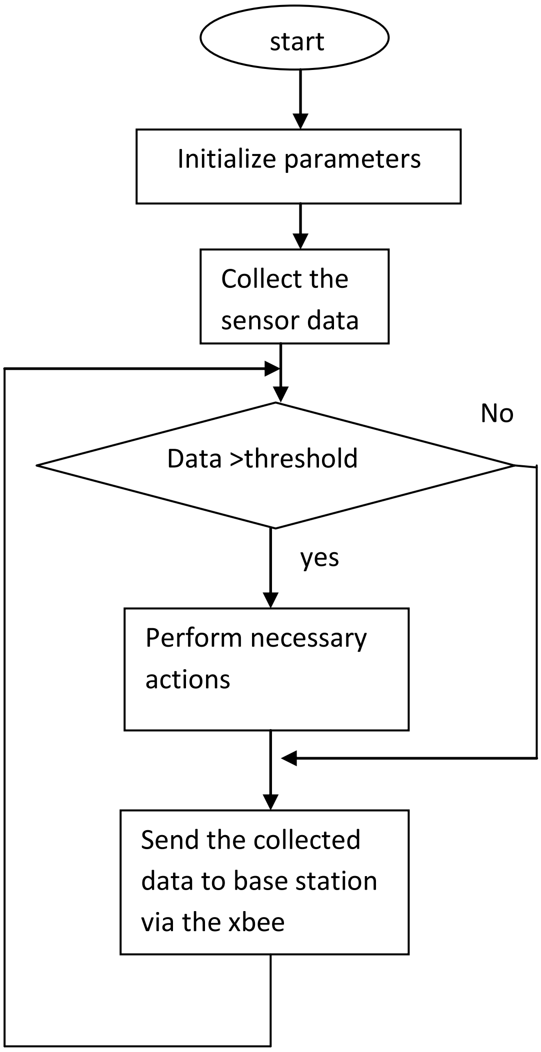


Figure 1 Flow chart of remote station

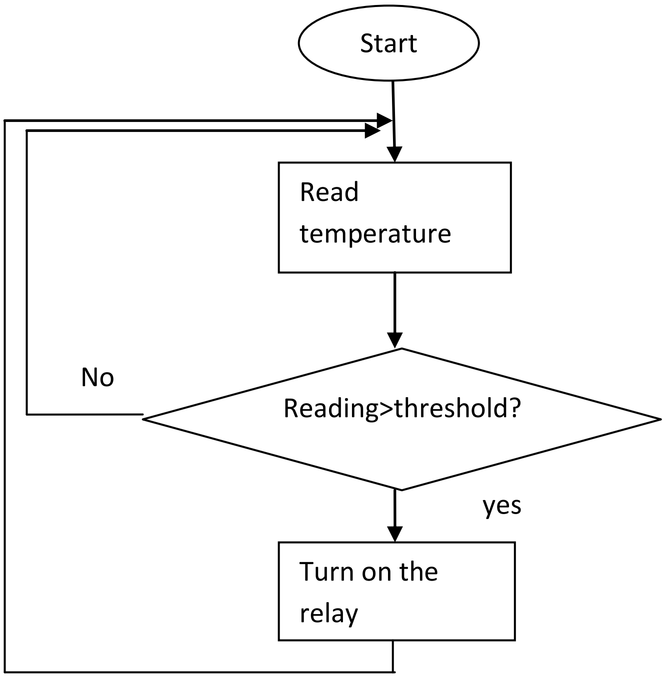


Figure 2 flow diagram of temperature sensor

The initialization of all the parameters is done. The code for interfacing and reading the sensor data is written. The condition is checked to know if the assigned threshold is reached. If the temperature is above the threshold then the relay is turned ON(figure 2). If the gas temperature detects any combustible gas then an exhaust fan is turned on and alarm is activated as shown below(figure 3).

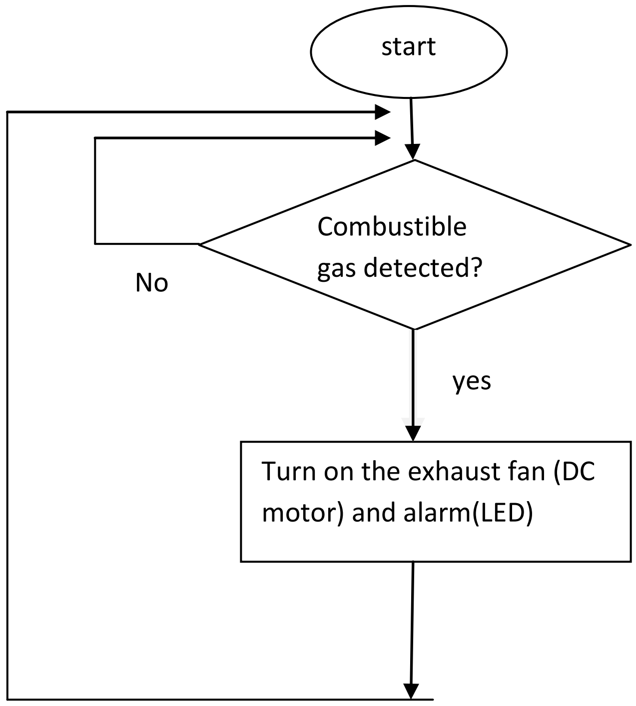


Figure 3 flow diagram of gas sensor

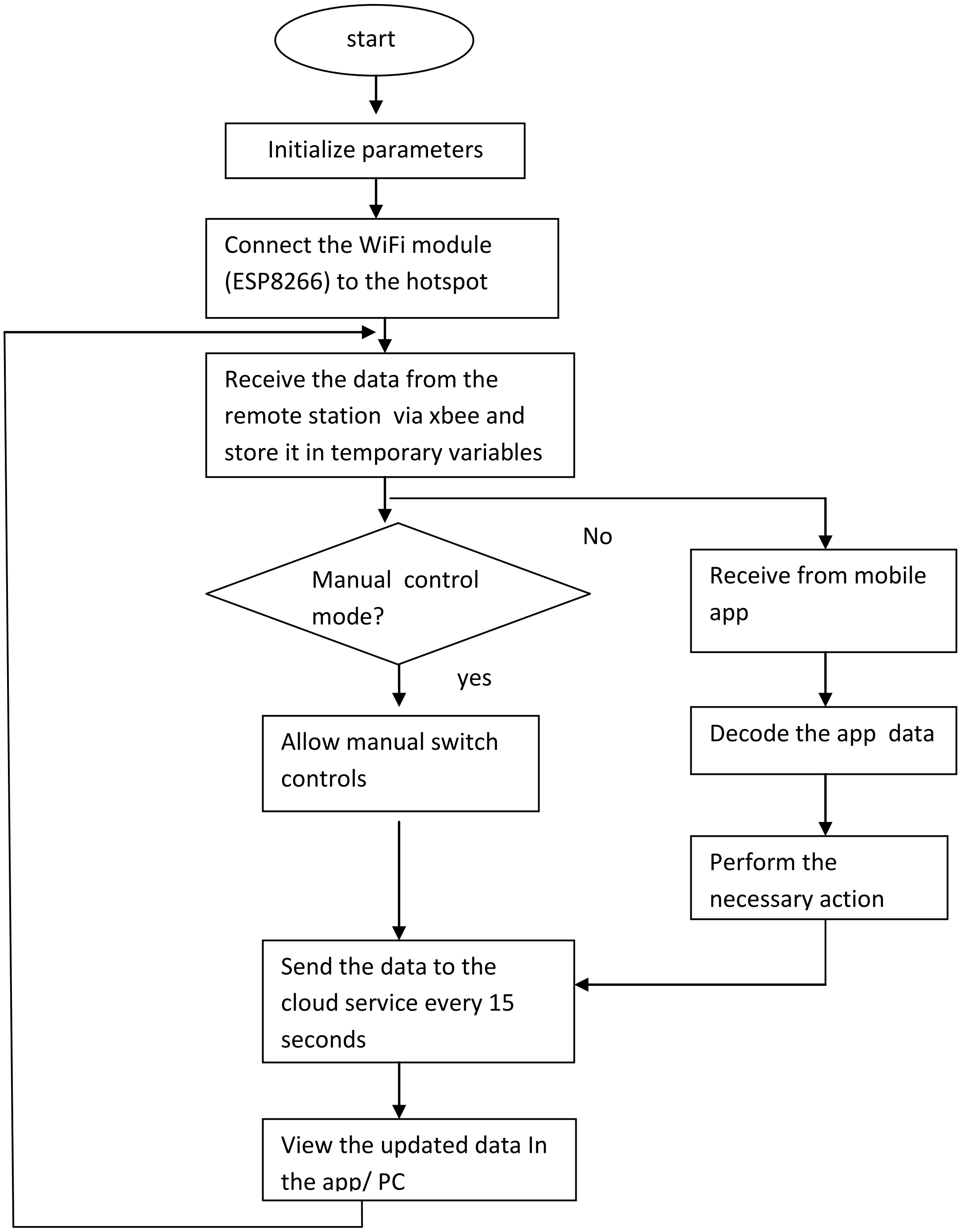


Figure 4 flow chart of Base station

The initialization of the parameters is done. The esp8266 (wi-fi module) is connected to the hotspot to enable the internet connection. The mode of operation is checked. Depending on the mode of operation the respective codes are executed.

Datasheets

BMP180(DIGITAL TEMPERATUR AND PRESSURE SENSOR)

Key features

Pressure range:	300 ... 1100hPa (+9000m ... -500m relating to sea level)
Supply voltage:	1.8 ... 3.6V (VDD)
	1.62V ... 3.6V (VDDIO)
Package:	LGA package with metal lid
Small footprint:	3.6mm x 3.8mm
Super-flat:	0.93mm height
Low power:	5µA at 1 sample / sec. in standard mode
Low noise:	0.06hPa (0.5m) in ultra low power mode
	0.02hPa (0.17m) advanced resolution mode

-Temperature measurement included

- I2C interface

- Fully calibrated

- Pb-free, halogen-free and RoHS compliant,

- MSL 1

Typical applications

- Enhancement of GPS navigation (dead-reckoning, slope detection, etc.)
- In- and out-door navigation
- Leisure and sports
- Weather forecast
- Vertical velocity indication (rise/sink speed)

General description

The BMP180 is designed to be connected directly to a microcontroller of a mobile device via the I2C bus. The pressure and temperature data has to be compensated by the calibration data of the E2PROM of the BMP180.

General function

The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with E2PROM and a serial I2C interface. The BMP180 delivers the uncompensated value of pressure and temperature. The E2PROM has stored 176 bit of individual calibration data. This is used to compensate offset, temperature dependence and other parameters of the sensor.

- UP = pressure data (16 to 19 bit)
- UT = temperature data (16 bit)

IR SENSOR

- **Key Features**
- 1) Applications - Obstacle Detector Sensor, Line Following Sensor, Proximity Sensor
- 2) 10-12cm range. Potentiometer for maximum range setting.
- 3) Can be used to differentiate between black and white (Can be used for line sensing)
- 4) Onboard LED indication for detection
- 5) Works on 5V or 3.3V input. TTL compatible output

MQ-2 GAS SENSOR

Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor’s conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

Characterestics:

*Good sensitivity to Combustible gas in wide range

* High sensitivity to LPG, Propane and Hydrogen

* Long life and low cost

* Simple drive circuit

Application:

* Domestic gas leakage detector

* Industrial Combustible gas detector

* Portable gas detector

RELAY(SRD-05VDC)

Main features

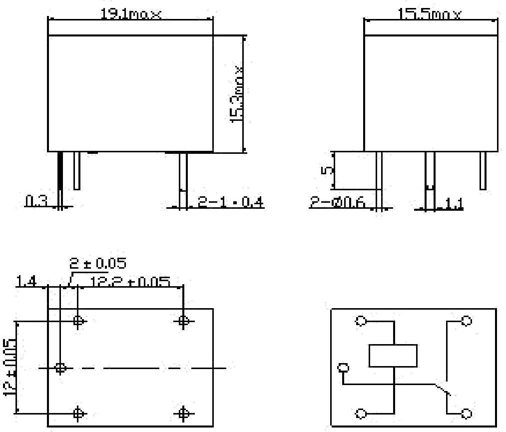
- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

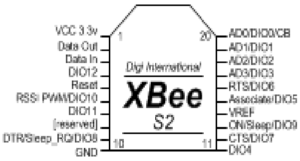
Application

- Domestic appliance, office machine, audio, equipment, automobile, etc.

(Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

DIMENSION(unit:mm)

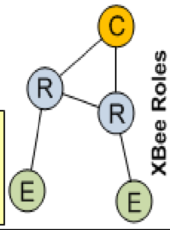




XBee S2 Quick Reference Guide

IEEE 802.15.4 = Zigbee Protocol. XBee is a microcontroller made by digi which uses the Zigbee protocol. The XBee uses 3.3V and has a smaller pin spacing than most breadboards/proto boards. Because of this, it is often useful to purchase a kit to interface the XBee with a breadboard.

Sept/2012 http://tunnelsup.com



XBee Roles

- Coordinator – 1 required in every network. In charge of setting up the network. Can never sleep
- Router – multiple may exist. Can relay signals from other routers/EPs. Can never sleep
- End Point – multiple may exist. Cannot relay signals. Can sleep to save power

Specs

Operating Voltage: 2.1 – 3.6V	Digital I/O pins: 11	RF Data Rate: 250kbps
Operating Current: 40mA@3.3V	Analog input pins: 4	Throughput speed: 35kbps
Indoor range: 40 Meters	Mesh routable	Frequency: ISM 2.4GHz
Line of sight range: 120 Meters	Self Healing network	OK Temp: -40 to 85C
Max Analog Pin Reading: 1.2V	Firmware: ZB ZigBee	

XBee Modes

Transparent – Communication through the XBee. If data is not generated from the XBee itself then both XBees should be set to AT.
Command – Communication to the XBee. If one XBee is sensing data, that XBee should be in AT mode while the receiving one should be in API mode.

XBee Setup

Connect the XBee to a TTL Serial FTDI adapter – OR – Arduino hack: Connect RX to RX, TX to TX, RESET to ground to bypass the Arduino entirely and get serial to XBee.
Use the free X-CTU software to configure the XBee.
Baud: 9600 – FC: Hardware – Data Bits: 8 – Parity: None – Stop Bits: 1

Basic Settings

PAN ID – The network to communicate over. If 0, the XBee will join any.
DH/DL – Destination Serial number. Used to send to a specific XBee's Serial. Set to 0 to send to just the Coorindator. Set to 0x000000000000FFFF to broadcast.
JV – Router/EP should be set to 1 so it rejoins the network on startup

Pin Settings

For pin settings to work, receiver XBee must be in API mode
D0 – Set pin 0 to start sensing
IR – Collect data on sensing pins every XX millisecs

Arduino Connectivity:
Arduino TX connects to XBee RX (Data in)
Arduino RX connects to XBee TX (Data out)

Arduino Integration:
Data sent to Serial.print() will go out TX port of Arduino which is then connected to the RX port of XBee. If XBee is in AT mode it will transmit it wirelessly. Data received from XBee will be sent to the Serial.

Arduino Example: Read an analog value using API

```
// Remote XBee: AT, Base XBee: API
if (Serial.available() >= 21) {  // Make sure the frame is all there
 if (Serial.read() == 0x7E) {    // 7E is the start byte
  for (int i = 1; i<19; i++) {   // Skip ahead to the analog data
    byte discardByte = Serial.read();
  }
  int analogMSB = Serial.read(); // Read the first analog byte data
  int analogLSB = Serial.read(); // Read the second byte
  int analogReading = analogLSB + (analogMSB * 256);
 }
}
```

API format for Remote AT Command Request

Byte	Example	Description
0	0x7e	Start byte – Indicates beginning of data frame
1	0x00	Length – Number of bytes (ChecksumByte# – 1 – 2)
2	0x10	
3	0x17	Frame type - 0x17 means this is a AT command Request
4	0x52	Frame ID – Command sequence number
5	0x00	64-bit Destination Address (Serial Number)
6	0x13	MSB is byte 5, LSB is byte 12
7	0xA2	
8	0x00	0x0000000000000000 = Coordinator
9	0x40	0x000000000000FFFF = Broadcast
10	0x77	
11	0x9C	
12	0x49	
13	0xFF	Destination Network Address
14	0xFE	(Set to 0xFFFE to send a broadcast)
15	0x02	Remote command options (set to 0x02 to apply changes)
16	0x44 (D)	AT Command Name (Two ASCII characters)
17	0x02 (2)	
18	0x04	Command Parameter (queries if not present)
19	0XF5	Checksum

Arduino Example: Change the pin setting on a remote Xbee

```
// Remote XBee: AT, Base XBee: API
Serial.write(0x7E);  // Sync up the start byte
Serial.write((byte)0x0);   // Length MSB (always 0)
Serial.write(0x10);  // Length LSB
Serial.write(0x17);  // 0x17 is the frame ID for sending an AT command
Serial.write((byte)0x0);   // Frame ID (no reply needed)
Serial.write((byte)00);   // Send the 64 bit destination address
Serial.write((byte)00);   // (Sending 0x000000000000FFFF (broadcast))
Serial.write((byte)00);
Serial.write((byte)00);
Serial.write((byte)00);
Serial.write((byte)00);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF); // Destination Network
Serial.write(0xFE); // (Set to 0xFFFE if unknown)
Serial.write(0x02); // Set to 0x02 to apply these changes
Serial.write('D');   // AT Command: D1
Serial.write('1');
Serial.write(0x05); // Set D1 to be 5 (Digital Out HIGH)
long chexsum = 0x17 + 0xFF + 0xFF +0xFF + 0xFE + 0x02 + 'D' + '1' + 0x05;
Serial.write( 0xFF – (chexsum & 0xFF)); // Checksum
```

API format for I/O Data Sample RX Indicaton

Byte	Example	Description
0	0x7e	Start byte – Indicates beginning of data frame
1	0x00	Length – Number of bytes (ChecksumByte# – 1 – 2)
2	0x14	
3	0x92	Frame type - 0x92 indicates this will be a data sample
4	0x00	64-bit Source Address (Serial Number)
5	0x13	MSB is byte 4, LSB is byte 11
6	0xA2	
7	0x00	
8	0x40	
9	0x77	
10	0x9C	
11	0x49	
12	0x36	Source Network Address – 16 Bit
13	0x6A	
14	0x01	Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet
15	0x01	Number of sample sets. Always set to 1 due to XBEE limitations
16	0x00	Digital Channel Mask – Indicates which pins are set to DIO
17	0x20	
18	0x01	Analog Channel Mask – Indicates which pins are set to ADC
19	0x00	Digital Sample Data (if any) – Reads the same as Digital Mask
20	0x14	
21	0x04	Analog Sample data (if any)
22	0x25	There will be two bytes here for every pin set for ADC
23	0xF5	Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)

Sleep Mode

Endpoints can sleep to save power. An endpoint that only wakes up every 5 minutes to send data may only be awake for 6 seconds a day.
SM – 4 = Cyclic sleep
SP – Sleep time (up to 28 secs)
SN – Number of sleep cycles
ST – Time awake

Pin I/O Options

0 – Disabled
1 – N/A
2 – ADC
3 – Digital IN
4 – Digital OUT, LOW
5 – Digital OUT, HIGH

Digital Ch Mask

First Byte
n/a n/a n/a D12 D11 D10 n/a n/a
Second Byte
D7 D6 D5 D4 D3 D2 D1 D0
Example:
0x00 0x0D = 0000 0000 0000 1101
Pins D3, D2 and D0

Analog Ch Mask

(volt) n/a n/a n/a A3 A2 A1 A0
Example:
0x05 = 0000 0101 = Pin A2 and A0

Notes