# IMPLEMENTATION OF DATA LOGGING WITH WIRELESS SENSOR NETWORKS

BY


**OGOKE ONYEKACHI CHIADIKOBI 20111787293**

**AJOKU IKENNA PETRUS 20111787063**

SUPERVISED BY:

**ENGR. DR. C.C. MBAOCHA**

**SUBMITTED TO**

ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTMENT

SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY

FEDERAL UNIVERSITY OF TECHNOLOGY, OWERRI


IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR DEGREE (B.ENG) IN ELECTRICAL AND ELECTRONIC ENGINEERING

OCTOBER 2016.

# CERTIFICATION

This is to certify that this work "IMPLEMENTATION OF DATA LOGGING WITH WIRELESS SENSOR NETWORK" was carried out by AJOKU PETRUS IKENNA - 20111787063 and OGOKE, ONYEKACHI CHIADIKOBI – 20111787293, in partial fulfillment of the degree of the requirements for the award of bachelor of Engineering (B.Eng) degree in partial award of bachelor degree (B.Eng) in Electrical and Electronic Engineering, Federal University of Technology, Owerri.

Approved by;

______________________ ____________________

Engr. Dr. C. C. Mbaocha Date

(Project Supervisor)

______________________ ____________________

Engr. Dr. D. O. Dike Date

(Head of Department)

______________________ ____________________

External Examiner Date

# DEDICATION

I dedicate this work to my Almighty God, my creator and helper, my parents and siblings for their unflinching support, to all Electrical and Electronic Engineering students and to all those that love the work of Engineers.

# ACKNOWLEDGEMENT

We sincerely appreciate our parents and siblings whose financial and moral supports were pivotal to the actualization of this report.

We thank our supervisor Engr. Dr. C. C. Mbaocha and his assistant Engr. Akande for their guidance and immense contributions towards the successful completion of this work.

We also want to use this opportunity to thank our Course Adviser, Engr. I.F. Ezebili for his support and inspiration he gave us throughout our stay in FUTO.

We also want to acknowledge all the Lecturers of the department of Electrical Electronic Engineering, without whom I would not have been where I am today.

We cannot conclude without acknowledging the Creator of all there is, God the Father Almighty for the graces he gave to us during the compilation of this report.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

The goal of this project was to design a data logging interface using wireless network principle in order to facilitate the process of monitoring physical parameters such as temperature, pressure, humidity, e.tc in the environment. There is a growing desire to find an efficient and cost effective means to monitor these physical parameters, especially at home and industries. The absence of a means to monitor these quantities in an environment has led to so many inadequacies.

The method adopted is a star network based topology in which each sensor node has a direct communication to the computer that performs the data logging function and also stores the information in its database. The design uses both Arduino platform for the sensor network which requires of C programming and a python platform for the data logging interface which requires the knowledge of python scripting language.

Each sensor node was connected wirelessly to the data logging device using the Wi-Fi protocol such that each device used in the network is Wi-Fi enabled including database server device.

The results from the project shows that a user-friendly user interface was created from which can be used to perform further actions by supervisors

# CHAPTER ONE

# INTRODUCTION

## 1.1. BACKGROUND OF STUDY

Advances in wireless communication technologies have made it possible for electronic devices of various size and complexity to interface and share information amongst them and other systems connected to them. These technologies led to the implementation of wireless sensor networks, allowing easily configured, adaptable sensors to be placed almost anywhere, and their observations similarly transported over large distances via wireless networks [4]. Database technologies are beginning to have a significant impact in the emerging area of wireless sensor networks (sensornets). The sensornet community has embraced declarative queries as a key programming paradigm for large sets of sensors. As seen in academia call for papers leading conferences and workshops in the sensornet area [31] and other research publications [33]. Therefore, implementing data logging services i.e. monitoring services with help of the improving technologies in wireless sensor networks which particularly focuses on dissemination and collection of environmental data in order to monitor the various states of matter as well as natural and human-made elements is now the emerging technological idea turning our society around especially in the industry and engineering fields as a whole

## 1.2. STATEMENT OF PROBLEM

In a digitized world, there is always a need to know what happening around us and motivated by the predicted tremendous growth in this field, the interesting applications, engaging challenges of WSNs. As they say, Knowledge is power. How do people know what is happening around them as humans depend on their senses; by using the sense organs; the eyes, ears, nose, and our skin to sense it.

But consider a situation where there is a need to know, monitor, store, and if possible control the events or environmental changes happening in a place where supervision may or may not be physically available or when information needs to be collected as accurately and quickly as possible, one may decide make use of electronic devices to carry out this operation and one of which is a SENSOR.

However, it doesn't just end there, the information collected by this sensor needs to be gathered, stored, analysed and outputted. Furthermore, in a situation or a process maybe an industrial plant process where a lot of physical quantities need to be acquired by various sensors, analysed, outputted and monitored. It definitely takes more than just a sensor or sensors.

Therefore, a data logging system would be needed with the help of a sensor network or a network of sensors either wired or wireless but because of the difficulty or complexity of cabling, someone may want to go the easy way out which is **implementing this database logging system using wireless sensor network,** especially in an industrial plant process but the database logging system is not just limited to an industrial plant process particularly. It can be used in different processes e.g. environment monitoring, homeland security and disaster relief operations.

## 1.3 OBJECTIVES OF THIS PROJECT

The Primary Objective of this project is to design a data logging system based on Wireless Network principles.

The Secondary Objectives of this project are stated below;

- To design a network of wireless sensor nodes with Arduino boards and a central computer as the gateway to a server using a star topology

- To design a data logging interface on the server using the python scripting program even while enabling communication between all the devices using Wi-Fi protocol compliance radio devices

## 1.4 JUSTIFICATION

The use of data logging is so common in modern day applications that are hard to find where it is not applied in modern day social life from environmental monitoring process to an industrial plant process to a homeland security system to even vehicles and so on. But depending on your specific process, data logging can have varying significance for the user. For some processes, the user is simply concerned that the products were exposed to the process and the particulars (though noteworthy) are simply an afterthought. While some users are concerned with how data logging uses the parameters of nodes joining the cluster so that the data attributes are selected and stored in a logged format for further evaluation and usage. Logging simply refers to the technique that models the data and information in a dimensional construct that is easy to store and retrieve. The data collection technique is being employed to store and collect data items and parameters on a database server [20]. This is a very important significance of the project.

However, the use of Wireless Sensor Networks (WSNs) in data logging in this modern day social life of advancing technology as one of the most exciting and growing branches of wireless communication. As predictions show the market is soon to cross the billion-dollar mark. Our world is going toward autonomous, sustainable and environment conscious systems with technologies which can monitor the environment, provide valuable and timely measurements. WSNs enable such drives in a non-intrusive manner yet with extended coverage of area thanks to their underlining distributed and miniature nature [13].

The first, and for many the most notable, feature is their cost. Using low-power and relatively inexpensive microcontrollers and transceivers, the sensor nodes used in wireless sensor networks are often less than one hundred dollars in cost. This opens the doors for many commercial or military applications, as the relatively diminutive cost of nodes allows for not only large numbers of sensors to be deployed, but also for large numbers of sensors to be lost. For example, sensor nodes can be dropped from a plane, allowing widespread coverage of an area with minimal effort involved in positioning the individual nodes. The relatively low cost of the sensors allows for some nodes to be damaged or lost without compromising the system, unlike larger, more centralized sensors [37].

Another advantage wireless sensor networks hold over traditional wireless sensing technology lies in the mesh networking scheme they employ. Due to the nature of RF communication, transmitting data from one point to another using a mesh network takes less energy that transmitting directly between the two points. While embedded systems must respect their power envelope, the overall energy spent in RF communication is lower in a mesh networking scenario than using traditional point-to-point communication [37]. Sensor networks can also offer better coverage than more centralized sensing technology.

Utilizing node cost advantage and mesh networking, organizations can deploy more sensors using a wireless sensor network than they could using more traditional technology. This decreases the overall signal-to-noise ratio of the system, increasing the amount of usable data. For all these reasons and more, wireless sensor networks offer many possibilities previously unavailable with traditional sensor technology [37].

However, this growing trend in data logging with the help of Wireless Sensor Networks have been perceived to have various and vast applications or where it can be applied and they include;

- Unattended Weather station recording (such as wind speed/ direction, temperature, relative humidity, solar radiation).
- Unattended hydrographic recording (such as water level, water depth, water flow, water pH, water conductivity).
- Unattended soil moisture level recording.
- Unattended gas pressure recording
- Offshore buoysfor recording a variety of environmental conditions
- Road Traffic counting
- Measure temperatures (humidity, etc.) of perishables during shipments
- Measure variations in light intensity.
- Process monitoring for maintenance and troubleshooting applications
- Process monitoring to verify warranty conditions
- Wildlife Research with pop-up archival tags
- Measure variations and handling shock (drop height) environment of distribution packaging
- Tank level monitoring
- Deformation monitoring of any object with geodetic or geotechnical sensors controlled by an automatic deformation monitoring system.
- Environmental monitoring
- Vehicle Testing (including crash testing)
- Motor Racing
- Monitoring of relay status in railway signalling
- For science education enabling ‘measurement’, ‘scientific investigation’ and an appreciation of ‘change’

- Record trend data at regular intervals in veterinary vital signs monitoring.
- Load profile recording for energy consumption management
- Temperature, humidity and power use for heating and air conditioning efficiency studies
- Digital electronic bus sniffer for debug and validation

## 1.5 SCOPE OF STUDY

This study is restricted to the data logging system based on Wireless Network principles.

The design is such that sensors will be interfaced together with the Wi-Fi module, the batteries and the ATmega328P microcontrollers using an Arduino board as well as the Arduino IDE create a sensor node.

More than one sensor nodes are intended to be used in a star topology wirelessly communicating with a server for storage, all these sensors working for the sole purpose of data acquisition or data logging of the information gathered from each of the sensor nodes.

The programming language used to program each of the components in the sensor node is Arduino programming which basically utilizes c programming. While the programming language used to program the data logging interface is the python scripting language. Also this data logging interface is created using an excel datasheet and an excel graph in order to make the interface user friendly.

All the data transmitted to the data logging system were done within the 2.4GHz unlicensed frequency band, which operates between 2,400MHz to 2,483.5MHz only.

## 1.6. REPORT ORGANIZATION

The report is written with the aim of performing data logging using Wireless Sensor Network principle in a simple and understandable manner such that the individuals reading the physical parameters from the data logging interface appreciates the work done in designing and implementing this system.

This report is an amalgamation of five chapters that contains and elaborates precise areas that are related to this project.

Chapter one; Introduction, gives a brief description of the work, states the problem to be solved, the significance of the project and the Scope and limitations of this project.

Chapter two; Literature review, describes the work at hand and its evolution. It states previous related works that have been done by other designers; how it was done and the materials used. It also contains a brief description of the methods and materials adopted for this project.

Chapter three; Methodology, materials and design gives an insight about the materials used in the design (both hardware and software), the technology deployed and also describes the methods implemented in the design.

Chapter four; Results and discussion, elaborates on the results and comments on the functionality of the system.

Chapter five; Conclusion and Recommendation, gives a summary of the work done, the constraint encountered during the project, the limitation of the project and proffers recommendations on how best data logging with Wireless Sensor Network can be further developed and deployed on a broader scale.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 BRIEF HISTORY

Data Acquisition (often abbreviated as DAQ) has become an industry standard term that refers to almost any type of computer based system comprised of analogue and/or digital inputs and outputs. Today most systems are based upon PCs, but there is still a large market for systems based upon other platforms such as VME and PXI as well as those based on proprietary embedded controllers.

It was not that long ago the stereotype of an engineer was someone in a white lab coat watching one or more meters and writing results on a clipboard. The advent of extremely low cost computers, combined with the development of a wide variety of powerful data acquisition interfaces has driven this stereotype farther back in history than black and white movies. Also, most college graduates in science and engineering disciplines have enough programming training and experience to make the programming required for DAQ applications quite straightforward.

There are a wide variety of interfaces used to connect data acquisition hardware to computers. Most popular among these are the PCI bus, Ethernet and USB protocols. In addition to interfaces connecting DAQ systems to standard PCs, there is a rapidly growing market for embedded systems where a program is written and deployed on the data acquisition system itself, allowing it to run standalone.

In the early 1980's, when the PC-based DAQ market was in its infancy, most analog input and output devices offered 12-bit resolution. There were also some lower speed products with greater than 12-bit resolution and a number of high speed products offering resolution in the 8 to 10-bit range. Today, the technology has changed and the standard resolution is 16-bit, with DAQ products offering resolutions up to 24-bit while some of the higher speed, lower resolution products are even challenging the performance of low end digital oscilloscopes [40].

But, in recent times due to advances in technology data logging has employed the use of wireless sensor networks in its implementation especially when varieties of data from different sources are to be logged to achieve a particular goal in a given environment, habitat, factory etc.

Although origins of WSN can be traced back to the beginning of 1980s research on Distributed Sensor Network (DSN) backed by Advanced Research Projects Agency (ARPA). Operating System (OS) suited for communication from CMU and helicopter tracking application at MIT were a couple of the pioneering works [41]. At the time such initiatives faced the obstacle of underdeveloped sensing and communication platforms. The pre-cursor to the current DARPA, the US military advance research unit.

In the late 1990s advances in wireless technologies, very large scale integration of components and Micro electro-mechanicals (MEMs) systems revived the attention given to WSNs. Notable projects included, the Wireless Integrated Network Sensors (WINSs) at the rock well science centre UCLA [42] and the famous Smart dust project, sensing nodes with grain size at University of California at Berkeley [43].

In continuation, the past decade has been accelerated growth period for WSNs in every aspect. From the hardware perspective advance in miniaturization and integration of sensors, processors and actuators including widely available of-the-shelf platforms such as micaz and telosB were achieved. In terms of software development OSs targeting embedded systems and sensor networks such as TinyOS and Contiki have evolved into existence. Accompanying communication protocols have been standardized, including IEEE 802.15.4 Media Access Control (MAC) and Routing Protocol for Low-Power and Lossy Networks (RPL).

The academia has also been actively engaged in tackling new challenges that accompany WSNs and NCSs. Some works include sampled system control and stability [44], network life time, coverage and architecture [45], co-design methodologies for control and networking [46, 47, 48], networking protocol evaluation and optimization [49]. In general the surveys [50, 51] present the researches around WSNs while [52] summarize NCSs researches.

The ultimate goal and vision of WSNs field is to make sensor networks a fabric of social life. Minute sensors will be embedded in every conceivable manner to provide accurate timely information to sinks nodes. These nodes interested with the information will enable proactive response hence improving the user experience, quality of life as well as fortunes spent in industries and manufacturing. In [53] picturesque vision for the year 2050 is introduced.

Aside from visionary predictions recent times some actual practical projects worth mentioning have been implemented. To name a few large scale deployments, habitat monitoring [54], structural monitoring project of the golden gate bridge [55, 56], healthcare [57], large scale water monitoring [58], and the HYDROBIONETs (HBNs) for large scale plants [13]. From the given examples of where and how wireless sensor networks have been implemented, the goal and vision of the WSNs field of sensor network becoming part of social lifeis already becoming a reality. For example, Data logging is more like processing information of a physical quantity but this time collected from a sensor. In other words, data logging is social life made easy for day to day work with the help of wireless sensor network.

## 2.2. REVIEW OF RELATED LITERATURE:

A wireless sensor network (WSN) consists of several small, inexpensive sensor nodes that communicate detected events wirelessly. Quite often, WSNs are used to monitor certain delicate situations such as floods, volcanoes, radioactive materials, wildfire, ethylene, methane, *etc.* However, they can also be used in other less critical cases, like the one covered in this work, where the information to be gathered is so exhaustive that several simple sensors needed. In these cases, it is the information flow that becomes critical.

Various researches, projects and several studies have been carried out covering this topic some of which are stated below;

### 2.2.1. Wireless Sensor Network for Monitoring Applications

They addressed the problem of detecting physical intrusion by designing and building a wireless sensor network system which was done following an exploration of personal area networks and mesh networking. To that end, the network employed sensor nodes equipped with motion sensors and accelerometers. The network communicated with a generic infrastructure, adaptable to future wireless sensor projects, which stored sensor data in a database. Also included was a user interface to monitor the status of the entire system. From results, the proof of concept demonstrates the viability

of WPAN-based wireless sensor networks despite the immaturity of the Tmote Sky platform.

[Chanin J.I, Halloran A.R, Agu E., Lou .W (2010)]

### 2.2.2. Process Control over Wireless Sensor Network

He made an experimental study on how wireless sensor network and network controlled systems can be used to improve the performance of controllers in microbiological processes for water treatment plants in the industries. He adopted the IEEE 802.15.4 standard and IETF RPL (Routing Protocol for Low power and lossy networks) to provide stable communication that is reliable enough for actual media access layers for the WSNs while the IETF RPL defines the functionality of the routing layer. The study of the performances of controller in the different scenarios also confirm the viability of applying feedback controller over the protocols IEEE 802.15.4 and RPL. Introducing the wireless network doesn't introduce unexpected phenomena on the theoretical control system only the requirement that real implementation need to be conservative than theoretical predictions. The study of the results in the presence of network delay was shown to eject the system with faster dynamics more which is greatly rejected on percent overshoot. The practical implication of this is if the WSN is expected to introduce delay close to the system limit then viability of the system need to be guaranteed by the use of multi-tiered network topology as explained. Last but not least the study of the results in the presence of packet drop has revealed for the theoretical stability prediction to hold the underlining packet loss probability must be uniformly distributed.

[ Zelalem Teffera (2013)].[13]

### 2.2.3. Wireless Sensor Network to Monitor Spatio-Temporal Thermal Comfort Of Polyhouse Environment

They conducted a research on how electronic systems are designed to provide thermal comfort to the crops of a polyhouse environment. In their work, wireless sensor node was designed to monitor the thermal status of polyhouse environment in spatio-temporal domain. A precision temperature sensor, LM35 was employed to read the thermal status. Also, signal conditioning circuit was deployed in the design which was wired about TLC 271, which operates on one power supply only and exhibits very high input impedance with rail to rail input.

A Zigbee model was deployed as the RF module in the design to ensure wireless communication. Also, each sensor node was assigned a network ID and it communicates its data to their base station with their respective ID in a star topology. The Zigbee RF module tends to be more complex than other RF modules because it has its own operating system which can be a bit difficult to use for some designers. In his results, the temperature reading from the sensors in the polyhouse environment shows a close agreement with what was observed from a standard digital thermometer. Therefore, the sensor network designed can be said to be of standard with other standard temperature monitoring devices.

[Pawar A.M., Patil S.N., Powar A.S., and Ladgankar B.P. (2013)]

### 2.2.4. SEP: Wireless Sensor Project

He designed a wireless sensor network that collects data from sensors, and transmits the data wirelessly from each node to a PC. In his work, the wireless protocol was designed as a star network using a Time Division Multiple Access (TDMA) approach. A Master-Slave TDMA communication technique was used in which the master transmits a beacon frame, which also contains communications to a specific node. Once this beacon frame has been received by a slave, the slave will wait for a fixed period of time before transmitting its data to the master. The nodes used in this design were pre- programmed with the device address and slot allocation. This helps to avoid overhead due to authentication and data collision.

The designer adopted a radio frequency band of 2.4GHZ for a Bluetooth module. This is an unlicensed frequency band that is suitable for the TDMA approach because more channels are available in this frequency band.

Although the Project was not completed as at the time of his report, the result he got shows sequence of temperature readings at a specific interval of time which partially correspond to the values he got while using a standard thermometer.

For this design, an **Atmel's ATMega328p Microcontroller chip** was used for the controlling unit. This is a 28 pin DIP package which combines 32Kb ISP flash memory with read-while-write capabilities, 1kb EEPROM, 2kb SRAM, 23 general purpose input/output lines, 6 of which are channels for the 10-bit Analogue-to-Digital Converter. It runs up to 20MHz with external crystal. It requires 1.8 volts to 5 volts for its operating voltage. This ATMega328P can be programmed using an Arduino. Arduino is an open source project that creates microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

The ATMega328p microcontroller chip is inserted on the Arduino board, which features serial communication interfaces, including Universal Serial Bus on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an Integrated Development Environment (IDE) based on a programming language named processing, which also supports the language Java, C and C++.

The Wireless Communication protocol adopted in this design is **IEEE 802.11b (WLAN) protocol**. The 802.11b WLAN protocol can achieve data rates of 11Mbps, using the 2.4GHz frequency band. It is widely used in PC networking. 802.11b uses timeslots of 20Gs, and a CSMA/CA transmission scheme. Between each transmission, a back-off time must take place.
This time is frozen whilst the transmission medium is busy. This back off time is a random value between 31 and 1023 time slots (640Gs and 20460Gs).
The advantages of this protocol are:

1. High data rate
2. Priority possible for delay-sensitive packets

## 2.3. ALTERNATE WIRELESS NETWORKING PROTOCOLS FOR THE DESIGN

There are several wireless networking standards defined, and others still being finalized that can be used for this project. This wireless networking protocols are selected based on their range, coverage area, frequency band of operation, data rate, latency, power consumption, etc. below are some of the wireless networking protocols that can be used in this project with the advantages and disadvantages of each protocol;

### 2.3.1. Bluetooth Wireless Protocol:

Bluetooth is an open specification for short range voice and data communications [11]. It operates in the 2.4GHz ISM band [12]. Bluetooth uses a frequency hopping sequence based on the master clock and master address [12]. Devices are authenticated using a challenge response mechanism based on a user defined PIN. It is also possible to encrypt the link [12]. Power consumption can be up to 100mW. It has a typical range of less than 10m and a peak data rate of 1Mbps is available [12]. Bluetooth uses timeslots of 625s [12]. The master may only begin a transmission on an even timeslot, and slaves may only begin transmission in an odd timeslot. A slave may only transmit (normally) if it received a packet from the master in the previous timeslot [12]. This means a maximum of 16 slaves to meet the specifications [4].

**Advantages of Bluetooth Protocol**

1. Frequency hopping provides resilience to DoS attacks.
2. An Ack/NAck is used to confirm successful data transmission.
3. No transmission contention delays [4]

**Disadvantages of Bluetooth Protocol**

1. Bluetooth devices must remain in an active state for the entire time they are connected to the network [4].

### 2.3.2. ZigBee Wireless Protocol:

The ZigBee protocol is designed for use in wireless sensor networks. ZigBee uses the 2.4 GHz frequency band for higher bandwidth and worldwide acceptance along with the ETSI 868 MHz and the FCC 900 MHz bands [16]. ZigBee uses a CSMA/CA transmission system, based on the IEEE 802.15.4 MAC layer [16]. The MAC layer provides AES encryption. ZigBee devices have been proven to communicate effectively in environments with significant interference [17]. ZigBee supports a data rate of 250kbps

**Advantages of ZigBee Wireless Protocol**

1. Standard based security protocol
2. Designed for low power applications
3. Mesh network means better range when many nodes are involved.

**Disadvantages of ZigBee Wireless Protocol**

1. Complex (relative to required system)
2. Transmission contention (CSMA/CA) means no guarantee of how long a transmission can take.
3. Low data rate

## 2.4. Radio Frequency Regulations

There are two frequency bands which are free to use for Industrial, Scientific and Medical applications. These bands are at 868MHz, and 2.4GHz. Each band has different regulations, which are defined by the European Telecommunications Standards Institute (ETSI).

For this project, the frequency band adopted is the 2.4GHz (between 2400MHz to 2483.5MHz), which is explained further below[4].

### 2.4.1 2.4GHz Frequency Band Regulations

The 2.4GHz band between 2400MHz and 2483.5MHz is available for generic use [15]. This band has no duty cycle restrictions [15]. LBT is not required [15]. The output power is restricted to 10mW (20dBm) for non-spread spectrum uses [15]. If FHSS is used, then an output power of 100mW (10dBm) is allowed [15]. For non-adaptive FHSS, the system must *"...make use of at least 15 well defined, non-overlapping hopping channels separated by the channel bandwidth as measured at 20 dB below peak power… the minimum channel separation shall be 1 MHz, while the dwell time per channel shall not exceed 0,4 s…* each channel of the hopping sequence shall be occupied at least once during a period not exceeding four times the product of the dwell time per hop and the number of channels." [15] This gives a maximum of 83 channels [4].

**Advantages of 2.4GHz frequency band**

1. More channels are available.
2. Higher output power available for FHSS systems.
3. Smaller antenna used. (Due to physical laws).
4. No duty cycle or LBT limitations [4]

## 2.5. COMMUNICATION PROTOCOLS

### 2.5.1. SPI PROTOCOL:

The serial peripheral interface (SPI) is a synchronous data link. It is invented by Motorola. Its operation is in fully duplex mode. Single master and short distance communication link uses this interface. In this link the device communicates in master- slave configuration.

### 2.4.1. I2C PROTOCOL

I²C is a multi-master protocol that uses 2 signal lines. The two I²C signals are called 'serial data' (SDA) and 'serial clock' (SCL). There is no need of chip select (slave select) or arbitration logic

### 2.4.2. UART PROTOCOL

UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C Data Terminal Equipment ( DTE ) interface so that it can "talk" to and exchange data with modems and other serial devices.

# CHAPTER THREE

# METHODOLOGY, MATERIALS AND DESIGN

The aim of this project is to design a data logging system based on Wireless Network principles.

The Objectives of the project is as follows;

1. To design a network of wireless sensor nodes with Arduino boards and a central computer as the gateway to a server using a star topology
2. To design a data logging interface on the server using the python scripting program even while enabling communication between all the devices using Wi-Fi protocol compliance radio devices

## 3.1 DESIGN REQUIREMENTS AND SPECIFICATION

### 1. CONTROLLER/PROCESSOR MODULE REQUIREMENTS

To control the activities of the sensor node and its mode operation effectively i.e. how each module should function collectively as a node, a high performance microcontroller with such abilities is considered. The following are the requirements of the microcontroller to be considered;

- High performance
- Programming ease
- Support for peripherals
- Low energy consumption
- The number of external components required for the sensor node
- Voltage and current requirements

## 2. COMMUNICATION MODULE REQUIREMENTS

To communicate over a wireless link, protocols as well as the devices that support these wireless communication protocols must be considered carefully. The following requirements were put into consideration before choosing the wifi esp-01 module;

- Over 100m line of sight coverage
- Little power down leakage current
- Supports embedded system protocol interface e.g SPI, UART, I2C etc,
- Low standby power consumption

## 3. SENSOR REQUIREMENTS

Having decided that no specific target application is intended but a general real time monitoring application such that the project could be used in environmental, industrial and medical applications, three sensors as well as sensor nodes were considered

- Air-pressure sensor
- Gas sensor
- Temperature and humidity sensor

## 4. SENSOR NODE REQUIREMENTS

Considering all the modules and their requirements needed for the sensor node and the fact that the Arduino board where the microcontroller was programmed all needed to be integrated in the sensor node. However, the following requirements is being considered;

- A general regulated 5v power source for all the modules in the system

- A regulated 3.3v circuit for the specified communication module
- A long-lasting battery power source with low internal resistance
- Resistors, 10k, 1k, 2.2k, 3.3k, 220 ohms resistors to replicate the Arduino board on the sensor node as well as manipulate some I/O ports of the microcontroller to suite our design
- Capacitors 22pF to replicate the Arduino board on the sensor node
- 16MHz oscillator to create clocking circuit for the microcontroller on the sensor node
- Voltage regulators for the power source

## 5. NETWORK REQUIREMENTS

As already stated in the design objectives, topology to be used is the star topology. The following requirements is needed create our network

- Three (3) wi-fi enabled sensor nodes since wi-fi protocol is the desired communication protocol
- A Laptop PC  wi-fi enabled system as the Master computer in the topology

## 6. DATABASE SERVER REQUIREMENTS

Since the database server will also be serving as the whole system interface, then basically software requirements is what will be needed. The laptop PC wi-fi enabled system has a large hard-disk external storage. Therefore, the intent is to use that master PC system as the database server and the user interface. The requirements to achieve this is as follows;

- Python Scripting software

- A python written program on python scripting software to create the database server

## 3.2. SOFTWARE PLATFORMS USED IN THE DESIGN

For the design and implementation of this project, two software platforms were used for interfacing, programming, and communicating between component parts of the wireless sensor network as well as displaying the data received by the server (PC) during data logging process. They include;

1. Arduino platform
2. Python platform

### 3.2.1. The Arduino Platform

Arduino was developed to teach Interaction Design [3], which means that it required the ability to sense the surroundings and do something about it. The platform is equipped with simple digital and analogue input/output interfaces, which can be programmed to sense or react to some events. The C programming langiage is the software language used in the Arduino environment to program ATmega328P microcontrollers. The Arduino Integrated Development Environment (Arduino IDE) makes the programming easier and less complex.

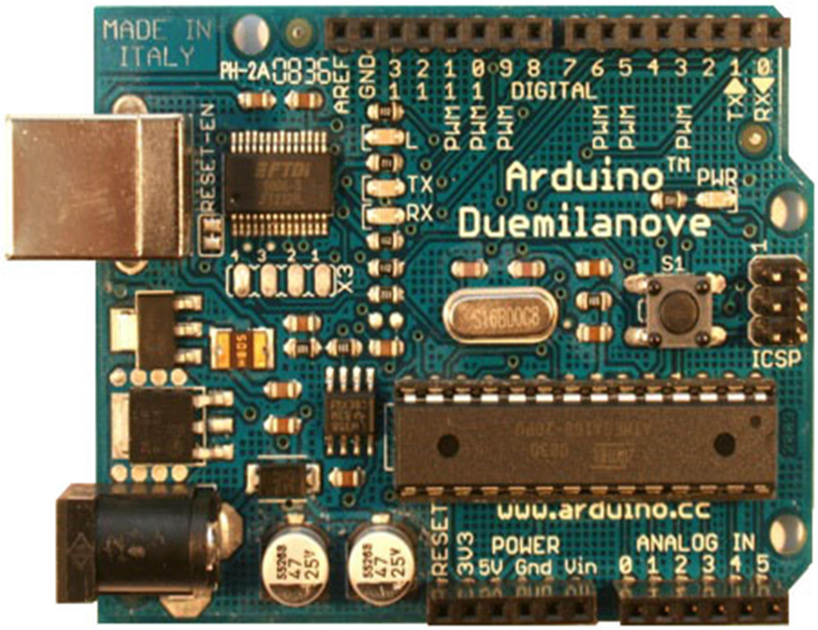



Figure 3.1 Arduino Duemilanove board

There are numerous sensors and actuators that work with Arduino. In relation to sensors: temperature, air pollution, light, GPS modules and sound are among the popular; as LEDs, speakers and digital/analogue outputs are common actuators. Also, interfaces like buttons can be programmed and used as a human interactive input [3].

### 3.2.2. THE PYTHON PLATFORM

Python is one of the more popular Open Source programming languages, owing largely to its own native expressiveness as well as to the variety of support modules that are available to extend its capabilities. One of these modules is DB-API, which, as the name implies, provides a database application programming interface. DBAPI is designed to be relatively independent of details specific to any given database engine; this helps you write database-access scripts that are portable between engines. DB-API's design is similar to that used by Perl's DBI module, the PHP PEAR DB class, and the Java JDBC interface. It uses a two level architecture in which the top level provides an abstract interface that is similar for all supported database engines, and a lower level consisting of drivers for specific engines that handle engine dependent details. This means, of course, that to use DB-API for writing Python scripts, you must have a driver for your particular database system [5].

## 3.3. COMPONENTS OF THE SYSTEM

A data logging wireless sensor network system comprises of the sensor nodes as the centre stones, the communication and networking protocols, the gateway, the database server and interface and the desired applications.

## 3.4. THE SENSOR NODES

Basic sensor nodes consist of four main hardware, which are;

1. The Processor/Controller module
2. The Transceiver module
3. The Sensor unit
4. The Power supply

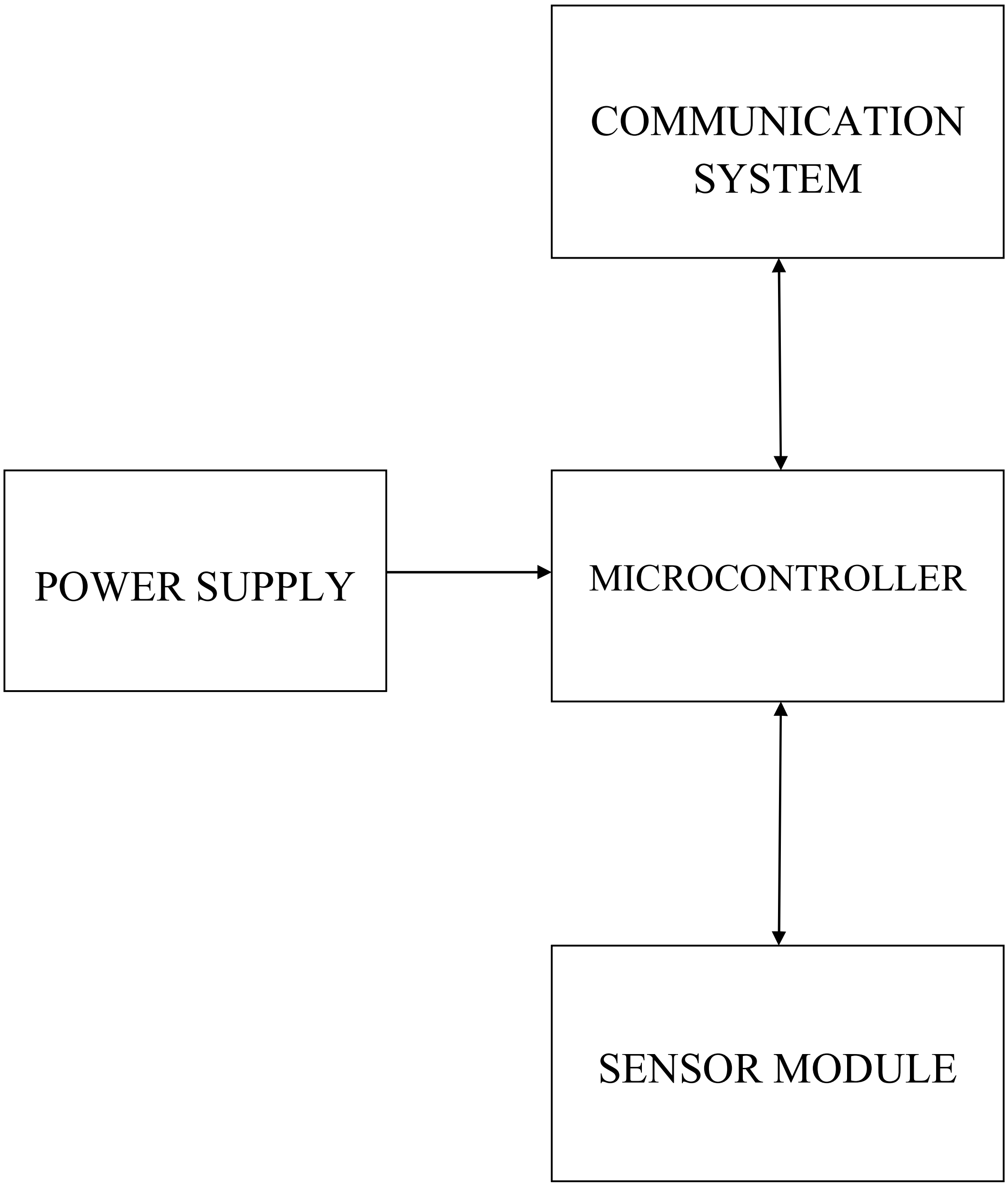



Fig.3.0: Block diagram showing the interconnectivity between each component of a sensor node

## MODE OF OPERATION OF A SENSOR NODE

Wireless sensor network consists of several sensor nodes, either interconnected to each other or connected directly to a database system. Each sensor node comprises of a power source, a controlling unit, a communication module and one or more sensor module. Once the node is powered, the sensor senses physical parameters which can be in form of analog or digital signal. The data sensed is transmitted to the controlling unit that processes the information and transmits the information through a communication module to a database wher it is stored and displayed.

### 3.3.1. **PROCESSOR/CONTROLLER MODULE**:

The Controller module executes all the calculation and processing tasks. Operating and application software codes run on it while coordinating other resources and components. In this project, Atmel's ATmega328p microcontroller chip was used as the would be using an Arduino board which is an open hardware such that the device's manufacturing schematics, programming language and software development environment are free and open source thereby doesn't need an operating system [19].

**ATMEGA328P MICROCONTROLLER**

Atmel's ATMega328P Microcontroller is a high-performance Atmel pico Power 8-bit AVR RISC-based microcontroller in 28 pin DIP package. It combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. Packaged, can be programmed in circuit, such as the Arduino. The device operates between 1.8V to 5V for its operating voltage. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

For better Synchronization when the ATmega328P Microcontroller is used as a Standalone device, 16MHz external crystal Oscillator with 2 100uF Capacitor should be used for its Clocking circuit.

The System Specifications for the ATmega328P Microcontroller chip is stated in the table below;

*Table 3.1: System Specifications for the ATmega328P Microcontroller chip [14]*

| | |
|---|---|
| Product Category | 8-bit Microcontrollers-MCU |
| Manufacturer | Atmel |
| Mounting style | Through Hole |
| Package/ Case | PDIP-28 |
| Core | AVR |
| Data Bus Width | 8 bits |
| Maximum Clock Frequency | 20MHz |
| Program Memory Size | 32 Kb |
| Data RAM Size | 2kB |
| ADC Resolution | 10 bits |
| Operating Supply Voltage | 1.8 V to 5.5 V |
| Maximum Operating Temperature | +85c |
| Processor Series | MegaAVR |
| Data RAM Type | SRAM |
| Data ROM Size | 1Kb |
| Data ROM Type | EEPROM |
| Height | 4.57mm |
| Interface Type | I2C, SPI, USART |
| Length | 34.8mm |
| Minimum Operating Temperature | - 40C |
| Number of inputs/outputs | 23I/O |
| Number of ADC Channels | 6 |
| Program Memory Type | Flash |
| Supply Voltage-Max | 5.5V |

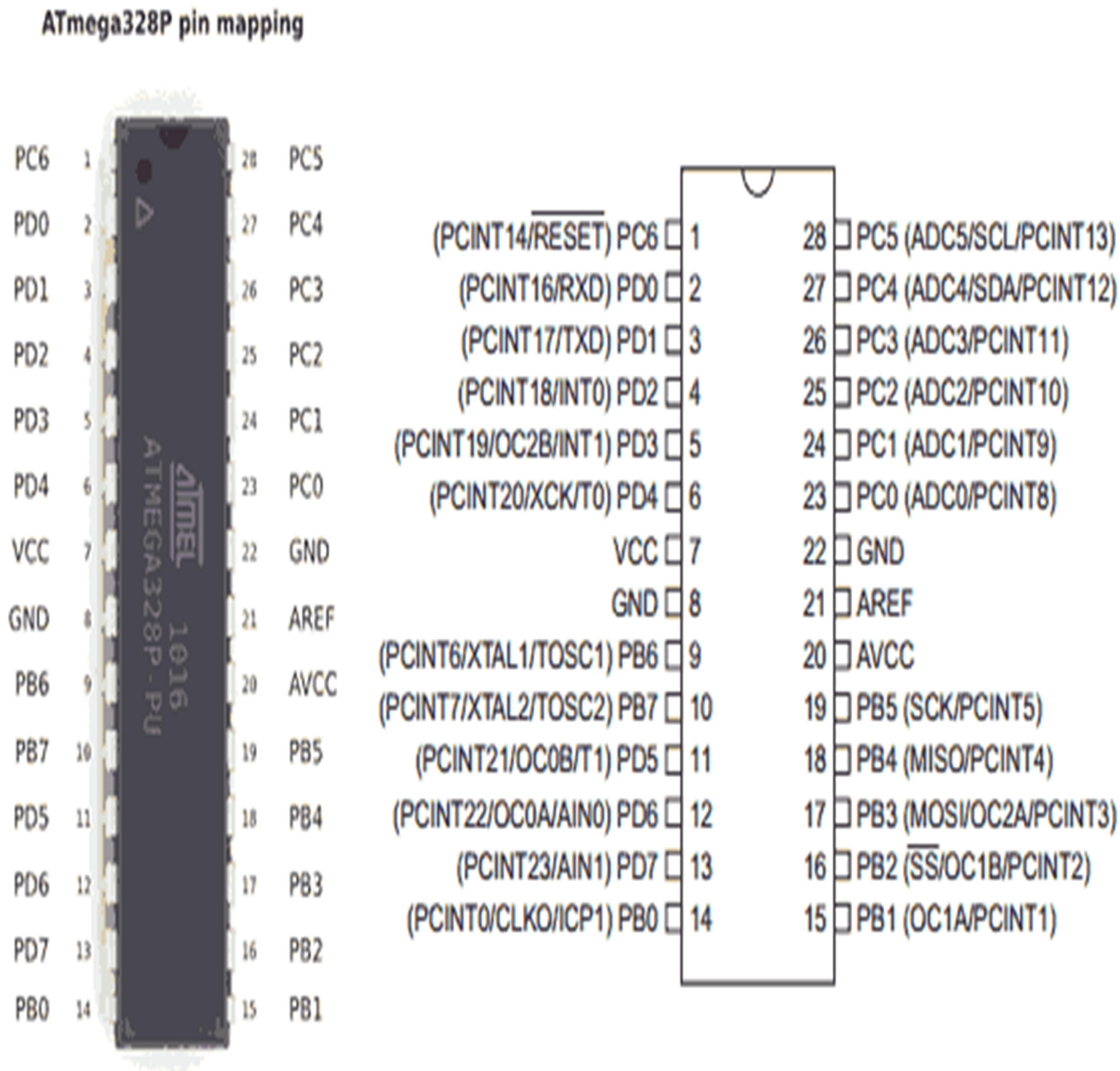



*Fig 3.2: ATmel ATMega328P Microcontroller [8]*

### 3.3.1.1 GENERAL OVERVIEW OF A COMPUTER ARCHITECTURE

Computer architecture is a specification detailing how a set of software and hardware technology standards interact to form a computer system or platform. In short, computer architecture refers to how a computer system is designed and what technologies it is compatible with.

As with other contexts and meanings of the word architecture, computer architecture is likened to the art of determining the needs of the user/system/technology, and creating a logical design and standards based on those requirements.

And a computer architecture can be very well likened to the ATMEGA 328P which is shown in details in the diagrams below.

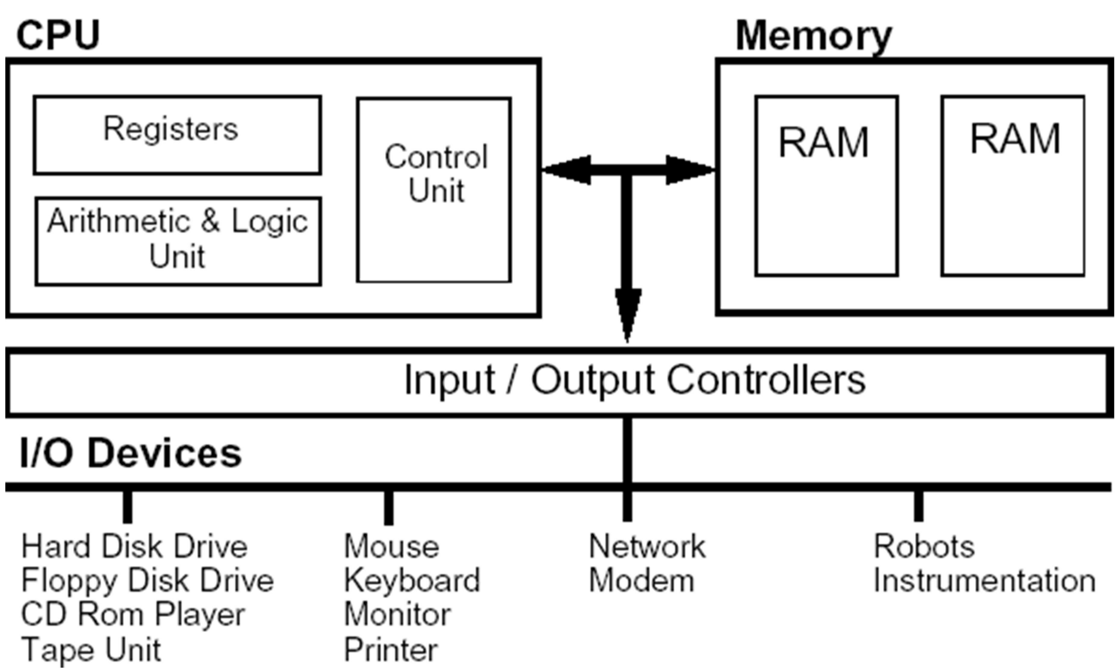



*Fig 3.3: General Computer Architecture Diagram [18]*

1. **Arithemetic And Logic Unit :** An arithmetic-logic unit (ALU) is the part of a computer processor (CPU) that carries out *arithmetic* and *logic* operations on the operands in computer instruction words
2. **Registers:** A processor register (CPU register) is one of a small set of data holding places that are part of the computer processor. A register may hold an instruction, a storage address, or any kind of data (such as a bit sequence or individual characters). Some instructions specify registers as part of the instruction.
3. **Control Unit:** The control unit (CU) is a component of a computer's central processing unit (CPU) that directs operation of the processor. It tells the computer's memory, arithmetic/logic unit and input and output devices how to respond to a program's instructions. ... Most computer resources are managed by the CU.
4. **Memory**: In computer architecture, the memory where memory hierarchy is used to discuss performance issues in computer architectural design, algorithm predictions, and lower level programming constructs involving locality of reference. The memory hierarchy in computer storage separates each of its levels based on response time.
5. **Input/ Output Controllers**: In computing, input/output or I/O (or, informally, io or IO) is where the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system is established

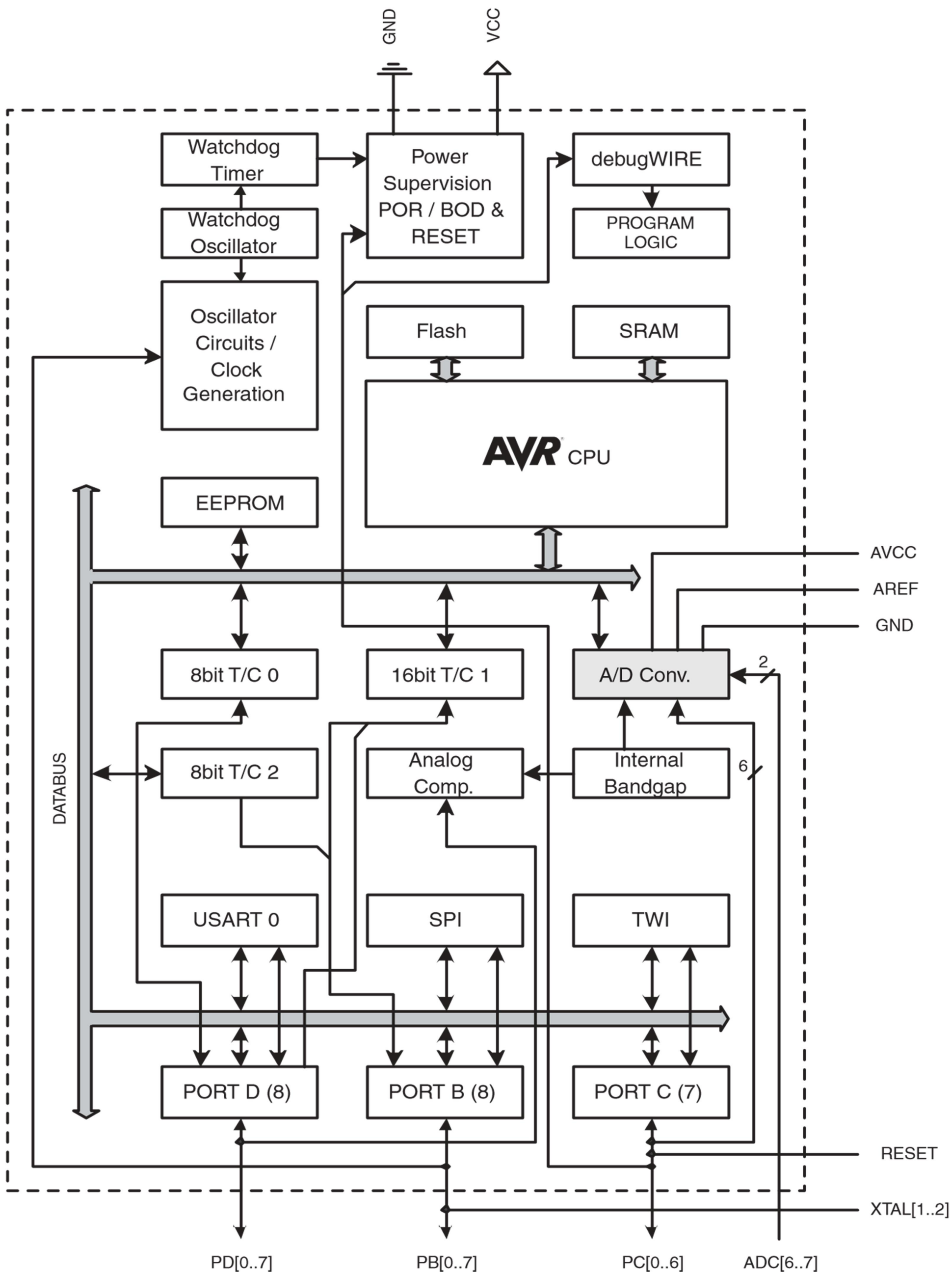



*Fig 3.4: The Atmega 328P Architecture [18]*

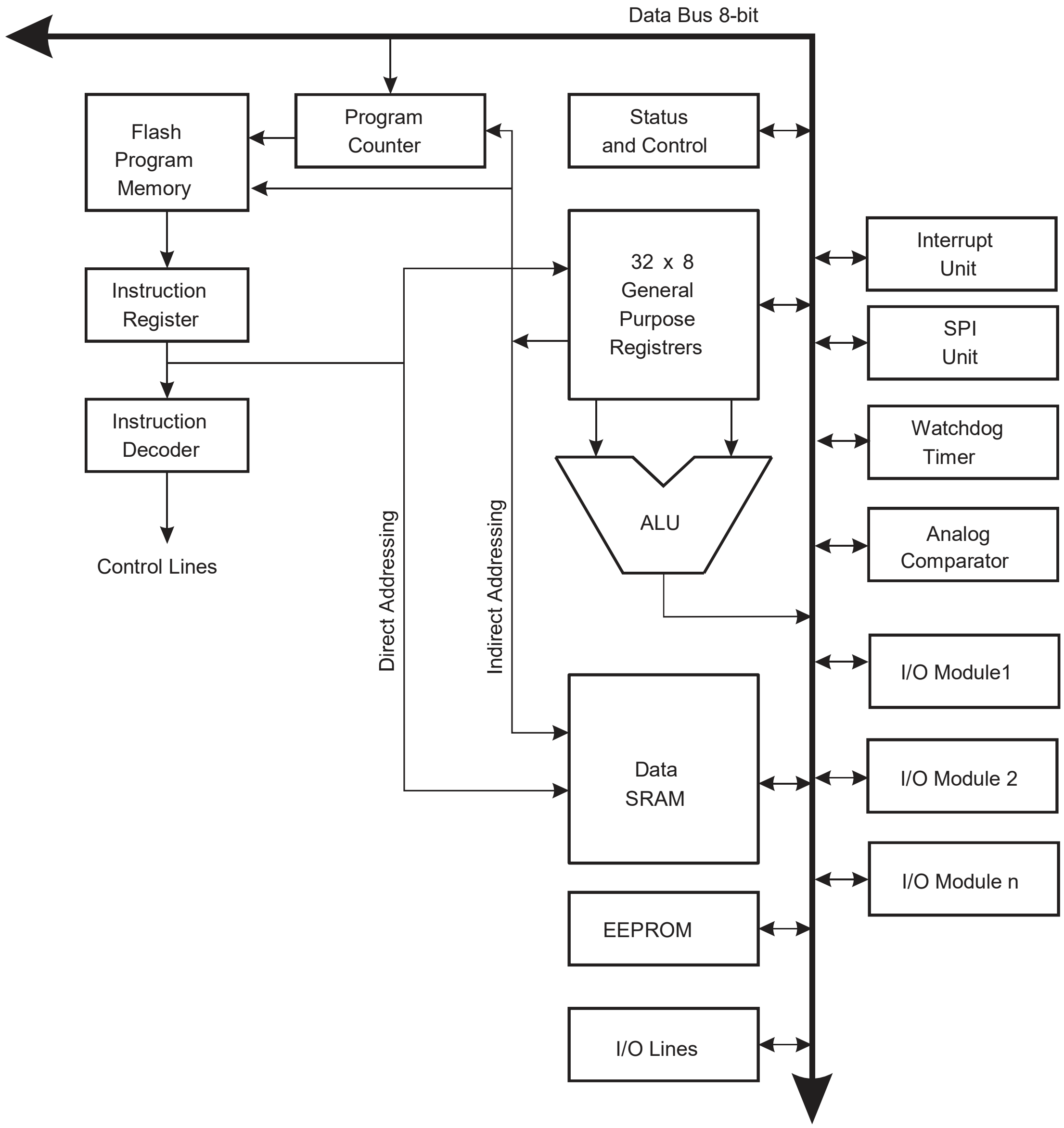



*Fig 3.5. Block Diagram of the AVR Architecture [18]*

1. **PIN count**: Atmega328 has got 28 pins. Two for Power (pin no.10: +5v, pin no. 11: ground), two for oscillator (pin 12, 13), one for reset (pin 9), three for providing necessary power and reference voltage to its internal ADC, and 32 (4×8) I/O pins.
2. **About I/O pins:** ATmega328 is capable of handling analogue inputs. Port A can be used as either DIGITAL I/O Lines or each individual pin can be used as a single input channel to the internal ADC of ATmega32,

plus a pair of pins AREF, AVCC & GND (refer to ATmega328 datasheet) together can make an ADC channel.

No pins can perform and serve for two purposes (for an example: Port A pins cannot work as a Digital I/O pin while the Internal ADC is activated) at the same time. It's the programmer's responsibility to resolve the conflict in the circuitry and the program. Programmers are advised to have a look to the priority tables and the internal configuration from the datasheet.

3. **Digital I/O pins:** ATmega328 has 28 pins (4portsx8pins) configurable as Digital I/O pins.
4. Timers: 3 Inbuilt timer/counters, two 8 bit (timer0, timer2) and one 16 bit (timer1).
5. **ADC:** It has one successive approximation type ADC in which total 8 single channels are selectable. They can also be used as 7 (for TQFP packages) or 2 (for DIP packages) differential channels. Reference is selectable, either an external reference can be used or the internal 2.56V reference can be brought into action. There external reference can be connected to the AREF pin.
6. **Communication Options:** ATmega328 has three data transfer modules embedded in it. They are

I. Two Wire Interface

II. USART

III. Serial Peripheral Interface [18]

7. **Analog comparator:** On-chip analog comparator is available. An interrupt is assigned for different comparison result obtained from the inputs.
8. **External Interrupt:** 3External interrupt is accepted. Interrupt sense is configurable.

9. **Memory:** It has 32Kbytes of In-System Self-Programmable Flash program memory, 1024 Bytes EEPROM, 2Kbytes Internal SRAM. Write/Erase Cycles: 10,000 Flash / 100,000 EEPROM.
10. **Clock:** It can run at a frequency from 1 to 16 MHz. Frequency can be obtained from external Quartz Crystal, Ceramic crystal or an R-C network. Internal calibrated RC oscillator can also be used.
11. **More Features**: Up to 16 MIPS throughput at 16MHz. Most of the instruction executes in a single cycle. Two cycle on-chip multiplication. 32 × 8 General Purpose Working Registers
12. **Debug:** JTAG boundary scan facilitates on chip debug.
13. **Programming:** Atmega328 can be programmed either by In-System Programming via Serial peripheral interface or by Parallel programming. Programming via JTAG interface is also possible. Programmer must ensure that SPI programming and JTAG are not be disabled using fuse bits; if the programming is supposed to be done using SPI or JTAG [18].

### 3.3.2. **Transceiver module:**

The Transceiver module is concerned with converting data in between bit stream, bytes or frames from microcontroller to and from radio waves of the wireless channel. WSNs utilize predominately radio frequency (RF) in the frequency spectrum 433 MHZ - 2.4GHz in half duplex mode and with no requirement on line of sight communication. In this project, an ESP-01 module, a Wi-Fi module which uses an IEEE 802.11 standard was used.

**The ESP-01 Wi-Fi Module:**

The ESP-01 Wi-Fi module utilizes 802.11b WLAN protocol. The 802.11b WLAN protocol can achieve data rates of 11Mbps, using the 2.4GHz frequency band. The ESP-01 Wi-Fi module is based on the ESP8266 chipset [8]. The ESP8266 chipset is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. This means that the ESP01 Wi-Fi module has an on board processor which can also perform the function of the ATmega328P microcontroller chip [8]. Each ESP8266 module comes pre-programmed with an AT command set firmware. This means that the ESP01 module can be plug onto the Arduino device and get about as much Wi-Fi ability as a Wi-Fi Shield. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime [7]. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces; it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts [14].

The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Therefore, a regulating circuit should be design that regulates the 5 volts from the microcontroller to 3.3 volts. This is done to prevent the 5 volts from the microcontroller from damaging the ESP8266 Wi-Fi module.

#### 3.3.2.1. Features of the ESP01 WLAN Module [14]

1. 802.11 b/g/n protocol (Wi-Fi protocol)
2. Wi-Fi Direct (Point-to-Point), soft-AP (Access Point)
3. Integrated Transport Control Protocol/ Internet Protocol (TCP/IP) stack
4. Integrated Transmitter/Receiver switch, Low Noise Amplifier, power amplifier and matching network

5. It requires Integrated PLLs, regulators, DCXO and power management units
6. It has a +19.5dBm output power in 802.11b mode
7. Power down leakage current of <10uA
8. It has a 1MB Flash Memory
9. Integrated low power 32-bit CPU could be used as application processor
10. It uses SPI (Serial Peripheral Interface), SPI Data Input Output 1.1 / 2.0, or Universal Asynchronous Receiver Transmitter (UART) protocol mode for its Embedded System
11. It uses either STBC, 1×1 MIMO (Multi Input Multi Output), or 2×1 MIMO mode of Antenna
12. It Wake up and transmit packets in < 2ms
13. Standby power consumption of < 1.0mW (DTIM3)

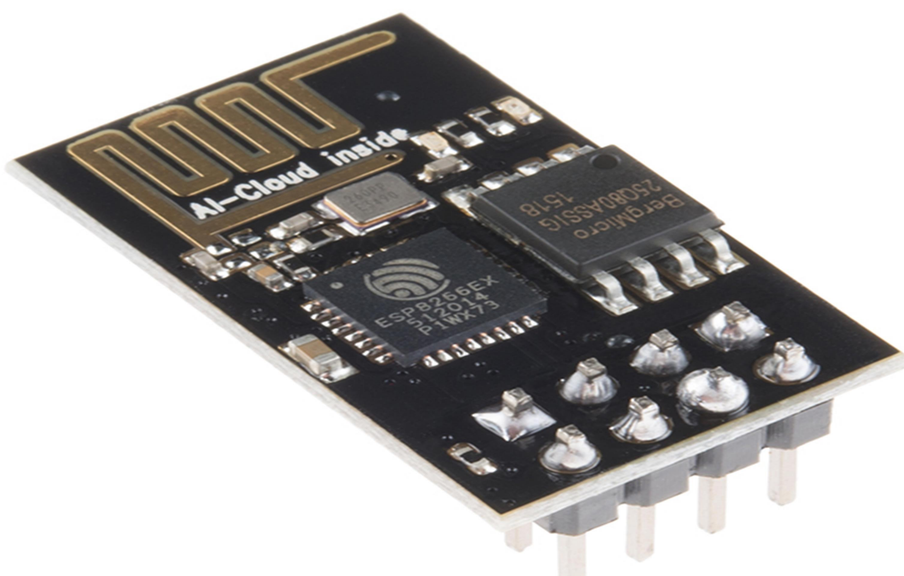



*Fig 3.6: ESP-01 WLAN Module [14]*

**ARCHITECTURE OF THE ESP01 MODULE**

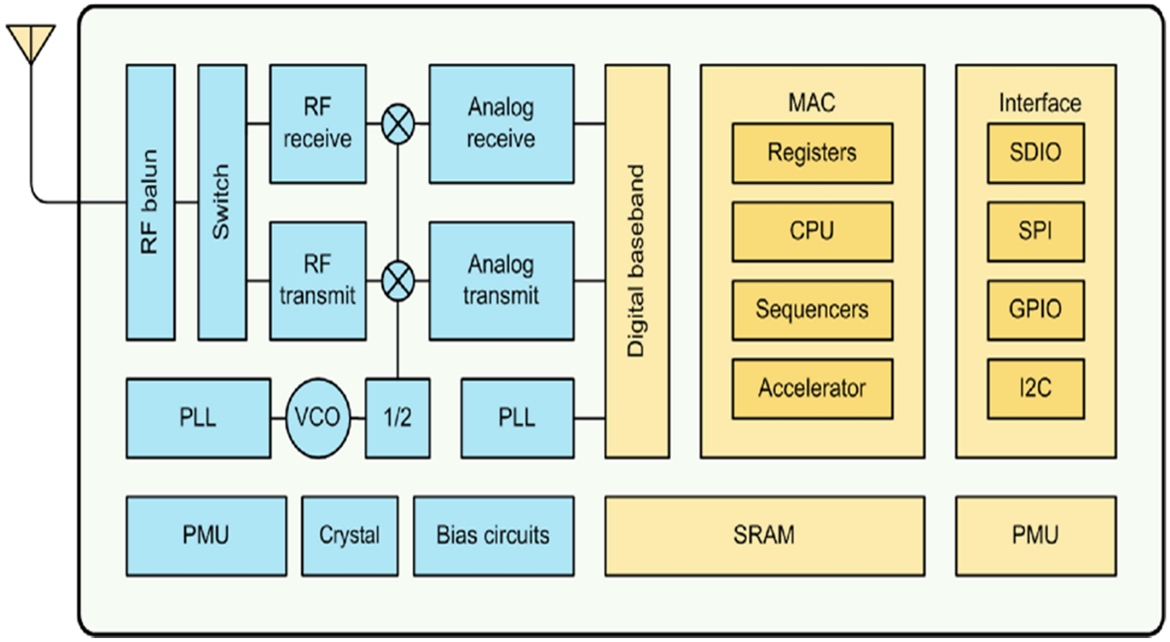



Fig.3.6a: Block Diagram of ESP8266EX Chipset

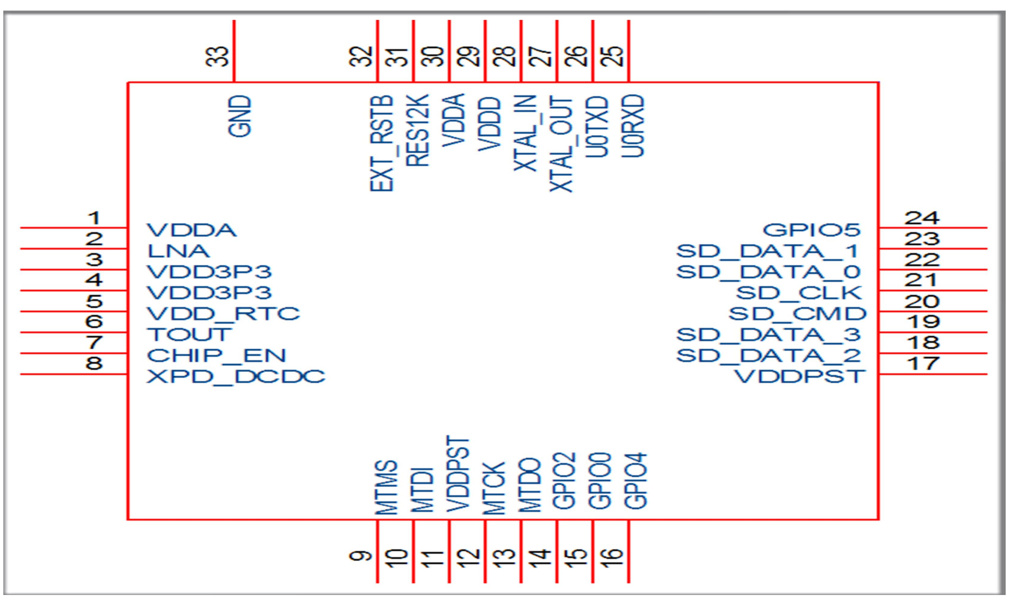



Fig.3.6b: Pin assignment for the ESP32866EX Chipset [12]

Table below presents an overview on the general pin attributes and functions of each pin.

Table : Pin Definitions [14]

| Pin | Name | Type | Function |
|---|---|---|---|
| 1 | VDDA | P | Analog Power 3.0 ~3.6V |
| 2 | LNA | I/O | RF Antenna Interface. Chip Output Impedance = 50Ω No matching required but we recommend that the π-type matching network is retained. |
| 3 | VDD3P3 | P | Amplifier Power 3.0~3.6V |
| 4 | VDD3P3 | P | Amplifier Power 3.0~3.6V |
| 5 | VDD_RTC | P | NC (1.1V) |
| 6 | TOUT | I | ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously. |
| 7 | CHIP_EN | I | Chip Enable.<br>High: On, chip works properly; Low: Off, small current |
| 8 | XPD_DCDC | I/O | Deep-Sleep Wakeup ; GPIO16 |
| 9 | MTMS | I/O | GPIO14; HSPI_CLK |
| 10 | MTDI | I/O | GPIO12; HSPI_MISO |
| 11 | VDDPST | P | Digital/IO Power Supply (1.8V~3.3V) |
| 12 | MTCK | I/O | GPIO13; HSPI_MOSI; UART0_CTS |
| 13 | MTDO | I/O | GPIO15; HSPI_CS; UART0_RTS |
| 14 | GPIO2 | I/O | UART Tx during flash programming; GPIO2 |
| 15 | GPIO0 | I/O | GPIO0; SPI_CS2 |
| 16 | GPIO4 | I/O | GPIO4 |
| 17 | VDDPST | P | Digital/IO Power Supply (1.8V~3.3V) |
| 18 | SDIO_DATA_2 | I/O | Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9 |

| | | | |
|---|---|---|---|
| 19 | SDIO_DATA_3 | I/O | Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10 |
| 20 | SDIO_CMD | I/O | Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11 |
| 21 | SDIO_CLK | I/O | Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6 |
| 22 | SDIO_DATA_0 | I/O | Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7 |
| 23 | SDIO_DATA_1 | I/O | Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8 |
| 24 | GPIO5 | I/O | GPIO5 |
| 25 | U0RXD | I/O | UART Reception during flash programming; GPIO3 |
| 26 | U0TXD | I/O | UART Transmission during flash progamming; GPIO1; SPI_CS1 |
| 27 | XTAL_OUT | I/O | Connect to crystal oscillator output, it can be used to provide BT clock input |
| 28 | XTAL_IN | I/O | Connect to crystal oscillator input |
| 29 | VDDD | P | Analog Power 3.0V~3.6V |
| 30 | VDDA | P | Analog Power 3.0V~3.6V |
| 31 | RES12K | I | Serial connection with a 12 kΩ resistor and connect to the ground |
| 32 | EXT_RSTB | I | External reset signal (Low voltage level: Active) |

## EMBEDDED MICROCONTROLLER ON THE ESP01 WiFi MODULE

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the WiFi stack, the rest can all be used for user application programming and development. The following interfaces can be used to connect to the MCU embedded in ESP8266EX:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller,

  and can also be used to visit external flash;
- Data RAM interface (dBus), which can be connected with memory controller;
- AHB interface, can be used to visit the register.

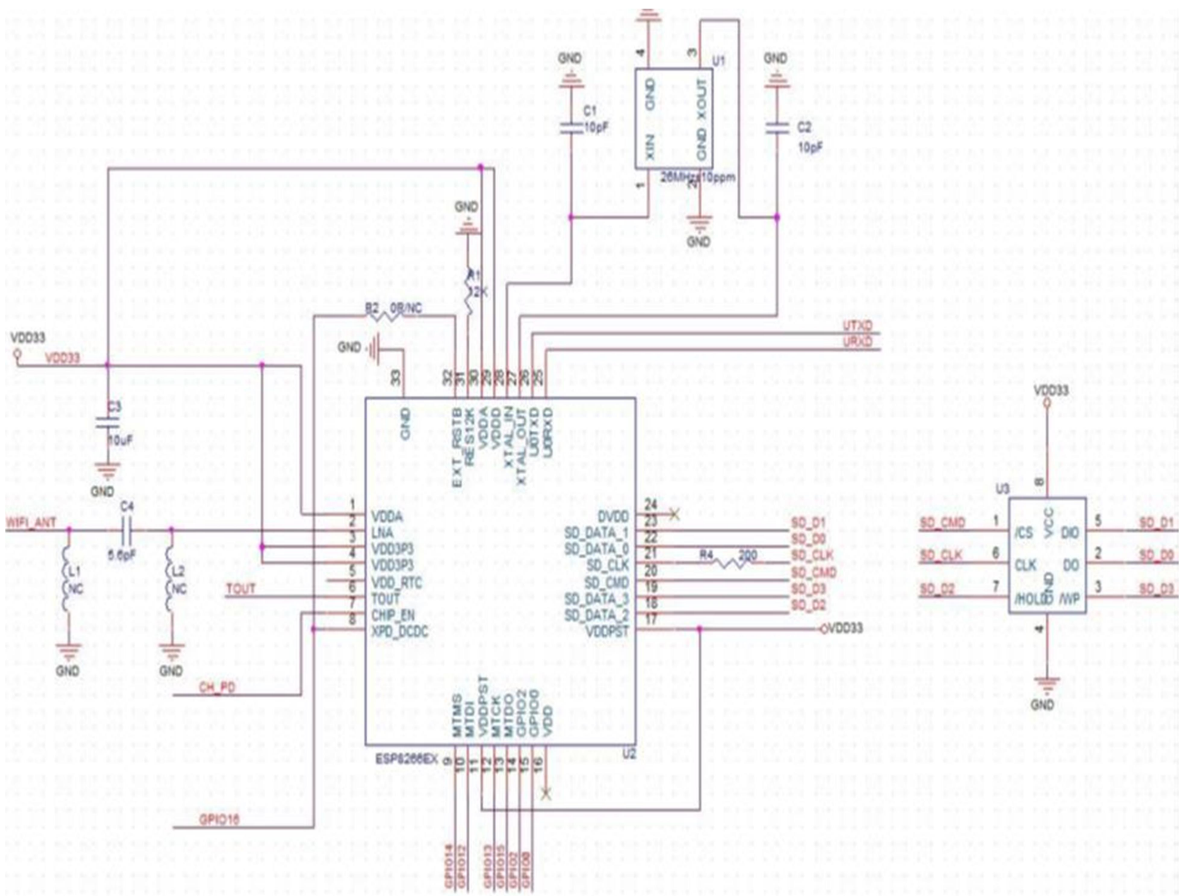



*Fig.3.6c: Schematic diagram showing the interconnectivity between the ESP8266EX and L106 Microcontroller [14]*

The Schematic diagram above shows the interconnectivity between the

ESP8266EX and the L106 Microcontroller in the ESP01 Module. An external clock circuit that is generated from an internal crystal oscillator and an external crystal in which the crystal frequency can range from 26MHz to 52MHz.

ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor. [14]

When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface or the CPU AHB bridge interface

ESP8266 on-board processing and storage capabilities allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. With its high degree of on-chip integration, which includes the antenna switch balun, power management converters, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. [14]

### 3.3.3. **Power supply module:**

The mechanisms for power supply in sensor nodes are either non rechargeable primary batteries or energy scavenging from the environment. While incorporating, primary battery is easier and the prevalent method, in the many deployment scenarios replacement is impossible thus the later method of energy scavenging necessary. Photo voltaic recharging based on solar cells, thermoelectric generators based on temperature differences and electromagnetic or electrostatic generators based on mechanical vibrations are among the possibilities to recharge secondary batteries. The main limitation of these approaches is that the technologies are not mature enough to enable recharging at very low current levels and efficiently manage intermittent charging cycles [38]. In this design, 5 AA batteries are used to power each node. Each battery produces an E.M.F of 1.5 volts.

### 3.3.4. **Sensors:**

Sensors, as the first interface to the physical world [13], a device which detects or measures a physical property and records, indicates, or otherwise responds to it [2]. It is a type of transducer; sensors may provide various types of output, but typically use electrical signals.

In this project, four different Sensors were used in the design and there include;

1. DHT11 Temperature and Humidity Sensor
2. GY-91 built in 3-in-1 (Barometric pressure, Gyroscopic, and Accelerometer) Sensor

3. MQ-2 Gas Sensor

1. **DHT11 SENSOR**

The DHT11 Sensor is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spends out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds. For efficient communication with other components connected to it, a 10K resistor, which will serve as a pull-up from the data pin to VCC pin of the sensor should be connected.

DHT11 humidity and temperature sensors can be use in remote weather stations, home environment control systems, and agricultural/garden monitoring systems.

**Technical Specifications of the DHT11 Sensor**

1. Operating Voltage: 3V to 5.5V
2. Low cost
3. 2.5mA max current use during conversion (while requesting data)
4. Good for 20-90% humidity readings with 5% accuracy
5. Good for 0-50°C temperature readings ±2°C accuracy
6. No more than 1 Hz sampling rate (once every second)
7. Body size 15.5mm x 12mm x 5.5mm
8. 4 pins with 0.1" spacing

The DHT11 humidity and temperature sensor measures relative humidity (RH) and temperature. Relative humidity is the ratio of water vapour in air vs. the saturation point of water vapour in air. The saturation point of water vapour in air changes with temperature. Cold air can hold less water vapour before it is saturated, and hot air can hold more water vapour before it is saturated. The formula for relative humidity is as follows:

*Relative Humidity = (density of water vapour / density of water vapour at saturation) x 100%*

Basically, relative humidity is the amount of water in the air compared to the amount of water that air can hold before condensation occurs. It’s expressed as a percentage. For example, at 100% RH condensation (or rain) occurs, and at 0% RH, the air is completely dry.

**How the DHT11 Sensor Measures Humidity and Temperature**

The DHT11 sensor calculates relative humidity by measuring the electrical resistance between two electrodes. The humidity sensing component of the DHT11 is a moisture holding substrate (usually a salt or conductive plastic polymer) with the electrodes applied to the surface. When water vapour 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. Inside the DHT11 sensor, there are electrodes which are applied to a substrate in the front of the chip.

The DHT11 Sensor converts the resistance measurement to relative humidity on an IC mounted to the back of the unit and transmits the humidity and temperature readings directly to the Microcontroller chip on the Arduino. This IC also stores the calibration coefficients and controls the data signal transmission between the DHT11 and the Arduino.

The temperature readings from the DHT11 come from a surface mounted NTC temperature sensor (thermistor) built into the unit. The DHT11 uses one signal wire to transmit sensor readings to the Arduino digitally. The power comes from separate 5V and ground wires. A 5K – 10K Ohm pull-up resistor is connected from the signal line to 5V to make sure the signal level stays high by default.

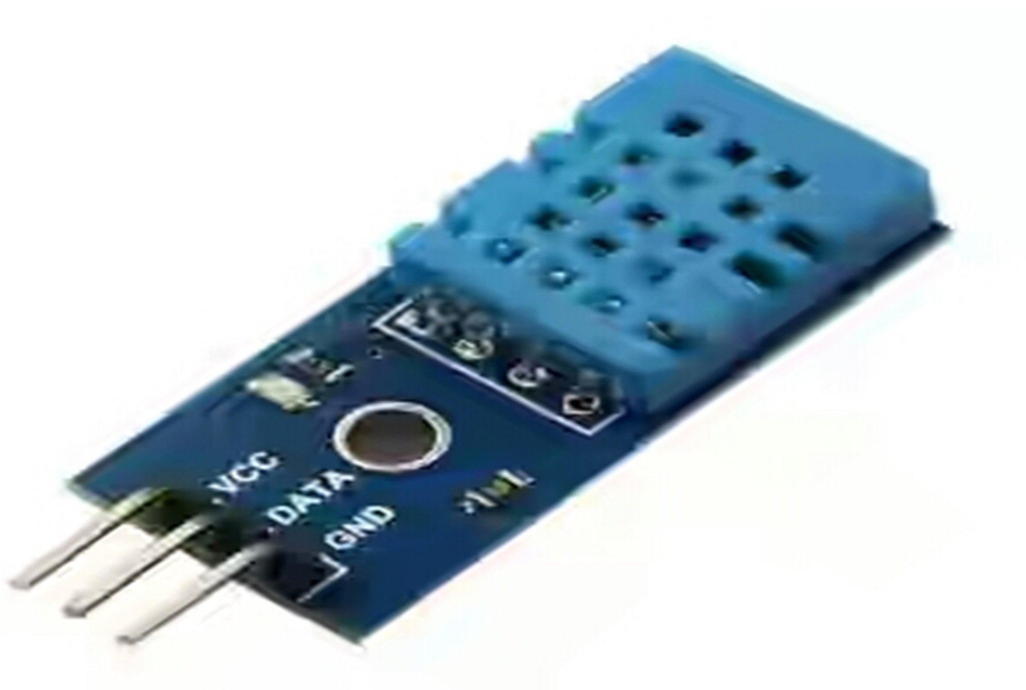



*Fig 3.7: DHT11 Temperature and Humidity Sensor [14]*

**MQ-2 GAS SENSOR**

This sensor module utilizes an MQ-2 as the sensitive component and has a protection resistor and an adjustable resistor on board. The MQ-2 gas sensor is sensitive to LPG, i-butane, propane, methane, alcohol, Hydrogen and smoke. It could be used in gas leakage detecting equipment in homes and industries. The resistance of the sensitive component changes as the concentration of the target gas changes.

**Features of the MQ-2 Gas Sensor**

1. Continuous Analog output
2. 3-pin interlock connector
3. Low cost, compact size and operates at Standard Working Condition


*Fig 3.8: MQ-2 Gas Sensor [14]*

***Table 3.2: Characteristics of the MQ-2 Gas Sensor [14]***

| Symbol | Parameter Name | Technical Condition | Remark |
|---|---|---|---|
| VC | Circuit Voltage | 5V±0.1 | AC or DC |
| VH | Heating Voltage | 5V±0.1 | AC or DC |

| | | | |
|---|---|---|---|
| RL | Load Resistance | Adjustable | |
| RH | Heater Resistance | 33Kohm±5% | Room temperature |
| PH | Heating Consumption | Less than 800Mw | |

***Tab 3.3: Environmental Condition of the MQ-2 Gas Sensor [14]***

| Symbol | Parameter Name | Technical Condition | Remark |
|---|---|---|---|
| TO | Operating Temperature | -20°C-50°C | |
| TS | Storage Temperature | -20°C-70°C | |
| RH | Relative Humidity | <95% | |
| O2 | Oxygen Concentration | 21% (standard condition) | Oxygen concentration can affect sensitivity |

***Tab 3.4: Sensitivity Characteristics of the MQ-2 Gas Sensor [14]***

| Symbol | Parameter Name | Technical Condition | Remark |
|---|---|---|---|
| RS | Sensor Resistance | 3Kohm-30Kohm (1000ppm iso-butane) | Detecting |

**Concentration scope:**

1. 200ppm-5000pp LPG and propane
2. 300ppm-5000ppm Butane
3. 5000ppm-20000ppm Methane
4. 300ppm-5000ppm Hydrogen
5. 100ppm-2000ppm Alcohol (3000ppm/1000ppm iso-butane)
6. Concentration slope rate ≤0.6

7. Standard detecting Condition
8. Temperature: 20°C±2°C
9. Operating Voltage: 5V±0.1
10. Humidity: 65%±5% VH: 5V±0.1
11. Preheating Time Over 24 hours

**GY-91 sensor module**

The GY-91 sensor module is a multi functioning sensor. The GY-91 Sensor is fabricated by the combination of Single chip MPU-9250 with built-in 3 axis Gyroscope, 3 axis Accelerometer, digital Compass and BMP280 improved barometric pressure sensor.

The MPU-9250 is the company's second generation 9-axis Motion Tracking device for smart phones, tablets, wearable sensors, and other consumer markets. The MPU-9250, delivered in a 3x3x1mm QFN package, is the world's smallest 9-axis Motion Tracking device and incorporates the latest InvenSense design innovations, enabling dramatically reduced chip size and power consumption, while at the same time improving performance and cost.

Improvements in this sensor compared to previous versions include supporting the accelerometer low power mode with as little as 6.4µA of and it provides improved compass data resolution of 16-bits (0.15 µT per LSB). The full scale measurement range of ±4800µT helps alleviate compass placement challenges on complex PCBs.

In this design, the GY-91 Sensor was used as a BMP280 sensor, which is a barometric pressure sensor.

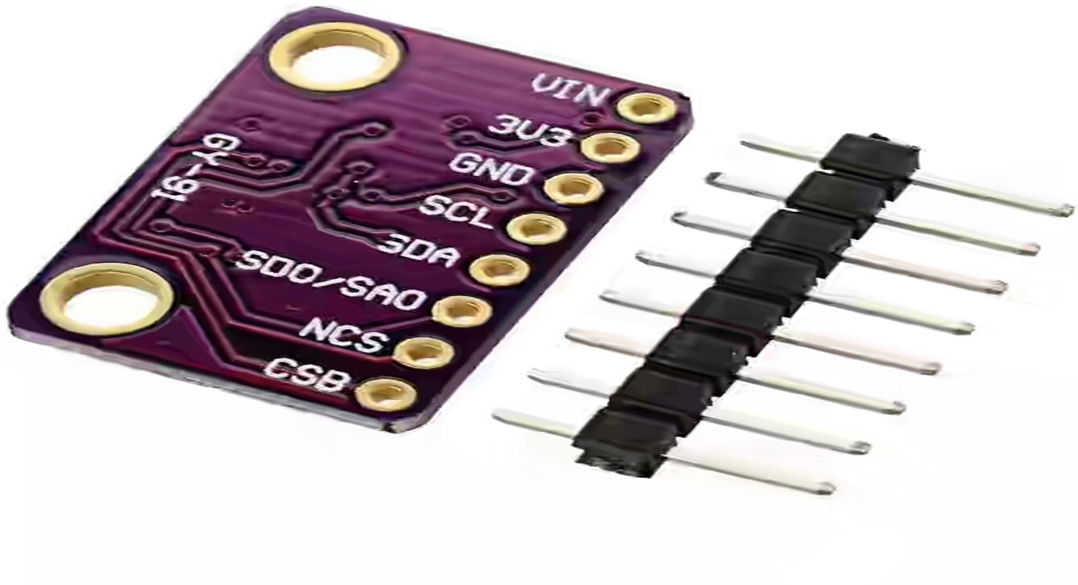



*Fig 3.9 GY-91 Multi-purpose Sensor [8]*

**Technical Specifications of the GY-91 Sensor [8]**

1. Model / Part Number: GY-91
2. Sensor Chips: MPU-9250 + BMP280
3. Interface: standard IIC / SPI communications protocol
4. Operating Voltage range: 3.0V – 5.0V (on-board low dropout regulator)
5. Module size 14.3mm * 20.5mm (More compact than GY-80)
6. Resolution: 16bit AD converter, 16-bit data output
7. Gyroscopes range: ± 250, 500, 1000, 2000 ° / s
8. Acceleration range: ± 2 ± 4 ± 8 ± 16g
9. Field range: ± 4800uT
10. Pressure range: 300-1100hPa

**Pin Configuration of the GY-91 Sensor [8]**

VIN: Voltage Supply Pin

3V3: 3.3v Regulator output

GND: 0V Power Supply

SCL: I2C Clock / SPI Clock

SDA: I2C Data or SPI Data Input

SDO/SAO: SPI Data output / I2C Slave Address configuration pin

NCS: Chip Select for SPI mode only for MPU-9250

CSB: Chip Select for BMP280

**Areas of Application for the GY-91 Sensor [8]**

The GY-91 Sensor can be utilized in the following fields

1. Quad-copter or any embedded application requiring multiple sensors
2. Robotic Arm stabilization
3. 2 wheel drive and 4 wheel drive robots where platform stability is a concern
4. Environmental monitoring using Sensor networks
5. Other Electronically Stable platforms

## BLOCK DIAGRAM OF A PRESSURE SENSOR

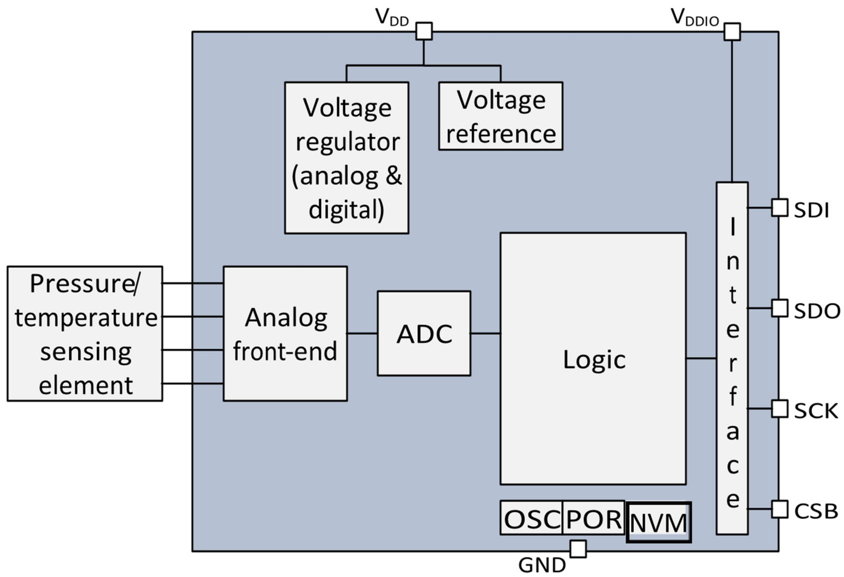



*Fig.: Block diagram of BMP280 Pressure Sensor [17]*

**Functional description**

The BMP280 consists of a Piezo-resistive pressure sensing element and a mixed-signal ASIC. The ASIC performs A/D conversions and provides the conversion results and sensor specific compensation data through a digital interface.

BMP280 provides highest flexibility to the designer and can be adapted to the requirements regarding accuracy, measurement time and power consumption by selecting from a high number of possible combinations of the sensor settings.

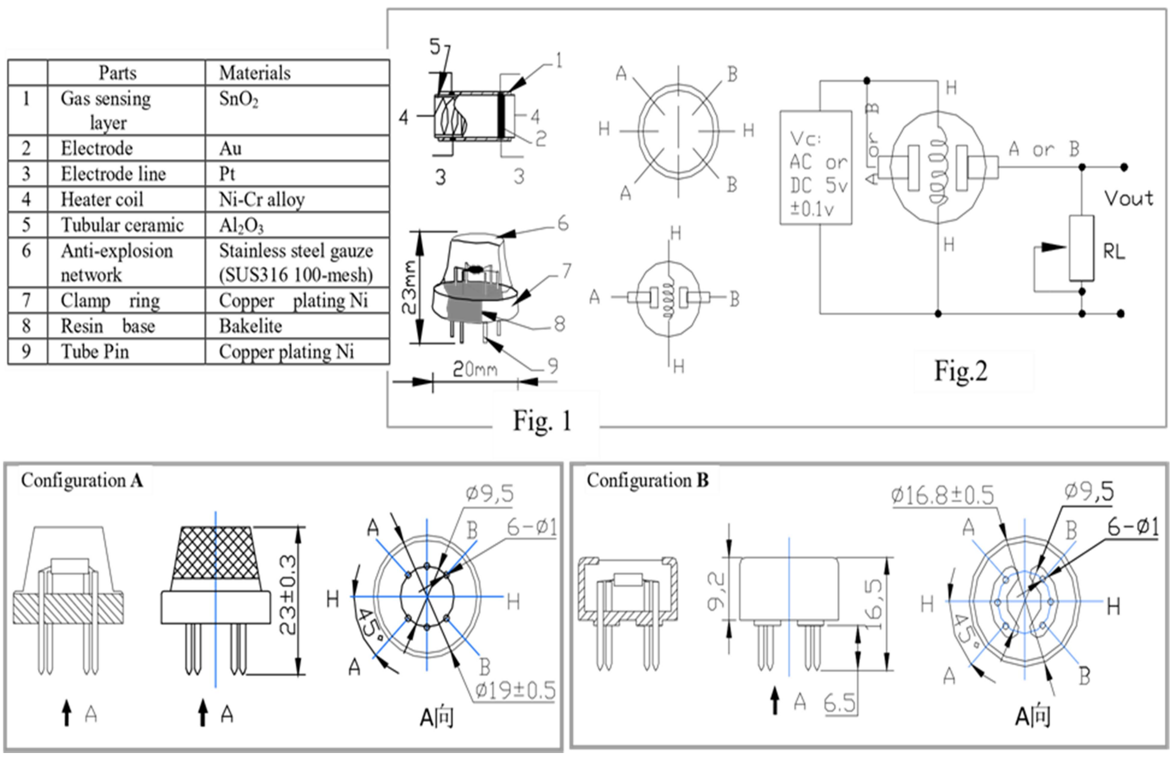
BMP280 can be operated in three power modes

- sleep mode
- normal mode

- forced mode

In sleep mode, no measurements are performed. Normal mode comprises an automated perpetual cycling between an active measurement period and an inactive standby period. In forced mode, a single measurement is performed. When the measurement is finished, the sensor returns to sleep mode.

## STRUCTURE AND CONFIGURATION OF AN MQ-2 GAS SENSOR



| | Parts | Materials |
|---|---|---|
| 1 | Gas sensing layer | $SnO_2$ |
| 2 | Electrode | Au |
| 3 | Electrode line | Pt |
| 4 | Heater coil | Ni-Cr alloy |
| 5 | Tubular ceramic | $Al_2O_3$ |
| 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 |



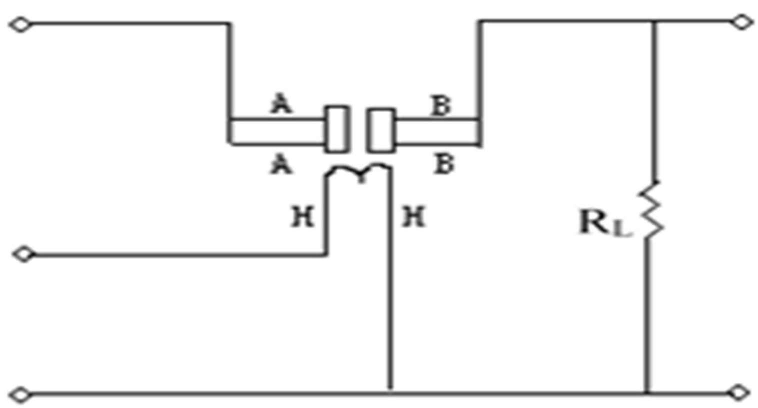
*Fig.: Structure and Configuration of the MQ-2 gas sensor [18]*

The figure above shows the Structure and configuration of MQ-2 gas sensor (Configuration A or B). The Sensor compose of micro AL2O3 ceramic tube, Tin Dioxide (SnO2) sensitive layer, measuring electrode and heater are fixed into a Resin base which is surrounded by the clamp ring. The 4 tube pins are connected to the voltage source, the ground, the digital and analog lines of the when interfacing with other modules. [18]

Sensitive material of MQ-2 gas sensor is $SnO_2$, 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. [18]

MQ-2 gas sensor has high sensitivity to Liquefied Petroleum Gas, also to Carbon monoxide and Smoke. The sensor could be used to detect different combustible gas, especially Methane, it is with low cost and suitable for different application

Vc




VRL

VH

*Fig.: Basic test circuit of the MQ-2 Sensor [18]*

The above is basic test circuit of the MQ-2 sensor. The sensor need to be put 2 voltage, heater voltage VH and test voltage VC. VH used to supply certified working temperature to the sensor, while VC used to detect voltage (VRL) on load resistance, RL whom is in series with sensor. The sensor has light polarity, Vc need DC power. VC and VH could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable RL value is needed.

DHT11 SENSOR PIN CONFIGURATION

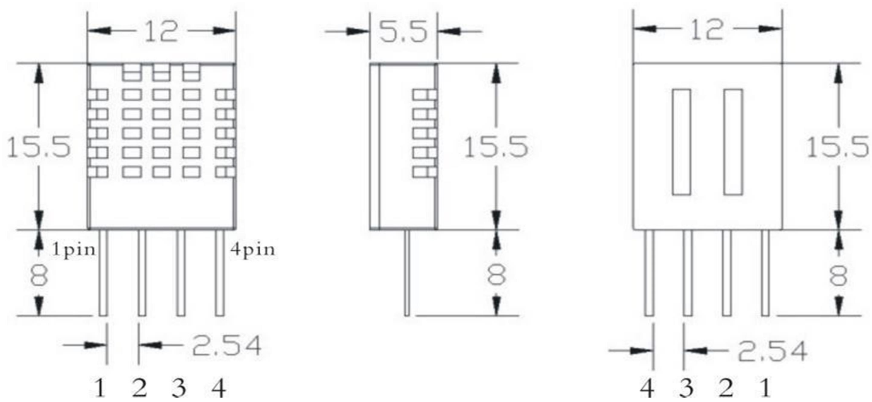



*Fig.3.7a: Pin Configuration of the DHT11 Sensor [19]*

**DHT11 Sensor Pin Description [19]**

1. the VDD power supply 3.5～5.5V DC
2. DATA serial data, a single bus
3. NC, empty pin
4. GND ground, the negative power

## A TYPICAL CIRCUIT THAT DESCRIBES THE OPERATION OF A DHT11 SENSOR

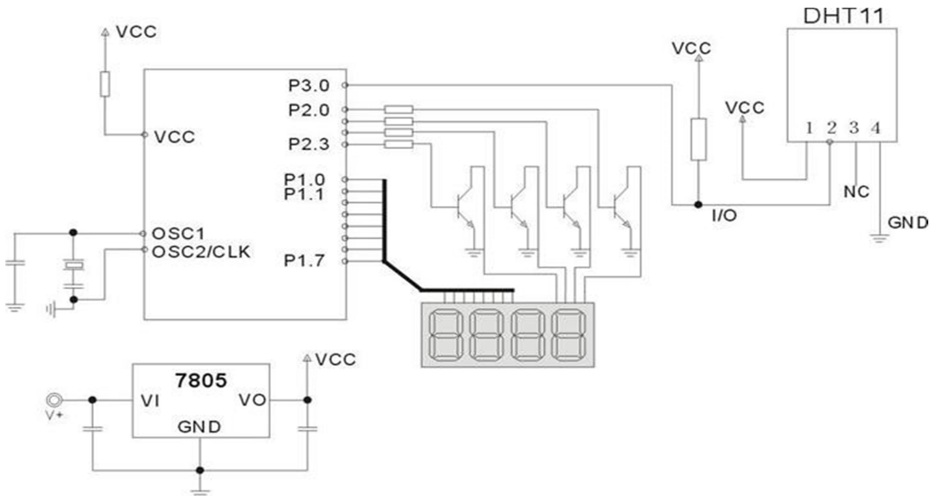



*Fig.3.7b: Schematic diagram showing the operation of a DHT11 Sensor node [19]*

Microprocessor and DHT11 of connection typical application circuit as shown above, DATA pull the microprocessor I / O ports are connected.

- Typical application circuit recommended in the short cable length of 20 meters on the 5.1K pull-up resistor, the resistance of greater than 20 meters under the pull-up resistor on the lower of the actual situation.
- When using a 3.5V voltage supply cable length shall not be greater than 20cm. Otherwise, the line voltage drop will cause the sensor power supply shortage, caused by measurement error.
- Each read out the temperature and humidity values are the results of the last measurement for real-time data, sequential read twice, but is not recommended to repeatedly read the sensors, each read sensor interval is greater than 5 seconds can be obtained accurate data.

## SERIAL COMMUNICATION INSTRUCTIONS (SINGLE-WIRE BIDIRECTIONAL)

**Single Bus Description:**

DHT11 uses a simplified single-bus communication. Single bus that only one data line, the system of data exchange, control by a single bus to complete. Device (master or slave) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external one about 5.1kΩ pullup resistor, so that when the bus is idle, its status is high. Because they are the master-slave structure, and only when the host calls the slave, the slave can answer, the host access devices must strictly follow the single-bus sequence, if the chaotic sequence, the device will not respond to the host.[19]

### 3.4 HAREWARE DESIGN OF THE WIRELESS SENSOR NETWORK

The design of the Wireless Sensor Network involves 7 processes namely;

1. Interfacing the sensors on the arduino board
2. Interfacing the wifi module to microcontroller on the arduino board
3. Programming of the microcontroller
4. Building a standalone arduino using the atmega328p microcontroller on a vero board
5. Designing a 3.3v regulating circuit for the communication module using lm117t voltage regulator on a vero board
6. Placement and connection of sensors and wifi modules on the vero board
7. Placement and connection of power source on each sensor node of the network

#### 3.4.1. INTERFACING THE SENSOR TO THE ARDUINO

The Sensors used in this design are sensitive components which should be handled with care when connecting to a voltage source. Sensors used in this design require a voltage between 3.3 volts to 5 volts. If the voltage supplied to the Sensors is above 5 volts, it will damage the circuitry of the sensors.

For a Sensor to function in this design, it must interface with the Arduino that will be use to program the ATmega328P Microcontroller chip. This involves uploading an Arduino Library for each of the Sensor used in the Arduino Integrated Development Environment (IDE).

Arduino Libraries are a collection of codes that are used for interfacing an Arduino to physical devices. This makes it easier for Designers to connect to Sensors, Display modules, Actuators, etc. For example, the DHT11 Library allows the Arduino to interface with the DHT11 Temperature/Humidity Sensor.

If the Library for a sensor, is not present in the Arduino IDE, it should be downloaded as a Zip/compressed file from the internet and uploaded into the Arduino Libraries.

For this project, the following procedures were performed during the interfacing of the sensors;

1. The Arduino application used for interfacing the sensors was installed on a Computer.
2. The specific Libraries for each sensor used in the design, which were not available in the Arduino library were downloaded from the internet
3. The downloaded libraries were saved in the Computer where the Arduino application has been installed
4. The library information contained in the downloaded Zip file was extracted and stored in a folder, having the same name as the Zip file
5. The Arduino application installed on the computer was opened
6. In the Arduino IDE, click on “Sketch” then click on “include library” after which click on “add zip library”

7. A search engine came up from which the folder containing the library was located and “open” was clicked on
8. Restart the Arduino application and ensure that the new libraries appears in the “Sketch>import library menu item of the software

It should be noted that the library file contains .cpp files, .h files, example files and some other files that ensures the proper functioning of the particular interfacing device.

### 3.4.2. INTERFACING THE WI-FI MODULE TO MICRO-CONTROLLER ON THE ARDUINO BOARD

The Esp-01 wi-fi module used in this design are sensitive components which should be handled with care when connecting to a voltage source. The esp-01 wi-fi module requires 3.3v to power it up. Anything above that value could damage the circuitry. So, it is required that a LM117T voltage regulator is connected between the 5v or above voltage source and the esp-01 when wiring it to the Arduino board. A breadboard might also be needed.

As previously stated, arduino Libraries are a collection of codes that are used for interfacing an Arduino to physical devices. So, definitely a library is needed to interface the esp-01 wi-fi with the Arduino board. The following procedures were performed during the interfacing of the esp-01 wi-fi module;

1. The Arduino IDE application is installed on a Computer.
2. The Library for the wi-fi module which is not available in the Arduino library is downloaded from the internet
3. The downloaded library is saved in the Computer where the Arduino application is installed
4. The library information contained in the downloaded Zip file is extracted and stored in a folder, having the same name as the Zip file
5. The Arduino application installed on the computer is opened

6. In the Arduino IDE, click on “Sketch” then click on “include library” after which click on “add zip library”
7. A search engine came up from which the folder containing the library was located and “open” was clicked on
8. Restart the Arduino application and ensure that the new libraries appear in the “Sketch>import library menu item of the software
9. An sample program called “webclient” was used to interface the wi-fi module with the Arduino.

The sample program is used to program the esp-01 such that it connects to the network with the SSID and password in the program.

### 3.4.3. PROGRAMMING OF THE MICROCONTROLLER

The Microcontrollers used in this design are sensitive components which should be handled with care when connecting to a voltage source. The microcontroller used in this design is the Atmel’s ATmega328P chip. It requires a voltage between 3.3 volts to 5 volts. If the voltage supplied to the Microcontroller exceeds 5 volts, it will damage the circuitry of the Microcontroller.

The following operations were performed while programming the microcontroller in this design;

1. Open the Arduino application installed on the Computer
2. Place the ATmega328P Microcontroller chip on the Arduino board. The board used was an Arduino Uno board
3. Plug the Arduino board to the Computer where the Arduino application was installed on using a USB cable. The Computer supplies the voltage to the Arduino

4. Burn the bootloader into the microcontroller chip, which was done by uploading the Arduino in-system program (ISP) Sketch onto the Arduino. This involves to;
    i. Select the board and the serial port from the "tools menu" that corresponds to the Arduino board
    ii. Select "Arduino/ATmega328P" from the Tools>Board menu
    iii. Click on " Run Tools>Burn Bootloader>w/Arduino as ISP
5. After running the bootloader, the codes for each sensor was written on the Arduino IDE after which the codes where debugged, uploaded into the Microcontrollers for each sensor.

### 3.4.4. BUILDING A STANDALONE ARDUINO USING THE ATmega328P MICROCONTROLLER

This design involves building an Arduino system, which is alternative compared to the Arduino board itself. The whole design was performed for four (4) Sensor nodes and was built around the ATmega328P Microcontroller chip.

The following components were used during the design of this system;

1. 4 x Vero boards
2. Jumper wires
3. 4 x 7805 Voltage regulator (5 volts)
4. 4 x 2 LEDs
5. 4 x 2 220 Ohm resistors
6. 4 x 1 10k Ohm resistor
7. 4 x 2 10 uF capacitors
8. 4 x 1 16 MHz clock crystal
9. 4 x 2 22 pF capacitors

### 3.4.4.1. OPERATIONS PERFORMED WHILE BUILDING THE STANDALONE ARDUINO SYSTEM

1. **Creation of Power lines and ground lines**

   Power and ground wires for where the voltage regulator will be placed. Also, power and ground wires were created at the bottom of the vero board connecting each rail.

2. **Addition of the L7805 voltage regulator and 10uF decoupling capacitors**

   A L7805 voltage regulator was fixed on the Vero board. The regulator is a TO-220 package where the Input from the external 9 volts power supply goes input on the left, ground is in the middle and the 5V output is on the right (when facing the front of the regulator). A power OUT and ground wires were fixed to the board which connects to the right and left rails of the Vero board. A 10uF capacitor fixed between the Input pin of the regulator and the ground. Also, a 10uF capacitor was fixed on the right rail between power and ground. The silver strip on the capacitor signifies the ground leg.

3. **Introduction of LED for power display**

   A 220-ohm resistor and an LED were fixed on the left side of the vero board across the voltage regulator (5 volts line). The LED attached to 5 volts line is used to know when the board is being powered as well as know if the board is being shorted.

4. **Addition of supporting circuitry**

   A 10kilo-ohm pullup resistor was connected to +5V supply from the RESET pin in order to prevent the chip from resetting itself during normal operation. The RESET pin reboots the chip when pulled down to ground.

In this design, the following pins of the ATmega328P chip were noted;

i. Pin 7 - Vcc - Digital Supply Voltage

ii. Pin 8 - GND

iii. Pin 22 - GND

iv. Pin 21 - AREF - Analog reference pin for ADC (Analog to Digital Converter)

v. Pin 20 - AVcc - Supply voltage for the ADC converter.

5. **Addition of the External Clock**

A 16 MHz Crystal Oscillator which serves as an external clock was introduced between pin 9 and 10, and two 22 pF capacitors were also added at each legs of the Crystal Oscillator running to ground from each of those pins.

### 3.5.2 DHT11 TEMPERATURE AND HUMIDITY SENSOR NODE

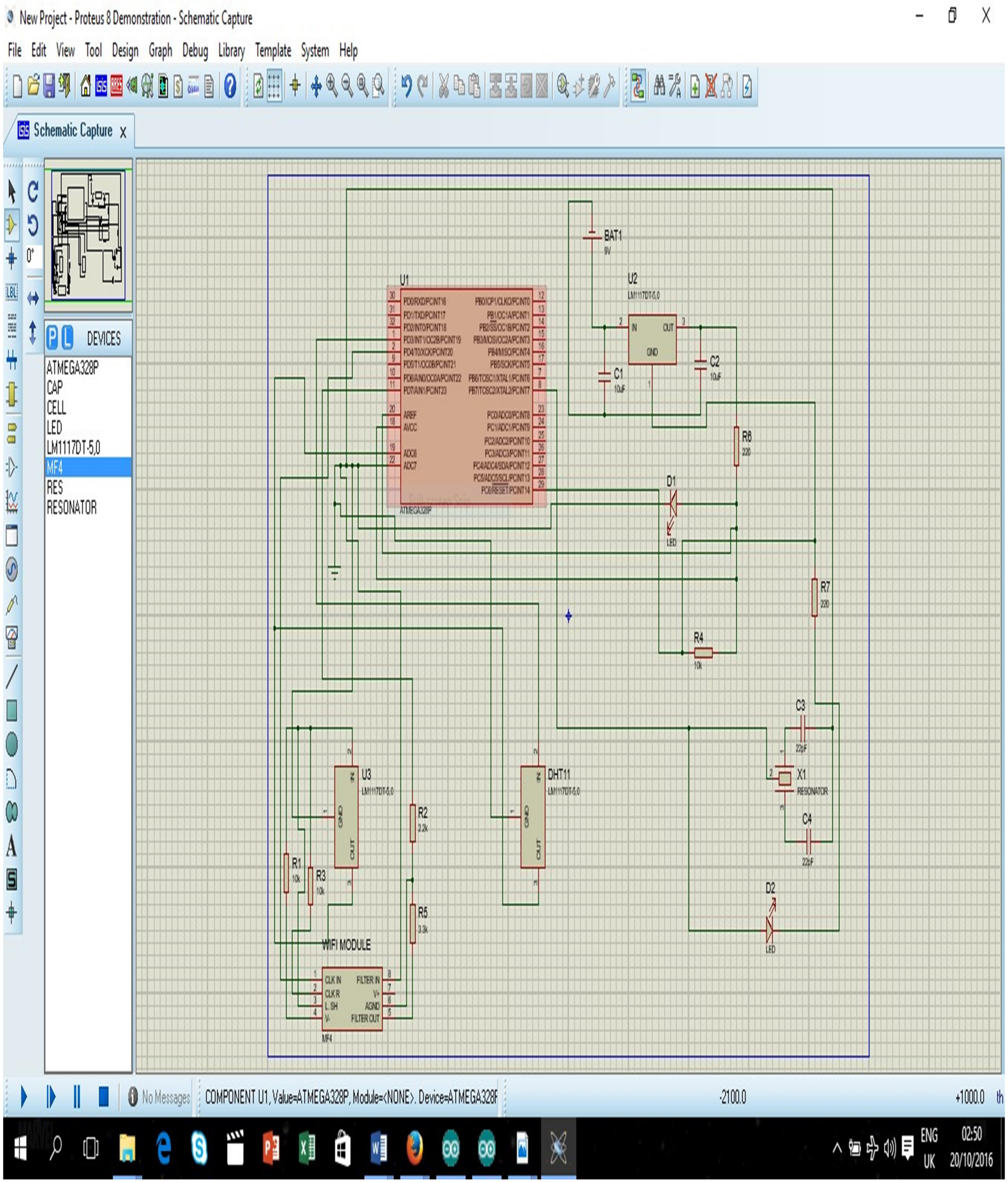



*Fig. 3.11: Circuit Simulation of the DHT11 Temperature and Humidity Sensor node*

### 3.5.3 GAS SENSOR NODE (MQ-2 node)

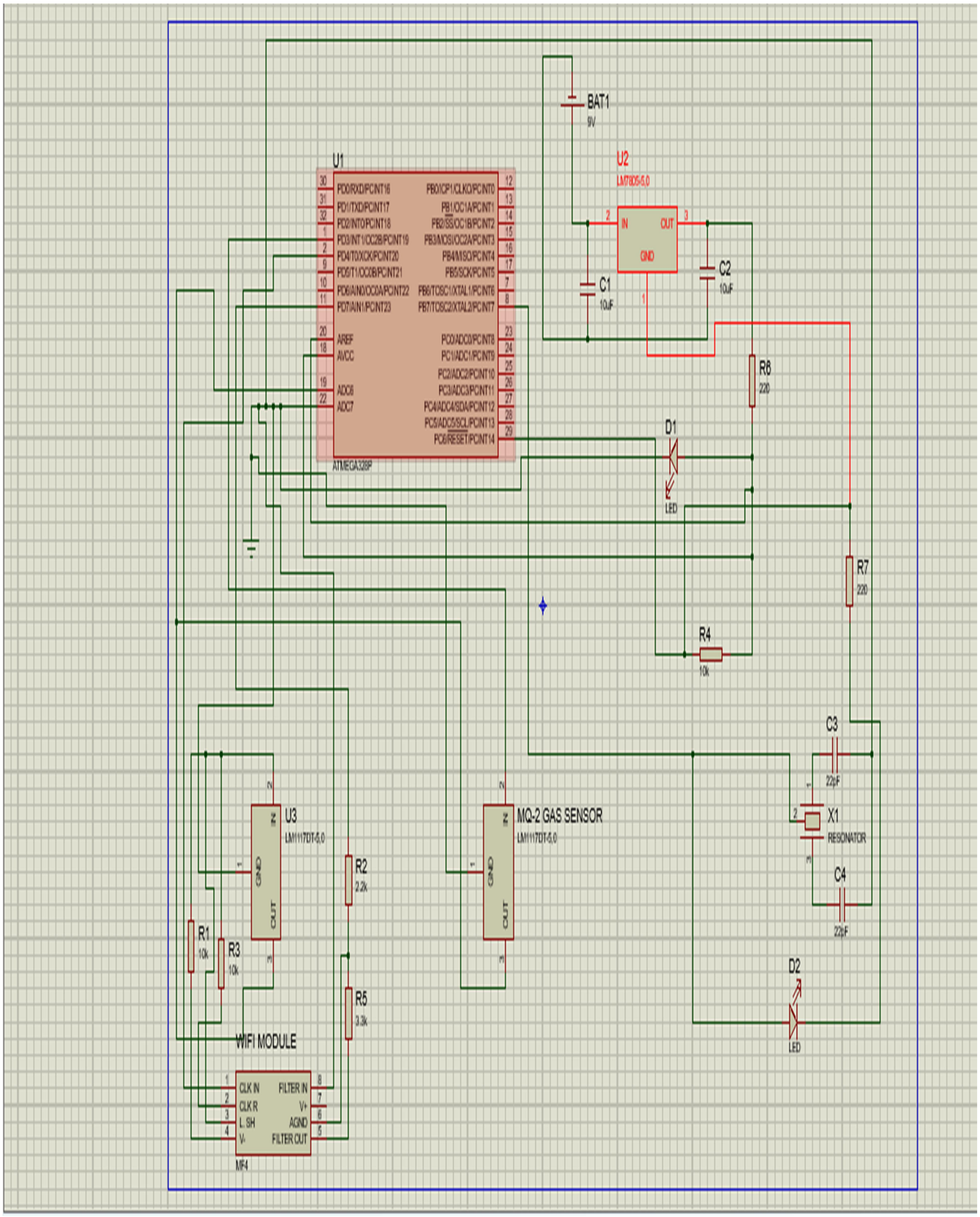



*Figure 3.13: Block Diagram of the Architecture of the Gas Sensor Node*

## 3.6. BLOCK DIAGRAM OF THE DATA LOGGING SYSTEM USING WIRELESS SENSOR NETWORK

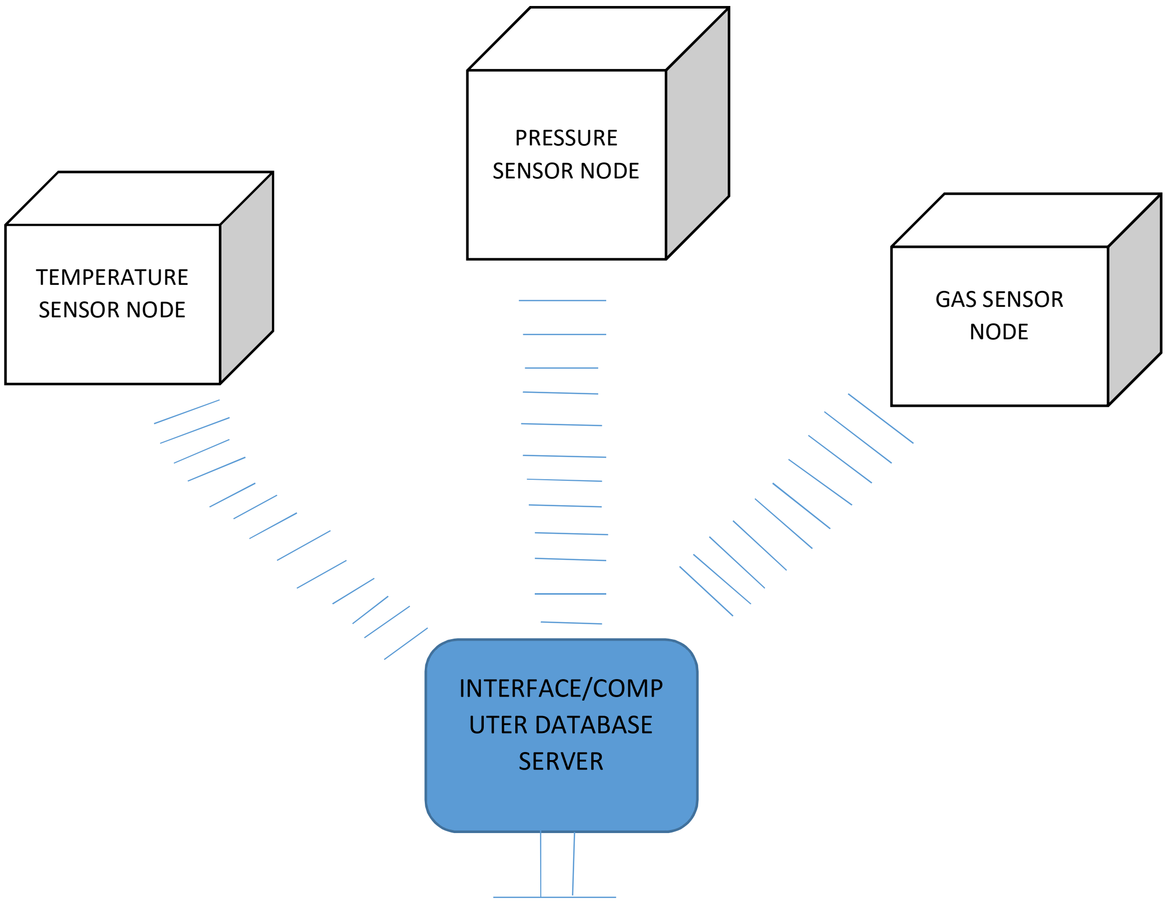
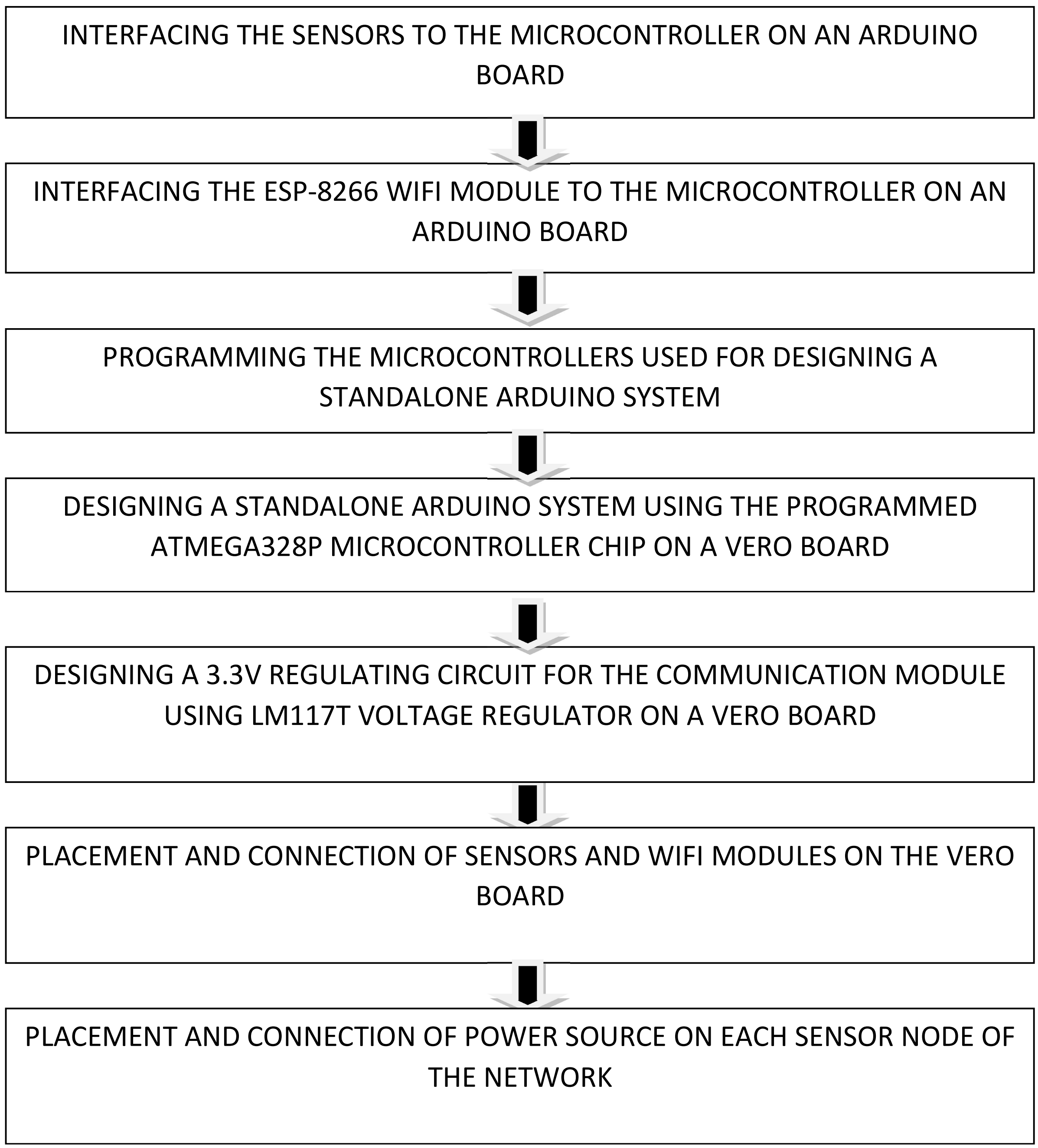



*Fig 3.14: Block Diagram of the Datalogging interface using Wireless Sensor Network*

### 3.6.1 MODE OF OPERATION OF EACH SENSOR NODE

At this stage each sensor node sends the values of the different parameters they have taken to the database system which is accepted and understood by the python code written on the database computer and displayed on the interface of the computer system

*Fig 3.15: Block diagram showing the step by step design of each Sensor node in the network*



## 3.7. SOFTWARE DESIGN OF THE DATA LOGGING INTERFACE

1. Creation of Hotspot Network on Database Server Computer
2. Installation of Python Scripting Software On the Database Server Computer
3. Creation of a Database Server on the computer using Python Scripting language

### 3.7.1. CREATION OF HOTSPOT NETWORK ON DATABASE SERVER COMPUTER

Hotspot is simply a wi-fi coverage area. A host computer is where this hotspot is created. In our design, the database server computer is also our datalogging interface, so it is paramount that the hotspot is created on the database computer or PC. The hotspot created is also dependent on if the computer is Wi-Fi enabled with the help of a driver as well as the operating system. These are the steps taken in a windows 8 system:

I. Open the command line prompt on the system
II. Run as administrator
III. Enter the following code on command line prompt:
   a) Netsh wlan set hosted network mode=allow ssid=KACHMAN key=onyekachi20
   b) Netsh wlan start hosted network

### 3.7.2. INSTALLATION OF THE PYTHON SCRIPTING SOFTWARE

1) Go to the python software foundation to download a reliable python software program
2) Run the setup as an administrator and install the program

### 3.7.3. CREATION OF A DATABASE SERVER USING PYTHON SCRIPTING LANGUAGE

In this part of the project, we will use some Python libraries to receive the data transmitted by the sensor nodes in a computer instead of the Arduino. Other devices such as a Raspberry pie can be used to receive data, store this data, manage and gather this data and display the data in a tabular form as well as a graphical form on an interface. This interface could be a monitor, laptop screen or a television screen.

This part of the project describes the concept of data logging or data acquisition that is key thing implemented in this project. Here is a step by step procedure how the database server is created:

1) Downloading and Installing a python library or command line tool;
   First you'll need to install the python Wi-Fi library. The python wi-fi library allows your Python script to talk with the network interface card in which the sensor nodes are connected to using wi-fi protocol. i.e. think of it as a stream connecting the Arduino code to the Python code. This library enables the python to talk to a MySQL database created. You can download the python wi-fi library here:
   https://pypi.python.org/pypi/wifi
2) Set up a MySQL database;
   a) Download and install the python library for MySQL
   b) Create a MySQL database and table to store our data from the sensor nodes depending on the sensor node that is connected and the parameter it is sensing
3) Write a Python script to insert the values gotten from the sensor node through the wi-fi protocol into the MySQL database.

# CHAPTER FOUR

# RESULTS AND DISCUSSION

As at this point of the report, the data logging interface for the wireless sensor network was not available. To verify if the sensor nodes designed were working, few experiments were conducted to test the sensor node components like ADC's, the ESP-8266 Wi-Fi module and onboard sensors. These experiments also make sure that code programmed into the Microcontroller runs on each of the sensor node. These experiments helped to understand the behavior of Wireless Sensor network.

Although the data logging interface was not available to view the readings, a serial monitor in the Arduino application was used as an alternative to monitor the display the readings from the Sensor nodes.

## 4.1. EXPERIMENTS CONDUCTED ON SENSOR NODES

### 4.1.1. DHT11 Temperature and Humidity Sensor node Setup:

Setup contains DHT11 temperature and Humidity sensor, a 10 kilo-ohms pull up Resistor, and the Standalone Arduino system to form a sensor node. In this design, the ATmega328P Microcontroller was programmed in such a way that temperature and humidity readings after every 5 seconds. The following readings were taken from an enclosed room under normal Atmospheric conditions on 15$^{TH}$ October at 9:40am. Sampling rate of the following readings was 2Hz and the duration of test is 15 minutes.

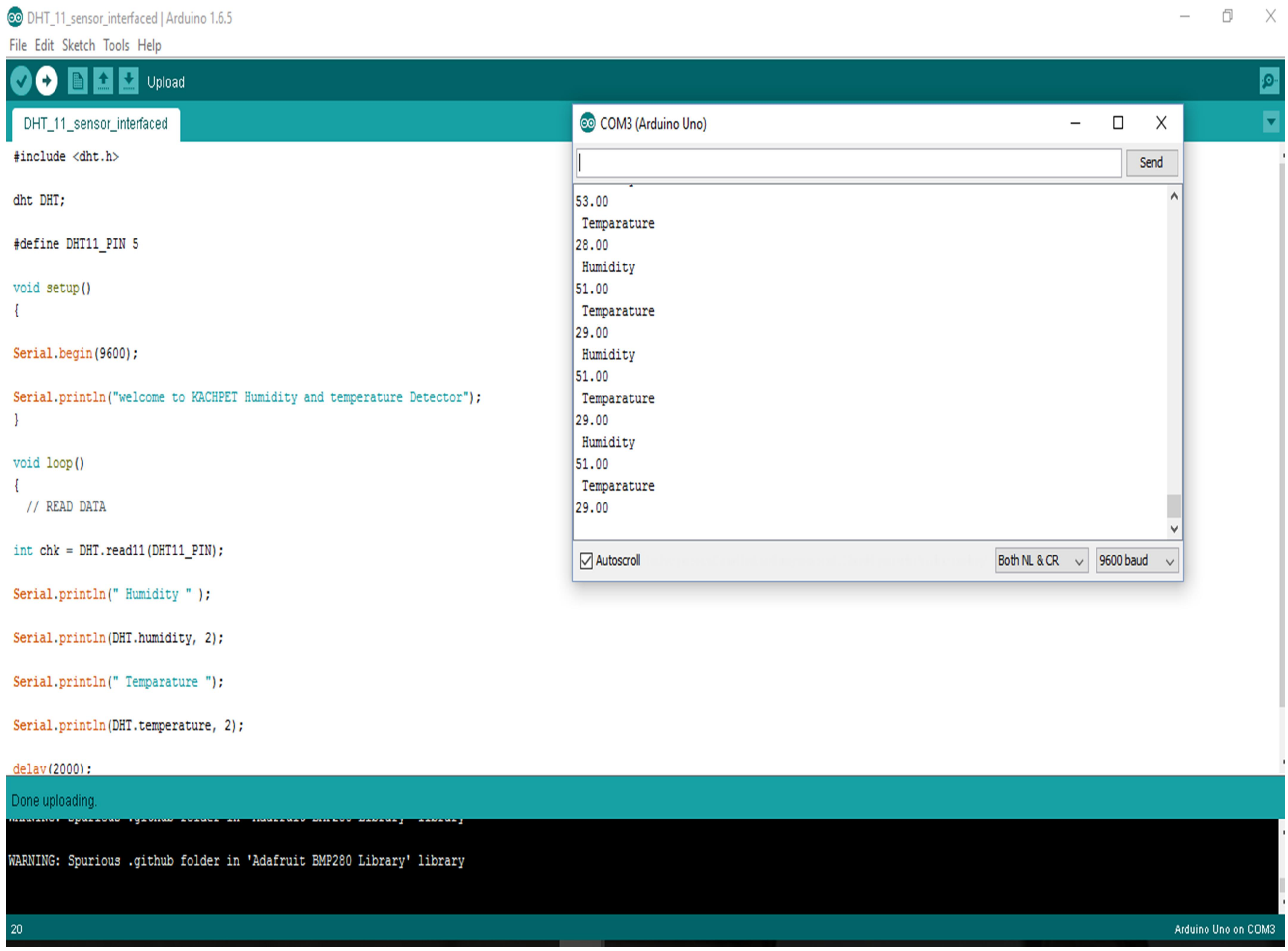



*Fig 4.1: Results from the temperature and humidity monitoring conducted in an enclosed room.*

## Observation and Conclusion

From the results derived, it was observed that the temperature and humidity of the enclosed room at that time interval varied from 28-29 degrees Celsius in temperature and 51-53 % humidity. This partially corresponded to the actual temperature readings of the enclosed room at that instance using a Standard temperature measuring device. Therefore, the sensor node can be considered as a standard monitoring device.

### 4.1.2. MQ-2 Gas Sensor node Setup:

The Setup of this node contains MQ-2 gas sensor, which placed on a bread board and connected directly to an Arduino Uno board using male jumper wires to form a sensor node. The ATmega328P Microcontroller was programmed to sense the environment for physical parameters every 1 second. This node senses Liquefied Petroleum Gas (LPG), Carbon Monoxide (CO) and smoke, all measured in PPM. An Experiment was carried out in an enclosed room on $16^{TH}$ of October at 2:15pm to confirm whether the node actually senses Carbon Monoxide (CO) and Smoke. The experiment was performed by placing burning materials close to the Sensor node for 5 minutes and the following readings were gotten from the experiment;

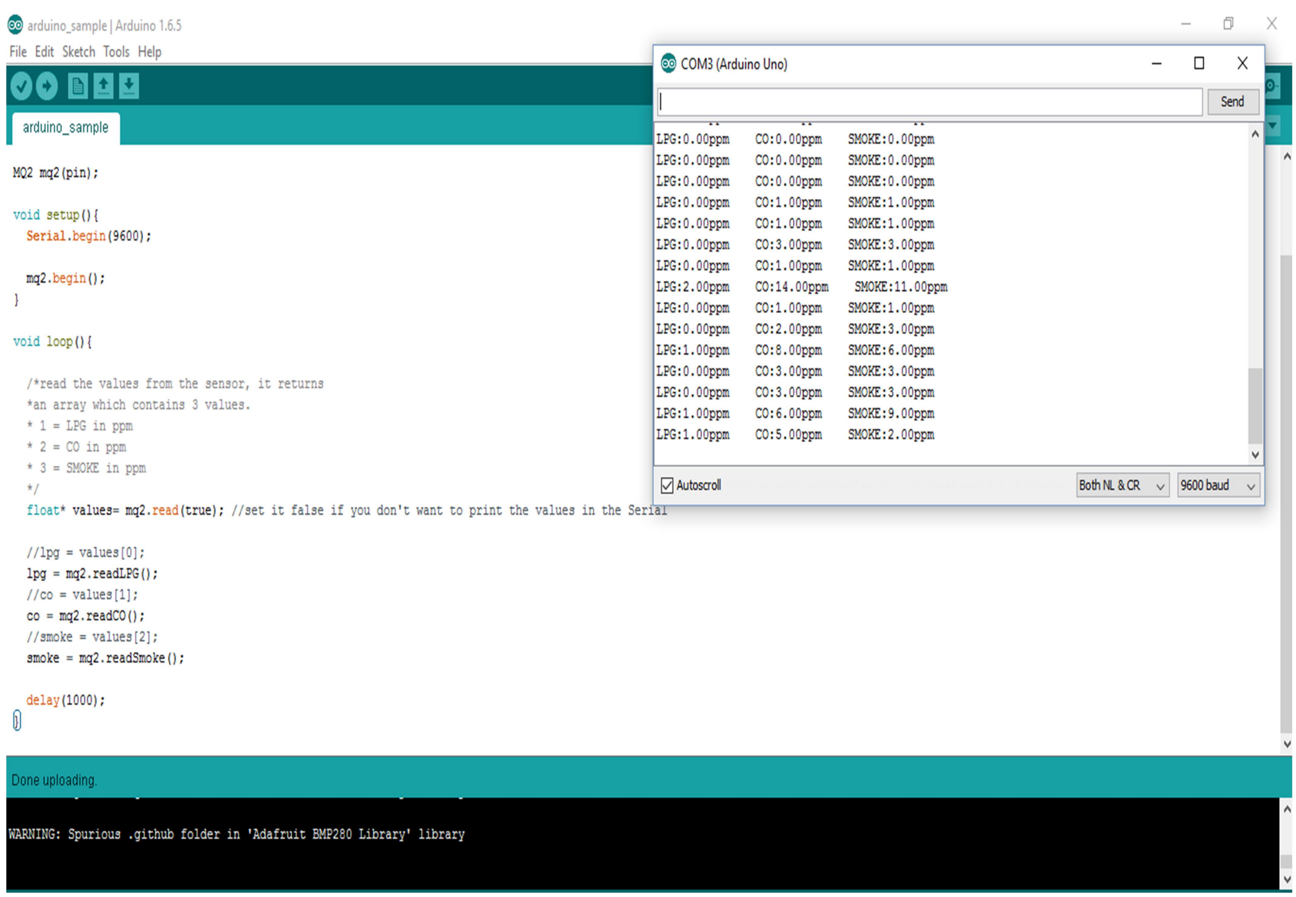



*Fig 4.2: Results from the gas monitoring conducted in an enclosed room.*

**Observation and Conclusion**

During the experiment, it was discovered that as the intensity of the smoke (which also contains CO gas) increases, the readings displayed on the Serial monitor increased and also decreases as the intensity of the smoke decreases. This shows that the MQ-2 gas sensor node functions as expected and gives a precise result.

### 4.1.3. GY-91 Pressure Sensor node Setup:

Setup contains an 8-pin GY-91 Pressure sensor, which was properly connected to the Standalone Arduino system to form a sensor node. In this design, the ATmega328P Microcontroller was programmed in such a way that temperature and humidity readings after every 7 seconds. The readings were taken from an enclosed room under normal Atmospheric conditions on 16$^{TH}$ October at 9:30am. The duration of test is 10 minutes.

**Observation and Conclusion**

From the results derived, it was observed that the temperature and humidity of the enclosed room at that time interval varied from 758-762mmHg. This partially corresponds to the actual pressure readings of the enclosed room at that instance using a Standard measuring device. Therefore, the sensor node can be considered as a standard monitoring device.

After the Sensor nodes were tested, the Wi-Fi module was interfaced on a bread board. A 3.3 volts regulating circuit was used to regulate the voltage from the microcontroller. After the modules have been properly connected, an AT command was sent to the ESP-8266 Module to initiate connection to other devices by creating a Hotspot using an IP address.

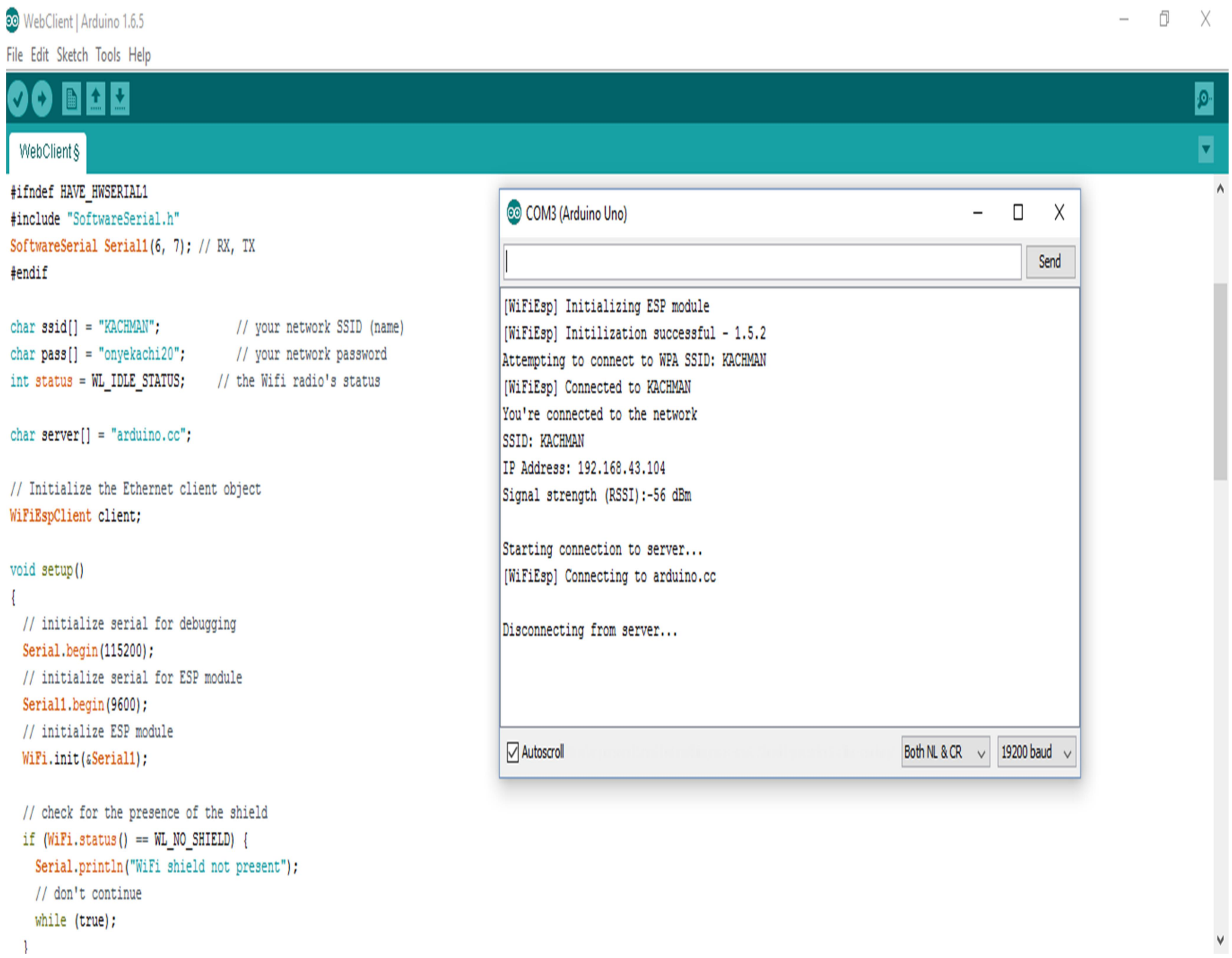



*Fig 4.3: A Serial monitor displaying the ESP-8266 connection to other devices.*

The diagram above show how the ESP-8266 Modules create connection to a client. In this design, the client is a PC. Once connection has been made, data from each node are sent wirelessly to the PC that performs the data logging function.

# CHAPTER FIVE

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 CONCLUSIONS

With the goal creating a data logging system, a network of wireless sensors was created. This enabled us to build sensor nodes with different sensors taking values of different parameters. The project was implemented such that the system can be used to log data in any environmental or industrial process such as a gas plant, a chemical processing plant, vehical testing plants, tank level determination or any manufacturing plant as long as parameters such as temperature, pressure, water flow, altitude, humidity etc. are involved.

Real time monitoring using wireless sensor networks have been in existence before this project was carried out. The technology used in implementing the wireless sensor nodes are improved of ways implanting it. The use Arduino boards and the Arduino IDE software for programming the controller chips made the job easier and more efficient in its implementation. Also, the use of python scripts programming has also proven to be a very easy and efficient way of collecting and storing data in a database server created using MySQL. Previous technologies used are either too tedious and complex or outdated such as timote sky platform and Java programming.

Despite several limitations that wireless sensor networks are faced with, they are still among the best techniques that can be adopted in real time monitoring and data logging.

## 5.2. KNOWLEDGE ACQUIRED

1. By carrying out this project, we have been able to acquire programing skills and knowledge that can be useful later in life.
2. We were able to develop our abilities to work as team players
3. As a result of working on this project, we have been able to suggest a different method that can be used in implementing data logging and wireless sensor networks

## 5.3. PROJECT CONSTRAINTS

The most prominent constraint encountered in the project was majorly the bulk of the work as well as good knowledge of all softwares and programming language used within limited time. Due to this problem, the creation of a data logging interface for the sensor network was actualized. The alternate means for displaying the sensed parameters was with the use of the serial monitor of the Arduino application, which was not as efficient as the data logging interface.

Another problem encountered in this design was the availability of the sensor network components in the local market. Most of the components used in the design were unavailable in the market, leading to the importation of these components, which is not cost effective and takes a lot of time to arrive.

## 5.4. PROJECT LIMITATIONS

1. The system is not sturdy, and consequently, leaves no guarantee against the elements of the weather such as thunder, heavy rain etc. when used in the outdoor places or field.
2. The system is also not protected against manmade disturbances or interference

## 5.5. RECOMMENDATIONS

a.) Due to time constraint, data logging and real time remote monitoring was not achieved. We advised that it should be very much considered if this project is to be improved upon

b.) Due to advancement in technology and the speed at which it is moving,

c.) For commercial purposes, the frame of the wireless sensor nodes should be made of a robust material that can withstand harsh climatic conditions.

d.) A more efficient battery power module can be used.

## REFERENCES


[2] J.I. Chanin, A.R. Halloran, Prof. E. Agu, Prof. W. Lou. Wireless Sensor Network for Monitoring Applications. In Project Report WORCESTER POLYTECHNIC INSTITUTE

[3] A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, W.Hong. "Model-Driven Data Acquisition in Sensor Networks". In Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

[4] Adam Linsay, Dr.R.Bowden, "Wireless Sensor Project"

[11] K.Pahlavan, A.H Levesque "Wireless Information Networks (2nd edition)". In John Wiley and Sons Inc.

[12] C.S.Murthy, B.S.Manoj "Ad Hoc Wireless Networks; Architectures and Protocols". In Published by Prentice Hall

[13] Z. Teffera. Process Control over Wireless Sensor Networks. In Master's Degree Project, Stockholm, Sweden 2013

[14] *http://ww.sparkfun.com/products/13678*

[15] http://www.etsi.org/. In ETSI: EN 300 328; European Standard, version 1.7.1 (Published by ETSI)

[16] Prof. Dr.-Ing. A.Sikora "ZigBee Competitive Technology Analysis" Section Summary. In ZigBee organization. http://www.zigbee.org/en/resources/whitepapers2.asp

[17] http://www.zigbee.org/en/resources/whitepapers2.asp "ZigBee and Wireless Radio Frequency Coexistence". In ZIgbee Organization

[18] Rakesh "AVR Microcontroller (Atmega32) – An Introduction"**.** In http://www.circuitstoday.com/atmega328-avr-microcontroller-an-introduction

[19] L.Sanabria, J.Barcelo ”A Course On Wireless Sensor Network”. (April 2012)

[20] S. Chaudhary, N. Singh, A. Pathak and A.K Vatsa. “Energy Efficient Techniques for Data aggregation and collection in Wireless Sensor Networks”. In IIMT Institute of Engineering & Technology, Meerut, U.P, India

[31] SenSys 2004 Call for Papers. http://www.cis.ohio-state. edu/sensys04

[32] IPSN 2004 Call for Papers. http://ipsn04.cs.uiuc.edu/ call_for_papers.html.

[33] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In ACM SIGMOD, 2003.

[34] Y. Yao and J. Gehrke. Query processing in sensor networks. In Conference on Innovative Data Systems Research (CIDR), 2003.

[35] Kollios, J. Considine, F. Li, and J. Byers. Approximate aggregation techniques for sensor databases. In ICDE, 2004.

[36] Global Markets and Technologies for Wireless Sensors, http://www.bccresearch.com/report/ wireless-sensors-technologies-markets-ias019a.html

[37] Guibas, Leonidas and Zhao, Feng, Wireless Sensor Networks: An Information Processing Approach, Morgan Kaufman Publishers, San Francisco, CA, 2004

[40] Data Acquisition Systems Then and Now, UEI Webnars ,http://www.ueidaq.com/data-acquisition.html

[41] Wang, Qinghua, and IlangkoBalasingham. ”Wireless sensor networks-an introduction.” Wireless Sensor Networks: Application-Centric Design, ISBN: 978953-307-321-7, DOI: 10.5772 13225 (2010)

[42] Pottie, Gregory J., and William J. Kaiser. ”Wireless integrated network sensors.” Communications of the ACM 43.5 (2000): 51-58

[43] Smart Dust, Autonomous sensing and communication in a cubic millimeter, http://robotics.eecs.berkeley.edu/pister/pisters/pre99/SmartDust.html , Accessed March 2013

[44] Zhang, Wei, Michael S. Branicky, and Stephen M. Phillips. ”Stability of networked control systems.” Control Systems, IEEE 21.1 (2001): 84-99

[45] Dietrich, I. and Dressler, F. ”On the life-time of wireless sensor networks”, ACM Trans-actions on Sensor Networks (TOSN), v.5 n.1, p.1-39, February 2009.

[46] Liu, A. Goldsmith. Cross-layer design of distributed control over wireless network. Systems and Control: Foundations and Applications, (Ed. T .Basar), Birkhauser, 2005.

[47] P. Park, J. Araujo, and K. H. Johansson, Wireless networked control system co-design, IEEE International Conference on Networking, Sensing and Control, Delft, the Netherlands, 2011

[48] Park, Pangun, et al. ”Breath: an adaptive protocol for industrial control applications using wireless sensor networks.” Mobile Computing, IEEE Transactions on 10.6 (2011): 821-838

[49] Park, Pangun, Carlo Fischione, and Karl Henrik Johansson. ”Adaptive IEEE 802.15. 4 protocol for energy efficient, reliable and timely communications.” Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks. ACM, 2010

[50] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 104112

[51] Yick, Jennifer, Biswanath Mukherjee, and DipakGhosal. ”Wireless sensor network survey.” Computer networks 52.12 (2008): 2292-2330.

[52] Hespanha, Joao P., PayamNaghshtabrizi, and Yonggang Xu. ”A survey of recent results in networked control systems.” Proceedings of the IEEE 95.1 (2007): 138162

[53] Holger Karl, Andreas Willig, Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, 2005

[54]Szewczyk, Robert, et al. ”An analysis of a large scale habitat monitoring application.” Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 2004

[55] S. Kim. Wireless sensor networks for structural health monitoring. Master’s thesis, University of California at Berkeley, May 2005

[56] Swartz, R. Andrew, et al. ”Structural monitoring of wind turbines using wireless sensor networks.” Smart Structures and Systems 6.3 (2010): 183-196

[57] Jeong Gil Ko, Tia Gao, Richard Rothman, Andreas Terzis. ”Wireless Sensing Systems in Clinical Environments: Improving the Efficiency of the Patient Monitoring Process”. IEEE Engineering in Medicine and Biology (EMB) Magazine, Volume 29, Issue 2, Pages 103-109, 2010.

[58] Cantoni, Michael, et al. ”Control of large-scale irrigation networks.” Proceedings of the IEEE 95.1 (2007): 75-91