IoT Based Coal Mine Safety System and Monitoring Using Arduino UNO



SONARGAON UNIVERSITY (SU)

Supervised By

Md. Rakibul Alam
Assistant Professor EEE, SU

Submitted By

Md. Tanveer Alam	**ID: EEE1902017054**
Setu Rani Datta	**ID: EEE1902017087**
Md. Martuza Alam	**ID: EEE1902017105**
Md. Abul kalam	**ID: EEE1902017109**
Abu Bakker Siddik	**ID: EEE1902017097**

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthapath,
Dhaka-1215, Bangladesh

Date of Submission: January - 2023

Declaration

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree.

.....	….....
Md. Tanveer Alam	**Setu Rani Datta**
.....	….....
Md. Martuza Alam	**Md. Abul kalam**
.....	
Abu Bakker Siddik	

Under Supervision of

.....

Md. Rakibul Alam

Assistant Professor EEE, SU

Certification

This is to certify that this project entitled **"IoT Based Coal Mine Safety and Monitoring System Using Arduino UNO"** is done by the following students under my direct supervision. This project work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervisor

.....

Md. Rakibul Alam

Assistant Professor EEE, SU

ACKNOWLEDGEMENT

The report titled as on ”**IoT Based Coal Mine Safety and Monitoring System Using Arduino UNO”** has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation.

At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable teacher **Md. Rakibul Alam,** Assistant Professor, Department of Electrical & Electronics Engineering, SU– Sonargaon University, Dhaka, for his valuable and patient advice, sympathetic assistance, co-operation, contribution of new idea. Deep theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

Finally, we would like to thanks again to the respected **Vice- Chancellor of SU, Professor Dr. Md. Abul Bashar** also thanks to **Head of Department of SU , Electrical & Electronics Engineering, Professor Dr. M. Bashir Uddin** because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under Bsc in EEE program, and that will be very helpful for our prospective career.

We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report

ABSTRACT

Coal Mining is a dangerous and potentially lethal job. Workers in coal mines face many environment related problems such as high temperature, humidity, release of harmful gases to name a few. As a means of reducing the intensity of the problem, our present work focuses on sensing these environmental parameters from different locations in the mines and sends alert to the mine control room operator who uses a Blynk Application whenever there seems to be a dangerous situation. For this project, a coal mine safety system is implemented by the use of a webpage as a medium to transmit the data. The device is introduced to monitor various parameters inside the coal mines, such as light detection, gas leakage, temperature and humidity conditions, and fire detection. This system of sensors is known as one large device and is mounted in the coal mines. All the sensor rankings are automatically fed into the thinking processors, which generates a multitude of ideas for them to explore. Here, gas monitoring equipment is still running in order to detect any possible issues, and a buzzer is used to warn the staff. Laser-detection (LDR) sensors are used in this framework to detect the presence of light. Automatically lights illuminate when you have them on, and they can be operated using the LED button. To ensure the proper response to any potential fire hazard, a notice is sent to the mail of the designated individual. A collection of constantly measured and recorded temperature values is available as well, shown on the serial monitor and the website platform. Additionally, a vibration sensor is used to detect any movements in the mine.

TABLE OF CONTENTS

Topics	Page No
DECLARATION	II
CERTIFICATION	III
ACKNOWLEDGEMENT	IV
ABSTRACT	V
TABLE OF CONTENTS	VI-VII
LIST OF FIGURES	VIII
LIST OF TABLES	IX

CHAPTER I: INTRODUCTION

TOPICS		Page No.
1.1	Introduction	10
1.2	Background Study	11
1.3	Objectives	11
1.4	Project Organization	11

CHAPTER II: DISCRIPTION OF THIS EQUIPMENTS

2.1	Introduction	12
2.2	Theory	12
2.3	Arduino Uno	14
2.4	SIM 800l GSM Module	15
2.5	Buck Converter	19
2.6	SMPS	20
2.7	Smoke Sensor (MQ-2)	20
2.8	Ultrasonic Sensor	21
2.9	Soil Moisture Sensor	22
2.10	LRD	22

2.11	Buzzer	24
2.12	Temperature Sensor	23

CHAPTER III: DESIGN OF THE PROJECT

3.1	Introduction	25
3.2	Block Diagram	25
3.3	Circuit Diagram	26
3.4	Full Over View	27
3.5	Working Principle	28
3.6	Conclusion	28

CHAPTER IV: SOFTWARE DEVELOPMENT

4.1	Arduino IDE	29
4.2	Proteus 8 Professional	37
4.3	Blynk App	38

CHAPTER V: RESULT & DISCUSSION

4.1	Introduction	45
4.2	Result	45
4.3	Advantages of smart grid	45
4.5	Discussion	46

CHAPTER VI: CONCLUSION

5.1	Conclusion	47
5.2	Future Work	47
References		48
Appendix		50-59

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
2.3	Arduino UNO Pin Diagram	14
2.4	SIM 800l GSM Module	16-18
2.**5**	Buck Converter	19
2.6	SMPS	20
2.7	Smoke Sensor (MQ-2)	20
2.8	Ultrasonic Sensor	21
2.9	Soil Moisture Sensor	22
2.10	LDR	22
2.11	Buzzer	23
2.12	Temperature Sensor	24
3.1	Block Diagram of this Proposed System	25
3.3	Circuit Diagram of the Project-1	26
3.4	Overview of Proposed System	27

LIST OF TABLE

Table 2.1: Technical specifications of Arduino UNO 17

CHAPTER I

INTRODUCTION

1.1 Introduction

Coal Mining is inherently dangerous with zero margin for error even under the best conditions. Even a small lapse in safety can be catastrophic. The principal airborne hazards in the mining industry include several types of particulates, naturally occurring gases, engine exhaust and some chemical vapours; the principal physical hazards are noise, segmental vibration, heat, changes in barometric pressure and ionizing radiation. The present paper is about the system which uses an IoT platform as a medium to transmit data. The system is implemented to monitor various parameters in the coal mines such as light detection, leakage of gas, temperature and humidity conditions and fire detection in the coal mine. These all sensors are together placed as a unit and sent into the coal mine and can be monitored by the operator using Blynk Application. The Internet of Things (IoT) is just another term for devices that connect to the internet and talk to each other. There are several different types of IoT applications on a wide scale. Cluster on the internet of Things classifies IoT technologies as some of the most critical areas for building thoughtful facilities, the advancement of smart transport, the advance of sensible power, smart industry, smart health, and smart environments. The IoT may be a technology that serves as a trendsetter that collects and stores all device information in the cloud wherever it is accessible via the internet. This technology also uses sensors and actuators for gathering information and distributing it across the network. Additionally, we use cloud to store data, but we also use it to conduct data analysis, collect data, and visually present data. So, if increasing technologies such as associate degree in applied science rises is often utilized to expand existing systems and make them more effective in numerous IoT applications, for example, agriculture, health, sensible home, etc., In order to say something positive about coal, you should say that it is a non-renewable energy source, and as a result, it can never be commonly replaced by humans. As a result, incidents of coal mine explosions occur, and miners put their lives at risk every day, since they are often employed in the coal mines.

1.2 Background Study

Coal Mining is a dangerous and potentially lethal job. In the Coal Mine Safety System and Monitoring, the Previous generation used 3 sensors and a buzzer. In this project, we have used 5 sensors and a Light. We look forward to future improvements to make our system design more efficient. In the future, we plan to add additional features to the system, such as surveillance camera monitoring and a voice broadcasting system. In the future, we are planning to launch an IoT system that can be tracked wirelessly by Man Power for any occurrence.

1.3 Objectives

There have many many importance and objective of this IoT based Coal mine safety system. We included here some objectives in bellow:

- ❖ To monitor the temperature value in coal mine using Smart Phone from anywhere in the world.
- ❖ To detect the lighting value in the coal mine using LDR.
- ❖ To Monitoring the water level in ground side of coal mine.
- ❖ To define the presense of smoke and Gas quantity in the Coal Mine side.
- ❖ To detect the soil level in the working area in the coal mine.

1.4 Project Organization

This project book is arranged into the following chapters:

- Chapter I: gives a brief introduction to the Theoretical background of the project.
- Chapter II: focuses on the disign and fabrication scheme of this system.
- Chapter III: focuses on the design of real-time based monitoring and efficient transformer protection system with GSM communication. Also discusses the performances of the implemented prototype system.
- Chapter IV: concludes the overall discussion software development in this chapter.
- Chapter V: Includes Results and advantages.
- Chapter VI: Includes the conclusion of this system.

CHAPTER II

DESCRIPTION OF THIS EQUIPMENTS

2.1 Introduction

This chapter includes the total over view of the device. In this chapter we have followed-up the theory of Arduino Uno, Soil sensor, temperature Sensor, Smoke Sensor, LDR, SMPS, GSM Module, Buck Converter, and other components. Here we can know that the total system overview of the projects. And we will also know that how the equipment are working with each other.

2.2 Theory

We've developed the system that can monitor every danger situation in every second and can make sure the safety by sending realtime data to every smart phone user who will connected with this system. In this system here used some specific sensors to sence the data and this all data are sended by IoT platform. Here used Blynk app which is generaly used in any IoT based operation. In this system we can measure Temperature, Smoke, Gas, Water level, Soil Moisture and Brightness of light. As the administrators of the Coal Mine Monitoring section, Yongping Wu and Guo Feng make use of A Bluetooth wireless transmission device has been implemented in coal mine monitoring to greatly expand the scope of the technology. Currently, the industry believes that to control costs and provide a standard in the short-range wireless networking market, Bluetooth is supposed to implement a powerful kit gap scheme, and it is also expected to provide a minimum power. demand for low-cost, low-power air interfaces. This paper describes the context of the situation, as well as different technical choices, as well as the architecture of the protocol stack of Bluetooth technology. Also, the paper goes into possible solutions for the Bluetooth HCI wireless networking, which is a necessary component in the development of this technology [2]. Also at the same same moment, the system has completed the integration of wired and wireless information transfer. This use of the new technologies, known as the "Wired Bus," is being employed. The problems that come along with this strategy include the main ones. For one, Bluetooth is a short-range wireless system. The implementation

of cabling, in this case, can be very complicated. In the case of a natural disaster or roof failure, the cabling is most likely damaged. That the long lifespan and extensive burden of standard communication systems is problematic. Even though it's difficult to work in the mine due to the abrasive environment, it is even more difficult to install and maintain the wired communication system. The DCS Coal Mine Monitoring System was developed by Zhenzhen Sun. A computerized bus that runs on the same lines as RS485 uses a bus structure that supports multipoint and two-way communication. So, the design of such a watching device will be done using commonly available 8-bit microcontrollers. With a circuit structure that is both simple and inexpensive, you enjoy the advantages of inexpensive circuit design. Thus, on the contrary, due to the use of a master-slave network arrangement, it is almost impossible to ensure the liability of the network structure. In addition, the amount of information that can be transmitted over a line with a subpar real-time output is also limited [3-4]. the JINGLING SONG and YINGLI ZHU's design of an automated mine safety monitoring device assisted by a wireless sensing element network. MSP430F and nRF2401 have been implemented in the process design to watch for mine protection. Sensor teams in the device intensively track temperature, humidity, and various parameters in the underground mine. Temperature, humidity, and various parameters are monitored by sensors and sent to a wireless communication module by the microcontroller. That information is sent via cable to the far-off location where it is watched by the observers [5]. The problem with this implementation is that hardware is installed in the coal mines when disasters or roof collapses occur, and once the damage has occurred, it's impossible to repair the device. Typical communication systems have a bad track record when it comes to reliability and long life. A great amount of work is needed to keep this mine in operational condition because of the harsh working conditions inside it. The other drawback is that the working state of the coal pit is extremely shaky, and if the gap between the manual labourer and the system is long, the miner will not receive the correct instructions. In the style of Y. S. Dohare and Tanmoy Maity, the Ogendra S. Dohare and Tanmoy Maity police investigation and protection scheme for underground coal mines used Low Power WSN. while the device is operating, a coffee control, costeffective, and Zigbee protocol based mostly wireless detector network that delivers an intelligent police investigation associate-trained protection system for underground coal mines. This is an example of a wireless networking scheme, where several nodes are associated with one another through a wireless connection. It is very simple to set up this network within underground mines. Underground coal miners have an efficiently policed

investigation and safety framework due to their having earned a degree. These nodes are used to link the miners on the surface to the room system. These nodes provide real-time electronic contact between miners and the room system. This approach is applied to my equipment and hence the drawback occurs because manual labourers don't seem to be sufficiently proficient with the procedure. Although this device is primarily focused on the underground mine, it does not control the physical health of the miners.

2.3 Arduino Uno

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

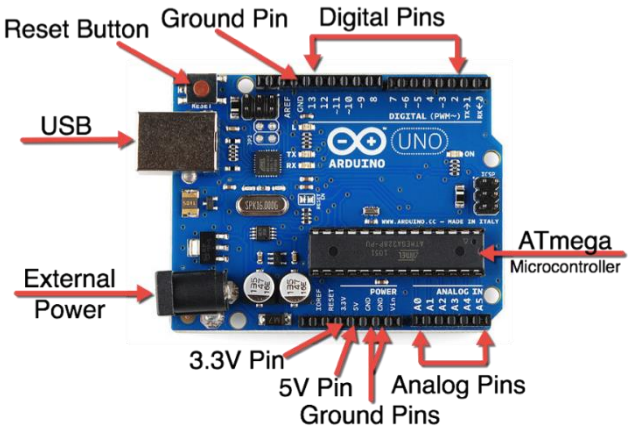


Fig: 2.3 Arduino UNO Pin Diagram

You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language and the Arduino Software (IDE), based on Processing.

Technical specifications:

Table 2.1: Technical specifications of Arduino UNO

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

2.4 SIM 800l GSM Module

TheSIM800l module is used to send the data from the microcontroller unit to the mobile phone by using AT commands. First put the SIM card in the SIM slot given in the SIM800l module. The receiver and transmitter pins of this module is connected to the transmitter and receiver pins of the ARDUINO board and also connect the ground and VCC pins of both ARDUINO and SIM800l modules.

At the heart of the module is a SIM800L GSM cellular chip from SimCom. The operating voltage of the chip is from 3.4V to 4.4V, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.



Figure 2.4.1: Different part of GSM Module

All the necessary data pins of SIM800L GSM chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 1200bps to 115200bps with Auto-Baud detection.

The module needs an external antenna to connect to a network. The module usually comes with a Helical Antenna and solders directly to NET pin on PCB. The board also has a U.FL connector facility in case you want to keep the antenna away from the board.

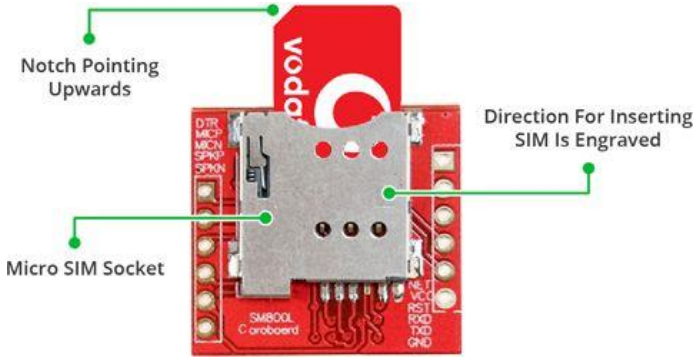


Figure 2.4.2: SIM inserting Procedure

There's a SIM socket on the back! Any activated, 2G micro SIM card would work perfectly. Correct direction for inserting SIM card is normally engraved on the surface of the SIM socket. This module measures only 1 inch² but packs a surprising amount of features into its little frame. Some of them are listed below:

- Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900
- Connect onto any global GSM network with any 2G SIM
- Make and receive voice calls using an external 8Ω speaker & electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts
- Transmit Power:

- Class 4 (2W) for GSM850
- Class 1 (1W) for DCS1800
- Serial-based AT Command Set
- FL connectors for cell antennae
- Accepts Micro SIM Card

There There is an LED on the top right side of the SIM800L Cellular Module which indicates the status of your cellular network. It'll blink at various rates to show what state it's in:



Figure 2.4.3: Blink every 1s time

The module is running but hasn't made connection to the cellular network yet.



Figure 2.4.5:Blink every 2s time

The GPRS data connection you requested is active.



Figure 2.4.6:Blink every 3s time

The module has made contact with the cellular network & can send/receive voice and SMS.

An antenna is required to use the module for any kind of voice or data communications as well as some SIM commands. So, selecting an antenna could be a crucial thing. There are two ways you can add an antenna to your SIM800L module.

The first one is a Helical GSM antenna which usually comes with the module and solders directly to NET pin on PCB. This antenna is very useful for projects that need to save space but struggles in getting connectivity especially if your project is indoors.



Figure 2.4.7: Use a Helical GSM antenna

You can snap-fit this antenna to small connector located on the top-left corner of the module. This type of antenna has a better performance and allows putting your module inside a metal case – as long the antenna is outside.



Figure 2.4.8: Use a Long Antenna

One of the most important parts of getting the SIM800L module working is supplying it with enough power.

Depending on which state it's in, the SIM800L can be a relatively power-hungry device. The maximum current draw of the module is around 2A during transmission burst. It usually won't pull that much, but may require around 216mA during phone calls or 80mA during network transmissions. This chart from the datasheet summarizes what you may expect: Since SIM800L module doesn't come with onboard voltage regulator, an external power supply adjusted to voltage between 3.4V to 4.4V (Ideal 4.1V) is required. The power supply should also be able to source 2A of surge current, otherwise the module will keep shutting down. Here are some options you can consider to correctly power your GSM module.

The SIM800L GSM/GPRS module is a miniature GSM modem that can be integrated into a large number of IoT projects. You can use this module to accomplish almost anything that a

normal cell phone can do such as sending SMS messages, making phone calls, connecting to the Internet via GPRS, and much more.

To top it all off, the module supports quad-band GSM/GPRS networks, which means it works pretty much anywhere in the world. The chip's operating voltage ranges from 3.4V to 4.4V, making it an ideal candidate for direct LiPo battery supply. This makes it a good option for embedding in projects that are short of space. All the necessary data pins of the SIM800L GSM chip are broken out to a 0.1″ pitch headers, including the pins required for communication with the microcontroller over the UART. The module supports baud rate from 1200bps to 115200bps with auto-baud detection. An external antenna is needed for the module to be connected to the network. The module usually comes with a helical antenna that can be soldered onto the module. The board also features a U.FL connector in case you want to keep the antenna away from the board. An antenna is required to use the module for any type of voice or data communication. So selecting an antenna can be a crucial thing. There are two ways to add an antenna to the SIM800L module.

2.5 Buck Conveter

Buck Converter is a type of chopper circuit that is designed to perform step-down conversion of the applied dc input signal.

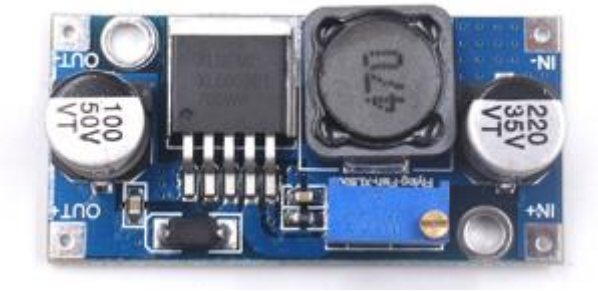


Figure 2.5: Buck Converter

In the case of buck converters, the fixed dc input signal is changed into another dc signal at the output which is of lower value. This means it is designed to produce a dc signal as its output that possesses a lower magnitude than the applied input.

2.6 SMPS

A switched-mode power supply (SMPS) is an electronic circuit that converts power using switching devices that are turned on and off at high frequencies, and storage components such as inductors or capacitors to supply power when the switching device is in its non-conduction state.



Figure 2.6: SMPS

Switching power supplies have high efficiency and are widely used in a variety of electronic equipment, including computers and other sensitive equipment requiring stable and efficient power supply. A switched-mode power supply is also known as a switch-mode power supply or switching-mode power supply.

2.7 Smoke Sensor (MQ-2)

The MQ-2 is a smoke and combustible gas sensor from Winsen. It can detect flammable gas in a range of 300 - 10000ppm. It's most common use is domestic gas leakage alarms and detectors with a high sensitivity to propane and smoke.



Figure 2.7: Smoke Sensor

The sensing material is SNO_2 for which the conductivity is lower in clean environments and higher when smoke or flammable gas is present. A simple circuit makes measuring this change

in conductivity and turning it into data fairly straight-forward, but does require some calibration.

2.8 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

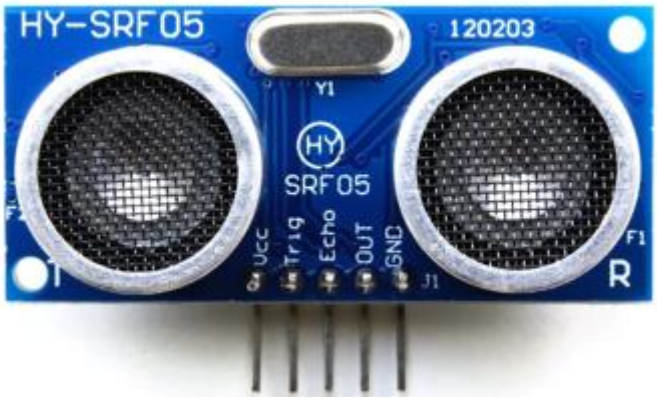


Figure 2.8: Ultrasonic Sensor

Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

2.9 Soil Moisture Sensor

A The Soil moisture sensor unit consists of soil moisture sensor and LM393 comparator chip. The soil moisture sensor is used to detect the soil moisture. The soil probe is dip in to the soil such that when the soil moisture is LOW the module output is HIGH indicated by using RED led on the comparator chip. This sensor measures the dielectric constant of the soil by using transmission line techniques.

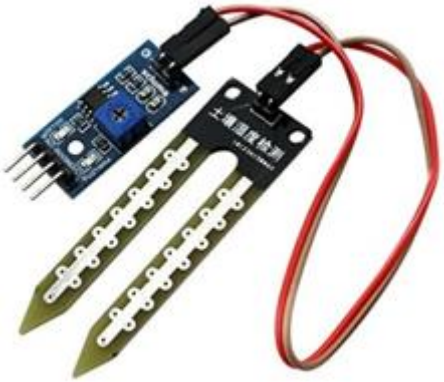


Figure 2.9: Soil Moisture Sensor

This circuit consists of four pins such as power supply pin, ground pin, analog and digital pins. The analog pin A0 connected to the analog pin of ARDUINO board and digital D0 pin connected to the digital pin of ARDUINO board. This sensor is a dual output mode in which analog output is more accurate. The operating voltage is up to 5V.

2.10 LDR

A Light dependent resistor (LDR) is resistor whose resistance decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. We are using LDR as another input to the controller. A photo-resistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

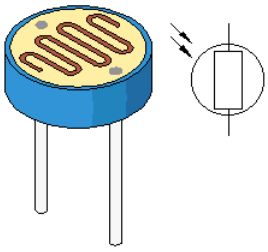


Figure 2.10: Light dependent resistor.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, for example, silicon. In intrinsic

devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire band gap. Extrinsic devices have impurities, also called do pants, and added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (that is, longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

2.11 Buzzer

For alarm purposes a lot of electric bells, alarms and buzzers are available in the market that has got different prices and uses. The buzzer being used in this project is a 5-12 V buzzer and has got enough alarm sound to be used in a fire alarm system. Louder buzzer would have been even better but then their operating voltages are high as we had a supply of maximum up to 12V available with us on the board.



Figure 2.11: Typical buzzer arrangement.

2.12 Temperature Sensor

They are devices to measure temperature readings through electrical signals. The sensor is made up of two metals, which generate electrical voltage or resistance once it notices a change in temperature. The temperature sensor plays a critical role in maintaining a specific temperature within any equipment used to make anything from medicine to beer. To produce these types of content, the accuracy and responsiveness of the temperature and temperature control are critical to ensuring the end product is perfect.



Figure 2.12: Temperature Sensor

Temperature is the most common physical measurement type in industrial applications. Accurate measurements are vital in ensuring the success of these processes. There are many applications that are not-so-obvious, which use temperature sensors. Melting chocolate, using a blast furnace, controlling a hot air balloon, freezing substances in a lab, running a motor vehicle, and firing a kiln.

Temperature sensors come in different forms, which are used for different methods of temperature management. There are two categories of temperature sensors which are contact and non-contact. Contact sensors are used mainly in hazardous areas.

CHAPTER III

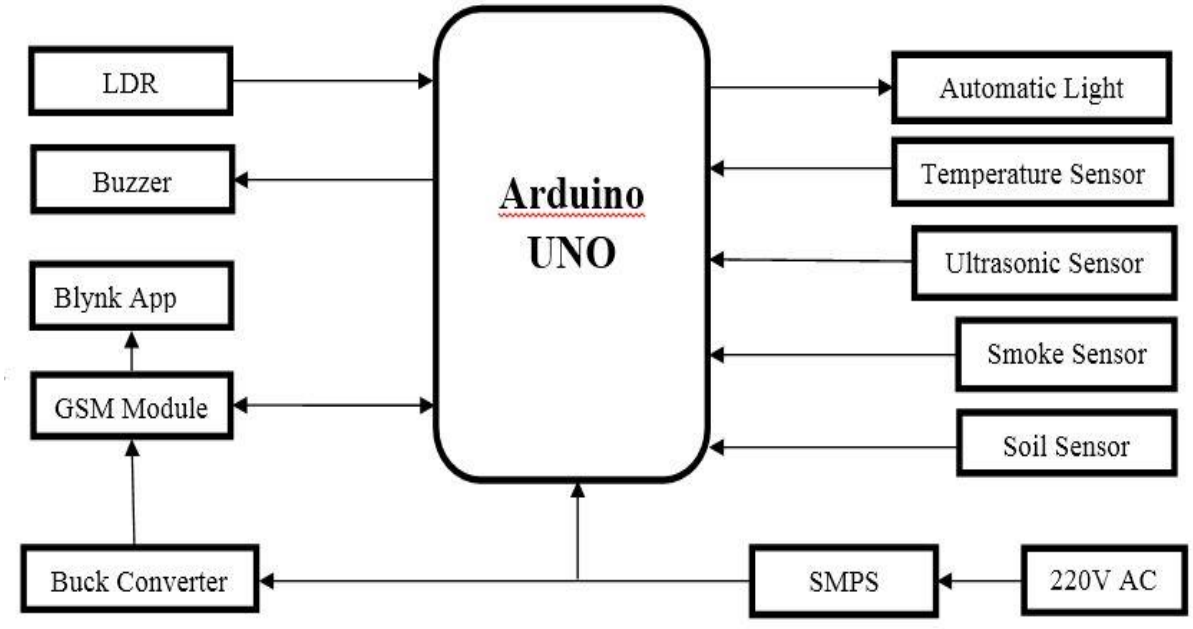
DESING OF THE PROJECT

3.1 Introduction

In this chapter fully cover with discuss design and fabrication of this project. Here we will discuss about developed block diagram and briefly describe about the circuit description and also learn about working principle.

3.2 Block Diagram

In this part of this project we added here Block diagram of IoT based Coal mine safety System. Here we are going to describe about our hardware part by showing part by part in this block diagram. We use here a Arduino Uno to control this total system which is shown here in the middle point of this diagram. In the Right side in bellow added a SMPS, which is included here to produce needed ampere or voltage to operate this all instruments.



Figur 3.2: Block Diagram of this Proposed System

After powering up everything these will be start normally and then every part of this system will play there responsibility as per command. Arduino Uno will control this total system using

the Arduino program which is uploaded to operate this total system properly. In the right side we use here a Temperature sensor, which will sense the temperature of this environment and when it will see the temperature is more than expectations than it will be notify using Buzzer alarm and Blinking Red Light. In the right side we used here a Smoke sensor, which will detect the presence of smoke and Gas in the min location and notify using Buzzer and Blinking Light. Here we use a Soil Sensor, this will detect the value of soil in the coal mine an if the soil level shown high at that moment it will notify us by alarming. Ultrasonic sensor is used here to detect the water level in the coal mine if the level of water are increase more than limit at that moment it will notify us using alarm and blinking Red Light. The brightness of light always be stable in minimum range but when it will detect over brightness at that moment it will notify us as per conditions. GSM module (SIM800L) are used here to connect this total system using operator Network connection.

3.3 Circuit Diagram

In this part of this book we attached here full circuit diagram of this proposed system.

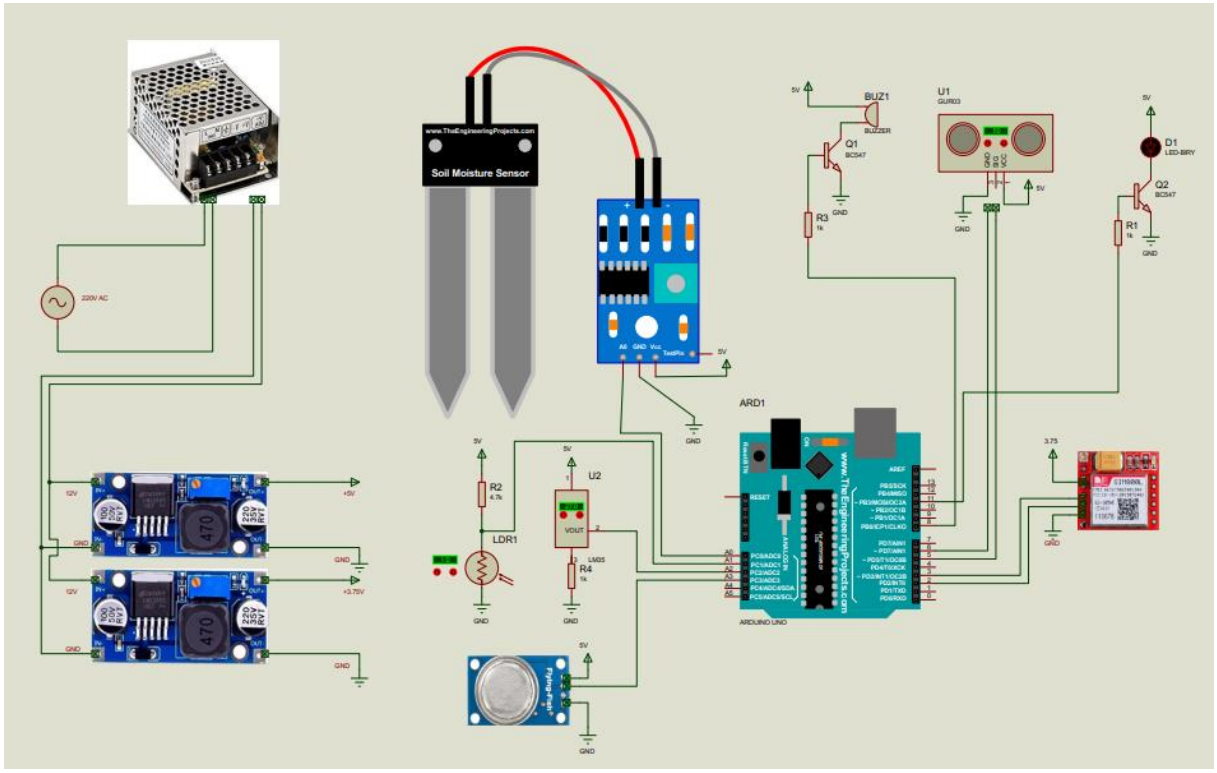


Fig: 3.3: Circuit Diagram of Proposed System

Here we use Porteous simulation software to design this circuit diagram which version is Porteous 8.0. Here we use a Arduino uno to control our total system by using the Arduino

program. We use here a Temperature sensor, which will sense the temperature of this environment and when it will see the temperature is more than expectations than it will be notify using Buzzer alarm and Blinking Red Light. we used here a Smoke sensor, which will detect the presence of smoke and Gas in the min location and notify using Buzzer and Blinking Light. Here we use a Soil Sensor, this will detect the value of soil in the coal mine an if the soil level shown high at that moment it will notify us by alarming. Ultrasonic sensor is used here to detect the water level in the coal mine if the level of water are increase more than limit at that moment it will notify us using alarm and blinking Red Light. The brightness of light always be stable in minimum range but when it will detect over brightness at that moment it will notify us as per conditions. GSM module are used here to connect the system with operator Network without any wi-fi connection or hotspot connection. It will burn money and it will provide the net connection from used operator company network.

3.4 Full Overview

This is the output or full over view of Coal mine safety system.

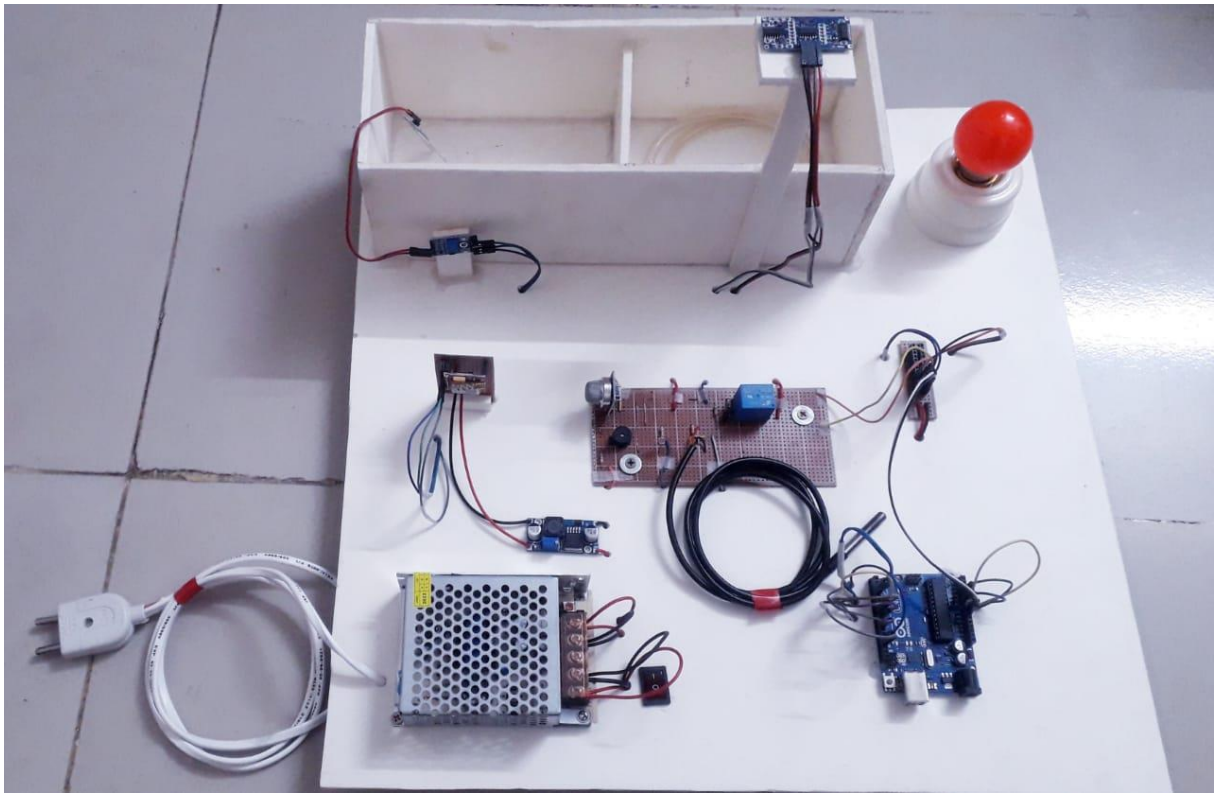


Fig: 3.4: Overview of Proposed system.

In this system you see here that a SMPS are used to make sure the power supply for all components. Here a Arduino Uno is used to operate this total system as per program.

We use here a Red LED to show the abnormal conditions by blinking. One GSM module are used here to connect this total system with IoT using operator network. Temperature sensor, Buzzer, Soil Sensor, Ultraosnic Sensor, Smoke sensor also used here to measure different environmental situations.

3.5 Working Principle

The picture above is the actual demo project we've made. Now let's take a look at the step by step working process.

After powering up everything these will be start normally and then every part of this system will play there responsibility as per command. Arduino Uno will control this total system using the arduino program which is uploaded to operate this total system properly. Temperature sensor will sense the temperature of this environment and when it will see the temperature is more than expectations than it will be notify using Buzzer alarm and Blinking Red Light. Smoke sensor will detect the presence of smoke and Gas in the min location and notify using Buzzer and Blinking Light. Soil Sensor will detect the value of soil in the coal mine an if the soil level shown high at that moment it will notify us by alarming. Ultrasonic sensor is used here to detect the water level in the coal mine if the level of water are increase more than limit at that moment it will notify us using alarm and blinking Red Light. The brightness of light always be stable in minimum range but when it will detect over brightness at that moment it will notify us as per conditions.

3.8 Conclusion

In this chapter we have discussed the block diagram,circuit discription, working principle, flow chart also show the real image of the project and others. In the later part of this paper we discussed the result & discussion of the project to make the concept clear to anyone.

Chapter IV

Software Developement

4.1 Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as **Windows, Mac OS X, and Linux**. It supports the programming languages C and C++. Here, IDE stands for **Integrated Development Environment**.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

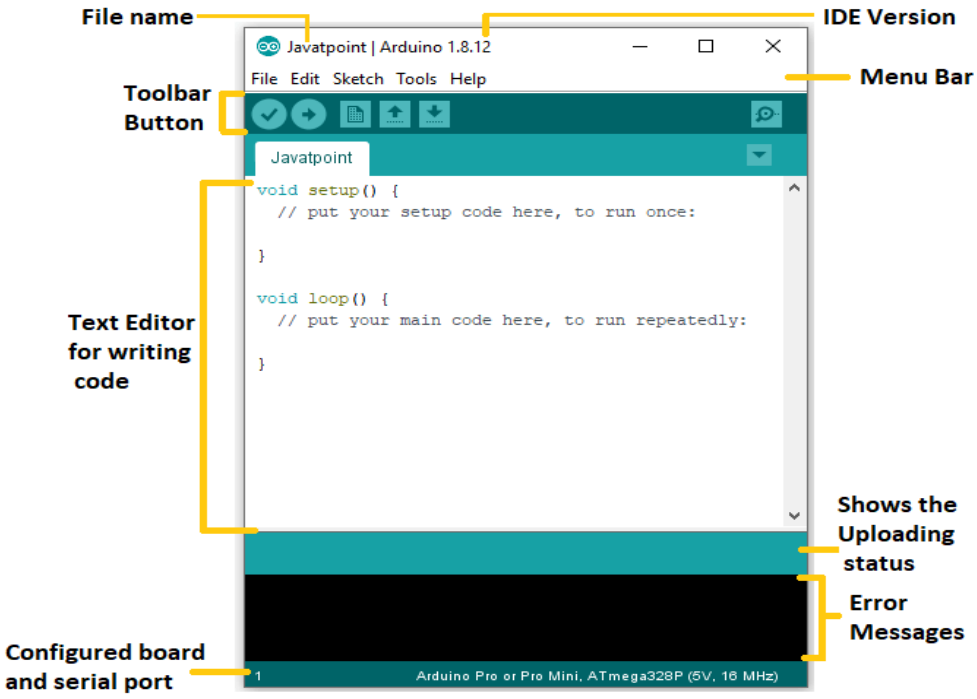


Figure 4.1: Java Point

4.1.1 Toolbar Button

The icons displayed on the toolbar are **New, Open, Save, Upload,** and **Verify**.

It is shown below:

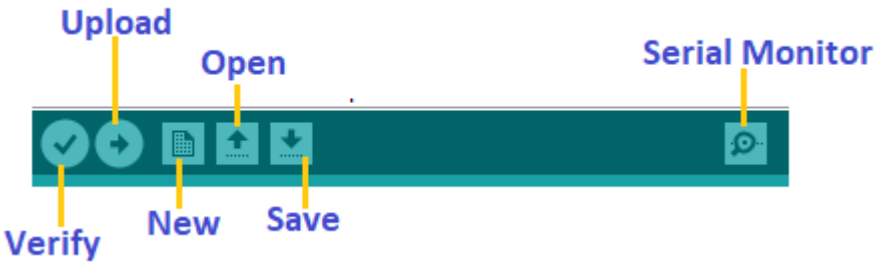


Figure 4.1.1: Toolbar Button

4.1.2 Upload

The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected.

We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

The latest Arduino boards can be reset automatically before beginning with Upload. In the older boards, we need to press the Reset button present on it. As soon as the uploading is done successfully, we can notice the blink of the Tx and Rx LED.

If the uploading is failed, it will display the message in the error window.

We do not require any additional hardware to upload our sketch using the Arduino Boot loader. A Boot loader is defined as a small program, which is loaded in the microcontroller present on the board. The LED will blink on PIN 13.

4.1.3 Open

The Open button is used to open the already created file. The selected file will be opened in the current window.

Save

The save button is used to save the current sketch or code.

4.1.4 New

It is used to create a new sketch or opens a new window.

4.1.5 Verify

The Verify button is used to check the compilation error of the sketch or the written code.

4.1.6 Serial Monitor

The serial monitor button is present on the right corner of the toolbar. It opens the serial monitor. When we connect the serial monitor, the board will reset on the operating system Windows, Linux, and Mac OS X. If we want to process the control characters in our sketch, we need to use an external terminal program. The terminal program should be connected to the COM port, which will be assigned when we connect the board to the computer.

4.1.7 Menu Bar

- File

When we click on the File button on the Menu bar, a drop-down list will appear. It is shown below:

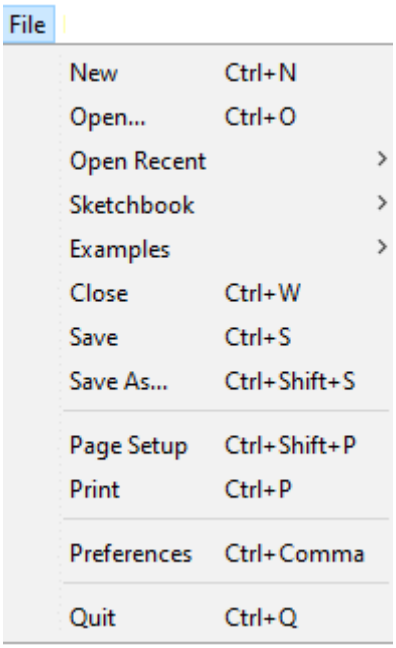


Figure 4.1.7: File

Let's discuss each option in detail.

New

The New button opens the new window. It does not remove the sketch which is already present.

Open

It allows opening the sketch, which can be browsed from the folders and computer drivers.

Open Recent

The Open Recent button contains the list of the recent sketches.

Sketchbook

It stores the current sketches created in the Arduino IDE software. It opens the selected sketch or code in a new editor at an instance.

Examples

It shows the different examples of small projects for a better understanding of the IDE and the board. The IDE provides examples of self-practice.

Close

The Close button closes the window from which the button is clicked.

Save

The save button is used to save the current sketch. It also saves the changes made to the current sketch. If we have not specified the name of the file, it will open the 'Save As...' window.

Save As...

We can save the sketch with a different name using the 'Save As...' button. We can also change the name accordingly.

Page Setup

It allows setting the page margins, orientation, and size for printing. The 'Page Setup' window will appear as:

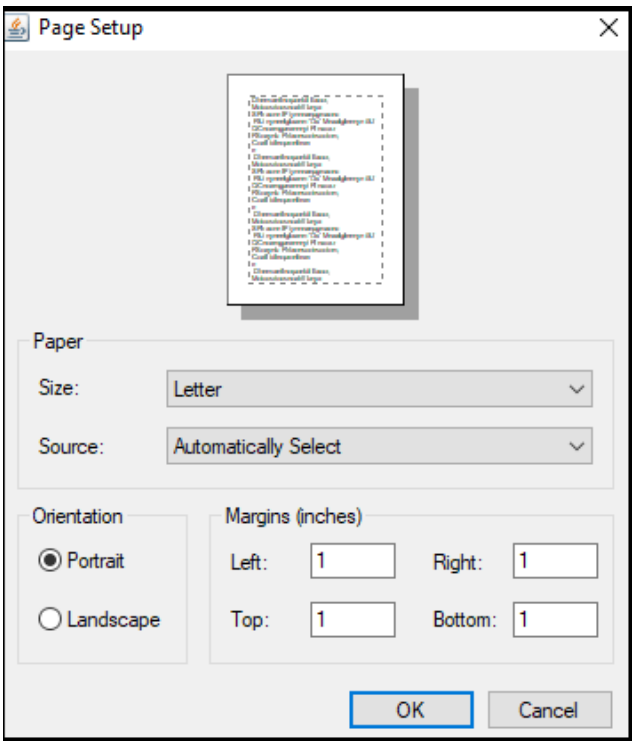


Figure 4.4: Page Setup

Print

According to the settings specified in the ‘Page Setup’, it prepares the current sketch for printing.

Preferences

It allows the customization settings of the Arduino IDE.

Quit

The Quit button is used to close all the IDE windows. The same closed sketch will be reopened when we will open the Arduino IDE.

- Edit

When we click on the Edit button on the Menu bar, a drop-down list appears. It is shown below:

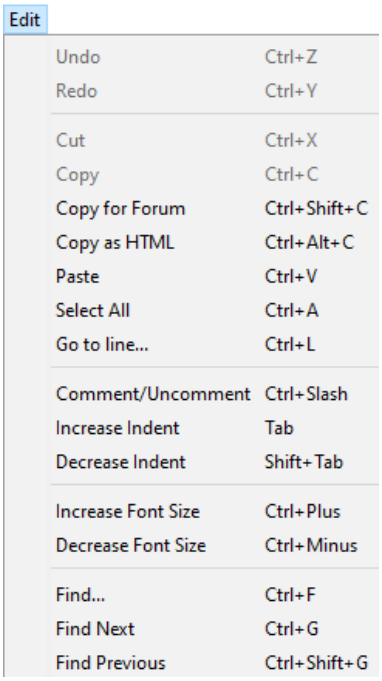


Figure 4.5: Edit

Let's discuss each option in detail.

Undo

The Undo button is used to reverse the last modification done to the sketch while editing.

Redo

The Redo button is used to repeat the last modification done to the sketch while editing.

Cut

It allows us to remove the selected text from the written code. The text is further placed to the clipboard. We can also paste that text anywhere in our sketch.

Copy

It creates a duplicate copy of the selected text. The text is further placed on the clipboard.

Upload

The Upload button is used to configure the code to the specified board through the port.

Upload Using Programmer

It is used to override the Boot loader that is present on the board. We can utilize the full capacity of the Flash memory using the 'Upload Using Programmer' option. To implement this, we need to restore the Boot loader using the Tools-> Burn Boot loader option to upload it to the USB serial port.

Export compiled Binary

It allows saving a .hex file and can be kept archived. Using other tools, .hex file can also be sent to the board.

Show Sketch Folder

It opens the folder of the current code written or sketch.

Include Library

Include Library includes various Arduino libraries. The libraries are inserted into our code at the beginning of the code starting with the #. We can also import the libraries from .zip file.

Add File...

The Add File... button is used to add the created file in a new tab on the existing file.

For example, let's add 'Blink' file to the 'Java point' file. The tab will now appear as:

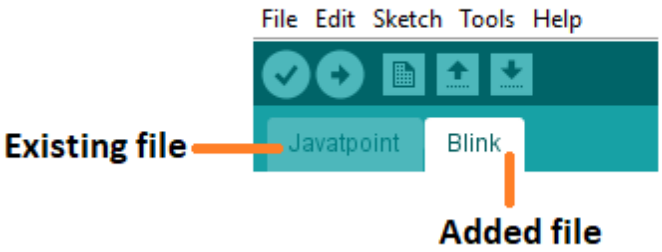


Figure 4.6: Added file

We can also delete the corresponding file from the tab by clicking on the small triangle -> Delete option.

4.1.8 Tools

When we click on the Tools button on the Menu bar, a drop-down list appears. It is shown below:

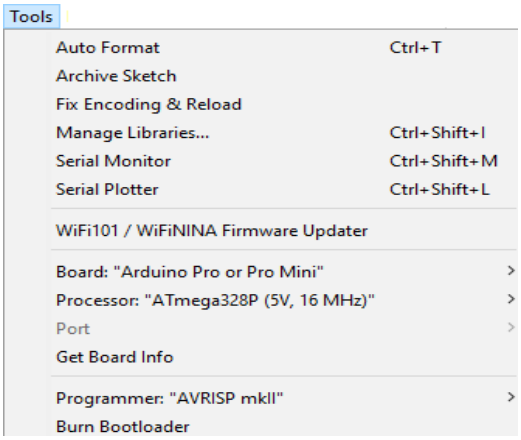


Figure 4.7: Tools

Let's discuss each option in detail.

Auto Format

The Auto Format button is used to format the written code. For example, lining the open and closed curly brackets in the code.

Archive Sketch

The copy of the current sketch or code is archived in the .zip format. The directory of the archived is same as the sketch.

Fix Encoding and Reload

This button is used to fix the inconsistency between the operating system char maps and editor char map encoding.

Manage Libraries…

It shows the updated list of all the installed libraries. We can also use this option to install a new library into the Arduino IDE.

Serial Monitor

It allows the exchange of data with the connected board on the port.

Serial Plotter

The Serial Plotter button is used to display the serial data in a plot. It comes preinstalled in the Arduino IDE.

WiFi101/WiFi NINA Firmware Updater

It is used to check and update the Wi-Fi Firmware of the connected board.

Board

We are required to select the board from the list of boards. The selected board must be similar to the board connected to the computer.

Processor

It displays the processor according to the selected board. It refreshes every time during the selection of the board.

Port

It consists of the virtual and real serial devices present on our machine.

Get Board Info

It gives the information about the selected board. We need to select the appropriate port before getting information about the board.

Programmer

We need to select the hardware programmer while programming the board. It is required when we are not using the onboard USB serial connection. It is also required during the burning of the Boot loader.

Burn Boot loader

The Boot loader is present on the board onto the microcontroller. The option is useful when we have purchased the microcontroller without the boot loader. Before burning the boot loader, we need to make sure about the correct selected board and port.

4.1.9 Help

When we click on the Help button on the Menu bar, a drop-down list will appear. It is shown below:

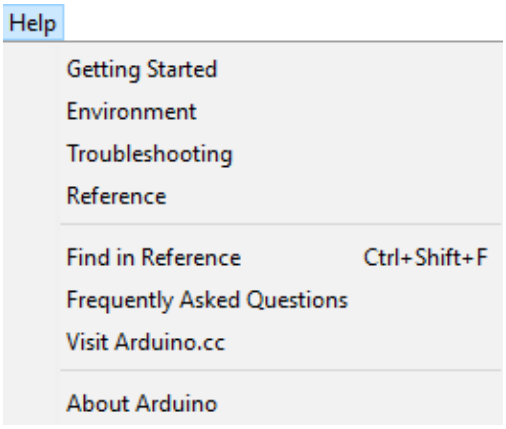


Figure 4.8: Help

The Help section includes several documents that are easy to access, which comes along with the Arduino IDE. It consists of the number of options such as Getting Started, Environment, Troubleshooting, Reference, etc. We can also consider the image shown above, which includes all the options under the Help section. Some documents like Getting started, Reference, etc.,

can be accessed without the internet connection as well. It will directly link us to the official website of Arduino.

4.2 Proteus 8 Professional

This is a brief introduction to Proteus application. The special features and advantages of this application will be immediately below. Hopefully this article will give you the most extensive and useful view on how to use Proteus 8 Professional.

Want to check the circuit diagram? Seeing circuit layout seems like a good choice but it is not easy to do it for large circuit boards. Or you can try creating a circuit diagram (PCB) printout and testing but it takes a lot of time-consuming operations. It seems, using Proteus 8 Professional is the most ideal. Proteus 8 Professional is a software used to plot, sketch PCB and simulate diagrams.

This software was developed by **Lab center Electronic Ltd**.

4.2.1 Features of Proteus 8 Professional

Very easy to draw diagrams on Proteus 8 Professional. Click "select device" and select the desired detail, you can draw the wire by clicking on the output of electricity or Vcc, Ground, etc.

Is the circuit completed? The simulation feature of Proteus 8 Professional can display most details in the circuit. There are two options for this feature: run the simulation and run each table. "Running simulation" shows circuit board at normal speed (if the circuit is not too complicated). "Running each table" will run each circuit board at every click. This customization is useful for fixing digital circuits. You can simulate microcontrollers. Depending on each microcontroller will be outlined based on PIC24, dsPIC33, 8051, Arduino, or ARM7. You can download the compiler and dump hex file into the microcontroller contained in Proteus. Moreover, real-time integration with simulation using switches, resistors, photoelectric resistors, etc.

Is an easy-to-use feature in Proteus. You can create your own designs or make Proteus work for you. Creating your own design is easy, you just need to put the details into the diagram and draw the electrical circuit through.

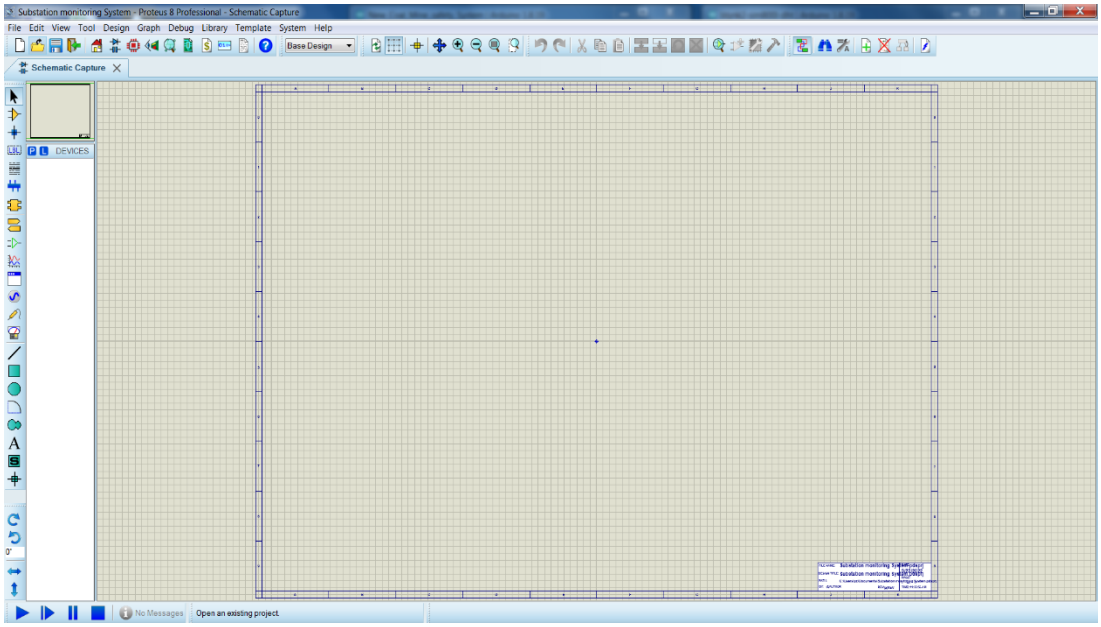


Figure 4.9: Schematic Capture

Don't worry about breaking any design rules because it will automatically detect errors. If you want Proteus to do it for you, just put the details in the corresponding position and run it automatically. It will draw the way to set the circuit and choose the best version. And now there's another custom "Auto placer", which requires you to set the board size by drawing the shape and size of the board. After that, it automatically puts the details into the mold. After that, all you have to do is make a circuit diagram.

4.3 Blynk App

With Blynk, you can create smartphone applications that allow you to easily interact with microcontrollers or even full computers such as the Raspberry Pi. The main focus of the Blynk platform is to make it super-easy to develop the mobile phone application. As you will see in this course, developing a mobile app that can talk to your Arduino is as easy as dragging a

widget and configuring a pin. With Blynk, you can control an LED or a motor from your mobile phone with literally zero programming. This is actually the first experiment that I will demonstrate in this course. But don't let this simplicity make you think that Blynk is only useful for trivial applications. Blynk is a robust and scalable tool that is used by hobbyists and the industry alike.

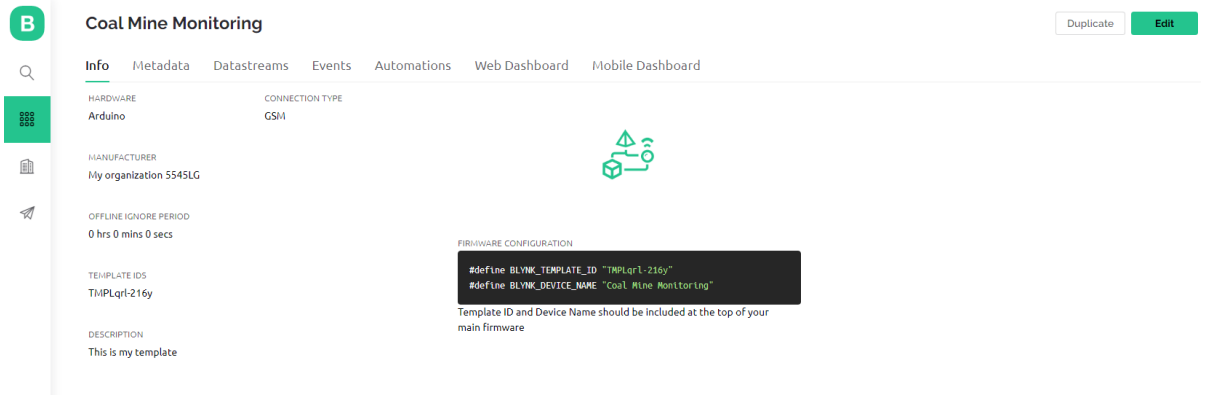


Figure 4.10: Info

You can use it to monitor the soil humidity of your vegetable garden and turn on the water, or open up your garage door, with your phone. You can also use it to control smart furniture that can learn from your routines, or embed IoT and AI to traditional industrial products such as a boiler, or for improving the integrity and safety of oilfields. Blynk is free to use for personal use and prototyping. Their business model generates profits by selling subscriptions to businesses that want to publish Blynk-powered apps for their hardware products or services.

The Blynk innovation is the smartphone app. The Blynk app is is really an app editor.
It allows you to create one or more projects. Each project can contain graphical widgets, like virtual LEDs, buttons, value displays and even a text terminal, and can interact with one or more devices.

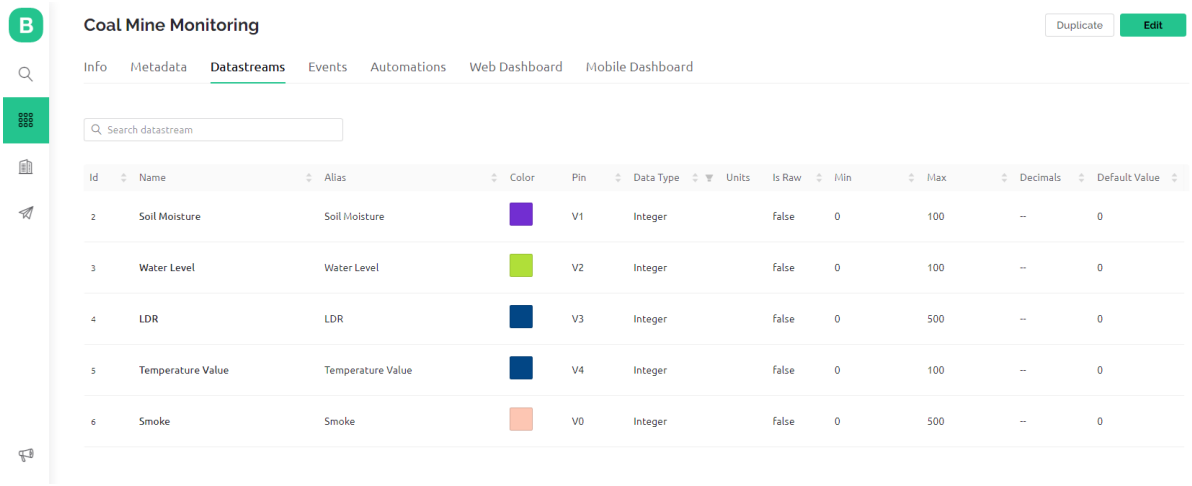


Figure 4.11: DataStream

With the help of the Blynk library, it is possible to control Arduino or ESP32 pins directly from your phone, without having to write any code at all. It is also possible to share a project with friends and even customers so that they can access the connected devices but not be able to modify the project. Imagine a scenario where you build a smartphone application where you can control lights, window blinds and room temperature from your phone. You can share the project with other family members so that they can also access the functionality.

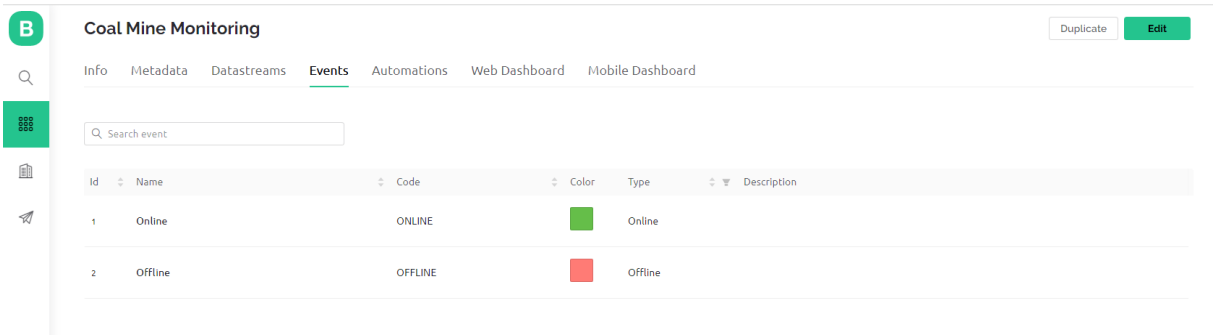


Figure 4.12: Events

I am amazed by the range of devices and connectivity types that the Blynk platform supports. The support is implemented by means of a Blynk library that targets a device and connectivity type combination. For example, if you want to use your Arduino Uno with an Ethernet shield, you would use the library “BlynkSimpleEthernet” which contains the Blynk firmware plus the required connectivity support.

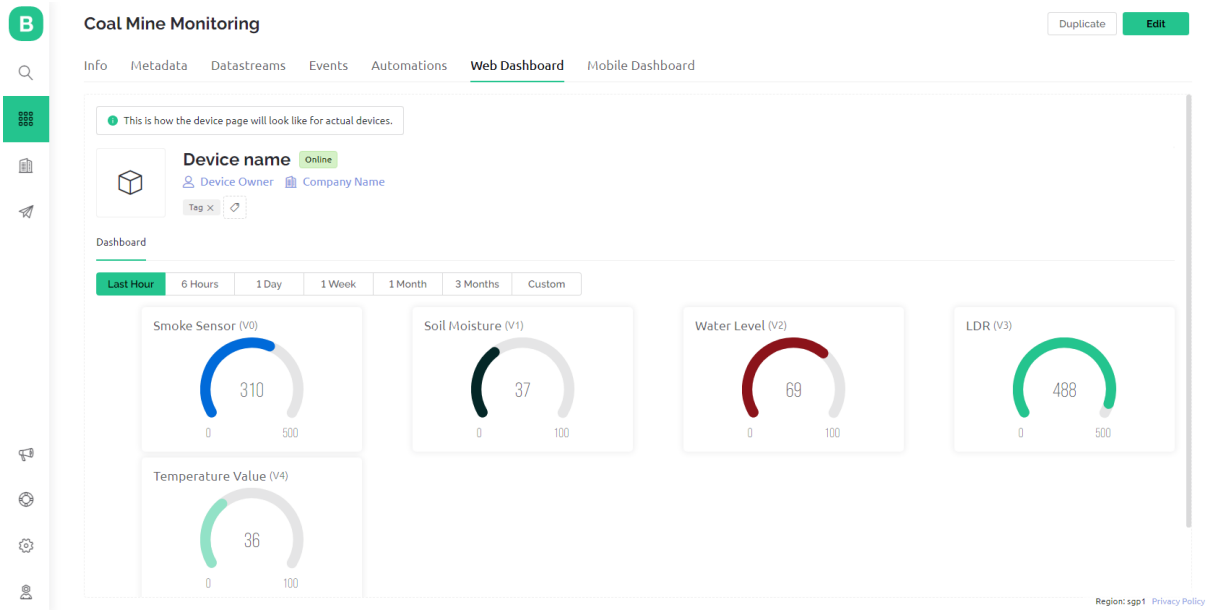


Figure 4.13: Web Dashboard

Both sketches would use the same Blynk infrastructure, such as the physical pins that are part of the Arduino's hardware, as well as the virtual pins which are implemented in software by the Blynk Platform. This means that with a bit of planning, you can write sketches that can be easily shared among different target devices. You can potentially write a sketch for an Arduino Uno and and with minimal modification run it on an Arduino MKR1010. Blynk also supports clients that are not a microcontroller. You can write client code is JavaScript, Python or Lua thanks to the available Blynk libraries for these languages. This means that you can have a Blynk project that interacts with an Arduino and with Python code running on a Linux virtual machine somewhere on the Cloud. The possibilities are really very exciting.

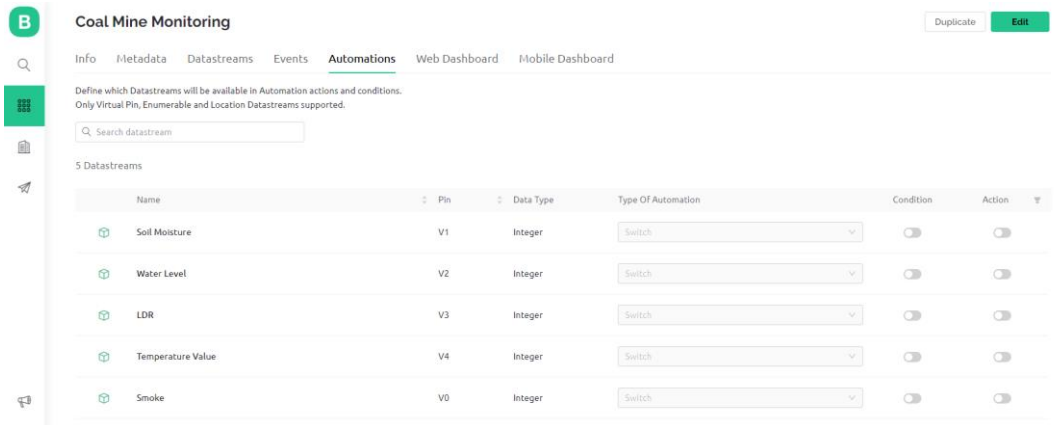


Figure 4.14: Automations

Unlike IoT platform such as IFTTT, Twillio, and even Adafruit IO, you can host a private instance of the full Blynk server and connect your smartphone Blynk app to it. The Blynk Cloud server is an excellent choice for most projects, as it is always there, ready to use. We will use the Cloud server in the first few experiments in this course to help you get started with minimal effort. However, as you will see, the Cloud Blynk server has imposed limitations. Some limitations are due to the topology of the server: depending on your geographical location, the server may be in a different continent, which makes communications between the app, the devices and the server slow due to the amount of time it takes packets to travel across the Internet.



Figure 4.15: Dashboard

Another imposed limitation is that in the Cloud server, you can only use a small number of widgets. Blynk is using the concept of "energy" to implement a pricing system for its widgets. In the Cloud server you may start a new project with 1000 energy units.

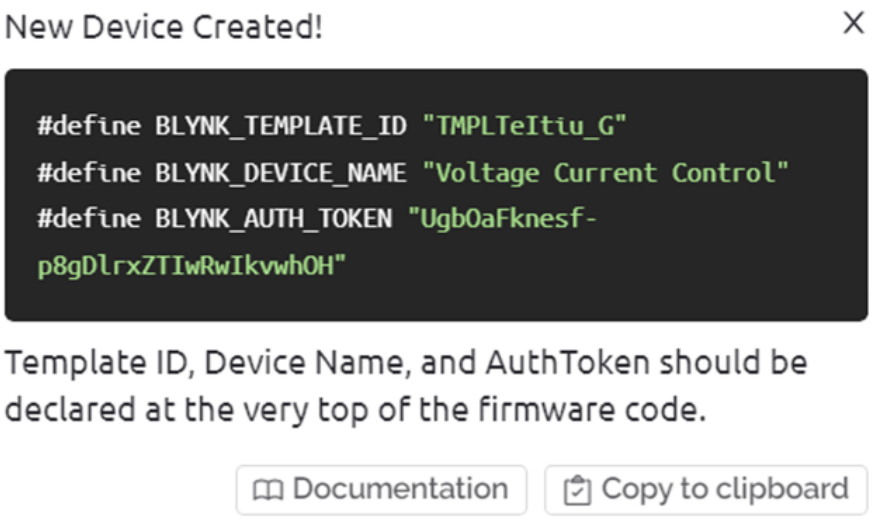


Figure 4.16: New Device Created

An LED widget may cost you 200 units, leaving 800 units for other widgets. On a private server, you can set your own energy limits. You can configure your server to allocate 100,000 energy units to new users. It's totally up to you.
Of course, you can purchase additional energy units to spent on the Cloud Blynk server, and this is a legitimate consideration especially if the users of your Blynk project are distributed around the world (therefore latency is not a big issue).

However, a private Blynk server gives you additional benefits:

- Essentially unlimited energy units, so that you can build any Blynk application you can think of.
- Minimal latency, which is useful when your application is used in a limited geographic area and responsiveness is important.
- Total control of your data. You can keep your own backups of your Private server, migrate your server to a new host, implement whichever security mechanisms you wish, and finely control your users.

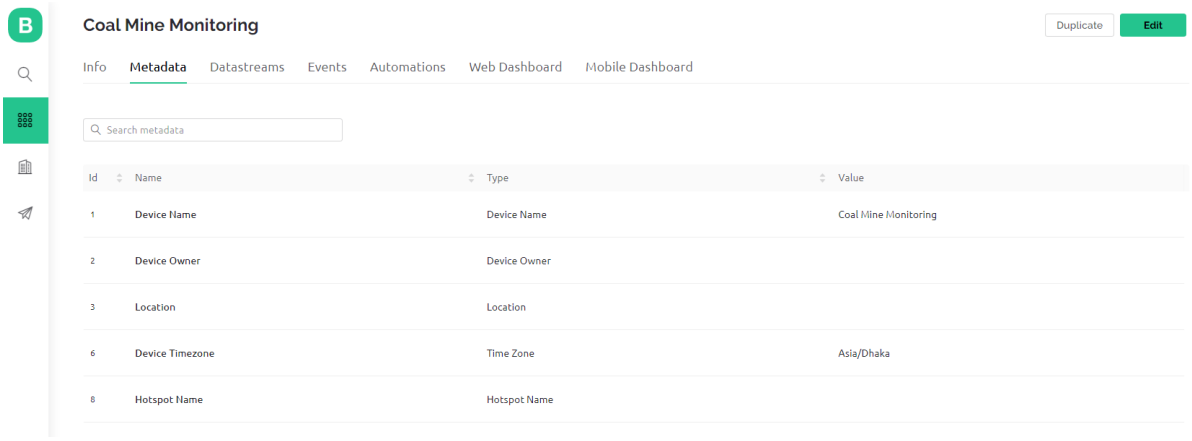


Figure 4.17: Dashboard

In this course, I will show you how to install and configure a private Blynk server so you can see the related benefits and costs. You can run an instance of the Blynk server on any host that has a Java runtime environment. In this course, I use a $15 Raspberry Pi Zero W, running the Raspberry Pi operating system.

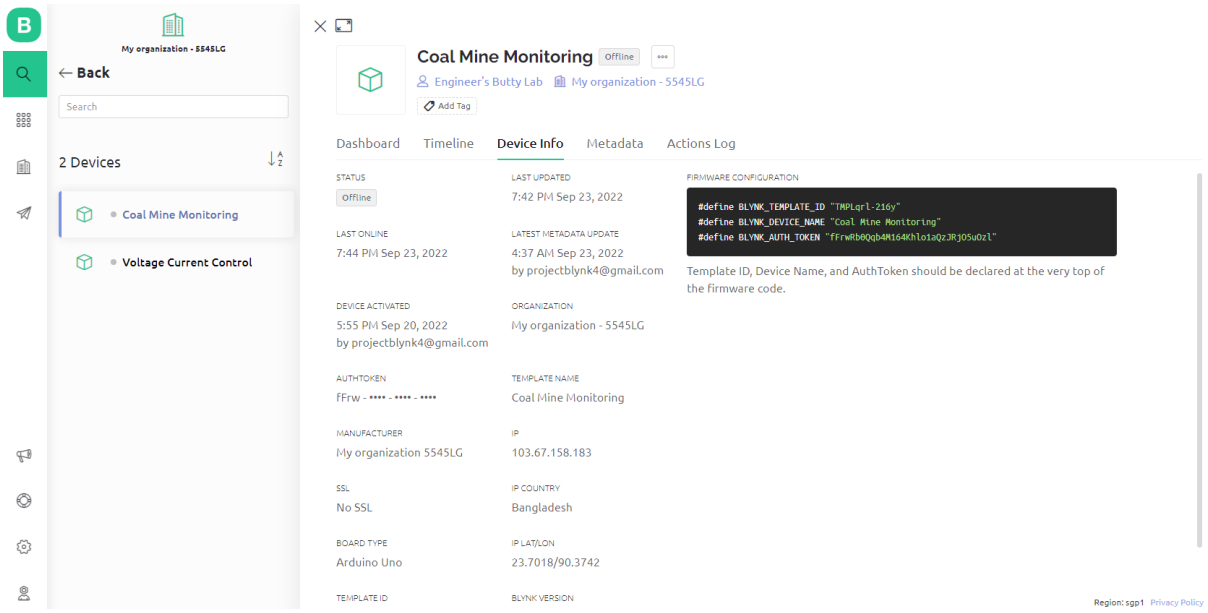


Figure 4.18: Device Info

This is a computer with just half a gigabyte of RAM, and still runs the Blynk server without any issues at all. You can use any Raspberry Pi with WiFi, it does not need to be a Zero W. In fact, you can use any Linux or Windows host you want, including virtual machines.

Chapter V

RESULT & DISCUSSION

5.1 Introduction

This chapter contains the results obtained and discussion about the project. We have also covered discussions about advantages, disadvantages and limitation of current version of the protection system.

5.2 Result

We show here the exact value for every fault level which is used in our program to monitoring our Coal Mine safety system.

Serial	Smoke	LDR	Soil	Water Level	Temperature	Statuse
1	Smoke>150	LDR>150	Soil>70	Water <60	Temp <50	Smoke Fault
2	Smoke<150	LDR<150	Soil>70	Water <60	Temp <50	Light Fault
3	Smoke<150	LDR>150	Soil<70	Water <60	Temp <50	Soil Fault
4	Smoke<150	LDR>150	Soil>70	Water >60	Temp <50	Water Fault
5	Smoke<150	LDR>150	Soil>70	Water <60	Temp >50	Temp Fault
6	Smoke<150	LDR>150	Soil>70	Water <60	Temp <50	Normal

Figure 5.1: Result Data

5.3 Advantags

This system successfully automates the current coal mine safety system and replaces wired communication.

It will greatly improve the performance and efficiency of data transmission of the coal mine safety system and reduce the costs of extending the system.

The application of IoT can realize the real-time monitoring of working system. Existing networks are slow and heavily rely on manual labour. Use of IoT can enhance the whole performance of the existing networks.

The loss of human life can be avoided significantly as this system assesses the situation inside the mine without the need of a human inside the mine.

- We can monitor all the data from anywhere of world.
- It’s possible to avoid any unexpected danger by using this monitoring system.
- Anyone can see the report by using his smartphone.

5.4 Discussion

The Internet of Things (IoT) is just another term for devices that connect to the internet and talk to each other. There are several different types of IoT applications on a wide scale. Cluster on the internet of Things classifies IoT technologies as some of the most critical areas for building thoughtful facilities, the advancement of smart transport, the advance of sensible power, smart industry, smart health, and smart environments. The IoT may be a technology that serves as a trendsetter that collects and stores all device information in the cloud wherever it is accessible via the internet. This technology also uses sensors and actuators for gathering information and distributing it across the network. Additionally, we use cloud to store data, but we also use it to conduct data analysis, collect data, and visually present data. So, if increasing technologies such as associate degree in applied science rises is often utilized to expand existing systems and make them more effective in numerous IoT applications, for example, agriculture, health, sensible home, etc., In order to say something positive about coal, you should say that it is a non-renewable energy source, and as a result, it can never be commonly replaced by humans. As a result, incidents of coal mine explosions occur, and miners put their lives at risk every day, since they are often employed in the coal mines. A significant number of mishaps, both small and large, occur due to the recent instrumentation and wired equipment, which then leads to the destruction, mishandling, spillage of the deteriorating gases in the coal mines, thereby resulting in considerable hazards for the excavators. lightweight is a critical consideration in the particular duties of an underground coal miner, since they are unable to proceed into the mine if there is no legal illumination that is capable of hindering their vision due to the heavy workload underground. Since it was obvious that the coal mine protection system must avoid this issue, a design for the system was developed. we are predisposed to solving the problems that we identify in our research by testing of the information gathered by the sensors, we are able to make use of and finish the study victimization the Thinker method. Automatic or manual control may be used to regulate how much to use. It is the Arduino microcontroller that we utilized in our project.

Chapter VI

CONCLUSION

6.1 Conclusion

This paper mainly focuses on the system successfully automates the current coal mine safety system and replaces wired communication. With this implementation of wireless communication between mines and ground level, better security and working environments are made possible for miners. This system allows better chances of preventing disasters from occurring and for quicker response by the authorities in case of disaster. The components used in this system are really compact and cost a small amount in terms of size as well as money. This system can help in saving lives of workers by giving handy information, beforehand. With the installation of a real-time observation device, a clearer and additional objective is given for assessing mine point perspective, which leads to greater accuracy. In this case, this technique will display the parameters on the monitoring screen. Everybody who is currently working in the mine, as well as any worker, will profit from this proposal., who will be able to use it to avoid losing their lives in a work-related accident. When sensing element values have crossed the alarm threshold, the alarm goes off.

6.2 Future Work

At present we have make this coal mine safety system using blynk app only but there have no Phone alarming or vibration or auto calling system are adder here. In future we will add more and more extra facility than any responsible persone will know about any abnormal condition without seeing the screen value.

References

[1] S.Wei, L. Li-li, ”Multi-parameter Monitoring system for Coal Mine based on Wireless Sensor Network Technology”, Proc. International IEEE conference on Industrial Mechatronics and Automation, pp 225-227, 2009.

[2] E.K Stanek, “Mine Electro technology Research: The Past 17 Years”, IEEE transactions on industry applications, vol. 24[5], pp 818-19, 1988.

[3] S.Wei, L. Li-li, ”Multi-parameter Monitoring system for Coal Mine based on Wireless Sensor Network Technology”, Proc. International IEEE conference on Industrial Mechatronics and Automation, pp 225-227, 2009.

[4] S.Jin-Jing, G. Heng-wei, S.Yu-jun, “Research on Transceiver System of WSN Based on V-MIMO Underground Coal Mines”, Proc. International Conference on Communications and Mobile Computing, pp 374-378, 2010.

[5] Dong “Coal Mine Safety Monitoring System Based on Zigbee and GPRS”, Applied Mechanics and Material Volume 422, 2013.

[6] LiHui “Design of Monitoring system for Coal mine safety based on Wireless sensor Networks”, 2008 International Conference on Mechatronic and Embedded systems and Applications(ASME).

[7] Gautam Gowrishankankaran and Charles He, “ Productivity, safety and regulation in undergroundcoalmining:Evidencefromdiastersandfatalities,” Arizoneducation,March2017.

[8] Yongping Wu and Guo Feng, “The study on coal mine monitoring using the Bluetooth wirelesstransmission system” , 2014 IEEE Workshop on Electronics, Computer and Applications, pp. 1016-1018, 2014.

[9] XiaolongFeng,JianshengQian,ZhenzhenSun,XingWang,"WirelessMobileMonitoringSystem for Tram Rail Transport in Underground Coal Mine Based on WMN," cason, pp.452-455,2010International Conference on ComputationalAspectsofSocial Networks,2010.

[10] Yi-ming Tian,You-ruiHuang,Yi-qingHuang,"IntelligentInformationProcessingofWSNBased on Vague Sets Theory and Applied in Control of Coal Mine Monitoring,"cccm, vol. 2, pp.649-652,2008ISECSInternationalColloquiumonComputing,Communication,Control,andManagement,2008.

[11] Jingjiang Song ,Yingli Zhu and Fuzhou DongK, “automatic monitoring system for coal minesafety based on wireless sensor network”, IEEE Radio Science and Wireless Technology Conference,pp.933-936, 2011.

[12] Yogendra S Dohare and Tanmoy Maity, “surveillance and safety system for underground coalminesbasedon LowPowerWSN”,IEEE,pp.116-119, 2014.

[13] ValdoHenriquesandRezaMalekian,“Minesafetysystemusingwirelesssensornetwork”,IEEE,pp. 1-12,2016.

[14] PranjalHazarika,“implementationofsafetyhelmetforcoalmineworkers”,1stIEEEInternationalConferenceonPowerElect ronicsIntelligentControlandEnergy Systems,pp.1-3, 2016.

[15] TanmoyMaityandParthaSarathi,“AwirelesssurveillanceandsafetysystemformineworkersbasedonZigbee”,1stInt’lConf. onRecentAdvancesinInformationTechnologyRAIT-2012

Appendix

Code:

```
// Created By: Mehedi Hasan
//Organization: Engineers Butty Lab BD
//  Email: engineersbuttylab@gmail.com
//Contact: +88 01888011500


#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID "TMPLqrl-216y"
#define BLYNK_DEVICE_NAME "Coal Mine Monitoring"
#define BLYNK_AUTH_TOKEN "fFrwRb0Qqb4M164Khlo1aQzJRjO5uOzl"

#define TINY_GSM_MODEM_SIM800

#include <TinyGsmClient.h>
#include <BlynkSimpleTinyGSM.h>
#define trig 7   // Trig pin
#define echo 8
char auth[] = BLYNK_AUTH_TOKEN;

char apn[]  = "internet";
char user[] = "";
char pass[] = "";

int value = 0;
int depth =20;

#include <SoftwareSerial.h>
SoftwareSerial SerialAT(2, 3); // RX, TX
TinyGsm modem(SerialAT);
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
#include <DHT.h>
#define DHT11_PIN 4
#define DHTTYPE DHT11
DHT dht(DHT11_PIN, DHTTYPE);

#define ONE_WIRE_BUS 10 // DS18B20 on arduino pin2 corresponds to D4 on physical board "D4 pin on the ndoemcu Module"
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
float temp;
#define sensor A0
#define buzzer 6
#define relay 13



BlynkTimer timer;
#define INTERVAL 1000L



//////////////////////////////// SMOKE ////////////////////////////////

void SendSmokeData()
{
int sensorValue = analogRead(A1);

  Blynk.virtualWrite(V0,sensorValue);
  Serial.println(sensorValue);

  if(sensorValue>150)
  {

    digitalWrite (6,HIGH);
```

```
  digitalWrite (13,HIGH);

   delay(3000);
 }
 else
 {

    digitalWrite (6,LOW);
  digitalWrite (13,LOW);

   delay(1000);
 }

}


//////////////////////////////////// LDR ////////////////////////////////////
void SendLdrData()
{
  int LDR = analogRead(sensor);
 Blynk.virtualWrite(V3,LDR);
 Serial.println(LDR);

 if(LDR<150)
 {
  digitalWrite (6,HIGH);
  digitalWrite (13,HIGH);
   Blynk.notify ("LIGHT ON");

   delay(3000);
 }
 else
 {
   digitalWrite (6,LOW);
  digitalWrite (13,LOW);
```

```
    delay(1000);
  }
}

//////////////////////////////////// Soil Moisture ////////////////////////////////////


void moisture()
{
  int value = analogRead(A2);
  value = map(value, 0, 1023,0, 100);
  Blynk.virtualWrite(V1,value);
  Serial.println(value);


   if (value < 70)


  {

  digitalWrite (6,HIGH);
   digitalWrite (13,HIGH);
    Blynk.notify ("SOIL LEVEL HIGH");
    delay(3000);
 }

 if (value > 70)


  {


  digitalWrite (6,LOW);
   digitalWrite (13,LOW);
    delay(1000);
```

```
 }

}
```

```
 ////////////////////////////////////////// Water Level //////////////////////////////////////

void waterlevel()
{
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long t = pulseIn(echo, HIGH);
  long cm = t / 29 / 2;
  Serial.println(cm);
  long level= depth-cm;
  if (level<0)
  level=0;
  level = map(level,0,depth-3,0,100);
  Blynk.virtualWrite(V2, level);
  delay(1000);


 if (level < 60)


  {

    digitalWrite (6,LOW);
   digitalWrite (13,LOW);
```

```
 delay(1000);


 }



  if (level > 60)

 {


 digitalWrite (6,HIGH);
  digitalWrite (13,HIGH);
   Blynk.notify ("WATER LEVEL HIGH");
 delay(3000);


 }

}

////////////////////////////////////////// Temp //////////////////////////////////////

void getSendData()
{
 DS18B20.requestTemperatures();
 temp = DS18B20.getTempCByIndex(0); // Celcius
   Blynk.virtualWrite(V4, temp);

   delay (1000);
 if (temp>50)
{
 digitalWrite (6,HIGH);
```

```
   digitalWrite (13,HIGH);
    Blynk.notify ("TEMPERATURE HIGH");
  delay(3000);

}

//////////////////.....//////////////////
}



void SendDhtData()
{

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print("Temperature: ");
  Serial.println(t);
  Serial.print("Humidity: ");
  Serial.println(h);
  Serial.println("----------------------------------------");

  Blynk.virtualWrite(V5, t);
  Blynk.virtualWrite(V6, h);

}
```

```
/*********************************************************************************
***
 *  setup() function

**********************************************************************************
/
void setup()
{

  Serial.begin(115200);
  delay(10);

  SerialAT.begin(9600);

  delay(3000);
  DS18B20.begin();
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(6,OUTPUT);
  pinMode(13,OUTPUT);

  dht.begin();

  Serial.println("Initializing modem...");
  modem.restart();

  Blynk.begin(auth, modem, apn, user, pass);

  timer.setInterval(INTERVAL, SendDhtData);
    timer.setInterval(INTERVAL, waterlevel);
  timer.setInterval(INTERVAL, moisture);
   timer.setInterval(INTERVAL, getSendData);
   timer.setInterval(INTERVAL, SendLdrData);
```

```
 timer.setInterval(INTERVAL, SendSmokeData);


  //---------------------------------------------------------
}


/*********************************************************************************
***
 *  loop() function

*********************************************************************************
/
void loop()
{
  Blynk.run();
  timer.run(
```