

LIGHT FIDELITY(Li-Fi) BASED INDOOR COMMUNICATIONSYSTEM

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

By

GULLANKI MADHURI (Reg. No. 37130149)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**SCHOOL OF ELECTRICAL AND ELECTRONICS
ENGINEERING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **GULLANKI MADHURI** (Reg.No.37130149) who carried out the project entitled "**LIGHT FIDELITY(Li-Fi) BASED INDOOR COMMUNICATION SYSTEM**" under my supervision from December 2020 to April 2021.

Internal Guide

Ms. R. Sakthi Prabha, M.E., (Ph.D.)

Head of the Department

Dr. T. RAVI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

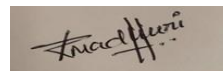
I GULLANKI MADHURI (Reg. No.37130149) hereby declare that the Project Report entitled “LIGHT FIDELITY(Li-Fi) BASED INDOOR COMMUNICATION SYSTEM” done by me under the guidance of Ms. R. Sakthi Prabha, M.E., (Ph.D.), is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in ELECTRONICS AND COMMUNICATION ENGINEERING.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

1.

A rectangular box containing a handwritten signature in black ink. The signature appears to be 'Gullanki Madhuri' written in a cursive style.

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N.M. NANDITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to my Project Guide **Ms. R. Sakthi Prabha, M.E., (PhD.)**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Light fidelity (Li-Fi) technology is a wireless communication system which uses visible light spectrum for communication and to transmit data with high speeds in a secure manner compared to all existing traditional methods like (Wireless Fidelity) Architecture. In this project we employ Li-Fi technology in data acquisition of sensor data from Building Management systems. Building Management system consists of many sensors which can give a lag in data in turn a slower analysis when health's of the building or its related equipment's are measured. By employing this simple method in devising the fast response to alerts and alarms using this Li-Fi IoT combined devices we can eliminate more error losses. A simple sensor integrated circuit is designed in this project to acquire data and transform into user readable data with the help of high-speed Li-Fi communication. After the initial stages practical examinations and evaluations will be done to check the performance of the circuit and give out the best Key performance index of this project. As per the suggested plan by increasing the communication speed between sensor connected devices and the building equipment's we can observe a drastic change in the various test environments and in real world cases.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	1-9
2	LITERATURE SURVEY	7
3	REQUIREMENTS	10-11
	3.1 EXISITING SYSTEM	12
	3.1.1 DRAWBACKS OF THE EXISTING SYSTEM	
	3.2 PROPOSED SYSTEM	
	3.2.1 ADVANTAGES OF PROPOSED SYSTEMS	13
	3.2 BLOCK DIAGRAM	13
	3.4 ARDUINO DESCRIPTION	14
	3.4.1 DIGITAL PIN	15
	3.4.2 ANALOG PIN	16
	3.4.3 OUTPUT SIGNALS	17
	3.4.4 INPUT SIGNALS	18
	3.4.5 SOFTWARE TIPS	22
	3.5 Li-Fi (LIGHT FEDILITY TECHNOLOGY)	23
	3.5.1 HISTORY OF Li-Fi TECHNOLOGY	24
	3.6 SENSORS DESCRIPTION	26
	3.6.1 PIR SENSOR	27
	3.6.2 MQ 135 SENSOR	28
	3.6.3 MQ 4 SENSOR	29
	3.6.4 MQ 7 SENSOR	30
	3.6.5 DTH 11 SENSOR	31
	3.6.6 ESP 32	32
	3.6.6.1 FEATURES OF THE ESP32	32
	3.6.7 Li-Fi TRANSMITTER AND RECIVER	36
	3.6.8 DISPLAY MODULE	37

4.1 ARDUINO IDE SOFTWARE DESCRIPTION

4.1.1 WHY ARDUINO?	39
4.1.2 ARDUINO SOFTWARE	40
4.1.3 WRITING SKETCHES	40
4.1.4 FILE	42
4.1.5 SKETCHES	43
4.1.6 TOOLS	44
4.1.7 HELP	45
4.1.8 SKETCHBOARD	46
4.1.9 TABS, MUTIPLE FILES AND COMPLATION	46
4.1.10 UPLOADING	46
4.1.11 LIBRARIES	47
4.1.12 THIRD-PARTY HARDWARE	47
4.1.13 SERIAL MONITOR	48
4.1.14 PREFERENCES	48
4.1.15 LANGUAGE SUPPORT	49

4.2 NODE RED

4.2.1 NODE RED TOLL	54
4.2.2 BROWSER- BASED FLOW EDITION	54
4.2.3 BUILT ON NODE.JS	55
4.2.4 SOCIAL DEVELOPMENT	56
4.2.5 GET STARTED	56
4.2.6 PREREQUISITES	56
4.2.7 INSTALLING WITH DOCKER	57
4.2.8 INSTALLING WITH SNAP	57
4.2.9 RUNNING	57
4.2.10 OVERRIDE INDIVIDUAL SETTTING	59
4.2.11 UPGRADING NODE-RED	60
4.2.12 OVERVIEW	61
4.2.13 SUMMARY	61-62

5	STIMULATION RESULT	63-66
6	CONCLUSION AND FUTURE SCOPE	67-68
	REFERENCES	69-70

LIST OF FIGURES

FIGURENO:	FIGURE NAME	PAGE NO:
3.3	BLOCK DIAGRAM	13
3.4	ARDUINO UNO	14
3.4.4	PIN CONFIGURATION OF ARDUINO UNO	19
3.5	BASIC DIAGRAM OF Li-Fi	23
3.5.2	WORKING OF Li-Fi	25
3.6.1	PIR SENSOR	27
3.6.2	MQ 135 SENSOR	28
3.6.3	MQ 4 SENSOR	29
3.6.4	MQ 7 SENSOR	30
3.6.5	DTH 11 SENSOR	31
3.6.6	ESP 32	32
3.6.7	Li-Fi RECEIVER AND TRANSMITTER	36
3.6.8	ILI 9341 DISPLAY MODULE	37
4.2	THE MAIN TYPES OF NODES: INPUT, OUTPUT AND PROCESSING	52
4.2.1	NODE RED APPEARANCE	
4.2.2	BROWSER- BASED FLOW EDITING	54
4.2.3	BUILT ON NODE.JS	55
5.1	PIR VALUE OBTAINED	62
5.2	HYDROGEN OBTAINED	63
5.3	CARBON MONOXIDE OBTAINED	63
5.4	TEMPERATURE OBTAINED	64
5.5	SMOKE OBTAINED	65

CHAPTER 1

INTRODUCTION

In this project Li-Fi technology in data acquisition of sensor data in building management system. By employing this simple method in BMS systems we can increase the speed of communication between the sensors and node, paves the way to higher speeds in data information exchange and wireless communication using light. The concept behind this technology is that the data can be transmitted with the help of light emitting diode (LED) bulbs and transmission rate can be control by using intensity of LED bulb which can be varies even faster than light intensity human eye can observe. Keeping in mind number of users increase day by day and heavy traffic of data, Li-Fi technology can be used as a solution to provide users an environment of high-speed data transmission.

According to, Li-Fi and Wi-Fi has the difference related to the congestion, density, security, safety, and speed. The more Wi-Fi enabled device is exist, the congestion may occur. In the technology of Wi-Fi, we can't add more routers if the user is increased, while we can add the light in Li-Fi. Efficiency and safety of the internet are the dominating issues right now. The performance of Li-Fi is claimed that more better than the performance of Wi-Fi. The rate speed of Li-Fi is 1000 times faster than Wi-Fi. For safety of the internet, Li-Fi is more secure than the Wi-Fi based on the spread of the signal. Li-Fi has a light characteristic that light cannot go through the wall. It is different from the signal of Wi-Fi can go through anywhere. Based on those two technologies, in a simple conclusion is Li-Fi has more secure communication rather than the Wi-Fi. The vulnerability exists if there are leakage in the wall while having an indoor communication. Security threat that may exist is an intruder can spoof the data usage.

Basic concept for working principle in Li-Fi Technology are pointing into: Transceiver and Light as a media transmission. This basic concept indicates as a duplex communication. The rates of Li-Fi are 14Gbps using three off- the- shelf laser diodes (red, green, and blue) and predict the rate until 100Gbps when the whole visible spectrum is used. According to [3] Li-Fi and VLC used a similar medium as a data communication that is light. While the Li-Fi technology is fully networked, bidirectional, and high-speed

wireless communication. Others said Li-Fi is the incorporation of Wi-Fi. Li-Fi enabling the system using fully networked wireless communication and could provide a connection that's 100 times faster than Wi-Fi. It can reach speeds up to 3Gbps by using DMT modulation. In Li-Fi, the transmit speed is up to 10Gbps. The rate speed of Li-Fi can be higher than 3Gbps while the technology is on research and developing. The rate speed of Li-Fi is based on the using of the modulation. For security, Li-Fi is more thought than a Wi-Fi. It is because the coverage area of Li-Fi is only on their illuminate area. The signal of Li-Fi cannot go through the wall. While the signal of Wi-Fi can go through the wall, it can cause the vulnerabilities in data loss and data leakage.

Building Management System

The building management system (BMS) is an overarching control system that is responsible for the automatic regulation and control of non-GMP facility subsystems, maintaining predefined parameters (or set points) and the control of their functionality. The major aim of the BMS is to guarantee the safety of facility operation, while also monitoring and optimizing the use and efficiency of its supervised subsystems to allow more efficient operation. Examples of the major subsystems controlled by the BMS are

A Building Management System (BMS) is a crucial component to any data centre facility. It allows for centralized management of critical infrastructure equipment, and ensures operations are being run securely and efficiently. The Building Management System monitors each area in the following ways:

- **Power** distribution must be consistent and clean once it reaches the computer equipment. The system helps ensure there are no power surges or inconsistent “spikes” of power that could damage the equipment. It also provides alerts to system managers if there are any power failures, surges or other issues with the power system.
- **Temperature** control is vital for any facility. The humidity needs to remain at 45 percent, because too much can damage the equipment and cause it to fail. The temperature needs to be consistent between 70 and 74 degrees, and can be adjusted or have circulation added as necessary by the BMS.
- **Physical security** guarantees only the appropriate and approved individuals enter the colocation centre at all times. This is managed on-site through key card readers, biometric scanners and tailgating sensors to gain access and move throughout the facility. The Building Management System can be set to sound an alarm and notify operators if there is a physical breach of the system or entrance at a time when no one should be in the building.

- The **fire suppression** component uses a Very Early Smoke Detection Apparatus (VESDA) tied to the Building Management System to alert operators or sound an alarm if there is a potential fire.
- In addition, the BMS also collects and stores historical data that can be used for trend analysis and examination. The data can help operators predict when there might be an issue or when maintenance needs to occur within a certain area of the BMS. The information comes through in real time, so engineers can immediately resolve an issue.
- A Building Management System can help operators monitor the facility from any location, on any device. This allows the data center manager to be proactive if there's ever an issue with part of the system instead of taking a reactive approach.
- If there is ever an alarm for a system failure or issue, a facility engineer is notified and can immediately go to the site and resolve any possible problem. Some issues, such as temperature control, can be fixed through automation, so the engineer or manager does not need to return to or be at the site.
- **Semiconductor-based sensors:** Placed on an integrated circuit, these identical diodes use temperature sensitive voltage compared with current conditions, allowing them to record changes in temperature.
- **Thermocouple:** As the name suggests, this consists of two wires – these are made from different metals and placed at different points, with the change in voltage between the two points showing change in temperature.
- **Resistance Temperature Detector:** A film or wire is wrapped around a ceramic or glass core, with temperature measured from the resistance between the element with temperature. These tend to be the most accurate type of sensor, but can also be the most expensive.

Negative Temperature Coefficient Thermistor: Providing high resistance at low temperatures, as temperature increases resistance quickly drops – reflecting changes quickly and accurately.

Since early 2014, the mobile has overtaken the personal computer (desktop/laptop) as the leading device used to navigate the Net. Along with the mobile, a number of other portable devices that connect to the Internet have also started proliferating at a very quick rate. Nowadays, most of us carry or possess at least one Internet-based device and mobile. So, the Internet of Things ([IoT](#)) now doesn't only mean different 'things', but has evolved into 'intelligent things' which have on board computation and network connections. Most importantly, they have the capability to sense the environment around us and, accordingly,

act intelligently. These devices are now referred to as connected devices, smart objects or the Web of Things.

This adaptation of technology has led to many developments in the visual programming and visual coding arenas. As a result, it has evolved into a visual programming/coding tool based on the already popular Node.js (a server side Java Scripting platform).

CHAPTER 2

LITERATURE REVIEW

Goswami, P. and Shukla, M.K. In this paper, a hybrid network combining light fidelity (Li-Fi) with a radio frequency (RF) wireless fidelity (Wi-Fi) network is considered. An additional tier of very small Li-Fi attocells which utilize the visible light spectrum, offers a significant increase in the wireless data throughput in an indoor environment, while at the same time providing room] illumination. Importantly, there is no interference between Li-Fi and Wi-Fi. A Li-Fi attocell covers a significantly smaller area than a Wi-Fi access point (AP). This means that even with a moderate user movement a large number of handover between Li-Fi attocells can occur, and this compromises the system throughput. Dynamic load balancing (LB) can mitigate this issue so that the quasi-static users are served by Li-Fi attocells, while moving users are served by a Wi-Fi AP. However, due to the user movement, local overload situations may occur which prevent handover, leading to a lower throughput. This research studies LB in a hybrid Li-Fi/Wi-Fi network by taking into account user mobility and handover signaling overheads. Furthermore, a dynamic LB scheme is proposed, where the utility function considers system throughput and fairness. In order to better understand the handover effect on the LB, the service areas of different APs are studied, and the throughput of each AP by employing the proposed LB scheme is analyzed.

Haas H, Yin L, Wang Y, and Chen C Li-Fi is networked, bidirectional wireless communication with light. It is used to connect fixed and mobile devices at very high data rates by harnessing the visible light and infrared spectrum. Combined, these spectral resources are 2600 times larger than the entire radio frequency (RF) spectrum. This paper provides the motivation behind why Li-Fi is a very timely technology, especially for 6th generation (6G) cellular communications. It discusses and reviews essential networking technologies, such as interference mitigation and hybrid Li-Fi/Wi-Fi networking topologies. We also consider the seamless integration of Li-Fi into existing wireless networks to form heterogeneous networks across the optical and RF domains and discuss implications and solutions in terms of load balancing. Finally, we provide the results of a real- world hybrid Li-Fi/Wi-Fi network deployment in a software defined networking testbed. In addition, results from a Li-Fi deployment in a school classroom are provided, which show that Wi-Fi

network performance can be improved significantly by offloading traffic to the Li-Fi.

Toshihiko mine, Student Member, Smart phones, tablets, and the rise of the Internet of Things are driving an insatiable demand for wireless capacity. This demand requires networking and Internet infrastructures to evolve to meet the needs of current and future multimedia applications. Wireless Het Nets will play an important role toward the goal of using a diverse spectrum to provide high quality-of- service, especially in indoor environments where most data are consumed. An additional tier in the wireless Het Nets concept is envisioned using indoor gigabit small-cells to offer additional wireless capacity where it is needed the most. The use of light as a new mobile access medium is considered promising. In this article, we describe the general characteristics of Wi-Fi and VLC (or Li-Fi) and demonstrate a practical framework for both technologies to coexist. We explore the existing research activity in this area and articulate current and future research challenges based on our experience in building a proof-of-concept prototype VLC Het Net.

Jovicic, A., Li, J. and Richardson, T. White LED offers advantageous properties such as high brightness, reliability, lower power consumption and long lifetime. White LEDs are expected to serve in the next generation of lamps. An indoor visible-light communication system utilizing white LED lights has been proposed from our laboratory. In the proposed system, these devices are used not only for illuminating rooms but also for an optical wireless communication system. Generally, plural lights are installed in our room. So, their optical path difference must be considered. In this paper, we discuss about the influence of interference and reflection. Based on numerical analyses, we show that the system is expected to be the indoor communication of the next generation.

Kahn, J.M. and Barry, J.R. The use of infrared radiation as a medium for high-speed short-range wireless digital communication is discussed. Available infrared links and local-area networks are described. Advantages and drawbacks of the infrared medium are compared to those of radio and microwave media. The physical characteristics of infrared channels using intensity modulation with direct detection (IM/DD) are presented including path losses and multipath responses. Natural and artificial ambient infrared noise sources are characterized. Strategies for designs of transmitter and receivers that maximize link signal-to-noise ratio (SNR) are described. Several modification formats are discussed in detail, including on-off keying (OOK) pulse-

position modulation (PPM), and subcarrier modulation. The performance of these techniques in the presence of multipath distortion is quantified. Techniques for multiplexing the transmissions of different users are reviewed. The performance of an experimental 50-Mb/s on- off-keyed diffuse infrared link is described.

Yunluo Wang Li-Fi R&D The VLN comm LumiNex has an Ethernet port available through which LED panels can provide wireless internet access as it emits optical signals through visible light. This Li-Fi enabled LED panel provides secure and private connection with a coverage area of 37 m² (400 ft²). The Li-Fi system uses a hybrid visible light/infrared system for downlink that can still provide Li-Fi connectivity even when the lights are off. Devices can connect to the Li-Fi network using the LumiStick USB dongle.

Harald Haas Li-Fi R&D In order to minimize infection risks for patients with artificial hearts, wireless data transmission methods with electromagnetic or optical transcutaneous data transmission have been developed. However, using these methods, it tends to become difficult to transmit data if the external data transceiver moves from its proper position because the communication distance is short. To resolve this problem, the purpose of this study is to develop a bidirectional data communication system using ultra high frequency radio wave for implantable artificial hearts, and confirm its performance. Due to its long communication distance, high-speed communication rate, low power consumption, high tolerance to electromagnetic disturbances, and secure wireless communication, we adopted Bluetooth radio wave technology for our system. The system consists of an internal unit and an external unit, each having a Bluetooth module (radio field intensity: 4 dBm, receiver sensitivity: -80 dBm). In addition, the internal unit has an 8-bit D/A converter and a micro controller with an 8-channel 10-bit A/D converter, while the external unit also has a RS-232C converter. We experimented with the internal unit implanted into pig meat (tissue thickness: 5 mm), carrying out a data transmission test and a control test in order to evaluate the performance of this system. As a result, data transfer speeds of about 20 kbps were achieved within the communication distance of 8 m. Moreover, the speed control of an artificial heart was achieved at a distance of 1 m. In conclusion, we confirmed that the system can wirelessly transmit data from the inside of a body to the outside while controlling an artificial heart, thereby preventing unstable data transmission due to accidental movements of the external data transceiver unit. It will contribute to effective medical treatment by continually monitoring the conditions of both the patient and the artificial heart and adequately controlling it.

Dushyant A. Basnayake Light Emitting Diodes are set to penetrate many areas of everyday life. An interesting property of these devices in addition to their lightening capabilities, is that they can also be utilized for data transmissions as well. LED's require low power for their operation and have very high switching speeds. This dual-purpose application can reduce our dependency on the densely used means of communication the Radio frequency communication. The project aims to build a system which carries out communication using visible light that are employed for indoor illuminations. The higher bandwidth of the visible light is exploited for high-speed data transmissions. For the communication to be full duplex it requires a downlink and an uplink. The downlink is implemented using the visible light. Whereas the uplink is provided using Infrared LED's, as visible light would serve no purpose of illumination. Furthermore, the IR link also eliminates the chance of interference. When communication is carried out through visible light there are problems associated with it. The data transmission rates should be high enough so that flickering of the LED's should not be detected. Sometimes it is possible for complete fadeout of the signals. To protect the data integrity the data should be framed, so as to detect lost signals and to ensure correct transmission and reception of the data. Computer network protocols like stop and wait algorithms [1] are employed to solve this problem. The data can be of any form such as text, an audio, image or the video. This project shows how visible light can be used to transfer text data from one computer to the other computer. The computers are used as the end device. The computers with the help of Python [2] can be used to interface software to the hardware.

Hossein Kazmi, Majid Safari Radio-frequency (RF) communication saw a progression of innovation throughout the 20th century. In recent years, it has been transformed profoundly by technological advances, both in the capabilities of individual radios and in the design of networks and other systems of radios. This discussion presents some highlights of recent advances and their implications for the design of radios and radio systems and for regulation and policy. It does not aim to describe the full range of technical challenges associated with wireless communications; the interested reader is referred to the 1997 NRC report *The Evolution of Untethered Communications*,¹ which describes many of the fundamental challenges

associated with wireless communications or, for a more recent view of the technology and its applications, several recent textbooks on wireless communications.

Xiping Wu, Majid Safari For being highly energy efficient and inexpensive, the use of current commercial LEDs for transmitting data is considered as a significant milestone in the area of optical wireless communications. Light-fidelity (Li-Fi) and optical camera communications (OCC) are two latest technologies which use LED as transmitter, with PD and camera as receivers, respectively. Currently, Li-Fi is emerged as highly promising as it supports high data rate and security. But it suffers from some limitations, e.g., short communication distance, low signal-to-interference-plus-noise ratio (SINR) etc. OCC can overcome these limitations, but it only supports lower data rate. Therefore., to provide proper solution, we propose a hybrid OCC/Li-Fi system. We are not going to use both the technologies at the same time, rather access point is selected based on fuzzy logic, considering different application scenarios. Furthermore, at the end of this paper, the performance of the selection mechanism is discussed briefly.

CHAPTER 3

REQUIREMENTS

3.1 EXISITING SYSTEM

In this Light-fidelity the Indoor is very Hybrid Wi- Fi in particularly Systems and Omnidirectional communication using Li-Fi Networks of Navigation system using Li-Fi in a subtle way.

3.1.1 DRAWBACKS OF THE EXISTING SYSTEM

Interference of kind of other light sources in the receiver terminals range limitations in basically omnidirectional communication using Li-Fi for all intents and purposes High power supply rates which limits the usage of battery and battery time, which particularly is fairly significant.

3.2 PROPOSED SYSTEM

To specifically Convert the Building Management System's Sensor data into light and transmitting over a fairly few distance using Li-Fi technology Data Acquisition from the sensor networks and display in a dashboard for sort of High-level user experience Encryption of sensor data while transmission using perfect coding methods and achieving the output at a desired time without any hinderance, contrary to popular believe. The Li-Fi system consists of mainly two parts, the transmitter and the receiver. LED light is used as signal source between two end systems. The transmitter part modulates the input signal with the required time period and transmits the data in the form of 1's and 0's using a LED bulb.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

In this Light- fidelity its higher data speeds achieved during transmission encryption of data for a secured line of communication actually ease in data acquisition of sensors for hassles dashboard user experience, or so they literally thought and li-fi applications literally are varied as a result of its pretty key features, pretty such as directional lighting, energy efficiency, intrinsic security, pretty high data rate capability, definitely signal blocking by walls and integrated networking capability security Dense urban environments Cellular communication EMI definitely sensitive environments Augmented reality Localized

advertising particularly Underwater communication Safety environments Intelligent transportation systems Connectivity definitely Sensitive data and as well as Indoor navigate, which specifically is quite significant. Speed-The speed of the Li-Fi is very high, and we can watch the videos without buffering. security- The light of the Li-Fi doesn't run through the partition, therefore, it is more protected and hacking is not possible, risk-free-Li-Fi utilizes light waves which are harm less, consistent- The data transfer is more protected.

3.3 BLOCK DIAGRAM

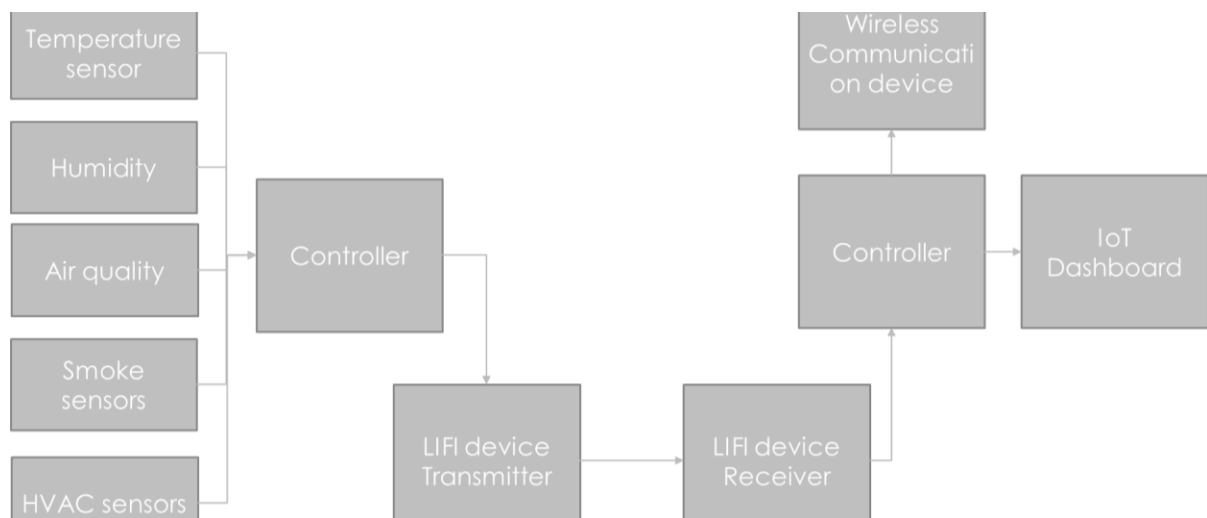


Fig:3.3 BLOCK DIAGRAM

The above figure 3.3 shows the block diagram of proposed methodology. The system consists of temperature sensor, humidity, air quality, smoke sensors, HAVC sensors, the sensors are connected to the controller where a power supply of 5v range (the volt supply varies from 12v but we consider 5v) is connected to it. The data from all the sensors are collected and transmitted to the receiver and it is displayed on the LCD screen. This whole process is done by the Li-Fi connection where the data is transmitted in the form of light and the solar plate is arranged to get the light impulse. At the receiver side we have amplifier so that we can avoid the unwanted signals.

3.4 ARDUINO IDE DESCRIPTION

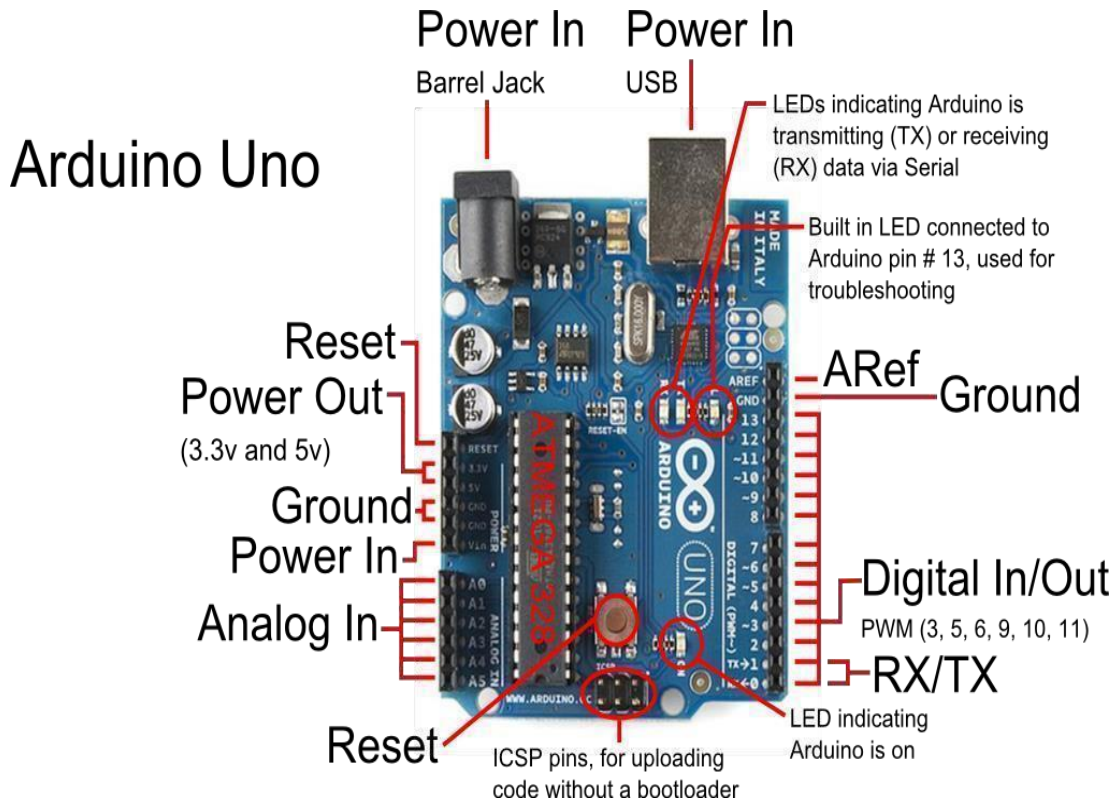


Fig:3.4 Arduino Uno

To use the voltage divider as a sensor reading device first you need to know the maximum voltage allowed by the analog inputs you are using to read the signal. On an Arduino this is 5V. So, already we know the maximum value we need for V_{out} . The V_{in} is simply the amount of voltage already present on the circuit before it reaches the first resistor. You should be able to find the maximum voltage your sensor outputs by looking on the Datasheet, this is the maximum amount of voltage your sensor will let through given the voltage in of your circuit. Now we have exactly one variable left, the value of the second resistor. Solve for R_2 and you will have all the components of your voltage divider figured out! We solve for R_1 's highest value because a smaller resistor will simply give us a smaller signal which will be readable by our analog inputs.

Powering an analog Reference is exactly the same as reading a sensor except you have to calculate for the Voltage Out value you want to use as the analog Reference.

All of the electrical signals that the Arduino works with are either Analog or Digital. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

3.4.1 DIGITAL PIN

An electronic signal transmitted as binary code that can be either the presence or absence of current, high and low voltages or short pulses at a particular frequency.

Humans perceive the world in analog, but robots, computers and circuits use Digital. A digital signal is a signal that has only two states. These states can vary depending on the signal, but simply defined the states are ON or OFF, never inbetween.

In the world of Arduino, Digital signals are used for everything with the exception of Analog Input. Depending on the voltage of the Arduino the ON or HIGH of the Digital signal will be equal to the system voltage, while the OFF or LOW signal will always equal 0V. This is a fancy way of saying that on a 5V Arduino the HIGH signals will be a little under 5V and on a 3.3V Arduino the HIGH signals will be a little under 3.3V.

To receive or send Digital signals the Arduino uses Digital pins # 0 - # 13. You may also setup your Analog In pins to act as Digital pins. To set up Analog In pins as Digital pins use the command: `pinMode(pinNumber, value);`

where `pinNumber` is an Analog pin (A0 – A5) and `value` is either INPUT or OUTPUT. To setup Digital pins use the same command but reference a Digital pin for `pinNumber` instead of an Analog In pin. Digital pins default as input, so really you only need to set them to OUTPUT in `pinMode`. To read these pins use the command: `digitalRead(pinNumber)` where `pinNumber` is the Digital pin to which the Digital component is connected. The `digitalRead` command will return either a HIGH or a LOW signal. To send a Digital signal to a pin use the command: `digitalWrite(pinNumber, value);` where `pinNumber` is the number of the pin sending the signal and `value` is either HIGH or LOW.

The Arduino also has the capability to output a Digital signal that acts as an Analog signal, this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command: `analogWrite(pinNumber, value);` where `pinNumber` is a Digital Pin with PWM capabilities and `value` is a number between 0 (0%) and 255 (100%). For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

THINGS TO REMEMBER ABOUT DIGITAL:

- Digital Input/Output uses the Digital pins, but Analog pins can be used as Digital
- To receive a Digital signal use: `digitalRead(pinNumber);`
- To send a Digital signal use: `digitalWrite(pinNumber, value);`
- Digital Input and Output are always either HIGH or LOW

All of the electrical signals that the Arduino works with are either Analog or Digital. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

3.4.2 ANALOG PIN

Humans perceive the world in analog. Everything we see and hear is a continuous transmission of information to our senses. The temperatures we perceive are never 100% hot or 100% cold, they are constantly changing between our ranges of acceptable temperatures. (And if they are out of our range of acceptable temperatures then what are we doing there?) This continuous stream is what defines analog data. Digital information, the complementary concept to Analog, estimates analog data using only ones and zeros.

In the world of Arduino an Analog signal is simply a signal that can be HIGH (on), LOW (off) or anything in between these two states. This means an Analog signal has a voltage value that can be anything between 0V and 5V (unless you mess with the Analog Reference pin). Analog allows you to send output or receive input about devices that run at percentages as well as on and off. The Arduino does this by sampling the voltage signal sent to these pins and comparing it to a voltage reference signal (5V). Depending on the voltage of the Analog signal when compared to the Analog Reference signal the Arduino then assigns a numerical value to the signal somewhere between 0 (0%) and 1023 (100%). The digital system of the Arduino can then use this number in calculations and sketches.

To receive Analog Input the Arduino uses Analog pins # 0 - # 5. These pins are designed for use with components that output Analog information and can be used for Analog Input. There is no setup necessary, and to read them use the command: `analogRead(pinNumber);` where `pinNumber` is the Analog In pin to which the Analog component is connected. The `analogRead` command will return a number including or between 0 and 1023.

The Arduino also has the capability to output a digital signal that acts as an Analog signal, this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command: `analogWrite(pinNumber, value);` where `pinNumber` is a Digital Pin with PWM capabilities and `value` is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by a ~ sign. For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

THINGS TO REMEMBER ABOUT ANALOG:

- Analog Input uses the Analog In pins, Analog Output uses the PWMpins
- To receive an Analog signal use: `analogRead(pinNumber);`
- To send a PWM signal use: `analogWrite(pinNumber, value);`
- Analog Input values range from 0 to 1023 (1024 values because it uses 10 bits, 2^{10})
- PWM Output values range from 0 to 255 (256 values because it uses 8 bits, 2^8)

All of the electrical signals that the Arduino works with are either input or output. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

3.4.3 OUTPUT SIGNALS

Output to the Arduino pins is always Digital, however there are two different types of Digital Output; regular Digital Output and Pulse Width Modulation Output (PWM). Output is only possible with Digital pins # 0 - # 13.

The Digital pins are preset as Output pins, so unless the pin was used as an Input in the same sketch, there is no reason to use the `pinMode` command to set the pin as an Output. Should a situation arise where it is necessary to reset a Digital pin to Output from Input use the command: `pinMode(pinNumber, OUTPUT);` where `pinNumber` is the Digital pin number set as Output. To send a Digital Output signal use the command: `digitalWrite(pinNumber, value);` where `pinNumber` is the Digital pin that is outputting the signal and `value` is the signal. When outputting a Digital signal value can be either HIGH (On) or LOW (Off). `analogWrite(pinNumber, value);` where `pinNumber` is a Digital Pin with PWM capabilities and `value` is a number between 0 (0%) and 255 (100%). For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

Output can be sent to many different devices, but it is up to the user to figure out

which kind of Output signal is needed, hook up the hardware and then type the correct code to properly use these signals.

THINGS TO REMEMBER ABOUT OUTPUT:

- Output is always Digital
- There are two kinds of Output: regular Digital or PWM (Pulse Width Modulation)
- To send an Output signal use `analogWrite(pinNumber, value);` (for analog) or `digitalWrite(pinNumber, value);` (for digital)
- Output pin mode is set using the `pinMode` command: `pinMode(pinNumber, OUTPUT);`
- Regular Digital Output is always either HIGH or LOW
- PWM Output varies from 0 to 255

All of the electrical signals that the Arduino works with are either input or output. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

3.4.4 INPUT SIGNALS

Analog Input enters your Arduino through the Analog In pins # 0 - # 5. These signals originate from analog sensors and interface devices. These analog sensors and devices use voltage levels to communicate their information instead of a simple yes (HIGH) or no (LOW). For this reason, you cannot use a digital pin as an input pin for these devices. Analog Input pins are used only for receiving Analog signals. It is only possible to read the Analog Input pins so there is no command necessary in the `setup ()` function to prepare these pins for input. To read the Analog Input pins use the command: `analogRead(pinNumber);` where `pinNumber` is the Analog Input pin number. This function will return an Analog Input reading between 0 and 1023. A reading of zero corresponds to 0 Volts and a reading of 1023 corresponds to 5 Volts. These voltage values are emitted by the analog sensors and interfaces. If you have an Analog Input that could exceed $V_{cc} + 5V$ you may change the voltage that 1023 corresponds to by using the Aref pin. This pin sets the maximum voltage parameter your Analog Input pins can read. The Aref pin's preset value is 5V.

Digital Input can enter your Arduino through any of the Digital Pins # 0 - # 13. Digital Input signals are either HIGH (On, 5V) or LOW (Off, 0V). Because the Digital pins can be

used either as input or output you will need to prepare the Arduino to use these pins as inputs in your setup () function. To do this type the command: `pinMode(pinNumber, INPUT);`

Inside the curly brackets of the setup () function where pinNumber is the Digital pin number you wish to declare as an input. You can change the pinMode in the loop () function if you need to switch a pin back and forth between input and output, but it is usually set in the setup () function and left untouched in the loop () function. To read the Digital pins set as inputs use the command: `digitalRead(pinNumber);` where pinNumber is the Digital Input pin number.

Input can come from many different devices, but each device's signal will be either Analog or Digital, it is up to the user to figure out which kind of input is needed, hook up the hardware and then type the correct code to properly use these signals.

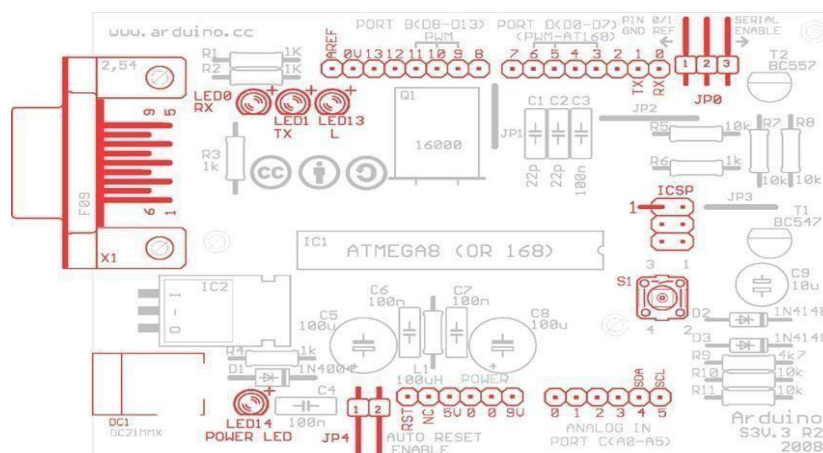


Fig 3.4.4 Pin Configuration of Arduino Uno

X1:

DE-9 serial connector Used to connect computer (or other devices) using RS-232 standard. Needs a serial cable, with at least 4 pins connected: 2, 3, 4 and 5. Works only when JP0 is set to 2-3 position.

DC1:

2.1 mm. power jack Used to connect external power source. Centre positive. Voltage Regulator Works with regulated +7 to +20 volts DC (9v. to 12v. is recommended). It is

possible to alternatively connect external power using 9v. pin or 5v. pin. (see POWER PINOUT).

ICSP:

2x3 pin header Used to program Atmega with bootloader. The number 1 on both sides of the board indicates cable pin1 position. Used to upload sketches on Atmega ICs without bootloader (available only in Arduino IDE versions 0011 and 0012).

JP0:

3 pins jumper when in position 2-3, this jumper enables serial connection (through X1 connector) to/from computer/devices. Use this as default position. When in position 1- 2, it disables serial communication, and enables external pull-down resistors on pin0 (RX) and pin1 (TX). Use this only to prevent noise on RX (that seems incoming data to Atmega), that sometimes makes sketch not starting. When removing this jumper, serial communication is disabled, and pin0 and pin1 work as a normal (floating) digital pin. Useful when more digital pins are needed, but only when serial communication is not necessary. External pull-down/pull-up resistor is required.

JP4:

2 pins jumper when in position 1-2, this jumper enables auto reset feature, useful when uploading a sketch to Arduino, resetting Atmega automatically. It makes unnecessary to press reset button (S1) when uploading sketches. Be sure that computer COM Port speed is set to 19200bps otherwise auto reset will not work properly. If removed, disables auto reset feature. Very useful to prevent undesired Atmega reset when using sketches that needs serial communication. Auto reset works with DTR pulse on serial pin4. Sometimes Arduino senses a DTR pulse when connecting X1 (serial connector) and some softwares sends a DTR pulse when it starts or when it closes, that makes Atmega reset when not desired.

S1:

Tactile button This button resets Atmega, to restart uploaded sketch or to prepare Arduino to receive a sketch through serial connector (when auto reset is not active).

LEDS:

Indicative led POWER led Turns on when Arduino is powered through DC1, +9v. pin or +5v. pin. RX led Blinks when receiving data from computer/device through serial connection. TX led Blinks when sending data to computer/device through serial connection. L led This led is connected to digital pin13 with a current limiter resistor (that doesn't affect pin13). Useful to test sketches. It is normal to blink when boot loading too.

POWER PINOUT:

6 pin header RST pin:

Makes Atmega reset when connected to GND. Useful for Shield Boards, or to connect external reset.

NC pin:

This pin is not connected in Arduino S3v3. Arduino Diecimila has a 3.3 volts pin in the same position.

+9v pin:

When Arduino DC1 is powered (with battery or DC adaptor), this pin is used as Vout, with the same voltage supplied on DC1 (see DC1), minus 0,7 volts. The total supplied current depends on external power source capacity. When Arduino DC1 is not powered, +9v. pin can be used as Vin, connecting it to external regulated power source (+7 to +20 volts) and connecting 0v. pin to external power source GND. In this case,

+5v pin:

Can be used as Vout, supplying +5 volts. +5v pin. When Arduino DC1 is powered (with battery or DC adaptor), +5v. pin supplies +5 volts as a Vout pin. The total supplied current depends on Voltage Regulator (7805 supplies up to 1A). This applies only to

+5v pin:

Atmega in/out pins only supplies max. 40mA on each pin. When Arduino DC1 is not powered, this pin can be used as Vin, connecting it to a regulated +5v. and connecting 0v. pin to power source GND. In this case, +9v. pin is inactive. 0v. pin (GND) Two 0v. pins between +5v. and +9v.

0v pin:

Beside AREF pin. When Arduino DC1 is powered, 0v. pin supplies 0 volts reference (GND) for +5v. pin and +9v. pin. When DC1 is not powered, and Arduino is powered through +5v. pin or +9v. pin, 0v. pin must be used as GND reference, connecting it to the external power source GND.

GND pin:

See 0v pin (GND).

AREF pin:

The AREF can be set to AVcc (default), internal 2.56 volts (Atmega8), internal 1.1 volts (Atmega168), or external AREF. In case of AVcc or internal AREF, AREF pin can be used to attach an external capacitor to decouple the signal, for better noise performance. In case

of external AREF, AREF pin is used to attach the external reference voltage. Remember that it is necessary to change de fuses (wiring.c file), and re-upload sketch, before connecting external voltage to AREF.

3.4.5 SOFTWARE TIPS

When bootloading na Atmega8 chip with Arduino 0010, there is a command (- i800) that makes bootloaddelay 10 minutes. So, if you need to use bootloader, use command line instead of IDE, removing “-i800” command and adding “-F” command, or use Arduino 0007 IDE. To upload sketches Arduino 0010 worksfine.

ARDUINO S3v3 NEW FEATURES

- Full compatible with Shield Boards (Version 2 is the only Arduino Board not compatible with Shield Boards because of ICSP header wrong position, and tall components);
 - A Vcc LP filter to reduce noise level on ADC;
 - Auto reset feature;
 - Auto reset enable/disable jumper, to avoid not desired reseating;
 - Arduino Diecimila compatible reset pin;
 - Pin13 onboard led, with current limiter resistor;
 - TX and RX onboard leds;
 - Power led with appropriate current limiter resistor (less 20mA of consumption);
- Jumper to disable serial communication and to enable RX external pulldown resistor, to avoid “RX floating error”. This feature allows to use digital pin0 and pin1 as a normal pin, when serial communication is not needed;
- All similar components (diodes, transistors, leds, capacitors) has the same board orientation (to makes easier to mount with less mistakes);
- No wires between pads, more space between wires, larger wires, larger pads (better for etching, soldering and drilling, with no short circuits, soldering bridges or open wires in corrosion);
 - Only 3 wire bridges;
 - Electrolytic capacitor (in serial to TTL circuit) changed to bipolar type (to avoid inverted voltage problem when serial cable is not connected);
- All jumpers are right angle type, to allow Shield Boards use.

3.5 Li-Fi (Light Fidelity TECHNOLOGY)

Li-Fi is the short form of “Light Fidelity”. It works on the principle of Visible Light Communication (i.e., VLC). The network is also referred as VPAN or VLC Personal Area Network. The VLC transmits data by intensity modulation. It uses LEDs and Laser diodes (or photo detectors) at transmit and receive ends respectively. It works in 380 nm to 780 nm optical band which is visible light and hence the name VLC.



Fig 3.5 basic concept of Li-Fi

The VLC standard or VPAN standard defines three classes of devices viz. infrastructure, mobile and vehicle. These devices operate in one of the three topologies mentioned below. The different device has different coverage range, data rate and other requirement

Li-Fi Vs Wi-Fi

Li-Fi and Wi-Fi are both used for wireless communication and transmission of data. However, they differ in the medium used. Wi-Fi, for example, uses radio waves transmitted over modems whereas on the other hand Li-Fi uses visible light from LED bulbs for transmission and receiving of data.

3.5.1 History of Li-Fi Technology:

The term Li-Fi was first coined by the University of Edinburgh, however, this technology was first introduced by Harald Has, a German professor, and physicist. He introduced and demonstrated it at a TED talk in July 2011. He believed light bulbs could act as wireless routers for Data transmission. His demonstration showed how changes in the amplitude of a light bulb at high speeds can be used for energy transmission. He demonstrated the use of LED for the transmission of a high-definition video and then showed the same video to the audience. He also formed a company later on by the name of pure Li-Fi. Since then, the use of this wireless technology has increased, now being deployed in more than 20 countries.

3.5.2 Working of Li-Fi

Li-Fi is a VLC (visible light communication) system. It uses a photo Detector (photodiode) to detect incoming signals and decode the data received into digital form. The LED light bulb uses a fast-alternating stream of dim and bright signals that are invisible to the human eye.

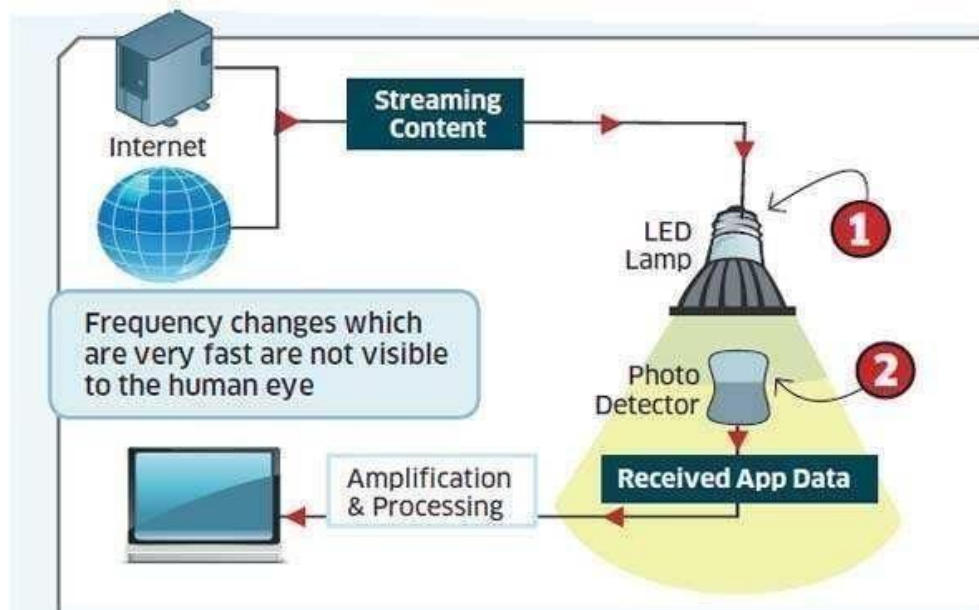


Fig 3.5.2 Working of Li-Fi

First, data is fed to the LED bulb. The bulb operates on signal processing technology so it is capable of sending the data in an embedded format high rapid speeds to the Detector which in this case is the photodiode. The photodiode then converts the incoming beam of light into electrical signals. The electrical signals are then converted into steerable data.

Advantages of Li-Fi Technology

- Li-Fi technology is much faster as compared to Wi-Fi technology. It has the capability to transmit data up to 100 Mbps or 1 Gbps. This means you can download a high-definition video in seconds.
- Li-Fi can help in keeping up with the demand of wireless data networking. After all the increase in radio frequency spectrum occupancy has limited the capacity of data transmission.
- radio waves are penetrable and can be intercepted by unwanted people. On the other hand, Li-Fi is much more secure as the signals are confined to the same room and cannot be intercepted easily.
- Since the Li-Fi systems use LED lights which are already installed in major of our homes and commercial areas, therefore, is an efficient use of energy as compared to Wi-Fi. Wi-Fi networking faces problems of neighboring network interference or radio interference whereas Li-Fi is free from all types of interferences that could affect Wi-Fi system and are therefore more efficient for use.

Disadvantages of Li-Fi Technology

- Li-Fi Can only function If the light source provided is continuous and uninterrupted. You can dim the light but never shut it off. It cannot, therefore, function without a continuous light source.
- Li-Fi is more secure as it is limited to the area of transmission. This might be an advantage but it also limits the range for Li-Fi to work in.

Applications of Li-Fi

- Li-Fi is perfect for use in military bases. This is because of the confinement of the data transmitted to a small area which cannot be penetrated by outside sources. Also, it does not require any complex infrastructure for its application and is simple to use.
- Li-Fi can be used in traffic lights to update drivers on weather conditions and traffic information so they can have a smooth journey on the road. Especially useful for sending critical information.
- Wi-Fi, as we know, uses radio waves which are not suitable for use underwater. Li-Fi can, however, be used for underwater communication as light can penetrate water easily. This will help to make underwater communication much easier.
- Urban areas are already using large quantities of LED bulbs and light sources. The Use of Li-Fi here is therefore very useful. This will not only connect more users in urban areas but will provide fast data to its customers.

3.6 SENSORS DESCRIPTION

Most of the android devices have built-in sensors that measure motion, orientation, and various environmental condition. The android platform supports three broad categories of sensors.

- Motion Sensors
- Environmental sensors
- Position sensors

Some of the sensors are hardware based and some are software-based sensors. Whatever the sensor is, android allows us to get the raw data from these sensors and use it in our application. For this android provides us with some classes. Android provides Sensor Manager and Sensor classes to use the sensors in our application. In order to use sensors, first thing you need to do is to instantiate the object of Sensor Manager class.

3.6.1 PIR SENSOR



Fig 3.6.1 PIR sensor

PIR sensors allow you to sense motion. They are used to detect whether a human has moved in or out of the sensor's range. They are commonly found in appliances and gadgets used at home or for businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

PIRs are made of pyroelectric sensors, a round metal can with a rectangular crystal in the centre, which can detect levels of infrared radiation. Everything emits low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is split in two halves. This is to detect motion (change) and not average IR levels. The two halves are connected so that they cancel out each other. If one-half sees more or less IR radiation than the other, the output will swing high or low.

Advantages of PIR Sensors

- Small in size
- Wide lens range
- Easy to interface
- Inexpensive
- Low-power
- Easy to use
- Do not wear out

3.6.2 MQ 135 sensor



Fig 3.6.2 MQ 135 sensor

MQ135 Gas Sensor is an air quality sensor for detecting a wide range of gases, including NH₃, NO_x, alcohol, benzene, smoke and CO₂. Ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulphide and Benzene steam, also sensitive to smoke and other harmful gases. It is with low cost & particularly suitable for Air quality monitoring application.

Features of MQ135 Gas Sensor

- High sensitivity to Ammonia, Sulphide and Benzene
- Stable and Long Life
- Detection Range: 10 - 300 ppm NH₃, 10 - 1000 ppm Benzene, 10 - 300 ppm Alcohol
- Heater Voltage: 5.0V
- Dimensions: 18mm Diameter, 17mm High excluding pins, Pins - 6mm High
- Long life and low cost

Applications of MQ135 Gas Sensor

- Domestic air pollution detector
- Industrial air pollution detector
- Portable air pollution detector

3.6.3 MQ 4 SENSOR

Sensitive material of MQ-4 gas sensor is SnO_2 , which with lower conductivity in clean air. When the target combustible gas exists the sensors, conductivity increases higher along with the gas concentration rising.

MQ-4 gas sensor has high sensitivity to Natural gas, Methane and could be used to detect both Methane and Propane. The sensor could be used to detect different combustible gas especially Methane, its low cost and suitable for different application.



Fig 3.6.3 MQ 4 SENSOR

Features

- High sensitivity to Combustible gas in wide range
- High sensitivity to Natural gas Methane, Coal gas, CO etc gas
- Fast response
- Wide detection range
- Stable performance, long life, low cost
- Simple drive circuit

APPLICATIONS

- Domestic gas leakage detector
- Industrial Combustible gas detector
- Portable gas detector
- Gas leak alarm, Gas detector

3.6.4 MQ 7 SENSOR



Fig 3.6.4 MQ 7 SENSOR

Sensitive material of MQ-7 gas sensor is SnO_2 , which with lower conductivity in clean air. It makes detection by method of cycle high and low temperature, and detect CO when low temperature (heated by 1.5V). The sensor's conductivity is higher along with the gas concentration rising. When high temperature (heated by 5.0V), it cleans the other gases adsorbed under low temperature. Please use simple electro circuit, convert change of conductivity to correspond output signal of gas concentration. MQ-7 gas sensor has high sensitivity to Carbon Monoxide. The sensor could be used to detect different gases contains CO, it is with low cost and suitable for different application.

Characteristics

- Good sensitivity to Combustible gas in wide range
- High sensitivity to Natural gas
- Long life and low cost
- Simple drive circuit

Application

- Domestic gas leakage detector
- Industrial CO detector
- Portable gas detector

3.6.5 DTH 11 SENSOR

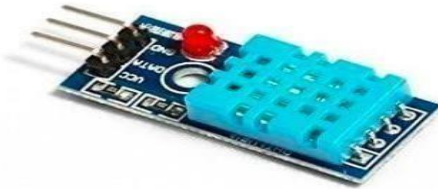


Fig 3.6.5 DHT 11 SENSOR

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

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

Working Principle of DHT11 Sensor

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy.

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure.

3.6.6 ESP 32



Fig 3.6.6 ESP 32

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process.^[2] It is a successor to the ESP8266 microcontroller.

3.6.6.1 Features of the ESP32

- Processors
 - CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
 - Ultra-low power (ULP) co-processor

- Memory
 - 520 KiB SRAM
- Wireless connectivity
 - Wi-Fi: 802.11 b/g/n
 - Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)
- Peripheral interfaces
 - 12-bit SAR ADC up to 18 channels
 - 2 × 8-bit DACs
 - 10 × touch sensors (capacitive sensing GPIOs)
 - 4 × SPI
 - 2 × I²S interfaces
 - 2 × I²C interfaces
 - 3 × UART
 - SD/SDIO/CE-ATA/MMC/eMMC host controller
 - SDIO/SPI slave controller
 - Ethernet MAC interface with dedicated DMA and IEEE 1588 Precision Time Protocol support
 - CAN bus 2.0
 - Infrared remote controller (TX/RX, up to 8 channels)
 - Motor PWM
 - LED PWM (up to 16 channels)
 - Hall effect sensor
 -
- Security
 - IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
 - Secure boot
 - Flash encryption
 - 1024-bit OTP, up to 768-bit for customers
 - Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)
- Power management
 - Internal low-dropout regulator

- Individual power domain for RTC
- 5 μ A deep sleep current
- Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt

Processors

As previously mentioned, the ESP32 uses a Tensilica Xtensa 32-bit LX6 microprocessor. This uses 1 or 2 cores (*all chips in the series are dual-core, except the ESP32-S0WD). The clock frequency reaches up to 240MHz and it performs up to 600 DMIPS (Dhrystone Million Instructions Per Second). Moreover, its low power consumption allows for ADC conversions, computation, and level thresholds, all while in deep sleep mode.

Wireless connectivity

The ESP32 enables connectivity to integrated Wi-Fi through the 802.11 b/g/n/e/i/. Moreover, dual-mode Bluetooth is made possible with the v4.2 BR/EDR and features Bluetooth Low Energy (BLE).

Memory

Internal memory for the ESP32 is as follows

ROM: 448 KiB (for booting/core functions), SRAM: 520 KiB (for data/instructions), RTC fast SRAM: 8 KiB (for data storage/main CPU during boot from sleep mode), RTC slow SRAM: 8 KiB (for co-processor access during sleep mode), and eFuse: 1 KiBit (256 bits used for the system (MAC address and chip configuration) and 768 bits reserved for customer applications). Moreover, two of the ESP32 chips – ESP32-D2WD and ESP32-PICO-D4 – have internally connected flash. The others are as follows: 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips), 2 MiB (ESP32-D2WD chip), and 4 MiB (ESP32-PICO-D4 SiP module).

External Flash and SRAM

ESP32 supports up to four 16 MiB external QSPI flashes and SRAMs with hardware encryption based on AES to protect developers' programs and data. It accesses the external QSPI flash and SRAM through high-speed caches.

Security

IEEE 802.11 standard security features are all supported, including WPA, WPA/WPA2 and WAPI. Moreover, ESP32 has a secure boot and flash encryption.

ESP32 Functions

ESP32 has many applications when it comes to the Internet of Things (IoT).

Here are just some of the IoT functions the chip is used for

Networking

The module's Wi-Fi Antenna and dual-core enables embedded devices to connect to routers and transmit data.

Data Processing

Includes processing basic inputs from analog and digital sensors to far more complex calculations with an RTOS or Non-OS SDK.

P2P Connectivity

Creates direct communication between different ESPs and other devices using IoT P2P connectivity.

Web Server

Access pages written in HTML or development languages.

ESP32 Applications

The ESP32 modules are commonly found in the following IoT devices:

- Smart industrial devices, including Programmable Logic Controllers (PLCs)
- Smart medical devices, including wearable health monitors
- Smart energy devices, including HVAC and thermostats
- Smart security devices, including surveillance cameras and smart locks

3.6.7 Li-Fi transmitter and receiver

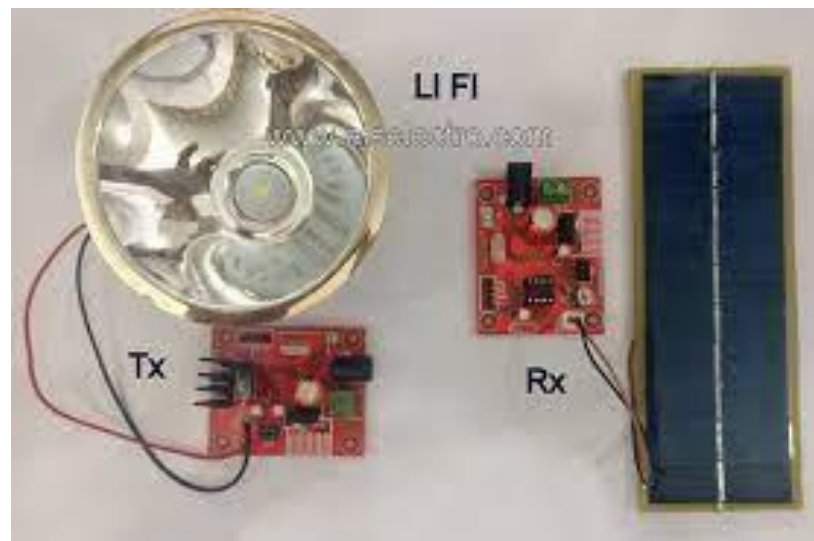


Fig 3.6.7 Li-Fi receiver and transmitter

- Li-Fi can be thought of as a light-based Wi-Fi.
- It uses light instead of radio waves to transmit information and instead of Wi-Fi modems, Li-Fi would use transceiver-fitted LED lamps that can light a room as well as transmit and receive information.
- LI-FI is transmission of data through illumination, i.e., sending data through an LED light bulb that varies in intensity faster than human eye can follow the new LIFI receiver is designed with solar panel which receives the light with data.
- Baud rate of operation is 9600 distance covered is maximum 15 feet without any obstacle.
- Moderate sunlight will affect distance of reception. Trim pot on receiver board can be adjusted to increase the distance.
- Connect Tx of LIFI to Rx of TTL, Rx of Li-Fi to Tx of TTL Power supply 12v (5v regulator is on board) LED source is soldered to Tx board.
- ATMEGA8 IC used on Tx & Rx boards for handling serial data.
- Operational procedure is very simple, if the led is on, it transmits a digital 1, if its off transmits a 0.

LIMITATIONS

1. Interferences from external light sources like sun light, normal bulbs, and opaque materials in the path of transmission will cause interruption in the communication.
2. light can't pass through objects
3. A major challenge facing Li-Fi is how the receiving device will transmit back to transmitter.

3.6.8 Display module



Fig 3.6.8: ILI 9341 Display Module

The display controller for the ILI9341 TFT module is also called ILI9341. It's a low-cost, easy-to-use color display that uses the SPI interface protocol and includes 4 or 5 control pins. This TFT monitor has a resolution of 240 x 320 pixels, or 76800 pixels. This module only accepts 3.3V and does not support 5V. (not 5V tolerant).

This LCD module's controller chip is the ILI9341, which provides different pin-saving SPI communication without sacrificing data transmission speed.

Users will need 6 Dupont lines (SCK, MISO, MOSI, CS, D/C, RST) to monitor this LCD module because its interface is SPI. This module also includes an SD card slot.

CHAPTER 4

HARDWARE DESCRIPTION

4.1 ARDUINO IDE SOFTWARE EXPLANATION

Arduino is an open-source electronics 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. 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 (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

4.1.2 Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for

installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

4.1.3 ARDUINO SOFTWARE

The Arduino Software (IDE) includes a text editor for writing code, a message field, a text console, a toolbar with buttons for common functions, and a set of menus. It communicates with the Arduino and Genuino hardware by connecting to them and uploading programmes.

4.1.4 WRITING SKETCHES

Sketches are programmes created with the Arduino Software (IDE). These sketches were made in a text editor and saved with the .ino file extension. Cutting/pasting, as well as searching/replacing text, are all available in the editor. The message area shows errors and provides input when saving and exporting. The Arduino Program (IDE) outputs text to the console, which includes complete error messages and other information. The designed board and serial port are shown in the window's bottom righthand corner. You can verify and upload programs, make, open, and save sketches, and open the serial monitor using the toolbar buttons. NB: Previous versions of the Arduino Software (IDE) used the extension .pde to save sketches. These files can be opened with version 1.0, but you'll be asked to save the sketch with the .ino extension when you save it.



Verify

Checks your code
for errors
compiling it.



Upload

Compiles your code
and uploads it to the
configured

board.

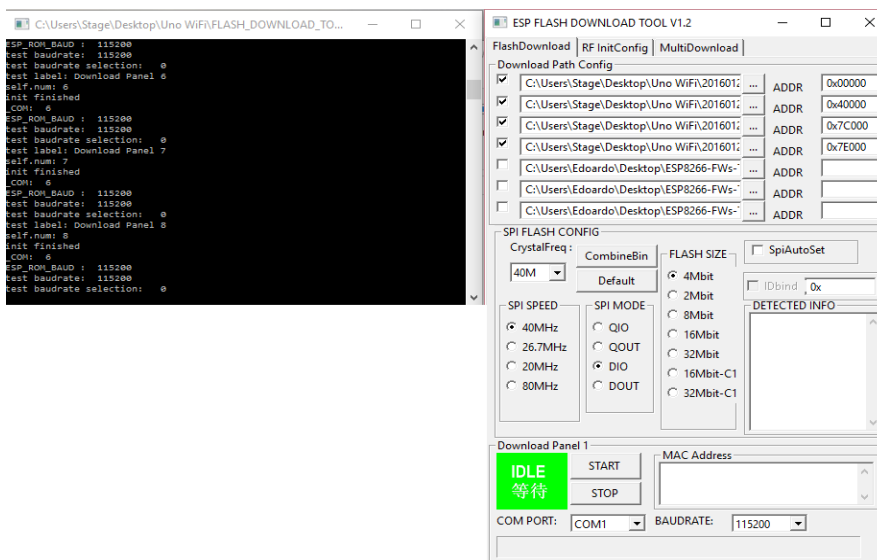
See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"



New

Creates a new sketch.



Open

Presents a menu of all the sketches in your sketchbook.

Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu



doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.



Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

The five menus Paper, Edit, Sketch, Resources, and Support include additional commands. The menus are context sensitive, which means they only display things that are important to the work being performed at the moment.

4.1.5 FILE

Open

Allows to load a sketch file browsing through the computer drives and folders.

Open Recent

Provides a short list of the most recent sketches, ready to be opened.

Sketchbook

Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

New

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

Copy as HTML

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

paste

Puts the contents of the clipboard at the cursor position, in the editor.

select All

Selects and highlights the whole content of the editor. Comment/Uncomment
Puts or removes the // comment marker at the beginning of each selected line.

Increase/Decrease Indent

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

Find

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

Find Next

Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

Find Previous

Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

4.1.6 SKETCH**Verify/Compile**

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

Upload

Compiles and loads the binary file onto the configured board through the configured Port.

Upload Using Programmer

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port

again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.

Export Compiled Binary

Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder

Opens the current sketch folder.

Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

Add File

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

4.1.7 TOOLS

Auto Format

This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

Archive Sketch

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

Fix Encoding &Reload

Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

Serial Monitor

Opens the serial monitor window and initiates the exchange of data with

any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

Board

Select the board that you're using. See below for descriptions of the various boards.

Port

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

Programmer

For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

Burn Bootloader

The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also sets the fuses.

4.1.8 HELP

You will find a variety of documents that come with the Arduino Program here (IDE). Without an internet connection, you can access Getting Started, Reference, this IDE guide, and other documents locally. The documents are a local copy of the online versions, and they may provide a link to our website.

This is the Help menu's only interactive feature: it selects the appropriate page from the local copy of the Reference for the function or order under the cursor.

4.1.9 SKETCHBOOK

The Arduino Software (IDE) is based on the idea of a sketchbook, which is a typical location for storing your programmes (or sketches). The Open button on the toolbar or the File > Sketchbook menu can be used to open the sketches in your sketchbook. The Arduino programme will build a directory for your sketchbook the first time you run it. With the Preferences dialogue, you can see or adjust the position of the sketchbook.

Files are saved with a .ino extension starting with version 1.0. The .pde extension was used in previous versions. In version 1.0 and later, you can still open .pde files; however, the programme will automatically rename the extension to .ino.

4.1.10 TABS, MULTIPLE FILES, AND COMPILATION

Allows you to handle drawings from several files (each of which appears in its own tab). There can be plain Arduino code (no extension), C files (.c extension), C++ files (.cpp extension), or header files (.h).

4.1.11 UPLOADING

You must pick the correct items from the Tools > Board and Tools > Port menus before uploading your sketch. The boards are listed in the following sections. The serial port on a Mac is possibly /dev/tty.usbmodem241 (for an Uno, Mega2560, or Leonardo), /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a Duemilanove or earlier USB board). It's most likely COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) on Windows; to find out, search for USB serial device in the Windows Device Manager's ports section. It should be /dev/ttyACMx,

/dev/ttyUSBx, or something similar on Linux. Click the upload button in the toolbar or pick the Upload item from the Sketch menu once you've selected the right serial port and board. Current Arduino boards will automatically reset and start uploading. You'll need to click the reset button on the board just before beginning the upload on older boards (pre- Diecimila) that don't have auto-reset. When the sketch is posted, the RX and TX LEDs on most boards can blink. When the upload is complete, the Arduino Program (IDE) will show a message or an error.

When you upload a sketch, you're using the Arduino bootloader, which is a tiny programme loaded onto your board's microcontroller. It enables you to upload code without the need for any extra hardware. When the board resets, the bootloader is active for a few seconds before starting whatever sketch was most recently uploaded to the microcontroller. When the bootloader begins, the on-board (pin 13) LED will blink (i.e. when the board resets).

4.1.12 LIBRARIES

Libraries add extra functionality to drawings, such as the ability to interact with hardware or manipulate data. Pick a library from the Sketch > Import Library menu to use it in a sketch. This will compile the library with your sketch and add one or more `#include` statements to the top of the sketch. Libraries take up more room on the board than your sketch since they are uploaded with it. Remove the `#include` statements from the top of your code if a sketch no longer needs a library.

In the reference section, there is a list of libraries. The Arduino app comes with a few libraries. Others are available for download from a number of places, like the Library Manager. You can import a library from a zip file and use it in an open sketch starting with version 1.0.5 of the IDE. To build a third-party library, follow these instructions.

4.1.13 THIRD-PARTY HARDWARE

Support for third-party hardware may be linked to your sketchbook directory's hardware directory. Board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions are some of the platforms that can be installed. Create the hardware directory first, then unzip the third-party platform into its own subdirectory. (If you use "arduino" as the sub-directory name, the built-in Arduino framework will be overridden.) Simply delete the program's directory to uninstall it.

The Arduino IDE 1.5 3rd party Hardware specification has more information on developing packages for third-party hardware.

4.1.14 SERIAL MONITOR

The serial data sent from the Arduino or Genuino board is shown (USB or serial board). Enter text and press enter or click the "send" button to send data to the monitor. Select the baud rate that corresponds to the rate passed to Serial from the drop-down menu. start with your sketch. When you connect with the serial monitor on Windows, Mac, or Linux, the Arduino or Genuino board will reset (restart your sketch execution from the beginning).

You can also communicate with the board using Processing, Flash, MaxMSP, and other programming languages (see the interfacing page for details).

4.1.15 PREFERENCES

In the preferences dialogue, you can set some preferences (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest is in the preferences file, which is shown in the preference dialogue.

4.1.16 LANGUAGE SUPPORT



The Arduino Software (IDE) has been translated into 30+ languages since version 1.0.1. The IDE starts up in the language that your operating system has

chosen by default. (Note that on Windows and probably Linux, the locale environment, which governs currency and date formats, determines this, not the language in which the operating system is displayed.)

If you want to adjust the language manually, open the Preferences window in the Arduino Software (IDE). A dropdown menu of currently supported languages appears next to the Editor Language. Choose your favorite language from the menu, and then restart the programmer to use it. The Arduino Program (IDE) will switch to English if your operating system language is not enabled.

Select System Default from the Editor Language drop-down to return the programme to its default setting of choosing the language depending on your operating system. When you restart the Arduino Program, this setting will take effect (IDE). Similarly, you must restart the Arduino Software (IDE) after updating your operating system's settings to upgrade it to the new default language.

4.2NODE RED

Node RED is a programming tool for building new and interesting links between hardware devices, APIs, and online services. It comes with a browser-based editor that makes it simple to link flows using the palette's large range of nodes, all of which can be deployed to the runtime in a single click.

The change node is another helpful node that allows you to change the message payload or add new properties. You may use this node to modify a message's properties, either by modifying existing ones, removing them, or adding new ones.

In this example, you'll continue with your MQTT theme and see how you can add a new message property `msg.payload`. note now that you've successfully "switched" the message flow based on the incoming `MQTTmessage`.

You can use Node-RED to link nodes to generate flows that carry out your programming mission. Messages move between nodes, from input nodes to processing nodes and eventually to output nodes. Let's look at nodes, flows, and messages.

There are three main types of nodes:

1. Input Nodes (e.g. inject)
2. Output Nodes (e.g. debug)
3. Processing Nodes (e.g. function)

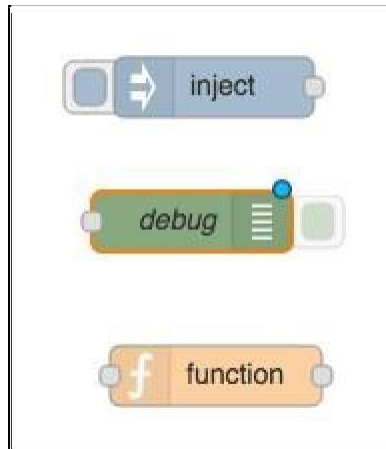


Fig 4.2 The main types of nodes: input, output and processing

You may use input nodes to enter data into a Node-RED application or "flow." They only have one output endpoint on their right hand, which is defined by the small grey rectangle. Input nodes are used to link data from other services, such as Twitter, Google, serial, web sockets, or tcp, or to manually enter data into a flow using the inject node.

You can send data outside of a Node-RED flow using output nodes. On their left hand, they have a single input endpoint. You can send data to other services using output nodes, such as Twitter, TCP, serial, or email nodes, or you can use the debug node to output to the debugpane.

You can process data using processing nodes. They have one or more output endpoints and an input endpoint. They allow you to convert data types (e.g. json, csv, xml) nodes, use data to trigger a message (e.g. trigger, delay) nodes, and write custom code that makes use of the data obtained (e.g. function node).

Some nodes, such as the inject and debug messages, have a button that enables you to actuate a node (in the case of the inject node) or activate and disable a node (in the case of the debug node) (in the case of the debug node).

Flows are made up of multiple nodes that are connected by output tabs that are linked to the input tabs of the next node in the flow. Messages travel across the nodes, bringing data from one node to the next.

Nodes in the Node-RED system consume input messages and generate output messages. Messages are JavaScript objects with at least one "payload" parameter, as seen here:

Listing 2.1 A basic Node-RED message structure

```
1.msg = {  
  1.payload:"message payload"  
3.};
```

Nodes consume and generate messages, and the data they consume and produce is normally stored in msg.payload. Messages, on the other hand, may be expanded to provide additional parameters. To set a message's topic and add a new parameter, position, for example, you could build a new msg object as shown in listing 2.2.

Listing 2.2 A more complex Node-RED message structure

```
1.msg = {  
2.payload:"message payload",  
3.topic:"error",  
4.location:"somewhere in space and time" 5.}
```

4.2.1 Node – red Toll

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

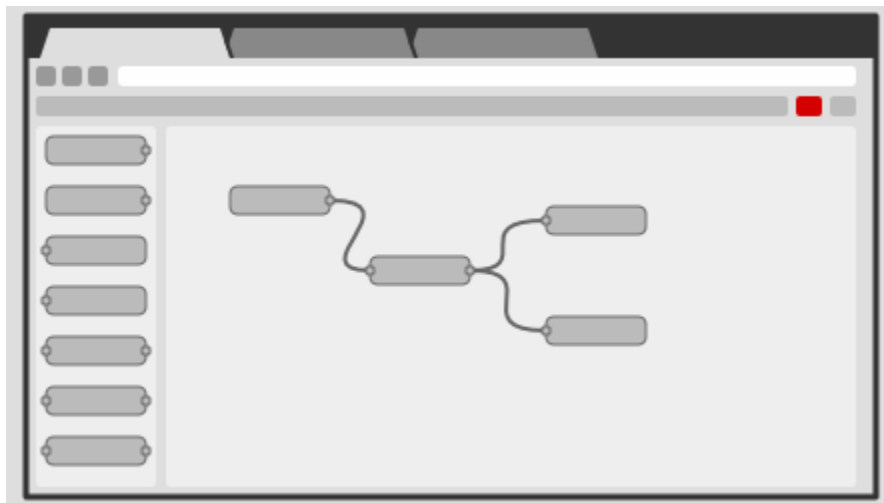


Fig 4.2.1 node red appearance

4.2.2 Browser-based flow editing

Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.

JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use

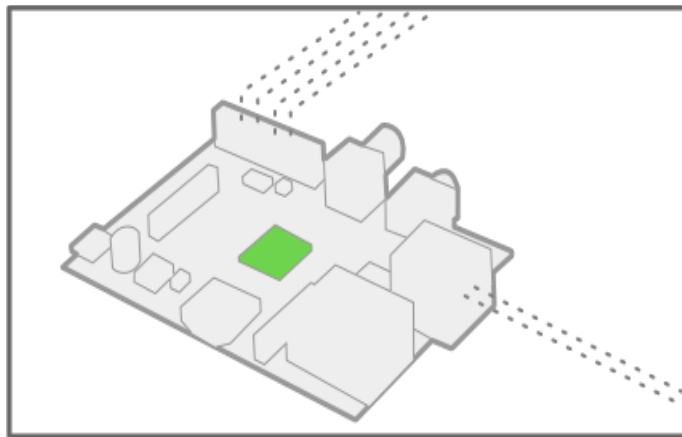


Fig 4.2.2 Browser-based flow editing

4.2.3 Built on Node.js

The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities.

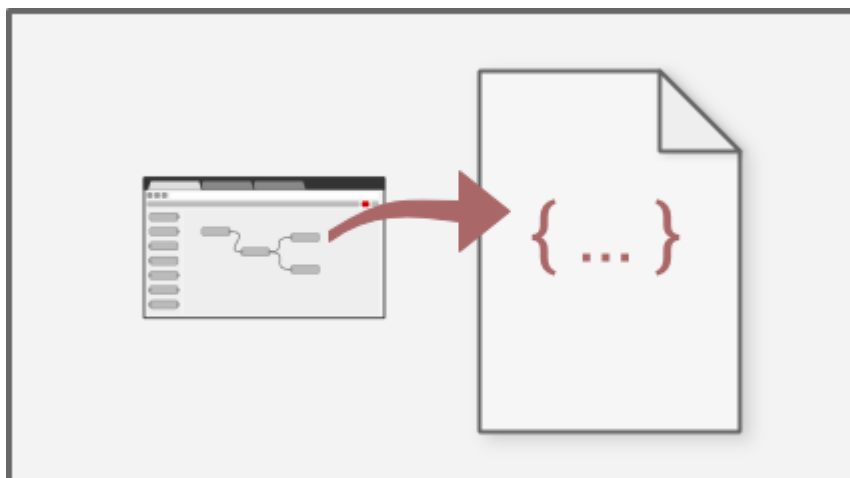


Fig 4.2.3 Built on Node.js

4.2.4 Social Development

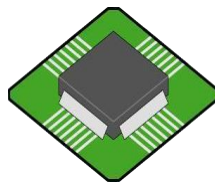
The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others. An online flow library allows you to share your best flows with the world.

4.2.5 GET STARTED

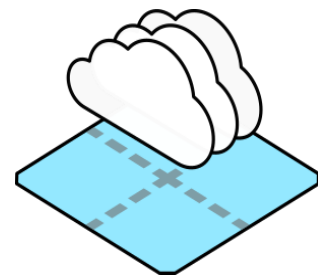
Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.



Run locally



Raspberry pi



IBM

4.2.6 Prerequisites

To install Node-RED locally you will need a supported version of Node.js.

Installing with npm

To install Node-RED you can use the `npm` command that comes with node.js:

```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`.

That command will install Node-RED as a global module along with its dependencies.

You can confirm it has succeeded if the end of the command output looks similar to:

```
+ node-red@1.1.0
added 332 packages from 341 contributors in 18.494s
found 0 vulnerabilities
```

4.2.7 Installing with docker

To run in Docker in its simplest form just run:

```
docker run -it -p 1880:1880 --name mynodered nodered/node-red
```

For more detailed information see our docker guide.

4.2.8 Installing with snap

If your OS supports **Snap** you can install Node-RED with:

```
sudo snap install node-red
```

When installed as a Snap package, it will run in a secure container that does not have access to some extra facilities that may be needed for you to use, such as:

- access to main system storage. Can only read/write to local home directories.
- `gcc` - needed to compile any binary components of nodes you want to install
- `git` - needed if you want to use the Projects feature
- direct access to gpio hardware
- access to any external commands your flows want to use with the Exec node (for example).

You can run it in “classic” mode which reduces the container security but then does provide wider access.

4.2.9 Running

Once installed as a global module you can use the `node-red` command to start Node-RED in your terminal. You can use `Ctrl-C` or close the terminal window to stop Node-RED.

```
$ node-red

Welcome to Node-RED
=====
```

```

30 Jun 23:43:39 - [info] Node-RED version: v1.1.0
30 Jun 23:43:39 - [info] Node.js version: v10.21.0
30 Jun 23:43:39 - [info] Darwin 18.7.0 x64 LE
30 Jun 23:43:39 - [info] Loading palette nodes
30 Jun 23:43:44 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
30 Jun 23:43:44 - [info] Settings file : /Users/nol/.node-red/settings.js
30 Jun 23:43:44 - [info] HTTP Static : /Users/nol/node-red/web
30 Jun 23:43:44 - [info] Context store : 'default' [module=localfilesystem]
30 Jun 23:43:44 - [info] User directory : /Users/nol/.node-red
30 Jun 23:43:44 - [warn] Projects disabled : set
editorTheme.projects.enabled=true to enable
30 Jun 23:43:44 - [info] Creating new flows file : flows noltop.json
30 Jun 23:43:44 - [info] Starting flows
30 Jun 23:43:44 - [info] Started flows
30 Jun 23:43:44 - [info] Server now running at http://127.0.0.1:1880/red/

```

You can then access the Node-RED editor by pointing your browser at <http://localhost:1880>.

The log output provides you various pieces of information:

- The versions of Node-RED and Node.js
- Any errors hit when it tried to load the palette nodes
- The location of your Settings file and User Directory
- The name of the flows file it is using.

Node-RED uses `flows_<hostname>.json` as the default flows file. You can change this by providing the flow file name as argument to t

4.2.10 Command-line Usage

Node-RED can be started using the command `node-red`. This command can take various arguments:

```

node-red [-v] [-?] [--settings settings.js] [--userDir DIR]
          [--port PORT] [--title TITLE] [--safe] [flows.json|projectName]
          [-D X=Y|@file]

```

he `node-red` command.

Option	Description
<code>-p, --port PORT</code>	Sets the TCP port the runtime listens on. Default: <code>1880</code>
<code>--safe</code>	Starts Node-RED without starting the flows. This allows you to open the flows in the editor and make changes without the flows running. When you deploy your changes, the flows are then started.
<code>-s, --settings FILE</code>	Sets the settings file to use. Default: <code>settings.js</code> in <code>userDir</code>
<code>--title TITLE</code>	Set process window title

Option	Description
<code>-u, --userDir DIR</code>	Sets the user directory to use. Default: <code>~/ .node-red</code>
<code>-v</code>	Enables verbose output
<code>-D X=Y @file</code>	Override individual settings
<code>-?, --help</code>	Shows command-line usage help and exits
<code>flows.json projectName</code>	If the Projects feature is not enabled, this sets the flow file you want to work with. If the Projects feature is enabled, this identifies which project should be started.

Node-RED uses `flows_<hostname>.json` as the default flows file. If the computer you are running on may change its hostname, then you should ensure you provide a static file name; either as a command-line argument or using the `flowsFile` option in your **settings file**.

4.2.11 Override individual settings

Since Node-RED 1.1.0

You can override individual settings on the command-line using the `-D` (or `--define`) option.

For example, to change the logging level you can use:

```
-D logging.console.level=trace
```

You can also provide the custom settings as a file:

```
-D @./custom-settings.txt
```

The file should contain a list of the settings to override:

```
logging.console.level=trace
logging.console.audit=true
```

4.2.12 passing arguments to the underlying Node.js process

There are occasions when it is necessary to pass arguments to the underlying Node.js process. For example, when running on devices like the Raspberry Pi or BeagleBone Black that have a constrained amount of memory.

To do this, you must use the `node-red-pi` start script in place of `node-red`. **Note:** this script is not available on Windows.

Alternatively, if are running Node-RED using the `node` command, you must provide arguments for the node process before specifying `red.js` and the arguments you want passed to Node-RED itself.

```
The following two commands show these two approaches: node-red-pi --max-old-space-size=128 --userDir /home/user/node-red-data/  
node --max-old-space-size=128 red.js --userDir /home/user/node-red-data/
```

Upgrading Node-RED

If you have installed Node-RED as a global npm package, you can upgrade to the latest version with the following command:

```
sudo npm install -g --unsafe-perm node-red
```

If you are using Windows, do not start the command with `sudo`.

4.2.12 Creating your first flow

Overview

This tutorial introduces the Node-RED editor and creates a flow that demonstrates the Inject, Debug and Function nodes.

1. Access the editor With Node-RED **running**, open the editor in a web browser.

If you are using a browser on the same computer that is running Node-RED, you can access it with the url: `http://localhost:1880`. If you are using a browser on another computer, you will need to use the ip address of the computer running Node-RED: `http://<ip-address>:1880`.

2. Add an Inject node. The Inject node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects. Drag one onto the **workspace** from the **palette**. Select the newly added Inject node to see information about its properties and a description of what it does in the **Information sidebar pane**.

3. Add a Debug node. The Debug node causes any message to be displayed in the **Debug sidebar**. By default, it just displays the payload of the message, but it is possible to display the entire message object.

4. Wire the two together Connect the Inject and Debug nodes together by **dragging between** the output port of one to the input port of the other.

5. Deploy At this point, the nodes only exist in the editor and must be deployed to the server. Click the Deploy button. With the Debug sidebar tab selected, click the Inject button. You should see numbers appear in the sidebar. By default, the Inject node uses the number of milliseconds since January 1st, 1970 as its payload.

6. Add a Function node The Function node allows you to pass each message through a JavaScript function. Delete the existing wire (select it and press delete on the keyboard). Wire a Function node in between the Inject and Debug nodes. Double-click on the Function node to bring up the edit dialog. Copy the following code into the function field:

```
// Create a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent on
return msg;
```

Click Done to close the edit dialog and then click the deploy button.

Now when you click the Inject button, the messages in the sidebar will now be formatted as readable timestamps.

4.2.13 Summary

This flow demonstrates the basic concept of creating a flow. It shows how the Inject node can be used to manually trigger a flow, and how the Debug node displays messages in the sidebar. It also shows how the Function node can be used to write custom JavaScript to run against messages.

Source:: The flow created in this tutorial is represented by the following json. To import it into the editor, copy it to your clipboard and then paste it into the Import dialog.

```
[{"id":"58ffae9d.a7005","type":"debug","name":"","active":true,"complete":false,"x":640,"y":200,"wires":[]},{ "id":"17626462.e89d9c","type":"inject","name":"","topic":"","payload":"","repeat":"","once":false,"x":240,"y":200,"wires":[["2921667d.d6de9a"]]}, {"id":"2921667d.d6de9a","type":"function","name":"Format timestamp","func":"// Create a Date object from the payload\nvar date = new Date(msg.payload);\n// Change the payload to be a formatted Date string\nmsg.payload = date.toString();\n// Return the message so it can be sent on\nreturn msg;","outputs":1,"x":440,"y":200,"wires":[["58ffae9d.a7005"]]}]
```

CHAPTER 5

STIMULATION RESULT

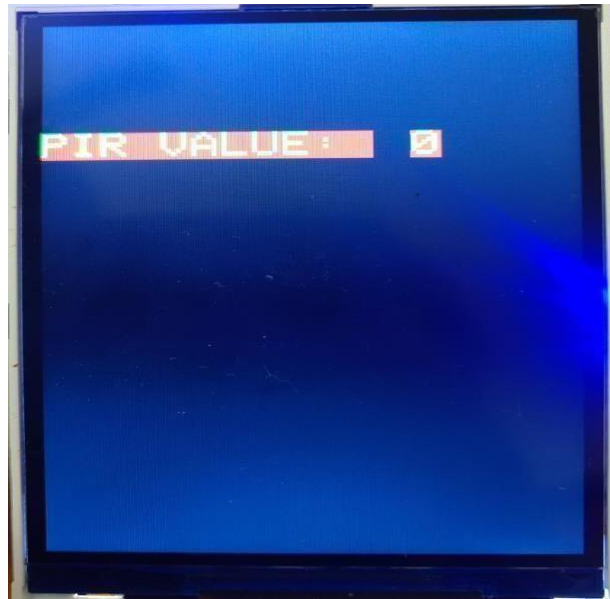


FIG 5.1 PIR VALUE obtained

From the above figure we can observe that the LCD display shows the value of PIR is 0, which is obtained from sensor connected to controlled A passive infrared sensor is an electronic sensor that measure infrared light radiating from objects in its field of view. They are most often used in PIR -based motion detectors. PIR sensors are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector". A PIR-based motion detector is used to sense movement of people, animals, or other objects. They are commonly used in burglar alarms and automatically activated lighting systems.

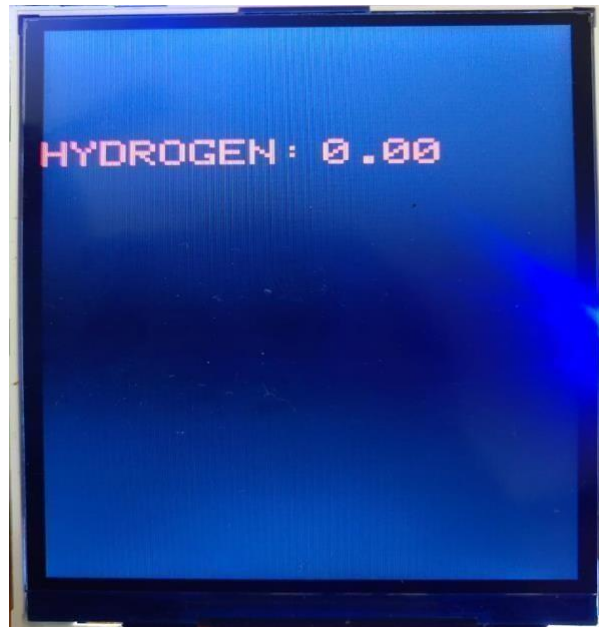


FIG 5.2 HYDROGEN VALUE OBTAINED

From the above figure we can observe that the LCD display shows the value of Hydrogen. A hydrogen 0.00, which is obtained from sensor connected to controller, A hydrogen sensor is a gas detector that detects the presence of hydrogen. They contain micro-fabricated point-contact hydrogen sensors and are used to locate hydrogen leaks. They are considered low -cost ,compact, durable and easy to maintain as compared to conventional gas detecting instruments. The absorbed hydrogen atoms give rise to a small voltage, which is amplified and transferred to the detector instrument.

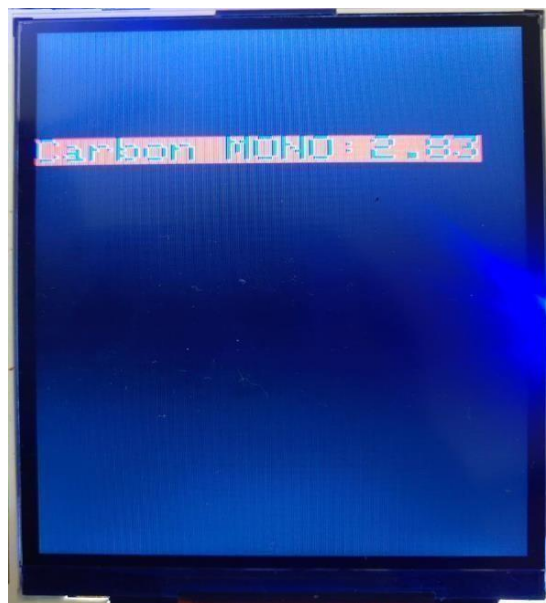


FIG 5.3 CARBON MONOXIDE VALUE OBTAINED

From the above figure we can observe that the LCD display shows the value of Carbon mono:2.83,which is obtained from sensor connected to controller ,A carbon monoxide detector is a device that detects the presence of the carbon monoxide gas to prevent carbon monoxide poisoning .CO is colourless, tasteless, and odourless gas produced by incomplete combustion of carbon-containing materials. CO detectors are designed to measure CO levels over time and sound an alarm before dangerous levels of CO accumulate in an environment, giving people adequate warning to safely ventilate the area or evacuate.

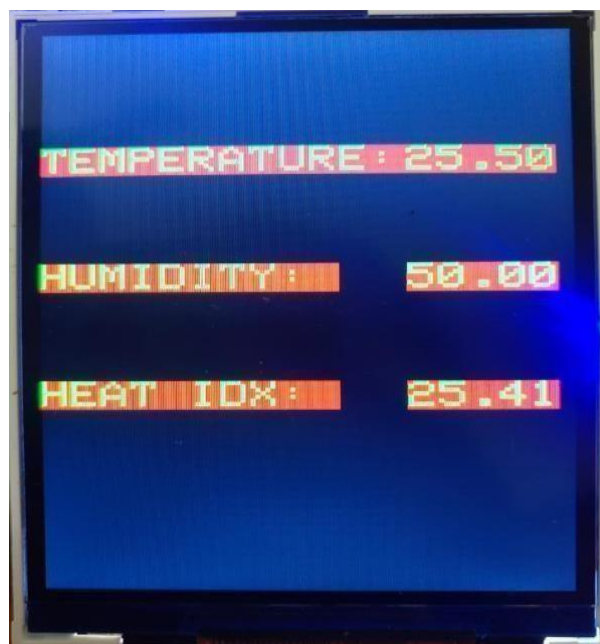


FIG 5.4 TEMPERATURE VALUE OBRTAINED

From the above figure we can observe that the LCD display shows the value of Temperature :25.50, Heat idx: 25.41, Humidity:50.0,which is obtained from sensor connected to controller, Temperature this sensor is a commonly used temperature measuring component in a temperature measuring instrument It directly measures the temperature and converts the temperature signal into a thermoelectromotive force signal, which is converted into the temperature of the measured medium by an electric meter. In industrial production, temperature is one of the important parameters that need to be measured and controlled.

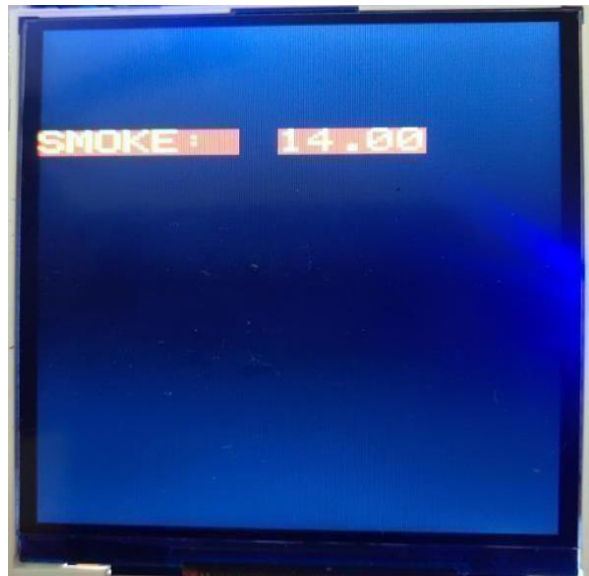


FIG 5.5 SMOKE VALUE OBTAINED

From the above figure we can observe that the LCD display shows smoke: 14:0, which is obtained from sensor connected to controller, A smoke detector is a device that senses smoke, typically as an indicator of fire .Smoke detectors are housed in plastic enclosures, typically shaped like a disk about 150 millimetres in diameter.

CHAPTER 6

CONCLUSION

In this report, a survey on Li-Fi technology, we can see that the Li-Fi is an advanced approach on design, having the best ever design of internet by largely reducing the size of device which transfers data, implementation- by means of having more than 1.4 million light bulbs all over the world if replaced by such LEDS can provide feasible access, and last but not the least enormous applications compared to any other networks in various fields which cannot be imagined by on use networks. Although there are some disadvantages, but can be eliminated by careful further research. Li-Fi has provided a step forward invention in the world of growing hunger communication, this is safe to all biodiversity including humans and progressing towards a greener, cheaper and brighter future of technologies.

FUTURE SCOPE

The Li-Fi Market was valued at USD 295.4 million in 2020 and is expected to reach USD 4157.38 million by 2026, at a CAGR of 69.7% over the forecast period 2021 - 2026. Li-Fi is a disruptive technology that is poised to impact many industries. The technology can unlock the potential of IoT, driving Industry 4.0 applications, and lead to the upcoming light-as-a-service (LaaS) in the lighting industry. The CAGR of wireless traffic has been 60% during the last decade. This growth is expected to sustain for the next 10 to 15 years, which, in the current scenario, is a reasonable assumption, due to the advent of Internet-of-Things and machine type communication (MTC). It corresponds to a demand of 12,000 times the current bandwidth, assuming the same spectrum efficiency.

REFERENCES

- [1] Global statistics - Global 5G subscriptions forecast 2019-2023, 06 2020, [online]
Available: <https://www.5gamericas.org/resources/chartsstatistics/global/>
- [2] A. Haka, R. Vasilyeva, V. Alekseev and H. Volchenkov, "Simulation Framework for Building of 6LiWPAN Network", XVI International Conference on Electrical Machines Drives and Power Systems ELMA 2019, pp. 522-526, June 2019.
- [3] S. A. Chaudhry and J. Zhang, "Network state" aware quality of service provisioning for the internet of things", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 7, no. 6, pp. 369-376, 2016.
- [4] N. K. Kapoor, S. Majumdar and B. "Scheduling of multiple applications in wireless sensor networks using knowledge of applications and network", International Journal of Information and Computer Science (IJICS), vol. 5, pp. 11-30, 2016.
- [5] V. Alekseev and D. Dine, "Simulation Framework for Realization of Quality of Service in Li-Fi Network", International Conference Automatics and Informatics'2017, pp. 201-204, 4-6 October 2017.
- [6] D. Dine, "Simulation framework for realization of horizontal handover in li-fi indoor network", 2019 28th International Scientific Conference Electronics ET 2019 - Proceedings, 12 September 2019.
- [7] N. Zahira, M. Phil and T. K. Pramod, "Implementation of increasing packet delivery ratio using li-fi technology", International Journal of Intellectual Advancements and Research in Engineering Computations, vol. 5, no. 1, pp. 1507-1510, 2017.
- [8] Y. T. Wu, "Research trends in technological pedagogical content knowledge (TPACK) research: A review of empirical studies published in selected journals from 2002 to 2011", British Journal of Educational Technology, vol. 44, no. 3, pp. E73-E76,
- [9] S. Konkani, P. Dillon, T. Vallone, L. Evonne, H. Kooskia and P. Vaananen, "Students' experiences of learning with iPads in upper secondary school - a base for prototype", Education and Information Technologies, vol. 22, no. 4, pp. 1299-1326, July 2017.
- [10] P. Goswami and M. K. Shukla, "Design of a Li-Fi Transceiver", Wireless Engineering and Technology, vol. 8, no. 4, pp. 71-86, 2017

[11] P. Shawsheen, E. Suresh and Ch. Job, "Performance Analysis of Visible Light Communication System for Free Space Optical Communication Link", *Procedia Technology*, vol.24, pp. 827-833, 2016.

[12] M. F. Issyk, B. Yautias and M. R. Haboku, "Applicability of Li-Fi Technology for Industrial Automation Systems", *International Journal of Electronics and Electrical Engineering*, vol. 5, no. 1, pp. 21-25, 2017.

[13] S. Shrestha, "Hadamard Coded Modulation for Visible Light Communication", *Journal of Nepal Physical Society*, vol. 4, no. 1, pp. 93-96, 2017.

[14] G. V. Krishnan, R. Nagarajan, T. D, M. M. Pushpa and S. Shamuyarira, "Vehicle Communication System Using Li-Fi Technology", *International Journal of Engineering and Computer Science*, vol. 6, no. 3, pp. 20651-20657, 2017.

[15] H.D. Kamel, "The Characteristic of Li-Fi Technology Comparing with Wi-Fi", *International Journal of Computation and Applied Sciences*, vol. 2, no. 2, pp. 51-56, 2017

[16] S. L. Walcheren, Sh. D. Pauer, V. V. Titmen and B. Shinde, "Overview of Li-Fi Technology", *International Research Journal of Engineering and Technology*, vol 4, no9, pp.442-426, 2017

[17] M. A. Hadis, "Wireless Communication tends to Smart Technology Li-Fi and its comparison with Wi-Fi", *American Journal of Engineering Research*, vol. 5, no. 5, pp. 40-47, 2016.

[18] K. Tanwar and S. Gupta, "Smart Class Using Li-Fi Technology", *The International Journal of Engineering and Science*, vol. 3, no. 7, pp. 16-18, 2014.

[19] Y. Prewet, "The Next Generation of Wireless Communication Using Li-Fi (Light Fidelity) Technology", *Journal of Computer Networks*, vol. 4, no. 1, pp. 20-29, 2017.

[20] S. Agarwal, Y. Omer, T. B. Patil and S. C. Sawant, "Solar Panel Cells as Power Source and Li-Fi Data Nodes Integrated with Solar Concentrator", *International Journal of Engineering and Applied Computer Science*, vol. 2, no. 5, pp. 172- 175, 2017.