A project report on

SMART HOME MONITORING SYSTEM ON VOICE COMMAND

Partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

V.LAVANYA - 208U1A0461

Team Members

B.GAYATHRI-208U1A0409   S.MITRANANDANA-208U1A0452

C.PRATHISHA-218U1A0423   K.LAKSHMI PRASANNA-208U1A0423

B.NAGALEELA-208U1A0405   N.SUMATHI-208U1A0434

Under The Esteemed Guidance Of

Mrs.S. SALEEMA, M.TECH

Assistant Professor,Department of ECE

(ESTD-2009)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

GOUTHAMI INSTITUTE OF TECHNOLOGY

AND MANAGEMENT FOR WOMEN

(Approved by A.I.C.T.E., New Delhi & Affiliated to J.N.T.University, Anantapur)

Sai Nagar,Peddasettypalli(V),PRODDATUR-516360,KADAPA Dist.A.P.

2020-2024

GOUTHAMI INSTITUTE OF TECHNOLOGY AND MANAGEMENT FOR WOMEN

(Approved by A.I.C.T.E., New Delhi & Affiliated to J.N.T.University, Anantapur)

Sai Nagar,Peddasettypalli(V),PRODDATUR-516360,KADAPA Dist.A.P.

CERTIFICATE

This is to Certify that this thesis entitled "SMART HOME MONITORING SYSTEM ON VOICE COMMAND" is the bonafide of work done by B.GAYATHRI[208U1A0409],V.LAVANYA[208U1A0461],C.PRATHISHA [218U5A0402],B.NAGALEELA[208U1A0405],S.MITRANANDANA[208U1A0452], K.LAKSHMI PRASANNA[208U1A0423],N.SUMATHI[208U1A0434] **under the guidance of** Mrs.S.SALEEMA **M.Tech** Assistant professor in the department of **"ELECTRONICS AND COMMUNICATION ENGINEERING" at GOUTHAMI INSTITUTE OF TECHNOLOGY AND MANAGEMENT FOR WOMEN**(Approved by AICTE, New Delhi Affiliated to J.N.T.University, Anantapur) submitted in partial fulfillment of the requirement for the award of the degree of "Bachelor of Technology.

Project Guide	**Head of the Department**
S.SALEEMA M.Tech	**Dr. B.BALASUBBANA** M.E, Ph.D ,
Assistant Professor	**Head of the Dept**
Electronics & Communication Engineering	Electronics & Communication Engineering

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

It gives us immense pleasure to thank my guide **Mrs**. **S.SALEEMA M.Tech** Assistant. Prof., ECE for having been a source of constant inspiration, precious guidance and generous assistance during the thesis work. I deem it as a privilege to have worked under his able guidance. Without his close monitoring and valuable suggestions this work wouldn't have taken this shape. I feel that his help is un-substitutable and unforgettable.

We also thankful to the honourable correspondent,**Smt K.C.BABAMMA** for providing us with good faculty and for her moral support throughout the course.

We also thankful to honourable Principal,**Dr. M**.**RAMA SUBBAMMA** and I express my sincere gratitude and deep sense of respect for making me available with all the required assistance and for his support and inspiration to carry out this thesis in the Institute.

We take privilege to record thanks to Head of the Department, ECE **Dr. B.BALASUBBANA** M.E, Ph.D , for his encouragement, cooperation and valuable suggestions to complete project in time.

We would like to thank our institution and all faculty members of ECE department for their help and guidance.

We would like to heart-full thanks to our parents and all our friends,especially our classmates for all thoughtful and mind stimulating discussions we had,which prompted us to thank beyond the obvious. We have enjoyed their championship so much during our stay at GITAMW, Proddatur.

IN ALL SINCERITY

B.GAYATHRI	208U1A0409
V.LAVANYA	208U1A0461
C.PRATHISHA	218U5A0402
B.NAGALEELA	208U1A0405
S.MITRANANDANA	208U1A0452
K.LAKSHMI PRASANNA	208U1A0423
N.SUMATHI	208U1A0434

DECLARATION

I, hereby declare that the work which is being presented in this dissertation entitled **“SMART MONITORING SYSTEM ON VOICE COMMAND”** submitted towards the partial fulfillment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** with specialization in **EMBEDDED SYSTEMS AND IOT** at **GOUTHAMI INSTITUTE OF TECHNOLOGY AND MANAGEMENT FOR WOMEN**, is an authentic record of my work carried out under the supervision of **S.SALEEMA,** M.Tech, Associate Professor In Department of ECE, **GOUTHAMI INSTITUTE OF TECHNOLOGY AND MANAGEMENT FOR WOMEN**, **PRODDATUR.**

The matter embodied in this dissertation report has not been submitted by me for the award of any other degree. Further the technical details furnished in the various chapters in this report are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.`

B.GAYATHRI - 208U1A0409
V.LAVANYA - 208U1A0461
C.PRATHISHA- 218U5A0402
B.NAGALEELA - 208U1A0405
S.MITRANANDANA - 208U1A0452
K.LAKSHMI PRASANNA- 208U1A0423
N.SUMATHI - 208U1A0434

Table of Contents

LIST OF FIGURES

ABSTRACT

Home automation involves introducing a degree of computerized or automatic control to certain electrical and electronic systems in a building. These include lighting, temperature control, security systems, garage doors, etc. A hardware system is installed to monitor and control the various appliances. The system would control the appliances based on its configuration. For example, it could automatically turn on the lights at a specified time in the evening, or it could measure the ambient light using a hardware sensor and turn on the lights when it grows dark. It can also allow a person to control appliances from a remote location, such as over the internet. For example, one could turn on the air conditioning from the office, before leaving for home. This project demonstrates a simple home automation system that allows the user to control it with a wireless device such as a Wi-Fi or Bluetooth enabled mobile phone. A desktop PC is used to run the server software. The system allows the user to control each of the lights and fans individually. It can automatically turn off the main lights and turn on a night lamp at a specified time. By measuring the signal strength, it can detect when the user enters a room and automatically turn on the light and fans, and then automatically turn them off when the user leaves the room.

Home automation system achieved great popularity in the last decades and it increases the comfort and quality of life. In this paper an overview of current and emerging home automation systems is discussed. Nowadays most home automation systems consist of a smartphone and microcontroller. A smart phone application is used to control and monitor the home appliances using different type of communication techniques. In this paper the working principle of different type of wireless communication techniques such as ZigBee, Wi-Fi, Bluetooth, and GSM are studied and their features are compared with each other so the users can choose their own choice of technology to build home automation system. Moreover in this research work the survey of different home automation systems is discussed and their advantages and drawbacks are also highlighted.

CHAPTER 1

OVERWIEW

INTRODUCTION

Home automation system is growing rapidly, they are used to provide comfort, convenience, quality of life and security for residents. Nowadays, most home automation systems are used to provide ease to elderly and disabled people and they reduce the human labor in the production of services and goods . Home automation system can be designed and developed by using a single controller which has the ability to control and monitor different interconnected appliances such as power plugs, lights, temperature and humidity sensors, smoke, gas and fire detectors as well as emergency and security systems . One of the greatest advantage of home automation system is that it can be controlled and managed easily from an array of devices such as smartphone, tablet, desktop and laptop . The rapid growth of wireless technologies influences us to use smartphones to remotely control and monitor the home appliances around the world . Several home automation systems use smartphones to communicate with microcontrollers using various wireless communication techniques such as Bluetooth , GSM , ZigBee , Wi-Fi and EnOcean Smartphone applications are used to connect to the network so that the authorized users can adjust the setting of system on their personal devices. Different type of home automation systems offer a wide range of functions and services, some of the common features are appliance control, thermostat control, remote control lighting, live video surveillance, monitor security camera, real time text alerts. Similar research was carried out by . A low cost and user friendly, smart living system is presented which also use android application to control home appliances. The wireless connection between android device and home appliances is developed via Bluetooth technology. It also provided security and alert system for proposed smart living system.

Chapter 2

LITERATURE SURVEY

1. Bluetooth based home automation system using cell phones:

In Bluetooth based home automation system the home appliances are connected to the Arduino BT board at input output ports using relay. The program of Arduino BT board is based on high level interactive C language of microcontrollers; the connection is made via Bluetooth.The password protection is provided so only authorized user is allowed to access the appliances. The Bluetooth connection is established between Arduino BT board and phone for wireless communication. In this system the python script is used and it can install on any of the Symbian OS environment, it is portable. One circuit is designed and implemented for receiving the feedback from the phone, which indicate the status of the device.

2. Zigbee based home automation system using cell phones:

To monitor and control the home appliances the system is designed and implemented using Zigbee. The device performance is record and store by network coordinators. For this the Wi-Fi network is used, which uses the four

switch port standard wireless ADSL modern router. The network SSID and security Wi-Fi parameter are preconfigured. The message for security purpose first process by the virtual home algorithm and when it is declared safe it is re-encrypted and forward to the real network device of the home. Over Zigbee network, Zigbee controller sent messages to the end. The safety and security of all messages that are received by the virtual home algorithm. To reduce the expense of the system and the intrusiveness of respective installation of the system Zigbee communication is helpful

Chapter 3

3.EMBEDDED SYSTEM

3.1 INTRODUCTION

An **embedded system** is a computer system—a combination of a computer processor,computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. Embedded systems control many devices in common use. In 2009, it was estimated that ninety-eight percent of all microprocessors manufactured were used in embedded systems.

Modern embedded systems are often based on microcontroller (i.e. microprocessors with integrated memory and peripheral interfaces), but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialized in a certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the DSP.

Since the embedded system is dedicated to specific tasks,design can optimize it to reduce the size and cost of the product and increase its reliability and performance. Some embedded systems are mass-produced, benefiting from economics on scale.

Embedded systems range in size from portable personal devices such as digital watches and MP3 players to bigger machines like home appliances, industrial assembly lines, robots, transport vehicles, traffic light controllers, and medical imaging systems. Often they constitute subsystems of other machines like avionics in aircraft and astrionics in spacecraft. Large installations like factories, pipelines and electrical grids rely on multiple embedded systems networked together. Generalized through software customization, embedded systems such as programmable logic controllers frequently comprise their functional units.

Embedded systems range from those low in complexity, with a single microcontroller chip, to very high with multiple units, peripherals and networks, which may reside in equipment racks or across large geographical areas connected via long-distance communications lines.

3.2 APPLICATIONS:

Embedded systems are commonly found in consumer, industrial, automotive, home appliances, medical, telecommunication, commercial, aerospace and military applications.

Telecommunications systems employ numerous embedded systems from telephone switches for the network to cell phones at the end user. Computer networking uses dedicated routers network bridges to route data.

Consumer electronics include MP3 players, television sets, mobile phones, video game consoles, digital cameras, GPS receivers, and printers. Household appliances, such as microwave ovens, washing machines and dishwashers, include embedded systems to provide flexibility, efficiency and features. Advanced heating, ventilation, and air conditioning (HVAC) systems use networked thermostats to more accurately and efficiently control temperature that can change by time of day and season. Home automation uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

also have considerable safety requirements. Spacecraft rely on astrionics systems for trajectory correction. Various electric motors — brushless DC motors, induction motors and DC motors — use electronic motor controllers. Automobiles, electric vehicles, and hybrid vehicles increasingly use embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems using embedded systems include anti-lock braking system (ABS), electronic stability control (ESC/ESP), traction control (TCS) and automatic four-wheel drive.

3.3 Characteristics:

Embedded systems are designed to perform a specific task, in contrast with general-purpose computers designed for multiple tasks. Some have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems are a small part within a larger device that serves a more general purpose. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is to play music.[9] Similarly, an embedded system in an automobile provides a specific function as a subsystem of the car itself.

3.4 Embedded Software Architectures:

Simple control loop

In this design, the software simply has a loop which monitors the input devices. The loop calls subroutines, each of which manages a part of the hardware or software. Hence it is called a simple control loop or programmed input-output.

Interrupt-controlled system

Some embedded systems are predominantly controlled by interrupts. This means that tasks performed by the system are triggered by different kinds of events; an interrupt could be generated, for example, by a timer at a predefined interval, or by a serial port controller receiving data.

This architecture is used if event handlers need low latency, and the event handlers are short and simple. These systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

Cooperative multitasking

Cooperative multitasking is very similar to the simple control loop scheme, except that the loop is hidden in an API.[3][1] The programmer defines a series of tasks, and each task gets its own environment to run in. When a task is idle, it calls an idle routine which passes control to another task.

The advantages and disadvantages are similar to that of the control loop, except that adding new software is easier, by simply writing a new task, or adding to the queue.

Preemptive multitasking or multi-threading

In this type of system, a low-level piece of code switches between tasks or threads based on a timer invoking an interrupt. This is the level at which the system is generally considered to have an operating system kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel.

As any code can potentially damage the data of another task (except in systems using a memory management unit) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy such as message queues, semaphores or a non-blocking synchronization scheme.

Because of these complexities, it is common for organizations to use an off-the-shelf RTOS, allowing the application programmers to concentrate on device functionality rather than operating system services. The choice to include an RTOS brings in its own issues, however, as the selection must be made prior to starting the application development process. This timing forces developers to choose the embedded operating system for their device based on current requirements and so restricts future options to a large extent.[19]

The level of complexity in embedded systems is continuously growing as devices are required to manage peripherals and tasks such as serial, USB, TCP/IP, Bluetooth, Wireless LAN, trunk radio, multiple channels, data and voice, enhanced graphics, multiple states, multiple threads, numerous wait states and so on. These trends are leading to the uptake of embedded middleware in addition to an RTOS.

Microkernels and exokernels

A microkernel allocates memory and switches the CPU to different threads of execution. User-mode processes implement major functions such as file systems, network interfaces, etc.

Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to and extensible by application programmers.

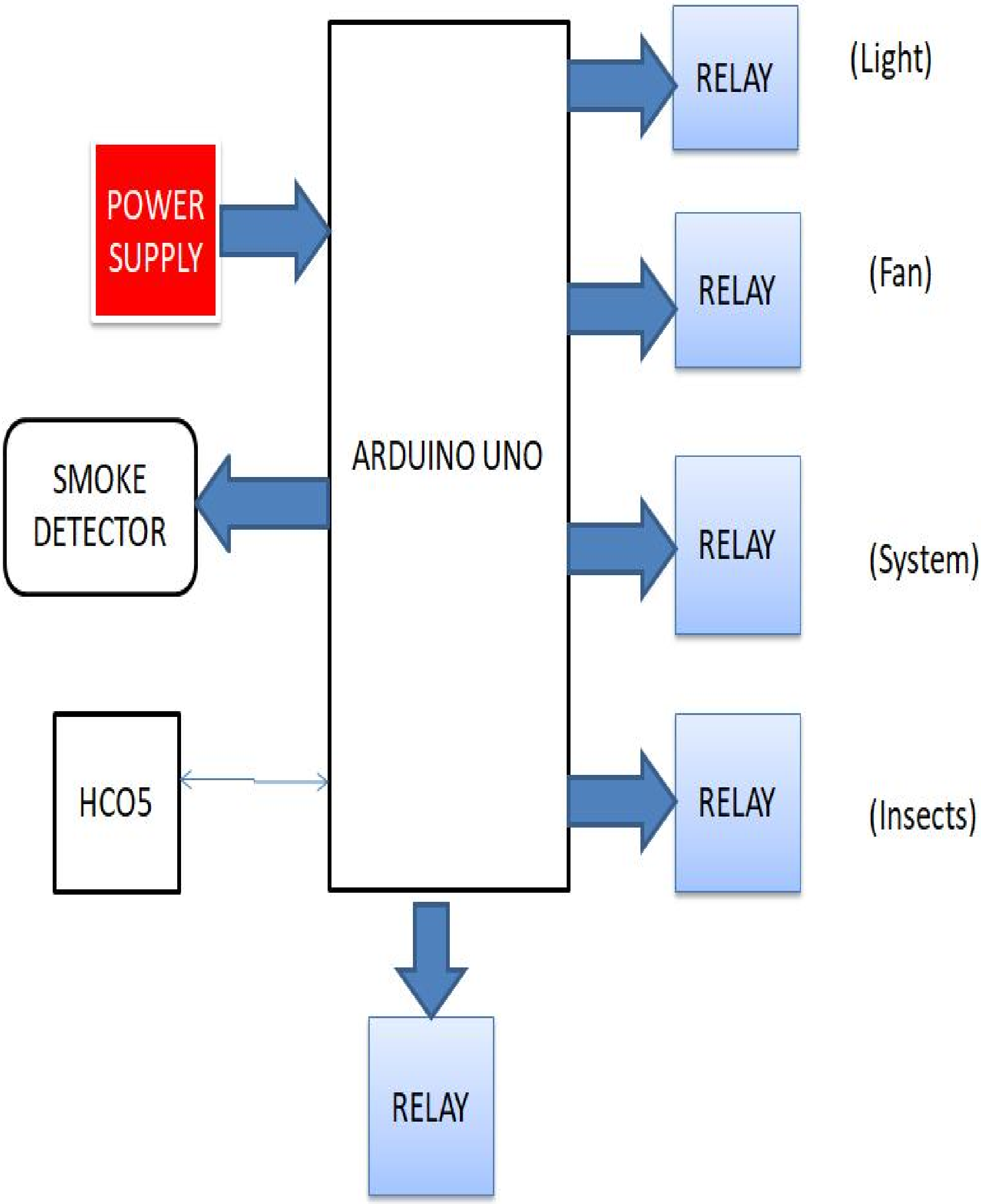
Monolithic kernels

A monolithic kernel is a relatively large kernel with sophisticated capabilities adapted to suit an embedded environment.

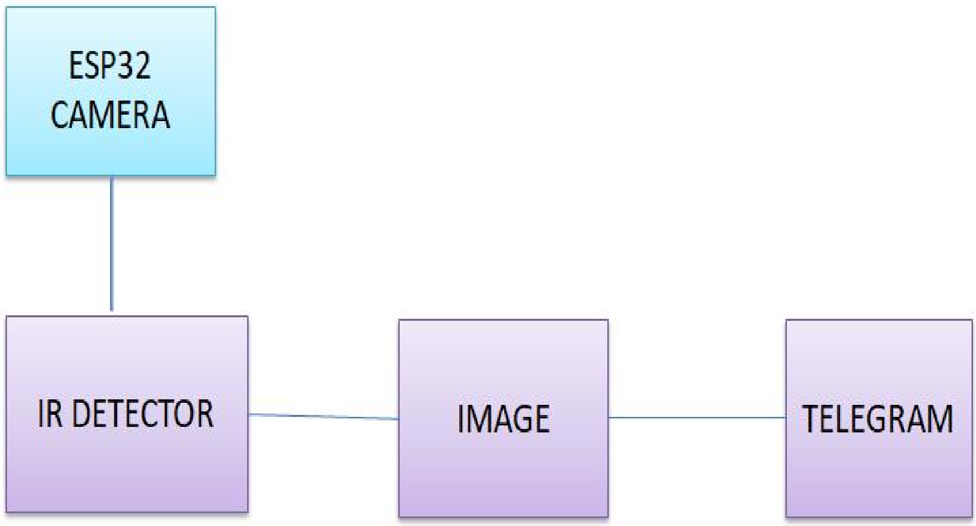
Chapter4

BLOCK DIAGRAM

4.1 EXISTING SYSTEM:



PROPOSED SYSTEM:



4.1 REQUIRED COMPONENTS

4.1.1 HARDWARE:

1. Power supply

2. Smoke detector

3.Relay

4.1.2 SOFTWARE AND LIBRARY TOOLS

1. Arduino IDE

2.Arduino Bluetooth Module

4.2 POWER SUPPLY:

Power supplies are devices that convert electrical power from one form to another, typically from alternating current (AC) to direct current (DC). They are essential components in various electronic devices, providing the necessary voltage and current for proper operation.

There are different types of power supplies, including:

1. **Linear Power Supplies:** These power supplies use a transformer to convert AC voltage to a lower AC voltage, then rectify and filter it to obtain a DC voltage. They are relatively simple but less efficient compared to other types.
2. **Switching Power Supplies:** These power supplies use high-frequency switching circuits to convert AC voltage to DC voltage. They are more efficient and compact compared to linear power supplies, making them suitable for a wide range of applications.
3. **Uninterruptible Power Supplies (UPS):** UPS systems provide backup power in case of mains power failure. They typically consist of a battery or flywheel energy storage system that kicks in when the main power source fails.
4. **Regulated Power Supplies:** These power supplies maintain a constant output voltage regardless of changes in input voltage or load conditions. They ensure stable power delivery to the connected devices.
5. **Variable Power Supplies:** Variable power supplies allow the user to adjust the output voltage and sometimes the current. They are useful for testing and prototyping electronic circuits.

When selecting a power supply, factors such as voltage output, current capacity, efficiency, size, and cost should be considered based on the specific requirements of the application. Additionally, safety features such as overload protection, short-circuit protection, and overvoltage protection are crucial for protecting both the power supply and the connected devices.

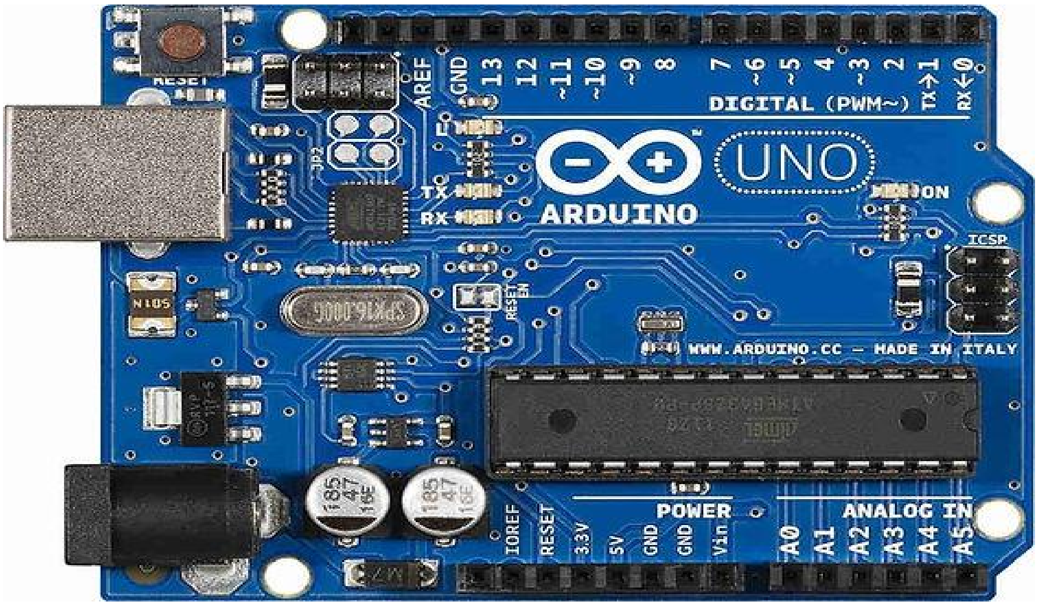
In embedded systems, power supplies play a critical role in providing the necessary electrical power for the system to function reliably. Embedded systems are often designed to operate in various environments and conditions, so choosing the right power supply is crucial for ensuring proper operation and longevity of the system.

Here are some key considerations for power supplies in embedded systems:

1. **Voltage and Current Requirements:** Embedded systems typically have specific voltage and current requirements for their components such as microcontrollers, sensors, and communication modules. The power supply should be able to provide stable voltage levels within the required range and sufficient current capacity to meet the system's demands.
2. **Efficiency:** Efficiency is essential in embedded systems, especially for battery-powered or energy-efficient applications. Choosing a power supply with high efficiency can help minimize power consumption and extend battery life, which is critical in portable or remote embedded systems.
3. **Size and Form Factor:** Embedded systems often have space constraints, especially in applications where size and weight are crucial factors. Selecting a compact and lightweight power supply that fits within the system's enclosure is essential for efficient integration and optimal use of available space.
4. **Input Voltage Range:** Embedded systems may need to operate in environments with varying input voltage levels, such as automotive or industrial applications. A power supply with a wide input voltage range can ensure compatibility with different power sources and protect the system from voltage fluctuations or transients.
5. **Regulation and Protection Features:** Built-in regulation and protection features such as overvoltage protection, overcurrent protection, and thermal shutdown are vital for ensuring the safety and reliability of the embedded system. These features help prevent damage to the system components and ensure stable operation under various operating conditions.
6. **Noise and Electromagnetic Interference (EMI):** Power supplies in embedded systems should generate minimal electrical noise and EMI to prevent interference with sensitive electronic components and communication signals. Choosing a power supply with low noise and EMI characteristics is crucial for maintaining signal integrity and system performance.

4.3ARDUINO UNO:

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.



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.

General pin functions:

LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.

VIN: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

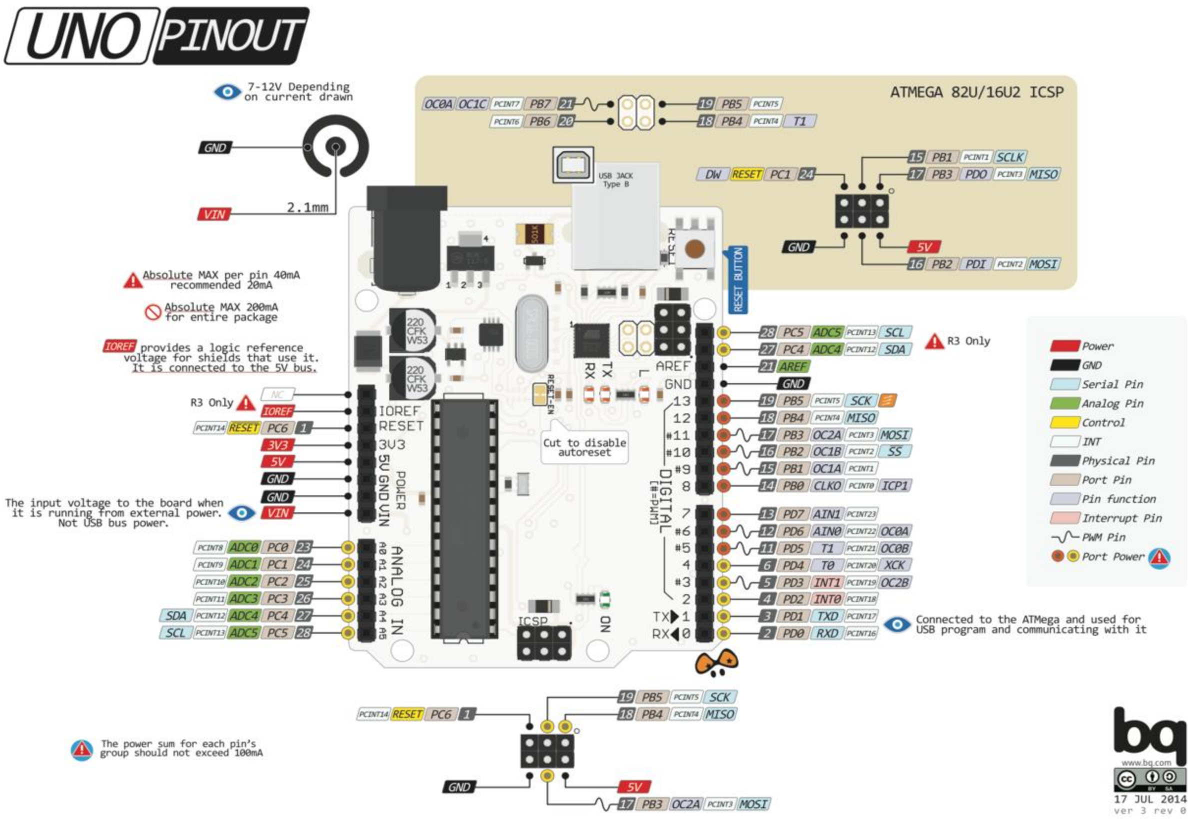
5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND: Ground pins.

IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.

Reset: Typically used to add a reset button to shields that block the one on the board.



Special pin functions:

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

Serial / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) **and** transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.

- External interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

- PWM (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.

- SPI (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.

- TWI (two-wire interface) / I²C: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.

- AREF (analog reference): Reference voltage for the analog inputs.

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

POWER SUPPLY:

A power supply is an electrical device that supplies electric power to an electrical load. The main purpose of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result, power supplies are sometimes referred to as electric power converters. Some power supplies are separate standalone pieces of equipment, while others are built into the load appliances that they power. Examples of the latter include power supplies found in desktop computers and consumer electronics devices. Other functions that power supplies may perform include limiting the current drawn by the load to safe levels, shutting off the current in the event of an electrical fault, power conditioning to prevent electronic noise or voltage surges on the input from reaching the load, power-factor correction, and storing energy so it can continue to power the load in the event of a temporary interruption in the source power (uninterruptible power supply).

TYPES:

An AC-to-DC power supply operates on an AC input voltage and generates a DC output voltage. Depending on application requirements the output voltage may contain large or negligible amounts of AC frequency components known as ripple voltage, related to AC input voltage frequency and the power supply's operation. A DC power supply operating on DC input voltage is called a DC-to-DC converter. This section focuses mostly on the AC-to-DC variant.

Linear Power Supply: An AC adapter disassembled to reveal a simple, unregulated linear DC supply circuit: a transformer, four diodes in a bridge rectifier arrangement, and an electrolytic capacitor to smooth the waveform

In a linear power supply the AC input voltage passes through a power transformer and is then rectified and filtered to obtain a DC voltage. The filtering reduces the amplitude of AC mains frequency present in the rectifier output and can be as simple as a single capacitor or more complex such as a pi filter. The electric load's tolerance of ripple dictates the minimum amount of filtering that must be provided by the power supply. In some applications, ripple can be entirely ignored. For example, in some battery charging applications, the power supply consists of just a transformer and a diode, with a simple resistor placed at the power supply output to limit the charging current.

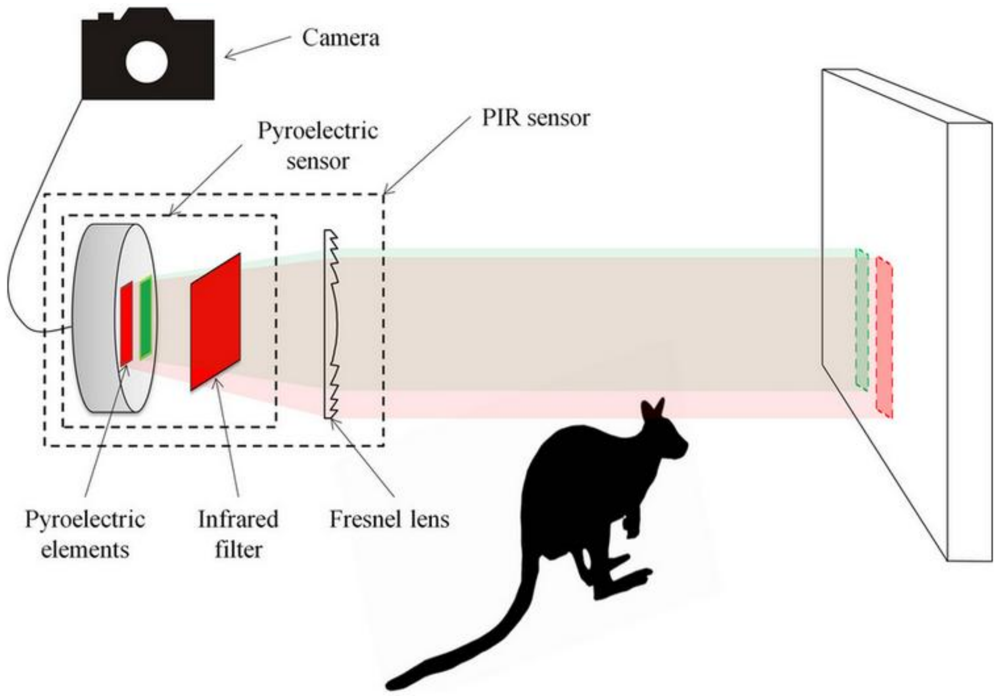
REGULATOR:

A regulated power supply is an embedded circuit; it converts unregulated AC (alternating current) into a constant DC. With the help of a rectifier it converts AC supply into DC. Its function is to supply a stable voltage (or less often current), to a circuit or device that must be operated within certain power supply limits. The output from the regulated power supply may be alternating or unidirectional, but is nearly always DC (direct current).[1] The type of stabilization used may be restricted to ensuring that the output remains within certain limits under various load conditions, or it may also include compensation for variations in its own supply source. The latter is much more common today.

4.4 SENSOR:

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation.



Passive infrared (PIR) sensors only detect infrared radiation and do not emit it from an LED. Passive infrared sensors are comprised of:

- Two strips of pyroelectric material (a pyroelectric sensor)
- An infrared filter (that blocks out all other wavelengths of light)
- A fresnel lens (which collects light from many angles into a single point)
- A housing unit (to protect the sensor from other environmental variables, such as humidity)

PIR sensors are most commonly used in motion-based detection, such as in-home security systems. When a moving object that generates infrared radiation enters the sensing range of the detector, the difference in IR levels between the two pyroelectric elements is measured. The sensor then sends an electronic signal to an embedded computer, which in turn triggers an alarm.

Active infrared sensors **work with radar technology and they both emit and receive infrared radiation**. This radiation hits the objects nearby and bounces back to the receiver of the device. Through this technology, the sensor can not only detect movement in an environment but also how far the object is from the device.

Working of an IR Sensor

The white LED here is an IR LED which works as the transmitter and the component next to the IR LED is a photodiode that works as the receiver in the IR sensor.

The IR transmitter continuously emits the IR light and the IR receiver keeps on checking for the reflected light. If the light gets reflected back by hitting any object in front it, the IR receiver receives this light. This way the object is detected in the case of the IR sensor.

The blue knob here is a potentiometer. You can control the range i.e. from how far you want to detect the object by changing the value of the potentiometer.

An IR sensor has two small LED indicators – one for power, which is ON the entire time the sensor is ON; the other is the Signal LED which detects the object. The signal LED has two states or situations:

- ON (Active) when it detects an object
- OFF (Inactive) when it doesn't detect any object

An **infrared detector** is a detector that reacts to infrared (IR) radiation. The two main types of detectors are thermal and photonic (photodetectors).

The thermal effects of the incident IR radiation can be followed through many temperature dependent phenomena.[2] Bolometers and microbolometers are based on changes in resistance.

4.5 RELAY:

A Relay is a simple electromechanical switch. While we use normal switches to close or open a circuit manually, a Relay is also a switch that connects or disconnects two circuits. But instead of a manual operation, a relay uses an electrical signal to control an electromagnet.

Basic design and operation:

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core (a solenoid), an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two contacts in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. The armature is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil it generates a magnetic field that activates the armature, and the consequent movement of the movable contact(s) either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

Every electromechanical relay consists of an consists of an

1.Electromagnet

2.Mechanically movable contact

3.Switching points and

4.Spring

Applications:

Protective relays:

For protection of electrical apparatus and transmission lines, electromechanical relays with accurate operating characteristics were used to detect overload, short-circuits, and other faults. While many such relays remain in use, digital protective relays now provide equivalent and more complex protective functions.

Railway Signaling:

Railway signalling relays are large considering the mostly small voltages (less than 120 V) and currents (perhaps 100 mA) that they switch. Contacts are widely spaced to prevent flashovers and short circuits over a lifetime that may exceed fifty years.

Since rail signal circuits must be highly reliable, special techniques are used to detect and prevent failures in the relay system. To protect against false feeds, double switching relay contacts are often used on both the positive and negative side of a circuit, so that two false feeds are needed to cause a false signal.

4.6 SMOKE DETECTOR:

In 1965 **Stanley Bennett Peterson and Duane D. Pearsall** invented the first home smoke detector. This device was cheap enough to be installed in most homes, although it was not widely used in homes until the 1970s.

Smoke alarms **detect fires by sensing small particles in the air** using a couple of different kinds of technologies. Once they detect those particles above a certain level, they signal the alarm to sound so that you and your family can get to safety.

There are three types of smoke alarms, ionization, photoelectric and a combination of the two which is commonly called a "dual" detector. Look for the UL stamp on any smoke alarm.

smoke detector, device used to warn occupants of a building of the presence of a fire before it reaches a rapidly spreading stage and inhibits escape or attempts to extinguish it. On sensing smoke the detectors emit a loud, high-pitched alarm tone, usually warbling or intermittent, and usually accompanied by a flashing light. There are two types of smoke detector: photoelectric and ionization. Photoelectric smoke detectors utilize a light-sensitive cell in either of two ways. In one type, a light source, e.g., a small spotlight, causes a photoelectric cell to generate current that keeps an alarm circuit open—until visible particles of smoke interrupt the ray of light, breaking the circuit and setting off the alarm. The other photoelectric detector, widely used in private dwellings, employs a detection chamber shaped so that the light-sensitive element cannot ordinarily "see" the light source (usually a light-emitting diode [LED]). When particles of smoke enter a portion of the chamber that is aligned with both the LED and the photocell, the particles diffuse or scatter the light ray so it can be "seen" by the photocell. As a result a current is generated by the light-sensitive cell and the alarm is triggered.

Ionization detectors employ radioactive material—in quantities so tiny they are believed to pose no significant health hazard—to ionize the air molecules between a pair of electrodes in the detection chamber. This enables a minute current to be conducted by the ionized air. When smoke enters the chamber, particles attach themselves to ions and diminish the flow of current. The reduction in current sets off the alarm circuit.

Photoelectric detectors respond faster and more effectively to the large smoke particles generated by a smoldering, slow-burning fire. Ionization detectors respond faster to the tiny smoke particles released by a fast-burning fire. For this reason some manufacturers produce combination versions of detectors. Many fire-prevention authorities recommend the use of both photoelectric and ionization types in various locations in a private home. Either type of detector can be powered by batteries or by house current.

There are three types of smoke alarms, ionization, photoelectric and a combination of the two which is commonly called a "dual" detector. Look for the UL stamp on any smoke alarm. Research has shown: Ionization smoke alarms detect flaming fires marginally earlier than photo-electric smoke alarms.

Smoke detector covers 100m2. Should not exceed 7.5m from any point. General spacing guidance: 5m from a wall, 10m between detectors.

4.7 HC-O5:

The HC-05 is **a class 2 Bluetooth module designed for transparent wireless serial communication**. It is pre-configured as a slave Bluetooth device. Once it is paired to a master Bluetooth device such as PC, smart phones and tablet, its operation becomes transparent to the user.

In AT mode, the consumption is under 3 mA. While pairing or searching for devices, the consumption is under 40 mA. After pairing, the current consumption is under 8 mA if there is no communication. While communicating, the consumption of the bluetooth device is about 20 MA.

In AT mode, the consumption is under 3 mA. While pairing or searching for devices, the consumption is under 40 mA. After pairing, the current consumption is under 8 mA if there is no communication. While communicating, the consumption of the bluetooth device is about 20 mA.

It uses the 2.45GHz frequency band. The transfer rate of the data can vary up to 1Mbps and is in range of **10 meters**. The HC-05 module can be operated within 4-6V of power supply.

HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC.

HC-05 Bluetooth Module is normally used for wireless data transmission among multiple microcontrollers. It can also be used to communicate between electronic devices like mobile, laptop, computers for data transmission. It also used in different information and data logging applications.

It communicates with microcontrollers using serial communication (USART). Default settings of HC-05 Bluetooth module can be changed using certain AT commands. As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, there is no need to shift TX voltage level of HC-05 module.

The HC-05 Wireless Bluetooth Module without Baseplate is **a slave only device**. This means that it can connect to most phones and computers with Bluetooth but it cannot connect to another slave-only device such as keyboards and other HC-06 modules.

The **HC-05 Bluetooth Module** adds wireless communication to your project to communicate via Bluetooth to any Bluetooth enabled Laptop or Mobile Device. The module communicates at 9600 baud rate via USART protocol. It can be used in applications like communication between two microcontrollers, data logging, wireless robots, wireless sensors data acquisition and home automation.

The module has two operating modes Data mode and Command Mode. To connect it with your Arduino or Raspberry Pi connect the TX and RX pins to the controller and power the device with VCC pin connected to 5V pin. While in data mode it can send or receive the data from other Bluetooth devices and in command mode it will change default device settings.

Applications

- Industrial Applications
- GPS receiver
- Computer and portable Devices
- Embedded Projects

4.8 ESP32 CAMERA:

The ESP32-CAM can be widely used in intelligent IoT applications such as **wireless video monitoring, WiFi image upload, QR identification**, and so on. The ESP32-CAM suit for IOT applications such as: Smart home devices image upload. Wireless monitoring.

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.

The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and an ESP32-CAM-MB micro usb to serial port adapter.

The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, WiFi image upload, QR identification, and so on.

There are three GND pins and two pins for power: either 3.3V or 5V.

GPIO 1 and GPIO 3 are the serial pins. You need these pins to upload code to your board. Additionally, GPIO 0 also plays an important role, since it determines whether the ESP32 is in flashing mode or not. When GPIO 0 is connected to GND, the ESP32 is in flashing mode.

FEATURES:

- Onboard ESP32-S module, supports WiFi + Bluetooth
- OV2640 camera with flash
- Onboard TF card slot, supports up to 4G TF card for data storage
- Supports WiFi video monitoring and WiFi image upload
- Supports multi sleep modes, deep sleep current as low as 6mA
- Control interface is accessible via pinheader, easy to be integrated and embedded into user products

Specifications :

1. Wireless Module: ESP32-S WiFi 802.11 b/g/n + Bluetooth 4.2 LE module with PCB antenna, u.FL connector, 32Mbit SPI flash, 4MBit PSRAM.
2. External Storage: micro SD card slot up to 4GB.
3. Camera
   - FPC connector.
   - Support for OV2640 (sold with a board) or OV7670 cameras.
   - Image Format: JPEG(OV2640 support only), BMP, grayscale.
   - LED flashlight.
4. Expansion: 16x through-holes with UART, SPI, I2C, PWM.
5. Misc: Reset button.
6. Power Supply: 5V via pin header.
7. Power Consumption.

   1. Flash LED off: 180mA @ 5V.
   2. Flash LED on to maximum brightness: 310mA @ 5V.
   3. Deep-sleep: 6mA @ 5V min.
   4. Modem-sleep: 20mA @ 5V min.
   5. Light-sleep: 6.7mA @ 5V min

4.9 INTERNET OF THINGS[IOT]:

The internet of things, or IoT, is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. IoT devices

are typically embedded with technology such as sensors and software and can include mechanical and digital machines and consumer objects.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, deliver enhanced customer service, improve decision-making and increase the value of the business.

With IoT, data is transferable over a network without requiring human-to-human or human-to-computer interactions.

A *thing* in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low, or any other natural or man-made object that can be assigned an Internet Protocol address and is able to transfer data over a network.

Key components of IoT include:

1. **Devices/Sensors**: These are physical objects embedded with sensors, actuators, and other technologies to collect and transmit data.
2. **Connectivity**: IoT devices are connected to the internet or other networks, allowing them to communicate with each other and with central systems.
3. **Data Processing**: The data collected by IoT devices is processed and analyzed to derive insights, make decisions, and trigger actions.
4. **Applications/Use Cases**: IoT technology is applied across various domains such as smart homes, healthcare, agriculture, manufacturing, transportation, and more.
5. **Security**: Securing IoT devices and networks is crucial to prevent data breaches, privacy violations, and unauthorized access.
6. **Interoperability**: Ensuring compatibility and seamless communication between different IoT devices and systems is important for their effective functioning.

IoT has the potential to revolutionize industries, improve efficiency, and enhance quality of life. However, it also raises concerns regarding privacy, security, data ownership, and ethical implications. As IoT continues to evolve, addressing these challenges will be essential to realizing its full potential.

IoT enables these smart devices to communicate with each other and with other internet-enabled devices. Like smartphones and gateways, creating a vast network of interconnected devices that can exchange data and perform various tasks autonomously. This can include:

- monitoring environmental conditions in farms
- managing traffic patterns with smart cars and other smart automotive devices
- controlling machines and processes in factories
- tracking inventory and shipments in warehouses

The potential applications of IoT are vast and varied, and its impact is already being felt across a wide range of industries, including manufacturing, transportation, healthcare, and agriculture. As the number of internet-connected devices continues to grow, IoT is likely to play an increasingly important role in shaping our world. Transforming the way that we live, work, and interact with each other.

Three types of IOT:

Billions of devices are connected to the internet, collecting and sharing information with one another. They range from smart home setups like cooking appliances and smoke detectors to military-grade surveillance equipment. The list below outlines a few of the most common types of IoT applications.

1. Consumer IoT

Consumer IoT refers to personal and wearable devices that connect to the internet. These devices are often referred to as smart devices.

2. Industrial Internet of Things (IIoT)

The industrial Internet of Things is the system of interconnected devices in the industrial sector. Manufacturing machinery and devices used for energy management are a part of the industrial Internet of Things.

3. Commercial IoT

Commercial IoT refers to the tools and systems used outside of the home. For example, businesses and health care organizations leverage commercial IoT for auditable data trails and consumer management.

How does IOT works?

The next few sections break down the components that make the Internet of Things work.

Internet of Things platform

An IoT platform manages device connectivity. It can be a software suite or a cloud service. The purpose of an IoT platform is to manage and monitor hardware, software, processing abilities, and application layers.

Sensor technologies

IoT sensors, sometimes called smart sensors, convert real-world variables into data that devices can interpret and share. Many different types of sensors exist. For example, temperature sensors detect heat and convert temperature changes into data. Motion sensors detect movement by monitoring ultrasonic waves and triggering a desired action when those waves are interrupted.

The core concept of the IoT is communication among devices and users. Unique identifiers (UIDs) establish the context of a device within the larger network to enable this communication. Identifiers are patterns, like numeric or alphanumeric strings. One example of a UID that you might be familiar with is an internet protocol (IP) address. They can identify a single device (instance identifier) or the class to which that device belongs (type identifier).

Internet connectivity

Sensors can connect to cloud platforms and other devices through a host of network protocols for the internet. This enables communication between devices.

Types of Sensors:

- Temperature Sensors
- Image Sensors
- Gyro Sensors
- Obstacle Sensors
- RF Sensor
- IR Sensor
- MQ-02/05 Gas Sensor
- LDR Sensor
- Ultrasonic Distance Sensor

Control Units: It is a unit of small computer on a single integrated circuit containing microprocessor or processing core, memory and programmable input/output devices/peripherals. It is responsible for major processing work of IoT devices and all logical operations are carried out here.

Cloud computing: Data collected through IoT devices is massive, and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.

Availability of big data: We know that IoT relies heavily on sensors, especially in real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.

Networking connection: In order to communicate, internet connectivity is a must, where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

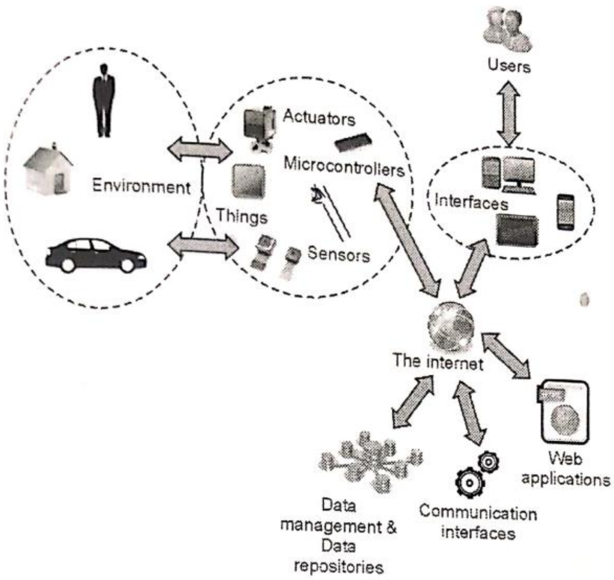
Working with IoT Devices

- **Collect and Transmit Data :** For this purpose sensors are widely used they are used as per requirements in different application areas.

- **Actuate device based on triggers produced by sensors or processing devices:** If certain conditions are satisfied or according to user's requirements if certain trigger is activated then which action to perform that is shown by Actuator devices.

- **Receive Information:** From network devices, users or devices can take certain information also for their analysis and processing purposes.

- **Communication Assistance:** Communication assistance is the phenomenon of communication between 2 networks or communication between 2 or more IoT devices of same or different networks. This can be achieved by different communication protocols like: MQTT, Constrained Application Protocol, ZigBee, FTP, HTTP etc.



Characteristics of IOT:

- Massively scalable and efficient
- IP-based addressing will no longer be suitable in the upcoming future.
- An abundance of physical objects is present that do not use IP, so IoT is made possible.
- Devices typically consume less power. When not in use, they should be automatically programmed to sleep.
- A device that is connected to another device right now may not be connected in another instant of time.
- Intermittent connectivity – IoT devices aren't always connected. In order to save bandwidth and battery consumption, devices will be powered off periodically when not in use. Otherwise, connections might turn unreliable and thus prove to be inefficient.

IoT applications span across various domains, including:

- **Smart Home**: IoT devices such as smart thermostats, lighting systems, security cameras, and appliances enable homeowners to monitor and control their home environment remotely for energy efficiency, convenience, and security.
- **Industrial IoT (IoT)**: In industrial settings, IoT technology is used to optimize processes, monitor equipment health, improve safety, and enable predictive maintenance in sectors like manufacturing, logistics, agriculture, and utilities.
- **Healthcare**: IoT devices like wearable fitness trackers, remote patient monitoring systems, and smart medical devices allow for continuous health monitoring, personalized treatment, and early detection of health issues.
- **Smart Cities**: IoT is employed in urban infrastructure to manage traffic flow, optimize waste management, monitor air and water quality, enhance public safety through smart surveillance, and improve the overall quality of life for residents.

4.91 ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is an open-source software platform used for programming Arduino microcontroller-based devices. It provides an intuitive interface and a set of tools for writing, compiling, and uploading code to Arduino boards. Here's a breakdown of its key features and functions:

1. **Code Editor**: The IDE includes a text editor where users write their Arduino sketches (programs). It supports syntax highlighting, auto-completion, and code formatting to make coding easier and more efficient.
2. **Sketch Management**: Arduino programs are called sketches, and the IDE organizes them within a project structure. Users can create, open, save, and manage multiple sketches within the IDE.
3. **Compilation**: The IDE integrates the Arduino compiler, which translates the human-readable Arduino code (written in C/C++) into machine-readable instructions that the Arduino board can understand. Users can compile their sketches directly within the IDE.
4. **Serial Monitor**: The IDE includes a built-in serial monitor tool that allows users to communicate with the Arduino board over a serial connection. It displays messages sent from the board and allows users to send commands or data to the board for debugging or interaction purposes.

5. **Board Manager**: Arduino supports a variety of microcontroller boards with different architectures and specifications. The IDE includes a board manager tool that simplifies the process of installing board definitions for different Arduino-compatible boards. Users can select their target board from a list and configure settings such as processor speed and port.
6. **Library Manager**: Arduino libraries are collections of pre-written code that provide additional functionality and features. The IDE includes a library manager tool that allows users to easily install, update, and manage libraries from the Arduino Library Manager repository.
7. **Upload**: Once a sketch is written and compiled, users can upload it to the Arduino board using the IDE. The IDE handles the process of transferring the compiled code to the board via a USB connection or other communication interfaces.
8. **Debugging**: While the Arduino IDE does not have full-fledged debugging capabilities like traditional IDEs for desktop programming languages, users can still implement debugging techniques using print statements and the serial monitor for basic troubleshooting and error detection.

The Arduino IDE (Integrated Development Environment) is software used to program Arduino boards. It provides a simple interface for writing, compiling, and uploading code to Arduino microcontrollers. Here's a basic overview of the Arduino IDE:

1. **Download and Installation**: You can download the Arduino IDE from the official Arduino website (https://www.arduino.cc/en/software). It's available for Windows, Mac OS X, and Linux.
2. **Writing Code**: The Arduino IDE uses a simplified version of C/C++ programming language. You write your code in the editor window. The basic structure of an Arduino sketch consists of two functions: `setup()` and `loop()`. The `setup()` function is called once when the board boots up, while the `loop()` function runs continuously after `setup()` finishes.
3. **Verify/Compile**: Once you've written your code, you can click the verify button (a checkmark icon) to compile it. This step checks your code for any syntax errors and compiles it into machine code that the Arduino board can understand.

4. **Upload**: After verifying, you can upload the compiled code to your Arduino board by clicking the upload button (a right arrow icon). This step sends the compiled code to the Arduino board via a USB connection.
5. **Serial Monitor**: The Arduino IDE includes a serial monitor tool that allows you to communicate with your Arduino board via the serial port. This is useful for debugging and for sending/receiving data between your Arduino board and your computer.
6. **Library Manager**: Arduino IDE comes with a library manager that allows you to easily add external libraries to your projects. These libraries provide pre-written code for various functions and components, saving you time and effort.
7. **Board Manager**: You can select the type of Arduino board you're using from the Tools menu. The board manager allows you to install additional board definitions, enabling you to work with a wide range of Arduino-compatible boards.
8. **Preferences**: Arduino IDE also allows you to customize settings such as the editor theme, font size, and more through the preferences menu.

Overall, the Arduino IDE is a user-friendly tool that simplifies the process of programming and uploading code to Arduino boards, making it accessible even for beginners in electronics and programming.

The Arduino IDE (Integrated Development Environment) is a software application designed to facilitate the development of code for Arduino microcontroller boards. Here's a more detailed description of its features and functionalities:

1. **Cross-Platform Compatibility**: The Arduino IDE is available for Windows, macOS, and Linux operating systems, ensuring accessibility to a wide range of users regardless of their preferred platform.
2. **Code Editor**: The IDE provides a built-in code editor with features such as syntax highlighting, auto-indentation, and code completion, making it easier for developers to write and manage their Arduino sketches (programs).
3. **Sketch Management**: Arduino programs are referred to as sketches, and the IDE allows users to create, open, save, and manage multiple sketches simultaneously. This enables developers to work on different projects seamlessly.
4. **Compiler and Uploader**: The IDE includes a compiler that translates Arduino sketches written in the Arduino programming language (a simplified version of

C/C++) into machine code that can be executed by the Arduino board's microcontroller. Additionally, it provides an uploader tool to transfer the compiled code to the connected Arduino board via a USB connection.

5. **Serial Monitor**: The Arduino IDE features a built-in serial monitor tool that allows developers to send and receive data between their Arduino board and their computer over a serial connection. This is particularly useful for debugging purposes and for interacting with peripherals connected to the Arduino board.
6. **Library Manager**: Arduino libraries contain pre-written code that simplifies the development process by providing ready-to-use functions and modules for various tasks, such as interfacing with sensors, actuators, and communication protocols. The IDE includes a library manager that allows users to easily search for, install, and manage libraries within their projects.
7. **Board Manager**: Arduino boards come in various configurations and models, each requiring specific settings and configurations within the IDE. The board manager simplifies this process by providing a graphical interface for selecting the appropriate board type, processor, and other parameters, ensuring compatibility between the IDE and the connected Arduino board.
8. **Preferences and Settings**: The IDE allows users to customize various settings and preferences according to their preferences and requirements. This includes options for configuring the editor, setting default compiler options, managing serial port settings, and more.

Overall, the Arduino IDE serves as a comprehensive development environment for programming Arduino microcontroller boards, offering a range of features and tools to support both novice and experienced developers in their Arduino projects.

The Arduino IDE (Integrated Development Environment) possesses several characteristics that make it a popular choice for programming Arduino microcontroller boards:

1. **User-Friendly Interface**: The Arduino IDE features a simple and intuitive interface, making it accessible to beginners in electronics and programming. Its ease of use allows users to quickly get started with writing and uploading code to Arduino boards.
2. **Open Source**: The Arduino IDE is open-source software, which means its source code is freely available for modification and redistribution. This fosters a

collaborative development environment and allows for community contributions and improvements.

3. **Cross-Platform Compatibility**: The IDE is compatible with multiple operating systems, including Windows, macOS, and Linux, ensuring that users can develop Arduino projects regardless of their preferred platform.
4. **Built-In Code Editor**: The IDE includes a built-in code editor with features such as syntax highlighting, auto-indentation, and code completion, enhancing the coding experience and facilitating code readability and organization.
5. **Compiler and Uploader**: Arduino IDE integrates a compiler that translates Arduino sketches written in the Arduino programming language into machine code executable by Arduino boards. It also provides an uploader tool for transferring compiled code to connected Arduino boards via USB.
6. **Serial Monitor**: The IDE features a serial monitor tool that allows developers to communicate with their Arduino boards via the serial port. This enables real-time monitoring of sensor data, debugging, and interaction with external devices connected to the Arduino.
7. **Library Manager**: Arduino IDE includes a library manager that simplifies the process of adding external libraries to Arduino projects. These libraries provide pre-written code for various functionalities, such as sensor interfacing, communication protocols, and hardware control, reducing development time and effort.
8. **Board Manager**: The IDE's board manager facilitates the selection and configuration of Arduino boards, ensuring compatibility between the IDE and the connected hardware. Users can easily choose the appropriate board type, processor, and settings for their projects.
9. **Community Support**: The Arduino IDE benefits from a vibrant and active community of developers, enthusiasts, and educators who contribute tutorials, libraries, and troubleshooting resources. This extensive support ecosystem helps users overcome challenges and learn from others' experiences.
10. **Extensibility**: The Arduino IDE's modular architecture allows for extensions and customizations through plugins and additional tools. This flexibility enables users to tailor the IDE to their specific needs and integrate third-party components seamlessly.

Overall, the Arduino IDE's user-friendly design, comprehensive features, and strong community support make it an excellent choice for both beginners and experienced developers working on Arduino-based projects.

The Arduino IDE (Integrated Development Environment) is a versatile tool used in various applications across different domains. Here are some common applications of the Arduino IDE:

1. **Prototyping and Development**: The primary use of the Arduino IDE is for prototyping and developing electronic projects. It allows developers to quickly write, compile, and upload code to Arduino microcontroller boards, enabling rapid iteration and testing of hardware and software designs.
2. **DIY Electronics Projects**: Arduino IDE is widely used by hobbyists, makers, and DIY enthusiasts to create a wide range of electronic projects, including home automation systems, robotics projects, LED displays, interactive art installations, and more.
3. **Education and Learning**: Arduino IDE is popular in educational settings, including schools, universities, and workshops, as a tool for teaching electronics, programming, and physical computing concepts. Its user-friendly interface and extensive documentation make it accessible to learners of all ages and skill levels.
4. **Internet of Things (IoT) Development**: With the proliferation of IoT devices, the Arduino IDE is frequently used for developing IoT applications and prototypes. Developers can use Arduino boards equipped with Wi-Fi or Ethernet capabilities, along with sensors and actuators, to build IoT projects that connect to the internet and interact with cloud services.
5. **Sensor Data Logging and Monitoring**: Arduino IDE is often employed for collecting and logging sensor data from various environmental sensors, such as temperature, humidity, light, and motion sensors. This data can be monitored in real-time using the serial monitor tool or stored for later analysis.
6. **Automation and Control Systems**: Arduino IDE is utilized in automation and control systems for tasks such as controlling motors, servos, relays, and other actuators based on sensor inputs or predefined conditions. These systems find

applications in home automation, industrial automation, agricultural automation, and more.

7. **Wearable Technology**: Arduino IDE is used in the development of wearable electronics and e-textiles projects. Developers can create wearable devices such as fitness trackers, smart garments, and gesture-controlled accessories using Arduino boards and compatible sensors.
8. **Interactive Art and Installations**: Artists and designers often use Arduino IDE to create interactive artworks, installations, and exhibits that respond to user input or environmental stimuli. These projects combine electronics, programming, and creative expression to engage audiences in unique and immersive experiences.
9. **Scientific Research and Prototyping**: Arduino IDE is utilized in scientific research and prototyping across various disciplines, including physics, biology, environmental science, and engineering. Researchers can use Arduino boards and sensors to collect data, perform experiments, and prototype custom scientific instruments.
10. **Embedded Systems Development**: Arduino IDE can be used for developing embedded systems and custom electronic devices for specific applications. Developers can create custom firmware and software to control embedded hardware and interface with external peripherals.

Chapter 5

5.1 ADVANTAGES:

- ✓ Improve security and safety.
- ✓ Use less energy.
- ✓ Create convenience.
- ✓ Use appliances more effectively.
- ✓ Increase home comfort.
- ✓ Gain insights about your home.
- ✓ Enhance accessibility.

5.2 DISADVANTAGES:

- ✓ Security Issues
- ✓ Cost: Extremely expensive.
- ✓ Greater acceptance
- ✓ Dependency on internet

CHAPTER 6

CONCLUSION and FUTURE POSSIBILITY

The home automation using Internet of Things has been experimentally proven to work satisfactorily by connecting simple appliances to it and the appliances were successfully controlled remotely through internet.

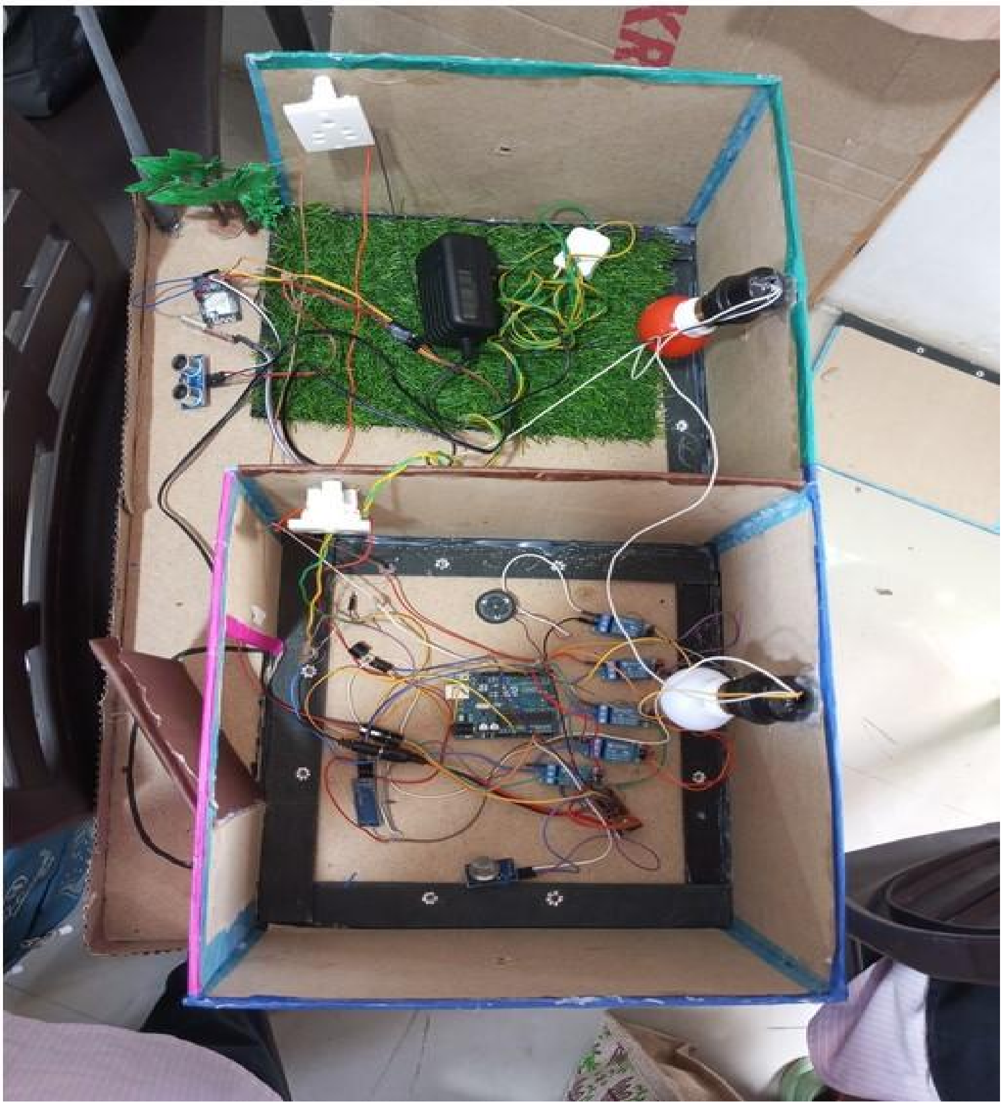
Home automation **makes life more convenient and can even save you money on heating, cooling and electricity bills**. Home automation can also lead to greater safety with Internet of Things devices like security cameras and systems. Home automation makes it possible to **automate tasks related to security, well-being, and comfort through a smart system installed in a home or building**. In other words, it integrates technology into the design of a space. One of the main advantages of home automation systems is energy efficiency.

In this work, a real-time home automation system has been successfully implemented which is quite effective in terms of performance and technology. Home automation systems are not so popular yet. But, there is a very large probability that it will be the trend in future. The log file been generated as a result of changing state of appliances. It can be used to track the user’s behavior i.e. the time of controlling the appliances. Using this log file, we can apply Machine Learning to the system, through which the system will learn how the user operates the appliances in his house.

Chapter 7

RESULT AND ANALYSIS

The process of this smart monitoring approach Here first activity allows the user to scan a QR code to get the server information since each user has a unique dashboard on the server. The second activity asks the user to connect to the MQTT broker and the third activity shows all the data provided by the broker. We can control the appliances from the MQTT dashboard. shows that the light is On.



Chapter 8

REFERENCE CODE

```
int gassensor = A0;

int bs = 0;

int buzzer = 7;

String data;

int relay1 =2;

int relay2 =3;

int relay3 =4;

int relay4 =5;

int relay5 =6;

void setup()

{

  Serial.begin(9600);

  pinMode(relay1,OUTPUT);

  pinMode(relay2,OUTPUT);

  pinMode(relay3,OUTPUT);

  pinMode(relay4,OUTPUT);

  pinMode(relay5,OUTPUT);

  pinMode(buzzer,OUTPUT);

  pinMode(gassensor,INPUT);

  digitalWrite(relay1,HIGH);

  digitalWrite(relay2,HIGH);

  digitalWrite(relay3,HIGH);

  digitalWrite(relay4,HIGH);
```

```
 digitalWrite(relay5,HIGH);
}
void loop()
{
 bs = analogRead(gassensor);
 Serial.print("value = ");
 Serial.println(bs);
 delay(500);
 if(bs > 500)
  {
   Serial.println("smoke detected");
   digitalWrite(buzzer,HIGH);
  }
  else
  {
   Serial.println("smoke not detected");
   digitalWrite(buzzer,LOW);
  }
 if(Serial.available()>0)
 {
  data = Serial.readString();
  Serial.println(data);
  data.trim();
  delay(500);
  if(data == "device1 on")
```

```
{
  digitalWrite(relay1,LOW);
 Serial.println("device1 on");
}
if(data == "device2 on")
{
  digitalWrite(relay2,LOW);
  Serial.println("device2 on");


}
 if(data == "device3 on")
{
  digitalWrite(relay3,LOW);
  Serial.println("device3 on");
}
if(data == "device4 on")
{
 digitalWrite(relay4,LOW);
 Serial.println("device4 on");
}
if(data == "device5 on")
{
 digitalWrite(relay5,LOW);
 Serial.println("device5 on");
}
```

```
  if(data == "device1 off")
 {
  digitalWrite(relay1,HIGH);


 }
 if(data == "device2 off")
 {
  digitalWrite(relay2,HIGH);


 }
  if(data == "device3 off")
 {
  digitalWrite(relay3,HIGH);
 }
 if(data == "device4 off")
 {
  digitalWrite(relay4,HIGH);
 }
  if(data == "device5 off")
 {
  digitalWrite(relay5,HIGH);
 }
```

Chapter9

REFERENCES

[1] R. S. Ransing and M. Rajput, "Smart home for elderly care, based Wireless Sensor Network," 2015 International Conference on NascentTechnologies the Engineering Field (ICNTE), Navi Mumbai, 2015

[2] S. Kumar and S. R. Lee, "Android based smart home system with control via Bluetooth and internet connectivity," The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), JeJu Island, 2014

[3] Young-Guk Ha, "Dynamic integration of Zigbee home networks into home gateways using OSGI service registry," IEEE Transactions on Consumer Electronics., vol. 55, Issue. 2, pp.470-476, May. 2009.

[4] J. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges," Proc. of the Workshop on High Confidence Medical Devices Software and Systems, Jun 2005.

[5] H. Noguchi, T. Mori and T. Sato, "Human daily behavior accumulation in real home environment via distributed sensors in a long term," 2011 IEEE/SICE International Symposium on System Integration (SII), Kyoto, 2011, pp. 368-374.

[6] S. M. Brundha, P. Lakshmi and S. Santhanalakshmi, "Home automation in client-server approach with user notification along with efficient security alerting system," 2017 International Conference on Smart Technologies forSmart Nation (SmartTechCon), Bangalore, 2017, pp. 596-601.

[7] N. Vikram, K. S. Harish, M. S. Nihaal, R. Umesh, A. Shetty and A. Kumar, "A Low Cost Home Automation System Using Wi-Fi Based Wireless Sensor Network Incorporating Internet of Things (IoT)," 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, 2017, pp. 174-178.

[8] M. Tharaniya soundhari and S. Brilly Sangeetha, "Intelligent interface based speech recognition for home automation using android application," 2015 International Conference

on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-11.

[9] C. Chiu-Chiao, H. C. Yuan, W. Shiau-Chin, and L. Cheng-Min, "Bluetooth-Based Android Interactive Applications for Smart Living," in 2nd International Conference on Innovations in Bioinspired Computing and Applications (IBICA 2011), 2011, pp. 309-312.

[10] A. Fleury, M. Vacher and N. Noury, "SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results," in IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 2, pp. 274-283, March 2010.

[11] L. Yang, M. Ji, Z. Gao, W. Zhang and T. Guo, "Design of Home Automation System Based on Zigbee Wireless Sensor Network," 2009 First International Conference on Information Science and Engineering, Nanjing 2009, pp. 2610-261