CHAPTER THREE

3.0 METHODOLOGY

3.1 The stages of home automation project

The methodology for this home automation project encompasses four main stages:

1. System Design and Planning
2. Software Development
3. Integration and Testing
4. Security Implementation

System Design and Planning

I. Requirements Gathering: Identify hardware components (Arduino UNO, Bluetooth module, relay module, and power supply) and determine the necessary software (Arduino IDE, Bluetooth control app) to meet the project's objectives.

II. Circuit Design: Design the electrical circuit to connect the Arduino UNO, Bluetooth module, relay modules, and household appliances. The relay modules are used to switch appliances on and off based on commands received by the Arduino.

III. Component Layout: Plan the physical layout of the components to ensure safe wiring, minimize electrical interference, and make the system compact and manageable.

Software Development

I. Arduino Programming: Write code in the Arduino IDE to control the relays based on Bluetooth commands received from a mobile device.

The code will: - Enable the Bluetooth module to receive commands. - Interpret commands (e.g., "ON" or "OFF") for specific appliances. - Activate or deactivate the connected relays based on the received commands.

II. Mobile Application Configuration: - Use a pre-existing Bluetooth control app, or configure a simple custom interface to send commands from the user's mobile device to the Bluetooth module. - Assign unique commands for each appliance and create buttons or control options for user- friendly interaction.

Integration and Testing

I. Hardware Integration: Connect the Arduino, Bluetooth module, relay modules, and appliances according to the designed circuit. Ensure all connections are secure and verify the operation of each relay.

II. System Testing: - Test the connection between the mobile device and the Bluetooth module to ensure reliable pairing and data transmission. - Test each appliance control feature individually to ensure the correct relay activates with the corresponding Bluetooth command. - Perform range tests to confirm the system operates effectively within typical Bluetooth limits (approximately 10 meters).

III. Debugging: Identify and fix any issues related to connection stability, command accuracy, or relay operation.

Security Implementation

I. Bluetooth Pairing Security: Configure the Bluetooth module with a unique pairing PIN to restrict system access to authorized devices only.

II. User Authentication: Implement basic authentication within the app if possible, ensuring only authorized users can control the appliances. This could involve a simple password prompt or device pairing verification.

III. Documentation and User Training User Documentation: Create a user manual with step-by-step instructions for system setup, connection, and troubleshooting.

IV. Safety Guidelines: Include safety instructions for handling the hardware, especially with appliances connected to high-voltage power sources.

V. End-User Training: Provide guidelines on how to use the mobile application, pair the device, and interpret system indicators (e.g., LEDs for on/off status).

Evaluation and Final Adjustments

I. User Feedback: Collect feedback from users (or testers) to identify any usability or performance issues.

II. Adjustments and Optimization: Make any final adjustments to the code, circuit, or app configuration based on feedback to enhance system performance and user satisfaction.

3.2 Diagram Of The Home Automation System using Arduino UNO and Bluetooth.

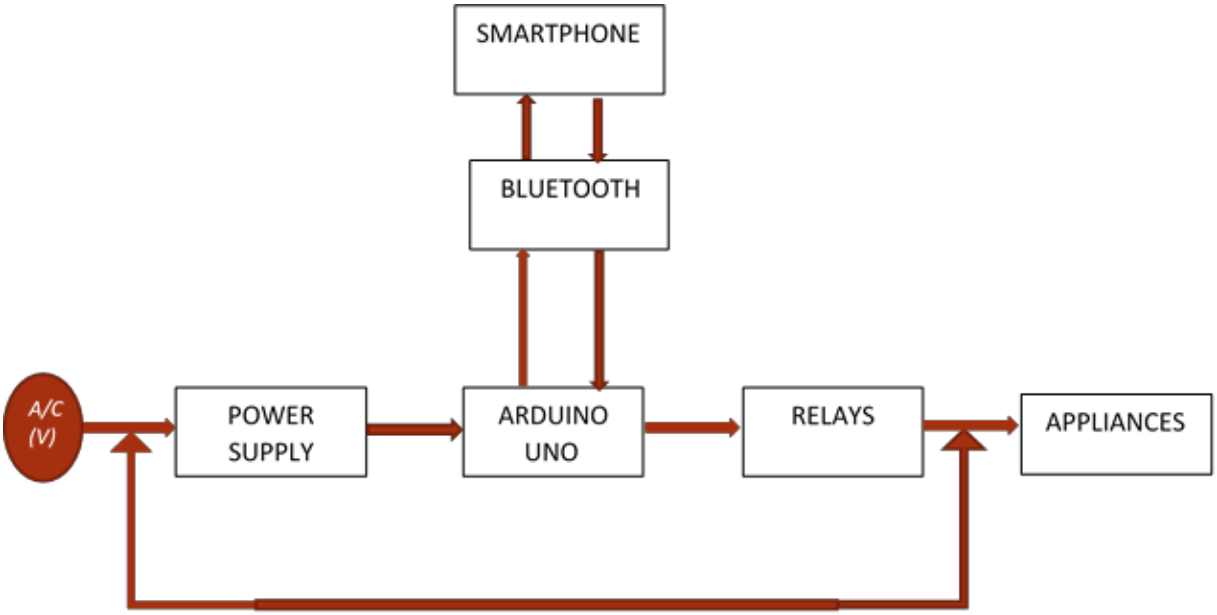

Figure 3 the block diagram of a home automation system.

3.2.1 The power supply

The design is powered by one of rechargeable 18650 battery. The battery is rated 7AH, 12V. This implies that the battery will supply a total of: 12 $volts$ and 7AH capacity. The voltage levels required by the circuits are 5$volts$ and12$volts$; all the inner circuits except the pump is powered by a 5v but the pump is powered with 12v. In other to get 5 volts from 12 volts, a buck converter is used. A buck converter (step down converter) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor) and an inductor. Its name derives from the inductor that “bucks” or opposes the supply voltage.

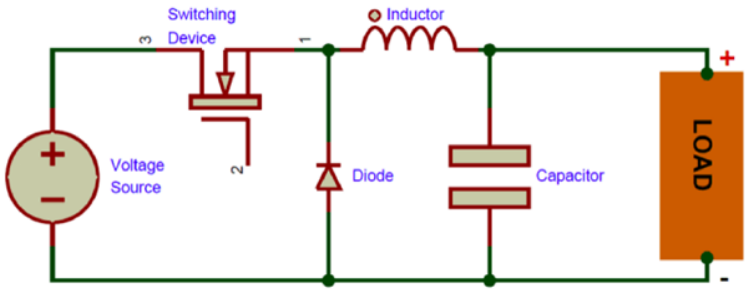

Figure 4 the buck converter circuit

In comparison to linear regulators, which are more straightforward circuits that lower voltages by dissipating power as heat but do not step up output current, switching converters (such as buck converters) offer a far higher level of power efficiency as DC-to-DC converters. Buck converters are essential for jobs like converting a computer's primary supply power down to lesser voltages needed because of their great efficiency, which is frequently over 90%.

Switching frequencies used by buck converters typically range from 100 kHz to a few MHz Smaller inductors and capacitors can be used with greater switching frequencies, but the efficiency is lost more frequently due to transistor switching.

3.2.2 Arduino UNO

Arduino UNO is based on ATmega328P Microcontroller, an 8-bit AVR Architecture based MCU from ATMEL. Arduino UNO comes in two variants: one consists of a 28-pin DIP Microcontroller while the other consists of 32 lead Quad Flat Package Microcontroller.

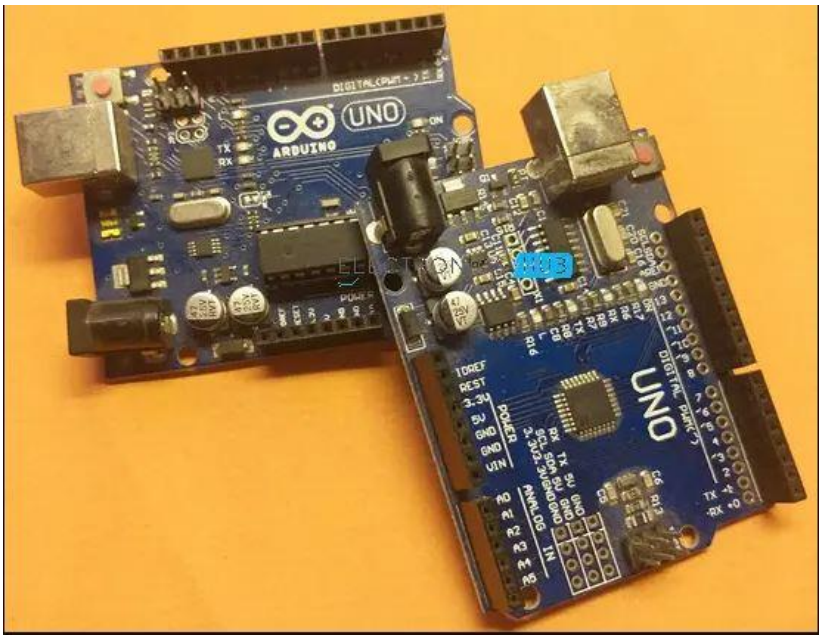

Figure 5 The Arduino UNO Board for but SMD and Thresiode

3.2.3 Arduino UNO Board Layout

The following image shows the layout of a typical Arduino UNO board. All the components are placed on the top side of the PCB.

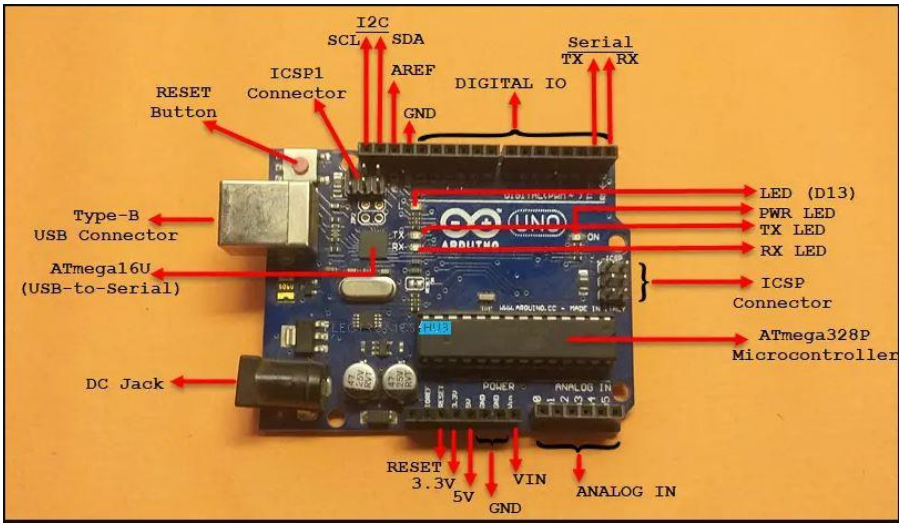

Figure 6 The pinout diagram of Arduino UNO board

As you can notice, there is a Type-B USB connector on the left short edge of the board, which is used for powering on the board as well as programming the Microcontroller. There is also a 2.1 mm DC jack to provide external power supply.

3.2.3.1 Technical Specifications of Arduino UNO

MCU	ATmega328P
Architecture	AVR
Operating Voltage	5V
Input Voltage	6V – 20V (limit) 7V – 12V (recommended)
Clock Speed	16 MHz
Flash Memory	32 KB (2 KB of this used by bootloader)

SRAM	2 KB
EEPROM	1 KB
Digital IO Pins	24 (of which 6 can produce PWM)
Analog Input Pins	6

As Arduino UNO is based on ATmega328P Microcontroller, the technical specifications of Arduino UNO are mostly related to the ATmega328P MCU. But none the less, let me give you a brief overview about some important specifications of Arduino UNO.

3.2.3.2 How to power up the Arduino UNO?

There are a couple of ways in which you can power the UNO board. The first and easy way is using the Type-B USB Connector. The next way is to provide an unregulated supply in the range of 6V to 20V to VIN pin of the UNO (Pin number 26).

You can also supply the unregulated supply through the 2.1mm DC Jack, in which case, you can access the supplied voltage through the VIN Pin.

3.2.3.3 What are Different Memories of Arduino UNO?

Strictly speaking, this is specific to the MCU i.e., ATmega328P, used on the Arduino UNO Board. There are three different memories available in ATmega328P. They are:

1. 32 KB of Flash Memory

2. 2 KB of SRAM

3. 1 KB of EEPROM

4. 0.5 KB of the Flash Memory is used by the bootloader code.

3.2.3.4 The Input and Output Pins of Arduino UNO

The 32 pins available on the UNO board, 22 pins are associated with input and output. In that 14 pins (D0 to D13) are true digital IO pins, which can be configured as per you application using pinMode(), digitalWrite() and digitalRead() functions.

All these Digital IO pins are capable of sourcing or sinking 20mA of current (maximum 40mA is allowed). An additional feature of the Digital IO pins is the availability of internal pull-up resistor (which is not connected by default).

The value of the internal pull-up resistor will be in the range of 20KΩ to 50KΩ.

There are also 6 Analog Input Pins (A0 to A5). All the analog input pins provide a 10-bit resolution ADC feature, which can be read using analogRead() function.

An important point about Analog Input pins is that they can be configured as Digital IO pins, if required.

Digital IO pins 3, 5, 6, 9, 10 and 11 are capable of producing 8-bit PWM Signals. You can use analogWrite() function for this.

3.2.3.5 Communication Interfaces available on Arduino UNO

Arduino UNO supports three different types of communication interfaces. They are:

1. Serial
2. I2C or I^2C
3. SPI

Perhaps the most common communication interface in the Arduino universe is the Serial Communication. In fact, the Arduino boards (UNO or Nano or Mega) are programmed using the serial communication.

Digital IO pins 0 and 1 are used as Serial RX and TX pins to receive and transmit serial data. These pins are connected to the serial pins of the on-board USB to Serial Converter IC.

Analog Input Pins A4 and A5 have alternative functions. They can be configured as SDA (A4) and SCL (A5) to support I2C or I^2C or Two Wire Interface (TWI) communication.

The final communication interface is the SPI. Digital IO Pins 10, 11 12 and 13 can be configured as SPI pins SS, MOSI, MISO and SCK respectively.

3.2.3.5 Additional features

There is an on-board LED connected to digital IO pin 13. Use this LED to perform Blinky operations. The reference voltage for the internal ADC is by default set to 5V. But using the AREF

pin, you can manually set the upper limit of the ADC. Using the IOREF pin, you can set the reference voltage for Microcontroller operations.

To reset the microcontroller, you can use the on-board RESET button. Although you can program the Arduino UNO using the USB cable, there is a provision to program the MCU using the In-Circuit Serial Programming (ICSP) interface. The UART bootloader, which is preloaded in to the ATmega328P microcontroller, enables programming through serial interface. But ICSP doesn't need any bootloader. You can program Arduino UNO using ISCP or use the ISCP of Arduino UNO to program other Arduino Boards.

Digital IO Pins 2 and 3 can be configured as External Interrupts Pins INT0 and INT1 respectively. Use attachInterrupt() function to configure the Interrupt for rising edge, falling edge or level change on the pin.

3.2.3.6 Arduino UNO Pinout

Now that we have seen a little bit about Arduino UNO and its important features and specifications, let us dive into the Arduino UNO Pinout. The following image shows the complete pinout of Arduino UNO Board.

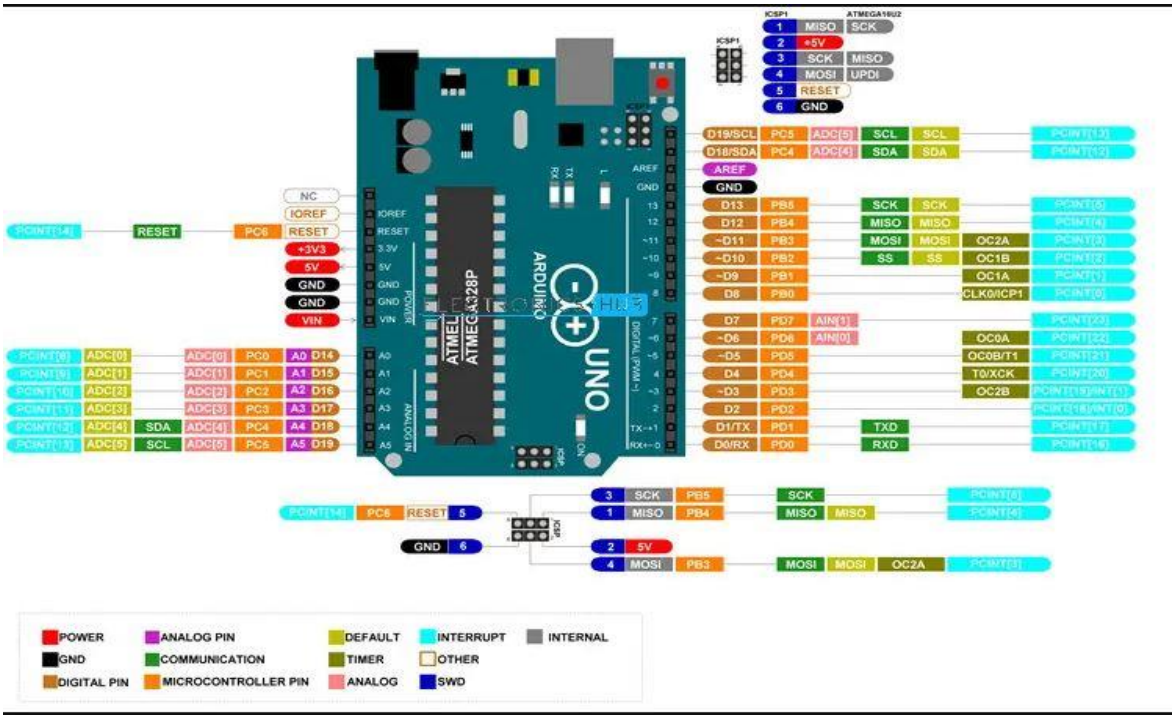

Figure 7 Arduino UNO pinout diagram

As you can see from the image, I described each pin of the Arduino UNO with its microcontroller equivalent pin, alternative functions, default functionality and other additional features.

3.2.3.7 Pin Description

For pin description of Arduino UNO, let us assume some basic numbering. Let the numbering begin with the RX Pin (D0). So, RX is Pin 1, TX is Pin 2, D2 is Pin 3 and so on. On the other side, NC is Pin 19, IOREF is Pin 20 etc. Overall, there are 32 pins on the Arduino UNO Board.

Pin Number	Pin Name	Description	Alternative Functions

1	RX / D0	Digital IO Pin 0 Serial RX Pin	Generally used as RX
2	TX / D1	Digital IO Pin 1 Serial TX Pin	Generally used as TX
3	D2	Digital IO Pin 2	
4	D3	Digital IO Pin 3	Timer (OC2B)
5	D4	Digital IO Pin 4	Timer (T0/XCK)
6	D5	Digital IO Pin 5	Timer (OC0B/T1)
7	D6	Digital IO Pin 6	
8	D7	Digital IO Pin 7	
9	D8	Digital IO Pin 8	Timer (CLK0/ICP1)
10	D9	Digital IO Pin 9	Timer (OC1A)
11	D10	Digital IO Pin 10	Timer (OC1B)
12	D11	Digital IO Pin 11	SPI (MOSI) Timer (OC2A)
13	D12	Digital IO Pin 12	SPI (MISO)
14	D13	Digital IO Pin 13	SPI (SCK)

15	GND	Ground	
16	AREF	Analog Reference	
17	SDA / D18	Digital IO Pin 18	I2C Data Pin
18	SCL / D19	Digital IO Pin 19	I2C Clock Pin
19	NC	Not Connected	
20	IOREF	Voltage Reference	
21	RESET	Reset (Active LOW)	
22	3V3	Power	
23	5V	+5V Output from regulator or +5V regulated Input	
24	GND	Ground	
25	GND	Ground	
26	VIN	Unregulated Supply	
27	A0	Analog Input 0	Digital IO Pin 14
28	A1	Analog Input 1	Digital IO Pin 15
29	A2	Analog Input 2	Digital IO Pin 16
30	A3	Analog Input 3	Digital IO Pin 17

31	A4	Analog Input 4	Digital IO Pin 18 I2C (SDA)
32	A5	Analog Input 5	Digital IO Pin 19 I2C (SCL)

The following table describes the pins of the ICSP Connector.

MISO	Master In Slave Out (Input or Output)
5V	Supply
SCK	Clock (from Master to Slave)
MOSI	Master Out Slave In (Input or Output)
RESET	Reset (Active LOW)
GND	Ground

There is also a similar ICSP connector known as ICSP1 associated with the ATmega16U Microcontroller. For more information on this connector, take a look at the Arduino UNO Pinout diagram.

3.2.3 Relay Module

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, which in turn

connects or disconnects another circuit. Relays can be of different types like electromechanical, solid state. Electromechanical relays are frequently used. Let us see the internal parts of this relay before knowing about working of relay. Although many different types of relay were present, their working is same.

Every electromechanical relay consists of an consists of an

1. Electromagnet
2. Mechanically movable contact
3. Switching points and
4. Spring

Electromagnet is constructed by wounding a copper coil on a metal core. The two ends of the coil are connected to two pins of the relay as shown.

These two are used as DC supply pins.

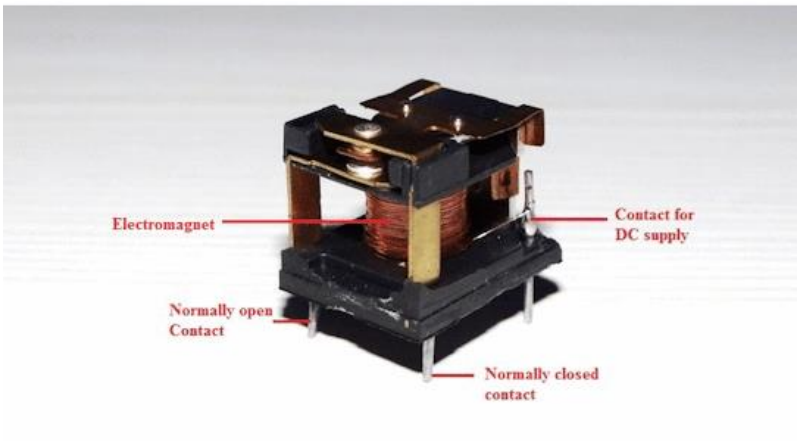

Figure 8 typical diagram of a relay module

Generally two more contacts will be present, called as switching points to connect high ampere load. Another contact called common contact is present in order to connect the switching points. These contacts are named as normally open (NO), normally closed(NC) and common(COM) contacts.

We can use a Relay either in a AC circuit or a DC Circuit. In case of AC relays, for every current zero position, the relay coil gets demagnetized and hence there would be a chance of continues breaking of the circuit.

So, AC relays are constructed with special mechanism such that continuous magnetism is provided in order to avoid above problem. Such mechanisms include electronic circuit arrangement or shaded coil mechanism.

Relay Working Principle?

The following animation shows how a Relay works.

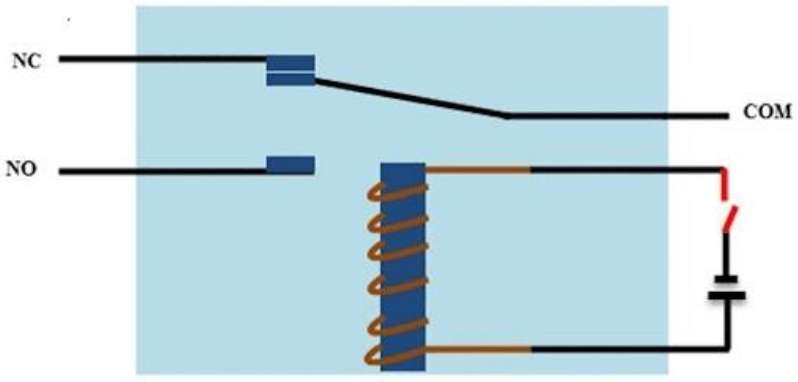

Figure 9 The Diagram of the Principle of Operation of Relay

1. When the electromagnet is applied with some current, it induces a magnetic field around it.
2. Above image shows working of the relay. A switch is used to apply DC current to the load.
3. In the relay, Copper coil and the iron core acts as electromagnet.
4. When the coil is applied with DC current, it starts attracting the contact as shown. This is called energizing of relay.
5. When the supply is removed it retrieves back to the original position. This is called De energizing of relay.

There are also such relays, whose contacts are initially closed and opened when there is supply i.e. exactly to opposite to the above shown relay.

Solid state relays will have sensing element to sense the input voltage and switches the output using opto-coupling.

Relay Applications

Relays are used to protect the electrical system and to minimize the damage to the equipment connected in the system due to over currents/voltages. The relay is used for the purpose of protection of the equipment connected with it.

These are used to control the high voltage circuit with low voltage signal in applications audio amplifiers and some types of modems.

These are used to control a high current circuit by a low current signal in the applications like starter solenoid in automobile. These can detect and isolate the faults that occurred in power transmission and distribution system. Typical application areas of the relays include

1. Lighting control systems
2. Telecommunication
3. Industrial process controllers
4. Traffic control
5. Motor drives control
6. Protection systems of electrical power system
7. Computer interfaces
8. Automotive
9. Home appliances

3.2.4 Bluetooth Module

The Bluetooth which is also known as HC-05 is a popular Bluetooth module which can add two-way (full-duplex) wireless functionality to any of our projects in a range of about 100m without any obstruction or barrier.

HC-05 Pinout Configuration

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module 1. Blink once in 2 sec: Module has entered Command Mode

		2. Repeated Blinking: Waiting for connection in Data Mode 3. Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

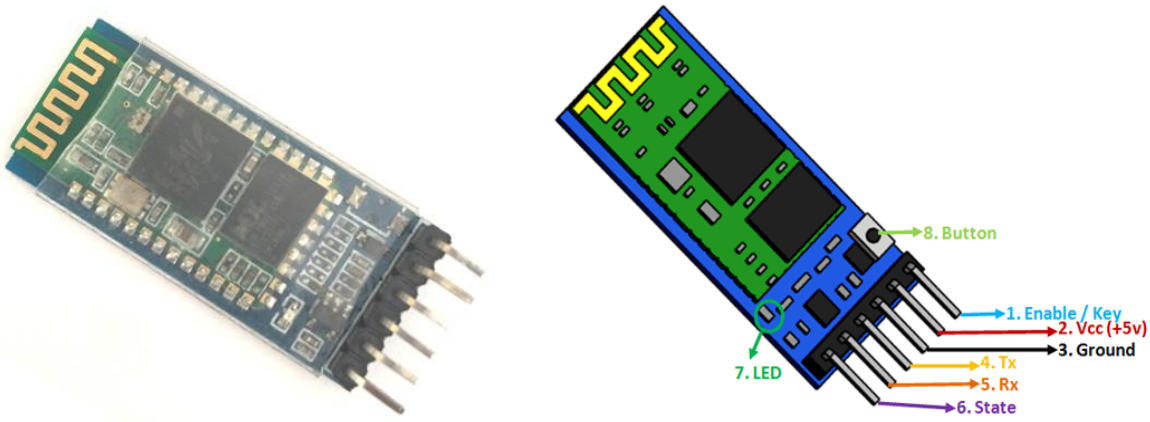

Figure 10 is a pictorial image of a Bluetooth module (HC-05).

Where to use HC-05 Bluetooth module

The **HC-05** is a popular module which can add two-way (full-duplex) wireless functionality to your projects. You can use this module to communicate between two microcontrollers like Arduino or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to

interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. So if you looking for a Wireless module that could transfer data from your computer or mobile phone to microcontroller or vice versa then this module might be the right choice for you. However do not expect this module to transfer multimedia like photos or songs; you might have to look into the CSR8645 module for that.

How to Use the HC-05 Bluetooth module

The **HC-05** has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in the figure below

During power up the key pin can be grounded to enter into Command mode, if left free it will by default enter into the data mode. As soon as the module is powered you should be able to discover the Bluetooth device as “HC-05” then connect with it using the default password 1234 and start communicating with it. The name password and other default parameters can be changed by entering into the

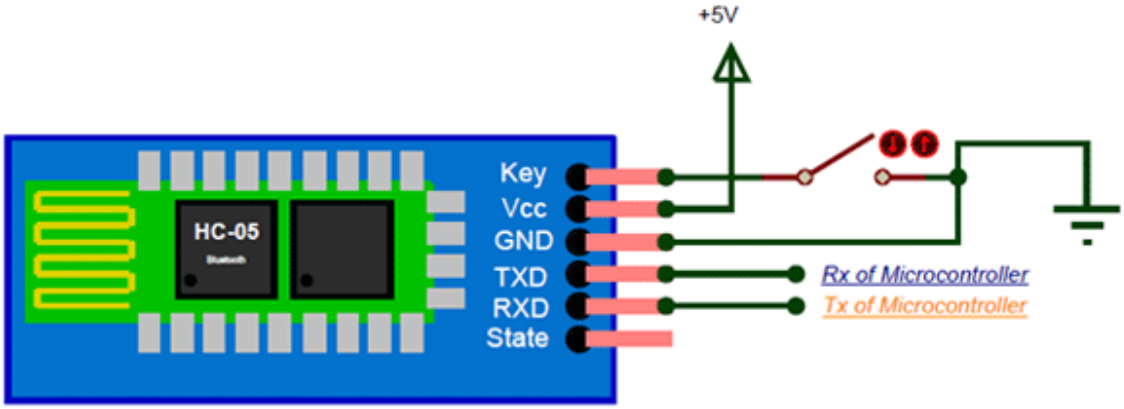

Figure 11 is the wiring diagram of Hc-05

Applications

1. Wireless communication between two microcontrollers

2. Communicate with Laptop, Desktops and mobile phones

3. Data Logging application

4. Consumer applications

5. Wireless Robots

6. Home Automation

3.3 The Code Implementation of the Home Automation

```
//HOME AUTOMATION SYSTEM USING ARDUINO UNO
//KWPOLY/2025/PROJECT
 char val; // Variable declaration


void setup() {//Initiating variables to pinout
```

```
pinMode(7,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
Serial.begin(9600);
 digitalWrite(7,HIGH);
 digitalWrite(12,HIGH);
 digitalWrite(11,HIGH);
 digitalWrite(10,HIGH);
 digitalWrite(9,LOW);
}
void loop() {
  if(Serial.available()){
 val = Serial.read();
 Serial.println(val);
}
if(val=='3'){
  digitalWrite(12,LOW);
}
else if(val=='4'){
  digitalWrite(12,HIGH);
```

```
}
if(val=='0'){
  digitalWrite(11,LOW);
}
else if(val=='6'){
  digitalWrite(11,HIGH);
}
if(val=='7'){
  digitalWrite(10,LOW);
}
else if(val=='8'){
  digitalWrite(10,HIGH);
}
if(val=='9'){
  digitalWrite(9,LOW);
}
else if(val=='5'){
  digitalWrite(9,HIGH);
}
if(val=='1'){
  digitalWrite(7,LOW);


}
```

```
else if(val=='2'){
  digitalWrite(7,HIGH);
}
delay(100);
}
```

3.4 CONSTRUCTION

The project was divided into three sections for construction: the Power Distribution Board (PDB), the Controller Board and the Farm prototype construction.

Figure 12 The Inner Construction of a Home Automation System

The board contains all the attached component including the power supply, relay, microcontroller to act as the brain.

3.4.1 CASING

The casing was constructed using a polyethelyn material which makes the module light in weight and makes the work neater and more professional. The boards are carefully arranged into its casing. The markings were done with a marker and ruler to make the cutting straight and neat.

Figure 13 The Home Automation System Casing