**DAYALBAGH EDUCATIONAL INSTITUTE**

**Dayalbagh , Agra**

**home automation**

**(An IoT Application Specially for Build Smart home)**

***Internship Project Report***



**Bachelor of Vocation (IOT)**

**Department of Physics & Computer Science**

**Faculty of Science**

**Dayalbagh Educational Institute**

**Dayalbagh, Agra**

**Declaration**

**This is to declare that this report has been written by me. In this report No part of the is plagiarized from other sources. All information included from other sources has been duly acknowledged.**

**By Submitted to –**

**Tinkal Shakya (1803776) Dr. R.S.Pavitra**

**Shubham Rathor (1803772)**

**Rohit Jain (1803763) Under –**

**Prashant Kumar (1803757) Amarjeet singh chauhan**

**Deepam Kumar (1803745)**

**Vicky (1803777)**

**Raja Babu (1803759)**

**Chandan Azad (1803742)**

**Sachin Kumar (1803764)**

**Abstract**

**The Main objective of this project is to develop a Home Automation System using an Arduino Board with NodeMCU 8266 being remotely controlled by any Android OS Smart phone. As technology is advancing so houses are also getting smarter.**

**Modern houses are gradually shifting from conventional switches to centralized control system. Presently, conventional wall switches located in different parts of the house make it difficult for the user to go near them to operate. Even more it become more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most solution with Smart phones.**

**In order to achieve this, a WiFi module NodeMCU 8266 is interfaced to the NodeMCU 8266 at the receiver end while on the transmitter end, a Blynk application on the Smart phone sends ON/OFF commands to the receiver where loads are connected, By touching the specified location on the Blynk , the loads can be turned ON/OFF remotely through this technology.**

**Introduction**

**Home automation*systems are advancement to the mechanization processes wherein human efforts are needed with the machinery equipments to operate various loads in*homes*.***

***Home automation system makes the operations of various home appliances more convenient and saves energy. With the energy saving concept, home automation or building automation makes life very simple nowadays. It involves automatic controlling of all electrical or electronic devices in homes or even remotely through***[***wireless communication***](https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/)***. Centralized control of lighting equipments, air conditioning and heating, audio/video systems, security systems, kitchen appliances and all other equipments used in home systems is possible with this system.***





**Smart Fan**

**Smart Light**

** Smart window**

**Smart Dustbin Smart Keypad Door Lock**

***All Lights , Fan and Window etc is connected to Smart Phone***



Machine, sensors & components—

* Fan & electronic machine and components
* Arduino
* Smart phone or laptop
* ESP8266 Node MCU
* Relay module
* **Keypad Door lock**
* Dustbin
* Bulb
* F to F, M to F & M to M Jumping wires
* Ultrasonic Sensor
* Servo Motor
* Stepper Motor etc.

**Description of Components**

**Fan & electronic machine and components /--**

****

**Fan**

****

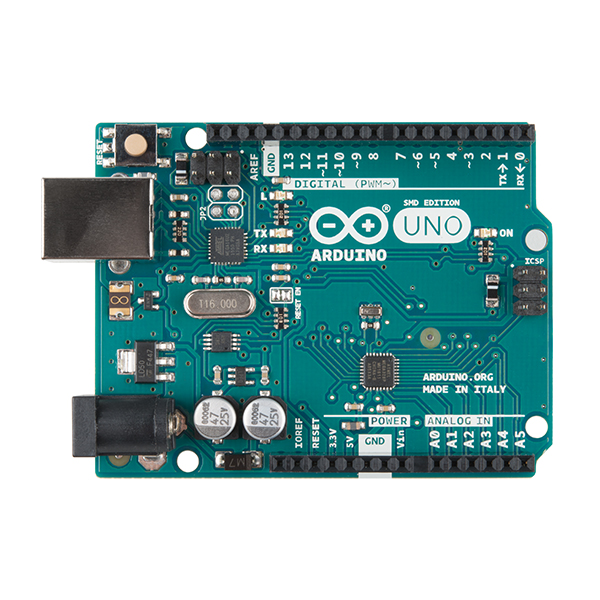
**BULB**

Arduino:-

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects**.**

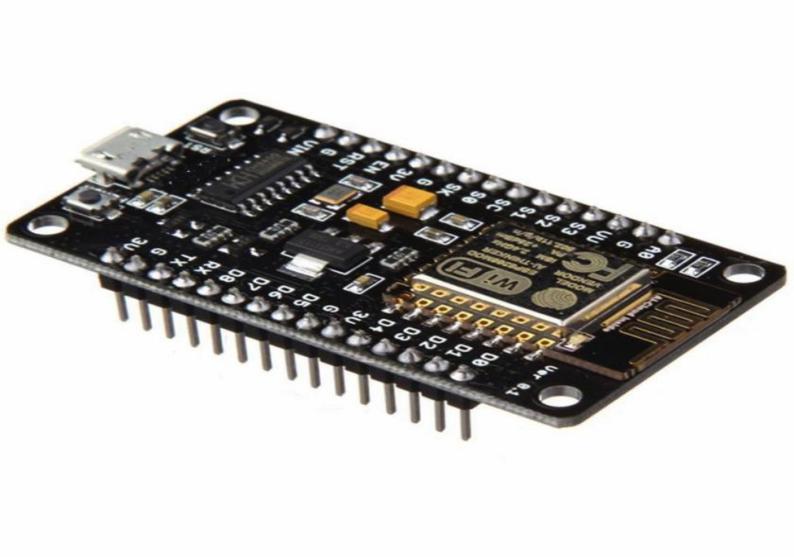
**Features of the Arduino UNO:**

* **Microcontroller: ATmega328.**
* **Operating Voltage: 5V.**
* **Input Voltage (recommended): 7-12V.**
* **Input Voltage (limits): 6-20V.**
* **Digital I/O Pins: 14 (of which 6 provide PWM output)**
* **Analog Input Pins: 6.**
* **DC Current per I/O Pin: 40 mA.**
* **DC Current for 3.3V Pin: 50 mA.**



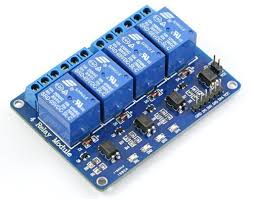
ESP8266 Node MCU :-

**NodeMCU is an open source**[IoT](https://en.wikipedia.org/wiki/Internet_of_Things)**platform.**[[4]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-nodemcu_firmware-4)[[5]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-5)**It includes**[firmware](https://en.wikipedia.org/wiki/Firmware)**which runs on the**[ESP8266](https://en.wikipedia.org/wiki/ESP8266)[Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi)**module from**[Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1" \o "Espressif Systems (page does not exist))**, and hardware which is based on the ESP-12 module.**



Relay Module :-

In this relay, when a current flows through the coil, it turns it into an electromagnet. The magnet pushes a switch to the left, forcing the spring contacts together, and completing the circuit they're attached to. This is a relay from an electronic, hot-water immersion heater programmer.



Keypad Door lock :-

The keypads used with keypad locks can vary greatly. ... When the code is entered correctly, a tone or other sound indicates that the lock has disengaged. Some keypad locks offer a security feature that prevents the lock from opening for 15 minutes after several failed attempts at entering the code.

****

Dustbin :-

Dustbin. a large container used for putting rubbish/garbage in, usually kept outside the house.



Jumping wires :-

The term "jumper wire" simply refers to a conducting wire that establishes an electrical connection between two points in a circuit. You can use jumper wires to modify a circuit or to diagnose problems in a circuit**.**



Ultrasonic Sensor :-

Ultrasonic sensors measure distances based on transmitting and receiving ultrasonic signals.An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back**.**

****

Servo Motor :-

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basicallyservo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears.

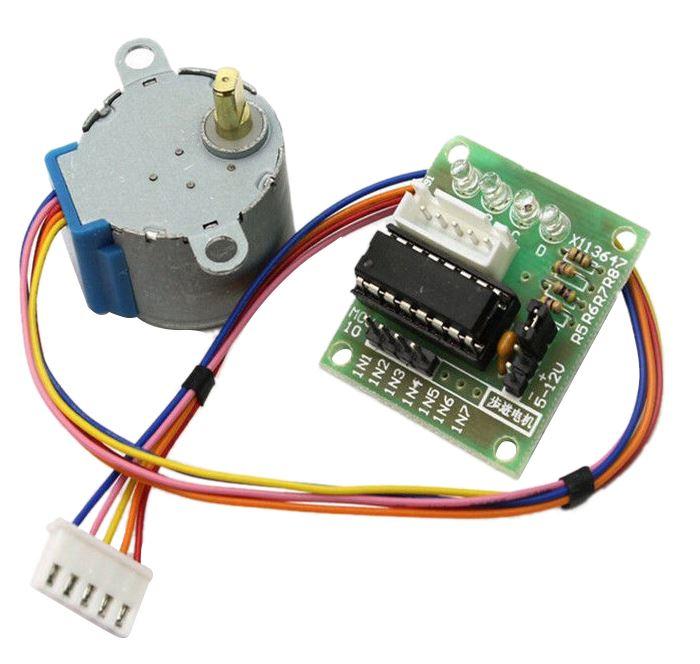


Stepper Motor :-

A stepper motor, also known as step motor or stepping motor, is a brushless DC electric motor that divides a full rotation into a number of equal steps.

Motor Driver :-

The function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.

****

Steps or Algorithm—

Step. 1 – Collect all components, wires and any other

Equipments for use in this project.

Step. 2 – Connection one by one components with.

Step. 3 – Will do wiring for all component.

**Step. 4 – Sample Source Code**

* **Enter the source code for this implementation.**

**Step. 5 – Result**

* **Result of this implementation**

**Step.6 – HTML Coding**

* **Coding for this implementation**

# **Door Lock with keypad using Arduino**

1. **Introduction:-**

Password Based Door Lock System using Arduino UNO is a simple project where a secure password will act as a door unlocking system. Old fashioned lock systems use mechanical locking and these can be replaced by new advanced techniques of locking systems

1. **Component:-**
2. Arduino uno
3. 3\*4 Keypad
4. Servo moter
5. Jumper Wire
6. Bread Board
7. Arduino USB
8. Solid wire
9. **Connections**

**Arduino to Keypad Connection**

|  |  |
| --- | --- |
| **Arduino** | **keypad** |
| **Digital pin4** | **Pin1** |
| **Digital pin5** | **Pin2** |
| **Digital pin6** | **Pin3** |
| **Digital pin7** | **Pin4** |
| **Digital pin8** | **Pin5** |
| **Digital pin9** | **Pin6** |
| **Digital pin10** | **Pin7** |

**Arduino to Servo Motor**

|  |  |
| --- | --- |
| **Arduino** | **Servo Motor** |
| **Orange wire** | **Pin11** |
| **Red wire** | **Gnd** |
| **Brown wire** | **5v** |

1. **Code**

**#include <Servo.h>**

**#include <Keypad.h>**

**Servo ServoMotor;**

**char \*password = "1234"; // change the password here, just pick any 4 numbers**

**int position = 0;**

**const byte ROWS = 4;**

**const byte COLS = 3;**

**char keys[ROWS][COLS] = {**

**{'1','2','3'},**

**{'4','5','6'},**

**{'7','8','9'},**

**{'\*','0','#'}**

**};**

**byte rowPins[ROWS] = { 10,9,8,7};**

**byte colPins[COLS] = { 6,5,4 };**

**Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );**

**int RedpinLock = 4;**

**int GreenpinUnlock = 5;**

**void setup(){**

**ServoMotor.attach(11);**

**LockedPosition(true);**

**}**

**void loop()**

**{**

**char key = keypad.getKey();**

**if (key == '\*' || key == '#')**

**{**

**position = 0;**

**LockedPosition(true);**

**}**

**if (key == password[position])**

**{**

**position ++;**

**}**

**if (position == 4)**

**{**

**LockedPosition(false);**

**}**

**delay(100);**

**}**

**void LockedPosition(int locked)**

**{**

**if (locked)**

**{**

**digitalWrite(RedpinLock, HIGH);**

**digitalWrite(GreenpinUnlock, LOW);**

**ServoMotor.write(11);**

**}**

**else{**

**digitalWrite(RedpinLock, LOW);**

**digitalWrite(GreenpinUnlock, HIGH);**

**ServoMotor.write(90);**

**}**

**}**

**Curtains Control Using Node MCU With Internet**

**Introduction**

Curtain Control solution is designed to open and close window curtains and blinds automatically. Curtains can be controlled with a Mobile phone. The motion of the curtain is in the form of a smooth, continuous movement initiated by pressing and holding down the button on the mobile phone.

**Component**

* ESP8266 NodeMCU
* Stepper Motor
* Stepper Motor Driver
* F to F Jumpers wire
* Breadboard

**Connections:--**

**Node MCU to Motor Driver**

|  |  |
| --- | --- |
| **Node MCU** | **Motor Driver** |
| **Int1** | **Pin D8** |
| **Int1** | **Pin D7** |
| **Int1** | **Pin D6** |
| **Int1** | **Pin D5** |
| **Vcc (+)** | **3v** |
| **Gnd (-)** | **Gnd** |

**Code:-**

**#include <Blynk.h>**

**#define BLYNK\_PRINT Serial**

**#include <Stepper.h>**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

**char auth[] = "11f8e73cee4545238e27d866b1b44af6";**

**char ssid[] = "Redmi3s";**

**char pass[] = "1234567t";**

**// change this to fit the number of steps per revolution**

**const int stepsPerRevolution = 1000;**

**Stepper myStepper(stepsPerRevolution, 14, 12, 13, 15);**

**int xVal=512;**

**void setup() {**

**Serial.begin(115200);**

**Blynk.begin(auth, ssid, pass);**

**}**

**void loop() {**

**Blynk.run();**

**if( (xVal < 500) ){**

**myStepper.step(1);**

**delayMicroseconds(3000);**

**}else if( xVal > 550){**

**myStepper.step(-1);**

**delayMicroseconds(3000);**

**}**

**}**

**BLYNK\_WRITE(V1)**

**{**

**xVal = param[0].asInt();**

**int y = param[1].asInt();**

**}**

**BLYNK\_WRITE(V2)**

**{**

**if( param.asInt() == 1){**

**xVal = 0;**

**}**

**else{**

**xVal = 512;**

**}**

**}**

**BLYNK\_WRITE(V3)**

**{**

**if( param.asInt() == 1){**

**xVal = 1000;**

**}else{**

**xVal = 512;**

**}**

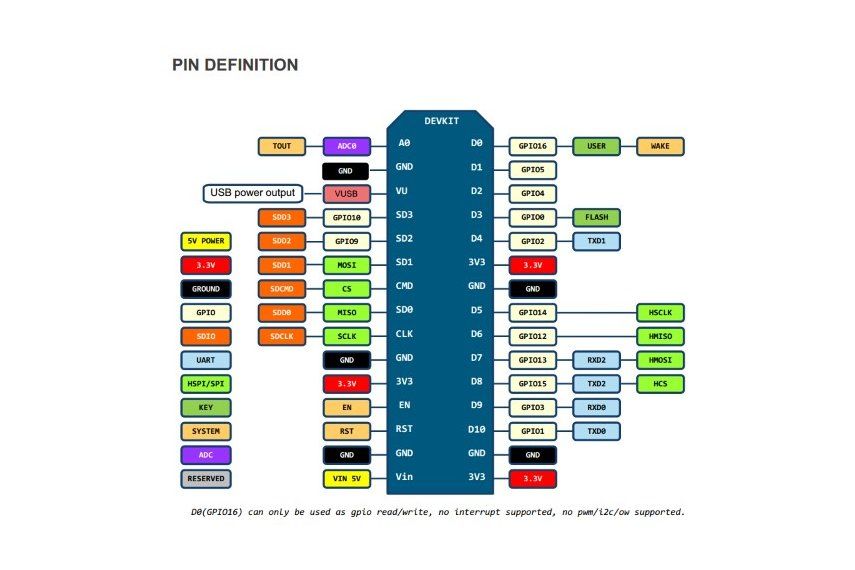
**}**

**Fan Control Using Node MCU Connect with Internet**

**Component :-**

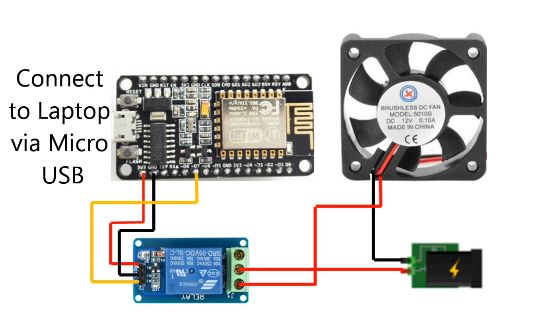
* ESP8266 NodeMCU
* Single Channel 5V Relay Module
* Fan
* Battery 12V
* Jumpers
* Breadboard

**Step 1: Pin Definition**

[](https://cdn.instructables.com/FA3/6B49/IT22LKOP/FA36B49IT22LKOP.LARGE.jpg)

**ESP8266 NodeMCU**

## Step 2: Hardware Installation



**Set up the connection as referred in the figure above.**

**Connection:--**

**NodeMCU to Relay**

|  |  |
| --- | --- |
| **NodeMCU** | **Relay** |
| **Pin D5** | **Signal** |
| **Gnd** | **Vcc (+)** |
| **3v** | **Gnd (-)** |

**Then connect +ve terminal of battery to the Common (C) of Relay, -ve terminal of battery connect to the fan's black wire and Normally Open (NO) of the relay connect to fan's red wire.**

**Code**

**#define BLYNK\_PRINT Serial**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

**auth[] = "11f8e73cee4545238e27d866b1b44af6";**

**char ssid[] = "Redmi3s";**

**char pass[] = "1234567t";**

**void setup()**

**{**

**Serial.begin(15200);**

**Blynk.begin(auth, ssid, pass);**

**}**

**void loop()**

**{**

**Blynk.run();**

**}**

Note :- Light connection is Same as fan connection

**A Smart Dustbin using a Arduino**

**Introduction**

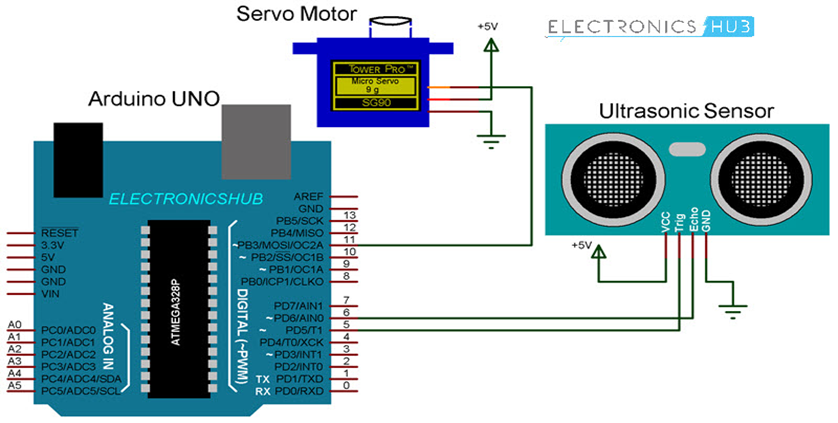
In this project, I have designed a simple system called Smart Dustbin using Arduino, Ultrasonic Sensor and Servo Motor, where the lid of the dustbin will automatically open itself upon detection of human.

**Connection:--**

* Arduino Uno
* Battery 12v
* Ultrasonic Sensor
* Servo Motor
* Wires

**Working :-**

After making a smart dustbin and making all the necessary connections, upload the code to arduino and provide 5v power supply to circuit. And then if ultrasonic sensor detects any object like hand arduino calculate its distance and then arduino will activate the servo motor and servo motor will lid open and closed automatically.

**

**Code:--**

**#include <Servo.h>**

**Servo servo;**

**int trigPin = 5;**

**int echoPin = 6;**

**int servoPin = 7;**

**int led= 10;**

**long duration, dist, average;**

**long aver[3];**

**void setup() {**

**Serial.begin(9600);**

**servo.attach(servoPin);**

**pinMode(trigPin, OUTPUT);**

**pinMode(echoPin, INPUT);**

**servo.write(0);**

**delay(100);**

**servo.detach();**

**}**

**void measure() {**

**digitalWrite(10,HIGH);**

**digitalWrite(trigPin, LOW);**

**delayMicroseconds(5);**

**digitalWrite(trigPin, HIGH);**

**delayMicroseconds(15);**

**digitalWrite(trigPin, LOW);**

**pinMode(echoPin, INPUT);**

**duration = pulseIn(echoPin, HIGH);**

**dist = (duration/2) / 29.1;**

**}**

**void loop() {**

**for (int i=0;i<=2;i++) {**

**measure();**

**aver[i]=dist;**

**delay(10);**

**}**

**dist=(aver[0]+aver[1]+aver[2])/3;**

**if ( dist<50 ) {**

**servo.attach(servoPin);**

**delay(1);**

**servo.write(0);**

**delay(3000);**

**servo.write(150);**

**delay(1000);**

**servo.detach();**

**}**

**Serial.print(dist);**

**}**

**Advantages –**

* **The main advantage of “Home Automation” is that the “Physically challenged and Disabled people”.**
* **Controlled through one common device**
* **Fast enough to realize the true power of wireless technology**
* **Replace television, air conditioner etc., remote for sleep mode.**
* **Smart Home.**

**Disadvantage –**

* **It is expensive too.**
* **It can not repair easily.**

**After complete this project**

****