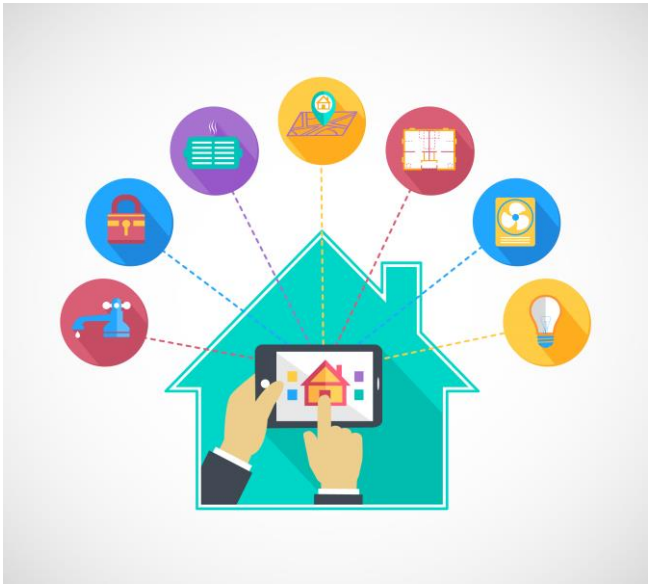
INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR



ARDUINO BASED HOME AUTOMATION SYSTEM WITH ANDROID AND BLUETOOTH

PRESENTED BY-

SAYAN SETH

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

INTRODUCTION

What is a Smart Home Automation System?

Home automation (also known as domotics) refers to the automatic and electronic control of household features, activity, and appliances. Various control systems are utilized in this residential extension of building automation. Some components of an automated home may include the centralized control of security locks on doors and gates, appliances, windows, lighting, surveillance cameras and HVAC systems (heating, ventilation and air conditioning).

The Advantages of an Automated Home

Home automation has greatly increased in popularity over the past several years. One of the greatest advantages of an automated home is the ease with which functionality can be managed on an array of devices: desktop, laptop, tablet or smartphone.

1. Security
2. Energy Efficiency
3. Savings
4. Convenience
5. Comfort

How Does Smart Home Automation Work?

Home automation systems are composed of hardware, communication and electronic interfaces that work to integrate electrical devices with one another. Domestic activities can then be regulated with the touch of a button. From any remote location, users can adjust the controls on home entertainment systems, limit the amount of sunlight given to houseplants, or change the temperatures in certain rooms. Home automation software is often connected through computer networks so that users can adjust settings on their personal devices.

Components of an Automation System:

- Controlled Devices
  They include household appliances, door openers, power door locks, sprinkling systems, lighting systems, HVAC systems, audio/video systems, home theatre equipment, security systems, telephone systems, intercoms, messaging systems, information systems, and many other types of equipment.

- ## Sensing Devices
  Sensing devices can report values, such as temperature, humidity, light levels, sound levels, etc., or states (such as on, off, open, closed, etc.). The signals sent by sensors are converted into data that can be displayed to the user or used by a controller program to make informed decisions based on certain conditions.

- ## I/O Interface Devices
  I/O (input/output) interface devices provide the logical communication link between the controller(s) and the controlled devices in a system. An I/O interface device can serve several communications functions, including:
  Converting analog signals to digital signals that can be used by the controller.

- ## Controllers
  Controllers provide the intelligent control functions in a home automation system. The control functions may be contained in a single central controller, or there may be other controllers besides the central controller that have a limited subset of control functions. All controllers must have sufficient data in order to control the controlled devices.

- ## User Interface
  User interfaces allow the user to interact with the system by sending information to the controller or by presenting information to the user about the system. Typical user interface devices include:

  - Push-button panels, with or without visual displays.
  - Touch-panel displays, with fixed or programmable screen layouts.
  - Computer keyboards and monitors.
  - Hand-held remote controls.
  - Telephone interfaces to allow long-distance remote control.
  - Television controllers with on-screen menus.

- ## System Network
  The system network includes all of the controllers, sensors, wires, cables, RF (radio frequency) links, IR (infrared) links, adapters, connectors, junction boxes, dimmers, ballasts, power supplies, etc. that connect the various system components.

- ## Programming Computer
  Some system controllers allow the user to program the system with the system's own user interface(s). Other systems require the use of a separate computer (typically a PC) to program the system controller.

PHYSICAL COMPONENTS USED IN THIS PROJECT

- Arduino UNO (Atmega 328 microcontroller)
- HC-05 Bluetooth Module
- Light Dependent Resistor(LDR)x1
- Light Emitting Diode(LED)x2
- 1k Ohm Resistors x6
- Servo Motor x1
- Breadboard
- Jumper wires

SOFTWARES AND DEVELOPMENT ENVIRONMENTS USED

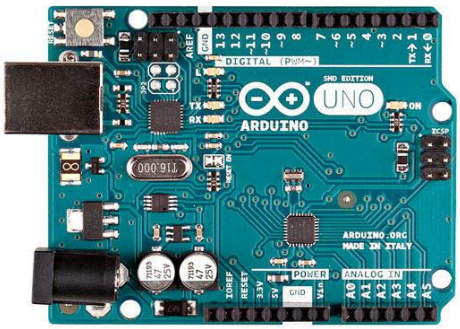
- Arduino IDE
- MIT App Inventor 2
- Fritzing
- Proteus 8 Professional

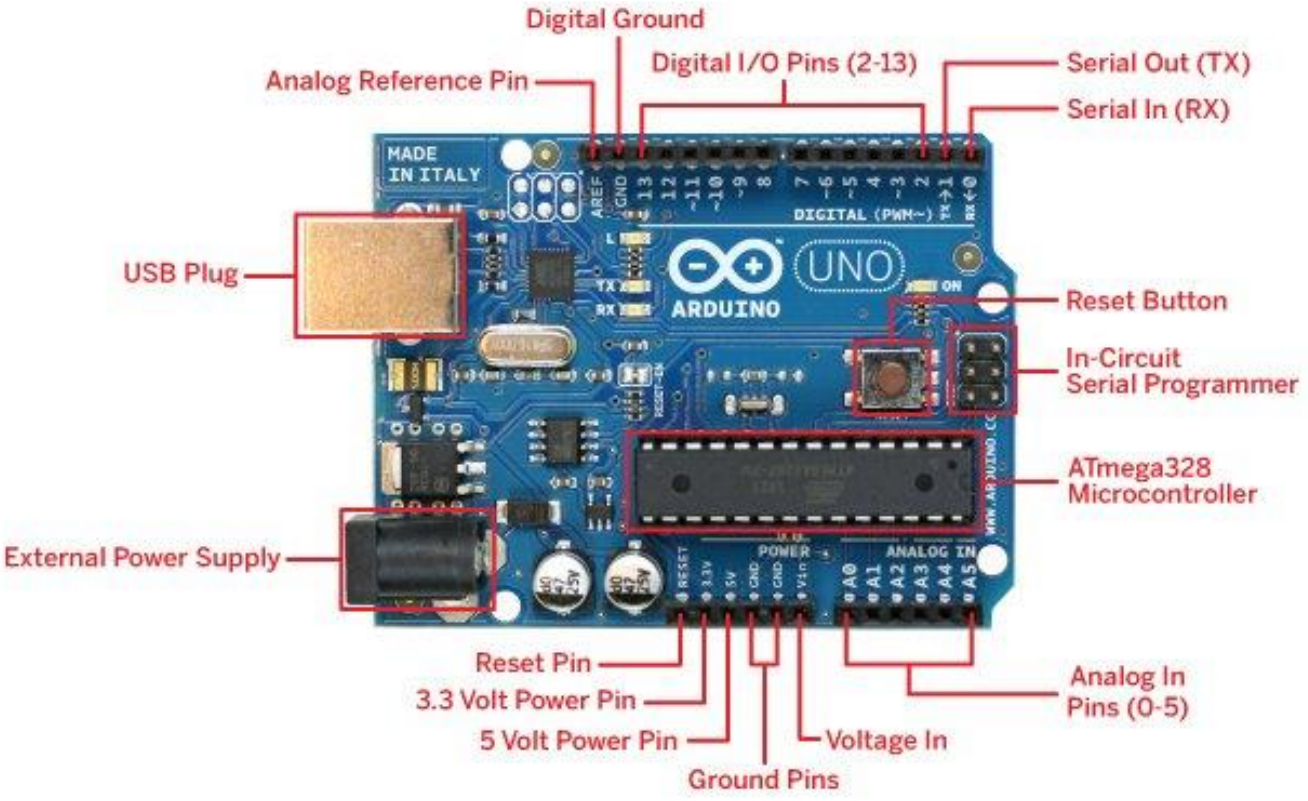
Arduino Uno R3

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP(In Circuit Serial Programmer) header and a reset button.

Features

- ATmega328 microcontroller
- Input voltage - 7-12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32k Flash Memory
- 16Mhz Clock Speed
- SRAM 2 KB
- EEPROM 1KB





DIFFERENT PARTS OF ARDUINO UNO:

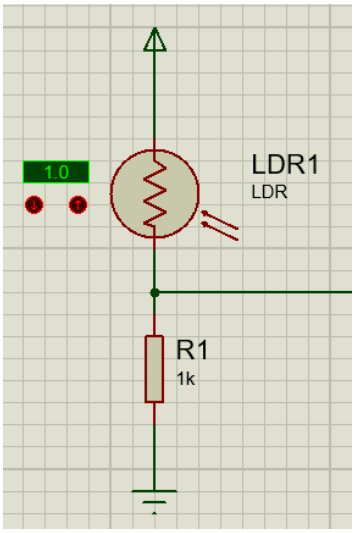
Serial: 0 (RX) and 1 (TX): Used to receive (RX) and transmit (TX) TTL serial data.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

Analog Pins: It support 10-bit Analog -to-digital conversion (ADC) using the analogRead() function.

LDR

A (light-dependent resistor) LDR is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits.

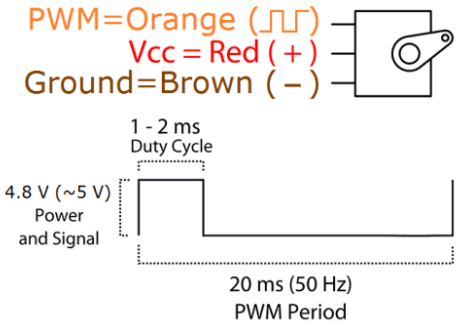


SERVO MOTOR

Because servo motors use feedback to determine the position of the shaft, Its position can be controlled very precisely. As a result, servo motors are used to control position of objects, rotate objects, move legs, arms or hands of robots, move sensors etc. with high precision.

Most servo motors have the following three connections:

1. Black/Brown ground wire.
2. Red power wire (around 5V).
3. Yellow or White PWM wire.



HC 05 Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Hardware Features:

1. Typical -80dBm sensitivity.

2. Up to +4dBm RF transmit power.

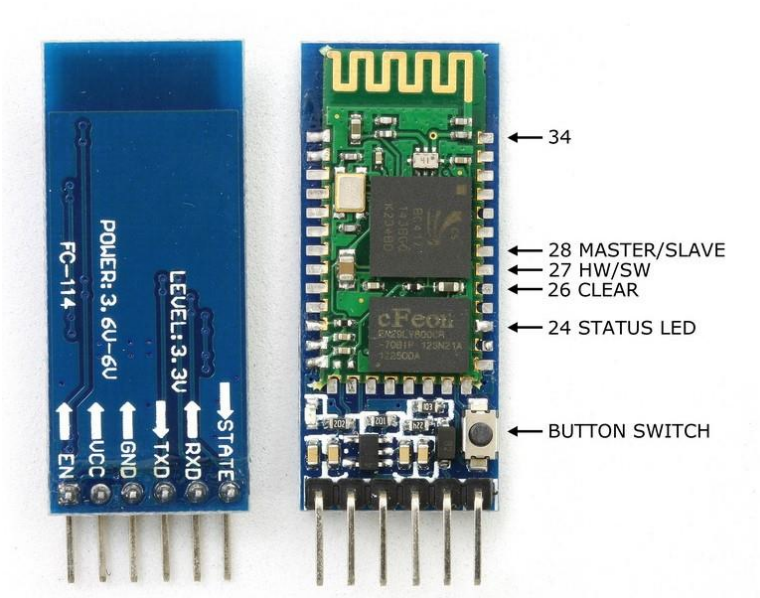
3. 3 to 5 V I/O.

4. PIO (Programmable Input/Output) control.

5. UART interface with programmable baud rate.

6. With integrated antenna.

7. With edge connector.



Software Features:

1. Slave default Baud rate: 9600, Data bits:8, Stop bit:1, Parity: No parity.

2. Auto-connect to the last device on power as default.

3. Permit pairing device to connect as default.

4. Auto-pairing PINCODE:”1234” as default.

Pin Description:

The HC-05 Bluetooth Module has 6pins.

ENABLE: When enable is pulled **LOW**, the module is disabled which means the module will **not turn on** and it **fails to communicate.** When enable is **left open or connected to 3.3V**, the module is enabled i.e. the module **remains on** and **communication also takes place**.

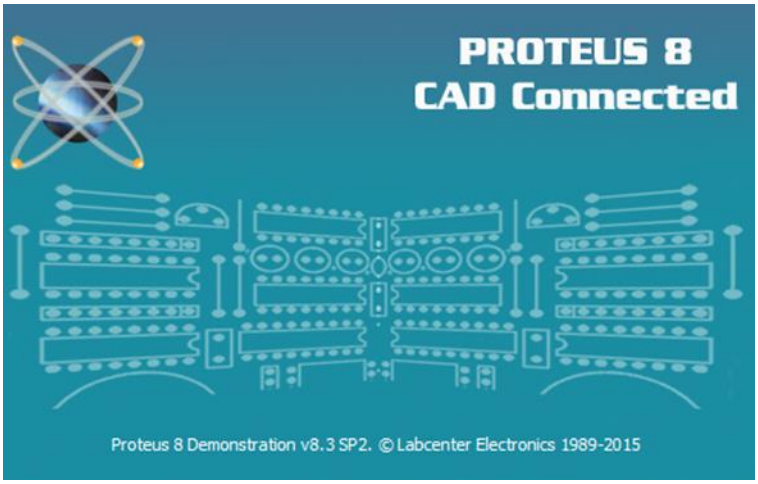
VCC: Supply Voltage 3.3V to 5V

GND: Ground pin

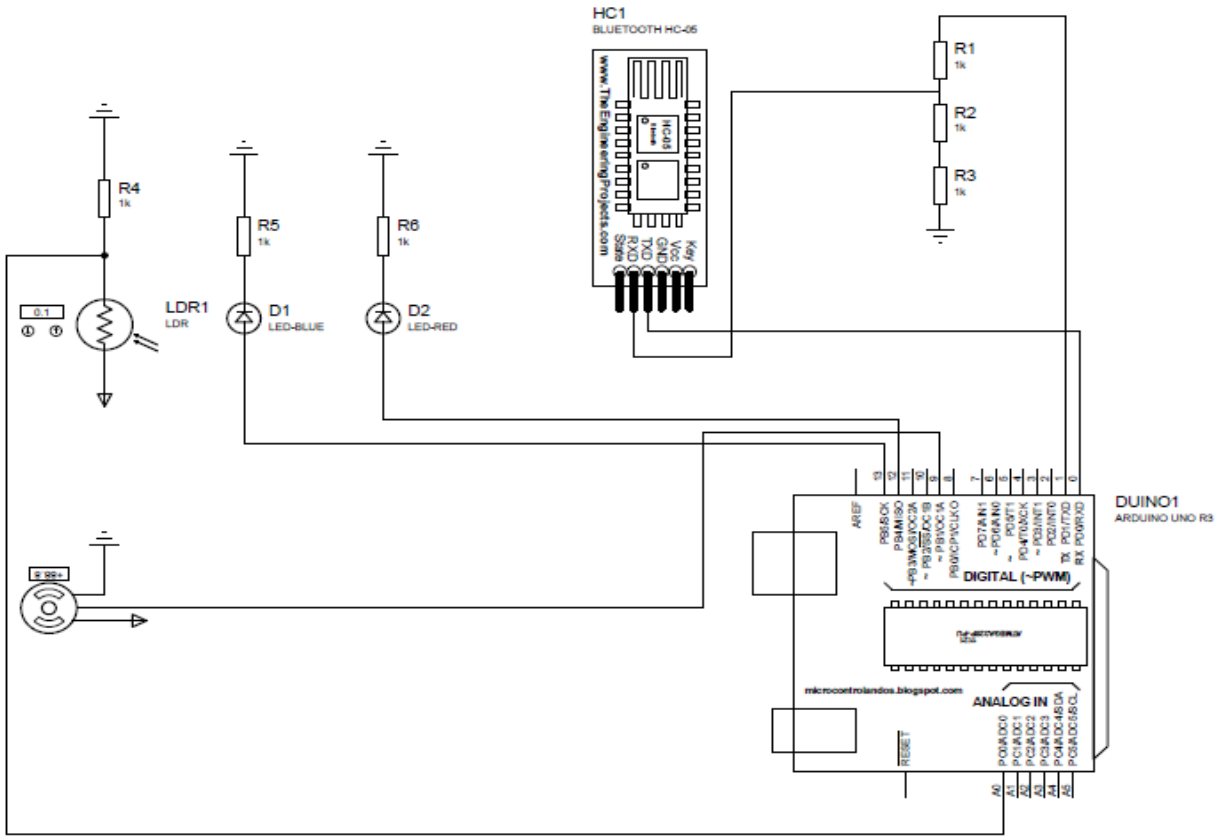
TXD & RXD: These two pins acts as an UART interface for communication.

STATE: It acts as a status indicator. When the module is **not connected to / paired** with any other Bluetooth device, signal goes **Low**. At this **low state**, the **led flashes continuously** which denotes that the module is **not paired** with other device. When this module is **connected to/paired** with any other Bluetooth device, the signal goes **High**. At this **high state**, the **led blinks with a constant delay** say for example 2s delay which indicates that the module is **paired**.

SOFTWARES:



CIRCUIT DIAGRAM



ARDUINO CODE

```
/*
  Home Automation with Arduino,Android,Bluetooth

  Created 4 Dec 2016
  by Sayan Seth
  */
#include <Servo.h>

Servo myservo;

#define ledPin1 13
#define ledPin2 12
#define servoPin 9
#define thresh 500
int state = 0;
int flag=0;   //check if led 1 is changing state.If it is in auto mode, check the
lighting condition in every loop. If it is not, don't.
void auto_light();
void setup()
{
  pinMode(ledPin1, OUTPUT);
  digitalWrite(ledPin1, LOW);
  pinMode(ledPin2, OUTPUT);
  digitalWrite(ledPin2, LOW);
  myservo.attach(servoPin);
  Serial.begin(9600);// Default communication rate of the Bluetooth module
}
void loop() {
  if(Serial.available() > 0) // Checks whether data is comming from the serial port
  {
  state = Serial.read();  // Reads the data from the serial port
  }
  if (state == '2')
 {
  digitalWrite(ledPin1, LOW); // Turn LED OFF
  state = 0;
  flag=1;  //We are chnging state of LED 1, no need to run the auto function
 }
  if (state == '1')
 {
  digitalWrite(ledPin1, HIGH); // Turn LED ON
  state = 0;
  flag=1; // We are changing state of LED 1, no need to run the auto function
 }
  if (state == '4')
  {
  digitalWrite(ledPin2, LOW);
  state = 0;

 }
  if (state == '3')
 {
  digitalWrite(ledPin2, HIGH);
  state = 0;
 }
  if (state == '5')
 {
  myservo.write(10);
  delay(1000);
  state = 0;
 }
  if (state == '6')
 {
  myservo.write(90);
```

```
  delay(1000);
  state = 0;
 }
  if (state == '7')
 {
  myservo.write(170);
  delay(1000);
  state = 0;
 }
  if (state == '8' || flag==2) //Auto mode is activated, run the auto function
 {
  auto_light();
  state = 0;
 }
}
void auto_light()
  {

    int val;
    flag=2; // signifies that auto mode is on
    val = analogRead(A0);
    if(val>=thresh)
    {
      digitalWrite(ledPin1,LOW);
    }
    else
      digitalWrite(ledPin1, HIGH);


   }
```


ACKNOWLEDGEMENT

We thank our beloved professors in The Department of Electronics and Telecommunication Engineering, IIEST Shibpur for giving us advices & necessary helps to successfully complete this project.

RESOURCES

1. Wikipedia
2. www.arduino.cc
3. https://diyhacking.com/diy-android-home-automation/