**TABLE OF CONTENTS**

1. Abstract
2. Arduino
3. Working and Programming in Arduino
4. Advantages of Arduino
5. Circuit Diagram
6. Components Required
7. Arduino Code
8. Difficulty and learning.

**ABSTRACT**

One of the most wonderful creations by human is the automaton and its derivation, robot. Automatizing things give us many benefits such as accuracy, time saver, specifications and what not. One such application is home automation. It has a wide range of applications in day to day life. For an instance, a washing machine or an air conditioner can be remotely controlled using the arduino.

* We are going to use an Arduino YUN controller to control the home appliances without any wired connection.
* The wireless connectivity of the Arduino based automation system uses the simple components like relay and resisters.
* It controls the home appliances by using the Wi-Fi enabled device.
* We can also use an Arduino UNO Board along with Arduino Ethernet Shield to give it a wireless connectivity.
* Arduino runs a code to control a Relay board according to the input and also serves a web page through which respective output to the relay board can be controlled.

Through relay controller board, we can control lamps, tubes or any AC power sockets.

**WIFI MODULE**

The crucial part of home automation is wireless relay control. Usually, relays are used to control high powered devices which consume/require high current. We need a local host or a server space to host a web application or rather use an app as an interface for the user to control the components wirelessly and the challenge is to connect the code logic with the web interface which can be done using Wi-Fi module or Ethernet shield combined with arduino.

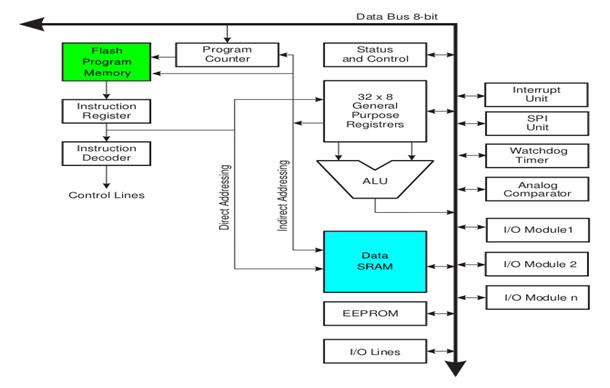
**ARDUINO**

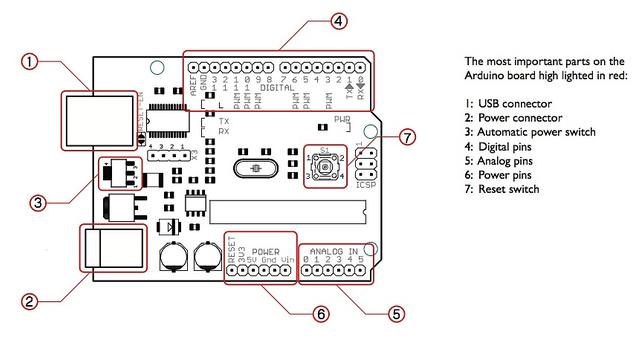
It is an open source hardware, software and user interface that can be used for designing/manufacturing interactive objects which can sense or control the physical world.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits.

**Working of Arduino:**

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.





**Programming in an Arduino:**

The most important advantage with Arduino is the programs can be directly loaded to the device without requiring any hardware programmer to burn the program. This is done because of the presence of the 0.5KB of Bootloader which allows the program to be burned into the circuit. All we have to do is to download the Arduino software and write the code.

* Programs written in Arduino are known as sketches. A basic sketch consists of 3 parts

1. Declaration of Variables  
 2. Initialization: It is written in the setup () function.  
 3. Control code: It is written in the loop () function.

* The sketch is saved with a .ino extension. Any operations like verifying, opening a sketch, saving a sketch can be done using the buttons on the toolbar or using the tool menu.
* The sketch should be stored in the sketchbook directory.
* Chose the proper board from the tools menu and the serial port numbers.
* Click on the upload button or chose upload from the tools menu. Thus the code is uploaded by the bootloader onto the microcontroller.

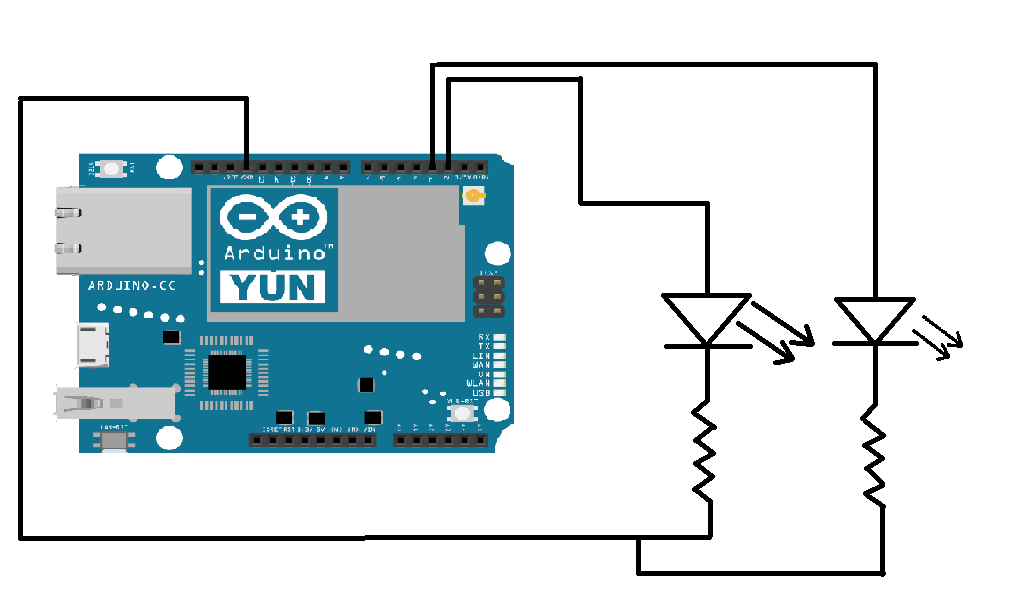
**Advantages of Arduino:**

* It is inexpensive
* It comes with an open source hardware feature which enables users to develop their own kit using already available one as a reference source.
* The Arduino software is compatible with all types of operating systems like Windows, Linux, and Macintosh etc.
* It also comes with open source software feature which enables experienced software developers to use the Arduino code to merge with the existing programming language libraries and can be extended and modified.
* It is easy to use for beginners.
* We can develop an Arduino based project which can be completely stand alone or projects which involve direct communication with the software loaded in the computer.
* It comes with an easy provision of connecting with the CPU of the computer using serial communication over USB as it contains built in power and reset circuitry.

**Arduino Specifications:**

* 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
* Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
* SRAM 2 KB (ATmega328)
* EEPROM 1 KB (ATmega328)
* Clock Speed 16 MHz
* Length 68.6 mm
* Width 53.4 mm
* Weight 25 g

**SCHEMATIC CIRCUIT DIAGRAM**

****

**COMPONENTS REQUIRED**

1. 1k ohms resistors
2. Arduino YUN
3. Connecting wires
4. Led (blue, red, green, yellow)
5. Breadboard

We have used Arduino YUN which is an another type/version of arduino board which has an inbuilt Wi-Fi/Ethernet controller and does not need an external connection. In this mini-project relay has not been used since the devices used are low power devices and relays can rather be used when there are high power devices.

**ARDUINO CODE**

#include <Bridge.h>

#include <YunServer.h>

#include <YunClient.h>

int LEDPIN2 = 2;

int LEDPIN = 3;

YunServer server;

void setup() {

Serial.begin(9600);

pinMode(LEDPIN,OUTPUT);

digitalWrite(LEDPIN,HIGH);

digitalWrite(LEDPIN2,HIGH);

Bridge.begin();

digitalWrite(LEDPIN,LOW);

digitalWrite(LEDPIN2,LOW);

delay(500);

digitalWrite(LEDPIN,HIGH);

digitalWrite(LEDPIN2,HIGH);

delay(500);

digitalWrite(LEDPIN,LOW);

digitalWrite(LEDPIN2,LOW);

delay(500);

digitalWrite(LEDPIN,HIGH);

digitalWrite(LEDPIN2,HIGH);

delay(500);

digitalWrite(LEDPIN,LOW);

digitalWrite(LEDPIN2,LOW);

delay(500);

server.begin();

}

void loop() {

YunClient client = server.accept();

if (client) {

process(client);

client.stop();

}

delay(50);

}

void process(YunClient client) {

String command = client.readStringUntil('\\');

client.println("Status: 200");

client.println("Content-type: text/html");

client.println();

client.println("<B><Center>");

client.println("LED 1");

client.println("<button><a href='http://yun.local/arduino/1\\'>Turn ON LED</a></button>");

client.print("<button><a href='http://yun.local/arduino/2\\'>Turn OFF LED</a></button><br>");

client.println("</B></Center>");

client.println("<B><Center>");

client.println("LED 2");

client.println("<button><a href='http://yun.local/arduino/3\\'>Turn ON LED</a></button>");

client.print("<button><a href='http://yun.local/arduino/4\\'>Turn OFF LED</a></button><br>");

client.println("</B></Center>");

int str=command.toInt();

if (str == 1) {

digitalWrite(3,HIGH);

client.println("<B><Center>");

client.print("Command: ");

client.println("ON");

client.println("</B></Center>");

}

if (str == 2) {

digitalWrite(3,LOW);

client.println("<B><Center>");

client.print("Command: ");

client.println("OFF");

client.println("</B></Center>");

}

if (str == 3) {

digitalWrite(2,HIGH);

client.println("<B><Center>");

client.print("Command: ");

client.println("ON");

client.println("</B></Center>");

}

if (str == 4) {

digitalWrite(2,LOW);

client.println("<B><Center>");

client.print("Command: ");

client.println("OFF");

client.println("</B></Center>");

}

}

**DIFFICULTY AND LEARNING**:

Understanding the concept of a Wi-Fi module and how it can be used to relate wireless control of devices.

Learning arduino programing.

Setting up the hardware.