CHAPTER 4

SYSTEM ANALYSIS

Automation of a device has a wide scope for this Generation as well as in forthcoming generation. In this wide scope, Mobile communication technology is playing a major role in the world of automation. This article is fully based on low cost and reliable home control monitoring system for accessing and controlling devices and appliances remotely using Android based Smart phone application. While using this technology the system improves the living standard at home, reduces human effort, energy efficient and time saving and thus make a smart home .And also it was very helpful for providing support to disabled people and fulfil their needs in home and thus they leads a normal life. The proposed systems consist of android mobile, Arduino Uno board, Wi-Fi module and a relay circuit. We are using Wi-Fi technology to monitor the device because of its accuracy, high range and instant connectivity. This module controls the home appliances with a very ease of installation and it is user friendly.

Home Automation usually is comprised of three main parts:

1. Main Controller
2. Interfaces
3. Control methods

4.1. MAIN AUTOMATION CONTROLLER:

A .ARDUINO UNO BOARD :

The 8-bit AT mega 328P microcontroller based on Arduino UNO is used in this proposal is to control the different components like Wi-Fi module and relay circuit networks. The advantage to having a separate controller is to focus only on the desired task.

4.2 INTERFACES:

An Interface is the way we interact with the Home automation controller. There are many types of interfaces like Touch Panels, Keypads, Remotes, Mobile Devices and Internet.

In this proposal, we used a Mobile device (Android smart phone).Nowadays it is a very common device for every user. We need to install an appliance controller application in it. In addition, within the mobile interface it can be able to control all the respected appliances of the home.

4.3 CONTROL METHODS:

We now have a controller, interfaces to interact with the controller, and sensors that tell the controller whatthings are occurring in the house.

Controllers can communicate and control the many different parts of a Home Automation System in a variety of ways. Some of these are IP (Internet Protocol sp), Wi-Fi,Bluetooth, Zig-bee, IR, Serial Data, and Relays (for motorization).

A .Wi-Fi technology:

Wi-Fi is a great option when you cannot get Ethernet wiring to desired locations. It is a good medium for communicating to different locations in the house, and will allow large bits of information to be passed back and forth with no wires. It is always best if we can get a wire to the location you are trying to control, but sometimes this is not possible or would be cost prohibitive.

B. Advantage of using Wi-Fi technology:

- Equipment can be placed almost anywhere
- No unsightly cords running throughyour home
- No need for additional Ethernet output
- Provide wide range and more efficient

4.3 BLOCK DIAGRAM OF ARDUION UNO CONTROLLER

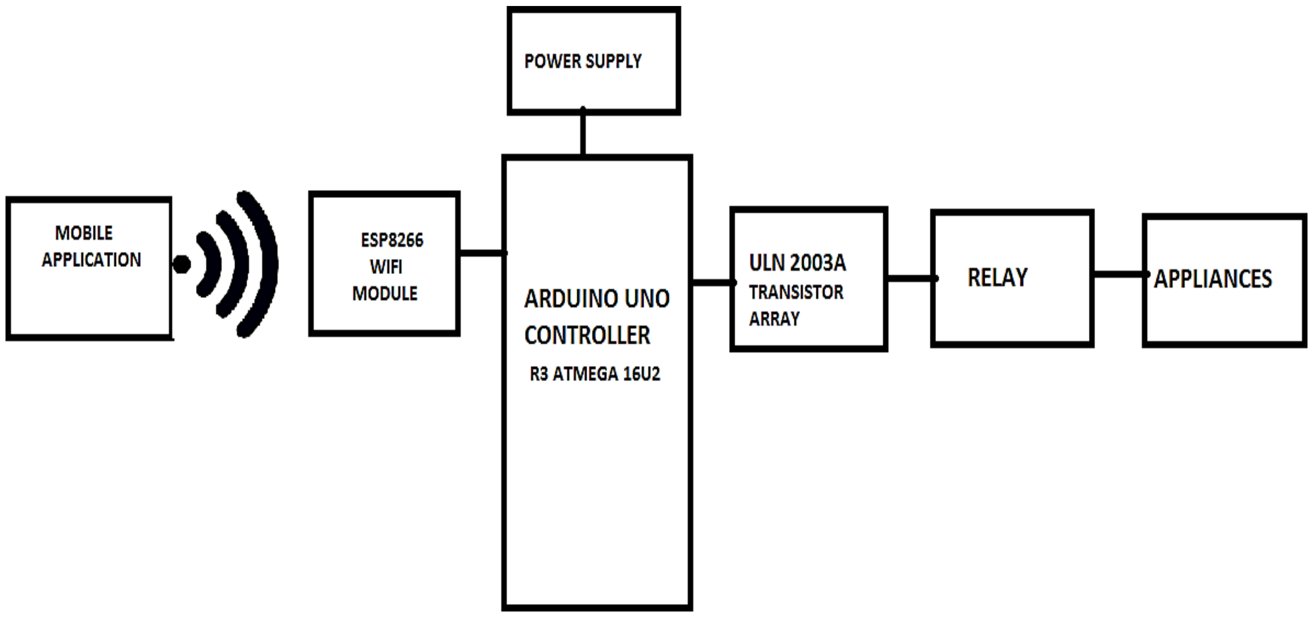

Figure 4 :Block diagram

In this system, we are going to make a home automation system using ESP8266 Wi-Fi module and Arduino Uno Controller R3. With the help of these modules, we will be able to control lights, electric fan and other home appliances through a Wi-Fi application using our Android Smart phone. All the appliances will be connected to relays, which are controlled by the Arduino.

ESP8266 and Arduino together act as a web Server and it will send control commands through the mobile application i.e.android software.

We are implementing withthe help of ESP8266 Wifi Module, as it is a self-contained SoC with integration of TCP/IP stack,which may help any microcontroller having UART to access a Wi-Fi network. It can act as both Wi-Fi access point as well as a Wi-Fi client.It is pre-programmed with AT commands, so we can easily access and configure it using a microcontroller. First, we can connect ESP8266 with Arduino Uno. The ESP8266 runs on 3.3V, it may damage if you connect it directly to 5V from Arduino. Now we can connect relays to Arduino. Here a ULN IC is connected which is used as relay driver. Then all the AC devices are connected to relay output to ON/OFF the AC devices.

A. *ULN IC:*

It is a relay driver IC and it is a Darlington array having high voltages and high currents.

It is made up of seven open collector Darlington pairs having common emitter which show ULN2003 has a capability of handling seven different relaysat a time.

A single Darlington pair consists of two bipolar transistors and operates in the range of 500mA and 600mA.

CHAPTER 6
SYSTEM DESIGN

6.1 Circuit Diagram:

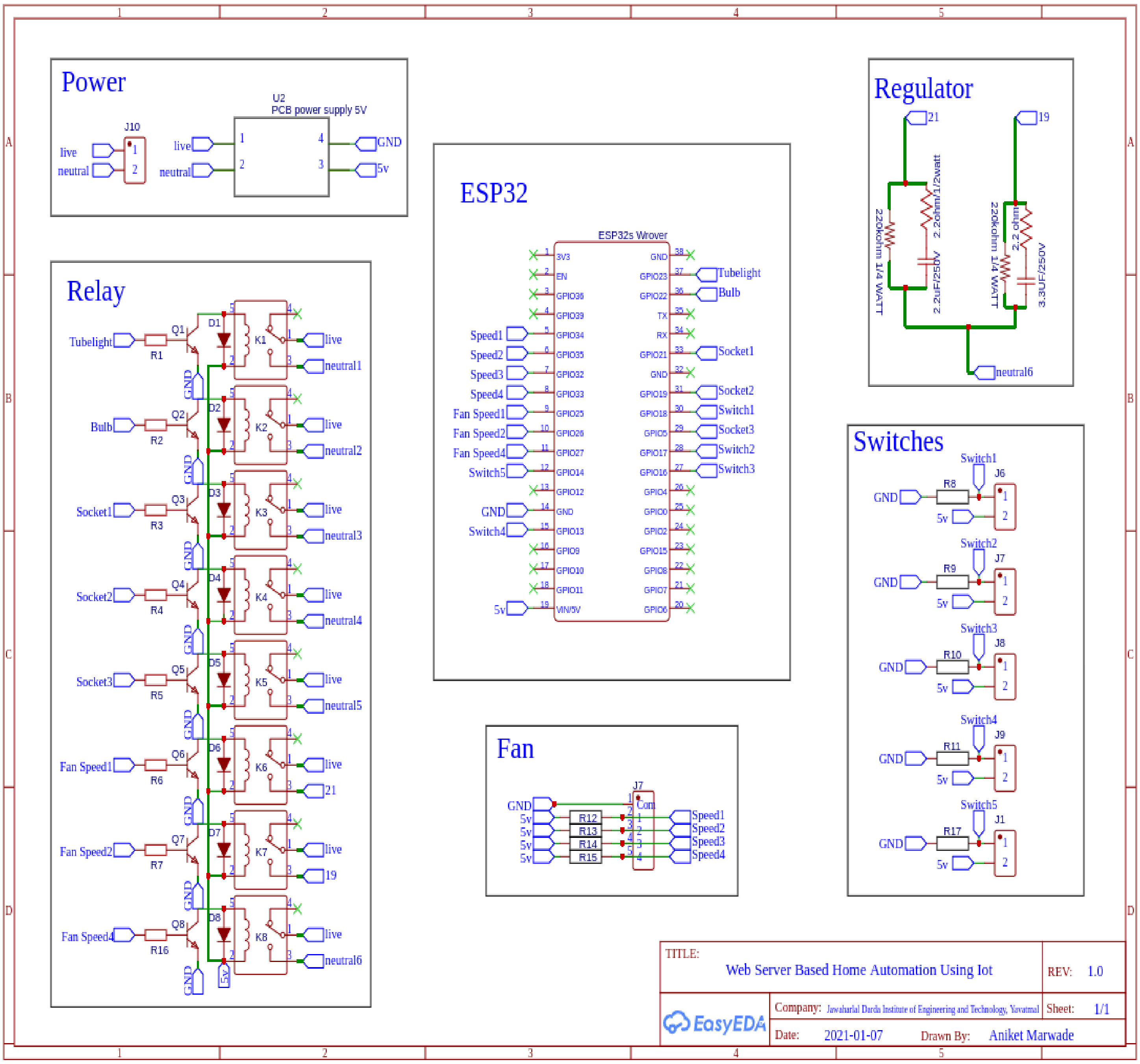

Figure 5: Circuit Diagram

6.2 Pin Description

ESP32-WROVER has 38 pins.

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12 [1]	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2 [2]	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3 [2]	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD [2]	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK [2]	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0 [2]	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1 [2]	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
NC1	27	-	-
NC2	28	-	-
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7

Name	No.	Type	Function
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Table 1: Pin Definitions

1. GPIO12 is internally pulled high in the module and is not recommended for use as a touch pin.
2. Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the SPI flash integrated on the module and are not recommended for other uses.

6.3 Strapping Pins :

ESP32 has five strapping pins, which can be seen in Chapter 6 Schematics:

6.3.1 MTDI

6.3.2 GPIO0

6.3.3 GPIO2

6.3.4 MTDO

6.3.5 GPIO5

Software can read the values of these five bits from register ”GPIO_STRAPPING”.

During the chip’s system reset release (power-on-reset, RTC watchdog reset and brownout reset), the latches ofthe strapping pins sample the voltage level as strapping bits of ”0” or ”1”, and hold these bits until the chip is powered down or shut down. The strapping bits configure the device’s boot mode, the operating voltage of VDD_SDIO and other initial system settings.

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if astrapping pin is unconnected or the connected external circuit is high-impedance, the internal weak
pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the hostMCU’s GPIOs to control the voltage level of these pins when powering on ESP32.

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Booting Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don’t-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Booting					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Table 2: Strapping Pins

- Firmware can configure register bits to change the settings of ”Voltage of Internal LDO (VDD_SDIO)” and ”Timingof SDIO Slave” after booting.
- The MTDI is internally pulled high in the module, as the flash and SRAM in ESP32-WROVER only support a powervoltage of 1.8 V (output by VDD_SDIO).

6.4 Flow Chart :

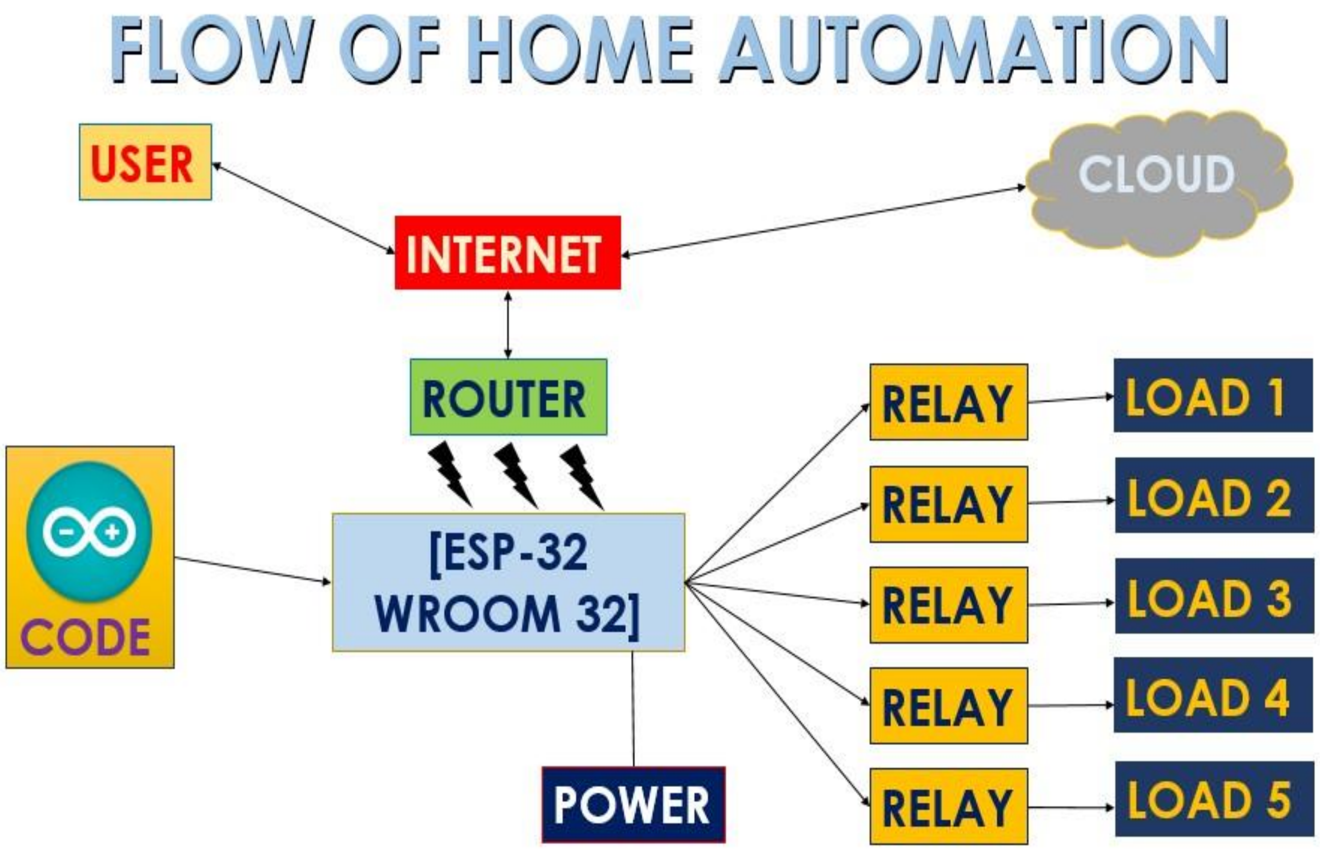

Figure 2: Flow Chart of Home Automation System

Above figure shows the flow chart of Home Automation System in which we have the various components such as User,Cloud, Esp-32 Wroom 32, Relay, Router and Load. The user is connected to the internet which it is connected to router and the cloud of the blink app server, The app gets activated when it is connected to the internet then it gets updated form the cloud platform.The user do the changes it switch ON the fan, light and OFF it, it reflect in the app the router provides the internet connection to the circuit.The ESP 32 can perform a complete standalone system or as a slave device to the Host MCU, reducing communication stack overhead on the main application process, it is connected to the power and with all realys.
The ESP32 is coded in the C++ language, it generates the token and provide to blink app, The load 1, 2, 3, 4 can have various appliances connected to it.

6.5 Semantic Flowchart:

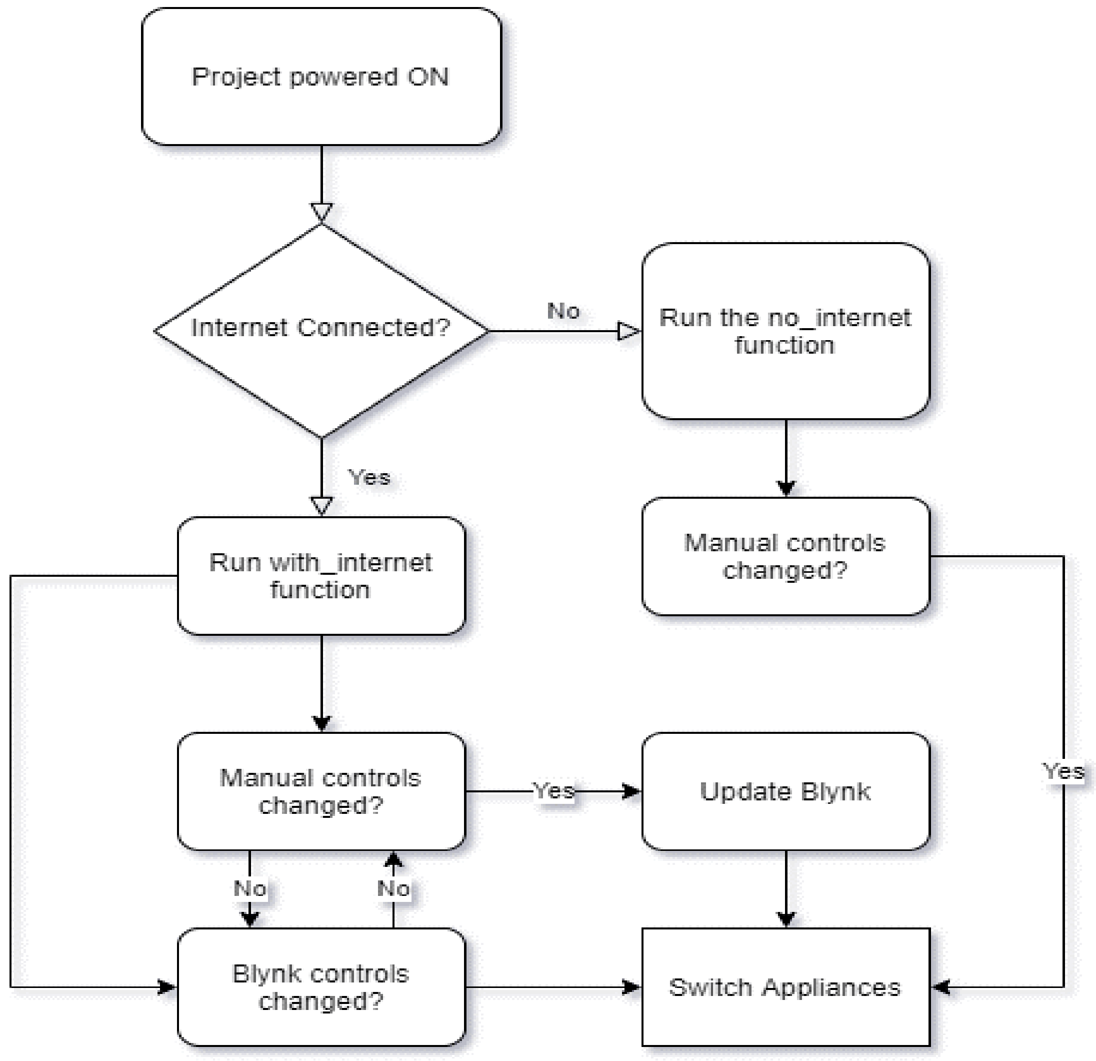

Figure 3: Semantic Flowchart of Home Automation System

We have the Semantics Flowchart of Home Automation System in which we able to see first we give the power to the circuit then it checked whether it is connected to internet or not if it is connected to internet it run with the internet function and its code then it see whether we controlling manually or not by the app if controlling manually then Blynk controls will be changed it update the blynk app and appliances will be switch on.

If the internet is not connected then it run with the no internet function code and check whether the control changed manually or by app and then the appliances switch on.