International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

IOT Based Voice Controlled Notice Board

Pradnya Kawade[1], Tejal Bhise[2], Disha Jadhav[3], Harshali Koli[4]

[1234]*Department of Electronics Engineering, Datta Meghe College of Engineering, Airoli, Navi Mumbai, Maharashtra, India*

Abstract -In current situation notice board is needed in lots of organizations. A notice board display is used to display the message or data sent through the excessive authorities of the organization. To display numerous notice a separate individual is appointed to do such work & it's also remember as a tough task. Here this task is handling enlistment Wi-Fi Electronic notice board. Whenever a notice is sent from the device via any tool, notice can be displayed on Wi-Fi digital notice board. This message may be sent from any tablet/smart-phone/computer etc. with Android OS upon a GUI primarily based totally on contact display operation. When the user is sending the message from the tool this may be obtained through the WI-FI modular. As the WI-FI module and port number information is only known to the user so it will be recognizedthe user who's operating. Later it's far sent to the Microcontroller that similarly enables in displaying the notice in Wi-Fi digital notice board that's equipped with LED.

Key Words:Wi-Fi technology, LED Display, Web Browser, Mobile phone

1.INTRODUCTION

The paper pursuits at designing a LED primarily based totally scrolling message display managed from mobile smartphone over IOT Web interface. The proposed gadget uses IOT Wi-Fi generation to speak from mobile smartphone to LED display board with textual content messaging.
The messages to be displayed are sent over android smartphone based browser app to Node-MCU microcontroller. Microcontroller decodes the message and sends LED Display modules. We have alternatives of sending messages concurrently or selectively to this display. Upon receiving the message it will likely be scrolled on LED display connected.
The proposed gadget makes use of Wi-Fi based wi-fi serial data communication in displaying messages on a remote digital notice board. Android based Application programs available for Wi-Fi communication used for transmitting the alpha-numeric textual content messages. Using Wi-Fi based serial data communication technique, the corresponding transceiver module has been interfaced with microcontroller board on the receiver end. For this purpose, a low price microcontroller board is programmed to obtain alphanumeric textual content messages in any of the above selected communication modes. The proposed gadget willassist in decreasing the human effort, paper, printer ink and price for manual changing of the notices.

2. HARDWARE REQUIREMENTS

2.1 NodeMCU ESP8266 Microcontroller

NodeMCU is an open source IoT platform. It consists of firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware that's based on the ESP-12E module. The term "NodeMCU" by default refers to the firmware instead of the improvement kits. The firmware makes use of the Lua scripting language. It is based on the eLua project, and constructed on the Espressif Non-OS SDK for ESP8266. Since Node MCU is open source platform, their hardware layout is open for edit/- modify/build. Node MCU Dev. Kit/board include ESP8266 Wi-Fi enabled chip. The ESP8266 is a low-price Wi-Fi chip developed with the aid of using Espressif Systems with TCP/IP protocol. There is Version2 (V2) to be had for NodeMCU Dev. Kit i.e. NodeMCU Development Board v1.0 (Version2), which generally is available in black colored PCB. ESP8266 is Wi-Fi enabled system on chip (SoC) module developed with the aid of using Espressif system. It is usually used for improvement of IoT (Internet of Things) embedded applications. ESP8266 comes with abilities of

→ 2.4 GHz Wi –Fi (802.11 b/g/n, assisting WPA/WPA2).
→ general-purpose input/output (sixteen GPIO).
→ Inter-Integrated Circuit (IC) serial communication protocol.
→ 10-bit ADCanalog to digital conversion
→ (SPI) Serial Peripheral Interface and serial communication protocol.
→ IS (Inter-IC Sound) interfaces with DMA (Direct Memory Access) (sharing pins with GPIO).
→ UART (on dedicated pins, plus a transmit-simplest UART may be enabled on GPIO2).
→ (PWM) pulse-width modulation Dept. Of ISE - Feb - May 2018-19

Wi-Fi Based Digital Notice Board employs a 32-bit RISC CPU primarily based totally on the Tensilica Xtensa L106 running at 80 MHz (or over clocked to 160 MHz). Has64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed via SPI. NodeMCU ESP8266 module is low cost Wi-Fi transceiver and can be used for end-point IoT developments. To communicate with the ESP8266 module microcontroller wishes to apply set of AT commands. Microcontroller communicates with ESP8266 module the usage of UART having particular Baud rate. There are many third-party producers that produce unique modules based on this chip. So, the module comes with unique pin availability options like

→ ESP-01 consists of 8 pins (2 GPIO pins) PCB trace antenna.
→ ESP-02 includes 8 pins, (3 GPIO pins) U-FL antenna connector.
→ ESP-03 includes 14 pins, (7 GPIO pins) Ceramic antenna.
→ ESP-04 have 14 pins, (7 GPIO pins) No ant

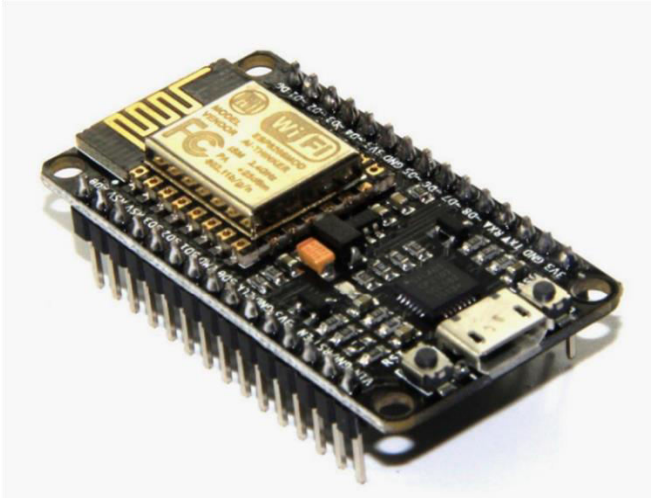

Fig -1: Node ESP8266

Programming the ESP8266 with Arduino IDE

As Arduino.cc started to broaden new MCU boards based on non-AVR processors they had to adjust the Arduino IDE that allows you to assist the spare tool. Doing so permits the Arduino C / C++ code to be compiled into those new processors. They finished this via way of means of introducing a board manager and a SAM core. When we communicate about the "core", it is a set of software program additives required by the board manager and Arduino IDE to compile the Arduino C / C++ source files into the machine language of the goal MCU. Some innovative ESP8266 supporters have evolved the Arduino middle for the ESP8266 Wi-Fi SoC, which may be discovered on the GitHub ESP8266 core web page. This is generally known as the "ESP8266 core for the Arduino IDE", which has come to be one of the main software program improvement platforms for numerous ESP8266-based modules and development boards (which includes NodeMCU).

2.2. MAX7219 dot matrix Display:

Fig -2: MAX7219 dot matrix display

Sooner or later, Arduino enthusiasts continually come upon the dot matrix display. It is a dimensional patterned LED array that is used to symbolize characters, symbols, and images. Almost all modern out of doors LED displays use dot matrices. And there may be hardly any better choice than the MAX7219 IC to control them. This is a easy and incredibly less expensive manner of controlling a single matrix. In addition they may be chained collectively to control or greater dot matrices for large projects.

MAX7219 Module Overview

There are numerous MAX7219 breakout boards available, of which are greater popular – one is the generic module and the other is the FC-16 module. A typical MAX7219 module consists of 8 ×8 dot matrix display and a MAX7219 LED display driver. Let's get to know them one by one.

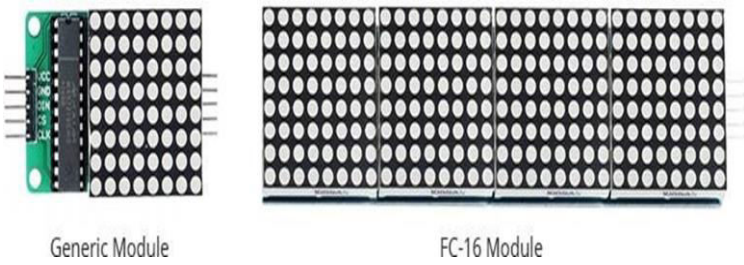

Fig -3: MAX 7219 module

The Dot Matrix Display

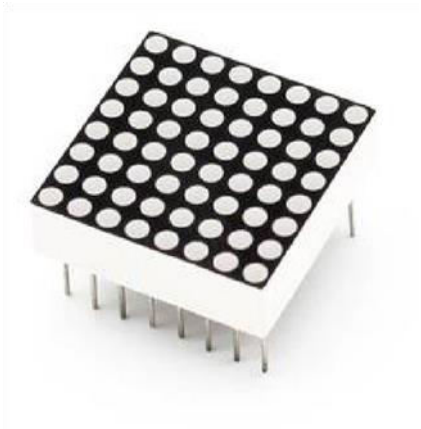

Fig -4: Dot matrix display

A typical single color 8×8 dot matrix unit has 16 pins, 8 for every row and 8 for every column. The purpose for all rows and columns being wired collectively is to lessen the range of pins required. If this had been now no longer the case, an 8×8 dot matrix unit could require 65 pins, one for every LED and one for a common anode or cathode connector. By wiring rows and columns together, only 16 pins are required. This approach of controlling a massive range of LEDs with fewer pins is known as Multiplexing

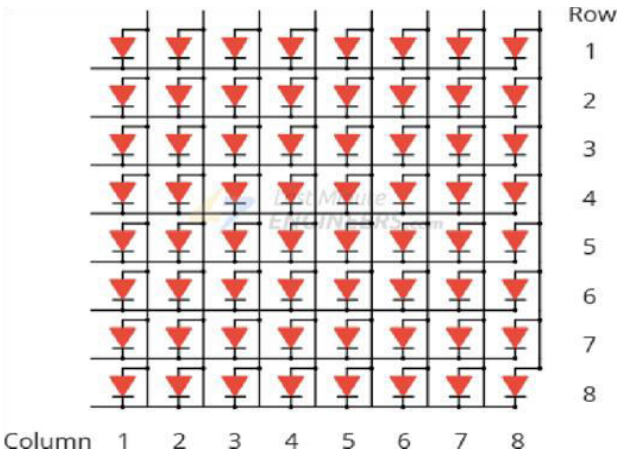

Fig -5: 8×8 dot matrix unit

In this approach, every column is activated for a very short time period and at the same time LEDs on that column are lit through addressing the corresponding row. The columns are switched so fast (hundreds or thousands of instances a second) that the persistence of the human eye perceives the display to be absolutely lit. Therefore only a maximum of 8 LEDs are lit at a time.

MAX7219 Chip

The traumatic issue approximately multiplexing is which you want to apply 8 power transistors and you need to continuously refresh the display to hold the image stable. Enter the amazing MAX7219 Chip, which does all of the control and refresh work for you. All you need to do is send it serial commands via the 4-pin SPI interface and it's going to automatically take care of the rest.

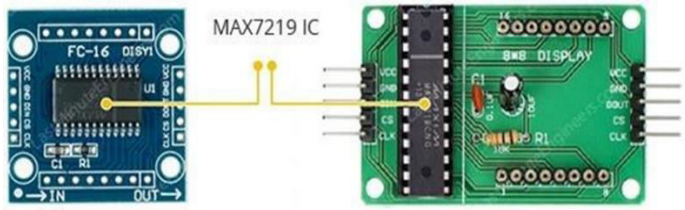

Fig -6:MAX 7219 chip

It can completely control 64 individual LEDs – which includes retaining the same brightness, and permitting you to alter the brightness of the LEDs with hardware or software program (or both). Once the display is updated through the microcontroller, the MAX7219 then looks after all of the work of refreshing the display at 800 Hz. Thereby eliminating the overhead from the microcontroller, which can be off doing different important things. You may even flip the display off for energy saving mode, and nevertheless send data while it is off. And another excellent factor is that once powered on, it continues the LEDs off, so no wacky displays for the primary seconds of operation. The MAX7219 chip communicates through the SPI interface, so it only require 3 data pins to connect with a microcontroller, and what's greater we are able to daisy-chain numerous modules collectively for large display using the same 3 wires.

Setting Maximum Current and Brightness

The MAX7219 chip permits you to alter the brightness of the display with hardware or software program (or both). To alter the brightness on the hardware level, the MAX7219 breakout board comes with a resistor (Reset)

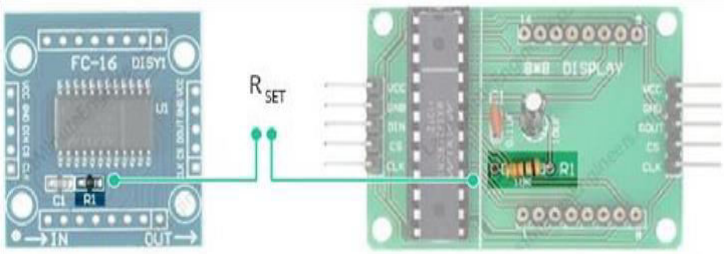

Fig -7: MAX7219 chip brightness

LED Matrix	ESP8266
VCC	3V (3V on Node MCU))
GND	GND (G on NodeMCU)
DIN	D7
CS	D8
CLK	D5

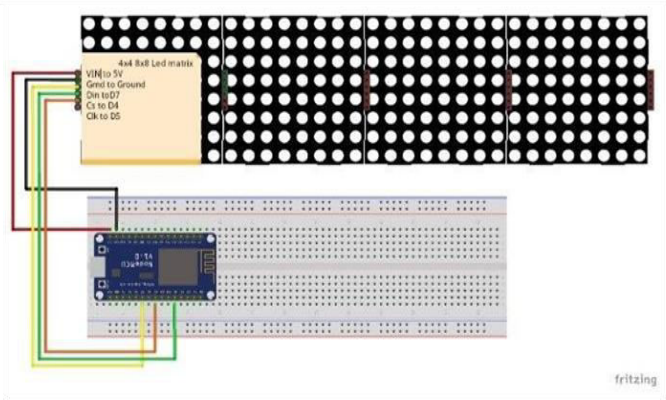

Fig -8: Connections between the LED matrix and ESP8266

2.3. Power supply

5V power supply is needed for this project. We can powered up this device through

- Batteries.
- Chemical fuel cells and different varieties of energy storage structures.
- Solar power
- Generators or alternators.

3. SOFTWARE REQUIREMENT

3.1. Arduino IDE

Arduino IDE is an open source software program this is especially used for writing and compiling the code into the Arduino Module. It is an professional Arduino software program, making code compilation too easy. It is effortlessly available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that incorporates inbuilt capabilities and commands that play a essential function for debugging, modifying and compiling the code within side the environment. The foremost code, also called a sketch, created on the IDE platform will ultimately generate a Hex File that is then transferred and uploaded in the controller on the board.

The IDE environment especially includes fundamental parts: Editor and Compiler in which former is used for writing the specified code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

Open Arduino IDE and set up the libraries, Libraries are very beneficial for including the greater functionality into the Arduino Module. There is a list of libraries you can upload as you click on the Include Library and upload the respective library it will on the top of the sketch with a #include sign. Suppose, we include the ESP8266WiFi library, it'll seem on the text editor as

#include <ESP8266WiFi.h>

Most of the libraries are preinstalled and come with the Arduino software program. However, you may additionally download them from the external sources. Next, choose the board and upload the source code. Insert your own SSID and password in the marked segment of the code.

3.2. Fritzing

Fritzing is an open source initiative to expand amateur or interest CAD software program for the design of electronic hardware, to guide designers and artists equipped to move from experimenting with a prototype to constructing a greater everlasting circuit. The software program is created in the spirit of the Processing programming language and the Arduino microcontroller and permits a designer, artist, researcher, or hobbyist to file their Arduino-based prototype and create a PCB format for manufacturing.

The related website allows customers share and discuss drafts and experiences as well as to lessen production costs. Fritzing may be seen as an electronic design automation (EDA) tool for non-engineers. The input metaphor is stimulated through the environment of designers (the breadboard-based prototype), even as the output is focused on accessible method of production. Figure No. 13 that is Connections between the LED matrix and ESP8266 is designed using this fritzing software.

4. METHODOLOGY

This project makes use of an LED dot matrix based on the MAX7219 from Maxim, that's an 8-bit LED display driver chip. These chips are designed to control as much as 8-digit 7-segment digital LED displays, bar graph displays, or 8x8 LED dot matrix displays. Arduino IDE provides a library known as Matrix with sample code written for the MAX7219 chip. Along with the MAX7219, we are able to use an ESP8266 and a dot matrix display to display messages sent over the nearby Wi-Fi network.

The system hardware includes a MAX7219 driver based 64x8 LED dot matrix display module. This module was interfaced with the NodeMCU board via 5-pins. Out of those 5 pins, 2were power pins. VCC connected to +3.3V pin of the NodeMCU board and GND was connected to the GND pin of the same board. Other 3 pins of the LED matrix module were control pins or data transfer pins. The MAX7219 driver communicated with the NodeMCU through SPI (Serial-Peripheral Interface) protocol and hence only 3 wires had been required to control the display. As the MAX7219 can control a most of 64 LEDs, the maximum size dot matrix display it can drive is 8×8 pixels. The 64 LEDs are driven through 16 output pins of the IC. The maximum number of LEDs that may be triggered at a time is actually eight.

The LEDs are organized in a matrix association of 8×8 rows and columns. So, the MAX7219 driver turns on every column periodically for a totally short period of time and on the same time it additionally drives every row. So, through rapidly switching thru those columns and rows and through exploiting the natural human eye perception limits it seemed like a continuous pattern running over the LEDs in the dot matrix display.

The communication protocol used right here is SPI and there may be usually one master device (the NodeMCU) that controls the peripheral devices or slaves (the display). We can control the display either via the NodeMCU's hardware SPI interface or 3 arbitrary digital pins can be configured as software SPI. Hardware SPI is faster than the software SPI. The SPI interface usually calls for four pins committed for signals MOSI (Master out Slave In), MISO (Master in Slave Out), and SCK (Serial Clock), SS (Slave Select). MOSI (Master out Slave In) pin of the NodeMCU is attached to the DIN pin of the LED matrix display. This Master line sends facts to the slave devices. SCK (Serial Clock) pin of the NodeMCU is attached to the CLK pin of the LED matrix display. The clock pulses which synchronize data transmission had been generated through the master. SS (Slave Select) pin of the NodeMCU is connected to the CS pin of the LED matrix display. This pin on each slave device is

controlled through the master to allow and disable a particular slave device. By the use of daisy chaining a couple of displays, one huge display can be created through simply linking DOUT (Data Out) pin of the primary dot matrix display module to the DIN (Data In) pin of the following display module. Rest of the pins like VCC, GND, CLK, and CS are shared among all the connected display modules. One greater aspect to observe down right here is that a common breakout board for the MAX7219 comes with a resistor among the +5V and the IC pin number 18. The resistor is used for setting the brightness or the current flow to the LEDs

The thesis explains the implementation of "IOT based voice controlled Notice Board" the use of Node MCU microcontroller. We can do successful implementation of this project with the aid of using following steps:

Step 1: Do the hardware connections as given, we need to join 5 pins of NodeMCU to LED display of them are VCC and GND and other 3 are data transfer pins D7, D8, D5 of Node MCU have to connect with Din, CS, CLK of LED display respectively.

Step 2: Upload the project code in Node MCU ESP8266, make sure your mobile phone share same Wi-Fi as NodeMCU.

Step 3: Open the IP address that is "192.168.4.1" in browser, we are able to see the web page from where we will send the message through voice commands

Step 4: After giving voice message, it'll be received by NodeMCU, it'll then decode it and display the given message on LED display

5. BLOCK DIAGRAM & SCHEMATIC DIAGRAM

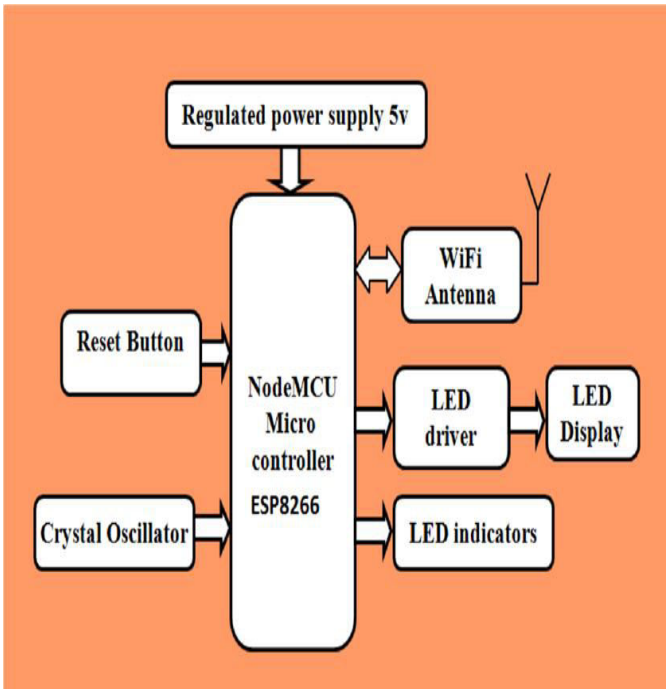

Fig -9: - Block Diagram of Project

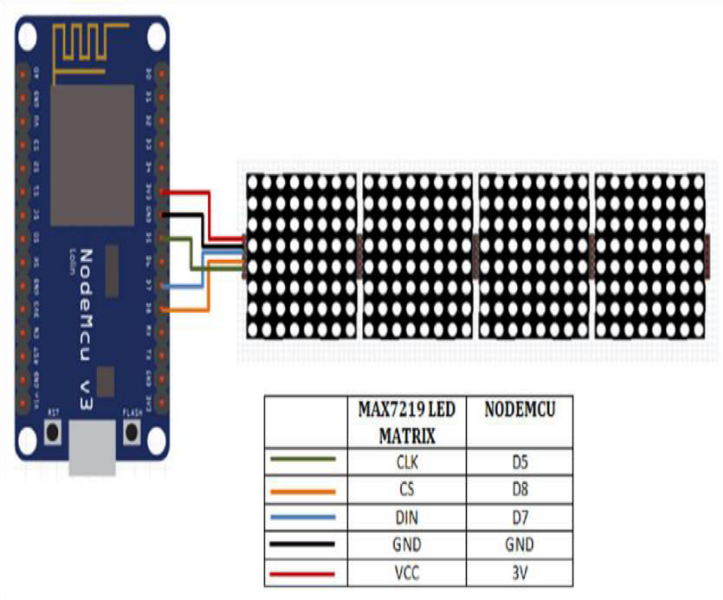

Fig -10: Schematic Diagram

6. EXPERIMENTAL SETUP & RESULT

The 64x8 LED dot matrix display module was interfaced to the NodeMCU board. The entire system was powered up through a +5V regulated dc power supply. Other DC power sources like battery, adaptors, SMPS may also be used. The current consumption of the system was very low and consequently it is able to be powered even through a laptop's USB port. As already mentioned in preceding sections it is able to be found that how a couple of MAX7219 driver ICs can be daisy chained to drive multiple LED dot matrix modules. Here an LED dot matrix display device was controlled through web browser. The source code was compiled and uploaded into the NodeMCU and powered up. The cellular hotspot in the person's mobile phone was turned ON. The system automatically established the wireless connection thru Wi-Fi hotspot and the LED dot matrix display commenced with default message string scrolling over it continuously. This system accepts the command given through the person and displayed the text on LED dot matrix display. The work was tested, calibrated and validated after multiple iterations.

Many text strings of various length had been tried and tested over this platform. Also each the Voice Commands in addition to Typing Text commands had been tested right here on the Google platform. So, the hardware system developed right here validated the work.

Fig -11: Project setup

7.CONCLUSION

Finally an application has been applied which included IoT based hardware system using Wi-Fi protocol for organising communication. In brief it was an IoT based application for controlling LED scrolling message display was effectively implemented right here on this work. This system was completely user friendly and will accept commands from specific users. This system provided user the freedom to apply both a voice based command or a text command. The developed system was highly responsive and overall performance wise much more efficient than the conventional systems available. Updating a message on the display has come to be very quick now through adopting this technique. Overall it's a worth device.

8. FUTURE SCOPE

.This work may be further extended to new levels as adoption rate of AI is increasing at a rapid pace and IOT has already captured the market. The combination of would lead to the development and implementation of lots more sophisticated systems. These systems could restrict human interventions as maximum of the responsibilities could be successfully and correctly accomplished through those smart systems only. Google Assistant based controlling of various parameters and devices could be quite beneficial for disabled sufferers or elder people. One can combine multiple sensors and domestic home equipment with the assist of IOT systems and their controlling can be made easy using those AI powered VPAs like Google Assistant over the phone.

9. REFERENCES

1. HarisIsyanto; AjibSetyoArifin; Muhammad Suryanegara, "Design and Implementation of IoT Based Smart Home Voice Commands for disabled people using Google Assistant", International Conference on Smart Technology and Applications (ICoSTA), 2020, Publisher: IEEE.

2.Tae-Kook Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant", International Conference on Electronics, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072 © 2020, IRJET | Impact Factor value: 7.529 | ISO 9001:2008 Certified Journal | Page 2652 Information, and Communication (ICEIC), 2020, Publisher: IEEE.

3.Vanathi K. Lalitha; B. Mahalakshmi; S. Madhusudan; M. Srinivasaperumal; S. Srikanth; Sathish R. Kumar, "Smart Control Of Home Amenities Using Google Assistant And Clap Switch Circuit", 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, Publisher: IEEE.

4.Avid Sheppard; Nick Felker; John Schmalzel, "Development of Voice Commands in Digital Signage for Improved Indoor Navigation Using Google Assistant SDK", IEEE Sensors Applications Symposium (SAS), 2019, Publisher: IEEE.

5. Mokh. SholihulHadi; Maulana Ahmad As Shidiqi; Ilham Ari ElbaithZaeni; Muhammad AlfianMizar; MhdIrvan, "Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API via Google Assistant", International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2019, Volume6, Publisher: IEEE.

6.İlkanYıldırım; ErkanBostancı; Mehmet SerdarGüzel, "Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and GoogleAssistant Build-In Smart Home Speakers", 4th International Conference on Computer Science and Engineering (UBMK), 2019, Publisher: IEEE.

7. M. Sundarramurthi; A. M. AnjanaSundari; AnandiGiridharan, "A Method of Designing Home Automation Control System (HACS) Using Virtual Assistant and Mobile Application", International Conference on contemporary Computing and Informatics (IC3I), 2019, Publisher: IEEE.

8.VetonKëpuska; GamalBohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)", IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, Publisher: IEEE.

9.SeptimiuMischie; LilianaMâţiu-Iovan; Gabriel GăŠpăresc, “Implementation of Google Assistant on Raspberry Pi,” International Symposium on Electronics and Telecommunications (ISETC), 2018, Publisher: IEEE.

10.AleksandarLazić; Milan Z. Bjelica; DejanNad; Branislav M. Todorović, “Google Assistant Integration in TVApplication for Android OS”, 26th Telecommunications Forum (TELFOR), 2018, Publisher: IEEE.