VOICE CONTROLLED SMART HOME AUTOMATION FOR DISABLED PERSON

Dr . Kavitha A S[1], Mr. Vishwas Rudramurty[2], Akash G[3], Syed Murtuza[4], Shreyas S K[5], Vinayak S[6]

[1]Professor, Department of CSE, HMS Institute Of Technology, Tumkur, India

[2]Asst.Professor, Department of CSE, HMS Institute Of Technology, Tumkur, India

[3]UG student, Department of CSE, HMS Institute Of Technology, Tumkur, India

ABSTRACT

Voice-controlled smart home technology is a breakthrough for people with disabilities. This paper outlines a system that lets users manage home appliances through voice commands, reducing the need for physical interaction with switches and controls. The system uses Arduino, Zigbee, and other hardware to control lights, fans, and more.Voice recognition is achieved with Python, using natural language processing and machine learning for reliable interpretation of commands. Security is enhanced with voice authentication, ensuring that only authorized users can control the system. An emergency SMS/notification feature provides added safety. Overall, this voice-controlled smart home system is affordable, flexible, and designed to help people with disabilities gain more independence at home. An android application is used to get the voice commands from user. A Wi-Fi enabled WEMOS D1 Mini board is used and the home appliances

Keywords

Smart home automation, voice control, Arduino, Zigbee, voice authentication, emergency notification, IoT.

INTRODUCTION

Home automation is one of the major growing industries that can change the way people live. Some of these home automation systems target those seeking luxury and sophisticated home automation platforms; others target those with special needs like the elderly and the disabled. Typical wireless home automation system allows one to control house hold appliances from a centralized control unit which is wireless. These appliances usually have to be specially designed to be compatible with each other and with the control unit for most commercially available home automation systems. The developed system can be integrated as a single portable unit and allows one to wirelessly control lights, fans, air conditioners, television sets, security cameras, electronic doors, computer systems, audio/visual equipment's etc. and turn ON or OFF any appliance that is plugged into a wall outlet, get the status of different sensors and take decision accordingly. The system is portable and constructed in a way that is easy to install, configure, run, and maintain. The perfect user interface still does not exist at present and to build a good interface requires knowledge of both sociology and technology fields. According to major companies that are involved in speech recognition researches, voice will be the primary interface between humans and machines in the near future. The problem lies with the situation of the elderly or disabled people.

LITERATURE SURVEY

A. Iqbal, M. Arifin, and A. Hossain, "Smart Home Appliance Control System for Physically Disabled People using Kinect and X10", in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 891-896.

H. Rashid, S. B. Osman, N. Hassan, I. U. Ahmed, R. Das, and M. Karim, "A New Design Approach of Home Automation System for Patients with Physical Disability to Reduce Water Wastage and Power Consumption using Renewable Energy" in *Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering*, 2830 September, 2017, Dhaka, Bangladesh, pp. 770-774

K. V. Sai, B. Vamshi, and V. K. Mittal, "Wireless Voice-Controlled Multi-Functional Secure eHome", *IEEE Transactions*, pp. 2235-2240.

D. Sunehra, and V. Tejaswi, "Implementation of Speech Based Home Automation System using Bluetooth and GSM" in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016*, pp. 807-813.

H. Basanta, Y. Huang, and T. Lee, "Assistive Design for Elderly Living Ambient using Voice and Gesture Recognition System" in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff Center, Banff, Canada,

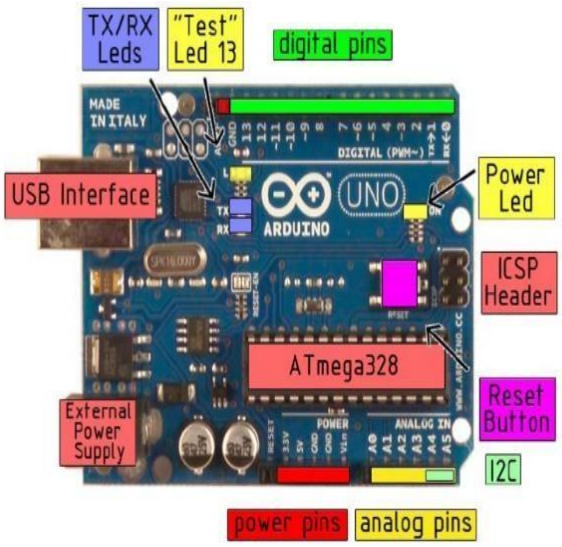
MATERIALS AND METHODS

2.1 Hardware Requirement:

- USB microphone
- ZigBee
- Relay
- Arduino UNO
- Power supply board

Arduino uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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..



positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

A brief description of the components Arduino UNO

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

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

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is

it possible to change the upper end of their range using the AREF pin and the analogReference() function.

ZIGBEE

ZigBee is the most popular industry wireless mesh networking standard for connecting sensors, instrumentation and control systems. ZigBee, a specification for communication in a wireless personal area network (WPAN), has been called the "Internet of things." Theoretically, your ZigBee-enabled coffee maker can communicate with your ZigBee-enabled toaster. ZigBee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks. ZigBee and IEEE 802.15.4 are low data rate wireless networking standards that can eliminate the costly and damage prone wiring in industrial control applications. Flow or process control equipment can be place anywhere and still communicate with the rest of the system. It can also be moved, since the network doesn't care about the physical location of a sensor, pump or valve. The ZigBee RF4CE standard enhances the IEEE 802.15.4 standard by providing a simple networking layer and standard application profiles that
can be used to create interoperable multi-vendor consumer electronic solutions.
The benefits of this technology go far beyond,
ZigBee applications include:

- Home and office automation
- Industrial automation
- Medical monitoring
- Low-power sensors
- HVAC control
- Plus, many other control and monitoring uses

RELAYS

Relays are electromechanical switches. They have very high current rating and both AC and DC motors can be controlled through them because motor will be completely isolated from the remaining circuit. An electromechanical relay contains an electromagnetic coil that moves a metal arm to make and break an electrical connection. They provide electrical isolation between the control signal and the load and are relatively low cost. No common ground connection between the control signal and load is needed. A standard digital logic GPIO output pin does not supply enough current to drive a relay coil directly. When using logic signals to control a relay, a driver circuit must be used to boost the current needed to energize the relay's electromagnetic coil. The load is switched on and off using the relay's metal contacts that move when the coil is energized. Since the metal contacts actually touch, relays will have less of a voltage drop across the switch point than transistor circuits.

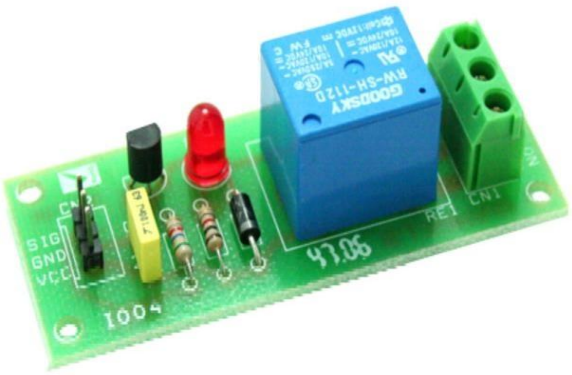


Fig 5.6: Relay motor driver

Specification of relay

- Supply input 12V DC at 75 mA
- Power LED
- Relay Output: up to 7 A
- Onboard tactile switch for direction control
- Relay based drive design with diode protection
- LED indicator for direction indication
- Power-On LED indicator
- Terminal pins and screw terminal connector for easy input / output connection
- Four mounting holes of 3.2 mm each
- PCB dimensions 41 mm x 81 mm

Regulator Power Module

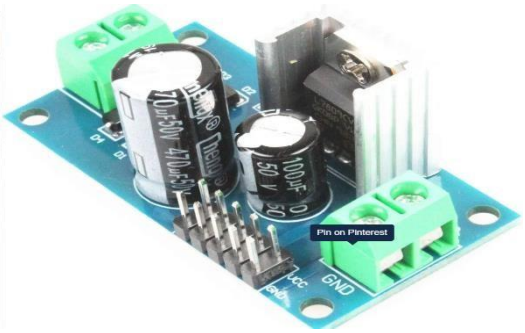
The LM7812 is a voltage regulator module that provides a stable 12V output, even if the input voltage changes. It's useful in many electronics projects where a consistent 12V is needed, like powering microcontrollers, sensors, or automotive electronics.

It has three terminals: one for input (where you connect the power source), one for output (where you get the regulated 12V), and one for ground. It can work with a range of input voltages, usually from 14V to 35V, but performs best with 18V to 24V.

The module includes safety features like protection against overheating and overcurrent, which helps keep it from burning out. It also has a low noise output, making it suitable for audio applications.

If you use this regulator, remember it can generate heat, so you might need a heat sink. Adding small capacitors to the input and output can help improve stability and reduce electrical noise.

Overall, the LM7812 is a solid choice when you need a reliable 12V power supply.

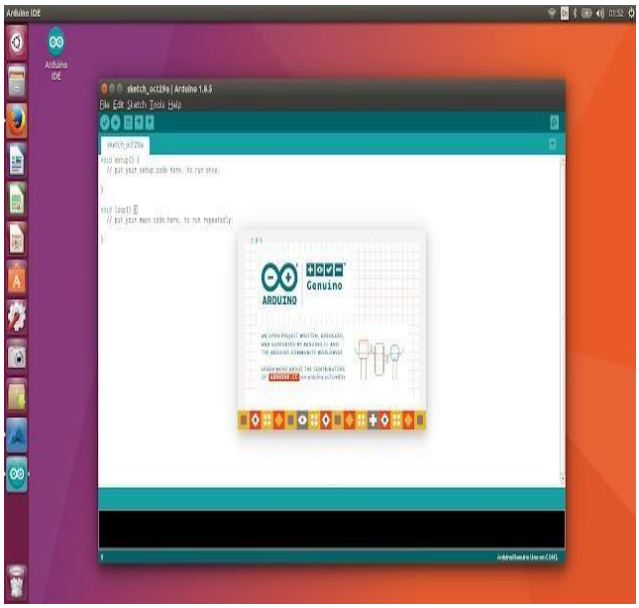


Software.

Arduino IDE

The Uno board can be programmed with the Arduino Software (IDE). For details, see the reference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).



You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog

input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

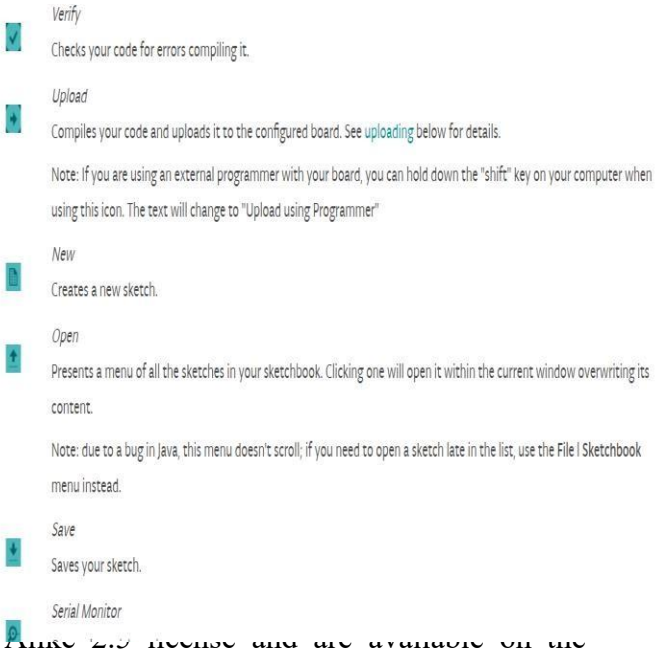
The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.[3]

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.



Arduino website. Layout and production files

for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -duino.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the on board voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the optiboot bootloader.[27] Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila,[a] Duemilanove,[b] and current Uno[c] provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs,

which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board[28] and Boarduino[29] boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python interpreters are available for many operating systems.

SYSTEM DESIGN .

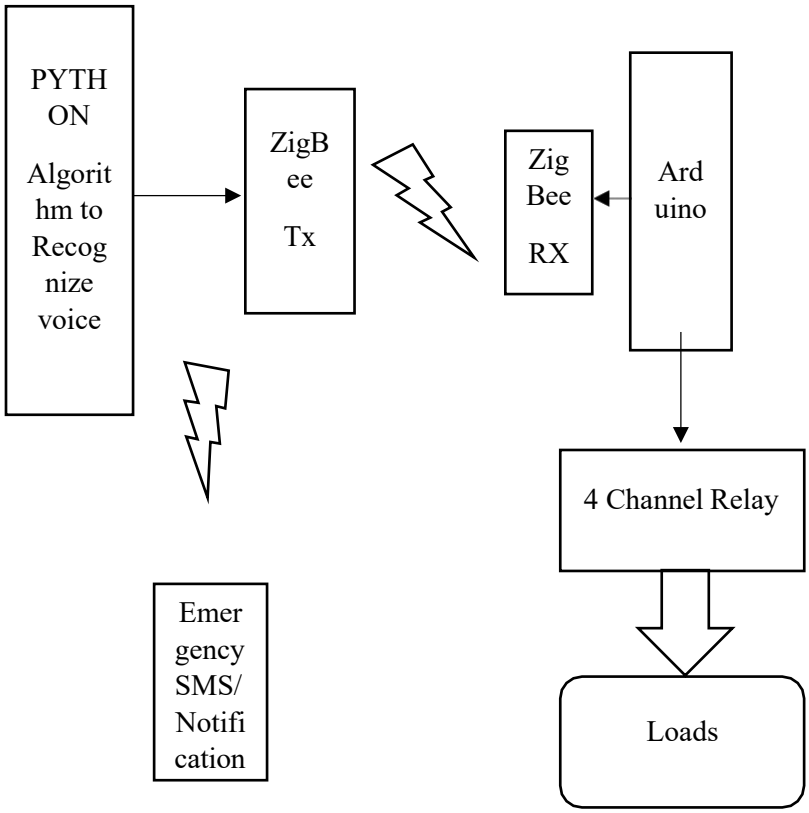


Fig 1: Block diagram of the proposed system

Requirement analysis: Understand the specific needs and requirements of the disabled person and their home environment. Speech recognition: Implement a speech recognition system that can accurately convert spoken words into text. Command processing: This involve creating a command dictionary and defining the corresponding action for each command. Integration with smart home devices: Establish communication to connect the voice control system with the smart home devices. User interface design: Create a user-friendly interface that enables the disabled person to easily interact with system using their voice commands. Testing and optimization: Test the system to ensure accurate speech recognition. Optimize the system-based user feedback and improve its performance.

REFERENCES

- Ajmeera Kiran, G R Sakthidaharan, D Divya Priya, B Uma Mahesh, K Prem Kumar, K Pavan Kumar 2022 for voice controlled home automation system using google assistants- IEEE.
- Dusi Durga Prasad, Golla Jyothi Mallika, Shaik Umer Farooq, S Pramod for voice controlled home automation, 2021 second international conference on electronics and sustainable communication systems, IEEE.
- Progress Mtshali, Freedom Khubisa for a smart home appliance control system for physically disabled people, 2019 conference on information communication technology and society, IEEE.
- Haris Isyanto, Ajib Setyo Arifin, Muhammad Suryanegara for design and implementation of IoT-based smart home voice commands for disabled people using google assistant, IEEEE.
- Tanzim Bin Khalid, Muhtasim Alam Chowdhury, Kazi Azmain Faieq for Near-field communication enabled voice and touch integrated home automation for elderly and disabled, 2020- IEE.
- Gaikwad V S, Saurabh Ghadge, Vaibhav More for Smart home automation with voice authentication using IoT, 2020
- Juel E. Titus, Kavyalakshmi P. A, Merin Job for IoT enabled home automation for differently challenged and elderly people using gesture and voice recognition, 2022- IEEE.
- M. Ebrahim Abidi, Ani Liza Asnawi, N.FM. Azmin, A.Z. Jusoh for development of voice control and home security for smart home automation, 2018- IEEE

AUTHORS

Akash G [1], student, Department of CSE, HMS Institute of Technology, Tumkur, India

Syed Murtuza[2]**,** student, Department of CSE, HMS Institute of Technology, Tumkur, India

Shreyas S K[3], student, Department of CSE, HMS Institute of Technology,Tumkur, India

Vinayak S [4], student, Department of CSE, HMS Institute of Technology, Tumkur, India