

*Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Kerbala
College of Engineering
Department of Electrical and Electronic Engineering*



DESIGN AND IMPLEMENT OF AN INTELLIGENT HUMAN- INTERACTIVE SYSTEM BASED ON VOICE RECOGNITION

*A project
Submitted to the Department of Electrical and Electronic
Engineering, University of Kerbala in a Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science in Electrical
Engineering*

by

Ali Abd Alkareem Ahmed Al-Shami

Kawther Diyha Yassen Al-Mosawi

Supervised by

Assist. Prof. Dr.

Haider Ismael Shahadi Al-Mayaly

August 2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَى
إِلَيْكَ وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا)

إهداء

الى رسول الهدى والسلام الذي لولاه ما سلكننا طريق الإسلام الى المبعوث رحمة للأنام
محمد المصطفى (صلى الله عليه وعلى إله الكرام)

الى من إنتهلنا من علومهم وإنتسبنا لهم و كنا ذريتهم أئمة الهدى و سفن النجاة (عليهم السلام) أخص
بالذكر قائمهم المهدي المنتظر (عجل الله فرجه الشريف) الذي يملئ الأرض عدلا و قسطا بعد ان
ملأت ظلما و جورا

الى رمز التضحية و الوفاء وساقى عطاشا كربلاء أبي الفضل العباس (عليه السلام)
إلى اللذين بذلوا النفس والنفيس والغالي والرخيص لأجلنا، إلى الذين لولاهم بعد الله لما كتبت هذه
الكلمات على هذه الورقة، إلى اللذين وصينا بهم حسنا
"أبأنا الأعراء رحم الله الماضين منهم وحفظ الباقيين "

إلى الذين ربينا معهم في حضن واحد، إلى الذين لا نطيق الحياة من دونهم يوما واحدا إخوتنا
وأنفاسنا التي تنفس بها.
"إخوتنا وإخواتنا"

الى الاصلاب الشامخة والارحام الطاهرة "اجدادنا وجداتنا" رحم الله الماضين منهم وأطال في
عمر الباقيين

الى "حشد الله المقدس" الذي لولاه ما كنا لنكتب هذه الكلمات
وإلى من لم تسعه ورقتي هذه ولكن ذكراه ومكانته في قلب أوسع

علي عبد الكريم احمد هاشم
كوثر ضياء ياسين فاخر

Acknowledgement

We thank God because we have succeeded in completing this research, We thank our generous families for their unlimited encouragement and support for us, we also extend my sincere thanks to the Honorable Professor, who gave us his knowledge and helped us in this achievement, the Supervisor **Assist. Prof. Dr. Haider Ismael Shahadi Al-Mayaly**.

We also thank everyone who helped and supported us for the success of this project.

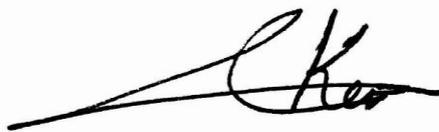
Undertaking

This is to declare that the project entitled “*Design and implement of an intelligent human interactive system based on voice recognition*” is an original work done by undersigned, in partial fulfillment of the requirements for the degree of “Bachelor of Science in Electrical Engineering” at Electrical and Electronic Engineering Department, College of Engineering, University of Kerbala.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.



Ali Abd Alkareem Ahmed Alshami



Kawther Diyha Yassen Almosawi

Abstract

With the help of Internet tools, doctors can monitor patients remotely and give medicines based on tracked information. This bed not only helps patients but will also help to advance the health system and provide medical and health status and easy handling with the patient. The proposed aim is to build a prototype bed which can respond to basic commands and helps the user. The voice recognition (VR3) module is used to detect the keywords for the trained voice when the user is passing commands to perform any of the actions such as Up, Down, etc.

Also, a mobile application has been attached to the bed to facilitate the work of the medical staff and also to shorten the time and provide the needs of the patient quickly.

List of Contents

Abstract	I
List of Contents	II
List of Figures	IV
List of Tables	V
Chapter One: Introduction	1
1.1 Overview.....	1
1.2 Smart patient monitoring.....	1
1.3 Motivation.....	2
1.4 Problem statement	2
1.5 Project objectives.....	3
1.6 Project outlines.....	4
Chapter Two: Background and Theory	5
2.1 Introduction	5
2.2 Hospital bed types.....	5
2.2.1 Stretchers	5
2.2.2 Semi-electric	5
2.2.3 Electric beds.....	5
2.2.4 Smart hospitals beds.....	6
2.3 Smart system requirements.....	6
2.3.1 Hardware requirements	6
2.3.2 Software and framework	11
2.3.3 Application of smart system in hospitals	13
Chapter Three: Interactive smart system for bed control and services providing	15
3.1 Introduction	15
3.2 Voice system attached to the electrical bed	15

3.2.1 Part of system.....	15
3.2.2 The Connect between external relays and control bed system relays	16
3.2.3 The Experiments of voice recognition module V3:	21
3.3 The services providing at the bed	23
3.3.1 Connect Method	23
3.3.2 Nurse Application	32
3.3.3 Feature of application.....	32
3.4 General block diagram of system	33
Chapter Four: Result and Discussion	34
4.1 Introduction	34
4.2 Test Bed	34
4.3 Test and problem of part of the voice system	36
4.3.1 Voice recognition module V3 kit.....	36
4.3.2 Arduino Mega and ESP32	36
4.3.3 Application	37
Chapter Five: Conclusion and Future work	39
5.1 Conclusion	39
5.2 Future work	39
References	A

List of Figures

Fig. 2.1 Arduino Mega	7
Fig. 2.2 ESP-WROOM-32	8
Fig. 2.3 Voice Recognition V3.....	9
Fig. 2.4 SIM808.....	9
Fig. 2.5 Relays	10
Fig. 2.6 Actuator	11
Fig. 2.7 Arduino UNO logo	11
Fig. 2.8 ThingSpeak logo	12
Fig. 2.9 MIT APP INVENTOR logo	13
Fig. 3.1 Principle operation of relay.	16
Fig. 3.2 Block Diagram Connect between Relays.....	17
Fig. 3.3 Circuit of Voltage Divider.....	18
Fig. 3.4 Relays bed system.....	19
Fig. 3.6 connection of emergency pin.....	25
Fig. 3.7 General block diagram	33
Fig. 4.1 The wave signals from remote control of bed. (a) Right side motion, (b) Left side motion, (c) Rise leg motion, (d) Down leg motion, (e) Rise back motion, (f) Down back motion, (g) Open bath motion, (h) Close bath motion, (i) Wheelchair motion, (j) Rest mode.....	36
Fig. 4.2 flowchart of turn on/off emergency pin	Error! Bookmark not defined.
Fig. 4.3. Flowchart of check services	Error! Bookmark not defined.
Fig. 4.4 view of MIT app, (a) view of designer, (b) view of blocks	37
Fig. 4.5 (a) view of application when all services is appearing, (b) the bed bottom will be green color when receive any services, (c) the bed bottom will be red color when emergency pin is active.....	38

List of Tables

Table 3.1 Connection between Arduino and Feedback of motor in bed	18
Table 3.2 Arrangement Relays bed system	19
Table 3.3 function of each relay in the main control board	20
Table 3.4 : connection between VR3 and Arduino	20
Table 3.5 Groups of VR3	22
Table 3.7 connection pins of service at ESP32 and Arduino	24
Table 3.8 test and emergency pins.....	25

Chapter One: Introduction

1.1 OVERVIEW

Technology has become the center of the main life stream. Which Smart Systems is a new generation of computing systems and information architecture that when combined with artificial intelligence, machine learning and Internet of Things technologies are breaking away from today's information, computing and telecom (ICT) paradigms to enable intelligent real-world physical systems to be integrated onto networks and the data from machines, sensors, video streams, maps, people, news-feeds and more to become an integral part of all information systems [1].

Hospital beds are the place most patients spend most of their times which need firm awareness regarding help patients remain safe while keen capacities help attendants investigate data and improve patient care. A definitive goal is to give patients a specific independency accordingly, enabling them to take some indispensable activities when nurses or care-takers are late or inaccessible [2].

For that is important to insert the smart system with hospital beds to get smart hospital beds.

The distinguishing characteristic of Internet of Things in the healthcare system is the constant monitoring a patient through checking various parameters and also infers a good result from the history of such constant monitoring. [3].

1.2 SMART PATIENT MONITORING

The healthcare monitoring systems has emerged as one of the most vital system and became technology oriented from the past decade.

Lack of medical care to the patients at right time causes various illness that lead to unexpected death.

The primary goal was to develop a reliable patient monitoring system using IoT so that the healthcare professionals can monitor their patients to avoid the unexpected death, which can use the integrated healthcare system based on IoT at hospital or at home with ensuring patients are cared for better ,in addition, patients and their family members feel comfort knowing that they are being monitored and will be supported if a problem arises, so according to the constitutions of World Health Organization (WHO) the highest attainable standard of health is a fundamental right for an individual [4] .

Patient monitoring is the most extended group among comprehensive proposals for the care of high-risk or long-term patients.

1.3 MOTIVATION

Basically, IOT is the internetworking of electronic devices to enable exchange of data between devices for specific domain applications. This concept of internetworking in internet of things (IOT) makes human life much easier than before [5].

So smart healthcare it solves many types of problem that make patient life in danger. It also has the potential to reduce healthcare costs significantly and enhance the quality of life of patients [6].

In addition, specifically the medical bed enabling efforts that are more efficient for caregivers and more environments that are responsive for patients. smart technology to hospital beds to increase patient comfort and reduce the incidence of preventable problems [7].

1.4 PROBLEM STATEMENT

Problems related to hospital beds:

1. A nurse only is always on duty at the nurse's central area outside the ICU hospital rooms. If a patient needs help, the patient can press an attend button on the bedside, and an audible alarm should alert the nurse to come the patient's aid. If two or more patients press their buttons at the same time, that will cause delay and error in patient care.
2. The amount of data is too complicated and too large, which leads to difficulties in data sharing and communication and ability to analyze information from this big data.
3. Data is not saved securely.
4. The electric bed available on the market is very expensive with difficult maintenance.

1.5 PROJECT OBJECTIVES

The objectives of this project it to solve same of problem:

1. To improve existing patient care procedures and introduce new features such as smart electrical bed which provides more comfort for the patient than the regular hospital bed.
2. To provides guarantee the privacy of patient data.
3. To avoid difficulties in data sharing with doctors by transfers all patient data to the supervisor, especially in those cases, which require intense care.
4. To provide easy uses of technology for Patients people.
5. To Reducing costs with high efficiency.
6. To provide a system be easy to maintain with high efficiency.
7. To development of the regular electric bed.

1.6 PROJECT OUTLINES

The outlines of this project are to implementation of a smart system that works with voice commands, by use Specific microcontrollers that achieve this smart system.

In addition, this system attaches with regular electrical bed to achieve this integrated smart system that helps patients, the duration of the project will not exceed one year.

So, we divided the reset chapters in the following style:

- I. Chapter Two (background and Theory): in this chapter we discuss the kits of the project and its background.
- II. Chapter Three (Design and Implementation of the smart bed system): in this chapter we obtain our work and the circuits used in the project.
- III. Chapter Four (Results and Discussion): We view the results of our work.
- IV. Chapter Five (Conclusion and Future Work): We will explain possible developments of this project.

Chapter Two: Background and Theory

2.1 INTRODUCTION

To begin with the project, let's get the idea of all the components that we used for the project. It is very important to know all the information about both hardware and software specifications.

In this chapter, it is mainly focus on the background of smart system and employ it in the medical field to help the patients, and hospital bed types.

2.2 HOSPITAL BED TYPES

the hospital beds differ on the types of adjustments that are possible and how to accomplish those adjustments.

There are many types of hospital bed:

2.2.1 *Stretchers*

The first type of hospital bed is the stretchers, commonly used in the emergency room unit. This is designed specifically for the purpose of mobility. It is also equipped with folding legs so that patients can be transferred from the ambulance to the emergency room or from home to ambulance without any hassle [9].

2.2.2 *Semi-electric*

hospital bed is a bed that has electronic function of elevating the head and the leg portions, but the overall height must be adjusted manually [10].

2.2.3 *Electric beds*

These beds are designed using a button on the side rails which enable to lower and raise the bed easily. Patients can do this without any

help of others. But the disadvantage of full electric beds is slightly higher costs [9] .

2.2.4 Smart hospitals beds

Smart hospital beds contain sensors for body temperature, heartbeat, blood, oxygen and pressure sensors, among others. All of these signals are required and necessary for the doctors to monitor the health of the patients. This system is located in the hospital beds and transfers all patient signals to the supervisor, especially in those cases, which require intense care.

But this beds have high costs [11] .

2.3 SMART SYSTEM REQUIREMENTS

2.3.1 Hardware requirements

The hardware components that use are:

2.3.1.1 Microcontrollers

1. Arduino Mega.

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560; it has 54 digital input/output pins. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC to-DC adapter or battery to get started.

It operating at 5 volt and the DC Current per I/O Pin is 40 mA

In addition, 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 [12]. The Fig.2.1 below shown Arduino Mega.



Fig. 2.1 Arduino Mega

2. ESP32.

ESP32-WROOM-32 is a powerful that targets a wide variety of applications ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding [13].

It has two CPU cores that can be individually controlled, The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection [14].

It operate at 3.3V, and the absolute maximum current drawn per GPIO is 40mA and has 38 pins, so to restart the ESP32 use EN pin connected to a pushbutton (Enable (EN) is the 3.3V regulator's enable pin), Boot pin for reset [15] . the Fig.2.2 below shown ESP.



Fig. 2.2 ESP-WROOM-32

2.3.1.2 Sensors

A sensor is a device, module, machine, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics[16]. The sensors used in this project are explained as follows:

1.Voice Recognition V3

Voice Recognition Module is a compact and easy-control speaking recognition board, product is a speaker-dependent voice recognition module. It supports up to 80 voice commands in all. Max 7 voice commands could work at the same time. Any sound could be trained as command. Users need to train the module first before let it recognizing any voice command.

It has 2 controlling ways: Serial Port (full function), General Input Pins (part of function). General Output Pins on the board could generate several kinds of waves while corresponding voice command was recognized [17]. The Fig.2.3 shown VR3.



Fig. 2.3 Voice Recognition V3

2.SIM 808

SIM808 module is a GSM and GPS two-in-one function module. It is based on the latest GSM/GPS module SIM808 from SIMCOM, supports GSM/GPRS Quad-Band network and combines GPS technology for satellite navigation.

It features ultra-low power consumption in sleep mode and integrated with charging circuit for Li-Ion batteries, that make it get a super long standby time and convenient for projects that use rechargeable Li-Ion battery, it controlled by AT command via UART and supports 3.3V and 5V logical level, it consumption 1mA in sleep mode [18] . the Fig.2.4 below is shown the SIM808.

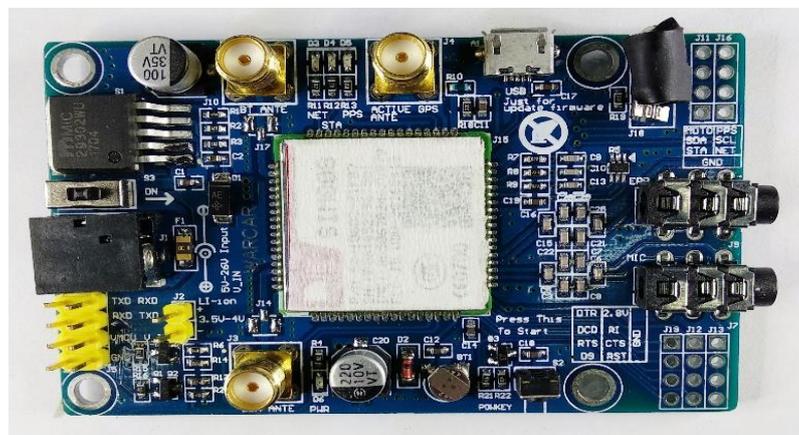


Fig. 2.4 SIM808

2.3.1.3 Relays

The relay module is an electrically operated switch it consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof [19].

Relays consist of three pins normally open pin, normally closed pin, common pin and coil. When coil powered on magnetic field is generated, the contacts connected to each other [20]. The Fig.2.5 is shown Relay.



Fig. 2.5 Relays

2.3.1.4 Actuator

An actuator is a component of a machine that is responsible for moving and controlling a mechanism or system, for example by opening a valve. In simple terms, it is a "mover".

It requires a control signal and a source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power. Its main energy source may be an electric current, hydraulic fluid pressure, or pneumatic pressure. When it receives a control signal, an actuator responds by converting the source's energy into mechanical motion [21]. The Fig.2.6 below is shown the actuator.



Fig. 2.6 Actuator

2.3.2 Software and framework

Software is a collection of data or computer instruction that tell the computer how to work [22].

In this project, the software and framework use:

2.3.2.1 Arduino IDE Environment

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software; this software can be used with any Arduino board. [23]. The Fig.2.7 below shown the logo of Arduino.



Fig. 2.7 Arduino UNO logo

2.3.2.2 IoT-Based Web Services thingspeak.com

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts [24].

In addition, Users of the free option will be limited to sending no more than 3 million messages each year to the ThingSpeak service, also be limited to 4 channels [25]. The Fig.2.8 below shown the logo of ThingSpeak.



Fig. 2.8 ThingSpeak logo

2.3.2.3 MIT app inventor

App Inventor lets you develop applications for Android phones using a web browser and either a connected phone or emulator. The App Inventor servers store your work and help you keep track of your projects [26].

The application build by work with [26]:

- a. The App Inventor Designer, where you select the components for your app.
- b. The App Inventor Blocks Editor, where you assemble program blocks that specify how the components should behave.

If you do not have an Android phone, you can build your apps using the Android emulator, software that runs on your computer and behaves just like the phone, and it supported for Mac OS X,

GNU/Linux, and Windows operating systems, and several popular Android phone models. Applications created with App Inventor can be installed on any Android phone [26].

MIT App Inventor does not work with Internet Explorer. For Windows users, we recommend using either Chrome or Firefox as your browser for use with MIT App Inventor, also can Installing the Windows software for it [27]. The Fig.2.9 below shown the logo of APP Inventor.



Fig. 2.9 MIT APP INVENTOR logo

2.3.2.4 TUNIOT

Is a code generator for the ESP8266\ESP32 boards, and do not need any coding skills to program it and make your IoT project. The tool is available on 7 languages and in active development and documentation is available [28].

NOTE:

All these software programs support with Arduino ID.

2.3.3 Application of smart system in hospitals

The applications of smart healthcare [29]:

1. in fitness tracking through a smart watch, along with parameters such as number of calories burned, steps taken, active hours.
2. Important to track the ph sensitivity of the sweat, oxygen intake of the body, heart rate monitoring.
3. Heart rate monitoring and remote ECG monitoring through

wearable's, have offered cost effective solutions in smart healthcare.

4. Monitoring the first responder's health status in an endemic or epidemic outbreak, getting ambulance assistance in case of emergency, developing evacuation schemes for disaster management in hospitals.
5. Maintaining active databases to ensure correct delivery of organs/blood to the users in need, accurate billing of surgical procedures through RFID tags are some of the significant applications in environmental management.

Chapter Three: Interactive smart system for bed control and services providing

3.1 INTRODUCTION

This project is about developed the usual hospital bed to make it work with voice commands and with the services attached.

This bed is an electrical bed that controlled it by wire control, and it includes four motors.

Classifying the motor based on the movements it performs.

1. Back and bath motor has five wires: two for power and other two-represent feedback of motor, one be always 29 volts.
2. Legs and the side motor have six wires: two for power and the other three represent the feedback of motor, (this type of motor has three feedback because it has a middle case of a move) and lest one is always 29 volts.
3. Perform the chair case is the three motors. side, back and legs motor.

The part of control board of electrical bed that we used it in our development.

1. Control system: include 12 relays each 3 relays connect with one motor.
2. Power system: convert 220 AC voltage to 29 DC voltage.

3.2 VOICE SYSTEM ATTACHED TO THE ELECTRICAL BED

3.2.1 *Part of system*

1. Arduino Mega.
2. Voice recognition module V3.
3. ESP32.
4. Relays.
5. Power part.

In this system uses 12 relays that a same to the number of relays in the control system bed.

Note:

The relays contain part (NO, NC, COM, coil)

NO: normally open.

NC: normally close. The Fig.3.1 shown the principle of relay.

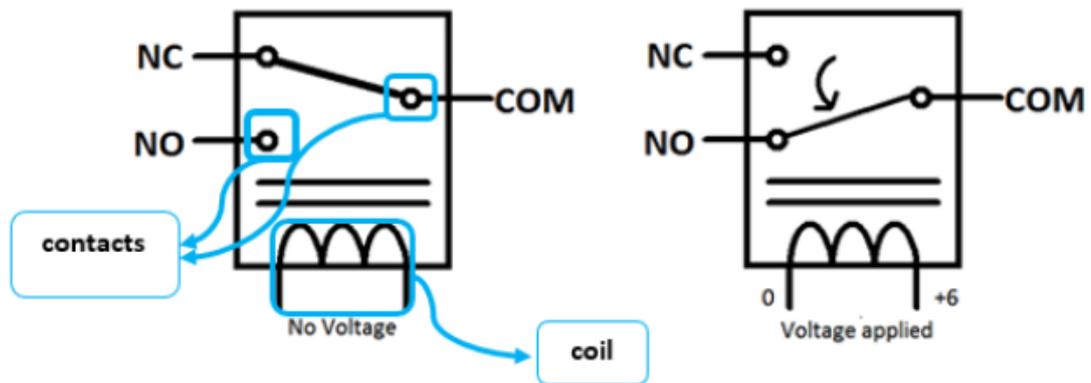


Fig. 3.1 Principle operation of relay.

3.2.2 The Connect between external relays and control bed system relays

We control the motor by using external relays to control internal relays. So, we connect one side of internal relay coil to the external relay COM and other side connected with the feedback of motor which gives either (0 Volt or 29 Volt), connects the NO in external relays with GND (of 29 volt) at control bed system relays. This method is applying to every 12 relays. The Fig.3.2 Shown the connection between relays.

So, to be able to control the voice system need to connect the microcontroller Arduino, connect at least 5 feedback of motor from the control bed system with the input pins of Arduino, between the inputs pins and feedback of motor connects voltage divider to get 5 volts to Arduino. The Table 3.1 show the pins connection.

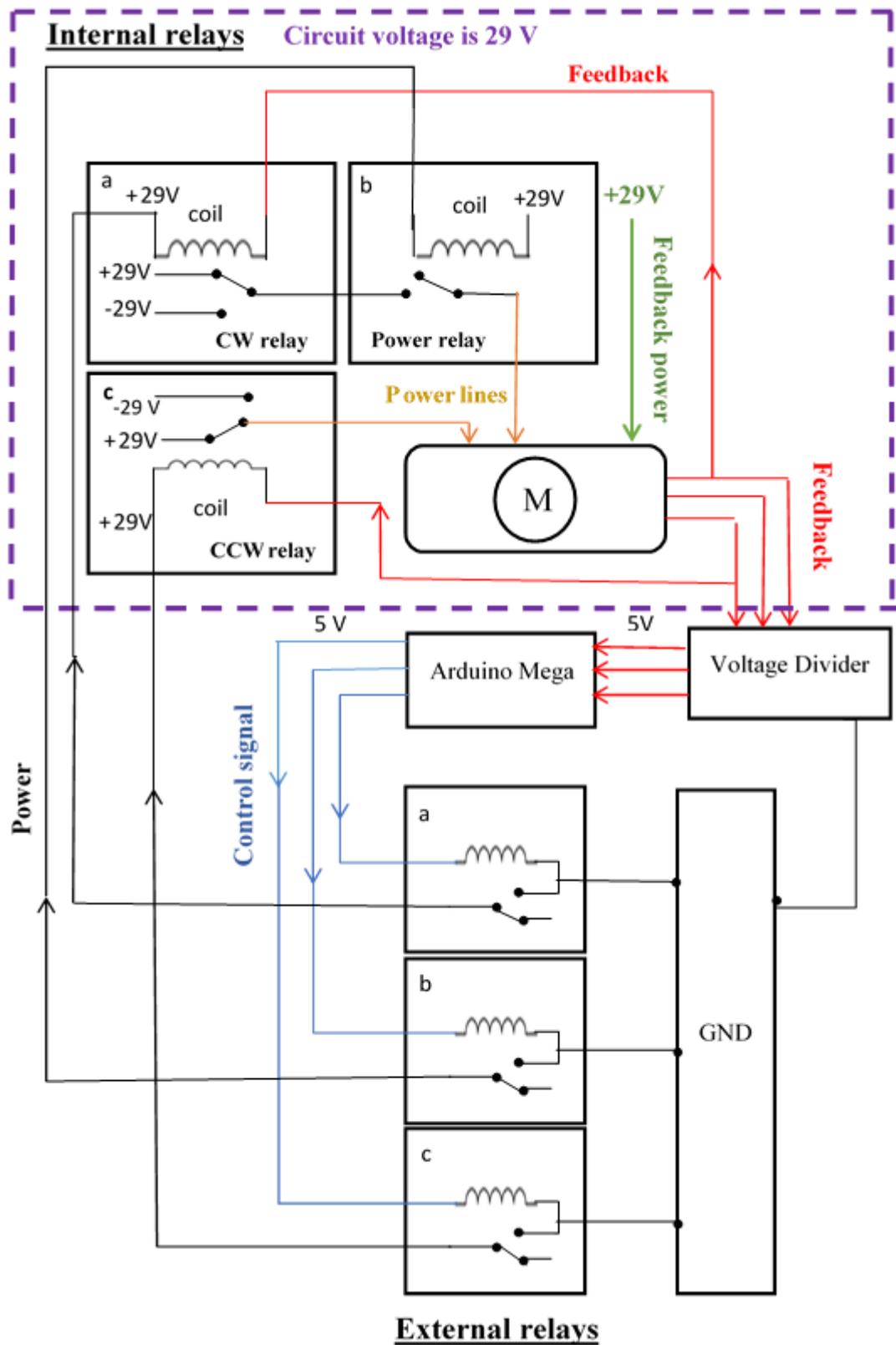


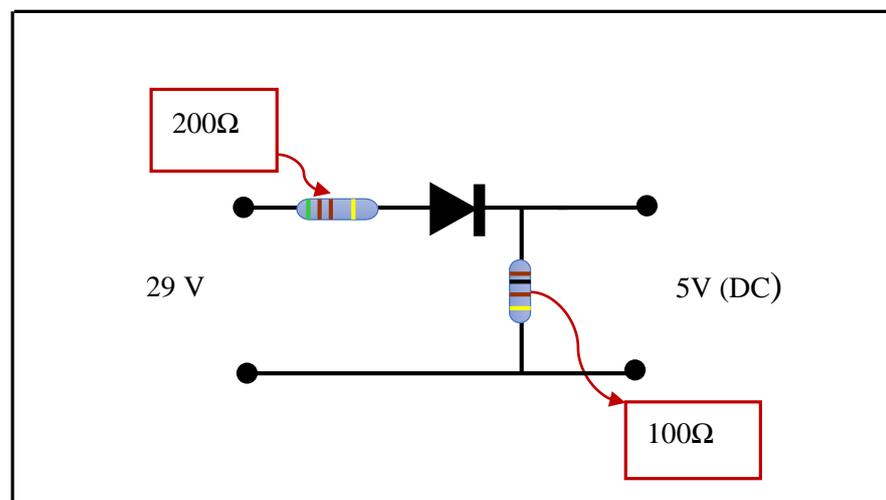
Fig. 3.2 Block Diagram Connect between Relays

Table 3.1 Connection between Arduino and Feedback of motor in bed

Pin of Arduino	Feedback of motor
Pin 40	Middle side
Pin 41	Right side
Pin 42	Middle leg
Pin 43	Up leg
Pin 44	Down back

Note:

The value of the resistors used in the voltage divider is 100 ohms and 500 ohms; (get 5 volts from 29 volts).and in this voltage divider use diode to prevent passes current in the reverse direction. The circuit shown in Fig.3.3.

**Fig. 3.3 Circuit of Voltage Divider**

After connecting the external relays with control, bed system relays connect it with Arduino Mega. Table 3.2 shown the pins uses in connection and Fig.3.4 below is shown the arrangement of bed system relays.

Table 3.2 Arrangement Relays bed system

bed system relays	External relays	Arduino Mega pins
A	1	22
B	2	23
C	3	24
D	4	25
E	5	26
F	6	27
G	7	28
H	8	29
I	9	30
J	10	31
K	11	32
L	12	33

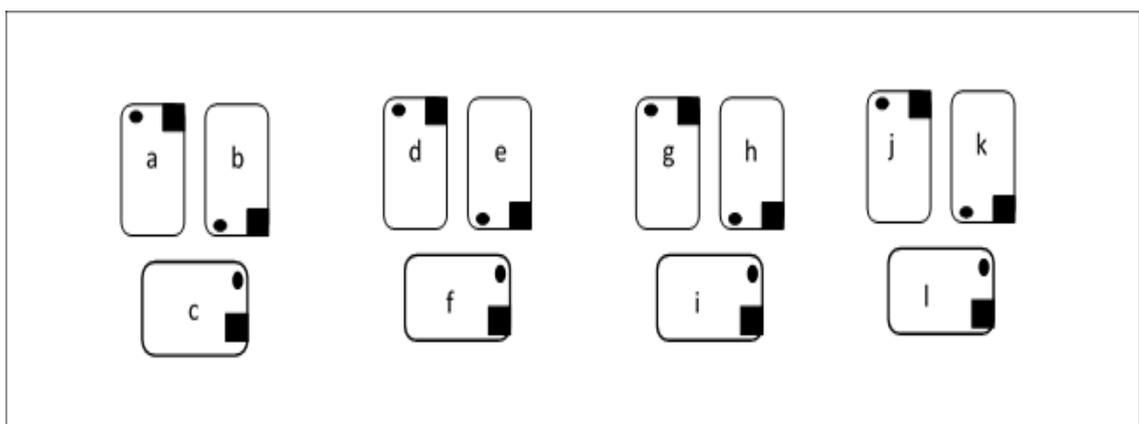
**Fig. 3.4 Relays bed system**

Table 3.3 function of each relay in the main control board

Sides		Legs		Head		Bed pan	
Right	Left	Down	Up	Down	Up	Close	Open
$b + c$	$a + b$	$e + f$	$d + e$	$h + i$	$g + h$	$k + f$	$j + k$

The above table 3.3 illustrate the function of each relay in the main control board and we depend on this table in design the voice controller.

Note:

The relay module used in this project is 16 relays and we need only 12 relays (external relays). Therefore, to control the other relays that do not necessary in work, connect it to Arduino to make it turn off always.

Responsible for voice commands is Voice Recognition module V3, this module need to installed it library in Arduino ID from Github [17].

Link of Voice Recognition V3 library [30]

After install library of it in Arduino, connect the Voice Recognition V3 with Arduino. The Table 3.3 shown the pin connection between VR3 and Arduino.

Table 3.4 : connection between VR3 and Arduino

Voice Recognition V3	Arduino mega
5 V	5V
TX	Pin 10
RX	Pin 11
GND	GND

Note:

To call commands send (load with index of voice record). And say your word to see if the Voice Recognition Module can recognize your words. If the voice is recognized, you can see.

3.2.3 *The Experiments of voice recognition module V3:*

3.2.3.1 Control LED sample

Control the LED on Arduino board connecting led to pin13 through voice commands. Before this example, need to train the VR module first.

- I. sigtrain 0 on Train the voice command used to light on the LED
- II. sigtrain 1 off Train the voice command used to turn off the LED

the steps of train:

1. Open **vr_sample_control_led** (File → Examples → VoiceRecognitionV3 → vr_sample_control_led)
2. Choose right Arduino board (Tool → Board, Mega recommended), Choose right serial port.
3. Click **Upload** button, wait until Arduino is uploaded.
4. Open **Serial Monitor**. Set baud rate 115200.

3.2.3.2 Control Motor sample

The way is same with control LED sample but use motor connection with relay. and control the direction of motor with voice command of VR3.

Note:

Because the Maximum voice commands, effective at the same time is seven, so we need group combinations to get more voice commands, and this achieve by using a word that is interplay between groups, and this word is Sumer (the name of bed). Table 3.4 shown the divide of groups.

Table 3.5 Groups of VR3

The groups	Voice index	Commend index
Group 0	→Back (0) →Leg (1) →Sides (2) →Bedpan (3) →Wheelchair (4) →Bed (5) →help (6)	→Group 1 →Group 1 →Group 2 →Group 4 →Group 0 →Group 0 →Group 3
Group 1 (back, legs)	→rise (7) →down (8) →set middle (9) →stop (10) →Sumer (11)	→Group 1 →Group 1 →Group 1 →Group 1 →Group 0
Group 2 (sides)	→left (12) →set middle (9) →right (13) →stop (10) →Sumer (11)	→Group 2 →Group 2 →Group 2 →Group 2 →Group 0
Group 3 (help)	→nurse (14) →medication (15) →sterilizer (16) →water (17) →food (18) →Sumer (19) →enquiry (20)	→Group 0 →Group 0 →Group 0 →Group 0 →Group 0 →Group 0 →Group 0
Group 4 (bath)	→open (21) →close (22) →stop (10) →Sumer (11)	→Group 4 →Group 4 →Group 4 →Group 0

Group 0 consider the main group which select the other groups from it, and the other groups consider subgroup, the switch word Sumer be in the subgroup and this word return to the main group (group 0).

The Working mechanism of groups:

The program starts at the main group. So, when the patient calling the Secondary group such as legs group by say the group name this group Contains some commands as shown in the above table 3.6 , when to choose up the legs move up and at this case, two relays are operation (relay d and e) according to table 3.3, or down also legs move down also two relays be the response to make this move (relay e and f) according to table 3.3, or choose Sumer this lead to going for the main group and when choose stop this case lead to stop the motors that response of legs motion and remain in the same group.

3.3 THE SERVICES PROVIDING AT THE BED

The services that add to the smart bed is a voice service that helps the medical staff and provides more comfort for the patient.

To achieve these services, the smart bed was connected to the Internet with ThingSpeak cloud, in ThingSpeak make a channel, each channel represents one bed.

3.3.1 Connect Method

The ThingSpeak connect with ESP32 over Wi-Fi by adding the name of network and password at the code of Arduino to make establish between ESP32 and the ThingSpeak, and need the ID of the channel and the write API key that uses at ThingSpeak to discernment between the channels that use.

After that connect the ESP32 with Arduino Mega to achieve the services by use voice because the Voice Recognition module V3 connects with Arduino Mega.

The type of connection between the ESP32 and Arduino Mega is analog connection by use the pulse of modulation PWM. the Table 3.5 shown the pins connection between ESP32 and Arduino.

Table 3.6 connection pins of service at ESP32 and Arduino

Arduino Mega		ESP32	
Pins	Services	Pins	Services
Pin 13	Nurse	Pin 12	Nurse
Pin 12	Medication	Pin 14	Medication
Pin 9	Sterilizer	Pin 27	Sterilizer
Pin 8	Water	Pin 26	Water
Pin 7	Food	Pin 25	Food
Pin 6	Enquiry	Pin 33	Enquiry

Note:

At the code of Arduino when using analog connection use analogWrite (pins number, 170), 170 mean voltage nearly equal to 3.3 volts (operation voltage of ESP32) according to this equation.

$$\frac{255}{x} = \frac{5V}{3.3V}$$

Also, connect LED with ESP32 that be turn on when any service be active.

Also, use test pin with ESP32, this pin must be active high before the service, meaning the services do not achieve if the test pin is active low.

Also, put the switch for an emergency this switch connects shown in Fig.3.5 and Table 3.6 shown pins connection of emergency and test pins.

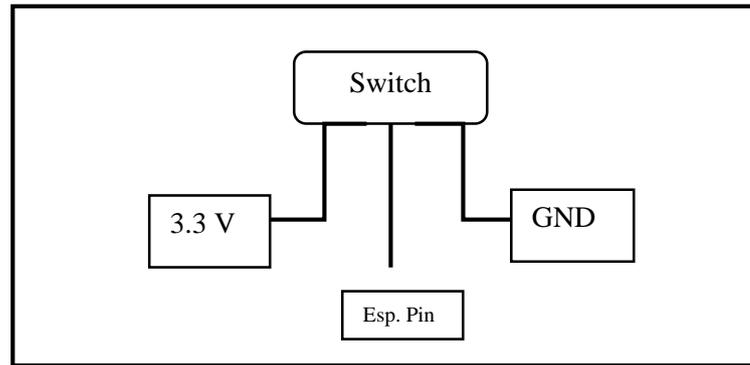


Fig. 3.5 connection of emergency pin

Table 3.7 test and emergency pins

Pin 13	LED
Pin 35	Test Pin
Pin 34	Emergency pin

3.3.2 Flowchart

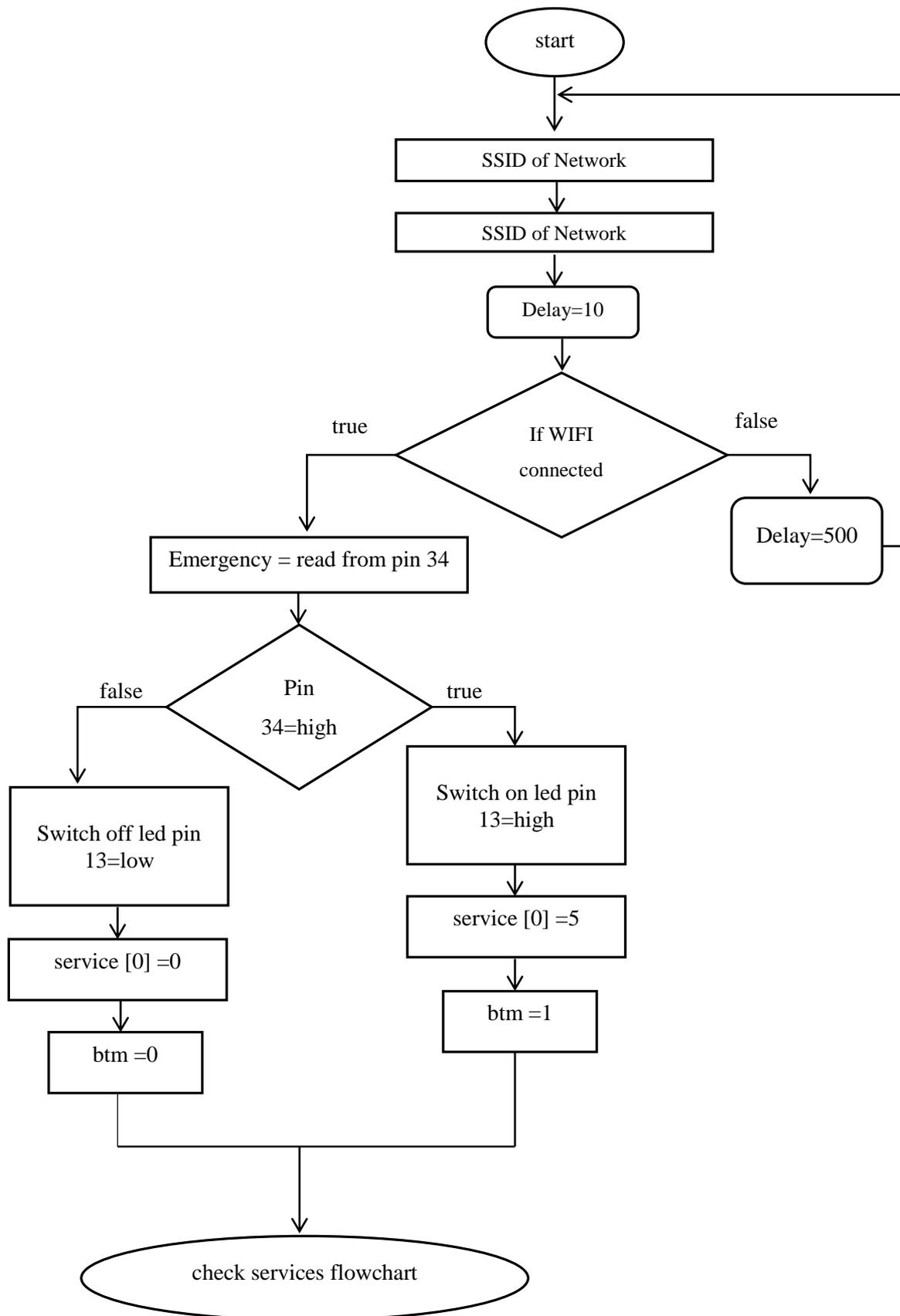
the flowchart describes the basic operation of the component, in this chart we will describe the esp32 operation, and because of the page limited we discuss it in many parts:

3.3.2.1 The flowchart of turn on/off emergency pin

the algorithm steps as follows:

- Step 1: connection ESP32 with the network by SSID of network and enter password
- Step 2: check if ESP32 connected with Wi-Fi or not, if it connects go to step 3, else after delayed by 500ms return to step 1.
- Step 3: read emergency pin from ESP32 by use pin 34.

- Step 4: check if the emergency pin is high or not, if it high it will "turn on" led that connected with pin 13 (pin 13= high), else "turn off" led (pin 13=low).
- Step 5: when the pin 13 is high, the services at " index 0" will take value equal to "5" this value to differentiation about services, also the counter "btm" take a value equal "1", else the service and "btm" take "0".
- Step 6: repeat the steps 5 if the pin 13 still high.
- Step 7: go to next flowchart of check services

**Fig. 3.6** flowchart of turn on/off emergency pin

3.3.2.2 Flowchart of check services

the algorithm steps as follows:

- Step 1: read test pin from pin 35 in ESP32
- Step 2: check if test pin active high or not, if its high go to step 3, else return to step 1.
- Step 3: check service pin, if its high the service achieve by take value equal "1", else the service take value equal "0" mean no service.
- Step 4: repeat step 3 for all services, and each service pin be active high or not, go to other services to check.
- Step 5: when all services check return to start.

Note: any service pin able to read and to store its value only when test pin is high.

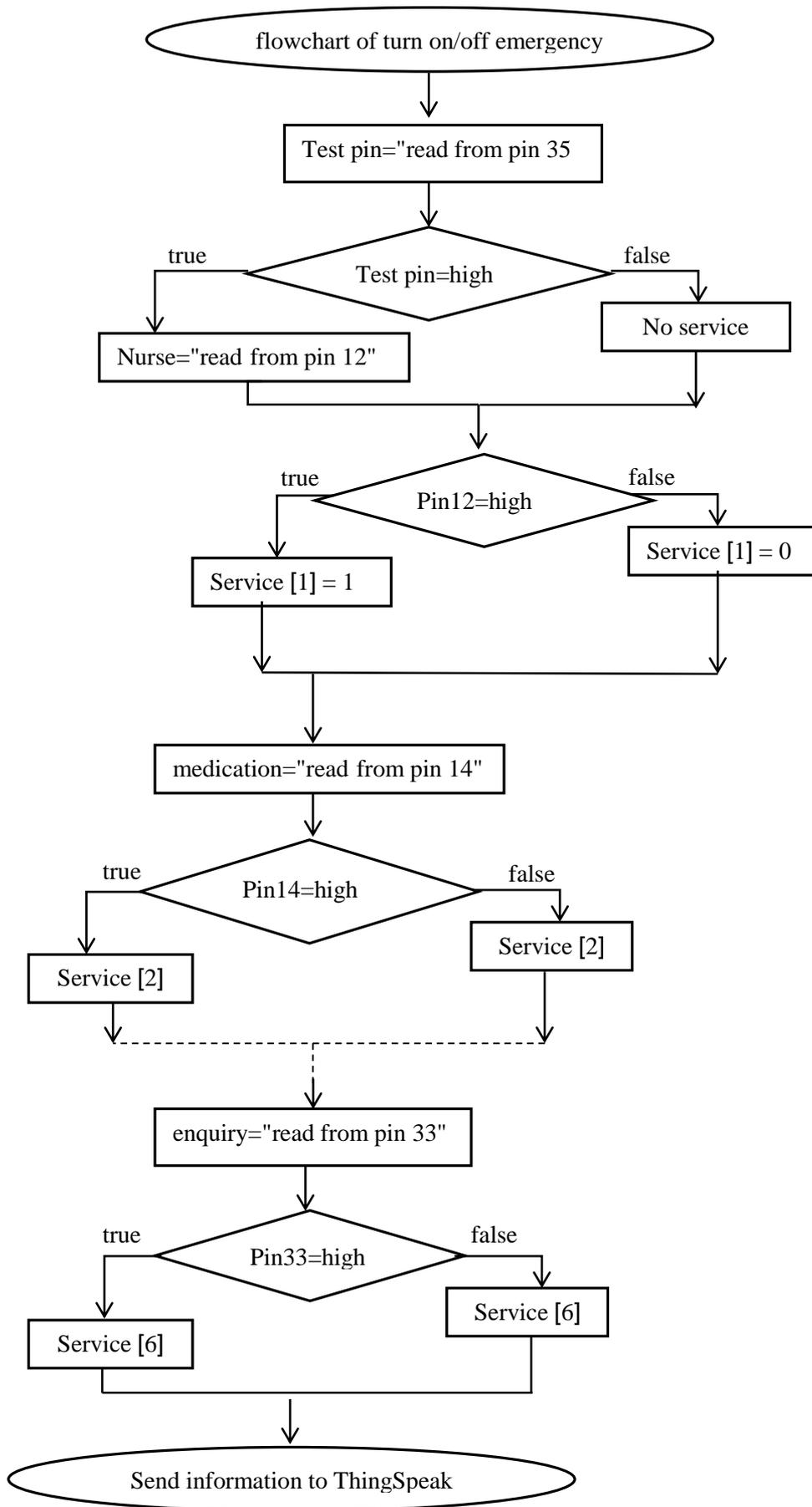


Fig. 3.7 Flowchart of check services

3.3.2.3 Flowchart of sending information from ESP32 to ThingSpeak cloud

the algorithm steps as follows:

- Step 1: enter the write API key of channel from ThingSpeak, to enable ESP32 access to ThingSpeak.
- Step 2: each service takes special field from 8 fields of channel
- Step 3: go to check emergency pin in the first figure with no need to reconnect with Wi-Fi network.

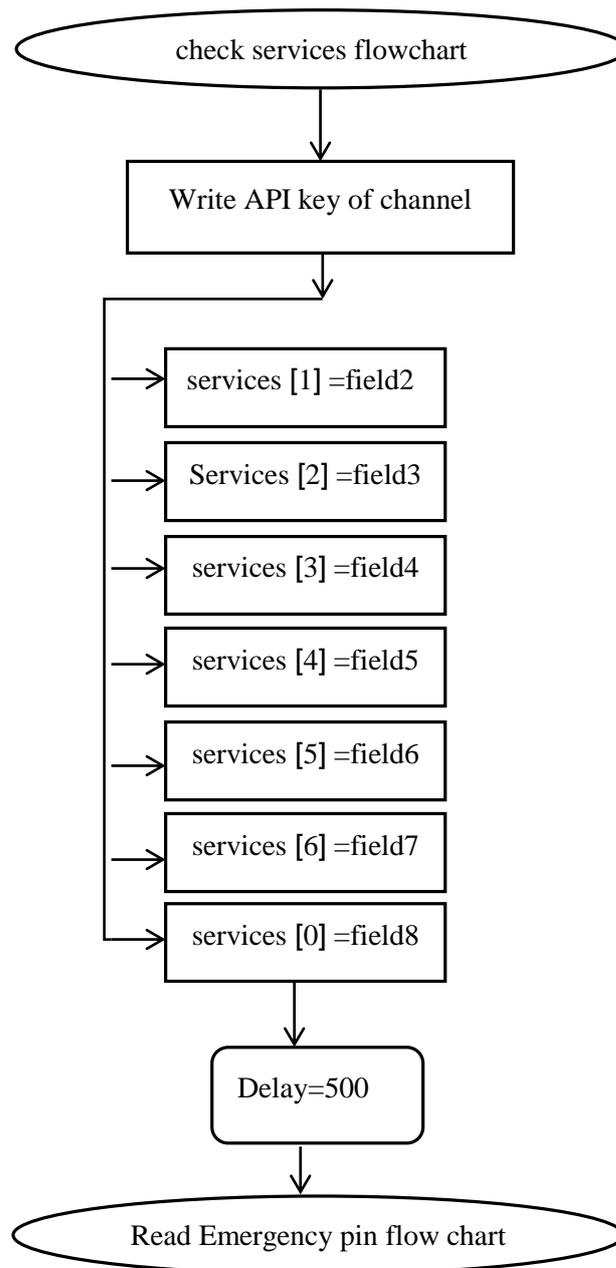


Fig. 3.8 send ESP32 information with ThingSpeak flowchart

3.3.3 Nurse Application

This application helps the medical staff and to shorten the time.

The application made with use program MIT APP inventor to make an Android application.

3.3.4 Feature of application

The view of this application has the table that have the number of bed and the unit where the bed in, also contains the services requested by the patient.

The application connects with ThingSpeak, to read data from the field channel and the connection made by using the read field http.

The number of beds represents by buttons each the bed has a button, the color of button changes to green when any service requests also come voice notification that talks the number of bed and unit which need help, and it changes to red color when the emergency the request also with voice notification.

Also, the services are hidden at the view of application and it is visible when it requests.

3.4 GENERAL BLOCK DIAGRAM OF SYSTEM

This section explains the block diagram of the over proposed system, this block diagram will clarify the connection between the part of the system

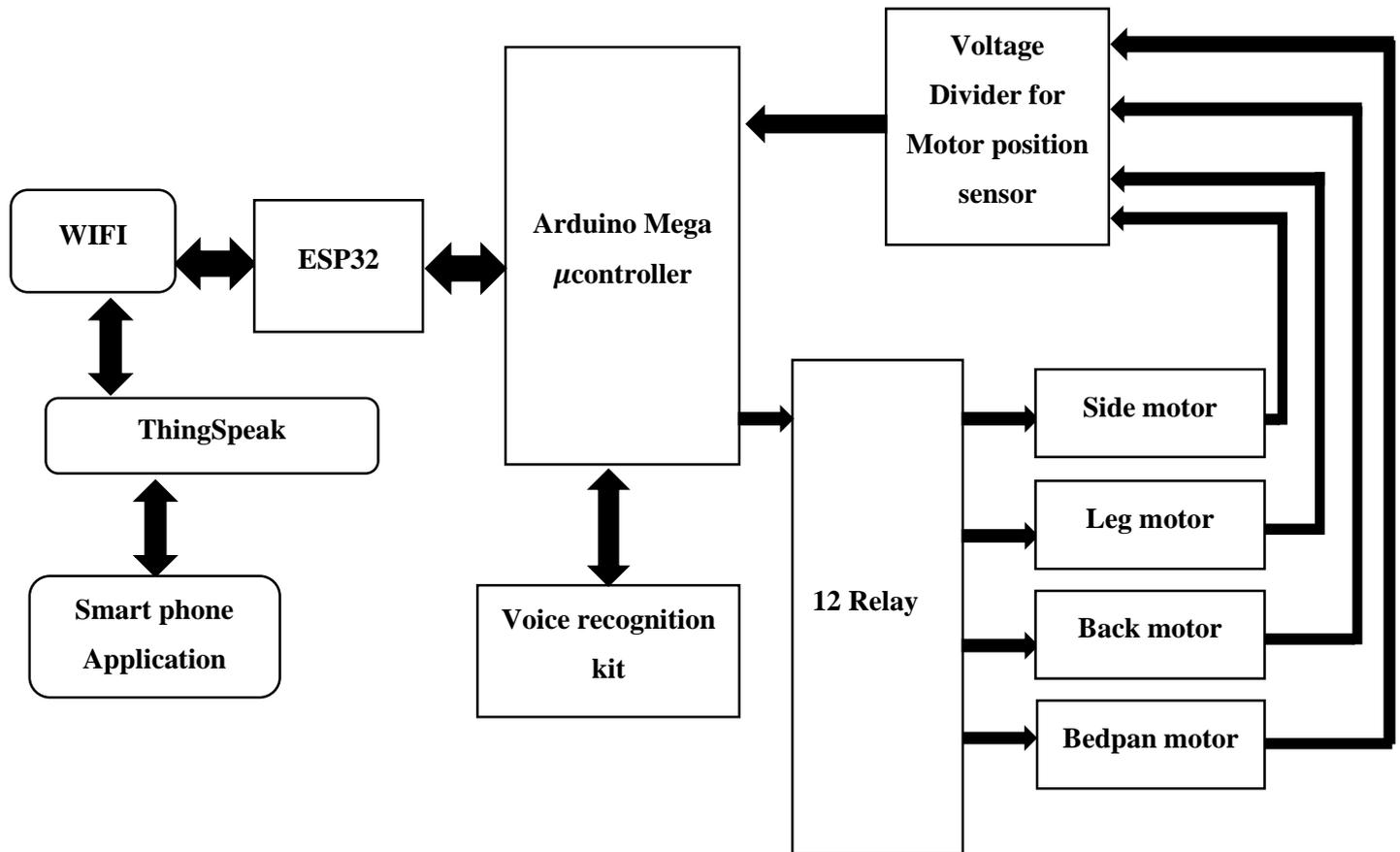


Fig. 3.9 General block diagram

Chapter Four: Result and Discussion

4.1 INTRODUCTION

After plugging and programming all components with, we conducted the experiment. We have operated all components according to the proposed system.

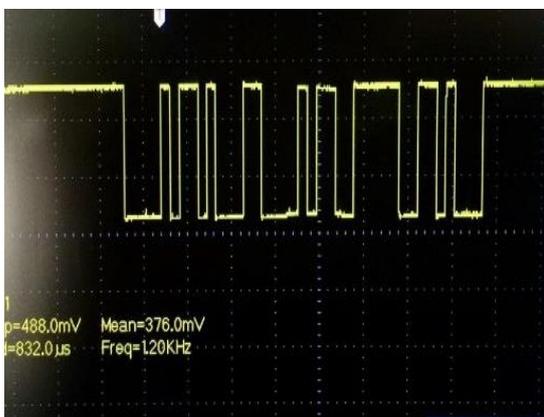
This chapter discusses this project work, which was done using the microcontroller that programmed by the PC

4.2 TEST BED

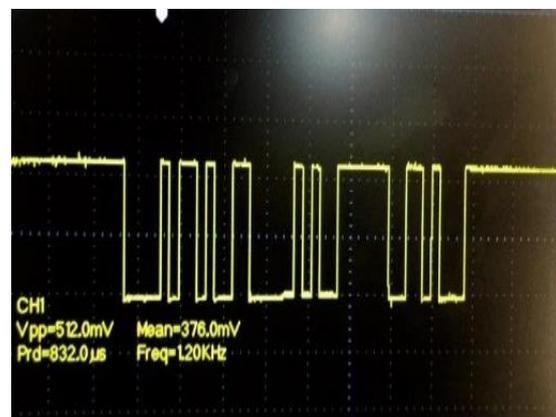
After reading and understanding each of the five-bed motions, we designed signals that is similar to its own signals by using the microcontrollers to control the motors firstly by using command enter to the serial monitor by keyboard and sending it the Arduino serially. but unfortunately this test failed with voice commend because it need to synchronization between external micro controller and the internally one.

So, we control the motor bed by control its own relay as we discussed in the previous chapter.

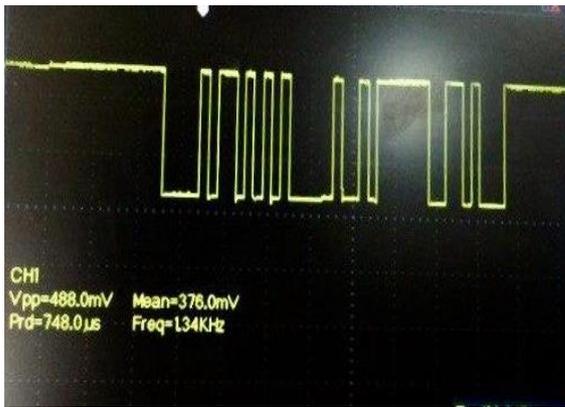
The wave signals from wire control of bed that we read by using oscilloscope:



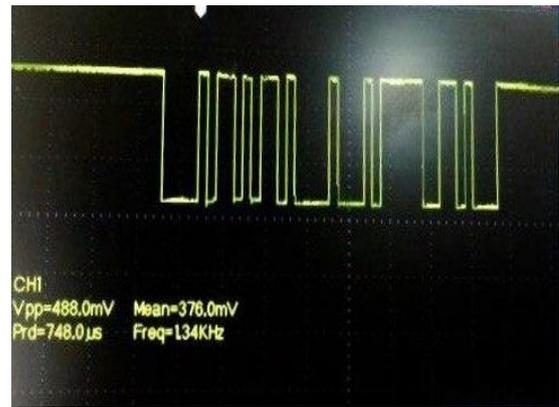
(a)



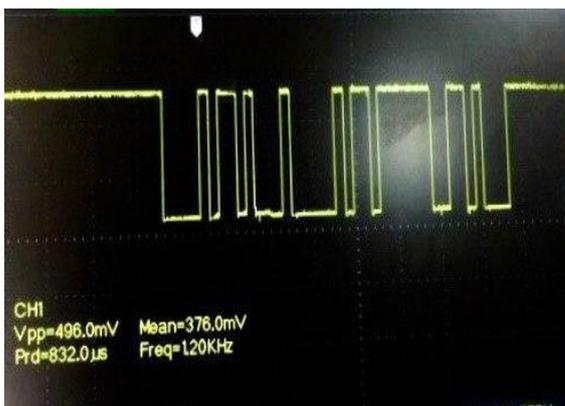
(b)



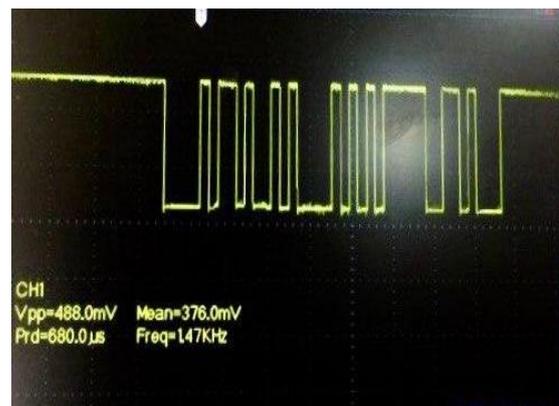
(c)



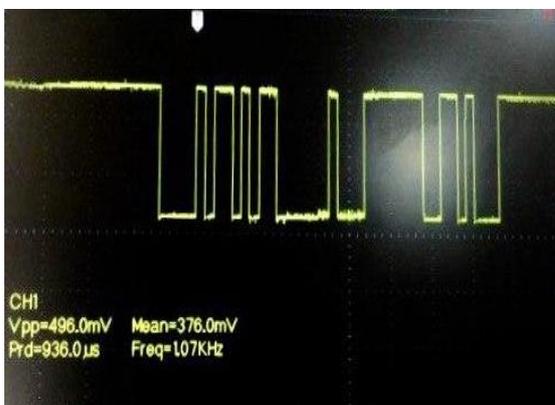
(d)



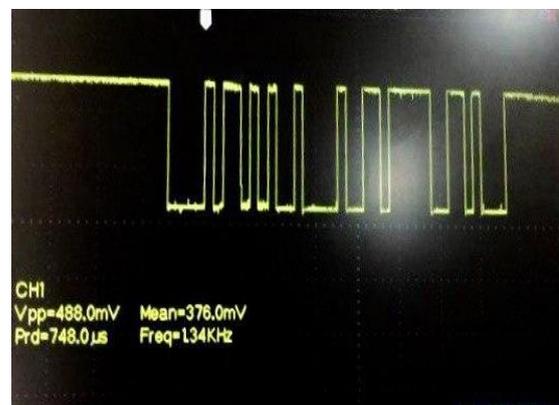
(e)



(f)



(g)



(h)

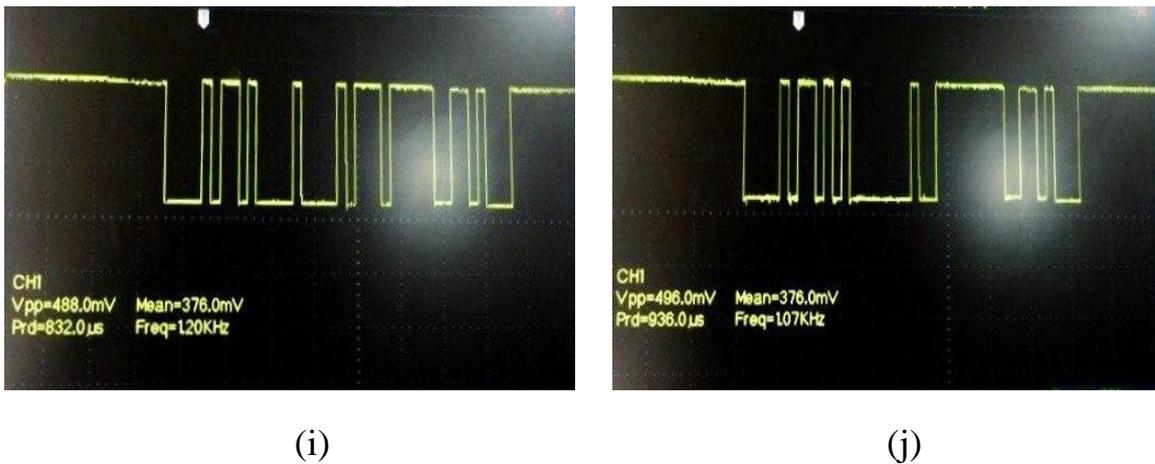


Fig. 4.1 The wave signals from remote control of bed. (a) Right side motion, (b) Left side motion, (c) Rise leg motion, (d) Down leg motion, (e) Rise back motion, (f) Down back motion, (g) Open bath motion, (h) Close bath motion, (i) Wheelchair motion, (j) Rest mode

4.3 TEST AND PROBLEM OF PART OF THE VOICE SYSTEM

4.3.1 Voice recognition module V3 kit

Through the experiences that we did at the beginning of the work with voice recognition, we find that the voice recognition module V3 must train in medium Without noise, and it does not work properly when it is in a medium containing noise and also when using a microphone has bad efficiency.

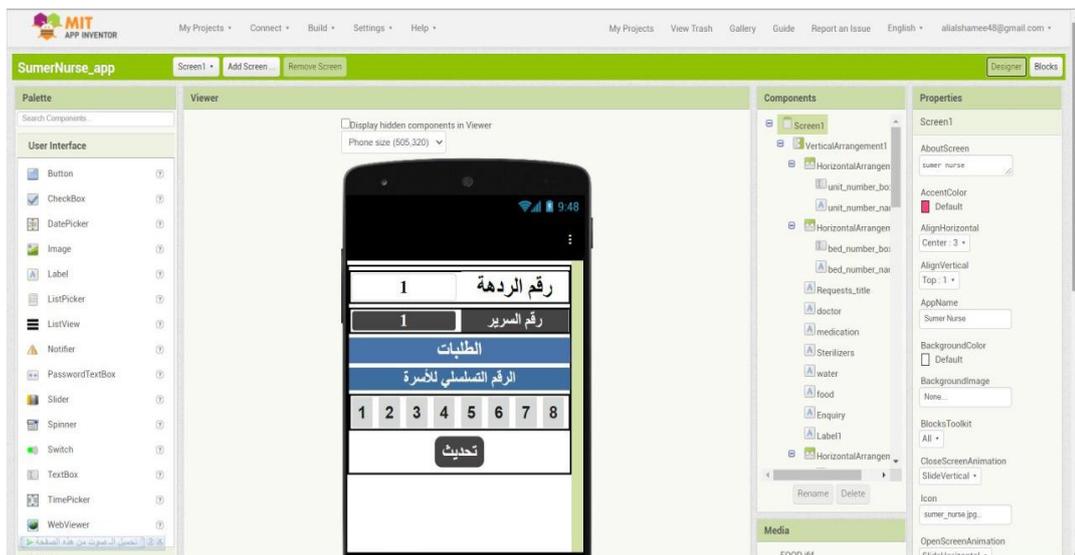
4.3.2 Arduino Mega and ESP32

We failed connect ESP32 with Arduino Mega serial connection because of different in voltage (esp32 work with 3.3 Volts & Arduino Mega work with 5 Volts). So, we use pin to pin connection and each pin has own meaning in our program.

Also, when connecting ESP32 to the internet with the use of Wi-Fi we use at Arduino code five-sec delay, and when connect it with 3G use 10 to 15 sec of delay in sending the information to the cloud. At 3G the number of delays large than Wi-Fi this means the 3G has less speed than Wi-Fi.

4.3.3 Application

The figures below shown the view of designer and blocks of MIT app.



(a)



(b)

Fig. 4.2 view of MIT app, (a) view of designer, (b) view of blocks

The application that we designed it should always be open to being able to see the required services as notifications only arrive when the application is open. The figures below shown the view of application.

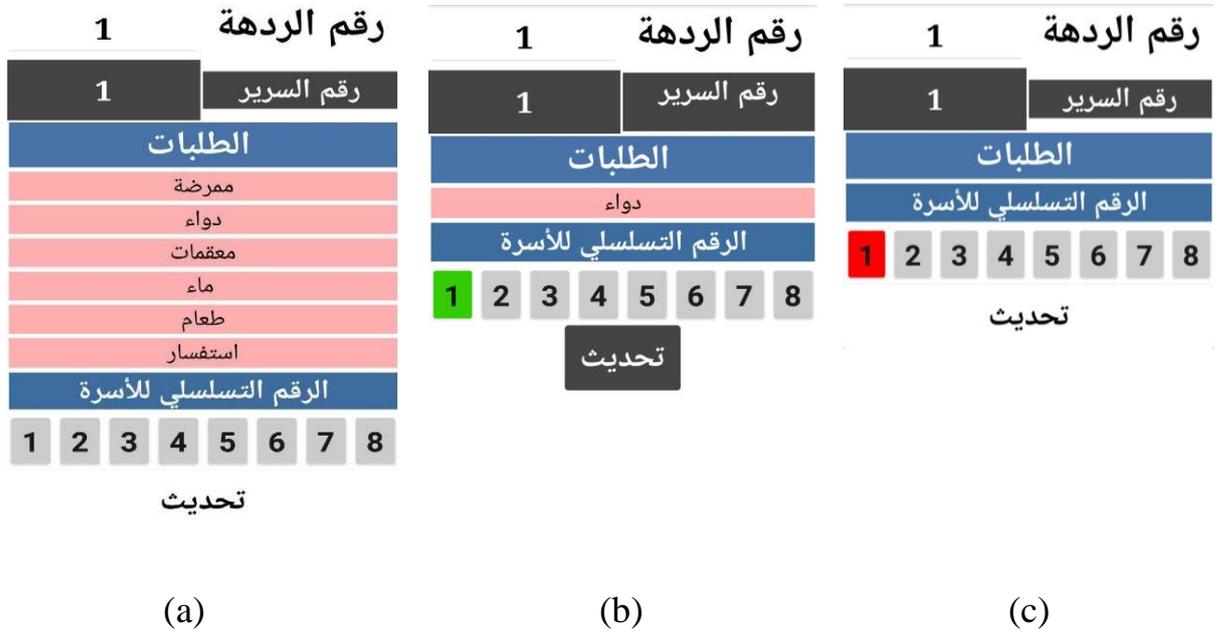


Fig. 4.3 (a) view of application when all services is appearing, (b) the bed bottom will be green color when receive any services, (c) the bed bottom will be red color when emergency pin is active.

Chapter Five: Conclusion and Future work

5.1 CONCLUSION

Creating a system for controlling a medical bed using voice commands with an application that also works with voice commands and that is connected to the Internet is a major step in developing this type of smart system.

The following points represent the project conclusion:

1. The project depends on creating an interactive smart system that uses the Internet of Things (IoT) and microcontrollers to accomplish many tasks based on the patient's voice commands the patient can, through voice commands only, control several movements in the bed.
2. The implement devices in this project are smart and depend on the feedback controlled by the sensor; also, the designed circuit for voice recognition can be trained for any language.
3. the patient can submit his requests to the nurses through the voice commands, where these orders are translated after distinguishing them and sent via the Internet cloud. A special platform has been programmed on the Internet to process the patient requests and send them to nurses through a talking application that has been designed for this purpose.
4. The designed application suitable for android mobile and LED/LCD screens. The patient requests are displayed on the Android application with a talking notification. Moreover, an emergency switch is attached for emergency cases to alert the doctors directly about the presence of an emergency and specify the room and bed number.
5. The components of this project are relatively cheap, which makes it easy to use without any problems.

5.2 FUTURE WORK

The following ideas represent the project future work:

1. Make the train of voice recognition module V3 auto .to enable the patient to train it to the commend that you need and with your language, and Arduino replaced with Raspberry Pi.

2. Development of the application so that the doctor can write medications and prescriptions, follow up on the patient's medication and follow his condition through the added measuring devices.
3. Add bed sensors such as heat, heartbeat, and layout.

References

- [1] L. Comms, “Smart Systems and Internet of Things Standards and Alliances,” p. 9368, 2015.
- [2] H. Kajol, S. Junaid, S. Jain, A. Chaudhary, and K. T. Prajwal, “Design and development of a voice actuated hospital bed for patient care,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 1751–1757, 2019, doi: 10.35940/ijrte.C5882.118419.
- [3] N. Arunpradeep, G. Niranjana, and G. Suseela, “Smart healthcare monitoring system using iot,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6, pp. 2788–2796, 2020.
- [4] J. J. Rani, “A SMART PATIENT HEALTH MONITORING SYSTEM,” vol. 119, no. 16, pp. 59–70, 2018.
- [5] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, “Internet of medical things (IOMT): Applications, benefits and future challenges in healthcare domain,” *J. Commun.*, vol. 12, no. 4, pp. 240–247, 2017, doi: 10.12720/jcm.12.4.240-247.
- [6] B. Friedman and D.G. Hendry, “Value Sensitive Design: A Twenty-Year Synthesis and Retrospective,” *Found. Trends Human-Computer Interact.*, vol. 8, no. 1–2, pp. 1–126, 2015, doi: 10.1561/XXXXXXXXXX.
- [7] M. M. Elsokah, H. M. Farkash, and A. R. Zerek, “Smart Beds For Hospitals with Internet of Things Solutions,” vol. 7, pp. 20–25, 2019.
- [8] https://en.wikipedia.org/wiki/Smart_system .
- [9] “Different Types of Hospital Beds For The Convenience of The Patients.”
- [10] U. R. Family and S. Yours, “If you have any questions about your Hospital Bed , do not hesitate to contact us at : 800-892-4044,” 1987.
- [11] <https://www.verifymarkets.com/blogs/news-and-interviews/smart->

[hospital-beds-enabling-effective-patient-monitoring-and-effective-delivery-of-care](#) .

- [12] Robotshop, “Arduino Mega 2560 Datasheet,” *Power*, pp. 1–7, 2015.
- [13] Espressif Systems, “ESP 32 Wroom Series Datasheet,” vol. 2.9, p. 20, 2019.
- [14] V. Version, “Nodemcu - 32s Datasheet,” 2019.
- [15] <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/> .
- [16] <https://en.wikipedia.org/wiki/Sensor> .
- [17] Wilson Shen, “VR3_manual,” 2014.
- [18] <http://acoptex.com/project/264/basics-project-053d-sim808-gsm-gprs-gps-bluetooth-evolution-board-evb-v32-at-acoptexcom/>
- [19] https://en.wikipedia.org/wiki/Relay#cite_ref-5 .
- [20] FEC, “Relay modules 1-channel features,” *Futur. Electron. Corp.*, no. 5 V, pp. 1–2, 2019.
- [21] <https://en.wikipedia.org/wiki/Actuator>.
- [22] <https://en.wikipedia.org/wiki/Software>.
- [23] <https://www.arduino.cc/en/main/software>.
- [24] <https://thingspeak.com/>.
- [25] https://thingspeak.com/pages/license_faq .
- [26] <https://appinventor.mit.edu/explore/content/what-app-inventor> .
- [27] <https://appinventor.mit.edu/explore/ai2/windows.html>
- [28] <http://easycoding.tn/index.php/2016/12/01/with-tuniot-internet-of-things-world-is-easy/>.
- [29] P. Sundaravadivel, E. Koungianos, S. P. Mohanty, and M. K. Ganapathiraju, “Everything You Wanted to Know about Smart Health Care,” *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 18–28, 2018, doi: 10.1109/MCE.2017.2755378.
- [30] “link of download the library of VR3,” p. 3.