Healthcare Robot for Inpatients

Enas Samir
Merna Mohamed
Hadeel Sameh
Yara Metwally

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
BACHELOR OF SCIENCE
in
Systems and Biomedical Engineering

Under the Supervision of
Prof. Dr. Muhammad Islam

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2020-2021

Table of contents:

Table of Figures:

Acknowledgment:

We would like to express our gratitude to our supervisor Dr.Muhammag Islam for the continuous support of our graduation project work and research. We are also grateful to Cairo University and ITIDA for providing us the needed fund to build our project. We thank Valeo, especially Eng. Ramy Soliman for the technical support and motivation. Finally, we thank our families for supporting us during our lives.

Abstract:

This project aims to reduce the direct interaction between patients and the medical staff because the medical staff is more susceptible to infection because they deal directly with patients with infectious diseases, so we started working on a robot that measures vital signs (temperature, heart rate, blood glucose, and SPO2) for each patient. Based on specific dates found in the schedule on the database, he also performs a video call between the patient and the doctor as needed, and there is a mobile app for the doctor through which he follows up the patient and sees his vital sign that is uploaded to the database after it is measured.

The movement of the robot is based on two methods, the first is the auto, which moves the robot in the corridor to the room, and the other is the manual through a mobile app to control the room between the beds.

Introduction:

Medical robots support minimally invasive procedures, customized and frequent monitoring for patients with chronic diseases, In addition, as robots alleviate workloads, nurses and other caregivers can offer patients more empathy and human interaction, which can promote long-term well-being.

Robots can also autonomously navigate to patients in examination rooms or a hospital, allowing doctors to interact from afar. Robots controlled by a remote specialist or another operator can also accompany doctors during hospital tours, allowing the specialist to contribute to on-screen counseling regarding patient diagnosis and care.

Market Research:

SWOT Analysis:

Strengths	Weaknesses
• Automated system. • Cost-effective. • Remote follow-up with an updated database.	• Internet dependency. • Commercial experience.
Opportunities	• Threats
• Need for automated systems. • Remote follow-up is highly needed now.	• Market acceptance. • New technologies.

Table 1

Chapter 3:Literature Review:

Method	author	year	How is it work	Notes
CGM(Continous Glucose Monitoring)	Abbott Laboratories' Freestyle Navigator	2008	A continuous glucose monitoring system, or CGM for short, is a compact medical system that continuously monitors your glucose levels in more or less real-time (there's normally a five-minute interval between readings). To use a CGM, you insert a small sensor onto your abdomen (or arm), with a tiny plastic tube known as a cannula penetrating the top layer of skin. An adhesive patch holds the sensor in place, allowing it to take glucose readings in interstitial fluid (the fluid that surrounds cells in the body) throughout the day and night. Generally, the sensors have to be replaced every 10 to 14 days.[1]	This technique is useless for us as it is evasive and needs direct contact with the patient also, it is better in case the patient is in his home not in the hospital.
Glucometer	Anton H. Clemens	1970	Glucose oxidase is an enzyme that oxidizes glucose and it is the main content in the glucometer test strips. When it reacts with glucose in the blood droplet (obtained by finger pricking method) an electrical signal is produced at the electrode interface. The strength of the electrical signal is directly proportional to the amount of glucose in the blood sample. That means, higher	This method needs direct contact with a patient.

			the strength of the electrical signals, the higher is the glucose concentration and vice versa. The glucometer is calibrated in such a way to display the strength of these electrical signals. Older meters used glucose dehydrogenase enzyme which was more sensitive than the glucose oxidase strips.[2]	
IR Glucometers	--	--	The patient puts his finger, source of IR goes through his tissues, wait some time while processing then device show the glucose level. Most of the time these devices are connected to mobile application	Sensors in the Egyptian market doesn't provide reasonable readings to work with
Autonomous Three-Wheeled Robot with Computer Vision System	--	--	The research is based on a three-wheel robot with computer vision. The strength of this research is that the product has many features such as GPS, accelerometer, and compass which can be used for data collection purposes. Higher accuracy results are achieved with higher resolution and frequency. The weakness of this research is having a bad robot control method that can change to Wi-Fi control that can support further control range. Other than that, the robot cannot operate in dark surroundings. The robot may have problems with the movement of the robot which cannot avoid the obstacles with the accurate turning angle of the robot.	--

Table 2

Materials and methods:

A. **Robot design and parts**:

A.1. STM32: We used this microcontroller to deal with temperature and HR & SPO2 sensors and do some processes on sensors reading before sending them to the raspberry pi. We use ST-Link to upload the code on stm32f3c8t6, one of the boot select must be one and the other must be zero. We also use the same ST-Link to clear the flash memory using STM32 ST-Link utility API, both boot select must be one.

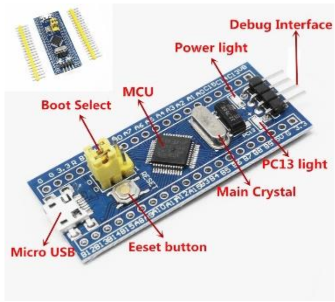


Figure 2:STM32

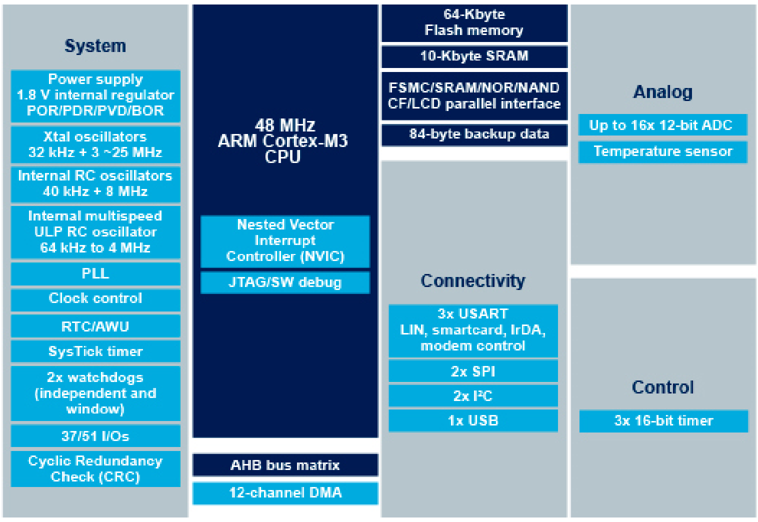
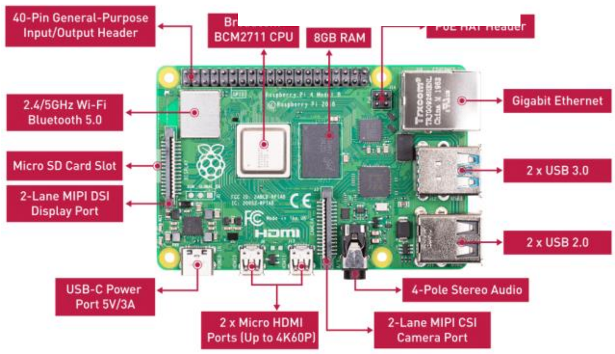


Figure 3:ST-Link

Figure 1: STM32 Specefications

A.2. MLX90416: we used it to measure temperature and estimate blood glucose. If we measure ambient and body temperature on the tip and wrest, we can estimate blood glucose. It used IR technology.

Figure 4:MLX90614



A.3. MAX30100: we used it to measure heart rate and SPO2 percentage.

A.4. Raspberry pi 4: We used it as our main controller to receive sensors' data, control motors, USB web camera, speakers, and pi camera.

Figure 5:Raspberry pi

A.5. Wheels: This wheel has two degrees of freedom and can traverse Front or Reverse. The center of the wheel is fixed to the robot chassis. The angle between the robot chassis and wheel plane is constant.

Figure 6:Wheel

A.6. Motor part: a 3D printed part to hold the DC motor.

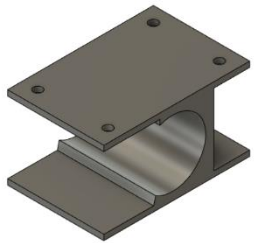


Figure 7:Motor Part

Figure 8:Body Design

A.7. Body design: 3D printed part hold sensors, have a way to integrate with the acrylic base to be able to afford the body design and wooden head to hold

A.8. Motors: DC Motor 5KG 12V 150RPM 3mm Shaft. 4 of them make robot could afford 20KG weight

Figure 9:DC Motor

A.9. Pi camera: It is the camera shipped along with Raspberry Pi. A pi camera module is also available to which can be used to take high-definition videos as well as still photographs.[3]

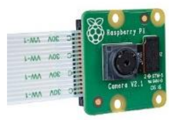


Figure 10:PI Camera

A.10. 7" HDMI TOUCH DISPLAY 1024 X 600 For Raspberry PI: 7" Touchscreen Display, Screen Dimensions: 194mm x 110mm x 20mm (including standoffs), Viewable screen size: 155mm x 86mm, Screen Resolution 800 x 480 pixels, 10 fingers capacitive touch, Connects to the Raspberry Pi board using a ribbon cable connected to the DSI port, the Adapter board is used to power the display and convert the parallel signals from the display to the serial (DSI) port on the Raspberry Pi.[4]

Figure 11: LCD

A.11. L298N Dual H-Bridge Motor Driver: The direction rotation of the DC motor is controlled by H-bridge. The DC motor can move clockwise or anti-clockwise when there is a voltage applied across a load in either direction. Input pins to control the DC motor, Enable pins to control the speed output pins connected to the motors

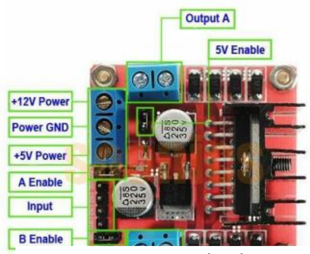


Figure 12:H-bridge

A.12. Raspbian OS: all the operating systems Arch, Risc OS, Plan 9, or Raspbian available for Raspberry Pi, Raspbian comes out on top as being the most user-friendly, best-looking, has the best range of default software, and optimized for the Raspberry Pi hardware. Raspbian is a free operating system based on Debian (LINUX), which is available for free from the Raspberry Pi website.[3]

A.13.Crontab: We can schedule jobs with the crontab command. A crontab file is a simple text file containing a list of commands meant to be run at specified times. It is edited using the crontab command. The commands in the crontab file are checked by the cron daemon, which executes them in the system background [5]. We used it to schedule the appropriate script depending on the visit type, where 0 stands for measure visit, 1 video visit, and 2 measure and video visit.

B. VitalSigns Measurements

B.1 Temperature: we connect MLX90416 with STM32 using the I2C protocol. I2C stands for Inter-Integrated Circuit. It uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high. Serial Data (SDA): Transfer of data takes place through this pin, Serial Clock (SCL) – It carries the clock signal. I2C operates in 2 modes, Master mode & Slave mode. Each data bit transferred on the SDA line is synchronized by a high to the low pulse of each clock on the SCL line.[6]

B.2 Heart Rate and SPO2: we measure them using max30100 which is connected to STM32 using I2C.

B.3 Sending data from sensors to the raspberry pi using UART communication protocol. By definition, UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end.[7]

Wires	2
Speed	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000
Methods of Transmission	Asynchronous
Maximum Number of Masters	1
Maximum Number of Slaves	1

Figure 13:UART Summery

B.4 Blood Glucose:

B.4.1 Collecting data: we collect it manually from 25 people. For every person, we ask him/her for his/her length, take his blood glucose level & finally measure their temperature on the tip and wrest. Finally, we record their records in an excel sheet. Where 0 means the temperature on tip & 1 means temperature on wrest.

B.4.2 Equations:

B.4.2.1 Heat Radiation:

$$h_r = 4 * \sigma * \varepsilon * T_o^3 * (T_s - T_0) \dots\dots\dots\dots (1)$$

$$G = g * \propto * (T_s - T_0) * \frac{L^3}{V^2} \dots\dots\dots\dots\dots\dots (2)$$

$$R = G * P \dots\dots\dots\dots (3)$$

$$N = 0.021 * R^4 \ldots\ldots\ldots\ldots\ldots..(4)$$

$$Heat\ convection\ , h_c = N * k * \frac{(T_s - T_0)}{L} \ldots\ldots\ldots\ldots.(5)$$

$$Blood\ flow\ rate\ , bf = \frac{4.36 * k * \pi * L * (T_{x2} - T_0)}{\rho * c_p * (T_s - T_0)} \ldots\ldots.(6)$$

$$Blood\ Glucose\ Level = a + b * h_r + c * h_c + d * bf \ldots\ldots\ldots(7)$$

where,

σ is Stefan Boltzmann constant, 5.67×10^{-8}

ε is the emissivity of measured surface

T_0 is surrounding temperature in °C

T_s is measured surface's temperature in °C

G is Grashof number

g is gravity

α is thermal expansion coefficient

L is subject's height in meter

v is kinematic viscosity

R is Rayleigh number

P is Prandatl number, usually 0.7

N is Nusselt number

k is the thermal conductivity of measured surface

h_r is heat radiation rate

h_c is heat convection rate

bf is blood flow rate

Figure 14: what coffetiants stands for

B.4.2.2 Preprocessing and Modelling:

we applied the past equations on the data collected to get this modified data source:

Finally, we applied a Multiple linear regression model to get the coefficients a,b,c,d we used the model to estimate blood glucose level.[8][9]

To	Ts	bf	Tall	BloodG
27	31	1.85E-08	155	106
27	31	1.85E-08	159	106
36	28	5.13E-08	178	100
17	28	2.14E-08	158	204
47	29	-5.54E-08	179	156
31	29	3.64E-08	160	108
37	30	-1.96E-08	165	115
36	30	-9.72E-09	174	133
23	32	1.38E-08	183	93
31	30	-2.37E-09	179	95

Figure 15: preprocessing result

C. DataBases:

We implemented the Databases on the Xampps server which is a free, open-source, web server.it works Via MariaDB and MySQL.

Our DataBases contains 5 main interties:

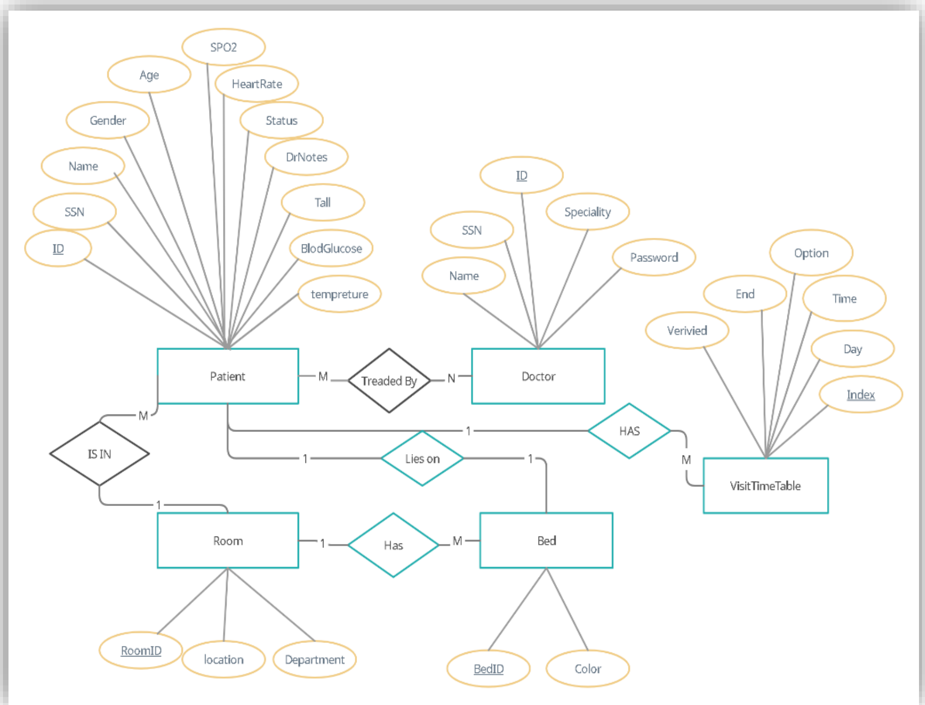


Figure 16:ERD of our databases

- Patient: contains patient data, doctor notes, and Vital signs measured.it has relation “treated by” with Doctor table,” lies on” with Bed table, “is in ” relation with Room table and “Has” relation with VisitTimeTable.
- Doctors: contains doctors basic information needed to register the doctor as a user.
- Bed: contains information for every bed in the Hospital.
- Room: contains information for every room in the Hospital. It has a “Has” relation with the Bed table
- VisitTimeTable: contains the scheduled visits for the robot.

Attributes for every entity are Explained in the Entity Relation Diagram (ERD).

Communication with the database is based on APIs which we implemented by PHP. We have implemented 15 APIs to do different jobs in our system and their folder is placed in the server folder.[10]

D. Doctor Mobile Application:

We implemented a doctor mobile application using Android Java with IDE: Android Studio.it has 4 main activities (Sign up, log in, MainPage, and VideoCall). we move from one to another by starting new intent.

The doctor can signup if having no account or can move to login activity if already has an account. Then, App will take him to the main page where his name is saved during all times of his login period using a global class until he presses the signout button.

D.1. DataBases Communication Features:

We have 6 options for the doctor to do on the Main page: The First is to retrieve data of all patients that the doctor is responsible for. The second is to retrieve data of all patients with danger status and the logged doctor is responsible for. The third is to retrieve patient data by patient id. The fourth is to Edit the doctor notes column for a specific patient by id. The fifth is to add a new visit time in the robot schedule in case the doctor needs an emergency call with the patient. The last feature is to show all upcoming visits for a specific patient [11].

Main-Page have also Join button which starts new intent to video-call activity if the robot is ready which means that it arrived at the room and verified patient identity using the verification algorithm.

D.2 Video conference feature:

Background:

In this part, we used the android studio as an IDE and for android application development. the project structure is :

- **manifests**: Contains the AndroidManifest.xml file.
- **java**: Contains the Java source code files, including JUnit test code.
- **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

In the beginning, we had two different scenarios to implement the video conference feature. we will discuss the two scenarios and explain which scenario we decided to choose.

- The first scenario is using zoom SDK as a video conferencing tool:
  At an early stage of implementing this feature, we built it with the use of the zoom tool, but after discussing it with our supervisor, we recognized that it will not be the best choice. Now we will discuss the pros and cons of using zoom SDK in our mobile App.

Pros	Cons
Very reliable.	Limited time.
In-meeting chat is very good.	Security issues.
Remote control option.	Zoom lags a lot.

  After discussing the pros and cons of zoom SDK, and according to requirements needed for our mobile App, like the need for an open-source, free tool we decided to customize our solution and search for another tool. so we found Jitsi SDK, which we will discuss in the next section.
- The second scenario is using Jitsi SDK as a video conferencing tool:
  Why we preferred this scenario?
  1. Open-source.
  2. Entirely free conferencing.
  3. Unlimited time.[12]

E. Robot Motion:
E.2. Manual part:

The 4 wheels of the chassis are connected to 4 separate motors. The motor driver IC L298N is capable of driving 2 motors simultaneously [ ]. The rotation of the wheels is synchronized based on the sides i.e. the left front and right front wheels rotate in sync and left-back and right back wheels rotate in sync. Thus the pair of motors is given the same digital input from L298N at any moment.

The chassis has one shelf over the wheels, and the wheels are fixed by 3D printed parts. The IC is fixed over the shelf with the help of two 0.5 inch screws. It is permanently connected to the motor wires and necessary jumper wires are drawn from L298N to connect to Raspberry Pi to control the motors [ ]. And we use a power supply providing 10A-12V to run the motors.

L298N Pin	RPi GPIO Pin
Int 1,2,3,4 “first IC ”	8,9,11,25
Int 1,2,3,4 “second IC ”	17,18,22,27

```
def forward():
    GPIO.output(Backward1,GPIO.LOW)
    GPIO.output(Forward1,GPIO.HIGH)
    GPIO.output(Backward2,GPIO.HIGH)
    GPIO.output(Forward2,GPIO.LOW)

    GPIO.output(Backward3,GPIO.LOW)
    GPIO.output(Forward3,GPIO.HIGH)
    GPIO.output(Backward4,GPIO.HIGH)
    GPIO.output(Forward4,GPIO.LOW)
```

Figure 17: Python code to control the DC motor using the L298N motor driver.

E.3 Remote Control Feature:

The implementation of the Remote control feature is implemented by sending a command from an android device to a raspberry pi microcontroller from an Android smartphone.

- Steps of implementation:
    1. We create a helper task for TCP/IP interface. from this method we can create and send commands to raspberry pi over wi-fi[13].
    2. We create a method to get the IP address and port from the textbox.
    3. From six buttons, we can send left, right, forward, backward, start and stop commands same as to commands we defined in the raspberry server program.

F.**Verification Algorithm (OCR):**

We used OCR (optical character recognition) Algorithm to verify patient identity. We assumed that the patient is wearing a bracelet with his “id” on it. Firstly show instructions on the screen to move his hand in front of the camera then, we capture the image using our camera, then we apply a Bilateral filter and Canny edge detector from OpenCV on it to smooth the image and get contours. Secondly, we used the “easyocr” library from python to detect the number, and then we will compare it with the patient id for the visit if it is the same then we cover the verified column into “1” to enable the doctor to join the meating[14].

```python
def Verfy(img):
    bfilter=cv2.bilateralFilter(img,11,17,17)
    edges=cv2.Canny(bfilter,30,200)
    keypoints=cv2.findContours(edges.copy(),cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
    contours=imutils.grab_contours(keypoints)
    contours=sorted(contours,key=cv2.contourArea,reverse=True)[:10]
    location = None
    for cont in contours:
        approx=cv2.approxPolyDP(cont , 10 ,True)
        if(len(approx)==4):
            location=approx
            break
    mask=np.zeros(img.shape,np.uint8)
    new_img=cv2.drawContours(mask,[location],0,255,-1)
    new_img=cv2.bitwise_and(img,img,mask=mask)
    (x,y)=np.where(mask==255)
    (x1,y1)=(np.min(x),np.min(y))
    (x2,y2)=(np.max(x),np.max(y))
    cropped_img=img[x1:x2+1,y1:y2+1]
    reader=easyocr.Reader(['en'])
    result=reader.readtext(cropped_img)
    number=result[0][-2]
    print(number)
    return number
```

Figure 18:ocr algorithm

Results:

A. Robot design and parts:

Figure 16:Final robot design with components

B. Vital signs' measurements:

B.1 I2C settings:

Master Features	
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Slave Features	
Clock No Stretch Mode	Disabled
Primary Address Length sel...	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detect...	Disabled

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
USART2 global interrupt	☑	0	0

Figure 19: I2C Settings

B.2 UART Settings:

Basic Parameters	
Baud Rate	9600 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

Figure 20: UART Settings

B.3 welcome screen:

Figure 21: Welcome Screen

B.4 HR &SPO2 screen instructions:

PLEASE PUT YOUR HAND HERE
To measure your HR&SPO2

Figure 22: HR & SPO2 instruction

B.5 Temperature screen instructions:

PLEASE PUT YOUR HAND HERE
To measure your Temperature

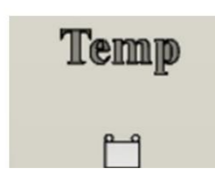


Figure 23:Temperature Screen

B.6 Sensor Measurements:

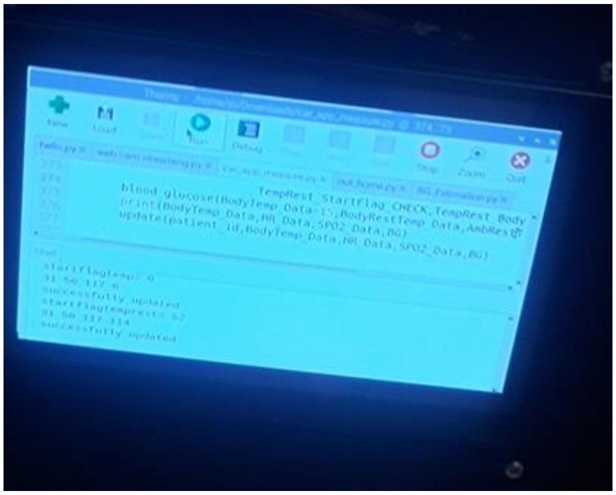


Figure24: sensor measurements

B.7 Video call:

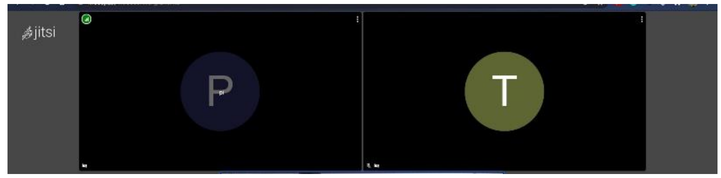


Figure 25: video call

B.8 Cron job:

```
27 18 * 5 * export DISPLAY=:0 &&  cd  /home/pi/Desktop/task3 && python3 hello.py
```

Figure 26: Cron Job

C. DataBases: These screen-shots are the results of our database

Options	id	ssn	name	gender	age	heartrate	temprature	spo	bloodglucose	bedid	roomid	status	doctornotes	tall
Edit Copy Delete	1	6457	ali	male	65	80	37	97	80	5	2	normal	He is under treatment	160
Edit Copy Delete	2	9675	asia	female	45	80	38	97	80	4	5	danger	about to check out	155
Edit Copy Delete	3	7429	raafat	male	55	85	39	90	70	2	2	danger	intensivecare	150
Edit Copy Delete	4	8608	adham	male	88	1000	40	30	190	8	2	danger	very danger	190
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	under	needs to measure	[illegible]

Figure2 7:Patients table

id	username	ssn	password	speciality
6	755	0	$2y$10$LkyydaLooXoZYFaaq6lvEe8w9BOq9zfWSwTGrHW4BAS...	surgery
7	ola	284	$2y$10$Fi9mkYm88.t4V.U/BqTagOpX2G9VPqrkKe9TxU9qe67...	cardio
8	hadeel	6327	$2y$10$bXMVyvJJHSlyKZ95.xrckuHEwHzrTMmR77C/euHgfVZ...	cardio
9	Hadeeel	2147483647	$2y$10$OYuwJl8.luObdes9D6qkROzHgFv.f67eM7B35jToH65...	cardio
10	KaylA	376473	$2y$10$Hf77s87yScWHWY8TbYYBhux6oo/4UqGmewQX.myGrP...	Cardio
11	hala	25730422	$2y$10$HbP6jcj7f4xXxiNVngmMUudMxi3AgxpAeJLdhZEjNuW...	internal
12	tia	1234556	$2y$10$2DJTs1/jS4LW04hdWj7ZtekePy4rt.2NDLmFNxGcLWH...	liver
13	lama	12345786	$2y$10$AB7vHEalkuJ7jzauc2FM4eC8Bt1yXwOozmD9PC8LUWb...	internal
14	asmaa	347789	$2y$10$XeLS1qbAxWlr/fFAxh5RB.97yrxnBBHcFTa3TgrDTh8...	cardio

Figure 28:Doctor table

doctorid	patientid
7	3
7	4
8	1
8	2
8	5

Figure 29:treated by relation table.

roomid	location	department
2	2floor	cardio
5	2floor	cardio

Figure 10:roms table

ind	patientid	day	time	isvideocall	verified	end
8	3	2021-05-27	20:05:00	2	1	1
9	5	2021-05-27	23:07:00	1	1	1
10	2	2021-05-27	10:07:00	1	1	1
11	1	2021-07-13	09:00:00	2	0	0

Figure 30:VisitTimeTable

id	color	roomid
1	yellow	2
2	pink	5
3	yellow	5
4	black	5
5	red	2

Figure 31:Bed table

D. Doctor Mobile Application:

D.1.DataBases Communication Features:

These screenshots are the results of the mobile app Database

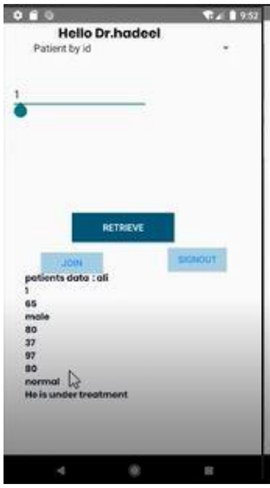


Figure 32:Retrieve patient data by id.

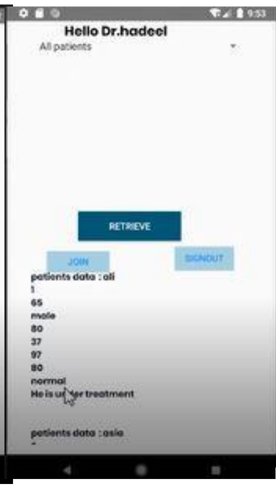


Figure 33: retrieve all patients treated by the logged-in doctor.

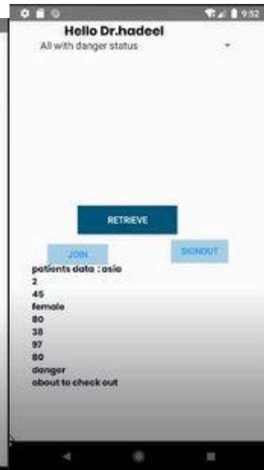


Figure 34:Retrieve all patients with danger status.

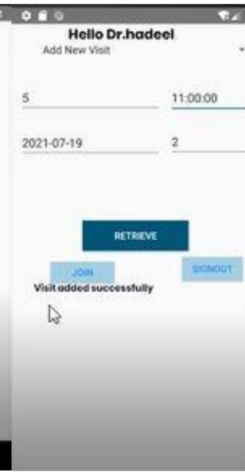


Figure 35:Add new visit

Figure 36:view upcomming visits

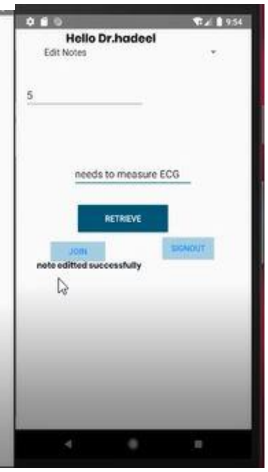


Figure 37:edit doctor notes for patient by id

D.2 Video Conference Feature:

we implemented the video conference feature between the robot and our Dr mobile application and here are the results of the feature.

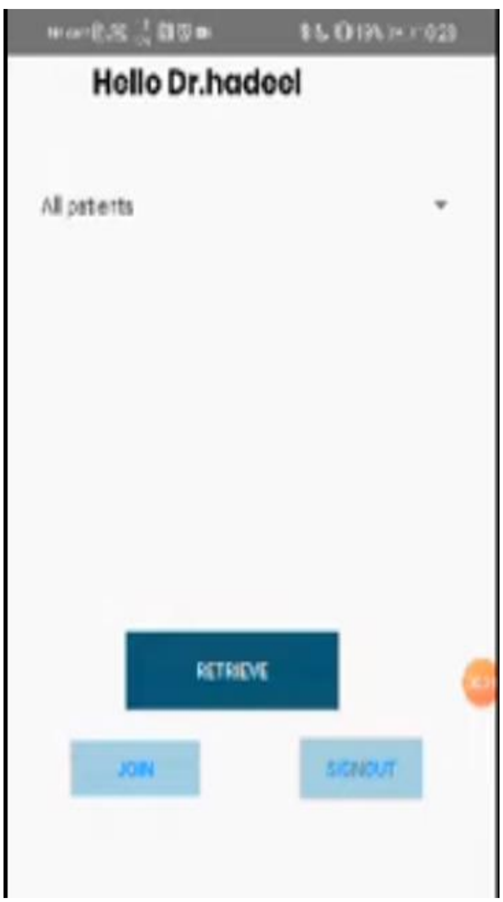


Figure38: Doctor logging in to start meeting

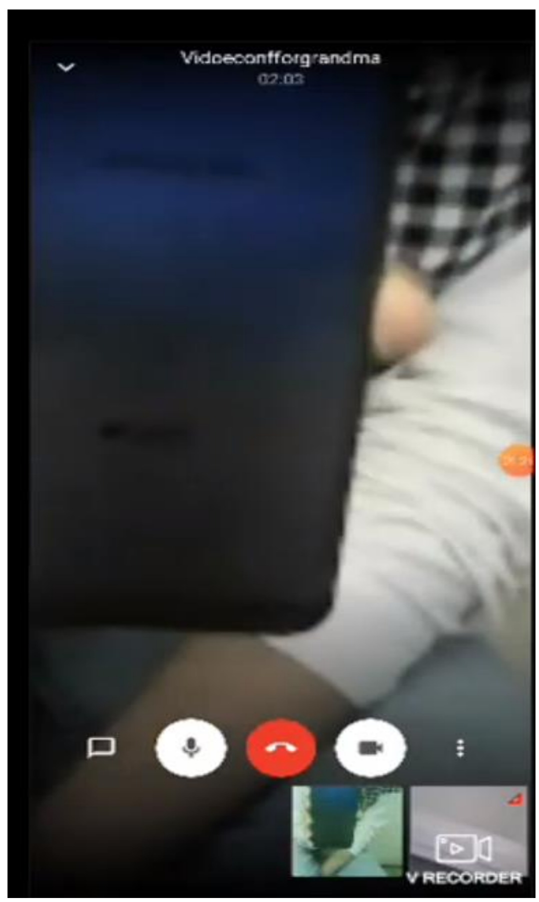
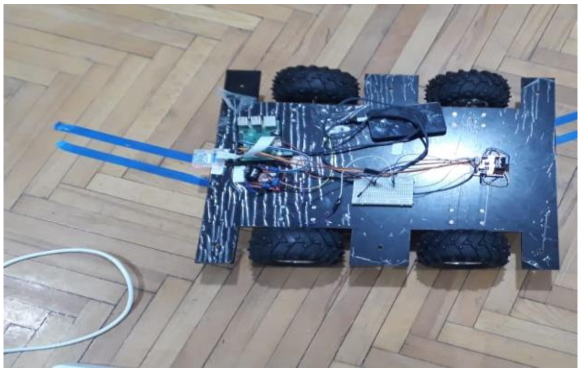


Figure39 : Result of launching video meeting from the android app to the Robot

E. Robot Motion:

E.1 Motors:

The robot determines the route by itself and finds the movement line in the corridor until it reaches the room, and then it is controlled by a mobile application so that it moves between patients based on the date of the visit.



E.2 Autonomous Part:

Autonomous Motion Algorithm:

- Test Case (figure5): results in <u>Forwarding</u> order.
- Test Case (Figure 6): results in <u>Right</u> order
- A test case in (Figure 7): results in <u>Left</u> order.

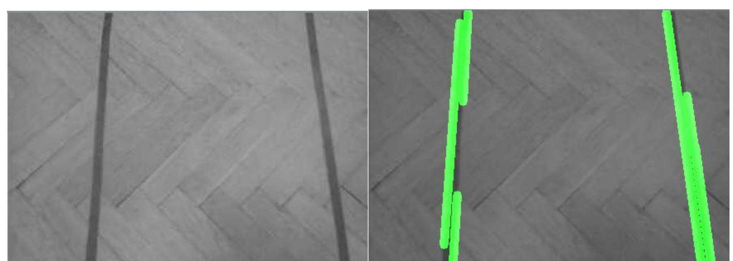


Figure 40:The image on the left is the caputered image and the image on the right shows lines detected

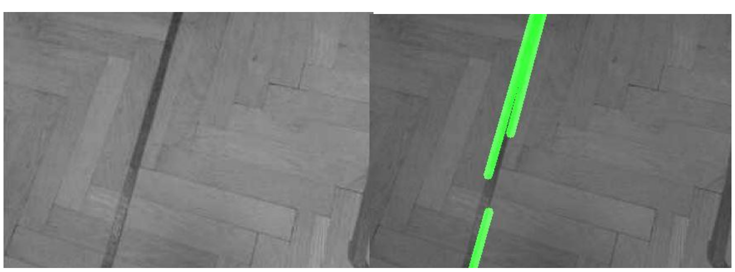


Figure 41:The image on the left is the caputered image and the image on the right shows lines detected

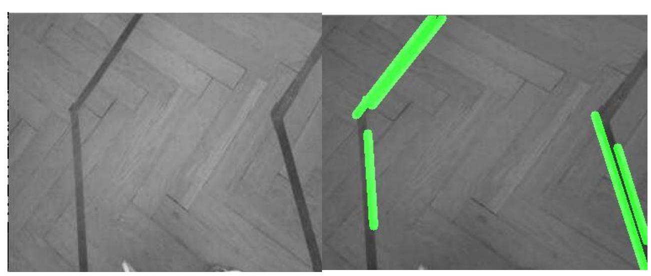


Figure 42:The image on the left is the caputered image and the image on the right shows lines detected

E.3 Remote Control Mobile Application:

we implemented the video conference feature between the robot and our Dr mobile application and here are the results of the feature.

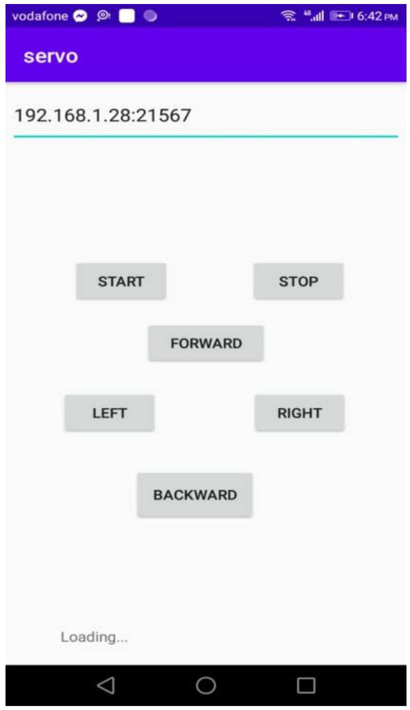


Figure 43: Result of Remote Control App

F. Verification Algorithm:

The following figure shows a test case result using our ocr algorithm:

Figure 44:OCR result

Discussion

A.Robot design and parts:

- We found that the base should be acrylic 6mm to afford the design weight and volume.
- We decided to make the body design from PLA material and to be 3D printed to be able to shape its curves. We consider curves, final shape, its computability to go for a market, weight controllability as we control the body weight by controlling infill in our case was 10% and result in our 3D printed part 1800gm.
- We decided to make our head from wood as its density and weight are less than acrylic 3mm. Although acrylic is better than wood as the prototype and from a robot look point of view, we decided to use MDF 2mm as the overall weight of the head must not exceed 500gm (where LCD was almost 250gm and speakers were about 200gm). MDF 2mm satisfies this requirement.
- One of the drawbacks in design was the position of nails to integrate the head with the main body. It should be centered. Its position affects the robot when it is moving. It wasn't stable while moving.
- Another drawback was about the base of a 3D printed part it was 480*410 it must be wider around 480*480 it will be suitable. Take care the weight may increase by 200gm.

B.Vital signs' measurements:

- We decided to work with STM32 as the main microcontroller. We did that for learning purposes as we want to work with ARM architecture. STM32 was available in the Egyptian market and a good mid-range controller. Also, its specifications were compatible with our requirements. It has debugger mode and it was extremely useful to troubleshoot any problem in our system. One of its main drawbacks is it doesn't have a strong community to support any problem. For a better understanding of HAL drivers, this document is really useful[8].
- MLX90416 was our best choice in the Egyptian market. Its readings' accuracy is better compared to using MAX30100 for measuring temperature. Although both MLX90614 & MAX30100 use the same technology, we prefer to use both of them as we are concerned about accuracy when this sensor will be in a healthcare robot.
- MAX30100 was the best we found to measure SPO2 & HR. Its accuracy is bad and affected by environmental factors. We divide the reading of SPO2 & HR by 100 to get some reasonable readings
- We decided to work with pi 4 as it has a better processor and UART transferring data was more stable on it.
- First of all, we work with pi camera v1 and it was a better choice for us. But we needed to adjust its focus but it wasn't adjustable. So, we decided to upgrade to pi camera v2. Sadly we couldn't find an experienced person to adjust focus for us. If your application needs both near and far focus or near focus only, and ardu camera will be your best choice[9].

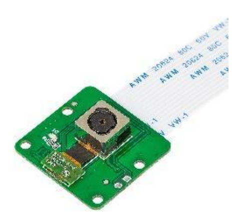
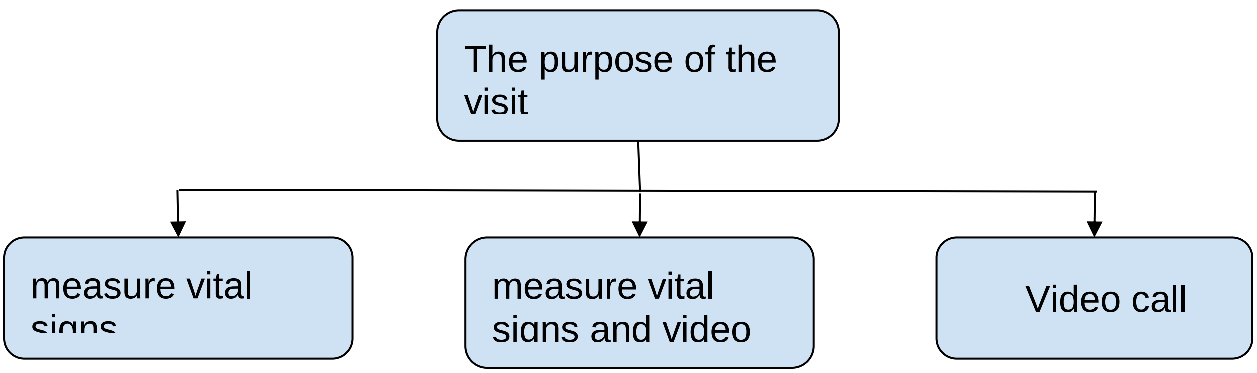


Figure 45: Ardu camera

- Watch the difference between 7 inch HDMI LCD and 5 inch HDMI LCD. The 5 inch HDMI LCD needs to use pi pins and an HDMI port so watch your application requirements.
- We decided to use raspbian os as it has a community to support any problem.
- UART problems: STM32 has 2 UART but the first UART1 could receive only but UART2 works properly. Keep your baud rate the lowest to reduce the noise that could be found while transferring data.
- The screen that gave patients instructions couldn’t last all the time as it is python code using Tkinter and it must end to continue the rest of the code.
- Cron-tab python library was our best choice to apply scheduling concepts.
- Sensor readings are biased depending on the environment.
- Blood glucose estimation depends on temperature readings which depend on the environment, So it depends.
- People using the traditional way of
- The script which retrieves the visit timetable runs every hour to update all meetings.

B.1.The purpose of the visit:

The three scripts work based on the purpose of the visit specified in the schedule



If the purpose of the visit was to measure vital signs, the first script will be run:
First, an instruction window will appear to instruct the patient to place their finger through a class that contains two functions.

```
window_1=popup().alert_popup("Follow instructions","hr-spo2.jpeg")
for i in range(5):
    MAX30100_CheckFlag(MAX30100_StartFlag,SPO2_Flag,HR_Flag,
                       MAX30100_StartFlag_CHECK,SPO2_Flag_CHECK,HR_Flag_CHECK)
    print(BodyTemp_Data,HR_Data,SPO2_Data)
    update(patient_id,BodyTemp_Data,HR_Data,SPO2_Data)
window_3=popup().alert_popup("Follow instructions","temp.jpeg")
for i in range(5):
    TempTip_CheckFlag(TempTip_StartFlag,TempTip_BodyFlag,TempTip_AmbFlag,
                      TempTip_StartFlag_CHECK,TempTip_BodyFlag_CHECK,TempTip_AmbFlag_CHECK)
    print(BodyTemp_Data,HR_Data,SPO2_Data)
    update(patient_id,BodyTemp_Data,HR_Data,SPO2_Data)
window_3=popup().alert_popup("Follow instructions","temp.jpeg")
for i in range(5):
    TempRest_CheckFlag(TempRest_StartFlag,TempRest_BodyFlag,TempRest_AmbFlag,
                       TempRest_StartFlag_CHECK,TempRest_BodyFlag_CHECK,TempRest_AmbFlag_CHECK)
```

The first function (alert_popup) create a popup window and determine its shape by the Tkinter library

```python
    def alert_popup(self,title, img):
        """Generate a pop-up window for special messages."""

        self.root.title(title)
        w = 1200      # popup window width
        h = 750      # popup window height
        sw = self.root.winfo_screenwidth()
        sh = self.root.winfo_screenheight()
        x = (sw - w)/2
        y = (sh - h)/2
        self.root.geometry('%dx%d+%d+%d' % (w, h, x, y))
        x = Image.open(img)
        x = ImageTk.PhotoImage(x)
        w = Label(self.root ,image=x ,bg="white" , width=1200, height=750)
        w.pack()
        self.countdown(5)
        mainloop()
```

The second (countdown) is to turn off the screen and destroy the root after a certain time that we have specified.

```python
    def countdown(self,time):
        if time == -1:
            self.root.destroy()
        else:
            self.root.after(1000, self.countdown, time-1)
```

Then we start calling the function "CheckFlag" which sends flags to make the sensor start reading the vital sign from the patient and then sends the data to the Raspberry Pi sequentially through other flags.

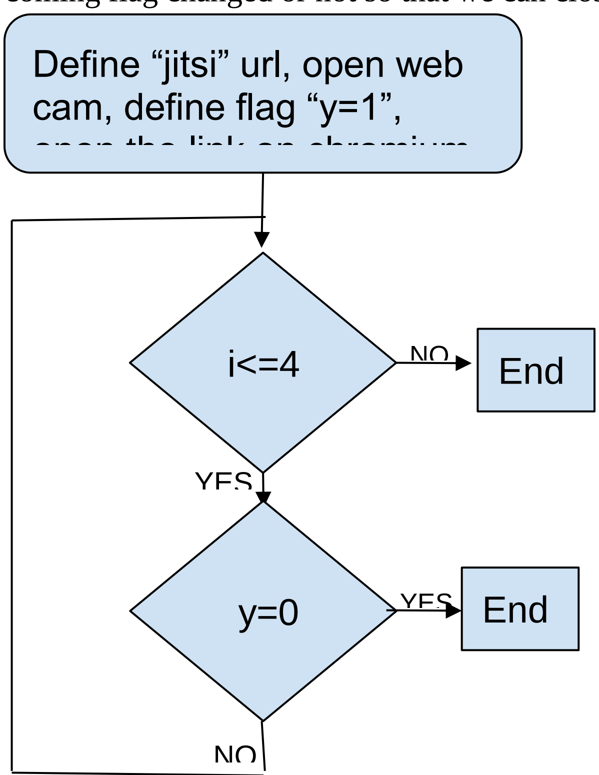
```python
def MAX30100_CheckFlag(StartFlagMAX30,HR_DataFlagMAX30,SPO2_DataFlagMAX30,
                                StartFlagCheckMAX30,HR_DataFlagCheckMAX30,SPO2_DataFlagCheckMAX30):
    global HR_Data
    global SPO2_Data
    ser.write(StartFlagMAX30.encode())
    time.sleep(1)
    Recive_StartFlagMAX30=ser.read()
    Recive_StartFlagMAX30=int.from_bytes(Recive_StartFlagMAX30,byteorder='big')
    print('startflag=',Recive_StartFlagMAX30)
    if(Recive_StartFlagMAX30 == StartFlagCheckMAX30):
        print("ok")
        ser.write(HR_DataFlagMAX30.encode())
        Recive_HR_DataFlagMAX30 = ser.read()
        Recive_HR_DataFlagMAX30 = int.from_bytes(Recive_HR_DataFlagMAX30,byteorder='big')
        print('HRflag=',Recive_HR_DataFlagMAX30)
        HR_Data = ser.read()
        HR_Data = int.from_bytes(HR_Data,byteorder='big')
        print('HRdata=',HR_Data)
        if(Recive_HR_DataFlagMAX30 == HR_DataFlagCheckMAX30):
            print("yes")
            s=ser.write(SPO2_DataFlagMAX30.encode())
            Recive_SPO2_DataFlagMAX30 = ser.read()
            Recive_SPO2_DataFlagMAX30 = int.from_bytes(Recive_SPO2_DataFlagMAX30,byteorder='big')
            print('spo2flag=',Recive_SPO2_DataFlagMAX30)
            Recive_SPO2_DataFlagMAX30 = ser.read()
            Recive_SPO2_DataFlagMAX30 = int.from_bytes(Recive_SPO2_DataFlagMAX30,byteorder='big')
            print('spo2flaggg=',Recive_SPO2_DataFlagMAX30)
            SPO2_Data = ser.read()
            SPO2_Data = int.from_bytes(SPO2_Data,byteorder='big')
            print('spo2data=',SPO2_Data)
            if(Recive_SPO2_DataFlagMAX30 == SPO2_DataFlagCheckMAX30):
                print("done")
```

After that, we send the data to the Database through the "update" function, and we make sure that the data is in a reasonable range by putting a condition to reduce the error.

```python
def update(n,temp_tip,hr,spo2):
    x.update({"patientid": n})
    temperature=temp_tip
    heartrate=hr
    spo=spo2
    if(temperature<=25 & temperature<=45 ):
        x.update({"temprature": temperature})
    if(heartrate>=40 & heartrate<=200):
        x.update({"heartrate": heartrate})
    if(spo>=80 & spo<=100):
        x.update({"spo": spo})
    sorted_string = json.dumps(x, indent=4, sort_keys=True)
    url="http://10.10.10.188/LoginRegister/update_VS.php"
    r = requests.post(url, json=x)
```

Iterate these functions four or five times in a for loop to make sure that the sensor got the correct data due to the low accuracy of the sensor.

If the purpose of the visit was video call, the second script will be run:
To open the video call first define the jitsi URL, open webcam, define the flag that has an initial value of zero and open the link on chromium.
When the doctor closes the call, the raspberry pi receives a flag so we loop to check if the coming flag changed or not so that we can close the call.



If the purpose of the visit was video call and measure vital signs, the third script will be run:
It is a combination of previous scenarios
First, measure the vital signs and then start the video call with the doctor.

C.DataBases:

We automated the record of patients' data which saves valuable data and makes it easy for doctors to review the case of their patients and this meets our objectives.

D.Doctor Mobile Application:

D.1.Retrieving Features:

The features we have provided in the doctor mobile app ensure that we can achieve distancing between medical staff and patients unless in emergency cases.

D.2.Video conference feature:

We implemented this feature to provide communication between the doctor through this embedded feature in the mobile app and the patient through the robot. With jitsi SDK in our project, we provided free and unlimited time conferencing which will be a very customized and cost-effective solution. This feature may be implemented with Android and IOS devices, but in our project, it is limited to Android smartphones.

E.Robot Motion:

First, we had to calculate the expected loads the motors would carry so that we could purchase the appropriate motors and wheels that would accommodate the weight of the robot.

We had problems with the movement of the robot. At first, the movement was unstable, or the motors stopped suddenly until we used a power generator to power 10 amps and 12 volts to the used motors.

E.1.Autonomous Motion:

Our result is better than old robots that only depend on ultrasonic sensors as it knows it passes not only avoiding obstacles, but it has some limitations as if the angle that the robot starts the track with is different, then the ranges of the order will be different.

E.2.Manual Motion:

In our project, the manual movement of the robot is implemented based on the Wifi technology which makes sending the commands very responsive and the range of the radio signals is a hundred meters, but on the other hand, it consumes high power. In some other applications, wireless communication is implemented with Bluetooth technology which consumes low power, but the radio signal range is ten meters[3].

F.Verification Algorithm:

Our verification algorithm is proof of our concept to verify the identity of the patients, but it has limitations as if the image captured was rotated, the OCR will not be able to detect the id number correctly. So, in future work, we can use more professional cameras that can focus to use QR codes.

Conclusion and future work

We can conclude that we built a telemedicine system with three main features, measuring vital signs, medical records, and video call feature. This system satisfies the main objectives of our project which are, Reducing medication error by automating the follow-up process and replacing the traditional way of follow-up for patients.

Future Work:

Hardware Side:

- Using design patterns as layered architecture, observer pattern, and state machine in STM32 parts.
- Writing STM32 drivers from scratch instead of using STm32CUBEMAX.
- Using ardu camera with a servo motor to be the only used camera in the robot.
- Turn robot movement completely autonomous using RPLIDAR and ROS.

Figure45: RPLIDAR

Figure44: RPLIDAR

Software Side:

- Building a mobile application that is compatible with both Android and IOS.
- Implementing the mobile Application using newer technology like Flutter.

References:

[1] "What's a CGM (Continuous Glucose Monitor) and Picking One." https://www.healthline.com/diabetesmine/what-is-continuous-glucose-monitor-and-choosing-one#What-is-continuous-glucose-monitoring-(CGM)? (accessed Jul. 23, 2021).

[2] "Glucose meter - Wikipedia." https://en.wikipedia.org/wiki/Glucose_meter (accessed Jul. 23, 2021).

[3] G. SinghPannu, M. Dawud Ansari, and P. Gupta, "Design and Implementation of Autonomous Car using Raspberry Pi," *Int. J. Comput. Appl.*, vol. 113, no. 9, pp. 22–29, 2015, doi: 10.5120/19854-1789.

[4] "Raspberry Pi 7" Touchscreen Display (Official) – Pi Supply." https://uk.pi-supply.com/products/raspberry-pi-7-touchscreen-display-official (accessed Jul. 22, 2021).

[5] "How To Schedule Python Scripts As Cron Jobs With Crontab (Mac/Linux) | Python Engineer." https://www.python-engineer.com/posts/cron-jobs-for-python/ (accessed Jul. 22, 2021).

[6] "I2C Communication Protocol - GeeksforGeeks." https://www.geeksforgeeks.org/i2c-communication-protocol/ (accessed Jul. 22, 2021).

[7] "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices." https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html (accessed Jul. 22, 2021).

[8] S. Y. H. Kit and N. M. Kassim, "Non-invasive blood glucose measurement using temperature-based approach," *J. Teknol. (Sciences Eng.*, vol. 64, no. 3, pp. 105–110, 2013, doi: 10.11113/JT.V64.2087.

[9] H. E. Farag, "A non-invasive method for measuring blood flow rate in A non-invasive method for measuring blood flow rate in superficial veins from a single thermal image. superficial veins from a single thermal image," 2013, doi: 10.18297/etd/426.

[10] "Build a Simple REST API in PHP | Okta Developer." https://developer.okta.com/blog/2019/03/08/simple-rest-api-php (accessed Jul. 24, 2021).

[11] "VishnuSivadasVS/Advanced-HttpURLConnection: Making HttpURLConnection easy and secure. Best method to impliment httpurlconnection in android." https://github.com/VishnuSivadasVS/Advanced-HttpURLConnection

(accessed Jul. 24, 2021).

[12] “Jitsi vs Zoom | 2021 Comparison.” https://stackshare.io/stackups/jitsi-vs-zoom (accessed Jul. 24, 2021).

[13] “Difference between Bluetooth and Wi-Fi - GeeksforGeeks.” https://www.geeksforgeeks.org/difference-between-bluetooth-and-wi-fi/ (accessed Jul. 24, 2021).

[14] “Car plate recognation using open CV | (OCR) | Computer vision شرح عربي - YouTube.” https://www.youtube.com/watch?v=eM9uaVUFmCY (accessed Jul. 24, 2021).