Title of project: Smart Portable Ventilator

Author DETAILS:

SURNAME & INITIALS: Phahlamohlaka M

EMAIL ADDRESS: mphophahhh@gmail.com

CONTACT NUMBERS: 0648917189

TABLE OF CONTENTS

PROPOSAL 4
INTRODUCTION 5
PROJECT OBJECTIVES 5
PROJECT DESCRIPTION 5
PROJECT TECHNOLOGY 5
TIMETABLE 6
BUDGET 6
CONCLUSION 6
REFERENCE 6
DESIGN DOCUMENT 7
INTRODUCTION 8
Purpose 8
Scope 8
Intended Audience 8
SYSTEMS OVERVIEW 8
DESIGN CONSIDERATIONS 8
ASSUMPTIONS AND DEPENDENCIES 9
Related software and hardware 9
End-user characteristics 9
General Constraints 9
GOALS AND GUIDELINES 9
DEVELOPMENT METHODS 10
ARCHITECTURAL STRATEGIES 10
SYSTEM ARCHITECTURE 11
NODEMCU ESP8266 Microchip 11
Blynk mobile application platform 11
Motor and Pulse heart rate sensor 11
Air pressure bag 11
LED's 11
Buzzer 11
LCD Display 11
Block diagram 12
DETAILED SYSTEM DESIGN 12
Activity diagram 13
Operational Status table 13
CONCLUSION 13
GLOSSARY 14
BIBLIOGRAPHY 14
USER DOCUMENT 15
SCOPE 16
Identification *16*
System overview *16*
Document overview *16*
REFERENCED DOCUMENTS 17
SOFTWARE SUMMARY 17

Software application *17*
Software inventory *17*
Software environment *17*
Contingencies and alternate states and modes of operation *18*
Security and privacy *18*
Assistance and problem reporting *18*

ACCESS TO THE SOFTWARE **18**

PROCESSING REFERENCE GUIDE **21**

NOTES **21**

REFERENCE **22**

APPENDIXES **22**

APPENDICES **23**

PROPOSAL

INTRODUCTION

The objectives of the project are simply to build a small smart ventilator that will be able to reduce patient work of breathing through using pressure to blow air into the patient lungs. The project will also have a feature that will able to measure patient heart beat rate using a sensor. The physical processes and the measuring of the patient heart beat rate will be monitored through LCD display, LED'S, Buzzer and a mobile application. Challenges related to the projects are shortage of materials to build the projects and some materials are expensive like self-inflating bag and also the design of the project power supply as the project have components that required uses different voltage as such the circuit has to have voltage divides. Problems/needs the project is solving are helping ordinary people to have their own ventilator in case of emergency, the ventilator is also portable you can just put it inside a bag and travel with, It also helps in case of pandemics where now we are experiencing global shortage of ventilators. Solving these problems will help save life and also improve health care mobility where people can just stay at home knowing they don't require to go to hospital for ventilator it will also help reduce hospital crowds. It cost less money to build the project compare to traditional ventilator, it can also be operated and monitored by someone who does not have any knowledge, the mobile application is easy to install, setup also to read values. Project uses less power and it can be operated by a portable battery that can fit inside a hand bag and also the battery can be charge with less expensive solar panel. To replace faulty components is easy and cheap.

PROJECT OBJECTIVES

Objectives that will be achieved include pressure to blow air into the patient lungs, measure heart beat rate using a sensor. Air pressure and the heart beat rate will be monitored on the mobile application and through buzzer and LED's on the circuit board. The project will be implemented by first gathering electronics components and materials to build the enclosure/cover. Second I will design the electric circuit that will be able to supply power to the entire system and attached the component on the design like buzzer, LED, LCD display, resisters and sensors. Third I will design the enclosure/cover using woods and plastic materials. Forth I will put the circuit inside the enclosure/cover to test if it fits. Firth I will write basic Arduino code to test all the components one by one if they are working. And if all the components are working I will download Blynk application on google play store and configure it to match the Arduino connections, After I will add GUI components on the app like buttons and sliders. After creating the app gui I will run test to check if there is a connection between the app and the circuit. Lastly I will run the test of the entire operation to check if the sensors, motors and other components react according to the way are expected to react also on the application if the data is showing is the data expected to show at that time.

PROJECT DESCRIPTION

I am building a smart portable ventilator system that uses less electricity and can be operated by a battery that can be charged by a small solar panel. People can take It anywhere as it is very portable. The project is currently in demand due to the outbreak of the corona virus. African countries are experiencing shortage of this type of medical equipment and with this project I believe it might be a way that can help with this shortage as is easy to build and cost less expensive compare to traditional ventilator. The target market for this project are people who are having breathing problem. Also it will be a good advice for everyone person to own one in case of an emergency, because I believe no one can predict when they might need it. The entire project is my own concept, it covers almost everything from supplying air, monitoring heart beat rate and use of mobile application to monitor all the processes. The project it won't include the air mask that the patient wear to receive the air pressure, because the air blower does not provide enough air is use only when designing a ventilator prototype.

PROJECT TECHNOLOGY

The technology that I will be using include a ESP8266 Microcontroller that will be used to control the entire process, sensors, buzzer, LCD display, LED's and communicate with the mobile application. Sensor I will be using Pulse Heart rate sensor to measure the patient heart beat rate. For air pressure I will be using rubber air blower. Mobile application I will be using Blynk which is a new IOT platform to host application and data on the cloud, it allows users to create a custom application for IOT projects.

TIMETABLE

Provide detailed information on the expected timetable for the project. Break the project into phases, and provide a schedule for each phase.

Activity	Description of work	Start and end dates
Collect electronic components	Collection of electronics components, microcontroller, sensors, resistors, motor etc.	16/05/2021 - 22/05/2021
Collect cover/enclosure materials	Collection of enclosure/cover materials, wood, gear and gear motor plastics.	23/05/2021 - 29/05/2021
Design the circuit	Designing of the circuit, constructing power supply points on the circuit, connecting the sensor to the circuit, connecting buzzer and LED's to the circuit.	16/05/2021 - 22/05/2021
Design the enclosure/cover	Designing the enclosure/cover on a paper to check how the circuit, motor and other components will fit.	23/05/2021 - 29/05/2021
Build the circuit and test	Building and testing the circuit to check if there are no errors.	01/06/2021 - 18/06/2021
Creating mobile application	Building the mobile application and link it to the circuit board and running tests to check for connectivity and physical responds from the ventilator.	19/06/2021 - 22/06/2021
Finalizing the project	Putting the circuit inside the enclosure/cover for final step in order to check the complete project if It is working.	23/06/2021 - 31/06/2021

BUDGET

State the proposed costs and budget of the project. Also include information on how you intend to manage the budget.

Items	Supplier	Quantity	Unit price	Total
Silicon rubber air blower	Takealot	1	R60	R60.00
High Torque Spur Gear DC Motor 640RPM 12V	Micro Robotics	1	R138.00	R158.70
NodeMCU ESP8266 – Original AI Thinker	Micro Robotics	1	R98.00	R112.70
Pulse Heart Rate Sensor	Micro Robotics	1	R130.00	R149.00
Resistors Pack	Communica	1	R23.68	R25.00
Magnetic Buzzer Round, orange, 1.5VDC	Communica	1	R11.70	R13.46
LED'S pack	Communica	1	R15.56	R17.45
PCB FG SS P15CM	Communica	1	R16.25	R18.69
RIBBON JUMPER 40W M/F 15CM	Communica	1	R20.00	R23.00
L298 DUAL H-BRIDGE MOTOR DRIVER	Micro Robotics	1	R40.00	R46.00
			Total budget	R624.00

CONCLUSION

Overall I feel that the project is will be pretty successful although there are going to be few problems here and there that I will encounter. This is low cost smart ventilator that will help reduce health care over load and it will work with no errors or failures, because I was able to reduce the complexity.

REFERENCE

Smart, Low Cost Ventilator Tunes to Patient's Own Breathing. (2020, APRIL 29). MEDGADGET EDITORS. Http: //www.medgadget.com/2020/04/smart-low-cost-ventilator-tunes-to-patients-own-breathing.html

DESIGN DOCUMENT

Introduction

Purpose

All presently available ventilator uses lots of power, they are complex and they are very expensive to own and they is also a very huge shortage of ventilator due to covid-19. The smart ventilator system that I am currently building uses less power, less complicated to build and also it cost less money to build it while still provide same pressure of air as traditional ventilator. The type of microchip used is very advance, allow all the processes to be monitored through mobile application and any user can read the data it does not need medical experience to understand the data that will be displayed on the application.

Scope

This document is intended to give detailed technical design of the entire ventilator project system. Making it easy for engineers to understand the architecture of the hardware and the application. The document also gives guide on how the different components work together the connection between them and the condition that the components work under. And the document also contains steps how to construct the project design, Process flow table and status flow table that shows different status and explanation of those status.

Intended Audience

The intended audience for this document are engineers specifically computer systems engineers, who want to understand the project design and also do modification where is necessary or improve the design to make it less expensive and use less power or even add more advance features to improve work efficiency. The engineers should have some knowledge of c++, electronics and have design skills.

Systems overview

The Project design consist of two sections the mobile application and the ventilator case. The ventilator case consists of electronic circuit and electronic components that controls the entire physical process using microchip output including the LED's, buzzer and motor which control the gears to put pressure on the air ball and provide the patient with air. After the microchip will receive data through inputs from the heart rate sensor. And the program will make adjustment depending on the data received from the input and also in a case where the patient heart beat is normal but struggling to breath the output signal can be send to the motor to increase the rate at which the motor control the gear so that the system starts to pump the air quickly. The microchip is connected to the internet cloud using Wi-Fi. The internet connection to happen the NodeMCU ESP8266 libraries must be installed to the Arduino ide also the blynk library. The mobile application blynk monitors the entire process, after the data have been sent to the cloud from the microchip the mobile app will connect to the internet cloud using Wi-Fi or sim card. Then the data is going to be fetch and displayed on the screen. Also in a situation where the user wants to manually increase the speed of the motor they can press buttons on the mobile app and data signal will be send from the app to the cloud and from the cloud to the microchip and the microchip will process the input data and make decisions based out of it will result in a physical form. The is a cooling fan inside of the case to cool down the circuit and the microchip in order to avoid heat failures and false input readings from the sensor. LED's and buzzer gives feedback regarding the state of the patient where the air pressure must be increased or decreased based on the heart rate sensor.

Design considerations

First issue that I have encountered is the size of the air pressure bag, I wanted to use an air pressure bag that will be able to supply enough air that can be used in real life scenario but the price was too expensive and I end up using a small air bag that can only be used for demo purposes. The second issue that I have encountered is the issue of voltage, the microchip outputs 3.3V and the motor needs 12V, Buzzer uses 5V and the LCD display uses 1.8V max, so I had to design the circuit board with transistors, diodes and regulators to supply those different voltages to the components. The other issue that I have encountered was the way in which the motor and the gear will be used in order to compress the air pressure bag to provide the air after de-compress and I ended up using two motor plastic gears the circle and the long ruler shaped gear to push the air pressure bag.

Last issue that I have encountered was the Microchip internet connectivity, it was resolved by downloading the Blynk library and NODEMCU ESP8266 library they provide easy to use internet connectivity to the cloud.

Assumptions and dependencies

Related software and hardware

The Project design involves many electronic components therefore they will long code to control all these components and that might require more advanced microchip, cause If the microchip speed becomes an issue the mobile application might not receive data in real time they might be delays and that will cause the systems to not give correct respond. The speed of the internet also is a key to the project to respond in real time if it happens the microchip network fails or the mobile application internet fails the system will fail and that might put the patient in danger. Also the power that supplies the microchip and mobile phone must be monitored to that it does not goes off and also use backup battery power in case the main fails

End-user characteristics

The mobile application settings are easy to change and as such the user must be careful when using the mobile application and also make sure that there is internet connection on the microchip and the mobile is working so that the system does not fail.

General Constraints

The hardware must be place in a room temperature not high temperature it might affect the outputs values. The user must use this project in an environment where there is high speed internet so that the connection between the microchip and the mobile applications does not fail. Also the project must be place in a room temperature or below room temperature, so that all the components can work properly and for a long period of time. Currently shortage of electronic components and air pressure bags, I had to change the project design to the type of components currently available.

Goals and guidelines

One of the goal of the system design is to build the circuit board that will be able to supply different voltage levels and current, also the circuit must have one point where you put input voltage to supply to all the direct point on the circuit including the microchip. And also the goal is to build a circuit board that consist of all components being independent of each other so that in a case of one component failing the entire system does not fail. The other goal is to make the project to be small in size so that it can fit in a normal bags that people carry in everyday life, so that in case of emergency the person can just pull it inside the bag and use it. It must also uses less power so that that it can be carried alongside the battery it uses or use solar panel to recharge it. The other goal is to make the microchip code too short so that the microchip can function at the highest speed with no delay that will result to system not responding to a way it was supposed to respond. And with short code the data will be transmitted fast to the cloud without delays and also when the mobile application send data to the cloud the microchip should be to get or import the data quick from the cloud. The other goal is to be able to monitor the process using chats and graphics and the blynk application provides that also to be able to monitor the process anywhere, so I decided to save the data from the microchip and the mobile application to the cloud using internet connectivity by connecting to the WIFI, and making sure that the WI-FI speed is very fast it does not cause delays during the operation. Also the goal is to use a heart rate sensor that will be able to measure heart rate on the hands figures so that it can be easy for the patient to wear the sensor. And the main goal is to be able to provide enough air pressure on the air pressure bag by using a 12V motor with high power to controller the gear with a high speed to compress the air pressure bag fast.

Development Methods

- First I downloaded the Arduino IDE, NODEMCU ESP8266 library and blynk library.
- Connect the NODEMCU ESP8266 to the computer via an usb cable, then go to the Arduino IDE and select the port on which the NodeMCU ESP8266 is connected and flash the microchip to remove any code that might be inside the microchip.
- Import the NODEMCU ESP8266 library and Blynk library to the code.
- Connect the NodeMCU ESP8266 to the WI-FI and after access the blynk cloud on the internet.
- Download blynk mobile application on an android mobile phone after sign-in.
- Link the mobile application with the microchip.
- Add two chart, and three buttons and one label to help monitor the process.
- Go to the Arduino platform and start coding the system, starting with writing code form main selection operation like what conditions will result in the motor turning and controlling the gear to put pressure on the air pressure bag.
- Code the heart rate sensor to check conditions so that the system adjust to meet the requirements at that time.
- Code LED'S so that they give feedback regarding what condition needs to be met and how will I go about to achieve the tasks of each component.
- Build the circuit board, by connecting the components to the circuit, sensors, led's, resistors, regulators and transistors.
- Create a voltage divider on the circuit to provide different voltages.
- upload the code to the NodeMCU ESP8266
- Make a rectangular shape with wood after cover five sides leave and leave the top side open.
- Attached the motor to the circle gear and attach the circle gear to the long ruler shaped gear that compress the air pressure bag.
- Attach the circuit board, cooling fan and the air pressure bag on the box.
- Put the power on and test if the system turns on, if the systems works try to check if the is connection between the box and the mobile application.

Architectural Strategies

I decide to create mobile application using Blynk platform, it provide easy and reliable ways to send the data between the mobile application and the Microchip. The Blynk send data at high speed to the cloud and you can use WI-FI or mobile data to connect to the cloud and also the application uses less mobile phone memory, CPU and storage which will results in the application working well without any disruptions. And to create the application the platform you don't need to know a programming language you just drag and drop widgets and configure the app to link with the Microchip. I decided to use NODEMCU ESP8266 Microchip it has libraries that allow easy internet connectivity, it also sends data at high speed to the cloud, uses less power and has many inputs and outputs pins to connect electronic components. The Microchip had RTS and flush buttons so that if the system fails the user can just press the buttons and remove the code inside and upload the new code. I decided to use 12V motor to provide more pressure to compress the air pressure bag, and the project design is easy to expand it as time goes on like changing the size of the air pressure bag putting bigger one so that it provides more air to the patient. And the control the 12V motor I decide to use voltage transistor out the output of the Microchip and supply 12V to the transistor to turn on the motor same applies to the motor in the future can be changed to 24V to provide even more power to the system without changing the Microchip or the entire design. I also decided to add old computer cooling fan to cool down the system when the components heat increases so that the sensor provide accurate values. The heart rate sensor I decided to use the Pulse Heart Rate Sensor since you can get readings by just placing the sensor on the hand finger not straight to the heart.

System Architecture

NODEMCU ESP8266 Microchip

I use Arduino IDE with c++ programming language to write the code that is going to be uploaded to the microchip to control the GPIO pins, write and read data from the external components. To link the application to the blynk cloud I downloaded the BLYNK library and import it to the code.

Blynk mobile application platform

First step I download the blynk mobile application on play store, sign-in and configure the application to send the data to the blynk cloud service and the blynk cloud service send and get the data to the microchip. After drag and drop GUI components to the dashboard.

Motor and Pulse heart rate sensor

The motor is connected to the microchip via two GPIO pins. The motor can be control through the mobile application button and the motor then control the gears that will be able to compress the air pressure bag. Also pulse heart rate sensor control the motor, when the microchip detects high pulse through analog pin from the sensor the microchip will activate the GPIO pins where the motor is connected with high speed and when the microchip detects low pulse from the sensor the motor GPIO pin will be activated in a way that it turn the motor with less power.

Air pressure bag

Air pressure bag conditions depend on the status of the patient, the air pressure pumps air at high level when the motor speed is high and pumps air low when the speed of the motor is normal. The air pressure is compress by the gears connected to the motor physically.

LED’s

The LED’s are connected to the microchip GPIO pins to detect system different status. When the green LED is on it means the heart rate sensor detected normal readings from the patient. When the orange LED is on it means the sensor now detected up normal level of heart rate and final when the red LED is on it means that the sensor detected high level of heart rate and as such the patient they might be in danger as such the microchip will adjust the operation.

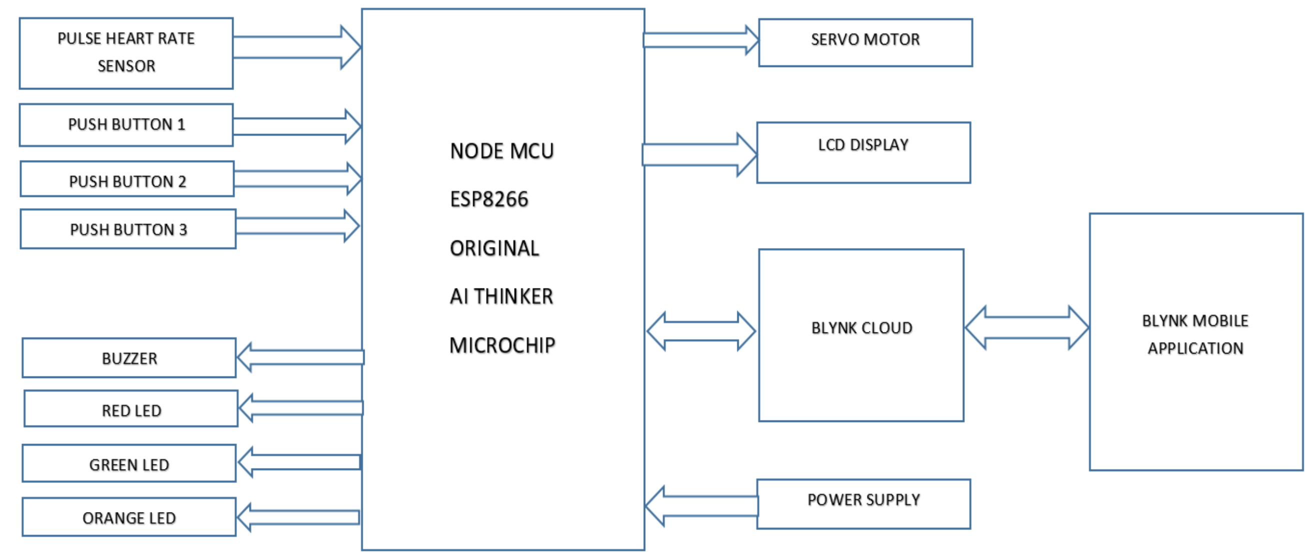
Buzzer

When the buzzer sound turns on it means the pulse heart rate sensor detected high level of heart rates from the patient as such the sensor send back the data to the microchip and the microchip send output to the GPIO pin where the Buzzer is connected.

LCD Display

LCD display values according to the system status, when there are high values detected by the sensor the microchip activates the GPIO pins to display three (3) on the LCD display, average sensor values LCD display displays two (2) and below average sensor values the LCD display displays 1.

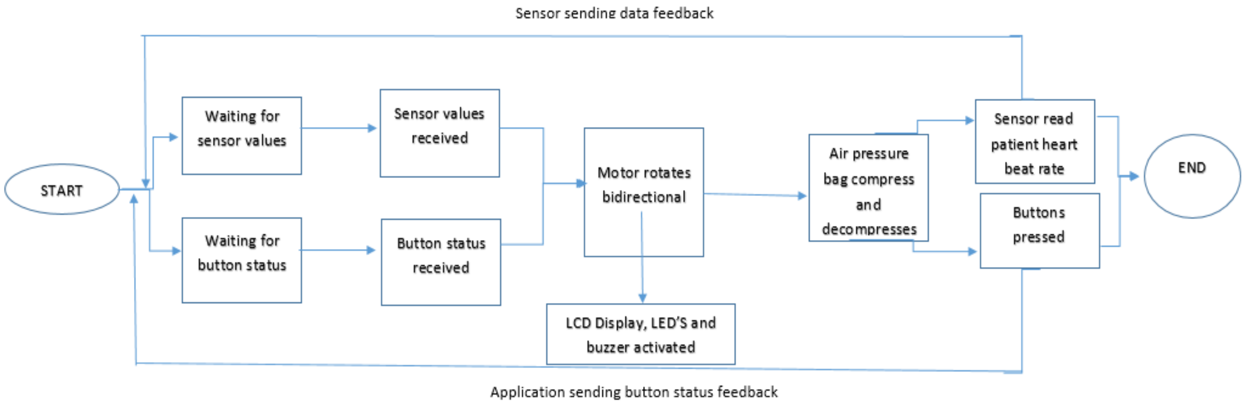
Block diagram



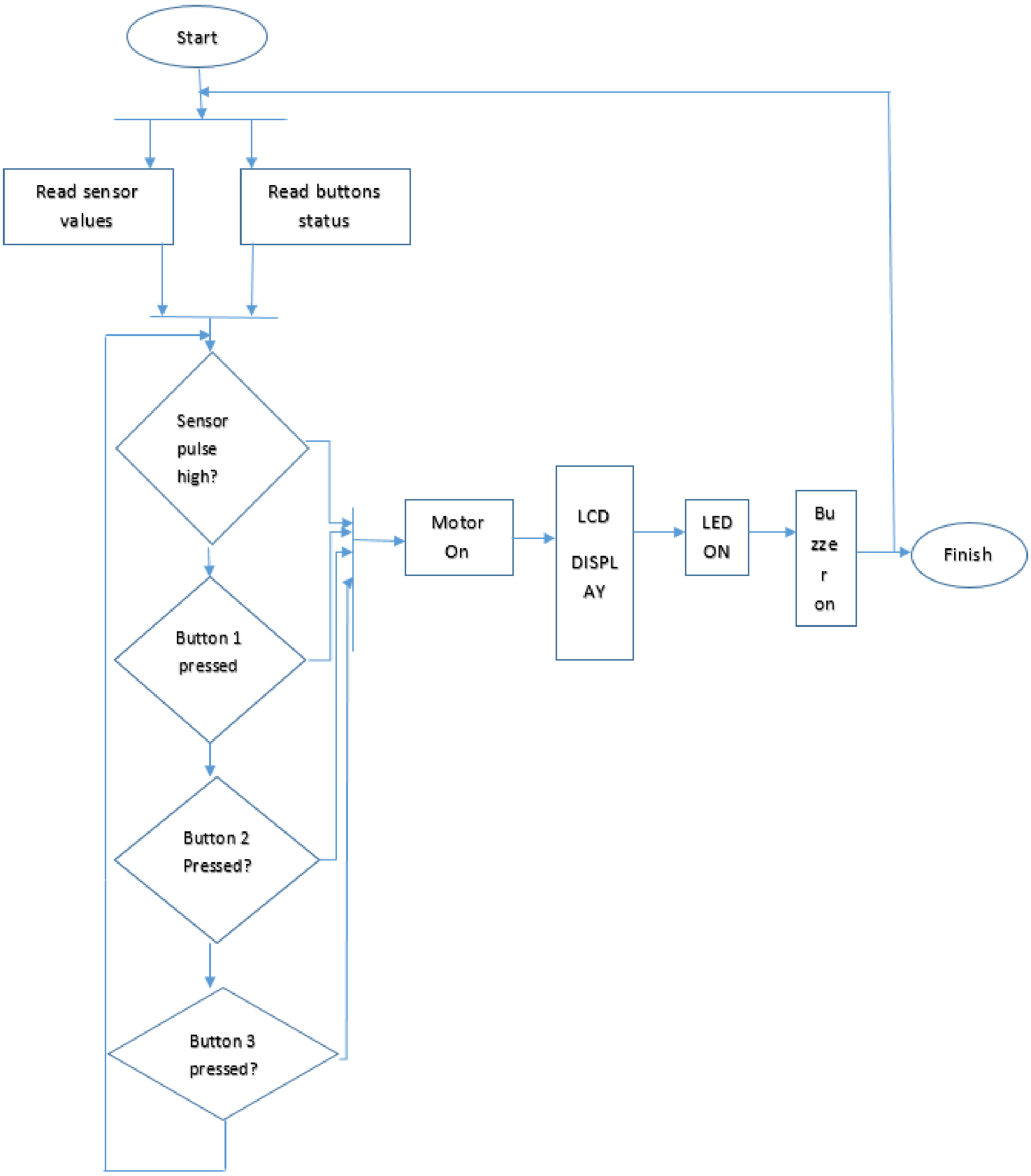
Detailed System Design

The NODEMCU ESP8266 microchip will control the entire system, the microchip have 32 KiB instruction and 80 KiB data. And there are 16 GPIO pins to connect inputs and outputs also there is 1 analog pin. The high torque spur gear dc motor 100RPM 12V will be connected to two GPIO pins via a transistor that supply 12V to the motor and the base of the transistor is connected to the microchip for bidirectional purpose and the spur gear motor produce high torque relative to the physical size and use a simple metal gearbox and the motor advantage is the compact design which enables the high speed. The pulse rate sensor will be connected to the ADC0 which is an analog pin and it essentially combines a simple optical heart rate sensor with amplification and noise cancellation circuitry making it fast and easy to get heart readings, you just have to put it on the fingertip and supply it with the 3.3V from the GPIO. The buzzer will be connected to one GPIO pin, each LED's also will be connected to one GPIO pin and uses 3.3V. The LCD display will be connected to 7 GPIO pins and supply 3.3V. And to setup the blynk mobile application I click on the new project and it show pop screen where you have to enter the name of the project and select the board as ESP8266 and below the textbox the is a authentication token that I copy to the Arduino IDE after I click create button, It will take me to the dashboard then I drag and drop the two chart to represent the Pulse heart rate and the motor levels and configure them as Digital GP pins.

Flow diagram



Activity diagram



Operational Status table

Control condition	Pulse heart rate sensor	APP buttons	Motor	LCD display display number	Buzzer	Red LED	Orange LED	Green LED
CASE A	HIGH	0	ON	3	ON	ON	OFF	OFF
CASE B	LOW	1	ON	1	OFF	OFF	OFF	ON
CASE C	LOW	2	ON	2	OFF	OFF	ON	OFF
CASE D	LOW	3	ON	3	ON	ON	OFF	OFF

Conclusion

The project design document provides is easy to build guide, also it cost less money to build the project and work very excellent like a traditional ventilator. The design was broken down it to different subsystems so that if one subsystem fails that part can be identified easily and also if there are modifications the parts effected can be easily be removed. This design uses less memory since half of the operation is taken by microchip the other half by the mobile application.

Glossary

- Ventilator, is an appliance for artificial respiration.
- Microchip, is a tiny piece of material that contains electronic circuits allowing it to store a lot of information.
- C++, is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language.
- NodeMCU ESP8266, is an open-source Lua based firmware and development board specially targeted for IoT based Applications.
- Blynk, is designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.
- Heart pulse, is the number of times your heart beats per minute. A normal resting heart rate should be between 60 to 100 beats per minute, but it can vary from minute to minute.

Bibliography

ASOK,A.(2020,April,29).COVID-19 Emergency Ventilator. create.arduino.cc. (https://create.arduino.cc/projecthub/anuvindasokin/covid-19-emergency-ventilator-6d4821)

Schalk, G.(2004). Software design document, Albany New York, Eberhard Karls, University Tubingen.

USER DOCUMENT

SCOPE

Identification

Smart portable ventilator is a system that will be able to reduce patient work of breathing, the system consists of microchip, sensor, LCD display, buzzer, LED's and motor. The entire system can be monitored via a mobile app. The project will be the first demo and version. The code is written in c# Arduino IDE version 1.8.15 and the mobile application BLYNK version is 1.8.15. The microchip used is NodeMCU ESP8266-Original AI Thinker, features Wi-Fi connectivity, onboard CP2102 and keys. The Arduino IDE and libraries are installed on windows 10 pro, Version 21H1. The hardware and components will be placed inside the enclosure made out of light wood and the design is flexible, project can be placed anywhere as long temperature is room temperature.

System overview

System Purpose: The purpose of the hardware system is to provide the patient with enough air when they are having breathing and to check their heart beat rate. The patient will receive the air pressure from the air pressure ball and the heart rate measure sensor will be placed on the patient finger after the sensor will send data to the microchip. The microchip will process the data and read three button status to make decisions. The microchip will also send data to the Blynk cloud service.
Software Purpose: The Arduino ide will be used to program the microchip, also they will be libraries that has to be downloaded like ESP8266 and the BLYNK library to allow communication between the microchip and the mobile application. The mobile application helps with monitoring the entire system, the motor speed, heart beat rate and the patient breathing conditions.
Software User: The ventilator can be used by anyone; you don't have to understand advanced technology in order to use it. Is user friendly and the hardware have only three buttons to control the operation. The mobile application the user just opens it and automatically it will start receiving values from the hardware as long as the mobile is connected to the internet.
Developer: in order to develop the system and the mobile application, you will have to know how to code in c# also understand how to integrate mobile application and hardware using networking and systems designs skills.
Operating sites: The system can be operated at any environment as the as the temperature is at room temperature. And the system must be placed on top of the table or something that is sustainable so that it can be balanced properly.

Document overview

The purpose of the document is to describe how the entire system including hardware and mobile application where constructed using various technologies. Is a guide to what the system does, how the system operate, List the features and functionalities the system provides to the uses or patients. The document must be capped safe since it contains detailed information about the entire system, hardware architecture, network architecture and the mobile application architecture, with this user manual anyone with technology background might be able to edit the system and if proper process are not followed correctly the system might cause harm to human. The system enclosure must always be closed tight so that there are no external interferences. The document also show how the data is being transferred from the sensor to the microchip via Analogue pin and how the microchip process all the data from buttons status to sensor and send it to the BLYNK cloud service and there after share with the mobile application.

REFERENCED DOCUMENTS

- Smart, Low Cost Ventilator Tunes to Patient's Own Breathing. (2020, APRIL 29). MEDGADGET EDITORS. Http: //www.medgadget.com/2020/04/smart-low-cost-ventilator-tunes-to-patients-own-breathing.html

- ASOK,A.(2020,April,29).COVID-19 Emergency Ventilator. create.arduino.cc. (https://create.arduino.cc/projecthub/anuvindasokin/covid-19-emergency-ventilator-6d4821)

- Schalk, G.(2004). Software design document, Albany New York, Eberhard Karls, University Tubingen.

SOFTWARE SUMMARY

Software application

- **The software application** will be used to write instruction to the microchip, so that the microchip can execute task based on the code written. The mobile application plays a role of sending data to the BLYNK cloud service, allow users to monitor the patient progress anywhere around the world as long the app is connected to the internet. The app also changes the system state by pressing any on the three buttons on the gui, and the app can also be used to check if the hardware is failing or working based on the data fetched from the cloud service that was send from the microchip to the cloud. On the app dashboard there is also a graphs representing heart beat rate and the state of the patient as a whole.
- **The hardware** is responsible for all the physical process and microchip operations. Including controlling the speed of the motor to compress the air pressure bag to provide the patient with air, Sending input signals to the LCD display, LED'S and the buzzer. The microchip will receive sensor values from the analogue pin and then check the level of patient heart beat rate after send signals back to the GIOP and also send the sensor readings to the cloud, so that the mobile application can receive the same data.
- **The hardware is link to the mobile application** via a cloud based communication. The ESP8266 and the BLYNK libraries also enable fast and easy internet connections between the hardware and the application.

Software inventory

The document contains all the operation of the entire systems and list of important libraries, software and features that makes up the system. The document shows how the system can be maintained, modified and configured to be used in any situation or that required by the engineer who will be doing the changes. The code that is inside the microchip is written in c# and uploaded to the microchip using Arduino IDE. As a result If there is an error in the code the IDE will pick it up before programing the microchip. Same applies to the BLYNK mobile application, if the app is not configured correctly they won't be any communication between the microchip and the app, for security purpose. The must always be backup power to the system, because if the main power goes off and the is no backup power the patient can be at serious risk. Access to the system must be restricted, only the people who are taking care of the patient can monitor the system and they must also be trained on how to use the system.

Software environment

The hardware can be placed anywhere under room temperature, the user must make sure the hardware is stable, there is a backup power supplied to the inverter incase the main power goes off. The hardware

must also be monitored everyday checking if all the components are not over heating or starting to fail. He hardware can also be used while in motion or at home if the patient did not have enough time to reach hospital. The hardware comes with everything connected so that the user can just plug and use without any configurations.
The mobile application need to be downloaded in google play store and be installed on an android mobile application or download the application on apple store and install it on IOS.
The hardware must be connected to the WIFI also the mobile must have data on to provide connection between the hardware and the application.

Software organization and overview of operation

The whole system depends on the hardware, and the hardware is link to the cloud and also the mobile application link to the cloud. The hardware and the mobile application communication architecture is designed to achieve the following properties:

- Reduce patient work of breathing through using smart technology that is portable.
- Helps the patient to breath out carbon dioxide, a harmful waste gas the body needs to get rid of.
- Cuts cost and save electricity as compare to traditional ventilator.
- The operation can be monitored in two ways in case one way fails, on the hardware enclosure and through mobile application.
- Easy to use and provide fast runtime.

Contingencies and alternate states and modes of operation

If it happens the system fails, putting the patient in danger, the user can quickly switch off the entire system by pressing the power button on the enclosure, or the user can just check on the mobile phone the readings that are incorrect then after try to replace the component that correspond with those readings. Also the users must first check if the systems respond to all the condition before assigning the system to the patient, by just tracking the status on the LCD display or the mobile application. For future purpose the system will be built in a way that all the electronic components including the sensors are just plug or pinned on the circuit board without being soldered so that when one component fails the user can just check the user manual to trace which components failed and replace it easily without having to wait for someone who know advance technology to come and repair the system.

Security and privacy

The operating system used to install the Arduino IDE, also the Arduino IDE and the microchip are all protected by the international copyright law that state that in order to do any modification you must first contact the developer of the system, hardware or the software. And there are also under laws that protect the design and the ways the project is being created.

Assistance and problem reporting

The mobile application can also be used as an error check, when the system fails the users can easily see via the mobile application that the graphs they are not showing the data that correspond with the condition of the patient at that moment in time. The user also has access to contact the developer of the system to guide them on the troubleshooting process, since is easy for anyone to operate the ESP8266 microchip on any project.

ACCESS TO THE SOFTWARE

Access control

- The hardware operation does not restrict user access; anyone can have access to the hardware if is placed in an open space or as long they have the patient or patient care taker permission.
- The hardware must be placed in a safety location, where few people have access since it does not have security features on it.
- On the mobile application the user in order to monitor the system progress they will have to enter security pin on the mobile then after they can have access to use the application.

Installation and setup

The following are the steps of how the user will get to have and use the software on their device.

- First the user must power on the system hardware and check if the power light is on.
- Connect the hardware to the Wi-Fi and also the user must make sure the Wi-Fi have internet access.
- Download the BYLNK mobile application on play store or IOS store and make sure the mobile is connected to the Wi-Fi or data.
- Configure the mobile application to be link to the microchip via a BLYNK cloud service.
- Run the mobile application and check if the app picks up the microchip from the cloud.
- Test the system by putting the sensor on your finger tip and check if there are readings on the LCD display or mobile application.

Initiating procedure

- The first step in developing the system, was to make sure the microchip is connected to the internet via Wi-Fi, making sure the microchip receives the sensor values taken from the patient also the button states when the user presses the buttons.
- The mobile application was downloaded from the android play store, configured to work with the ESP8266 microchip.
- The last step is check if the sensor values correspond with the speed of the motor.

Description of inputs

Input conditions

- The mobile application has three inputs, the buttons that control the state of the motor, which is low speed, average speed and fast speed.
- The hardware has four outputs, the sensor and three buttons that also control the state of the motor.
- There is also a power button on the hardware that control the entire system power.

Input formats

- The Input format, the user can press three buttons on the mobile application to control the operation of the motor, the buttons state is represented as Boolean inside the application.
- Second Input format, the user can press three buttons on the hardware that send Boolean data to the microchip to make decisions.

Composition rules

- The system forces all the inputs from the mobile application entered by the user also on the hardware entered by the user to be Boolean.
- But inside the code the strings are used and integers are used to make the logic of the program and to transfer data easy to the cloud.

Input vocabulary

- The mobile application use string, Integers and Boolean only as input.

- The hardware receive data as string and Boolean from the GIO pins.

Sample inputs

- The mobile application uses only buttons as inputs to enable the user to change the operation of the entire system. The buttons inside the application is converted to Boolean after send to the cloud based services where the microchip can read the button states.
- The hardware also has buttons as input and the microchip reads the GIO pins after converts the status of the buttons to Boolean inside the code. Also the sensor works as an input to the microchip and converts the data to string.

Description of outputs

General description

The system consists of hardware and the mobile application. The hardware is link to the mobile application via the internet and the BLYNK cloud service. The system has three different states first state the system provides low air pressure when the heart rate sensor detects low level of patient heart beat rate, the second the state the system provide average air pressure to the patient if the sensor detects normal heart beat rate. The last state the system provides high air pressure to the patient when the heart beat rate is very high. All this three state can be configured manually by pressing the three buttons on the hardware enclosure. And also the three states can be configured automatically by the heart beat rate sensor detecting different levels of stages of the patient heart beat rate.

Output formats

The mobile application outputs data to the BLYNK cloud service. The data that the mobile application outputs is strings, integers and Boolean when the user presses buttons on the application. Same applies to the microchip, the microchip outputs data to the cloud service in a form of strings, integers and Boolean. The microchip also send output signals to the GIO pins to activate the motor, LCD display, LED's and the buzzer.

Sample outputs

The Hardware activates the gio pins using string and Boolean inside the code to send signal to the motor, LED's and buzzer.

Recovery and error correction procedures.

- When the mobile application is not responding or working properly, the user must first check if the mobile data is on or the Wi-Fi connected to is on.
- The user must check also check if the application is still configured to link with the microchip.
- If the application is still failing the user must uninstall it and download it again from the play store or iOS store.
- If the hardware fails, the user must first start to checking if the power is supplied to the hardware.
- When the motor is not working the way is supposed to work, the user must first check if there are reading coming from the heart rate sensor on the LCD display or on the mobile application.
- When the entire system fails the user can simply visit the system manual and disconnect everything after connect it again including re-downloading the mobile application.

Stopping and suspending work

If the system losses power and the backup power also runs out, the system will fail automatically. When the user wants to switch off the system they can press the power button on the system enclosure. The user can also switch off the link between the hardware and the application by just disabling the internet connection or pressing disconnect button on the mobile application.

PROCESSING REFERENCE GUIDE

Capabilities: The functions of the hardware and the application is to let the user be able to monitor the entire process on the mobile application and on the enclosure of the hardware. The user also has access in deciding what state the system can operate under that matches the patient health condition. If the user wants to stop the entire operation first they must disconnect the microchip from the mobile application and close the application, after the user can switch off the system by pressing the power button on the enclosure.

Conventions: The user can physical press the state buttons on the mobile application, and the application will send data in a form of a string to the cloud and the microchip will receive that string and process it then sends the feedback and signal to the GIO pins to activate the components that are aligned to the string operation. The user can also press the button on the enclosure, the microchip will receive the Boolean status of the buttons after activates the align components and send the data to the cloud service for the mobile application to get the data.

Processing procedures: In order to operate the system, fist the user must switch the system on, connect the system to the Wi-Fi, Open the mobile application, connect the mobile phone to the Wi-Fi or switch the data on. The user must also make sure the backup power battery is full and last every time the system is being used. Lastly the user must test if the system is still working correctly by placing the heart rate sensor on tip of the finger to check if it records corrects readings and also put a hand on the air pressure bag to feel the level of air pressure coming out of the air bag.

Aspect of software use: The mobile application has three gui buttons that represent air pressure state and gui graphs that represent the level of the patient heart rate. The user must make sure the application is link to the hardware so that they exchange data through the cloud. Power is the only thing that can affect the operation of the system as a whole if the is no backup power placed aside or connected to the inverter.

Related processing: If the system does not respond at all, the user must start troubleshooting checking if the system is connected to the Wi-Fi, if the sensor is still working by placing the sensor on the fingertip and monitor the reading from the LCD display and also on the mobile phone. If the motor does not respond according to the way is expected to work, the problem is caused by the buttons and the sensor if the microchip reads wrong data coming from them.

Recovery from errors, malfunctions, and emergencies: When the system is failing to work and the user have gone through all the troubleshooting processes, the user must press the power button to turn off the entire system. The system can also automatically fix an error after start to display the correct readings.

NOTES

Glossary

- The ESP8266 - is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espressif Systems in Shanghai, China. The chip first came to the attention of Western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker.

- BLYNK – is a mobile application that was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool applications.
- c# - is a general-purpose, multi-paradigm programming language encompassing static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.
- Arduino IDE - is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

REFERENCE

APPENDIXES

Smart, Low Cost Ventilator Tunes to Patient’s Own Breathing. (2020, APRIL 29). MEDGADGET EDITORS. Http: //www.medgadget.com/2020/04/smart-low-cost-ventilator-tunes-to-patients-own-breathing.html

ASOK,A.(2020,April,29).COVID-19 Emergency Ventilator. create.arduino.cc. (https://create.arduino.cc/projecthub/anuvindasokin/covid-19-emergency-ventilator-6d4821)

Schalk, G.(2004). Software design document, Albany New York, Eberhard Karls, University Tubingen.

APPENDICES

(Source code and picture of project)

CONFIGURING THE MOBILE APPLICATION

ESP8266 source code (Arduino IDE)

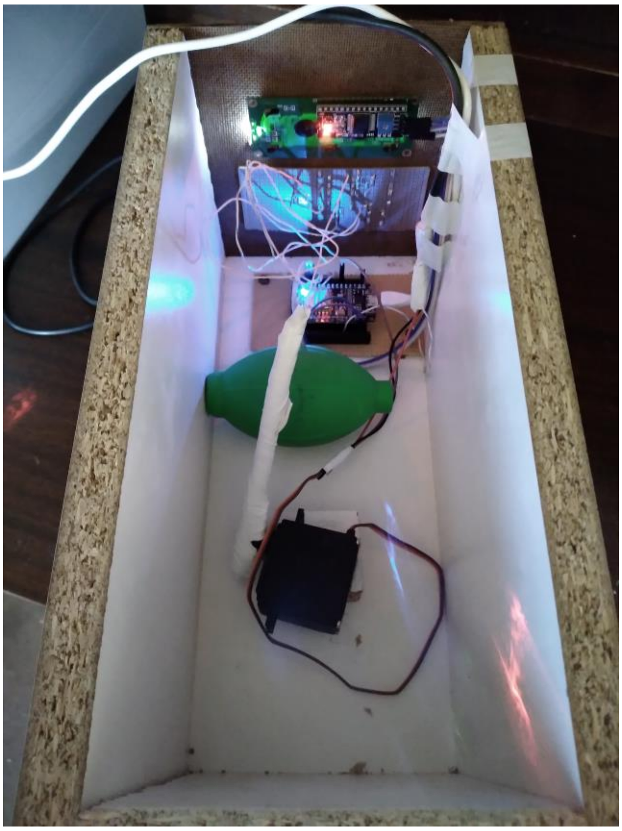
```
#define BLYNK_PRINT Serial // Blynk cloud communication
#include <ESP8266WiFi.h> // library for intergrading esp8266 to arduino ide
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>  // Controlling servo motor
#include <Wire.h>   //  library for I2C protocol
#include <LiquidCrystal_I2C.h>  // library for I2C LCD

//Get LCD library object
LiquidCrystal_I2C lcd(0x27,16,2);

//unique key to connect to the Blynk
char auth[] = "itiQ9xAuIvA6KyfEG6scNfA3SU1cv7Na";

// WiFi credentials.
// Wifi name and password for the network.
char ssid[] = "12345";
char pass[] = "@Mphoza20";

//Servo motor object
Servo myservo;

//Application buttons status virables
int level1;
int level2;
int level3;

//Application vitual pins
#define VPIN_BUTTON_1    V1
#define VPIN_BUTTON_2    V2
#define VPIN_BUTTON_3    V3

//physical inputs and outputs
int Button1 = D6;
int Button2 = D7;
int Button3 = D8;
int Buzzer = D5;


//heart rate sensor stored virables
int PulseSensorPurplePin = 0;
String Signal="";
int Threshold = 550;

//Setup controls and Internet connctions
void setup()
{
  //Setting LED, Buttons and Buzzer as inputs and outputs
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(Button1, INPUT);
  pinMode(Button2, INPUT);
  pinMode(Button3, INPUT);
  pinMode(Buzzer, OUTPUT);

  //Attaching the pin the servor motor is connected
```

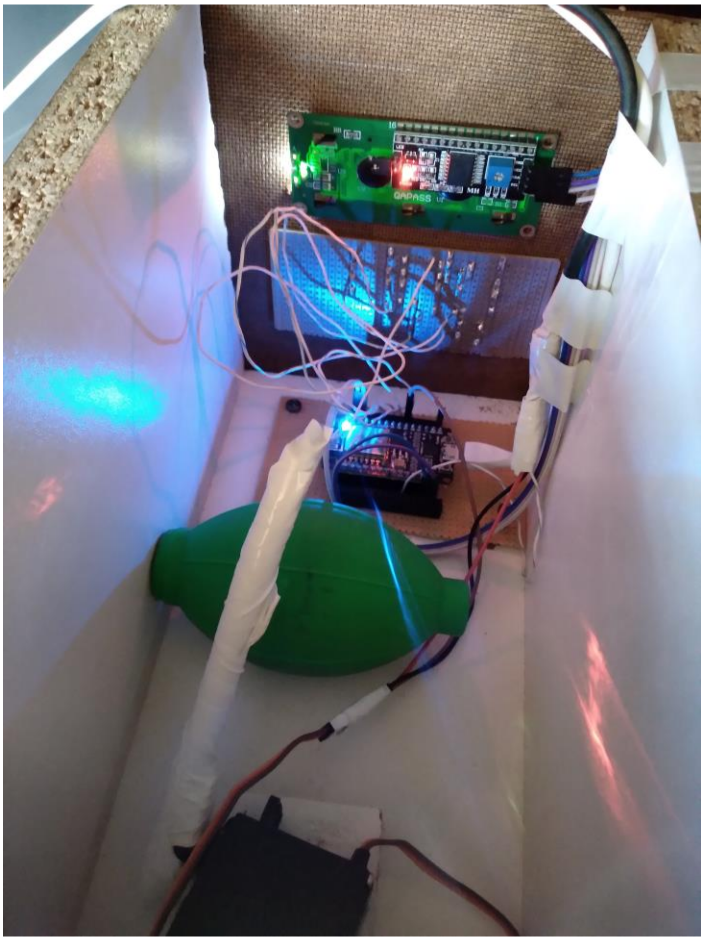
```
  myservo.attach(D7);

  // Debug console
  Serial.begin(9600);

  //Starting LCD Display
  lcd.begin(16,2);
  lcd.init();
  lcd.backlight();

  //Connecting to the Blynk Cloud service
  Blynk.begin(auth, ssid, pass);

}

//Send data to the Blynk cloud patform(representing heart rate sensor)
BLYNK_READ(V5)
{

  if(digitalRead(Button1)==HIGH)
  {
    Blynk.virtualWrite(V5,50);
  }
  else if(digitalRead(Button2)==HIGH)
  {
    Blynk.virtualWrite(V5,75);
  }
  else if(digitalRead(Button3)==HIGH)
  {
    Blynk.virtualWrite(V5,100);
  }else{
    Blynk.virtualWrite(V5,0);
  }

}

//Send data to the Blynk Cloud platform(representing stages of the system)
BLYNK_READ(V4)
{

  if(level1==1)
  {
    Blynk.virtualWrite(V4,1);
  }
  else if(level2==1)
  {
    Blynk.virtualWrite(V4,2);
  }
  else if(level3==1)
  {
    Blynk.virtualWrite(V4,3);
  }
  else{
    Blynk.virtualWrite(V4,0);
  }

}

//Get Mobile application Buttons Status from the Blynk platform
```

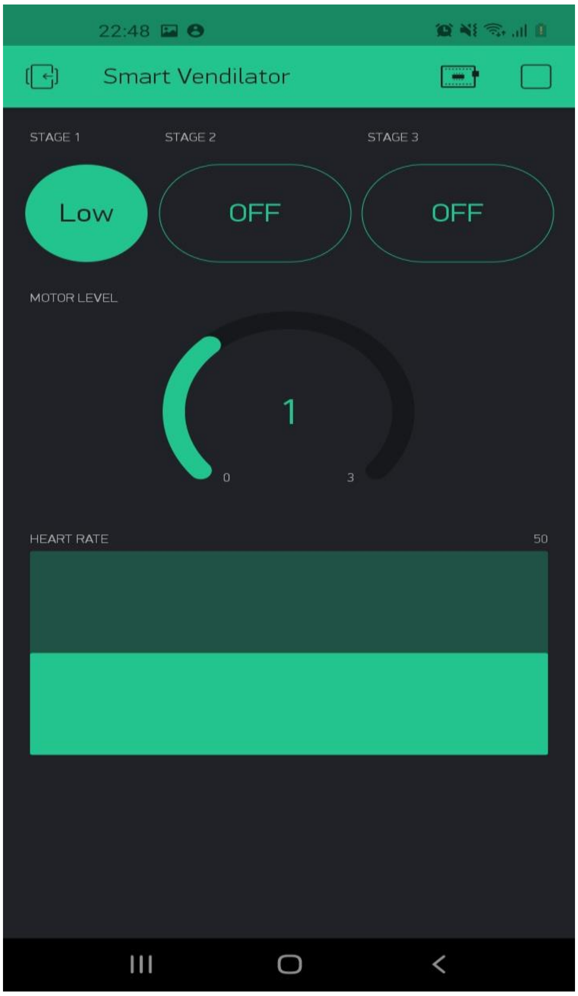
```
BLYNK_WRITE(VPIN_BUTTON_1)
{
 level1 = param.asInt();
}
BLYNK_WRITE(VPIN_BUTTON_2)
{
 level2 = param.asInt();
}
BLYNK_WRITE(VPIN_BUTTON_3)
{
 level3 = param.asInt();
}

//Runtime operation  of the system
void loop()
{
 //Intergrade the Blynk Library to arduino
 Blynk.run();

 //heart rate sensor data tracking
 Signal = "0%";
 String Display = "";

 //Status of the Physical Buttons that represent the sensor
 int button1Status = digitalRead(Button1);
 int button2Status = digitalRead(Button2);
 int button3Status = digitalRead(Button3);
 String SystemStatus ="";

 //Print data to the first line of the LCD
 lcd.setCursor(0,0);
 lcd.print("Vendilator On");

 //Stage 1 operation, where the systems povide low air pressure
 if(level1 == 1 && level2 == 0 && level3 == 0)
 {
  if(button1Status ==HIGH)
  {
   Signal ="50%";
  }

  digitalWrite(LED_BUILTIN, HIGH);

  SystemStatus = "LOW";

  Display = "BPM:"+String(Signal);


  lcd.setCursor(0,1);
  lcd.print(SystemStatus+" "+Display);

  //run the servo motor
  myservo.write(0);
  delay(2000);
  myservo.write(180);
  delay(2000);

 }
 // Stage 2 operation, where the system provide normal air pressure
```

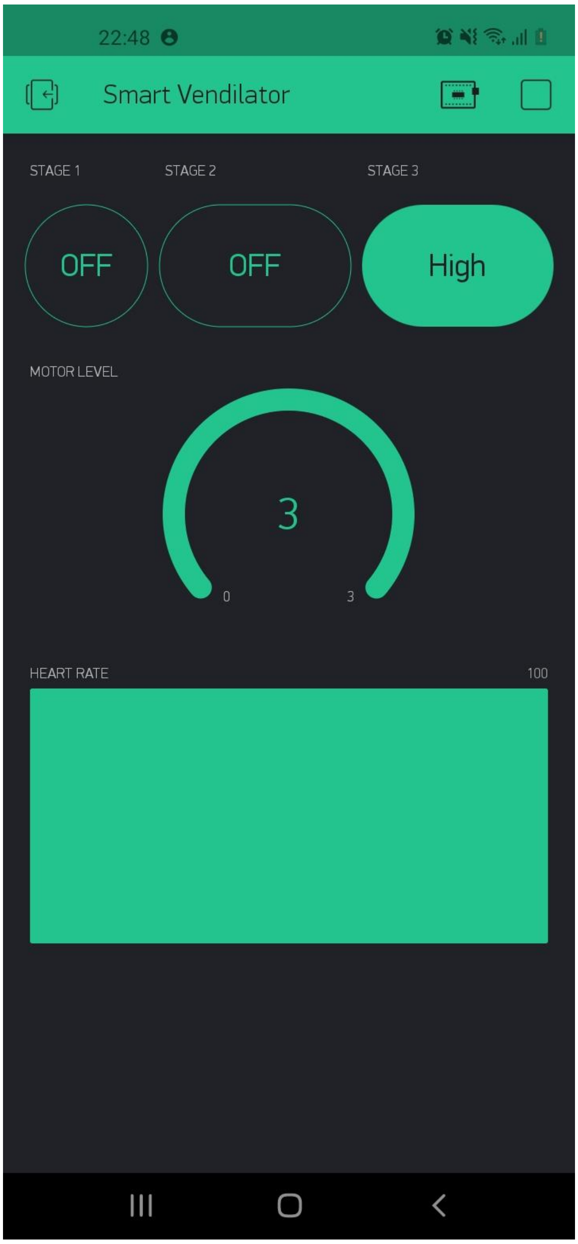
```
 else if(level2 == 1 && level1 == 0 && level3 == 0)
 {
   if(button2Status ==HIGH)
   {
     Signal ="75%";
   }

  SystemStatus = "Normal";
  Display = "BPM:"+String(Signal);

  lcd.setCursor(0,1);
  lcd.print(SystemStatus+" "+Display);

  digitalWrite(LED_BUILTIN, HIGH);

  //run the servo motor
  myservo.write(0);
  delay(1000);
  myservo.write(180);
  delay(1000);

 }
 // Stage 3 operation, where the system provide high air pressure
 else if(level3 == 1 && level1 == 0 && level2 == 0)
 {
  if(button3Status ==HIGH)
  {
  Signal ="100%";
  digitalWrite(Buzzer,HIGH);
  }else{
    digitalWrite(Buzzer,LOW);
  }

  SystemStatus = "HIGH";
  Display = "BPM:"+String(Signal);

  lcd.setCursor(0,1);
  lcd.print(SystemStatus+" "+Display);

  digitalWrite(LED_BUILTIN, HIGH);

  //run the servo motor
  myservo.write(0);
  delay(500);
  myservo.write(180);
  delay(500);

 }
 //Turn the system off
 else {
  myservo.write(0);
 }

 //Clear LCD Data
 lcd.clear();
}
```

Picture of Project








22:48
Smart Vendilator
STAGE 1
STAGE 2
STAGE 3
OFF
OFF
High
MOTOR LEVEL
3
0
3
HEART RATE
100