

IoT BASED SLEEP APNEA PREVENTION AND DETECTION

Submitted in partial fulfillment of the requirements for the award of

Master of Technology degree in Medical Instrumentation

By

Jennifer. J (41440007)



DEPARTMENT OF BIOMEDICAL ENGINEERING

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY – 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Jennifer. J** (Reg. No. **41440007**) who carried out the project entitled "**IoT Based Sleep Apnea Prevention and Detection**" under our supervision from **November 2022 to April 2023**.

P. Grace
Dr. P. GRACE KANMANI PRINCE, M.E., Ph.D.,

Guide

T. Sudhakar
Dr. T. SUDHAKAR, M.Sc., Ph.D.,

Head of the Department

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

Chennai Nagar Rajiv Gandhi Salai

Submitted for Viva-voce Examination held on 05.05.2023

T. Sudhakar
Internal Examiner

R. M. 552
External Examiner

DECLARATION

I, **Jennifer. J** (Reg. No. **41440007**) hereby declare that the project report entitled "**IoT Based Sleep Apnea Prevention and Detection**" done by me under the guidance of Dr. **P.GRACE KANMANI PRINCE**, M.E., Ph.D., Department of Biomedical Engineering is submitted in partial fulfillment of the requirements for the award of **Master of Technology** degree in **Medical Instrumentation**.

DATE: 05/05/2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. **T. SUDHAKAR**, M.Sc., Ph.D., Head of the Department, and Department of Biomedical Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. **P. GRACE KANMANI PRINCE**, M.E., Ph.D., for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all **Teaching** and **Non-teaching staff** of the Department of Biomedical Engineering who were helpful in many ways for the completion of the project.

I want to extend my sincerest thanks to all my **Family members** and **Friends** for their encouragement and support during my project work.

ABSTRACT

Sleep is necessary for human health, although its specific physiological purpose is uncertain. Breathing dynamics are a critical diagnostic tool in therapeutic settings such as sleep analysis, intensive care, and central nervous and physiological disease analysis. Sleep apnea occurs when airflow to the lungs is interrupted for 10 seconds or more during sleep, often due to a loss of neuronal input from the central nervous system (Central Sleep Apnea) or upper airway collapse. (Obstructive Sleep Apnea). To solve this issue, a microcontroller-based sleep apnea monitor has been created. A respiration sensor monitors breathing conditions, a pulse sensor and SPO2 track heart rate and blood oxygen saturation, a digital humidity and temperature sensor, and a microphone detects snoring sounds are all included in the device. The microcontroller continuously checks all data, and if any breathing issues are identified, the patient's support system is activated immediately. This helpful technique assists patients in achieving consistent breathing, which can lessen or eliminate snoring.

Keywords: *Sleep Apnea, Compressor, ThingSpeak*

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
	Abstract	V
	List of Abbreviations	Viii
	List of Figures	X
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
3	AIM AND SCOPE	6
	3.1 Existing System	6
	3.1.1 Disadvantages	6
	3.2 Proposed System	6
	3.2.1 Advantages	6
	3.3 Scope	6
4	METHODS AND MATERIALS	7
	4.1 Block Diagram	7
	4.2 Circuit Diagram	8
	4.3 Flow Diagram	9
	4.4 Hardware Requirements	10
	4.4.1 Arduino Nano	10
	4.4.2 ESP32	13
	4.4.3 DHT11Sensor	16
	4.4.4 Respiration Sensor	19
	4.4.5 MAX30100	21
	4.4.6 Accelerometer Sensor	24
	4.4.7 Compressor	26
	4.4.8 Power Supply	28
	4.5 Software Requirements	32
	4.5.1 Arduino IDE	32
5	RESULTS AND DISCUSSION	40
6	CONCLUSION AND FUTURE SCOPE	46

6.1 Conclusion	46
6.2 Future scope	46
REFERENCE	48

LIST OF ABBREVIATIONS

AC	- Alternating Current
ADC	- Analog -To- Digital Conversion
AMPS	- Ampere
ATmega	- Atmal, mega
BLE	- Bluetooth Low Energy
CAN	- Controller Area Network
COM port	- Communication Port
CPAP	- Continuous Positive Airway Pressure
CPU	- Central Processing Unit
CSA	- Central Sleep Apnea
DAC	- Digital-To- Analog Conversion
DC	- Direct Current
EDR	- Endpoint Detection and Response
ESP32	- Espressif system
ESP-IDF	- Espressif IoT Development Framework
GND	- Ground
I/O	- Input /Output
I2C	- Inter -Integrated Circuit
I2S	- Integrated Inter-IC Sound Bus
IC	- Integrated Circuit
IDE	- Integrated Development Environment
IoT	- Internet of Things
JTAG	- Joint Test Action Group
LCD	- Liquid Crystal Display
LED	- Light Emitting Diode
MEMS	- Micro- Electro Mechanical System
MMC	- Multi Media Card

NC	- No Connection
OSA	- Obstructive Sleep Apnea
PCB	- Printed Circuit Board
PWM	- Pulse Width Modulation
RAM	- Random Access Memory
RH	- Relative Humidity
RTC	- Real -Time Clock
SD	- Secured Digital
SPI	- Serial Peripheral Interface
SRAM	- Static Random Access Memory
UART	- Universal Asynchronous Receiver-Transmitter
USB	- Universal Serial Bus
VCC	- Voltage common collector

LIST OF FIGURES

FIGURE NO.	DESCRIPTION	PAGE NO.
4.1	Block diagram	7
4.2	Circuit diagram	8
4.3	Flow diagram	9
4.4	Arduino Nano	11
4.5	ESP32	14
4.6	DHT11Sensor	17
4.7	Respiration Sensor	20
4.8	MAX30100	22
4.9	Accelerometer Sensor	24
4.10	Compressor	27
4.11	Block Diagram	28
4.12	Circuit Diagram	29
4.13	Bridge Rectifier	30
4.14	Output of a Rectifier	31
4.15	7812 regulators	32
4.16	Types of Regulators	32
5.1	Hardware Setup	40
5.2	LCD Output	41
5.3	Graphical representation of Humidity	41

5.4	Graphical representation of Heart Rate	42
5.5	Graphical representation of Spo2	42
5.6	Graphical representation of temperature	43
5.7	Graphical representation of snoring	43
5.8	Graphical representation of acceleration	44
5.9	Graphical representation of compressor	45

CHAPTER – 1

INTRODUCTION

Millions of people across the world suffer from the widespread sleep ailment known as sleep apnea. It is characterized by respiratory pauses during sleep that might last several seconds or more. If left addressed, this interruption, which can occur numerous times throughout the night, might result in serious health issues.

Daytime exhaustion, memory loss, headaches, and other symptoms are all caused by sleep apnea. Additionally, it has been connected to more severe medical disorders like diabetes, high blood pressure, heart disease, and stroke. Up to 1 billion individuals may be affected by sleep apnea, according to estimates of the disorder's increasing prevalence globally.

There are various forms of sleep apnea, but obstructive sleep apnea is the most prevalent. (OSA). The person snores or gasps for air as a result of the muscles in the back of the throat failing to keep the airway open while they are sleeping. When the brain fails to tell the muscles to breathe, central sleep apnea (CSA), a less common variant, develops.

Even though sleep apnea is a serious disorder, it is treatable. Treatment options range from less intrusive approaches like lifestyle modifications like weight loss and abstaining from sedatives and alcohol to more invasive procedures like surgery or the help of a continuous positive airway pressure (CPAP) machine. Preventing long-term health issues and enhancing the general quality of life requires early detection and treatment of sleep apnea. Receiving care is essential. Therefore, it is always better to prevent illness than to treat it. Building a detection and prevention tool is so important.

The system's design and implementation were made possible by the use of the flexible and programmable processors Arduino and ESP32. The MAX30100, DHT11, respiration sensor, and acceleration sensor are just a few of the sensors that have been added to track breathing patterns and other physiological data as you sleep. The system also comprises crucial parts including the compressor, mask, step-down transformer, and relay for delivering the necessary air pressure for treating sleep apnea.

The system also includes manual and automatic modes, giving users more freedom and control over how the system behaves. While the manual mode enables users to change the settings and get feedback based on their particular needs, the automated mode may be utilized to continuously monitor and adjust the air pressure.

Healthcare providers can assess patient progress and change treatment parameters as necessary with the help of IoT-based sleep apnea prevention and detection systems. For patients who reside in remote or underdeveloped locations or who have limited access to specialized medical care, this can be especially beneficial.

The potential to improve the quality of life for the millions of people worldwide who are plagued by this sleep condition exists in the field of IoT-based sleep apnea prevention and detection. The continuous research and development in this field has the potential to fundamentally alter how sleep apnea is identified, treated, and managed, leading to better health outcomes and an overall improvement in quality of life.

CHAPTER – 2

LITERATURE SURVEY

Scott F. Roush et.al. (2004) Conducted an observational study, Preeclampsia was identified and treated in a 25-year-old woman who was 37 weeks pregnant and gravida 4 para 2. Maternal oxygen desaturation and a concurrent lowering of the foetal heart rate were seen during the patient's apneic episode. A child that was too small for its gestational age was then born to her. The obstructive sleep apnea was formally established by polysomnography in this case, and maternal desaturation was noted along with foetal heart rate slowdowns. Negative effects for the mother and foetus may be avoided with early identification of obstructive sleep apnea (OSA).

Zafar A. Usmani et.al. (2012) conducted an observational study, matching participants for age, sex, and BMI included 28 pSS patients and 18 healthy volunteers. Each patient underwent a polysomnography test the next day. We also compared how frequently obstructive Apneas and hypopneas occurred during sleep in the two groups, as well as their levels of fatigue, sleepiness, anxiety, and sadness. Through the use of correlation analysis, the relationships between sleep study variables and excessive sleepiness and tiredness were examined.

Mamoun Al-Mardini et.al. (2014) designed a device, the approach makes use of the built-in sensors, pervasiveness, computational power, and user-friendly interface of a smartphone to screen OSA. We generally use three sensors to get physiological data from patients: an oximeter to track oxygen saturation, a microphone to capture breathing effort, and an accelerometer to record body movement. Finally, we compare the ability of our method to screen for the condition to the industry standard using 15 samples to evaluate it. The results showed that while 100% of patients were correctly identified as having the condition, only 85.7% of patients were correctly identified as not having it. These first results underscore the critical role smartphones play in healthcare and emphasise the significance of the built system's performance in relation to the gold standard.

Ina Djonlagic et.al. (2015) conducted a study. At the Brigham and Women's Hospital Sleep Disorders Laboratory, the first 44 patients who met the inclusion criteria had either a diagnostic polysomnography or a CPAP titration trial were included in the

study. Following polysomnography (PSG), participants were divided into three groups: (1) healthy controls (control group), (2) obstructive sleep apnea patients who did not receive treatment (OSA group), and (3) OSA patients who had a full night of CPAP therapy. (CPAP group). In contrast to the OSA group, participants had to have an AHI of more than 15/h in order to be classified as healthy controls.

Ahmed S. BaHammam et.al. (2017), the majority of TECSA cases experience self-limited and persistent apnea episodes during the initial PAP titration, which has been suggested as a therapy method. A follow-up is necessary to make sure that the emergent important events disappear with continued therapy. This will be possible by tracking the PAP saved data. However, the therapist ought to recognise and deal with any problems that could be causing treatment-emergent CSA. PAP should be reduced and titration should be done gradually to address the issue of over-titration. When a major mask leak is discovered, the interface needs to be changed. PSG may be repeated in difficult circumstances.

Mohamed Nabi EL Gharib (2019) developed a system, In this method, the patient falls asleep with a mask placed over their nose, forcing air into their nasal passages with the force of an air blower. The air pressure needs to be precisely controlled, constant, and continuous in order to prevent throat contractions while you sleep. Although nasal CPAP prevents airway obstruction, sleep apnea episodes can occasionally reoccur when it is used improperly or is stopped altogether. (Figure 1). using a CPAP machine while ill Women who have gestational sleep apnea during their pregnancy run the chance of continuing to have severe respiratory suppression after giving birth. As a result, utilising medicines that prevent breathing should be limited and regulated by experts.

Anatolii Petrenko (2019) developed a system, the accuracy of determining the breathing pattern in various settings for various physical states of patients is improved when two-dimensional conversion accelerometer signals are used as opposed to not using them for diagnosing and monitoring sleep apnea at home. One-dimensional (1d) accelerometer signals are transformed into two-dimensional (2d) graphical images, which are then analysed by CNN using numerous processing layers.

Hiwa Mohammadi et.al. (2021) conducted a case study, 23 healthy controls and

31 OSAS patients (mean age, 48.50). Every individual had polysomnography the entire night. For those who had OSAS, it was further divided into groups of mild, moderate, and severe cases.

Balavenkata B chatruvedula et.al. (2021) developed a method for the treatment, Patients with underlying pathologies such as obstructive adenoids, chronic mouth breathing, asthma, and those with a BMI of greater than 24.9 were also disqualified, as were those who had already finished an interceptive orthodontic/orthopaedic therapy phase. The study was limited to the lateral cephalograms that closely fitted the aforementioned specifications. the cerebral pictures. The upper and lower airways' dimensions were also measured. The airway measurements were compared between the three groups using one-way ANOVA tests, and the odd's ratio was calculated to find a relationship between the airway measurements and the growth trend of the person. In order to statistically examine the data, IBM SPSS software, version 20, was used.

Prathap chandar manivannan et.al (2021) developed a system for making an empirical and historical diagnosis of OSAS. Confirmation is provided by polysomnography. (PSG). Background and analysis: A thorough medical history should be taken, including details on both nocturnal and diurnal symptoms as well as OSAS-related morbidities like neurobehavioral deficits, behaviour, fatigue, failure to thrive, and systemic hypertension. A person's history of snoring at night is the most crucial factor in predicting OSA. Allergy rhinitis or any other ailment that could make nasal airflow resistance worse, as well as the relative size (micrognathia) and location (retrognathia) of the mandible, should all be taken into account.

CHAPTER – 3

AIM AND SCOPE

The aim of the project is to develop an Internet of Things-based system for the prevention and detection of sleep apnea using a variety of sensors and processors including the Arduino and ESP32. For individualized treatment choices, the system will have both automatic and manual modes. In order to provide users with an intuitive experience, the project will investigate remote monitoring and integration with other IoT devices.

3.1 EXISTING SYSTEM

Monitoring respiratory dynamics is a crucial diagnostic technique in a number of therapeutic settings, including sleep analysis, intensive care, and the study of physiological and central nervous system disorders. The interruption of airflow to the lungs for 10 seconds or more during sleep is referred to as sleep apnea. All of the systems and research that are now in place merely monitor and detect sleep apnea.

3.1.1 Disadvantages

- Even if there were a cure, none of them would be affordable, thus there is no effective way to treat them.
- The average person cannot afford such high prices.

3.2 PROPOSED SYSTEM

According to the protocols and studies in place, sleep apnea can only be monitored and detected. There isn't a good way to treat them, or if there was, it wouldn't be economical. That price is beyond the means of the average person.

3.2.1 Advantages

- Sensors are utilised to analyse and provide measures for healthy sleep without inconvenience
- Average person to readily afford it.

3.3 SCOPE

To improve the quality of life for those who have sleep apnea, the ultimate goal is to create a system that is effective and affordable.

CHAPTER – 4

METHODS AND MATERIALS

To collect real-time data, this methodology connects a number of sensors, including the MAX30100, DHT11, Respiration Sensor, and Accelerometer Sensor, with an Arduino Nano and ESP32, LCD display, and ThingSpeak platform. The Arduino Nano reads the sensor data, which is then transferred to the ThingSpeak platform through ESP32 for analysis and visualisation. These sensors work best together for a variety of applications, including smart home automation, environmental monitoring, and health monitoring systems.

4.1. BLOCK DIAGRAM

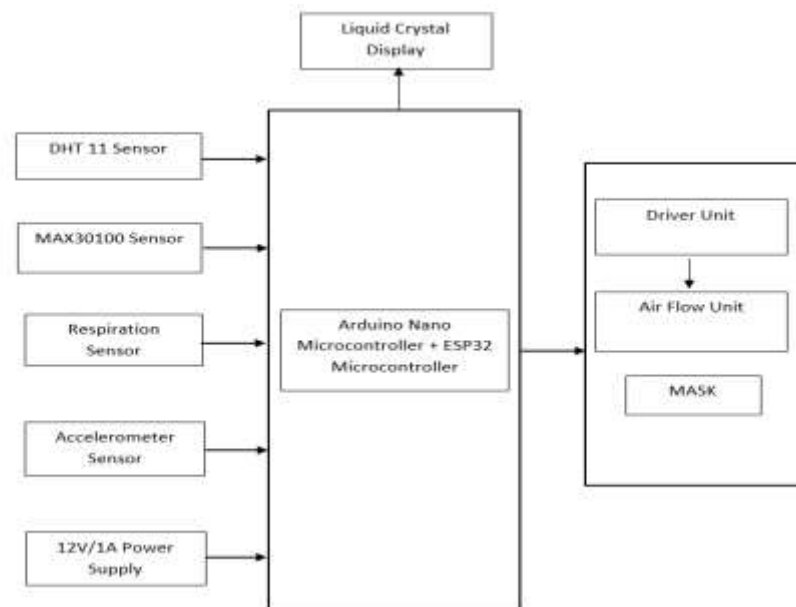


Fig. 4.1: Block diagram

Two microcontrollers, the Arduino Nano and the ESP32, are used. The core controller board for interacting with various sensors and parts is called an Arduino nano. For monitoring and control, the ESP32 module offers Wi-Fi and Bluetooth communication. The real-time data and system status are shown on the LCD panel as shown in fig 4.1. The respiration sensor computes the rate of breathing. The acceleration sensor is employed to gauge the body's motion and vibration. The patient's temperature and humidity are measured via the DHT11 sensor. The MAX30100 sensor detects pulse oximetry, which is used to gauge heart rate and blood oxygen saturation

levels. the compressor, step-down transformer, and relay that make up the driving unit. The power supply's voltage can be increased or decreased using a transformer. Relays are used to turn on or off high current loads. This device operates at 230 volts; the step-down transformer converts 230 volts to 12 volts, which is then supplied to the relay, which then converts the 12 volts to 9 volts, which is then supplied to the device as 5 volts.

4.2 CIRCUIT DIAGRAM:

Phase P is connected to the positive terminal of the transformer and neutral N is connected to the negative terminal of the transformer. Phase P is connected to the compressor and NO of the relay is connected to another terminal of the compressor. The positive terminal and negative terminal of 2 pushbuttons are connected to digital pins and ground. The positive terminal of the LED is connected to the Vin of the Arduino Nano, and the negative terminal is connected to the Ground Gnd of the Arduino Nano. Two data pins of the LED display are connected to the analog pins of the Arduino Nano. The Vin of Max30100 is connected to the Vin of Arduino Nano, and the SCL of Max30100 is connected to the digital pin of Arduino Nano. X, Y, and Z of the accelerometer are connected to the analog pins of Arduino Nano. 0 pin is connected to the ground of the Arduino Nano. The positive and negative terminals of the respiration sensor is connected to ESP32 analog pins. The positive and negative terminal of DHT11 sensor is connected to ESP32 analog pins. 3 terminals of a relay are connected to the positive terminal, Ground and Digital pin of ESP32 as shown in fig. 4.



Fig. 4.2: Circuit diagram

4.3 FLOW DIAGRAM

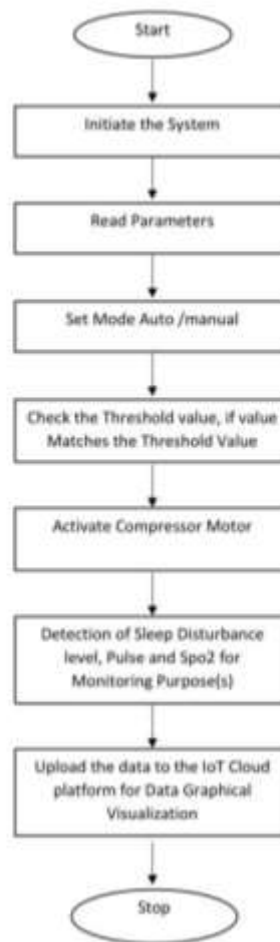


Fig. 4.3: Flow diagram

ThingSpeak is a program used for the IoT platform. A hotspot connection to a computer or laptop is required before turning on the gadget. Using the associated email ID and password, log in. The caregiver or doctor should sign in using the channel ID on their smartphone or another device; the application displays the channel ID for personal use. Before starting the kit, read all the parameters. There are two modes; Automatic and manual. The device will detect the patient's breathing if the automated mode is activated; if there is no breathing, the compressor starts and blows air through the mask. On the LCD, a 10-second countdown will be displayed as shown in fig.4.3.

It will automatically shut off the compressor after 10 seconds. If necessary, they can manually turn the air supply on and off while it is in manual mode only if the carer

or nurse always monitors the patient. If not, we can turn it off. With the help of the ON/OFF switch, they can manage the kit. The LCD will show the sensor output, including heart rate, SPO2, humidity, temperature, and acceleration. It appears steady if breathing is normal. With the help of the ON/OFF switch, they can manage the kit. The LCD will show the sensor output, including heart rate, SPO2, humidity, temperature, and acceleration. It appears stable if breathing is normal. The output of the readings will be displayed as a data sheet and a graphical representation, which will help the doctors diagnose the patient because it will be automatically stored in the cloud for later use.

4.4 HARDWARE REQUIREMENTS

- Arduino Nano Microcontroller
- ESP32
- DHT 11 Sensor -Temperature and Humidity Sensor
- Respiration sensor
- Max 30100 sensor-Pulse oximeter and heart rate sensor
- Accelerometer sensor
- Driver unit
- Air flow unit
- Mask
- 12V/1A power supply
- Power chord

4.4.1 ARDUINO NANO

A lightweight, open-source microcontroller board named Arduino Nano is based on the ATmega328P processor. Because of its small size and low power consumption, it is a popular board in the Arduino family. The board is an adjustable platform for designing and testing new electronic devices and can be used in a wide range of electronic projects, from robotics to home automation.

The Arduino Integrated Development Environment (IDE), a free software tool for developing and uploading code to the board, can be used to program the Arduino Nano

board. The IDE is user-friendly and includes several of libraries and examples to assist newcomers in learning how to program.

One of the unique characteristics of the Arduino Nano is its small size, which makes it suitable for use in portable and wearable devices is shown in fig. 4.4. The board is smaller than a credit card measuring approximately 18 mm by 45 mm. Because of its small size, the Arduino Nano can be utilized in a variety of devices, including sensors, data loggers, and remote controls.

Low power consumption, which makes it excellent for usage in battery-powered devices, is another important feature of the Arduino Nano. With a maximum consumption of current of 200mA, the board can function in the 5–12V voltage range. Despite its low power requirements, the Arduino Nano can be powered for long periods of time using small batteries, like coin cell batteries.



Fig. 4.4: Arduino nano

The Arduino Nano is known for its adaptability in addition to its compact size and low power consumption. The board may be programmed to carry out several of activities and is compatible with a wide range of sensors and electronic components. Robotics, home automation, and Internet of Things (IoT) projects frequently use the Arduino Nano.

The airflow of a CPAP machine can be monitored and controlled using an Arduino Nano in sleep apnea devices. It can be set up to read data from sensors that measure air pressure and flow, and then adjust the flow rate in order to maintain airway pressure at an appropriate level. In order to effectively manage sleep apnea, it is necessary to make sure the patient is receiving the right amount of air pressure for keeping their airway open.

Monitoring the patient's sleep patterns is another way that Arduino Nano can be included in sleep apnea devices. It can be set up to warn the doctor or guardian when

the patient stops breathing. This may help to spot patterns in the patient's breathing during sleep and alter the treatment strategy accordingly.

It works effectively with portable medical equipment due to its tiny size and low power consumption. It is an essential instrument for the treatment of sleep apnea due to its capacity to track and control the airflow in a CPAP machine as well as recognize changes in a patient's sleep patterns.

As a result, Arduino Nano is a potent and adaptable microcontroller board with several of uses in the healthcare sector. It is a vital instrument for the treatment of sleep apnea because of its tiny size, low power consumption, and capacity to monitor and regulate the airflow in a CPAP machine. The Arduino Nano can enhance the efficiency of sleep apnea therapy and patient health in general by providing precise pressure and flow control, monitoring patient compliance, and observing sleep patterns.

SPECIFICATIONS

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (6 of which are PWM)
- Analog Input Pins: 8
- Clock Speed: 16 MHz
- Dimensions: 18mm x 45mm
- Weight: 5 g
- Programming Interface: USB (CH340G driver)
- Operating Temperature: -40°C to 85°C

TYPES OF ARDUINO BOARD

- Arduino Nano
- Arduino Nano 33 IoT
- Arduino Nano Every
- Arduino Nano BLE

- Arduino Nano BLE Sense
- Arduino RP2040 connect

ADVANTAGES:

- Wide availability Compatible with a wide range of sensors and components
- Low cost
- Small size
- Low power consumption
- Easy to use
- Wide range of libraries and examples.

4.4.2 ESP32:

Espressif Systems designed and manufactures a strong microcontroller known as the ESP32. It is the ESP8266's replacement and is powered by a dual-core 32-bit CPU with integrated Wi-Fi and Bluetooth connections. The ESP32 is a flexible microcontroller used in a variety of applications, such as IoT, robotics, automation, and more. It has a wide range of features and capabilities.

Low power consumption is one of the ESP32's primary characteristics, which makes it the perfect option for battery-powered applications. A deep sleep mode on the microcontroller can considerably increase the battery life of devices that use the ESP32 is shown in fig. 4.5. The microcontroller offers numerous power-saving modes.

Additionally, the ESP32 offers a variety of communication interfaces, such as SPI, I2C, UART, CAN, and Ethernet, which allow connections to other components and sensors. The ESP32 is a great choice for Internet of Things applications since it also has built-in Wi-Fi and Bluetooth connectivity that enables wireless communication with other devices.

The ESP32's extensive array of peripherals, which includes ADC, DAC, PWM, RTC, and other features, is another crucial aspect. A variety of applications are made possible by these peripherals, which make it simple to interact with sensors, actuators, and other devices.



Fig. 4.5: ESP32

The ESP32 is supported by a strong software ecosystem that includes the Arduino IDE, the ESP-IDF (Espressif IoT Development Framework), and Micro Python in addition to its hardware capabilities. Even for developers with little to no expertise with microcontrollers, this software support makes it simple to develop and deliver apps on the ESP32.

In general, the ESP32 is an effective and adaptable microcontroller that is suitable for a variety of applications. It is the perfect option for IoT, automation, robotics, and many other applications due to its low power consumption, communication interfaces, a wide range of peripherals, and software support.

The multipurpose ESP32 microcontroller can be used in many different industries, including sleep apnea. There are many applications for ESP32 in sleep apnea that can enhance patient treatment. Monitoring important parameters including heart rate, oxygen saturation, and respiratory rate is one of the key applications of ESP32 in sleep apnea. This can be accomplished by gathering information from sensors positioned on the patient's body and sending it to a monitoring system for evaluation. Additionally, ESP32 is capable of identifying sleep apnea events including breathing pauses and snoring, which can then set off an alert or notify the patient or caretaker.

Additionally, ESP32 have the capacity to control CPAP machines, which are frequently used to treat sleep apnea. Depending on the requirements of the patient, it can change the air pressure and flow rate. Additionally, ESP32 is capable of tracking a

patient's sleep habits, including their sleep cycles and the duration of each stage of sleep. The usage of this data can enhance treatment strategies and raise sleeping standards. Finally, ESP32 allows users to provide feedback on their experiences with sleep apnea symptoms and therapy. These comments can be utilized to enhance treatment plans and deliver more individualized care. Overall, ESP32 can enhance patient care and treatment results for sleep apnea and has a wide variety of possible applications.

SPECIFICATIONS

- CPU: Dual-core Tensilica LX6 240 MHz
- RAM: 520 KB SRAM
- Wi-Fi: 802.11 b/g/n/e/i
- Bluetooth: v4.2 BR/EDR and BLE
- GPIO: 34 pins
- ADC: 12-bit SAR ADC up to 18 channels
- DAC: 2 x 8-bit DACs
- Interfaces: UART, SPI, I2C, I2S, CAN, Ethernet, SD/MMC, JTAG
- Operating Voltage: 2.2V to 3.6V
- Dimensions: 18mm x 25.5mm

TYPES

- ESP32-WROOM-32
- ESP32-SOLO-1
- ESP32-PICO-D4
- ESP32-WROVER
- ESP32-CAM

ADVANTAGES

- *Low power consumption:* The ESP32 offers a low power consumption mode that can be utilised to improve battery life, which is essential for wearable technology that must be worn for lengthy periods of time.

- *Wireless connectivity:* The ESP32 includes integrated Wi-Fi and Bluetooth capabilities that can be used to transfer data to a remote monitoring system or a healthcare professional. This can make it simpler for patients to get prompt feedback and for healthcare professionals to keep track of patients' development.
- *Real-time data processing:* The ESP32 has a quick processor and is capable of real-time data processing, which is essential for sleep apnea monitoring. This makes it possible for the patient or healthcare professional to receive feedback right away, which can help to avert significant consequences.
- *Low cost:* The ESP32 is comparatively cheap in comparison to other microcontrollers, making it a viable alternative for patients and healthcare professionals.

Overall, the ESP32 has a number of benefits that make it a potential solution for the detection and treatment of sleep apnea. It is a viable solution for both patients and healthcare providers because to its low power consumption, wireless connectivity, real-time data processing, and inexpensive cost.

4.4.3 DHT11SENSOR

The DHT11 sensor is a versatile, low-cost digital temperature and humidity sensor shown in fig. 4.6. Because of its low price, simplicity, and dependability, it is a well-liked option among makers and beginners.

The DHT11 sensor measures temperature and humidity using a thermistor, and vice versa. Its single-wire serial interface makes it simple to connect to microcontrollers like the Arduino, Raspberry Pi, or ESP8266. Using a unique communication protocol that provides data in the form of digital pulses, the sensor connects with the microcontroller.

The DHT11 sensor provides a 20% to 90% relative humidity detection range with an accuracy of 5% RH. With a 2°C accuracy, the temperature measuring range is 0°C to 50°C. For humidity, the sensor responds in around 2 seconds, and for temperature, it takes about 5 seconds.

The affordability of the DHT11 sensor is one of its key benefits. One of the most reasonably priced temperature and humidity sensors on the market is this one. The sensor is also very simple to use and doesn't need any extra parts to connect to a microcontroller.

The DHT11 sensor does have certain restrictions, though. Compared to other sensors on the market, it has a lesser accuracy. Additionally, it is not advised for use in high-precision applications.



Fig. 4.6: DHT11SENSOR

In many applications, including the monitoring of sleep apnea, the DHT11 sensor is a common choice for measuring temperature and humidity. The DHT11 sensor can be used to track the patient's breathing in sleep apnea because variations in humidity and temperature can be signs of breathing problems.

Sleep apnea, a disorder that causes brief breathing pauses or shallow breathing while sleeping, can interfere with a person's normal breathing patterns. This may result in lower blood oxygen levels and other severe health issues. The DHT11 sensor can assist in identifying these breathing disturbances by tracking temperature and humidity levels, alerting the patient or healthcare professional to take the necessary action.

In order to assess the temperature and humidity of the exhaled air, the DHT11 sensor can be integrated into a sleep apnea monitoring system by positioning it close to the patient's mouth or nose. These readings can be wirelessly communicated to a microcontroller, like an Arduino or an ESP32, which can evaluate the data and, if necessary, send alarms to the patient or healthcare professional.

The DHT11 sensor can be used to track breathing patterns as well as the environment during sleep. The sensor can assist in identifying elements, such as dry

air or high temperatures, that may contribute to sleep apnea by analyzing the temperature and humidity levels in the bedroom.

The DHT11 sensor can be a beneficial tool for tracking sleep apnea and enhancing patient outcomes, all things considered. Both patients and healthcare practitioners find it to be a desirable option because to its low cost, simplicity of use, and dependability. To provide a complete picture of the patient's sleep health, it is crucial to keep in mind that the DHT11 sensor has some limits in terms of accuracy and precision and should be used in conjunction with other monitoring equipment.

SPECIFICATIONS

- Operating Voltage: 3.5V to 5.5V
- Humidity Measurement Range: 20% to 90% relative humidity with an accuracy of 5%
- Range of Temperature Measurement: 0°C to 50°C with 2°C Accuracy
- Response Time: 2 seconds for humidity, 5 seconds for temperature
- Output: Digital data via a single-wire serial interface
- Sampling Rate: One measurement per second
- Power Consumption: 1mA during active measurement, 50A during standby
- Dimensions: 12 mm x 15.5 mm x 5.5 mm

TYPES

DHT11 3-pin: The most popular DHT11 sensor is this one. Three pins are present: VCC, GND, and DATA. The power supply is connected to the VCC pin, the ground is connected to the GND pin, and the DATA pin is connected to the microcontroller's input/output (I/O) pin. A tiny plastic cover surrounds the sensor, shielding it from the elements. This kind of sensor is simple to use and appropriate for the majority of applications.

DHT11 4-pin: The four pins on this particular DHT11 sensor are VCC, GND, DATA, and NC (not connected). The DATA pin is connected to the microcontroller's I/O pin, while the VCC and GND pins are connected to the power supply and ground,

respectively. The NC pin is not connected because it is not in use. In PCB designs, this kind of sensor is frequently employed where the extra pin is required for stability or other functions.

DHT11 bare: A chip without a casing or packaging is referred to as a DHT11 sensor. It can be easily integrated into other devices and is appropriate for situations where size is a consideration. However, it requires extra precautions to prevent damage because it is more vulnerable to external elements like humidity and temperature.

ADVANTAGES

- Not intrusive
- Effective in terms of cost
- Accurate
- Real-time observation
- Simple integration
- reduced power use

4.4.4 RESPIRATION SENSOR

The sound sensor module offers a simple method for hearing and is typically used to measure sound intensity. Applications involving security, switches, and monitoring can all use this module. For ease of use, its precision may be quickly modified.

It makes use of a microphone as the input source for an amplifier, peak detector, and buffer. The sensor processes an output signal voltage that is transmitted to a microcontroller and then completes the necessary processing when it detects sound.

The Arduino sound detection sensor module determines if sound levels have risen above a certain level. A microphone picks up sound, which is then sent into an LM393 op amp. An on-board potentiometer is used to change the sound volume set point. An LED on the module illuminates and the output is reduced when the sound level exceeds the specified point.



Fig. 4.7: RESPIRATION SENSOR

In sleep apnea monitoring systems, sound detection sensors are a type of sensor that can be utilized as a respiration sensor shown in fig. 4.7. Since snoring is a typical sign of sleep apnea and is a result of vibrations in the upper airway during breathing, sound detecting sensors are used to monitor sleep apnea. The patient's breathing patterns during sleep can be inferred from the vibrations that sound detection sensors pick up on.

For the purpose of monitoring sleep apnea, a variety of sound detection devices, including piezoelectric sensors, MEMS microphones, and acoustic sensors, can be utilized. The vibration of the larynx during breathing can be picked up by piezoelectric sensors because they are sensitive to pressure fluctuations. MEMS microphones are inexpensive, tiny sensors that are simple to connect with other electronics like microcontrollers or smartphones. Acoustic sensors are capable of detecting breathing noises and are sensitive to pressure variations.

In order to monitor sleep apnea, a sound detection sensor is used as a respiration sensor that is put close to the patient's nose or throat. The sensor picks up the vibrations or sounds that come from breathing, and the data is then analyzed to identify the patient's breathing patterns, including rate and depth.

Sound detection sensors are non-invasive and do not require contact with the patient's skin or bodily fluids, making them pleasant and safe for the patient. This is one benefit of employing sound detection sensors as respiration sensors for sleep apnea monitoring. Another benefit is the affordability and simplicity of integrating sound detection sensors with other electronics like microcontrollers or cell phones. But not all individuals may benefit from sound detection devices, particularly those who don't snore or exhibit other sleep apnea symptoms. Other sensor types, including pressure sensors or breathing belts, may be more suitable in certain circumstances.

In conclusion, sound detection sensors can be utilized in sleep apnea monitoring systems as a respiration sensor to identify breathing patterns based on snoring sounds. They may not be appropriate for all people, but they do have benefits like non-invasiveness, cost, and ease of integration.

SPECIFICATIONS

- Working voltage: 3.3-5 volts DC
- Flexibility in Sensitivity
- 32 x 17 mm Signal output indication Dimensions
- Single-channel signal output
- With the retaining bolt hole, simple installation
- when there's sound, it generates low-level output and the signal light.
- Digital switching output (0 and 1 high and low) is the output format.

ADVANTAGES

- Non-invasiveness
- Real-time observation
- accurate observation
- Customization
- Arrangement with additional equipment

4.4.5 MAX30100

The MAX30100 is a highly integrated sensor module for monitoring heart rate and pulse oximetry shown in fig. 4.8. One compact module house two LED drivers, a photodetector, and signal-processing circuitry. In many medical, fitness, and wearable devices, the MAX30100 sensor module is used to track heart rate and blood oxygen saturation levels.

Two LED drivers are used by the MAX30100 sensor module, one for red light and the other for infrared light. Deoxygenated haemoglobin absorbs infrared light while oxygenated haemoglobin absorbs red light. The MAX30100 can determine the oxygen saturation level in the patient's blood by observing the variation in absorption between the two light wavelengths.

The MAX30100 can monitor blood oxygen saturation as well as heart rate by identifying the pulsatile component of blood flow. Changes in the amount of light passing through the patient's skin are detected by the sensor module using a photodetector. A pulsatile signal that can be utilised to determine the patient's heart rate is produced as blood flows through blood vessels and alters the amount of light that reaches the photodetector.



Fig. 4.8: MAX30100

A built-in ambient light rejection algorithm in the MAX30100 sensor module reduces noise from sources of ambient light, such as sunlight or fluorescent lights. In order to enhance the signal-to-noise ratio and boost measurement accuracy, it also contains an adaptive gain control method.

The MAX30100 sensor module has a number of benefits over conventional pulse oximeters, including its compact size, low power requirement, and simplicity of device integration. The module can provide continuous monitoring of heart rate and blood oxygen saturation levels and is simple to connect into wearable gadgets like smartwatches or fitness trackers.

The MAX30100 sensor module is a highly integrated sensor that may be used to monitor blood oxygen saturation levels and heart rate during sleep apnea. A common sleep disease called sleep apnea involves breathing pauses or shallow breathing while you're asleep, which lowers the amount of oxygen in your blood. These variations in oxygen levels can be recognised by the MAX30100 sensor module, which can also notify medical personnel of potential breathing issues.

Two LED drivers, one for red light and the other for infrared light, are used by the MAX30100 sensor module. Deoxygenated haemoglobin absorbs infrared light while oxygenated haemoglobin absorbs red light. The MAX30100 can determine the oxygen saturation level in the patient's blood by observing the variation in absorption between the two light wavelengths. A drop in blood oxygen levels in people with sleep

apnea may be a sign that they are having a breathing episode.

The MAX30100 can detect the pulsatile component of blood flow and measure heart rate in addition to blood oxygen saturation monitoring. The amount of light that reaches the photodetector changes as blood flows through the blood vessels, resulting in a pulsatile signal that can be used to determine the patient's heart rate. The patient may be having a respiratory episode if there are changes in heart rate.

A built-in ambient light rejection algorithm in the MAX30100 sensor module reduces noise from sources of ambient light, such as sunlight or fluorescent lights. In order to enhance the signal-to-noise ratio and boost measurement accuracy, it also contains an adaptive gain control method. These algorithms aid in ensuring that, despite external noise, the sensor module delivers precise measurements.

Overall, the MAX30100 sensor module is a useful device for tracking sleep apnea. The device's capacity to assess blood oxygen saturation and heart rate enables medical personnel to spot impending breathing episodes and modify treatment regimens accordingly.

It is a desirable alternative for continuous monitoring of vital signs while you sleep due to its tiny size, low battery consumption, and simplicity of interaction with other devices.

SPECIFICATIONS

- Operating Voltage: 1.8V to 5.5V
- Supply Current: 600 μ A
- LED Driver Output Voltage: 1.7V to 3.3V
- LED Current Range: 0mA to 50mA
- Photodetector Sensitivity: 220 μ A/W
- Sampling Rate: Up to 3200 samples per second
- LED Wavelengths: Red: 660nm; Infrared: 880nm
- ADC Resolution: 18-bit
- Output Data Format: I2C-Compatible
- Operating Temperature Range: -40°C to +85°C
- Package Dimensions: 5mm x 4mm x 1.2mm

TYPES

- Reliable Monitoring
- Constant Watching
- Non-Invasive
- Power Usage very Low
- Easy Integration, Small Size Ambient Light Rejection

4.4.6 ACCELEROMETER SENSOR

A specific kind of sensor called an accelerometer is shown in fig. 4.9 detects an object's acceleration or the rate at which its velocity changes. Automotive, aerospace, and consumer electronics are just a few of the industries that use accelerometers. In addition, they are utilized in healthcare and medical settings for monitoring sleep apnea.

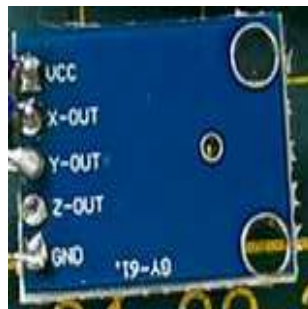


Fig. 4.9: ACCELEROMETER SENSOR

An accelerometer sensor's fundamental operating mechanism is based on the piezoelectric effect or microelectromechanical systems (MEMS) technology. In most cases, an accelerometer is made up of a mass attached to a spring system, with one end of the spring fixed and the other coupled to the mass. The mass moves in relation to the fixed end of the spring when the accelerometer encounters acceleration, compressing or stretching the spring. In order to calculate the acceleration, a measurement of the capacitance of the MEMS sensor or a voltage across the piezoelectric material caused by the mechanical stress can be made.

One, two or three dimensions of acceleration can be measured by accelerometer sensors. While two-dimensional accelerometers record acceleration on two perpendicular axes, one-dimensional accelerometers only measure acceleration along a single axis. Accelerometers that measure acceleration in all three dimensions

are referred to as three-dimensional or three-axis accelerometers.

Accelerometer sensors can be used to track body movements and breathing effort when monitoring sleep apnea. An accelerometer can be used to measure variations in acceleration brought on by the movement of the chest or abdomen while breathing. This allows for the detection of respiratory effort. Monitoring changes in acceleration can also be used to identify body movements like turning over or getting out of bed.

Low noise floors and great sensitivity to minute changes in acceleration are common characteristics of accelerometer sensors used for sleep apnea monitoring. They can be set up to filter out higher frequency noise from sources like ambient vibration, which is often designed to function at low frequencies like the range of human breathing and body motions.

In order to provide a complete picture of sleep apnea, accelerometer sensors can also be used in concert with other sensors, such as the sound detection sensor and the MAX30100 sensor. For instance, the MAX30100 sensor can track oxygen saturation levels while the sound detection sensor may hear snoring or other respiratory noises. This knowledge along with accelerometer data can be used to determine the type and severity of sleep apnea, as well as the effectiveness of various treatment methods.

Overall, accelerometer sensors are a crucial piece of equipment for tracking sleep apnea. They can gain important insights on the severity of sleep apnea and the efficacy of treatment choices due to their ability to detect breathing effort and body movements when they are asleep.

SPECIFICATIONS

- Measurement range: ± 2 g or ± 16 g
- Sensitivity: 10–100 pF/g or 0.1–1 mV/g
- Frequency response: low-pass filter with a cut-off frequency in the range of 0.5 to 20 Hz
- Noise level: less than 1 mg/ $\sqrt{\text{Hz}}$
- Output format: Analog voltage or a digital signal (I2C or SPI)

- Size & package: Dependent on the exact sensor model, but typical ranges are 3 x 3 mm to 10 x 10 mm, with surface-mount or through-hole packaging being the most frequent.

ADVANTAGES

- Not intrusive
- Real-time observation
- Precision Portability
- Effective in terms of cost
- Simple to use

4.4.7 COMPRESSOR

A compressor is a mechanical device is shown in fig. 4.10 that raises the pressure of a gas or vapour in order to decrease the volume of the gas or vapour. It is frequently utilised in industrial and commercial contexts for a variety of purposes, including gas compression, air conditioning, refrigeration, and the operation of pneumatic tools.

Positive displacement and dynamic compressors are the two basic types of compressors. In order to compress the gas, positive displacement compressors trap a predetermined volume of gas in a chamber and then reduce the chamber's capacity. Reciprocal compressors, rotary screw compressors, and scroll compressors are a few types of positive displacement compressors.

On the other side, dynamic compressors function by giving a gas velocity, which is subsequently transformed into pressure via diffusers and nozzles. Centrifugal and axial compressors are two types of dynamic compressors.

The particular application and needs determine the type and size of compressor to use. For instance, a centrifugal compressor may be better suited for large-volume, low-pressure applications as opposed to a reciprocating compressor for low volume, high pressure applications.

Efficiency, reliability, maintenance needs, and cost are all things to keep in mind when choosing a compressor in addition to compressor type. To guarantee optimum performance and longevity, a compressor must be installed, used, and maintained

properly.

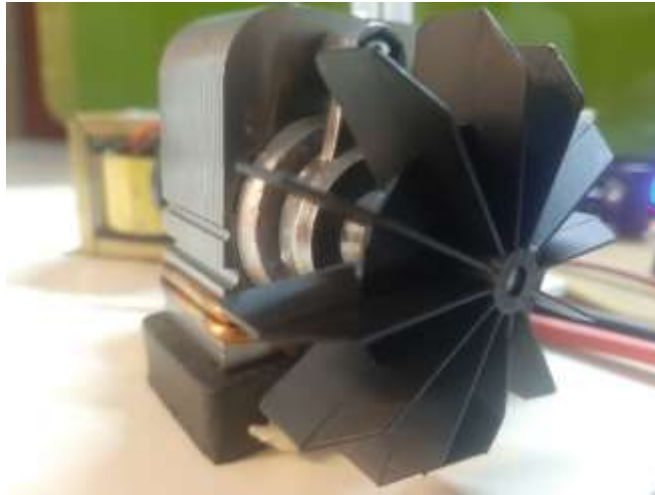


Fig. 4.10: Compressor

Compressors enable the effective compression of gases and vapours for a variety of functions, making them crucial parts in many industrial and commercial applications.

A mask that the patient wears over their face. By keeping the airway open throughout sleep, this air flow helps to avoid breathing pauses. The compressor is an important component of sleep apnea equipment because it creates the steady stream of air that is necessary to keep the airway open. Intake air from the room is passed through a filter by the compressor, which gets rid of any contaminants or allergies that could be there.

Following that, the air gets pressurized to a fixed level according to the patient's recommended pressure setting. It is extremely important to remember that a doctor normally prescribes the CPAP machine's pressure setting according to the level of severity of the patient's sleep apnea. To make sure the patient is getting the right amount of pressure to maintain their airway during sleep, the pressure setting can be changed.

In conclusion, the compressor in a sleep apnea machine is in charge of producing a constant flow of pressurized air that aids in maintaining the airway open while you sleep. This component is vital to the CPAP therapy's efficiency and is important in assisting individuals with sleep apnea in getting uninterrupted, deep sleep

during the night. A mask that the patient wears over their face. By keeping the airway open throughout sleep, this air flow helps to avoid breathing pauses.

The compressor is an important component of sleep apnea equipment because it creates the steady stream of air that is necessary to keep the airway open. Intake air from the room is passed through a filter by the compressor, which gets rid of any contaminants or allergies that could be there. Following that, the air gets pressurized to a fixed level according to the patient's recommended pressure setting. It is extremely important to remember that a doctor normally prescribes the CPAP machine's pressure setting according to the level of severity of the patient's sleep apnea. To make sure the patient is getting the right amount of pressure to maintain their airway during sleep, the pressure setting can be changed.

In conclusion, the compressor in a sleep apnea machine is in charge of producing a constant flow of pressurized air that aids in maintaining the airway open while you sleep. This component is vital to the CPAP therapy's efficiency and is important in assisting individuals with sleep apnea in getting uninterrupted, deep sleep during the night.

4.4.8 POWER SUPPLY

4.4.8.1 BLOCK DIAGRAM

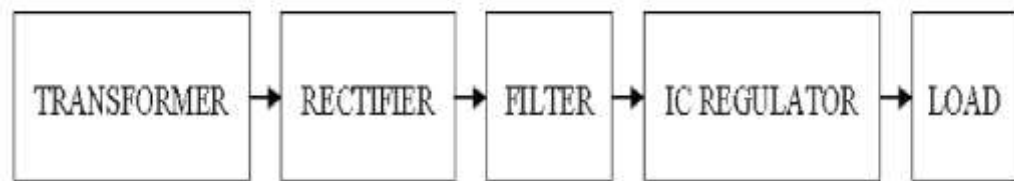


Fig. 4.11: Block Diagram

The block diagram of a power supply circuit is shown in fig. 4.11. A transformer that transforms high voltage AC current into low voltage AC current, then a bridge rectifier that changes AC to DC. Any leftover AC ripple is eliminated by the filter, which results in a more consistent DC output. The final stage is an IC voltage regulator, which controls the output voltage in order to ensure that downstream components receive steady and consistent power for the correct operation of electronic devices.

4.4.8.2 CIRCUIT DIAGRAM:

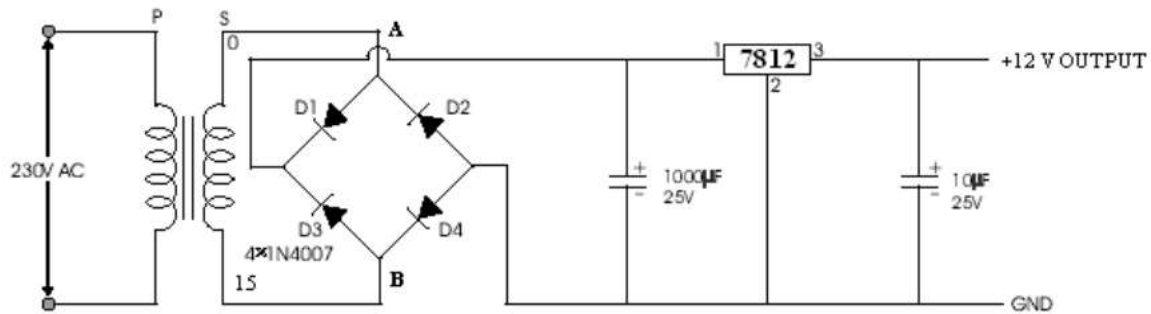


Fig. 4.12: Circuit Diagram

The circuit diagram fig. 4.12 shows various parts sequentially arranged to form a reliable and effective power supply system. A transformer, which is essential for converting high-voltage AC current into low-voltage AC current, the bridge rectifier, which converts the AC current into the DC current, is connected to the transformer.

After being rectified, the filtered DC current goes through a filter to remove any lingering AC ripple, which further stabilizes the current. Then an IC voltage regulator is shown, which keeps the output voltage constant and supplies steady, consistent power to the components further down.

4.4.8.3 WORKING PRINCIPLE

TRANSFORMER

Power supply voltage (0-230V) will be reduced to (0-15V) level using the potential transformer. The secondary coil's voltage will decrease and the current, or AMPS, will increase or decrease depending on the wire gauge if the secondary has fewer turns in the coil than the primary is known as a STEP-DOWN transformer. The rectifier will then be linked to the secondary of the potential transformer.

BRIDGE RECTIFIER

The circuit is known as a bridge rectifier is shown in fig. 4.13 when four diodes are linked as in the figure. The network's diagonally opposed corners serve as the circuit's input, and the remaining two corners serve as the network's output. Assume

that the transformer is in good working order and that points A and B have positive and negative potentials, respectively. With point A's positive potential, D3 will be skewed forward, and D4 will be skewed backwards.

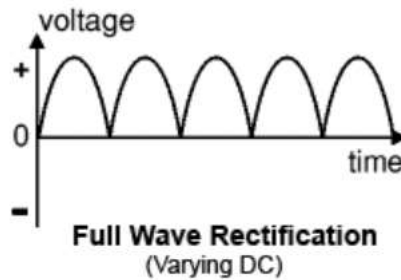


Fig. 4.13: Bridge Rectifier

Point B's negative potential will cause D1 and D2 to move in the opposite direction. D3 and D1 are currently forward-biased, allowing current to flow through them; D4 and D2 are reverse-biased, preventing current flow.

From point B, current flows through D1, up through the load, through D3, through the transformer secondary, and back to point B. A half cycle later, the polarity across the transformer's secondary reverses, driving D2 and D4 in one direction while driving D1 and D3 in the opposite direction. The direction of current flow will now be through D4, up through Load, through D2, through the transformer secondary, and then back to point A. between D2 and D4. The direction of the current flow through Load is constant. This current generates a voltage that corresponds to it as it passes through the load. This bridge rectifier is a full-wave rectifier because current flows through the load during both half cycles of the applied voltage.

With a given transformer, a bridge rectifier delivers an output voltage that is approximately double that of a standard half-wave circuit, which is one advantage over a traditional full-wave rectifier.

Due to the diode, this bridge rectifier consistently loses 1.4 Volts of the input voltage. The cut-off region of the 1N4007 PN junction diode, which we are employing, is 0.7 Volt.

The total drop voltage is 1.4 volt and any two diodes are therefore always conducting.

FILTER

The output of a rectifier is shown in fig. 4.14 will be changed into a more stable DC voltage if a capacitor is added in parallel with the load resistor of the rectifier to create a simple filter circuit. The capacitor is initially charged to the rectified Waveform's peak value. Up until the rectified voltage surpasses the capacitor voltage, the capacitor is discharged through the load resistor after the peak. The process is then repeated after charging the capacitor once more.

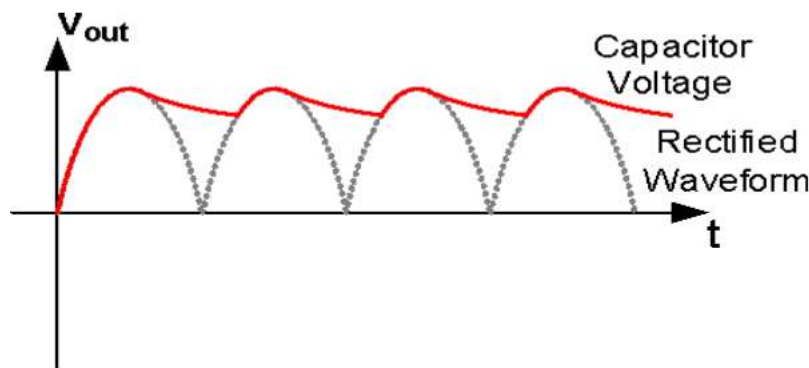


Fig. 4.14: Output of a rectifier

IC VOLTAGE REGULATORS

A class of frequently used ICs includes voltage regulators. The circuitry for the reference source, comparator amplifier, control device, and overload protection are all combined in a single IC in regulator IC units. A fixed positive voltage, a fixed negative voltage, or an adjustable set voltage can all be regulated by IC devices.

A fixed three-terminal voltage regulator features a grounded second terminal, an unregulated dc input voltage applied to one input terminal, and a regulated dc output voltage coming from a third terminal.

Voltages between 5 and 24 volts are fixed positively regulated by the series 78 regulators. The series 79 regulators work similarly, offering set negative regulated voltages between 5 and 24 volts. The 78xx IC series is utilised in this circuit to provide regulated power. These regulators have a fixed voltage range and a current capacity of 1 to 1.5 A. 5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, and 24V are the most often used regulated voltages. To enhance the regulation, it is critical to place capacitors across

the regulator IC's input and output.

The 7812 regulators are used in this circuit to convert fluctuating dc into a constant positive 5V power supply. The output will vary if the input voltage drops below 14.6 volts. 1)Input, 2) Ground, 3) Output is shown in fig. 4.15. We are utilizing a 230/15V step-down transformer because of this. Fig.4.16 shows the regulator's minimum level input is higher than the transformer's output.

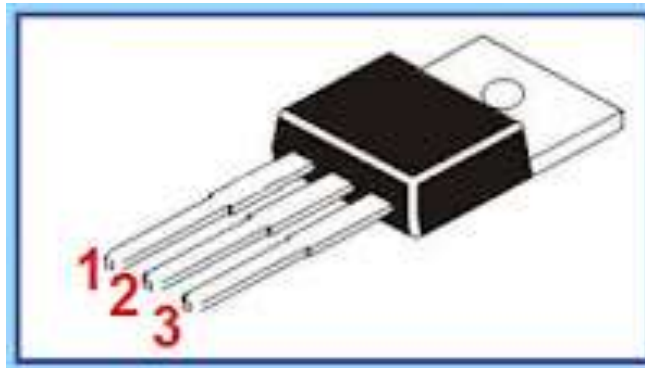


Fig. 4.15: 7812 regulators

Type	Min Input Voltage	Output Voltage
7805	7V	+5V
7806	8V	+6V
7808	10V	+8V
7809	11V	+9V
7812	15V	+12V
7815	18V	+15V
7818	22V	+18V
7824	30V	+24V

Fig. 4.16: Types of regulators

4.5 SOFTWARE REQUIREMENTS

4.5.1 ARDUINO IDE

Arduino Integrated Development Environment (IDE) v1 Learn how the Arduino IDE v1 works, such as compiling & uploading sketches, file management, installing dependencies and much more.

The Arduino Integrated Development Environment – or Arduino Software (IDE) – contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Writing Sketches

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

IDE VERIFY File

Verify Checks your code for errors compiling it.

IDE UPLOAD File

Upload Compiles your code and uploads it to the configured board.

IDE NEW File

New Creates a new sketch.

IDE OPEN File

Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

IDE SAVE File

Save Saves your sketch.

IDE SERMON File

Serial Monitor Opens the serial monitor.

File

- **New** Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- **Open** Allows to load a sketch file browsing through the computer drives and folders.
- **Open Recent** Provides a short list of the most recent sketches, ready to be opened.
- **Sketchbook** Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- **Close** Closes the instance of the Arduino Software from which it is clicked.
- **Save** Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as." window.
- **Save as...** Allows to save the current sketch with a different name
- **Page Setup** It shows the Page Setup window for printing.
- **Print** Sends the current sketch to the printer according to the settings defined in Page Setup.
- **Preferences** Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- **Quit** Closes all IDE windows. The same sketches open when quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- **Undo/Redo** Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- **Cut** Removes the selected text from the editor and places it into the clipboard.
- **Copy** Duplicates the selected text in the editor and places it into the clipboard.
- **Copy for Forum** Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax colouring.
- **Copy as HTML** Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- **Paste** Puts the contents of the clipboard at the cursor position, in the editor.

- Select All Selects and highlights the whole content of the editor.
- Comment/Uncomment Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- Verify/Compile Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder Opens the current sketch folder.
- Include Library Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

- **Add File...** Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the data subfolder of the sketch, which is intended for assets such as documentation. The contents of the data folder are not compiled, so they do not become part of the sketch program.

Tools

- **Indents** it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- **Archive Sketch** Archives a copy of the current sketch in zip format. The archive is placed in the same directory as the sketch.
- **Fix Encoding & Reload** Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- **Serial Monitor** Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- **Board** Select the board that you're using. See below for descriptions of the various boards.
- **Port** This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- **Programmer** For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- **Burn Bootloader** The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

Help

Find in Reference This is the only interactive function of the Help menu: it

directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Before compiling the sketch, all the normal Arduino code files of the sketch (.ino, .pde) are concatenated into a single file following the order the tabs are shown in. The other file types are left as is.

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an UNO or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port

and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the Rx and Tx LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e., when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g., working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or

you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

Serial Monitor

This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to `Serial.begin` in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor +does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

CHAPTER – 5

RESULTS AND DISCUSSION

The hardware setup of Sleep apnea is shown in fig. 5.1. The sensors were able to gather information on a variety of factors, including heart rate, temperature, mobility, and respiration rate, which were then shown in real-time on the LCD display fig. 5.2. The patient received continuous positive airway pressure (CPAP) from the compressor, which helped maintain their airways open and enhanced breathing as they slept. The ThingSpeak app was able to gather and process the sensor data, creating a graphical depiction of the patient's sleep metrics and patterns. This made it possible for doctors to keep track of the patient's development and modify their treatment plans as necessary.

Patients may select between automated and manual control of the CPAP equipment thanks to the usage of an automatic and manual mode switch, which allowed for flexibility in the management of sleep apnea. The patient's general health and well-being have been improved by this personalized approach to sleep apnea care, which assisted in the detection and prevention of sleep apnea.



Fig. 5.1: Hardware setup



Fig. 5.2: LCD Output



Fig. 5.3: Graphical representation of Humidity

The graphical representation of humidity is shown in fig. 5.3 The correlation between humidity and time for patients, with the y-axis representing a normal range for humidity and the x-axis representing time and date. A continuous line on the graph depicts the changes in humidity levels throughout time. The humidity levels for patients, people who need respiratory support, or those who are recuperating from respiratory illnesses are frequently monitored and managed using this sort of graph. Therefore, the graphical representation is useful for overseeing patient care and making sure that the humidity levels are ideal for better respiratory function and comfort.

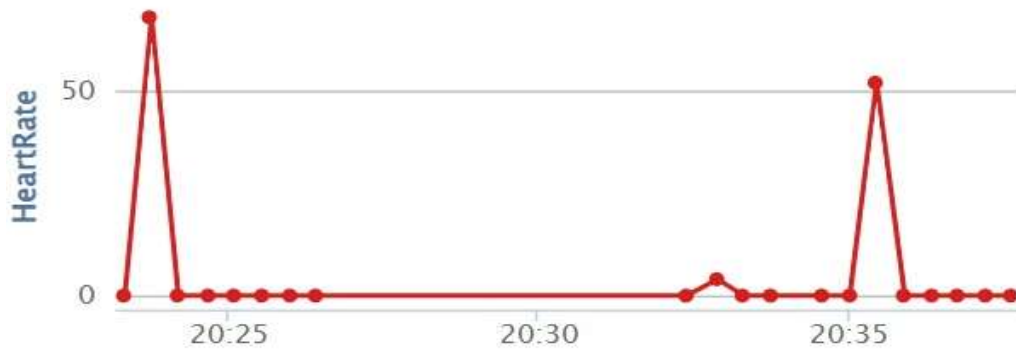


Fig. 5.4: Graphical representation of Heart Rate

The graphical representation of heart rate is shown in fig. 5.4. A normal heart rate range is shown on the y-axis of the graph, The x-axis displays the current time and date. The heart rate variations throughout time are represented by a continuous line on the graph. This kind of graph is utilized to track and monitor the patient's heart rate.

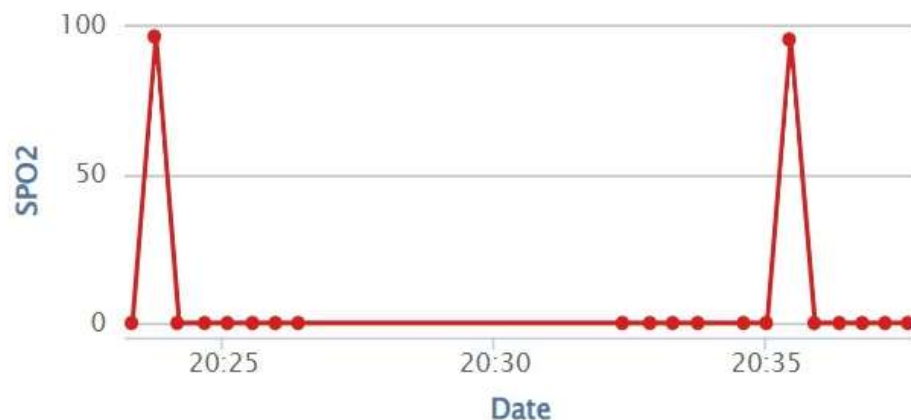


Fig. 5.5: Graphical representation of SpO2

The graphical representation of SpO2 is shown in fig. 5.5. A normal range for SpO2 is shown on the y-axis, while the x-axis represents time and date. The graph illustrates the correlation between time and SpO2 (blood oxygen saturation) for patients. A continuous line that shows the changes in SpO2 levels over time may be seen on the graph. The oxygen levels of patients, people who are getting respiratory support, or those who have respiratory diseases are tracked and managed using this graph. Therefore, this kind of graphical representation is helpful in monitoring patient

care and making sure that oxygen levels are at their ideal levels for better respiratory function and general health.

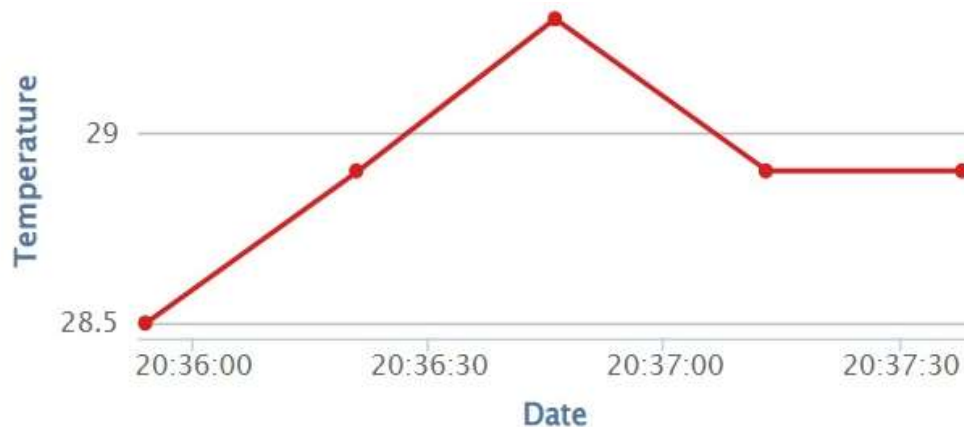


Fig. 5.6: Graphical representation of temperature

The graphical representation of temperature is shown in fig. 5.6. A typical temperature range is shown on the y-axis, and the link between time and temperature for patients with sleep apnea is depicted graphically on the x-axis. The graph shows an uninterrupted line that depicts how the body's temperature changes over time. Patients body temperatures are monitored and the quality of their sleep is evaluated using this graph.

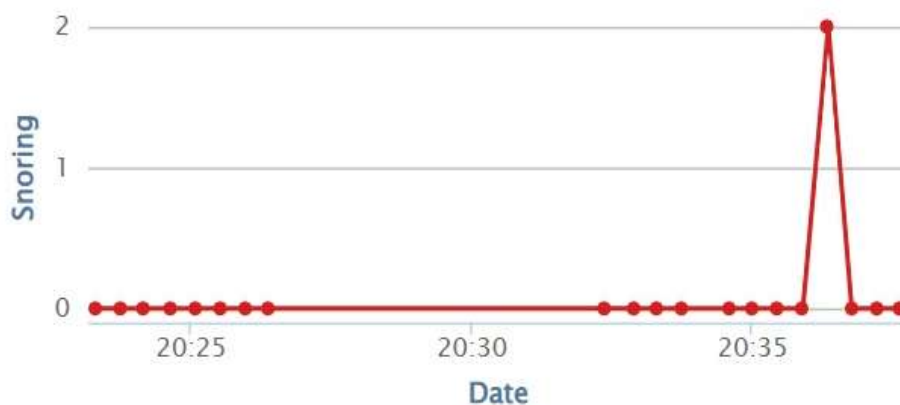


Fig. 5.7: Graphical representation of Snoring

The graphical representation of snoring is shown in fig. 5.7. A line graph with the x-axis denoting the time and date and the y-axis denoting the typical range for snoring in patients with sleep apnea is displayed in the graphical representation. The center of the normal range is indicated by a horizontal line that runs across the chart's various

parts. Two parallel lines above and below the midline show the upper and lower limits of the typical range. Each data point is depicted by a dot connected by a line to make a continuous curve on the line graph, which shows the frequency of patients' snoring with time. The curve fluctuates up and down within the usual range, showing fluctuations in snoring intensity. The visual summary of the snoring levels of patients with sleep apnea throughout time provided by the graphical depiction allows for simple interpretation.

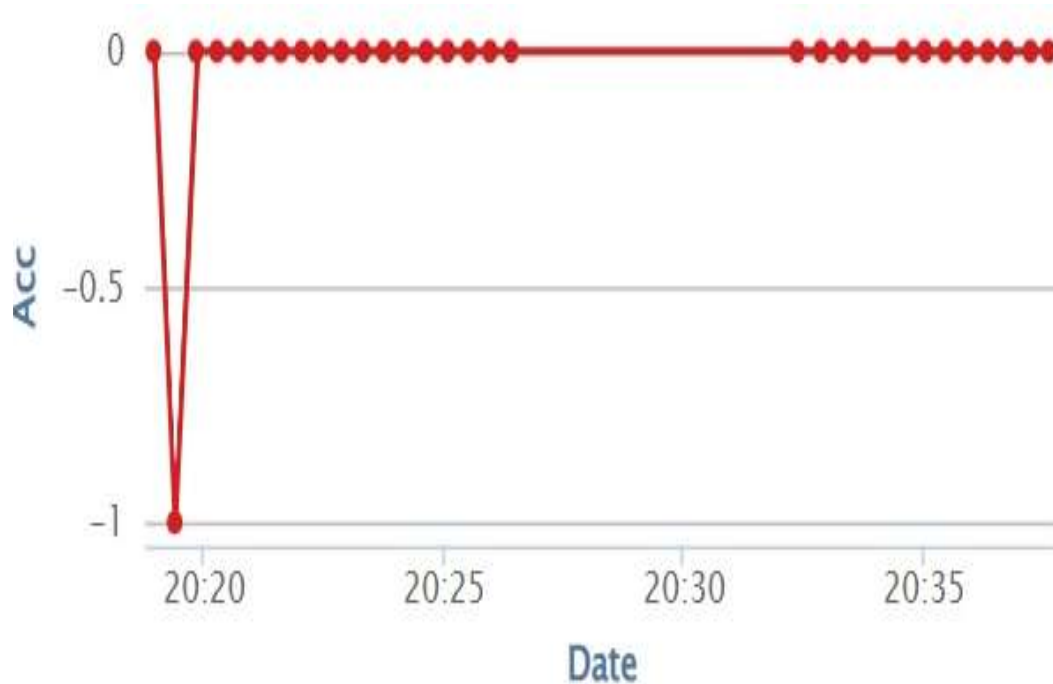


Fig. 5.8: Graphical representation of acceleration

The graphical representation of acceleration is shown in fig. 5.8. While the x-axis displays the time and date, the y-axis displays an acceleration's normal range. A continuous line on the graph depicts how the acceleration changes over time. This type of graph is frequently used to track patients' movements while they sleep and evaluate the quality of their sleep. Healthcare professionals regularly utilize this type of graphical representation to track patient care and assess how well sleep apnea treatments are working.



Fig. 5.9: Graphical representation of Compressor

The graphical representation of compressor is shown in fig. 5.9. The y-axis shows the compressor reading, while the x-axis shows the time and date. The graph shows a straight line that depicts the changes in the compressor reading over time. This kind of graph is frequently used in sleep studies to check the efficiency of a sleep apnea machine and make sure it is operating properly.

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Sleep apnea is a serious disorder that can negatively affect a patient's health and quality of life. To avoid problems and enhance the patient's quality of life, it is critical to adequately recognize and manage the illness. Continuous monitoring of multiple parameters, including heart rate, temperature, acceleration, and respiration rate, is provided by this device. The severity of the problem can be determined using this comprehensive method, and personal treatment programs can be developed to properly control the symptoms. It has been demonstrated that using a CPAP machine may reduce the severity of sleep apnea symptoms, including snoring and daytime tiredness, and enhance the patient's quality of life. A complete and efficient method of managing sleep apnea is made possible by the use of a device that includes a compressor, a number of sensors, and the ThingSpeak app. The tool can considerably enhance the patient's general health and well-being by identifying and preventing sleep apnea.

6.2 FUTURE SCOPE

While the studies mentioned above have shown high accuracy in detecting sleep apnea events, there is always room for improvement. Future research can focus on improving the accuracy of the monitoring system by incorporating additional sensors or using advanced algorithms. Sleep apnea is a complex disorder that affects individuals differently. Future research can focus on using data from the monitoring system to personalize treatment for each patient. This can involve adjusting the pressure settings of the CPAP machine or altering the sleeping position to improve breathing. Real-time feedback can be valuable for patients with sleep apnea. Future research can focus on developing a monitoring system that provides real-time feedback to the patient, such as alerts when a sleep apnea event occurs or suggestions for adjusting the treatment. Sleep apnea is often associated with other health conditions such as obesity, hypertension, and diabetes. Future research can focus on integrating the sleep apnea monitoring system with other health technologies to provide a comprehensive health

monitoring system for patients. Telemedicine is becoming increasingly popular, especially in the wake of the COVID-19 pandemic. Future research can focus on developing a sleep apnea monitoring system that can be remotely accessed by healthcare providers to provide remote monitoring and management of sleep apnea patients.

REFERENCES

1. Abbasi, Anna, Gupta, Sushilkumar S., Sabharwal, Nitin, Meghrajani, Vineet, Sharma, Shaurya, Kamholz, Stephan, and Yizhak Kupfer. "A comprehensive review of obstructive sleep apnea." *Sleep Science* 14, no. 2 (2021): 142-154. <https://doi.org/10.5935/1984-0063.20200056>.
2. Al-Mardini, Mamoun, Fadi Aloul, Assim Sagahyroon, and Luai Al-Husseini. "Classifying obstructive sleep apnea using smartphones." *Journal of biomedical informatics* 52 (2014): 251-25
3. Badr, M. S., Dingell, John D., and Shahrokh Javaheri. "Central Sleep Apnea: a Brief Review." *Current pulmonology reports* 8, no. 1 (2019): 14.
4. BaHammam, Ahmed S. "Treatment-emergent central sleep apnea (complex sleep apnea)." *Sleep and Vigilance* 1, no. 2 (2017): 53-56.
5. Chang, Hong-Po, Yu-Feng Chen, and Je-Kang Du. "Obstructive sleep apnea treatment in adults." *The Kaohsiung journal of medical sciences* 36, no. 1 (2020): 7-12.
6. Chaturvedula, Balavenkata B., Ashwin M. George, Naveen K. Mani, Saravana Dinesh, and Aravind K. Subramanian. "Pediatric Obstructive Sleep Apnea vs Adult Obstructive Sleep Apnea: An Orthodontic Perspective." *Indian Journal of Sleep Medicine* 15, no.2 (2020): 18.
7. de Menezes Duarte, Ricardo L., Guimarães Pereira Togeiro, Sonia M., Oliveira Palombini, Luciana D., Galhardo Rizzatti, Fabíola P., Fagundes, Simone C., Magalhães-da-Silveira, Flavio J., Cabral, Marília M. et al. "Brazilian Thoracic Association Consensus on Sleep-disordered Breathing." *Jornal Brasileiro de Pneumologia* 48, no. 4 (2022). <https://doi.org/10.36416/1806-3756/e20220106>.
8. Djonlagic, Ina, Mengshuang Guo, Paul Matteis, Andrea Carusona, Robert Stickgold, and Atul Malhotra. "First night of CPAP: impact on memory consolidation attention and subjective experience." *Sleep medicine* 16, no. 6 (2015): 697-702.
9. Ho, Matthew L., and Steven D. Brass. "Obstructive sleep apnea." *Neurology International* 3, no. 3 (2011): e15.
10. Javaheri, Shahrokh, Barbe, Ferran, Campos- Rodriguez, Francisco, Dempsey,

- Jerome A., Khayat, Rami, Javaheri, Sogol, Malhotra, Atul et al. "Sleep Apnea: Types, Mechanisms, and Clinical Cardiovascular Consequences." *Journal of the American College of Cardiology* 69, no. 7 (2017): 841-858.
11. Lai, Victor, Jayne C. Carberry, and Danny J. Eckert. "Sleep apnea phenotyping: implications for dental sleep medicine." *J Dent Sleep Med* 6, no. 2 (2019): 1-12.
 12. Malhotra, Atul, Ayappa, Indu, Ayas, Najib, Collop, Nancy, Kirsch, Douglas, Mcardle, Nigel, Mehra, Reena et al. "Metrics of sleep apnea severity: beyond the apnea-hypopnea index." *Sleep* 44, no. 7 (2021).
 13. Malhotra, Atul, Ayappa, Indu, Ayas, Najib, Collop, Nancy, Kirsch, Douglas, Mcardle, Nigel, Mehra, Reena et al. "Metrics of sleep apnea severity: beyond the apnea-hypopnea index." *Sleep* 44, no. 7 (2021).
<https://doi.org/10.1093/sleep/zsab030>
 14. Mohamed Nabih EL Gharib. Gestational Sleep Apnea. *Am J Biomed Sci & Res.* 2019 - 2(5). AJBSR.MS.ID.000601. DOI: 10.34297
 15. Mohammadi H, Aarabi A, Rezaei M, Khazaie H and Brand S (2021) Sleep Spindle Characteristics in Obstructive Sleep Apnea Syndrome (OSAS). *Front. Neurol.* 12:598632. doi: 10.3389
 16. Pavwoski, Patrick, and Anita V. Shelgikar. "Treatment options for obstructive sleep apnea." *Neurology: Clinical Practice* 7, no. 1 (2017): 77-85.
 17. Petrenko, Anatolii. "Breathmonitor: AI Sleep Apnea Mobile Detector." In *IEEE International Conference on System Analysis & Intelligent Computing*, pp. 201-208. Springer, Cham, 2022.
 18. Prathap Chandar Manivannan, Swarna Swathi Silla, Sarada Penmetcha. Obstructive Sleep Apnea In Children – A Review. *Int J Dentistry Oral Sci.* 2021;8(2):1447-1452
 19. Reshale Aiman Johar, Farah Hassan Bakhamees, Jawahir Omar Altamimi, Raghad Abdullah Alahmadi, Rahaf Abdulelah Alqurashi, Reem Mahmoud Hersi, Refal Mowaffaq Bougis and Rinad Haitham Bohairy "Obstructive Sleep Apnea: A Review Article". *Saudi Journal of Dental Research* (2021)
 20. Roush, Scott F., and Laird Bell. "Obstructive sleep apnea in pregnancy." *The Journal of the American Board of Family Practice* 17, no. 4 (2004): 292-294

21. Usmani, Zafar A., Michael Hlavac, Maureen Rischmueller, Subash S. Heraganahally, Cassie J. Hilditch, Sue Lester, Peter G. Catcheside, Nick A. Antic, Ching Li Chai-Coetzer, and R. Doug McEvoy. "Sleep-disordered breathing in patients with primary Sjögren's syndrome: a group-controlled study." *Sleep medicine* 13, no. 8 (2012): 1066-1070
22. Wang, Juan, Wang, Yan, Feng, Jing, Chen, yuan, and Jie Cao. "Complex sleep apnea syndrome." *Patient preference and adherence* 7, (2013): 633-641.