

ADVANCED SOCIAL INTERNET OF THINGS FOR REAL TIME MONITORING OF DIABETICS PATIENTS IN HEALTH CARE SYSTEM

Submitted in partial fulfillment of the requirements for the award of Master
of Technology degree in Medical Instrumentation

by

Mercy Grace M (41440008)



**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

APRIL – 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

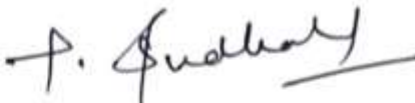
DEPARTMENT OF BIOMEDICAL ENGINEERING

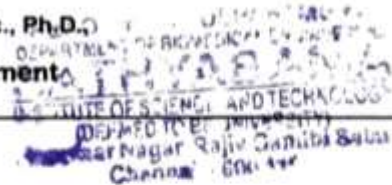
BONAFIDE CERTIFICATE

This is to certify that this Project report is the bonafide work of **Mercy Grace M** (Reg.No. **41440008**), who carried out the project entitled "**Advanced social Internet of Things for real time monitoring of diabetics patients in health care system**" under our supervision from **March 2023 to May 2023**.


Dr. ANIMA NANDA, Ph.D., FNRS, FIAS, FAB
Professor & Dean-IQAC
Internal Guide


Dr. KRISHNAMOORTHY, Ph.D.,
External Guide


Dr. T. SUDHAKAR, M.Sc., Ph.D.
Head of the Department



Submitted for Viva-voce Examination held on 5.5.23


Internal Examiner


External Examiner

DECLARATION

I, **Mercy Grace M** (Reg. No. **41440008**) hereby declare that the Project report entitled "**Advanced social Internet of Things for real time monitoring of diabetics patients in health care system**" done by me under the guidance of **Dr. Anima Nanda**, Ph.D., FNRS, FIAS, FAB, **Professor and Dean-IQAC**, Department of Biomedical Engineering is submitted in partial fulfillment of the requirements for the award of **Master of Technology** degree in **Medical Instrumentation**.

DATE: 5.5.23

PLACE: CHENNAI

M. Mercy Grace

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 would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Anima Nanda**, Ph.D., FNRS, FIAS, FAB, Professor and Dean, for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I convey my thanks to **Dr. T. Sudhakar**, M.Sc., Ph.D., Head of the Department, 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 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 extend my sincerest and heartfelt gratitude to my father **Mr. B. Maria Navis** and mother **Mrs. M. Shobana** and my **sisters** for their constant support for bringing this work into a good shape and unbound love and affection. I sincerely thanks all my **Family members** and **Friends** for their encouragement and support during my project work.

Mercy Grace
(Mercy Grace M)

ABSTRACT

Internet of Things (IoT) has advanced quickly in order to remotely monitor patients and improve the quality of healthcare services. Unfortunately, current systems have some shortcomings when it comes to informing patients of their health status based on ongoing changes in their health. The proposed project aims to develop an effective health monitoring tool for diabetic patients using ESP32 and Arduino Nano as microcontrollers. The system incorporates various sensors to monitor different health parameters such as non-invasive glucose detection, heart rate, step count, calorie output, sweat monitoring, and BMI calculation. The non-invasive glucose detection is done using a Temp6000 sensor, which measures the glucose levels in the patient's blood without the need for blood sampling. This sensor uses infrared light to measure the glucose levels in the skin's tissue. The heart rate and SpO2 levels are monitored using dedicated sensors connected to the Arduino Nano microcontroller. The step count and calorie output are monitored using an accelerometer sensor, which calculates the number of steps taken by the patient and then calculates the calories burnt based on this information. The sweat monitoring is done using a GSR sensor that measures the amount of sweat produced by the patient's body, which can be an indicator of stress or exercise intensity. The BMI calculation is done in the program using the patient's height and weight. All these parameters are monitored effectively through an IoT Blynk application, which displays real-time data and alerts the patient in case of any abnormalities or variations in the monitored parameters. The system's results were encouraging, and the data it collected is stored fast. This IoT-based health monitoring system makes it simple for doctors to get real-time data. Using this proposed method, doctors can monitor patient health from any location. Overall, this health monitoring tool provides an effective way for diabetic patients to monitor their health parameters and take necessary actions to manage their condition.

Keywords: IoT-based health monitoring system, diabetic patients, ESP32, Arduino Nano microcontroller, Temp6000 sensor

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF ABBREVIATIONS	iv
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
	1.2 DIABETES	2
	1.2.1 TYPES OF DIABETES	3
	1.2.2 SIGNS AND SYMPTOMS	4
	1.2.3 BLOOD GLUCOSE MONITORING	5
	1.2.3.1 INVASIVE METHODS	6
	1.2.3.2 NON-INVASIVE METHODS	7
	1.3 IOT IN DIABETES MONITORING	7
2	LITERATURE SURVEY	10
3	AIM AND SCOPE	18
	3.1 AIM	18
	3.2 SCOPE	18
4	MATERIALS AND METHODS	20
	4.1 ESP32	20
	4.1.1 GENERAL PIN FUNCTION	21
	4.1.2 SPECIAL PIN FUNCTION	21
	4.2 ARDUINO NANO	22
	4.2.1 GENERAL PIN FUNCTION	23
	4.2.2 SPECIAL FUNCTION	24
	4.3 TEMP6000 SENSOR	24

	4.3.1	PIN FUNCTION	26
4.4		ADXL335 SENSOR	27
	4.4.1	ADXL335 PIN DIAGRAM	28
	4.4.2	ADXL335 PIN FUNCTION	28
4.5		MAX30100 SENSOR	29
	4.5.1	MAX30100 PIN DIAGRAM	29
	4.5.2	PIN FUNCTION	30
4.6		GSR SENSOR	30
	4.6.1	PIN FUNCTION	31
4.7		ARDUINO IDE	31
	4.7.1	ARDUINO IDE PROCESSING	39
4.8		BLYNK APP	40
4.9		METHODOLOGY	42
	4.9.1	WORK FLOW	45
5		RESULTS AND DISCUSSION	48
	5.1	RESULTS	48
	5.1.1	PROTOTYPE OUTPUT	49
	5.1.2	BLYNK APP OUTPUT	50
	5.2	DISCUSSION	51
6		SUMMARY AND CONCLUSION	53
	6.1	SUMMARY	53
	6.2	CONCLUSION	54
		REFERENCES	55

LIST OF ABBREVIATIONS

ADC	-	ANALOG TO DIGITAL CONVERTER
AI	-	ARTIFICIAL INTELLIGENCE
API	-	APPLICATION PROGRAMMING INTERFACE
AREF	-	ANALOG REFERENCE
ASIoT	-	ADVANCED SOCIAL INTERNET SOCIAL OF THINGS
AWS	-	AMAZON WEB SERVICES
BG	-	BLOOD GLUCOSE
BMI	-	BODY MASS INDEX
CGM	-	CONTINUOUS GLUCOSE MONITORING
CMOS	-	COMPLEMENTARY METAL-OXIDE SEMICONDUCTOR
CSII	-	CONTINUOUS SUBCUTANEOUS INSULIN INFUSION
DAC	-	DIIGITAL TO ANALOG CONVERTER
DM	-	DIABETES MELLITUS
DMS	-	DIABETES MONITORING SYSTEM
EEPROM	-	ELECTRICALLY ERASABLE PROGRAMMABLE READ- ONLY MEMORY
FGM	-	FLASH GLUCOSE MONITORING
FPGA	-	FIELD PROGRAMMABLE GATE ARRAY
FSW	-	FITTING-BASED SLIDING WINDOW
GCP	-	GOOGLE CLOUD PLATFORM
GPIO	-	GENERAL PURPOSE INPUT/OUTPUT
GPRS	-	GENERAL PACKET RADIO SERVICES
GPS	-	GLOBAL POSITIONING SYSTEM
GSR	-	GALVANIC SKIN RESPONSE
HBA1C	-	GLYCATED HAEMOGLOBIN
HIPAA	-	HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT
I2C		INTER-INTEGRATED CIRCUIT
ICSP	-	IN-CIRCUIT SERIAL PROGRAMMING
IDE	-	INTEGRATED DEVELOPMENT ENVIRONMENT
IOT	-	INTERNET OF THINGS
JTAG	-	JOIN TEST ACTION GROUP

LADA	-	LATENT AUTOIMMUNE DIABETES IN ADULTS
LCD	-	LIQUID CRYSTAL DISPLAY
LED	-	LIGHT EMITTING DIODE
MCU	-	MICTURATING CYSTO URETHROGRAM
MI	-	MINIMALLY INVASIVE
ML	-	MACHINE LEARNING
MODY	-	MATURITY-ONSET DIABETES OF THE YOUNG
NFC	-	NEAR-FIELD COMMUNICATION
NI	-	NON-INVASIVE
NIR	-	NEAR INFRARED
OLED	-	ORGANIC LIGHT-EMITTING DIODE
PD	-	PARTIAL DISCHARGE
PPG	-	PHOTOPLETHYSMOGRAPHY
PSO	-	PARTICLE SWARM OPTIMIZATION
PWM	-	PULSE WIDTH MODULATION
RFID	-	RADIO-FREQUENCY IDENTIFICATION
RHM	-	REMOTE HEALTH MONITORING
RPI	-	RASPBERRY PI
RTC	-	REAL-TIME CLOCK
SAPT	-	SENSOR AUGMENTED PUMP THERAPY
SDA	-	SERIAL DATA LINE
SCL	-	SERIAL CLOCK LINE
SMO	-	SOCIAL MEDIA OPTIMISATION
SMS	-	SHORT MESSAGES
SPI	-	SERIAL PERIPHERAL INTERFACE
T1D	-	TYPE 1 DIABETES
T2D	-	TYPE 2 DIABETES
UART	-	UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER
VCC	-	VOLTAGE COMMON COLLECTOR
VIN	-	VOLTAGE INPUT
WHO	-	WORLD HEALTH ORGANIZATION
WIFI	-	WIRELESS FIDELITY
WSN	-	WIRELESS SENSOR NETWORK

LIST OF FIGURES		
FIG.NO	TITLE	PAGE.NO
4.1	ESP32	20
4.2	ARDUINO NANO	23
4.3	TEMP6000 SENSOR	26
4.4	ADXL335 PIN DIAGRAM	28
4.5	MAX30100 PIN DIAGRAM	29
4.6	GSR SENSOR	30
4.7	WORKING OF BLYNK APP	41
4.8	WORK FLOW DIAGRAM	45
5.1	PROTOTYPE OF HEALTH MONITORING DEVICE	48
5.2	DISPLAY OF HEALTH MONITORING DEVICE	49
5.3	BLYNK APP OUTPUT	50

LIST OF TABLES

TABLE	TITLE	PAGE NO.
5.1	PROTOTYPE OUTPUT	50
5.2	BLYNK APP OUTPUT	51

CHAPTER 1

INTRODUCTION

1.1. OUTLINE OF THE PROJECT

Instead of only being free from disease, health is defined as a complete condition of physical, mental, and social well-being. People's desire for a better life is fundamentally based on their state of health. IoT has been seen as the next technological revolution because it has recently enabled internal connectivity amongst all objects. The most impressive application of IoT is in healthcare management, which offers tools for monitoring environmental and physical conditions. IoT is nothing more than employing sensors and networks to connect computers to the internet. Diabetes is a chronic disease that affects millions of people worldwide. It requires continuous monitoring of various health parameters to manage the condition effectively. Our proposed project aims to develop a health monitoring tool specifically for diabetic patients that can monitor their vital signs, physical activity, and glucose levels. The proposed system utilizes advanced sensors such as an accelerometer, GSR sensor, Temp6000 sensor, heart rate and SpO2 sensors to monitor different health parameters. These sensors are integrated with ESP32 and Arduino Nano microcontrollers to collect and process data efficiently. The system also incorporates a BMI calculator that takes the patient's height and weight to calculate their body mass index. The system is designed to be portable and compact, allowing patients to carry it with them wherever they go. By using an IoT Blynk application, patients can access real-time data and receive alerts in case of any abnormalities or variations in the monitored parameters. The proposed system provides an effective and convenient way for diabetic patients to monitor their health parameters and take necessary actions to manage their condition. The proposed project offers a comprehensive solution for diabetic patients to monitor their health continuously. By integrating advanced sensors with ESP32 and Arduino Nano microcontrollers and an IoT Blynk application, the system provides a reliable and efficient way to collect and process data, enabling patients to manage their condition effectively. This enabled healthcare providers to remotely monitor patients' conditions and modify treatment plans as needed. By producing immediate results, this system will reduce patient waiting time. Doctors can monitor

patient health from any location using this proposed method. The physiological parameters chosen for this project are non-invasive glucose sensor, heart rate sensor, GSR sensor and accelerometer sensor. Parameters for this project are set based on data collected from healthy individuals. The result obtained are discussed and concluded.

1.2. DIABETES

Diabetes mellitus, usually referred to as "diabetes," has become an epidemic on a global scale. Diabetes is a long-term metabolic condition defined by high blood glucose (also known as blood sugar), which over time can seriously harm the heart, blood vessels, eyes, kidneys, and nerves. In 1936, diabetes mellitus types 1 and 2 were distinct from one another (Olokoba et.al., 2012). Blood glucose (BG) levels fluctuate due to the long-term metabolic disorder known as diabetes, which can be caused by either the body producing insufficient insulin or by the body's inability to utilise the insulin that is produced (Alfian et.al., 2018). Regular blood glucose monitoring is crucial for controlling diabetes. Type 2 diabetes is the most prevalent kind and typically affects adults. It develops when the body stops producing enough insulin or grows resistant to it. Type 2 diabetes prevalence has sharply increased during the past three decades in nations of all income levels.

The estimated global prevalence of diabetes mellitus in 2011 was 366 million. This number will rise to 552 million by the year 2030. Type 2 diabetes is becoming more common worldwide, with 80 percent of individuals affected living in low- and middle-income countries. 4.6 million people died from DM in 2011. The majority of the income nations, and diabetes is directly responsible for 1.5 million fatalities annually. Throughout the past few decades, diabetes has been progressively rising in both incidence and prevalence. According to predictions from 2011, 439 million people will have type 2 diabetes by 2030 (Olokoba et.al., 2012, p.269). According to the World Health Organization (WHO), there are currently 450 million cases of diabetes globally. By 2045, that number may increase to 700 million cases, and in the United States alone, it may reach 39.7 million cases by 2030 and 60.6 million cases by 2060 (Tang et.al., 2020, p. 6925)). Sridevi et. al.,(2021) states that it was extremely concerning that 82 million individuals solely in Southeast Asia have diabetes. In the US, 96 million persons, or more than one in three, have prediabetes. More than 80% of them are unaware that they have it.

1.2.1. Types of Diabetes

Type 1 diabetes (T1D) and Type 2 diabetes (T2D) are the two main subtypes of diabetes, respectively. As a result of the body's limited ability to produce insulin, type 1 diabetes, also known as T1D, causes variations in blood glucose levels. Juvenile diabetes or insulin-dependent diabetes, often known as type 1 diabetes, is a chronic illness in which the pancreas produces little to no insulin on its own. Type 2 diabetes, also referred to as T2D, is characterised by variations in blood glucose levels as a result of the body's inability to use the insulin that it produces. Non-insulin-dependent diabetes was the previous term used to describe type 2 diabetes. Type 2 diabetics often do not need to take as many frequent measures or receive insulin therapy. They should instead switch to a healthy lifestyle, it is urged. The majority of Type 1 patients are children (kids), who cannot produce their own insulin and need to be continuously given IV insulin (AL-Suhail et.al., 2017). An inadequate amount of insulin is produced by the pancreas, which causes type 1 diabetes. On the other hand, type 2 is primarily characterised by patients' rising insulin resistance and decreasing insulin sensitivity, which leads to poor insulin utilisation (Tang et.al., 2020, p. 6925).

Other kinds of diabetes include: type 3C diabetes, Latent autoimmune diabetes in adults (LADA), Maturity-onset diabetes of the young (MODY), Neonatal diabetes and brittle diabetes. Type 3c diabetes happens when a pancreas experiences damage (other than autoimmune damage), which affects its ability to produce insulin. Pancreatitis, pancreatic cancer, cystic fibrosis and hemochromatosis can all lead to pancreas damage that causes diabetes. Having their pancreas removed (pancreatectomy) also results in Type 3c. Latent autoimmune diabetes in adults (LADA) is like Type 1 diabetes. LADA also results from an autoimmune reaction, but it develops much more slowly than Type 1. People diagnosed with LADA are usually over the age of 30.

A genetic abnormality that changes how your body generates and uses insulin causes maturity-onset diabetes of the young (MODY), also known as monogenic diabetes. MODY now comes in more than ten different varieties. Up to 5% of

diabetics might develop it, and it frequently runs in families. In Neonatal diabetes, during the first six months of life, this unusual form of diabetes develops. It also has a monogenic diabetes subtype. The lifelong form of neonatal diabetes known as persistent neonatal diabetes mellitus affects about 50% of newborns. The illness vanishes for the other half after a few months of beginning, but it may reoccur later in life. This condition is known as transitory neonatal diabetes mellitus. Type 1 diabetes known as "brittle diabetes" is characterised by frequent, significant spikes and falls in blood sugar levels. Hospitalization results frequently from this instability. Rarely, brittle diabetes may need to be treated permanently with a pancreas transplant.

1.2.2. Signs and Symptoms

Numerous individuals disregard diabetes' warning signs and symptoms because it is a chronic condition. Since hyperglycemia's effects take time to manifest, unlike those of many other diseases, people do not consider this to be a serious problem. The fact that damage may start years before symptoms show up is a fact that few people are aware of. This is regrettable since early signs can be identified, the disease could be treated straight away, and vascular issues would be avoided (Ramachandran et.al., 2014, p.579).

The severity of diabetes symptoms is related to blood sugar levels. Some people may not exhibit symptoms, particularly if they have prediabetes, gestational diabetes, or type 2 diabetes. Symptoms of type 1 diabetes frequently appear suddenly and are more severe. Diabetes type 1 can develop at any age. Yet, it frequently begins in childhood or adolescence. The more prevalent kind of diabetes, type 2, can manifest at any age. Those over 40 are more likely to have type 2 diabetes. Moreover, type 1 diabetics may experience nausea, vomiting, or stomach discomfort. Severe type 1 diabetes symptoms might emerge in as little as a few weeks or months. The first signs of type 2 diabetes frequently appear years later. Some individuals report no symptoms at all. Though it increasingly affects kids and teenagers, type 2 diabetes typically develops in adults. It's crucial to understand the risk factors for type 2 diabetes because the symptoms are difficult to detect. The majority of the time, gestational diabetes (diabetes during pregnancy) has no

symptoms. Between 24 and 28 weeks of pregnancy, a doctor should check a pregnant woman for gestational diabetes.

Symptoms of diabetes includes:

1. Unexplained weight loss
2. Frequent fatigue
3. Irritability
4. Repeated infections especially in the Genital area, Urinary tract, Skin, Oral cavity
5. Delayed wound healing
6. Dry mouth
7. Itching
8. Decreased vision
9. Burning sensation on feet
10. Pain and numbness on feet
11. Impotence or erectile dysfunction
12. Reactive hypoglycaemia (Ramachandran et.al., 2014, p.579).

1.2.3. Blood Glucose Monitoring

The management of diabetes requires frequent blood glucose testing. Nivetha et.al., (2018) states that the one of the most significant health issues facing people today was diabetes, which causes blood glucose levels to rise exponentially. Regular blood glucose testing is necessary to prevent such consequences from diabetes (Jamal et.al.,2021). The normal nature of regular glucose testing equipment is obtrusive. Invasive procedures cause pain, require a lot of time and money, penetrate healthy tissue, and increase the risk of catching infectious diseases. As a result, employing a non-invasive method to measure blood glucose can be pleasurable, safe, and economical (Nivetha et.al.,2018).

To avoid irregularities in their glucose ranges,a person with diabetes needs to keep their blood glucose level under control. The following three types of monitoring methods—invasive, minimally invasive, and noninvasive—have been established (Valenzuela et.al.,2020, p.921). In Valenzuela's et.al., (2020) study found the following:

- The majority of operations fall under the "invasive" category since they directly contact the patient's blood, which provides the highest level of precision in results. The usual approach involves topping a finger, which creates discomfort for the sufferer. Since infections are a possibility, the measures must be carried out in accordance with a rigid cleaning schedule.
- One of the techniques under investigation is minimally invasive. This procedure makes use of micropores, which are microscopic skin holes produced caused by laser light. The pores are opened and a device produces a steady vacuum pressure in order to collect a small amount of transdermal body fluid. The received sample is processed by an electrode made of enzymes, which then calculates the glucose content.
- The noninvasive group includes noninvasive glucometers, whose purpose is to replace a blood variable with one that may be accessed by an external technique. A sensor is attached to a specific part of the body in order to test blood glucose. Because doctors have been given the assurance that the amounts of glucose found in a patient's blood and on their skin are nearly 100% accurate, the values are now considered to be nearly 100% dependable. There have been studies on noninvasive glucose tests using saliva, urine, sweat, and tears, however it is not possible to continuously monitor the glucose levels(p.921).

Blood glucose monitoring can be categorised as either invasive or non-invasive depending on whether the test has damaged human skin (Tang et.al., 2020 ,p.6925).

1.2.3.1. Invasive Methods

The invasive method causes pain, drains vital blood, and intrudes on healthy tissues. Due to injection, it also transmits infection. To measure the glucose levels, serum, plasma, or whole blood samples can all be used. Based on the study of Villena et.al., (2019), glucose monitoring was only feasible in laboratories due to issues with non-specificity, toxicity, and cross-reaction with other drugs. The methods of choice for determining blood glucose levels in laboratories include enzymatic-amperometric and hexokinase. They asserts that they could all detect a wide range of glucose concentrations, including in

hypoglycemia and hyperglycemia, and are all very sensitive and specific (Villena et.al., 2019, p.800).

1.2.3.2. Non-Invasive Methods

Painless non-invasive treatment is used. Outside of a body opening that is either naturally occurring or purposefully made, there is no skin breach, contact with the mucosa, skin, or any interior body cavities. Non-invasive and minimally invasive techniques are the two primary divisions of this approach (NI). NI technologies are those that need interstitial fluid and tears be removed from the body in order to measure the concentration of glucose through an enzyme reaction. As stated by Villena et.al., (2019), NI technologies just rely on radiation, not any physiological fluids, as is the case with conventional technology (p.800). The radiation that is produced by this procedure is dangerous, thus people should avoid it. The numerous investigations for non-invasive blood glucose measurement have been done employing near-infrared photoacoustic spectroscopy, Raman spectroscopy, polarisation technique, and light scattering approach (Sridevi et.al.,2021, p.1-13).

1.3. IOT IN DIABETES MONITORING

The healthcare sector has undergone a technological revolution, particularly in the treatment of long-term conditions like diabetes. Millions of individuals throughout the world suffer with diabetes, which needs to be managed and monitored constantly to avoid long-term consequences. A paradigm shift in how healthcare professionals treat diabetic patients has occurred with the introduction of the internet of things (IoT). An innovative idea called the Advanced Social Internet of Things (ASIoT) combines the internet of things with social media platforms to enable the sharing of real-time data and information amongst healthcare professionals, patients, and carers.

Several IoT technologies can help patients and medical staff monitor these aspects without the need for frequent hospital visits and lessen the workload on the typically understaffed medical staff. A few of the current options and services are networked sensors, RFID tags, smart watches, and wearable sensors. Several advantages pertaining to patient safety and effectiveness are promised to be provided by these technologies (A.M. et.al., 2019, p.03016).

The management of chronic diseases like diabetes depends on effective communication between various stakeholders in the healthcare ecosystem, which is made possible by the ASIOT framework.

Diabetes is a chronic condition that necessitates ongoing attention to medication compliance, nutritional consumption, and blood glucose levels. Patients used to be compelled to visit medical institutions on a regular basis for checkups, which might be expensive and time-consuming. With the ASIOT method, patients can communicate real-time data about their blood sugar levels, nutritional consumption, physical activity, and medication adherence with healthcare professionals, family members, and carers via social media platforms. This strategy may lead to better patient outcomes, fewer readmissions to the hospital, and lower healthcare costs.

ASIOT uses a variety of IoT devices, including glucose metres, insulin pumps, smartwatches, and fitness trackers, to monitor diabetic patients in real-time. Using social media networks, these gadgets broadcast real-time data on blood glucose levels, physical activity, and medication adherence to healthcare professionals and carers. With this strategy, healthcare professionals may monitor patients remotely, act quickly, and give individualised care.

Using social media platforms to improve communication between patients, healthcare professionals, and carers is another component of the ASIOT architecture. In today's world, social media sites like Facebook, Twitter, and Instagram are commonplace, and they provide fantastic opportunities for healthcare professionals to interact with patients and carers. Patients can ask questions, share their experiences, and get support from other patients and carers via these platforms. Social media channels can be used by healthcare professionals to distribute instructional content, offer guidance, and respond to patient inquiries.

A promising strategy to improve patient outcomes in the management of chronic diseases like diabetes is the enhanced social internet of things for real-time monitoring of diabetic patients. This method makes use of social media platforms and Internet of Things (IoT) devices to provide seamless communication between patients, healthcare professionals, and carers. Patient outcomes, readmission rates to hospitals, and healthcare expenditures may all be improved by real-time monitoring of diabetes patients utilising ASIOT. IoT and social media platform

integration in healthcare systems will probably become more common in the future as technology develops.

For even more individualised care for diabetic patients, IoT devices can be combined with other technologies like artificial intelligence (AI) and machine learning (ML). In order to forecast future glucose levels, AI algorithms can examine data from CGMs and other devices. These algorithms can also notify patients and their healthcare professionals of potential issues before they arise. Large datasets can be analysed by ML algorithms to find patterns and trends in glucose levels, which can assist healthcare professionals in creating more effective treatment plans. The Internet of Things application can be expanded upon by developing a mobile application for a more effective personal monitoring system. The user of the mobile application can monitor their health in a more interactive way and the mobile application can retrieve data and information from the database (Ruhani Ab. Rahman et.al., 2017, pp. 168-173).

CHAPTER 2

LITERATURE REVIEW

Amine Rghioui et.al., (2020) developed an IoT based diabetic patient monitoring system based on the patient's blood glucose level, body temperature and their location. In this work, they developed a monitoring system for diabetic patients to help them self-manage their chronic diseases using the Node MCU. This technique automatically transfers glucose readings from diabetic patients' monitoring devices to a mobile app, where they are stored in the cloud. Data can be categorised by the server through processing. The MCU node is connected to the glucometer to continuously record the glucose level in a diabetic patient. Patients can be followed up on using the information gathered. It can be thought of as a platform for glucose data that facilitates interaction between patients and medical experts. Using the learning machine algorithms installed on the server, it can automatically help users track their blood glucose levels and predict upcoming changes in health. On the diabetic datasets, they used the following four primary algorithms: OneR, Random Forest, SMO, and Naive Bayes. They made an effort to assess the effectiveness and efficiency of the stated algorithms' accuracy, precision, and sensitivity. The Random Forest algorithm generally outperforms SMO, OneR, and Naive Bayes algorithms at anticipating the onset of diabetes disease [20].

Ruwaida Ramly et.al., (2020) designed an IoT based glucose monitoring system. The project's blood glucose sensor monitors the amount of glucose. The Arduino Uno serves as the brain of the system and retrieves and processes data from the sensor before transmitting it to the cloud via a Wi-Fi module. Information is sent to the smartphones of the patient and the doctor after data processing. The warning indication states that if the glucose level is higher than the permitted range, the patient must provide the proper dose of insulin. An iterative waterfall methodology was utilised to facilitate the development of this project. It is possible to use this approach to loop back to the previous iteration if the process encounters any mistakes or doesn't satisfy all the requirements. The project's outcome is clear. Users can view the project's outcomes thanks to the message delivered to Blynk,

which presented the glucose data and dose unit of insulin that the patient must administer when the glucose level is abnormal [21].

In order to display real-time glucose, body temperature, and contextual data (i.e., environmental temperature) to end-users like patients and clinicians in graphical and human-readable forms, Tuan Nguyen Gia et. al. (2017) developed an Internet of Things (IoT)-based system architecture from a sensor device to a back-end method. Additionally, by adapting the nRF communication protocol to the glucose monitoring system, a high level of energy economy is attained. Additionally, they designed energy collecting components for the sensor device after analysing its energy use. Last but not least, the project provides a wide range of advanced services at the gateway level, including a push notification service for warning patients and medical staff in case of emergency situations (i.e., too low or too high glucose level). The results reveal that the technology can monitor glucose levels continuously, remotely, and in real time. The results also demonstrate that a high level of energy efficiency can be achieved in the sensor device by integrating the power management unit, the energy harvesting unit, and the customised nRF component [9].

Antonio Alarcón-Paredes et.al.,(2019) modelled a non-invasive blood glucose monitoring that is built on the Internet of Things (IoT). A visible laser beam, a Raspberry Pi Camera, and a Raspberry Pi Zero (RPI) that is powered by a battery bank were all used in this experiment and were all built inside a glove. The glove embodiment incorporates the Raspberry Pi Zero board and the IoT sensor's image acquisition stage. To collect data for the non-invasive monitoring, the RPi Zero takes a number of pictures of the user's fingertip and computes their histograms. On a Flask microservice using the Tensorflow libraries, generated data is handled by an artificial neural network (ANN). All measurements were made in-vivo, and the mean absolute error (10.37%) and Clarke grid error (90.32%) were used to compare the findings to laboratory blood testing. The predicted glucose values can be collected and used for monitoring by an end device, like a smartphone [1].

In order to monitor the blood glucose level, Omar Alfandi (2022) used non-invasive Wireless Sensor Network (WSN)-based approaches. As an optical method of the non-invasive technology, near-infrared (NIR) has been used to help diabetic

patients continuously test their blood without feeling any pain. In order to alert the patient's parents or guardians of their situation when they are about to arrive at critical circumstances, especially at night, the proposed solution will send an alarms and notifications using Short Messages (SMS) to up to three persons along with the patient's current location. In addition, a Machine Learning (ML) model is utilised to predict scenarios in which the patient may encounter significant issues. Because this model has never been used to anticipate a future projection of the patient record, it mirrored best practises in this area of chronic health. A multivariate time- series data collection, known as AIM '94, was used to train the proposed ML model. The collected data shows that serious critical hyperglycemia situations can be accurately identified with a high degree of precision[3].

M.Rahmat et.al., (2017) modelled a device called GluQo, a non-invasive method to monitor glucose level. To monitor blood glucose optically, a near-infraredLED was put over the fingertip. The amount of glucose in the blood was calculatedbased on the brightness of the light received. The signal was filtered and amplifiedbefore being delivered to the microcontroller to be shown on an LCD display. The estimated glucose level for an individual was based on the investigated voltages received. Additionally, an Android app displayed the glucose levels after being transferred through WiFi to a phone. Validation and calibration for the prototype were carried out. The devised method's percentage error of the glucose reading incomparison to the finger-prick method was 7.20%. Voltage was calibrated against glucose levels, and a correlation coefficient of 0.9642 was discovered. The correlation for this parameter was strong. This indicates that the voltage signals from the sensor and the anticipated glucose readings exhibit a strong correlation [18].

Jose Jorge Rodrigues Barata et.al., (2019) designed a hardware to a portableIoT-based device called LibreMonitor in order to remotely and continuously check the health of diabetes patients. This hardware integrates an NFC module and an IoT card to gather data from the Freestyle Libre sensor's glucose measurement and transmit it over Bluetooth LE communication to a mobile application. The ability to access records and check blood glucose levels in real time is made possible by the patient's ability to see the screen. The information obtained and shown in the app is transmitted through WiFi, 3G, or 4G to a secure cloud-based platform were

healthcare professionals will store and analyse it. An application built on the cloud was used to save, examine, and report data regarding the glucose levels of diabetic patients. In addition, data mining and machine learning techniques can be used thanks to the available data. Furthermore, they looked at the glycemic system and compared it to the Freestyle Libre; based on the minor percentage changes, they got to the conclusion that it was effective. The system met the criteria for glycemic average, precision, and accuracy in the documentation of blood glucose measurements recorded by the sensor, and it demonstrated that its use is practical when done so under the supervision of a qualified professional [3].

A study of prior studies employing mobile health apps for treating, diagnosing, monitoring, detecting, and predicting diabetes was suggested by Omar AlShorman et.al., (2021). One of the main advantages of mHealth, according to them, is real-time, continuous monitoring with high dependability, accessibility, and availability. In addition, mHealth is used to control and monitor diabetes-related aspects of diet, smoking, glycemic management, obesity, and physical activity. They said that mHealth, wireless body sensor networks (WBSN), the Internet of Things (IoT), and communication generations and technologies (including 3G, 4G, GPRS, GPS, and bluetooth) are all becoming crucial elements for keeping an eye on patients and the elderly. They aim to review remote health monitoring (RHM) for diabetic patients based on mHealth technologies, including diabetic management and control, diabetes prevention, diabetes intervention programme, diabetes self-efficacy, continuous glucose monitoring, improved glycemic control, diabetic patient treatment, diabetes prediction system, improved diabetes care, continuous and remote monitoring system, insulin dose management, and carbohydrate monitoring. The main problems and trends that this technology may encounter in the future are also discussed [5].

M.Saravanan et.al., (2018) developed a brand-new assortment of wearable gadgets, including a pair of smart socks, a smart wristband, and a smart neck band, to continuously check the health of diabetes patients. These devices are networks of different sensors that report data on weight gain and loss, movement patterns, skin moisture, heart rate, and food intake. To send a packet of all sensor readings to the mobile app via Bluetooth, the devices require controllers. With the use of a machine learning system, they have predicted changes in patient status. They use

the neck band's output to calculate how much food is being consumed, which helps people control their meals to lower their blood sugar levels. Smart socks will locate the area that may be more prone to developing foot ulcers so that the user may start treating it. Based on the data they had collected; they calculated the current level and the expected future levels. The coefficient for each of the three wearable modules is established following the processing of the data from the three wearables, and it is then sent into the particle swarm optimization (PSO) algorithm as input. PSO is then used to forecast the individual's current level of diabetes. These results are shown on the smartphone app so the user may check whether their blood glucose levels have increased or decreased [23].

Praful p. Pai et.al., (2018) designed a portable embedded system for taking photoacoustic measurements on tissues to estimate glucose concentration by using field programmable gate array (FPGA). The back-end architecture enables rapid data gathering and noise reduction for photoacoustic measurements while operating at 274.823 MHz on a Xilinx Virtex-II Pro FPGA. The glucose measuring method is proven using glucose solutions in vitro and on tissues in vivo. The photoacoustic signal amplitude varies linearly with sample glucose content. Using a range of photoacoustic signal features, a kernel-based regression technique is utilised to forecast glucose concentration from photoacoustic measurements. In comparison to past attempts, the calibration method performs better, with a mean absolute relative difference of 8.84% and a Clarke Error Grid distribution of 92.86% and 7.14% over Zones A and B of the grid, respectively. A cloud computing infrastructure for automatic blood glucose monitoring has been suggested to make it easier for diabetics to communicate with their doctors and carers [16].

Gaoba Zhang et.al., (2020) proposed a smart, non-invasive blood glucose monitoring system that uses smartphone PPG signals to classify a user's blood glucose level as normal, borderline, or warning. The key implementation procedures of the proposed system are as follows: (1) a novel algorithm for acquiring photoplethysmography (PPG) signals using only smartphone camera videos; (2) a fitting-based sliding window (FSW) algorithm to remove varying degrees of baseline drifts and segment the signal into single periods; (3) extracting distinguishing features from the Gaussian functions by contrasting PPG signals at various blood glucose levels; and (4) categorising the valid systolic signals. They evaluated the

proposed system using data from 80 patients. Experimental results show that the system can estimate blood glucose levels with an overall accuracy of 81.49% and can distinguish between genuine and false signals with an accuracy of 97.54%. The recommended approach serves as a manual for incorporating non-invasive blood glucose technologies into commonplace or clinical settings. This study demonstrates the significant potential of smartphone-based PPG signals for measuring a person's blood glucose level [29].

A noninvasive blood glucose metre with multi-sensor fusion was proposed by Feng Rui and his colleagues in 2021. It is useful for rapid yet accurate blood glucose monitoring and diabetes screening. Principle components analysis and discriminant analysis were used to build a classification model to ascertain whether the subjects had diabetes. One-factor analysis of variance was used to extract the traits of diabetics. The subjects were divided into different categories using the unsupervised K-means clustering algorithm, and models were then developed separately for each category to provide more coverage and greater accuracy. This was done due to the difficulties in modelling noninvasive blood glucose monitoring caused by various pathogeneses, disease processes, and intricate complications among various diabetics. Clinical testing of the precision involved 254 patients with type II diabetes. The results in the Parkes error grid were as follows: 58.33% in Zone A, 39.43% in Zone B, and 2.24% in Zone C, with a correlation coefficient of 0.69 and a root mean squared error of 2.67 mmol/L [8].

Maria Kamusheva et.al., (2021) examined whether diabetes monitoring systems (DMS) were a feasible method for routine glucose level control in patients with Type 1 or Type 2 diabetes in order to enhance therapeutic outcomes or to quickly identify and correct incorrect patient behaviours. To locate every study relevant to the research question, Cochrane was employed. The type of study, diabetes type, type of intervention, changes in clinical parameters, and any other relevant results were all obtained and summarised. 33 of the 1,495 studies that were initially found included more than 44,100 patients with Type 1, Type 2, or gestational diabetes. They comprised Closed-Loop Systems, Flash Glucose Monitoring (FGM), Continuous Subcutaneous Insulin Infusion (CSII), Sensor Augmented Pump Therapy (SAPT), and Telemonitoring. Additionally highlighted were Continuous Glucose Monitoring (CGMS) and SMBG. Most studies discovered that DMS has a

modest nominal effectiveness. The use of DMS reduced glycated haemoglobin (HbA1c) readings from 0.17 to 0.70%, according to an overall analysis of 15 meta-analyses and 11 systematic reviews, the majority of which (10 and 6, respectively), focused on patients with Type 1 diabetes [12].

Shalini R and et.al., (2019) proposed a non-invasive, sensitive acetone sensor device that provides quick and real-time electronic readouts of acetone levels in the breath that are converted to suitable glucose levels in the blood utilising the acetoneglucose negative correlation. The breakdown of excess acetyl-CoA produced by the metabolism of fatty acids in diabetic patients causes an increase in acetone levels in the blood. This acetone enters the lungs, where it is released through breathing and removed by the urine. As a result, the amount of acetone in someone's breath might be used to determine their blood sugar levels. A TGS822 tin oxide (SnO_2) sensor was used to determine how much acetone was present in the exhaled air. There was a correlation between blood glucose levels and acetone levels in exhaled breath. Additionally, doctors can telemonitor this system at any time, from any location, using laptops, smartphones, and other internet-connected electronic devices. For showing real-time glucose to end users like patients and clinicians in graphical and human-readable forms, they developed an IoT-based system architecture from a sensor device to a back-end system. This sensor gadget talks with the server using an API key, and the sensor readings are displayed graphically for easier understanding by the end user. The findings show that breath acetone levels can be used to precisely measure blood glucose levels [24].

Shygyri Haxha et.al., (2016) reported a non-invasive glucose monitoring system based on spectroscopy that measures glucose concentration. Both in vitro and in vivo investigations are conducted, and the technique used is near-infrared transmission spectroscopy. According to the study, sensor output voltage and glucose concentration levels are related. They demonstrated an inexpensive model of a noninvasive glucose monitoring system based on spectroscopy, which exhibits encouraging results in vitro and creates a link between varying blood glucose levels and optical signals. When the preliminary findings are benchmarked using the Clarke grid analysis, the results are encouraging and the bulk of the markings (obtained test results) fall within the permitted range. Invasive and non-invasive readings were taken at intervals of one minute. The range of invasive values is from

70 to 105, while the range of non-invasive values is from 70 to 110. The obtained results from the test measures fall within the permissible range, proving the accuracy of the created and suggested non-invasive glucose sensor [10].

Md. Iftekharul Alam Efat et.al., demonstrated a smart health monitoring system for diabetic patients that keeps track of their condition based on blood sugar, heart rate, food intake, sleep quality, and exercise. As an example, this technology continuously receives variables (data) from sensors, processes them with a neural network to assess the data, and then outputs four levels of health risk status: low, medium, high, and extreme. Depending on the type of patient and their medical histories, the risk status can vary. Additionally, if the patient's health is in a high- or extreme-risk category, an automatic phone call and/or SMS notification will be sent to the patient's relative along with the patient's location. In addition, it calls the patient's closest hospital in cases of high danger. Yet, in both instances, the system gives allied instructions as voice commands to the patient's mobile device. Using 25 diabetes patients as test subjects, this technology effectively identified the appropriate risk level with an accuracy rate of 84.29%, which is a level of accuracy for assessing health risk that is extremely acceptable [7].

CHAPTER 3

AIM AND SCOPE

3.1. AIM

The aim of ASIOT for real-time monitoring of diabetic patients in healthcare systems is to improve the quality of care for people with diabetes by providing timely and accurate information to healthcare providers. This system is designed to develop a health monitoring tool for diabetic patients using ESP32 and Arduino Nano microcontrollers, which can monitor their vital signs, physical activity, and glucose levels and to provide significant insights to patients and healthcare professionals that can enhance diabetes treatment and avoid complications. The system is designed to be portable and compact, allowing patients to carry it with them wherever they go.

3.2. SCOPE

The Advanced Social Internet of Things (ASIOT) has a wide range of potential uses and advantages for real-time monitoring of diabetes patients in healthcare systems. The aim of this project is to offer a comprehensive and individualized approach to diabetes management by using the power of IoT devices and data analytics to the full potential. This system incorporates advanced sensors such as an accelerometer, GSR sensor, Temp6000 sensor, heart rate and SpO2 sensors to monitor different health parameters. By using an IoT Blynk application, patients can access real-time data. In medical terms, the proposed project aims to provide a comprehensive solution for diabetic patients to monitor their health continuously.

Diabetes is a chronic condition that requires regular monitoring of various health parameters to manage the condition effectively. The proposed system offers a convenient and efficient way for patients to monitor their health parameters, enabling them to take necessary actions to manage their condition effectively. In medical terms, the proposed project aims to provide a comprehensive solution for diabetic patients to monitor their health continuously.

Firstly, the system enables continuous monitoring of vital signs, physical activity, and glucose levels, allowing patients to manage their condition effectively. This can be especially useful for patients who have difficulty managing their diabetes with

traditional methods. Secondly, Patients who utilise this system receive a user-friendly dashboard that shows their glucose levels and other health parameters in a simple and understandable manner. Thirdly, the system can be used for remote patient monitoring, which can be especially useful for patients who live in remote areas or have limited access to healthcare providers. Healthcare providers can monitor the patient's health parameters and make necessary adjustments to the treatment plan remotely.

Fourthly, the system can be used for research purposes, allowing researchers to collect data on various health parameters and analyze it to gain insights into the management of diabetes. This can lead to the development of more effective treatments and management strategies for diabetes. Fifthly, the system can be used for personal health monitoring by individuals who want to monitor their health parameters regularly, enabling them to maintain a healthy lifestyle and prevent the onset of diabetes. Lastly, the system can be used for public health purposes, allowing researchers to collect data on various health parameters of the population. This can provide insights into the prevalence and management of diabetes in the community, potentially leading to the development of more effective public health strategies to manage diabetes.

The proposed health monitoring tool has various potential applications in the medical field, ranging from the management of diabetes to research and public health. It enables continuous monitoring of vital signs, physical activity, and glucose levels, allowing patients to manage their condition effectively, healthcare providers to provide optimal care, and researchers to gain insights into the management of diabetes.

This system also increases the effectiveness and efficiency of healthcare providers. These details can help medical professionals make more educated judgements on the dosage of medications, suggested diets and lifestyles, and other diabetes management-related issues. Additionally, by minimising the need for in-person visits and facilitating more effective and efficient care, the system can expedite interactions between patients and healthcare professionals.

CHAPTER 4

MATERIALS AND METHODS

4.1. ESP32

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios. ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends.



Fig 4.1. ESP32

The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption. ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also

integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

4.1.1. General pin function

- Digital Input/Output (GPIO): These pins can be used for digital input or output, and can be configured as input, output or input/output. The ESP32 has a total of 36 GPIO pins.
- Analog Input (ADC): The ESP32 has 18 analog input pins that can be used to read analog signals. These pins are capable of measuring a voltage between 0 and 3.3V.
- Pulse Width Modulation (PWM): The ESP32 has 16 PWM output pins that can be used to generate analog signals. PWM pins can be used to control the brightness of LEDs or the speed of motors.
- Serial Communication: The ESP32 has two UART interfaces, two SPI interfaces, and two I2C interfaces for serial communication with other devices.
- I2S Interface: The ESP32 has an I2S interface that can be used to interface with audio devices such as speakers, microphones, and audio codecs.
- Capacitive Touch Interface: The ESP32 has a built-in capacitive touch sensor, which can be used to detect touch input.
- SD Card Interface: The ESP32 has an SD card interface, which can be used to read and write data to an SD card.
- Ethernet Interface: The ESP32 has a built-in Ethernet MAC interface, which can be used to connect to wired networks.

4.1.2. Special Pin Function

- Boot mode selection pins: The ESP32 has two pins, GPIO0 and GPIO2, that are used to select the boot mode of the chip. These pins are used to determine whether the ESP32 boots from internal flash memory, external flash memory, or over the UART interface.
- JTAG interface: The ESP32 has a JTAG interface that can be used for debugging and programming the chip. The JTAG interface uses GPIO12,

GPIO13, GPIO14, GPIO15, and GPIO25 pins.

- RTC pins: The ESP32 has two pins, GPIO34 and GPIO35, that are used to interface with the real-time clock (RTC) module. These pins can be used to read the time and date from the RTC module.
- LED control pins: The ESP32 has two pins, GPIO2 and GPIO4, that are connected to the onboard LED. These pins can be used to control the state of the LED.
- DAC pins: The ESP32 has two DAC pins, GPIO25 and GPIO26, that can be used to output analog signals. These pins can be used to generate audio signals or control the output voltage of a power supply.
- Camera interface: The ESP32 has a dedicated camera interface that can be used to connect a camera module. The camera interface uses several pins, including GPIO5, GPIO14, GPIO15, GPIO18, GPIO19, and GPIO21.

4.2. ARDUINO NANO

Arduino Nano is a microcontroller-based development board that is designed to be small, compact, and easy to use. It is based on the ATmega328P microcontroller, which is an 8-bit AVR microcontroller that provides 32KB of flash memory, 2KB of SRAM, and 1KB of EEPROM. The board has a compact form factor, measuring only 45mm x 18mm, which makes it easy to use on a breadboard or in a small project.

The Arduino Nano has 14 digital input/output pins, 8 analog input pins, and 6 PWM (Pulse Width Modulation) pins. These pins can be used to interface with a variety of sensors, actuators, and other electronic components. The board also has a reset pin, a UART (Universal Asynchronous Receiver/Transmitter) interface, and a SPI (Serial Peripheral Interface) interface, which allows it to communicate with other devices, such as sensors and displays.

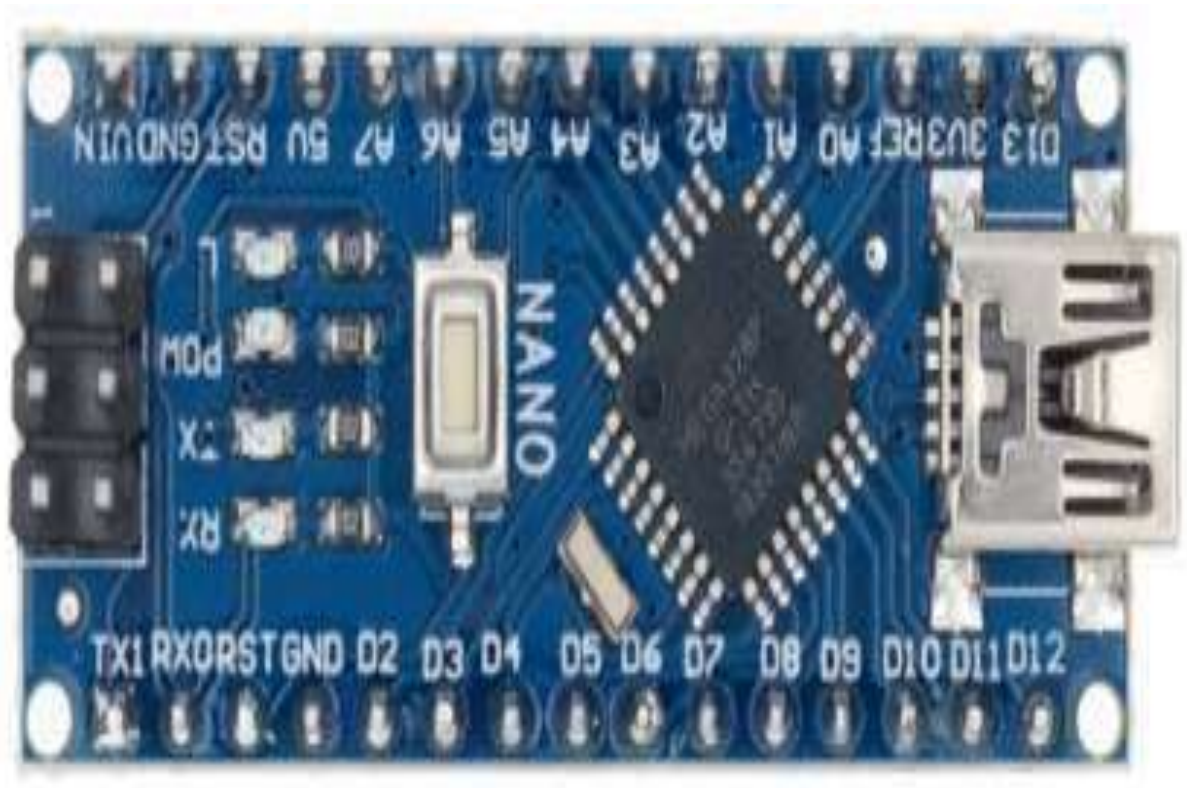


Fig 4.2. Arduino Nano

4.2.1. General Pin Function

- Digital pins 0-13: These are the digital input/output pins that can be used to interface with a variety of sensors, actuators, and other electronic components. These pins can be configured as inputs or outputs using software.
- Analog pins A0-A7: These are the analog input pins that can be used to read analog voltages from sensors, such as light sensors and temperature sensors.
- PWM pins 3, 5, 6, 9, 10, and 11: These are the pulse width modulation (PWM) pins that can be used to control the brightness of LEDs, the speed of motors, and other similar applications.
- Reset pin: This is the reset pin that can be used to reset the board or to put it into bootloader mode for programming.
- 5V and 3.3V pins: These are the voltage output pins that provide a regulated voltage of 5V and 3.3V, respectively, for powering external components.
- Ground pins: These are the ground pins that provide a common ground

reference for the board and external components.

- UART pins: These are the pins that provide a serial communication interface that can be used to communicate with other devices, such as sensors and displays.
- SPI pins: These are the pins that provide a SPI communication interface that can be used to communicate with other devices, such as SD cards and Ethernet modules.

4.2.2. *Special Pin Function*

- AREF pin: This pin is used to provide a reference voltage for the analog-to-digital converter (ADC) when using analog inputs. It can be used to set the maximum voltage for the ADC conversion, which can be useful for accurately measuring small voltage differences.
- I2C pins: The Arduino Nano has two pins (A4 and A5) that can be used for I2C communication. I2C is a popular communication protocol that is used to connect multiple devices together in a network.
- RX and TX pins: These pins are used for serial communication and can be used to send and receive data to and from other devices, such as a computer or another microcontroller.
- INT0 and INT1 pins: These pins can be used as interrupt pins, which allow the microcontroller to respond quickly to external events, such as a button press or a sensor reading.
- Analog reference voltage pins: The Arduino Nano has two pins (AREF and AREF-SEL) that can be used to set the reference voltage for the ADC. This can be useful for accurately measuring small voltage differences.
- ICSP pins: These pins are used for in-circuit serial programming (ICSP) and can be used to program the microcontroller using a programmer.

4.3. TEMP6000 SENSOR

Temp6000 is a non-invasive glucose monitoring sensor that uses infrared technology to measure glucose levels in the body. It is a promising technology that has the potential to revolutionize glucose monitoring, as it eliminates the need for painful and invasive blood tests. The Temp6000 sensor works by emitting infrared

light onto the skin, which is absorbed by the glucose in the interstitial fluid. The sensor then measures the temperature of the skin, which is directly related to the glucose concentration in the interstitial fluid. The sensor is able to calculate glucose levels based on the measured temperature and a calibration factor.

The Temp6000 sensor consists of a thin, flexible film that is placed on the skin and contains an array of temperature sensors. The sensors detect the temperature difference between the skin and the underlying tissues and transmit the data to a microcontroller or computer for processing. The sensor uses an algorithm that converts the temperature readings into glucose concentrations, which can be displayed on a screen or transmitted wirelessly to a mobile device. The sensor can be worn continuously, providing real-time glucose monitoring without the need for invasive blood sampling.

The Temp6000 sensor has several advantages over traditional invasive glucose monitoring methods, including:

- Non-invasive: The Temp6000 sensor does not require blood sampling, making it less painful and more convenient for patients.
- Continuous monitoring: The sensor can be worn continuously, providing real-time glucose monitoring and allowing for timely adjustments to treatment plans.
- Accurate: The Temp6000 sensor has been shown to provide accurate glucose readings, with comparable accuracy to traditional invasive methods.
- Portable: The sensor is small and portable, making it easy to use and carry around.

One of the advantages of the Temp6000 sensor is its non-invasiveness, which makes it more comfortable and convenient for patients. It also reduces the risk of infection and other complications associated with traditional blood glucose monitoring methods. However, there are still some challenges associated with the use of the Temp6000 sensor. One of the main challenges is the need for calibration, which can be affected by various factors such as skin temperature, humidity, and the presence of other substances in the interstitial fluid. Additionally, the accuracy of the sensor may be affected by individual differences in skin type and thickness. . Overall, the Temp6000 sensor shows promise as a non-invasive glucose monitoring

device that could greatly improve the quality of life for individuals with diabetes. With further research and development, it could become a valuable tool for diabetes management.



Fig 4.3. Temp6000 Sensor

4.3.1. Pin Function

The sensor has two pins, which are as follows:

- VCC - This pin is used to supply power to the sensor. It should be connected to a voltage source of 3.3V to 5V.
- OUT - This pin is the output of the sensor and provides an analog voltage that is proportional to the light intensity detected by the sensor. The voltage output ranges from 0V to the VCC voltage level. The output signal can be connected to an analog input pin of a microcontroller or ADC (Analog to Digital Converter) module.

It is important to note that the Temp6000 sensor requires an external resistor between the VCC and OUT pins to function properly. This resistor is typically a 10k ohm resistor that is connected in series between the VCC and OUT pins. This resistor sets the sensitivity of the sensor and should be chosen based on the specific

application requirements.

4.4. ADXL335 SENSOR

The ADXL335 is a 3-axis accelerometer sensor that is used to measure acceleration in three perpendicular axes. The sensor measures acceleration up to $\pm 3g$ (g being the acceleration due to gravity) and outputs an analog voltage proportional to the acceleration measured along each axis. This sensor can be used in a wide range of applications, such as motion and tilt sensing, robotics, and gaming.

The ADXL335 sensor works on the principle of measuring the change in capacitance of three small, flexible, and thin-film structures, known as "proof masses", that are suspended between two fixed plates. These proof masses are connected to the plates by springs, and when the sensor experiences acceleration, the proof masses move relative to the fixed plates, causing a change in the capacitance between them. This change in capacitance is converted into a voltage by the sensor's internal electronics, which is proportional to the acceleration along each axis.

The sensor includes a built-in signal conditioning circuitry that filters and amplifies the analog voltage outputs from the XOUT, YOUT, and ZOUT pins. The sensor also includes a self-test feature that can be used to verify that the sensor is functioning correctly. To use the ADXL335 sensor with a microcontroller, the analog voltage outputs from the XOUT, YOUT, and ZOUT pins should be connected to the analog input pins of the microcontroller. The microcontroller can then use the appropriate analog-to-digital converter (ADC) function to read the analog voltage and convert it into an acceleration value. The sensitivity and other settings of the sensor can be adjusted using software to match the specific requirements of the application.

4.4.1 ADXL335 Pin Diagram

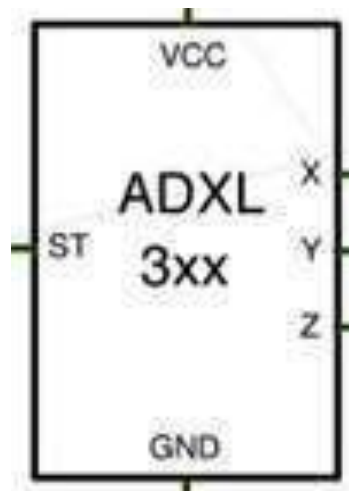


Fig 4.4. ADXL335 Pin Diagram

4.4.1. Pin Function

- VCC: This is the power supply pin for the sensor. It should be connected to a voltage supply between 1.8V to 3.6V.
- GND: This is the ground pin for the sensor. It should be connected to the ground of the microcontroller or power supply.
- XOUT: This pin outputs an analog voltage corresponding to the acceleration measured in the X-axis.
- YOUT: This pin outputs an analog voltage corresponding to the acceleration measured in the Y-axis.
- ZOUT: This pin outputs an analog voltage corresponding to the acceleration measured in the Z-axis.
- ST: This pin is used to set the sensitivity of the sensor. It can be left unconnected or connected to VCC or GND, depending on the desired sensitivity setting.

The XOUT, YOUT, and ZOUT pins output a voltage proportional to the acceleration measured along the corresponding axis. The voltage output is typically centered around a voltage of $VCC/2$, which means that the acceleration due to gravity along that axis will be represented by a voltage of $VCC/2$. The voltage output can be converted into an acceleration value using the appropriate formula and calibration.

4.5. MAX30100 SENSOR

The MAX30100 sensor is a high-sensitivity pulse oximeter and heart-rate sensor designed for use in a wide range of applications, such as fitness and wellness monitoring, medical diagnostics, and wearable devices. The sensor uses two LEDs and a photodetector to measure the absorption of light by blood vessels, which can then be used to determine the oxygen saturation level and heart rate of the user.

The MAX30100 sensor includes a built-in algorithm that performs signal processing and noise reduction to ensure accurate and reliable measurements. The sensor also includes an interrupt pin that can be used to alert the microcontroller when a new measurement is available, and a FIFO buffer that can be used to store multiple measurements for later processing.

To use the MAX30100 sensor with a microcontroller, the SDA and SCL pins should be connected to the corresponding I2C pins of the microcontroller. The LED control and photodetector output pins should be connected to the appropriate control and input pins of the microcontroller, and the interrupt pin should be connected to an external interrupt pin of the microcontroller if needed.

4.5.1. MAX30100 Pin Diagram

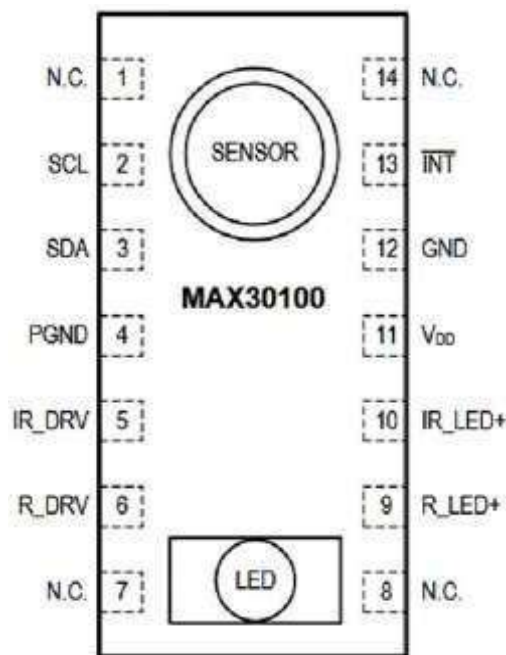


Fig 4.5. MAX30100 Pin Diagram

4.5.2. Pin Function

- VIN: This is the power supply pin for the sensor, and should be connected to a voltage supply between 1.8V and 5.5V.
- GND: This is the ground pin for the sensor.
- SDA: This is the serial data input/output pin for the I2C interface.
- SCL: This is the serial clock input pin for the I2C interface.
- INT: This is the interrupt output pin, which can be used to indicate that a new measurement is available or to signal other events.
- LED1: This is the anode pin of the first LED, typically a red LED.
- LED2: This is the anode pin of the second LED, typically an infrared LED.
- PD: This is the photodiode output pin, which outputs the intensity of light detected by the photodiode.

4.6. GSR SENSOR

GSR (Galvanic Skin Response) sensor, also known as skin conductance sensor, is a type of sensor that measures the electrical conductance of the skin. It is commonly used in various applications such as lie detection, stress monitoring, and biofeedback. The GSR sensor typically consists of two electrodes that are placed on the skin. When a voltage is applied across the electrodes, a small current flows through the skin, and the resistance of the skin determines the magnitude of this current. The GSR sensor measures this current and converts it into a voltage signal that can be processed by a microcontroller or other device.

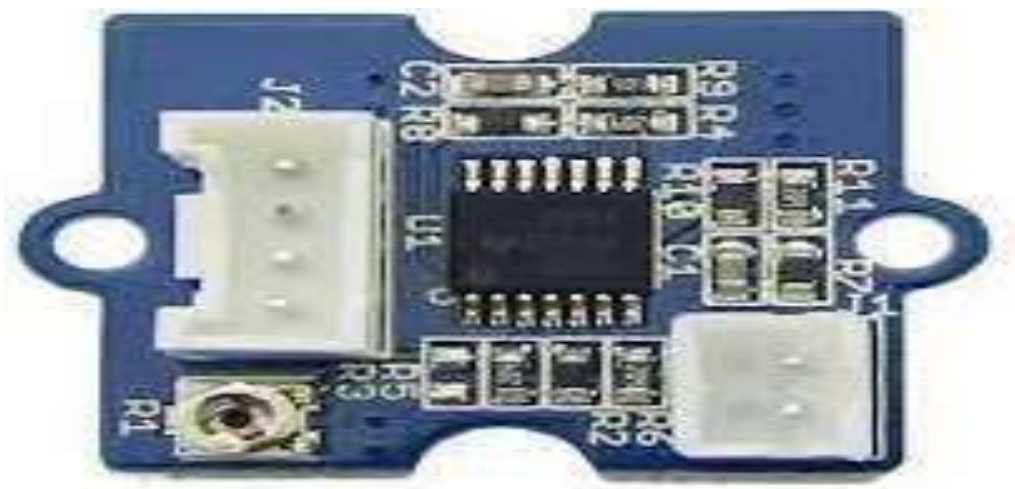


Fig 4.6. GSR Sensor

The GSR sensor can be used in two modes: DC and AC. In DC mode, a constant voltage is applied across the electrodes, and the resulting current is measured. In AC mode, a high-frequency AC voltage is applied across the electrodes, and the resulting current is measured. AC mode is generally more sensitive than DC mode, but it is also more prone to noise and interference. The GSR sensor typically has two output pins: a signal pin and a ground pin. The signal pin outputs the voltage signal generated by the sensor, while the ground pin is connected to the ground of the system. Some GSR sensors may also have additional pins for calibration or temperature compensation.

4.6.1. Pin Function

- Signal Pin: This pin is used to output the voltage signal generated by the sensor. The voltage signal is proportional to the electrical conductance of the skin and can be measured and processed by a microcontroller or other device.
- Ground Pin: This pin is connected to the ground of the system to provide a reference for the signal pin.

Some GSR sensors may also have additional pins for calibration or temperature compensation, but the signal and ground pins are the two essential pins required for measuring skin conductance.

The GSR sensor can be connected to a microcontroller or other device using analog or digital input pins. The voltage signal generated by the GSR sensor is typically in the range of a few millivolts to several volts, depending on the design of the sensor and the level of electrical conductance of the skin being measured. The signal can be amplified and filtered as needed to obtain accurate measurements and reduce noise and interference.

4.7. ARDUINO IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

Arduino IDE (Integrated Development Environment) is a software application used to program and develop software for Arduino microcontrollers. It is an open-source platform that allows users to write, compile, and upload code to Arduino boards.

Arduino IDE provides a user-friendly interface that simplifies the programming process for beginners and advanced users alike. It has a text editor for writing code, a serial monitor for debugging, and a library manager for accessing a wide range of pre-built code libraries.

The IDE supports C and C++ programming languages and comes with a set of pre-installed libraries that can be used to interface with various sensors, actuators, and other components. Users can also download additional libraries from the Arduino library manager or create their own libraries to reuse code in different projects. Arduino IDE supports multiple operating systems, including Windows, Mac OS, and Linux, and it can be used with various Arduino boards, such as the Arduino Uno, Mega, Nano, and Due.

Arduino IDE v1 is an integrated development environment (IDE) used for programming and developing software for Arduino microcontrollers. It is an open-source platform that provides a user-friendly interface for writing, compiling, and uploading code to Arduino boards. Arduino IDE v1 is based on the Processing programming language and uses the GCC compiler toolchain for compiling code. It supports C and C++ programming languages and comes with a set of pre-installed libraries that can be used to interface with various sensors, actuators, and other components.

The IDE has a simple and intuitive interface, making it easy for beginners to get started with Arduino programming. It includes a text editor for writing code, a serial monitor for debugging, and a library manager for accessing a wide range of pre-built code libraries. Arduino IDE v1 is compatible with a wide range of Arduino boards, including the Arduino Uno, Mega, Nano, and Due. It supports multiple operating systems, including Windows, Mac OS, and Linux.

One of the strengths of Arduino IDE v1 is its ease of use, which makes it a popular choice for beginners and hobbyists. The platform has a large and active community that provides support, tutorials, and code examples, making it easy to get started with Arduino programming. However, one limitation of Arduino IDE v1 is its lack of more advanced features, such as code analysis and refactoring tools, which can make programming more efficient and streamlined. Nevertheless, Arduino IDE v1

remains a popular choice for many developers due to its simplicity, stability, and wide range of library support.

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 serialmonitor.

➤ ***IDE VERIFY File***

Verify Checks the code for errors compiling it.

➤ ***IDE UPLOAD File***

Upload Compiles the code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer".

➤ ***IDE NEW File***

New Creates a new sketch.

➤ ***IDE OPEN File***

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

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

➤ **IDE SAVE File**

Save-Saves the sketch.

➤ **IDE SERMON File**

Serial Monitor-Opens the serial monitor. Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

➤ **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. Examples: Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

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 coloring.

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 the 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**

Auto Format-This formats the code nicely: i.e. 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 the 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**

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

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 the programs (or sketches). The sketches in the 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.

➤ **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 the 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. You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

4.7.1. Arduino IDE Processing

The Arduino IDE is primarily used for writing and uploading code to Arduino microcontrollers. However, it also includes a software sketchpad called Processing that can be used for creating interactive graphics, animations, and visualizations. Processing is an open-source programming language and development

environment that was initially developed for creating visual arts and graphics, but has since been expanded to include other applications such as data visualization and educational tools. To use Processing in the Arduino IDE, users need to first enable it by installing the "Arduino Create Agent" on their computer.

Once installed, users can access Processing by opening the Arduino IDE and navigating to the "Tools" menu. From there, they can select "Arduino Create Editor" and then "Processing" to launch the Processing sketchpad. In the Processing sketchpad, users can write code using the Processing language, which is similar to Java. The sketchpad includes various tools and libraries that make it easy to create interactive graphics, animations, and visualizations. Users can also interface with the Arduino microcontroller by using the "Serial" library to send and receive data between the Arduino and the Processing sketchpad. Overall, the integration of Processing in the Arduino IDE provides users with a powerful tool for creating interactive projects that combine physical computing and visualizations.

The programming method for the proposed project involves writing code for ESP32 and Arduino Nano microcontrollers to collect and process data from various sensors and display the results on an IoT Blynk application. The code is written in the Arduino Integrated Development Environment (IDE) using the C++ programming language. The first step in the programming method is to set up the Arduino IDE and install the necessary libraries for the sensors and Blynk application. Then, the pins to which the sensors are connected are defined and initialize them in the `setup()` function. Next, the code to collect data from each sensor are written and store it in a variable. For example, for the accelerometer sensor, we can collect data on the number of steps taken and calculate the calorie output based on the step count. Similarly, for the GSR sensor, we can collect data on the amount of sweat produced, which can be an indicator of stress or exercise intensity. Finally, the data is displayed on the IoT Blynk application in real-time. The Blynk application is configured to receive data from the microcontroller and display it in a user-friendly format, such as graphs or charts.

4.8. BLYNK APP

Blynk is a mobile app that allows users to control and monitor various internet of things (IoT) devices, sensors, and other connected devices. The app provides an

easy-to-use interface for creating custom dashboards, controlling devices, and visualizing data. The Blynk app uses a cloud-based server to connect the user's mobile device to the IoT devices. Users can choose from a variety of communication protocols, including Wi-Fi, Ethernet, Bluetooth, and cellular data. The app provides an extensive library of pre-built widgets, including buttons, sliders, graphs, and gauges, that users can drag and drop onto their dashboard to create a customized interface.

One of the primary features of the Blynk app is the ability to create custom notifications and alerts. Users can set up triggers based on sensor readings or other events and receive push notifications when the trigger is activated. This feature is particularly useful for monitoring and controlling devices remotely. Blynk also provides an API that allows developers to integrate their IoT devices with the app. This API includes libraries for various programming languages, including Arduino, Raspberry Pi, and ESP8266. The app can also integrate with popular IoT platforms, such as Amazon Web Services (AWS) and Google Cloud Platform (GCP).

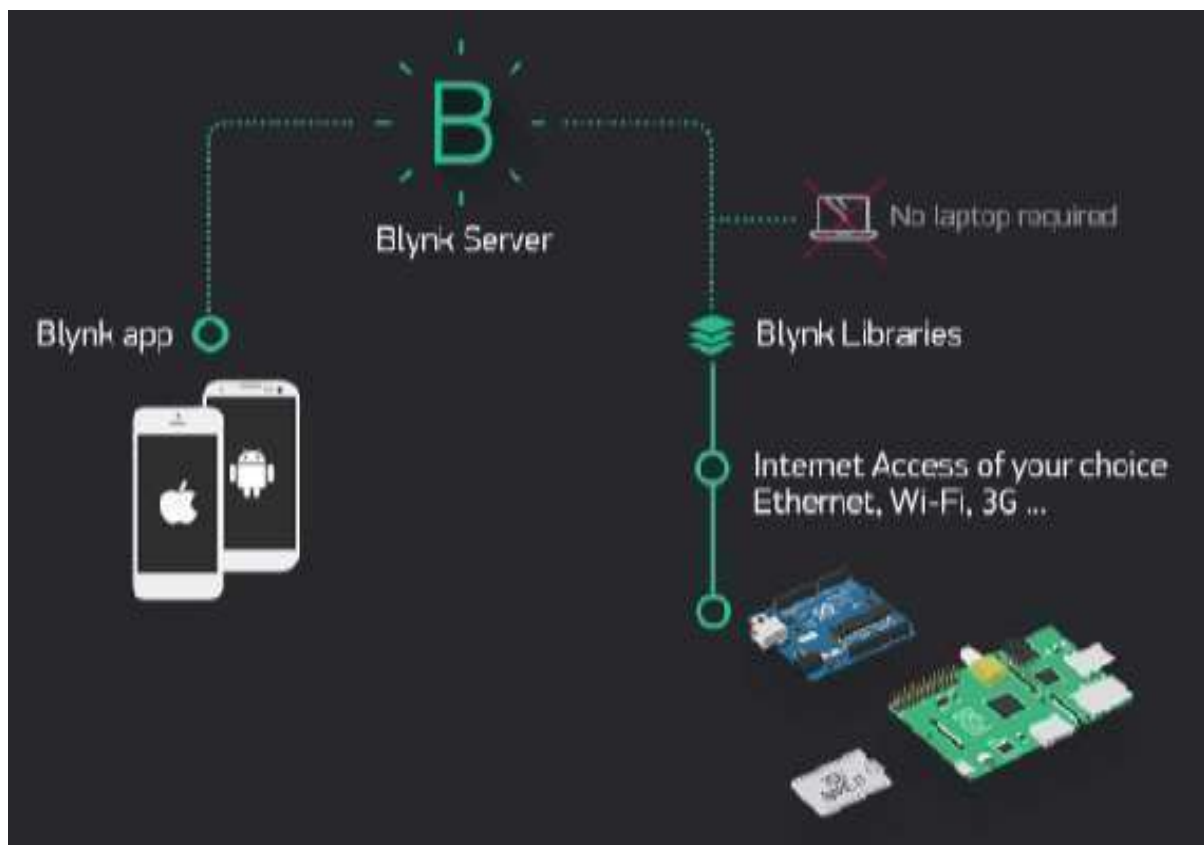


Fig.4.7 Working of Blynk app

The Blynk app works by providing an interface for users to control and monitor various IoT devices and sensors. The app communicates with the devices through a cloud-based server, which acts as an intermediary between the mobile device and the IoT devices. When a user creates a new project in the Blynk app, they can choose from a variety of communication protocols, including Wi-Fi, Ethernet, Bluetooth, and cellular data. They can then drag and drop various widgets onto the app's dashboard to create a customized interface for controlling and monitoring their devices. Each widget in the Blynk app is associated with a specific function or task, such as turning a device on or off, adjusting a variable, or displaying sensor data. When the user interacts with a widget, the app sends a command to the cloud-based server, which then forwards the command to the corresponding IoT device.

The Blynk app also allows users to create custom notifications and alerts based on sensor readings or other events. For example, a user could set up an alert to notify them when a temperature sensor detects a temperature above a certain threshold. The app will send a push notification to the user's mobile device when the alert is triggered. In addition to its built-in widgets and features, the Blynk app also provides an API for developers to integrate their IoT devices with the app. This API includes libraries for various programming languages, such as Arduino, Raspberry Pi, and ESP8266. Overall, the Blynk app provides an easy-to-use and flexible interface for controlling and monitoring IoT devices. It simplifies the process of creating custom dashboards and notifications, making it an excellent tool for both hobbyists and professionals.

4.9. METHODOLOGY

Methodology in advanced social IoT for real-time monitoring of diabetic patients in the healthcare system involves the systematic and structured approach to designing, developing, and implementing a comprehensive IoT-based solution that enables continuous monitoring of diabetic patients. An advanced social IoT system for real-time monitoring of diabetic patients in healthcare systems requires a comprehensive methodology that takes into account the needs of patients, healthcare providers, and the healthcare system. The methodology of Advanced Social IoT (ASIoT) for real-time monitoring of diabetic patients in a healthcare system can be broadly divided into the following steps:

- Requirement Analysis: The first step in developing an ASIOT system for real-time monitoring of diabetic patients is to conduct a thorough analysis of the requirements of the system. This involves identifying the needs of the patients, healthcare providers, and other stakeholders, as well as the technical requirements of the system.
- System Design: The system architecture is designed and select the appropriate sensors to be used in the system. The system design will include the placement of sensors on the patient's body and the communication protocol between the sensors and the microcontrollers.
- Sensor Selection: Once the requirements have been identified, the next step is to select the appropriate sensors for monitoring the health of the patients. In the case of diabetic patients, this may include glucose sensors, heart rate monitors, activity trackers, and other medical devices.
- IoT Device Selection: The sensors selected in the previous step are then connected to IoT devices such as wearables or smartphones that are worn by the patients. These devices must be selected based on their compatibility with the selected sensors, as well as their reliability, ease of use, and other factors.
- Hardware Integration: In this step, the selected sensors with ESP32 and Arduino Nano microcontrollers are integrated. The microcontrollers collect data from each sensor and process it to obtain meaningful information about the patient's health parameters.
- Code Development: The code for ESP32 and Arduino Nano microcontrollers are developed to collect and process data from each sensor and display it on the IoT Blynk application. The codes were written in the Arduino IDE using the C++ programming language.
- Testing: After developing the code, the system is tested by monitoring the health parameters of diabetic patients in real-time. A reliability and performance analysis of the system is performed.
- Deployment: Once the system has been tested and validated, the system for use by diabetic patients is deployed. The system will be made portable and compact, allowing patients to carry it with them wherever they go.

- **Cloud-based Server Setup:** The data collected from the sensors is transmitted to a cloud-based server, which must be set up with appropriate database systems, machine learning algorithms, and other technologies required for real-time data processing and analysis.
- **Mobile Application Development:** A mobile application must be developed for patients to access their health data, receive alerts, and communicate with their healthcare providers. The application must be designed with a user-friendly interface and must be compatible with the sensors and IoT devices selected in earlier steps.
- **Dashboard Development:** A real-time dashboard must be developed for healthcare providers to monitor the health status of patients, view alerts, and make treatment recommendations. The dashboard must be designed to be easy to use and customizable to meet the needs of different healthcare providers.
- **Implementation and Deployment:** Once the ASIOT system has been developed and tested, it must be implemented and deployed in the healthcare system. This involves training healthcare providers and patients on how to use the system, as well as integrating the system with existing healthcare infrastructure. The system will be made portable and compact, allowing patients to carry it with them wherever they go.
- **Maintenance and Support:** The ASIOT system must be maintained and supported to ensure that it continues to function effectively over time. This involves monitoring the system for errors or bugs, providing technical support to users, and updating the system with new features and capabilities as needed.

The methodology for ASIOT for real-time monitoring of diabetic patients in a healthcare system involves a range of technical and non-technical activities, including requirements analysis, sensor selection, cloud-based server setup, mobile application development, dashboard development, data analytics, implementation and deployment, and maintenance and support.

4.9.1. Work Flow

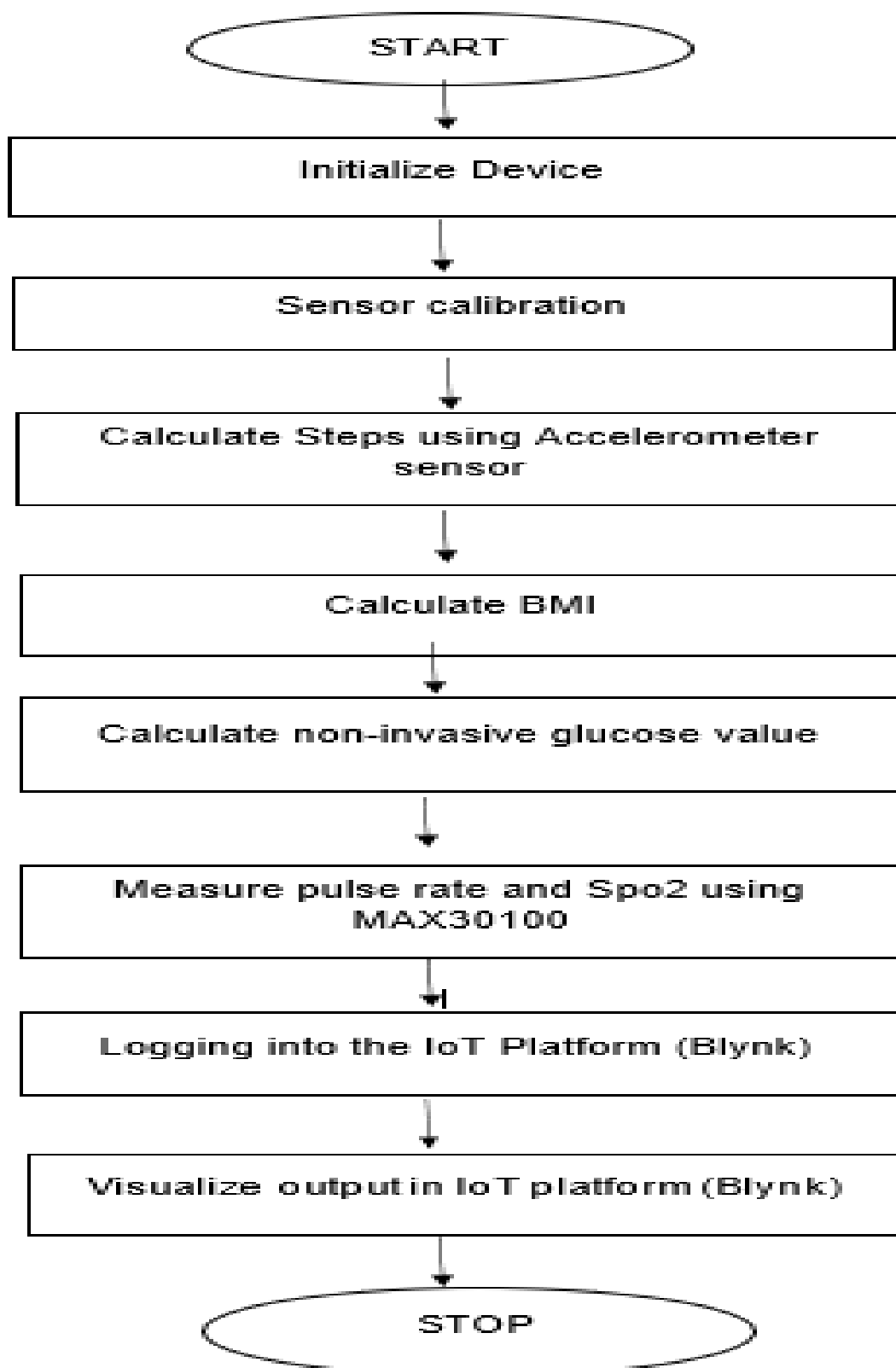


Fig. 4.8 Work Flow Diagram

The workflow for a system that includes the initialization of a device, calibration of sensors, calculation of vital signs, non-invasive glucose level monitoring, logging into IoT, and visualization of output in IoT, stop the device can be broken down into the following steps:

- **Start:** The system starts with the power-up of the device and the initialization of the software.
- **Initialize the device:** The device is initialized, and the sensors are checked to ensure that they are properly connected and functioning.
- **Sensors calibration:** The sensors are calibrated to ensure accurate readings. Calibration may involve adjusting sensor settings or comparing sensor readings to known standards.
- **Calculate steps:** The system calculates the user's daily step count using the data from the accelerometer sensor.
- **Calculate BMI:** The user's weight and height have to be noted manually and it calculates the BMI (Body Mass Index) of the user.
- **Calculate Non-invasive glucose level:** The system uses the Temp6000 sensor to measure glucose levels non-invasively. This involves shining light on the skin and measuring the reflected light to determine glucose levels.
- **Calculate Pulse rate and SpO2:** The system measures the user's pulse rate and SpO2 using a MAX30100 sensor. The sensor uses light to measure the user's heart rate and oxygen saturation levels.
- **logging into IoT:** The system logs all the sensor data and vital signs measurements onto an IoT platform for real-time monitoring.
- **Visualize output in IoT:** The logged data is visualized using graphs and charts on the IoT platform, allowing healthcare providers to monitor the user's health in real-time.
- **Stop:** The system shuts down or enters a low-power mode to conserve battery life when not in use.

This workflow provides a comprehensive approach to real-time monitoring of diabetics patients in a healthcare system using advanced social IoT technology. By incorporating multiple sensors and logging vital signs data onto an IoT platform,

healthcare providers can more effectively monitor patients and provide timely interventions as needed.

The workflow diagram for ASIOT for real-time monitoring of diabetics patients in a healthcare system shows how IoT technology can be used to improve the quality of care for patients with chronic conditions like diabetes. By collecting and analyzing real-time health data, healthcare providers can make informed decisions about patient care and help patients better manage their condition.

CHAPTER-5

RESULTS AND DISCUSSION

5.1. RESULTS

A non-invasive glucose detection (Temp6000 sensor), GSR sensor, heart rate and SpO2 sensors, accelerometer sensor, ESP32 and Arduino nano were used to collect readings of glucose levels, heart rate, Spo2, sweat monitoring and physical activity. The result of this project yielded promising results. The system was able to collect and transmit real-time data points such as glucose levels, heart rate, Spo2, step count, calories output, sweat monitoring and BMI calculation to healthcare providers and carers. This allowed healthcare providers to remotely monitor the patients' conditions and adjust the treatment plan as needed.

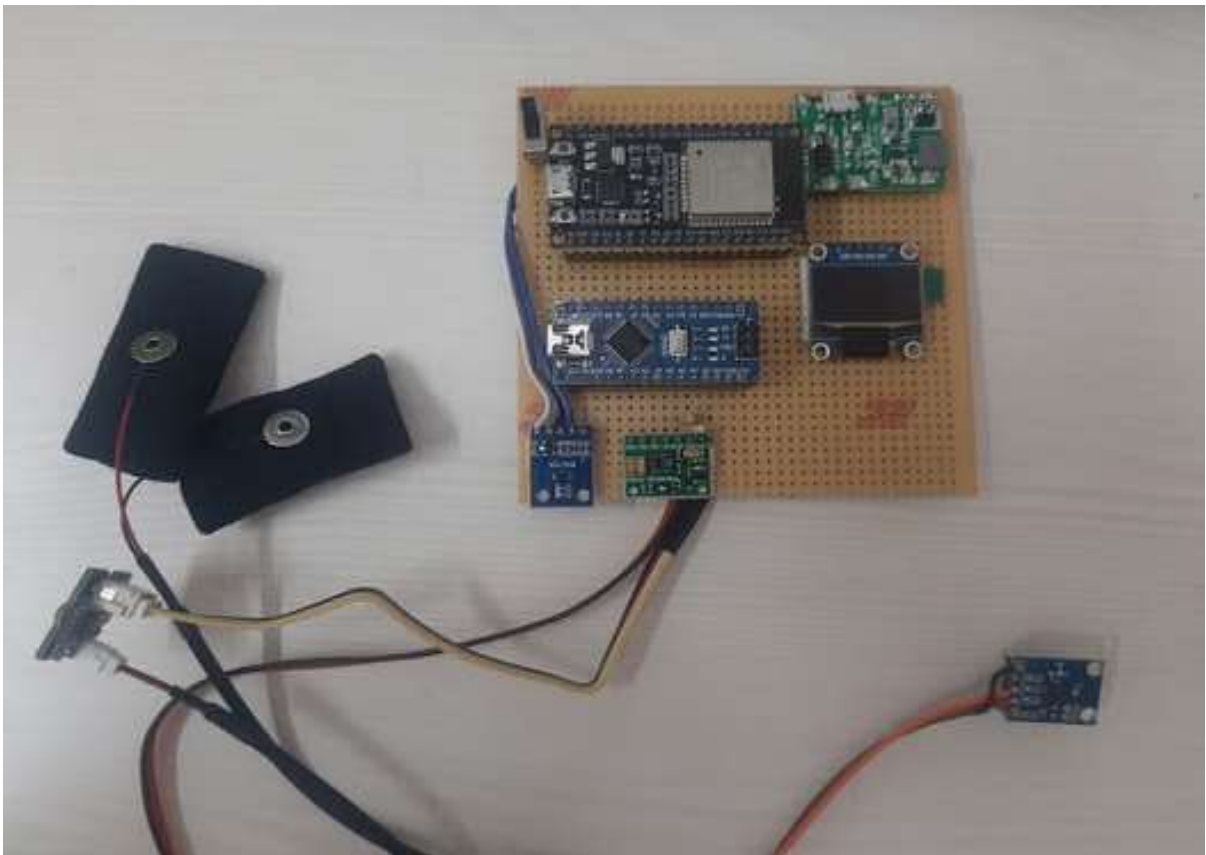


Fig.5.1. Prototype of Health Monitoring Device

Figure 5.1 represents the prototype of the Health Monitoring device. The principle components of the system that is Arduino Nano, ESP32, Temp6000 (noninvasive glucose sensor), heart rate & Spo2 sensor, GSR sensor and accelerometer sensor

have been connected. The inputs are obtained from the sensors and the output is obtained in both the prototype and IoT platform.

5.1.1. Prototype Output

Heart rate, Spo2, GSR output, glucose level, steps calculation and calories burn are calibrated using the microcontroller. The output values of the sensors calculated and displayed in the OLED, so that these values are visible to the users. The parameters include heart rate, Spo2, GSR output, glucose level, steps calculation and calories burn are displayed in the OLED of the prototype.

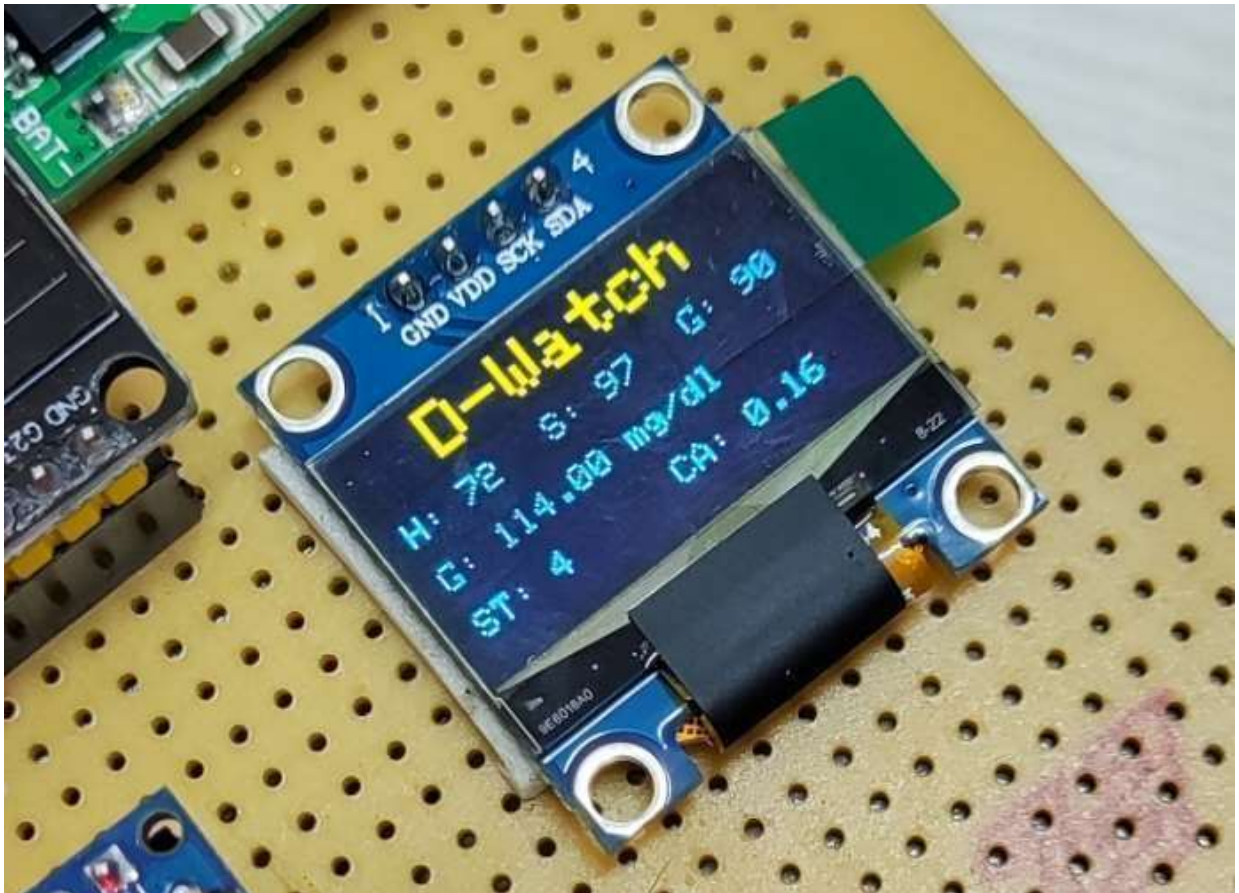


Fig.5.2. Display of the Health Monitoring Device

Fig.5.2. represents the display of the health monitoring device. In the OLED display, H, S, G, G, ST and CA denotes the measurements of Heart rate, Spo2, GSR output, Glucose level, Steps calculation and calories burn respectively. The below table 5.1 shows the measurand values (i.e) displayed in OLED.

Table 5.1. Prototype Output

Parameters	Measurand
Heart rate	72 bpm
Spo2	97%
GSR	90 μ S
Glucose level	114.0 mg/dl
Steps calculation	4
Calories burn	0.16 cal

5.1.3. Blynk app Output



Fig 5.3. Blynk app Output

The obtained output is displayed on the OLED display and additionally with the aid of IoT-cloud based app (blynk) the doctor can monitor real time health data. The output is displayed on Blynk android Application. The screenshot of the result of the designed system obtained on android application (Blynk app) are shown in Figure 5.3. The figure shows the measurands of Heart rate, Spo2, GSR, BMI calculation, steps count, calories and Blood glucose. The below table 5.3 shows the measurand values (i.e) obtained in Blynk app.

Table 5.2. Blynk app output

Parameters	Measurand
Heart rate	47 bpm
Spo2	96%
GSR	87 μ S
BMI	Healthy weight
Steps count	5
Calories	0.20 cal
Blood Glucose	114 mg/dl

5.2. DISCUSSION

Using advanced social IoT (Internet of Things) to monitor diabetic patients in real time can provide several advantages. The system can be configured to collect various data points such as glucose levels, heart rate, Spo2, physical activity, and GSR output which can then be transmitted to healthcare providers and carers in real time. According to Alekya et.al., (2021) the cloud app, which is based on IoT and its design principles, allows for direct communication of sensor devices while also being versatile and effective in serving stored data, users, and sensors.

This allows healthcare providers to remotely monitor and adjust the patient's treatment plan, potentially improving care quality and reducing hospitalisations. The system can also be configured to enable patient empowerment and self-management. This has the potential to improve patient engagement and adherence to treatment plans, which is a major challenge in diabetes management.

Furthermore, the system can make use of cutting-edge social IoT technologies to build a supportive community of patients and carers. This can offer patients emotional support, education, and personalized peer-to-peer coaching to help them improve their self-management skills. The community can also be used for real-time social networking, allowing patients to connect with others who are dealing with similar issues and share their experiences. However, there are some issues that must be addressed when putting such a system in place. For example, patient data privacy and security must be ensured. To protect sensitive patient information, the

system must be designed to comply with regulations such as HIPAA (Health Insurance Portability and Accountability Act).

Moreover, the system must be designed to accommodate patients with varying levels of technology literacy, which can be difficult in the elderly. To enable widespread adoption, the system must also be scalable and cost-effective. Finally, using advanced social IoT to monitor diabetic patients in real time has the potential to improve patient outcomes, increase patient empowerment, and create a supportive community of patients and carers. However, in order to reap these benefits, the system must address issues such as privacy and security, technology literacy, scalability, and cost-effectiveness.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1. SUMMARY

The proposed project used ESP32 and Arduino Nano microcontrollers to create an effective health monitoring tool for diabetic patients. The implemented IoT-based architecture is a complete system, beginning with a sensor node and ending with a back-end server. Doctors and carers can easily monitor their patients using the system at any time and from any location by using a browser or a smartphone application. The system includes a variety of sensors to track various health parameters such as step count, calorie output, sweat monitoring, non-invasive glucose detection, heart rate, and BMI calculation. An accelerometer sensor is used to monitor the patient's step count and calorie output, which calculates the number of steps taken and then calculates the calories burned based on this information. Sweat monitoring is performed using a GSR sensor, which measures the amount of sweat produced by the patient's body and helped to predict stress or exercise intensity. A Temp6000 sensor was used for non-invasive glucose detection, which measures glucose levels in the patient's blood without the need for blood sampling. This sensor measures glucose levels in skin tissue using infrared light. Dedicated sensors connected to the Arduino Nano microcontroller monitor the heart rate and SpO2 levels. The programme computes the BMI based on the patient's height and weight. The system was built on the Arduino IDE with the C++ programming language. The system's sensors were used to collect and wirelessly transmit the data (non-invasive glucose detection, heart rate, step count, calorie output, sweat monitoring, BMI calculation and physical activity). The sensors were linked to the Arduino Nano processor, where the data is acquired and processed. The processor processes the data collected by the sensors, and the processed data was sent to the cloud via a Wi-Fi module. All of these parameters are effectively monitored by an IoT Blynk application, which displays real-time data and it was viewed on the webpage using a computer and parallel with a mobile device and the sensor output is also displayed on the OLED. Overall, this health monitoring tool allowed diabetic patients to effectively monitor their health parameters and took the necessary actions to manage their health condition. The samples were also taken remotely and the glucose parameter was demonstrated in real-time monitor. The device is

also user-friendly and safe for the human body.

6.2. CONCLUSION

A real-time health monitoring system based on IoT made life easier for all patients because they were able to measure various health parameters automatically. This device is intended to not only save money and time, but also to reduce the risk of the patient's health deteriorating due to travel. The project is a successful healthcare monitoring system which can be used to wireless monitor for the continuous verification of the health of patients. The effective health monitoring system was made by the help of IoT and wireless sensor technology. The device uses inputs from the patient and perform medical tests digitally and display the results on an OLED display and a mobile phone app. Sensor data were collected and an information was saved to view on the mobile application like Blynk. This system reduced patient waiting time by producing immediate results. This project also helped the doctors to monitor patient health from any location at any time, which is very important in health care system. Though the project is very useful for patient and doctor but in future I would like to improve this with Big data analysis so that the patient's health history can be transferred to any part of the world for any future health complicity.

REFERENCES

1. Alarcón-Paredes, Antonio, Victor Francisco-García, Iris P. Guzmán-Guzmán, Jessica Cantillo-Negrete, René E. Cuevas-Valencia, and Gustavo A. Alonso-Silverio. "An IoT-based non-invasive glucose level monitoring system using raspberry pi." *Applied Sciences* 9, no. 15 (2019): 3046.
2. Alekya, R., Neelima Devi Boddeti, K. Salomi Monica, R. Prabha, and V. Venkatesh. "IoT based smart healthcare monitoring systems: A literature review." *Eur. J. Mol. Clin. Med* 7 (2021): 2020.
3. Alfandi, Omar. "An Intelligent IoT Monitoring and Prediction System for Health Critical Conditions." *Mobile Networks and Applications* (2022): 1-12.
4. Alfian, Ganjar, Muhammad Syafrudin, Muhammad Fazal Ijaz, M. Alex Syaekhoni, Norma Latif Fitriyani, and Jongtae Rhee. "A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing." *Sensors* 18, no. 7 (2018): 2183.
5. AlShorman, Omar, Mahmoud Saleh Masadeh, and Buthaynah AlShorman. "Mobile health monitoring based studies for diabetes mellitus: a review." *Bulletin of Electrical Engineering and Informatics* 10, no. 3 (2021): 1405-1414.
6. Barata, José Jorge Rodrigues, Roberto Munoz, Rafael D. De Carvalho Silva, Joel JPC Rodrigues, and Victor Hugo C. De Albuquerque. "Internet of Things based on electronic and mobile health systems for blood glucose continuous monitoring and management." *IEEE Access* 7 (2019): 175116-175125.
7. Efat, Md Iftekharul Alam, Shoaib Rahman, and Tasnim Rahman. "IoT based smart health monitoring system for diabetes patients using neural network." In *Cyber Security and Computer Science: Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, February 15-16, 2020, Proceedings 2*, pp. 593-606. Springer International Publishing, 2020.
8. Feng, Rui, Zhanxiao Geng, Ang Li, Yao Chu, Chenyang Wu, Ning Zhang, Xiaohui Guo, Junqing Zhang, Xiaohao Wang, and Fei Tang. "Noninvasive blood glucose monitor via multi-sensor fusion and its clinical evaluation." *Sensors and Actuators B: Chemical* 332 (2021): 129445.
9. Gia, Tuan Nguyen, Mai Ali, Imed Ben Dhaou, Amir M. Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. "IoT-based continuous glucose monitoring system: A feasibility study." *Procedia Computer*

- Science* 109 (2017): 327-334.
10. Haxha, Shyqyri, and Jaspreet Jhoja. "Optical based noninvasive glucose monitoring sensor prototype." *IEEE Photonics Journal* 8, no. 6 (2016): 1-11.
 11. Jamal, Amr, Shabana Tharkar, Weam Saleh Babaier, Shrooq Faisal Alsomali, Allulu Saad Alsulayhim, Monera Abdulkareem Alayuni, Nada Abdulaziz Aldakheel, Safa Sultan Al-Osaimi, Norah Alshehri, and Mohammed Batais. "Blood Glucose Monitoring and Sharing Amongst People With Diabetes and Their Facilitators: Cross-sectional Study of Methods and Practices." *JMIR diabetes* 6, no. 4 (2021): e29178.
 12. Kamusheva, Maria, Konstantin Tachkov, Maria Dimitrova, Zornitsa Mitkova, Gema García-Sáez, M. Elena Hernando, Wim Goettsch, and Guenka Petrova. "A systematic review of collective evidences investigating the effect of diabetes monitoring systems and their application in health care." *Frontiers in endocrinology* 12 (2021): 636959.
 13. Longva, Anne Marit, and Moutaz Haddara. "How can IoT improve the life-quality of diabetes patients?." In *MATEC web of conferences*, vol. 292, p. 03016. EDP Sciences, 2019.
 14. Nivetha, K., N. Ramya, R. Thendral, and A. Gopikrishnan. "Blood glucose measurement by sweat using Arduino." *BLOOD* 4, no. 2 (2018).
 15. Olokoba, Abdulfatai B., Olusegun A. Obateru, and Lateefat B. Olokoba. "Type 2 diabetes mellitus: a review of current trends." *Oman medical journal* 27, no. 4 (2012): 269.
 16. Pai, Praful P., Pradyut K. Sanki, Sudeep K. Sahoo, Arijit De, Sourangshu Bhattacharya, and Swapna Banerjee. "Cloud computing-based non-invasive glucose monitoring for diabetic care." *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, no. 2 (2017): 663-676.
 17. Rahman, Ruhani Ab, Nur Shima Abdul Aziz, Murizah Kassim, and Mat Ikram Yusof. "IoT-based personal health care monitoring device for diabetic patients." In *2017 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 168-173. IEEE, 2017.
 18. Rahmat, MA Aizat, E. L. M. Su, M. Mohd Addi, and C. F. Yeong. "GluQo: IoT-based non-invasive blood glucose monitoring." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 9, no. 3-9 (2017): 71-75.

19. Ramachandran, A. "Know the signs and symptoms of diabetes." *The Indian journal of medical research* 140, no. 5 (2014): 579.
20. Rghioui, Amine, Assia Naja, Jaime Lloret Mauri, and Abedlmajid Oumnad. "An IOT based diabetic patient monitoring system using machine learning and node MCU." In *Journal of Physics: Conference Series*, vol. 1743, no. 1, p. 012035. IOP Publishing, 2021.
21. Ruwaida Ramly, Aznida Abu Bakar Sajak and Mohammad Syazwan Syahmi, "IOT-based glucose monitoring system," Aug 2020.
22. Saijo, Yuto, Hiroshi Okada, Masahide Hamaguchi, Momoko Habu, Kazushiro Kurogi, Hiroaki Murata, Masato Ito, and Michiaki Fukui. "The Risk Factors for Development of Type 2 Diabetes: Panasonic Cohort Study 4." *International Journal of Environmental Research and Public Health* 19, no. 1 (2022): 571.
23. Saravanan, M., and R. Shubha. "Non-invasive analytics based smart system for diabetes monitoring." In *International Conference on IoT Technologies for HealthCare*, pp. 88-98. Springer, Cham, 2017.
24. Shalini.R, Madhumita.M, ManojKumar.A.S, Pratheba.D and Sangeetha.K, "An IoT Based glucose monitoring system", *Journal of Biotechnology and Biomaterials*, 2019.
25. Sridevi, Parama, ASM Shamsul Arefin, and Abu Shahadat Md Ibrahim. "A Feasibility Study of Non-invasive Blood Glucose Level Detection Using Near-Infrared Optical Spectroscopy." *Bangladesh Journal of Medical Physics* 14, no. 1 (2021): 1-13.
26. Tang, Liu, Shwu Jen Chang, Ching-Jung Chen, and Jen-Tsai Liu. "Non-invasive blood glucose monitoring technology: a review." *Sensors* 20, no. 23 (2020): 6925.
27. Valenzuela, Francisco, Armando García, Erica Ruiz, Mabel Vazquez, Joaquin Cortez, and Adolfo Espinoza. "An IoT-based glucose monitoring algorithm to prevent diabetes complications." *applied sciences* 10, no. 3 (2020): 921.
28. Villena Gonzales, Wilbert, Ahmed Toaha Mobashsher, and Amin Abbosh. "The progress of glucose monitoring—A review of invasive to minimally and non-invasive techniques, devices and sensors." *Sensors* 19, no. 4 (2019): 800.
29. Zhang, Gaobo, Zhen Mei, Yuan Zhang, Xuesheng Ma, Benny Lo, Dongyi Chen, and Yuanting Zhang. "A noninvasive blood glucose monitoring system

based on smartphone PPG signal processing and machine learning." *IEEE Transactions on Industrial Informatics* 16, no. 11 (2020): 7209-7218.