Ministry of Higher Education &Scientific Research

Al-Nahrain University

College of Engineering

Computer Engineering Department

**Design and implementation of blood glucose for diabetes management using microcontroller**

A Final Year Project Submitted to the Department of Computer Engineering in the College of Engineering of Nahrain University in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Engineering.

**By**

**Ali Abdul Hussein Khalil**

**Ramadan 1442**

**May 2021**

**SUPERVISOR CERTIFICATION**

I certify that this final year project was prepared under my supervision at the Department of Computer Engineering, Al-Nahrain University as a partial fulfillment of the requirements of Bachelor of Science in Computer Engineering.

Signature:

Name:

(Supervisor)

Date: / / 2021

Signature:

Name: Dr. Ayman J. Salman

(Head of Department)

Date: / / 2021

**CERTIFICATE**

We certify, as an examining committee, that we have read this project entitled "**PROJECT TITLE**" examined the student **STUDENT NAME** in its content and found it meets the standard of final year project for the degree of Bachelor of Science in **Computer Engineering**.

Signature:

Name:

(Chairman)

Date: / /2021

Signature: Signature:

Name: Name:

(Member) (Member)

Date: / /2021 Date: / /2021

Signature:

Name:

(Supervisor and member)

Date: / /2021

Approval of the College of Engineering/ Al-Nahrain University.

**ABSTRACT**

This research (Design and implementation of blood glucose for diabetes management using microcontroller) proceeds towards the conclusion that bridging the gap between medical measuring devices that used in homes for personal use and the Internet, Pour this search to using the information obtained by traditional measuring devices for various purposes, including the ease of collecting large numbers of data and analyzing them technically, which leads to ease of diagnosis and formulation of medical policies And therapeutic side by side other benefits such as sustainable monitoring of developments in the health or disease situation and overcoming sudden cases, as this research places its first priority on improving the health status of individuals and groups alike by raising the level of technical monitoring to reduce the risks arising from cases of biological disorder and thus increase the chances of surviving these concomitant disorders This research follows a path dedicated to designing a blood glucose meter as a model for a group of other devices of a similar and similar character, which do not differ much in principle and mechanism from the desired design, in addition to that this research continues to be in the ranks of the research that achieves Research precedence in the field of Internet of things for the medical side, but the research is unique With a detailed presentation methodology of the way to design components of measurement, transmission and cloud services in a simplified manner that adopts ease and reduces material costs.

At the end of this research, we ask God for empowerment and payment in providing the benefit of this project to patients with chronic diabetes, the original target of the plan of this research, hoping for a speedy recovery for them first and foremost, for He is over all things powerful.

**TABLE OF CONTENTS**

[**ABSTRACT** III](#_Toc77177272)

[**TABLE OF CONTENTS** IV](#_Toc77177273)

[**LIST OF Figures** V](#_Toc77177274)

[**LIST OF Tables** VI](#_Toc77177275)

[**CHAPTER 1** 1](#_Toc77177276)

[1.1. Introduction 1](#_Toc77177277)

[1.2. Aim of the project 1](#_Toc77177278)

[1.3. Blood Glucose Measurement 1](#_Toc77177279)

[1.4. The principle of Invasive Glucose Measurement 3](#_Toc77177280)

[ Colorimetric Method 3](#_Toc77177281)

[ Amperometric Method 4](#_Toc77177282)

[**CHAPTER 2** 5](#_Toc77177283)

[2.1. Problem Statement 5](#_Toc77177284)

[2.2. Solution Methodology: 6](#_Toc77177285)

[**CHAPTER 3** 8](#_Toc77177286)

[3.1. Experimental and Design Procedures for Invasive Glucomete 8](#_Toc77177287)

[3.2. Hardware Components of (IGMSI) 19](#_Toc77177288)

[3.3. The Glucose measurement & Implementation 19](#_Toc77177289)

[3.4. Result on LCD 20](#_Toc77177290)

[3.5. Result on Smartphone 20](#_Toc77177291)

[3.6. Connect Arduino Wi-Fi by Smart Phone 21](#_Toc77177292)

[3.7. Connect NodeMCU with The Router 23](#_Toc77177293)

[3.8. Result on ThingSpeak Cloud 23](#_Toc77177294)

[3.9. Results or measurements on Telegram 24](#_Toc77177295)

[3.10. Software Flowchart 30](#_Toc77177296)

[3.11. Code for Display the Measurements on Smartphone 32](#_Toc77177297)

[3.12. Code for measurement the Glucose and Display it on LCD Then Send it to Telegram and ThingSpeak Cloud 34](#_Toc77177298)

[**CHAPTER 4** 40](#_Toc77177299)

[4.1. Conclusion 40](#_Toc77177300)

[4.2. Suggestion for Future Work 41](#_Toc77177301)

[ Make the glucometer work with multiuser 41](#_Toc77177302)

[ Add new features such as electronic diagnostic 41](#_Toc77177303)

**LIST OF Figures**

[Figure ‎1:1 Blood Glucose Measurement Methods 2](#_Toc77118538)

[Figure ‎3:1 : Completing Vision of Internet Glucometers System. 9](#_Toc77118539)

[Figure ‎3:2 IGMSI Design 10](#_Toc77118540)

[Figure ‎3:5 Token blood Sample From Finger 14](#_Toc77118541)

[Equation 3‑1 glucose oxidation 15](#_Toc77118542)

[Table ‎3:1 LCD to Microcontroller connection 16](#_Toc77118543)

[Table ‎3:2 NodeMCU Pin Categories 17](#_Toc77118544)

[Figure ‎3:6 Helping Telegram commands 24](#_Toc77118545)

[Figure ‎3:7 Help Commands 25](#_Toc77118546)

[Figure ‎3:9 Set Username and the telegram give the token access HTTP API. 27](#_Toc77118547)

[Figure ‎3:10 Change The Bot Name 27](#_Toc77118548)

[Figure ‎3:11 Chat Between Doctor and Glucometer 29](#_Toc77118549)

**LIST OF Tables**

[Table ‎3:1 LCD to Microcontroller connection 16](#_Toc77162292)

[Table ‎3:2 NodeMCU Pin Categories 17](#_Toc77162293)

**Nomenclature**

**IGMSI**: Invasive Glucose Monitoring System over Internet.

**NodeMCU**: standalone programmable microprocessor.

**Biosensor**: blood glucose sensor

**Thingspead**: cloud

**Workstation**: reaction or monitoring interface. is client, maybe indecat’s for hospitals, doctors or any important person.

# 

**GENERAL VISON OF THE GLUCOSE MEASUREMENT**

### Introduction

In the past two decades, the field of medicine has undergone many changes due to advances in computer technology. With the introduction of new devices and patient care techniques have improved significantly and this is reflected in the growth of life life expectancy in developed. Wireless technology is one of the major contributors to increased efficiency and the reliability of the healthcare system. One example of this is the remote monitoring of blood sugar levels of patients with diabetes. Future patients can use their cell phones to receive notifications about measured blood sugar levels, food intake, physical activity levels and medications. Then the cell phone notifies the patients of their health and transmits this information to the doctor to review it . This not only makes it easy for doctors to access patient information, but also provides the patient with assurance that they are being taken care of.The “IGMSI” device is also used for continuous health monitoring and diagnosis.

An example of this is the Polysomnographic Diagnostic System which was successfully used to examine causes of sudden infant death syndrome (SIDS)[1]

Wireless technology also expands the network of information systems present in hospitals. Picture Archiving and Communication Systems (PACS) is an example

of such a system. With PACS, diagnostic images taken from MRI, CT Scan, PET/CT, Ultrasound or a number of other imaging modalities, can be uploaded to universal servers, from where it can be accessed from anyone in any hospital connected to the network[2]. This increases the ease and speed with which a diagnosis can be made, and also eliminates the problems associated with previous methods of storing images (hard copies which could be lost or misplaced).

Diabetes Mellitus (DM) is a chronic disease characterized by the body’s inefficiency in producing / metabolizing insulin[3]. It is widely recognized as one of the biggest threats to public health today, acting as a “silent killer”. It not only seriously affects one’s quality of life, but also plays a significant role in the onset and development of other life-threatening illnesses[4][5].

### Aim of the project

This project aims to provide a service to a different segment of people, specifically those who suffer from chronic diabetes, by designing a blood glucose meter connected with the Internet and thus building an integrated health system to help the target group in this research.

### Blood Glucose Measurement

Using to blood glucose measurement, there are two methods to measurement blood glucose saturation, the invasive and non-invasive methods. Figure ‎1:1 Blood Glucose Measurement Methods illustrates the overall concept to glucose measurement methods.



Figure ‎1:1 Blood Glucose Measurement Methods

* Invasive Method

The invasive method done by using the needle to prick the finger and get the blood out, and then placed the blood sample onto the test strip to measurmrnt the concentration of glucose by a glucose meter .

A glucose meter is a medical device used to determine the concentration of glucose in the solution. The glucose concentration is measured in units of milligram per decilitre (mg/dl) or millimole per litre (mmol/L), depending on the different regions[6].

The purpose of this device is to measure biological signal and transmitit via Internet to use it in various aplications.

* Non-Invasive Method

Non-invasive method is painless. There is no break in the skin is created and there is no contact with the mucosa, or skin break, or internal body cavity beyond a natural or artificial body orifice. In this type of method have a advantage of not entering or penetrating the body. Some of the non-invasive method techniques are UV spectroscopy method, optical spectroscopy method, glucose oxidation method etc. But such method causes harmful to radiation which affects human being. Figure 3 shows the techniques for blood glucose measurement which is non-invasive.

### The principle of Invasive Glucose Measurement

Most glucose meters are based on electrochemical technology. They use electrochemical test strips to perform the measurement. A small drop of the solution to be tested is placed on a disposable test strip that the glucose meter uses for the glucose measurement. The two most common methods used in electrochemical measurement of glucose are the Colorimetric method (Non Invasive Method) and the Amperometric (Invasive Method) method[6].

* C**olorimetric Method**

In this method, the typical sensors such as LEDs or photo sensors form the analog interface. These sensors are followed by a Transimpedance Amplifier (TIA) for the glucose concentration measurement in the solution. The Color Reflectance principle is used in this method to sense the color intensity in the reaction layer of the test strip by the photometry. The glucose meter generates a numerical value, that is a measurement of the glucose concentration present in the solution[6][7].

* Amperometric Method

In this method, the electrochemical test strip contains a capillary that is used to draw in the solution placed at one end of the test strip. The test strip also contains an enzyme electrode containing a reagent such as Glucose Oxidase. Glucose undergoes a chemical reaction in the presence of enzymes and electrons are produced during the chemical reaction[8]. These electrons (i.e., the charge passing through the electrode) are measured and this is proportional to the concentration of glucose in the solution. An ambient temperature measurement is also made in order to compensate for the effect of temperature on the rate of the reaction[6][7].

# 

**STATEMENT OF PROBLEM AND METHODOLOGY OF SOLUTION**

This chapter will outline the technical objective of the project and provide the theoretical and technical knowledge related to the development of the final design for the IGMSI device. Tools and programs essential in developing the final solution are also discussed to give the reader practical knowledge regarding the IGMSI design process (detailed in the following chapters).

### Problem Statement

For simplicity, the design objective for this search can be broken down into three categories:

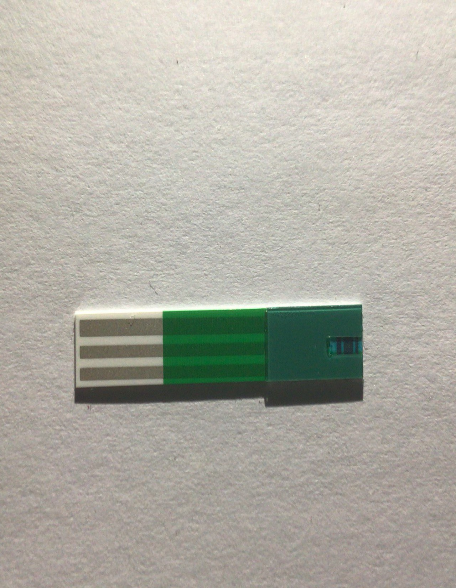
* handling the microcontroller with biosensor signal,
* display the measurement of glucose on LCD and smartphone without internet.
* connection the glucometer with the internet and sending the measurements to the Telegram application and ThingSpeak cloud.

### Solution Methodology:

For biosensor design,we used “oncall” test strip. To handling test strip with microcontroller need to use soket to insert the strips in it, the reaction current(working current ) is token by A0 (A/D:Anylog to Digital Converter) to filtering then convert to the voltage then sampling voltage and measure the glucose independent on the voltage amplitude. Figure 2.1 Shows how to test strip connect with NodeMCU microcontroller.

Figure ‎2:1 Glucose Recognition

Working Electrode



A/D

Microcontroller

D4

D0

D7

Reference Electrode

# 

IMPLEMINTATION CHAPTER

In this chapter, a detailed account of the design procedures and experiments to test for proper functionality are given. A logical order is followed in which procedures and experiments are introduced in generally the same order in which they were performed.

### Experimental and Design Procedures for Invasive Glucomete

The Biosensors are rapidly becoming part of our everyday life. As technology advances, we may soon be able to monitor many aspects of our health at home, in real-time, and without going to the hospital. Therefore, we see the increasing interest of researchers and technology companies to developing biometric devices and linking them with the Internet and building databases for them, that work for archiving or data analysis, as well as using artificial intelligence techniques for the purpose of analysis, diagnosis and future prediction cases[9], and this helps many people to live a safe life away from sudden health threats.

Currently, point-of-care applications constitute the main use of biosensors. In this area we see companies such as Luciole Medical[10] in Switzerland, which is developing a [minimally invasive probe](https://www.luciolemedical.com/technology) to measure blood oxygen levels in the brain for intensive care. This make us thinking to bulding glucometr device handle with internet. Figure ‎3:1 illustrates linked set of glucometer devices (A, B, C, D) with the internet and shared thire informations with workstalion interface.

B Glucose

Meter

A Glucose

Meter

D Glucose

Meter

C Glucose

Meter

Internet

Workstation

Workstation

Data Center

Figure ‎3:1 : Completing Vision of Internet Glucometers System.

In this sherch we take one glucometer device to implement this task, and the other devices (glucose meter or any biologic device working as same these procedures). This device can be called IGMSI. Figure ‎3:2 illustrate how IGMSI designing and connect with internet.

4- LCD 2\*16

1. Glucose Strip

2- NodeMCU Microcontroller

5- ThingSpeak Cloud

3- Smartphone to connect on LAN

7- Button

8- LED

6- Telegram

Figure ‎3:2 IGMSI Design

Now let us describe all components of this device

1. Glucose Strip(Biosensor):

The Glucose Strip forms the main biochemical sensor where the sample of solution is placed. The test strip has the following Three electrodes[7][6] as illustrate in Figure ‎3:3 :

Working electrode: Electrons are produced here during the chemical reaction. This electrode is connected to the current-to-voltage amplifier[6].

Reference electrode: Held at a constant voltage with respect to the working electrode to push the desired chemical reactions[6].

Counter electrode: Supplies current to the working electrode[6].

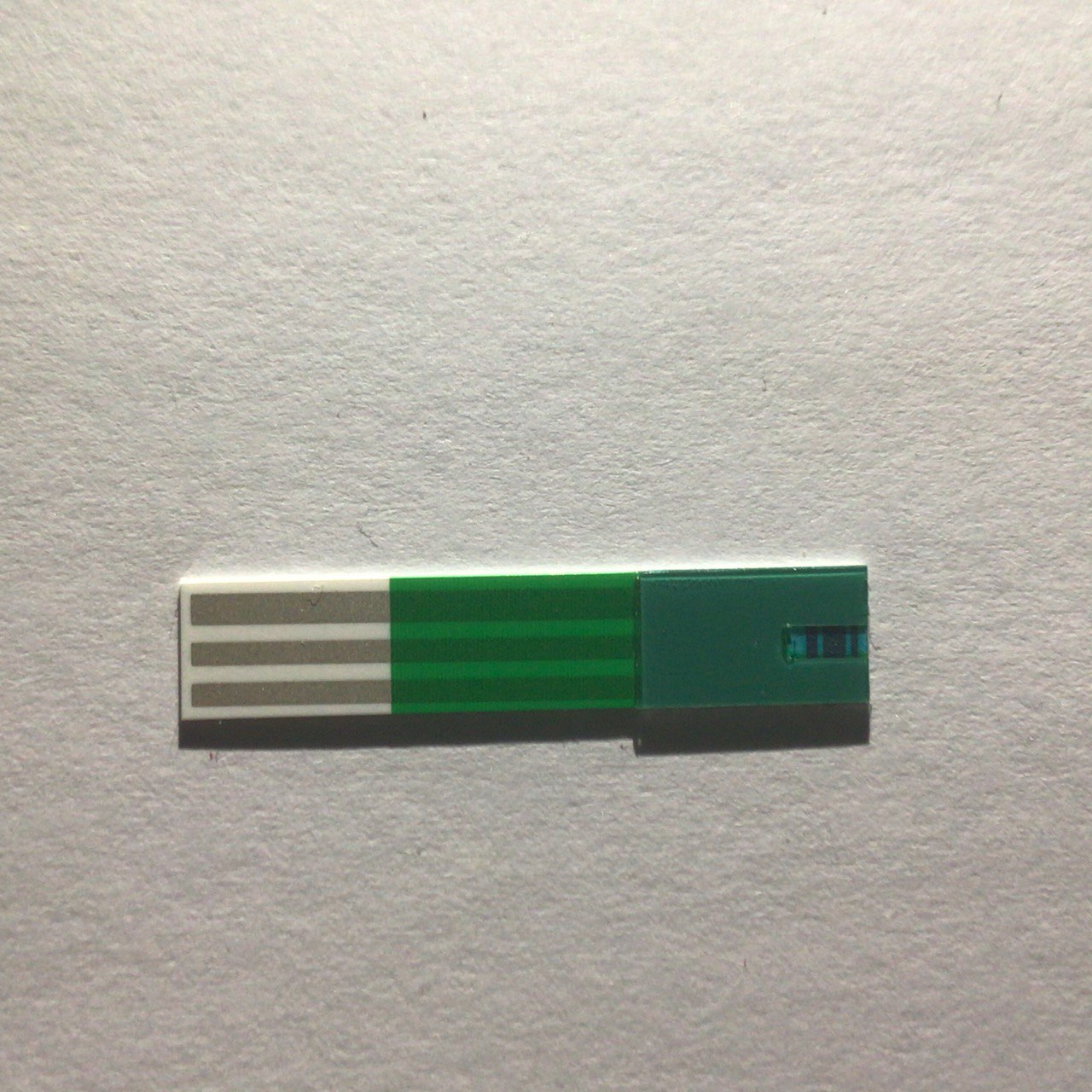


Figure ‎3:3 Three Electrode Strip Model

Most of the glucose meters design use only two electrodes, reference electrode and working electrode as illustrate in **Error! Reference source not found.**.



Figure ‎3:4 Schematic Diagram for Tow Electrodes Test Strip [3]

But the biosensor strips hold the same idea to measure the glucose in blood sample that token from figure[3] that ilustarate this Figure ‎3:5



Figure ‎3:5 Token blood Sample From Finger

A precise reference voltage (VREF) is applied to the reference electrode and a precise bias voltage (VBIAS) is applied to the op amp or in this search on Ina219 current sonser. This way the precise potential difference is maintained across the working electrode and the reference electrode. This voltage is the stimulus which drives the test strip’s output current[5]. The magnitude of the output current is then used to calculate the number of electrons produced[6][7].

“ A voltage is applied in the WE and RE electrodes with a range of -400 millivolts to 8 volts. This is used to define the voltage at which the sensor is able to perform at the maximum current. This electrical current is produced by the very selective oxidation of glucose in the blood sample, which is catalyzed by two reagents which are pre-coated inside the test strip: (1) an enzyme and (2) a mediator molecule” In addition to that, some of references referre to take between 2 and 6 seconds before taking measurement from strip to have peak voltage value and stable.The time token dependent in some times on the type and the producer of the glucose strip[3].

The solution sample is deposited on the test strip and the reaction of the glucose with the enzyme takes place. Electrons are generated during the chemical reaction and the sample of blood work as a jumper between the reference and working electrods. Flow of electrons will correspond to the flow of current through the working and the reference electrode. This current will change according to the glucose concentration[5]. The current is measured using a transimpedance amplifier (current-to-voltage converter) for the measurement with an Analog-toDigital Converter (ADC)[7][5]. The output of the transimpedance amplifier will be seen as a variation in the voltage with varying glucose concentrations in the solution[6].

The basic operation of glucose biosensor is based on the fact that the enzyme glucose oxidize(GOD) catalyses the oxidation of glucose to gluconic acid. The enzyme acts as a biorecognition element, which recognizes glucose molecules. These enzyme molecules are located on an electrode surface, which acts as a transducer. As soon as the enzyme recognizes the glucose molecules, it acts as a catalyst to produce gluconic acid[11][5] as illustrate in **Error! Reference source not found.** [5].

Equation 3‑1 glucose oxidation



Although there are many differences between the various commercially available test strips, they all rely on the fundamental mechanism of discussed above[12].

By using “oncall” test strip to measurment the current ampletude by ADC(Anaylog Digital Converter) is dicresed over the time, this illustrates that the concentration of blood glucose decreases with time due to the oxidation reaction[3].

1. LCD 2\*16

16×2 LCD is a 32 digits display screen .In Liquid crystal display 16×2, there are 2 rows and 16 columns. Any digit from ASCII code is viewable on the module. It supports the custom signs and designs but those require some specific methods and have some limitations.

Table ‎3:1 LCD to Microcontroller connection

|  |  |
| --- | --- |
| LCD display | Geekcreit NodeMCU board |
| VCC | +5v |
| GND | Ground |
| SDA | Pin D1 |
| SCL | Pin D2 |

1. **NodeMCU Board Pinout Configuration:**



Figure 3.6 NodeMCU Pin.

Table ‎3:2 NodeMCU Pin Categories

|  |  |  |
| --- | --- | --- |
| Pin Category | Name | Description |
| Power | Micro-USB, 3.3V, GND, Vin | **3.3V:** Regulated 3.3V can be supplied to this pin to power the board  **GND:** Ground pins  **Vin:** External Power Supply |
| Control Pins | EN, RST | The pin and the button resets the microcontroller |
| Analog Pin |  | Used to measure analog voltage in the range of 0-3.3V |
| GPIO Pins | GPIO1 to GPIO16 | NodeMCU has 16 general purpose input-output pins on its board |
| SPI Pins | SD1, CMD, SD0, CLK | NodeMCU has four pins available for SPI communication. |
| UART Pins | TXD0, RXD0, TXD2, RXD2 | NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins | CLk, SAD | NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C. |

**4. Internet**

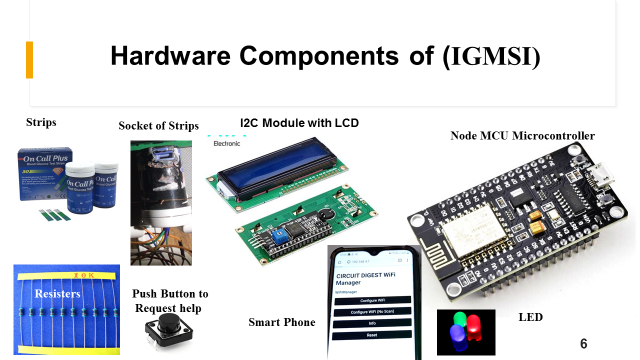
To connect hardware device withe internet there are building wifi with Esp8266 chip in Nodemcu controller, can use #include <ESP8266WiFi.h> library to connect wifi with LAN network.

**5. ThingSpeak Cloud**

There are many of IOT cloud to handle with nodemcu microcontroller, we will use “ThingSpeak” cloud here.

The “*ThingSpeak*” is where we are going to store the data collected by our glucometer and where we can see the data that we collected. Visit ThingSpeak.com and Sign Up for an account. Once you have a user account, you need to create a channel. ThingSpeak channels are where data gets stored. Create a new channel by selecting Channels, My Channels, and then New Channel. Name the channel, “ali\_99” for example, and name Field 1, “Glucose Reading CurveI”. Click “Save Channel” at the bottom to finish the process.

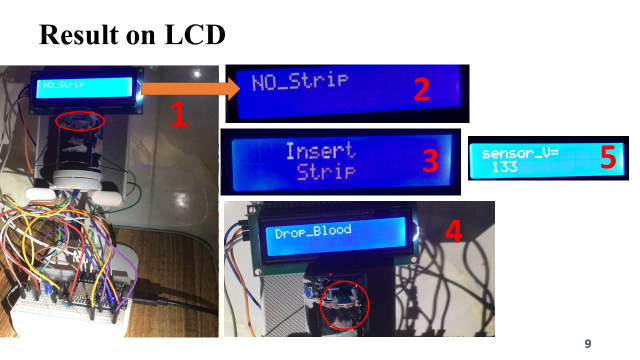
### Hardware Components of (IGMSI)



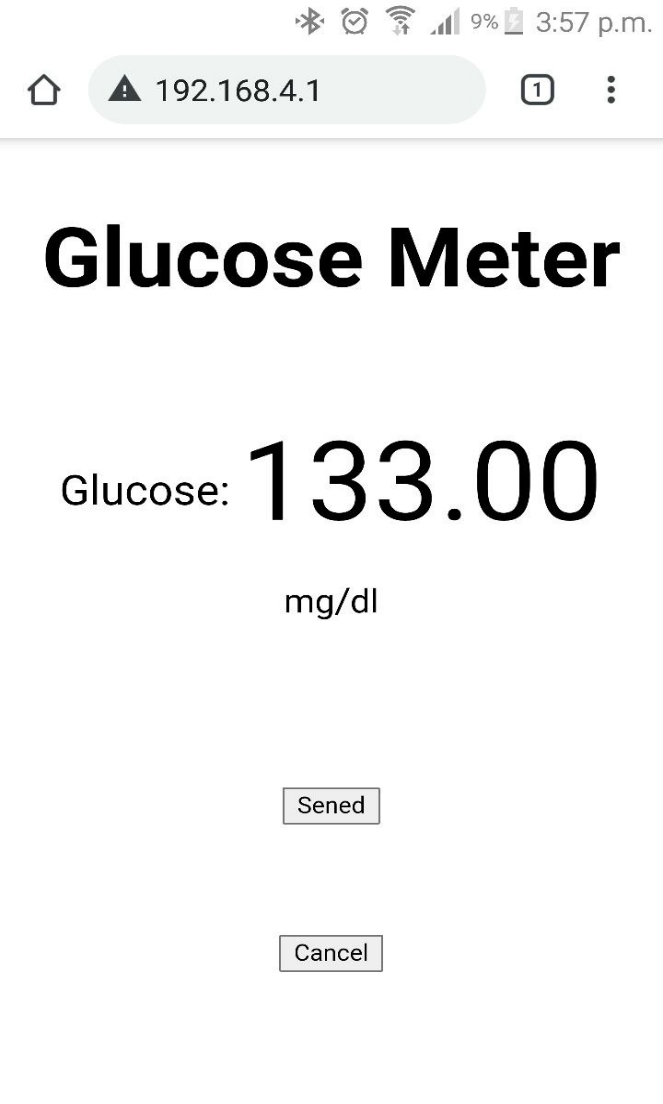
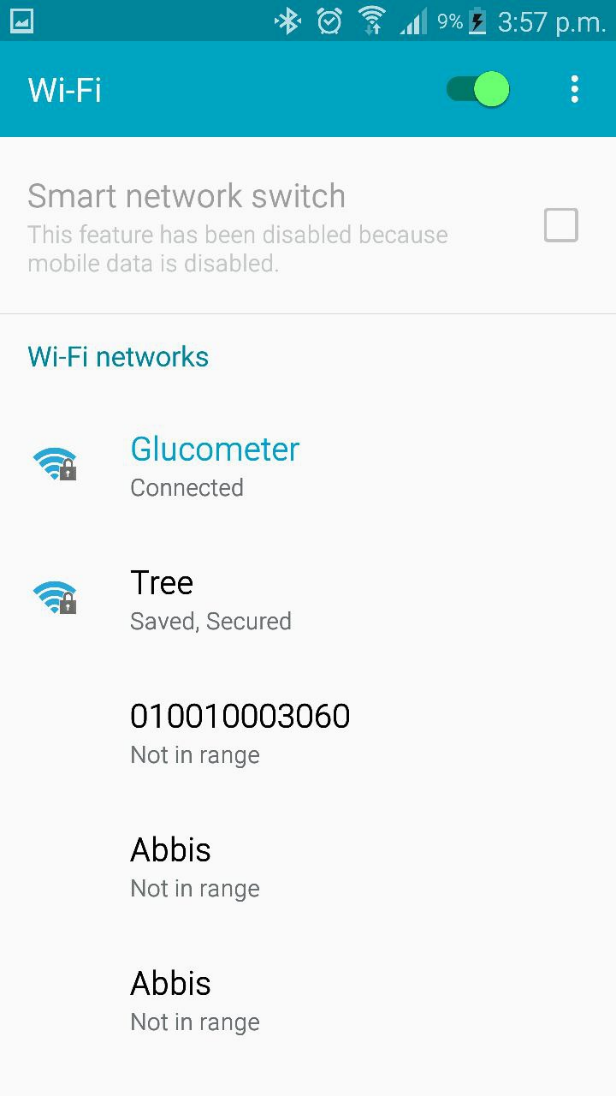
### The Glucose measurement & Implementation

|  |  |  |
| --- | --- | --- |
| Pin | Strip | Controllor |
| 1 | Count electrode | D0&D1 |
| 2 | Reference electrode | D5 |
| 3 | Working electrode | A0 |

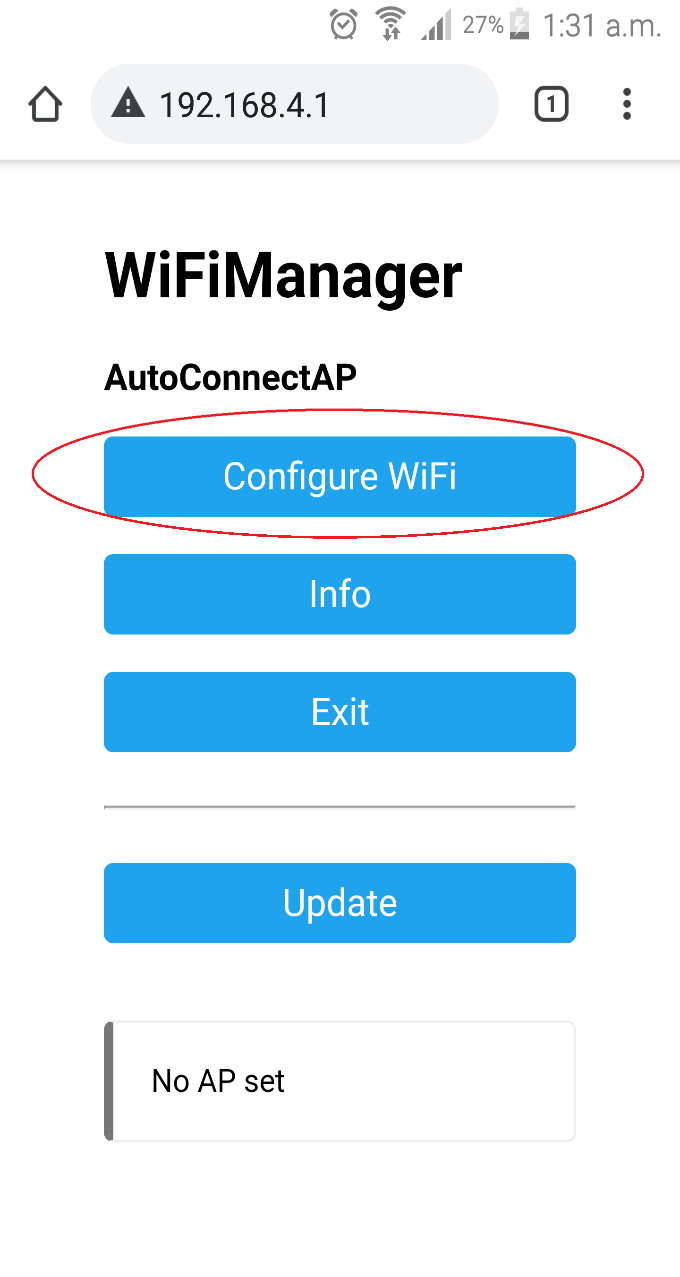
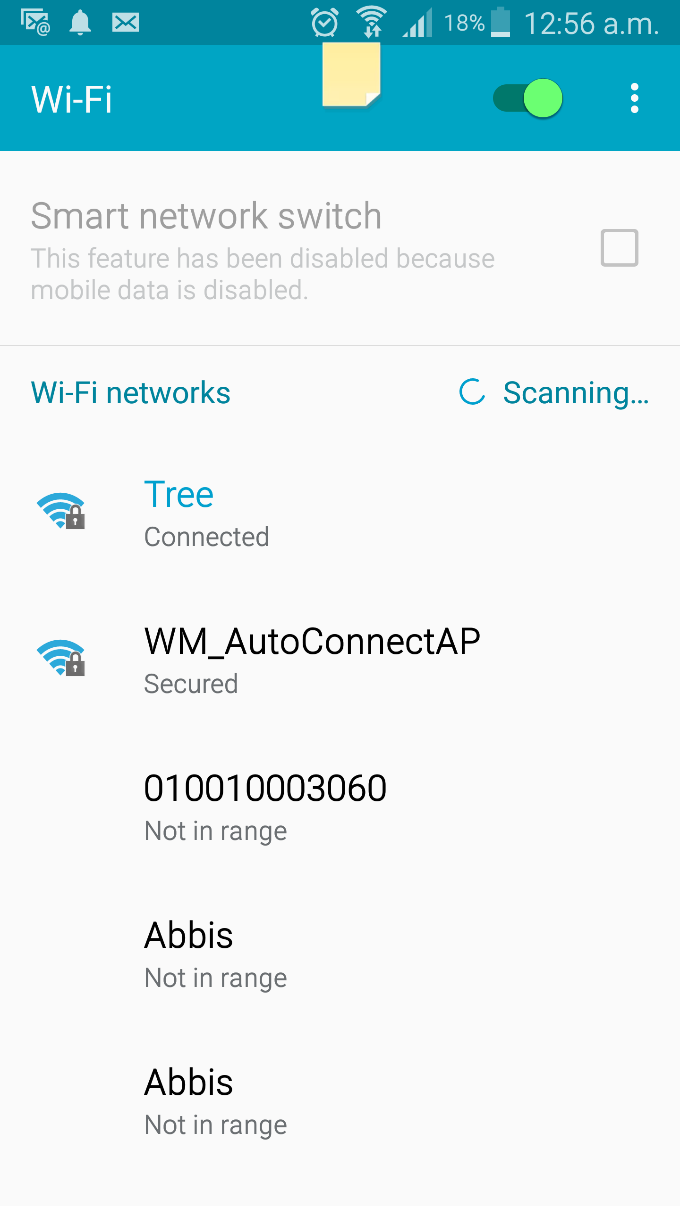
### Result on LCD

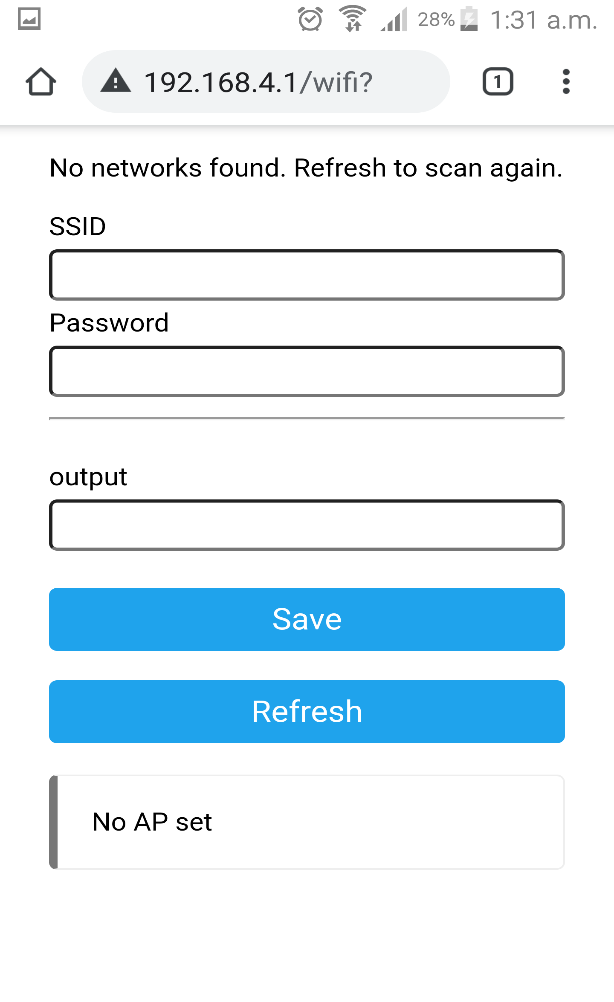


### Result on Smartphone



### Connect Arduino Wi-Fi by Smart Phone



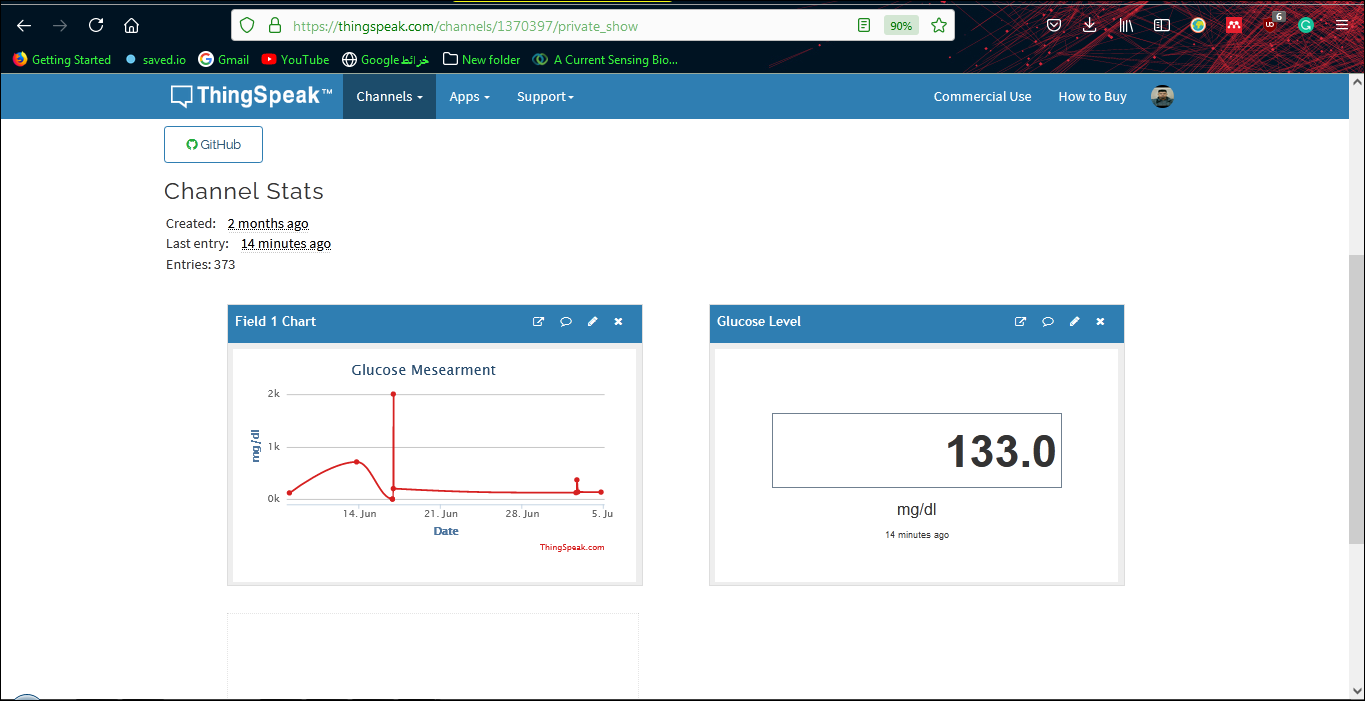


### Connect NodeMCU with The Router

Graphical user interface

Description automatically generated

### Result on ThingSpeak Cloud



### Results or measurements on Telegram

* How do I create a bot[13]?

From Botfather page can find this helping bot list by enter “/help” command, as illustrate in Figure ‎3:6 A and B



Figure ‎3:6 Helping Telegram commands

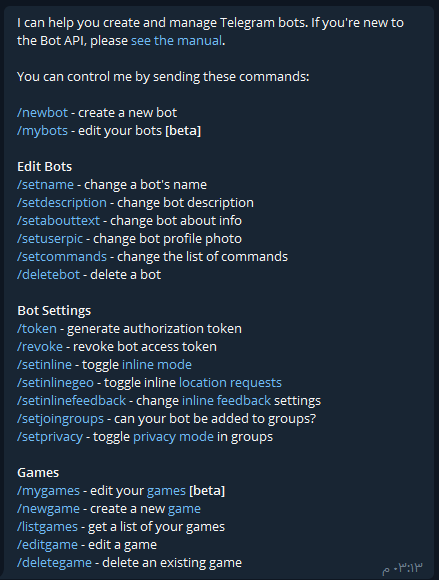
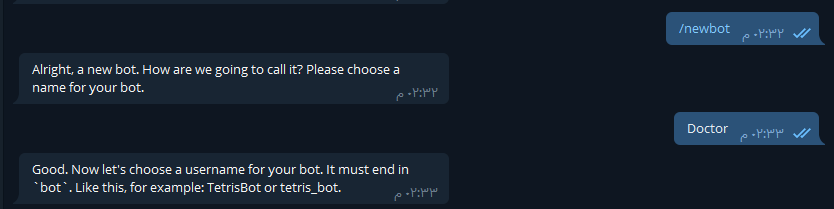


Figure ‎3:7 Help Commands

Use the **/newbot** command to create a new bot. The BotFather will ask you for a bot name. we will give name “Doctor” as illustrate in Figure ‎3:8.

The **name** of our bot is displayed in contact details and elsewhere.

Figure ‎3:8 Create New Telegram chat Bot

After that Botfather will ask to choose username for this bot. This bot must end in “bot”. will give usename “PatientDoctor\_bot”. as illustrste in Figure ‎3:9

The **Username** is a short name, to be used in mentions and t.me links. Usernames are 5-32 characters long and are case insensitive, but may only include Latin characters, numbers, and underscores. Your bot's username **must** end in 'bot', e.g. 'tetris\_bot' or 'TetrisBot'.

Then generate an authorization token for your new bot. The token is a string along the lines of “*1775492194:AAEtCKNab3vpbzrLpL9\_oEr22V1U-gzfRUg* “ that is required to authorize the bot and send requests to the [Bot API](https://core.telegram.org/bots/api). Keep your token secure and store it safely, it can be used by anyone to control your bot.



Figure ‎3:9 Set Username and the telegram give the token access HTTP API.

To change bot name show Figure ‎3:10



Figure ‎3:10 Change The Bot Name

Use the **/newbot** command to create a new bot. The BotFather will ask you for a name and username, then generate an authorization token for your new bot.

The **name** of your bot is displayed in contact details and elsewhere.

* Doctor Telegram

A screenshot of a computer

Description automatically generated

Figure ‎3:11 Chat Between Doctor and Glucometer

### Software Flowchart

Configure the Wi-Fi connection

Network System

Configure the APIs connection

Data Transmitting

Workstation 1 (Telegram)

Waiting to insert test strip

Check test strip valid

NO

Glucose Measurement System

Start

Drop blood

Measurement

Buffer storage

Workstation 2 (Thingspeak)

Ending Connection

EEPROM

Access Point

Yes

LCD

Used

Push Button to Request help

Waiting 0.7 min

### Code for Display the Measurements on Smartphone

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <Hash.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

const char\* ssid     = "Glucometer";

const char\* password = "123456789";

float t = 0.0;

AsyncWebServer server(80);

unsigned long previousMillis = 0;    // will store last time DHT was updated

// Updates DHT readings every 10 seconds

const long interval = 10000;

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML><html>

<head>

 <meta name="viewport" content="width=device-width, initial-scale=1">

 <style>

   html {

    font-family: Arial;

    display: inline-block;

    margin: 2px auto;

    text-align: center;

   }

   h2 { font-size: 3.0rem; }

   p { font-size: 4.0rem; }

   .units { font-size: 1.2rem; }

   .dht-labels{

     font-size: 1.5rem;

     vertical-align:middle;

     padding-bottom: 15px;

   }

 </style>

</head>

<body>

 <h2>Glucose Meter</h2>

 <p>

   <span class="dht-labels">Glucose: </span>

   <span id="G">%G%</span>

   <sup class="units">mg/dl</sup>

 </p>

 </p><button>Sened</button></p>

 </p><button>Cancel</button></p>

</body>

<script>

setInterval(function ( ) {

 var xhttp = new XMLHttpRequest();

 xhttp.onreadystatechange = function() {

   if (this.readyState == 4 && this.status == 200) {

     document.getElementById("G").innerHTML = this.responseText;

   }

 };

 xhttp.open("GET", "/G", true);

 xhttp.send();

}, 10000 ) ;

setInterval(function ( ) {

 var xhttp = new XMLHttpRequest();

 xhttp.onreadystatechange = function() {

   if (this.readyState == 4 && this.status == 200) {

     document.getElementById("humidity").innerHTML = this.responseText;

   }

 };

 xhttp.open("GET", "/humidity", true);

 xhttp.send();

}, 10000 ) ;

</script>

</html>)rawliteral";

String processor(const String& var){

 //Serial.println(var);

 if(var == "G"){

   return String(t);

 }

 return String();

}

void setup(){

**Serial**.begin(115200);

**Serial**.print("Setting AP (Access Point)…");

**WiFi**.softAP(ssid, password);

**IPAddress** IP = **WiFi**.softAPIP();

**Serial**.print("AP IP address: ");

**Serial**.println(IP);

 // Print ESP8266 Local IP Address

**Serial**.println(**WiFi**.localIP());

 // Route for root / web page

 server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

   request->send\_P(200, "text/html", index\_html, processor);

 });

 server.on("/Glucose: ", HTTP\_GET, [](AsyncWebServerRequest \*request){

   request->send\_P(200, "text/plain", String(t).c\_str());

 });

 // Start server

 server.begin();

}

void loop(){

 unsigned long currentMillis = millis();

 if (currentMillis - previousMillis >= interval) {

   // save the last time you updated the DHT values

   previousMillis = currentMillis;

   float newT = 133;

   if (isnan(newT)) {

**Serial**.println("Failed to read from DHT sensor!");

   }

   else {

     t = newT;

**Serial**.println(t);

   }

 }

}

### Code for measurement the Glucose and Display it on LCD Then Send it to Telegram and ThingSpeak Cloud

/\*

 -------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Glucose Measurement System \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 Editing by Ali Abdul Hussein Khalil

 2020/2021

 Al-Nahrain University

 -------------------------------------------------------------------------

\*/

#include <FS.h>

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <**ThingSpeak**.h>

#include <**DNSServer**.h>

#include <**ESP8266WebServer**.h>

#include <**WiFiManager**.h>

#include <ArduinoJson.h>

#include "CTBot.h"

#include <**LCD\_I2C**.h>

**LCD\_I2C** lcd(0x27);

#define test\_strip\_d4 D4

#define insert\_strip\_d0 D0

#define write\_d7 D7

#define led\_d8 D8

#define help\_d3 D3

**CTBot** myBot;

const char \* myWriteAPIKey = "V9JFKXZV1YIKAKH2"; //Your Write API Key

long ID = 818222669; //Your Channel Number (Without Brackets)

unsigned long myChannelNumber = 1370397;

const char\* server = "api.thingspeak.com";

String token = "1828120441:AAF5PsHpWyTILga5JAq7j\_Hgazrlm4YiIY8";

long analogInPin\_A0 = A0;

WiFiClient client;

int buttonState = 0;

// Set web server port number to 80

//WiFiServer server(80);

**WiFiManager** wifiManager;

// Auxiliar variables to store the current output state

String outputState = "off";

// Assign output variables to GPIO pins

char output[2] = "5";

//flag for saving data

bool shouldSaveConfig = false;

//callback notifying us of the need to save config

void saveConfigCallback () {

**Serial**.println("Should save config");

 shouldSaveConfig = true;

}

void setup() {

**Serial**.begin(115200);

 pinMode(analogInPin\_A0, INPUT);

 pinMode(insert\_strip\_d0, INPUT);

 pinMode(test\_strip\_d4, OUTPUT);

 pinMode(help\_d3, INPUT);

 pinMode(led\_d8, OUTPUT);

 digitalWrite(led\_d8, LOW);

 //read configuration from FS json

**Serial**.println("mounting FS...");

 if (**SPIFFS**.begin()) {

**Serial**.println("mounted file system");

   if (**SPIFFS**.exists("/config.json")) {

     //file exists, reading and loading

**Serial**.println("reading config file");

     File configFile = **SPIFFS**.open("/config.json", "r");

     if (configFile) {

**Serial**.println("opened config file");

**size\_t** size = configFile.size();

       // Allocate a buffer to store contents of the file.

       std::unique\_ptr<char[]> buf(new char[size]);

       configFile.readBytes(buf.get(), size);

**DynamicJsonBuffer** jsonBuffer;

**JsonObject**& json = jsonBuffer.parseObject(buf.get());

       json.printTo(**Serial**);

       if (json.success()) {

**Serial**.println("\nparsed json");

         strcpy(output, json["output"]);

**Serial**.println(output);

         delay(7000);

       } else {

**Serial**.println("failed to load json config");

       }

     }

   }

 } else {

**Serial**.println("failed to mount FS");

 }

 //end read

 WiFiManagerParameter custom\_output("output", "output", output, 2);

 //set config save notify callback

 wifiManager.setSaveConfigCallback(saveConfigCallback);

 // set custom ip for portal

 //wifiManager.setAPConfig(IPAddress(10,0,1,1), IPAddress(10,0,1,1), IPAddress(255,255,255,0));

 wifiManager.addParameter(&custom\_output);

 // fetches ssid and pass from eeprom and tries to connect

 // if it does not connect it starts an access point with the specified name

 // here  "AutoConnectAP"

 // and goes into a blocking loop awaiting configuration

 wifiManager.autoConnect("AutoConnectAP");

 // or use this for auto generated name ESP + ChipID

 //wifiManager.autoConnect();

**Serial**.println("Connected.");

 strcpy(output, custom\_output.getValue());

 //save the custom parameters to FS

 if (shouldSaveConfig) {

**Serial**.println("saving config");

**DynamicJsonBuffer** jsonBuffer;

**JsonObject**& json = jsonBuffer.createObject();

   json["output"] = output;

   File configFile = **SPIFFS**.open("/config.json", "w");

   if (!configFile) {

**Serial**.println("failed to open config file for writing");

   }

   json.printTo(**Serial**);

   json.printTo(configFile);

   configFile.close();

   //end save

 }

 String ssid = (String)wifiManager.getWiFiSSID();

 String pass =  (String)wifiManager.getWiFiPass();

 delay(10);

**WiFi**.begin(ssid, pass);

 while (**WiFi**.status() != WL\_CONNECTED)

 {

   delay(500);

**Serial**.print(".");

 }

**ThingSpeak**.begin(client);

**Serial**.println("");

**Serial**.println("WiFi connected");

 myBot.wifiConnect(ssid, pass);

 myBot.setTelegramToken(token);

 if (myBot.testConnection())

**Serial**.println("\ntestConnection OK");

 else

**Serial**.println("\ntestConnection NOK");

}

void loop() {

 lcd.begin();

 lcd.backlight();

 lcd.print("   Insert");

 lcd.setCursor(4, 1);

 lcd.print("Strip");

 delay(700);

 for (int i = 0; i < 6; ++i)

 {

   lcd.backlight();

   delay(60);

   lcd.noBacklight();

   delay(60);

 }

 delay(1000);

 lcd.backlight();

 lcd.clear();

 delay(500);

 digitalWrite(test\_strip\_d4, HIGH);

 if ( digitalRead(insert\_strip\_d0) == LOW) {

   lcd.clear();

   lcd.print("NO\_Strip");

   delay(1000);

   return;

 }

 else if ( digitalRead(insert\_strip\_d0) == HIGH) {

   // digitalWrite(test\_strip\_d4,LOW);

   digitalWrite(write\_d7, HIGH);

   // lcd.clear();

   //lcd.print("Drop\_Blood");

   long average = 3;

   average = analogRead(analogInPin\_A0);

**Serial**.println(average);

   if (average > 50 && average < 200) {

     lcd.clear();

     lcd.setCursor(1, 1);

     lcd.print("used strip");

**Serial**.println(">>>>>>");

     delay(2200);

   }

   else if (average >= 0 && average <= 50) {

     lcd.clear();

     lcd.print("Drop\_Blood");

     long average = 0;

     delay(10600);

     average = analogRead(analogInPin\_A0);

     if (average >= 200 && average < 1024) {

       average = average \* 0.16438;

       lcd.clear();

       lcd.print("sensor\_V= ");

       lcd.setCursor(1, 1);

       lcd.print(average);

       TBMessage msg;

       myBot.sendMessage(ID, "Glocuse Mesuerment...");

       myBot.sendMessage(ID, "PN:Zaid Ali");

       String lo = String(average);

       myBot.sendMessage(ID, lo);

**ThingSpeak**.writeField(myChannelNumber, 1, average, myWriteAPIKey); //Update in ThingSpeak

**Serial**.println("telebot");

       long x = 1;

       for (x = 1; x < 100000000; x++) {

**Serial**.println("wait");

         TBMessage msg;

         digitalWrite(help\_d3, LOW);

         // if there is an incoming message...

         if (myBot.getNewMessage(msg)) {

           // ...forward it to the sender

           myBot.sendMessage(msg.sender.id, msg.text);

**Serial**.println(msg.text);

           String mm = String(msg.text);

           lcd.clear();

           lcd.print(mm);

           if (msg.text.equalsIgnoreCase("LT ON")) {

      // if the received message is "LIGHT ON"...

             digitalWrite(led\_d8, HIGH);

         // turn on the LED (inverted logic!)

             myBot.sendMessage(ID, "Light is now ON");

             lcd.backlight();

             lcd.print("   HIGH");

             lcd.setCursor(4, 1);

             lcd.print("GLUCOSE");

             delay(700);

             for (int i = 0; i < 6; ++i)

             {

               lcd.backlight();

               delay(60);

               lcd.noBacklight();

               delay(60);

             }

           }

         }

         else if (digitalRead(help\_d3) == LOW) {

           lcd.clear();

           myBot.sendMessage(ID, "help");

           lcd.print("help");

         }

         delay(3000);

       }

     }

   }

 }

}

# 

**CONCLUSION AND SUGGESTION FOR FUTURE WORK**

### Conclusion

* This Project presented a real-time remote IoT-based continuous glucose monitoring system. The implemented IoT-based architecture is complete system starting from sensor to a back-end server.
* The main goal is to design and implement a very low cost with The hardware and software features of Arduino NodeMCU based system used to determine the approximate concentration of glucose in the blood is described.
* The“on call” Bio sensor (Test Strips) are used for monitoring blood glucose levels.
* when a drop of blood is placed on the test strip it interacts with the elements on the strip that causes a reaction and an electric is generated.
* The wifi is smart to connect with LAN network, used EEPROM to store wifi configuration or Access point mode to connect by smart phone
* The glucometer device use the telegram bot to transfer the messages between it and telegram also too send the glucose measurement to thingspeak cloud.
* The necessary software is developed in C programming language,
* Through the system, doctors and caregivers can easily monitor their patient anytime, anywhere via a browser or a smart-phone.

### Suggestion for Future Work

#### Make the glucometer work with multiuser

#### Add new features such as electronic diagnostic

**REFERENCES**

[1] D. Barschdorff, “Information processing of biomedical sensor signatures for ensuring the quality of life,” *18th IMEKO World Congr. 2006 Metrol. a Sustain. Dev.*, vol. 2, pp. 951–962, 2006.

[2] “Diagnostic Imaging Services - St. Joseph Hospital.” https://www.stjosephhospital.com/services/diagnostic-imaging-services/diagnostic-imaging-services (accessed May 20, 2021).

[3] S. Saha, N. Sarker, and A. Hira, “Design & implementation of a low cost blood glucose meter with high accuracy,” *1st Int. Conf. Electr. Eng. Inf. Commun. Technol. ICEEICT 2014*, no. January 2019, 2014, doi: 10.1109/ICEEICT.2014.6919050.

[4] W. Zhang *et al.*, “Self-Powered Implantable Skin-Like Glucometer for Real-Time Detection of Blood Glucose Level In Vivo,” *Nano-Micro Lett.*, vol. 10, no. 2, pp. 1–11, 2018, doi: 10.1007/s40820-017-0185-x.

[5] M. G. Yanez, “Glucose Meter Fundamentals and Design,” *Glucose M. Fundam. Des.*, pp. 1–34, 2013.

[6] N. Dalvi, “Glucose Meter Reference Design,” p. 20, 2013, [Online]. Available: http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en566535.

[7] “Microchip Technology - Designing a glucose meter using an 8bit microcontroller.” https://www.epdtonthenet.net/article/85026/Designing-a-glucose-meter-using-an-8bit-microcontroller.aspx (accessed May 18, 2021).

[8] G. Hernandez-Vargas, J. E. Sosa-Hernández, S. Saldarriaga-Hernandez, A. M. Villalba-Rodríguez, R. Parra-Saldivar, and H. M. N. Iqbal, “Electrochemical biosensors: A solution to pollution detection with reference to environmental contaminants,” *Biosensors*, vol. 8, no. 2, pp. 1–21, 2018, doi: 10.3390/bios8020029.

[9] B. Balkau *et al.*, “Predicting diabetes: Clinical, biological, and genetic approaches,” *Diabetes Care*, vol. 31, no. 10, pp. 2056–2061, 2008, doi: 10.2337/dc08-0368.

[10] “Technology / Luciole Medical AG.” https://www.luciolemedical.com/technology (accessed May 21, 2021).

[11] A. S. A. El-hamid, A. E. Fetohi, R. S. Amin, and R. M. A. Hameed, “Design of Digital Blood Glucose Meter Based on Arduino UNO,” *IJournals All Rights Reserv.*, vol. 3, no. 8, p. 8, 2015, [Online]. Available: www.ijournals.in.

[12] K. K, “Application Note : カイネティクス特性解析 Application Note : ■ カイネティクス スクリーニングおよび選択,” *Reproduction*, vol. 06, no. 805, pp. 2–3, 2008.

[13] “Bots: An introduction for developers.” https://core.telegram.org/bots (accessed May 22, 2021).

**الخلاصة**

يسري هذا البحث (

Design and implementation of blood glucose for diabetes management using microcontroller)

تجاه خلاصة مفادها تجسير الفجوة الحاصلة بين اجهزة القياس الطبي المستخدمة في المنازل ذات الاستخدام الشخصي والانترنيت مما يصب في إطار استخدام المعلومات المستحصلة بواسطة أجهزة القياس التقليدية لأغراض متعددة منها سهولة جمع اعداد ضخمة من البيانات وتحلياها تقنيا مما يفضي الى سهولة التشخيص ورسم السياسات الطبية والعلاجية جنبا الى جنب فوائد اُخر أمثال المراقبة المستدامة للتطورات الحالة الصحية او المرضية وتجاوز الحالات المفاجئة. حيث يضع هذا البحث أولى أولوياته تحسين الحالة الصحية للأفراد والجماعات على حد سواء برفع مستوى المراقبة التقنية لتقليص المخاطر الناجمة من حالات الاضطراب البايلوجي وبالتالي زيادة فرص النجاة من هذه الاضطرابات الملازمة والعرضية الطارئة، حيث يسلك هذا البحث مسارا مخصصا لتصميم جهاز قياس نسبة السكر في الدم أنموذجا عن مجموعة من الأجهزة الاخرى ذات الطابع المماثل والمشابه والتي لا تختلف كثيرا من حيث المبدأ والالية عن التصميم المنشود، علاوة على ذألك يدأب هذا البحث ليكون في مصاف الابحاث التي تحقق الاسبقية البحثية في مجال انترنيت الأشياء لدى الجانب الطبي بيد ان البحث يتفرد بمنهجية عرض تفصيلية لطريقة تصميم مكونات القياس والارسال والخدمات السحابية بطريقة مبسطة تعتمد السهولة وخفض التكاليف المادية.

وفي نهاية هذا البحث نسأل من الله التمكين والسداد في تقديم مفاد هذا المشروع لمرضى داء السكر المزمن، الجهة المستهدفة بالأصالة من خطة هذا البحث راجين الله الشفاء العاجل لهم أولا وأخيرا فهو على كل شيء قدير.