Bachelor Thesis
Electrical Engineering
Thesis no
03 04 2021

Measurement of Core Body Temperature with a Powered Toothbrush

Vamsi Reddy Konduru
Satya Sai Naga Jaya Koushik Pilla

Dept. of Mathematics & Natural Sciences
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Department of Mathematics and Natural Sciences at Blekinge Institute of Technology in partially fulfilment of the Bachelor of Science in Electrical Engineering. The thesis is equivalent to 10 weeks of full-time studies.

Contact Information:
Author(s):
Satya Sai Naga Jaya Koushik Pilla
E-mail: acroyy@student.bth.se
Vamsi Reddy Konduru
E-mail: vako20@student.bth.se

University advisor:
Prof. Wlodek J Kulesza
Dept. of Mathematics and Natural Sciences

Dept. Electrical Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se/didd
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

We need to employ modern-day technologies in our daily lives to make our life easier. In this thesis, we analyze the user's oral hygiene habits and core body temperature to help people to track the state of their health. We designed a system in which the data are directly extracted from the mouth, then stored and presented in the application. By means of the system, the people can keep track of their core temperature without any extra function apart from their daily chores. The system can also help to keep track of the health of people with symptoms of dementia and Alzheimer.

We used an LM35 temperature sensor to measure temperature. The sensor is mounted on a powered toothbrush head. When the user starts to brush, the sensor records the temperature data for seven times in two minutes of an average brush time and sends it to the mobile application using WiFi. We used a NodeMCU board, ThingSpeak server and MIT inventor app to transfer the data and design the mobile application.

Keywords: Buzzer, Core temperature, NodeMCU, LM35 temperature sensor, Powered toothbrush.

Acknowledgments

We want to express our gratitude towards Prof. Wlodek J Kulesza, Irina Gertsovich and Johan Flyborg for their kind co-operation, supervision, and support in completing the project; they are at every walk of this project with full support.

Sincerely,

Konduru Vamsi Reddy
Satya Sai Naga Jaya Koushik Pilla

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In our day to day lives, the core body temperature is of paramount concern for health monitoring. Measurement of core body temperature helps to detect diseases and metabolic changes [4]. One area that has long been of interest is the ability to predict female ovulation and thus find the fertile window to facilitate the possibility of pregnancy. The fertility apps available today require to separate temperature measurement, the value of which is then manually entered into the app. This procedure involves several risks for data gaps and incorrect registrations. Diseases such as hypothyroidism and inflammatory conditions can be controlled and diagnosed using regular core body temperature measurements. In future, we could use it for health communication for elderly living in a home environment without home care by professionalises [5]. We can make use of the web application and IoT [6] to plot the graph of the obtained real-time temperature values from the WiFi [7] module and thereby study the core temperature dynamics [8] in the case taken using the obtained statistics.

The aim is to measure core body temperature with a temperature sensor integrated into the brush head of a powered toothbrush [9]. The data are transferred through the ThingSpeak to the app, and the graph are displayed in the app [10]. In order to get the core body temperature from the user, we place a temperature sensor on the toothbrush. Using the current IoT microcontroller, we collect the raw data from the user and store it in compatible electronic device and use it for further analysis. The data collected daily, the information can be stored efficiently on the ThingSpeak app server. The project benefits the user essentially, as instead of occupying a user's local hard disk space, the recorded information is made available for user's reach at the Internet using cloud technology / ThingSpeak app server. So, it enables the user to get his data on any electronics device he is wanted it to be. It is useful even for elderly, dementia suffering patients who cannot keep track of their core body temperature values, but it is necessary for them in order to diagnose their medical condition daily. This model can make it happen without much effort from the user.

This report consists of seven chapters, where Chapter 1 consists of the introduction, which is about the project problem and the motivation. Chapter 2 explains related work, which gives a brief explanation about the important terms and some of the methods proposed by other researchers. Chapter 3 explains the problem statement, objectives and main contributions of the project. Chapter 4 describes the modelling of the project, which consists of the project's user-driven data, which contains the data of all the tools, methods, apparatus, the particular constraints and web applications in a consolidated table, block diagram of the model and flow chart. Chapter 5 presents the implementation of the project and provides the information related to the electronic components used to build the system. The limitations and rated values of the devices are mentioned in this chapter. The software code that is required to execute the model as per the desired conditions too mentioned in this chapter. Chapter 6 consists of the testing and validation of the project results. Chapter 7 explains the conclusion and developments of the project according to the team.

Chapter 2

Related Works

Core body temperature [11] plays a vital role in tracking the menstrual cycles in women [12]. The core body temperature is the body's lowest resting temperature, which can be measured as soon as a person wakes up in the early morning. There is a correlation between the core body temperature and the fertile window of a female. Therefore by make use of the correlation between the core body temperature and ovulation, we can simulate the response required to make the correlation of core body temperature and ovulation with greater accuracy.

Core body temperature is bio mark of healthy ageing [13]. Scattered evidence indicates that a lower basal body temperature may be associated with prolonged health span, yet few studies have directly evaluated this relationship. They examined cross-sectional and longitudinal associations between early morning oral temperature (36.1°C to 37.2°C) and usual gait speed, endurance walk performance, fatigability, and grip strength in 762 non-frail men (52%) and women aged 65–89 years participating in the Baltimore Longitudinal Study of Ageing. Since excessive adiposity may alter temperature set points, associations were also examined within adiposity strata.

Core body temperature measurement using a multi-sensor armband is a study conducted exclusively on young Australian women as a part of observational study [14]. The upward shift in core body temperature, which occurs shortly after ovulation and continues until the next menses, is a potentially useful marker of ovulation, which has been exploited in clinical and research settings. The objective of this research is to investigate the utility of Body Media Sense Wear (BMSW) in monitoring ovulation in young women by analyzing the correlation and agreement of basal temperatures measured using BMSW and a digital oral thermometer.

Luteal phase length can be measured using the quantitative basal body temperature methods [15]. This research depicts the validation of the mid-cycle against LH peak will give the useful data for documenting hormonal characteristics. The primary purpose of that project is to evaluate qualitative basal body temperature methods for their accuracy and the reliability against mid cycle serum LH data.

Advances in information and communication technologies have led to the emergence of the IoT [16]. In the modern health care environment, the IoT technologies bring convenience to physicians and patients, since they are applied to various medical areas such as real-time monitoring, patient information management, and healthcare management. The body sensor network (BSN) technology is one of the core technologies of IoT developments in the healthcare system, where a patient can be monitored using a collection of tiny-powered and lightweight wireless sensor nodes. However, the development of this new technology in healthcare applications without considering security makes patient privacy vulnerable. In that paper, the authors highlight the critical security requirements in the BSN-based modern healthcare system. Subsequently, they propose a secure IoT-based healthcare system using BSN, called BSN-Care, which can efficiently accomplish those requirements.

IoT based device for fertility monitoring facilitates healthcare practitioners to be aware and associate with the patients proactively [17]. Fertility monitoring devices are non-invasive devices that assist in predicting the probability of conception. The conventional methods for fertility monitoring are frequently unspecific or cause discomfort as they mainly depend on body temperature alone. In this paper, a wearable device for fertility monitoring [4] that tracks ovulation by using abdominal thickness and basal body temperature is proposed.

Over the last few decades, life expectancy has increased significantly. However, older adults who live on their own often need assistance due to mobility difficulties, symptoms of dementia [18] or other health problems. In such cases, an autonomous supporting system may be helpful. This paper proposes the Internet of Things (IoT)-based information system for indoor and outdoor use [19], [20]. Since the conducted survey of related works indicated a lack of methodological approaches to the design process, therefore a Design Methodology (DM), which approaches the design target from the perspective of the stakeholders, contracting authorities and potential users are introduced. The implemented solution applies the three-axial accelerometer [21] and magnetometer, Pedestrian Dead Reckoning (PDR), thresholding and the decision trees algorithm.

The number of patients requiring medical assistance increases each day while the staff-patient ratio is not balanced, causing issues such as treatment delay and often leading to patient dissatisfaction. Besides that, healthcare devices are getting complex and challenging for them to be handled and interpreted personally by the patient [22]. Lack of staff and challenges in operating the medical devices affect patients in the hospital and home care patients that require full attention and constant monitoring. This urges for the development of new methods or technology. At present, Wireless Sensor Network (WSN) [23] is gaining interest as one of the major components in enabling the Internet of Things (IoT) since it offers low cost, low power monitoring besides reducing devices dependency on wires or cable [24].

Chapter 3

Problem Statement, Objectives and Main Contribution

As Survey of Related Works shows, the people that suffer memory loss or memory related health issues may forget to perform daily chores, for example brushing teeth. They could even have a problem to to track their health state by e.g. measure the core body temperature. The problem that also need to be solved in a simple way is to help a healthcare provider to remind a patients with memory impairment to perform their dally tasks if they forget them. The information about a basic health parameters such as core body temperature could be also useful for the care service. The problem is to acquire needed info and make access to it for the health provider.

The research question, that defines the objectives of our project is what are the functionalities and the structure a system which can support a health provider to help the people with memory issues to monitor their daily choirs. Meanwhile, the system should even provide a basic information about possible health issues of the patients without any additional feedback from them.

At this instant, we are interested in the generation of the core body temperature values seven times in the average brush time, using a temperature sensor placed at the rear end of the brush head, transmitting the obtained values with the use of inbuilt WiFi connection [25] and after that, depicting a graph of the temperature against the day and time of brushing. It also makes the user acknowledge the time instant of the brush using the different frequency tones. One thereby store these values and make a record of their teeth health.

By this project, we make the data digitally accessible to any number of health monitoring [26] centres at any time by using Cloud Computing Algorithms[27], as the system transmits every sample recorded by the sensor.

The main contribution of the thesis is the design, implementation and validation of an autonomous model for retrieving regular core body temperature data. We develop an app using the MIT app Inventor [28] in which we see the curves of core temperature and the time of data reading taken several times during the brush time.

We store the patient data in the cloud server, so that one access the patient records from anywhere in the world. With the Internet of Things applications, we access the core body temperature values at any desired electronic device available nearby. With the installation of the ThingSpeak or MIT application on that particular device, one get logged in and review their previously saved records in it. The device helps users with special conditions, having difficulties in memorizing things.

Chapter 4

Modelling

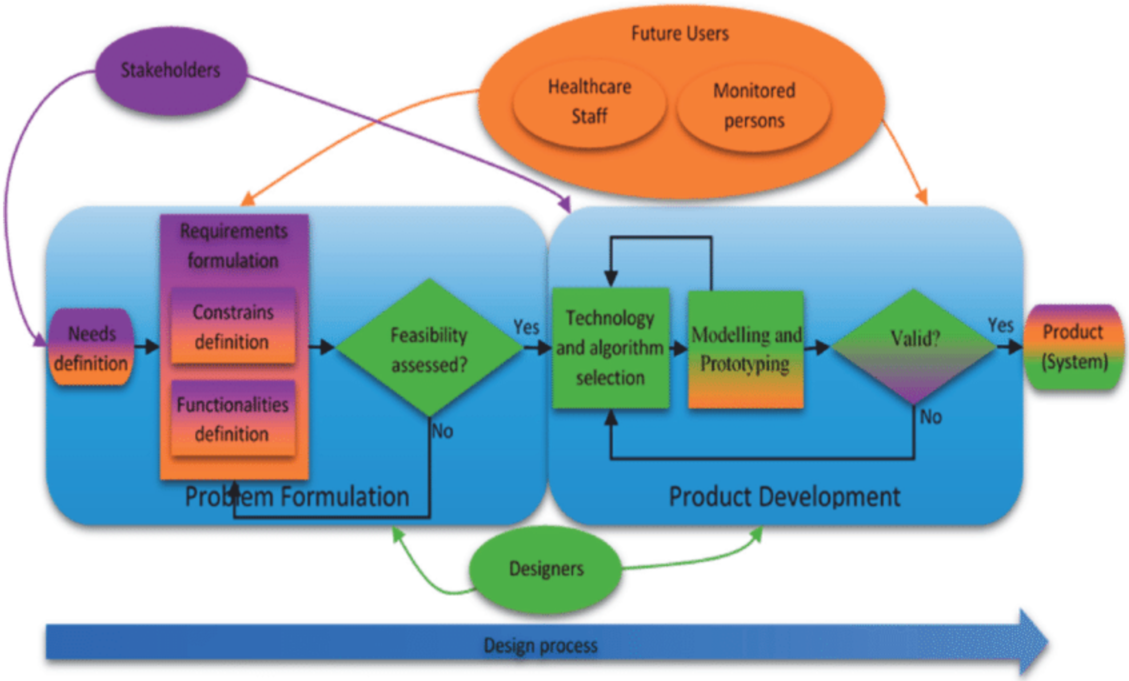


Figure 4.1: Flowchart of Proposed Design Methodology [1]

Problem Formulation part has three parts, they are Needs definition, Requirements formulation and Feasibility assessment as shown in Figure 4.1. The Needs definition initializes the design process, where the supervisor introduces the concept and the problem that needs to be dealt with. The designer team checks all the requirements specified by the client in the Requirements formulation stage. The main aim of the Requirements formulation is to check whether the given functionalities can be implemented and constraints can be met. Feasibility assessment is the stage where we analyze the every part of the system individually, if we encounter any kind of problem in implementing the particular constrain

Functionalities		Particular constraints	Possible technologies & algorithms
General	Itemized		
Communication	Sending data to MIT app	Mobile internet	MIT app Inventor, ThingSpeak
	Sending data to ThingSpeak server	Wi-Fi	NodeMCU
Data analysis	Identifying abnormalities	Daily statistics and app functionality	ThingSpeak
Acquiring the data about "brushing teeth"	Core temperature	5% uncertainty, sampling period 1 s	LM35 Temperature sensor
	Time	Brushing time 2 min	NodeMCU

Table 4.1: User Driven Design

raised by the client, we take this issue with client and come to a conclusion about what to do to it.

Product Development stage is the other part of the design methodology where there are main sub phases like Technologies and Algorithms Selection, Modeling and Prototyping, Validation. While choosing the Algorithm and Technologies we must not only look for the efficiency but also the cost must be within the client budget. Modeling and Prototyping is the part where the models and prototypes must be backed by the both the designers and future users, whether there is a room for any improvements, bugs in the code, updating in the technology and is going on unit the clients need it to be. The final phase is the validation phase where the whole system is ready for deployment and then tested for system acceptability in the market.

From the User-Driven Design, the functionalities like communication, data analysis, core temperature measurement have been defined. To assure the communication functionality, we have to establish a communication between the temperature sensor placed at the toothbrush head and the mobile application or computer. The constraint is to store the core body temperature readings. The tools and algorithms used for communication purposes are NodeMCU [29] and C++ programming language [30]. Data analysis functionality requires a particular data form to interpret them. The plot of the temperature values in a graph against the day of the reading can be done using the ThingSpeak application. Core temperature measurement is the most rudimentary parameter of the experiment, with an uncertainty of 5%, and the minimum brush time must be 2 minutes. The temperature and time measurement are functionalities of importance.

The temperature sensor used is LM35, and the timer used is the built-in timer of the microcontroller. According to the proposed UDD, the system must assure communication between the temperature sensor, the microcontroller, and the client-side server. It should measure core body temperature and transmit the readings and be able to analyze the obtained temperature values and plot the graph of the temperature obtained against the time. Through this, one can quickly identify the curves to find out the abnormalities. We use an Internet-based IoT analytic platform ThingSpeak to analyze the data. It also transmits the temperature at 1 reading per second, with 5% uncertainty. The whole process must go on for an average brush time of 2 minutes.

Figure 4.2 shows the sequence of steps performed during the brushing process by the designed system. The timer starts when the program is uploaded and gets executed in the microcontroller. The temperature sensor continuously records the values seven times throughout the brush time. Based on the time, there are different tones for the buzzer. When the time is more than 30 seconds and less than 1 minute, it gives one kind of sound, and as the time is more than 1 minute and less than 2 minutes, it provides a different sound, and when the time is more than 2 minutes, it gives another different sound. The ThingSpeak server consolidates all the data. The WiFi enabling module which is inbuilt the NodeMCU transmits the data to the destined server. The MIT app receives the data, and the app displays the statistics.

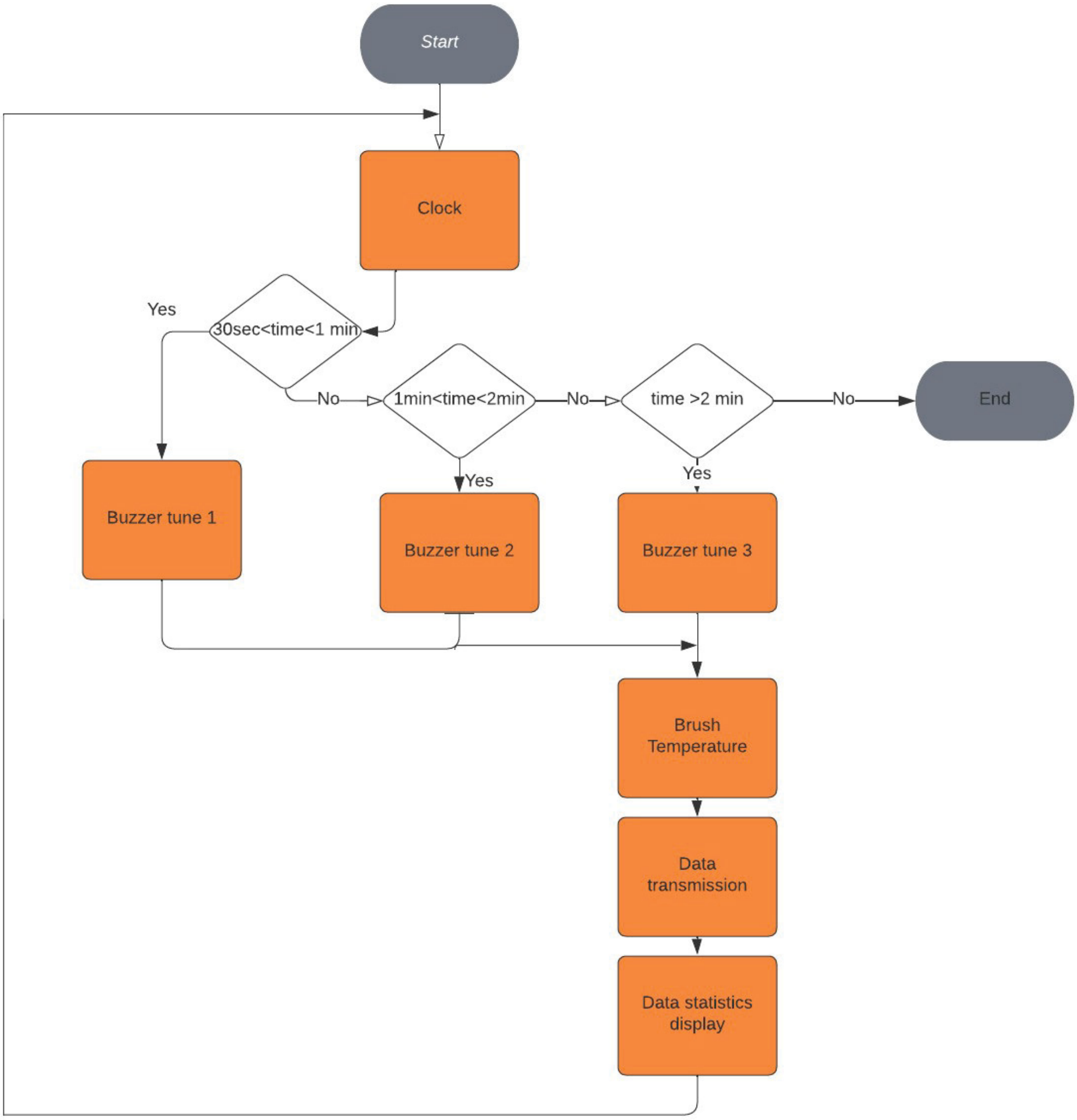


Figure 4.2: Flowchart of the Design

Figure 4.3 depicts the NodeMCU board, LM35 [31] temperature sensor, buzzer and ThingSpeak server. We presented the inputs on the left side of the NodeMCU block and the outputs to the right side of the NodeMCU block. The temperature sensor records the values and sends to the NodeMCU board, then according to the code, the buzzer makes different frequencies sounds at the different intervals of time. The NodeMCU board sends the data to the ThingSpeak server [32], and the

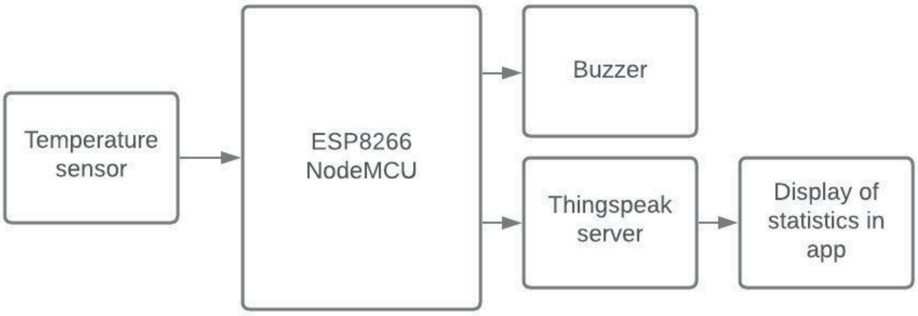


Figure 4.3: Structure of the Tracking "Brushing Teeth" Activity System

data are transferred to the app, the app displays the graphs of the consolidated values.

Chapter 5

Implementation

This chapter discusses the implementation of the system. Section 5.1 provides the circuit diagram of the system and technical information about the components. Section 5.2 provides the code for the entire system and set up of the ThingSpeak server.

5.1 Hardware implementation

5.1.1 Circuit diagram

Now, we will assemble all the components as shown in Figure 5.1 The circuit connections are as follows:

Pin 1 of the LM35 goes into +3V of the NodeMCU.

Pin 2 of the LM35 goes into analog pin A0 of the NodeMCU.

Pin 3 of the LM35 goes into the ground pin (GND) of the NodeMCU.

Buzzer one pin to the ground of NodeMCU.

Another pin of the buzzer is connected to pin D5 of NodeMCU.

We can use the transistors [33] and resistors also, but we can also build the circuit without having them since we get the same results. We built the circuit without using the transistors since they make the hardware expensive.

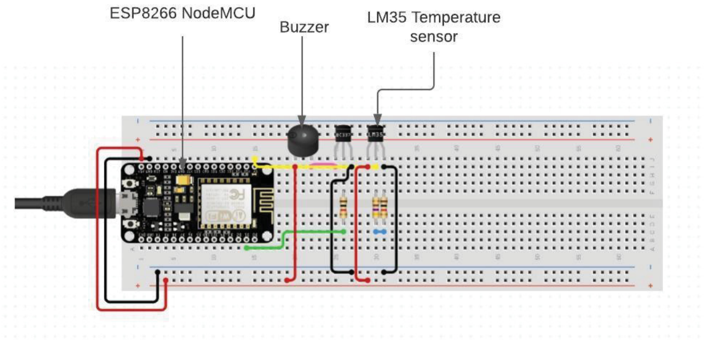


Figure 5.1: Circuit diagram of the design

5.1.2 Technical specifications of the components

NodeMCU: The development board equips the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa [34] 32-bit LX106 RISC microprocessor, which operates at 80 to 160 MHz adjustable clock frequency [35] and supports RTOS. As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with an LDO voltage regulator to keep the voltage steady at 3.3V. The ESP8266 NodeMCU features two buttons [36]. One marked as $\boxed{RST}$ located on the top left corner is the Reset button, used to reset the ESP8266 chip [37]. The other $\boxed{FLASH}$ button on the bottom left corner is download button used while upgrading firmware. The ESP8266 NodeMCU has 30 pins that interface it to the outside world.

The system's validity depends on the microcontroller taken to generate the output. We had taken a NodeMCU module to operate the whole system. NodeMCU is an open-source IoT platform. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems and hardware, based on the ESP-12 module [38]. It is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266. It uses many open-source projects, such as Lua-cjson and spiffs [38]. So, the module can accurately transfer the readings because it can withstand harsh conditions.

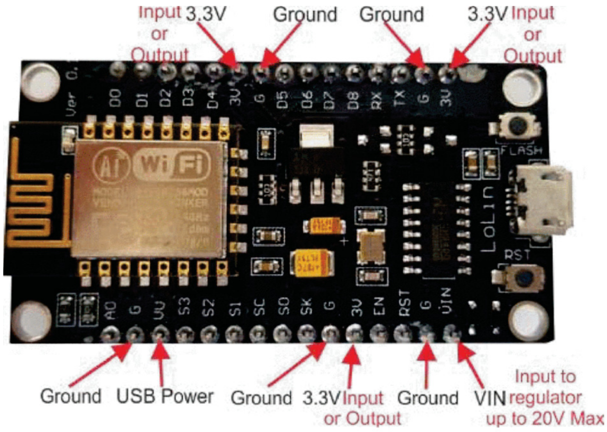


Figure 5.2: NodeMCU [2]

LM35 Temperature Sensor LM35 was used to measure temperature with an electrical output comparative to the temperature in °C. LM35 shows higher voltage values than thermocouples and may not need to amplify output voltage. The output voltage of LM35 [39] is proportional to the temperature. The scale factor is 0.01 V/°C.

LM35 has three pins which are:
PIN 1: Vcc - +5 V input voltage.
PIN 2: the output signal.
PIN 3: the ground.

Figure 5.3: LM35 sensor [3]

A Buzzer is a small yet efficient component to add sound features to our project/system. It is a tiny and compact 2-pin structure; hence can be easily used on the BreadBoard, PerfBoard, and even on PCBs, making this a widely used component in most electronic applications. For making beeps, alarms and tones, we use piezo buzzers. We can use them in alarm systems [40], for keypad feedback or some games.

The specification of the buzzer includes the following:
The frequency range - 3300 Hz.
Operating temperature range from –20°C to +60°C.
Operating voltage range from 3 V to 24 V DC.
The sound pressure level is 85 dBA.
The supply current below 15 mA.

Figure 5.4: Buzzer

The brush, we use is the Oral-B Genius 8000 electric toothbrush model. The model comes with a rechargeable handle and charger. The brush operates on

110 v - 120 V. We could mounted the LM35 temperature sensor on the backside of the brush head by soldering. Since we do not have technology to mount it correctly without damaging the sensor, we used the sensor directly without mounting it on the brush head.

Figure 5.5: Brush

5.2 Software implementation

Setting up ThingSpeak account for LM35 sensor:

First, we need to create a new channel by clicking the $\boxed{NewChannel}$ button. We can name it according to our preference by giving names to both axes.

Now click on the $\boxed{APIKeys}$ tab, which are the access keys to our channel using which we can write and read values. We use only the Write API Key generated for our channel. Therefore, we created a channel on ThingSpeak.

Now, we upload the code into the Arduino IDE.
1) Go to Tools;
2) Board > NodeMCU 1.0 (ESP - 12E Module);
3) Port (Choose the right Port);
4) Select baud rate "115200".

Here, we will change the API Write Key and SSID and password according to our connection, which we noted earlier. API Write Key is from the mentioned ThingSpeak channel. Now we will click the upload button. Then we can see the outputs in the serial monitor and the ThingSpeak server.

The code for the project is listed below:

```
#include <ESP8266WiFi.h>
String apiWritekey = "VJB26XDL5FBPVPAY";
const char* ssid = "POCOF1";
const char* password = "vamsi721" ;

```

```
const char* server = "api.thingspeak.com";
int sensorPin = 0;
int buzzer = D5;
WiFiClient client;
const unsigned long event_1 = 30000;
const unsigned long event_2 = 60000;
const unsigned long event_3 = 120000;
void setup() {
  Serial.begin(115200);
  pinMode(buzzer,OUTPUT);
  WiFi.disconnect();
  delay(10);
  WiFi.begin(ssid, password);

  Serial.println();
  Serial.println();
  Serial.print("Connecting␣to␣");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("NodeMcu␣connected␣to␣wifi...");
  Serial.println(ssid);
  Serial.println();
}

void loop() {

 int reading = analogRead(sensorPin);
 float voltage = reading * 3.3;
 voltage /= 1023.0;


 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;

  if (client.connect(server,80)) {
    String tsData = apiWritekey;
```

```
    tsData +="&field1=";
    tsData += String(temperatureC);
    tsData += "\r\n\r\n";

    client.print("POST␣/update␣HTTP/1.1\n");
    client.print("Host:␣api.thingspeak.com\n");
    client.print("Connection:␣close\n");
    client.print("X-THINGSPEAKAPIKEY:␣"+apiWritekey+"\n")
      ;
    client.print("Content-Type:application/x-www-form-
      urlencoded\n");
    client.print("Content-Length:␣");
    client.print(tsData.length());
    client.print("\n\n");
    client.print(tsData);

    Serial.print("Temperature:␣");
    Serial.print(temperatureC);
    Serial.println("uploaded␣to␣Thingspeak␣server....");
  }
  client.stop();

  Serial.println("Waiting␣to␣upload␣next␣reading...");
  Serial.println();
  if( millis() > event_1){
     tone(buzzer,1000);
    }
    if( millis() > event_2){
     tone(buzzer,4000);
    }
    if( millis() > event_3){
     tone(buzzer,6000);
    }
  delay(1000);
}
```

Chapter 6

Validation

6.1 Testing the buzzer

We test the buzzer for three cases. The test cases are divided based on the time instant of the test. From 30th second to 60th second, we enable an unique tone of the buzzer to acknowledge the user about the brush time. After that, we enable another kind of tone with a different frequency for the next 1 minute of the brush time. It indicates the user the completion of the two minute brush time. After 2nd minute, there is a different buzzer sound.

Test case 1 When the toothbrush is ON, the temperature sensor takes the readings at the particular region inside the mouth. With the help of the inbuilt WiFi of the *NodeMCU* module, we transmit the temperature recorded at the sensor to the dedicated electronic device by the mobile application, which supports WiFi communication [41]. We hear a unique tone for the time period from 30th second to 60th second different from the other tone for the remaining time.

Test case 2 After 1 minute of the brush time, there is a change in the buzzer's frequency and tone, indicating that it has passed a threshold amount of brush time. It will simultaneously receive and transmit the data between the sensor and the system in which we are storing the data.

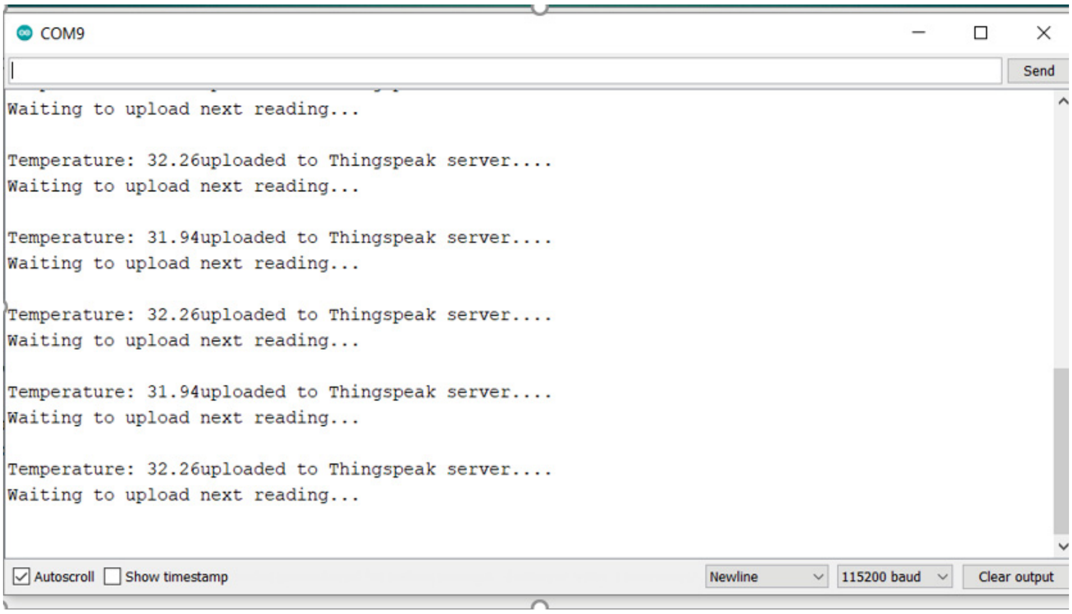
Test case 3 Once the brush time is more than two minutes, the buzzer changes its frequency and tone, signaling the users that they can stop the brushing as the time exceeds the average brush time prescribed by the dentists [42].

We uploaded the code for the buzzer into the Arduino IDE. The buzzer started to generate a tone from 30th second, and then the buzzer generated a different tone from 60th second. Then the buzzer started a different tone after completing the 2 minutes.

6.2 Verification of measurements

We recorded temperature using the LM35 temperature sensor. The range of the LM35 temperature sensor is -55°C to +150°C. We checked the range from 0°C to 45°C by keeping the sensor in the refrigerator and near hot air. When we kept the sensor near the refrigerator, we got the values from 0°C to 4°C and 40°C to 43°C when kept near hot air.

The output readings we get after uploading the code in the Arduino IDE in serial monitor are shown in Figure 6.1.



```
Waiting to upload next reading...

Temperature: 32.26uploaded to Thingspeak server....
Waiting to upload next reading...

Temperature: 31.94uploaded to Thingspeak server....
Waiting to upload next reading...

Temperature: 32.26uploaded to Thingspeak server....
Waiting to upload next reading...

Temperature: 31.94uploaded to Thingspeak server....
Waiting to upload next reading...

Temperature: 32.26uploaded to Thingspeak server....
Waiting to upload next reading...
```

Figure 6.1: Output in Arduino IDE

We have verified the total design by comparing the obtained temperature values with the typical human being values. Using the designed prototype, we recorded the output of the temperature sensor, see Figure 6.2. We see that the recorded temperature values are in a range of the "normal" oral temperature, which is from 35.73°C-37.41°C [43], [44].

As shown in Table 6.1, seven temperature samples were taken from an individual in 2 minutes of one brush time. To validate the design constrains, the measurement uncertainty in terms of standard deviation is used.

The relative uncertainty is given in terms of relative standard deviation $\sigma_{\%}$, denoted as:

$$\sigma_{\%} = (\sigma/\mu) * 100\% \tag{6.1}$$

where the standard deviation σ is defined as:

$$\sigma = \sqrt{\sum_{i=1}^{N}(x_i - \mu)^2/N} \tag{6.2}$$

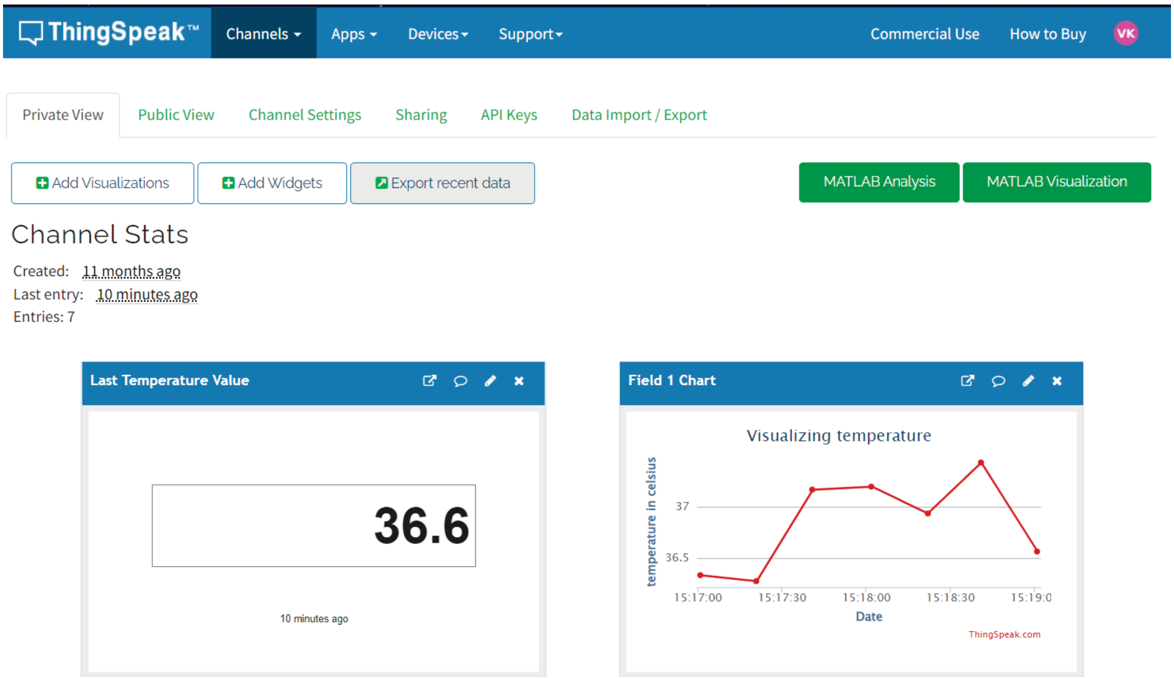
where:
μ - mean of the given samples
x_i - i-th sample value
N - a number of samples

In Table 6.1 the example of measurement analyse is shown.

Table 6.1: Uncertainty calculation

Temperature °C	Mean value μ	Relative standard deviation $\sigma_{\%}$	Constrain	Test result
36.3, 36.3, 37.2, 37.2, 36.9, 37.4, 36.6	36.8°C	1.1%	5.0%	Passed for 3σ criterion

Output in ThingSpeak server: After creating the channel, the ThingSpeak page looks as shown in Figure 6.2. We can adjust the channel by going into the channel settings. We can make the channel visible to only us by keeping it in private view. We can make the channel visible to all people by keeping it in public view. We can also add widgets by using the $\boxed{AddWidgets}$ button. We can export the data from this channel using the $\boxed{DataImport/Export}$ button. If we have any data with us already, we can also import the data into the channel using the $\boxed{DataImport/Export}$ button. We can even change the title of the graph. We can see the API key by clicking on the API keys button. The API key we used in the code is from ThingSpeak website.



Channel Stats

Created: 11 months ago
Last entry: 10 minutes ago
Entries: 7

36.6

10 minutes ago

Figure 6.2: Output in ThingSpeak Server

Output in mobile app: Figure 6.3 shows the view of the mobile interface, where we can see the temperature graph. The y-axis is about temperature values ans the x-axis is about the time and date. By using the $\boxed{+}$ and $\boxed{-}$ buttons we can even enlarge it if we have more differences in the temperature values. If we press on a dot in the graph, we can read the temperature value, date, and time of the reading. The below temperature value widget, there is the enlargement of the reading. The temperature values are round off to the one decimal place. The graph in Figure 6.3 shows an example of temperature measurements, which represents the example of historical record, available in the designed application.

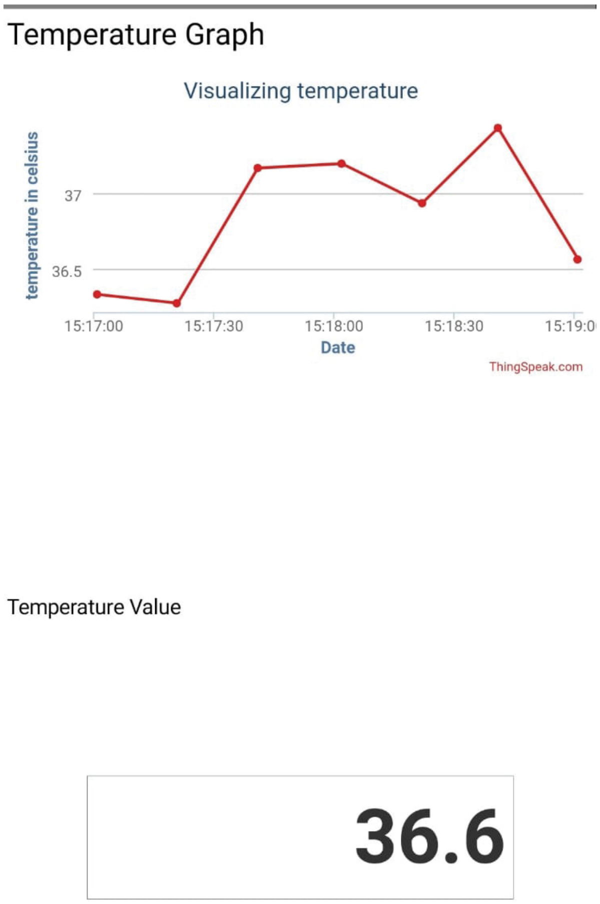


Figure 6.3: Output in Mobile App

Chapter 7

Conclusions and Future Work

This project aimed to design a system that helps people keeping track of their health by a means of their core body temperature. We successfully designed a system for monitoring the core body temperature using the LM35 temperature sensor. We determined that the the oral region is a body area where the core temperature can be measured in user-friendly and easily accessible way. The core body temperature read from the designed system can also be used in the health monitoring systems [45].

A statistical line graph that depicts the core body temperature is presented at a screen. By comparing the obtained temperature in the mobile application with the temperature observed using a digital thermometer we validated the designed system with respect to required constrains.

One of the important requirement mentioned in the User Driven Data table is to measure the temperature with 5% uncertainty. By calculating the relative error we proved that the uncertainty of the instrument is less than 5%.

The extensions of the thesis can be possible in many ways, such as enhancing the traditional LM35 temperature sensor with integrating a live video camera with the sensor. We can make use of other devices that can be aligned with the LM35, which give us more information regarding the body conditions like pH values, respiration, amount of oxygen levels. In the event of any undesirable, unpredictable events, there can be swift, reliable source of communication being established immediately, therefore alerting the nearby heath emergencies with the required information about the victim.

A voice alert [46] or mail alert can be issued regarding any jagged peaks inside the user's body. In cases of the patients with conditions of dementia [47] or Alzheimer's when the patients have trouble in memorizing the things around them, they can have a track of their day-to-day life, life events, and the history of their past healthy life [48].

With the advancement of the engineering and medical technologies, especially in electronics and computer engineering including artificial intelligence [49], machine learning [50] algorithms, neural networks [51], it will be possible to identify the user present health conditions [52] without the need of human monitoring [53] or additional support from the user [54].

References

[1] Damian Dziak, Bartosz Jachimczyk, and Wlodek J Kulesza. Iot-based information system for healthcare application: design methodology approach. *Applied sciences*, 7(6):596, 2017.

[2] Lamir Shkurti, Xhevahir Bajrami, Ercan Canhasi, Besim Limani, Samedin Krrabaj, and Astrit Hulaj. Development of ambient environmental monitoring system through wireless sensor network (wsn) using nodemcu and "wsn monitoring". In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–5. IEEE, 2017.

[3] Mitar Simić, Goran M Stojanović, Libu Manjakkal, and Krzystof Zaraska. Multi-sensor system for remote environmental (air and water) quality monitoring. In *2016 24th telecommunications forum (TELFOR)*, pages 1–4. IEEE, 2016.

[4] Lan Luo, Xichen She, Jiexuan Cao, Yunlong Zhang, Yijiang Li, and Peter XK Song. Detection and prediction of ovulation from body temperature measured by an in-ear wearable thermometer. *IEEE Transactions on Biomedical Engineering*, 67(2):512–522, 2019.

[5] Sandro Pinto, Jorge Cabral, and Tiago Gomes. We-care: An iot-based health care system for elderly people. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 1378–1383. IEEE, 2017.

[6] Luca De Nardis, Giuseppe Caso, and Maria Gabriella Di Benedetto. Thingslocate: a thingspeak-based indoor positioning platform for academic research on location-aware internet of things. *Technologies*, 7(3):50, 2019.

[7] Shahid Raza, Prasant Misra, Zhitao He, and Thiemo Voigt. Bluetooth smart: An enabling technology for the internet of things. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 155–162. IEEE, 2015.

[8] Xinfan Lin, Hector E Perez, Jason B Siegel, Anna G Stefanopoulou, Yonghua Li, R Dyche Anderson, Yi Ding, and Matthew P Castanier. Online parameterization of lumped thermal dynamics in cylindrical lithium ion batteries

for core temperature estimation and health monitoring. *IEEE Transactions on Control Systems Technology*, 21(5):1745–1755, 2012.

[9] Munirah Yaacob, Helen V Worthington, Scott A Deacon, Chris Deery, A Damien Walmsley, Peter G Robinson, and Anne-Marie Glenny. Powered versus manual toothbrushing for oral health. *Cochrane Database of Systematic Reviews*, (6), 2014.

[10] S Monisha, M Monisha, P Deepa, R Sathya, and K Gunasekaran. An android application for exhibiting statistical chronicle information. *International Journal of Communication and Computer Technologies*, 7(1):7–9, 2019.

[11] LEON LACK and Kurt Lushington. The rhythms of human sleep propensity and core body temperature. *Journal of sleep research*, 5(1):1–11, 1996.

[12] RAYMOND L VANDE WIELE, JEFF BOGUMIL, INGE DYRENFURTH, MICHEL FERIN, RAPHAEL JEWELEWICZ, MICHELLE WARREN, TAWFIK RIZKALLAH, and GEORGE MIKHAIL. Mechanisms regulating the menstrual cycle in women. In *Proceedings of the 1969 Laurentian Hormone Conference*, pages 63–103. Elsevier, 1970.

[13] Juan José Carmona and Shaday Michan. Biology of healthy aging and longevity. *Revista de investigacion clinica*, 68(1):7–16, 2016.

[14] John D Wark, Lucy Henningham, Alexandra Gorelik, Yasmin Jayasinghe, Stefanie Hartley, and Suzanne Marie Garland. Basal temperature measurement using a multi-sensor armband in australian young women: A comparative observational study. *JMIR mHealth and uHealth*, 3(4):e4263, 2015.

[15] Leonard F Blackwell, Pilar Vigil, María Elena Alliende, Simon Brown, Mario Festin, and Delwyn G Cooke. Monitoring of ovarian activity by measurement of urinary excretion rates using the ovarian monitor, part iv: the relationship of the pregnanediol glucuronide threshold to basal body temperature and cervical mucus as markers for the beginning of the post-ovulatory infertile period. *Human Reproduction*, 31(2):445–453, 2016.

[16] Prosanta Gope and Tzonelih Hwang. Bsn-care: A secure iot-based modern healthcare system using body sensor network. *IEEE sensors journal*, 16(5):1368–1376, 2015.

[17] Revathy Prasannan and Greeshma Sarath. Iot based device for fertility monitoring. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 813–817. IEEE, 2020.

[18] Shirin Enshaeifar, Payam Barnaghi, Severin Skillman, Andreas Markides, Tarek Elsaleh, Sahr Thomas Acton, Ramin Nilforooshan, and Helen Rostill. The internet of things for dementia care. *IEEE Internet Computing*, 22(1):8–17, 2018.

[19] Areej Almazroa and Hongjian Sun. An internet of things (iot) homecare management system using cardiac arrhythmia classification. In *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2021.

[20] Areej Almazroa and Hongjian Sun. An internet of things (iot) management system for improving homecare-a case study. In *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–5. IEEE, 2019.

[21] Sebastijan Sprager and Matjaz B Juric. An efficient hos-based gait authentication of accelerometer data. *IEEE transactions on information forensics and security*, 10(7):1486–1498, 2015.

[22] Noraini Azmi and Latifah Munirah Kamarudin. Enabling iot: Integration of wireless sensor network for healthcare application using waspmote. In *AIP Conference Proceedings*, volume 1808, page 020010. AIP Publishing LLC, 2017.

[23] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 483–488. IEEE, 2005.

[24] Xiong Xiong, Kan Zheng, Rongtao Xu, Wei Xiang, and Periklis Chatzimisios. Low power wide area machine-to-machine networks: key techniques and prototype. *IEEE Communications Magazine*, 53(9):64–71, 2015.

[25] Neil Cameron. Bluetooth communication. In *Arduino Applied*, pages 289–310. Springer, 2019.

[26] Daniel Balageas, Claus-Peter Fritzen, and Alfredo Güemes. *Structural health monitoring*, volume 90. John Wiley & Sons, 2010.

[27] Martin Randles, David Lamb, and Azzelarabe Taleb-Bendiab. A comparative study into distributed load balancing algorithms for cloud computing. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pages 551–556. IEEE, 2010.

[28] Shaileen Crawford Pokress and José Juan Dominguez Veiga. Mit app inventor: Enabling personal mobile computing. *arXiv preprint arXiv:1310.2830*, 2013.

[29] Andrej Škraba, Andrej Koložvari, Davorin Kofjač, Radovan Stojanović, Vladimir Stanovov, and Eugene Semenkin. Prototype of group heart rate monitoring with nodemcu esp8266. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4. IEEE, 2017.

[30] Simon Monk. *Programming Arduino: getting started with sketches*. McGraw-Hill Education, 2016.

[31] Cuihong Liu, Wentao Ren, Benhua Zhang, and Changyi Lv. The application of soil temperature measurement by lm35 temperature sensors. In *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, volume 4, pages 1825–1828, 2011.

[32] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, 4(5):1125–1142, 2017.

[33] Balraj Arunachalam, Amit Patvardhan, R Aiswarya, V Prasen, et al. Embedded temperature monitoring and control unit. In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*, pages 293–297. IEEE, 2010.

[34] Javier Acevedo, Robert Scheffel, Simon Wunderlich, Mattis Hasler, Sreekrishna Pandi, Juan Cabrera, Frank HP Fitzek, Gerhard Fettweis, and Martin Reisslein. Hardware acceleration for rlnc: A case study based on the xtensa processor with the tensilica instruction-set extension. *Electronics*, 7(9):180, 2018.

[35] Keith Bowman, Carlos Tokunaga, James Tschanz, Arijit Raychowdhury, Muhammad Khellah, Bibiche Geuskens, Shih-Lien Lu, Paolo Aseron, Tanay Karnik, and Vivek De. Dynamic variation monitor for measuring the impact of voltage droops on microprocessor clock frequency. In *IEEE Custom Integrated Circuits Conference 2010*, pages 1–4. IEEE, 2010.

[36] Prakhar Srivastava, Mohit Bajaj, and Ankur Singh Rana. Iot based controlling of hybrid energy system using esp8266. In *2018 IEEMA Engineer Infinite Conference (eTechNxT)*, pages 1–5. IEEE, 2018.

[37] Marco Schwartz. *Internet of Things with ESP8266*. Packt Publishing Ltd, 2016.

[38] Ravi Kishore Kodali and SreeRamya Soratkal. Mqtt based home automation system using esp8266. In *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 1–5. IEEE, 2016.

[39] A Goswami, T Bezboruah, and KC Sarma. Design of an embedded system for monitoring and controlling temperature and light. *International journal of electronic engineering research*, 1(1):27–36, 2009.

[40] Omar Asif, Md Belayat Hossain, Mamun Hasan, Mir Toufikur Rahman, Muhammad EH Chowdhury, et al. Fire-detectors review and design of an automated, quick responsive fire-alarm system based on sms. *Int'l J. of Communications, Network and System Sciences*, 7(09):386, 2014.

[41] Andrzej Sikora, Mateusz Krzysztoń, and Michał Marks. Application of bluetooth low energy protocol for communication in mobile networks. In *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–6. IEEE, 2018.

[42] Andrew Gallagher, Joseph Sowinski, James Bowman, Kathy Barrett, Shirley Lowe, Kartik Patel, Mary Lynn Bosma, and Jonathan E Creeth. The effect of brushing time and dentifrice on dental plaque removal in vivo. *American Dental Hygienists' Association*, 83(3):111–116, 2009.

[43] Adele Diamond, Carolyn T Lye, Deepali Prasad, and David Abbott. Correction: One size does not fit all: Assuming the same normal body temperature for everyone is not justified. *Plos one*, 16(10):e0259428, 2021.

[44] Ivayla I Geneva, Brian Cuzzo, Tasaduq Fazili, and Waleed Javaid. Normal body temperature: a systematic review. In *Open Forum Infectious Diseases*, volume 6, page ofz032. Oxford University Press US, 2019.

[45] Wataru Honda, Shingo Harada, Takayuki Arie, Seiji Akita, and Kuniharu Takei. Printed wearable temperature sensor for health monitoring. In *SENSORS, 2014 IEEE*, pages 2227–2229. IEEE, 2014.

[46] Eleanna Kafeza, Dickson KW Chiu, Shing-Chi Cheung, and Marina Kafeza. Alerts in mobile healthcare applications: requirements and pilot study. *IEEE Transactions on Information Technology in Biomedicine*, 8(2):173–181, 2004.

[47] Partha Pratim Ray, Dinesh Dash, and Debashis De. A systematic review and implementation of iot-based pervasive sensor-enabled tracking system for dementia patients. *Journal of medical systems*, 43(9):1–21, 2019.

[48] S Jayapradha and PM Durai Raj Vincent. An iot based human healthcare system using arduino uno board. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pages 880–885. IEEE, 2017.

[49] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in

healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.

[50] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

[51] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.

[52] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[53] Znaonui Liang, Gang Zhang, Jimmy Xiangji Huang, and Qmming Vivian Hu. Deep learning for healthcare decision making with emrs. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 556–559. IEEE, 2014.

[54] Maarten G Poirot, Praneeth Vepakomma, Ken Chang, Jayashree Kalpathy-Cramer, Rajiv Gupta, and Ramesh Raskar. Split learning for collaborative deep learning in healthcare. *arXiv preprint arXiv:1912.12115*, 2019.