Telegram Based Mobile Terminal for Body Temperature Data Storage of COVID-19 Patients

Putra Wisnu Agung Sucipto 1,*, Raden Hengki Rahmanto 2

1 Electronics Engineering Department; Universitas Islam 45 Bekasi; Jl. Cut Mutia no. 83 Bekasi, Jawa Barat, e-mail: wisnu@unismabekasi.ac.id
2 Mechanical Engineering Department; Universitas Islam 45 Bekasi; Jl. Cut Mutia no. 83 Bekasi, Jawa Barat, e-mail: hengki rahmanto@gmail.com

* Correspondence: e-mail: wisnu@unismabekasi.ac.id

Received: 4 August 2020; Review: 18 August 2020; Accepted: 1 September 2020; Published: 25 September 2020

Abstract

Health monitoring reports against COVID-19 suspect must be presented every day at 7 o'clock in the morning and 7 o'clock at night. In this report, every patient vital sign must be monitored in order to prevent a potential decline in the level of health from the patient. One of the symptoms observed in this monitoring is a change in body temperature. This monitoring should be carried out without touching the patient’s skin to reduce potential risks of droplets that contain the virus from spreading. This work proposes an idea related to non-contact temperature monitoring which is suitable to use in this pandemic condition. This idea realization process by building a data storage terminal for body temperature scans that is mobile, cheap and has a large storage capacity. This mobile terminal is an integration of the MLX90614 temperature sensor series, Arduino UNO and Telegram apps. In addition, the communication process between the sensor and the Telegram apps utilizes the GPRS cellular network services. Before arriving at the Telegram apps, this data will pass through gateway site which acts as a collector and relay. In the process of sending the temperature data is then converted into two body temperature categories, normal temperature and abnormal temperature. Based on the trial results that have been carried out, this system consumes 2A of current and 5V of voltage. The communication time duration between the temperature data acquisition system and the collector site is 28 seconds. While the length of time for sending the data from the collector site to the Telegram apps is 8 seconds.

Keywords: Temperature, MLX90614, Telegram, COVID-19

1. Introduction

COVID-19 disease control spread focuses on early detection capabilities and efforts to prevent the spread of infection through limiting social activities. Activity restrictions are pursued by requiring self-isolation for people who have been infected with or without symptoms. For people who were suspected with COVID-19 by category without symptoms who undergo self-isolation at home, their health monitoring is carried out by the Fasilitas Kesehatan Tingkat Pertama (FKTP) team. This team is in charge of conducting health monitoring, which includes monitoring body temperature by measuring body temperature every 7 o'clock in the morning and 7 o'clock in the night (Ministry of Health of the Republic of Indonesia 2020).

Measuring a human body temperature can be done by measuring the skin temperature and core body temperature. Measuring the core body temperature can be done by attaching a measuring instrument to the skin surface or placing it directly in the stomach. Measurements by attaching the tool to the skin surface were carried out using the LM73 sensor which is glued to aluminum to receive the body heat flux (Huang et al. 2017). Meanwhile, the measurement by attaching the measurement instrument directly into the stomach can be reached by placing a small thermometer in which the TMP112 sensor has been installed (Yoshida, Miyaguchi, and Nakamura 2017).
With its small size close to a nanometer, this thermometer is very practical to use. Apart from its size, the nanotechnology makes it easy to integrate into a wide variety of instruments, not just for measurement. Nonetheless, telecommunications equipment can also be installed that can send temperature scan data, from the inside to the outside of the human body and store it into smartphones and computers (Wong et al. 2020) (Lee et al. 2019) (Pereira, Medina, and Santos 2017) (Wu et al. 2017) (Chen et al. 2015) (Malekian 2015). As a connection, it is common to use RF, UHF, zigbee, and Bluetooth to transmit the scanned data to the final terminal (Chuo et al. 2010) (Vaz et al. 2010) (Malhi et al. 2012) (Wang et al. 2010).

However, even though the packaging of the tool is already very small so that it is easy to move and it is easy to receive the scan results directly to a cell phone, the process of measuring the temperature is still not right to be used when the current pandemic is still occur. Because the process of measuring the temperature is still touching human skin, so it is possible to reduce the efforts to control the spread of infection. So, a measuring device is needed that can work in a non-contact manner as described in (Song et al. 2018).

This research offers an alternative idea to help overcome the problem of body temperature scanning of COVID-19 patients which must be done according to the scanning schedule set by the FKTP team. This idea realization is by integrating non-contact temperature sensor technology using the MLX90614 sensor, GPRS services on cellular telecommunication networks and social media apps Telegram. The MLX sensor plays a role in scanning the temperature, which the results are then acquired by Arduino Uno and sent over the cellular telecommunications network by using the SIM900 module. This data will be sent to the Telegram application, after first being collected by a gateway site that acts as a relay. Based on the data, identity of the person who owns the temperature that were being scanned and the identity data of the Telegram account that has been registered first hand, this temperature data is then forwarded to the Telegram application as the final terminal for data receiver. The data that has been stored in the Telegram application is a reference data for monitoring and evaluating the patient's body temperature as one of the record for vital signs from body's health.

The combination of these three technologies is intended to build a data terminal that have mobile capability and affordable. By utilizing GPRS cellular network services, it will be able to minimize telecommunication constraints between patients at home and the FKTP team at the community health center (Puskesmas) and the National Task Force. In addition, the use of the Telegram application is increasingly drawing the attention considering that it is low cost or cheap and has the ability to store data with a very adequate capacity.

2. Research Methods
This research has undergone trials to obtain evidence of a complete system performance. The trial was carried out by taking a sample of human body temperature data directly and sending it to the Telegram application. Sampling of temperature data were taken by the MLX90614 sensor which is integrated within the Arduino Uno as its temperature data acquisition system. Furthermore, temperature data is passed to the SIM900 module to be superimposed on the cellular network via GPRS service using AT Command. The AT Command used in this activity is a data transfer instruction by using the TCP and HTTP GET Request protocols. This protocol was chosen because the regulated data must be received by the gateway site before being relayed to the Telegram application. The gateway site used in this trial is a site with the URL www.sarangsemut.online.

When the data has arrived at the gateway site, the data will be relay its journey to the Telegram application based on the chat id and token that was already registered before hand. Chat id and token are the identity of the destination driver in the form of Application Programming Interface (API) which is used for data transactions to the Telegram application. The Telegram API is obtained by registering a Telegram bot profile via BotFather on the Telegram application (Telegram 2020) Figure 1 illustrates a detailed block diagram of data transmission from the MLX90614 sensor to the Telegram application tested in this research.
Telegram Based Mobile Terminal for Body Temperature Data Storage of COVID-19 Patients

As illustrated in Figure 1, initially the temperature data acquisition process was carried out using the MLX90614 sensor. This sensor is a small infrared-based non-contact temperature sensor with a temperature reading range from -40 °C to +125 °C. The measurement resolution of this sensor is 0.02 °C (Melexis 2020).

This sensor works in an integrated manner with the ultrasonic sensor, buzzer and 128x32 OLED LCD. The integration with the ultrasonic sensor is intended to ensure that the scanning process is in accordance with the reading distance within a maximum range of 5 cm from the skin. If the presence of the sensor on the measured object is within a distance of 5 to 15 cm in a row, a notification is given in the form of a LED light as a marker. In addition, if the scanning process has been carried out, a buzzer will sound as a signal. In detail, Figure 2 shows a schematic of a data acquisition system circuit for measuring body temperature in this trial.

Figure 1. Block Diagram of Data Transmission from Sensor to Telegram Application

Figure 2. Schematic for Temperature Data Acquisition Circuit Using MLX90614 Sensor
Furthermore, the acquired temperature data is forwarded to the SIM900 module to be inserted in the AT Command before being sent to the gateway site. The AT Command statement used for TCP and HTTP GET Request protocols is AT + CIPSTART and AT + CIPSEND. AT + CIPSEND uses three input parameters, namely the type of protocol, IP (Internet Protocol) number and server port number. In this work the type of protocol used is TCP, with a blanket DNS IP number is the URL of the gateway site with the server port number is 80. As a whole, the details of the HTTP GET Request algorithm using AT Command in sending temperature data from the sensor to the gateway site are described in the following pseudocode.

ALGORITMA 1. Protokol HTTP GET Request Komunikasi data dengan instruksi AT+COMMAND via SIM900

DEKLARASI:
String link = "GET /sensor/KirimTelegram.php?nilai="+Suhu+" HTTP/1.1\r\nHost: www.sarangsemut.online\r\n\r"

DESKRIPSI:
AT+CREG=1
AT+CGATT=1
AT+CIPSHUT
AT+CIPMUX=0
AT+CSTT="internet"
AT+CSTT?
AT+CIICR
AT+CIFSR
AT+CIPSTART="TCP","www.sarangsemut.online",80"
AT+CIPSEND= >
Link
SEND OK
AT+CIPCLOSE

After the temperature data arrives at the site www.sarangsemut.online, this data is immediately forwarded to the Telegram application. The relay process to Telegram application is run by executing the CURL instruction on PHP. The parameters needed to carry out these instructions are the Telegram bot token data, chat id and the contents of the message to be sent to Telegram. Figure 3 shows a snippet of CURL instructions that are executed for the process of relaying messages to Telegram, based on the Telegram bot token data that has been registered with the name DomudManiaBot (some of the information is closed, to maintain the confidentiality of the bot's identity).

```php
<?php
define('BOT_TOKEN', 'untingKukupKucing2GJ1Ji84rcs="Tok"');
define('CHAT_ID', '123456');

function kirimTelegram($pesan) {
    $json = json_encode($pesan);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, TRUE);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $json);
    $result = curl_exec($ch);
    if ($result == 'OK') {
        echo 'Sukses kirim pesan ke Telegram.
    } else {
        echo 'Gagal kirim pesan ke Telegram.
    }
}

kirimTelegram('Hi, pesan ini dari sarangsemut.online. Suhu tubuh Anda, '+$_GET['Suhu']+' Derajat Celcius. Salam sehat, ya..!');
?>
```

Figure 3. CURL Instruction Code Snippet for Message Relay Process to Telegram
Finally, the data that has been received on the Telegram will be saved as a conversation message. This printing record is a collection of historical temperature data readings that have been sent remotely from the sensor. This message contains data that includes temperature values, hours and days of the scan from body temperature. The format for this message writing refers to the message content listed in the CURL instruction code snippet in Figure 3. The detail of the message statement is “Hi, pesan ini dari sarangsemut.online. Suhu tubuh Anda, xx.xx Derajat Celcius. Salam sehat, ya!!”.

At the end of this trial, several tests are carried out to evaluate the device performance. Device performance is reviewed based on the amount of electrical power used, review of temperature measurement results, GET HTTP Request testing for activation of GPRS service channels and sending data from sensors to gateway sites, and testing of temperature data transmission from gateway sites to telegrams. In testing this data transmission, a serial COM device is used to see the stages of the AT Command execution run by Arduino and SIM900. Apart from that, the message contents were also matched by using the CURL instruction and the contents stored in the message box of the Telegram application.

3. Results and Discussion

The electronic components described in Figure 2 have been assembled on a PCB to create a prototype that will undergo testing procedure. Figure 4 shows the hardware temperature scanning board prototype, which can be seen without the cover frame. This prototype is provided with a battery power source with a voltage of 9V. In addition, the MLX90614 sensor is connected to an ultrasonic sensor, Arduino Uno and a ringer for the temperature scanning process. Also, the SIM900 module has also been connected to the Arduino Uno to carry out the data transmission process.

![Temperature Scanner Prototype With MLX90614 Sensor](image)

When operating, this prototype consumes a current of 2A at a voltage of 5V. The nominal current of 2A refers to the amount of current consumption required by the SIM900 during the transmission process. If it is not in an active transmission condition, the SIM900 is in sleep mode with a current absorption of less than 2A. Thus, this condition allows this prototype to remain in an energy efficient state.

Temperature measurement tests have shown significant results. Body temperature can be measured non-contact without touching the skin of the object that being observed. Non-contact measurements with a maximum distance of 5cm from the skin have been carried out, and the results can also be displayed on the OLED LCD screen. Based on the scanning distance test, the distance indicator light has been able to show the difference between on-off lights which shows the range of the sensor's proximity to the skin during the process of adjusting the object's position to the MLX90614 sensor before starting the measurement. In addition, at the end of the scanning process the ringer can create a sound as a sign that the
scan has been successfully carried out and the temperature measurement results have been obtained.

On the other hand, the process of sending temperature data remotely by running the sending data protocol described in algorithm 1 has been successfully carried out. The test results show that, the transmission process per protocol stage can be run by SIM900. SIM900 has been able to deliver data from the source and to the destination address correctly and precisely. Starting with the network registration process, the AT + CREG and AT + CGATT instructions have successfully opened a communication channel through the GPRS service and initiated the TCP communication protocol. SIM900's request for IP has also been answered by the cellular network by providing an answer to the IP number, namely 10.197.245.132. For all requests executed by AT Command, the cellular network has given an OK answer which indicates that this prototype has been accepted by the cellular network and given access for data transmission. Table 1 shows in detail the results of the network registration test and answers to requests sent and received by SIM900 during the initiation process into the cellular network via GPRS service.

Table 1. Network Registration Test Results

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Meaning</th>
<th>Answer</th>
<th>Reply Time (millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AT+CREG=1</td>
<td>Activates network registration</td>
<td>OK</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>AT+CGATT=1</td>
<td>Entering gprs service</td>
<td>OK</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>AT+CIPSHUT</td>
<td>Disable GPRS</td>
<td>SHUT OK</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>AT+CIPMUX=0</td>
<td>Activate single IP connection</td>
<td>OK</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>AT+CSTT=&quot;internet&quot;</td>
<td>Settings APN sim card</td>
<td>OK</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>AT+CSTT?</td>
<td>Ensure the APN settings are correct</td>
<td>+CSTT: &quot;internet&quot;,&quot;&quot;,&quot;&quot;</td>
<td>5000</td>
</tr>
<tr>
<td>7</td>
<td>AT+CIICR</td>
<td>Initiate GPRS connection</td>
<td>OK</td>
<td>10000</td>
</tr>
<tr>
<td>8</td>
<td>AT+CIFSR</td>
<td>Ask for IP</td>
<td>10.197.245.132</td>
<td>5000</td>
</tr>
</tbody>
</table>

In addition to successfully entering the cellular network, the prototype has successfully transmitted data from sensors to the gateway site. Referring to the HTTP GET Request test that has been done, the connection experiment until sending the link to the site server www.sarangsemut.online has been successfully carried out. At this stage, this prototype has been able to deliver temperature data to move into the gateway site. Furthermore, referring to the AT + CIPSTART instruction test followed by the AT + CIPSEND instruction, SIM900 has received a SEND OK answer from the server which means that this prototype has been able to run an HTTP GET Request to the server which contains a call to a file named kirimTelegram.php which is stored on server as specified in the destination URL, and also create temperature data link to the sent URL parameter so that it can be read by the server. Table 2 describes in detail the stages of the HTTP GET Request to the gateway site with the input parameter in the form of url www.sarangsemut.online on server port number 80 and the Ok answer obtained.

Table 2. Details of HTTP GET Request Stages to Gateway Sites

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Meaning</th>
<th>Answer</th>
<th>Reply Time (millisecond)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AT+CIPSTART=&quot;TCP&quot;,&quot;&quot;,'www.sarangsemut.online&quot;,&quot;&quot;,&quot;&quot;,80&quot;</td>
<td>Activates connection with server</td>
<td>OK</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>AT+CIPSEND=95</td>
<td>Send link length</td>
<td>&gt;</td>
<td>Instantly</td>
</tr>
<tr>
<td>3</td>
<td>GET /sensor/KirimTelegram.php?nilai=&quot;Suhu+&quot; HTTP/1.1\nHost: <a href="http://www.sarangsemut.online">www.sarangsemut.online</a></td>
<td>Send an HTTP GET request</td>
<td>SEND OK</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>AT+CIPCLOSE*</td>
<td>Closed connection</td>
<td>OK</td>
<td>Instantly</td>
</tr>
</tbody>
</table>

After being able to send PHP file calling requests and URL parameters containing temperature data, this prototype has also succeeded in relaying temperature data to the...
Telegram application. The kurimTelegram.php file that is sent in the HTTP GET Request by this prototype, can be understood by the server as evidenced by the CURL instruction execution process. The CURL instruction contained in the kurimTelegram.php file is a statement of the process of relaying messages to the Telegram application. As the message narrative has been stipulated in this research method, the gateway site server has been able to forward this message using the CURL instruction contained in the kurimTelegram.php file and gives the answer status Return to SIM900. With the response to Pesan Telegram Terkirim received by Serial COM, testing the CURL instruction shows that this prototype has completely succeeded in sending data to the Telegram application in a complete data. Table 3 shows in detail the CURL instruction trial by the gateway site server and the answers received by SIM900 for the transmission process that occurred and the display of the contents received on the Telegram application message box.

Table 3. Testing of CURL Instructions by the Gateway Site Server and the Answer Received by SIM900 and the Display of the Contents of the Telegram Application Message Box

<table>
<thead>
<tr>
<th>Display</th>
<th>Testing 1</th>
<th>Testing 2</th>
<th>Testing 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial COM</td>
<td>HTTP/1.1 200 OK</td>
<td>HTTP/1.1 200 OK</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td></td>
<td>X-Powered-By: PHP/7.2.33</td>
<td>X-Powered-By: PHP/7.2.33</td>
<td>X-Powered-By: PHP/7.2.33</td>
</tr>
<tr>
<td></td>
<td>Content-Type: text/html; charset=UTF-8</td>
<td>Content-Type: text/html; charset=UTF-8</td>
<td>Content-Type: text/html; charset=UTF-8</td>
</tr>
<tr>
<td></td>
<td>Date: Thu, 24 Sep 2020 11:33:51 GMT</td>
<td>Date: Thu, 24 Sep 2020 11:33:59 GMT</td>
<td>Date: Thu, 24 Sep 2020 11:34:07 GMT</td>
</tr>
<tr>
<td></td>
<td>Server: LiteSpeed</td>
<td>Server: LiteSpeed</td>
<td>Server: LiteSpeed</td>
</tr>
<tr>
<td></td>
<td>X-Powered-By: PleskLin</td>
<td>X-Powered-By: PleskLin</td>
<td>X-Powered-By: PleskLin</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 27</td>
<td>Content-Length: 27</td>
<td>Content-Length: 27</td>
</tr>
<tr>
<td></td>
<td>Connection: keep-alive</td>
<td>Connection: keep-alive</td>
<td>Connection: keep-alive</td>
</tr>
<tr>
<td>Telegram</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Telegram Message Sent</td>
<td>Telegram Message Sent</td>
<td>Telegram Message Sent</td>
</tr>
</tbody>
</table>

Based on the appearance of the Telegram application described in Table 3, the narrative of the message received by the Telegram confirms the correctness of the message content sent by the CURL instruction as described in Figure 3. The contents of the messages sent and received are identical, so it can be said that this prototype has the ability to properly and correctly forward the message content from the server to the Telegram application. Besides that, in terms of time the performance of this prototype tends to work with a relatively fast duration. Referring to Tables 1, 2 and 3, the length of time required for this prototype to establish initial communication between the temperature data acquisition system and the collector site is 28 seconds, the HTTP GET Request to run the CURL instruction by SIM900 is 1.1 seconds, while the length of time sending data since CURL is executed by the gateway system to the Telegram application received by Serial COM as shown in Table 3 is for 8 seconds. This means that it only takes less than 10 seconds to deliver temperature data from the sensor to the Telegram application even though it has to pass through a gateway when it is connected to the cellular network.

4. Conclusion

This research has been able to provide suggestions for an alternative idea in providing a data terminal that have mobile capability and affordable. Although low cost or cheap, this research shows that the prototype is reliable and accurate. In addition, testing has shown that the Telegram application can be used as an alternative medium for storing and monitoring the body temperature of COVID-19 patients.

References


IEEE Transactions on Biomedical Circuits and Systems 4, no. 5: 281–94.


