IOT-BASED THERMAL DATA LOGGING AND USER IDENTIFICATION

B.Sc. Project

By Piyush Pattanayak

Guided By Rashmirani Panda and Arijit Ghosh

Date:04.05.2024

OUTLINES

- INTRODUCTION
- OBJECTIVES
- PREREQUISITES
- MODULE OVERVIEW
- DATA IMPLEMENTATION AND INTEGRATION
- CONCLUSION AND FUTURE WORK

Introduction

Introduction

- In today's interconnected world, efficient data collection and analysis are crucial for various applications, including healthcare, security, and inventory management.
- Our project aims to combine RFID technology with temperature sensing capabilities to create a comprehensive monitoring system.
- By integrating with Google Sheets, we make easy for seamless data logging and analysis, enabling real-time insights and decision-making.

Objectives

- Capturing real-time thermal data
- User identification with personalized data logging and analysis
- User-friendly interface
- Exploring potential applications

Introduction

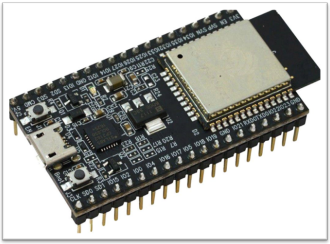
Prerequisites

Prerequisites

- Microcontroller(ESP32)
- Temperature Sensor(MLX90614)
- RFID(PN532)
- OLED(SSD1106)

ESP32

- ESP32 is a low-cost, low-power microcontroller chip developed by Espressif Systems.
- It has integrated Wi-Fi and Bluetooth connectivity.
- The ESP32 provides many GPIO pins that facilitate connection with and control of external devices and sensors

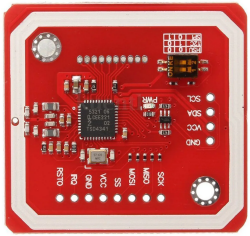


MLX90614

- The MLX90614 is an infrared thermometer for non-contact temperature measurements.
- The MLX90614 can be used to measure the temperature of a particular object ranging from -70° C to 382.2°C.
- The sensor uses IR rays to measure the temperature and communicates to the microcontroller using the I2C protocol.

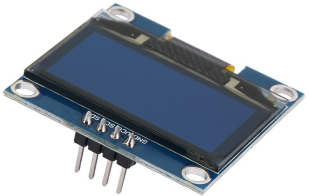
RFID PN532

- The PN532 is a highly integrated NFC (Near Field Communication) controller from NXP Semiconductors.
- RFID (Radio Frequency Identification) is a technology that allows for the wireless identification of physical objects using radio waves



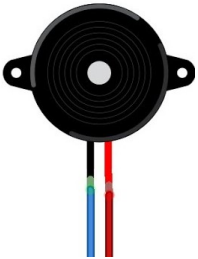
OLED SSD1106

- The SSD1106 is a popular OLED (Organic Light-Emitting Diode) display controller chip manufactured by Solomon Systech.
- It is widely used in various electronic devices, especially in small-scale applications like wearable devices, IoT gadgets and DIY electronics projects.



Buzzer

- A buzzer or beeper is an audio signaling device.
- The most common uses include alarm devices, timers, train and confirmation of user input such as a mouse click or keystroke.



Module Overview

Module Overview

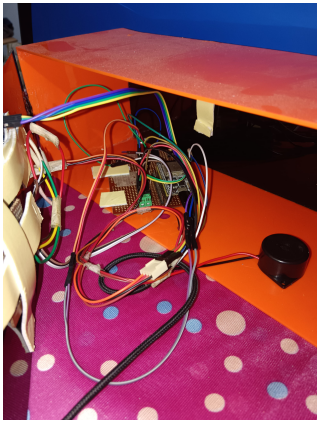
CONNECTION

PINOUTS

OLED	ESP32	RFID	ESP32	MLX90614	ESP32
SDA	GPIO21	SDA	GPIO21	SDA	GPIO21
SCL	GPIO22	SCL	GPIO22	SCL	GPIO22
VCC	3V3	VCC	3V3	VCC	3V3
GND	GND	GND	GND	GND	GND

(1)

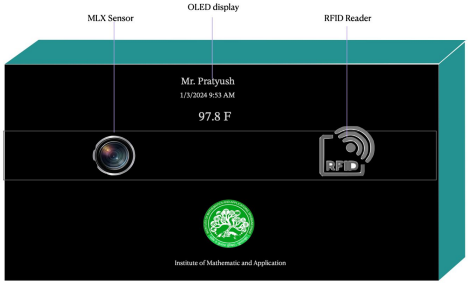
CONNECTION DIAGRAM



Module Overview

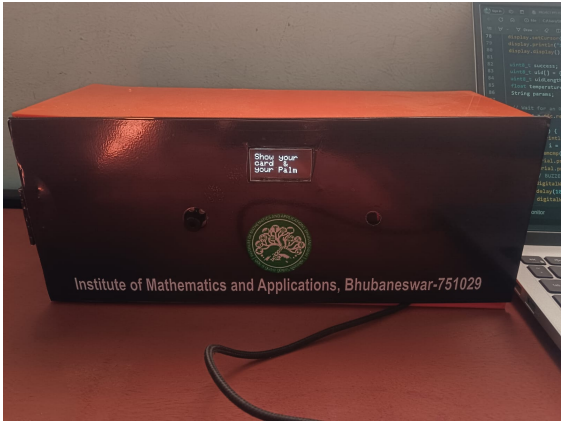
MODULE

Module


MLX Sensor
OLED display
RFID Reader
Mr. Pratyush
1/3/2024 9:53 AM
97.8 F
RFID
Institute of Mathematic and Application


MLX Sensor
OLED display
RFID Reader
Mr. Pratyush
1/3/2024 9:53 AM
97.8 F
Institute of Mathematic and Application

(2)


Show your
card &
your Pain
Institute of Mathematics and Applications, Bhubaneswar-751029

Data Implementation and Integration

Data Implementation and Integration

Implementation

Arduino IDE

- The Arduino IDE(Integrated Development Environment) is a software platform used for programming Arduino microcontrollers.
- The code can be used for lights, sensors and many electronic components.
- It has a Simple interface & offer libraries for common tasks.

ARDUINO CODE

SHEET.ino

```
#include <Wire.h>
#include <Adafruit_SH110X.h>
#include <Adafruit_PN532.h>
#include <Adafruit_MLX90614.h>
#include <HTTPClient.h>
#include <WiFi.h>
#define SDA_PIN 21 // Define your SDA pin here
#define SCL_PIN 22 // Define your SCL pin here
#define i2c_Address 0x3C // Change this to the desired I2C address

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET -1
Adafruit_SH1106G display = Adafruit_SH1106G(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
Adafruit_PN532 nfc(SDA_PIN, SCL_PIN);

const char* ssid = "PIYUSH";
const char* password = "12345678";
const char* GOOGLE_SCRIPT_ID = "AKfycbwF9okznQ1p491FANXnTnA_bon4748tMolF9im9wxRUZZvCfrCzmp5YGkeU494Yu6Vwtw";

const int buzzerPin = 23;

```

SHEET.ino

```
24  struct RFIDTag {
25    uint8_t uid[4]; // UID of the tag
26    uint8_t uidLength; // Length of the UID
27    const char* name; // Name associated with the tag
28  };
29
30  // Define an array of RFID tags and their associated names
31  RFIDTag knownTags[] = {
32    {0x73,0x83,0x36,0xFE},4,"Piyush"},
33    {0xB3,0x8F,0x3F,0xFE},4,"Madhav"},
34    {0x3,0x40,0x42,0xFE},4,"Sandip"},
35    {0x53,0x85,0x3E,0xFE},4,"Ashish"},
36    {0x83,0x29,0x3E,0XFE},4,"Asim"},
37    {0xC3,0XF3,0X92,0XF5},4,"Maharshi"},
38    {0xB3,0x73,0x2C,0xFE},4,"Bibhu"},
39    {0x93,0x89,0x2D,0xFE},4,"Ananta"},
40    {0x23,0xBA,0x2C,0xFE},4,"Sambit"},
41    {0xC3,0x5B,0x2D,0xFE},4,"Ashutosh"},
42    {0xF3,0x22,0x2A,0xFE},4,"Shiv"}
43  };
44
```

SHEET.ino

```
45  void setup() {
46    Serial.begin(115200);
47
48    Wire.begin(SDA_PIN, SCL_PIN); // Initialize I2C with defined SDA and SCL pins
49    mlx.begin();
50    nfc.begin();
51
52    if (!display.begin()) { // Initialize SH110X display
53      Serial.println(F("Something went wrong"));
54      for (;;);
55    }
56
57    display.clearDisplay();
58    display.setTextSize(2);
59    display.setTextColor(SH110X_WHITE);
60
61    // Connect to Wi-Fi
62    Serial.print("Connecting to ");
63    Serial.println(ssid);
64    WiFi.begin(ssid, password);
65    while (WiFi.status() != WL_CONNECTED) {
66      delay(500);
67      Serial.print(".");
68    }
69    Serial.println("");
70    Serial.println("WiFi connected");
71
72  pinMode (buzzerPin,OUTPUT);
```

SHEET.ino

```
76  void loop() {
77    display.clearDisplay();
78    display.setCursor(0, 16);
79    display.println("Show your card  &   your Palm");
80    display.display();
81
82    uint8_t success;
83    uint8_t uid[] = { 0, 0, 0, 0, 0 };  // Buffer to store the returned UID
84    uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on ISO14443A card type)
85    float temperature = mlx.readObjectTempC();
86    String params;
87
88    // Wait for an RFID card
89    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength);
90
91    if (success) {
92      Serial.println("Found an RFID tag!");
93      for (int i = 0; i < sizeof(knownTags) / sizeof(knownTags[0]); i++) {
94        if (memcmp(uid, knownTags[i].uid, uidLength) == 0) {
95          Serial.print("HELLO ");
96          Serial.println(knownTags[i].name);
97          // BUZZER SOUND
98          digitalWrite(buzzerPin, HIGH);
99          delay(1000);
100         digitalWrite(buzzerPin,LOW);
```

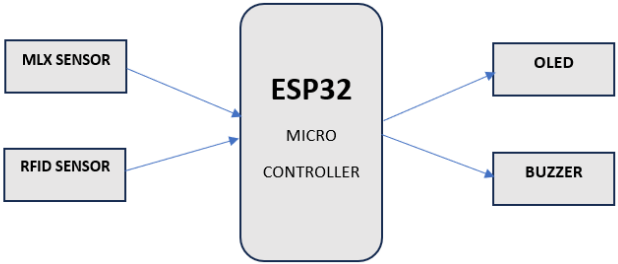
SHEET.ino

```
101         // RFID DATA
102         display.clearDisplay();
103         display.setCursor(0, 16);
104         display.print("Hello ");
105         display.println(knownTags[i].name);
106         display.display();
107         delay(2000);
108
109         // MLX DATA
110         display.clearDisplay();
111         display.setCursor(0, 16);
112         display.print("Temperature: ");
113         display.print(temperature);
114         display.println(" C");
115         display.display();
116         delay(3000);
117         //THANKS GIVING
118         display.clearDisplay();
119         display.setCursor(0,16);
120         display.print("Thank you ");
121         display.println(knownTags[i].name);
122         display.display();
123         display.clearDisplay(); // Delay for visibility, adjust as needed
124         delay(3000);
125
```

SHEET.ino

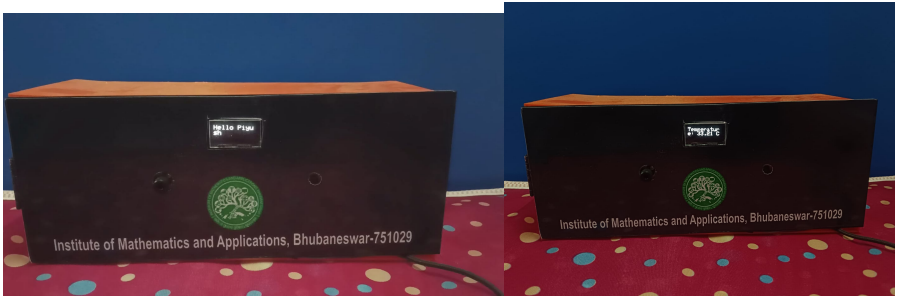
```
    // Prepare parameters for sending to Google Sheets
    params = "UID=";
    for (uint8_t i = 0; i < uidLength; i++) {
      params += String(uid[i], HEX);
    }
    params += "&Temperature=" + String(temperature);

    // Send data to Google Sheets
    sendData(params);

    delay(1000);
  }
}

void sendData(String params) {
  HTTPClient http;
  String url = "https://script.google.com/macros/s/" + String(GOOGLE_SCRIPT_ID) + "/exec?" + params;

  Serial.print("Requesting URL: ");
  Serial.println(url);

  if (http.begin(url)) {
    int httpCode = http.GET();
    http.end();
    Serial.print("Response code: ");
    Serial.println(httpCode);
  } else {
    Serial.println("Failed to connect to Google Sheets");
  }
```

Transfer of data from Sensors



BY I2C COMMUNICATION

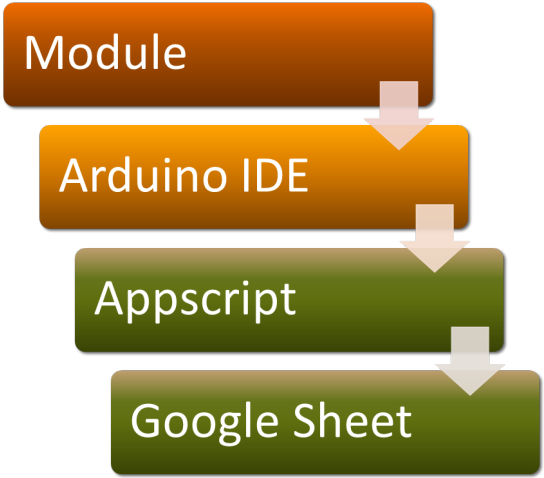
OUTPUT


Hello Piyu
Institute of Mathematics and Applications, Bhubaneswar-751029
Institute of Mathematics and Applications, Bhubaneswar-751029

(3)

Data Implementation and Integration

Integration


Module
Arduino IDE
Appscript
Google Sheet

Google Sheet

- Google Sheets is a free online tool for creating and sharing spreadsheets.
- It allows us to edit, share, and work together on data with formulas and charts in real time.

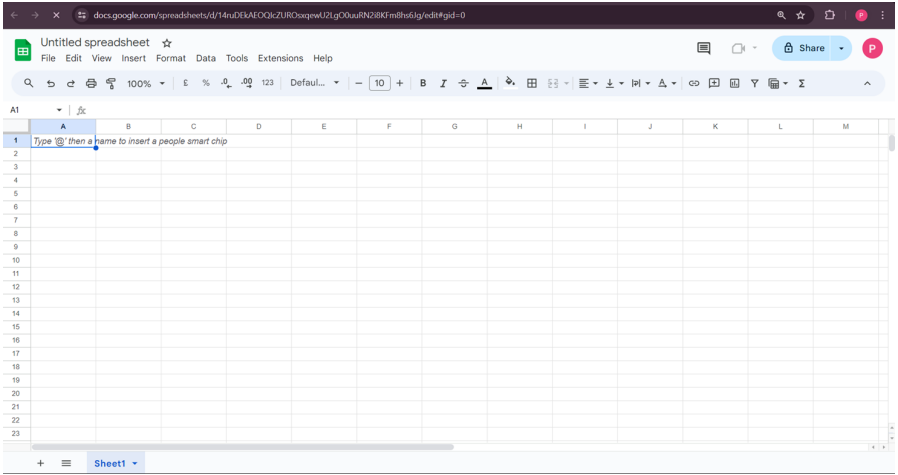


Figure 1: Create a new Sheet

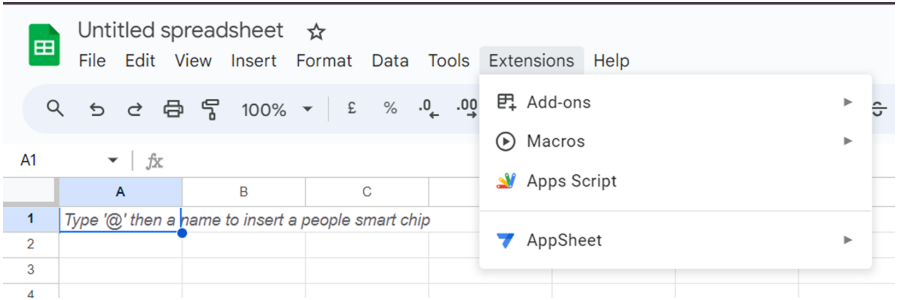


Figure 2: Click On Extensions then AppScript

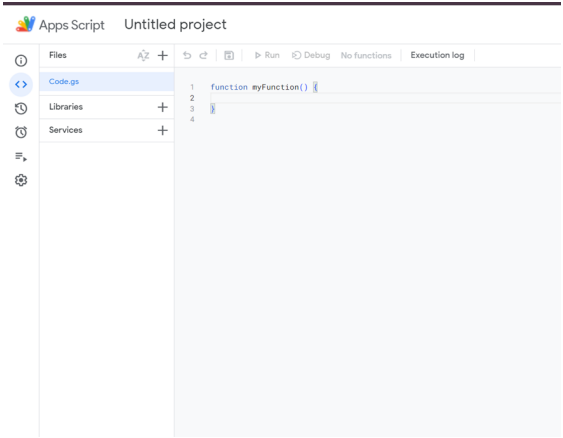


Figure 3: Copy the Require Script code and paste in Code

SCRIPT CODE

- The script code is written in JavaScript which serves as the back-end logic for web applications.
- In our project, the script code is used in conjunction with Google Apps Script to handle incoming requests from the ESP32 device and manipulate Google Sheets data accordingly.

```
var GOOGLE_SCRIPT_ID = "10EFDr8ay4xUXdsObm1zU90Y2Bt-sh5X6QXYbtcH6y70";

function doGet(e) {
  var sheet = SpreadsheetApp.openById(GOOGLE_SCRIPT_ID).getActiveSheet();
  var UID = e.parameter.UID;
  var temperature = e.parameter.Temperature;
  var timestamp = new Date();
  var names = {
    "738336fe": "Piyush",
    "b38f3ffe": "Madhav",
    "034042fe": "Sandip",
    "53853efe": "Ashish",
    "83293efe": "Asim",
    "c3f392f5": "Maharshi",
    "b3732cfe": "Bibhu",
    "93892dfe": "Ananta",
    "23ba2cfe": "Sambit",
    "c35b2dfe": "Ashutosh",
    "f3222afe": "Shiv"
    // Add more UID-name mappings as needed
  };

  sheet.appendRow([timestamp,names[UID], temperature]);

  return ContentService.createTextOutput("Data received successfully");
}
```

Figure 4: Required Code

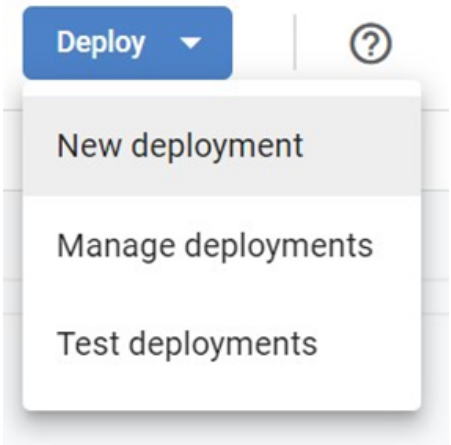


Figure 5: Deploy

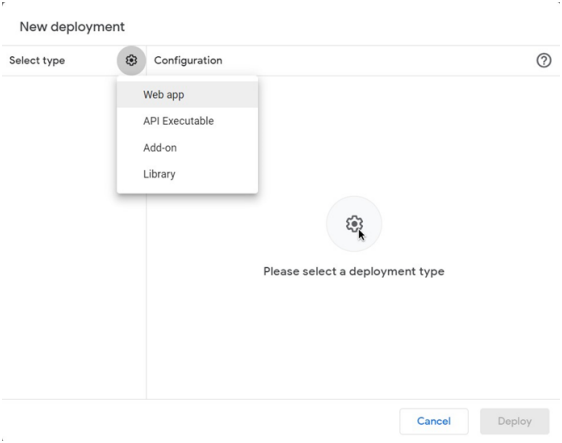


Figure 6: Deploy it as Web App

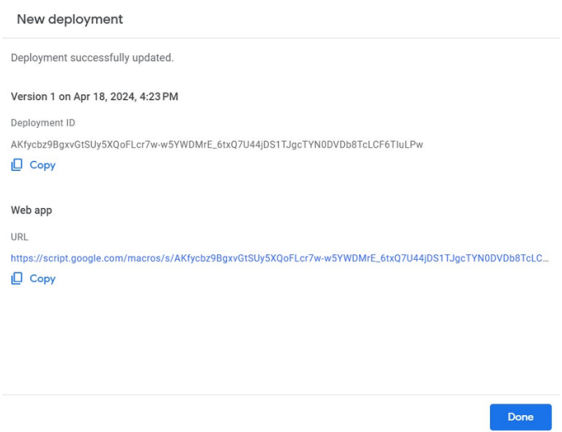


Figure 7: Deployment ID

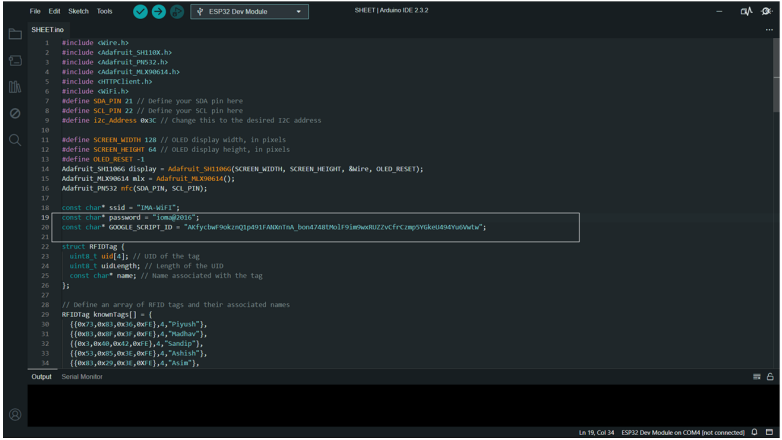


Figure 8: Copy the Deployment id and Paste it in Arduino IDE

OUTPUT

	A	B	C
1	DATE AND TIME	UID	TEMPERATURE
2	4/9/2024 13:02:41	Ashish	31.27
3	4/9/2024 13:03:09	Asim	31.27
4	4/9/2024 13:03:20	Maharshi	30.99
5	4/9/2024 13:03:52	Madhav	30.81
6	4/9/2024 13:17:30	Ananta	30.47
7	4/9/2024 13:18:56	Madhav	31.07
8	4/9/2024 13:21:22	Piyush	30.99
9	4/9/2024 17:50:15	Ashish	29.99
10	4/9/2024 17:50:43	Ananta	29.77
11	4/10/2024 19:01:1	Ashish	30.05
12	4/10/2024 19:03:3	Ashish	29.85
13	4/10/2024 19:15:5	Ashish	29.83
14	4/10/2024 19:27:4	Ashish	29.63

Conclusion

CONCLUSION

- Our project represents a successful combination of RFID technology, temperature sensing and cloud-based data management using the ESP32 microcontroller.
- Seamlessly integrating these components, we have created a versatile and efficient data collection system.
- Using Google Sheets as a back end database allows for convenient storage, retrieval and analysis of collected data.
- This system facilitates real-time monitoring and enables users to remotely access and analyze information.

FUTURE WORK

1. Enhanced data analysis
2. User Authentication
3. Mobile application
4. Cellular automata Integration

References I

[1] Asif Ahmed, Mohd Noor Abdullah, and Ishkrizat Taib. "Design of a contactless body temperature measurement system using Arduino". In: *Indonesian Journal of Electrical Engineering and Computer Science* 19 (Sept. 2020), p. 1251. DOI: 10.11591/ijeecs.v19.i3.pp1251-1258.

[2] Ridi Arif et al. "Use of the MLX 90164 sensor and the ThingSpeak platform for internet of things-based animal body temperature check". In: *ARSHI Veterinary Letters* 5.2 (Oct. 2021), pp. 35–36. DOI: 10.29244/avl.5.2.35-36. URL: https://journal.ipb.ac.id/index.php/arshivetlett/article/view/35221.

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.010. URL: https://www.sciencedirect.com/science/article/pii/S1389128610001568.

References II

[4] H.G Navin Kumar et al. "Literature review on "Hybrid temperature sensing and monitoring system with built-in sanitizer"". In: *International Advanced Research Journal in Science, Engineering and Technology* 7.12 (Dec. 2020), pp. 66–70. ISSN: 2393-8021(Online),2394-1588(Print). DOI: 10.17148/IARJSET.2020.71213.

[5] Puput Rusimamto et al. "Design and Implementation of Thermal Body System Employing Thermal Sensor MLX 90614 for Covid-19 Symptoms Early Detector". In: Jan. 2020. DOI: 10.2991/aer.k.201124.058.

THANK YOU