

# Designing an Internet Based Remote Diabetic Foot Wound Monitoring and Treatment System

Izyan Khan<sup>1</sup>, Micheal Buncek<sup>1</sup>, Zaid Ahmad<sup>2</sup>, Sayed Aamir<sup>3</sup>, Imteyaz Ahmad<sup>4,\*</sup>

<sup>1</sup>Lincoln Community School, Bayonne, USA

<sup>2</sup>Department of Computer Engineering, Zakir Hussain College of Engineering and Technology, Aligarh Muslim University, Aligarh, India

<sup>3</sup>Gwinnett Endocrinology, Snellville GA, USA

<sup>4</sup>Department of Pediatrics-Neonatology, Saint Peters University Hospital, Rutgers-Robert Wood Johnson Medical School, New Brunswick, USA

---

**Abstract** Chronic wounds are a major problem for diabetic people around the world. Sometimes when a diabetic person gets a wound and it becomes infected, it takes a long time to heal properly and – if left unnoticed and untreated – requires the limb to be amputated, usually happening to wounds on the foot of the diabetic person, thus called diabetic foot. To solve this problem, we devised a remote Internet of Things (IoT)-based system using an Arduino UNO Wi-Fi platform to remotely monitor and treat a wound, enabling physicians to provide an early diagnosis and medical intervention to prevent such amputations. This system records the pH, humidity, and temperature of the wound, which changes when a wound is getting infected by a bacterium and is becoming a chronic wound. Our system is also capable of incorporating newer evolving therapies by being able to actuate them, like sending electricity through wound crater by under skin wire to turn on an LED – broadly representing turning on an electrotherapy system that could be properly implemented later – or turning on a pump to deliver liquid medication of some kind, by a physician remotely switching them through an Internet of Things device if determined the therapy is needed. We made a silicone foot model to simulate diabetic foot. We made a crater to simulate a wound, tunneled wires inside, and placed an LED to demonstrate the actuation of the electrotherapy. We used anhydrous calcium chloride to generate a thermogenic reaction to simulate the increase in temperature, humidity, and pH changes, which are hallmarks of inflammation and infection. We would then check how the sensors recorded the data, comparing temperature, pH, and humidity from before and after the calcium chloride was added. Ultimately, we designed an innovative diabetic simulation model that uses an analog PH0-14 E-201-C pH sensor, a DS18B20 submersible temperature sensor, and a DHT11 humidity sensor to record data simulation infection; and wires connected to the Arduino's pins to actuate electrotherapy and power the pump to deliver liquid medicine remotely via an app which an iOS-based system, like an iPhone or iPad can access.

**Keywords** Diabetic foot, Cloud-based IoT device, Remote-monitoring system, Innovative early treatment

---

## 1. Introduction

Chronic wounds are a major problem for diabetic people around the world. Every year in the United States, about 120,000 amputations of the lower limb not related to trauma are performed on people with diabetes [1]. These amputations happen to people with uncontrolled diabetes because, according to the National Library of Medicine, their high blood sugar causes diabetic neuropathy, where they lose sensation in their nerves. When they get a cut or wound (usually on the foot), they do not notice or feel it, causing the wound to remain untreated and likely infected. This infection, along with the damaged blood vessels, causes the wound to become chronic and not heal, causing

the surrounding tissue to die and eventually requiring an amputation of the wounded limb. This commonly happens in the foot; thus, the name of this condition is diabetic foot [2].

Usually, to prevent this, diabetic people are encouraged to go regularly to the doctor to check for these worsening wounds, but for many reasons (usually not enough time or money), they do not. Because of this, many people who have difficulty controlling their diabetes may have a wound that they do not know is worsening and potentially threatening to their limbs.

However, if a wound could be monitored by a doctor remotely, and even could be treated remotely, this could greatly help many diabetic people and prevent them from being disabled from one of their limbs needlessly by giving doctors the ability to give an early diagnosis and medically intervene, curing the wound before it gets worse.

This paper aims to present an IoT-based diabetic foot online monitoring system and early treatment system to solve this problem, as a proof-of-concept design.

---

\* Corresponding author:

khanimteyaz001@gmail.com (Imteyaz Ahmad)

Received: Aug. 6, 2025; Accepted: Sep. 3, 2025; Published: Sep. 16, 2025

Published online at <http://journal.sapub.org/ajbe>

## 2. Background Study

### 2.1. Scope of Diabetic Foot and Impact of Treatment

Diabetic foot is a condition that causes a wound on a foot to become severely infected and incapable of proper healing, usually requiring the foot to be amputated for the patient's safety. It is caused mainly by two pathological processes due to diabetes: peripheral arterial disease and peripheral nerve damage. Peripheral arterial disease, in short, causes the blood vessels to thin and narrow causing poor blood supply (especially distal areas like the legs and feet), which can slow or stop the healing of a wound, while peripheral nerve damage causes loss of sensation in nerves, which causes wounds to go unnoticed, allowing for infections of an already unhealing wound to settle in quickly and cause severe damage due to the lack of treatment [3].

Signs of an infection occurring or worsening in a chronic wound are the increase in wound temperature, pus formation (measured by humidity), and in a change of pH, as noted by multiple papers [8], [9], and [10].

### 2.2. Related Works & Designs

The engineers of FlexiLab at Purdue University used an electrochemical sensor with an Arduino in their design to detect uric acid in the wound and determine its condition [4].

### 2.3. Electrotherapy Treatment of Diabetic Foot

To treat diabetic wounds, apart from the traditional treatment like liquid antibiotics, there are newer modalities like electrotherapy, which has shown that it can enhance healing of diabetic ulcers/wounds by flowing small amounts of current near the wound site, stimulating immune cells to react and try to heal and fight the infection sooner [5]. These are particularly useful through electronics, which can deliver said small current.

## 3. Design Specifications

This system is designed to remotely monitor and treat diabetic foot wounds using an IoT-enabled platform. It integrates sensors for real-time measurement of pH, temperature, and humidity, which are key indicators of wound infection. If signs of infection are detected, the system user can actuate electrotherapy or liquid medication remotely via an app-based interface, enabling early medical intervention and reducing the risk of amputation.

### 3.1. Materials Used

#### 3.1.1. Software Specifications

Firmware Features:

- Sensor data acquisition and calibration
- Wi-Fi transmission to cloud
- Remote control functions (electrotherapy, medication pump)

**Table 1.** Software Components

Software	Description
Arduino IDE	Used to program the microcontroller and integrate sensor logic.
Arduino IoT Cloud Platform	Enables device-to-cloud communication, real-time data streaming, and remote control.
Arduino IoT Remote App (iOS)	Allows doctors to monitor sensor readings and activate treatment options (LED or pump).

#### 3.1.2. Hardware Specifications and Materials

**Table 2.** Hardware Components

Component	Description
Arduino Uno R4 Wi-Fi	Microcontroller with built-in Wi-Fi & Bluetooth for IoT communication.
PH0-14 E-201-C Sensor	Analog sensor for detecting wound pH changes (range 0–14). Range of Error: $\pm 0.2\text{pH}$ [11]
DS18B20 Sensor	Waterproof digital sensor to monitor wound temperature. Range of Error: $\pm 0.5$ Celsius in the range -10 Celsius to +85 Celsius [12]
DHT11 Sensor	Measures humidity and ambient temperature for wound environment. Range of Error: $\pm 5\%$ RH in the range 20%~90% [13]
Pump (6V)	Delivers liquid medication remotely when activated.
Blue LED	Shows passage of current through under skin wire which broadly simulates the actuation of electrotherapy.
Calcium Chloride	Thermogenic agent to simulate infection/inflammation effects.
Breadboard & Connectors	For circuit prototyping and sensor integration.
Solar Power Bank	Enables portable, sustainable power supply.
USB-C Cable	For programming and power interface.
pH Buffer Packets (4 & 9)	For calibration of the pH sensor.
Plastic Medicine Cup (30 ml)	Medication reservoir for pump output.

Materials:



**Figure 1.** The materials used in this project

## 4. Methodology

### 4.1. Creation and Testing of Simulation Model and Monitoring System

To create a model diabetic foot to test and run our design on, we bought a standard fake foot made of silicone. We carved a circular crater (seen in Figure. 2), in which we fitted a plastic dose cup in the crater with modeling clay so our blood simulant (plain water) could be filled in the crater. To represent the electrotherapy treatment, we ran two wires underneath the skin of the foot (also seen in Figure. 2) and attached a blue LED that would turn on when current ran through the wires, representing the current flowing through the wound in electrotherapy. We acknowledge our simulation model is not capable of quantifying or controlling current density and waveform. In future, we plan to incorporate some of these capabilities in our simulation model.



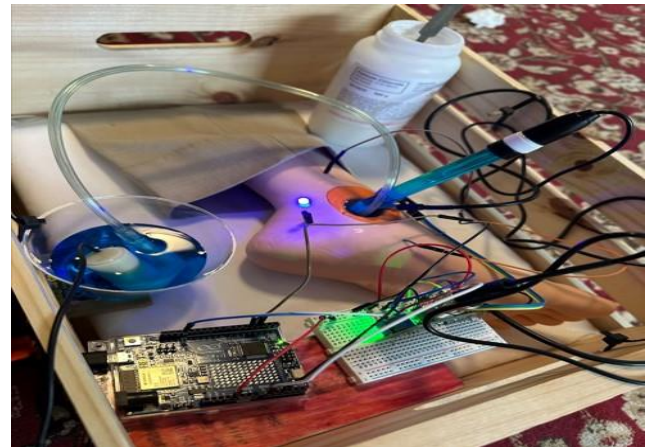
**Figure 2.** Creation of simulation model. Carving of wound crater (left), and under skin wiring (right)

For our monitoring system, we used a DHT11 humidity sensor placed directly above the wound cup to measure the humidity of wound (caused by pus formation), a DS18B20 temperature probe placed in the wound cup to measure the wound's temperature, and a PH0-14 E-201-C pH probe placed in the wound cup to measure the wound's pH, all wired and connected to the Arduino Uno (as shown in Figure. 3 and Figure. 4). The Arduino runs a cloud sketch from the Arduino IDE with the Arduino IoT Cloud Platform, which allows the chip to read values from all sensors and send them, through Wi-Fi, to the Arduino IoT Remote App. The cloud sketch (main program code written by us) is provided as the appendix, and it is available on Arduino website: <https://app.arduino.cc/things/495ce5a9-385a-4c64-9eb8-5c8e0aea7259/sketch> (requires user login).

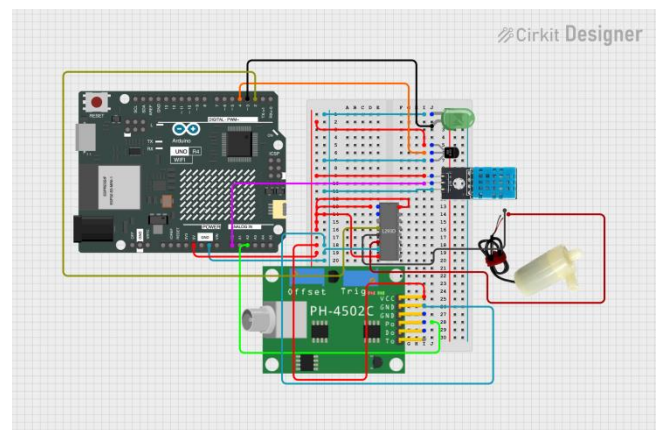
To simulate an infection on our model, we used anhydrous calcium chloride that, when mixed with the plain water in our wound cup, produces a thermogenic reaction that increases the temperature, and humidity (due to water), and changes the pH, mimicking an infection in a worsening wound becoming chronic. These changes are reflected in the app, allowing the user to see the reading of the sensors, which allowed us to evaluate how well our monitoring system was working and what changes were needed.

For the treatment system, the electrotherapy wires and liquid medication (represented by any liquid, we used dyed

water) pump are both wired to the Arduino, again shown in Figure. 3, that allows the user to turn each on or off through the app manually. The app also monitors the time that each treatment is on, indicating for what duration each is active, letting us know if each treatment method was connected correctly, through the Arduino, to the app, and told us what part of the treatment system would need rewiring or programming.



**Figure 3.** Completed assembly of model with Arduino and its temperature, humidity, and pH sensors along with pump and electrotherapy



**Figure 4.** The breadboard for wiring of sensors and controls to Arduino Uno R4 Wi-Fi

### 4.2. Calibration

We tested our system by checking how our device showed changes in temperature, humidity, and pH values to see if it could properly detect changes that indicated a bacterial infection in a worsening wound.

We did this test by filling the crater with water (a substitute for blood since around 50% of blood is water [7]) and putting all the sensor probes into the crater. The system would check and record the base values on the monitor to show stability.

After adding two teaspoons of calcium chloride to the water due to a reaction with water and calcium chloride producing calcium hydroxide, a strong base, and significant heat, which simulates the bacterial infection, increasing the pH and the temperature and humidity of the wound, we tested the control for electrotherapy to ensure it is working.

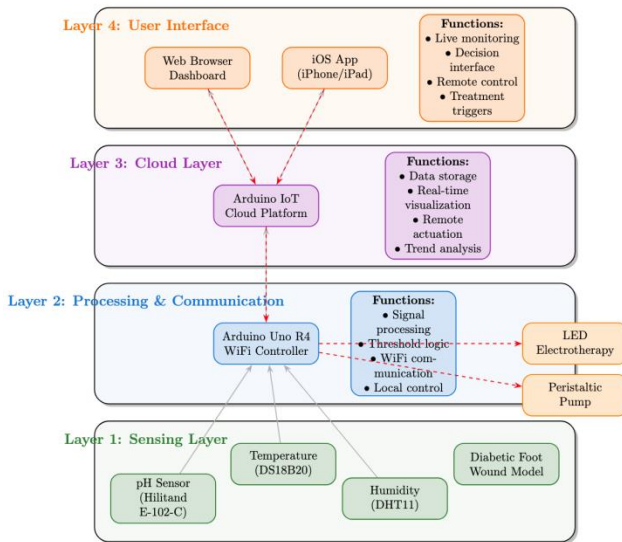
As the system finishes recording the data, a check is done to ensure that the value has changed accurately and reflects the change in the pH, temperature, and humidity. If this test had failed, we would need to reiterate the design.

With this, we also learned that our pH E-201-C probe would require a wait time before stabilizing to a correct value, not due to its error range.

The error range of our sensors were as follows:

pH sensor E-201-C error:  $\pm 0.2$ pH after proper calibration [11], temperature sensor DS18B20 error: the sensors have a quoted accuracy of  $\pm 0.5$  Celsius in the range -10 Celsius to +85 Celsius [12], and the humidity sensor DHT11 error:  $\pm 5\%$ RH in the range 20%~90% [13].

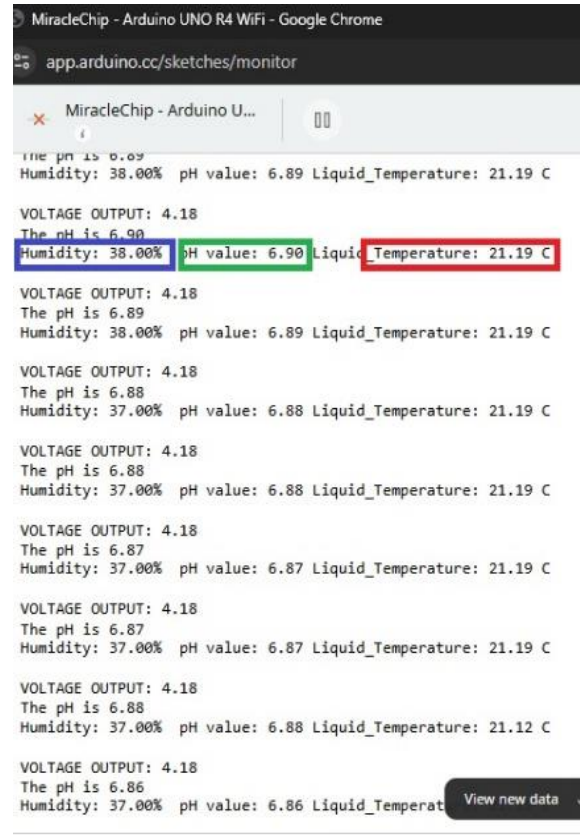
### 4.3. Flowchart Diagram



**Figure 5.** The flowchart diagram of our system, including both physical hardware and online software

## 5. Final Result

To use the diabetic foot monitoring system, the pH, temperature, and humidity probes are placed into the simulated wound and are then connected to the Wi-Fi. As the system records the values, they come as analogue numbers on the Cloud server (Figure. 6 and Figure. 7) that are viewed remotely as graphs in the iPhone app, in Figure. 8 and Figure. 9. The humidity was recorded as 38%, the pH of plain water was recorded as 6.9, and the temperature was recorded as 21.19 °C. Then we add about two teaspoons of calcium chloride to the water to simulate bacterial infection. The calcium chloride with water forms calcium hydroxide, a strong base, and also produces heat, which simulates the bacterial infection, increasing the pH, temperature, and humidity of the wound. After the addition of calcium chloride, the pH increases to 7.65 and the temperature increases to 32.63 °C. The humidity increases to 79%. The changes are noted remotely on the iPhone app, signifying worsening of the wound.



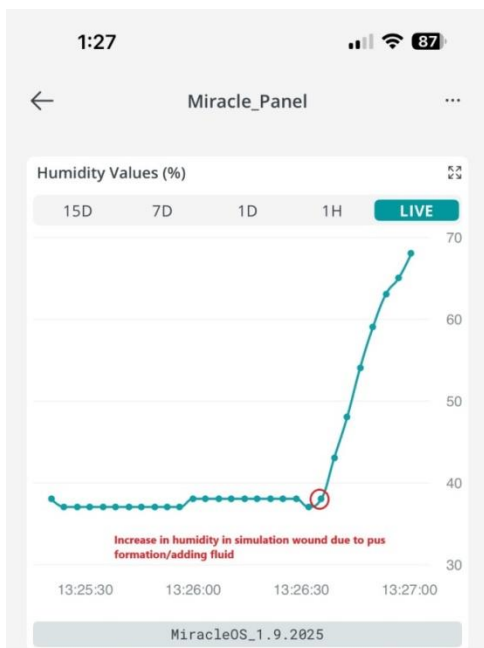
**Figure 6.** Values of the sensors from the terminal before the addition of calcium chloride



**Figure 7.** Values of the sensors from the terminal after the addition of calcium chloride



**Figure 8.** The graph of the values of the temperature (top) and pH (bottom), before and after calcium chloride was added

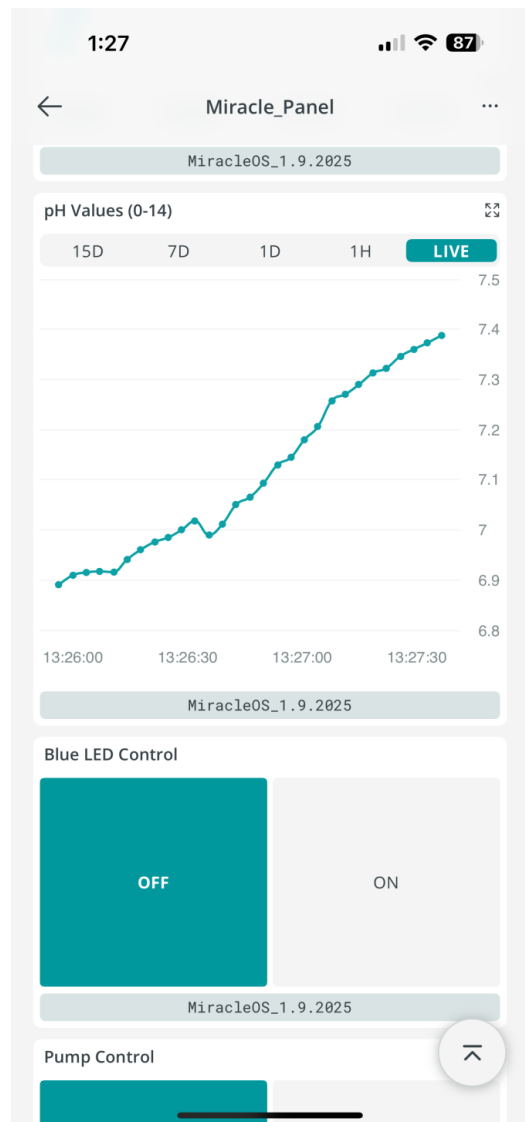


**Figure 9.** The graph of the humidity values, before and after calcium chloride was added



**Figure 10.** Picture of complete simulation model, with monitoring system app on iPhone (bottom-right)

The physician can then promptly initiate early treatment through the controlled electrotherapy and liquid medication pump, for which we have an on-off switch on the iPhone app to control each (please see Figure. 11).



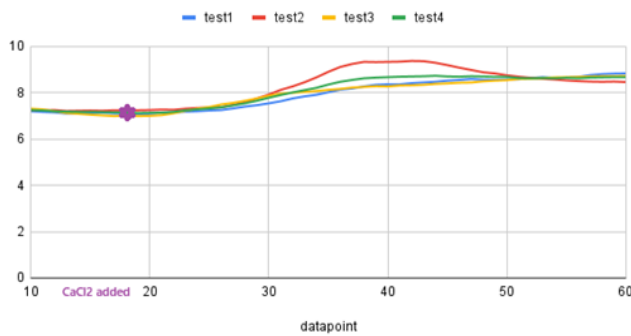
**Figure 11.** Screenshot of iPhone app with on-off switch for electrotherapy (Blue LED Control), and liquid medicine delivery pump

**Table 3.** T-Test Results

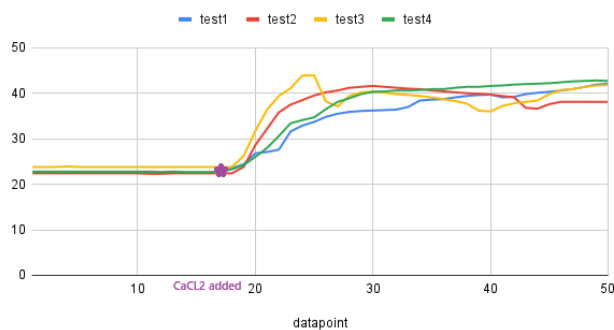
Measurement	Avg. Before Adding Calcium Chloride	Avg. After Adding Calcium Chloride	Avg. Change	T-statistic	P-value
pH in range 0-14, decimal.	7.14	8.21	+1.07	-20.62	5.27e-49
Temperature in Celsius.	22.90	38.09	+15.19	-39.16	2.03e-75
Humidity in percent.	29.74	81.84	+52.10	-27.45	2.37e-55

### 5.1. Statistical Analysis

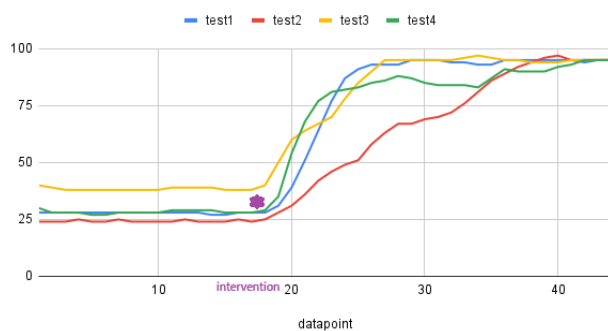
pH for test1, test2, test3 and test4

**Figure 12.** Graph of pH across the tests

Temperature across test1, test2, test3 and test4

**Figure 13.** Graph of temperature in Celsius across the tests

Humidity for test1, test2, test3 and test4

**Figure 14.** Graph of humidity in percentage across the tests

To demonstrate reproducibility, enhance validity, and ensure statistical significance, we conducted the test four additional times. The Arduino serial data was compiled into

a .csv file and imported into a Python program (supplied as appendix), which then performed an independent t-test (Welch's t-test) for each parameter (pH, temperature, and humidity sensor readings). The results for all three sensors in our design are presented in Table 3 and each sensor's reading are visualized in Figures. 12, 13, and 14.

All parameters are statistically significant (all P-values less than 0.05).

## 6. Conclusions

This study introduces a novel IoT-based simulation platform for the remote monitoring and early intervention of diabetic foot wounds. By incorporating integrated sensing of pH, temperature, and humidity, alongside remotely actuated treatment modules such as electrotherapy and liquid medication delivery, the system is designed to detect key pathological indicators associated with wound infection progression. Experimental testing using a foot model with simulated inflammatory responses demonstrated the system's ability to accurately record diabetic wound parameters in real time.

The integration of sensor data with cloud-based monitoring presents a practical approach for enhancing physician oversight, particularly for patients with limited access to in-person care. The model's modular design supports flexibility in future development, and its emphasis on real-time feedback aligns with current trends in personalized and connected healthcare. These results suggest the feasibility of using smart wound management systems to improve outcomes in diabetic care and reduce the incidence of preventable lower-limb amputations.

Future work will focus on proper integration of a more controlled and quantifiable electrotherapy circuit, the miniaturization and enhanced precision of sensors, the addition of imaging capability and artificial intelligence for automated wound assessment, and possible clinical validation. The proposed proof-of-concept model has the potential to contribute meaningfully to the advancement of biomedical engineering solutions for chronic wound management.

## ACKNOWLEDGEMENTS

This project received the Gold Medal in the Bayonne District's Project Innovate competition, and Silver Medal at

the 67<sup>th</sup> Hudson County STEM Showcase in the year 2025. This project has also been nominated to participate in the Thermo Fisher Scientific Junior Innovators Challenge 2025, to be held in Washington D.C., USA.

## The Appendix

### *MiracleOS\_1-7-2025.ino* (Main Code)

```

/*
  Sketch generated by the Arduino IoT Cloud Thing
  "Untitled"

  https://create.arduino.cc/cloud/things/534aba90-846c-4e48-
  8194-bfe7ad9541b3

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and
  updated when changes are made to the Thing

  float humidity;
  float liquidTemp;
  float pH;
  int led_blue;
  int pump;

  Variables which are marked as READ/WRITE in the
  Cloud Thing will also have functions
  which are called when their values are changed from the
  Dashboard.
  These functions are generated with the Thing and added
  at the end of this sketch.
  */
#include <DS18B20.h>
#include "thingProperties.h"
#include <dht.h>
#include <Wire.h>
#include <OneWire.h>

#define SENSOR A2
#define OFFSET 0.00
#define LED 13
#define SAMPLING_INTERVAL 20
#define PRINT_INTERVAL 800
#define ARRAY_LENGTH 40
int PH_ARRAY[ARRAY_LENGTH];
int PH_ARRAY_INDEX=0;

/* DHT uses analog pin A0
  pH Probe uses analog pin A2
  LED uses digital pin 3
  Pump uses digital pin 2
  DS18B20 uses pin digital pin 4

```

```

*/
#define dht_apin A0
dht DHT;

int temp;

DS18B20 ds(4); // Liquid Temp connects to pin 4.

double AVERAGE_ARRAY(int* ARR, int NUMBER){
  int i;
  int max,min;
  double AVG;
  long AMOUNT=0;
  if(NUMBER<=0){
    Serial.println("ERROR!/n");
    return 0;
  }
  if(NUMBER<5){ //less than 5, calculated directly
    statistics
    for(i=0;i<NUMBER;i++){
      AMOUNT+=ARR[i];
    }
    AVG = AMOUNT/NUMBER;
    return AVG;
  }else{
    if(ARR[0]<ARR[1]){
      min = ARR[0];max=ARR[1];
    }
    else{
      min=ARR[1];max=ARR[0];
    }
    for(i=2;i<NUMBER;i++){
      if(ARR[i]<min){
        AMOUNT+=min; //arr<min
        min=ARR[i];
      }else {
        if(ARR[i]>max){
          AMOUNT+=max; //arr>max
          max= AMOUNT+=ARR[i];
        }else{
          AMOUNT+=ARR[i]; //min<=arr<=max
        }
      }
    }
    }//for
    AVG = (double)AMOUNT/(NUMBER-2);
  }//if
  return AVG; }

void setup() {
  pinMode(LED,OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3,OUTPUT);
  led_blue = 0;
  pump = 0;

  Serial.begin(9600);

```

```

// Initialize serial and wait for port to open:
Serial.begin(9600);

// This delay gives the chance to wait for a Serial
Monitor without blocking if none is found
delay(1500);
Serial.println("DHT11 Humidity & temperature
Sensor\n\n");
delay(1000); //Wait before accessing Sensor

// Defined in thingProperties.h
initProperties();

// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/*
The following function allows you to obtain more
information
related to the state of network and IoT Cloud
connection and errors
the higher number the more granular information
you'll get.
The default is 0 (only errors).
Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  DHT.read11(dht_apin);

  static unsigned long SAMPLING_TIME = millis();
  static unsigned long PRINT_TIME = millis();
  static float VOLTAGE;
  if(millis()-SAMPLING_TIME >
  SAMPLING_INTERVAL)
  {
    PH_ARRAY[PH_ARRAY_INDEX++]=analogRead(SENSOR);

    if(PH_ARRAY_INDEX==ARRAY_LENGTH)PH_ARRAY_INDEX=0;
    VOLTAGE = AVERAGE_ARRAY(PH_ARRAY,
    ARRAY_LENGTH)*5.0/1024;
    SAMPLING_TIME=millis();
  }
  if(millis() - PRINT_TIME > PRINT_INTERVAL)
  //Every 800 milliseconds, print a numerical, convert the state
of the LED indicator
  {
    Serial.print("VOLTAGE OUTPUT: ");
    pH = 5.4348*(4.71 - VOLTAGE) + 4.00;
    Serial.println(VOLTAGE,2);
    Serial.print("The pH is ");
    Serial.println(pH);
    digitalWrite(LED,digitalRead(LED)^1);
    PRINT_TIME=millis();
  }

  Serial.print("Humidity: ");
  humidity = DHT.humidity;
  Serial.print(DHT.humidity);
  Serial.print("% ");
  Serial.print("pH value: ");

  Serial.print(pH);
  Serial.print(" ");

  ds.doConversion();

  while (ds.selectNext()) {
    Serial.print("Liquid_Temperature: ");
    liquidTemp = ds.getTempC();
    Serial.print(liquidTemp);
    Serial.println(" C\n");
  }
  if (led_blue == 1) {
    digitalWrite(3, HIGH); // Blue LED
  }
  else { digitalWrite(3, LOW);
  }

  if (pump == 1) {
    digitalWrite(2, HIGH); // Pump
  }
  else { digitalWrite(2, LOW);
  }

  delay(2000); //Wait 2 seconds before accessing the
sensors again.
}

/*
Since Pump is READ_WRITE variable,
onPumpChange() is
executed every time a new value is received from IoT
Cloud.
*/
void onPumpChange() {
  if (pump == 1) {
    digitalWrite(2, HIGH); // pump
  }
  else { digitalWrite(2, LOW);
}

```

```

    }
}

/*
  Since LedBlue is READ_WRITE variable,
  onLedBlueChange() is
  executed every time a new value is received from IoT
  Cloud.
*/
void onLedBlueChange() {
  if (led_blue == 1) {
    digitalWrite(3, HIGH); // Blue LED
  }
  else { digitalWrite(3, LOW);
  }
}

```

### **thingsProperties.h (Library for WiFi)**

```

// Code generated by Arduino IoT Cloud, DO NOT EDIT.

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char SSID[] = SECRET_SSID; // Network
SSID (name)
const char PASS[] = SECRET_OPTIONAL_PASS;
// Network password (use for WPA, or use as key for WEP)

void onLedBlueChange();
void onPumpChange();

float humidity;
float liquidTemp;
float pH;
int led_blue;
int pump;

void initProperties(){

  ArduinoCloud.addProperty(humidity, READ, 2 *
SECONDDS, NULL);
  ArduinoCloud.addProperty(liquidTemp, READ, 2 *
SECONDDS, NULL);
  ArduinoCloud.addProperty(pH, READ, 2 * SECONDS,
NULL);
  ArduinoCloud.addProperty(led_blue, READWRITE,
ON_CHANGE, onLedBlueChange);
  ArduinoCloud.addProperty(pump, READWRITE,
ON_CHANGE, onPumpChange);

}

```

```

WiFiConnectionHandler
ArduinoIoTPreferredConnection(SSID, PASS);

```

### **Python Program used for generating T-Test and Statistical Analysis:**

```

import pandas as pd
from scipy import stats

def analyze_characteristic(file_path, characteristic_name,
before_start, before_end, after_start):
    """
    Reads a CSV file, performs a t-test, and calculates the
    average change.
    """
    # Read the data from the CSV file
    df = pd.read_csv(file_path)

    # Define the columns containing the test
    measurements
    test_columns = ['test1', 'test2', 'test3', 'test4']

    # --- Grouping and Data Preparation ---
    # Create the 'before' group based on the specified
    datapoint range
    before_df = df[(df['datapoint'] >= before_start) &
(df['datapoint'] <= before_end)]

    # Create the 'after' group
    after_df = df[(df['datapoint'] >= after_start)

    # Flatten the data for each group into a single array of
    measurements
    # This pools all 'test' columns for the t-test
    before_values = before_df[test_columns].values.flatten()
    after_values = after_df[test_columns].values.flatten()

    # --- Calculations ---
    # Calculate average values for before and after
    avg_before = before_values.mean()
    avg_after = after_values.mean()

    # Calculate the average change
    avg_change = avg_after - avg_before

    # Perform the independent t-test (Welch's t-test)
    t_stat, p_value = stats.ttest_ind(before_values,
after_values, equal_var=False)

    # --- Print Results ---
    print(f"--- Analysis for {characteristic_name} ---")
    print(f"Before Group: Datapoints
{before_start}-{before_end}")
    print(f"After Group: Datapoints {after_start}-end")
    print(f"\nAverage {characteristic_name} Before:
{avg_before:.2f}")

```

```

    print(f"Average {characteristic_name} After:
{avg_after:.2f}")
    print(f"Average Change: {avg_change:+.2f}")
    print(f"\nT-statistic: {t_stat:.2f}")
    print(f"P-value: {p_value:.2e}")

# Interpret the p-value
if p_value < 0.05:
    print("Conclusion: The change is statistically
significant (p < 0.05).\n")
else:
    print("Conclusion: The change is not statistically
significant (p >= 0.05).\n")

# --- Main execution ---
# Analyze each file with the specified grouping rules
analyze_characteristic('ph_new.csv', 'pH', before_start=14,
before_end=17, after_start=19)
analyze_characteristic('temp.csv', 'Temperature (°C)',
before_start=1, before_end=18, after_start=19)
analyze_characteristic('humidity.csv', 'Humidity (%)',
before_start=1, before_end=18, after_start=19)

```

---

## REFERENCES

- [1] J. L. Harding *et al.*, "National and State-Level Trends in Nontraumatic Lower-Extremity Amputation among U.S. Medicare beneficiaries with Diabetes, 2000–2017," *Diabetes Care*, vol. 43, no. 10, pp. 2453–2459, Jul. 2020, doi: 10.2337/dc20-0586.
- [2] National Library of Medicine, "Diabetic foot," *MedlinePlus*. Available: <https://www.medlineplus.gov/diabeticfoot.html>.
- [3] "Preventing Diabetes-Related amputations," *Diabetes*, May 15, 2024. <https://www.cdc.gov/diabetes/diabetes-complications/preventing-diabetes-related-amputations.html>.
- [4] Hoilett, Orlando. "A Smart Bandage for Monitoring Chronic Wounds." *Hackaday*, 14 Apr. 2022, [www.hackaday.com/2020/05/12/a-smart-bandage-for-monitoring-chronic-wounds](http://www.hackaday.com/2020/05/12/a-smart-bandage-for-monitoring-chronic-wounds).
- [5] Lan, Xiaodong, *et al.* "Electrical Stimulation as an Adjunctive Therapy for Diabetic Ulcers: A Systematic Review and Meta-analysis." *International Wound Journal*, vol. 21, no. 12, Dec. 2024, <https://doi.org/10.1111/iwj.70104>.
- [6] R. Saracco, "Smart bandage," *IEEE Future Directions*, Dec. 22, 2021. <http://cmte.ieee.org/futuredirections/2021/12/24/smart-bandage>.
- [7] Institute for Quality and Efficiency in Health Care (IQWiG), "In brief: What does blood do?," *InformedHealth.org - NCBI Bookshelf*, Mar. 16, 2023. <https://www.ncbi.nlm.nih.gov/books/NBK279392>.
- [8] P. Salvo *et al.*, "Temperature- and pH-sensitive wearable materials for monitoring foot ulcers," *International Journal of Nanomedicine*, vol. Volume 12, pp. 949–954, Jan. 2017, doi: 10.2147/ijn.s121726.
- [9] P. Jindatanmanusan, S. Luanraksa, T. Boonsiri, T. Nimmanon, and P. Arnutti, "Wound fluid matrix metalloproteinase-9 as a potential predictive marker for the poor healing outcome in diabetic foot ulcers," *Pathology Research International*, vol. 2018, pp. 1–5, Oct. 2018, doi: 10.1155/2018/1631325.
- [10] L. Gould and M. Mahmoudi, "Analysis of biogenic amines and small molecule metabolites in human diabetic wound ulcer exudate," *ACS Pharmacology & Translational Science*, vol. 7, no. 9, pp. 2894–2899, Sep. 2024, doi: 10.1021/acspsci.4c00418.
- [11] "Grove - PH Sensor Kit (E-201C-Blue) | Seeed Studio Wiki," Jan. 06, 2023. <https://wiki.seeedstudio.com/Grove-PH-Sensor-kit/>.
- [12] "Temperature Sensing Using DS18B20 Digital Sensors — Open Energy Monitor 0.0.1 documentation." <https://docs.openenergymonitor.org/electricity-monitoring/temperature/DS18B20-temperature-sensing.html>.
- [13] dfrobot.com, "Upgrade Your Arduino with DHT11 Sensor: Accurate Readings!," *dfrobot.com*. <https://www.dfrobot.com/product-174.html>.