LAUTECH Journal of Computing and Informatics (LAUJCI) – ISBN : 2714-4194
Volume 5 No.1, April 2026 – www.laujci.lautech.edu.ng

LABVIEW-BASED HUMAN-MACHINE INTERFACE FOR AN AUTOMATED INCUBATOR

Ogidan, Olugbenga Kayode
Department of Electrical and Electronics Engineering
Elizade University,
Ilara-Mokin, Ondo State.
olugbenga.ogidan@elizadeuniversity.edu.ng

Shobowale, Yusuf Ishola
Department of Computer Engineering
Obafemi Awolowo University,
Ile-Ife, Nigeria
yishobowale@oauife.edu.ng

Hamza-Sambo, Surayyah
Department of Electrical and Electronics Engineering
Elizade University,
Ilara-Mokin, Ondo State.
hamzasurayyah1999@gmail.com

ABSTRACT
Microorganisms are temperature sensitive and as such they are usually kept in temperature-controlled chambers. Many existing incubators use Liquid Crystal Display (LCD) as temperature monitors which does not give users full monitoring and control of the incubator. This research develops a Human-Machine Interface (HMI) for an automated microbial incubator to enhance user's experience by allowing them full temperature monitoring and control without having to open the incubator from time to time. The HMI is developed with the National Instruments Laboratory Virtual Instrument Engineering Workbench (LabVIEW). Temperature acquisition and control were achieved using the Atmega 328 microcontroller on an Arduino board which was interfaced with LabVIEW. Tests performed reveal that the incubator was able to maintain a temperature between 23^{o} C and 30^{o} C while providing information for users through LCD, Arduino serial, and the developed LabVIEW-based user interface. The LabVIEW interface was able to provide a graphical display of incubator temperature rise and fall on a real-time basis thus enhancing users' experience more than LCD or serial monitor. This work will add value to microbial study enabling researchers' efficiency and productivity.

Keywords: Arduino, Automation, Human Machine Interface, Incubator, LabVIEW.

1. INTRODUCTION

Incubators serve a variety of functions in a scientific laboratory. They generally maintain a constant temperature, however, additional features are often built in. The incubator in this research is a microbial incubator used for the growth of a variety of organisms, including bacteria, cells, and embryos. Incubator types include neonatal incubator (Al-Sawaff, *et al*., 2020; Fahmi, *et al.*, 2020; Nwaneri, *et al*., 2020; Subramanian, *et al*., 2019; Marafa, *et al*., 2020; Prasad, *et al.*, 2015; Nisha, *et al*., 2014), egg incubtors (Kommey, *et al*., 2022; Peprah, *et al*., 2022), microbial incubators Sánchez, *et al*., (2020) to mention a few.

The problem addressed in this project is the need for a user-friendly and efficient interface for controlling and monitoring an automated incubation system. Sánchez, et al., 2020 authors developed an open-source, low-cost temperature-control chamber for Microbial Electrochemical Technologies (MET) research. The research aimed at providing a low-cost solution for controlling temperature in the range 20˚C to 55˚C. One of the limitations of the research was that the Graphical User Interface (GUI) written in Python was not user-friendly. The proposed solution is the development of a LabVIEW-based Human Machine Interface that allows for easy and intuitive operation of the incubation system, as well as real-time monitoring of the temperature. The incubation system's overall performance and user experience need to be enhanced.

2. RELATED WORKS

Interfaces are tools or systems that allow unrelated entities to communicate with one another. The interactions between people and computing systems are made easier by user interfaces. These user interfaces include a command-line interface in which texts are used as the basis of users' communication with the computer. Graphical User Interface (GUI) is another type of user interface. GUI is an alternative to text-only interfaces making use of programming language knowledge and it makes it possible for non-technical users to operate sophisticated devices. In the case of a Menu-driven interface, it uses a series of screens (menus) either a graphic or list format, through which the user can interact with devices. Through this, the user is able to complete a goal or advance to a secondary menu with additional options. Another form of user interface is the conversational user interface. In this case, users can control gadgets by speaking a series of commands with embedded voice recognition capabilities.

Human-Machine Interface (HMI) is a dashboard that enables a user to communicate with a machine, computer program, or system. HMIs provide users access to real-time data and a graphical user interface for monitoring and controlling equipment, especially in an industrial setting such as health, environmental monitoring Isa et al., (2018), energy, agriculture Sony (2020), oil and gas, power, water and wastewater Ogidan *et al*, (2014), transportation and the likes.

Examples of HMIs include computer monitors, built-in screens on machines, and tablets to mention a few. Their basic purpose is to provide insight into the mechanical and electrical performance of the system and measure progress. They can be used to communicate with Programmable Logic Controllers (PLCs) and input/output sensors to get and display information for users to view. They are also used for as monitoring and tracking, turning machines on and off or accelerating output. HMIs can be used to connect to the Supervisory Control and Data Acquisition (SCADA) systems and to access important, trigger alarms, and view crucial information presented in graphs, charts, or digital dashboards.

Presently, HMIs can be developed using web technologies to be used on PC or mobile devices (Al-Sawaff, *et al.*, 2020; Kommey, *et al.*, 2022; Fahmi, *et al.*, 2020) ladder programming as in the case of PLC, Python Sánchez et al., (2020), Microsoft Visual Studio Marafa et al., (2020), Laboratory Virtual Instrument Engineering Workbench (LabVIEW) (Al-Sawaff, *et al.*, 2020; Patel et al., 2019; Subramanian, *et al.*, 2019; Isa, *et al.*, 2018) to mention a few. LabVIEW is a powerful and versatile software development platform that is widely used in industrial and scientific applications. LabVIEW being a graphical programming environment was equipped with comprehensive toolkits for data acquisition, signal processing, and control, specifically useful for rapid development of automated experimental and testing systems. It is particularly required for multi-channel real-time data acquisition, automated experiment control, and synchronized communication with multiple instruments. By using LabVIEW, the HMI can be customized and tailored to the specific needs of the incubation system, and can easily interface with other instruments and equipment (LabVIEW, 2023a, LabVIEW, 2023b). LabVIEW based on the Queued Message Handler (QMH) architecture was used to analyse an automated multi-threaded control and data analysis framework (Luo *et al.*, 2026).

Some incubators are implemented with LCD Arduino serial monitors (Kommey, *et al*., 2022; Peprah, *et al*., 2022; Fahmi, (2020); Kumbhar, et al., 2022, Marafa et al., 2020; Prasad, et al., 2015; Nisha, et al., 2014) as the user's interface. This cannot effectively give users full control.

According to Athimoolam and Karuppasamy (2022), studies indicate that IoT-based smart incubators in comparison with conventional incubators possess the ability of reducing energy consumption, reducing noise and improving temperature stability. Smart incubators on the other hand have great potentials of providing personalized care and improving clinical outcomes, for preterm infants (Linu *et al*., 2024

Some of the neonatal incubator-focused studies reviewed in this paper used LabVIEW for the HMI. The authors in Al-Sawaff, *et al*., (2020) developed a LabVIEW-based temperature control system for Neonatal Incubators which is used to monitor and control changes in the air temperature of the incubator and the baby's skin temperature. A similar study was done by (Isa, *et al*., 2018) which focused on monitoring the temperature of a certain sensitive environment. This system is able to alert the user when the temperature is out of the specified range by giving a warning through a Light-Emitting Diode (LED) display on the LabVIEW software interface and provides an audible alarm by a buzzer. In (Prakash *et al.*, 2017), a more sophisticated method of monitoring human body parameters like temperature, and heartbeat rate and estimating the percentage of oxygen using an embedded web server and LabVIEW technology was developed. LM35 sensor was used as a temperature sensor to measure the temperature of the patient, pulse oximeter is the method used to estimate the percentage of oxygen in the body of the patient and TCRT1000 is the Integrated circuit used here to measure the heart pulse of the patient. These data are serially transmitted over the serial port of LabVIEW. In LabVIEW screen, current biomedical parameters are displayed for users to see.

Raharja, and Sugiyarta (2022) presented a machine learning-based smart incubators to analyze real-time data and predict the risk of neonatal complications by applying use artificial intelligence algorithms in providing early intervention and personalized care. While this improved the overall health outcomes of preterm infants, it lacks enough clinical validation and standardization of the systems.

Yan *et al.*, (2024) presented the establishment of the microscope incubation system (MIS). The work provided a system that was easily adapted to any inverted microscope stage by introducing an incremental PID control algorithm to provide temperature and gas concentration stability. The system could trace single cell migration in real time and support cell viability when compared to standard incubators.

Other LabVIEW-based temperature control projects did not only apply to incubators or health monitoring but also other sectors like agriculture. An example is a study done by (Soni, 2020) on Greenhouse Monitor and Control with LabVIEW. The project develops a data acquisition and data logging system, which monitors several important parameters of the greenhouse such as temperature, humidity, light intensity, and soil moisture while these parameters are made available to the users on a user-friendly interface developed with LabVIEW. In (Patel, *et al*., 2019), the authors developed a LabVIEW-based smart house control as a home-protecting system. This was used for the detection of hazardous parameters using sensors like smoke detection, Liquified Petroleum Gas (LPG) leakage detection, motion detection, and temperature monitoring. The beauty of the work of (Patel, et al., 2019) is that temperature wasn't the only focus. Temperature sensor (LM35) was the sensor used to measure the temperature. LPG sensor (MQ-6) is used to sense LPG. To interface the sensors with LabVIEW software running on a PC, National Instrument (NI) Data Acquisition (DAQ) National Instrument 6008 DAQ was connected to the laptop thereby making all the sensor readings to be obtained and controlled. The only limitation of this study is that LabVIEW can only be viewed with a computer (laptop) which is not portable. In (Ogidan, *et al*., 2014), the authors developed a Smith predictor-based controller to mitigate the effect of network delays on the dynamics of a non-linear wastewater distributed system. The control algorithm was developed using MATLAB while the implementation was done using LabVIEW for a graphical User Interface. In (Bolu, et al., 2022), instrumentation control over the network was developed and LabVIEW was used as the use interface. In the work of (Muñoz-Berbel *et al*., 2011), the authors developed an instrumentation system for a real-time measurement of Escherichia coli cultures concertation. They made use of platinum electrode chips and impedance spectroscopy. The sequential injection analysis (SIA) carried out during this research was implemented as a virtual instrumentation on a PC running LabVIEW. Modular task scheduling, asynchronous event handling, and parallel processing was enabled via queue-driven message dispatching to present LabVIEW's Queued Message Handler (QMH) architecture (Volponi, *et al*., 2024).

Wearable smart incubators in comparison with conventional incubators are systems worn by the infants which improve parent-infant bonding; and reduction in the risk of infection. It provides continuous monitoring and control of environmental parameters but poses the challenges of safety of wearable devices (Devaprasanth *et al*., 2022).

Indhuja *et al*., (2023) presented a more efficient and integrated system by combining IoT-based hybrid smart incubators with conventional incubators which could automatically adjust environmental parameters in real-time thereby providing integrated feeding systems and phototherapy lights features. This still has limitation in that limited evidence on the clinical outcomes and cost-effectiveness were provided.

Existing incubators have had significant changes over the years. From Table 1, one would discover that the most common are the neonatal (for infants) incubators (Al-Sawaff, et al., 2020; Fahmi, et al., 2020; Nwaneri, et al., 2020; Subramanian, et al., 2019; Marafa, et al., 2020; Prasad, et al., 2015; Nisha, et al., 2014) followed by the egg incubators (Kommey, et al., 2022; Peprah, et al., 2022). Whereas there is less attention on the microbial incubator (Sánchez, et al., 2020; Muñoz-Berbel et al., 2011). The dearth of research efforts in microbial incubators and the fact that off-the-shelf microbial incubators are quite expensive Sánchez, et al., (2020) is one of the motivations for this work.

Most of the existing automated neo-natal and egg incubation systems often have a limited user interface made of LCD (Kommey, *et al*., 2020; Fahmi, *et al*., 2020; Peprah, *et al*., 2022; Kumbhar, *et al*., 2022), which are small in size and contain limited information. These size-limiting interfaces makes them difficult to use, leading to errors and suboptimal performance. This work is unique as a smart combination of cheap hardware with professional software that provides a real-time non-intrusive GUI by showing graphs of temperature rise/fall without opening the door; making it suitable for sensitive experiments. It stands out as a game-changer in microbial incubation studies for resource-limited laboratories like in Nigeria.

The proposed solution is to enhance the microbial incubator by the development of a PC-based Human Machine Interface running on LabVIEW that allows for easy and intuitive operation of the incubation system, as well as real-time monitoring of important parameters such as temperature, humidity, and CO_2 levels. From Table 1, it could be observed that previous research work has shown the potential of LabVIEW for virtual instrumentation of incubators but most of them are applied in the neonatal incubators (Al-Sawaff *et al.*, 2020; Subramanian, et al., 2019; Prasad, *et al.*, 2015). This work is unique because it seeks to apply LabVIEW-based virtual instrumentation to microbial incubators. This is to enhance user's experience and to improve microbial incubator research especially in situations where the experiment to be performed requires a non-intrusive condition. The rest of the paper will be arranged in this manner. Section 3 will discuss methodology, results, and discussion will be the focus of section 4, while section 5 will be the conclusion.

Table 1. Analysis of papers reviewed on incubator neonatal, egg, and microbial incubators.

Paper	**Neonatal Incubator**	**Egg Incubator**	**Microbial Incubator**	**LCD/Serial monitor**	**Python**	**LabVIEW interface**	**Web/Mobile Application**
Al-Sawaff et al., 2020	✓					✓	✓
Sánchez et al., 2020			✓		✓		
Kommey, et al., 2020		✓		✓			✓
Fahmi, et al., 2020	✓			✓			✓
Peprah, et al., 2022		✓		✓			✓
Nwaneri, et al,. 2020	✓			✓			✓

Subramanian, et al., 2019	✓					✓	✓
Kumbhar, et al., 2022	✓			✓			
Marafa, et al., 2020	✓			✓			
Prasad, et al., 2015	✓			✓		✓	✓
Nisha, et al., 2014	✓			✓			

3. DESIGN AND IMPLEMENTATION

The aim of this research is to develop a LabVIEW-based HMI for the monitoring and control of microbial incubators. The procedure involves developing a microcontroller-based temperature control for the automated incubator, developing a HMI for the automated incubator and interfacing Arduino microcontroller-based temperature control system with LabVIEW-based HMI. These will be discussed one after the other.

Figure 1 is an architecture of communication between microbial incubator and the LabVIEW-based HMI. The incubator, through its temperature sensor sends signal to the microcontroller which is displayed on the LabVIEW-based HMI. The communication geos back in reverse order back to the microbial incubator.

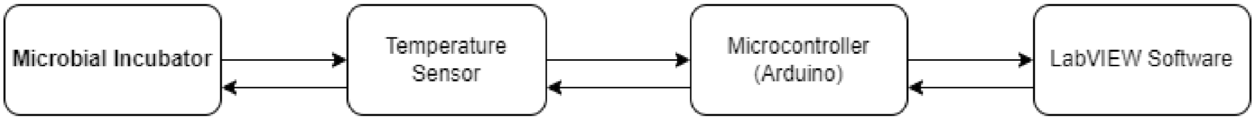


Fig 1: Architecture of communication between microbial incubator and the LabVIEW-based HMI

3.1 The incubator Heat Transfer Model.

The heat energy that is generated within the incubator in form of warming and cooling transient temperatures is described by the heat transfer (Yeler and Koseoglu,2021; Bahrami, 2009) shown in equations 1 and 2:

$$T = (T_i - T_n)e^{-t/RC} \qquad (1)$$

$$T = (T_i - T_n)e^{-t/(\left(\frac{1}{hA}\right))\,(\frac{\mathrm{p}}{C_pV}))} \quad (2)$$

Where:

T= Transient incubator temperature

T_i = initial temperature of the incubator before heating

T_n= Incubator bulk air temperature

t= Time (seconds)

C = Thermal capacitance (p/C_pV)

R = Thermal resistance $(1/hA)$

C_p= Specific Heat

V= Volume

p= Density

h=Heat transfer coefficient

A = Surface area

3.2 Developing Microcontroller-based Temperature Control

The system block diagram is shown in Figure 2a. The core of the incubator's temperature control system is an Arduino Uno microcontroller that coordinates all components. The sensing and actuation subsystem comprises a DS18B20 digital temperature sensor, a two-channel relay module, and a manual mode selector switch. The DS18B20 sensor, a single-wire digital thermometer operates across a 3.0 to 5.5 V supply range and measures temperatures from -55°C to +125°C with 0.0625°C resolution. Connected via the One Wire protocol to the Arduino Uno, it continuously monitors chamber temperature and transmits it to the microcontroller. Upon receiving the temperature, the microcontroller uses its control algorithm to make decisions. Based on the decision, a two-channel relay is made to switch between the upper and lower temperature thresholds based on feedback from temperature sensor DS18B20. The control algorithm maintains chamber temperature within 23°C to 30°C: Heating mode is activated when temperature, $T < 23°C$ while cooling mode is activated when $T > 30°C$. System visual output is made available through LCD, LED and LAbVIEW HMI. Heat is provided for the incubator through heating pads while cooling fans are employed to provide cooling when necessary to maintain temperature within specified limits. The entire system is powered by a 12 V DC power source.

3.3 Closed Loop Incubator Temperature Control System

The feedback control loop adopted by the developed device is shown in Figure 2b where the chosen threshold temperature, desired or reference input temperature is the *R(s)*. $M_{CONTROL}$ is the controller which is Arduino Uno microcontroller in this case. The plant being controlled is the incubator temperature denoted by B_{TEMP} while the measured temperature displayed on the LCD/serial monitor or LabVIEW HMI is the system output *Y(s)*. Feedback is the DS18B20 temperature sensor denoted by $\mathrm{SENSOR_{TEMP}}$ which continuously feeds the Arduino Uno microcontroller with the incubator temperature at any given time. Equations (3) to (7) describe the closed loop transfer function of the system. The open loop transfer function for the developed device is shown in Equations (3) to (6). The gain for the closed-loop control system of Figure 2b is obtained as follows:

$$M_{CONTROL} = \frac{U(s)}{E(s)} \tag{3}$$

$$U(s) = E(s) \times M_{CONTROL} \tag{4}$$

$$B_{TEMP}(s) = \frac{Y(s)}{U(s)} \tag{5}$$

$$Y(s) = B_{TEMP} \times U(s) \tag{6}$$

Substituting Equations (2), (3) and (4) into Equation (5) and rearranging terms gives

$$\frac{Y(s)}{R(s)} = \frac{M_{CONTROL} \times B_{TEMP}}{1 + M_{CONTROL} \times B_{TEMP}} \tag{7}$$

where *R(s)* = reference input, *E(s)* = error, *U(s)* = control input, *Y(s)* = system output, $M_{CONTROL}$ = microcontroller, B_{TEMP} = body temperature, and $SENSOR_{TEMP}$ = temperature sensor.

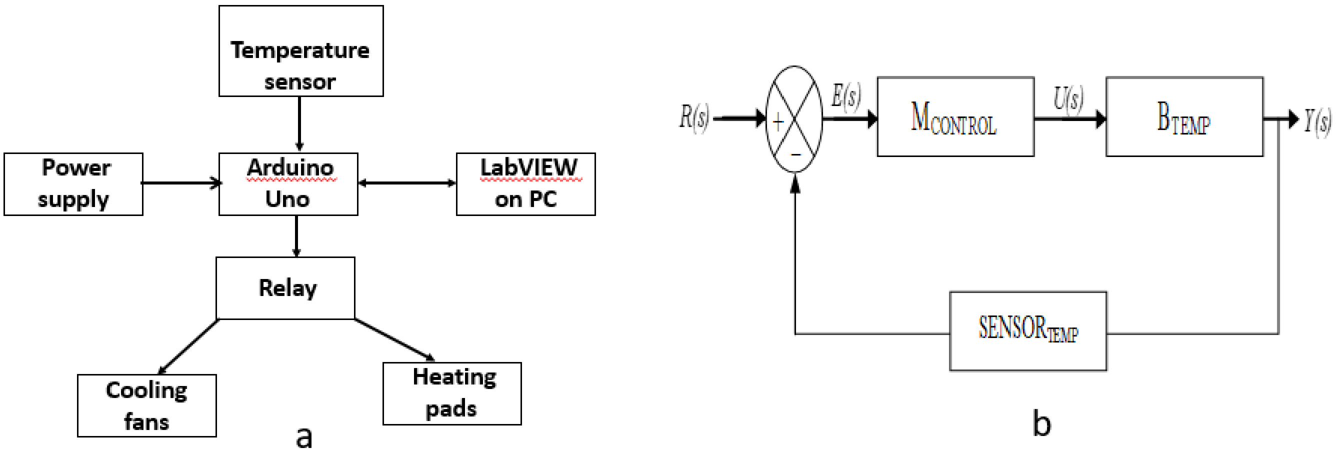


Fig 2: a) Block Diagram of the developed system, b) Closed-loop temperature control system of the automated incubator

3.4 Interfacing the Arduino Microcontroller with LabVIEW

From Figure 1, it can be observed that a temperature sensor is used to acquire the temperature in the incubation chamber and fed to the microcontroller. In the microcontroller, temperature control is done to ensure the incubator is maintained between 23 to 29 ^{o}C. The temperature sensor used in this research is a one-wire temperature sensor (DS18B20) shown in Figure 3a while Figure 3b shows the control unit of the incubation system consisting of the Atmega 328 microcontroller on the Arduino board and other components such as a relay for switching on and off the heating unit and Liquid Crystal Display (LCD) to view the temperature values. In this research, a Universal Serial Bus (USB) cable is used to connect the temperature control unit of the incubator to a PC hosting the LabVIEW software.

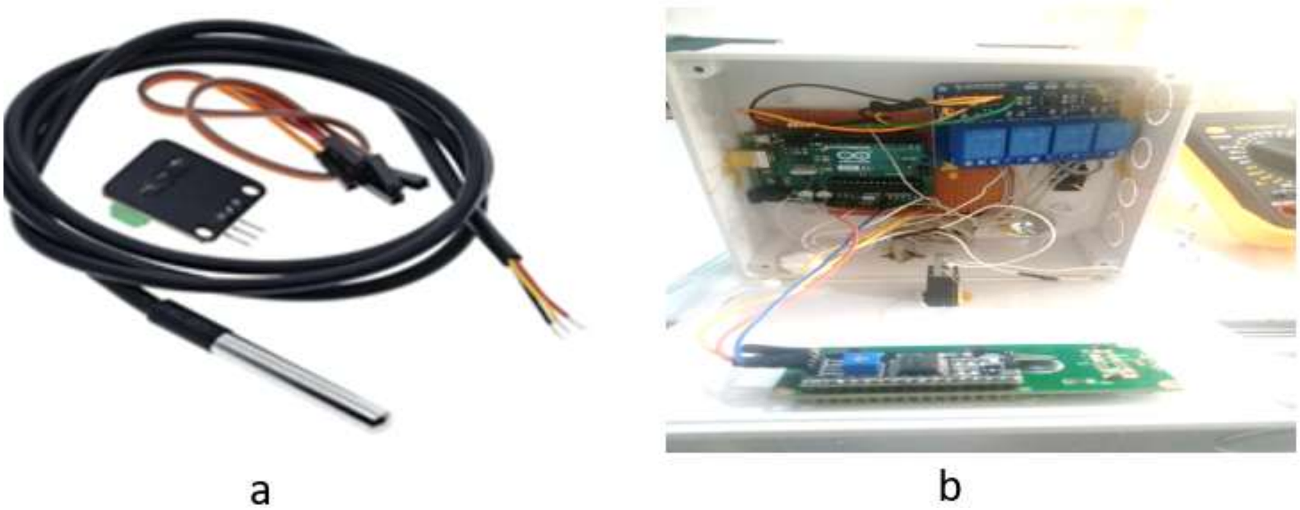


Fig 3: a) One-wire Temperature Sensor, b) Hardware Setup

Programing is done in the Arduino Integrated Environment (IDE) and uploaded to the Atmega 328 microcontroller on the Arduino Uno board to ensure a controlled temperature in the incubator based on the flowchart in Figure 4. When the incubator is switched on, the device initializes. Temperature sensor reads in the temperature of the chamber. If the acquired temperature is less than or 23 ^{o}C, heating unit (heater) is switched on while the cooling unit (cooler) is turned off. As the temperature of the chamber rises, the moment it is greater than or equal to 30 ^{o}C, the heating unit is turned off while cooling unit is turned on to dissipate the heat. The heating unit consists of heating pads while cooling unit comprises of Peltier modules, heat sing and cooling fan which is meant to dissipate the heat in the temperature chamber to achieve faster temperature regulation. The value of the temperature at any time is displayed on the LCD and the LabVIEW interface on a PC. In this manner, the incubator is maintained within the range of 23 to 30 ^{o}C as shown by the flowchart.

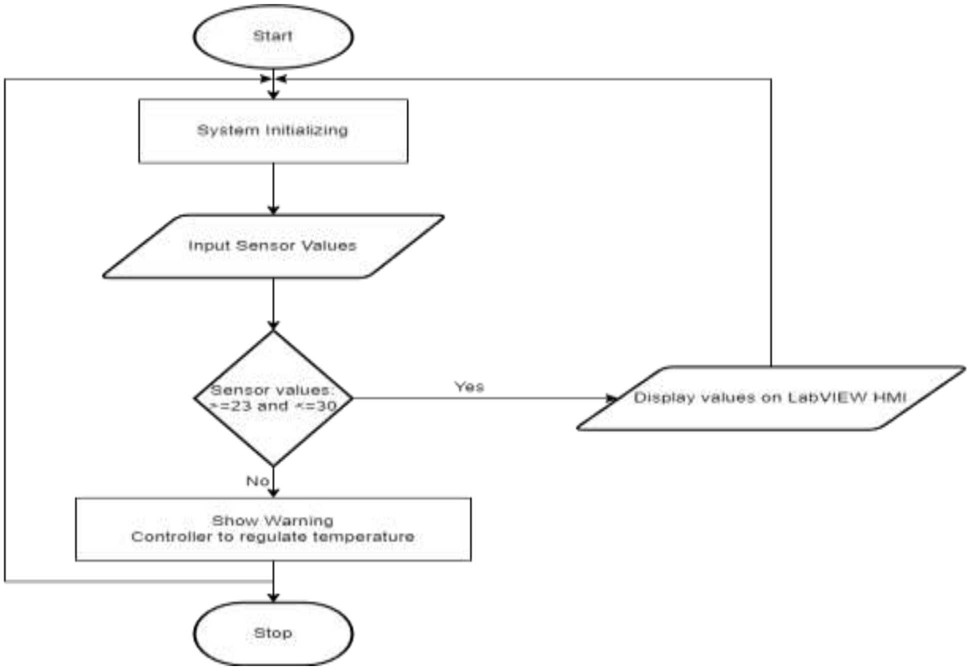


Fig 4: Flowchart of the incubator temperature control system

3.5 Development of LabVIEW-based HMI

While sections 3.1 to 3.4 emphasize mainly the hardware interfacing of the Arduino microcontroller and the PC hosting the LabVIEW, this section will be discussing the software configuration involved and graphical programming of LabVIEW to achieve the HMI. To do these different types of software are installed including LabVIEW software from National Instruments Kodosky, (2020). Visual Package Manager (VIPM), Digilent LINX and the Arduino Integrated Development Environment (IDE) which will be used to communicate with LabVIEW. Each of these software are shown in Figures 5a to 5d.

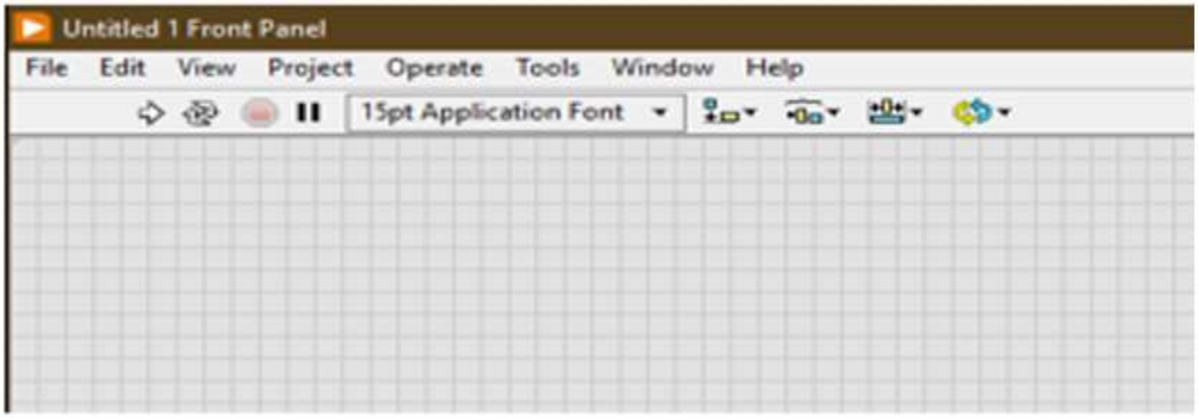
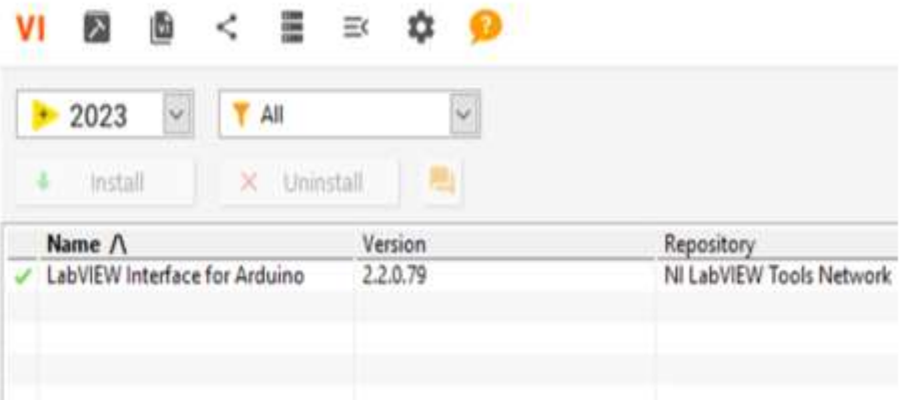


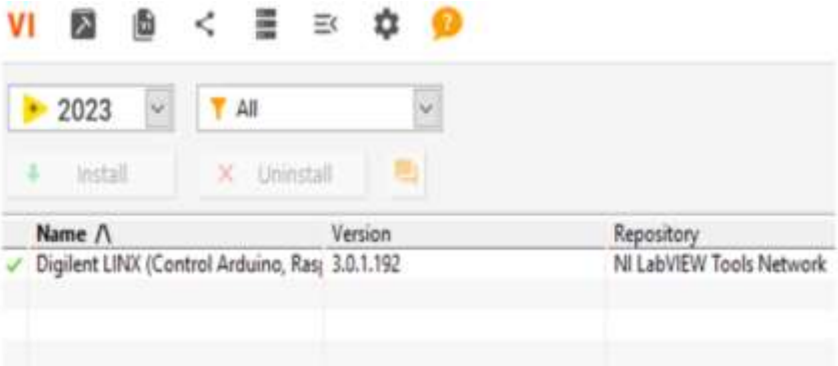
Fig 5a

Fig 5b



Name /\	Version	Repository
✓ LabVIEW Interface for Arduino	2.2.0.79	NI LabVIEW Tools Network

Fig 5c

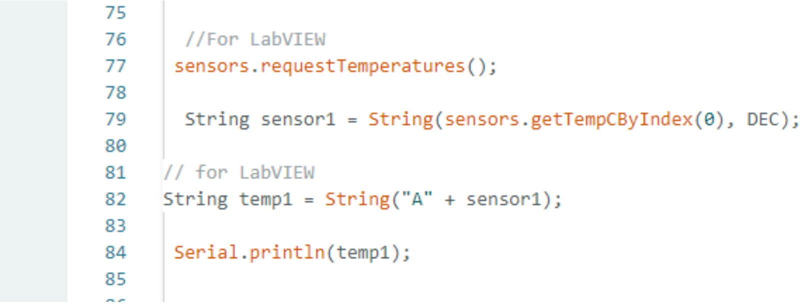


Name /\	Version	Repository
✓ Digilent LINX (Control Arduino, Ras	3.0.1.192	NI LabVIEW Tools Network

Fig 5d

Figs 5a-d: Software required for configuring the Arduino-LabVIEW interfacing

Communication between Arduino microcontroller and LabVIEW is achieved with the code in the Arduino IDE shown in Figure 6. In which temperature values were read as string variables which were later displayed in a user-friendly interface of LabVIEW.



```
  //For LabVIEW
 sensors.requestTemperatures();

  String sensor1 = String(sensors.getTempCByIndex(0), DEC);

// for LabVIEW
String temp1 = String("A" + sensor1);

 Serial.println(temp1);
```

Fig 6: Arduino code for LabVIEW

Unlike many code-based programming languages, LabVIEW is a graphical programming language that operates in two modes which are the block diagram and the front panel. The programmer uses a drag and drop procedure to achieve programming in the block diagram section and runs the programme at the front panel which is the user-friendly interface. For this research, Figures 7a and 7b is the Block diagram and front panel of the automated incubator.

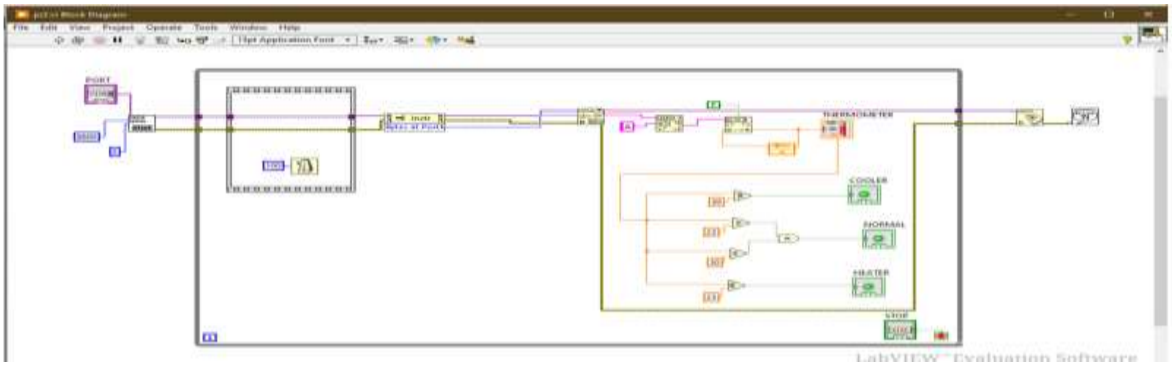


Fig 7a: LabVIEW block diagram

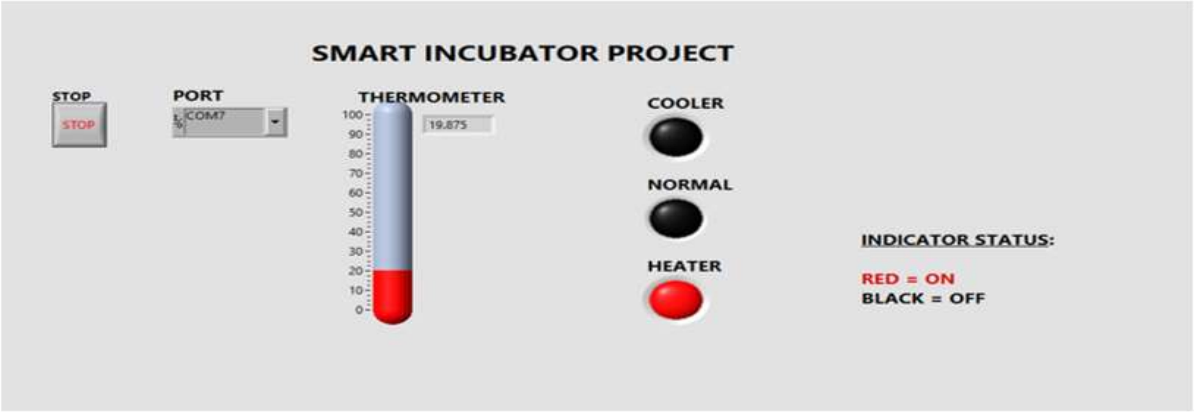


Fig 7 b: front panel for the automated incubator

4. SYSTEM PERFORMANCE EVALUATION

Tests were performed to verify the performance of the developed system as illustrated in sections 4.1 and 4.2.

4.1 Comparison of Developed Sensor and Digital Thermometer Output

The one-wire temperature sensor (DS18B20) used in this work is a factory-calibrated sensor. Its measurement was compared with a standard digital thermometer as a way of recalibrating it for better accuracy. Fig 8b is the graph obtained. Correlation of the developed temperature measuring device and the standard digital thermometer was calculated retuning a value of 0.999352. This shows a high degree of accuracy. Figures 8a and 8b respectively illustrate the experimental set up and the graph obtained when the measurement from the Arduino LCD / serial monitor was compared to the values obtained in LabVIEW front panel. The system was also tested under different conditions by turning on the incubator and observing it performance at the temperature rises, stabilize and confined within the specified range of 23 °C to 30 °C.

It could be observed from Figures 9a to 9c that if the temperature goes below 23°C the indicator “HEATER” will turn ON as shown in Figure 9b indicating that the heating pads in the incubator have come ON and will remain ON until the temperature value maintain the normal range and then the indicator “NORMAL” will turn ON as shown in Fig 9a. When the temperature starts to get high and above 30°C the “NORMAL” indicator goes OFF and the “COOLER” indicator comes ON. The “COOLER” indicator shows that the cooling device (Peltier) is ON and it is cooling the system in order for it to go back to the desired temperature range. This goes on and on and as such the incubator is able to maintain desired temperature and indicate in real-time on the LabVIEW user-friendly interface as well as the LCD / serial monitor. The developed system provides a test bed through which electrochemical sensors can be introduced for further study of microbial behaviours with non-intrusive constraints Song, (2016). Also, the GUI of this research is more user-friendly compared to the Python-based interface (Sánchez et al., 2020). The developed temperature chamber will also be useful in the simulation of temperature cycles including day and night temperature, and seasonal temperature, and the measured values will be made available to researchers on their PCs in real-time. When compared to microbial incubators being sold off-the-shelf, the developed LabVIEW-based incubator is much cheaper given the fact that it is developed from scrap refrigerators thereby creating wealth from electronic waste that would have been a menace to society.

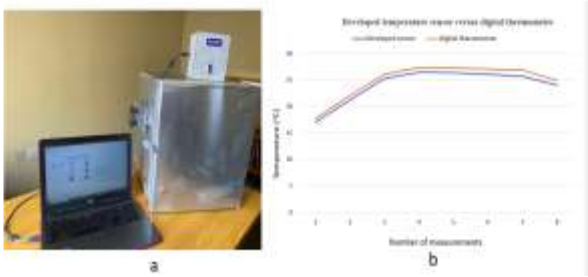


Fig: 8a) Experimental Setup, b) Graph of temperature from the Arduino serial monitor

Figure 9a: Automated Incubator in Normal Temperature mode

Figure 9d: LCD displaying temperature of the incubator

Figure 9b: Automated Incubator turns on the Cooler

Figure 9c: Automated Incubator turns on the Heater

Fig 9: The automated incubator under different testing modes shown in LabVIEW interface and LCD

4.2 Comparison of Developed Sensor Serial Monitor and Labview HMI Output

Experiment was carried out in which the automated incubator was made to operate for a period of 140 second during which the temperature of the developed sensor displayed on the computer serial monitor and that of the LabVIEW HMI were recorded as shown in Table 1 and Figure 10. The mean absolute error of the obtained temperature was computed using equation 8 and found to be 0.01 ^{o}C, depicting a very high degree of accuracy. From Figure 10. It could be observed that the automated incubator exhibited a high degree of reliability in maintaining the incubator temperature between the desired lower limit of 23 ^{o}C and upper limit of 30 ^{o}C. It could also be observed that the total period of ON and OFF switching of the incubator to regulate temperature was 140 seconds which is just a little above 2 minutes. In future designs, the proportional, integral, and derivative controller will be adopted to provide better performance.

Table 2. Comparison of developed LCD/Serial monitor and LabVIEW temperature

Time (sec.)	Developed Temperature sensor (^{o}C))	LabVIEW Temperature (^{o}C))	Absolute Error (^{o}C))
0	26.1900	26.1990	0.0090
5	25.6700	25.6700	0
10	25.1400	25.1420	0.0020
15	25.0000	25.0000	0
20	24.7900	24.7900	0
25	23.5000	23.5000	0
30	24.0000	24.0000	0

35	24.8900	24.8900	0
40	25.0100	25.0120	0.0020
45	25.4000	25.4000	0
50	26.0000	26.0000	0
55	26.2300	26.2320	0.0020
60	26.4400	26.4430	0.0030
65	27.1200	27.1240	0.0040
70	27.7500	27.7520	0.0020
75	28.2500	28.2530	0.0030
80	28.69000	28.6920	0.0020
85	29.0600	29.0610	0
90	29.4400	29.4432	0.0032
95	29.6900	29.6920	0.0020
100	29.6900	29.6930	0.0030
105	29.7800	29.2840	0.4960
110	30.0000	30.0200	0.0200
115	30.0000	30.0300	0.0300
120	29.8900	29.8920	0.0020
125	29.8000	29.8010	0.0010
130	29.2100	29.2120	0.0020
135	28.0300	28.0330	0.0030
	Mean Absolute Error (MAE)		0.014278571

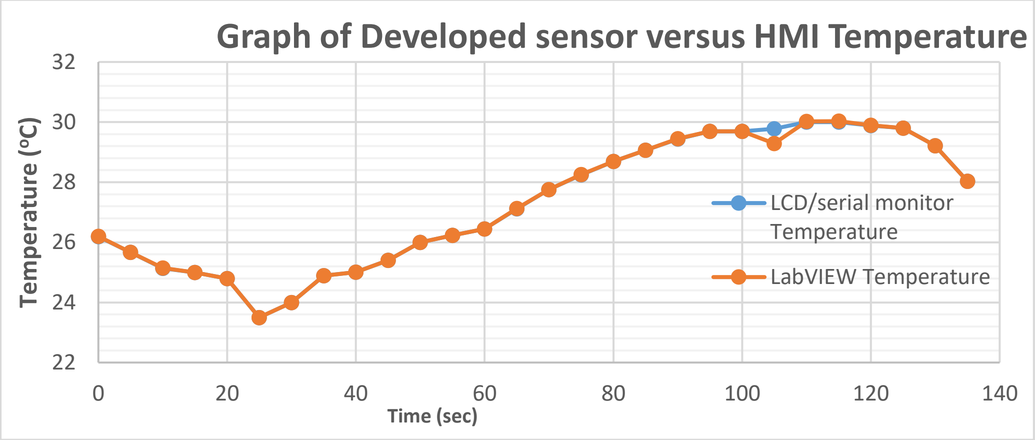


Fig 10: Comparison of LCD/Serial monitor and LabVIEW temperature

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_a - y_b| \qquad (8)$$

Where y_a is the developed sensor temperature, y_b is the LabVIEW temperature, \n\ is the number of samples obtained, which is 21 during this experiment. When the MAE for the temperature output of the developed sensor on the serial monitor and the LabVIEW-based HMI was computed using equation 1, the MER was found to be 0.014278571or approximately 0.01 seconds.

4.3 Comparison with Existing Manual and Automated Incubators

The developed system was compared with existing manual and automated incubators using varying parameters, as shown in Table 3. In terms of temperature control, manual incubators use basic thermostat or at times, they are manually regulated, while automated incubators use PID control. In case of the developed system, it uses a microcontroller-based control with lower and upper limit levels and displays its results in graphical form on a real-time basis. This is unlike the manual incubators where users need to open the door to monitor or the automated incubator that uses LCD. In terms of level of automation, the manual incubator is fully manual, automated incubator provides some degree of automatic control whereas the approach in this research provides a PC-based control with the potential of remote control over a local area or wide area network. The cost is manual incubator is low, an automated incubator is moderate, while in the case of our developed system, the use of a scrap refrigerator for developing the incubator has considerably reduced the cost of the system. The concept of repurposing scrap refrigerators as incubators is another way of reducing the amount of electronic waste with injurious health implications.

Table 3: Comparison between manual, automated and developed incubator with LabVIEW HM Microbe Notes (2026).

Aspect	Manual Incubators	Automated Incubators (Typical)	LabVIEW HMIIncubator
Temp Control	Basic thermostat; manual adjustments	Auto PID/heater-fan; set ranges (e.g., 20-55°C)	Arduino-based hysteresis (23-30°C); heater/peltier
User Interface	None or basic analog gauges	LCD/serial monitor or basic digital	Graphical LabVIEW front panel; real-time graphs
Monitoring	Open door checks; no real-time data	Limited LCD readout; some alarms	PC-based real-time temp plots, status indicators (heater/cooler/normal)
Cost	Low	Medium-high (off-the-shelf)	Low (scrap fridge + Arduino/LabVIEW)
Automation Level	Full manual turning/monitoring	Auto temp; some humidity/CO2	Auto temp; PC control potential
Portability	High; simple setup	Medium; needs power/outlets	Low; PC-tethered via USB

4.2 Limitations of the Developed System

The limitations of the developed device include: It is PC dependent and needs a laptop for full HMI; not standalone like LCD, in its present form, it measures only temperature (23-30°C) and no humidity or CO2 yet, which are common parameters in advanced microbial incubators. Its validation does not yet involve the use of live organisms. Other limitations include the fact that the system is dependent on LabVIEW environment, which might not be available in every laboratory and some hardware components that might need to be ordered from abroad if it has to be implemented in a developing economy such as Nigeria.

5 CONCLUSION

In this work, a LabVIEW-based Human Machine Interface (HMI) for a microbial incubator was developed. Microorganisms are temperature-sensitive and so LabVIEW HMI provides a user-friendly method of monitoring temperature in real-time that will enable scientists to perform experiments and study microorganisms more closely without having to open the incubator chamber frequently. From this work, LabVIEW HMI is found to be more user-friendly compared to the Arduino LCD in terms of esthetics, volume of information available, and intuitiveness thus enhancing the user’s experience and improving efficiency. LabVIEW also has the inherent capability of being able to control hardware (incubator temperature) from the LabVIEW interface on the PC. This will allow remote control of the incubator from remote locations over a network. The developed system is able to maintain temperature within the specified range of 23 ^{o}C to 30

^{o}C and reach steady state in approximately 2 minutes, making the incubator temperature available to the user on a real-time basis. The MAE of 0.01 ^{o} C between its serial monitor and HMI temperature outputs attests to a high degree of accuracy and reliability of the developed HMI, Future work will involve testing the developed device with microorganisms and controlling the incubator from the LabVIEW interface over a network.

REFERENCES

Al-Sawaff, Z. H., Yahya, Y. Z., & Kandemirli, F. (2020). LabVIEW BASED TEMPERATURE CONTROL SYSTEM FOR NEONATAL INCUBATOR. Eurasian Journal of Science Engineering and Technology, 1(1), 20–26. https://www.researchgate.net/publication/345241789

Athimoolam, S. and Karuppasamy, P. (2022). “Smart incubator for premature baby in an IoT Applications,” *Journal of Semiconductor Devices and Circuits*, vol. 9, no. 2, pp. 1 –9.

Bolu, C. A., Domfang, C., Obiazi, A., Falade, F., Musa, T. A., Alinaitwe, H., ... & Wara, S. T. (2022, November). Appropriate Online Laboratories for Engineering Students in Africa. In *2022 IEEE IFEES World Engineering Education Forum-Global Engineering Deans Council (WEEF-GEDC)* (pp. 1-5). IEEE.

Devaprasanth, S., Bhavadessh, D. and Dhanunjaya, G. (2022). “Wearable sensor with web server based incubator monitoring and control management system,” *International Journal of Research Publications and Reviews*, vol. 3, no. 5, pp. 501 –506.

Fahmi, F., Shalannanda, W., Zakia, I., & Sutanto, E. (2020). Design of an IoT-based smart incubator that listens to the baby. IOP Conference Series: Materials Science and Engineering, 1003(1), 012153. https://doi.org/10.1088/1757-899x/1003/1/012153

Indhuja, G., Abinaya, R., Sneka, S., Lakshmi, M. and Karpagam, R. (2023) “Development of hybrid power baby incubator,” *International Journal of Research in Engineering and Science*, vol. 11, no. 5, pp. 582 –584.

Isa, A., Abba, I., Abdullahi, S. I., Zakari, A., & Babalola, O. A. (2018). Temperature Monitoring and Alarming Device Using LabVIEW Interface with Microcontroller. International Journal of Engineering, Technology, Creativity and Innovation, 1(2), 2636–4867.

Kodosky, J. (2020). LabVIEW. *Proceedings of the ACM on Programming Languages*, *4*(HOPL), 1-54.

Kommey, B., Akudbilla, D., Doe, G., & Amponsah, C. O. (2022). A low-cost smart egg-incubator. Sustainable Engineering and Innovation, 4(1), 22–33. https://doi.org/10.37868/sei.v4i1.id152

LabVIEW, (2023a). What is LabVIEW: Graphical System Design» Electronics Notes. (n.d.). Www.electronics-Notes.com. Retrieved February 18, 2023, from https://www.electronics-notes.com/articles/test-methods/labview/what-is-labview.php#:~:text=History%20of%20LabVIEW.

LabVIEW, (2023b). What is LabVIEW: Graphical System Design» Electronics Notes. (n.d.). Www.electronics-Notes.com. Retrieved February 18, 2023, from https://www.electronics-notes.com/articles/test-methods/labview/what-is-labview.php#:~:text=History%20of%20LabVIEW

Linu, J., Emmanuel, B. P., Sandesh, M., Muhammed, A. P., Sankaranarayan, C. M. and Vismaya, R. (2024). "Smart incubator using IoT," *International Journal of Progressive Research in Science and Engineering*, vol. 5, no. 5, pp. 25 –30.

Luo C., Yu H., Naidong W., Long F., Hong H., Zhou W.,and Chen C. (2026). An Integrated Hardware–Software Platform for Automated Thermodynamic Characterization of Gas–Solid Interfaces Using a Resonant Microcantilever. Micromachines 2026, 17, 428 https://doi.org/10.3390/mi17040428

Marafa, I. S., Sanusi, B. A., & Olabiyisi, S. O. (2020). Fault Tolerance and Real Time Monitoring Infant Incubator Model. LAUTECH Journal of Computing and Informatics (LAUJCI), 1(1), 2714–4194

Michal Soni, M., Madhan Siva, R., Thirupathi Vasan, R. V., Vineeth, S. A., & Nehru, S. A. (2017). LabVIEW Based Health Monitoring System Using Arduino. International Journal of Innovative Research in Science, Engineering and Technology, 6(3), 2347–6710. https://doi.org/10.15680/IJIRSET.2017.0603040

Muñoz-Berbel, X., Escudé-Pujol, R., Vigués, N., Cortina-Puig, M., García-Aljaro, C., Mas, J., & Muñoz, F. X. (2011). Real time automatic system for the impedimetric monitoring of bacterial growth. *Analytical letters*, *44*(16), 2571-2581.

Nisha, Z. A., & Elahi, A. H. M. F. (2014). Low Cost Neonatal Incubator with Smart Control System. Research Gate, 978(1), 4799–6399. https://doi.org/10.13140/2.1.4591.5201

Nwaneri, S. C., Sojobi, J. W., Oyelade, A. O., Ezenwa, B. N., Balogun, O. J., & Uregbulam, U. C. (2020). A Low-Cost IoT Based Neonatal Incubator for Resource Poor Settings. The West Indian Journal of Engineering, 43(1), 0511-5728.

Ogidan, O. K. (2014). *Design of nonlinear networked control for wastewater distributed systems* (Doctoral dissertation, Cape Peninsula University of Technology).

Patel, A., Chudasama, K., Patel, R., & Patel, S. (2019). LabVIEW based smart house control. *International journal of advanced scientific research and management*, *4*(2), 234-238.

Peprah, F., Gyamfi, S., Amo-Boateng, M., Buadi, E., & Obeng, M. (2022). Design and construction of smart solar powered egg incubator based on GSM/IoT. Scientific African, 17, e01326. https://doi.org/10.1016/j.sciaf.2022.e01326

Prasad G J, Prasanna Valan P, & Preethika Britto. (2015). GSM based Remote Patient Monitoring System for Neonate in Rural Areas. International Journal of Engineering Research And, V4(11). https://doi.org/10.17577/ijertv4is110176

Raharja, N. M. and Sugiyarta, I. S. (2022). "Current Trends in Incubator Control for Premature Infants with Artificial Intelligence Based on Fuzzy Logic Control: Systematic Literature Review", *Journal of Robotics and Control*, vol. 3, no. 6, pp. 863 –877. Available: DOI: 10.18196/jrc.v3i6.13341.

Sánchez, C., Dessì, P., Duffy, M., & Lens, P. N. L. (2020). OpenTCC: An open source low-cost temperature-control chamber. HardwareX, 7, e00099. https://doi.org/10.1016/j.ohx.2020.e00099

Song, Y. E., Boghani, H. C., Kim, H. S., Kim, B. G., Lee, T., Jeon, B. H., ... & Kim, J. R. (2016). Maximum power point tracking to increase the power production and treatment efficiency of a continuously operated flat-plate microbial fuel cell. *Energy Technology*, *4*(11), 1427-1434.

Soni, S. K. (2020). GREENHOUSE MONITOR AND CONTROL WITH LABVIEW202. International Journal of Advanced Research in Engineering and Technology (IJARET), 11(10), 0976-6480. https://doi.org/10.34218/IJARET.11.10.2020.023

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User interface design and evaluation*. Elsevier.

Subramanian, M., Sheela, T., Srividya, K., & Arulselvam, D. (2019). Security And Health Monitoring System of The Baby in Incubator. International Journal of Engineering and Advanced Technology, 8(6), 3582–3585. https://doi.org/10.35940/ijeat.f9353.088619

Toby, B. H. (2001). EXPGUI, a graphical user interface for GSAS. *Journal of applied crystallography*, *34*(2), 210-213.

Vertegaal, R., Shell, J. S., Chen, D., & Mamuji, A. (2006). Designing for augmented attention: Towards a framework for attentive user interfaces. *Computers in Human Behavior*, *22*(4), 771-789.

Volponi, M.; Zieli ´ nski, J.; Rauschendorfer, T.; Huck, S.; Caravita, R.; Auzins, M.; Bergmann, B.; Burian, P.; Brusa, R.S.; Camper, A. (2024). Total Automation of LabVIEW Operations for Science: A framework for autonomous control systems for complex experiments. Rev. Sci. Instrum. 2024, 95, 085116. [CrossRef]

Yan H, Wu T, Li X, Feng Z, Ge M, Zhang L and Dong W-F (2024) Establishment of the microscope incubation system and its application in evaluating tumor treatment effects through real-time live cellular imaging. Front. Bioeng. Biotechnol. 12:1447265. doi: 10.3389/fbioe.2024.1447265.

Author's Brief Profile

Ogidan Olugbenga K. is a specialist in Systems Automation and Embedded Control. His research interests include: Smart (automated) Irrigation Systems, Internet of Things (IoT), Renewable Energy, Real-time and Embedded systems applications to E-Health. He received the degree "Doctor Technologiae" from the prestigious Cape Peninsula of University of Technology, South Africa. He is a recipient of two national patent awards from the Federal Government of Nigeria. He is presently a Reader at the Department of Electrical and Electronics Engineering, Elizade University, Ilara Mokin, Ondo State, Nigeria. His works have been presented in conferences and published in reputable journals both local and international.

Shobowale Yusuf I. obtained his B.Tech (Computer Engineering) degree from Ladoke Akintola University of Technology (LAUTECH), M.Sc (Computer Science) from Obafemi Awolowo University (OAU) and currently at the concluding stage of his P.hD. He has published works in different journals and conferences. His research areas are: Blockchain Technology, Wireless Communication, Network Security and Renewable Energy. He is presently a lecturer in the department of Computer Engineering at Obafemi Awolowo UniversityHe is presently a lecturer in the department of Computer Engineering at Obafemi Awolowo University, Ile-Ife

Hamza-Sambo Surayyha received the Bachelor of Engineering (B.Eng) degree from Elizade University, Nigeria in 2023. She is a graduate member of the Nigerian Society of Engineers (NSE). Her research interests include renewable energy, systems automation and embedded systems. She is presently undergoing her Master degree in Renewable Energy Engineering at the University of Aberdeen, United Kingdom.