

Article

Automation of a PCB Reflow Oven for Industry 4.0

Isaí Vilches ¹, Félix Juárez Durán ², Alfonso Gómez-Espinosa ^{1,*}, Mary Carmen García Carrillo ²
and Jesús Arturo Escobedo Cabello ¹

¹ Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Av. Epigmenio González 500, Fracc. San Pablo, Queretaro 76130, Mexico

² Center for Engineering and Industrial Development (CIDESI), Department of Electrical and Electronics Engineering, Av. Playa Pie de la Cuesta 702, Desarrollo San Pablo, Queretaro 76125, Mexico

* Correspondence: agomez@tec.mx; Tel.: +52-442-238-3302

Abstract: With the rise of Industry 4.0, its pillars (which include Internet of Things, “Big Data”, data analytics, augmented reality, cybersecurity, etc.) have become unavoidable tendencies for the automated manufacturing industry. Equipment upgrade is required to match the new standards of digitally assisted automation. However, not all factories in the medium to small range (or independent manufacturers) can afford to upgrade their equipment. Therefore, the availability of affordable Industry 4.0 upgrades for now-outdated devices is necessary for manufacturers in the SME range (Small-Medium Enterprises) to stay relevant and profitable. More specifically, this work revolves around the automation of printed circuit board (PCB) manufacturing, which is one of the most popular and profitable areas involved in this movement; and within it, the large majority of manufacturing defects can be traced to the soldering or “reflow” stage. Manufacturing research must, thus, aim towards improving reflow ovens and, more specifically, aim to improve their autonomous capabilities and affordability. This work presents the design and results of a controlling interface utilizing a Raspberry Pi 4 as a coupling interface between an MQTT Broker (which monitors the overall system) and the oven itself (which is, intentionally, a sub-prime model which lacks native IoT support), resulting in successful, remote, network-based controlling and monitoring of the oven. Additionally, it documents the design and implementation of the network adaptations necessary for it to be considered a cybersecure IIoT Module and connect safely to the Production Cell’s Subnet. All of this to address the inclusion of specific Industry 4.0 needs such as autonomous functioning, data collection and cybersecurity in outdated manufacturing devices and help enrich the processes of SME PCB manufacturers.

Keywords: Industry 4.0; PCB manufacturing; IIoT; reflow oven; soldering; Raspberry Pi



Citation: Vilches, I.; Juárez Durán, F.; Gómez-Espinosa, A.; García Carrillo, M.C.; Escobedo Cabello, J.A.

Automation of a PCB Reflow Oven for Industry 4.0. *Automation* **2023**, *4*, 78–93. <https://doi.org/10.3390/automation4010006>

Academic Editor: Eyad H. Abed

Received: 15 December 2022

Revised: 10 February 2023

Accepted: 13 February 2023

Published: 15 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

“The Internet of Things” (or IoT for short) is one of the pillars of “Industry 4.0” or “The Fourth Industrial Revolution”. Other pillars include Big Data and data analytics, augmented reality, cybersecurity, collaborative robots, additive manufacturing, cloud-based computing, artificial intelligence, and 5G networks [1,2].

Advancements in the 1970’s and 1980’s in electronics manufacturing technologies such as nanotechnology and Integrated Circuits (ICs) are now reflected in practically every electronic device we use; and even though prices have dropped dramatically since then, devices such as the PC, smartphone and tablet are still perceived as a considerable yet inevitable investment to all middle-class populations throughout the world.

We are, therefore, in need of reducing the time and cost associated with electronics manufacturing, and in a need of expansion in technology accessibility (and its benefits) to occur. Technology must transcend from a privilege (since currently only around 60% of world population have some form of Internet Access [3]) to a human right; while, at the same time, assuring product quality, traceability and abolition of unacceptable labor

conditions or involving employees in dangerous tasks during said processes. Innovations in electronics manufacturing, such as this, are of utmost importance at this time.

The most popular component format for mass produced circuits is, by far, SMT (Surface Mount Technology). Here, small components are located onto solder-paste pads by a Pick and Place machine [4]. The board is then placed inside a reflow oven and exposed to convective heat that reaches temperatures in the range 180–250 °C. As demand for smaller and more compact devices increases, SMDs grow to be both even more complex and more delicate. The miniaturization process includes the use of heat sensitive materials that could deteriorate on exposure to high temperatures [5]; thus, temperature control is necessary.

Not all manufacturing companies, however, possess the resources to upgrade their equipment to that of the latest tendencies (in this case, being Industry 4.0 compatible), and those that do will probably discard their past equipment to do so. This contradicts two of the most important pillars of Industry 4.0: Sustainability and reduced consumerism [6]. Ways to upgrade current equipment are to be sought, which is precisely what is done in this work.

Since this research addresses an Industrial IoT (IIoT) issue, a review of the associated literature is in order.

IIoT is a relatively recent and increasingly more relevant expansion of the IoT concept (defined to be the connection between people and things and between the physical and virtual objects for the exchange of information to perform coordinated tasks [7]). One of its primary characteristics is the use of a larger number of smaller, simpler devices (called modules) mutually connected through the Internet to gather data, which is later sent to a cloud-based server for analyzing (Data Science and Big Data pillars of Industry 4.0) [8].

However, relying on a cloud-based server for the allocation, storage, and request management of large volumes of information is known to have a number of potential downsides, and since access to robust cloud services can represent a significant investment for SME manufacturers, ways to reduce data traffic (and thus be able to rely on leaner servers) ought to be found.

The most common network architecture for industrial processes is the so-called “centralized architecture”, where sensors and actuators all communicate with a central processing unit (a PLC, for instance) but not with one another. This would increase data traffic to and from the cloud, as all data must pass through the processing unit first to be uploaded or retrieved. IIoT, on the other hand, favors and leverages a decentralized architecture, where IoT devices can communicate directly with both one another and the cloud server [9], thus reducing data traffic and the necessary server robustness.

The second strategy used to reduce network traffic is to use lightweight protocols. This topic alone has led to a number of research regarding the best network-layer protocol for IIoT, including Modbus TCP [10], QUIC [11], and MQTT [12].

According to the literature [13], the MQTT (Message Query Telemetry Transport) Protocol has been the most popular protocol for IoT applications for several years now, as it is lightweight and does not require large bandwidth. This has also made it popular for other IoT areas (besides Industrial), such as smart homes [14], medical equipment [15] and even agriculture [16].

Recorded implementations of IIoT in actual industrial processes are also already found in the literature, including a chemical production use case [17], Li-Ion battery manufacturing [18], and electric motor monitoring [19], but few integrations in the electronics manufacturing industry were found.

Géczy et al. [20] present a modular approach, by implementing new sensors to their devices through microcontrollers (such as the “Fishino”) ensembled with thermocouples and pressure sensors. However, this work focuses on transcending Serial Communication native to a Reflow Oven (RS-232) to network-based communication, elevating important data to and from an MQTT server.

Furthermore, no methods, proofs-of-concept or any form of research were found regarding the chosen niche: exploring potential, limitations, and strategies useful to upgrade

outdated devices as a solution to adopt newer and better technological tendencies without discarding previous equipment.

This work reports the design and performance of a Raspberry-Pi based network interface and controller for a Zhengbang ZB5040HL Reflow oven (as well as the followed strategies to the project network cybersecurity) to match the needs of an “Industry 4.0”-compliant PCB manufacturing cell. The interface is capable of operating the reflow oven’s actuators according to instructions allocated in an MQTT server and of uploading valuable monitoring data to said server, such as temperature history, process status, or any other relevant variable to be monitored. Collectively, these efforts result in a successful proof of concept of an interface capable of allowing deprecated devices to be upgraded to Industry 4.0 Standards. with autonomous operation, remote monitoring, and cybersecurity as the main criteria.

2. Materials and Methods

This section includes a more in-depth review of the concepts necessary for this work, such as reflow ovens and communication protocols.

2.1. State-of-the-Art Reflow Ovens

Currently, the most popular and effective technique to achieve solder-paste reflow is convection heating, as Infra-red heating has become a deprecated technology [21]. In convection ovens, a heat element is placed behind a flow-provoking element (most commonly, a resistance and a fan, respectively).

Reflow ovens can also be categorized into Drawer-Type Ovens and Conveyor-Type Ovens [22]. The first type is suitable for a lesser production volume and is less expensive; while the second type is used in high-yield factories, with size ranges upwards of 2 m long and weights of almost half a ton, as well as being more expensive.

As expected, the most fundamental need for Industry 4.0 automation is network communication. State-of-the-art reflow ovens and other manufacturing equipment (such as those distributed by Siemens [23] and Kurz Ersa [24] feature modern Industrial communication protocols such as ProfiNet or even native direct Internet connection to Broker-Client systems. However, the model used for this work only possesses a DB9 port for serial communication, which makes it a perfect fit for the “automation upgrade” proof-of-concept.

2.2. Communication Protocols

Communication protocols as a subject is a rather broad spectrum, including (for our purposes) UART (Universal Asynchronous Receive-Transmit) Protocol and the TCP/IP Model.

Both are binary protocols for serial communication, but with very different uses in industry. In this work, the MQTT fits the “Networks Layer” in the TCP/IP model connecting different computational devices (such as PCs or OS Embedded Processors), whilst RS-232 is simpler and is used for byte transfers between industrial devices.

2.2.1. RS-232

The RS-232 Communication Protocol is one of the most popular for unexpensive low-level industrial short-distance asynchronous serial communication. It utilizes higher tension levels than Transistor-Transistor Logic (TTL) and both positive and negative voltages for binary encoding (± 5 V with absolute maximum of ± 15 V), so proper transceiver hardware is required.

Whilst RS-232 is still popular among industrial applications, its domestic/development role is practically obsolete. The required hardware is not natively present in modern PC or laptops, as USB (which utilizes TTL level +5 V) has replaced RS-232. A proper adapter cable (which includes a level-adapting transceiver) is required and was purchased for this reason.

The DB9 connector (named this way for its 9-pin structure) is most often related to the RS232 Serial Protocol, but it is also commonly used for RS422 and RS485. Confirmation

could be found by disassembling the oven and finding the two components presented in Figure 1 soldered onto the main board.

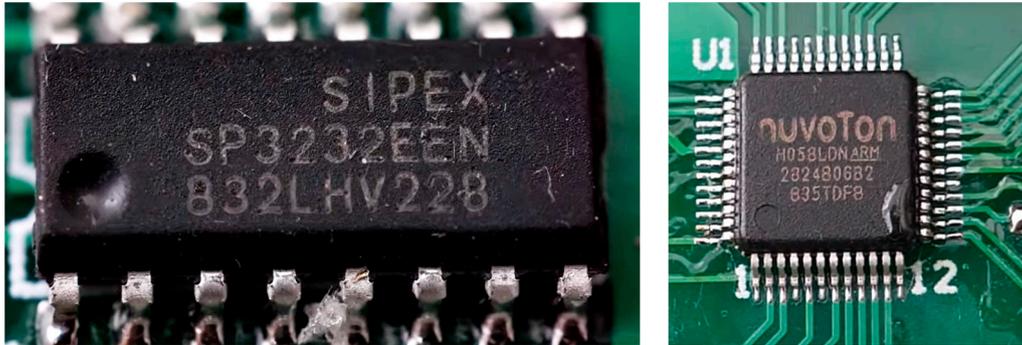


Figure 1. An SP3232EEN IC and an M058LDN 32-bit ARM Microcontroller. Both correspond to the RS-232 protocol.

The SP3232EEN is an RS-232 transceiver, while the diagram for the M058LDN Microcontroller (provided by NuvoTon) [25] confirms RS232 is the used protocol, as is its only supported protocol associated with the DB9 connector: While it also supports I2C and SPI, I2C commonly uses 4-pin connectors, while SPI can be found using 5-, 6- or 10-pin connectors.

2.2.2. MQTT

The Message Query Telemetry Transport Protocol (MQTT Protocol) is a subscribe-publish network-level protocol, and it has become the most popular for IIoT applications due to its reliability, low generation of network traffic and multi-programming-language library availability [26]. These features, coupled with the possibility of easily expanding the project to other devices in a manufacturing-cell network made MQTT the control signal distribution channel of choice.

The server (also called “Broker”) holds information organized into “topics”. When a client device is subscribed to the topic, the Broker makes sure to notify when a change in the allocated data occurs. Instead of generating constant traffic to deliver periodic, redundant, unchanged data, communication only occurs when changes do.

In this control architecture, MQTT Communication serves two purposes: Instructions such as Reflow profile settings or emergency stops are remotely sent to the coupling interface through “publishing” script, running in an external device, while valuable information such as real-time temperature monitoring or process status is also published to the corresponding topic.

As explained in Section 3.2, this project utilizes a dedicated workstation-based server to sustain the MQTT Broker; but the same concepts and benefits are transferable (and scalable) to cloud-based servers. This is useful to, for instance, multi-facility networks for larger manufacturing companies.

2.3. Coupling Interface

A device with internet connection and Serial RS-232 Voltage Level I/O was required as the coupling interface. This device would subscribe to a MQTT Broker and listen for instructions. It would then output an RS-232 signal to command the oven and publish a real-time update of the heat-cycle and any possible errors.

The network-connection implications make a Raspberry Pi 3 or Raspberry Pi 4 (coupled with a USB-RS232 transceiver-adapter) the best choice among the prototyping Microcontroller/Microprocessor selection, as their network configuration can be manipulated with ease using the Linux OS and both achieve optimal performance and be compatible with a cyber-secure network architecture which includes NAT routing, subnet configurations, etc.

3. Experimental Setup

3.1. Experimental Strategy

The main element of this project is a Wenzhou-Zengbang ZB5040HL Reflow oven, although the same conclusions and solutions should work for similar models such as ZB2015HL, ZB2520HL and ZB3530HL. These are all Drawer-type ovens which, as seen in Figure 2, include a female DB9 port for serial communication.



Figure 2. A ZB Series “Drawer type” reflow oven. Isometric (A) and back (B) views respectively.

According to the manufacturer’s website ZB series entry [22], the DB9 port is used for “PC On-Line operation”, heat profile modification and temperature monitoring. This software is said to be specifically designed for controlling a ZB Series Oven (shown in Figure 2). However, this is the only information given regarding the specific communication protocol or parameters.

The software does provide a way of “remotely” controlling the oven but typing and clicking is necessary in the GUI. The chosen solution was to implement a standalone controller which would replicate the output signals on the software which, according to the previous findings on Figure 1, are in the RS-232 Protocol.

With this information at hand, the provided software was installed, and a PC Serial IO monitor (Eltima Serial Monitor) was configured for RS232 Serial Communication.

Knowing that this software outputs messages in RS-232, we can now monitor the PC’s Serial Port and observe its behavior at different stages of the heat cycle. By combining the provided Zhengbang Software (Figure 3) and the Serial Monitor Software provided by Eltima LLC, the experimental setup shown in Figure 4 was used to find the RS-232 data packets corresponding to each command.

Said experimental Setup works as follows:

1. The user uses the PC (H1) to open the Zhengbang Software (S1) and manually executes different commands (like those shown in Figure 3).
2. H1 then outputs a logic level serial code through its USB port, to which a USB-DB9 transceiver (H2) is connected.
 - a. The Serial Monitor Software (S2) detects and reads said output; and shows it on screen. (Figure 5)
 - b. H2 translates the Logic Level Serial Code to RS-232 and delivers it to the DB9 port in the Reflow Oven (H3).
3. H3 Executes the command, e.g.: “Turn on the heating element with 50% power”, etc.

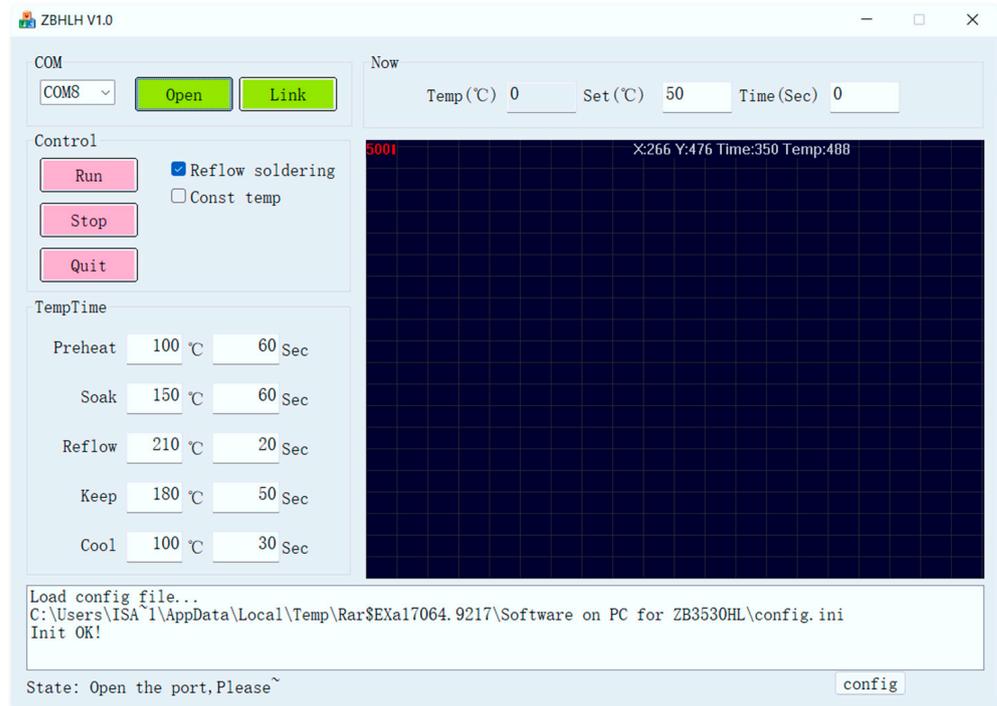


Figure 3. The Zhengbang Company Oven Control Software. Note buttons that indicate commands such as “Open”, “Link” or “Stop”. These are the necessary control outputs; each button corresponds to a specific command. The final interface is able to replicate such signals.

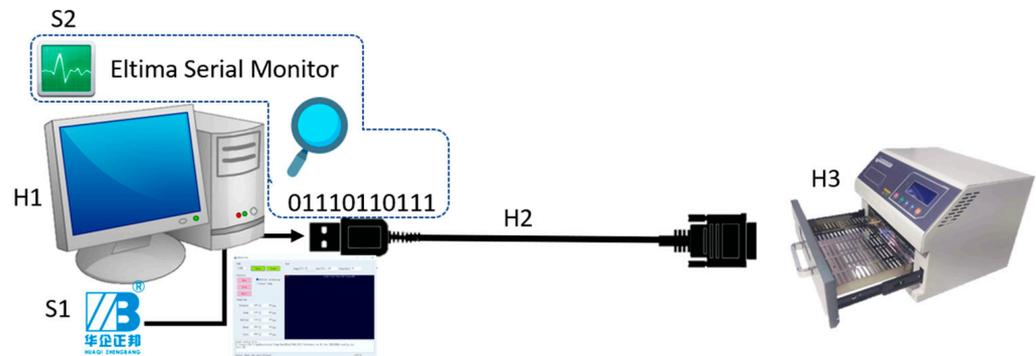


Figure 4. Experimental Setup. It consists of 2 pieces of Software (Labeled S1 and S2) and three pieces of hardware (Labeled H1, H2, and H3).

Figure 5 highlights in a red box a set of Hex Codes which stood out. IRP_MJ_WRITE Code with data column record 0x [A1 38 02 23 B2], for example, appeared immediately before the IRP_MJ_READ, which is a Serial Temperature Read command on which we can find the temperature 63 °C (encoded as 0x [00 3F]), Highlighted in the yellow square. This reading was confirmed by the provided software’s user interface.

A total of 40 tests were performed, following the methodology in Figure 6 (and utilizing the aforementioned Experimental Setup). This was done to clearly demonstrate that our interpretation of what each command does was correct. After the tests, Table 1, which includes the found Hex Codes, could be generated.

Function	Direct...	Status	Data
IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_PURGE)	UP	STATUS_SUCCESS	
IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_COMMSTATUS)	DOWN		
IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_PURGE)	DOWN		0f 00 00 00
IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_PURGE)	UP	STATUS_SUCCESS	
IRP_MJ_WRITE	DOWN		a1 38 02 23 b2
IRP_MJ_WRITE	UP	STATUS_SUCCESS	a1 38 02 23 b2
IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_COMMSTATUS)	DOWN		
IRP_MJ_READ	DOWN		
IRP_MJ_READ	UP	STATUS_SUCCESS	a1 38 02 00 3f 69 75

Figure 5. Example of a “WRITE” and a “READ” record on the Serial Monitor Software. Data corresponds to a Temperature Request and a Temperature Reading respectively.

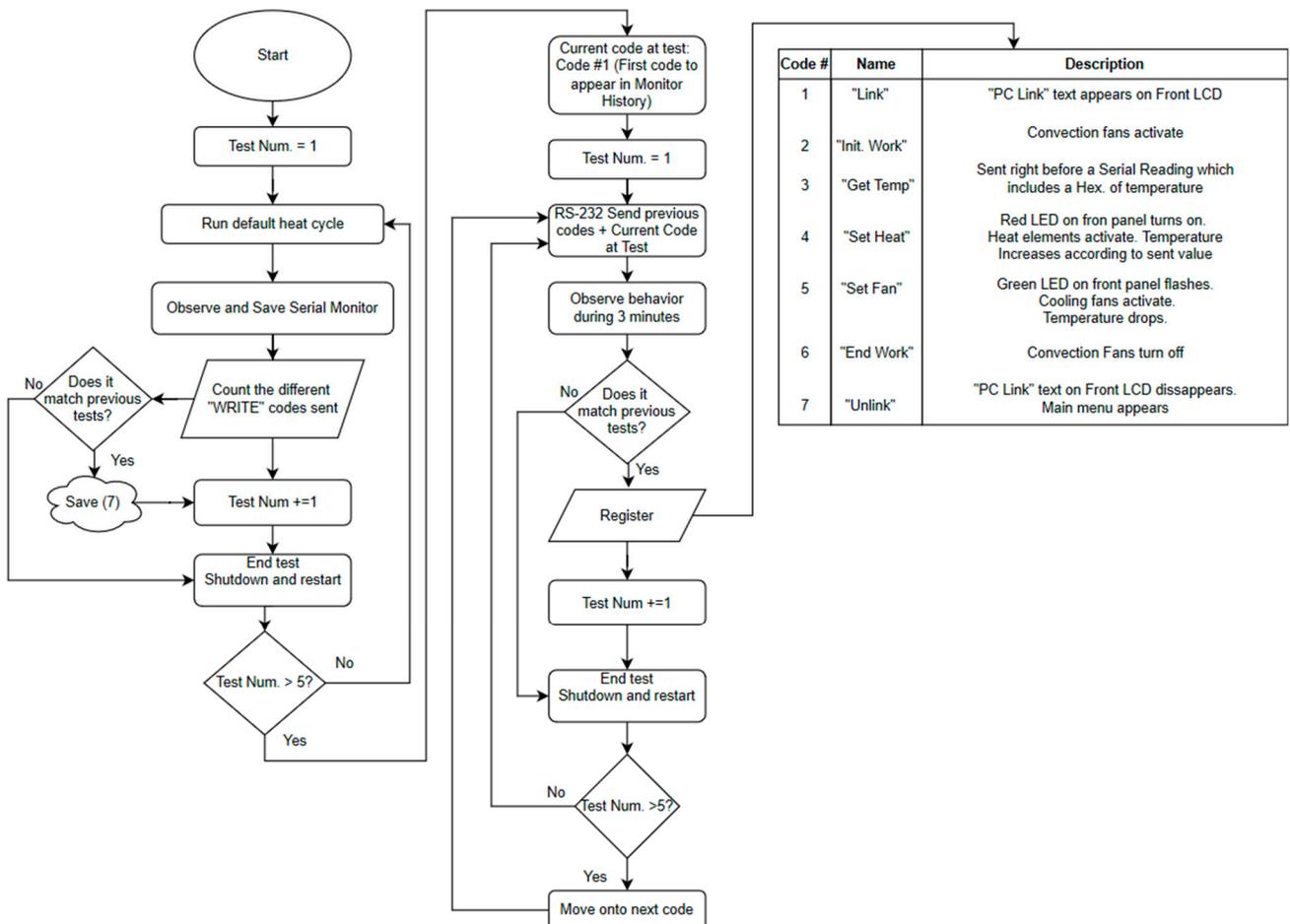


Figure 6. Flow diagram of experimental tests (Total of 40).

Table 1. The seven functions of PC-Link Mode.

Command	Serial HexCode (0x . . .)	Function
Link	[02] <i>[81]</i> [3E]	Oven enters "PC Link" Mode
Un-link	[03] <i>[00]</i> [00] [C0] <i>[81]</i>	Oven enters "Manual Mode"
Init. Work	[A0] [36] [01] [01] [7A] <i>[02]</i>	Heat element and cooling fan are enabled. Convection fan turns on.
End Work	[A0] [36] [01] [00] [BA] <i>[C3]</i>	Heat element and cooling fan are disabled. Convection fan turns off
Get Temperature	[A1] [38] [02] [23] [B2]	Request for temperature send (Integer in °C)
Set Heat	[A0] [34] [01] [heat] <i>[CRC16]</i>	Set heating element to power "heat" (8-bit integer)
Set Fan	[A0] [35] [01] [fan] <i>[CRC16]</i>	Set cooling fan to power "fan" (8-bit integer)

It is worth noting that the HexCodes described in Table 1 show bold bytes and italics bytes. Bytes written in bold are content bytes, whereas italics bytes (always the last two of the message) are generated using the Modbus sum check algorithm CRC16. This is a very curious design choice by Zhengbang, as RS-232 and Modbus have blatantly different message data structures.

This is important because while commands 1–5 can be treated as static data structures, commands 6 and 7 are dynamic, and will vary their CRC16 2-byte segment according to the "heat" or "fan" byte. Said bytes hold a "control action" value, ranging from decimal 0–255, which correspond to either the power of the heating element or the cooling fan respectively. Therefore, it is important to add a CRC16 library to the controller and generate the last two bytes before sending the UART message.

So far, the controlling interface device had to meet the following criteria: Both Wired (ethernet) and wireless network connection, adjustability of network parameters (such as setting a specific IP address for each NIC), a serial port for UART communication (preferable through USB connection) and to count with an available open-source process management daemon (for exception, errors, and reboot management) for end-purpose ergonomics.

Taking all of this into account, a Raspberry Pi 4 was seen as a perfect fit, and incorporated as the controller. A PID-Controller library, the "serialport" library (configuring one of the USB ports to asynchronous mode with a 9600 baud rate) and the "easy-crc" library were downloaded for their use during the final implementation.

3.2. Cybersecurity and the MQTT Server

As mentioned, the importance of cybersecurity has been increasing as an Industry guideline. A subnet arrangement as shown in Figure 7 was designed to protect the Production Cell (PrC) Local Network from a possible cyberattack.

The main constraint around which the subnet was designed was to prevent any possible "unclean device" (such as PCs which have not been factory reset prior to their involvement with the project) from directly connecting to the network.

By generating a Private WLAN (Wireless Local Area Network) utilizing the Wireless Router, PC ↔ Internet and PC ↔ Raspberry Pi traffic occurs separately, and through different NICs (Network Interface Cards). The same occurs for Raspberry Pi ↔ PrC Subnet (Accessed through the Ethernet Switch) and "PROJECTS" Network ↔ Internet. This last connection is additionally secured through the Research Center's NAT and DNS.

This, in turn, also keeps traffic from "unclean devices" apart from that of the Research Center's "PROJECTS" Network; since said network includes devices (such as servers, PLCs, etc.) which cannot risk being vulnerable to a cyberattack.

This model is, in an intentional manner, analog to the Local Networks of a factory's Production Cells and its Main Network respectively; model which SME manufacturers could leverage to improve their cybersecurity.

As for the MQTT server, it was also specially designed for the larger project containing this work: an Automated PCB Production Cell. It follows the architecture shown in Figure 8.

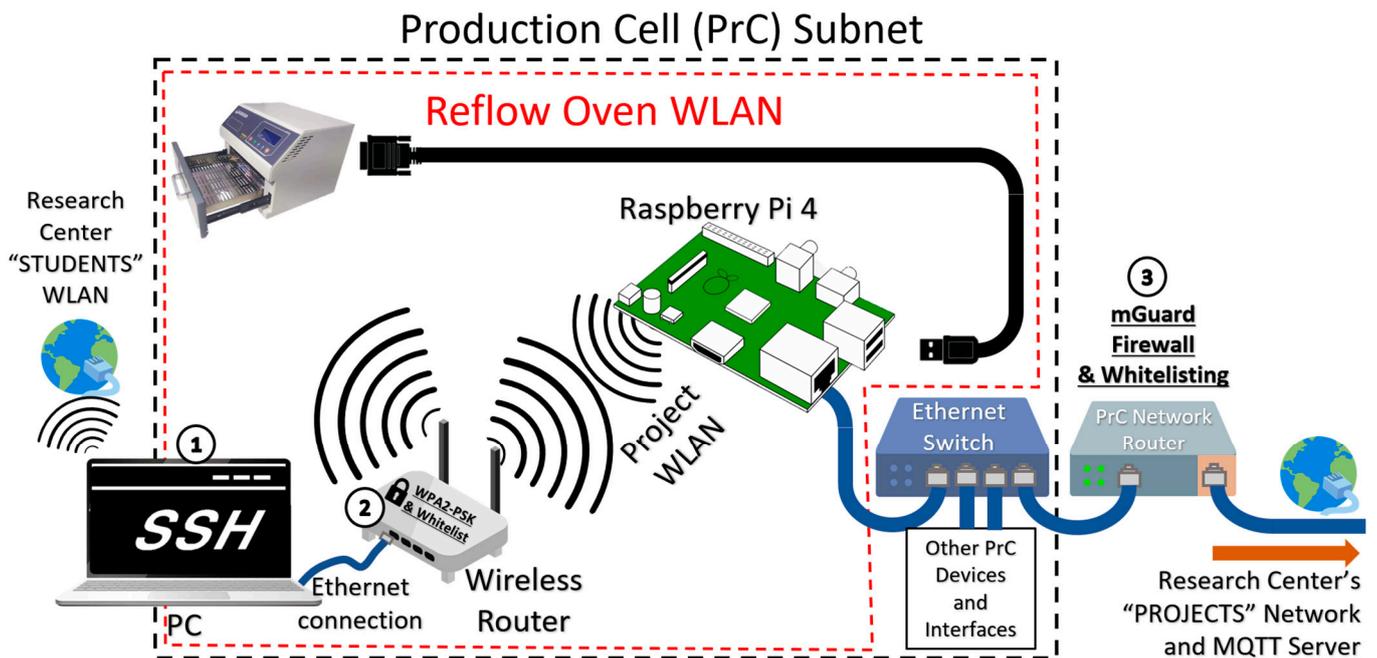


Figure 7. Labels 1–3 indicate the three different implemented strategies of cybersecurity: (1) Maintain data traffic from possibly “unclean devices” away from the reflow oven’s WLAN, by utilizing different NICs. (2) The wireless router managing the reflow oven WLAN follows the WPA2-PSK Security Standard and operates under a Whitelist. (3) The PrC Subnet is managed by a Phoenix mGuard Router, which runs the mGuard Firewall and enforces yet another Whitelist.

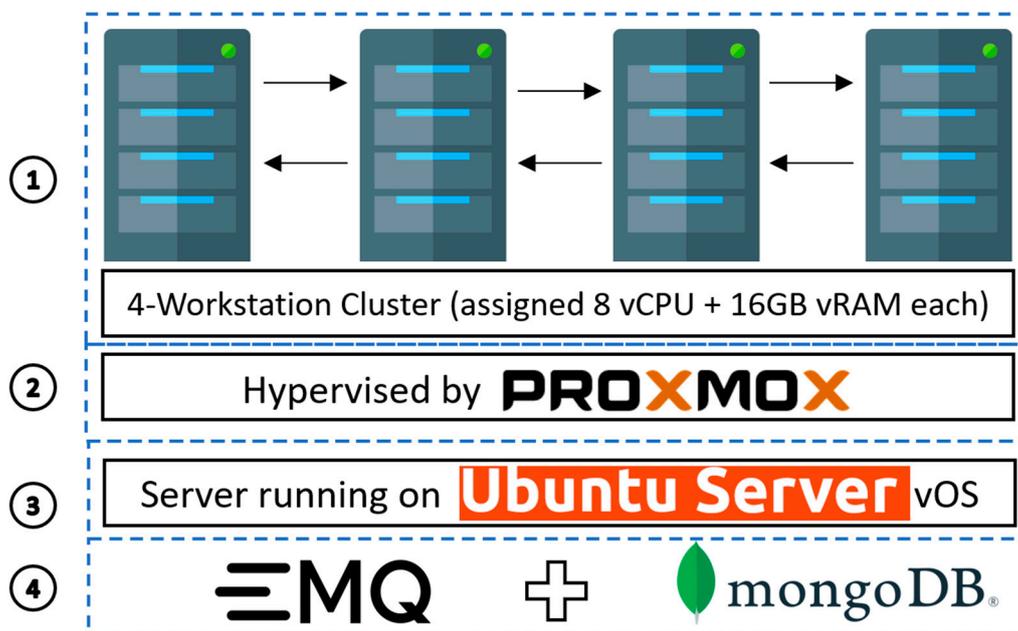


Figure 8. Labels 1–4 indicate the layers of the MQTT Server’s architecture. (1) Hardware: A workstation Cluster. (2) Virtualization Hypervisor: Proxmox assigns eight virtual nuclei of CPU and 16 GB of Virtual RAM from each workstation to the Clustered Virtual Machine running the Broker Services (3) OS: The Virtual Machine runs on the Ubuntu Server Virtual OS. (4) Services: EMQ and MongoDB.

The Broker itself runs on EMQ (An open-source MQTT Message broker, designed for IoT) and MongoDB (NoSQL Database). Gathered data, such as the temperature record during reflow, process status, and Control Actions would not benefit from the use of a

relational database; so, NoSQL (a non-relational database language) was chosen, as it presented a larger potential for scalability.

Since MongoDB already supports the JSON data structure format, this will prove useful during the code implementation, on which JavaScript was chosen as the final scripting language.

4. Results

4.1. Verification of the Found Command Codes

Whenever a device “Links” to the reflow oven (by sending “Link” Code UART 0x02 + CRC16) the Reflow Oven LCD displays a confirmation message (as shown on Figure 9) indicating successful connection was established.



Figure 9. Reflow Oven’s Liquid Crystal Display (LCD) confirms successful “PC-link” mode entered. Red Light Emitting Diode (LED) to the left of the display indicates resistance is currently active and temperature increase is occurring.

As with the “Link” command, every command in Table 1 had to be tested as manual UART Serial Sends (utilizing the “serialport” JavaScript library) to that our conclusion of their functionality was correct.

To attain confirmation that each command is, in fact, responsible for the attributed action, the commands were executed in a reciprocal “do-undo” method, as illustrated in Figure 10. Said diagram shows a gradual buildup, from the simplest commands (and their cancelling counterparts) to an actual temperature closed-loop control.

First, the system was linked and unlinked (Stage 1). Then, in addition, the convection fans were activated and deactivated in between (Stage 2). Then, a closed loop for temperature increase was added (Stage 3). Once that worked, it was exchanged for a temperature decrease loop (Stage 4) before, finally, combining them both (Stage 5).

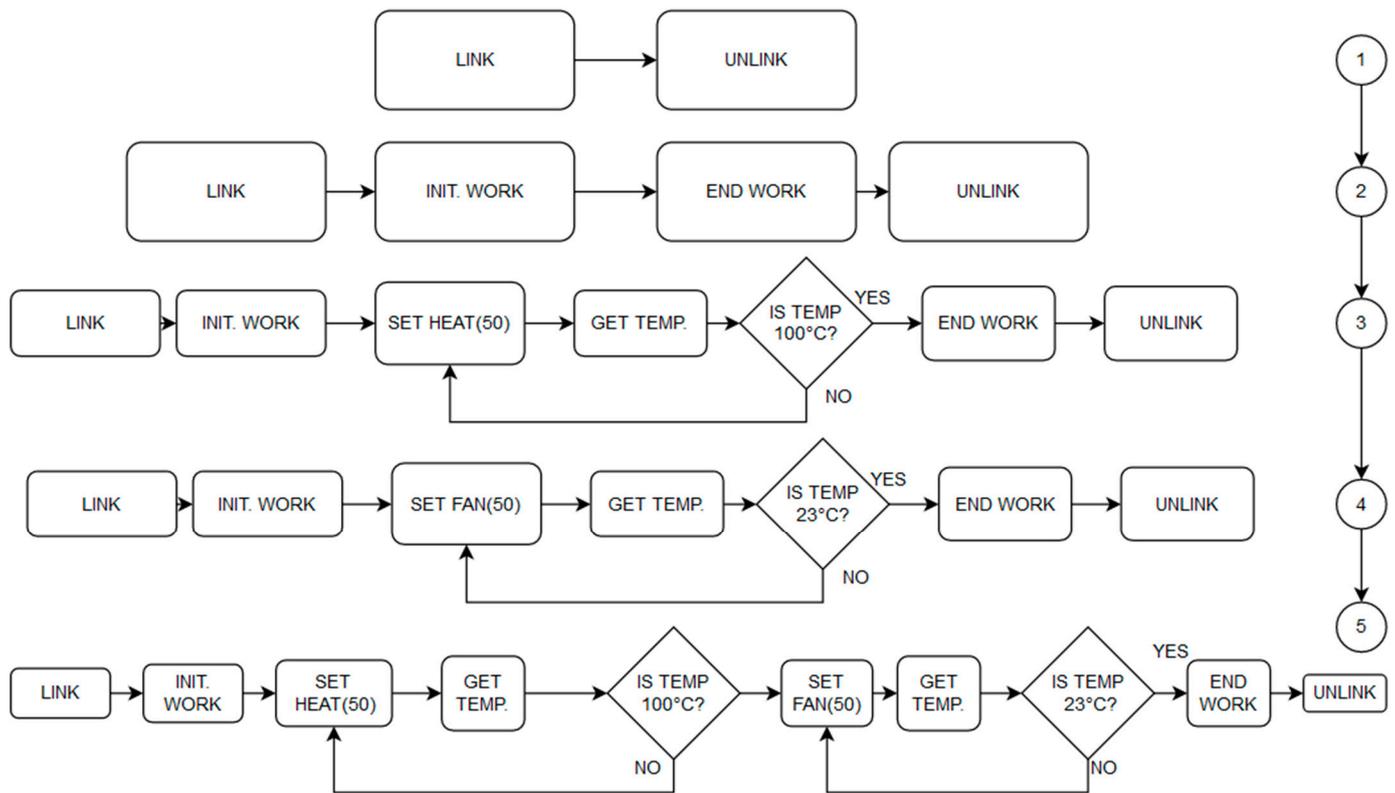


Figure 10. Gradual buildup of the code's logic, testing every command and their performance. Circular labels indicate "stages" within the gradual buildup.

4.2. Final Control Logic and Implementation

With the necessary architecture and materials (and having validated the interpretation of each command), the Autonomous Operation Flowchart was generated. It is shown in Figure 11, and it depicts the control logic used for the main code.

Said logic consists of a closed-loop control system, utilizing a PID Controller to adjust temperature (sensed from built-in thermocouples). A heating element and a fan (found inside the oven) are controlled to increase or decrease temperature respectively, and thus reach a set-point.

With each cycle, the "TEMP" topic (Temperature) is updated in the MQTT Server and, if necessary, the "STATUS" topic (which indicates current process status, such as the current stage within the reflow process) is updated as well. Such interactions are represented by clouds in the diagram.

The control logic was implemented into the Raspberry Pi utilizing the JavaScript language and the NodeJS Framework.

JavaScript is well known for being a suitable language for managing asynchronous tasks, leveraging features such as "callbacks" [27].

This was necessary to program periodic "interruptions", such as consistently updating the current temperature reading or the current control action.

However, JavaScript as a language is meant to run code over a web browser. That is why the NodeJS Framework had to be used, which simply allows us to run JavaScript Code directly from the Operating System (OS) or command terminal of, in this case, the Raspberry Pi 4.

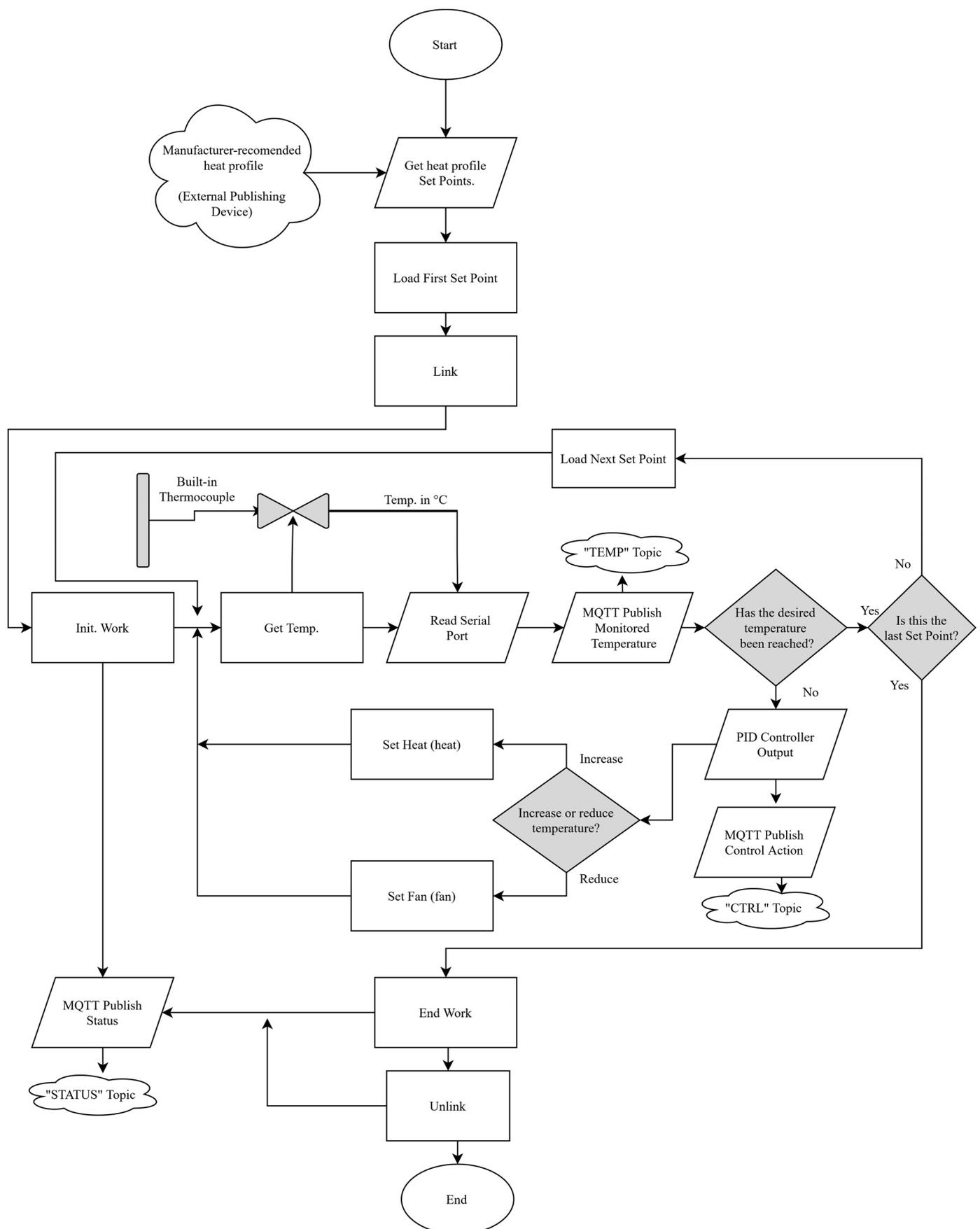


Figure 11. Automated Control Flowchart. Used as a guideline for programming the standalone controller.

4.3. Functional Validation and Results

Finally (and for communication validation purposes), a copper board with stencil-made solder paste pads was subjected to the temperatures shown in Figure 12 (an array of set-points based on the manufacturer's recommended reflow profile). The test was controlled remotely across the room.

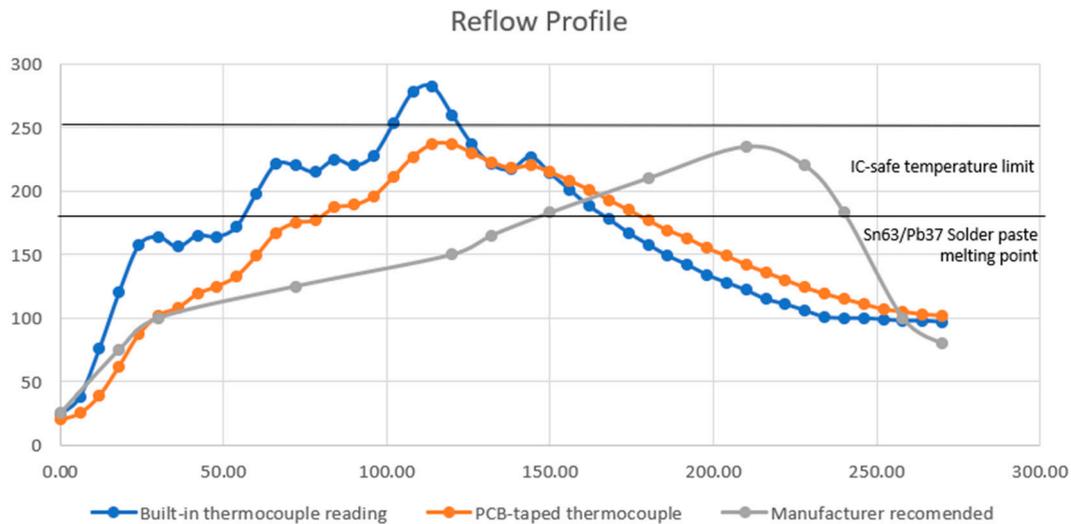


Figure 12. Reflow profile graphic. Offset is evident between the built-in ambient thermocouple (faulty reading) and the one taped directly onto the copper board (actual reading). The gray curve represents the recommended reflow profile.

Although the test resulted in successful solder reflow (which is discussed to greater detail in the next chapter), it certainly represents an area of opportunity for future research; as precise and accurate temperature control falls outside the scope of this research, which is to grant an outdated reflow oven with automation, cybersecurity, and data collection capacities, to comply with Industry 4.0 needs.

This, alongside electrical conductivity (which was tested using a multimeter) are the most important criteria to define successfully soldered pad.

5. Conclusions and Future Work

With these findings, functional remote and autonomous operation of the ZB5040HL was achieved, and successful reflow of solder paste was observed. We therefore consider this “Industry 4.0-compatible” modification (out of a previously outdated device) successful. The fact that the Command HexCodes have been (presumably) designed by Zhengbang for this Oven Series presents both an advantage and a disadvantage: This would make these exact same codes useful for other models of the same series, but also it would imply that a similar process to that described in Section 3.1 must be followed to obtain the corresponding command codes if the user is trying to implement this strategy over an oven of a different brand or series. The same is true for compatibility with other protocols, as this same strategy and components would also work for RS-422 and RS-485, but the right converter/transceiver would be needed, as these serial communication standards follow the same message structure as RS-232 but with different voltage levels and transfer-wiring types. It would not, however, work as easily with Modbus (another largely popular serial communication protocol) as the message structure between Modbus and the RS family are rather different. Reprogramming the interface for specific Modbus TTL serial outputs and purchasing the right transceiver would also be necessary.

Additionally, these findings may also be utilized to produce interfaces for other devices lacking native to IIoT capacities (such as the laser or mechanical etching machines which also belong to the PCB manufacturing process), and other communication protocols, such

as RS-485 or Modbus. In essence, this proof-of-concept provides guidelines for managing different industrial processes to upgrade their current equipment to comply with Industry 4.0's demand for automation, remote monitoring, and Industrial IoT.

As Industry 4.0 as a movement continues to influence the manufacturing industry; and the adoption of its features involves ever-more necessary upgrades, this work offers an option for SME manufacturers to embrace these innovations in an affordable manner, and thus, stay both relevant and profitable.

For future research, the improvement of temperature control precision and offset compensation are certainly major areas of opportunity. Figure 12 shows three different temperature profiles. The first one (blue plot) corresponds to the oven's built-in thermocouple found in the workspace near the heating elements (which reports a higher, yet incorrect, temperature). The orange curve plots the temperature sensed by an additional thermocouple, taped directly onto the copper board (which indicates the actual copper board temperature). This dual-measurement setup was made to find the magnitude in which the workspace and workpiece temperatures differ and determine if this can serve as a niche for future research. Finally, the gray curve plots the recommended manufacturer heat profile.

Although differences between the produced curve and the recommended profile may seem significant, the literature inclines towards considering the so-called "heating factor" (measured in $s^{\circ}C$) as the variable holding the most influence over solder joint strength and overall reliability [28,29]. For lead-free solder pastes, the ideal heating factor supposedly sits around $500 s^{\circ}C$ [30], whereas when using Sn63/Pb37 both the manufacturer recommended profile and the achieved profile easily reach upwards of $2500 s^{\circ}C$.

If, additionally, temperature could be measured accurately and without altering the thermodynamics of the reflow process (such as with IR sensors [31]), the whole system could serve to feed a sample space with data from multiple runs, that would allow for the same optimization contribution for different solder-paste types.

This is all to say that a robust scientific analysis to assess solder joint quality based on the heating factor criteria out leveraging on Big Data techniques (applied on an IoT module-fed database) is an excellent candidate for follow-on research. However, being such a relevant topic to the quality of the finished product (and falling outside the scope of this narrowly delimited research and its objectives), temperature and reflow control are not explored further in this article, as they represent a dedicated research effort focused only on optimization.

Whereas this work shows the first stage for Reflow Oven Upgrades, control precision must still climb another step for it to become a popular and reliable industrial-level solution.

Author Contributions: Conceptualization, F.J.D.; methodology, M.C.G.C.; software, F.J.D. and I.V.; validation, I.V.; formal analysis, A.G.-E.; investigation, I.V.; resources, M.C.G.C.; data curation, I.V. and F.J.D.; writing—original draft preparation, I.V.; writing—review and editing, I.V., J.A.E.C. and A.G.-E.; visualization, I.V.; supervision, J.A.E.C. and A.G.-E.; project administration, M.C.G.C.; funding acquisition, A.G.-E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The study did not report any data.

Acknowledgments: This work's authorship would like to thank project "321074-LANITED", of the National Council for Science and Technology (CONACYT), for providing the resources needed for the experimental stage of this research. As well, to Luis Govinda García-Valdovinos, Leonardo Barriga-Rodríguez, and Diego Iván Rodríguez Sanchez for their academic support and revision labor over this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Carvalho, N.G.P.; Cazarini, E.W. Industry 4.0—What Is It? In *Industry 4.0—Current Status and Future Trends*; IntechOpen: London, UK, 2020.
2. Ortiz, J.H.; Marroquin, W.G.; Cifuentes, L.Z. *Industry 4.0: Current Status and Future Trends*; IntechOpen: London, UK, 2020. [CrossRef]
3. International Telecommunication Union. *ICT Indicators Database*; UN: Geneva, Switzerland, 2020.
4. Iftikhar, B.; Malik, M.M.; Hadi, S.; Wajid, O.; Farooq, M.N.; Rehman, M.M.; Hassan, A.K. Cost-Effective, Reliable, and Precise Surface Mount Device (SMD) on PCBs. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *899*, 012007. [CrossRef]
5. Kamaruzzaman, L.S.; Goh, Y. Effects of alloying element on mechanical properties of Sn-Bi solder alloys: A review. *Solder. Surf. Mt. Technol.* **2022**, *34*, 300–318. [CrossRef]
6. Pereira, A.C.; Pereira, A.P.; Pereira, N.G. Industry 4.0 Technologies: What Is Your Potential for Environmental Management? In *Industry 4.0—Current Status and Future Trends*; IntechOpen: London, UK, 2020. [CrossRef]
7. International Telecommunication Unit. Internet of Things. Available online: <https://www.itu.int/en/ITU-T/techwatch/Pages/internetofthings.aspx> (accessed on 13 January 2023).
8. Hazra, A.; Adhikari, M.; Amgoth, T.; Srirama, S.N. A Comprehensive Survey on Interoperability for IIoT: Taxonomy, Standards, and Future Directions. *ACM Comput. Surv.* **2021**, *55*, 1–35.
9. Mirani, A.A.; Velasco-Hernandez, G.; Awasthi, A.; Walsh, J. Key Challenges and Emerging Technologies in Industrial IoT Architectures: A Review. *Sensors* **2022**, *22*, 5836. [CrossRef] [PubMed]
10. Jaloudi, S. Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study. *Future Internet* **2019**, *11*, 66. [CrossRef]
11. Khalifeh, A.; Mansour, M.; Alouneh, S. QUIC transmission protocol: Test-bed design, implementation and experimental evaluation. *J. Electr. Eng.* **2021**, *72*, 20–28. [CrossRef]
12. Mishra, B.; Kertesz, A. The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
13. Hillar, G.C. *MQTT Essentials—A Lightweight IoT Protocol: Send and Receive Messages with the MQTT Protocol for Your IoT Solutions*; Packt Publishing: Birmingham, UK, 2017.
14. Munshi, A. Improved MQTT Secure Transmission Flags in Smart Homes. *Sensors* **2022**, *22*, 2174. [CrossRef] [PubMed]
15. Tsao, Y.-C.; Cheng, F.-J.; Li, Y.-H.; Liao, L.-D. An IoT-Based Smart System with an MQTT Broker for Individual Patient Vital Sign Monitoring in Potential Emergency or Prehospital Applications. *Emerg. Med. Int.* **2022**, *2022*, 7245650. [CrossRef] [PubMed]
16. Jerrin Simla, A.; Chakravarthy, R.; Megalan Leo, L. An Experimental study of IoT-Based Topologies on MQTT protocol for Agriculture Intrusion Detection. *Meas. Sens.* **2022**, *24*, 100470.
17. Lam, A.N.; Haugen, O.; Delsing, J. Dynamical Orchestration and Configuration Services in Industrial IoT Systems: An Autonomic Approach. *IEEE Open J. Ind. Electron. Soc.* **2022**, *3*, 128–145. [CrossRef]
18. Liu, M.-C.; Chang, H.-C.; Huang, C.-H.; Hsu, F.-R. An implementation of industrial IoT: A case study in lithium-ion battery pack and assembly. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 3361–3375. [CrossRef]
19. Magadán, L.; Suárez, F.J.; Granda, J.C.; García, D.F. Low-Cost Industrial IoT System for Wireless Monitoring of Electric Motors Condition. *Mob. Netw. Appl.* **2022**. [CrossRef]
20. Geczy, A.; Kuglics, L.; Megyeri, I.; Gelbmann, R.; Harsanyi, G. Sensor-based IoT monitoring of Electronics Manufacturing in University Lab Environment. In Proceedings of the 2021 IEEE 27th International Symposium for Design and Technology in Electronic Packaging (SIITME), Timisoara, Romania, 27–30 October 2021.
21. Yu, H.; Kivilathi, J. CFD modelling of the flow field inside a reflow oven. *Surf. Mt. Technol.* **2002**, *14*, 38–44. [CrossRef]
22. Huaqi Zhengbang. *Drawer Type Reflow Oven*; Wenzhou Zhengbang Electronic Equipment Co., Ltd.: Wenzhou, China, 2021; Available online: <http://www.wzzbdz.com/en/product/302.html> (accessed on 25 October 2022).
23. Opcenter Execution Electronics IoT. Collecting Oven Data with Opcenter Execution. Siemens. 2020. Available online: https://www.plm.automation.siemens.com/media/global/en/KIC-Reflow-Machine-Data-Plugin-fs-82112-c6_tcm27-94905.pdf (accessed on 26 October 2022).
24. Ersä, K. Industry 4.0 Driven By Kurtz Ersä. Ersä, 2021. Available online: <https://www.driven-by-kurtzersa.com/> (accessed on 26 October 2022).
25. NuvoTon. M058LDN DataSheet. Nuvoton Technology Corporation. 2015. Available online: https://www.nuvoton.com/export/resource-files/DS_M051%28DN_DE%29_Series_EN_Rev1.03.pdf (accessed on 30 October 2022).
26. Technology Update. *MQTT Eases Software Development for Industrial Internet of Things*; Control Engineering: Downers Grove, IL, USA, 2016.
27. Kelhini, F. *Modern Asynchronous JavaScript*; Pragmatic Bookshelf: Frisco, TX, USA, 2021; ISBN 9781680509045.
28. Gao, J.G.; Wu, Y.P.; Ding, H.; Wan, N.H. Thermal profiling: A reflow process based on the heating factor. *Solder. Surf. Mt. Technol.* **2008**, *20*, 20–27.
29. Gao, J.; Wu, Y.; Ding, H. Optimization of a reflow soldering process based on the heating factor. *Solder. Surf. Mt. Technol.* **2007**, *19*, 28–33. [CrossRef]

30. Geczy, A.; Matyas, T.D.; Kaman, J.; Harsanyi, G. Application of Grid-Eye IR sensor for enhanced HMI and OSH purposes in Industry 4.0 reflow soldering environment. In Proceedings of the 2020 43rd International Spring Seminar on Electronics Technology (ISSE), Demanovska Valley, Slovakia, 14–15 May 2020.
31. Veselý, P.; Horynová, E.; Starý, J.; Bušek, D.; Dušek, K.; Zahradní, V.; Plaček, M.; Mach, P.; Kučí, M.; Ježek, V.; et al. Solder joint quality evaluation based on heating factor. *Circuit World* **2018**, *44*, 37–44. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.