Graduation Project
Bachelor's Degree Electronic Engineering

To Develop an Automatic Control Oven System to Mitigate Economic Losses for Delights of the Caribbean Food S.A. in San Juan, Tibás for the period of June to October 2017.

Student
Giovanni Heredia

Electronic Engineering Director
Ing. Jose Luis Medrano

Tutor
Ing. Mauricio Armas Sandi

October 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DEDICATION

I dedicate this project to God for providing me with the opportunity, intelligence, and perseverance to continue my dreams. Secondly, I owe my sincere appreciation to my entire family and relatives who have always encouraged and supported me over the years. Finally, I want to extend a special and profound appreciation to my wife, Rona and to my children, Chelsi and David.

ACKNOWLEDGEMENTS

I want to express my utmost appreciation and gratitude to our CEO, Lynn Young and to my company "Belize Electric Company Limited, a Fortis Inc. Company" for the scholarship and the study leave, in order to complete my goal in achieving my Bachelor's Degree in Electronic Engineering.

ABBREVIATIONS

A	Ampere, amps
ADC	Analog-Digital converter
ANSI	American National Standards Institute
ARM	Atmel ARM microprocessor chip
AVR	A "C" language environment for programming Atmel ARM processors
Bps	bits (baud) per second
Cloud	space in the internet
DAQ	Data Acquisition
GNU	An operating system known as: GNU not Linux
GPIO	General Purpose Input / Output pins
GPU	Graphic Processing Unit
GSM	Global System for Mobile communication
HD	High Definition
HDMI	High-Definition Multimedia Interface
H_r	Humidity (relative)
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICSP	In-Circuit Serial Programming pins (SPI)
IDE	Integrated Development Environment
I/O	Input / Output
IF	Forward current

Kg/m^3	Kilogram per cubic meter
LAN	Local Area Network
LCD	Liquid Crystal Display
LXT	Linux Terminal
LXDE	Lightweight X-11 Desktop Environment (Raspbian)
MISO	Master In, Slave out (SPI)
MOSI	Master Out, Slave In (SPI)
NTC	Negative Temperature Coefficient
NC	Normally Closed
NO	Normally Opened
OS	Operating System
PC	Personal Computer
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
Poly Fuse	Polymeric Positive Temperature Coefficient fuse
PMBOK	Project Management Body of Management
PPP	Point-to-Point Protocol
PTC	Positive Temperature Coefficient
P_{sat}	Pressure saturation
P_v	Pressure vapor
RTD	Resistance Temperature Detectors
RX	Receive signal
SCK	Serial Clock (SPI)

SCL	Serial Clock, I2C data bus
SDA	Serial Data, I2C data bus
SD micro	Secure Digital micro memorycard
SOW	Scope of Work
SS	Slave Select (SPI)
SPI	Serial Peripheral Interface communication (ICSP)
SSR	Solid State Relay
TX	Transmit signal
UART	Universal Asynchronous Receiver / Transmitter
USB	Universal Serial Bus
V_F	Voltage (forward)

CHAPTER I: PROJECT PROBLEM

1.1 SIGNIFICANCES OF THE PROBLEM

1.1.1 Introduction

This project pretends to automate a manual oven so that the operators' choices are limited to only choosing the specific product and the time to bake. After choosing the product, the automatic system then manages the required variables in order to produce an efficiently baked product.

This project overall design intends to include various means of safety precautions to protect the operators and personnel. One of the general idea and aim of this project's strategy includes the storage of the oven's usage records on an application utilizing the Wi-Fi/LAN via a local server. The expectation of this local design and implementation of this project is to reduce cost in comparison with the procurement of a similar solution of brand new automated oven.

1.1.2 Business Description

The micro business Delights of the Caribbean Food S.A. (known as Caribbean Delights) is dedicated to the production, distribution, and sale of Caribbean style food and pastries to supermarkets, restaurants and the public. A group of professional entrepreneurs, all originally from the Limon province, Costa Rica, formally registered the business in March 2009.

The company currently has two places both located in the San Juan district of Tibas, San Jose. One is a small restaurant that is opened to the public and the other is the production area. The production area is where the products' different processes transpire such as the preparation, baking, freezing and storage. The restaurant is located within the "Centro Comercial Plaza del Valle", the address is at the corner of "Calle central y Calle 55". The production area is located on the "Transversal 3", 100 meters east, and 300 meters south of the restaurant. Both of them are in the San Juan district, Tibas. The transportation of products is from the production area to the clients, supermarkets, snacks shops, and restaurants.

According to the company website: http://www.carideli.comthe company mission is expressed as "*Deleitar a nuestros clientes con lo mejor de los sabores exóticos del Caribe*" (Caribbean Delight, 2017), which translates to: "Delight our customers with the best of the exotic flavors of the Caribbean." The company vision according to the website is "*Ser reconocidos por los distribuidores y consumidores de productos alimenticios caribeños, como LOS MEJORES en sabor, calidad y servicio, por medio del compromiso, entusiasmo y motivación de los colaboradores*" (Caribbean Delight, 2017). Which translate to: "To be recognized by distributors and consumers of the Caribbean style food products, as THE BEST in taste, quality and service, through our commitment, enthusiasm and motivation of our coworkers".

The business has eleven employees; the administration section consists of a general manager, an administrator, and a sales representative. The restaurant comprises of two employees who attend the public. The production sections have six employees who work in the various processes of preparing the different products. Caribbean-style pastries and food such as patty, tart, "*panbon*" and "*galleta*" are the products that are baked by the crew in the production section.

1.1.3 Problem Justification

At Caribbean Delights, there are various productions' processes, all which are important for the productions of the final product. However, within all these processes is the baking process, which is specifically affected by the implementation of this project. Currently, the baking process includes the usage of a manual oven. The entire operation of the manual oven is dependent on human interventions and this creates losses, due to inconsistencies of the various variables. This may cause that the final products may be, over or under cooked, or too dry due to lack of moisture. Apart from products lost, other inefficiencies may be created such as excess usage of electricity and gas. All these operational inefficiencies may eventually cause an economic decline for the company.

1.2 PROBLEM DEFINITION

In the production area, the baking processes are manual procedures that depend on human functionality. For the process of preheating the oven, someone has to open the gas valve, light a pilot fire to ignite the burners of the ovens. The oven is preheated for approximately ten minutes and since there is no thermometer, there is no verification of the temperature. Then the products are placed inside for baking. The timer is set to the desire time and starts once the product is placed inside the oven. Periodic monitoring by viewing is used to verify if the product is baking properly. Since all this procedure is manual, there are many possibilities for human error. Currently, these errors are not affecting the production, since the baking is mostly scheduled to once or twice a day for two or three times per week.

The main reason for the need for automatic oven is that one of the medium-term goals of the business is to export their products to foreign markets. The plan is to export finished baked product, shipping them frozen. Therefore, the automatic system should manage the specific product preset variables. Once the baking process is automatic, losses are reduced since the control system manages the consumption of gas and electricity, which creates an increase of efficiency.

1.2.1 Question of the Problem

How to develop an oven automated control system utilizing the Arduino microcontroller in order to mitigate losses and increase efficiencies for the microbusiness Caribbean Delight by November 2017?

1.3 OBJECTIVES

1.3.1 General Objective

To develop an automated control system for a manual oven utilizing Arduino microcontroller in order to reduce losses and increase efficiencies of the micro business Caribbean Delight by November 2017.

11.3.2 Specific Objectives

- To define the minimum processes require utilizing the touchscreen located on the oven given that the Arduino is programed.

- To determine the hysteresis variables by measuring the time and temperature for the different products to guarantee properly baked products.

- To implement a log register to track the oven history using a database engine via a computer assuming that it communicates with the Arduino.

- To analyze the feasibility of designing and implementing a customized solution in comparison to an existing factory-made automated oven solutions.

1.4 OPPORTUNITIES AND LIMITATIONS

1.4.1 Opportunities

- The operators choose the product and this enables the control system to utilize the necessary variables to preheat the oven within a specific time frame (approx. 15 to 20 minutes) to the required temperature.
- To minimize human interface in order to allow the control system to manage the oven for properly baked products.
- Reduce the safety risk of leaking gas by installing a gas actuator valve and electrical igniter within the control system and removing the existing pilot system.
- Providing the Arduino is on the network, an authorized user is able to access a log register by utilizing a database to obtain a report of the oven's history.

1.4.2 Limitations

- A limitation on the efficiency of the system is that the oven has completed the preheat cycle, but the operator does not initiate the baking cycle immediately. This affects the efficiency of the usage of gas.
- Another limitation is the heat convection within the oven due to its design and airflow limitations.
- Also, a limitation is the possibility of implementation delay due to various causes such as shipment delay due to natural disasters and contractor's inefficiencies.

CHAPTER II: THEORETICAL FRAMEWORK

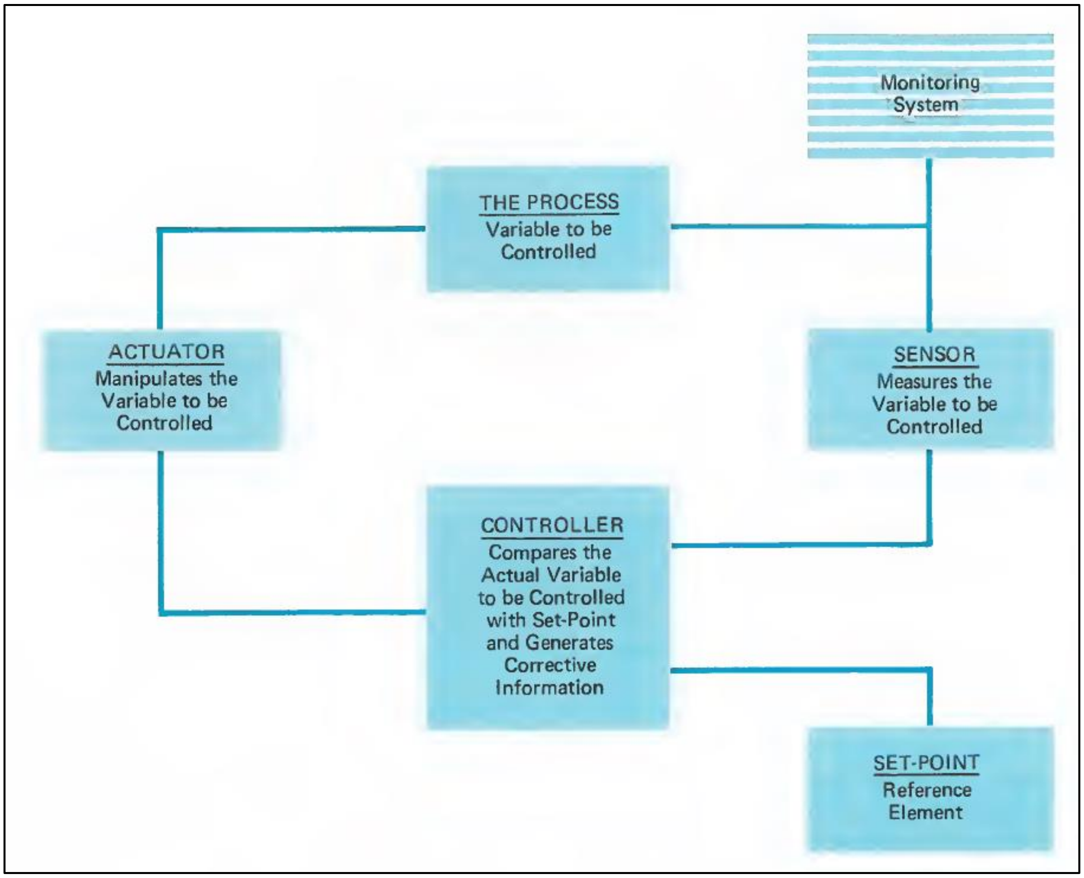
2.1. AUTOMATIC CONTROL SYSTEM

For a few decades, automatic control systems have continued to play an increasingly important role in the development and advancement of modern civilization and technology. The science of engineering is the composition of knowledge and control over resources and systems for the benefit of humanity, as mentioned in the book Modern Control Systems: “Engineering is concerned with understanding and controlling the materials and forces of nature for the benefit of humankind. Control system engineers are concerned with understanding and controlling segments of their environment, often called systems, to provide useful economic products for society” (Dorf & Bishop, 2011, p. 2). We used automated system solutions to control our surrounding systems to generate valuable products for the economic benefits of our society.

There are many benefits to implement and utilize an automatic control system. One of the principal functions of an automatic control system is to reduce human action in engineering processes and procedures. There are various motives to reduce human action especially if the equipment or the area where the equipment is located is in a dangerous zone. Other reasons are to mitigate the human error factors since automated control systems are consistent, efficient, durable, precise, accurate, and diligent. Presently, with the advantage of computers, modern automated control systems are developing further into self-learning, optimum, and adaptive systems.

An automatic control system interfaces with its environment using devices such as sensors and actuators to produce the desired system response. “A automatic control system is an interconnection of components forming a system configuration that will provide a desired system response” (Dorf & Bishop, 2011, p. 2). To obtain the output response, an automatic control system is utilized to regulate the actions of its devices. Generally, there are two main classifications of control systems, the open-loop system, and the closed-loop (feedback) system.

Figure 1: Block diagram of an Automatic Control System

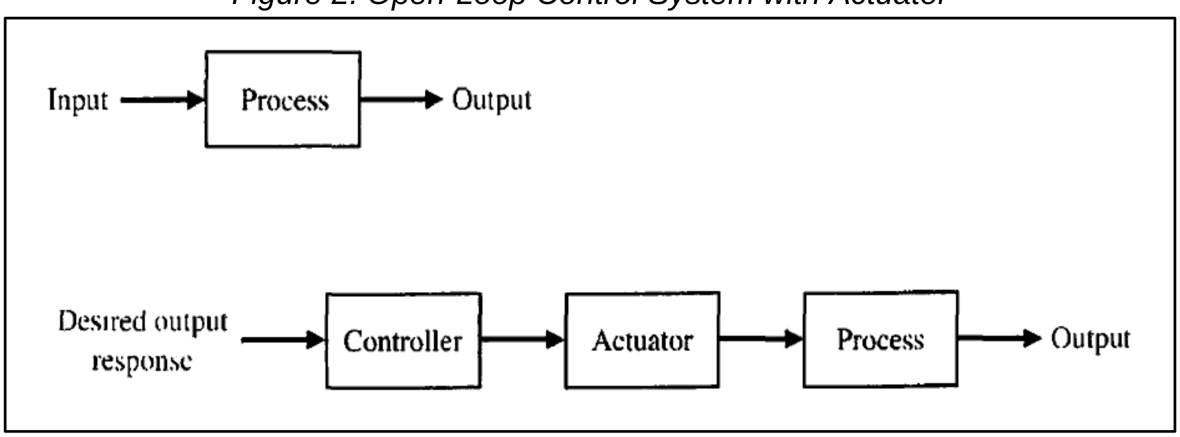


Source: (Dorf & Bishop, 2011)

2.1.1 Open-Loop Control System

An open-loop system is not necessarily for an automatic system since it lacks the capacity to adapt to a variety of operating conditions or to respond satisfactorily. This is principally since it does not have a feature of feeding the output signal to be compared with the command signal. The characteristic of an open-loop control system is that it utilizes an actuator to directly control the process of obtaining the desired system response without the use of feedback information. The authors Dorf & Bishop (2011) defined an Open-Loop System as “A system without feedback that directly generates the output in response to an input signal” (p. 303). In an open-loop system, the operation of the processes obtains the result independently of the desired result since it does not use feedback information to correct its actions. It is worthy to note that most open-loop automatic control systems are usually controlled by time. Examples of equipment that utilize the open-loop system functions are microwaves and washing machines, where the results are not necessarily the desired results.

Figure 2: Open-Loop Control System with Actuator

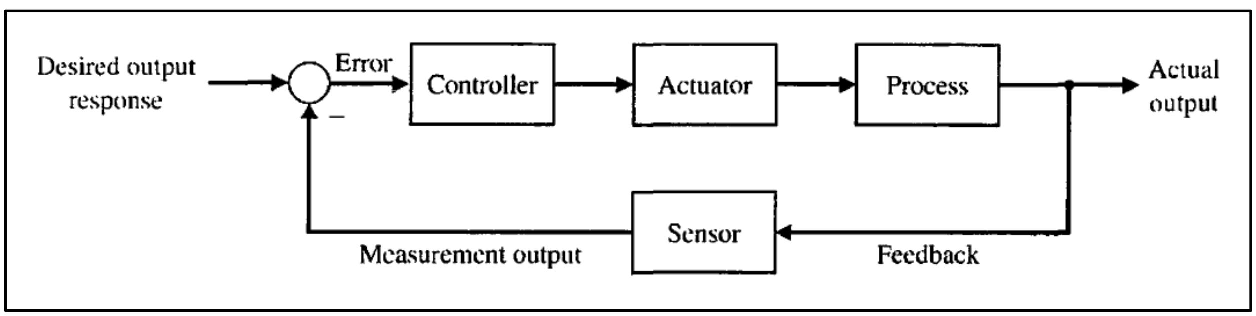


Source: (Dorf & Bishop, 2011)

2.1.2 Closed-Loop Control System

Distinctly, a closed-loop automated control system is different from an open-loop system, in that the closed-loop systems use a signal from its output response as a feedback and compare this with the reference or desired result. The definition of a "Closed-loop system: A system with a measurement of the output signal and a comparison with the desired output to generate an error signal that is applied to the actuator" (Dorf & Bishop, 2011, p. 303). A known factor is the desired result and since we know that there is a difference between the desired and the output response, it is possible to generate an error signal. The controller adjusts or corrects the actuator by using the amplification of this relative amplified error signal, in attempting to reduce the error signal to achieve the desired system response.

Figure 3: Closed-Loop Control System

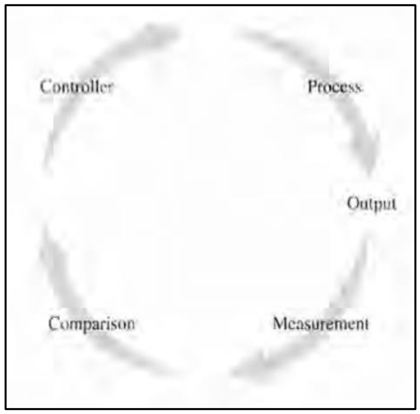


Source: (Dorf & Bishop, 2011)

The method of introducing the output response signal back into the system and then utilizing its error signal, after comparing it to the desired response, to control or correct the process in order to reduce the error signal is the feedback system. A closed-loop control system may be of a single feedback system or of multi-loop feedback systems.

Normally, sensors or transducers are utilized along the processes to detect the measurements of the outputs response and to send this signal to the controller.

Figure 4: Closed-Loop concept



Source: (Dorf & Bishop, 2011)

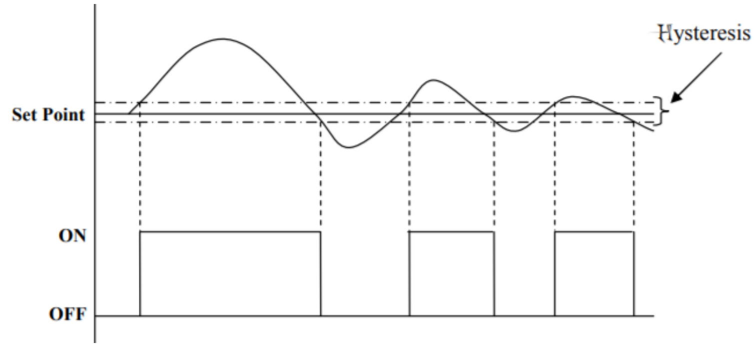
The major advantage of a closed-loop control system over an open-loop system is the capacity to offer stability, gain, and sensitivity over controls of the output system response even if there are external noise and disturbance signals, yet it achieves the desired system response even for a complex system. Open-loop system is mainly utilized in the design of a simple control system because of its deficient in accuracy and versatility.

2.2 ON / OFF TEMPERATURE CONTROLLER

The temperature controller is a system that has the ability to control temperature without much human intervention with the use of a controller and temperature sensors to compare the desired system response with the actual output temperature. There exist various types of controllers such as the ON/OFF, Proportional, Integral, Derivative, and the combinations between them; however, for the purpose of this project, only the ON/OFF temperature controller is described.

The ON/OFF temperature controller is one of the most basic temperature regulators, once the controller senses if it has exceeded a reference point it does one action. This controller has two states, ON or OFF; it does not have an intermediate state. Once the temperature is above the reference point, the controller turns off the heat and when it is below the reference point, it turns on the heat. This produces a change of state every time the reference point is crossed and therefore causes that the temperature oscillates continually between the peak-to-peak states, this variation of the peak-to-peak cycle depends on the desired system responses. Sometimes this peak-to-peak cycle is oscillating quickly and may damage relays and contactors, to reduce the possibility of damage the peak-to-peak cycle may be increased. Hysteresis is another name for the peak-to-peak cycle; the overshoot goes over the reference point and the undershoot goes below the reference point. The ON/OFF temperature controller is generally used when the precision control of temperature is not necessary and when the mass of the system is substantial sot the temperature change is slow.

Figure 5: ON / OFF Heating Temperature Controller



Source: *(Electrical4u, 2017)*

2.3 TRANSDUCERS AND SENSORS

In designing a project, it is of utmost importance to choose adequate transducers and sensors. Since the transducer is the principal means of communicating with the controller's environment. A transducer is a device that converts a physical variable to an electrical current or voltage signal. The types of variables are force, pressure, temperature, velocity or various others. There are some factors to consider when investigating the type of transducer or sensor is necessary, such as analog or digital. An analog sensor produces a continuous range of value as the output signals. A digital sensor produces discrete values as the output signal. Digital sensors are more compatible with computers or digital based controllers. Another factor to consider is the current or voltage output signal requirements. Other factors to investigate also are sensitivity, precision, range, speed, connector's types and calibration requirements.

2.3.1 Temperature Sensor

In the atomization industry, the various types of temperature measuring sensors are known as thermoelectric transducers. One of them is the Resistance Temperature Detector (RTD) sensor, these are normally made of a specific type of wire mounted in an insulation, and a temperature change affects a variation of the resistance. The resistance is the metal in the sensor is calculated with the following formula:

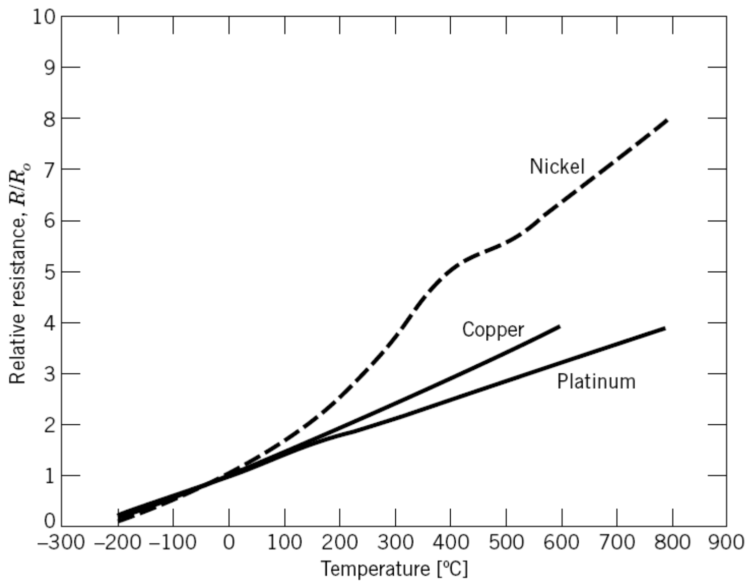
$$R = (l\ \rho_e) / A_c$$

Where: Resistivity = ρ_e; length = l; cross sectional area = A_c;

$$R = R_o \, [1 + k \, (T - T_o)]$$

Reference resistance = R_o; temperature coefficient = k; reference temperature = T_o

Figure 6: Relative Resistance of three pure metals



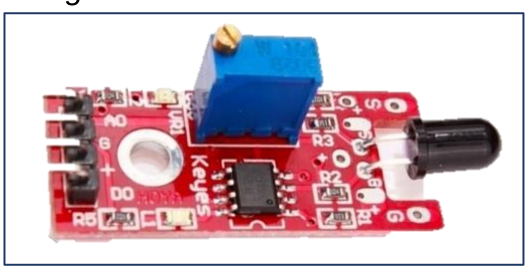
Source: (Figliola & Beasley, 2011, p. 318)

In the construction of RTD various materials are utilized however, "Platinum is the most common material chosen for the construction of RTDs. The principle of operation is quite simple: platinum exhibits a predictable and reproducible change in electrical resistance with temperature, which can be calibrated and interpolated to a high degree of accuracy" (Figliola & Beasley, 2011, p. 318). Two other pure metals are copper and nickel, which also produce predictable and reproducible resistances; they produce a positive temperature coefficient (PTC). As seen in figure 10 the relative resistance of three pure metals (R_O at zero ºC) resistance increases with the rise of the temperature. RTD sensors are mostly utilized in industrial applications since they are accurate and stable, the range of operation is between -150 to 650 ºC.

2.3.2 Photo diode Sensor

A photo diode sensor is a solid-state device that detects light. It is made of P-N semiconductor; its photoelectric effect has the ability to generate a current when photons are absorbed into the diode. A small leakage current I_d, known as dark current is produced even if there is no light. In photovoltaic mode, the diode’s photocurrent or photovoltage can be measured (Webster, 1999). Normally this photodiode is operated in reverse bias and has the same characteristics as a regular diode.

Figure 7: Arduino Flame sensor



Source: *(Arduino AG, 2017)*

The Arduino flame sensor module has the following features, it detects wave lengths between 760 nm to 1100 nm infrared, and it has a 60-degree angle of detection. It has two outputs mode, the first is analog output (AO), real-time output voltage signal on the thermal resistance; the second is digital output (DO), when the temperature reaches a certain threshold, the output high and low signal threshold adjustable via potentiometer; the working voltage is DC 3.0 to 5.5 V.

2.4 ACTUATOR VALVE AND RELAY

2.4.1 Actuator Valve

An actuator is primarily a mechanical device with the ability to generate a mechanical movement in an external device by activation of hydraulic, pneumatic or electrical signals. Actuator generates two main types of movements, linear or rotary. The linear actuator generates linear movement and the rotary actuator generates revolving movement such as a motor, pump or valve. The actuator normally receives instructions from controllers and regulators, example such as the Arduino. A regular valve is a mechanical gadget, which controls the flow of a fluid, gas or liquid within a pipe. Regular valves are operated manually and they have three basic modes of operations: open, closed or regulate a control flow of the fluid, gas or liquid. An electrical actuator valve has an electrical solenoid, which is managed by an electrical signal. Once the solenoid is activated by the electrical control signal, it opens the valve therefore allowing the flow of the fluid, gas or liquid.

2.4.2 Solid State Relays

Relays are key components in the design and development of an automatic control system. A relay is normally an electromagnetic device that functions as a switch. Its main functions of a relay are to control or protect an electrical circuit. Primarily, it can be controlled from a lower voltage to manage a higher output voltage. Relays are utilized in electrical automation, control of industrial motors, in electronics circuits and as an interface for computers, alarms and amplifiers.

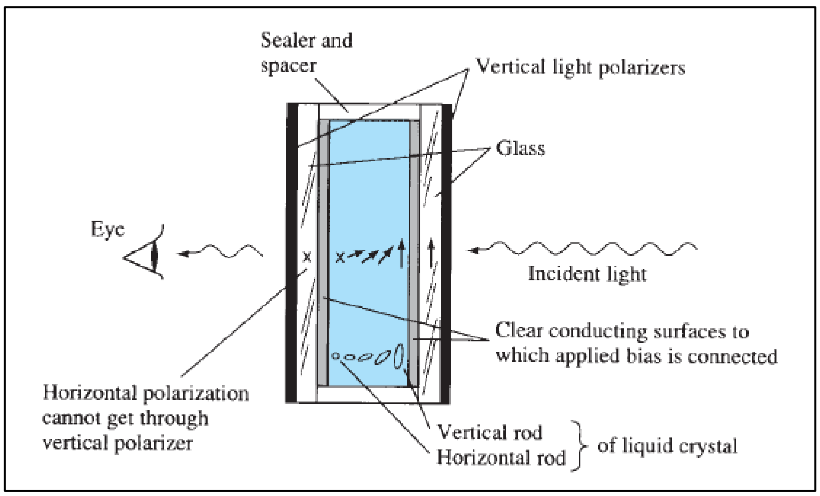
Due to the current limitations of the Arduino, a relay module is utilized as a solution between high current circuits and the Arduino. This relay module has eight relays to control external loads. It has a normally open interface (normally open) and normally closed NC. The control circuit operates between 4.75 ~ 5.25 VDC and between 8~250 mA. The switching circuit operates between 5 A / 7 A and 250 VAC / 30 VDC with total power of 70 W and up to a frequency of 1 Hz. A feature is an opto-coupler isolation that can be utilize as a protection, to isolate the input voltage from the coil control voltage, therefore it protects the Arduino in case of a spike or short. The general characteristics are that it has an active LED indicator, independent contacts and small.

Figure 8: Relay module

2.5 LCD DISPLAY

LCD (Liquid Crystal Display) is a display manufacture utilizing a liquid type substance, which is trap between two plates of crystal. When electric current passed within a certain area or zone the molecule within the liquid re-adjust, resulting in a change of the refraction indices, which then affects the opaqueness of the liquid. A backlight is normally used, and this sectional blocking of the backlight is how we see pixels, words, etc. The color LCD displays is designed so that each pixel is divided into three sub-pixel of red, green and blue (RGB), the mixture of these three colors produce a variety of palettes.

Figure 9: LCD Display



Source: (Boylestad & Nashelsky, 2013, p. 830)

Depending on the manufacture of the display, communication between the LCD display and the microprocessor may utilize any of the various protocols such as Parallel, Serial I^2C and Serial UART.

2.5.1 Nextion Touch LCD Display

Nextion NX3224K024 Touch 2.4” LCD Display is a smart type of display that is comprised of two components; a hardware section that includes a TFT (thin-film-transistor) boards and a software part (the Nextion editor). This LCD TFT board uses only one serial UART port to communicate. “Nextion is a seamless Human Machine Interface (HMI) solution that provides a control and visualization interface between a human and a process, machine, application or appliance. Nextion is mainly applied to Internet of thing (IoT) or consumer electronics field” (ITEAD Intelligent Systems Co.Ltd, 2017). The Nextion editor has features such as buttons, text, progress bar, slider, instrument panel, which assist in the interface designs. Other features include 320 x 240 resolution, true to life colors RGB 65K, 4M flash memory, on board micro-SD card for firmware upgrade, adjustable brightness.

Figure 10: Nextion Touch 2.4” LCD display

Source: (ITEAD Intelligent Systems Co.Ltd, 2017)

2.6 ARDUINO

The Arduino is an open-source microcontroller unit design with two integrated elements, the hardware and the software. The Arduino is a microcontroller with brain, it has the hardware element such as the I/O access pins, and these pins are configured and programed in the software, an open-source Integrated Development Environment (IDE) platform. Program codes are developed to manage the I/O pins to control devices such as sensors and actuators. “The process of interaction works: If you do something, you feel a change, and from that you can know something about the world” (Nussey, 2013, p. 8). This control of sensors and actuators allows the Arduino the facility to interface and interact with its environment. Sketch is the program files, which are the codes written and developed in the Arduino IDE to perform different functions. Sketch program files are normally saved with (.no) extensions or an anterior one (.pde).

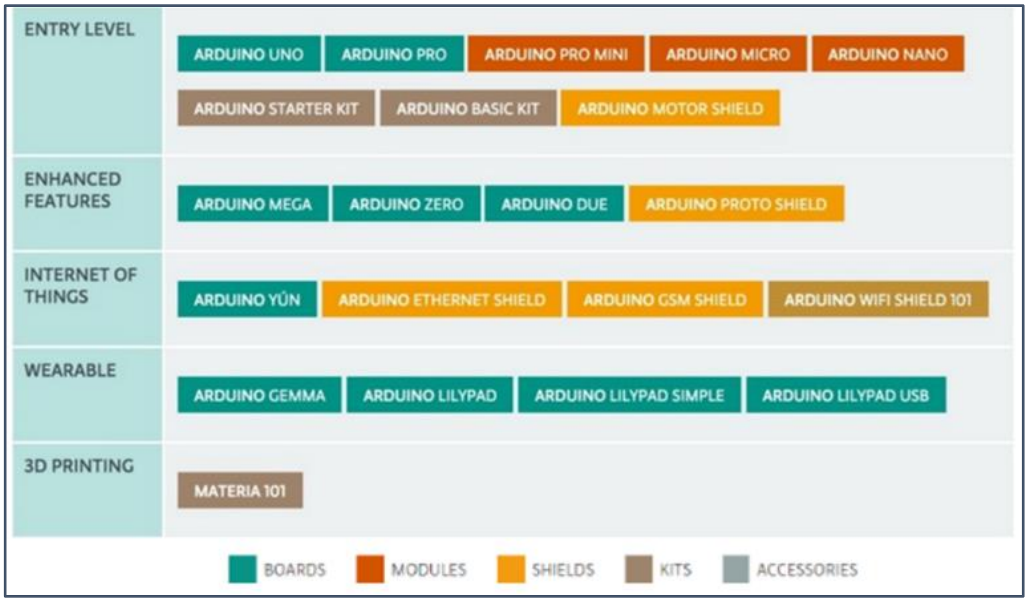
As mentioned, the software component of the Arduino’s configuration is developed and coded in the open-source license IDE platform, which is available in two different packages. The first is the cloud-based design IDE known as *Arduino Web Editor*. Access is only available via internet, on any web browser (*Chrome*, *Firefox*, *Safari* or *Edge)*. One of its functions includes the ability to save and retrieve Sketch files from the Cloud using any computer with access to internet. The other IDE is an application known as *Arduino Software,* its design with the function to install on a computer with any OS such as Windows, Mac or GNU/Linux. By utilizing the *Arduino Software,* its programming application and the Sketch files can be accessed and utilized without an internet access.

In the Arduino platform, there are files of collection of programming codes and instructions known as libraries. These libraries are programming files with specific information and functions with the capacity to manage different sensors, actuators and displays, etc. The IDE has various standard incorporated libraries to activate and use functions and devices such as LAN, Wi-fi, GSM, SD memory, Serial port, SPI, mouse, keyboards, etc. In addition, many libraries are available for download. However, personal unique library files may be designed and developed with specialize functions. An assemble programming language known as AVR is a handy and fast way to design, program and develop new library files for Arduino. However, attention has to be considered when manipulating and organizing the memory structure of the Arduino. To minimize complications, it is best not to use AVR unless you have the expertise, it is better to choose a higher-level programing language such as C, Basic or Java.

The Arduino has I/O pins that provide different functions, such as analog input, communication pins like Rx/T x, UART, SPI and I2C; and digital I/O pins. The digital pins may be configured as an input or output, in two states: high (five V) or low (zero V). As mentioned, the Arduino has analog input pins which function is to measure or accept an electrical range between zero V to five V. This range includes any value such as 0.1, 0.2, 0.5, 3.4, 4.5, up to 5 V. The Arduino does not have any analog outputs, if required, certain digital pins may be configured as Pulse Width Modulation (PWM) output.

Depending on the model of the Arduino, there exist various sizes of memory and they operate between 3.3 V or 5.0 V DC. As power source, the Arduino utilizes its USB connection or utilizes an external power supply. Advances and projects are constantly being developed for Arduino, many of such projects and programming codes are available at https://github.com/arduino/Arduino/. Depending on the operation and the necessity of the project, there are various Arduino models to choose from, see the next figure.

Figure 11: Arduino Models



Source: (Arduino AG, 2017)

Within the ranges of Arduino's products, there exist the *Shield* modules. These *Shield* modules are utilized to give additional functions and capacities to the different Arduinos. There is a wide variety of *Shield* modules with variety of functions such as Wi-Fi, proto, motor controllers, Bluetooth, GSM, relays, and some more.

2.6.1 Arduino Mega

The Arduino Mega 2560 is a microcontroller unit, which is easily utilized in projects. The Arduino webpage states, The Arduino Mega 2560 is a microcontroller board based on the AT mega 2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, four UARTs (Universal Asynchronous Receiver / Transmitter), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. (Arduino AG, 2017).

As mentioned, one of its features is the UART communication hardware serial port; it has four built-in ports label as Tx0/Rx0 to Tx3/Rx3 and they are located on pins 0/1/18/19/16/17/14/15. Note: the USB is hardwire to and shares the same port as the Tx0/Rx0 pins 0/1. The USB port is protected with an automatic poly-fuse against short circuits. Communications also include an I2C data bus (SDA/SCL) pins 20/21. Serial Peripheral Interface (SPI) is located on the ICSP header pins or digital pins 50/51/52/53 may be configured as SPI.

Figure 12: Arduino Mega 2560

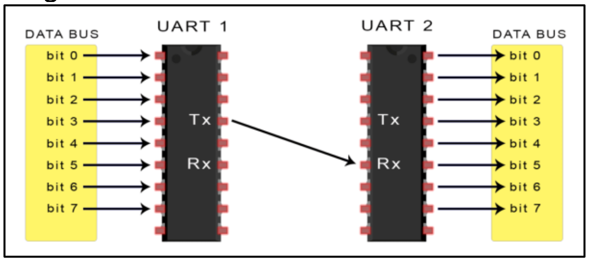
Source: *(Arduino AG, 2017)*

The Mega has six external interrupts located on the digital pins 2/3/18/19/20/21, these are utilized to trigger an external interrupt on low level, rising or falling edges or a change in state. Each I/O pin normally operates at 20 mA (max 40 mA). The analog input pins data at 10-bit resolution (1024 different values). The digital pins 2-13/44-46, in PWM mode operates at an 8-bit resolution. The Mega may get its power source from the USB connection, but for more current, it is better to use an external power supply within the range 7 - 12 V DC. The Arduino Mega is configured and programed utilizing Sketch files developed in the IDE software Arduino Web Editor (Cloud) or Arduino Software (OS) platforms.

2.6.2 UART

Universal Asynchronous Receiver Transmitter (UART) is not a communication protocol, but a small efficient communication IC that principally uses the transmitting (TX) port and the receiving (RX) port. The UART IC is responsible for the interfacing between the microprocessor parallel data buses, and converting the parallel data to the serial communications link port and vice-versa.

Figure 13: Serial UART communication



Source: *(Park & Mackay, 2003)*

The UART generates the start, stop, and parity bits used for checking data integrity in asynchronous transmission between the serial ports. During transmissions, the UART chip performs various functions, such as:

i. setting the correct baud rate;
ii. interfacing to the microprocessor data bus;
iii. accepts/sends characters one at a time;
iv. generates a start bit for each character;
v. adds/receives the data bits in a serial stream;
vi. calculates and adds the parity bit to the data stream;
vii. adds/reads the required stop bit(s).

Benefits of utilizing the UART are:

i. It uses two wires,
ii. It has no clock signal,
iii. It has a parity bit, (error checking) the flexible data packet structure once configured.

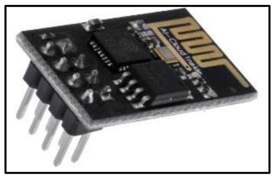
Drawbacks are:

i. The data frame size is limited to a max of 9 bits,
ii. The baud rates on both ends must be within +/-10% of each other.

2.6. Wi-Fi Shield Module

The ESP8266 module is a Wi-Fi network solution that permits the Arduino to join and communicate on a network. This Wi-Fi module connects to the Arduino via UART (RX/TX) serial interface. This module is able to establish communication to any access point devices using the 802.11 b/g/n and Wi-Fi Direct (P2P) protocols. As power source, it uses a 3.3 VDC voltage supply.

Figure 14: Wi-Fi ESP8266 module



Source: *(ITEAD CC)*

CHAPTER III: METHODOLOGY FRAMEWORK

3.1 TYPE OF INVESTIGATIONS

3.1.1 Purpose: Applied

The purpose of this investigation is defined as applied, since it involves the development and implementation of a project. The project is to design, develop, test and implement an automatic system with Arduino microcontrollers in order to manage and control variables such as duration time and temperature of an existing manual oven in Caribbean Delight micro business.

3.1.2 Term Dimension: Transversal

This project term dimension is a transverse type, since it is a non-experimental project, and which the design, development, collection of data, verification and the implementation of the automatic control system is to be completed by a definite time. The completion time for the implementation of this project is by the end of October 2017.

3.1.3 Framework: Micro

The framework of this project is defined as a micro type, since the problem and solution is very specific for the implementation of a control system baking solution on an specific equipment for a micro business with 2 locations and less than 20 employees.

3.1.4 Nature: Quantitative

The nature of this project is a quantitative one, since it engages a microprocessor to control and calculate the necessary variables of the measurements of some sensors in order to achieve the objective of the project. Furthermore, the expenses of the project are tracked in order to analyze and evaluate in reference to procuring a new automatic oven. This is used to verify if a self-design automatic system is economically feasible.

3.1.5 Character: Project

The character of the investigation is defined as a Project since it has clearly stated a general objective and specific ones, to solve and implement how to mitigate losses and increase efficiency of the oven by using a microcontroller. The objectives are implemented within a specific planned period which is by the end of October 2017. This project is to satisfy an economical and technological requirement for future business opportunities for Caribbean Delight microenterprise located in San Juan the Tibás, San Jose, Costa Rica.

3.2 PROJECT MANAGEMENT FRAMEWORK

The project is defined as a unique group of tasks that is performed to create a distinctive product, service or improvement to meet the requirements of a necessity or to take advantage of an opportunity. The project has clearly defined objectives and it is planned for a period. According to the PMBOK® guide (2013) .

A project is a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates that a project has a definite beginning and end. The end is reached when the project's objectives have been achieved (p. 3). A project reaches its end when the project's objectives cannot be met for various reasons or if there is no longer a need.

This project uses the Project Management Body of Management (PMBOK®) guide as the base to develop and guide the different processes which are categorized in five Process Groups which are Initiating, Planning, Executing, Monitoring Controlling, and Closing. The following is a brief description of each Process Group.

- Initiating Process Group. Those processes performed to define a new project or a new phase of an existing project by obtaining authorization to start the project or phase.

- Planning Process Group. Those processes required to establish the scope of the project, refine the objectives, and define the course of action required to attain the objectives that the project was undertaken to achieve.
- Executing Process Group. Those processes performed to complete the work defined in the project management plan to satisfy the project specifications.
- Monitoring and Controlling Process Group. Those processes required to track, review, and regulate the progress and performance of the project; identify any areas in which changes to the plan are required; and initiate the corresponding changes.
- Closing Process Group. Those processes performed to finalize all activities across all Process Groups to formally close the project or phase.

(PMBOK® Guide—5th Ed. Content Committee, 2013, p. 49)

3.3 PROJECT IMPACT FRAMEWORK

3.3.1 Short Term Impact

The short-term impact is seen within the first week after the project is implemented. This impact has two benefits, one in terms of qualitative and quantitative terms. Almost immediately, the workers and employees of the company have positive benefits in terms of qualitative since it is not necessary to preheat the oven manually, or monitoring of the product during baking. The automatic system controls the temperature, humidity, ventilation and the duration variables during baking process. Also in the short term, the company is going to have benefits in term of quantitative because they are not investing a larger capital to buy new automatic oven since they are investing in a local design and solution with implementation of this project.

3.3.2 Medium Term Impact

In reference to the qualitative terms, the first three months medium-term impact is the continuance of positive benefits for the operators and for the business. An operator's benefit is the reduction of monitoring the baking process of their products. This helps to achieve an improvement in their production efficiency and a reduction of products losses. In terms of quantitative, the medium-term impact of the efficiency of the oven's control system has positive financial benefits for the business in the reduction of the usage loss of the butane gas and of product's losses.

3.3.3 Long Term Impact

The one-year long-term impact of this project in reference to qualitative terms continues with positive benefits for ovens' operators. This is to be experienced especially when the company's output is increased if exporting its products abroad. Although production has increased, the extra baking process has not affect the operators because the automatic system continues to manage and control the variables, even if the baking cycle has increased. This long-term impact is going to produce benefits in the company in quantitative terms since it has the ability to increase its production. The company is more profitable since it is more competent in the baking processes. The reduction of gas wastage and product losses is kept at a minimum even though production has increased.

CHAPTER IV: DIAGNOSTIC

4.1 ACTUAL SITUATION DESCRIPTION

The project requirements of the business are determined by meeting with the owner. The outcome of the meeting produces the following requirements. The production department has a baking procedure, which includes the usage of a manual oven. The operation of the oven is manual and it is fully dependent on human intervention to function. This dependency on human intervention may create losses due to inconsistency of the baking process, and as a result, some of the final baked products may not be within standards.

After various visits to the production location, noting some observations and conversing with the main operator, the following is determined in the oven's preheating process and the baking process. Initially, the operator opens the main gas valve and then stoops under the oven to light the pilot fire. The oven's regulated gas valve is turned on and set to an approximately temperature setting, the burner ignites. The oven's regulated gas valve has an integrated thermostat, which turns off the gas supply if the oven heats up over 400º F as marked on the regulated valve. The operator normally allows approximately ten minutes for the preheating of the oven. However, since the oven is not equipped with a thermometer or a temperature-measuring device, there is no verification of the actual baking temperature. The products allocated for baking are pre-placed on trays. Once the oven is considered preheated, the trays are placed into the oven and at that time the baking process starts. The oven has a timer, which is set to the desire baking time in minutes. Both, the sounding of an alarm by the timer and the periodic monitoring by the operator are utilized to check if the product is properly baked.

Each product has a different baking procedure. The procedures include different variables within the process such as, specific baking time with the fan on or off, baking with humidly (water), baking at a specific temperature range. The following table 1 shows the required modes and variables required for each product during baking.

Table 1: Individual Products modes and variable requirement for baking

Products	Temperature F/C	Baking time fan-On (mins)	Baking time, fan-Off (mins)	Humidity moisture
Panbon	300/149	45	75	Yes
Patty	320/160	35	0	no
Tart	320/160	35	0	no
Galleta	320/160	25	0	no

Source: Caribbean Delight information

There were various design flaws and imperfections noted. The first is that once the main gas valve is turned on, most of the time the pilot remains unlit. The design of the pilot's supply is such that it is located before the oven regulated valve and therefore once the main gas valve is turned on, gas continually leaks into the general area since it is unlit. There is no indication or a written procedure to alert of the danger of the leaking gas from the unlit pilot. Another imperfection is that the oven has no installed sensor or temperature gauge to indicate or show if the temperature is below the baking level. Likewise, there is no sensor or alarm to indicate if the door is not properly closed. An additional imperfection noted is the lighting system within the oven, there were no bulbs in the light sockets since the bulbs always vibrate out, fall and break, and this is a health risk. The light sockets currently installed are not industrial type sockets. The next defect is airflow, since the fan blade is not adequately designed, it is a homemade design made of pieces of 90-degree angle iron. Additionally, the fan motor seems oversize, hence inefficiency of electrical consumption. The manual oven built imperfections may expose the business’ personnel and clients to health risks and personal safety danger.

Since all the processes are manually implemented, there are many possibilities for human error. The baking is scheduled to mostly once or twice a day, and only two or three times per week. The effect of the operational inefficiencies is not immediate, but eventually it may cause an economic decline for the company as production increases.

4.2 DATA COLLECTION, PROTOTYPE AND CHARACTERISTICS

4.2.1 Interview

The first interview of this study is to discuss the baking process and to obtain a better understanding of the operations of the oven. The purpose is to comprehend how the baking procedure fits into the overall operation process of the production section. Equally discussed is the likelihood of limitations of the existing manual oven. The operator mentioned the extra observation time needed since they have to check the products regularly. In his observation, he noted that the baking time depended on the amount of trays in the oven. This may be caused by an inadequate design flaw of the fan blades; therefore, airflow might not be consistent. Moreover, the oven overall design has one heat source (below). Part of the baking procedure is that the product, Panbon needs more moisture during baking than the other products. Note the normal operating baking temperature range is 149º to 160º C.

A second site visit at the production section is performed as part of the actual situation process. The purpose is to check the oven design, to have a better understanding of the component requirements for the automatic control and to draw the existing circuits and take pictures. The following circuits and components were checked: the pilot system and gas flow system, the fan system including its electrical circuit, the existing timer and buzzer, the lighting system and the existing safety system. This interview with the main operator mostly entailed discussions about the preheating procedure and the baking processes.

4.2.2 Observation

During the visits, the following items and issues are documented; the implementation of this project does not guarantee that all of them are to be addressed:

- No water line is installed to the oven
- No gas shutoff valve is installed on the oven
- Once the pilot is unlit, butane gas is released into the working area
- It does not have a gauge or sensor to measure temperature
- Unable to check the air flow and the motor of the fan system

  Regular breakage of light bulbs due to vibration, the socket is inadequate

CHAPTER V: PROJECT DESIGN AND DEVELOPMENT

5.1 DESIGN

As part of the initial design stage, two principal types of automatic control solutions are considered. One of them is the PLC type solution and the other is the microprocessor type solution. Overall, two different brands and three types of PLC solutions are researched and the material costs are compiled and listed in Table 2. Additionally, two microprocessor solutions are considered and researched; their material costs are listed in Table 3. The range between the lowest and highest material cost is $220.33 to $1158.90 US dollars which is approximately an 80.9% difference .The inexpensive PLC is mostly modular and additional modules are necessary to meet the project necessities, therefore raising the cost of materials. However, the microprocessors meet most of the necessary requirements.

Table 2: PLC Solutions, material cost

Kinco K5 low cost PLC Solution			
Parts Description	**Unit**	**Amount**	**Subtotal**
CPU(K504EX-14AR)	1	$161.50	$161.50
PM531/K531-04RD	1	$156.00	$156.00
SM541 / K541	1	$122.24	$122.24
USR232A	1	$ 8.57	$ 8.57
capacitors 100nF	5	$ 0.20	$ 1.00
Igniter	1	$ 12.50	$ 12.50
IR Flame sensor	1	$ 2.49	$ 2.49
Touch 2.4" display	1	$ 24.99	$ 24.99
Gas valve actuator	1	$ 38.00	$ 38.00
RTD sensor	1	$ 9.99	$ 9.99
Fan contactor	1	$ 8.49	$ 8.49
LEDs / Resistors	5	$ 0.50	$ 2.50
micro-switch	1	$ 1.76	$ 1.76
Water valve	1	$ 4.69	$ 4.69
Shipping	1	$ 60.00	$ 60.00
Unforeseen 10%	1	$ 65.00	$ 65.00
Price US Dollars	**Parts total**		**$679.72**

Kinco High performance PLC Solution			
Parts Description	**Unit**	**Amount**	**Subtotal**
PLCF1/F122D1608T	1	$415.00	$415.00
(RP2A-0402C1)	1	$215.15	$215.15
USR232A	1	$ 8.57	$ 8.57
Igniter	1	$ 12.50	$ 12.50
capacitors 100nF	5	$ 0.20	$ 1.00
Relay 8 port	1	$ 8.98	$ 8.98
IR Flame sensor	1	$ 2.49	$ 2.49
Touch 2.4" display	1	$ 24.99	$ 24.99
Gas valve	1	$ 38.00	$ 38.00
RTD sensor	1	$ 9.99	$ 9.99
Fan contactor	1	$ 8.49	$ 8.49
LEDs/resistor	5	$ 0.50	$ 2.50
micro-switch	1	$ 1.76	$ 1.76
Water valve	1	$ 4.69	$ 4.69
Shipping	1	$ 60.00	$ 60.00
Unforeseen 10%	1	$ 84.00	$ 84.00
		Parts total	**$898.11**

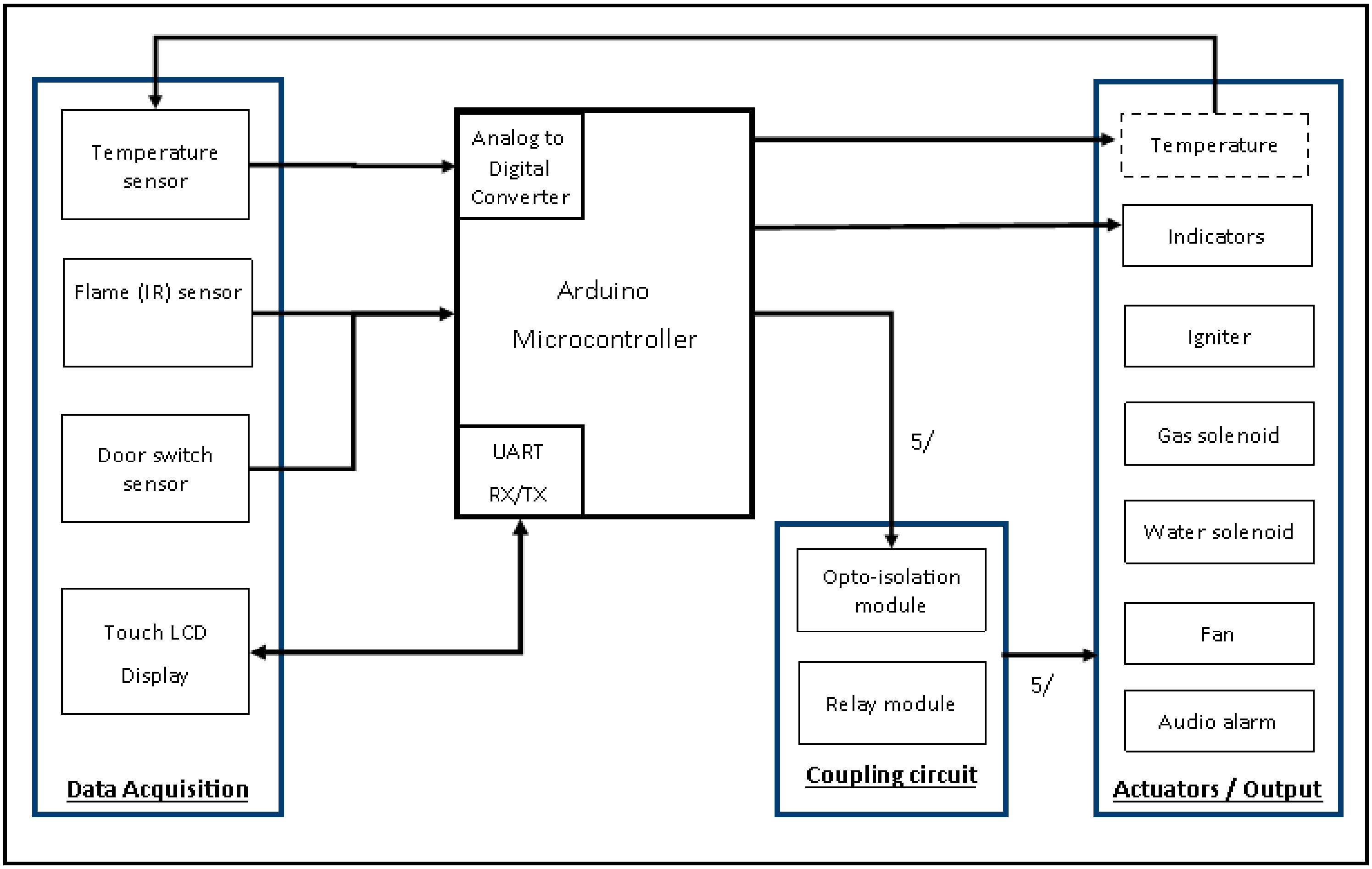
EATON Compact PLC Solution			
Parts Descriptior	**Unit**	**Amount**	**Subtotal**
EC4P222-MTXD1	1	$908.50	$ 908.50
Relay 8 port	1	$ 8.98	$ 8.98
Igniter	1	$ 12.50	$ 12.50
IR Flame sensor	1	$ 2.49	$ 2.49
Gas valve	1	$ 38.00	$ 38.00
RTD sensor	1	$ 9.99	$ 9.99
Fan contactor	1	$ 8.49	$ 8.49
LEDs	2	$ 1.00	$ 2.00
Resistors	3	$ 0.50	$ 1.50
micro-switch	1	$ 1.76	$ 1.76
Water valve	1	$ 4.69	$ 4.69
Shipping	1	$ 60.00	$ 60.00
Unforeseen 10%	1	$100.00	$ 100.00
		Parts total	**$1,158.90**

Table 3: Microprocessor Solutions, material cost

Arduino Microprocessor Solution				Raspberry Pi Microprocessor Solution			
Parts Description	**Unit**	**Amount**	**Subtotal**	**Parts Description**	**Unit**	**Amount**	**Subtotal**
Arduino Mega	1	$ 33.00	$ 33.00	Raspberry Pi3	1	$ 34.99	$ 34.99
Wi-Fi Shield	1	$ 7.95	$ 7.95	Igniter	1	$ 12.50	$ 12.50
Igniter	1	$ 12.50	$ 12.50	Raspberry case	1	$ 7.99	$ 7.99
Arduino case	1	$ 5.99	$ 5.99	Relay 8 port	1	$ 8.98	$ 8.98
Relay 8 port	1	$ 8.98	$ 8.98	IR Flame sensor	1	$ 2.49	$ 2.49
IR Flame sensor	1	$ 2.49	$ 2.49	Touch 2.4" display	1	$ 24.99	$ 24.99
Touch 2.4" display	1	$ 24.99	$ 24.99	Gas valve	1	$ 38.00	$ 38.00
Gas valve	1	$ 38.00	$ 38.00	RTD sensor	1	$ 9.99	$ 9.99
RTD sensor	1	$ 9.99	$ 9.99	A/D converter Servo	1	$ 29.95	$ 29.95
Fan contactor	1	$ 8.49	$ 8.49	Fan contactor	1	$ 8.49	$ 8.49
LEDs/resistor	5	$ 0.50	$ 2.50	LEDs/resistor	5	$ 0.50	$ 2.50
micro-switch	1	$ 1.76	$ 1.76	micro-switch	1	$ 1.76	$ 1.76
Water valve	1	$ 4.69	$ 4.69	Water valve	1	$ 4.69	$ 4.69
Shipping	1	$ 35.00	$ 35.00	Shipping	1	$ 35.00	$ 35.00
Unforeseen 10%	1	$ 24.00	$ 24.00	Unforeseen 10%	1	$ 25.00	$ 25.00
Price US Dollars		**Parts total**	**$220.33**			**Parts total**	**$247.32**

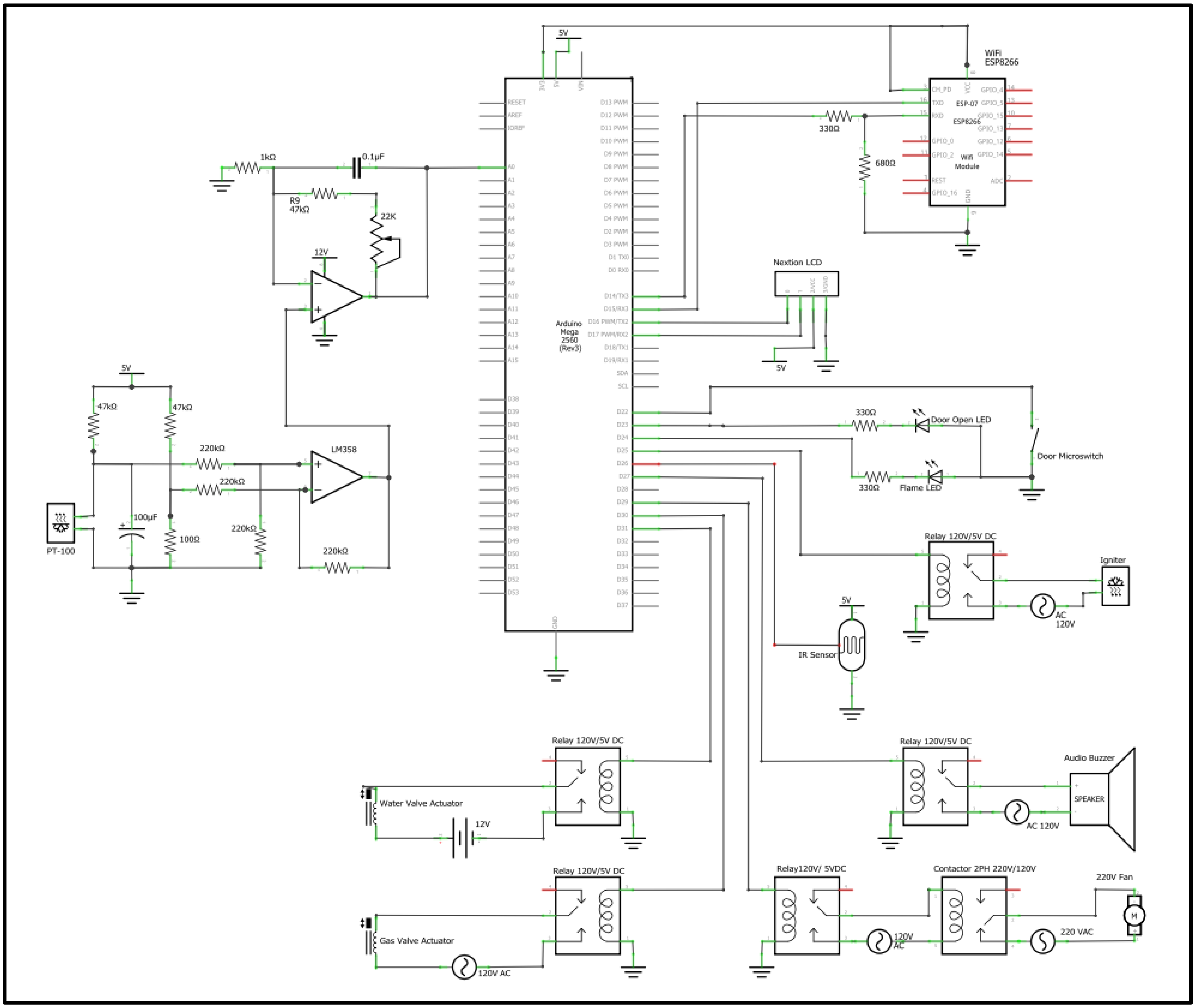
The microprocessors parts are lower in cost, which is the main factor in determining the chosen solution. Apart from the price, the Arduino microprocessor solution was decided on mainly because of its embedded system. This operates such that once it boots up the setup executes once, and immediately after, its starts to execute the loaded Sketch. In addition, it prioritizes all process loaded in its loop, hence monitoring of all input sensors and controlling all the outputs. As opposed to the Arduino, the Raspberry Pi does monitor its inputs, but it is not responsive fast enough since apart from its I/O, it also manages its operating system. Secondly, the Arduino has built-in analog inputs as opposed to the Raspberry Pi, which need an ADC for temperature analog sensor. The following Figure 15 is the block diagram of the oven automated control system utilizing the Arduino microprocessor.

Figure 15: Automatic Oven Controller Block Diagram



In the development of the overall design, one factor that is considered is the possibility to expand the system to control two or three ovens utilizing one microprocessor. Initially, the Arduino Uno was researched. However, the Arduino Mega has more features especially in considering future expansion and the price difference is $5.00 US dollars. The principal advantage for this project is that the Mega has its four UART ports in comparison with the Uno, which shares a single UART port with its USB port. For debugging and connecting additional displays, the four UART ports are useful. Another key advantage of the MEGA is the 54 digital ports in comparison to the Uno 14 digital ports.

Figure 16: Automatic Oven ControllerSchematic Diagram



Reference to the Figure 16, the PT-100 temperature sensor signal is conditioned with an LM358IC, which consists of dual operational amplifiers (OpAmp). In this design, one of the Op Amp is configured as a unity gain differential amplifier and the other Op Amp is configured as a variable gain amplifier circuit. This circuit measures variation of voltages of the sensor, which is triggered by temperature change, and delivers an output between zero V to five V. The output of the OpAmp supplies the analog input pin (A0) of the Arduino; the built-in 10-bit ADC then converts the temperature signal into digital values between 0 and 1024. Within the Sketch, these digital values are mapped to 8-bits values providing a temperature reading between 0 to 250 ºC.

A door switch (NO) is connected to ground and the pin (22) on Arduino. It is configured as an INPUT_PULLUP; hence, it utilizes a pull-up resistor within the Arduino. Pin (22) is HIGH when the door is open and vice-versa when the door is closed. A door indicator LED is connected via a resistor to pin (23). This pin is configured as an OUTPUT and the LED turns ON when the door is opened and turns OFF when the door is closed.

The igniter, audio alarm, fan, gas solenoid and water solenoid are all connected to the solid-state relay by means of the built-in opto-couplers feature of the relay. This is to protect and to isolate the Arduino's pins signal voltages from the relay's coil voltages. In the case of the fan relay, it connects via a double-pole 220 V contactor to the fan motor. The pins are configured as OUTPUT and are utilized as igniter- pin (25), audio alarm - pin (27), fan- pin (29), gas solenoid -pin (30) and water solenoid- pin (31).

The IR-sensor is connected to the Arduino digital Pin 26 as an INPUT. A flame indicator LED is connected via a resistor to Pin-24; this pin is configured as an OUTPUT. It is programmed in the Sketch such that the LED turns ON when flame is detected by the IR-sensor and turns OFF if no flame is detected.

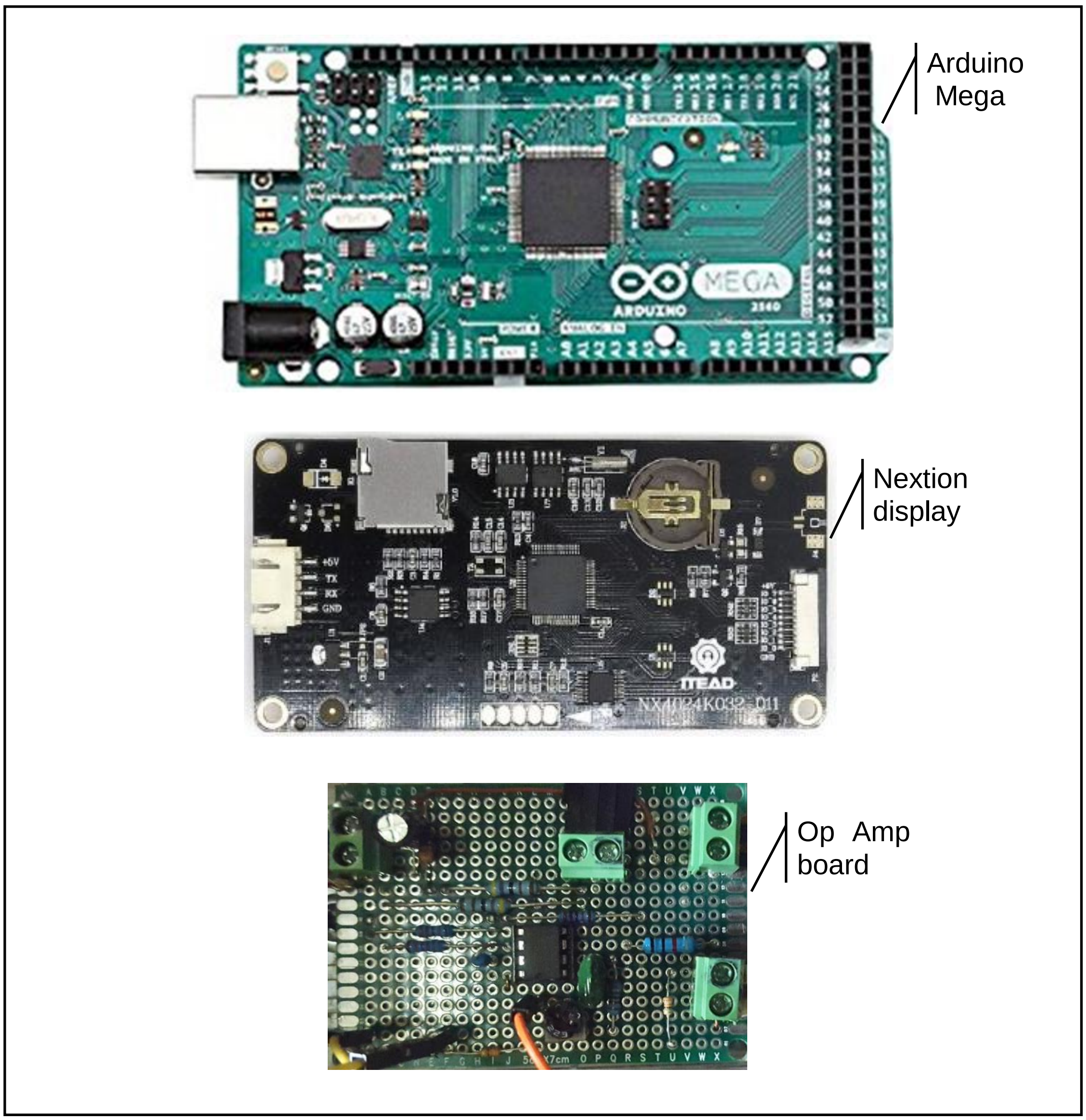
The Nextion display communicates via UART of the Arduino, it is setup as Serial2 on the Tx2/Rx2 Pins (16/17). The Arduino is programmed to transmit to the Nextion display any signal changes it detects from the temperature sensor, the door micro switch and the flame sensor. In return, the Nextion touch screen displays, transmits a signal to the Arduino once a button is pressed within it for a specific process. This activates an interrupt routine within the Arduino Sketch, which then activates the necessary process such as preheating or baking utilizing the sensors and actuators and therefore controlling the oven.

5.2 PROTOTYPE

5.2.1 Hardware

The hardware includes the following list of items, three input sensors, two actuator valves outputs, the Wi-fi module, a small 7 cm by 5 cm PCB board with the LM358 operational amplifier, the Arduino and the Nextion display, see the Figure below.

Figure 17: Various hardware circuit boards



5.2.2 Software

Figure 18: Project software flow chart

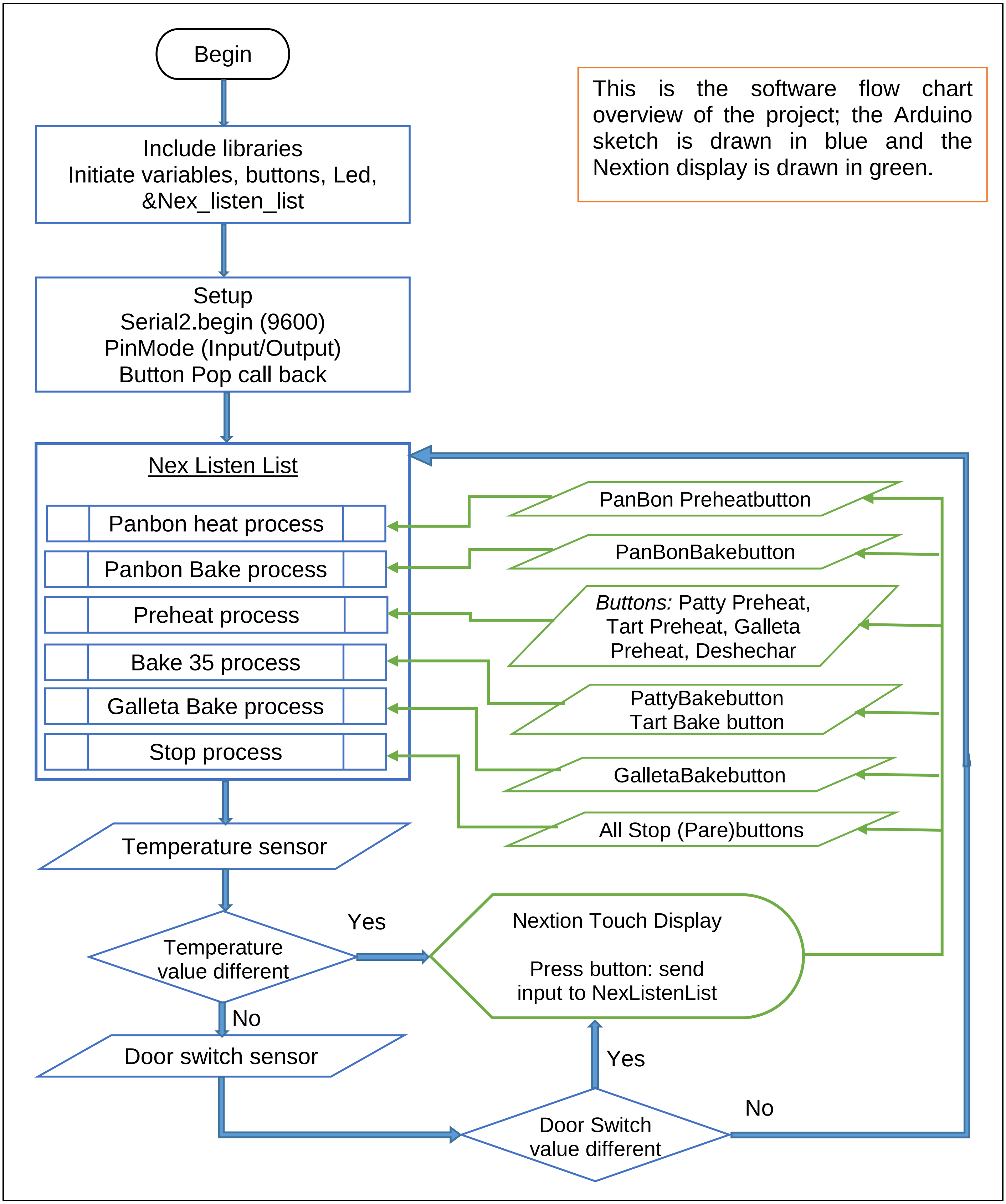


Figure 19: Preheat flow process

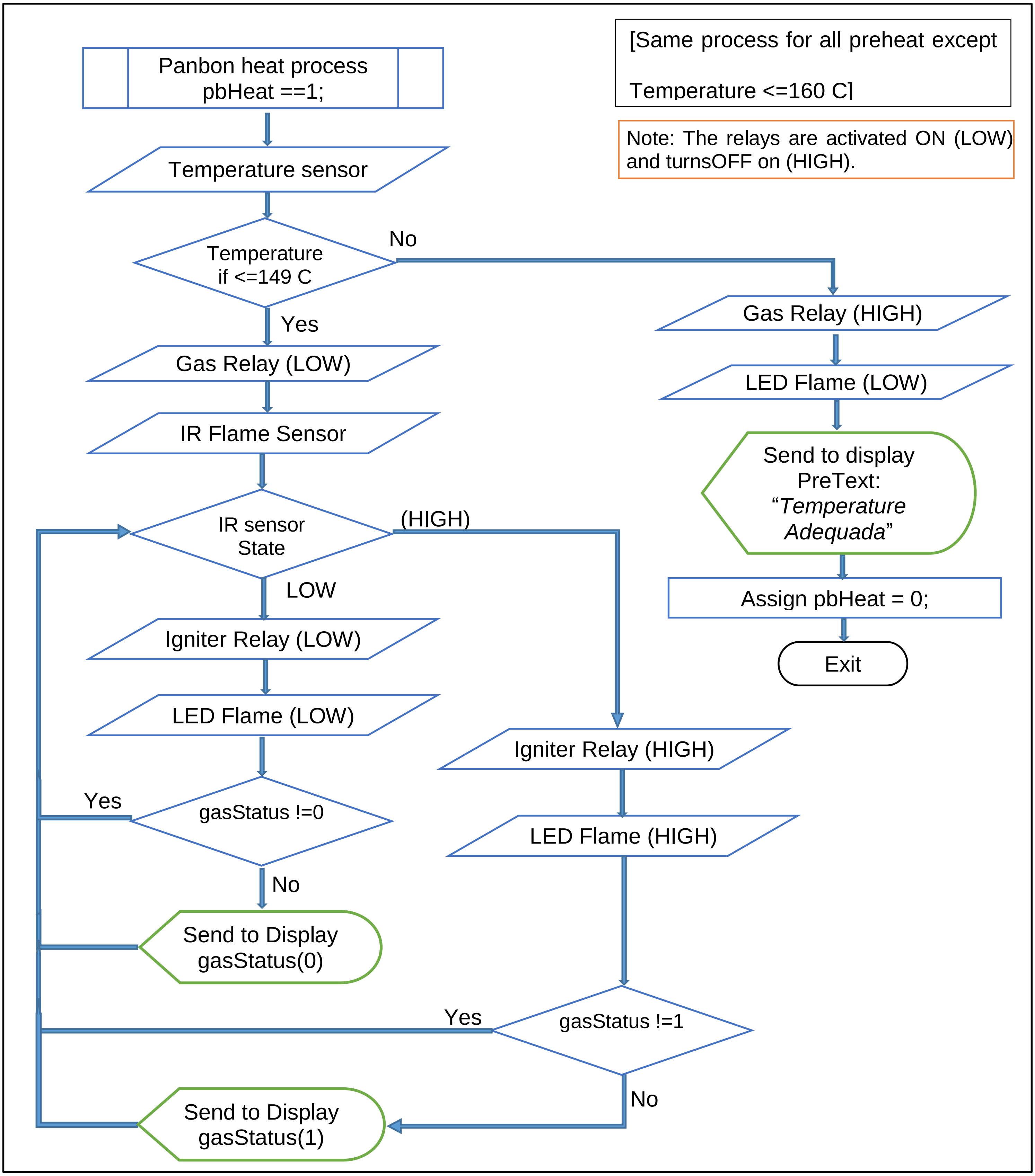


Figure 20: Section of the Arduino Sketch

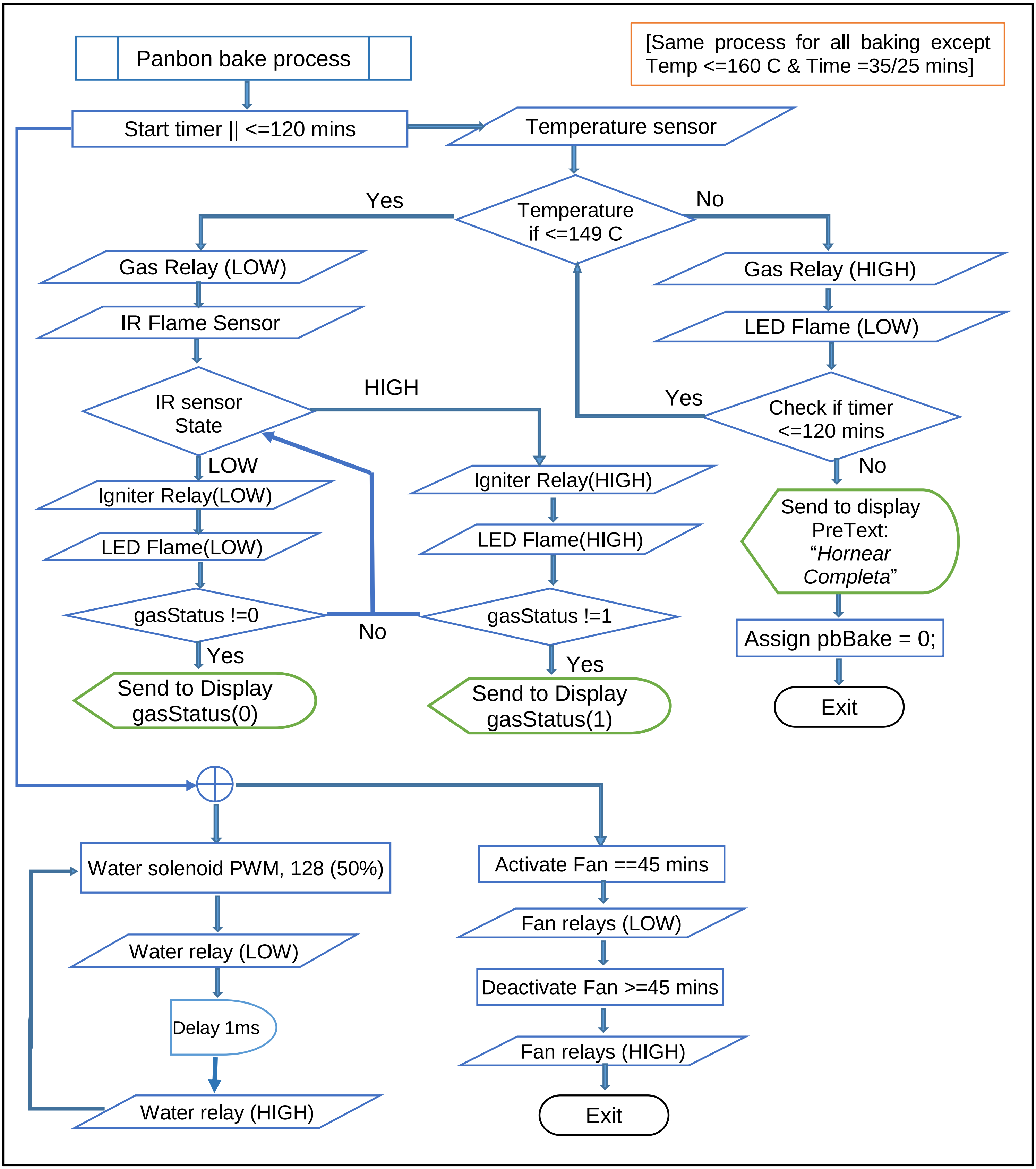
Graduation_Project_CDelight_draft1 §

```
void loop() {
  nexLoop(nex_listen_list); // monitor list for pressing of button in Nextion

///read the temperature sensor A0 analog in value/////////////
  RTDvalue = analogRead(RTDsensor);
  tempValue = map(RTDvalue, 0, 1023, 0, 250); // map it to the range of the analog out:
  if (marcadorTemp < tempValue-2 || marcadorTemp>tempValue+2){
      marcadorTemp = tempValue;
      TempInfo.setValue(tempValue);// send to Nextion display
   }
////Check if door is open - if open LED = On/HIGH//////////
  doorState = digitalRead(doorSensor);
  if (doorState == HIGH) {
         digitalWrite (fanRelay, HIGH);// turn off
         digitalWrite (ledDoor,HIGH); // door open LED(On)
     if (marcadorDoor != 1){  // check if change in last state
         marcadorDoor = 1;
         doorStatus.setValue(1);
      }
   } else {
    digitalWrite (ledDoor,LOW); // door close Led(off)
        if (marcadorDoor != 0){    // check if change in last state
        marcadorDoor = 0;
        doorStatus.setValue(0);
     }
   }
////check state of IR sensor (flame yes = High) and assign LedFlame (ON)
 IRstate = digitalRead (IRsensor);
 if (IRstate == HIGH){
     digitalWrite (ledFlame, HIGH);
  } else {
     digitalWrite (ledFlame, LOW);
    }
```

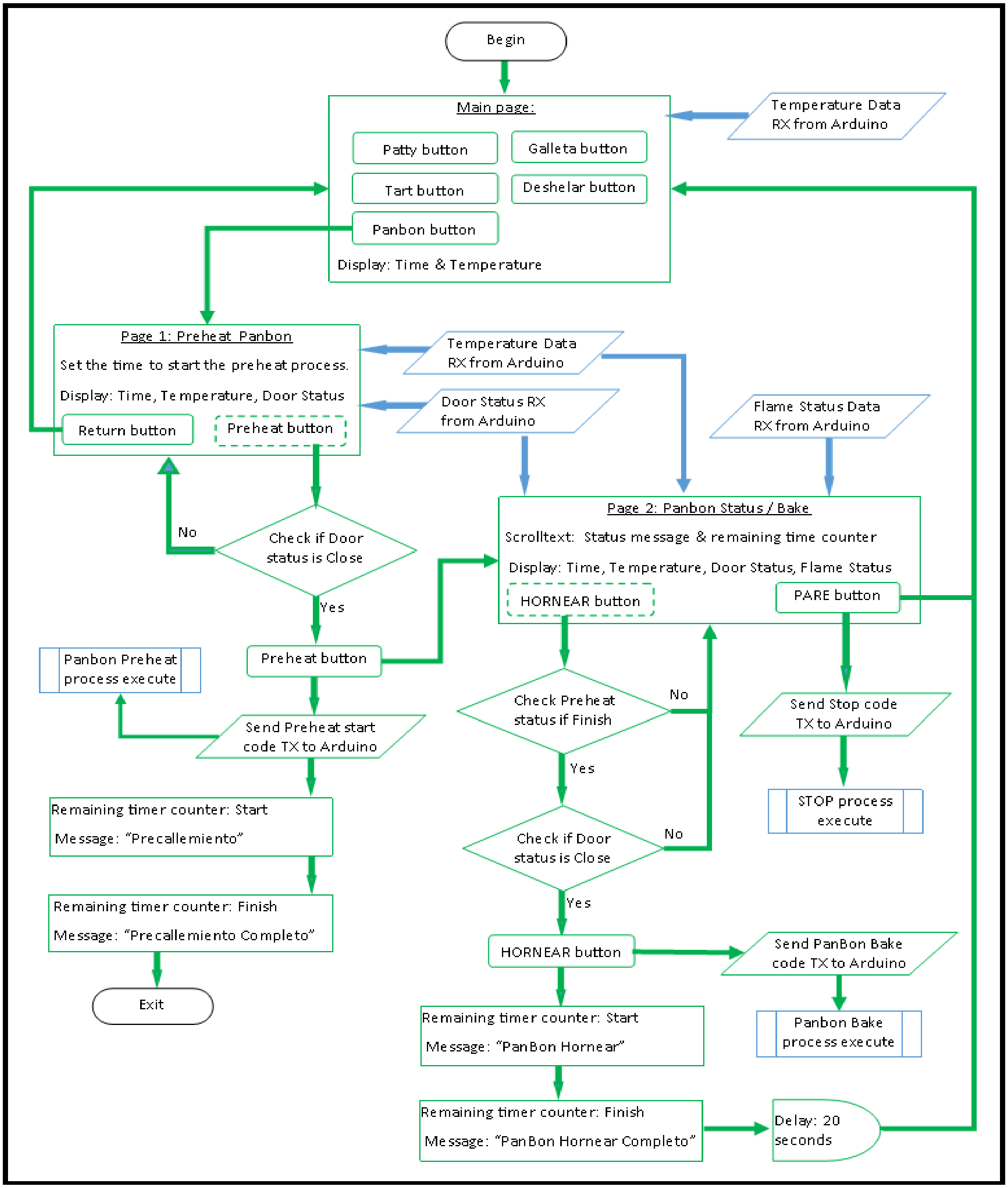
Source: Arduino Sketch final project

Figure 21: Bake flow process



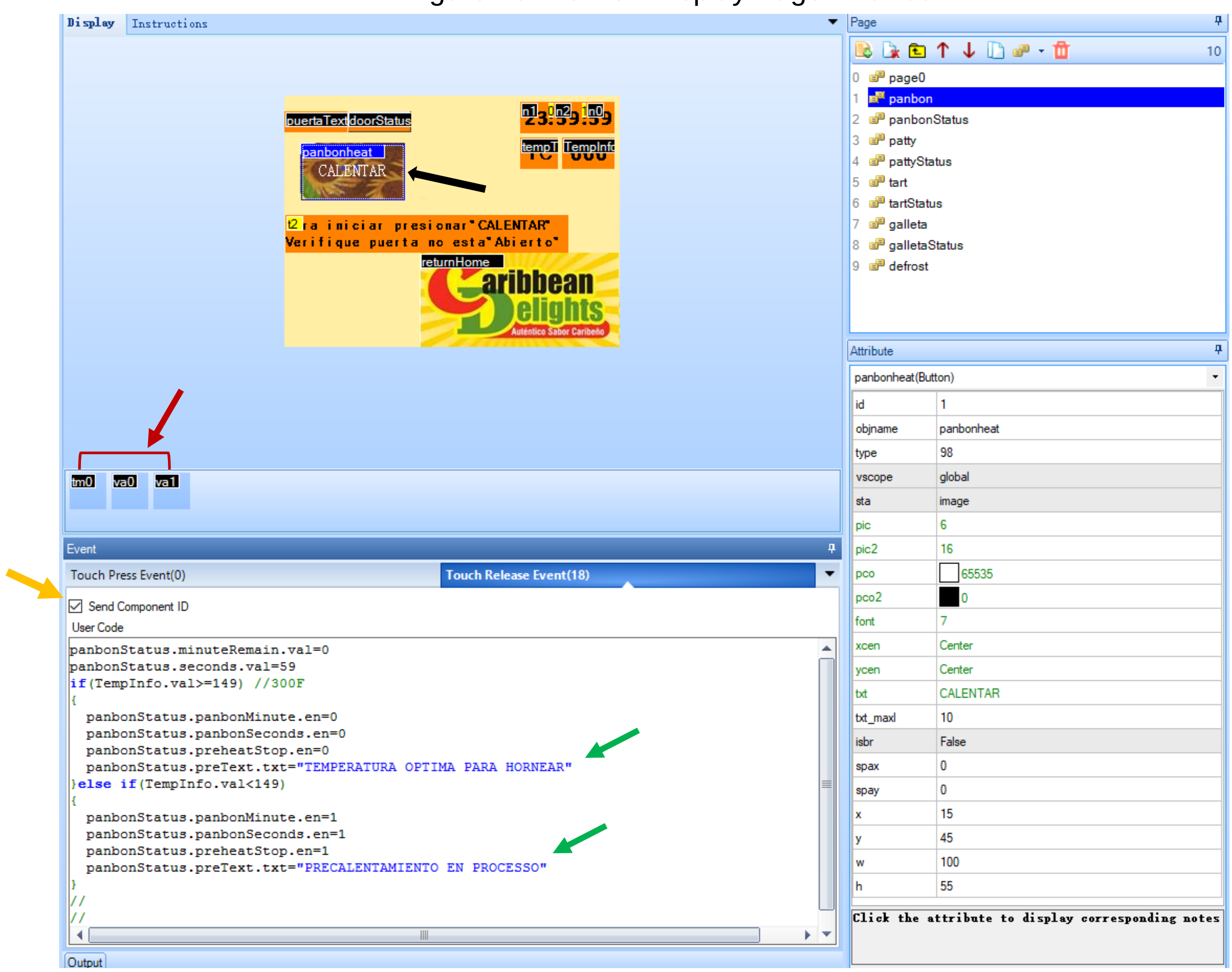
Nextion Display is developed in the Nextion IDE environment. The navigation, buttons, pictures, and texts and certain data flow are managed by the display.

Figure 22:Nextion Display Data Flow Chart



The following briefly layout the programming codes of the Nextion display. In Figure 23, the code that is associated with the push button **CALENTAR** is marked with a black arrow. This button attributes are id#: 1, objname: “panbonheat” and has two pictures #6 displayed normal and #16 displayed when pressed. The user codes shown in the white area is executed when the button is released as a Touch Release Event. The release of the button, also triggers a Send Component ID to the Arduino, as seen with the box that is checked (orange arrow). The Hex code of the page #1, the button id #1 and the button release # 1 are transmitted to the Arduino. The brown arrow shows the timer tm0, and its two variables, va0 and va1, which are described in the following paragraph.

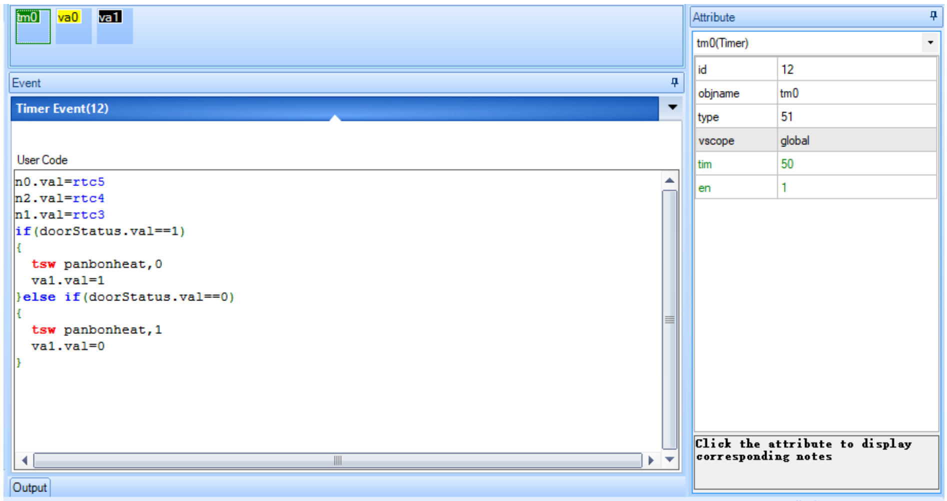
Figure 23: Nextion Display Page1-Panbon



Source: Nextion IDE editor

In Figure 24, the codes in white area are for the timer tm0; the first three lines assigns the display time such as the hour, minutes and seconds to the number boxes n0, n1, and n2. The next line then checks the door Status value, and if the door is open (1), it then disables (0) the button "panbon heat", but if the door is closed (0), it enables (1) the button.

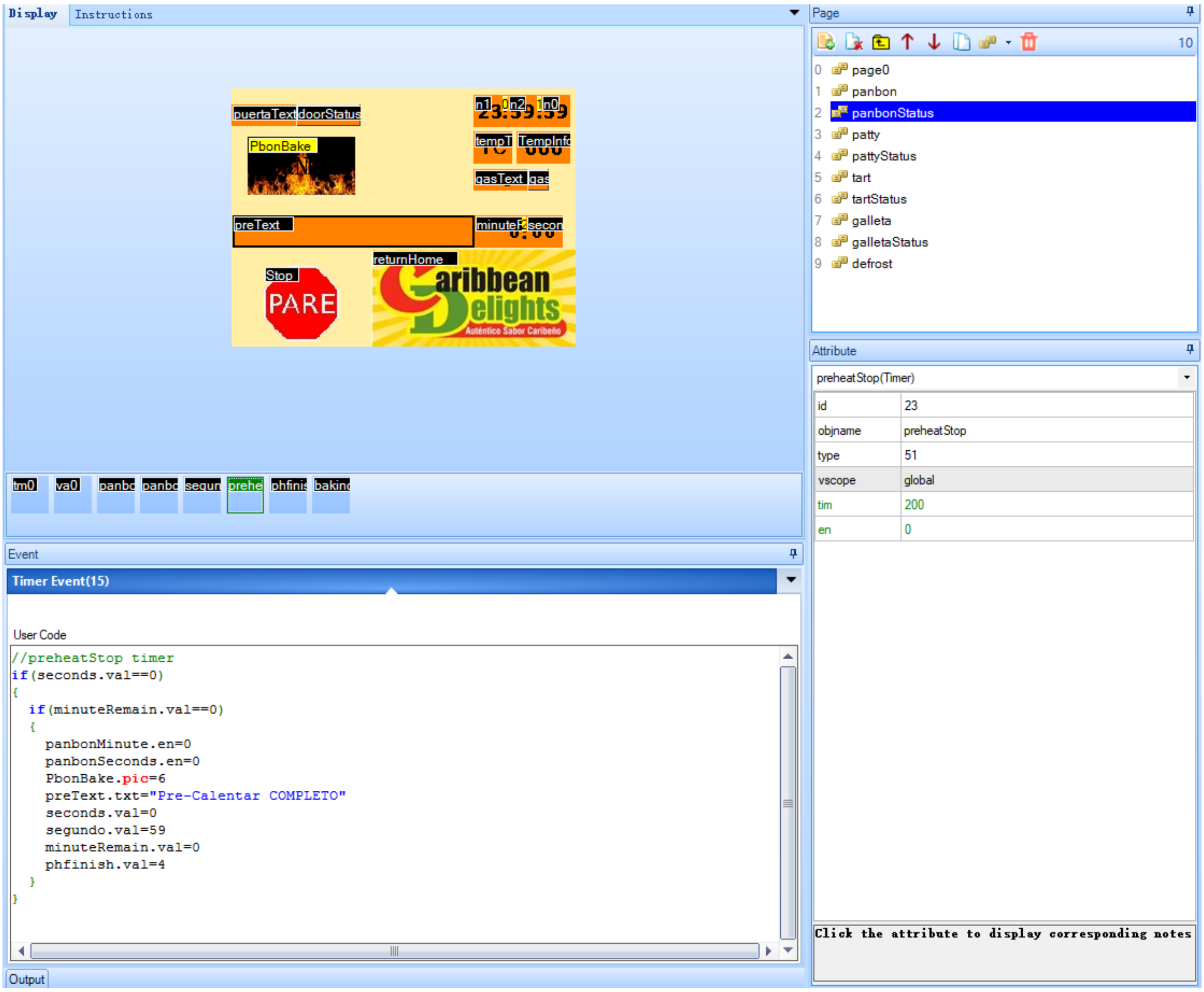
Figure 24: Nextion display page 1 timer tm0



Source: Nextion IDE editor

In reference to Figure 23, the codes in the white area show what is executed when the button **CALENTAR** is released (Touch Release Event). The first line of the code assigns the minute Remain value as (14) minutes to the Panbon Status (page 2). The second line updates the second Value to 59 seconds. The following step is to verify the temperature to see if the preheat process should start. If the temperature is less than 149º C, then it enables the remaining time counter (minutes and seconds) and displays the message in the scrolling text "*Precalentamiento*". However, if the temperature is more than 149º C, then it disable the remaining time counter and send the message "*Temperatura Optima para Hornear*". Afterwards, the display switches page to the pan bon Status page.

Figure 25: Nextion display page 2: PanbonStatus



Source: Nextion IDE editor

Figure 25 shows on page 2, "PanbonStatus", the various timers that create the remaining timer countdown. There are three timers, one for counting the seconds, one for counting the minutes, and one for when the countdown is complete. This timer is "preheatStop", and the codes are seen in the white area. Initially, it verifies if the values in the seconds and the minute timers are equal to (0). If they are equal to (0), then it disables both timers (0), sends the message to the scrolling text "*Pre-calentar COMPLETO*", resets the second and minute variables to (0), and finally, assigns the value (4) to the variable phfinish.

5.3 IMPLEMENTATION

The implementation consists of installing the display, microprocessor controller, relay, contactor, igniter, Wi-Fi module, three sensors and two actuator valves on various locations of the manual oven. The pictures in Figures 26 are taken before the implementation of the hardware.

Figure 26: The Oven controls cabinet before the upgrade to the automatic system

Figure 27: Installation of the controller, relay module and actuators

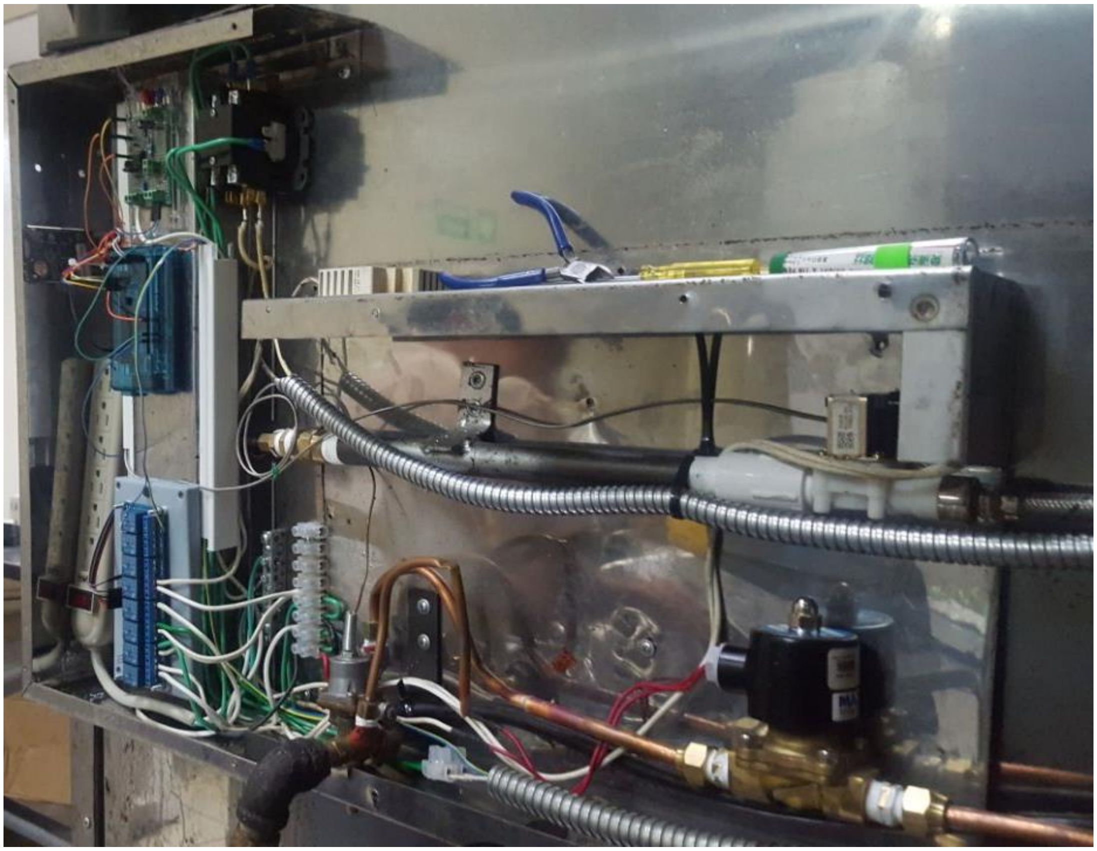


Figure 28: Installation of the IR sensor and the electric igniter

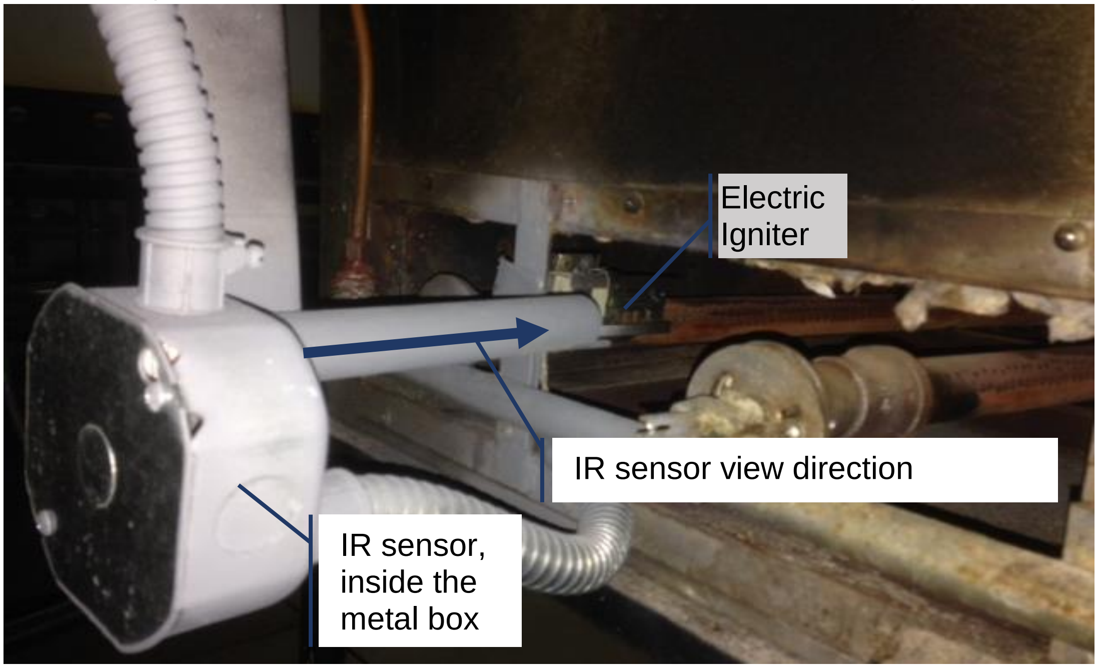


Figure 29: Installation of the controller, relay module and actuators

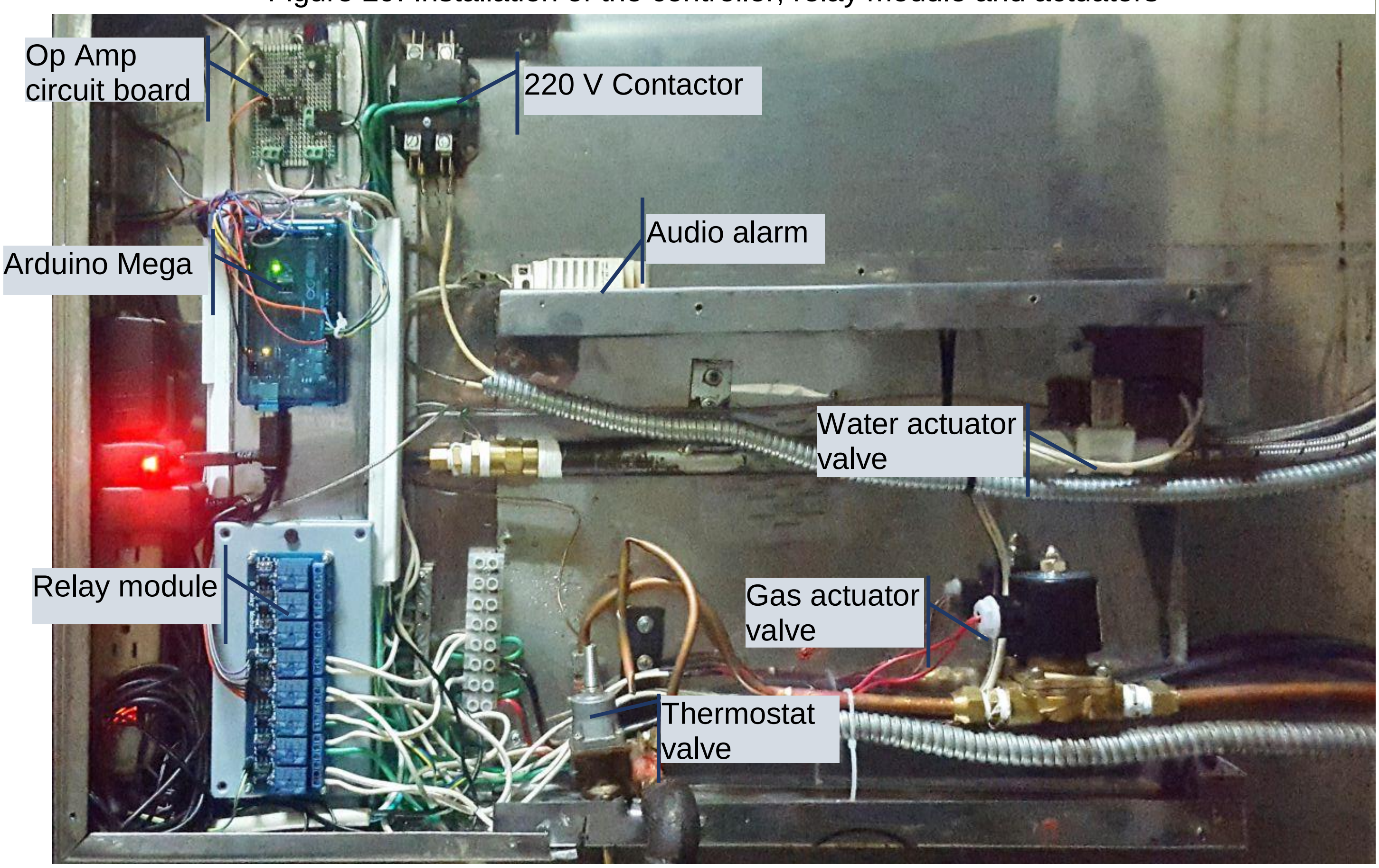


Figure 30: The oven’s controls cabinet after the upgrade ofthe automatic system

Figure 31: Processeswith the time and date saved in the MYSQL database

```
mysql> select * from report;
+---------+----------+-----------+------------+------------+------------+
| process | selector | startTime | cancelTime | finishTime | date       |
+---------+----------+-----------+------------+------------+------------+
|       1 |        0 | 09:02:10  | 09:05:50   | 00:00:00   | 2017-10-20 |
|       1 |        0 | 09:10:05  | 09:20:10   | 00:00:00   | 2017-10-20 |
|       1 |        0 | 09:20:18  | 09:26:43   | 00:00:00   | 2017-10-20 |
|       2 |        0 | 10:00:40  | 10:05:27   | 00:00:00   | 2017-10-20 |
|       2 |        0 | 10:30:10  | 10:40:04   | 00:00:00   | 2017-10-20 |
|       3 |        0 | 14:02:01  | 14:05:20   | 00:00:00   | 2017-10-23 |
|       3 |        0 | 14:20:30  | 00:00:00   | 14:55:30   | 2017-10-23 |
|       4 |        0 | 10:23:55  | 00:00:00   | 10:48:50   | 2017-10-24 |
|       1 |        0 | 11:18:23  | 00:00:00   | 13:18:23   | 2017-10-25 |
+---------+----------+-----------+------------+------------+------------+
9 rows in set (0.01 sec)

mysql>
```

Figure 32: Report application login page

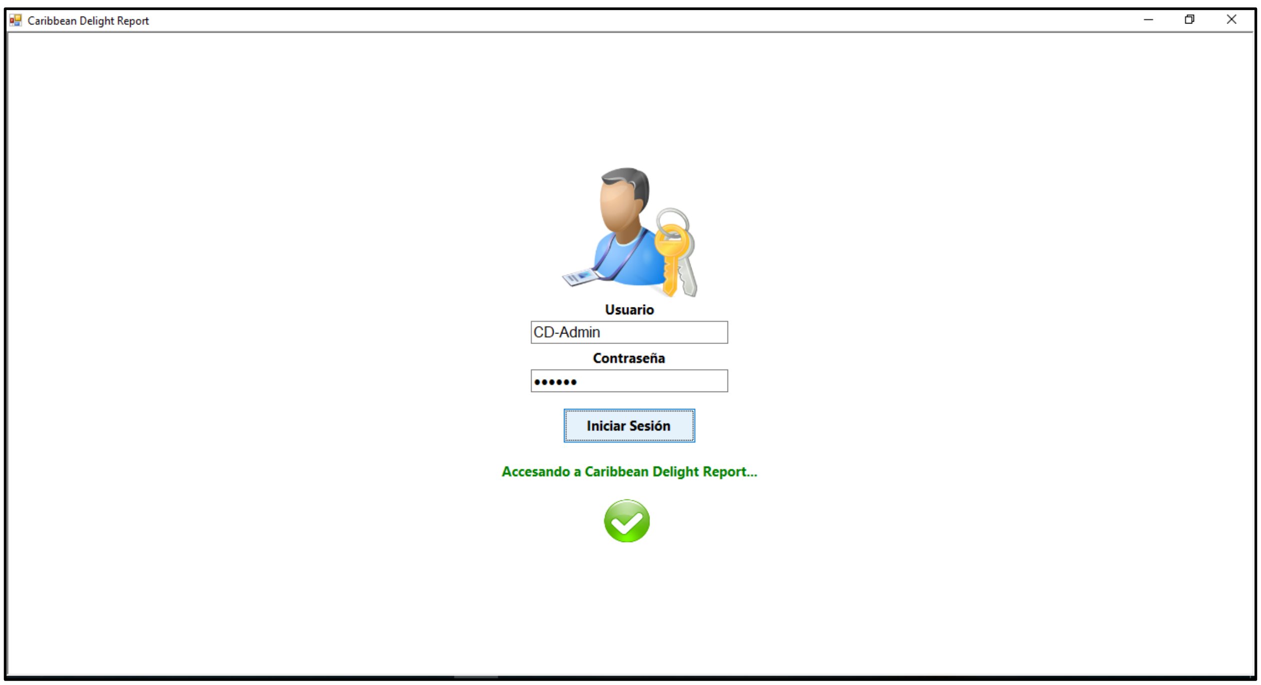
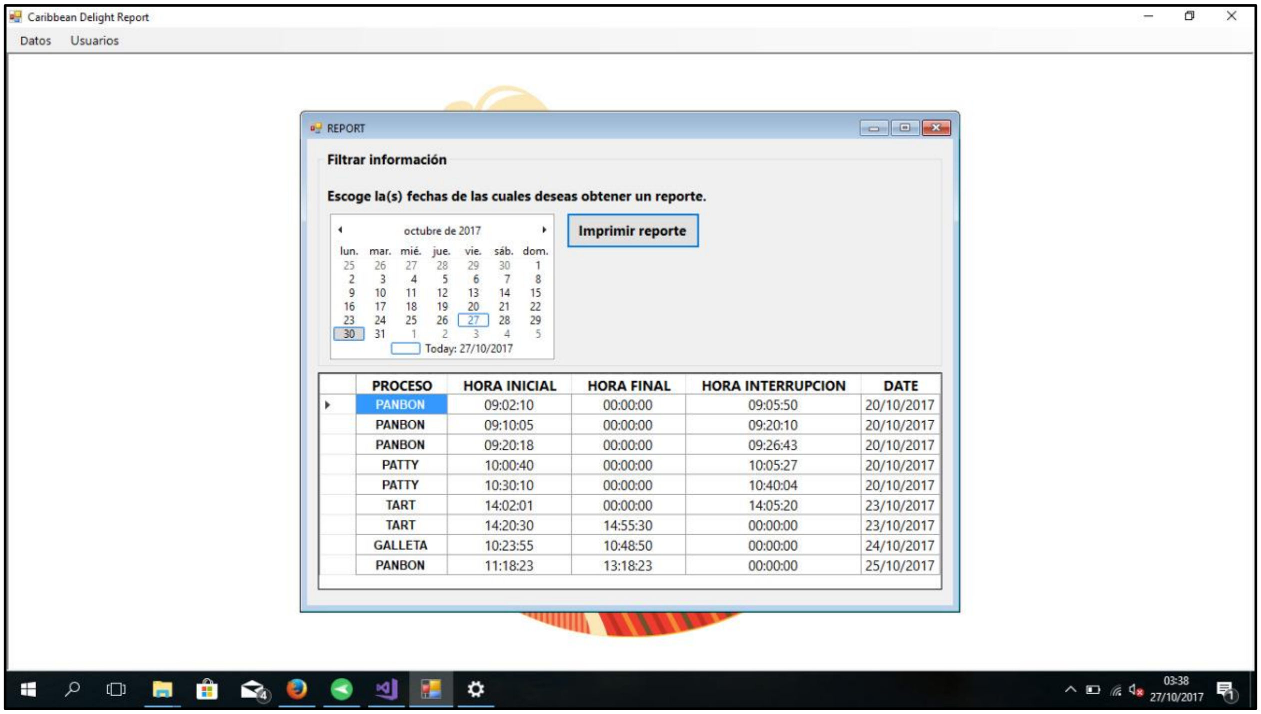


Figure 33: Sample of the Report generated from the MYSQL database



The following are unforeseen issues that occurred during the installation and testing:

- At the owner' requests, the manual thermostat valve was re-installed. An issue then occurred during testing. However, after servicing the thermostat valve it was discovered that the thermostat valve was partially blocked and hence restricting the flow of gas. The thermostat is now serviced and working.

- The existing control panel is not adequately sized for the new control system and the existing width is too narrow for fitting the Nextion display. The installation of the new control cabinet was delayed because the company contractor is busy on other projects. Therefore, another contractor is found to complete the installation of the new control cabinet.

- The system has two power supplies; one of 5 volts and the other 12 volts, each have a positive and negative ground. Initially, on booting up the system on the testing, the Wi-fi did not work. When checking the serial monitor, only errors were noted. It was noted that a similar issue occurred during prototype testing and therefore, the negative grounds of both power supplies were connected to form a common ground with each other. After, the Wi-fi worked well and it connected to the network allowing for saving of the process data from Arduino to database.

- During testing, it is observed that the preheating process is not working as the original design scope. It is noted that at the end of preheating, the system turns off the gas and waited for the bake process to initiate. However, after a while, the oven starts losing heat due to the lack of heat caused by turning off the gas. Upon checking the software, it was noted that two command lines were added to turn off gas and igniter. In the design specifications, these commands were only to be at end of baking processes, but due to an oversight, it was also placed at the end of the preheating processes. After they were removed, the system works as designed.

5.4 COST ANALYSIS

For the cost analysis, the engineer salary rate used is in accordance to the Costa Rican law decree No 40022-MTSS, published in gazette 230, dated 30/11/2016. Any calculations between Colones and US dollars are done as to the exchange rates dated 12/09/17. Both are shown in Table 4, also, all prices and costs are in US dollars.

Table 4: Salary Rate and Exchange Rate Colones to US Dollars

	Colones	
*Bachelor Degree	monthly	hour
Decreto N° 40022-MTSS, publicado en la gaceta 230,	524477.85	3025.83
Colegio Federado de Ingenieros y de Arquitectos de Costa Rica (minimum monthly salary)		
Colones to $1.00 US dollars exchange rate (12/10/17)	buy	578.5
	sell	566.5

Table 5: Project total cost - material and labor

Arduino microprocessor solution			
Parts Description	**Unit**	**Amount**	**Subtotal**
Arduino Mega	1	$ 33.00	$ 33.00
Wi-Fi Shield module	1	$ 7.95	$ 7.95
Igniter	1	$ 12.50	$ 12.50
Arduino Mega case	1	$ 5.99	$ 5.99
Relay 8 port	1	$ 8.98	$ 8.98
IR Flame sensor	1	$ 2.49	$ 2.49
Touch 2.4" display	1	$ 24.99	$ 24.99
Gas valve actuator	1	$ 38.00	$ 38.00
RTD temperature sensor	1	$ 9.99	$ 9.99
Fan contactor 2 pole/220V	1	$ 8.49	$ 8.49
Components LEDs/Resistors	5	$ 0.50	$ 2.50
Door micro-switch	1	$ 1.76	$ 1.76
water valve actuator	1	$ 4.69	$ 4.69
Shipping	1	$ 35.00	$ 35.00
cables/connectors 5%	1	$ 9.80	$ 9.80
		Parts total	$ 206.13
Labor	**hours**	**Rate/hour**	**Subtotal**
Engineering services per hour*	120	$ 5.23	$ 627.66
Transportation (12 bus trips)			$ 9.98
design 16 hr, prototype 64hr		Labor total	$ 637.64
implement 16hr, testing 24 hr		**Grand Total**	**$ 843.77**

The Project Cost Analysis, Table 6, compares the total cost of this upgrade project with the totals of procuring similar new automated ovens from different suppliers. As shown, the cost of this upgrade is clearly a better investment since it costs 96% less than acquiring a new automated oven. Similarly, the power consumption of the upgrade system is 614 watts, which consumes 44.18% less power than the newer oven. Also, the detail including labor and material parts of this automatic control project implementation total investment cost is shown below in the Table 5.

Table 6: Project cost analysis

Company	**Caribbean Delights**	**Electro Maz Ltda.**	**Diseños Metalmecánicos S.A**	**UNOX Inc.**
Product Aspect	Upgrade, own design	MZPG-HC10B	Touchline 10GN 1/1	ChefTop XAVC-10FSGPR
Oven type	Gas Automatic	Gas Automatic	Gas Automatic	Gas Automatic
Features	Controller only	Controller + new oven	Controller + new oven	Controller + new oven
Touch screen	Yes	No	Yes	Yes
Temperature (max)	250 C	400 C	300 C	260 C
Humidity	Yes	Yes	Yes	Yes
Fan / Speed control	Yes (single) / No	Yes (double) / Yes	Yes (double) / Yes	Yes (triple) / Yes
Alarm	Yes	Yes	Yes	Yes
Programmable	Yes	No	Yes (USB)	Yes (digital pen)
Steam generator	No	No	Yes	Yes
Generate report	Yes	No	No	No
Power (Watts)	614	1100	300	1000
Costs ($USD) w/ Tax	$843.77	$5,974.68	$18,617.34	$23,389.59

5.5 OPERATING CONDITIONS

NOTE: Since most of the users are Spanish speakers, the operating manual is written in the Spanish language.

Manual de usuario

El horno va a tener una pantalla de 2.3” que se usa para manejar el control del horno. Hay una pantalla principal que tiene opciones para escoger el producto que quiere hornear, ver la página siguiente.

Página principal

Se puede ver en la página principal que hay 5 botones para escoger: **PANBON**, **PATTY**, **TART**, **GALLETA** y **Deshelar**. Otra informacion que se puede ver en esta página es la temperatura del horno y la hora. Para escoger una de las 4 opciones, presiona el botón de **PANBON**, **PATTY**, **TART**, o **GALLETA**,y se cambia a la página de precalentamiento del horno. En la siguiente figura se muestra el ejemplo de la página de precalentamiento de **PANBON**.Las páginas de **PATTY**, **TART**, y **GALLETA** son semejantes. La opción de **Deshelar** es diferente y es explicada más adelante.

Página Panbonprecalentamiento (puerta: **Abierto**)

En la página de precalentamiento, presione el botón **CALENTAR** para iniciar el proceso de calentar el horno. Si escoge Panbon el horno se calentarahasta 145 ºC, las otras opciónesllegan hasta 160 ºC. Nota: si la puerta está abierta, el indicador "Puerta" va a mostrar **Abierto** .DE este modo, (como se ve en la figura de abajo) el botón **CALENTAR** no se activara hasta que la puerta esté cerrada.

Página Panbon precalentamiento (puerta: **cerrado**)

Cuando el presiona **CALENTAR**, inicia el proceso de calentar el horno y se pasa a la página de hornear, como se ve en las siguientes dos figuras.

Página Status (Panbon): Precalentamiento en proceso

En la parte superior de está página se ve el proceso de Precalentamiento y en el área de texto moviendo (flecha azul) se ve el mensaje “PRECALENTAMIENTO” y su tiempo restante (flecha verde). En está página también está el indicador de Fuego (flecha negra) que indica que el fuego esta encendido. Cuando el Precalentamiento está completo, el tiempo que falta llega a 0:00 y el mensaje en el área de texto (flecha azul) se cambia a “PRECALENTAMIENTO COMPLETO”.. En está página también hay dos botones, **PARE** y **HORNEAR**. El botón **PARE** es para parar cualquier proceso. Si están en el proceso de Precalentamiento o en el proceso de Hornear y por cualquier razón necesita terminar y parar el proceso – presionar el botón **PARE**. El controlador cierra de inmediato la válvula de gas y regresa la pantalla a su página principal.

Página Status (Panbon): Precalentamiento completo

En está misma página tambien esta el botón de **HORNEAR**. Cuando presiona el botón **HORNEAR,** como cuando está en Panbon, empieza el proceso de hornear. En este proceso, el controlador mantiene el horno a 145 ºC por 45 minutos con el ventilador encendido y 75 minutos sin ventilador.

Nota: el botón de **HORNEAR** no se activa cuando lo presiona si el indicador puerta esta en **Abierto**.

En el proceso de hornear de Patty y Tart el controlador mantiene el horno a 160 ºC por 35 minutos con ventilador. En el proceso de hornear de Galleta el controlador mantiene el horno a 160 ºC por 25 minutos con ventilador.

Cuando presiona el botón de Deshelar en la página principal se llega a la página Deshelar. El proceso de Deshelar es para descongelar productos. Este calienta el horno de temperatura ambiente hasta 145 ºC y lo mantiene allí por 5 minutos, después se apaga el horno y regresa a la página principal. Nota: El botón de Deshelar (flecha negra) no se activa si el indicador “Puerta” está en modo **Abierto**.

Página Deshelar: Inicio

En esta página también hay el botón **PARE**, si por cualquier razón necesita terminar este proceso – presionar el botón **PARE**.

CHAPTER VI: RECOMMENDATIONS AND CONCLUSIONS

6.1 RECOMMENDATIONS

- Install an individual shut-off gas valve for the oven. It shares a valve with the other oven; this affected the testing and usage of both ovens.
- Perform extensive research to verify that the parts and components procured have the required specifications, connections and dimensions necessary for the implementation of the project.
- Install industrial type light fixtures in the oven since the current one is inadequate. Its light bulb always falls due to the fan vibration and breaks within the oven. The broken glass may contaminate the baking products and hinders the production efficiency.
- Replace the fan blade with a more efficient one since the current one is made of angle iron bolted together; they do not effectively circulate the heated air.
- Install an electrical main control switch for the oven.
- Implement a design that allows variable speed control of the fan.
- Completely remove or replace the existing defected manual solenoid valve currently installed on the oven.
- After viewing the feature of a graphic input program concept for new bake process, this can be implemented on this oven also using a digital pen.

6.2 CONCLUSIONS

- Initially, the oven was operated manually and to function was fully dependent on human intervention. After the project is implemented, with a touch of the display screen, the operator chooses a product by pressing a virtual button on the display. The display communicates with the Arduino microcontroller and it initiates an interrupt routine, which starts the process of preheating and baking of the chosen product. The Arduino microcontroller manages all the necessary variables to complete the baking process without any more inputs from the operator.

- At first, during the design scope and prototype testing, the hysteresis was set at 1 ºC. However, during testing at various times, the oven control cycles the relays less than 20 seconds. To reduce the chance of damaging the relays and actuators, the hysteresis was adjusted to 3 ºC. Note: when the oven used the manual thermostat, the hysteresis was recorded at 5.0 ºC. The oven controller operates stable at a hysteresis of 3 ºC. The components such as the relays, igniter and actuator valve now oscillates, turning off and on, between a 1.0 to 2.0 minutes cycle. This adjustment of the hysteresis to 3 ºC to did not affect the quality of the baking.

- The owners of the business requested the ability to track the product baked by the oven. A MySQL database stores the following information: the process names, start time, finish time, or cancelation time along with the date. The Arduino sends the relevant data information of baking process to the database via its Wi-Fi that is connected to the LAN. An application designed in Visual Studio generates the report by accessing the information from the MySQL database. The information on the report assists the owner to track the oven baking of the products and its efficiency. Furthermore, this system has much potential for growth in the IoT (Internet of Things) spectrum due to features such as Wi-Fi and database capabilities. One example is mobility by implementing a remote access app to control the oven's operations from a cell phone, tablet, or PC.

- The following is the comparison of existing factory-made automated oven solutions to designing and implementing a customized solution, as with this project. In the analysis performed it was shown that the customized solution cost less than 96% in comparison to similar type factory-made solutions. Hence, a customized solution is definitely financially feasible. Added benefit of the customized solution is that any future upgrades, for added features such as variable fan speed or adding another manual oven, would require minimal monetary investments and minor programming modifications.

BIBLIOGRAPHY

Arduino AG. (2017). *Arduino AG*. Retrieved from http://www.arduino.cl/

Boylestad, R. L., & Nashelsky, L. (2013). *Electronic Device and Circuit Theory* (11 ed.). NJ, USA: Pearson Education, Inc.

Caribbean Delight. (2017). *http://www.carideli.com/quienessomos.php*. Retrieved from http://www.carideli.com/index.php

Dorf, R. C., & Bishop, R. H. (2011). *Modern Control Systems* (12th ed.). NY, USA: Pearson Education, Inc.

Electrical4u. (2017). *ONLINE ELECTRICAL ENGINEERING STUDY SITE*. Retrieved from https://www.electrical4u.com/

Encyclopedia Britannica, Inc. (2017). *ENCYCLOPEDIA BRITANNICA*. Retrieved from https://www.britannica.com/

Figliola, R. S., & Beasley, D. E. (2011). *Theory and Design for Mechanical Measurements* (5th ed.). MA, USA: John Wiley & Sons Inc.

ITEAD Intelligent Systems Co.Ltd. (2017). *ITEAD CC*. Retrieved from https://www.itead.cc/

Nussey, J. (2013). *Arduino for Dummies.* West Sussex, England: John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com.

Park, J., & Mackay, S. (2003). *Practical Data Acquisition for Instrumentation and Control Systems.* Oxford: Newnes publications. Retrieved from http://www.idc-online.com/

PMBOK® Guide—5th Ed. Content Committee. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (Fifth ed.). Pennsylvania, USA: Project Management Institute, Inc. Retrieved from www.pmi.org

Webster, J. G. (1999). *Measurement, Instrumentation, and Sensors Handbook.* FL: CRC Press LLC.

ANNEXES

ANNEX 1: ARDUINO LANGUAGE REFERENCE

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi-line comment)
- #define
- #include

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP
- LED_BUILTIN
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- short
- float
- double
- string - char array
- String - object
- array

Conversion

- char()
- byte()
- int()
- word()
- long()
- float()

Variable Scope & Qualifiers

- variable scope
- static
- volatile
- const

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Due & Zero only

- analogReadResolution()
- analogWriteResolution()

Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Time

- millis()
- micros()
- delay()
- delayMicroseconds()

Math

- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry

- sin()
- cos()
- tan()

Boolean Operators

- && (and)
- || (or)
- ! (not)

Pointer Access Operators

- * dereference operator
- & reference operator

Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- %= (compound modulo)
- &= (compound bitwise and)
- |= (compound bitwise or)

- const

Utilities

- sizeof()
- PROGMEM

- tan()

Characters

- isAlphaNumeric()
- isAlpha()
- isAscii()
- isWhitespace()
- isControl()
- isDigit()
- isGraph()
- isLowerCase()
- isPrintable()
- isPunct()
- isSpace()
- isUpperCase()
- isHexadecimalDigit()

Random Numbers

- randomSeed()
- random()

Bits and Bytes

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

External Interrupts

- attachInterrupt()
- detachInterrupt()

Interrupts

- interrupts()
- noInterrupts()

Communication

- Serial
- Stream

ANNEX 2: NEXTION DISPLAY INSTRUCTIONS

The Nextion Instruction Set

These are the set of commands that Nextion can use.
They are categorized into only a few categories

1 – General Rules and Practices

No.	General Rule or Practice
1	**All instructions over serial: *are terminated with three bytes of* 0xFF 0xFF 0xFF** ie: decimal: 255 or hex: 0xFF or ansichar: ÿ or binary: 11111111 ie byte ndt[3] = {255,255,255}; write(ndt,3); or print("\xFF\xFF\xFF"); or print("ÿÿÿ")
2	All instructions and parameters are in ASCII
3	All instructions are in lowercase letters
4	Blocks of code and enclosed within braces { } can not be sent over serial this means if, for, and while commands can not be used over serial
5	A space char 0x20 is used to separate command from parameters.
6	There are no spaces in parameters unless specifically stated
7	Nextion uses integer math and does not have real or floating support.
8	Assignment are non-complex evaluating fully when reaching value after operator.
9	Comparison evaluation is non-complex. Use nesting when and as required.
10	Instructions over serial are processed on receiving termination (see 1.1)
11	Character escaping is limited to 0x0D by using two text chars \r
12	Nextion does not support order of operations. sys0=3+(8*4) is invalid.
13	16-bit 565 Colors are in decimal from 0 to 65535 (see 5.Note)
14	Text values must be encapsulated with double quotes: ie "Hello"
15	Items within specific to Enhanced Models are noted with *K*
16	Transparent Data Mode (used by addt and wept commands)

1. MCU sending to Nextion
    1. MCU sends command. ie: wept 30,20ÿÿÿ or addt 1,0,320ÿÿÿ
    2. Nextion requires ~5ms to prepare for transparent mode data transfer
    3. Nextion sends "Ready" 0xFE 0xFF 0xFF 0xFF Return Data (see 7.30)
    4. MCU can now send specified quantity (20) of raw bytes to Nextion
    5. Nextion receives raw bytes from MCU until specified quantity (20) is received
    6. Nextion sends "Finished" 0xFD 0xFF 0xFF 0xFF Return Data (see 7.29)
    7. MCU and Nextion can proceed to next command

Reference to a component without specified Attribute can
create for long and potentially frustrating debug sessions
p[pageindex].b[component.id].attribute // global scope
b[component.id].attribute // local scope on current page

3 – Operational Commands

No.	Name	Param Count	Description and Usage/Parameters/Examples
1	page	1	Change page to page specified. Unloads old page to load specified page. Nextion loads page 0 by default on power on. usage: page <pid> <pid> is either the page index number, or pagename page 0 // Change page to indexed page 0 page main // Change page to the page named main
2	ref	1	Refresh component (auto-refresh when attribute changes since v0.38) – if component is obstructed (stacking), ref brings component to top. usage: ref <cid> <cid> is component's .id or .objname attribute of component to refresh – when <cid> is 0 (page component) refreshes all on the current page. ref t0 // Refreshes the component with .objname of t0 ref 3 // Refreshes the component with .id of 3 ref 0 // Refreshes all components on the current page (same as ref 255)
3	click	2	Trigger the specified components Touch Press/Release Event usage: click <cid>,<event> <cid> is component's .id or .objname attribute of component to refresh <event> is 1 to trigger Press Event, 0 to trigger Release Events click b0,1 // Trigger Touch Press Event of component with .objname b0 click 4,0 // Trigger Touch Release Event of component with .id 4
4	ref_stop	0	Stops default waveform refreshing (will not refresh when data point added) – waveform refreshing will resume with ref_star (see 3.5) usage: ref_stop ref_stop // stop refreshing the waveform on each data point added
5	ref_star	0	Resume default waveform refreshing (refresh on data point add) – used to resume waveform refreshing stopped by ref_stop (see 3.4) usage: ref_start ref_star // resume default refreshing, refresh on each data point added
6	get	1	Send attribute/constant over serial (0x70/0x71 Return Data) usage: get <attribute> <attribute> is either numeric value, .txt contents, or constant get t0.txt // sends text contents of t0.txt in 0x70 Return Data format get "123" // sends text constant "123" in 0x70 Return Data format get n0.val // sends numeric value of n0.val in 0x71 Return Data format get 123 // sends numeric constant 123 in 0x71 Return Data format
7	sendme	0	Sends number of currently loaded page over serial (0x66 Return Data) – number of currently loaded page is stored in system variable dp – used in a page's initialize event will auto-send as page loads usage: sendme sendme // sends the value of dp in 0x66 Return Data Format
8	cov	3	Convert variable from numeric type to text, or text to numeric type – text must be text ASCII representation of an integer value. – source and destination types must not be of the same type – when length is fixed and value is less, leading zeros will be added ie: src numeric value of 123 with length 4, result is dest text "0123"

			– when value is larger than length, .txt truncated to most significant digits ie: src numeric value of 23425 with length 4 result is dest text "2342" usage: cov <src>,<dest>,<length> <src> is text attribute (or numeric attribute when <dest> is text) <dest> is numeric attribute (or text attribute when <src> is numeric) <length> will determine if leading zeros added to conversion to text cov h0.val,t0.txt,0 // convert value of h0 into t0.txt without leading zeros cov t0.txt,h0.val,0 // convert integer in t0.txt into h0.val <length> ignored. cov h0.val,t0.txt,4 // convert value of h0 into t0.txt with exactly 4 digits Invalid: cov h0.val,va0.val,0 or cov t0.txt,va0.txt,0 // src & dest same type.
9	touch_j	0	Recalibrate the Nextion device's touch sensor – presents 4 points on screen for user to touch, saves, and then reboots. – typically not required as device is calibrated at factory – sensor can be effected by changes of conditions in environment usage: touch_j touch_j // trigger the recalibration of touch sensor
10	substr	4	Extract character or characters from contents of a Text attribute usage: substr <src>,<dest>,<start>,<count> <src> is text attribute where text will be extracted from <dest> is text attribute to where extracted text will be placed <start> is start position for extraction (0 is first char, 1 second) <count> is the number of characters to extract substr va0.txt,t0.txt,0,5 // extract first 5 chars from va0.txt, put into t0.txt
11	vis	2	Hide or Show component on current page – show will refresh component and bring it to the forefront layer – hide will remove component visually, touch events will be disabled – use layering with mindful purpose, can cause ripping and flickering. – use with caution and mindful purpose, may lead to difficult debug session usage: vis <comp><state> <comp> is either .objname or .id of component on current page, – valid .id is 0 – page, 1 to 250 if component exists, and 255 for all <state> is either 0 to hide, or 1 to show. vis b0,0 // hide component with .objname b0 vis b0,1 // show component with .objname b0, refresh on front layer vis 1,0 // hide component with .id 1 vis 1,1 // show component with .id 1, refresh on front layer vis 255,0 // hides all components on the current page
12	tsw	2	Enable or disable touch events for component on current page – by default, all components are enabled unless disabled by tsw – use with caution and mindful purpose, may lead to difficult debug session usage: tsw <comp><state> <comp> is either .objname or .id of component on current page, – valid .id is 0 – page, 1 to 250 if component exists, and 255 for all <state> is either 0 to disable, or 1 to enable. tsw b0,0 // disable Touch Press/Release events for component b0 tsw b0,1 // enable Touch Press/Release events for component b0 tsw 1,0 // disable Touch Press/Release events for component with id 1 tsw 1,1 // enable Touch Press/Release events for component with id 1 tsw 255,0 // disable all Touch Press/Release events on current page
13	com_stop	0	Stop execution of instructions received from Serial – Serial will continue to receive and store in buffer. – execution of instructions from Serial will resume with com_star (see 3.14) – using com_stop and com_star may cause a buffer overflow condition.

			– Refer to device datasheet for buffer size and command queue size usage: com_stop com_stop // stops execution of instructions from Serial
14	com_star	0	Resume execution of instructions received from Serial – used to resume an execution stop caused by com_stop (see 3.13) – when com_star received, all instructions in buffer will be executed – using com_stop and com_star may cause a buffer overflow condition. – Refer to device datasheet for buffer size and command queue size usage: com_star com_star // resume execution of instruction from Serial
15	randset	2	Set the Random Generator Range for use with rand (see 6.14) – range will persist until changed or Nextion rebooted – set range to desired range before using rand – power on default range is -2147483648 to 2147483647, runtime range is user definable. usage: randset <min>,<max> <min> is value is -2147483648 to 2147483647 <max> is value greater than min and less than 2147483647 randset 1,100 //set current random generator range from 1 to 100 randset 0,65535 //set current random generator range from 0 to 65535
16	code_c	0	Clear the commands queued in command buffer without execution usage: code_c code_c // Clears the command buffer without execution
17	print	1	Send raw formatted data over Serial to MCU – print/printh does not use Nextion Return Data, user must handle MCU side – qty of data may be limited by serial buffer (all data < 1024) – numeric value sent in 4 byte 32-bit little endian order value = byte1+byte2*256+byte3*65536+byte4*16777216 – text content sent is sent 1 ASCII byte per character, without null byte. usage: print <attr> <attr> is either component attribute, variable or Constant print t0.txt // return 1 byte per char of t0.txt without null byte ending. print j0.val // return 4 bytes for j0.val in little endian order print "123" // return 3 bytes for text "123" 0x31 0x32 0x33 print 123 // return 4 bytes for value 123 0x7B 0x00 0x00 0x00
18	printh	1 to many	Send raw byte or multiple raw bytes over Serial to MCU – printh is one of the few commands that parameter uses space char 0x20 – when more than one byte is being sent a space separates each byte – byte is represented by 2 of (ASCII char of hexadecimal value per nibble) – qty may be limited by serial buffer (all data < 1024) – print/printh does not use Nextion Return Data, user must handle MCU side usage: printh <hexhex>[<space><hexhex>][...<space><hexhex] <hexhex> is hexadecimal value of each nibble. 0x34 as 34 <space> is a space char 0x20, used to separate each <hexhex> pair printh 0d // send single byte: value 13 hex: 0x0d printh 0d 0a // send two bytes: value 13,10 hex: 0x0d0x0a
19	add	3	Add single value to Waveform Channel – waveform channel data range is min 0, max 255 – 1 pixel column is used per data value added – y placement is if value < s0.h then s0.y+s0.h-value, otherwise s0.y usage: add <waveform>,<channel>,<value> <waveform> is the .id of the waveform component <channel> is the channel the data will be added to

			<value> is ASCII text of data value, or numeric value – valid: va0.val or sys0 or j0.val or 10 add 1,0,30 // add value 30 to Channel 0 of Waveform with .id 1 add 2,1,h0.val // add h0.val to Channel 1 of Waveform with .id 2
20	addt	3	Add specified number of bytes to Waveform Channel over Serial from MCU – waveform channel data range is min 0, max 255 – 1 pixel column is used per data value added. – addt uses Transparent Data Mode (see 1.16) – waveform will not refresh until Transparent Data Mode completes. – qty limited by serial buffer (all commands+terminations + data < 1024) – also refer to add command (see 6.19) usage: add <waveform>,<channel>,<qty> <waveform> is the .id of the waveform component <channel> is the channel the data will be added to <qty> is the number of byte values to add to <channel> addt 2,0,20 // adds 20 bytes to Channel 0 Waveform with .id 2 from serial // byte of data is not ASCII text of byte value, but raw byte of data.
21	cle	3	Clear waveform channel data usage: cle <waveform>,<channel> <waveform> is the .id of the waveform component <channel> is the channel to clear <channel> must be a valid channel: < waveform.ch or 255 0 if .ch 1, 1 if .ch 2, 2 if .ch 3, 3 if .ch=4 and 255 (all channels) cle 1,0 // clear channel 0 data from waveform with .id 1 cle 2,255 // clear all channels from waveform with .id 2
22	rest	0	Resets the Nextion Device usage: rest rest // immediate reset of Nextion device – reboot.
23	doevents	0	Force immediate screen refresh and receive serial bytes to buffer – useful inside exclusive code block for visual refresh (see 3.26 and 3.27) usage: doevents doevents // allows refresh and serial to receive during code block
24	strlen	2	Computes the length of string in <txt> and puts value in <len> usage: strlen <txt>,<len> <txt> must be a string attribute ie: t0.txt, va0.txt <len> must be numeric ie: n0.val, sys0, va0.val strlen t0.txt,n0.val // assigns n0.val with length of t0.txt content
24a	btlen	2	Computes number of bytes string in <txt> uses and puts value in <len> usage: btlen <txt>,<len> <txt> must be a string attribute ie: t0.txt, va0.txt <len> must be numeric ie: n0.val, sys0, va0.val btlen t0.txt,n0.val // assigns n0.val with number of bytes t0.txt occupies
25	if	Block	Conditionally execute code block if boolean condition is met – execute commands within block { } if (condition) is met. – nested conditions () is not allowed. invalid: ((h0.val+3)>0) – block opening brace { must be on line by itself – no space between block close brace } and else. valid: }else invalid: } else – Text comparison supports ==, != – Numerical comparison supports >, <, ==, !=, >=, <= – nested "if" and "else if" supported. usage: if condition block [else if condition block] [else block] – (condition) is a simple non-complex boolean comparison evaluation valid: (j0.val>75) invalid: (j0.val+1>75) invalid: (j0.val<now.val+60)

```
if(t0.txt=="123456")
{
page 1
}if(n0.val==123)
{
b0.txt="stop"
}else
{
b0.txt="start"
}if(rtc==1)
{
t0.txt="Jan"
}else if(rtc1==2)
{
t0.txt="Feb"
}else if(rtc1==3)
{
t0.txt="Mar"
}else
{
t0.txt="etc"
}
```

26	while	Block	Continually executes code block until boolean condition is no longer met
			– tests boolean condition and execute commands within block { } if condition was met and continues to re-execute block until condition is not met.
			– nested conditions () is not allowed. invalid: ((h0.val+3)>0)
			– block opening brace { must be on line by itself
			– Text comparison supports ==, !=
			– Numerical comparison supports >, <, ==, !=, >=, <=
			– block runs exclusively until completion unless doevents used (see 3.23)
			usage: while condition block
			– (condition) is a simple non-complex boolean comparison evaluation
			valid: (j0.val>75) invalid: (j0.val+1>75)
			// increment n0.val as lon as n0.val < 100. result: b0.val=100
			// will not visually see n0.val increment, refresh when while-loop completes
			while(n0.val<100)
			{
			n0.val++
			}//increment n0.val as long as n0.val < 100. result: n0.val=100
			// will visually see n0.val increment, refresh each evaluation of while-loop
			while(n0.val<100)
			{
			n0.val++
			doevents
			}

27	for	Block	Iterate execution of code block as long as boolean condition is met – executes init_assignment, then tests boolean condition and executes commands within block { } if boolean condition is met, when iteration of block execution completes step_assignment is executed. Continues to iterate block and step_assignment until boolean condition is not met. – nested conditions () is not allowed. invalid: ((h0.val+3)>0) – block opening brace { must be on line by itself – Text comparison supports ==, != – Numerical comparison supports >, <, ==, !=, >=, <=

			– block runs exclusively until completion unless doevents used (see 3.23)
			usage: for(init_assignment;condition;step_assignment) block
			– init_assignment and step_assignment are simple non-complex statement
			valid: n0.val=4, sys2++, n0.val=sys2*4+3 invalid: n0.val=3+(sys2*4)-1
			– (condition) is a simple non-complex boolean comparison evaluation
			valid: (j0.val>75) invalid: (j0.val+1>75)
			// iterate n0.val by 1's as long as n0.val<100. result: n0.val=100
			// will not visually see n0val increment until for-loop completes
			for(n0.val=0;n0.val<100;n0.val++)
			{
			}//iterate n0.val by 2's as long as n0.val<100. result: n0.val=100
			// will visually see n0.val increment when doevents processed
			for(n0.val=0;n0.val<100;n0.val+=2)
			{
			doevents
			}

28 *K*	wepo	2	Store value/string to EEPROM – EEPROM valid address range is from 0 to 1023 (1K EEPROM) – numeric value length: is 4 bytes, -2147483648 to 2147483647 – numeric data type signed long integer, stored in little endian order. val[addr+3]*16777216+val[addr+2]*65536+val[addr+1]*256+val[addr] – string content length: .txt content is .txt-maxl +1, or constant length +1 usage: wepo <attr>,<addr> <attr> is variable or constant <addr> is storage starting address in EEPROM wepo t0.txt,10 // writes t0.txt contents in addresses 10 to 10+t0.txt-maxl wepo "abcd",10 // write constant "abcd" in addresses 10 to 14 wepo 11,10 // write constant 11 in addresses 10 to 13 wepo n0.val,10 // write value n0.val in addresses 10 to 13
29 *K*	repo	2	Read value from EEPROM – EEPROM valid address range is from 0 to 1023 (1K EEPROM) – numeric value length: is 4 bytes, -2147483648 to 2147483647 – numeric data type signed long integer, stored in little endian order. val[addr+3]*16777216+val[addr+2]*65536+val[addr+1]*256+val[addr] – string content length: .txt content is lesser of .txt-maxl or until null reached. usage: repo <attr>,<addr> <attr> is variable or constant <addr> is storage starting address in EEPROM repo t0.txt,10 // reads qty .txt-maxl chars (or until null) from 10 into t0.txt repo n0.val,10 // reads 4 bytes from address 10 to 13 into n0.val
30 *K*	wept	2	Store specified number of bytes to EEPROM over Serial from MCU – EEPROM valid address range is from 0 to 1023 (1K EEPROM) – wept uses Transparent Data Mode (see 1.16) – qty limited by serial buffer (all commands+terminations + data < 1024) usage: wept <addr>,<qty> <addr> is storage starting address in EEPROM <qty> is the number of bytes to store wept 30,20 // writes 20 bytes into EEPROM addresses 30 to 49 from serial // byte of data is not ASCII text of byte value, but raw byte of data.
31 *K*	rept	2	Read specified number of bytes from EEPROM over Serial to MCU – EEPROM valid address range is from 0 to 1023 (1K EEPROM) usage: rept <attr>,<addr> <addr> is storage starting address in EEPROM

ANNEX 3: SITE VISITS

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit #1

Date 7 April, 2017 **Start** 9:10 **Stop** 11:00

Topic: To better understand the baking procedures and to see the oven.
Talk to the operator

Description: Interview with the principal operator about the general baking processe
and the limitation or difficulties experience with the manual oven.

Advance: Existed the loss of time due to regularly checking of the products,
baking time depended on the amount of trays in the oven since the fan is inadequate,
the air flow is not consistent. I noted that the oven design has one heat source (below).
The product, Panbon needs humidity, moisture during baking.

Limitation: There is no water line source install to the oven, a water bowl is used.
It does not have a individual gas shutoff valve.
it does not have a gauge or sensor to measure temperature.

Note: The baking temperature is between 145 to 160 C,

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit #2
Date 1 August, 2017 **Start** 11:00 **Stop** 14:30

Topic: Site visit to perform a diagnostic of the actual situation of the project

Description: To check the oven design and the existing layout so as to have a better understanding of the components required for the automatic control. Draw the existing circuits and take pictures.

Advance: Check the following circuits and components: overall design; the pilot system and gas flow system; the fan system including its electrical circuit the existing timer and buzzer; the lighting system; the existing safety system; had an inteview with the main operator about the preheating process and the baking processes.

Limitation: Unable to check the air flow of the fan system and the fan motor specifications and also the lighting system since it did not have bulbs in place.

Note:

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit #3
Date 11 October, 2017 **Start** 14:45 **Stop** 16:00

Topic: Site visit to prepare for installation of the sensors and actuators

Description: Review cabinet for installation of display monitor and LED indicators. Review bracket type needed for installation of the IR sensor, igniter and gas solenoid.

Advance: View possible locations to install the sensors and solenoids.

Limitation: Need to obtain brackets, screws and electric drill for installation.

Note: Will meet with a contractor who specialize in gas oven to assist with the installation gas solenoid and igniter.

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit 4
Date 13 October, 2017 **Start** 7:30 **Stop** 10:30

Topic: Review oven: Temperature vs time
1 -Time to heat up to 150 C. 2 - time to cool down to 135 C
3 - Controller cabinet maximum temperature

Description Note the temperature reading of the oven in reference to time heating and note the change temperature in reference to time once there is no flame.
Note the maximum temperature of the controller cabinet - where the Arduino will be installed.

Advance: The oven was heated to 250 C and the controller cabinet maximum temperature recorded is at 52.5 C
Over 180 temperature readings were recorded of the heating up and cooling down.

Limitation Reading were taken at 1 mins, 30 sec, 20 sec and 15 sec intervals, however the time intervals were not exact +/- 1 sec, also a digital thermometer was utilize and the reading were not exact (stable).

Note: The Arduino processor operating temperature is -55 to +125 C.
The current baking process does not use exact temperature, +/- 5 C.

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit 5th
Date 14-Oct **Start** 14:00 **Stop** 20:00

Topic: Install water line, along with modify control cabinet which includes the modification of the gas line.

Description Install 9 meters (1/2") PVC water pipes with a cut off valve and flexible metal water hose to the oven along with the water solenoid valve. Removal of the existing components within the control cabinet, cutting it and installing a new one along with a part of the old cabinet.

Advance: Completion of the water line along with the water solenoid was successful. The gas line was not fully install due to missing of copper fittings and the welding tools.

Limitation unable to complete the gas installation because to the lack of copper fittings and the lack of the welding equipment to weld the copper pipes.

Note:

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit 6th
Date 16-Oct **Start** 15:00 **Stop** 21:00

Topic: Complete installation of the control cabinet. Install the gas line along v the gas solenoid valve. Install the igniter, the IR sensor and the door micro switch.

Description Cutting, drilling and shaping the hole for the installation display. Welding and finalize the gas line including the installation of the gas solenoid. Also the relocating of the existing thermostat valve to the new control cabinet. Building frames and installing the IR sensor along with the electric igniter.

Advance: Complete the physical installation of the gas line, control cabinet, IR sensor and the electric igniter.

Limitation Need to buy screws and tie wrap.

Note:

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit 7th
Date 17-Oct **Start** 12:00 **Stop** 20:00

Topic: Install wiring to the various devices, IR sensor, electric igniter, fan, water solenoid valve, gas solenoid valve. Install the door micro switch and the temperature sensor.

Description Initiate the installation of wires between the different sensors and the Arduino and relay. Install the door micro switch and the temperature sensor within the oven.

Advance: Completion of the installation of the micro switch and temperature sensor and the wiring between devices to the Arduino and relay.

Limitation

Note:

Universidad Hispanoamericana
Final Project
Electronic Engineering

Student Giovanni Heredia

Project Site Visits Registry

Visit 8th
Date 23-Oct **Start** 14:00 **Stop** 20:30

Topic: Install the microprocessor Arduino, relay, temperature OpAmp circuit, Wi-Fi module and start testing

Description Install the microprocessor Arduino, relay, temperature OpAmp circuit, Wi-fi and finalized the wiring of the sensors and solenoid valves to relay and Arduino. Initial testing of the control system.

Advance: Completed all wiring and start testing.
adjust the hysteresis to 3 °C so that the relays oscillates at minimum 1 minute cycle

Limitation The Wi-fi did not work, gave errors, found that the wiring was missing a common ground between the devices and the power supplies. During testing, the preheat process and upon completion of its cycle - turn off gas and igniter instead of maintaining the temperature.

Note: Found that the programming codes had extra codes different from design specifications which was over looked and should have been only at end of baking process.

ANNEX 4: APPROVED LETTERS

UNIVERSIDAD HISPANOAMERICANA
SERIEDAD Y PRESTIGIO

CARTA DEL TUTOR

San José, 31 de Octubre del 2017

Señores
Departamento de Registro
Universidad Hispanoamericana

Estimado señor:

El estudiante Giovanni Heredia, cédula de identidad número P0164476, me ha presentado, para efectos de revisión y aprobación, el trabajo de investigación denominado *"To Develop an Automatic Control Oven System to Mitigate Economic Losses for Delights of the Caribbean Food S.A. in San Juan, Tibas for the period of June to October 2017"*, el cual ha elaborado para optar por el grado académico de Bachillerato.

En mi calidad de tutor, he verificado que se han hecho las correcciones indicadas durante el proceso de tutoría y he evaluado los aspectos relativos a la elaboración del problema, objetivos, justificación; antecedentes, marco teórico, marco metodológico, tabulación, análisis de datos; conclusiones y recomendaciones.

De los resultados obtenidos por el postulante, se obtiene la siguiente calificación:

Tabla 1 Calificación del proyecto

#	Rubro	% Teórico	% Asignado
a	Original del tema.	10	10
b	Cumplimiento de entrega de avances de avances.	20	20
c	Coherencia entre los objetivos, los instrumentos aplicados y los resultados de la investigación.	30	30
d	Relevancia de las conclusiones y recomendaciones.	20	20
e	Calidad, detalle del marco teórico.	20	20
Total:		100	100

En virtud de la calificación obtenida, se avala el traslado al proceso de lectura.

Atentamente,

Nombre del profesor: Mauricio Daniel Armas Sandí
Cédula de identidad: 1-1361-0843
Carné colegio profesional: IEL-22359

UNIVERSIDAD HISPANOAMERICANA
SERIEDAD Y PRESTIGIO

CARTA DEL LECTOR

San José, 8 de noviembre del 2017

Señores
Departamento de Registro
Universidad Hispanoamericana

Estimado señor:

El estudiante Giovanni Heredia, pasaporte número P0164476, me ha presentado, para efectos de revisión y aprobación, el trabajo de investigación denominado *"To develo pan automatic control oven system to mitigate economic losses for Delights of the Caribbean Food S.A."*, el cual ha elaborado para obtener su grado de Bachillerato.

He revisado y he hecho las observaciones relativas al contenido analizado, particularmente lo relativo a la coherencia entre el marco teórico y análisis de datos, la consistencia de los datos recopilados y la coherencia entre éstos y las conclusiones; asimismo, la aplicabilidad y originalidad de las recomendaciones, en términos de aporte de la investigación. He verificado que se han hecho las modificaciones correspondientes a las observaciones indicadas.

Por consiguiente, este trabajo cuenta con mi aval para ser presentado en la defensa pública.

Atentamente,

Jorge Villalobos Cascante
Cédula de identidad: 1-1185-0467
Carné colegio profesional: IEL-22656

UNIVERSIDAD HISPANOAMERICANA
SERIEDAD Y PRESTIGIO

CARTA DEL FILOLOGO

San Jose, 16 de noviembre de 2017

Estimado señor

El estudiante Giovanni Heredia, pasaporte número P0164476 me ha presentado, para efectos de corrección de estilo, el trabajo de investigación denominado: “To Develop an Automatic Control Oven System to Mitigate Economic Losses for Delights of the Caribbean Food S.A. in San Juan de Tibás for the period of June to October 2017”, el cual ha elaborado para optar por el grado de Bachiller en Ingeniería Electrónica.

He revisado, de acuerdo con los lineamientos de la corrección de estilos señalados por la universidad, los aspectos de estructura gramatical, acentuación, ortografía, puntuación y los vicios de dicción, que se trasladan al escrito y he verificado que se han realizado todas las correcciones indicadas en el documento.

Por consiguiente, doy fe que este trabajo se encuentra listo para ser presentado oficialmente a la universidad.

Atentamente,

M.Ed. Martha Eugenia Rojas Fernández

Cédula de identidad: 1-330-145

Carné Colegio de Licenciados y Profesores 75186

DECLARACIÓN JURADA

Yo Giovanni Valentino Heredia, mayor de edad, portador de la cédula de identidad número Belize passport #P0164476 egresado de la carrera de Ingeniería Electrónica de la Universidad Hispanoamericana, hago constar por medio de éste acto y debidamente apercibido y entendido de las penas y consecuencias con las que se castiga en el Código Penal el delito de perjurio, ante quienes se constituyen en el Tribunal Examinador de mi trabajo de tesis para optar por el título de Bachillerato, juro solemnemente que mi trabajo de investigación titulado: To Develop an Automatic Control Oven System to Mitigate Economic Losses for Delights of the Caribbean Food S.A. in San Juan, Tibás for the period of June to October 2017., es una obra original que ha respetado todo lo preceptuado por las Leyes Penales, así como la Ley de Derecho de Autor y Derecho Conexos número 6683 del 14 de octubre de 1982 y sus reformas, publicada en la Gaceta número 226 del 25 de noviembre de 1982; incluyendo el numeral 70 de dicha ley que advierte; articulo 70. Es permitido citar a un autor, transcribiendo los pasajes pertinentes siempre que éstos no sean tantos y seguidos, que puedan considerarse como una producción simulada y sustancial, que redunde en perjuicio del autor de la obra original. Asimismo, quedo advertido que la Universidad se reserva el derecho de protocolizar este documento ante Notario Público.

En fe de lo anterior, firmo en la ciudad de San José, a los 30 días del mes de Octubre del año dos mil 17.

Firma del estudiante

Cédula: P0164476

Belize passport.