

## Development of a Raspberry Pi-Based Automation System for an Induction-Heated Milk Pasteurizer



Evgeny Neverov<sup>\*ID</sup>, Igor Korotkiy<sup>ID</sup>, Aleksandr Vladimirov<sup>ID</sup>, Pavel Korotkih<sup>ID</sup>, Elena Nikolaeva<sup>ID</sup>,  
Andrey Porokhnov<sup>ID</sup>

Institute of Engineering Technologies, Kemerovo State University, Kemerovo, 650000, Russia

Corresponding Author Email: [neverov42@mail.ru](mailto:neverov42@mail.ru)

<https://doi.org/10.18280/jesa.560511>

### ABSTRACT

**Received:** 22 June 2023

**Revised:** 18 August 2023

**Accepted:** 16 October 2023

**Available online:** 31 October 2023

#### Keywords:

*automation, pasteurizer, induction, magnetic flux, food, polymer, induction heating, process control*

Induction heating has recently gained prominence as a preferred technology in industrial, medical, and household systems, owing to its superior advantages over traditional heating methods. The key to devising an energy-efficient methodology for heat treatment of food raw materials using an induction heated pasteurization tank lies in the effectiveness of the process automation system. Addressing the automation issue pertaining to a pasteurization induction unit, the authors explore the capabilities of AVR and ARM microcontrollers. These are employed to establish a comprehensive development environment for managing, constructing, testing, and deploying an embedded microcontroller application. Utilizing the Thonny development environment, Python programming language version 3 is implemented to write and execute programs on the Raspberry Pi microcomputer. This microcomputer is wielded to regulate the operation of the pasteurizer prototype and its various associated peripherals, including sensors that measure diverse milk pasteurization parameters. Throughout the operation of the unit, all components maintain communication with the controller. The control panel facilitates the management of the installation and renders data output. As a result of this study, a control program and algorithm were developed for a prototype of an induction-type installation, empowering control and surveillance of milk pasteurization processes.

## 1. INTRODUCTION

The traditional methods used to sterilize liquid food products involve thermal pasteurization, a technology that has been developed over decades and is an efficient and cost-effective processing method in the food industry. These methods are mainly based on convective and conductive heat transfer, in which heat energy is transferred from the hot medium to the cooler product, resulting in large temperature gradients.

An alternative method of thermal pasteurization is magnetic induction heating, which is widely used in industry, households, and medicine and demonstrates the advantages of high energy efficiency, fast heating, safety, cleanliness, and precise control of the temperature of heated raw materials [1-7].

As a non-contact method of direct material heating, it is based on Joule heating and magnetic hysteresis. These two mechanisms cause heating through uneven current distribution in the area of the conducting material due to the formation of an alternating magnetic field.

Scientific works [8-15] describe universal induction systems applicable in the food industry and domestic use. In such systems, the main elements are induction coils, a high-voltage generator power supply, a microcontroller (MC), and an oscillating circuit.

Automation is of great importance in various technological processes. In particular, it exempts humans from routine tasks, increasing their productivity [16-20]. In addition, automatic

devices reduce human involvement in a given process, decreasing the probability of error in determining various values. Automatic systems respond much more accurately and promptly to changes in certain parameters than humans do. Automation of technological processes using a PID controller based on a microcomputer can be not simply useful but indispensable in cases where a large number of calculations must be performed in a short time [21-23]. Another development is the use of a PID controller to control the milk pasteurization process [24-28].

Milk pasteurization involves heating it to a set temperature for a given time to get rid of pathogens and prevent their proliferation. Therefore, high-precision temperature control is required. The article [24] is devoted to the use of PID (Proportional-Integral-Derivative) and MPC (Model Predictive Control) controllers to control milk pasteurization processes and their comparison. The authors found that MPC controllers are more suitable for controlling complex technological processes than PIDs.

The article [25] develops a pasteurizer control system, which is a PID controller based on Atmega16, to control the drives of electric heating elements that receive input data from PT 100 temperature sensors. Test results were obtained when heating 10 l of milk from 30°C to 65°C, using Ziegler Nichols tuning method,  $K_p = 18$ ,  $K_i = 0.45$ , and  $K_d = 180$ . The graph shows rise time ( $T_r$ ) = 2,400 s, delay time ( $T_d$ ) = 140 s, peak time ( $T_p$ ) = 2,700 s, dead time ( $T_s$ ) = 3,700 s, maximum rise ( $M_p$ ) = 2°C, and steady state error 1.5%.

Based on the described articles, we developed a software

and hardware complex based on a Raspberry Pi 4 B+ microcomputer and an induction heating installation, which allows pasteurization of various products in automatic mode.

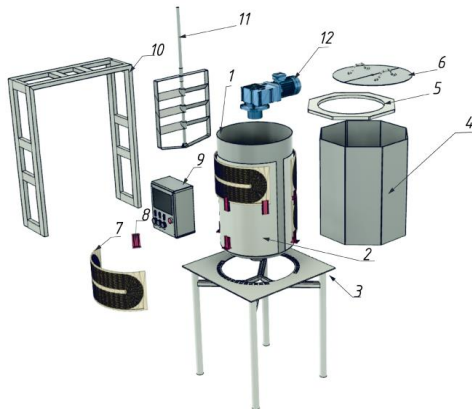
In contrast to manual or automatic control, PID control allows for increasing the quality of pasteurized milk and destroying the bacteria in raw milk with a higher probability.

The study aims to develop an automation system for the thermal treatment of food raw materials using an induction heated pasteurization tank.

In this framework, research objectives are to develop an application in the form of a window interface to control the induction unit under the control of the microcomputer Raspberry Pi and to select peripheral devices to control the unit, including actuators, sensors, and libraries for them in the Python programming language environment.

## 2. MATERIALS AND METHODS

The external appearance of the separated scheme of the developed prototype of the induction heating device is shown in Figure 1. The algorithm of device operation is as follows: from the control panel, pasteurization temperature values, stirrer rotation speed, induction heating power, and its duration are set; the voltage required for the power supply of main elements of the unit and the electric motor is set by the voltage regulator.



**Figure 1.** Distributed diagram of the pasteurizer prototype. 1. Tank, 2. Tank covers, 3. Base table, 4. Hull, 5. Upper part of the hull, 6. Lid, 7. Inductor frame, 8. Inductor frame mounts, 9. Power supply unit housing, 10. Mounting rack, 11. Stirrer, 12. Electric motor

In solving the problem of automation of the induction pasteurization unit, we analyzed the possibilities of AVR and ARM MCs because they are widely used in the creation of electronics due to their versatility, wide range of functionality, and high technical characteristics. The task of automating the pasteurizer operation was addressed with the use of these MCs.

The first solution was to use the AVR Atmega2560 MC. This controller operated the Arduino Mega 2560 board, which allowed it to automate the pasteurization process.

The second solution was to use the STM32F429 MC, the core of which is an ARM Cortex-M4. The STM32F429 was chosen instead of its predecessor AVR Atmega2560 due to its considerable advantages in features, such as higher controller frequency (180 MHz vs 16 MHz), more program (2,048 KB

vs 256 KB), and RAM (256 KB vs 4 KB), and the ability to select the supply voltage. This controller controls the STM32F429I-DISCOVERY debug board.

The AVR MC includes a processor, programmable input/output peripherals, and memory. AVR MCs are used for digital control of any kind of electrical, automotive, or mechanical system, industrial plants, various electronic devices, etc. These MCs are available as 8-, 16- and 32-bit integrated circuits. The most common AVR MCs are ATmega8, ATmega16, ATmega32, and Atmega328.

The AVR Atmega32 MC has EEPROM-1 KB, SRAM-2 KB, 32 general-purpose registers, 23 general-purpose input/outputs, a flexible timer or counter-3 including compare modes, serial programmable USART, internal and external interrupts, an SPI serial port, two-wire serial interface, a 10-bit 6-channel analog-to-digital converter, and a programmable timer including an internal oscillator and five programmable power save modes.

In this study, we used AVR Studio 4 – an Integrated Development Environment (IDE) designed for writing and debugging programs for AVR MCs. AVR Studio 4 contains an assembler and a simulator. In addition, this development environment supports tools for the AVR such as ICE50, ICE40, JTAGICE, ICE200, STK500/501/502, and AVRISP. AVR Studio supports COFF as an output format for symbolic debugging. Other third-party software tools can also be configured to work with AVR Studio.

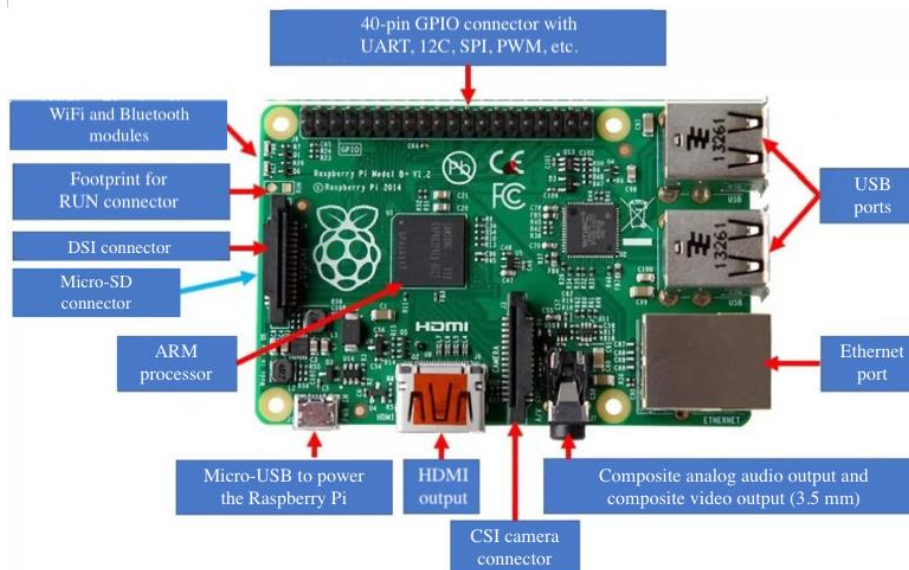
This processor (Figure 2) has a Reduced Instruction Set Computer (RISC) architecture. RISC is designed to process simple instructions from a limited command set. The simplicity of the instructions limits the number of tasks that the processor can perform but increases its performance. One command in such processors is executed in one machine cycle. In addition, processors of this architecture contain a relatively small number of transistors, due to which the processor consumes less power.

The ease of instructions and their limited number is the reason for some disadvantages of the RISC architecture. The limited number of instructions leads to their greater number, i.e., a program to perform the same task in the RISC architecture is 30% longer than a similar program in Complex Instruction Set Computer (CISC) architecture. Longer program code results in longer memory load time.

ISOC SEGGER Embedded Studio is a full-featured development environment for controlling, building, testing, and deploying an embedded ARM MC application in all stages of development: from a project generator that will create a project for most ARM MCs, a powerful project manager and source code editor including a C/C++ compiler and built-in debugger with an advanced debugging information window and direct integration of the J-LINK hardware debugger, to the features of version control for automatic application deployment.

The SEGGER Embedded Studio development environment is available for Windows, Linux, and MacOSX operating systems. SEGGER Embedded Studio has a cross-compiler for ARM MCs. The interface of the development environment is the same on all platforms to ensure the highest performance regardless of operating system preference.

The source code editor not only supports user-defined syntax highlighting, automatic code indentation, and bracket highlighting, but also provides code auto-completion for symbols, functions, and keywords in the application, as well as customizable code comment templates.

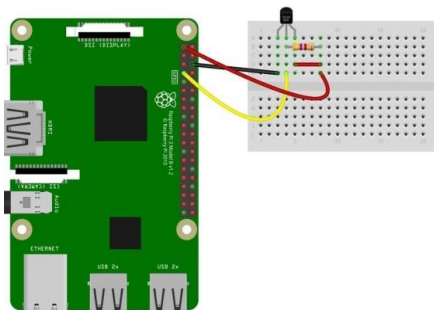


**Figure 2.** Raspberry Pi 4B+ microcomputer

Raspberry Pi 4 Model B is a miniature single-board computer. This platform has access to the Internet (with no additional device required) and its own operating system, which makes it possible to develop applications. Thus, this microcomputer can become a control element for different devices. Below we present an image of the Raspberry Pi 4B (Figure 3) with its constituent parts.

Raspberry Pi 4 reads pasteurization parameters (thermal treatment product, volume, inductor power) and controls temperature sensors, inductor, and stirrer. After starting the heating process, based on the value of the read temperature of the pasteurization product, the microcomputer stops heating, providing a constant temperature (holding temperature) for a certain time (holding time) for a particular product. During the entire pasteurization process, the microcomputer reads the temperatures of the transistor and inductor, and if the permissible temperatures are exceeded, Raspberry Pi 4 makes an emergency stop of the pasteurization process. After the pasteurization process is completed, the microcomputer automatically generates a report on the pasteurization session, saving the data in the form of CSV files and graphs and formatting the received data in the form of a report in docx format.

The microcomputer was used to control the operation of the pasteurizer prototype and its various peripherals. This includes sensors that measure various milk pasteurization parameters.



**Figure 3.** Wiring diagram for connecting DS18B20 to Raspberry Pi 4B

### 3. RESULTS

The first step is to develop a control algorithm with measurements of pasteurization temperature. A DS18B20 digital temperature sensor can be used for this assessment (Figure 3).

The sensor has the following specifications [29-31]:

- Power supply range: 3 V-5.5 V;
- Communication protocol: 1-Wire;
- Wiring method: direct/single line with parasitic power supply;
- Temperature conversion resolution: 9 bits – 12 bits;
- Operating range temperature:  $-55^{\circ}\text{C} - +125^{\circ}\text{C}$ ;
- Temperature measurement time at a maximum resolution of 12 bits: 750 ms;
- Type of indexing on the 1-Wire line: unique 64-bit ID.

The script written in the Python programming language allows setting the temperature measurement time in seconds, reading temperature values in degrees Celsius once per second, and writing temperature and time values to a CSV (Comma Separated Values) file.

All elements communicate with the controller during the operation of the unit. The unit is controlled and the data output is performed using the control panel. Before the beginning of the operation, the milk is poured into the pasteurization tank (from 10 l to 75 l).

The algorithm of the installation operation is shown in Figure 4.

After the installation is switched on, it is initialized by checking the basic parameters that do not correspond to the preset values and will not allow the installation to perform its work. These parameters are the current and power supply voltage, as well as the stirring speed. If these parameters do not correspond to the intended values, the unit is restarted. This process is not reflected in the algorithm, as it happens by default. The milk pasteurization process can be regarded as an experiment, so after activation and initialization, it is necessary to enter data about the particular experiment: username, date, duration of the experiment, and the name of the pasteurization product.

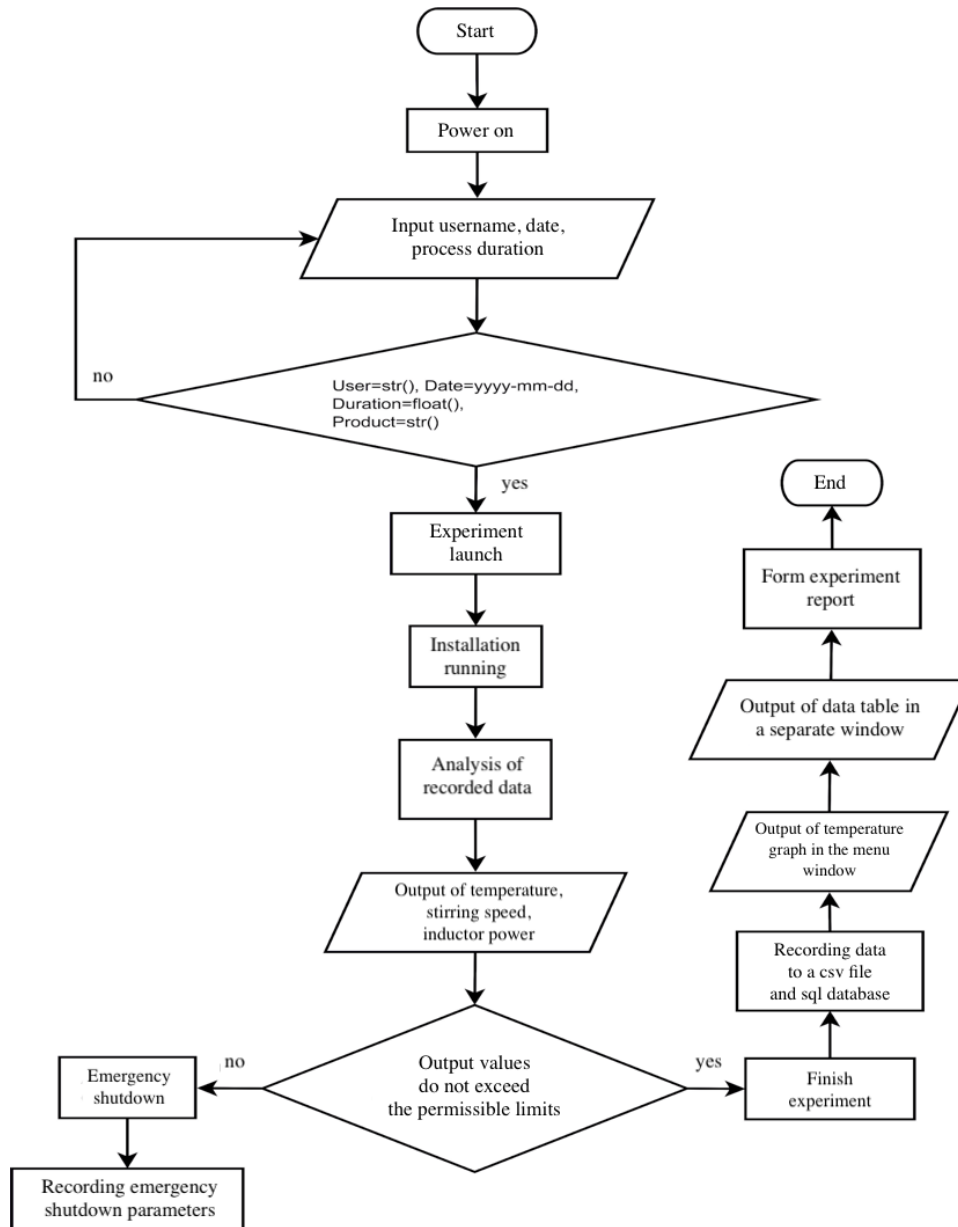


Figure 4. Algorithm of the experimental installation

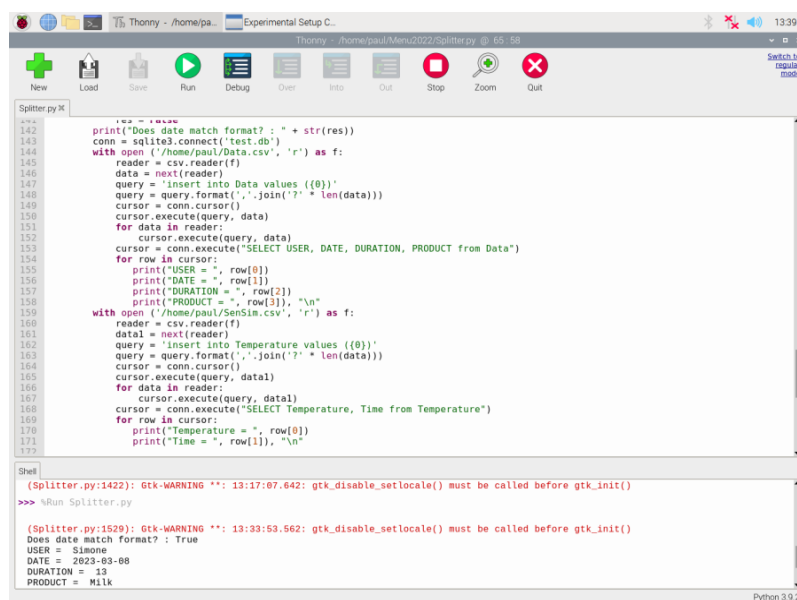


Figure 5. The Python Thonny development environment



The entered data are checked for compliance with the preset formats, and in case of inconsistency, re-entry of the data is requested. If the entered data are correct, the experiment is started, and the data received by the sensors are constantly analyzed and displayed on the screen (this can be an LCD of the control panel or a computer monitor). If this data does not correspond to the permissible values (e.g., exceeding the temperature), an emergency occurs, in which the operation of the installation is abruptly interrupted, and the data read at that moment are written to the file and/or the database. If during the whole experiment, all measured data do not exceed the permissible limits, the experiment is terminated in the normal mode, the data are written into files, and some data are displayed on the screen. From the recorded and entered data, a report on the experiment is formed.

To write and run Python programs on the Raspberry Pi microcomputer, we use the Thonny development environment, which is designed for Python version 3. This development environment features Read-Evaluate-Print-Loop (REPL) programming. This feature allows entering and executing Python language commands without using `print()`. Figure 5 shows the window of this development environment.

This development environment allows viewing the variables defined in the script and their values. This can be done by clicking the "View" button and selecting "Variables" in the drop-down menu. The window that opens will display a table with variable names and values. Another characteristic of variables that can be displayed is the address.

Another function of this development environment is the debugger. The debugger runs the program step by step through the structural parts of the code. The debugger detects errors in the part of the code that is currently being executed. It can be started by pressing the key combination `Ctrl + F5`.

Apart from these functions, Thonny has a step-by-step evaluation of expressions. When using the small-step debugger, the user can see the evaluation of an expression in terms of syntax contradiction and compactness.

One more useful function of this development environment is to view the calls of functions defined in the code. When this function is run, the program runs in steps, and if the next step includes the function to be called, a window opens with the code of this function and local variables (the window displays a table with the names and values of local variables).

In addition, the development environment detects and color-codes syntax errors. This feature works by default and does not require any action from the user to run it.

In addition to error highlighting, the development environment can highlight different variables. This function helps to distinguish local variables from global variables in the code and also helps to find typos in variable names.

To reduce code typing time, Thonny has an auto-completion feature: when a word is typed, the development environment gives options for completing that word in a drop-down list.

After that, the program code describes the function that inserts the image into the window (`wx.Image`). The parameters of this function are the address of the image on the computer and its type. It is followed by the function placing the image in the window (`wx.StaticBitmap`). It specifies the coordinates of the upper left corner of the image relative to the upper left corner of the window. The coordinate system is such that the vertical axis points down and the horizontal axis points to the right, so both coordinates are positive.

The definition of the location of the image to display the temperature vs. time graph at the end of the experiment is

followed by setting the text. This is performed by the `wx.StaticText` function. Its parameters are the location of the text (in this case, the window panel) and its content: the label (for example, "User"). There are many such elements of the "static text" type in the window: "Enter experiment parameters", "Duration, s", "Pasteurization process monitoring", etc.

The `sizer.Add` function is used to position any element on the window panel. It determines the position of the element on the panel relative to other elements. The arguments of this function are the name of the variable to which the function of creating this or that element is assigned; the position of the element in the invisible window table (for example, (3,1): first row, third column), row and column numbering starts from zero; a flag responsible for aligning or stretching the element by the column width; and the width of the border.

In addition to static text elements, the window contains input fields. They are created by the function `wx.TextCtrl`, the argument of which is the location of the element in the window, in this case, the panel of the window.

Buttons with text on them are used to trigger different events. Buttons are created with the function `wx.Button`. The arguments of this function are the same as those of the `wx.StaticText` function: location and content. For example, (panel, label = "Enter parameters").

Each button is bound to an event. This is done with the `Bind` function, whose parameters are event type (in this case, button press event, `wx.EVT_BUTTON`), event name (for example, `self.on_press`), and button name (one of the buttons in the menu is called `self.buttonParEnt`).

The first button that the user clicks after entering the experiment parameters ("User", "Date (yyyy-mm-dd)", "Durection, s", "Pasteurized product") is the button labeled "Enter parameters". It triggers an event, which is named `on_press2(self, event)` in the code and is declared as a function with the command `def`. If all the parameters are entered correctly, the program creates a "Data.csv" file and writes the entered data to it, separated by commas without spaces, using the function `f.write`. In the "Data.csv" file, the comma is the separator between the values, it separates some columns from others. Hence, the file format is called CSV. In addition to the entry of the input data into the "Data.csv" file, the same data is written to the previously created SQL database "Data" with the help of the `query` and `conn.exequte` functions. Then, to show the user that this sequence of operations has been completed, a small window is displayed using the same function as the window with the message about the incorrectly entered parameters. The panel of this window displays the text: "Parameters entered, saved to the Data.csv file, and written to the database". At the bottom of this window there is also an "ok" button (it is positioned in such windows by default when using the `wx.MessageBox` function).

After entering the parameters of the experiment, these parameters can be viewed in tabular form by pressing the "Show entered data" button.

Once all the parameters of the experiment are entered and viewed in the table, the experiment can be started. The experiment is started by pressing the "Launch experiment" button. Pressing this button starts the `on_press` event. The code of this event contains the opening of the file "Data.csv" and reading the value of the duration of temperature measurement in seconds from the third column of the first row (this CSV file has no headers). The value read is assigned to the variable `n`. This is followed by a loop to set two other variables: the first

element of the arithmetic progression and the difference of the next two elements. These are equal to zero and one, respectively. After these variables are set, the "for" loop is started, where the variable "moment" is set, having the value of the moment in seconds. It is equated to the sum of its own and the arithmetic difference. The temperature value as well as the inductor power and stirrer speed values are equated to the sine of the time value with different numbers added to the sine based on the operating values for milk pasteurization. This is how the data from the different sensors is simulated.

After these variables are set in the event code, the SetLabel functions follow, which set new values to the static texts responsible for outputting temperature, motor speed, inductor power, and time. Only the values of the "measured" (generated) values are new in these set values, the text remains the same. Following the setting of values of static texts the file "SenSim.csv" is opened and the values of temperature and time are written to it, separated by commas. The recording is carried out in mode "a+", due to which the program does not overwrite the file, but adds what is currently being written to what has already been written. With the function f.write() the program first writes the temperature value, puts a comma, then writes the time value, and then does a line feed (in Python the line feed is written as "\n"). Next, the whole content of the

window is updated using the self.Update() function, which also updates the values of the "measured" variables. The update is followed by a delay of one second, which is set by the function time.sleep(1), where one in brackets means one second. In this case, the user does not notice the updating of the entire content of the window, and only the updating of the values of temperature, speed of rotation of the electric motor, inductor power, and time is visible.

When the set time expires, the ChartBuilder function from the program, which is written as part of this work, runs and is imported into the main menu program with the "import" command, as libraries are imported. This program plots the graph of time-temperature dependence, taking values from the file "SenSim.csv" using the library matplotlib.pyplot. The x.append and y.append functions are used to set the values of the horizontal and vertical axes, respectively. Next, the function plt.title() sets the header that is placed above the graph "Dependence of temperature on time"; and the functions plt.xlabel() and plt.ylabel() set the headers of the graph axes: time and temperature, respectively. After the graph is plotted, it is saved in PNG image format using the plt.savefig() function, whose argument is the location of the graphical file on the computer. Figure 6 shows the menu window after the start of the experiment and before its completion.

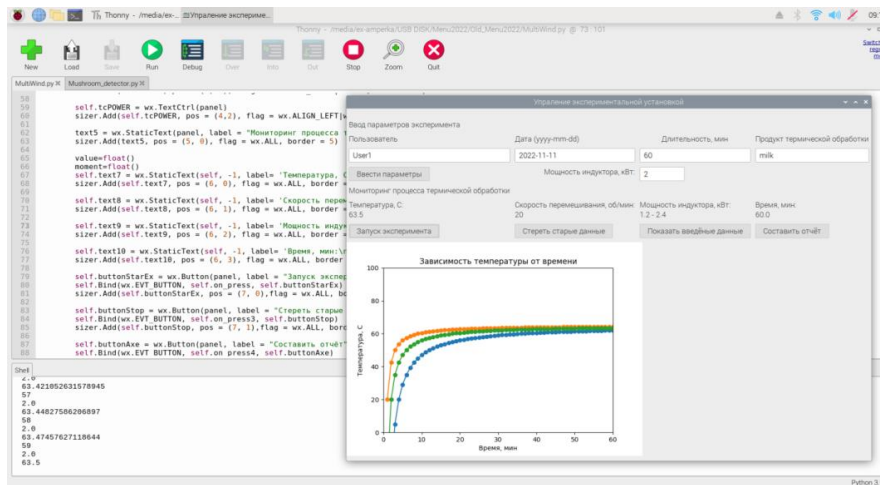


Figure 6. Experiment control menu during the measurement of different values

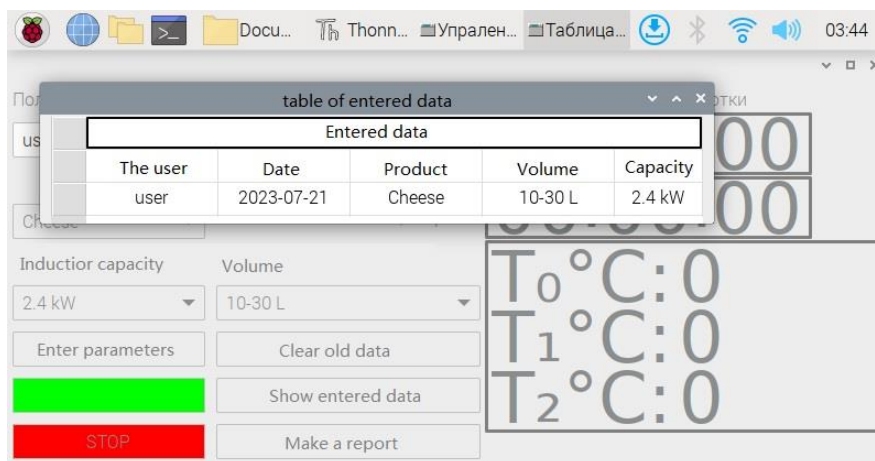
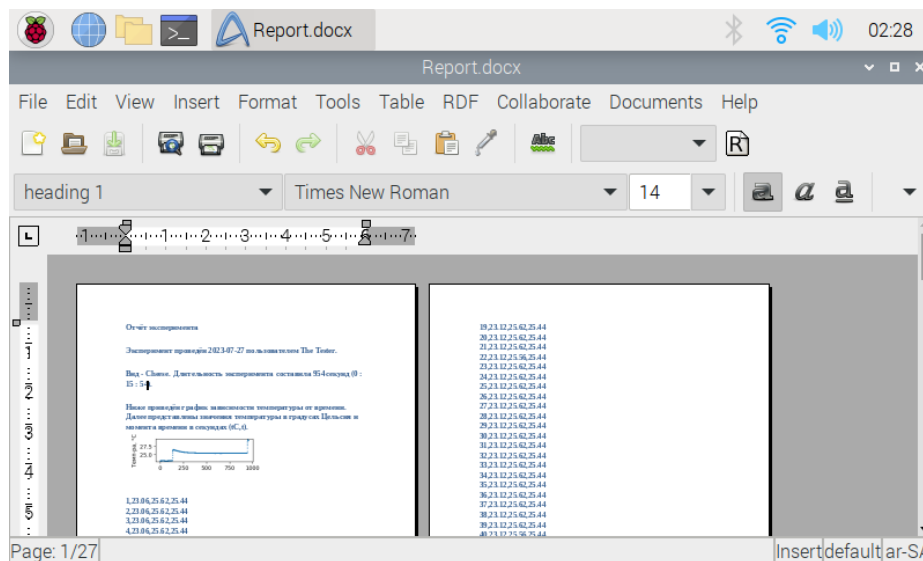


Figure 7. Table of entered data in the program window



**Figure 8.** Automatically generated pasteurization session report

After the end of measurements, which occurs after the time set by the user, a graph of the dependence of pasteurization temperature on time is displayed in the menu window.

To process all the entered and measured data of the experiment into a single report, there is a corresponding event in the menu program. This event is defined by the function "on\_press4(self, event)". First, the "Data.csv" file is opened, then several variables are assigned to the values of the cells in the table of this CSV file (the file contains one row and five columns). An example table is shown in Figure 7.

After that, the variable responsible for converting seconds to minutes with rounding to the thousandths is set using the round() function, the first argument being the value to be rounded, and the second argument being the number of digits after the decimal point.

This is followed by the creation of a docx document using the docx.Document function (Figure 8). Then the doc.add\_heading function creates headers with text with values written into it, read from the "Data.csv" file. All these texts have the format of the first-level header. After creating these texts, the temperature vs. time graph should be added as a PNG graphical image to this document using the doc.add\_picture function.

The addition of the image to the document is followed by opening the file "SenSim.csv" in read mode, reading all of its contents using read(), and writing this content to the document using doc.add\_heading. After all these operations, the document is saved as Report.docx using doc.save(), and the window with the message appears: "The report is compiled and saved as Report.docx" as the wx.MessageBox function is applied.

#### 4. DISCUSSION

Thus, we developed an automation system for a raw milk pasteurization installation, the principle of which is based on the influence of an alternating magnetic field with induction heating. In this way the time to enter the operating mode is reduced by a factor of 2, no auxiliary equipment is needed to heat the coolant, heat losses into the environment are reduced, and operating costs are cut down compared to electric heating with heating elements [32, 33]. The technology is

environmentally friendly and the installation is smaller in size compared to milk heat treatment plants using intermediate coolants (steam, water) [34]. Using a system of inductors, it is possible to heat only a certain local area, which reduces power consumption by 30%. The efficiency of such units increases by up to 92-95%. By the type of working cycle, the pasteurizer is periodical [34]. The type of pasteurization is continuous pasteurization [34], the milk is heated to 63-65°C and kept at this temperature for 30 minutes.

The proposed automation system provides a 30% increase in energy efficiency compared to manual control of this installation. A microcomputer was used to control the operation of the pasteurizer prototype and its various peripherals. During the operation of the installation, all elements exchange data with the controller. The installation control and data output are performed using the control panel.

#### 5. CONCLUSION

This study provides a rationale for the use of the necessary automation controls for the prototype induction milk pasteurization unit we developed. An algorithm was developed for control and PID regulation of the pasteurization process. Optimal automated control of this process is best carried out based on a Raspberry Pi microcomputer, the use of which in the future will allow to develop a control system using artificial intelligence and neural networks to control and monitor the process.

Due to the developed electromagnetic field control system, the heating process can be precisely controlled, which allows easily heating a small area to the required temperature without affecting the surrounding area. The electromagnetic field created by an induction heater can heat the material and products in contact with it very quickly. In addition, induction heating can be turned on and off very quickly, making it ideal for applications that require precise heating, such as pasteurization, where overheating compromises the technology.

The information in this article can be used to expand information about the possibility of introducing the Raspberry Pi microcomputer and induction heating into food production.

## REFERENCES

- [1] Park, S.M., Jang, E., Park, J.S., Kim, J.H., Choi, J.H., Lee, B.K. (2020). Optimal design of high-frequency induction heating apparatus for wafer cleaning equipment using superheated steam. *Energies*, 13(23): 6196. <https://doi.org/10.3390/en13236196>
- [2] Vairamohan, B., Bran, I., De Bellefon, G.M. (2011). What's new in electrotechnologies for industrial process heating? In *ACEEE Summer Study on Energy Efficiency in Industry*, pp. 1-156.
- [3] Solovev, S.V., Morozov, V.V., Radkevich, E.V., Kartashov, L.P., Makarovskaia, Z.V. (2019). Obosnovanie parametrov konstruktivnykh elementov induktsionnogo nagrevatel'ia dlia pasterizatsii moloka v potoke [Justification of parameters of design elements for an induction heater for pasteurizing milk in a flow]. *Izvestia Orenburg State Agrarian University*, 1(75): 100-103.
- [4] Sweeney, M., Dols, J., Fortenbery, B., Sharp, F. (2014). Induction cooking technology design and assessment. In *ACEEE Summer Study on Energy Efficiency in Buildings*, August 17-22, 2014. American Council for an Energy-Efficient Economy, pp. 370-379.
- [5] Makarova, G.V., Solovev, S.V., Shilin, V.A. (2013). Patent na poleznuiu model RF 137709 U1. Induktsionnyi nagrevatel zhidkosti s vyderzhivatelem [Useful model patent RF 137709 U1. Liquid induction heater with a hold tube]. *Bulletin No. 6. Federal Service for Intellectual Property of the Russian Federation*.
- [6] Lucia, O., Maussion, P., Dede, E., Burdío, J.M. (2013). Induction heating technology and its applications: Past developments, current technology, and future challenges. *IEEE Transactions on Industrial Electronics*, 61(5): 2509-2520. <http://doi.org/10.1109/TIE.2013.2281162>
- [7] Razak, R.A., Ibrahim, N.M., Rahman, A.S.F., Fayzul, M., Azizan, M.M., Hashim, U., Basir, I. (2021). Induction heating as cleaner alternative approach in food processing industry. *Journal of Physics: Conference Series*, 1878: 012053. <http://doi.org/10.1088/1742-6596/1878/1/012053>
- [8] Wang, G., Wan, Z., Yang, X. (2020). Induction heating by magnetic microbeads for pasteurization of liquid whole eggs. *Journal of Food Engineering*, 284: 110079. <http://doi.org/10.1016/j.jfoodeng.2020.110079>
- [9] Vladimirov, A.A., Beletskaiia, M.E., Beliakov, E.A., Tupitsyn, A.V., Prosekov, A.I. (2020). Patent RF 2741359 Ustroistvo induktsionnogo nagreva barabana kontaktnoi sushki [Patent RF 2741359 Device for induction heating of the contact drum dryer] *Bulletin No. 3. Federal Service for Intellectual Property of the Russian Federation*.
- [10] Kaunova, L.M. (2022). Obosnovanie konstruktivnykh parametrov induktsionnogo nagrevatel'ia dlia pasterizatsii moloka [Justification of design parameters of induction heater for milk pasteurization]. *AgroEkoInzheneriia*, 3(112): 141-148. <https://doi.org/10.24412/2713-2641-2022-3112-141-148>
- [11] Sanz, F., Sagües, C., Llorente, S. (2015). Induction heating appliance with a mobile double-coil inductor. *IEEE Transactions on Industry Applications*, 51(3): 1945-1952. <http://doi.org/10.1109/TIA.2014.2367136>
- [12] Alotto, A., Spagnolo, A., Paya, B. (2011). Particle swarm optimization of a multi-coil transverse flux induction heating system. *IEEE Transactions on Magnetics*, 47(5): 1270-1273. <http://doi.org/10.1109/TMAG.2010.2086439>
- [13] Lubin, T., Netter, D., Leveque, J., Rezzoug, A. (2009). Induction heating of aluminum billets subjected to a strong rotating magnetic field produced by superconducting windings. *IEEE Transactions on Magnetics*, 45(5): 2118-2127. <http://doi.org/10.1109/TMAG.2009.2014461>
- [14] Serrano, J., Acero, J., Lope, I., Carretero, C., Burdío, J.M. (2018). A flexible cooking zone composed of partially overlapped inductors. *IEEE Transactions on Industrial Electronics*, 65(10): 7762-7771. <http://doi.org/10.1109/TIE.2018.2801815>
- [15] Meng, L.C., Cheng, K.W.E., Chan, K.W., Lu, Y. (2012). Variable turn pitch coils design for heating performance enhancement of commercial induction cooker. *IET Power Electronics*, 5(1): 134-141. <http://doi.org/10.1049/iet-pel.2010.0388>
- [16] Glotko, A., Sycheva, I., Permyakova, E., Danilchik, T., Usenko, V. (2018). Prospects of production of ecologically clean agricultural products in the transboundary territories of the Great Altai. *Journal of Environmental Management and Tourism*, 8(7): 1361-1372. [http://doi.org/10.14505/jemt.v8.7\(23\).05](http://doi.org/10.14505/jemt.v8.7(23).05)
- [17] Bekezhanov, D., Rzabay, A., Nesipbaev, O., Kopbassarova, F., Halibiyati, H. (2022). Legal significance of digitalization of environmental information in ensuring environmental safety. *Journal of Environmental Management and Tourism*, 13(3): 656-664. [http://doi.org/10.14505/jemt.v13.3\(59\).06](http://doi.org/10.14505/jemt.v13.3(59).06)
- [18] Bekezhanov, D., Kopbassarova, G., Rzabay, A., Kozhantayeva, Z., Nessipbayeva, I., Aktymbayev, K. (2021). Environmental and legal regulation of digitalization of environmental protection. *Journal of Environmental Management and Tourism*, 12(7): 1941-1950. [https://doi.org/10.14505/jemt.12.7\(55\).19](https://doi.org/10.14505/jemt.12.7(55).19)
- [19] Nurgaliyeva, A.M., Kazbekova, Z.S., Bokenchina, L.K., Bekniyazova, D., Bokenchin, K.K. (2022). Opportunities for using green bonds to finance environmental projects in developing countries: Experience of the Republic of Kazakhstan. *Journal of Environmental Management and Tourism*, 13(7): 1918-1926. [http://doi.org/10.14505/jemt.v13.7\(63\).12](http://doi.org/10.14505/jemt.v13.7(63).12)
- [20] Rzabay, A., Seriyev, B., Beisov, E., Kopbassarova, G., Kurmanbayeva, D. (2022). Environmental and legal regulation of radioactive pollution management. *Journal of Environmental Management and Tourism*, 13(3): 633-642. [http://doi.org/10.14505/jemt.v13.3\(59\).03](http://doi.org/10.14505/jemt.v13.3(59).03)
- [21] Septiani, N.I., Bayusari, I., Caroline, Haiyunnisa, T., Suprpto, B.Y. (2017). Optimization of PID control parameters with genetic algorithm plus fuzzy logic in stirred tank heater temperature control process. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Palembang, Indonesia, IEEE, pp. 61-66. <https://doi.org/10.1109/ICECOS.2017.8167167>
- [22] Jibril, M., Tadese, M., Tadese, E.A. (2020). Temperature control of stirred tank heater using optimal control technique. *ScienceOpen Preprints*. <https://doi.org/10.14293/S2199-1006.1.SOR-PPVT203.v1>
- [23] Divya Priya, A.V., TamilSelvan, G.M., Rajesh, T. (2018). Real time implementation of fuzzy based PID



- controller tuning for Continuous Stirred Tank Heater (CSTH) process. *Journal of Fuzzy Systems*, 10(3): 78-81.
- [24] Alamirew, T., Balaji, V., Gabbeye, N. (2017). Comparison of PID controller with model predictive controller for milk pasteurization process. *Bulletin of Electrical Engineering and Informatics*, 6(1): 24-35. <https://doi.org/10.11591/eei.v6i1.575>
- [25] Singgih, H. (2017). Optimization of PID controller in temperature control system processes pasteurization of milk. *American Journal of Engineering Research*, 6(9): 175-187.
- [26] Neverov, E.N., Korotkiy, I.A., Korotkih, P.S. (2022). *Osnovy Avtomatizirovannogo Proektirovaniia [Fundamentals of Computer-Aided Design]. Training Manual*. KemSU. Kemerovo, p.108.
- [27] Korotkiy, I.A., Neverov, E.N., Vladimirov, A.A., Neverova, O.A., Proskuriakova, L.A. (2021). *Opredelenie teplofizicheskikh kharakteristik tropicheskikh fruktov dlia ikh ispolzovaniia pri proizvodstve sukhikh molochnykh produktov [Determination of thermophysical characteristics of tropical fruits for their use in the production of dried dairy products]*. *Food Processing: Techniques and Technology*, 51(2): 220-231.
- [28] Korotkiy, I.A., Korotkiy, E.V., Korotkaya, E.V., Neverov, E.N., Fedorov, D.E., Gushchin, A.A. (2020). *Razrabotka nizkotemperaturnoi tekhnologii izvlecheniia belka iz tvorozhnoi syvorotki [The development of low temperature technology for extracting protein from curd whey]*. *Bulletin of KSAU*, 2(155): 148-154. <https://doi.org/10.36718/1819-4036-2020-2-148-154>
- [29] Razooqi, R.A., Hassan, H.J., Saheb, G.M.A. (2022). Selection algorithm for reducing IoT service delay in the smart factory. *Journal Européen des Systèmes Automatisés*, 55(3): 419-426. <https://doi.org/10.18280/jesa.550315>
- [30] Danladi, M.S., Baykara, M. (2022). Design and implementation of temperature and humidity monitoring system using LPWAN technology. *Ingénierie des Systèmes d'Information*, 27(4): 521-529. <https://doi.org/10.18280/isi.270401>
- [31] Escandón-Panchana, P., Morante-Carballo, F., Herrera-Franco, G., Rodríguez, H., Carvajal, F. (2022). Fluid level measurement system in oil storage. *Python, Lab-based scale. Mathematical Modelling of Engineering Problems*, 9(3): 787-795. <https://doi.org/10.18280/mmep.090327>
- [32] Başaran, A., Yılmaz, T., Azgin, S.T., Civi, C. (2021). Comparison of drinking milk production with conventional and novel inductive heating in pasteurization in terms of energetic, exergetic, economic and environmental aspects. *Journal of Cleaner Production*, 317: 128280. <https://doi.org/10.1016/j.jclepro.2021.128280>
- [33] Başaran, A., Yılmaz, T., Civi, C. (2018). Application of inductive forced heating as a new approach to food industry heat exchangers. *Journal of Thermal Analysis and Calorimetry*, 134: 2265-2274. <https://doi.org/10.1007/s10973-018-7250-7>
- [34] Bredikhin, S.A. (2021). *Tekhnologicheskoye oborudovaniye pererabotki moloka: Uchebnoye posobiye dlya vuzov [Technological equipment for milk processing: A textbook for universities]*. Saint Petersburg: Lan.