ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Smart Vacuumed Rice Dispenser

Team #21

WANG SHINING
(sw63@illinois.edu)
LIU YUYANG
(yuyang11@illinois.edu)
ZHAO GAONING
(gaoning2@illinois.edu)
YANG ZIXIN
(zixin7@illinois.edu)

TA: Cui Xiaotong

May 15, 2026

Abstract

This report presents the design and verification of a Smart Vacuumed Rice Dispenser that combines automatic rice dispensing with pressure-based storage control. The system is designed to improve user convenience while reducing rice exposure to air during storage. The prototype uses an Arduino Nano as the main controller, a rotary encoder and OLED display for user interaction, an HX711 load-cell system for real-time weight measurement, and a BMP280 pressure sensor for internal pressure monitoring. Rice is dispensed through an Archimedes screw driven by a 24 V DC geared motor, while a 5 V air pump is controlled by a relay to reduce the pressure inside the rice container after each dispensing cycle. The software is implemented as a finite state machine, including user input, container taring, dispensing, motor stopping, vacuum recovery, and timeout protection. Verification results show that the 5 V control board operates reliably, with a measured supply voltage of 4.98 V, and that the sensors, display, relay-controlled pump, and MOSFET-controlled motor can function together as intended. The final prototype demonstrates the feasibility of integrating electromechanical dispensing, embedded control, and vacuum pressure maintenance into a compact rice storage system.

Contents

1 Introduction

1.1 Problem Statement and Motivation

Milled rice undergoes continuous lipid oxidation and degradation of starch structures during prolonged storage, which adversely affects its physicochemical properties and sensory profiles [1]. Furthermore, environments with fluctuating ambient humidity accelerate mold proliferation and insect infestation, a process primarily driven by mass moisture transfer between the grain and the surrounding air [2]. Traditional rice dispensers available in the consumer market merely provide physical containment and manual dispensing functionality, completely lacking the capability to actively regulate internal atmospheric conditions.

Therefore, there is an explicit engineering need for a smart vacuumed rice dispenser. By extracting air to establish a low-pressure, low-oxygen environment, the rates of aerobic respiration and lipid oxidation in stored grains can be significantly suppressed. Additionally, integrating modern microcontrollers and Internet of Things (IoT) architecture [3] allows for automated electromechanical dispensing, precise portion control, and real-time remote pressure monitoring. This systematic approach enhances both storage efficacy and user convenience, overcoming the fundamental limitations of existing passive storage solutions.

1.2 High-Level Requirements and Block Diagram

The final prototype is an automatic rice dispensing and vacuum storage system. The user sets the target rice amount from **50 g to 1000 g** using the rotary encoder and starts the dispensing process by pressing the button. During dispensing, the Arduino Nano controls the motor to rotate the Archimedes screw, while the load cell measures the dispensed rice mass in real time. When the measured weight reaches the selected target, the motor stops automatically.

After dispensing is complete, the air pump is activated to adjust the pressure inside the rice container. The pressure sensor monitors the internal pressure, and the pump stops when the preset pressure threshold of **900 hPa** is reached. The OLED display shows the target weight, current weight, pressure reading, and system status.

The high-level requirements are:

- The system shall allow the user to set a target rice amount between **50 g and 1000 g**.
- The system shall automatically dispense rice using the Archimedes screw mechanism.
- The motor shall stop when the load cell detects that the measured rice weight reaches the selected target.
- The air pump shall operate after dispensing and stop when the pressure sensor reaches the preset threshold of **900 hPa**.

- The OLED display shall show the target weight, current weight, pressure reading, and current system status.
- The mechanical structure shall prevent direct contact with moving parts and keep the rice path, air path, and electrical components separated.

Figure 1 shows the high-level block diagram of the final system.

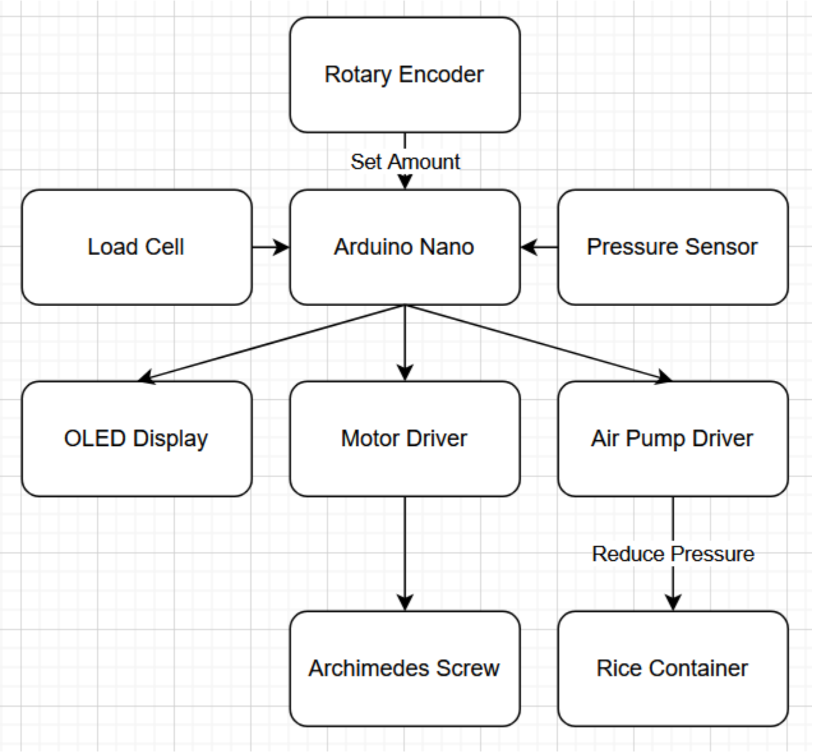


Figure 1: High-level block diagram

2 Design

2.1 Structural Design

2.1.1 Electromechanical Dispensing and Containment Subsystem

The electromechanical dispensing and containment subsystem acts as the fundamental physical infrastructure of the smart dispenser. As illustrated in the CAD cross-sectional view (Figure 2), the primary storage vessel is structurally partitioned into an upper and a lower housing. To ensure robust hermeticity under internal negative pressure, the two halves are mechanically coupled via a central flange fastened by twelve equidistant M8 structural bolts. A custom-molded rectangular silicone elastomer gasket is compressed at the interface. According to standard pressure vessel design principles [4], the total preload clamping force F_c generated by the twelve bolts must strictly exceed the opposing atmospheric force acting on the chamber walls to prevent gasket separation.

At the superior aspect of the upper housing, a threaded lid is employed for user access and grain refilling. This lid utilizes a concentric silicone O-ring to form a radial seal. Furthermore, a dedicated pneumatic through-hole is integrated into the lid, interfacing with flexible elastomeric tubing to connect the containment chamber to the vacuum evacuation subsystem.

For precise material extraction, the system incorporates an internal Archimedean screw pump installed at a fixed inclination angle. The assembly comprises a rotating helical auger, a cylindrical stator casing, a rigid base mount, and a DC motor secured by a custom stabilization bracket. Granular flow mechanics [5] dictates that the theoretical volumetric dispensing rate Q of the screw conveyor can be mathematically modeled as:

$$Q = \eta \cdot \frac{\pi}{4}(D^2 - d^2) \cdot p \cdot N$$

where D and d are the outer and inner diameters of the screw flight respectively, p represents the pitch length, N is the angular velocity in revolutions per minute (RPM), and η denotes the volumetric filling efficiency coefficient (typically empirically derived based on the internal friction angle of the rice grains).

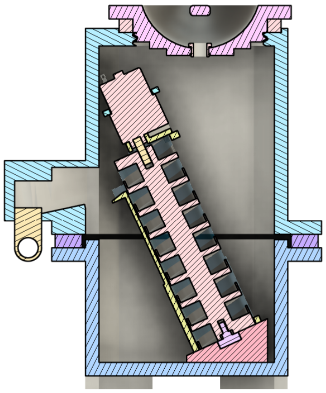
Finally, the distal discharge port is normally isolated utilizing a custom-fitted silicone stopper. This component operates as a passive hermetic seal during the vacuum storage phase and is only momentarily disengaged during the active motor-driven dispensing cycle.

(a) Home view

(b) Front view

(c) Right view

(d) Top view



(e) Cross-sectional view

Figure 2: Different views of the bucket structure

2.1.2 Vacuum Sealing and Pressure Maintenance Subsystem

The vacuum sealing and pressure maintenance subsystem is designed to create a low-pressure storage environment inside the rice container, thereby improving the preservation performance of the dispenser. By reducing the amount of air inside the chamber, the system can help slow down moisture absorption and quality degradation during storage. This subsystem mainly consists of the sealed bucket structure, silicone sealing elements, pneumatic tubing, and the air-pump interface.

The sealing strategy is based on both structural compression and material-level sealing. At the main flange interface between the upper and lower housings, a rectangular silicone gasket is compressed by twelve M8 bolts to prevent leakage along the joint surface. At the top lid, a circular silicone O-ring is used to provide radial sealing during repeated opening and closing. In addition, auxiliary openings are sealed by silicone plugs or through-hole sealing components. These measures are intended to reduce leakage paths and maintain a stable internal pressure after vacuuming.

However, because the rice container is assembled from multiple 3D-printed parts, air-tightness remains one of the main structural challenges. Although the interfaces are sealed by rubber components, the printed parts themselves may still contain microscopic pores or interlayer defects caused by the additive manufacturing process. Under vacuum-pumping conditions, these defects may allow slow air leakage through the container wall, which would reduce the pressure-holding capability of the system and affect long-term storage performance.

To improve the sealing performance of the prototype, the current solution is to apply a waterproof sealant coating to the surface of the 3D-printed parts. This treatment is expected to fill small pores on the printed surface and reduce air permeation through the material. Combined with the gasket compression at the flange and the O-ring sealing at the lid, this method provides a practical sealing solution for the current prototype stage.

From a mechanical point of view, the sealing reliability depends on whether the bolt preload is sufficient to keep the gasket compressed during vacuum operation. If the clamping force is too small, local separation may occur at the interface, leading to leakage. Therefore, the flange connection, gasket design, and surface treatment must work together to ensure that the chamber can maintain the required negative pressure during storage.

Future validation of this subsystem will include leakage testing and vacuum retention experiments. The main evaluation indices will include the achievable vacuum level, the pressure recovery rate over time, and the sealing stability after repeated dispensing and vacuuming cycles.

2.2 Electrical Design

2.2.1 PCB and Power Supply

The electrical subsystem is built around an Arduino Nano, which serves as the main controller for user input, sensor reading, display output, motor control, and pump control. The custom PCB is designed as a **5 V control board**. It provides organized connections for the Arduino Nano, rotary encoder, OLED display, HX711 load cell amplifier, BMP280 pressure sensor, relay control signal, and MOSFET motor control signal.

The final electrical design is different from the original design document, where the system included a temperature and humidity sensor, Peltier cooling device, and fan for climate control. In the final prototype, these components were removed, and the storage function is implemented through pressure monitoring and air-pump control instead. The original document also used a stepper-motor-based dispensing concept, but the final system uses a **24 V DC geared motor** to drive the Archimedes screw.

The PCB is powered by a regulated **5 V input**, which supplies the Arduino Nano and low-power modules. The OLED display, HX711 module, rotary encoder, and BMP280 pressure sensor are connected to the Arduino through the PCB. The 24 V DC geared motor is powered by a separate 24 V supply because its voltage and current requirements are higher than what the PCB and Arduino can provide. The motor is switched by a MOSFET circuit, where the Arduino only provides a logic-level control signal. The 5 V air pump is controlled by a relay module, allowing the Arduino to turn the pump on and off without directly driving the pump current.

To ensure correct signal reference, the PCB ground and the external actuator power ground are connected together. This common-ground design allows the Arduino control signals to drive the MOSFET and relay reliably while keeping the higher-current actuator paths separated from the low-power logic circuit.

Figure 3 (shows the PCB layout of the final electrical control board. The Arduino Nano is placed near the center of the board, while the connectors for other parts are arranged around it. This layout reduces loose wiring and provides a more stable connection between the control board and the mechanical prototype.

A full-size version of the PCB layout is provided in Figure 5

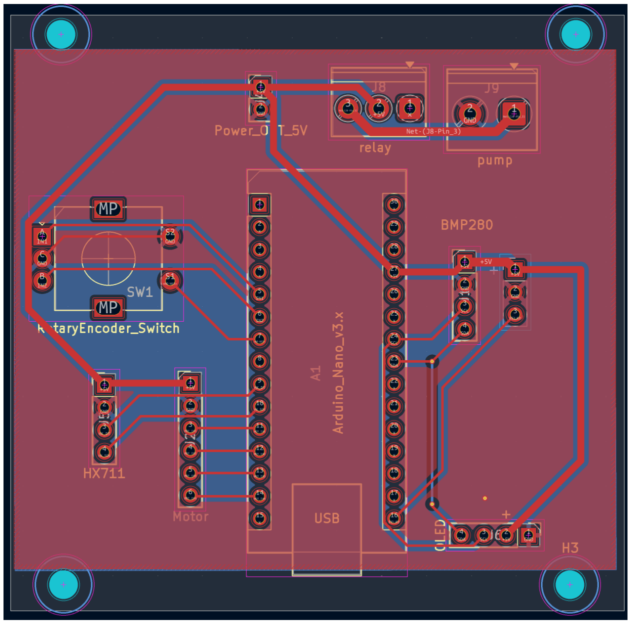


Figure 3: PCB layout of the final 5 V control board

It should be noted that the 01x06 Motor connector was originally designed for a stepper motor. After replacing the motor with a DC geared one, we only use two of these pins.

2.2.2 Sensors and Actuators

The electrical system includes user input devices, feedback sensors, display output, and two actuator-control circuits. The user interface consists of a rotary encoder with a push button and an OLED display. The rotary encoder allows the user to adjust the target rice amount, while the button starts the dispensing process. The OLED display shows the target weight, current measured weight, pressure reading, and system status.

The weight feedback is provided by a load cell connected to an HX711 amplifier module. The HX711 converts the small signal from the load cell into digital data that can be read by the Arduino Nano. During dispensing, the Arduino continuously reads the weight value and stops the motor when the measured rice mass reaches the selected target within the designed range of **50 g to 1000 g**.

The pressure feedback is provided by a BMP280 pressure sensor. After dispensing is complete, the Arduino activates the 5 V air pump through the relay. The pump continues operating until the measured pressure reaches the preset threshold of **900 hPa**. This pressure-based process replaces the original temperature and humidity control function.

The actuator side includes the 24 V DC geared motor and the 5 V air pump. The motor drives the Archimedes screw mechanism to dispense rice. Since the motor requires 24 V, it is powered separately and controlled through a MOSFET switching circuit. The air pump is controlled through a relay because it is only required to switch on and off after the dispensing process. Table 1 summarizes the main electrical components used in the final design.

Table 1: Main electrical components used in the final design

Component	Function	Interface
Arduino Nano	Serves as the main control unit for user input, sensor reading, display output, motor control, and pump control.	Digital/analog I/O
Rotary encoder	Allows the user to adjust the target rice weight.	Digital input
Encoder button	Starts the dispensing process after the target weight is selected.	Digital input
OLED display	Displays the target weight, current measured weight, pressure reading, and system status.	I2C
Load cell and HX711	Measures the dispensed rice weight and converts the load cell signal into digital data for the Arduino Nano.	Digital interface
BMP280 pressure sensor	Measures the internal pressure of the rice container for pump-control logic.	I2C
MOSFET circuit	Switches the 24 V DC geared motor used to drive the Archimedes screw.	Digital output
Relay module	Switches the 5 V air pump on and off after dispensing is complete.	Digital output

2.3 Software Design

The software of the smart rice dispenser was implemented on an Arduino-based controller. The main responsibility of the software is to coordinate the user input, weight sensing, pressure sensing, rice dispensing motor, vacuum pump, and OLED display. The program was designed as a state-based control system so that each operation, such as waiting for user input, dispensing rice, stopping the motor, and vacuuming the container, can be handled in a clear and reliable sequence.

2.3.1 User Interface

The user interface consists of a rotary encoder with a push button and a 128×64 OLED display. The rotary encoder allows the user to select the desired amount of rice before dispensing. In the current implementation, the target amount changes in 10 g increments and is limited between 10 g and 1000 g to prevent invalid input values.

The OLED display provides real-time feedback to the user. When the system is idle, the screen displays the selected target weight, the current pressure reading, and the system status. During dispensing, the display shows the measured rice weight and the target weight. During vacuuming, the display shows the current pressure inside the container. This allows the user to understand the current operating stage without using an external computer.

2.3.2 Control Logic

The control logic was implemented as a finite state machine. The main states are `IDLE`, `TARE_CONTAINER`, `DISPENSING`, `STOP_MOTOR`, `RE_VACUUM`, and `ERROR_TIMEOUT`. This structure makes the software easier to debug and prevents multiple actuators from being controlled incorrectly at the same time.

In the `IDLE` state, the motor is off and the system waits for the user to set the target weight and press the encoder button. After the button is pressed, the system enters the `TARE_CONTAINER` state. This state is important because the user places an empty container on the load cell before dispensing rice. The software performs a tare operation before starting the motor, which sets the current container weight as zero. As a result, the measured weight during dispensing only represents the weight of the rice added to the container.

After taring, the system enters the `DISPENSING` state. The DC gear motor is turned on and rice begins to flow into the container. The Arduino continuously reads the HX711 load cell module and compares the measured weight with the target weight. When the measured weight reaches or exceeds the selected target amount, the system enters the `STOP_MOTOR` state and turns off the motor.

A timeout condition was also included to improve reliability. If the system dispenses for more than 15 seconds without reaching the target weight, it enters the `ERROR_TIMEOUT` state. This prevents the motor from running continuously if the rice outlet is blocked or if the storage container is empty.

After the motor stops, the system enters the `RE_VACUUM` state. In this state, the vacuum pump is activated to reduce the pressure inside the rice storage container. The pressure is measured using the BMP280 sensor. When the measured pressure reaches the target vacuum threshold, the pump is turned off and the system returns to the `IDLE` state. This vacuum process helps improve rice storage by reducing air exposure after dispensing.

Figure 4 shows the finite state machine used in the Arduino software.

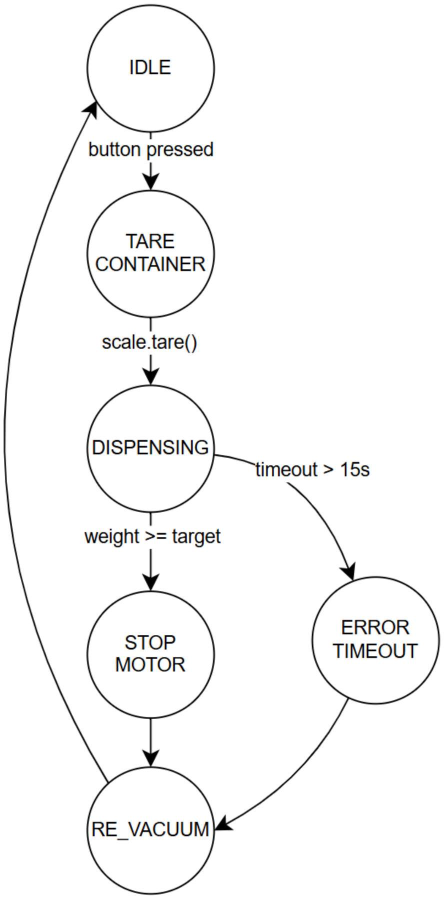


Figure 4: Finite state machine of the Arduino control logic.

3 Design Verification

3.1 Structure Verification

3.1.1 Electromechanical Dispensing Subsystem Verification

The verification of the electromechanical dispensing and containment subsystem is conducted to validate its two primary functional requirements: the mechanical hermeticity for vacuum support and the dispensing accuracy of the physical conveying mechanism.

The structural hermeticity of the assembled containment vessel is verified through an active evacuation and sealing test. This procedure explicitly evaluates the effectiveness of the physical interfaces, primarily the compression seal maintained by the twelve M8 structural bolts against the central silicone gasket, as well as the radial O-ring on the top lid. The vacuum pump is coupled to the designated pneumatic port on the lid. Upon system activation, the evacuation process is monitored to confirm that the internal air volume is continuously extracted. The mechanical assembly successfully passes this verification if the system visibly establishes and sustains a negative internal pressure state, demonstrating that there is no macroscopic atmospheric leakage across the fastened flange. This functional test confirms that the mechanical enclosure strictly isolates the internal volume from ambient conditions.
The performance and accuracy of the Archimedean screw dispensing mechanism are experimentally characterized. Because granular materials like milled rice exhibit complex internal friction and non-Newtonian flow behaviors, empirical measurement is mandatory. A standard target output mass of $M_{\text{target}} = 150$ g is defined for a single dispensing cycle. The integrated DC motor is actuated to drive the helical auger, and the actual dispensed mass M_{actual} is captured using a high-precision digital scale. Across multiple consecutive iterations, the relative mass error E is evaluated as:

$$E = \frac{|M_{\text{actual}} - M_{\text{target}}|}{M_{\text{target}}} \times 100\%$$

The physical dispensing subsystem meets its core engineering objective if the maximum recorded error margin strictly satisfies $E \leq 5\%$. This confirms that the internal geometric clearances of the screw pump and the motor operational timing are suitably optimized to overcome granular jamming and deliver reliable portion control.

3.1.2 Vacuum Sealing and Pressure Maintenance Subsystem Verification

The verification of the vacuum sealing and pressure maintenance subsystem is conducted to validate whether the storage chamber can establish and maintain a sufficiently stable negative-pressure environment during practical operation. This verification directly addresses the central engineering concern of the present prototype, namely, whether the combined sealing strategy based on flange compression, elastomeric sealing components, and surface treatment of the 3D-printed parts is adequate to suppress air leakage.

The verification procedure is performed through a vacuum retention experiment. First, the rice container is fully assembled with the rectangular silicone gasket positioned at the flange interface, the circular O-ring installed at the top lid, and all auxiliary openings sealed by the designated silicone plugs. The vacuum pump is then connected to the pneumatic port on the lid and activated until the internal chamber pressure reaches the target value measured by the pressure sensor. After this evacuation stage, the pump is turned off, and the internal pressure evolution is continuously monitored over a prescribed holding period.

The subsystem is considered to satisfy its sealing requirement if the pressure increase during the holding stage remains within an acceptable limit, indicating that no significant leakage path exists through the flange interface, lid connection, auxiliary sealing points, or 3D-printed chamber walls. In contrast, a rapid recovery of internal pressure toward atmospheric conditions would imply insufficient hermeticity and would indicate that the present sealing configuration requires further improvement.

To quantitatively characterize the pressure-holding performance, the relative pressure recovery ratio R_p is defined as:

$$R_p = \frac{P_t - P_{\min}}{P_{\text{atm}} - P_{\min}} \times 100\%$$

where $P_{\min}$ is the minimum pressure reached immediately after vacuum pumping, P_t is the chamber pressure measured after a specified holding time t, and P_{atm} is the ambient atmospheric pressure. A smaller value of R_p indicates better pressure retention capability and therefore better sealing performance.

In addition, comparative testing can be carried out before and after applying waterproof sealant to the surface of the 3D-printed parts. If the coated structure shows a significantly lower pressure recovery rate than the uncoated structure, the experiment will confirm that microscopic porosity in the printed material is one of the dominant leakage sources. In this way, the subsystem verification not only evaluates whether the current prototype meets its functional requirement, but also provides direct experimental evidence for the effectiveness of the proposed sealing improvement strategy.

3.2 Electrical Verification

The electrical verification was conducted to confirm that the PCB and connected electrical modules could support the final dispensing and pressure-control functions. The PCB was designed as a 5 V control board, so the first test was to verify that the Arduino Nano and low-voltage modules could operate reliably from the 5 V rail. With the Arduino Nano, OLED display, HX711 module, rotary encoder, and BMP280 pressure sensor connected, the measured supply voltage was **4.98 V**, which was within the acceptable range for the control circuit.

The weight-sensing circuit was tested using the load cell and HX711 amplifier. A **5 g calibration weight** was used to calibrate the weight reading, and the measured value changed consistently when additional weight was applied to the weighing platform. The BMP280 pressure sensor was also tested through the Arduino. Under atmospheric conditions, the displayed pressure value was slightly above 1000, which corresponds to a hectopascal-level reading. Therefore, the pump control threshold was treated as approximately **900 hPa**, in the final pressure-control logic.

The actuator control circuits were tested separately. The 5 V air pump was controlled through a relay module, and the relay reliably switched the pump on and off during repeated tests. The dispensing motor required more modification. The original design used a stepper motor, but during testing the stepper motor did not provide enough torque to drive the Archimedes screw under load. To solve this issue, the motor was replaced by a **24 V DC geared motor** controlled through a MOSFET switching circuit. After this change, the motor was able to rotate the screw and support the rice dispensing process.

Overall, the electrical verification showed that the 5 V control board, sensor modules, display, relay-controlled pump, and MOSFET-controlled motor could operate together. The main electrical design change during verification was the replacement of the original stepper motor with a higher-torque 24 V DC geared motor.

3.3 Software Verification

The software verification focused on confirming that the Arduino program correctly responded to user input, updated the OLED display, controlled the dispensing motor, read the weight sensor, and activated the vacuum pump after dispensing. The verification was divided into user interface tests and actuator control tests.

3.3.1 OLED and Input Test

The purpose of this test was to verify that the rotary encoder and OLED display worked correctly as the main user interface. During the test, the rotary encoder was rotated clockwise and counterclockwise while observing the value shown on the OLED display. The expected behavior was that the target rice amount would increase or decrease in 10 g increments. The lower and upper software limits were also tested to ensure that the value stayed within the allowed range.

The push button on the rotary encoder was then tested. When the system was in the `IDLE` state, pressing the button was expected to start the dispensing sequence. The OLED display was expected to change from the ready screen to the taring and dispensing screens.

Table 2: Electrical verification results

Test Item	Verification Method	Expected Result	Result
5 V control board	Measured the PCB 5 V rail with the Arduino Nano, OLED display, HX711 module, rotary encoder, and BMP280 sensor connected.	The measured voltage should remain close to 5 V and support all low-voltage control modules.	Pass, 4.98 V
Load cell and HX711	Calibrated the load cell using a 5 g reference weight and observed the measured value when weight was added.	The weight reading should respond consistently to applied weight after calibration.	Pass
BMP280 pressure sensor	Read the pressure value through the Arduino and observed the atmospheric pressure reading.	The sensor should provide a stable pressure reading for pump-control logic.	Pass
Relay pump control	Sent digital HIGH and LOW signals from the Arduino to the relay module.	The 5 V air pump should switch on and off correctly.	Pass
MOSFET motor control	Sent digital control signals from the Arduino to the MOSFET circuit after replacing the original stepper motor with a 24 V DC geared motor.	The motor should rotate the Archimedes screw and stop when the control signal is disabled.	Pass
Motor torque test	Tested the original stepper motor and then replaced it with a 24 V DC geared motor.	The motor should provide enough torque to drive the Archimedes screw under load.	Pass after revision

Continued on next page

Table 3 – continued from previous page

Test Item	Verification Method	Expected Result	Result

Table 3: OLED and input verification results

Test Item	Verification Method	Expected Result	Result
OLED idle display	Observed the OLED screen when the system was in the `IDLE` state.	The display should show the selected target weight, pressure reading, and ready status.	Pass
Rotary encoder clockwise input	Rotated the encoder clockwise while observing the target weight on the OLED display.	The target weight should increase in 10 g increments.	Pass
Rotary encoder counterclockwise input	Rotated the encoder counterclockwise while observing the target weight on the OLED display.	The target weight should decrease in 10 g increments.	Pass
Weight setting limits	Continued rotating the encoder beyond the minimum and maximum allowed values.	The target weight should remain between 10 g and 1000 g.	Pass
Push-button start input	Pressed the encoder button while the system was in the `IDLE` state.	The system should leave the idle screen and enter the taring and dispensing sequence.	Pass
OLED dispensing display	Observed the OLED screen during rice dispensing.	The display should show the current measured rice weight and the selected target weight.	Pass

3.3.2 Motor and Pump Control Test

The motor and pump control test verified that the Arduino could correctly control the dispensing motor and vacuum pump based on sensor readings. First, an empty container was placed on the load cell. After the button was pressed, the system performed the tare operation before starting the motor. This confirmed that the container weight was ignored and that the displayed weight started from approximately 0 g.

Next, several dispensing tests were performed using different target weights. For each test, the DC gear motor was expected to start during the `DISPENSING` state and stop when the measured rice weight reached the target value. The measured final weight was compared with the selected target weight to evaluate the accuracy of the software control.

The timeout protection was also tested by running the system under a condition where rice could not be dispensed normally. If the target weight was not reached within the timeout period, the system was expected to stop the motor and display an error message. This test verified that the software could prevent continuous motor operation during abnormal conditions.

Finally, the vacuum pump control was tested after the dispensing process. After the motor stopped, the pump was expected to turn on and reduce the pressure inside the storage container. The system continuously monitored the pressure sensor reading and turned off the pump once the target pressure threshold was reached. This verified that the software could automatically restore the vacuum condition after each dispensing cycle.

Table 4: Motor and pump control verification results

Test Item	Verification Method	Expected Result	Result
Container tare operation	Placed an empty container on the load cell and started the dispensing sequence.	The software should perform a tare operation before dispensing so that the container weight is ignored.	Pass
DC motor dispensing control	Started a dispensing cycle and observed the motor control signal and rice outlet.	The DC gear motor should turn on during the `DISPENSING` state and drive the dispensing mechanism.	Pass
Weight-based motor stop	Set a target weight and observed the motor behavior when the measured rice weight reached the target value.	The motor should stop when the measured weight is greater than or equal to the selected target weight.	Pass
Dispensing timeout protection	Tested the system under a condition where rice could not reach the target weight within the timeout period.	The system should stop normal dispensing and enter the `ERROR_TIMEOUT` state after 15 seconds.	Pass
Vacuum pump activation	Observed the pump after the dispensing motor stopped.	The vacuum pump should turn on automatically after dispensing to reduce the pressure inside the rice storage container.	Pass
Pressure-based pump stop	Monitored the BMP280 pressure reading while the pump was running.	The pump should turn off when the measured pressure reaches the target vacuum threshold.	Pass

4 Cost

The total project cost includes both labor cost and material cost. The labor cost was estimated using the formula recommended in the ECE 445 final report guideline:

$$\text{Labor Cost} = \text{Ideal Salary} \times \text{Actual Hours Spent} \times 2.5$$

For this project, the ideal engineering salary was assumed to be RMB 300 per hour. Each team member spent approximately 100 hours on the project, including mechanical design, electrical design, PCB design, software development, assembly, testing, debugging, and report preparation. Therefore, the estimated labor cost for each member is:

$$300\ \text{RMB/h} \times 100\ \text{h} \times 2.5 = 75{,}000\ \text{RMB}$$

Table 5 summarizes the estimated labor cost of the team.

Table 5: Estimated labor cost

Team Member	Hourly Rate	Hours Spent	Estimated Labor Cost
Wang Shining	RMB 300/h	100 h	RMB 75,000
Liu Yuyang	RMB 300/h	100 h	RMB 75,000
Zhao Gaoning	RMB 300/h	100 h	RMB 75,000
Yang Zixin	RMB 300/h	100 h	RMB 75,000
Total	–	400 h	RMB 300,000

The estimated labor cost of the project is RMB 300,000. In addition to labor cost, the prototype also required mechanical, electrical, and control components. The estimated material cost of the current prototype is summarized in Table 6. Based on the current bill of materials, the total known cost is **RMB 482.72**, excluding the cost of one motor whose price has not yet been finalized. Among all components, the PCB fabrication is the most expensive single item, with a cost of **RMB 200**. The 3D-printed structural parts cost **RMB 51**, based on a total printed mass of approximately **1546 g**. Overall, the electrical and control components account for a large proportion of the total cost, while the mechanical sealing and fastening parts remain relatively inexpensive.

5 Conclusion

This project designed and implemented a Smart Vacuumed Rice Dispenser that integrates automatic rice dispensing, real-time weight measurement, pressure monitoring, and vacuum recovery control. The final prototype uses an Arduino Nano as the central controller, together with a rotary encoder, OLED display, HX711 load-cell module, BMP280 pressure sensor, relay-controlled air pump, and MOSFET-controlled DC geared motor. Through this design, the system allows the user to select a target rice amount, automatically dispense rice using an Archimedes screw mechanism, stop the motor based on measured weight feedback, and restore a low-pressure storage condition after dispensing.

The final design also reflects several important engineering revisions made during the development process. The original stepper motor was replaced with a 24 V DC geared motor because the stepper motor did not provide enough torque to drive the screw under load. The original temperature and humidity control concept was also simplified into a pressure-based vacuum storage function, which better matched the final prototype scope. In addition, the electrical system was reorganized around a 5 V control board while using separate actuator power for the motor and pump.

Verification results show that the main subsystems functioned as intended. The 5 V control board supplied the Arduino Nano and low-voltage modules reliably, with a measured voltage of 4.98 V. The load cell and HX711 module responded correctly after calibration, the BMP280 sensor provided pressure readings for pump control, the relay module successfully switched the air pump, and the MOSFET circuit controlled the DC geared motor. The software finite state machine also successfully handled user input, container taring, dispensing, motor stopping, timeout protection, and vacuum recovery.

Because this project involves food storage and electromechanical actuation, safety and reliability are important ethical considerations. The dispenser should prevent direct user contact with moving parts, avoid unsafe wiring exposure, and use materials that are appropriate for contact with stored rice. In addition, the system should not mislead users about long-term food preservation performance unless vacuum retention and food-safety tests have been fully validated. These considerations are consistent with the engineering responsibility to prioritize public safety, health, and welfare.

From a broader perspective, the Smart Vacuumed Rice Dispenser may improve household convenience and reduce food waste by helping users store and dispense rice more consistently. Economically, the prototype remains relatively low-cost, although future versions should reduce PCB and assembly costs for practical production. Environmentally, better rice preservation may reduce unnecessary grain disposal, but the use of plastic printed parts and electronic components should also be considered in future design improvements.

Overall, the prototype demonstrates the feasibility of combining electromechanical dispensing and pressure-based storage control in a compact rice dispenser. However, several aspects can still be improved in future work. The airtightness of the 3D-printed container should be further tested and improved, especially because microscopic pores and assembly gaps may affect long-term vacuum retention. The dispensing accuracy can also be improved through repeated calibration, better motor speed control, and more detailed testing under different rice amounts. Future versions may further optimize the mechanical structure, reduce wiring complexity, improve food safety and durability, and add remote monitoring functions. Despite these limitations, the final system successfully achieves the major goals of automatic dispensing, sensor-based feedback control, and post-dispensing vacuum recovery.

References

[1] Z. Zhou, K. Robards, S. Helliwell, and C. Blanchard, "Ageing of stored rice: Changes in chemical and physical attributes," *Journal of Cereal Science*, vol. 35, no. 1, pp. 65–78, 2002.

[2] K. R. Bhattacharya, *Rice Quality: A Guide to Rice Properties and Analysis*. Woodhead Publishing, 2011.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] R. G. Budynas and J. K. Nisbett, *Shigley's Mechanical Engineering Design*, 11th. McGraw-Hill Education, 2020.

[5] A. W. Roberts, "The mechanics of powder handling and processing," *Advanced Powder Technology*, vol. 10, no. 1, pp. 63–83, 1999.

Appendix A

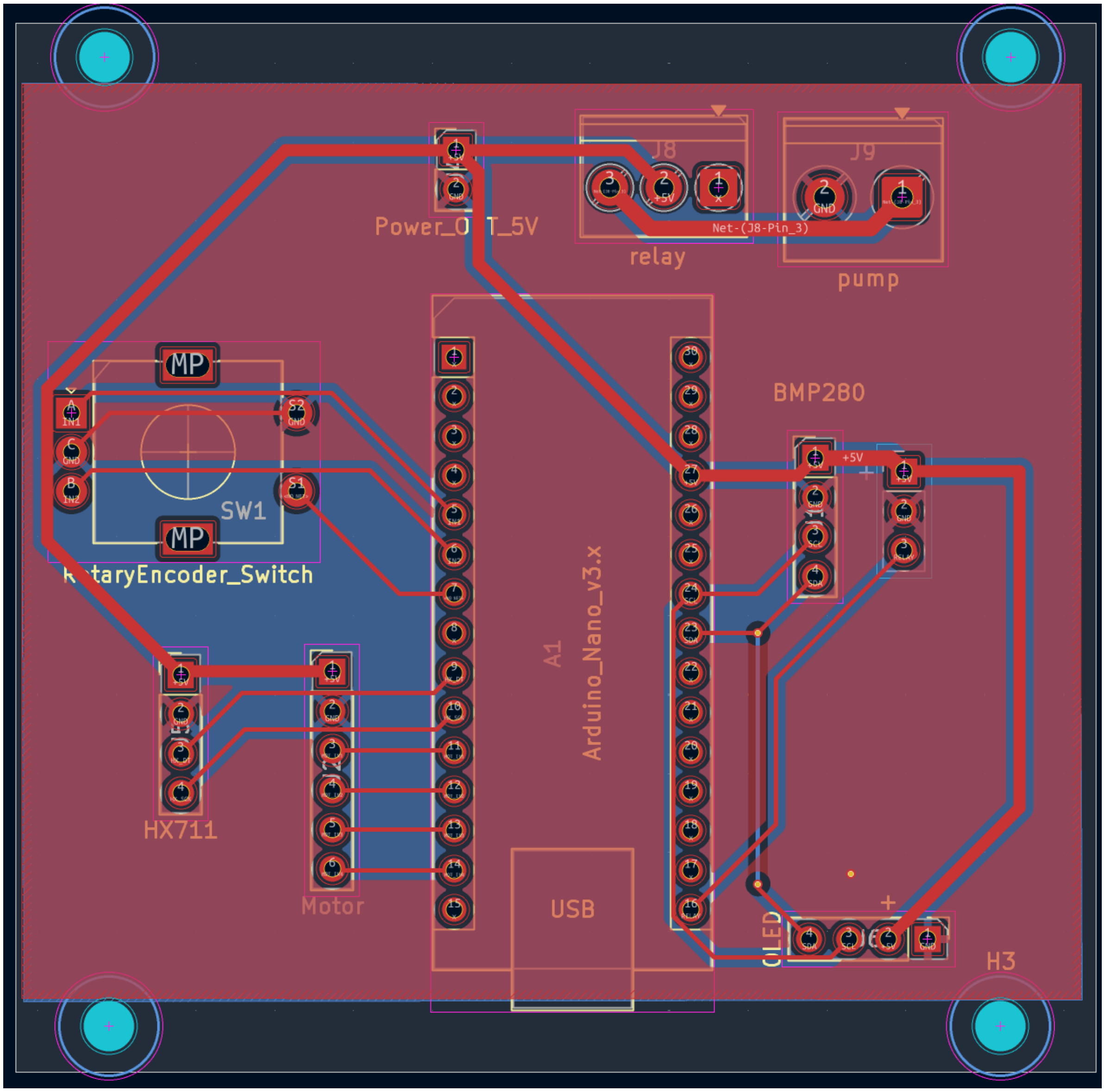


Figure 5: Full-size PCB layout of the final 5 V control board

Table 6: Bill of materials and estimated cost of the prototype

Item	Unit Price (RMB)	Quantity	Cost (RMB)
3D printed parts	0.033/g	1546 g	51.00
Square sealing ring	15.00	1	15.00
Circular sealing ring	15.00	1	15.00
Silicone gasket	0.11	4	0.44
Silicone through-hole plug	0.53	1	0.53
Silicone plug	4.87	1	4.87
Linear bearing	5.07	1	5.07
Air tube	5.50	1	5.50
M8 screws	1.16	12	11.60
M8 nuts	0.89	12	8.90
Load cell sensor	22.50	1	22.50
PCB	–	5	200.00
Transformer	–	1	25.74
Electronic switch control board	–	1	6.10
DC gear motor	–	1	20.76
5V 2A power adapter	–	1	10.69
Stepper motor	–	1	3.90
Rotary encoder	–	1	3.22
HX711 module	–	1	3.38
Pin headers	–	1	4.67
DC connector splitter	–	1	4.02
Power cable	–	2	11.44
Dupont wire splitter	–	2	5.50
Arduino Nano board	–	1	17.00
Transformer (second item)	–	1	18.81
Dupont wires 40P	–	1	7.08
Total known cost			482.72