

Design and Implementation of a Three-Bottle Automatic Water Conveyor System Using Arduino and LabVIEW

Atmaja Sonawane, Riya Sahay, Dr. Saravanakumar R

School of Electrical Engineering Vellore Institute of Technology
Vellore, India

Abstract- Liquid filling processes require a lot of manual attention and automating these processes can improve the operational consistency, reduce the manual intervention and simplify the filling tasks. This paper presents a simple and low-cost design and implementation of a three-bottle filling automatic conveyor which uses main hardware component as Arduino Uno and integrates a software component of LabVIEW for monitoring and alerts about the filling process. The proposed design consists of a stepper motor which drives a rotating platform on which the bottles are placed, a homing switch, an infrared sensor (IR sensor) which is used to detect the bottles and a DC water pump used in the filling of the bottles. The Arduino acts as the centre of signals and controls the process by sending required signals to the components, it establishes the initial platform position, detecting bottle presence, operating the pump for a predefined duration, and rotating the platform to position the next bottle. To send alerts to the user, a buzzer is integrated which beeps after each bottle is filled, while LabVIEW receives the status message through the Arduino through serial communication and displays it on the monitoring panel. The prototype is tested using three bottles where each filling process took a five second time period and a delay of 100 ms between each filling. All the three bottles were observed to be filled successfully and the system results were verified successfully.

Keywords— Automatic filling system, Arduino UNO, embedded control, LabVIEW, bottle positioning, virtual instrumentation.

I. INTRODUCTION

Automated liquid handling is an important part of packaging and processing operations, where the containers have to be positioned one after another, the dispensing has to be controlled and the actuators have to work in coordination. In manual filling, an operator has to repeatedly place the containers in position and also control the dispensing process by hand. By automating these steps, the dependence on manual handling can be reduced and the actions can be carried out in a fixed and repeatable sequence. A micro-controller-based system is a practical platform which can be used to connect the sensors, the actuators and the programmed control logic together. When the sensor inputs are coordinated with a predefined control sequence, such a system can detect the container, dispense the liquid and position the mechanism with very little help from an operator.

Along with this, the use of virtual instrumentation makes it possible to collect and display the status information of the system, which makes it easier to observe the process while it is running.

This paper presents the design and implementation of a three-bottle automatic water conveyor system which is built using an Arduino UNO and LabVIEW. In the physical arrangement, a rotating platform driven by a stepper motor is used to bring the three bottles one by one beneath the water outlet. A limit switch is used to set the reference position of the platform, and an infrared sensor (IR sensor) detects whether a bottle is present at the filling station. The water is dispensed by a DC water pump, which is controlled through a MOSFET-based driver circuit, and it runs for a predefined time of 5 s for each bottle. A buzzer is also included in the controller to give sound feedback, and interruption-handling routines are used to indicate some of the operating events and abnormal conditions.

A LabVIEW virtual instrument (VI) is connected to the embedded controller through USB serial communication at 9600 baud. The status messages generated by the Arduino are received and displayed by this interface, while the physical filling sequence is still controlled by the microcontroller on its own. Since the process execution and the monitoring are kept separate in this way, the system can operate without depending on the computer interface all the time.

The main contribution of this work is combining bottle detection, homing to the reference position, timed dispensing, moving the platform in sequence and monitoring through virtual instrumentation in a single automated system. The prototype was evaluated by functional testing of the three-bottle filling sequence. This implementation shows that sensing, actuation and status monitoring can be coordinated using a compact embedded architecture.

II. SYSTEM ARCHITECTURE AND DESIGN

The proposed system is made up of three main subsystems, which are the sensing subsystem, the embedded control and actuation subsystem, and the supervisory monitoring sub-system. The information about the presence of the bottle and the reference position of the rotating platform is given by the sensing subsystem. These inputs are processed by the embedded controller, which then coordinates the stepper motor, the water pump and the audible indicators. The status messages generated by the controller are displayed by the monitoring subsystem through a LabVIEW interface.

An Arduino UNO is used as the central processing unit. The infrared sensor (IR sensor) is connected to the controller to detect whether a bottle is present at the filling station, and a limit switch is used to set the reference position of the rotating platform. The stepper motor is driven through a ULN2003 driver module so that the bottles can be positioned one after another. A DC water pump, which is switched through a MOSFET-based driver circuit, supplies the water through a filling tube. An active buzzer is used

to give sound feedback during some of the operating events.

The Arduino is connected to the LabVIEW virtual instrument (VI) through a USB serial connection which is configured at 9600 baud. Through this channel, the status messages are transmitted from the embedded controller to the computer. These messages are displayed by the VI in a log indicator, but no control commands are sent from it to the machine. Because of this, the automatic filling sequence stays under the control of the Arduino and the LabVIEW interface does not have to be running continuously.

The overall architecture keeps the execution of the physical process separate from the supervisory monitoring. With this arrangement, the embedded controller can coordinate the sensing and actuation functions, and the computer gives a view of the operating status of the system.

III. HARDWARE DESIGN AND IMPLEMENTATION

1. Embedded Controller

An Arduino UNO is used to run the control algorithm and to connect with the sensors and the actuators. The AUTO input, the IR sensor and the home limit switch are read by the controller, and it generates the signals which are needed for the stepper motor driver, the pump driver and the buzzer. The firmware is written in Arduino C++ and the CheapStepper library is used for operating the stepper motor.

The controller also sends text status messages through its USB serial interface. These messages show events like the start of homing, the successful filling of a bottle, the indexing of the platform and the fault conditions.

2. Rotating Platform and Stepper-Motor Drive

A rotating platform holds three bottles and brings them one by one beneath the water outlet. The platform is driven by a 28BYJ-48-type stepper motor through a ULN2003 driver module, and this driver gives the switching interface which is needed to energize the windings of the motor.

In the firmware, a nominal value of 4096 half-steps per revolution is used. As the platform has three bottle positions, the nominal movement for each position is calculated as approximately 1365.5. Since the number of steps has to be an integer, the remainder is distributed by the firmware across the successive indexing movements. The sequence which results is 1365, 1365 and 1366 steps, which gives a nominal total of 4096 steps over the three indexing movements.

A micro limit switch is used to set the reference position. In the homing, the platform keeps rotating until the switch is activated. To avoid endless movement when the reference position is not detected, the controller limits the homing search to 5000 steps.

Bottle Detection An IR obstacle sensor is used to detect whether a bottle is present at the filling station. The output of the sensor is connected to the digital input A1 of the Arduino and it is configured as an active-LOW input, so a LOW signal means that a bottle is detected.

To reduce the effect of short fluctuations in the sensor, a stability check of 100 ms is applied by the firmware before the bottle detection is accepted. The filling starts only when the AUTO input is enabled and the condition of bottle presence is satisfied.

The sensor which is implemented gives only the information of bottle presence, and it does not measure the level of the liquid inside the bottle.

3. Water-Dispensing Mechanism

A DC water pump carries the water from the reservoir through a tube which is placed above the bottle-filling station. The pump is switched through a MOSFET-based driver circuit, because it needs more current than what an output of the microcontroller can supply directly.

For each bottle, the pump is turned on for a predefined time of 5 s. When this time ends, the pump is switched off by the controller and a completion message is generated. The actual volume which is dispensed depends on the flow

characteristics of the pump and on the hydraulic arrangement. Since the prototype has no flow sensor or liquid-level sensor, the implementation gives time-based dispensing and not direct volume-controlled filling.

4. Audible Feedback and Operator Input

An AUTO switch or push button which is connected to the Arduino is used to enable or disable the automatic operation. This input is checked by the firmware during the homing, filling and platform-movement routines.

An active buzzer is used to give sound indications for some selected events. Different beep patterns are associated by the firmware with events such as successful homing, completion of a filling operation, removal of the bottle during filling and failure to find the home position.

5. Power and Electrical Interfacing

The control electronics are supplied through a regulated power arrangement which includes a buck converter. A separate external supply is used for the water pump, so that its working current is not drawn directly from an Arduino output. A proper common ground is needed between the control and actuator circuits for the signal reference.

The stepper motor driver and the MOSFET driver act as the electrical interfaces between the low-power signals of the controller and the actuators.

IV. SOFTWARE DESIGN AND OPERATING METHODOLOGY

1. Arduino UNO Firmware

The Arduino firmware is responsible for running the automatic filling sequence, and it does this through a set of separate routines which handle input monitoring, homing, bottle detection, dispensing, indexing, and fault handling. The main parameters used in the program are the nominal steps per revolution, the number of bottle positions, the filling duration, the sensor-stability interval, and the maximum number of homing steps allowed.

When the system is powered on, the firmware first sets up the required input and output pins, starts the serial communication at a baud rate of 9600, and makes sure that the pump stays in the switched-off state. Once the automatic operation is enabled by the user, the controller first moves the platform to establish its reference position, and only after this homing is completed does it begin the filling sequence.

2. Homing Procedure

Homing is the process which establishes a reference position, and all the later movements of the platform are calculated starting from this position. Once the automatic operation is enabled, the controller first checks the state of the limit switch. In case the switch is not already active, the stepper motor is made to move forward in small increments, and during this movement the controller keeps monitoring the state of the switch continuously.

As soon as the switch gets activated, the controller resets the position count to zero and marks the platform as homed, so that the system knows it is at the starting point. If the switch is not detected even after 5000 steps, the controller treats this as a failure and records a fault, keeps the pump in the switched-off state, and also produces an audible fault indication through the buzzer to alert the user.

In this way, the procedure provides a reference point for the programmed indexing sequence that follows. However, the nominal step count by itself cannot be taken as a measurement of the actual angular positioning accuracy of the platform.

3. Bottle Detection and Filling Sequence

Once homing has been completed successfully, the controller waits until a bottle is detected at the filling position. The signal from the sensor has to stay active for the whole programmed stability interval of 100 ms before the bottle is accepted as present, which avoids false triggering.

When a bottle has been detected and the automatic operation is still enabled, the controller switches the pump on for 5 s. While the dispensing is going on, the firmware keeps monitoring both the AUTO input

and the bottle-presence signal. If the automatic operation gets disabled or the bottle is removed in the middle of the process, the pump is switched off immediately and the corresponding status message is generated.

When the programmed filling interval ends normally, the controller switches the pump off, reports that the filling is complete, and activates the buzzer to alert the user. After this, the platform is moved so that the next bottle position comes under the filling point.

4. Sequential Indexing

After a filling operation has been completed successfully, the firmware calculates the number of motor steps needed to move the platform to the next position. Since one revolution cannot be divided equally into three whole-number movements, the indexing routine spreads the leftover steps across the three movements, so that the programmed total adds up to one nominal revolution.

The bottle-position variable is updated once the indexing is done. After the third position is completed, the controller sends the sequence back to the first bottle position, so that the cycle can continue in a loop.

5. LabVIEW-Based Monitoring

A LabVIEW VI is used as a computer-based interface through which the operation of the filling system can be observed. The VI uses NI-VISA to set up the serial communication with the Arduino through the COM port selected by the user, and the serial configuration is kept at 9600 baud so that it matches the baud rate set in the Arduino firmware.

The VI is made up of a serial-port selection control, a stop button, and a string indicator labelled Arduino Log. Once the serial connection is configured, a While Loop repeatedly checks how many bytes are available at the serial port and reads the data that has arrived. The received text is joined to the existing log using string concatenation and a shift register, so that the messages build up one after another in the display. A wait interval of 100 ms is also placed inside the loop.

When the operator stops the VI, the loop terminates and the VISA connection is closed. This action only ends the monitoring loop, and it does not stop the filling process that is being controlled by the Arduino. The physical machine is operated through its hardware AUTO input.

graphicx float 0.48

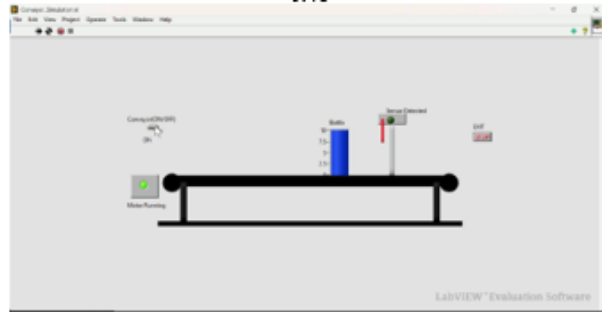


Fig. 1.

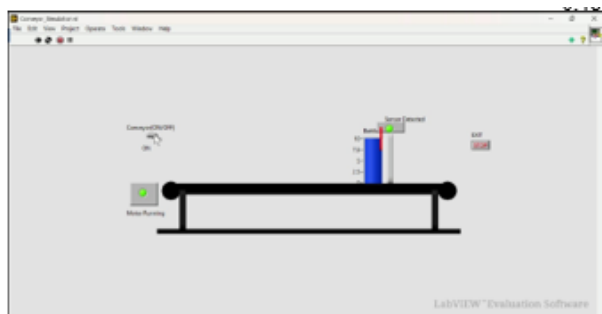


Fig. 2.

V. EXPERIMENTAL EVALUATION AND RESULTS

1. Experimental Setup

The prototype was evaluated by carrying out a functional demonstration of its three-bottle filling sequence. The setup was made up of the Arduino-based controller, the rotating platform, the stepper-motor driver, the limit switch, the IR bottle sensor, the water pump, the MOSFET driver, the buzzer, and the LabVIEW monitoring interface.

The pump activation interval was configured as 5 s for each bottle, and the LabVIEW VI was set to communicate with the Arduino through the USB serial interface at a baud rate of 9600.

The evaluation mainly concentrated on how the filling sequence operates and how the embedded controller interacts with the monitoring interface. Quantitative measurements of the dispensed volume, positioning error, cycle time, and repeatability were not recorded during the testing.

2. Functional Demonstration

During the demonstration that was carried out, all three bottle positions were observed to be filled successfully, with the pump being kept active for the programmed interval of 5 s for each bottle. The controller was configured in such a way that the rotating platform advances step by step from one bottle position to the next.

The system also includes several supporting functions in its firmware, namely the homing logic, the bottle-presence checks, the pump shutoff conditions, the audible feedback, and the serial status reporting. All of these functions form a part of the implemented firmware. However, the available test record does not contain separate quantitative results for the detection accuracy, the fault-response time, or the reliability over repeated cycles.

TABLE I
SUMMARY OF THE FUNCTIONAL DEMONSTRATION

Parameter	Implemented Configuration or Observation
Number of bottle positions	3
Pump activation interval	5 s per bottle
Total programmed pump-on time	15 s for three completed fills
Bottle-presence detection	IR sensor, active LOW
Sensor-stability interval	100 ms
Platform reference	Limit-switch-based homing
Nominal steps per revolution	4096 half-steps
Nominal indexing sequence	1365, 1365, 1366 steps
Serial communication rate	9600 baud
Filling demonstration	All three bottle positions filled successfully

The total programmed pump-on time of 15 s is obtained by adding up the three individual dispensing intervals of 5 s each and a delay of 100ms. It does not include the time taken for homing, detection, platform movement, and other processing, and therefore it cannot be taken as the total duration of one complete cycle.

3. Interpretation of Results

The successful demonstration of the three positions shows that the prototype is able to carry out its

intended time-based filling sequence under the conditions in which it was demonstrated. The combination of bottle detection, reference-position homing, motorized indexing, pump switching, and status reporting shows how several sensing and actuation functions are made to work together inside a single embed-ded system. The LabVIEW interface gives visibility into the operating states reported by the controller, while it has no responsibility for the physical control sequence. Because of this separation, the machine is able to execute its programmed operations independently of the monitoring application.

The results should be understood as a functional demonstration and not as a quantitative validation of performance. In particular, the fixed pump duration does not guarantee a specific dispensed volume, because the prototype does not measure the flow or the liquid level. Additional repeated trials and quantitative measurements would be needed to evaluate the filling consistency, the positioning accuracy, and the operational reliability of the system.

VI. DISCUSSION

The proposed implementation brings together mechanical positioning, bottle detection using a sensor, timed dispensing and computer-based monitoring of the status. The rotating platform gives access to the three bottle positions one after another, and the limit switch is used to set a reference point for the programmed indexing sequence. The IR sensor makes sure that the normal filling routine does not start unless a bottle is detected.

Since a dedicated motor driver and a MOSFET-based pump interface are used, the Arduino is able to control the actuators which need separate electrical drive stages. The firmware also has interruption-handling logic, which is meant to stop the dispensing when the AUTO input is disabled or when a bottle is removed. A homing routine with a fixed limit gives a defined response for the case where the reference switch is not detected. One important point in this implementation is the difference between timed

dispensing and volume-controlled dispensing. In the present system, the duration for which the pump runs is controlled, and the quantity of water which is delivered is not measured. Because of this, the final volume can be affected by differences in the pump flow, the supply conditions or the characteristics of the tubing. To have direct filling control based on feedback, a flow sensor or a liquid-level feedback mechanism would be required.

The LabVIEW interface adds a monitoring layer by using the serial status messages. As the VI is read-only in the present implementation, it gives no start/stop commands for supervision and no direct control of the actuators. These features can be added in a future version if bidirectional communication and a proper command-validation logic are implemented. The present evaluation of the prototype is limited to the functional demonstration which was reported. More testing over repeated operating cycles is needed before any conclusion can be made about the consistency, the reliability, the throughput or the suitability of the system for industrial production.

Limitations and Future Scope

The present prototype gives a functional implementation of sequential bottle filling which is based on time. There are several improvements which can be made to extend its capabilities.

Firstly, a flow sensor can be added to measure the quantity of water which is dispensed, so that the pump can be controlled on the basis of volume. As another option, a suitable liquid-level sensor can be used in the cases where the filling up to a target level is preferred.

Secondly, the LabVIEW interface can be extended so that it shows a bottle counter, a filling-status indicator and an event log which is structured. Bidirectional serial communication can also be introduced, so that validated start and stop commands can be given from the monitoring interface.

Thirdly, the firmware can be redesigned around a non-blocking state machine, which would improve the coordination between the sensor monitoring, the

operation of the actuators and the communication. This approach can also make it easier to expand the system in the future.

Fourthly, testing over repeated cycles can be carried out to measure the actual filling volume, the variation between the bottles, the repeatability of the positioning, the cycle time and the behaviour of the fault response. These measurements would give a stronger basis for the quantitative evaluation of the performance.

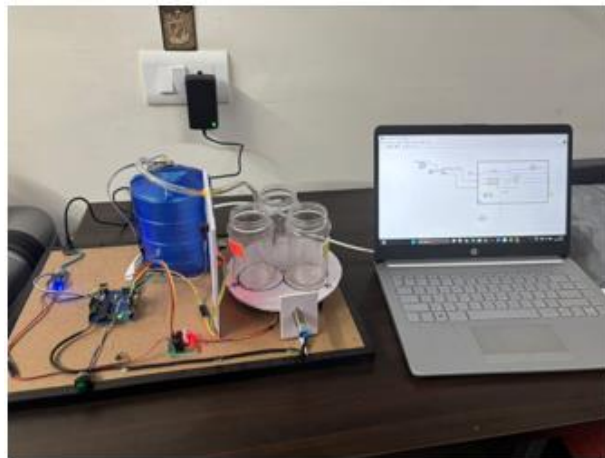


Fig. 4. Hardware Assembled



Fig. 5. Monitoring in LabView

Finally, wireless serial communication and data logging with timestamps can be added, which would allow remote monitoring and later analysis of the operation of the system.

VIII. CONCLUSION

This paper presented the design and implementation of a three-bottle automatic water filling system which is built using an Arduino UNO

and LabVIEW. The prototype combines an IR-based mechanism for detecting the bottles, a homing procedure which uses a limit switch, a rotating platform driven by a stepper motor, a water pump switched through a MOSFET, and sound indications for the status. The filling sequence is coordinated by the Arduino firmware, which also moves the platform from one bottle position to the next, and the pump is kept on for a programmed time of 5 s for each bottle.

A LabVIEW virtual instrument is connected to the Arduino through a USB serial connection at 9600 baud, and the status messages generated by the controller are displayed by it. The physical process is executed by the embedded controller on its own, without depending on the monitoring interface. In the functional demonstration which was reported, all three bottle positions were filled successfully.

This implementation shows that embedded control, electromechanical positioning, detection using sensors and virtual instrumentation can be integrated in a single prototype. But the present design uses time-based dispensing, so the volume which is delivered is not measured directly. In the future work, the focus will be on filling based on feedback, a wider supervisory monitoring, and quantitative testing of the filling consistency and the repeatability of the system.

REFERENCES

1. K. N. Tupsakri, H. R. Gowda, V. T. Kaling, M. Manjunath, and S. R. Gunaga, "Automatic Bottle Filling and Capping System," in Proc. 1st Int. Conf. on Intelligent and Sustainable Power and Energy Systems (IS-PES), SciTePress, 2023, pp. 177–182, doi: 10.5220/0012507800003808.
2. M. A. Fahmi, "Soy milk Bottle Filling and Closing Control System Using Arduino Uno Microcontroller," undergraduate thesis, Mechatronics Engineering Technology, Politeknik Negeri Jember.
3. Design of a Real Time Remote Monitoring and Controlling System Using LabVIEW," IEEE Conference Publication, Nov. 2023. [Online]. Available:

<https://ieeexplore.ieee.org/document/1030693>
3.

4. Jaskuła et al., Fast MEMS Application Prototyping Using Arduino/LabView Pair," Measurement Automation Monitoring, p. 548, Dec. 2015. [Online]. Available:
https://yadda.icm.edu.pl/baztech/element/bwm/eta1.element.baztech-eb19f2f9-a490-4828-8770-e571bbc74af8/c/jaskulaf_astM_AM122015.pdf.