Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2023, 10(2): 9-21

Research Article ISSN: 2394 - 658X

LabVIEW-Based Design of Robotic Arm-Based SCADA for Bottle Filling Plant

Reshail Abbasi[1], Muhammad Faisal Buland Iqbal*[2], Shehar Bano[3] and Hafiz Anns Nadeem[4]

[1]*Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, Pakistan*
[2] *Key Laboratory of Intelligent Computing & Information Processing, Ministry of Education, Xiangtan University, Xiangtan 411105, PR China*
[3] *School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, PR China*
[4] *School of Computing, Engineering, and Digital Technologies, Teesside University, UK*
*faisal@smail.xtu.edu.cn (corresponding author) **

ABSTRACT

The globe has undergone an industrial revolution. Automation is the process of handling various industrial operations without the assistance of humans by using control systems, such as robots, computers, and other technology. Most industries use robotic arms to control various operations, but the most cutting-edge ones are difficult for everyone to adopt since they are so expensive. Some of them use the Industrial Internet of Things (IIOT) and have a very large and complex mechanical assembly, which is not possible in developing countries or in industries that are just starting out. Implementing some of the ideas, which are particularly difficult to integrate with the current PLCs, will also drive up the cost of the procedure. Therefore, this research aims to create a digital robotic arm that will address all the issues mentioned. This will not need any hydraulics and will have a straightforward mechanical assembly, lowering maintenance costs. The robotic arm will not require a separate controller; since it is digital, it will be simple to incorporate into the system's PLC. To pick up a filled bottle and transfer it toward the packaging area, a SCADA for the bottle-filling facility was created in LabVIEW using this robotic arm design. This arm successfully picked up the bottle and moved toward the packing area. The system's control panel provides access to all of the process' digital signals. In this research, all predetermined tasks were completed, LabVIEW was used to create the SCADA for the bottle-filling plant, and the robotic arm's design and functionality were tested via SCADA to attest to the justification as mentioned earlier.

Key words: PLC (Programmable Logic Controller), LabVIEW, Robotics, Automation, SCADA.

INTRODUCTION

The automation of industries made life for humans relatively simple and improved their safety level. Automation's main objective is to minimise human involvement in industrial processes. Threats to human life can be easily reduced in this way. It decreases the likelihood of electric shocks or mishaps caused by hands or clothing being entangled in equipment or conveyors. Although automation has specific problems, they are outweighed by its advantages. Process automation uses computer technology and software engineering to improve the productivity and safety of power plants and factories in various sectors, including paper, mining, cement, etc. [9]. Any process automation impossible without the use of a robot or robotic arm. A robot is a mechanical device that operates automatically while being supervised by a human. A robot may operate directly under the direction of a computer programme or directly under the guidance of a human. The field of robotic applications has advanced significantly over the past 20 years. Future uses in the military, space-based research, and daily activities are anticipated to be numerous. Some processes have been completely automated [15].

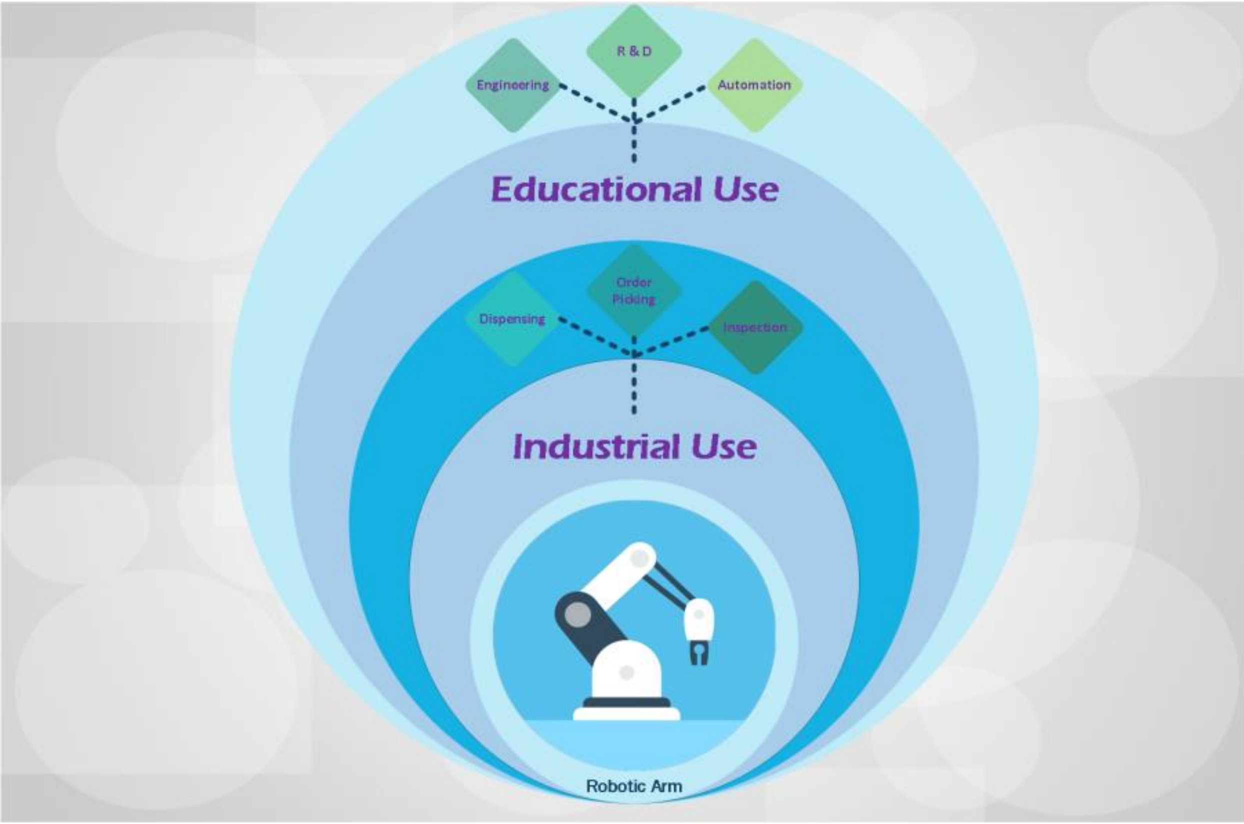

Fig .1 Robotic arm's applications

Nowadays, robotic arms are used worldwide to carry out jobs more quickly and effectively than humans. Many businesses, such as manufacturing, agriculture, food preparation, health care, etc., use robots. The industry influences any robotic arm's design it will operate. Robotic arms have a proven track record of dependability and an exceptional capacity to complete tasks quickly and effectively without the risks and issues associated with the human mistake. Robotic arms are also made to do challenging activities and procedures that require a degree of precision and accuracy that humans cannot match. Robot arms are capable of performing tasks repeatedly and accurately when they are correctly programmed. Robotic arms can handle materials that are brittle or need particular treatment with great accuracy by using proximity and pressure sensors.

Studies that detailed the creation of a robotic arm via a series of experiments were carried out in 2009. Three experiments were conducted. The learner may move a few sliders in the first activity to directly control the robotic arm. Each slider controls a motor. Students' first robotics-related homework would demonstrate how the robotic arm works. The second experiment demonstrates the effects of gravity at different data rates on a robotic arm. In the third experiment, students must program the settings for each motor for the robot's motions at each new level. In order to execute a task assigned by the instructor, students would assemble in this movement. Students can see how to program a robotic arm with gravity considerations in mind by participating in the Effect of Gravity Experiment. The Trajectory Planning Experiment offers students the chance to practice inverse kinematics. As a live webcam feed is required for each study, this lab's primary concern is a need for more bandwidth. As a result, efforts must be made to lower the lab's bandwidth requirements while maintaining the necessary visual, audio, and possibly haptic feedback for the user. Simulators must also be kept to a minimum so that students do not get the impression that they are in a virtual lab rather than an online lab [12].

Another article described the project's goal: to teach a robotic arm to utilise fundamental tools in a faraway laboratory. Acrylic sheet is used to make the robotic arm. Students' personal and remote lab computers may access Lab VIEW control panels. LabVIEW is used to link the student's computer to the remote lab computer. The equipment is linked to a computer in a separate laboratory [2]. The microprocessor manages the robotic arm, which also controls the lab equipment. A webcam with video chat is utilized to determine the location of the robotic arm. Based on student comments, the robotic arm is trained to interact with remote laboratory equipment. In this project, the robotic arm is commanded to do basic motions. It may be used for many things by including sensors or image tracking.

The LabVIEW Embedded Module provides a complete graphical development environment for embedded design for ARM Microcontrollers. The LabVIEW graphical development environment and ARM microcontrollers are easily integrated by this module, which was jointly created by Keil-An ARM Company and National Instruments [13].

According to the authors of this study [3], the beverage industry might become more productive by using effective production methods in industrial liquid/yogurt filling machines (YFM). This article described the second phase of our YFM's control architecture, which is based on industry 4.0 principles and incorporates an NFC platform to enhance customer satisfaction. During this pandemic period, wireless technology has grown ubiquitous and pervasive for custom-made goods. The YFM's fundamental components have been specified. The YFM is designed with a PFC-based controller, and high-level control architecture is used to construct fully automated filling processes.

Principal findings of their testing included: It calls for sophisticated Internet-based production process tracking and product development tracking. During the manufacturing process, intelligent control of the production line may be utilized to stop the line or activate field equipment over the Internet. A robotic arm is an example of sophisticated robotics used in production to carry out several jobs simultaneously. An innovative bidirectional strategy for changing the hardware and programming algorithm is now being researched in order to build a long-term model for cognitive computing. An NFC tag may be used to turn the item into a smart one (RFID technologies). This prototype model included the most current cutting-edge technologies that link industrial Internet of things (IIoT), near field communication (NFC) module, raspberry pi module, robotic arm, and cyber-physical systems (field devices). Their team created a prototype that will be used to educate, impart knowledge, and spark creative thinking. Their team was made up of researchers, academics, and industrial engineering grads.

Individual student projects and experiential learning are incorporated in this research. Up until client input is received, it depicts a customized manufacturing line with several stations, controllers, and operations, in addition to supply chain logistics. In addition, researchers utilized FluidSIM-MecLab, Ansys, MATLAB, and Simulink, which were previously accessible, to simulate and validate the designs and algorithms. They simplified the acquisition and processing of data online. A combination of formal, casual, and non-formal learning strategies were used to generate a tangible product.

The authors of this work [4] examine the issue of medication planogram optimization for robotic dispensing systems (RDSs) in mail-order pharmacy automation (MOPA) facilities. A MOPA is used by a high-throughput fulfillment center that handles many prescription order submissions. To accommodate the rising demand, each RDS unit in MOPA facilities is equipped with an auto-dispenser system and a robot arm that automatically count and distribute medications. An RDS planogram depicts the distribution of medications between one RDS unit and other RDS units. By optimizing the RDS planogram, the three objectives of the study—medication association, RDS workload balance, and robot arm travel distance are attained.

For each of the several performance evaluation criteria, the algorithm with the most exceptional performance is determined. This work formulates the simultaneous robotic dispensing planogram optimization issue as a MOO problem. Using ARM, the rules of drug associations are retrieved. Throughout the optimization process, the correlations between the three variables (medication association, machine workload balance, and total robot arm trip distance) are examined. Taking dispenser number, placement, and assigned responsibility into account, the proposed method develops the optimal Plano-gram design. The findings show that the challenge's size and criteria significantly influence the performance of various algorithms. SPEA-II provides a broader range and additional options for small-scale problems.

PAES has the lowest computational cost when handling minor issues, but as the problem's size grows, the computational difference diminishes in significance. Adaptability is a concern with planograms that is practical to have. According to the study of the examined MOPA, planogram results do not change significantly over a short time period, such as a year. The suggested model framework, which operates under the assumption that all demand can be satisfied in a single shift and that system downtime is not taken into consideration, will be used in future research to address the medication planogram optimisation issue. Future iterations of the suggested model will take into account circumstances in which there is an excess of demand and the demand may be deferred until the next shift. When developing stochastic or robust optimization models, the stochastic features of system aspects like counting speed, demand volatility, and downtime will also be considered. Alternative knee-based techniques should be looked into to improve the quality of the Pareto-optimal solution set since only the ASA method is employed in this study to identify knee regions.

The paper [5], discussed an embedded PLC design and its use in system automation. The ARM Microcontroller and the embedded LabVIEW software module are used to create the embedded PLC. PLCs, contemporary microprocessor-based electronics, are utilised to build complex control systems that are employed in almost every sector requiring automation. This PLC is adaptable, durable, and compatible with commercial parts. Additionally, new technologies may be added to it. Due to its dependability and simplicity of use, the PLC has become a crucial controller in industry today. In addition to controlling the voltage and frequency of power sources, it is used to control a range of mechanical movements in giant machines. Compared to other IEC 61131-3 standard languages, FBD has been found to be more approachable.

Because input and output ports are directly coupled to function blocks, FBD is easy to use and comprehend. It is well knowledge that embedded PLCs are less expensive than traditional PLCs. If the client wishes to expand the number of IO ports, there will be just a little increase in cost, since the processor board already has all of them.

The design and deployment of a laboratory-scale automated visual inspection system for the production line of the beverage sector are described in a journal paper [6]. The beverage bottle is examined for faults using the image processing technique in a case study. A number of issues, such as inaccurate labelling and liquid level, may be found using the image processing technique. A conveyor belt prototype was created in the lab, along with a filling plant prototype, to test the simulation results. This paper examines a case study of the Coca-Cola bottling plant in which image databases are used to identify various types of image defects utilizing image processing software. This study provides design parameters for a laboratory prototype vision system that simulates a bottling facility using a conveyor belt and a software-based vision system. As the photograph is taken, a photoelectric sensor detects the object's proximity and checks for faults.

According to the writers in [7], vending machines are crucial to the industrialisation process because they meet the pressing requirements of society. Because a system has to be adaptable, efficient, and cost-effective in today's fiercely competitive industrial world, machine automation is crucial. Automation has the potential to radically disrupt practically any sector since it is such a broad idea. In order to make soft drinks like Fanta and Maaza in a creative fashion, this project uses automation utilising PLC and SCADA for bottling, packaging, and bottling. New beverages are launched into the market every day. This campaign seeks to encourage small enterprises to automate the production and bottling of beverages. This work will reduce an industry's operational time, while also improving the system's accuracy, flexibility, and efficiency. There has been created an automated method for brewing drinks. The aforementioned arrangement has a wide variety of automation-related applications.

In the world of mass manufacturing, all units must be processed and monitored in a short length of time, increasing output. In this project, we demonstrate how PLC and SCADA may be utilised to automate the beverage preparation process using a vending machine. PLC and SCADA may be expensive to set up initially, but the advantages far outweigh the costs of building up an automated machine like this. Depending on the complexity and nature of the product produced, the kind of equipment utilised, and the operator's abilities, a different level of SCADA capability will be needed [14]. The project also incorporates the use of SCADA, which enables remote system operation. The SCADA system oversees all monitoring operations. Using SCADA, we are able to start and stop the system while sitting at a distance from the process. This concept facilitates the identification of process defects.

The importance of productivity in maximising the utilisation of resources like labour and materials is covered in this work [8]. Only after managing and reducing operating time is it possible to reduce cycle time by the greatest amount. Productivity may be increased with a better strategy for maximising resource use, labour effectiveness, and facility layout, among other things. This investigation seeks to ascertain if bottle filling may be accelerated using a PLC-controlled belt conveyer. The automated bottle-filling equipment includes a time limit in order to fill bottles accurately and quickly. With the aid of PLC programming, this is achievable. The reasoning used in this research to determine the bottle's location on the conveyor and its state—precisely, whether it is full or not—is described in depth. Additional sensors and a PLC are employed as inputs into the system. This will achieve the goal of quantity with quality for any manufacturing organisation by ensuring the accuracy of the amount to be filled and drastically cutting the cycle time to fill one bottle.

The current approach will have a variety of applications in the area of automation, particularly in sectors that depend on mass production and need rapid processing and handling of a huge volume of components to boost productivity. Although installation costs are high, it has a very long operational life. The system is more adaptable, dependable, and economical. The approach reduces both labor hours and cycle times. The results of a bottle-filling technique controlled by a PLC. The installation of 5 nozzles allows them to fill five bottles at once, decreasing filling time and improving output; nevertheless, human labor is still needed for pallet loading and unloading from the conveyer belt. Human interaction could be unnecessary if robotic arms that can be trained to do loading and unloading operations were used. In the future, this system will be able to accommodate more than five bottles.

According to the authors of the study [9], Factory Automation experts are looking for approaches to develop SCADA systems using Visual Basic for a number of reasons. The bulk of SCADA applications need a limited number of user input parameters and data recording, monitoring, and trending features that are user-viewable. Conventional SCADA-based systems provide the bulk of organizational features, but at exorbitant costs for small OEM clients. Provide frontend graphics, trends, communication, and warnings using Visual Basic's integrated event-based user interface design, SQL database access capabilities, and off-the-shelf plug-ins such as ActiveX controls or DLL, and MSCOMM components. Users, developers, OEMs, and integrators may create applications for industrial HMI or SCADA.

The VB-based SCADA that was built is compatible with a real-time view of the industrial process, faster downtime and fault detection times, and operator safety. OEM firms may rapidly adopt the usage of customized SCADA as a component of their industrial machine automation in order to optimize plant process visualisation, detect faults faster, and aid in the replacement of people in hazardous locations. This article illustrates how VB may be used to replace SCADA in packaging machines. The SCADA is intended to offer a real-time view of the industrial plant, expedite issue diagnostics, and safeguard operational personnel. Incorporating SCADA into an organization's industrial automation system may enhance plant process visualization, expedite issue detection, and aid in staff replacement in hazardous environments.

The research aims to create a virtual training simulator for the Programmable Logic Controller, according to the authors of this work [10]. Due to the software's extensive library of process models, which may simulate the functional behavior of real production processes, students can get experience using PLC tools to handle a variety of industrial processes. The analytic capabilities of this simulator provide the rapid gathering of data on the output performance of the controlled system. The descriptive analysis approach used in this study consists of five steps: problem identification, literature research, instrument design, implementation, testing, and assessment. This program is implemented using LabVIEW on a PC. Using an Arduino and a signal conditioning circuit, control signals are sent between the actual PLC (any PLC will do) and the PC. Three commercial models were created as a result of this study. LabVIEW is used to implement this program on the PC (Laboratory Laboratory Workbench). Control signals are sent between the real PLC (any PLC may be used) and the PC using an Arduino and a signal conditioning circuit.

As a consequence of this investigation, three commercial models were produced. Two traffic signal control systems comprise the first production system. The water level control of the tank is the second type produced. The third industrial design is a control system for an elevator. The first plant's analysis tools consist of information on the aggregate waiting periods of all vehicles passing through the intersection. The second plant's analytic tools include a system performance graph with information on the maximum overshoot, settling time, and steady-state error. For the third plant, the total average floor displacement time of all elevator users acts as the analytical instrument.

As a consequence of this research, a virtual education simulation including three plant simulations and accompanying analytical tools was created. This research seeks to enhance students' understanding of PLC and control mechanisms. The user may alter the simulator's dynamic settings. Utilizing this virtual simulation system and analytical tools, students will be able to experience controlling various industrial processes using different PLC functions, hence boosting their understanding of PLC and control approaches.

The researcher highlighted how, in a separate investigation, many sensors are utilized to conduct quality checks on different components of the final product from the small plant. Among the sensors are LDRs, PH meters, Thermocouples, and optical fiber sensors (OFS). These little industrial prototype's evaluation criteria include pH, level, temperature, weight, and turbidity. A bottle-filling facility scale model has a number of sensors and actuators installed. An optical sensor known as an OFS converts weight into an electrical signal. The microbending phenomenon is its basis. Through Compact Field Point, which also provides actuator signals, the LabVIEW Data Acquisition collects all sensor signals [11].

After reading the preceding material, we may use the strategies offered to address our problem statement. Without the need for hydraulics or substantial mechanical components, a simple and low-maintenance robotic arm may be made in its place. This would decrease the assembly's maintenance expenses and prolong the robotic arm's lifespan. The bottle-filling process may be automated using a single digital control panel that can quickly connect to the existing SCADA robotic arm system. A separate controller will not be necessary for the arm. The same PLC overseeing the process now would control this robotic arm, which would instantly connect to the existing system. In the past, sorting entailed loading items onto a conveyor and sorting only one item at a time. This process takes much time and costs money. Our research aims at sorting simultaneously [16].

In section 2, we will discuss the methodology, this explains the method used to complete the tasks at hand. After that, section 3 discusses the project's industrial applicability. This tells us about how the research went what was achieved and the results during this process in section 4. Section 5 discussed the conclusion and future recommendations.

METHODOLOGY

Because the work was complex, system engineering was the ideal approach to take. System engineering will be used to break the work into manageable components, and each component will be merged to produce the task's ultimate form.

System engineering

Systems engineering is a multidisciplinary and inclusive process that makes use of systems concepts and ideas in addition to scientific, technical, and management techniques in order to successfully design, test, utilize, and eventually retire engineered systems. System engineering places a primary emphasis, early on in the product development cycle, on providing, recognizing, balancing, and integrating the objectives, purposes, and success criteria of various stakeholders. At the same time, identifying the client's real or anticipated requirements, functional ideas, and needed functionality, as well as establishing an acceptable lifecycle model, process methodology, and governance structures. While keeping in mind the varying degrees of complexity, unpredictability, change, and variation; as well as developing and evaluating viable solutions.

The management of risk is the fundamental goal of any endeavor involving Systems Engineering. This includes the risk of failing to provide the client with what they want and need, the risk of a late delivery, the risk of a high cost, and the risk of unpleasant unexpected consequences. One of the criteria that may be used to evaluate the worth of the work that is done in the field of systems engineering is the degree to which a particular risk is mitigated. On the other side, a measure of acceptability is the degree of excess risk that results from the absence of a System Engineering activity.

Industrial Use

Industrial applications use system engineering to build manufacturing facilities, retire aging projects, and solve complicated issues. The difficult or massive job is divided into manageable bits. Before the little components are integrated to make the final goal or project, they are each independently solved, developed, and tested to check whether they match the criteria. System engineering will be used to break the work down into smaller components. To develop the necessary SCADA, each individual component will first be built and validated individually, after which it will be included into the larger system. In order to test and validate the stated work, a software program is going to be employed. LABVIEW is the name of the program that will serve as the application of choice.

The LabVIEW Software

Systems engineering software named LabVIEW was created for applications that need testing, measurement, and control as well as rapid access to hardware and data analysis. The LabVIEW programming environment simplifies hardware integration for engineering applications by facilitating standard data gathering from NI and third-party devices. LabVIEW simplifies programming so that engineers may focus on solving technical problems. Results may be viewed straight away because of LabVIEW's integrated drag-and-drop engineering user interface creation and integrated data viewers. Either utilise the math and signal processing IP that is currently given or build own library out of a range of tools transform data into useful business outcomes. LabVIEW can communicate with other applications and open-source programming languages, libraries from them may be utilised, demonstrating that other engineering tools are interoperable.

The LabVIEW programming language is named "G." The letter G stands for graphic. LabVIEW is more productive than traditional programming languages since developing applications takes much less time. Traditional programming languages take weeks or months to create. Using a robust graphical programming language, may do the same task more quickly.

The given work was broken down into doable parts using a system engineering technique to ensure efficient completion. Using this approach, the given work may be solved fast. All the smaller components will subsequently be combined into the final product or job. Below is a detailed breakdown of the parts that were developed and put together to form the completed SCADA.

Robotic arm

The primary objective of creating a two-dimensional robotic arm in LabVIEW was to create one that could pick up a full bottle and set it on a conveyor belt in the packing area. The manner of selecting relies on the use of the robotic arm; for instance, if it is used to pick up metallic bottles or other metallic objects, we may magnetize them according to the application's requirements. If the metallic claw is designed to pick up glass bottles, a stepper motor will be utilized to regulate its opening and closing. The primary objective was to construct a SCADA utilizing a robotic arm. Designing the robotic arm in LabVIEW to perform the required side-to-side and up-down motions was the first stage in developing the SCADA.

This was accomplished by selecting a conveyer from the DSC module and using its height and position-altering capabilities to move it vertically, horizontally, or sideways. This allowed the robotic arm to move as required after that.

The Bottle

The following operation in the SCADA system based on animation included moving the bottle in line with requirements. The bottle in the illustration was intended to move up and down as well as side to side, exactly like the robotic arm. So, a bottle was picked from the image navigator of the DSC module, and property nodes controlled its motion. The property node was used to adjust the bottle's height and placement following the task's specifications. The robot then moved the bottle in the following action. To choose the bottle, a Boolean was thus created; Once the pick button is pressed, the robot will grab the bottle with its robotic arm and transfer it to the desired location. After the bottle has been picked up by the robotic arm, the height or position of the robotic arm will change, which will cause the bottle's height and position to likewise shift. A robotic arm will be used by the SCADA system to pick up the bottle from the filling side of the machine and place it on the conveyer belt on the packaging side of the machine.

Conveyor

Conveyors are used in a range of sectors, factories, and retail establishments across the world. The conveyor is often used in the manufacturing business to transport raw materials, finished items, etc. A belt surrounds the drums from beginning to end, causing the drums to rotate and providing a circular motion. As the drums rotate, both the belt and anything is put on it advance. Depending on the circumstances, the conveyer belt's speed may be modified.

PROCESS OF AUTOMATION

Figure 2 depicts the flow chart for the automation process of the specified job:

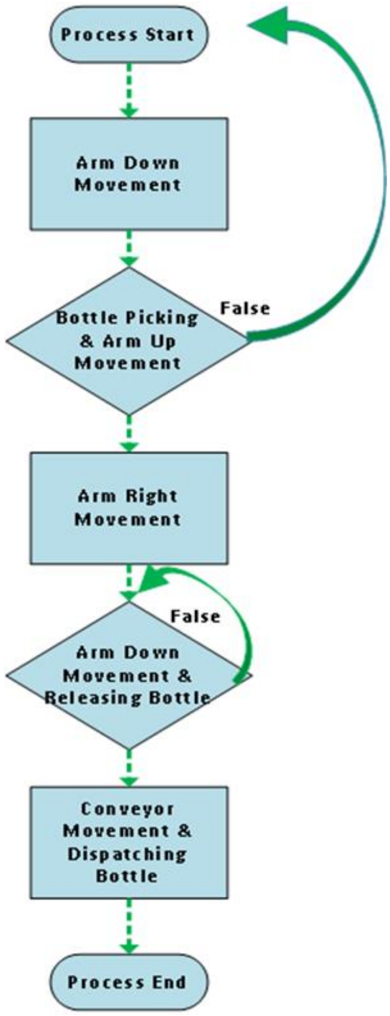

Fig. 2 Flow chart of the process

Automating any process is critical in the business since it ensures seamless operation and decreases operating costs by reducing the need to hire a staff person to supervise each stage of the process. Because the use of MCC and other automation technologies allows for pinpointing the issue, automation also makes tracing any process faults easier. It shortens the time required to correct any undesired faults in the running activity and makes them easy to correct.

Process Start Loop

As shown in fig 3 (1), At this point, a while loop was implemented to begin the automation. Shift registers were added to the while loop to save the initial value and return it to the loop's beginning. After seeing the beginning position and picking point of the robotic arm, the shift register's initial value, in this instance 100, was entered. The robotic arm needed to go from 100 to 420. Once the shift register was introduced, the starting value was raised by one, so whenever the while loop runs, an increment block will add one to the stored value in the shift register. A comparison block will be added to check both the new value and the one that was required for the robotic arm to be able to pick up the bottle after the increment. The new value will be verified and compared to the required value. If the value that is coming in is the same as the value that has been saved or referenced, the comparator will output 1. Upon detecting a one at the comparator's output, the operation will go on to the bottle choosing loop.

A Boolean false was added and connected to the right movement of the robotic arm to guarantee that it would only travel in the correct direction. The robotic arm can only descend downward before moving on to the next loop. In order to guarantee that this loop will not repeat itself when the comparator's output is one, a not gate from the comparator's output was also added to the Boolean start of the current loop. This loop also included a delay to smooth out and make it easier to see the robotic arm's action.

Bottle Pick

Following the completion of the first loop, the program will proceed to the next phase. In this phase, the arm will pick up the bottle and reposition it at the top of the stack. To do this, a shift register was added for the initial value and a while loop was implemented inside a case structure. The depicted scenario's initial value for the robotic arm's starting position was 420, as can be seen in Fig 3(2). In this loop, a decrease rather than an increment was introduced since the robotic arm was anticipated to go higher. Consequently, a decrement was added to the shift register, incrementing the shift register's value by one each time the while-loop runs. In addition, a comparator was added so that the value of the shift register could be compared to the value that was already recorded in the comparator. When the value coming from the comparator and the value stored in the shift register are equal, we will get a one at the output, which means the program may go on to the next section.

The program's up/down block and NAND gate were also attached to this output. While the NAND gate's second input becomes one, it outputs 0 to the bottle pick, stopping the bottle pick program loop from continuing to run when the NAND gate's first input is always one. Here, the NAND gate functions solely as the not gate used in the previous loop. The output of the comparator also carries out the up/down command. The up/down block controls the robotic arm as well as the upward movement of the bottle. As a result, the robotic arm raises the bottle when it reaches the desired position. This phase was also given a delay to make it clearer.

In the fabricated account of the event, the robotic arm was fastened to a construction up/down block. In the event that the bottle picks condition is not met, the robotic arm will move up and down but will not pick up the bottle. However, as the process gets more automated, the robotic arm will automatically pick up the bottle as it lowers and return to its starting position once it rises. This will occur as the method becomes more automated.

Robotic arm right movement

The robotic arm is designed to travel in the proper direction once it has completed the task of picking up the bottle and relocating itself to its initial location. The program was expanded with an additional while loop to get the required result. This will cause the robotic arm to move to the right. For the right movement of the robotic arm, we added a register and set its initial value in the while loop. The shift register in the aforementioned example originally had the value 0. The Robotic was developed so that it could make a right turn after going from 0 to 500. Every time the while-loop is executed, the value that is stored in the shift register will grow by one as a result of the addition of the increment that was made after it. It was at this point that the reference value of 500 was introduced for the decrement comparator. If the value coming from the shift register and the comparator reference are identical, the comparator will output 1. The right movement control receives this output. The robotic arm and bottle will shift to the right at 500 when desired value for the study is reached. The output of the comparator was also provided to the arm-down movement block, which will transfer control of the program to the next iteration of the loop. As a result of this action, the arm will drop, and the bottle will be put on the packing conveyor. A not gate was also utilized, as shown in Figure 3(3), in order to stop this loop from continuing once the required data were obtained. This is represented in the figure.

When the result of the comparator is 1, the signal passes via the not gate and then the loop control for the right arm movement. This will terminate the cycle and direct the program to the next stage. As demonstrated in the picture, a control slider was added to the while loop to guarantee that the robotic arm and bottle moved appropriately. There is also a delay in time.

Arm down movement

The following loop will guide the robotic arm downward and toward the conveyor belt. Next, go in the direction of the parking lot after dropping the bottle off there. A while loop was added to the case structure in order to get the desired outcome. The while loop transferred the robotic arm's starting value to the shift register again. The example shift register has a beginning value of 100. Then, an increment was introduced to increase the value by one each time the while-loop executed. Once the value compared by the comparator and the value stored in the shift register coincide, the output will be 1. The arm and bottle will move lower due to the loop's output, forcing the arm to slide down the block. The output of the comparator was also connected to the NOT gate, which prevents the while loop from running whenever the values in the comparator and the shift register are same. This is accomplished by connecting the NOT gate to the output of the comparator.

As demonstrated in Fig. 3 (4), the output will also handle the subsequent loop. After a bottle is received at the output, the program will proceed to the subsequent loop, which involves releasing the bottle onto the conveyor. When the instruction is received, both the bottle and the robotic arm will move lower due to the insertion of a delay and the local variable up/down.

Releasing bottle and upwards movement

According to the loop at hand, the robotic arm is set to drop the bottle onto the conveyor before rising. To do this, a while loop was added to the program. To provide a starting value for the loop, a shift register was once again added in order to do this. The bottle should rise after being dropped; therefore,, a decrement was also inserted using the shift register. Consequently, 420 was entered as the starting value in the shift register in this case. The after shift register decrement will subtract one after each while loop execution. In the above case, the comparator's reference value is 100, which it will get from the decrement.

Consequently, when the value from the shift register matches the value kept in the reference, the comparator's output will be 1. The local variable up/down receives this output and in this instance uses it to control the upward movement of the robotic arm. This local variable will elevate the robotic arm and release the bottle to the conveyor belt. The local variable of the arm-up movement is coupled to the NOT gate, which receives the comparator output. As a consequence, the loop ends when the arm is in the required position, and the software proceeds to the next stage as shown in Fig 3 (5).

The outermost while loop now has a whole new case structure that was added to it. Within this framework, the programming for the new bottle has been brought to a successful conclusion. After the final bottle has been released at the packing conveyor and the arm has returned to the top position, the structure of the case will ensure that a new bottle will spawn at the starting conveyor in order to replace the one that was just removed. The new bottle will be placed at a different location by setting the required constant to the position's local variable.

Movement of the arm to the left and resumption of the procedure

The robotic arm must be moved back to its initial starting position by moving it in the proper direction. As seen in the screenshot, a while loop was added to the case structure to achieve this goal. The while loop was given a shift register at the beginning to record the initial value of the loop. The beginning value in the example was 500. So that it would decrement one whenever the while-loop ran, a decrement was added after the shift register. A comparator uses the decrement's output to check if the value in the shift register is greater than or equal to the reference value, which in this instance is 0. The while-loop for the local variable will begin when the comparator output of 1 is reached, and this signal is also sent to the new bottle block. This means that the program will resume when this loop is complete, with a new bottle placed at the beginning.

The result of the decrement is sent to the local variable that controls whether the control moves to the left or right, as seen in figure 3. (6). The movement of the robotic arm to the left is under the direction of this geographical variable. While the while-loop is being executed, the robotic arm moves slightly to the left. A time delay was also used to see the loop correctly.

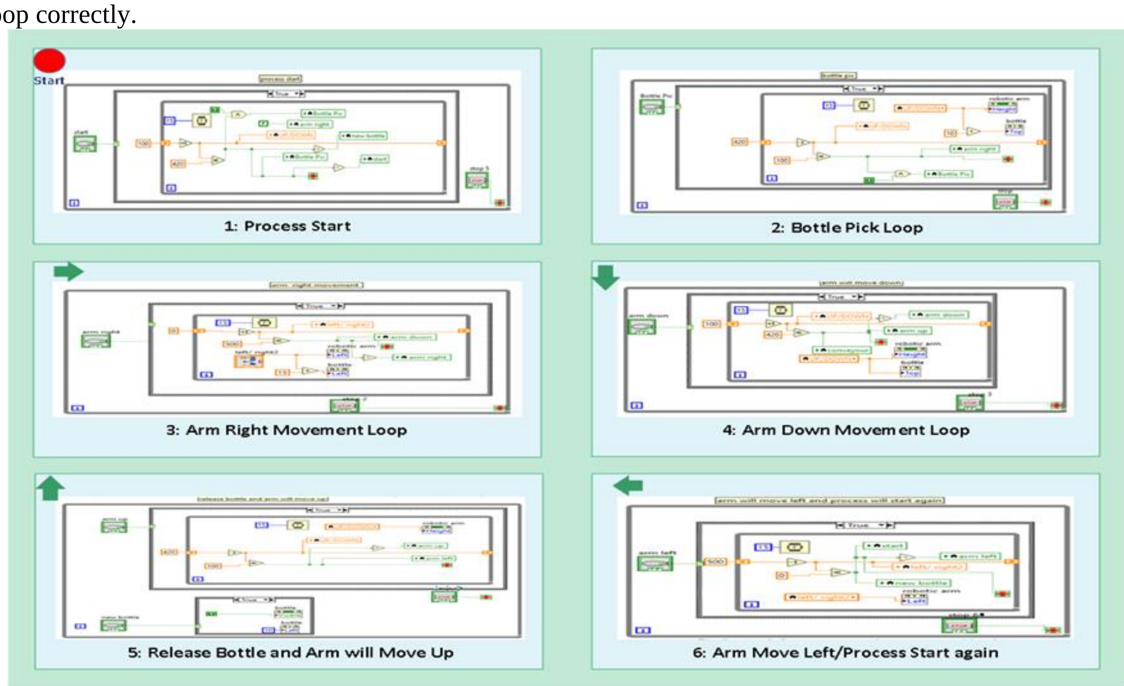

Fig. 3 Robotic Arm right movement

Conveyor Movement

After the bottle was placed on the conveyor by the robotic arm, the conveyor needed to be modified so that it could demonstrate how the bottle was moved from the conveyor to the packing truck. A while-loop was used in order to successfully finish this project. The true portion of the case structure was enhanced with the addition of a while loop. It was decided to construct a shift register in order to provide the while loop with the initial value for the bottle movement. In the above example, a while-loop had an increment after the shift register, which added one to the starting value each time the loop was executed. Several values were inserted into the shift register as the starting value and verified in the comparator as the finishing value to identify the beginning value. In this job, 770 served as the comparator's reference value, while 500 served as the bottle movement's beginning value. Because the left movement of the bottle was designated as the output of the increment, it was ensured that the bottle would progressively advance down the conveyor in the desired direction each time the while-loop was put into action. When both the reference value and the value that is coming from the shift register are identical, the inner while loop will finish, and the comparator output will change to 1. The while loop will not run until the required outcome has been obtained since this output is processed via a NOT gate before being transferred to the conveyor's local variable, as shown in figure 4.

A bottle position controller regulating the bottle's travel on the conveyor was installed inside the loop. The bottle will then be placed, followed by a robotic arm rising and moving along a conveyor toward the packaging truck. These two things will happen simultaneously. The procedure of collecting bottles and pushing them toward the packing conveyor will begin as soon as the bottle reaches the truck and the robotic arm returns to its starting position. This will cause the operation to continue. The conveyor belt that will be used in the company will be an environmentally friendly sensor-based belt that will begin moving as soon as it recognizes the presence of a bottle on it.

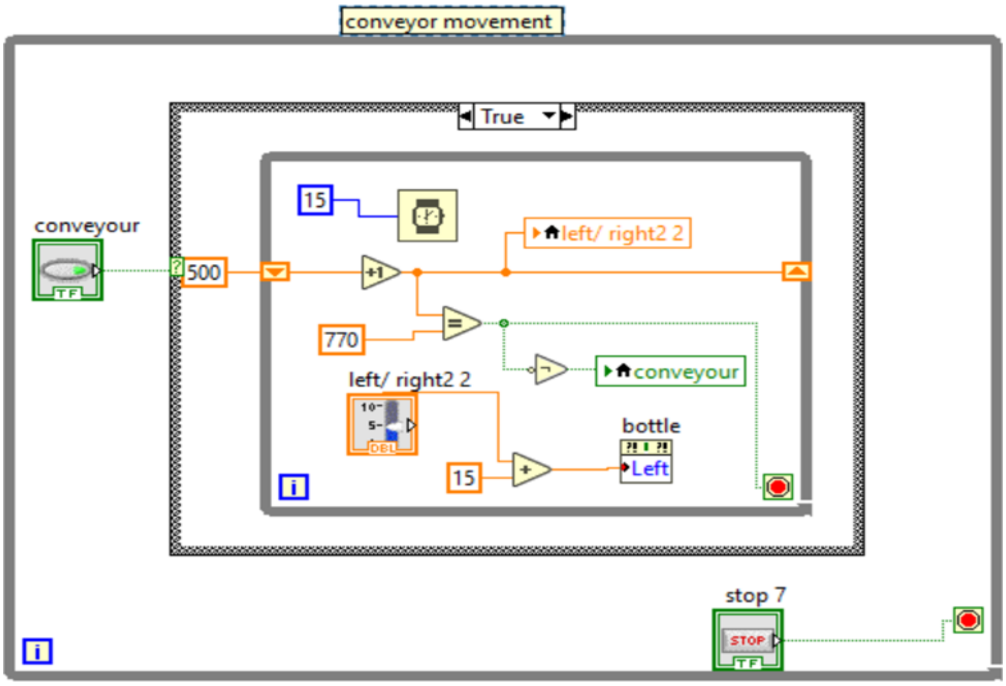

Fig. 4 Conveyor movement loop

RESULTS AND DISCUSSIONS

The final objective of the project was to develop a SCADA based on a robotic arm. LabVIEW was used to design and construct. The complete SCADA system based on robotic arms was monitored and controlled through a control panel. Figure 5 shows the designed control panel for the SCADA system with a robotic arm for the bottle-filling plant.

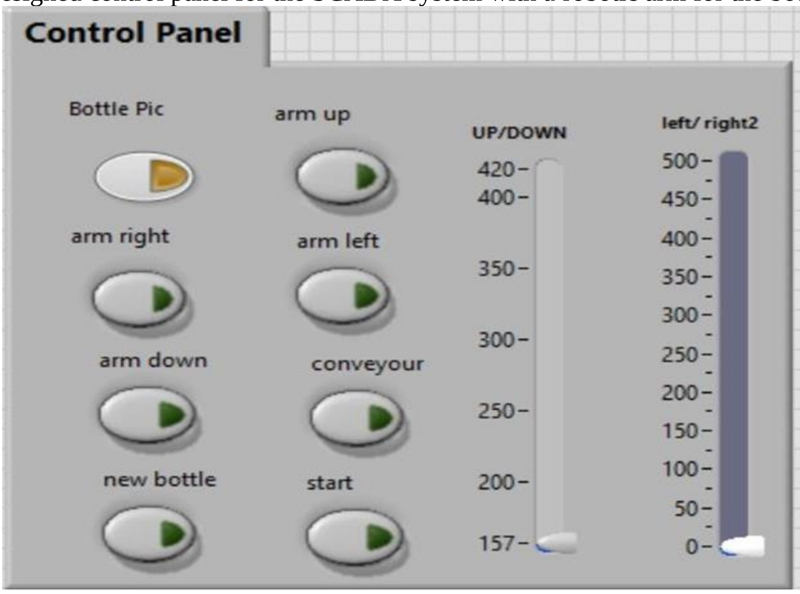

Fig. 5 Control Panel for SCADA

This control panel displays the movements of the robotic arm and the conveyor. Each move of the robotic arm is shown on the screen by means of digital signal indicators in real time. This control panel may also be used to regulate the motions of the conveyor and robotic arms. The no signal light in Figure 5 above indicates that the procedure has yet to start. Below is a thorough description of each phase of the process and its warning indications.

The process starts and the new bottle

Figure 6 (1) shows that when the procedure is signaled to begin, the robotic arm will begin lowering itself to pick up the bottle, and a fresh bottle will then appear on the conveyor. As a consequence of this, the figure that was provided earlier demonstrates that the process is beginning, and that a fresh bottle is now being used.

Bottle pick and Arm UP movement

Figure 6 (2) shows the robotic arm moving forward to its starting position after picking up the bottle. The control panel's bottle pick and up movement indicators are both on, suggesting that the procedure is running with the assistance of these two signals.

Right Movement indication

The robotic arm in figure 6 (3) will start to fall after picking up the bottle and travelling a certain distance in the right direction. The indication to move the arm to the right may be seen on the control panel. At this stage of the process, the arm travels in the required direction, and a strong signal indicates that the movement is appropriate.

Arm Down Indicator

As seen in figure 6 (4) below, The down movement signal indicator is switched on when the robotic arm moves lower. This shows that the robotic arm is being directed downward by the process. As a result, at this stage, the robotic arm will go downward and let go of the bottle so that it may be transported by the conveyer belt.

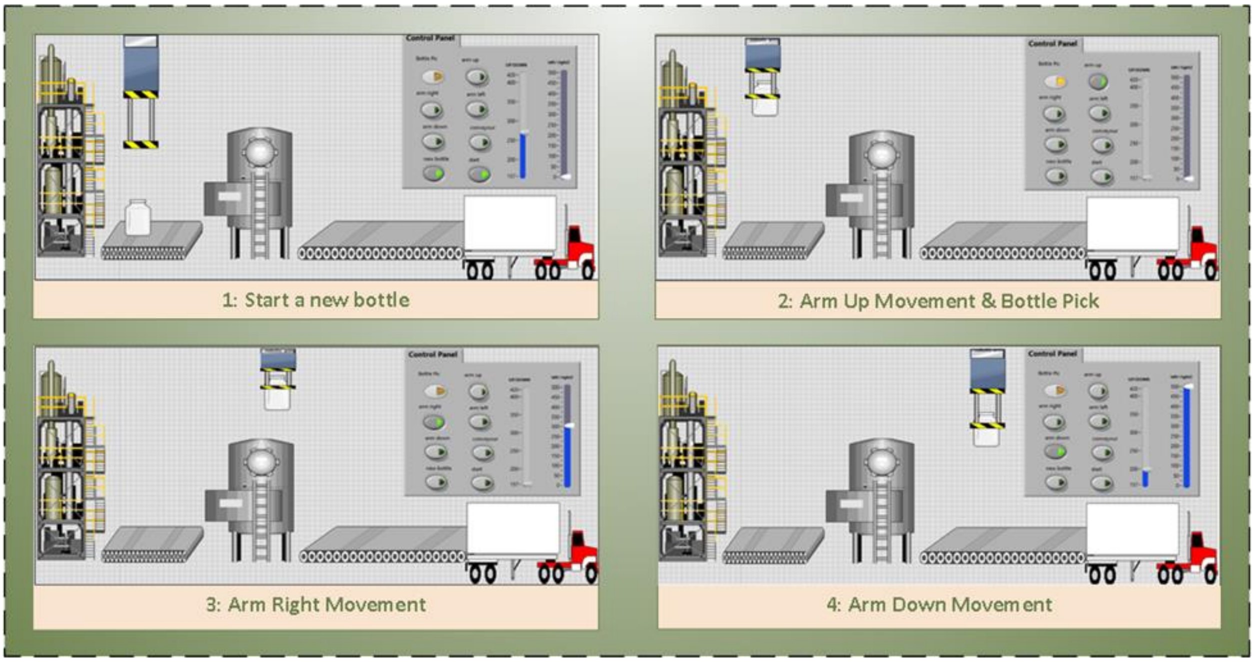

Fig. 6 SCADA Movement

Conveyor indicator and Arm Up Indicator

The upward movement of the robotic arms and the conveyor are shown in Figure 7. This shows that the process has picked up the conveyor movement signal, forcing the conveyor and the bottle it carries to advance. After the bottle is lowered into the conveyor, the arm will raise. As the bottle advances along the conveyor toward the packing truck, the arm will raise and return to its initial position. Conveyor motion and arm-up motion occur in parallel at the same time.

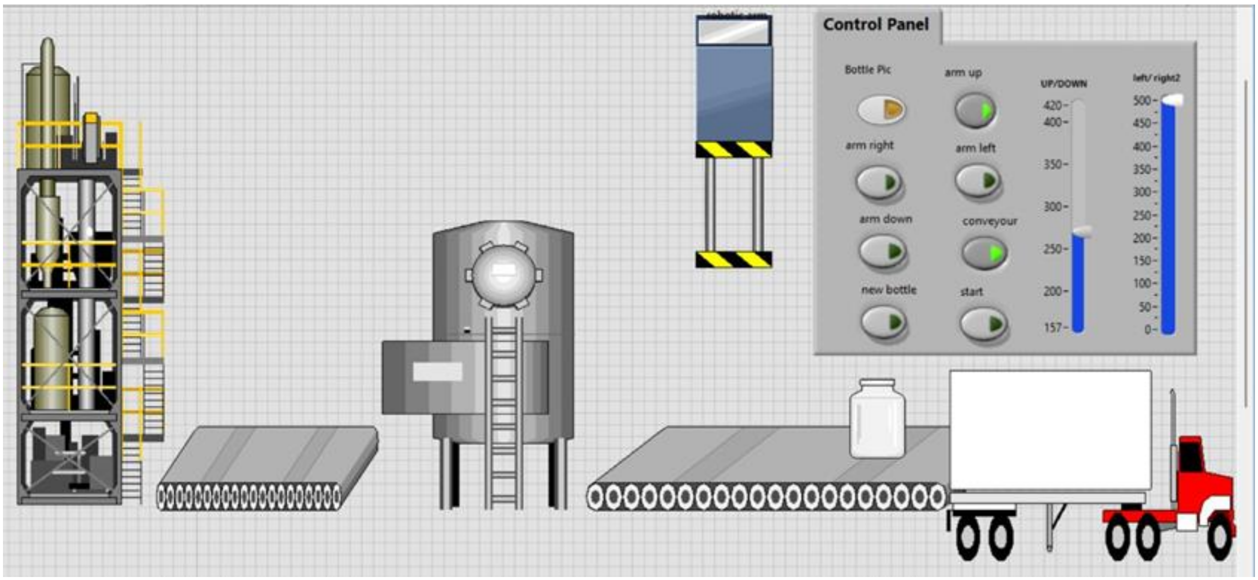

Fig. 7 Conveyor & up the movement of the arm

A few instruments will be used in the project's industrial application to get the required results from the SCADA. The bottle on the beginning conveyor will be found using the sensors on the robotic arm. So, when the bottle is genuinely present, only to the extent necessary to pick it up and place it on the packing side of the conveyor belt will it lower itself. The packaging conveyor belt will also include sensors that can recognise whether a bottle is on the belt; consequently, the conveyor will only move when a bottle is there.

This will result in very high energy efficiency. When the necessary number of bottles are in the package, the conveyor as well as the process will pause while waiting for the next box to be placed on the packing conveyor. The bottles will be tallied using a proximity counter across the conveyor belt. The job that the robotic arm is designed to do will determine how it operates. It may be driven to function as an electromagnet when picking up iron objects, or it can be used to pick objects made of glass or other materials by adding stepper motors.

CONCLUSION

The bottle-filling facility's mechanical arm-based SCADA design was adequate. The primary aim of the project, which was to develop a digital robotic arm-based SCADA that is capable of being readily integrated into the existing system, was also successfully completed. The robotic arm's mechanical setup should also be lightweight, easy to maintain, and power-efficient.

A control panel that is easily accessed was successfully built into the SCADA system based on a robotic arm. In order to make maintenance easy, the mechanical component was kept simple. Low power usage was considered while designing the automation. The whole SCADA-based bottle-filling process was successfully developed, simulated, and watched from the digital control panel.

FUTURE RECOMMENDATIONS

The future recommendations are given below:

- Currently a software-based SCADA, it could someday be used in the industry. The industry will utilise a variety of sensors and limit switches. Sensors that determine if the bottle is there at the start and while it is being packed. Furthermore, limit switches to prevent the robotic arm's extension from going beyond the required lengths on each end.
- This robotic arm design may have applications in the future. We may use the had as a rotatory arm by using servo motors to control its movement. The design of the robotic arm enables the installation of auxiliary motors for a range of applications.

Acknowledgements

We are thankful for all the authors of this paper for their team work and collaboration. We thank the relevant teachers for their support in the experiments, the students in the lab for their contributions, and Key Laboratory of Intelligent Computing & Information Processing, Ministry of Education, Xiangtan University, Xiangtan 411105, China.

REFERENCES

[1]. Akinwale, O., Kehinde, L., Ayodele, K. P., Jubril, A. M., Jonah, O. P., Ilori, S., & Chen, X. (2009, June). A Labview Based on Line Robotic Arm For Students' Laboratory. In 2009 Annual Conference & Exposition (pp. 14-39).

[2]. Sujatha, C. and Shanthakumari, G., Remote Robotic Arm Control Using LabVIEW (2015).
[3]. Salah, B., Alsamhan, A.M., Khan, S. and Ruzayqat, M., 2021. Designing and Developing a Smart Yogurt Filling Machine in the Industry 4.0 Era. Machines, 9(11), p.300.
[4]. Wang, H., Dauod, H., Khader, N., Yoon, S.W. and Srihari, K., 2018. Multi-objective parallel robotic dispensing planogram optimisation using association rule mining and evolutionary algorithms. International Journal of Computer Integrated Manufacturing, 31(8), pp.799-814.
[5]. Goyanka, M.S. and Patil, M.R., 2014. LabVIEW For ARM Based System Automation.
[6]. Padhee, S. and Nandan, D., 2021. Design of Automated Visual Inspection System for Beverage Industry Production Line. Traitement du Signal, 38(2), pp.461-466.
[7]. Kumar, A. and Puja Rani, A., 2016. Automation of beverage vending machines using PLC and Scada. An International Journal of Engineering & Technology, 3(5), pp.78-84.
[8]. Sabnis, N.V., Ingale, S.M. and Vede, S.S., 2017. Improvement in Productivity of Bottle Filling Operation by using Multi-Nozzle PLC System.
[9]. Kumbhar, V.B., Gokhale, D., Kumbhar, V.S. and Patil, M.K., 2017. Implementation of Customised SCADA for Cartoner Packaging machine for Cost Effective Solution. International Research Journal of Engineering and Technology, 4(1).
[10]. Aria, M., Utama, J., Fauzia, F., Rizal, M., Fahmi, M. and Yudha, M., 2020, July. Virtual Simulation System with Various Examples and Analysis Tools for Programmable Logic Controller Training. In IOP Conference Series: Materials Science and Engineering (Vol. 879, No. 1, p. 012108). IOP Publishing.
[11]. Venkateswarlu, G., 2014. Auto Quality Testing and Filling Station by using NI LabVIEW and Specially Designed OFS. CVR Journal of Science and Technology, 7, pp.90-96.
[12]. Pornjit Pratumsuwan and Watcharin Pongaen, "An Embedded PLC Development for Teaching in Mechatronics Education", 6th IEEE Conference on Industrial Electronics and Applications 2011 pp.1477-148.
[13]. "Getting Started with the LabVIEW Embedded Module for ARM Microcontrollers 1.1," National Instruments Corporation, 2008.
[14]. Tieyi Zhang, Yun Zhang, Xiaorong Zhou. DzDevelopment on Serial Communication between Panasonic Electrical PLC and Computer based on VBdz. Guangxi University, Nanning China. Applied Mechanics and Materials Vols, 170-173(2012)
[15]. https://en.wikipedia.org/wiki/Automation
[16]. C. Zhang and K. S. Suslick, "Colorimetric sensor array for soft drink analysis", J. Agric Food Chem, vol 55, pp 237-242, 2007.