AUTOMATIC BOTTLE FILLING SYSTEM USING PLC

A project report for Industrial Internship

In partial fulfillment for the award of the degree

of

B.TECH

In the department of

ELECTRICAL ENGINEERING

from

GREATER KOLKATA COLLEGE OF ENGINEERING AND MANAGEMENT

at

Ardent Computech Pvt. Ltd.

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent® Computech Pvt. Ltd.
Module – 132, Ground Floor, SDF Building,
Sector V, Salt Lake, Kolkata – 700091.
☎+91 33 40073507
✉ training@ardentcollaborations.com

CERTIFICATE FROM SUPERVISOR

This is to certify that student have successfully completed the project titled "**AUTOMATIC BOTTLE FILLING SYSTEM USING PLC”** under my supervision during the period from “**08/08/22**” to “**17/08/22**” which is in partial fulfillment of requirements for the award of the **B.TECH** degree and submitted to the Department of “**EE**” of **“GREATER KOLKATA COLLEGE OF ENGINEERING AND MANAGEMENT”**.

Name	signature
1) RAFAYAL NATHANIAL	Rafayal Nathanial
2) SUBHAJIT MONDAL	Subhajit Mondal.
3) DEBPRASAD NASKAR	Debprasad Naskar
4) MOUMITA SANPUI	Moumita Sanpui
5) DHIMAN DEY	Dhiman Dey
6) NILOY NASKAR	Niloy Naskar
7) KUNTAL NASKAR	Naske
8) ROUNAK MONDAL	Rounak Mondal.

Soumyadip Sarkar

____
Signature of the Supervisor

ARDENT COMPUTECH PVT. LTD. ★ KOLKATA

Date : ***17/08/2022***

Name of the Project Supervisor : **SOUMYADIP SARKAR**

ACKNOWLEDGEMENT

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, *[SOUMYADIP SARKAR]* for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

ABSTRACT

The objective of our project is to design, develop and monitor "Automatic bottle filling system using PLC". This work provides with a lot of benefits like low power consumption, low operational cost, less maintenance, accuracy and many more. This project is based on Industrial automation and is a vast application used in many industries like milk industries, chemical, food, mineral water and many industrial manufacturers. A prototype has been developed to illustrate the project.

Filling is the task that is carried out by a machine and this process is widely used in many industries. In this project, the filling of the bottle is controlled by using a controller known as PLC which is also the heart of the entire system. For the conveyor system, a dc motor has been selected for better performance and ease of operation. A sensor has been used to detect the position of the bottle. In our project we have used less number of systems hence the overall cost has been reduced to an extent. Ladder logic has been used for the programming of the PLC, which is the most widely used and accepted language for the programming of the PLC. The PLC used in this system is a Siemens S7 – 1200 which makes the system more flexible and easy to operate

INDEX

INTRODUCTION

Overview

The project is based on industrial automation and PLC is the heart of automation. The hardware and the software are the two important areas in our project.

1) HARDWARE DESCRIPTION:

In this project, Siemens PLC SIMATICS S7 1200 is used for controlling the inputs and outputs. Input supply to the PLC is given through a SMPS. The rating of the SMPS is 24V DC 5Amps. The PLC used here is a compact PLC which has fixed number of inputs and outputs. In this kind of PLC model, the CPU contains 14 digital inputs and 10 digital outputs. One diffused photoelectric sensor has been used for the positioning of the bottles. A geared DC motor has been used for running the conveyor system. The ratings of the DC motor is 12V and 50 RPM speed with a high starting torque of 70 Kg-cm (at no load). Toggle switches are used to serve the purpose of some inputs to the PLC.

2) SOFTWARE DESCRIPTION:

There are five important languages which are used for the programming of the PLC. The list of the methods is as follows:

- Functional block diagram (FBD)
- Structure text
- Instruction list
- Flow chart
- Ladder diagram

Out of these five languages, ladder diagram is the most widely used language and is simple as compared to other languages. Ladder diagram has been used for the programming of this PLC.

AUTOMATION

- Automation is a set of technology that results in operation of machine and systems without any significant human intervention and achieves performance superior to manual operation.
- Automation is the creation of technology and its application in order to control and monitor the production and delivery of various goods and services. It performs tasks that were previously performed by humans. Automation is being used in number of areas such as manufacturing, transport, utilities, defense, facilities, operations and lately, information technology.
- Automation or industrial automation is the use of control systems such as computers, controllers to control industrial machinery and processes, to optimize productivity in the production of goods and delivery of services. Automation is a step beyond mechanization. Whereas mechanization provides human operators with machinery to assist them with the muscular requirements of work, automation greatly decreases the need for human sensory and mental requirements.

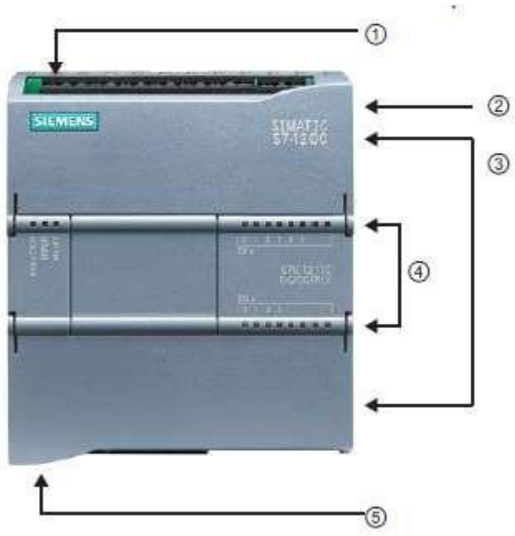
Automation impacts

- It increases productivity and reduce cost. It gives emphasis on flexibility and convertibility of manufacturing process. Hence gives manufacturers the ability to easily switch from manufacturing Product A to manufacturing product B without completely rebuilt the existing system/product lines. Automation is now often applied primarily to increase quality in the manufacturing process, where automation can increase quality substantially.

- Increased consistency of output. Replacing humans in tasks done in dangerous environments.

PLC (PROGRAMABLE LOGIC CONTROLLER)

Technical Specification of PLC



① Power connector

② Memory card slot under top door

③ Removable user wiring connectors (behind the doors)

④ Status LEDs for the on-board I/O

⑤ PROFINET connector (on the bottom of the CPU)

The CPU combines a microprocessor, an integrated power supply, input and output circuits, built-in PROFINET, high speed motion control I/O and on board analog inputs in a compact housing to create a powerful controller. After downloading a program, the CPU contains the logic required to monitor and control the devices in the application. The CPU monitors the imputs and changes the output according to the logic of the user's program.

The CPU provides a PROFINET port for communication over a PROFINET network.

Power Budget

The CPU has an internal power supply that provides power for the CPU, the signal modules, signal board and communication modules and for other 24VDC user power requirements.

5VDC logic budget supplied by the CPU and the 5VDC power requirements of the signal modules, signal boards and communication modules.

The CPU provides a 24 VDC sensor supply that can supply 24 VDC for input points, for relay coil power on the signal modules, or for other requirements. If your 24 VDC power requirements exceed the budget of the sensor supply, then you must add an external 24 VDC power supply to your system.

Some of the 24 VDC power input ports in the S7-1200 system are interconnected, with a common logic circuit connecting multiple M terminals. For example, the following circuits are interconnected when designated as "not isolated" in the data sheets: the 24 VDC power supply of the CPU, the power input for the relay coil of an SM, or the power supply for a nonisolated analog input. All non-isolated M terminals must connect to the same external reference potential.

General Specification of CPU 1214C

CPU features:-

Technical Data	Description
User memory – □ Work	□ 75 Kbytes
□ Load	□ 4 Mbytes internal, expandable upto SD card size
□ Retentive	□ 10 Kbytes
On – board digital I/O	14 inputs/ 10 outputs
On – board analog I/O	2 inputs
Process image size	1024 bytes of inputs (I)/ 1024 bytes of outputs (Q)
Bit Memory (M)	8192 bytes
Temporary (local) memory	• 16 kbytes for startup and program cycles (including associated FBs and FCs) • 4 kbytes for standard interrupt events including FBs and FCs • 4 kbytes for error interrupt events including FBs and FCs
Signal modules expansion	8 SMs max.

SB, CB, BB expansion	1 max
Communication module expansion	3 CMs max
High – speed counters	6 total • Single phase: 3 at 100kHz and 3 at 30kHz clock rate • Quadrature phase: 3 at 80kHz and 3 at 20kHz clock rate
Pulse outputs	4
Pulse catch inputs	14
Time delay/cyclic interrupts	4 total with 1ms resolution
Edge interrupts	12 rising and 12 falling (14 and 14 with optional signal board)
Memory card	SIMATIC Memory card (optional)
Real time clock accuracy	+/- 60 seconds/month
Real time clock retention time	20 days typ./12 days min. at 40^{o}C (maintenance free super capacitor)

Power Supply:-

Technical Data	
Voltage range	85 to 264 VAC
Line frequency	47 to 63 Hz
Input current (max. load)	• CPU only – 100 mA at 120VAC; 50mA at 240 VAC • CPU with all expansion accessories – 300mA at 120 VAC; 150mA at 240 VAC
Ground leakage, AC line to functional earth	0.5 mA max.
Hold up time (loss of power)	20ms at 120 VAC; 80ms at 240 VAC

Digital Inputs and Outputs:- □

Number of inputs: 14

- Rated voltage: 24 VDC at 4 mA, nominal
- Surge voltage: 35 VDC for 0.5 sec

- Logic 1 signal (min.): 15 VDC at 2.5 mA
- Logic 0 signal (max.): 5 VDC at 1 mA
- Number of outputs: 10
- Current (max.): 2A
- Surge current: 7A with contacts closed
- Maximum relay switching frequency: 1 Hz

PLC Concepts

Execution of the user program

The CPU supports the following types of code blocks that allow us to create an efficient structure for our user program:

- Organization blocks (OBs) define the structure of the program. Some OBs have predefined behavior and start events, but we can also create OBs with custom start events.
- Functions (FCs) and function blocks (FBs) contain the program code that corresponds to specific tasks or combinations of parameters. Each FC or FB provides a set of input and output parameters for sharing data with the calling block. An FB also uses an associated data block (called an instance DB) to maintain state of values between execution that can be used by other blocks in the program. Valid FC and FB numbers range from 1 to 65535.
- Data blocks (DBs) store data that can be used by the program blocks. Valid DB numbers range from 1 to 65535.

Execution of the user program begins with one or more optional start-up organization blocks (OBs) which are executed once upon entering RUN mode, followed by one or more program cycle OBs which are executed cyclically.

Operating Modes of the CPU

The CPU has three modes of operation: STOP mode, STARTUP mode, and RUN mode. Status LEDs on the front of the CPU indicate the current mode of operation.

- In STOP mode, the CPU is not executing the program. We can download a project.
- In STARTUP mode, the startup OBs (if present) are executed once. Interrupt events are not processed during the startup mode.
- In RUN mode, the program cycle OBs are executed repeatedly. Interrupt events can occur and be processed at any point within the RUN mode.

PLC data type

The PLC data type editor lets us define data structures that we can use multiple times in our program. We create a PLC data type by opening the “PLC data types” branch of the project tree and double – clicking the "Add new data type" item. On the newly created PLC data type item, we use two single-clicks to rename the default name and double-click to open the PLC data type editor.

We create a custom PLC data type structure using the same editing methods that are used in the data block editor. We then add new rows for any data types that are necessary to create the data structure that we want.

If a new PLC data type is created, then the new PLC type name will appear in the data type selector drop drop-lists in the DB editor and code block interface editor.

Potential uses of PLC data types:

- PLC data types can be used directly as a data type in a code block interface or in data blocks.
- PLC data types can be used as a template for the creation of multiple global data blocks that use the same data structure.

Programming Concepts

Structuring the user program

When we create a user program for the automation tasks, we insert the instructions for the program into code blocks:

- An organization block (OB) responds to a specific event in the CPU and can interrupt the execution of the user program. The default for the cyclic execution of the user program (OB 1) provides the base structure for your user program and is the only code block required for a user program. If we include other OBs in our program, these OBs interrupt the execution of OB 1. The other OBs perform specific functions, such as for startup tasks, for handling interrupts and errors, or for executing specific program code at specific time intervals.
- A function block (FB) is a subroutine that is executed when called from another code block (OB, FB, or FC). The calling block passes parameters to the FB and also identifies a specific data block (DB) that stores the data for the specific call or instance of that FB. Changing the instance DB allows a generic FB to control the operation of a set of devices. For example, one FB can control several pumps or valves, with different instance DBs containing the specific operational parameters for each pump or valve.
- A function (FC) is a subroutine that is executed when called from another code block (OB, FB, or FC). The FC does not have an associated instance DB. The calling block passes parameters to the FC. The output values from the FC must be written to a memory address or to a global DB.

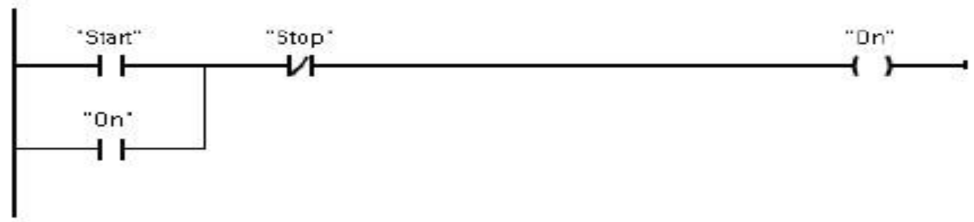
Programming Language

STEP 7 provides the following standard programming languages for S7-1200:

- LAD (ladder logic) is a graphical programming language. The representation is based on circuit diagrams.
- FBD (Function Block Diagram) is a programming language that is based on the graphical logic symbols used in Boolean algebra.
- SCL (structured control language) is a text-based, high-level programming language.

Ladder Logic (LAD)

The elements of a circuit diagram, such as normally closed and normally open contacts, and coils are linked to form networks.



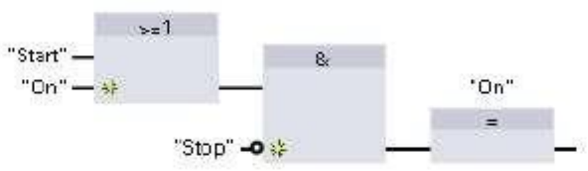
To create the logic for complex operations, you can insert branches to create the logic for parallel circuits. Parallel branches are opened downwards or are connected directly to the power rail. You terminate the branches upwards.

LAD provides "box" instructions for a variety of functions, such as math, timer, counter, and move.

STEP 7 does not limit the number of instructions (rows and columns) in a LAD network.

Function Block Diagram

Like LAD, FBD is also a graphical programming language. The representation of the logic is based on the graphical logic symbols used in Boolean algebra.



To create the logic for complex operations, insert parallel branches between the boxes.

Mathematical functions and other complex functions can be represented directly in conjunction with the logic boxes.

STEP 7 does not limit the number of instructions (rows and columns) in an FBD network.

SCL

Structured Control Language (SCL) is a high-level, PASCAL-based programming language for the SIMATIC S7 CPUs. SCL supports the block structure of STEP 7.

SCL instructions use standard programming operators, such as for assignment (:=), mathematical
functions (+ for addition, - for subtraction, * for multiplication, and / for division). SCL also uses standard PASCAL program control operations, such as IF-THEN-ELSE, CASE, REPEATUNTIL, GOTO and RETURN. We can use any PASCAL reference for syntactical elements of the SCL programming language. Many of the other instructions for SCL, such as timers and counters, match the LAD and FBD instructions.

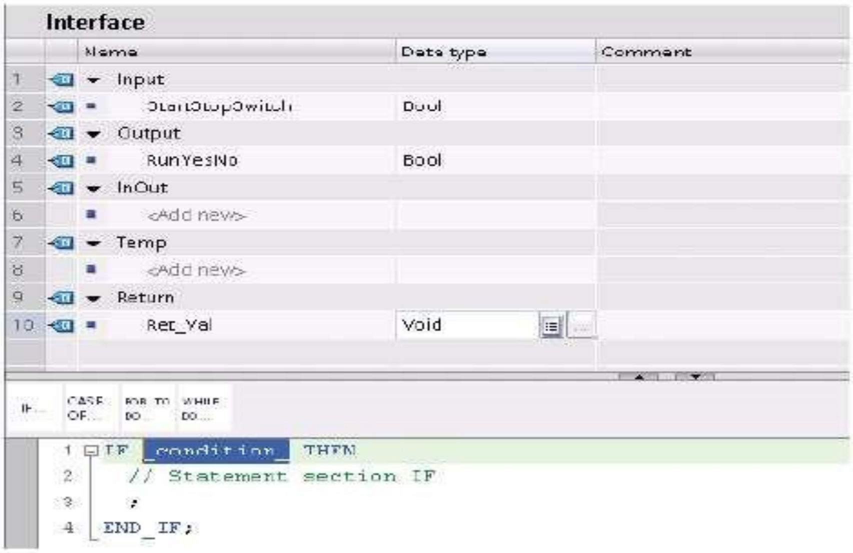
We can designate any type of block (OB, FB, or FC) to use the SCL programming language at the time we create the block. STEP 7 provides an SCL program editor that includes the following elements:

● Interface section for defining the parameters of the code block

● Code section for the program code

● Instruction tree that contains the SCL instructions supported by the CPU

We enter the SCL code for our instruction directly in the code section. For more complex instructions, we simply drag the SCL instructions from the instruction tree and drop them into our program. We can also use any text editor to create an SCL program and then import that file into STEP 7.

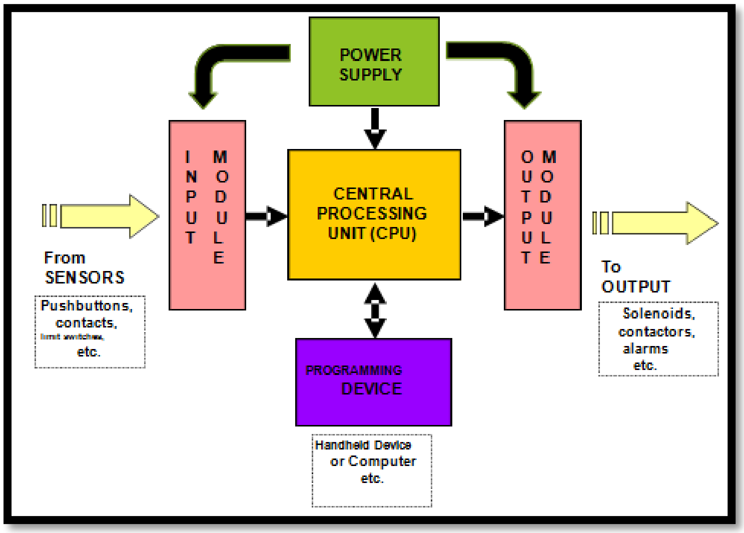


Interface

	Name	Data type	Comment
1	Input		
2	StartStopSwitch	Bool	
3	Output		
4	RunYesNo	Bool	
5	InOut		
6	<Add new>		
7	Temp		
8	<Add new>		
9	Return		
10	Ret_Val	Void	

```
IF _condition_ THEN
    // Statement section IF
    ;
END_IF;
```

PLC Architecture


POWER SUPPLY
INPUT MODULE
CENTRAL PROCESSING UNIT (CPU)
OUTPUT MODULE
From SENSORS
Pushbuttons, contacts, limit switches, etc.
To OUTPUT
Solenoids, contactors, alarms etc.
PROGRAMMING DEVICE
Handheld Device or Computer etc.

PLC System

- Input / Output Section
- CPU or Central Processing
- Programming Device
- Power supply
- Memory

Basic Elements

- **Input/ Output Section:** The input section or input module consists of devices like sensors, switches, and many other real-world input sources. The input from the sources is connected to the PLC through the input connector rails. The output section or output module can be a motor or a solenoid or a lamp or a heater, whose functioning is controlled by varying the input signals.
- **CPU or Central Processing Unit:** It is the brain of the PLC. It can be a hexagonal or an octal microprocessor. It carries out all the processing related to the input signals in order to control the output signals based on the control program.
- **Programming Device:** It is the platform where the program or the control logic is written. It can be a laptop or a computer itself.
- **Power Supply:** It generally works on a power supply of about 24 V, used to power input and output devices.
- **Memory:** The memory is divided into two parts- The data memory and the program memory. The program information or the control logic is stored in the user memory or the program memory from where the CPU fetches the program instructions. The input and output signals and the timer and counter signals are stored in the input and output external image memory respectively.

PLC hardware

On the automation point of view switches are classified on following manner.

- *Manual Actuation*
  Ex. Toggle switch & Push button.
- *Mechanical Actuation*
  Ex. Limit Switches
- *Electrical Actuation*
  Ex. Relay, Timer, Sensor, Contactor

Input/output Hardware

Contactor & Relays

Contractor :- A contactor is an electrically controlled switch used for switching an electrical power circuit.

A contactor has three components:-

a. The contacts are the current carrying part of the contactor. This includes power contacts, auxiliary contacts, and contact springs.
b. The electromagnet (or "coil") provides the driving force to close the contacts.
c. The enclosure is a frame housing the contacts and the electromagnet. Enclosures are made of insulating materials such as Bakelite, Nylon 6, and thermosetting plastics to protect and insulate the contacts and to provide some measure of protection against personnel touching the contacts.

Relay : - Relay is an electrical switch, which is turn ON /OFF under another switch.

Relay is two types –

I) Controlling type: - Controlling Relay takes lower amount of current & voltage and control higher amount of current & voltage.
Controlling Relay is two types –
a. EMR (Electromagnetic Relay).
b. SSR (Solid State Relay).
II) Protection type: - Protection Relay detects the isolated faults in the generation, transmission & distribution system. Protection Relay is two types – a. OLR (Overload Relay).
b. Microprocessor.

3.2 Toggle Switch: - A Toggle Switch is an electromechanical switch which uses a lever or baton as an actuator. Toggle switches are available in many sizes and configurations and offer a wide range of uses. They are popular for their ease of operation and generally offer 1-3 positions to open or close a circuit.

Different types of toggle switch-

- SPST single pole single throw
- SPDT single pole double throw
- DPST double pole single throw
- DPDT double pole double throw
- SPCO single pole change over
- QPDT quarter pole double throw

Limit Switch: - A limit switch is an electromechanical device that consists of an actuator, mechanically linked to a set of contacts. When an object comes into contact with the actuator, the device operates the contacts to make or break an electrical connection.

Benefits:-

i.Can be used in any industrial environment.
ii.Can switch load with high inductance.

Limitation:-

i.Restricted to equipment operating at lower speed.

ii.Must make direct contact with target.   iii. Moving mechanical parts will wear out.

Sensor:-Sensor is a part of controlling circuit not a switch . Sensor is two types

_

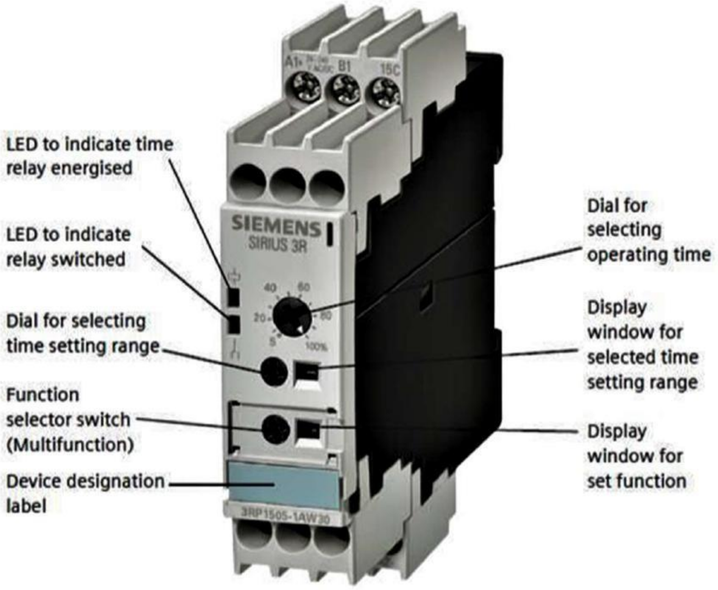
a. Analog (Thermocouple, Thermistor, RTD).

b. Digital (Inductive, Capacitive, Optical).

3.3 Timer:- A Timer is a device that provides time delay between two events or process.

Timer is two types -

I.**Mechanical Timer: -** It is a mechanical switching device. It's max time range 0.1sec – 30sec. It has no coil part only having contact part. Contact part changes its state by mechanical force.


LED to indicate time relay energised
LED to indicate relay switched
Dial for selecting time setting range
Function selector switch (Multifunction)
Device designation label
Dial for selecting operating time
Display window for selected time setting range
Display window for set function
SIEMENS
SIRIUS 3R
A1
B1
15C
3RP1505-1AW30

II.**Electrical Timer:-**It is an electrical switching device. It's timer max range 0.1sec – 30min. It has one relay coil, two contact parts, and two NO & two NC part. Contact part changes its state by the help of electricity.

<u>Sensor:-</u>

A **sensor** is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena.

Category of Sensor: -

1.Analog category
2.Digital category
1.Analog category-
A. Temperature
B. Humidity
C. Heart-Beat
D. Rain, Fog etc.
2.Digital category-
A. Inductive
B. Capacitive
C.IR(Infra-Red)
D. Optical

Advantages of PLC

1. **Reduced space:** PLCs are completely solid-state devices and hence are extremely compact in comparison to hard-wired controller where electro-mechanical devices are used.
2. **Higher Life and Reliability:** These devices are extremely rugged. The chances of defect/damage are very less as there is very lesser moving mechanisms here.
1. **Economical:** As the defect probability is very less, one can consider it as one-time investment. In this way, PLCs are undoubtedly most economical systems. Cost of PLC recovers within a short period.
1. **Energy saving:** Average power consumption is just 1/10th of power consumed by an equivalent relay-based control.

5.**Ease of maintenance:**

- Modular replacement
- Easy trouble shooting
- Error diagnostics with programming unit.

1. **Tremendous flexibility:** There is no requirement of rewiring if any change is required to be implemented. It can carry out complex functions like arithmetic operations, counting, comparing, generation of time delay etc. It has a very high processing speed and greater flexibility in both analog and digital process. “On Line”/ ‘Off Line’ programming is also possible in it.
1. **Shorter Project Time:** The hard-wired control system can be constructed only when the task is fully defined. However, in case of PLC, the construction of the controller and wiring are independent of control program definition.
1. **Easier Storage Archiving and Documentation:** This is due to its compatibility with PC-AT, Printer and Floppy disk.

Disadvantage of PLC

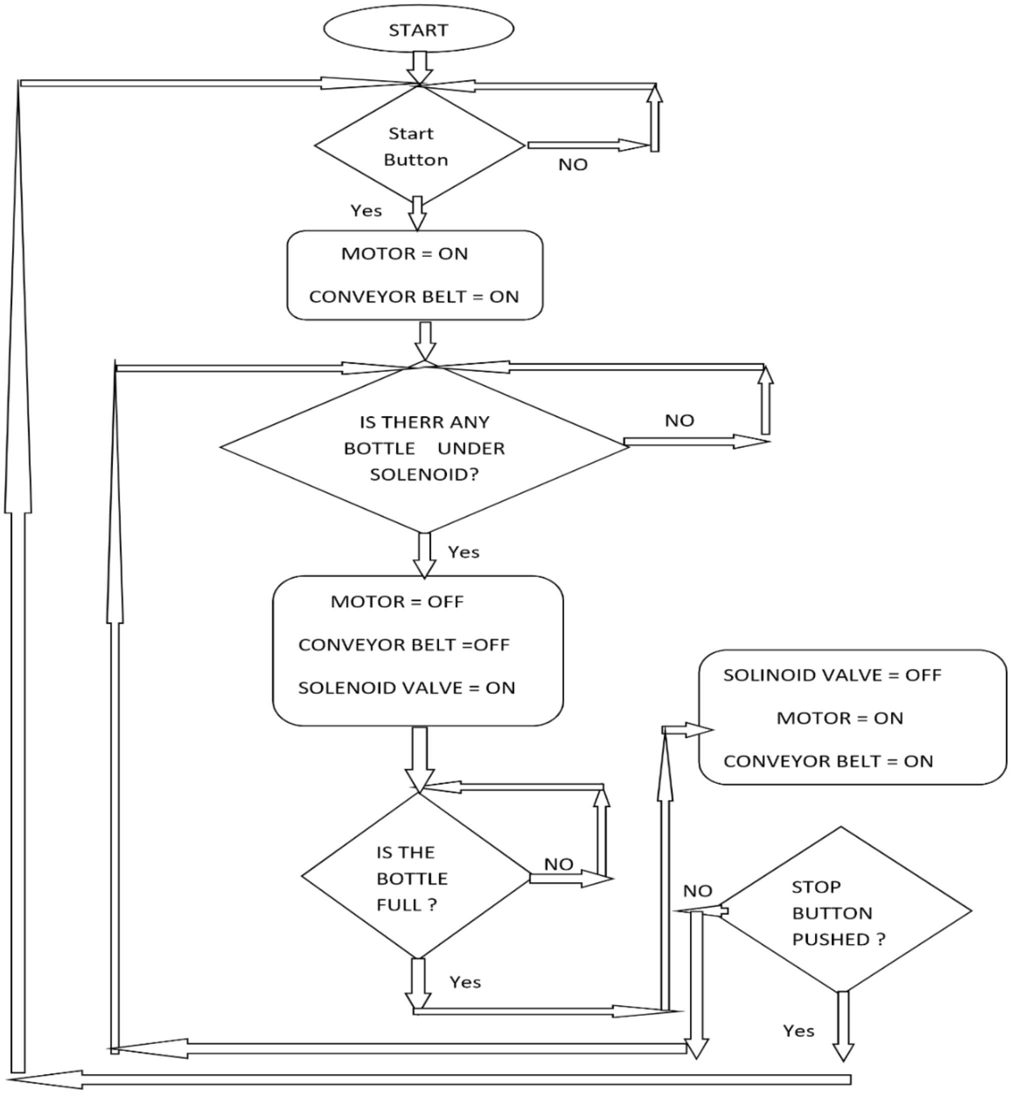
1. There's too much work required in connecting wires.
2. There's difficulty with changes or replacements.
3. It's always difficult to find errors; And require skilful work force.
4. When a problem occurs, hold-up time is indefinite, usually long.

SOFTWARE IMPLEMENTATION

Objective

we are controlling watering of any agricultural field automatically such that when we using the water of reservoir tank we can control the time and pressure of water and when tank get low level in water watering of field get stop automatically and tank pump gets on when pump gets to full position watering of field get start automatically again.

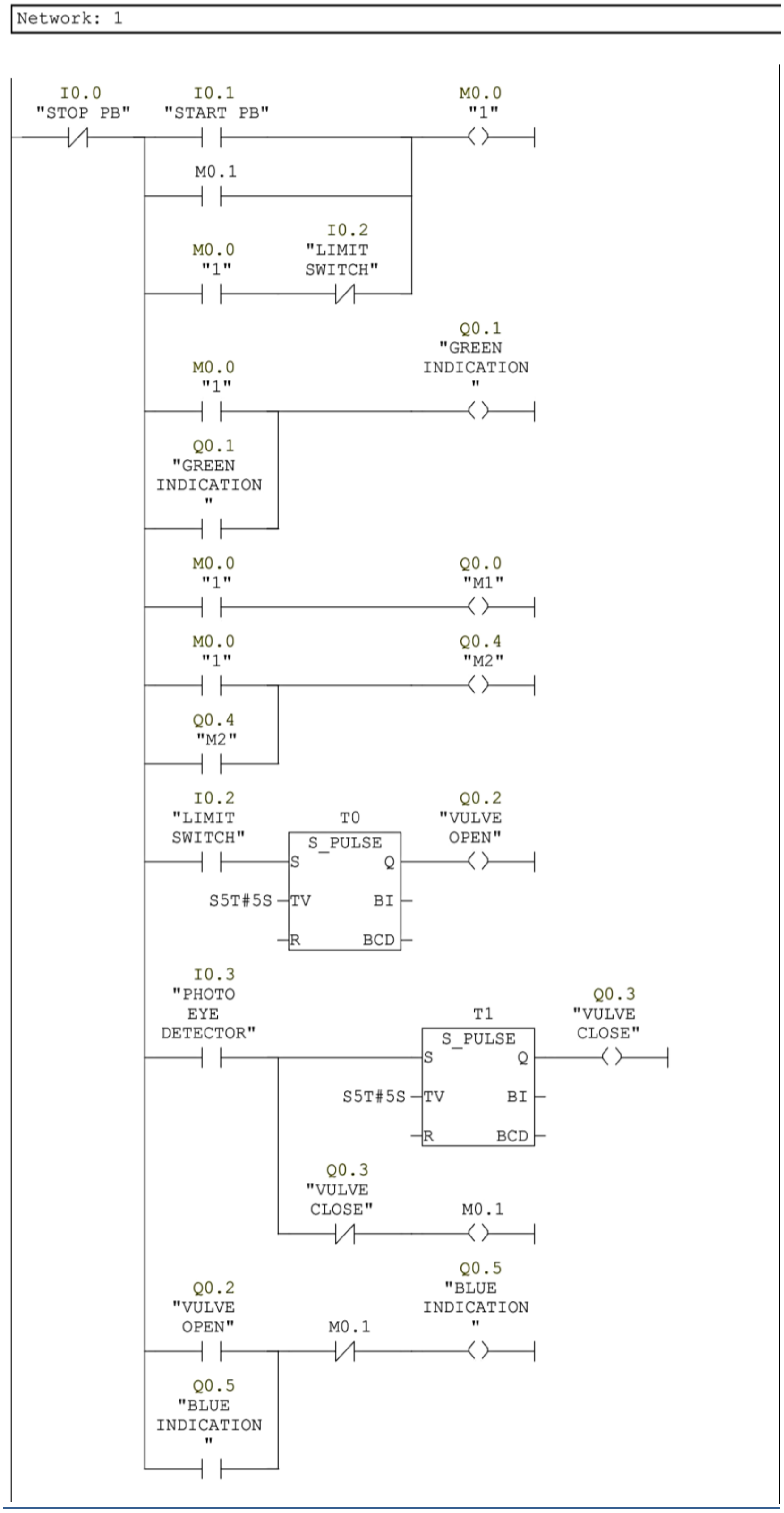
Flowchart



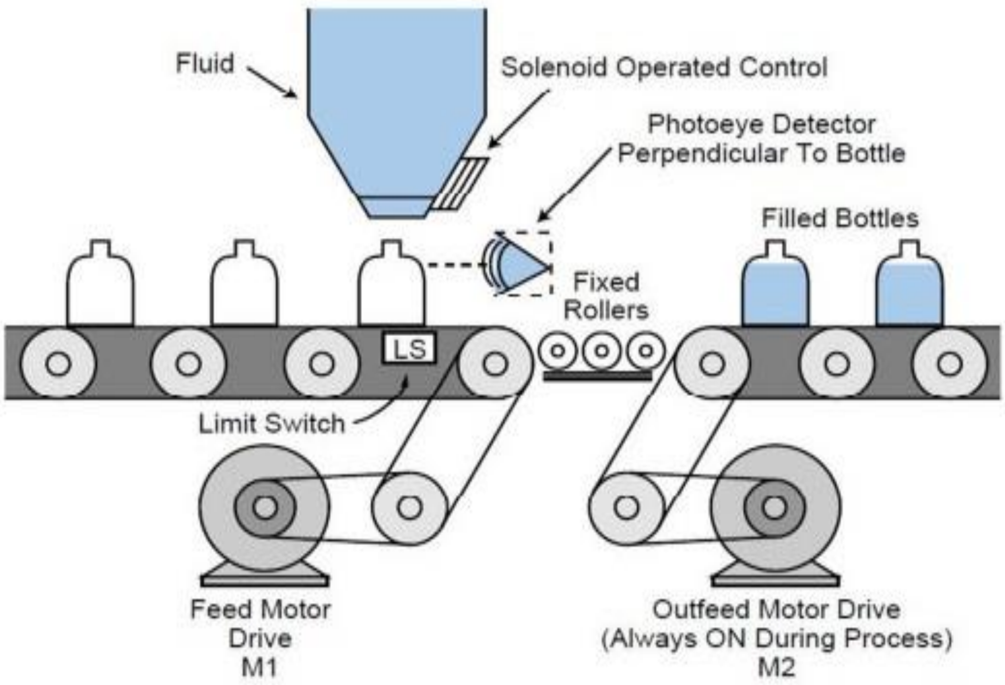
Algorithm

STEP 1	:	Start button is pressed & green light turns on and remains on and remains on until stop button is pressed.
STEP 2	:	As light turns on the motor 1 starts running.
STEP 3	:	After motor 1 runs and if their limit switch has not signaled timer T0 gets activated.
STEP 4	:	After T0 gives done and photo detector is disabled, solenoid valve gets in operation.
STEP 5	:	As PE signals solenoid stops and buzzer sounds after which timer T1 gets enabled which stops the process for 0.5sec.
STEP 6	:	Once the filled bottle condition is activated the cycle starts again.

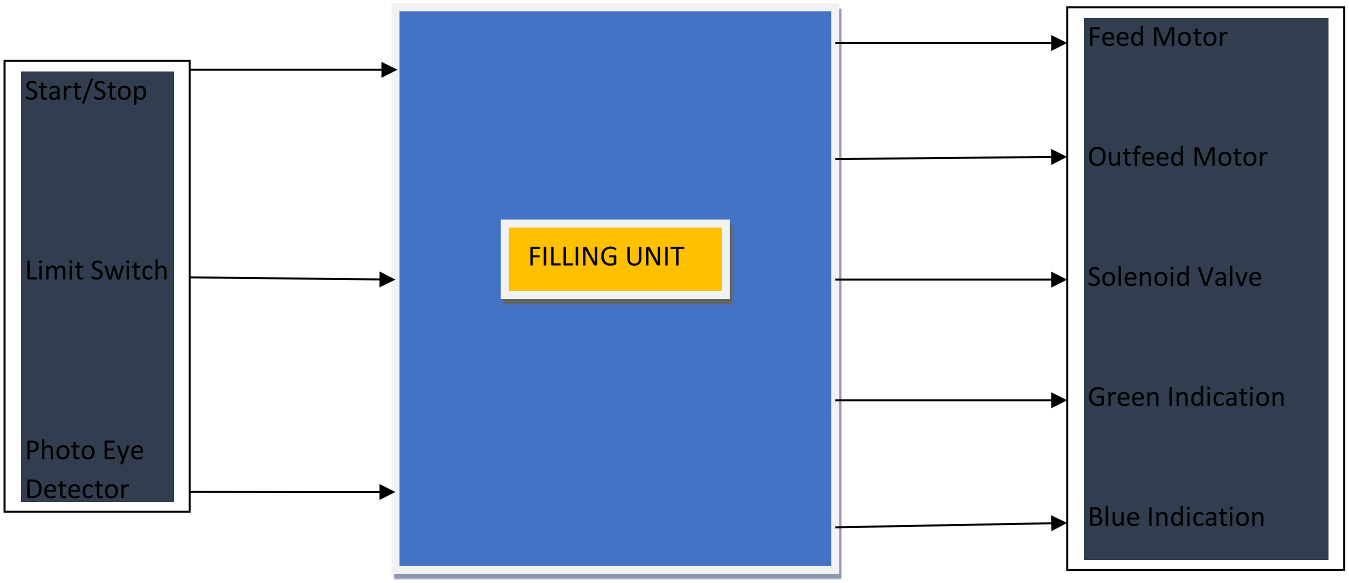
Ladder Logic Diagram (LAD)



Schematic Diagram



Block Diagram



Methodology

1. At first the NC (stop push button) and NO (start push button) connection has been placed as input to enable and disable the output (memory build).
2. Secondly we are holding that memory build with the NO (memory build) and NC (limit switch) connection as we are using push button here.
3. Next NO (memory build) has been placed to indicate a Green signal as an output. Again we are holding the Green Indication by NO (green indication) connection.
4. Next we are connecting the main output Motor1 with NO (memory build) connection.
5. Now the same connection we are doing with the Motor2 and holding the output (Motor2).
6. In the next step we are controlling the valve (open condition) with the limit switch (NO connection) and we set the timer (S_PULSE) for 0.5 sec.
7. After that we are connecting the valve (close condition) with the input photo eye detector with the second timer (S_PULSE) for again 0.5 sec.
8. After closing the valve we have to start the motor again so we are taking another memory build as output with the NC connection of close valve. Also we are holding the memory build 1 with the second memory build.
9. Now as it is said that the blue indication remain on during the time open condition of the valve we are taking the blue indication as output with NO(valve open) and NC(memory build 2) with holding the blue indication.

CONCLUSION

The main objective of this project was to develop a bottle filling system based on certain specifications. The project presents an automatic filling system controlled by PLC as per the filling requirement which has simple operation. The system has the advantages as simple structure and reliable operation. The system is controlled by PLC. This was successfully implemented. We consider this project as a journey where we acquired knowledge and also gained some insights into the subject which we have shared in this report.

By the installation of jet nozzle and strong solenoid valve can reduce the time to fill bottles and can efficiently increase productivity. A guide way could be used in case of vibration.

A capping section could also be introduced. The nozzle positioning must be given more care and concentration. The system could be redesigned for increased bottle size and productivity.

ARDENT COMPUTECH PVT. LTD. ★ KOLKATA