CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF AUTOMATION

In order to keep up with the rapidly growing technologies, industries are shifting towards automation and IoT. Before the evolution of industrial automation, industries had mechanization system where the systems were manually operated by humans. Since the need for highly accurate, high quality and increased productivity became important, automation of the process with minimum human intervention came out as a better option.

Industrial automation is about the usage of control devices such as PLCs to control the various processes and the machinery by minimizing human intervention as much as possible. In industrial automation, various process variables such as pressure, temperature, flow, level of liquid are acquired with the help of sensors. Once this data is acquired, it is processed by systems like microcontrollers or PLCs.

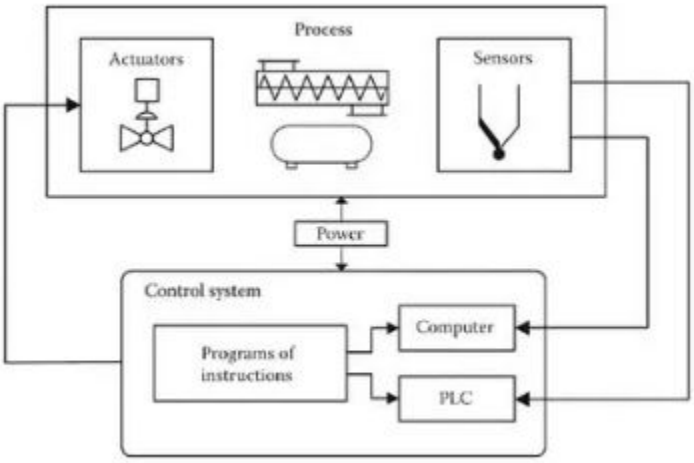


Fig 1.1.: Industrial automation

Various process variables are acquired with the help of sensors which are present at the process station. This acquired data is them provided to the systems like PLCs, which contain the program instructions according to which the process has to run. After processing the variables, the output is provided to the actuators which in turn acts on the process station to provide the desired result (shown in Fig 1.1).

Bottle filling is a process where the bottle is filled with liquids. Automatic bottle filling system play an important part in pharmaceutical, agricultural, chemical, food and various other industries where bottles are to be filled in bulk.

1.2 AIM & OBJECTIVES OF THE PROJECT WORK

The main objectives of the project are:

- Develop an automatic liquid filling system using PLC
- Control the START/STOP of the motor via the phone application (Blynk)
- Control the movement of the conveyor belt by the PLC based on the position of the bottle
- Sensing the position of the bottle using the proximity sensor and stopping the movement of the belt once it is in the right position
- Control the action and timing of the solenoid valve using the PLC that is used to fill the liquid into the bottle.
- Collecting the data from the Blynk application via Wifi using the NodeMCU module

1.3 PROBLEM STATEMENT AND METHODOLOGY

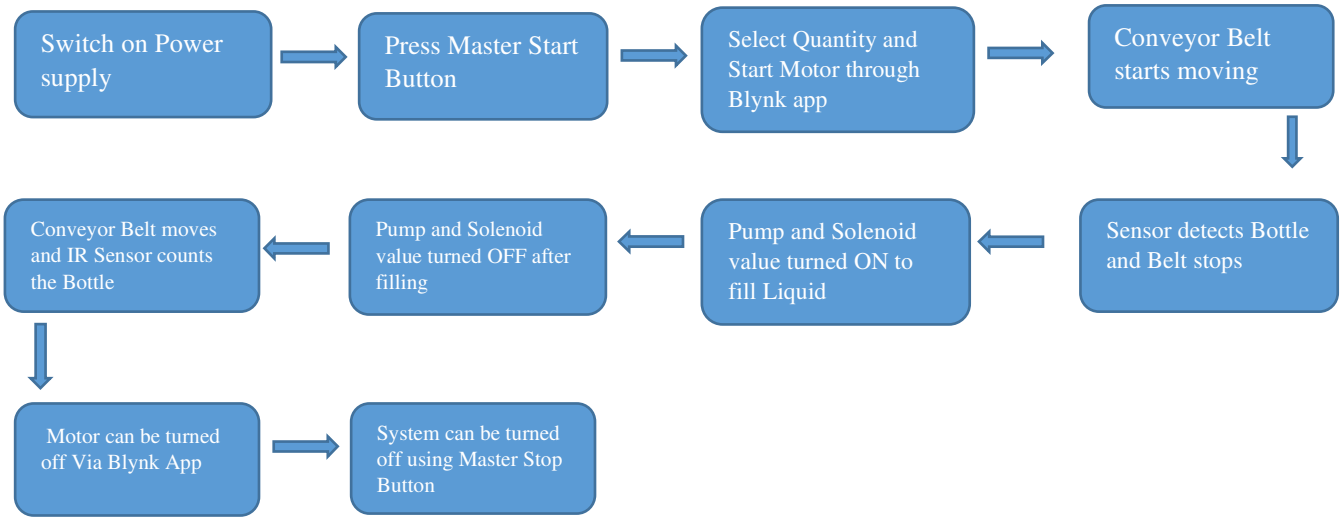
In Large industries PLCs are used in automatic liquid filling systems which either HMI or RFID sensors. As an alternative we have used IoT in our project which allows the user to change the quantity of liquid to be filled and control ON/OFF of the motor through an application on their phone.

Working Methodology

In our project, we control the quantity of liquid that is to be filled in the bottle through the BLYNK app. The quantity is selected and the motor to run the conveyor belt is also controlled using the app. Once both the parameters are selected, the system keeps running for the same quantity until the motor is manually stopped through the app and the quantity is changed.

Timers are included which are used to fill the liquid in the bottles and respective delays are provided once the bottles are filled.

Once the MASTER START button is pressed, the system starts. Once the quantity of liquid that is to be filled is selected via the app and the motor is started through the app, the conveyor belt starts running with the empty bottles on it. The motor stops running automatically once the proximity sensor senses the bottle in the right position, the solenoid valve turns ON and the liquid is filled into the bottle according to the quantity selected and the timing provided in the ladder logic. Once the bottle is filled, the motor starts moving after the delay, in turn which moves the conveyor belt. As the bottle moves, an IR sensor is placed which detects the bottles and displays the count of the bottles that are filled.



1.4 ORGANISATION OF REPORT

This project report is divided into 6 Chapters. After this introduction chapter, **Chapter 2** provides information regarding the literature survey and the advantages of using automatic liquid filling systems.

Chapter 3 contains the general block diagram of the project, the hardware components such as DC geared motor, solenoid valve, NodeMCU, its pin configurations and other components required for the project.

Chapter 4 contains the details of FX1N-20MR PLC and the details of the software GX-Works2 used to develop the ladder logic.

Chapter 5 contains the connection diagrams, ladder logic and the software details required for the project

Chapter 6 provides the result and conclusion of the project.

CHAPTER 2
LITERATURE SURVEY

"PLC Based Automatic Bottle Filling and Capping System with User Defined Volume Selection [1].address about detection of bottle position using IR sensor. The filling operation is accompanied with a user defined volume selection menu. Capping of bottles.

“Automatic liquid filling to bottles of different height using programmable logic controller”[2].focuses on development of a filling machine which can fill bottles of different heights using PLC. Here capacitive sensors are used for sensing the bottles.

The advantages of using automatic filling systems are as follows

- It enables faster processes and saves a lot of time
- Since it is an automatic system, it is more consistent and reliable. They provide accurate results whether it be volume of the liquid filled, level of filling etc.
- Product wastage due to errors while filling are reduced as a result of their accuracy
- Operating such systems are easy
- These systems can be easily upgraded to meet the necessary requirements

CHAPTER 3

SYSTEM DESCRIPTION

3.1 BLOCK DIAGRAM

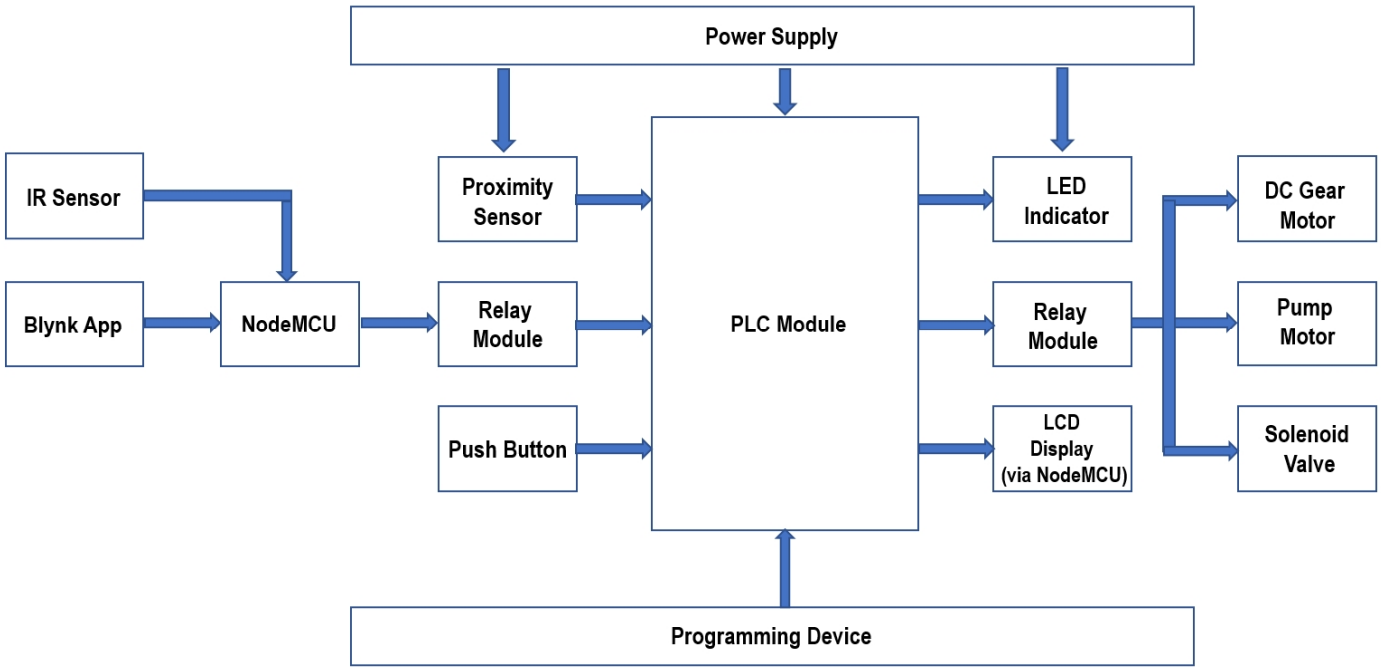


Fig 3.1: Block diagram

The main aim of our project is to use PLC which controls the various process in the system. It takes the signals from the sensors, processes it and sends signal to the actuators. This signal to the actuators is based on the programming instructions in the PLC. The programming instructions is the flow of the whole process that the system has to follow in order to get the desired result.

Fig 3.1 shows the block diagram of our project. The PLC module receives its inputs from the push buttons, relay module and the proximity sensor. The push buttons are used to operate the Start/Stop of the system. The relay module provides the input from the NodeMCU, which receives the data from the user via the Blynk application. The proximity sensor detects the bottle.

The outputs are shown towards the right of the PLC module in the block diagram. LED indicators are used to indicate system on, motor running, etc. DC motor, Solenoid Valve and the Water pump are all controlled by the PLC through a relay module. The LCD display shows the quantity that is being filled into the bottle and also the count of the bottles.

3.2 HARDWARE DESCRIPTION

Hardware refers to the physical elements present in any system. In our project, PLC plays a very important role as it helps in functioning of the system based on the ladder logic (program instructions). The NodeMCU is responsible to communicate the input from the user to the PLC. The various components used in project are discussed in the further topics.

3.2.1 PROGRAMMABLE LOGIC CONTROLLER (PLC)

A programmable logic controller, also known as PLC, is a digital computer used as a major component in the field of automation. It acts as an interface between the inputs and the program. It is designed to operate reliably under harsh industrial environments – such as extreme temperatures, wet, dry, and/or dusty conditions. It has been adapted for the control of manufacturing processes, such as assembly lines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis.

A PLC continuously monitors the state of the input devices that are connected, makes decisions based on the program instructions and controls the state of the output devices.

The five different programming languages used for PLCs are

- Ladder Diagram (LD)
- Structured Diagram (SD)
- Instruction List (IL)
- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)

Among the above mentioned programming languages, we have used ladder logic programming.

The PLC that we have used in our project is FX1N-20MR which is discussed in Chapter 4.

3.2.2 PROXIMITY SENSOR

A proximity sensor is a non-contact sensor that detects the presence of an object when it enters the sensor's field. This object that is to be detected by the proximity sensor is often referred to as the target. There are many types of proximity sensors, and they each sense targets in distinct ways. The two most commonly used proximity sensors are the inductive proximity sensor and the capacitive proximity sensor.

Proximity sensors have high reliability and a long functional life because of the lack of physical contact between the sensor and the target object.

An inductive proximity sensor is a non-contact electronic proximity sensor. It is used for positioning and detection of metal objects. The sensing range of an inductive switch is dependent on the type of metal being detected. The proximity sensor that is used in the project is a 3-wired NPN inductive proximity sensor RM18 (Fig 3.2.2). Table 3.2.2 provides the specifications of this proximity sensor.

Fig 3.2.2: NPN inductive proximity sensor

The features of this proximity sensor are as follows:

- Operating Voltage (VDC): 6 ~ 36.
- Detection Distance (mm): 8.
- Outer Thread Size: M18.
- Object Type: Inductive.
- Output Type: NPN Normally Open Three-line.
- Installation Type: Non-Flush (Unshielded).
- Max Output Current: 200 mA.
- Setting Distance (mm): 0 ~ 5.6.

Table 3.2.2 Specifications of RM18 sensor

Model	Orange RM18~3008 NA
Operating Voltage (VDC)	6 ~ 36
Detection Distance (mm)	8
Outer Thread Size	M18
Object Type	Inductive
Output Type	NPN Normally Open Three-line
Installation Type	Non-Flush (Unshielded)
Max Output Current (mA)	200
Setting Distance (mm)	0 ~ 5.6
Insulation Resistance (MΩ)	50
Operating Temperature (°C)	-25 to 70
Storage condition (℃)	-30 to 80
Cable Diamensions	Ø5, 3-Wire, 2 Meter
Weight (gm)	100
Shipment Weight	0.12 kg
Shipment Dimensions	11 × 8 × 5.5 cm

3.2.3 IR SENSOR (INFRARED SENSOR)

Infrared radiation is an electromagnetic wave with wavelength of 700nm to 1mm. It is emitted by objects with temperature above 0 kelvin. Infrared radiation intensity and wavelength depends on temperature of the object. An infrared (IR) sensor (shown in Fig 3.2.3.1) is an electronic device that measures and detects infrared radiation from its surrounding environment.

An IR sensor consists of an emitter and receiver (shown in Fig 3.2.3.2). The emitter is an IR LED and the detector is an IR photodiode. The IR photodiode is sensitive to the IR light emitted by an IR LED. When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the reception by the IR receiver, the output of the sensor is defined.

Fig 3.2.3.1: IR sensor

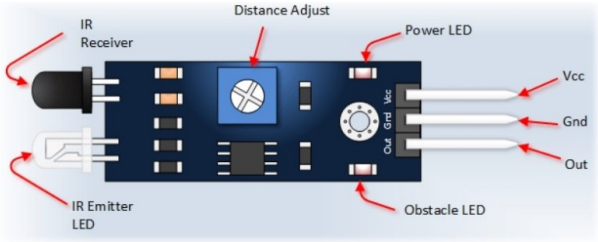


Fig 3.2.3.2 : Parts of IR sensor

3.2.4 PUSH BUTTONS

A push-button is a simple switch mechanism to control some part of the machine or some part of the process. These buttons are made out of hard material, usually plastic or metal. The surface is shaped in a way, so that it can be easily pushed or depressed by humans.

With respect to our project, push buttons (Fig 3.2.4) are used as Master START and Master STOP buttons. The Master START button turns ON the entire system whereas the Master STOP button stops the entire process once pressed.

Fig 3.2.4: Push Buttons

3.2.5 RELAY

A relay is an electrically operated switch. Relays are mainly used where it is necessary to control a circuit by a separate low-power signal, or where several circuits are supposed to be controlled by a single signal. We have used 2 different relays – 5V relay (shown in Fig 3.2.5.1) and 12V relay (shown in Fig 3.2.5.2)

5V Four-Channel Relay Module:

The four-channel relay module contains four 5V relays and the associated switching and isolating components, which makes interfacing with a microcontroller or sensor easy with minimum components and connections. The contacts on each relay are specified for 250VAC and 30VDC and 10A in each case, as marked on the body of the relays. Since it is a low triggered relay, we have used a NOT gate (IC 7404) to make the relay high triggered.

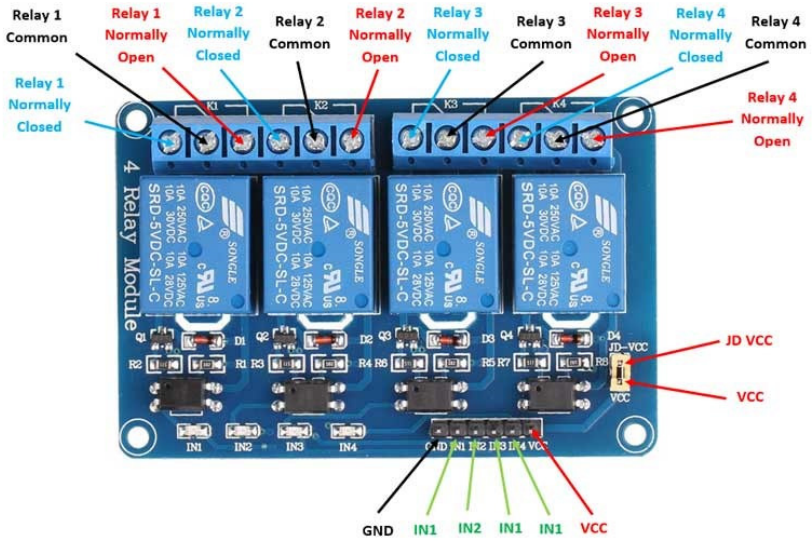


Fig 3.2.5.1: 5V four channel relay

Specifications:

- Supply voltage – 3.75V to 6V
- Trigger current – 5mA
- Current when the relay is active - ~70mA (single), ~300mA (all four)
- Relay maximum contact voltage – 250VAC, 30VDC
- Relay maximum current – 10A

12V Four-Channel Relay Module:

The four-channel relay module contains four 12V relays and the associated switching and isolating components, which makes interfacing with a microcontroller or sensor easy with minimum components and connections

The triggering of 4 Channel Relay Module is reliable, more stable. Relays terminals (C, NC, NO) are accessible through screw terminals which makes wiring up the board very easy. The use of such high-voltage relay eliminates the risk of heating up of the relay as electromechanical relay limit the current consumption in accordance with a voltage rating.

Fig 3.2.5.2: 12V four channel relay

Four-Channel Relay Module Pin Descriptions

Table 3.2.5 provides the pin description of a four-channel relay module.

Table 3.2.5 Pin description

Pin Number	Description
GND	Ground reference for the module
IN1	Input to activate relay 1
IN2	Input to activate relay 2
IN3	Input to activate relay 3
IN4	Input to activate relay 4
VCC	Power supply for the relay module
VCC	Power supply selection jumper
JD-VCC	Alternate power pin for the relay module

3.2.6 LEDs

A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. The colour of the light, corresponding to the energy of the photons, is determined by the energy required for electrons to cross the band gap of the semiconductor.

LEDs have many advantages, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. LEDs are used in applications as diverse as aviation lighting, fairy lights, automotive headlamps, traffic signals, camera flashes, medical devices and many more.

Fig 3.2.6: LEDs

An indicator LED (shown in Fig 3.2.6) is used to show the status of certain operations such as system ON, motor ON, solenoid valve ON, etc.

In our project:

- Green LED indicates system ON
- Yellow LED indicates motor ON
- Blue LED indicates solenoid valve ON

3.2.7 LCD

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc.

16X2 LCD:

A 16X2 is shown in Fig 3.2.7. The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.

- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

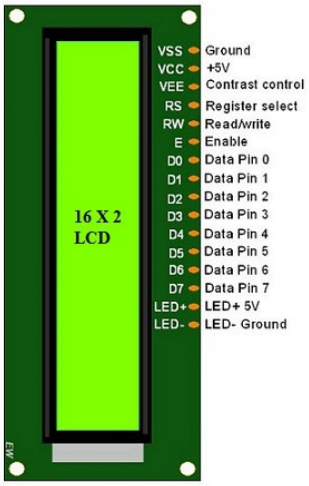


Fig 3.2.7: Pin diagram of LCD 16X2

- **Pin1** (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- **Pin2** (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- **Pin3** (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- **Pin4** (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- **Pin5** (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).

- **Pin6** (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- **Pins7-14** (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the Microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- **Pin15** (+ve pin of the LED): This pin is connected to +5V
- **Pin 16** (-ve pin of the LED): This pin is connected to GND.

3.2.8 SOLENOID VALVE

Solenoid valves (shown in Fig 3.2.8) are usually used to control the flow of any type of liquids. They are electromechanically operated valves. Their task is to open and close the valve automatically, based on the requirement the entry of water is controlled, in automatic irrigation sprinkler systems and many of the mechanism. There are many other applications of the solenoid valve, in washing machines and dishwashers where others.

Their tasks are to shut off, release, dose, distribute or mix fluids. They are found in many application areas. Solenoids offer fast and safe switching, high-reliability, long service life, good medium compatibility of the materials used, low control power and compact design.

Fig 3.2.8.: Solenoid valve

3.2.9 DC PUMP

A pump is a device that moves fluids (liquids or gases), or sometimes slurries, by mechanical action. Pumps can be classified into three major groups according to the method they use to move the fluid: direct lift, displacement, and gravity pumps.

Fig 3.2.9: DC Pump 12V

Pumps operate by some mechanism (typically reciprocating or rotary), and consume energy to perform mechanical work moving the fluid. Pumps operate via many energy sources, including manual operation, electricity, engines, or wind power, come in many sizes, from microscopic for use in medical applications to large industrial pumps. Fig 3.2.9 shows a DC water pump.

3.2.10 DC GEARED MOTOR

Geared DC motor (shown in Fig 3.2.10) can be defined as an extension of DC motorA geared DC Motor has a gear assembly attached to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM .The gear assembly helps in increasing the torque and reducing the speed. Using the correct combination of gears in a gear motor, its speed can be reduced to any desirable figure. This concept where gears reduce the speed of the vehicle but increase its torque is known as gear reduction.

Fig 3.2.10: DC geared motor

3.2.11 CONVEYOR BELT

A conveyor belt is one of the types of a conveyor systems. They are majorly used in industries for material handling like moving boxes from one place to another within a factory or even for bulk material handling where large quantity of resources are moved like sand, grain etc.

This conveyor belt system consists of two pulleys (or also called as drums) with a closed loop of the conveyor belt that rotates about them. The pulleys are powered and therefore they help in moving the belt and as well as the material on the belt in the forward direction. The powered pulley is called the drive pulley while the unpowered pulley is called the idler pulley. There are two main industrial classes of belt conveyors.

One of them are used for general material handling such as those moving boxes along inside a factory and the other type are used for bulk material handling such as those used to transport large volumes of resources and agricultural materials, such as grain, salt, coal, sand and more.

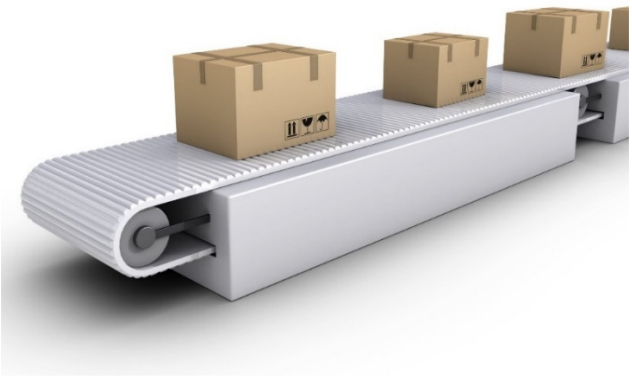


Fig 3.2.11: Conveyor Belt

Belt conveyors (shown in Fig 3.2.11) are the most commonly used powered conveyors because they are the most versatile and the least expensive. Products are conveyed directly on the belt so both regular and irregular shaped objects, large or small, light and heavy, can be transported successfully.

3.2.12 POWER SUPPLY

We have used a 12V Adaptor to power the entire circuit. This 12V power is required to provide power to PLC and its I/O devices. In order to provide power to LCD, NodeMCU, 5V relay, and IR sensor this 12V is converted to 5V by using LM7805 IC. Fig 3.2.12 shows the circuit diagram for a 12V to 5V converter.

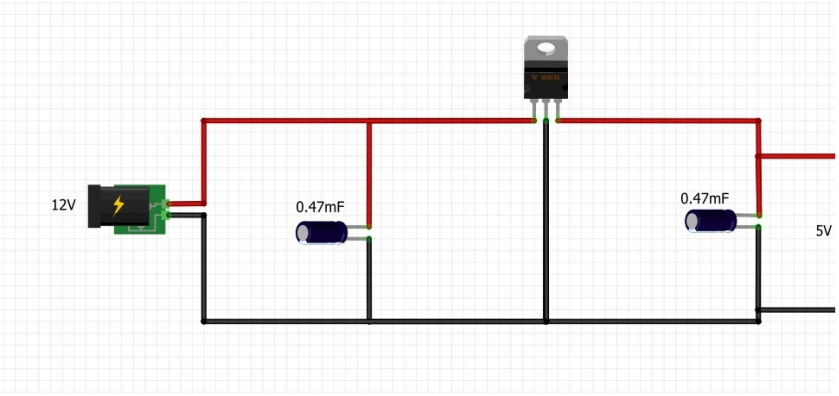


Fig 3.2.12: 12V to 5V converter

3.2.13 NODE-MCU

INTRODUCTION

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work drastically. Connecting the physical objects to the digital world has become a lot easier and simpler. A Shanghai-based Semiconductor Company, named Espressif Systems, released a bite-sized WiFi enabled microcontroller – ESP8266. It can be used to monitor and control things from anywhere in the world which is perfect for just about any IoT project.

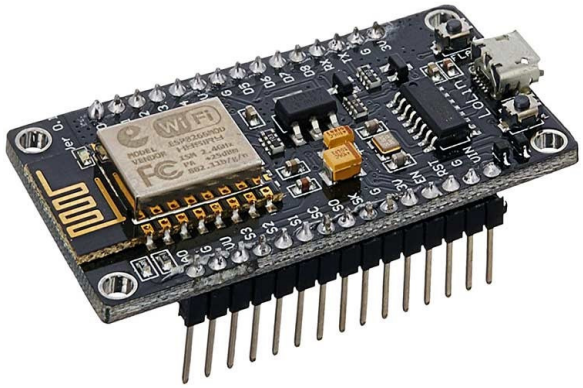


Fig 3.2.13.1: Node MCU ESP8266

The **NodeMCU ESP8266** (shown in Fig 3.2.13.1) **development board** comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can connect to a WiFi network and interact with the Internet. It can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU more versatile.

Specifications and features of Node-MCU ESP8266

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz

- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

POWER REQUIREMENT

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which is enough when ESP8266 pulls as much as 80mA during RF transmissions.

The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

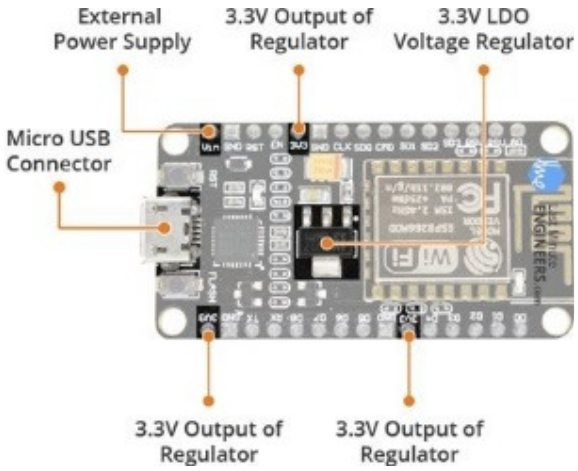


Fig 3.2.13.2: Node MCU power pins

NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface. Fig 3.2.13.2 shows the power pins in a Node MCU ESP 8266 board.

PIN CONFIGURATION OF NODE MCU

The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. Fig 3.2.13.3 shows the different pins of NodeMCU-ESP8266.

These pins can be assigned to all sorts of peripheral duties, including:

- ADC channel – A 10-bit ADC channel.
- UART interface – UART interface is used to load code serially.
- PWM outputs – PWM pins for dimming LEDs or controlling motors.
- SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface – I2S interface if you want to add sound to your project.

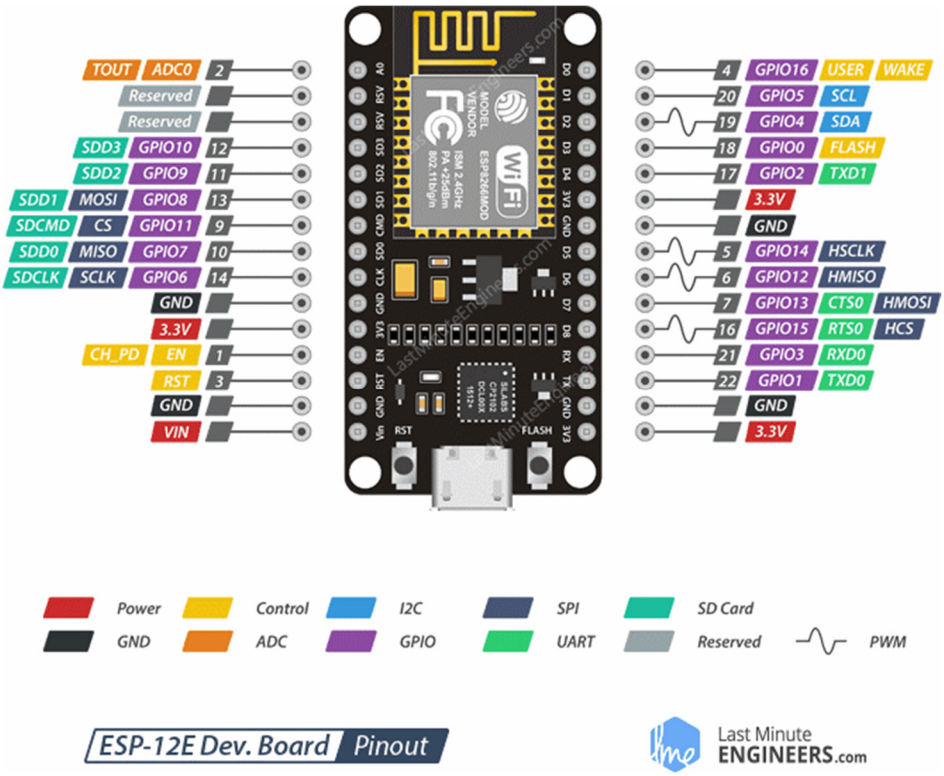


Fig 3.2.13.3: Node MCU pin diagram

Power Pins There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND is a ground pin of ESP8266 NodeMCU development board.

I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins ESP8266 NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

SDIO Pins ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 µs to 10000 µs, i.e., between 100 Hz and 1 kHz.

Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- RST pin – RST pin is used to reset the ESP8266 chip.
- WAKE pin – Wake pin is used to wake the chip from deep-sleep.

CHAPTER 4

PLC DETAILS

4.1 FX1N – 20MR PLC

This programmable logic controller uses industrial-grade 32-bit STM32F103 MCU .with strong anti-interference and faster speed, high quality components, stable performance, on-line download, on-line monitoring, automatic save when power off. The program for this PLC is written using ladder logic programming language.

This PLC supports Mitsubishi GX-Developer, and GX-work2 PLC software. The port that is used for program upload, download, test and monitor the PLC is known as the Programing port. It is reusable, stable, and easy to learn. The PLC transmission baud rate is 9600.

STM32F103 microcontrollers use the Cortex-M3 core, with a maximum CPU speed of 72 MHz. The portfolio covers from 16 Kbytes to 1 Mbyte of Flash with motor control peripherals, USB full-speed interface and CAN.

Fig 4.1 shows a FX1N-20MR PLC and Table 4.1 provides the specifications of this PLC.

The technical specifications of this PLC are as follows:

- Model: FX1N-20MR
- Make : Chinese (Mitsubishi Compatible)
- CPU Chip: STM32F103
- Output Type: Relay
- Working Voltage: DC24（V）
- Wide Power Supply Voltage: 10-28VDC
- Baud Rate: 9600
- Programming Software: for GX-Developer, for GX-works2.
- Size: Approx. 86 x 73 x 20mm / 3.4 x 2.9 x 0.7inch
- Weight: 120g

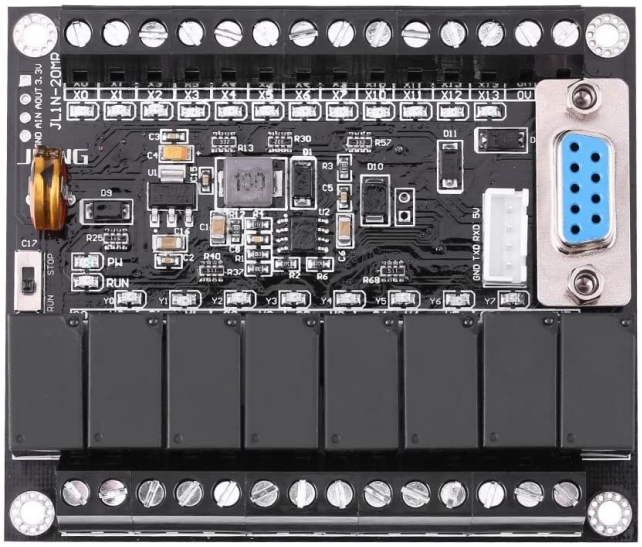


Fig 4.1: FX1N- 20MR PLC

Table 4.1 Specifications of FX1N-20MR PLC

ATTRIBUTE	DESCRIPTION
Model	FX1N-20MR
CPU Chip	STM32F103
Output Type	Relay
Operating Voltage	10-28V DC
Baud Rate	9600
Programming Software	GX-works2
Programming Language used	Ladder Logic
Number of Inputs	12
Number of Outputs	8
Input type	Sink
Communication port type	RS 232C
Network Type	point-to-point communication, Master-Slave communication, UART communication
Memory	16-bit Address bus

4.2 GX-WORKS 2

GX Works2 is a programming tool for designing, debugging, and maintaining programs on Windows. GX Works2 has improved functionality and operability, with easier-to-use features compared to existing GX Developer

Main functions of GX Works2:

GX Works2 can manage programs and parameters in units of projects for each programmable controller CPU.

- Programming:
  Programs can be created in a Simple project in a similar way with existing GX Developer. Structured programming in a structured project is also available with GXWorks2

- Setting parameters:
  The parameters for programmable controller CPUs and network parameters can be set with GX Works2. Intelligent function module parameter can be set as well.

- Writing/reading data to/from a programmable controller CPU:
  Created sequence programs can be written to/read from a programmable controller CPU using the Read from PLC/Write to PLC function. Also, with the online program change function, the sequence programs can be changed even when the programmable controller CPU is in RUN

- Monitoring/debugging:
  Created sequence programs can be written to the programmable controller CPU and device values at operation can be monitored online/offline

- Diagnostics:
  The current error status and error history of the programmable controller CPU can be diagnosed. With the diagnostics function, the recovery work is completed in a short time. With the System monitor function (for QCPU (Q mode)/LCPU), detailed information on such as intelligent function modules can be obtained. This helps to shorten the recovery work time at error occurrence

4.2.1 STEPS TO CREATE A LADDER LOGIC AND DUMP INTO PLC

- Open the software GX works 2
- In the menu bar select Project > New
- A small New project window will open shown in Fig 4.2.1.1. Select the following option:
    - Series -> FXCPU
    - Type -> FX1N/FX1NC
    - Project Type -> Simple Project
    - Language -> Ladder

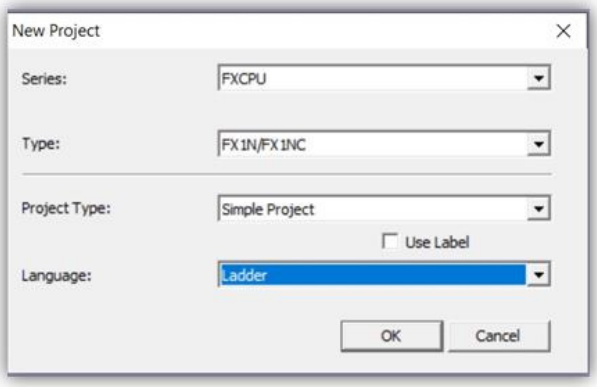


Fig 4.2.1.1: New Project Window

- Click Ok to proceed. A new window will open where we have to design the Ladder logic.
- We created the ladder logic using the components present in the toolbar (shown in Fig 4.2.1.2) at top.

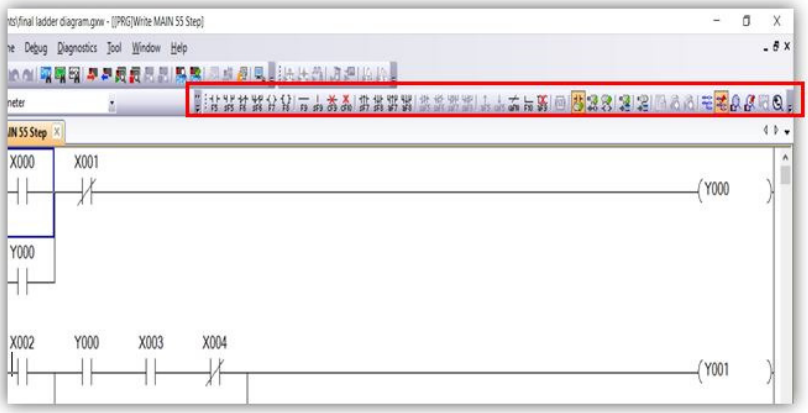


Fig 4.2.1.2: Toolbar

- After ladder logic is created, we tested the logic using the simulator function (shown in Fig 4.2.1.3) inside the software

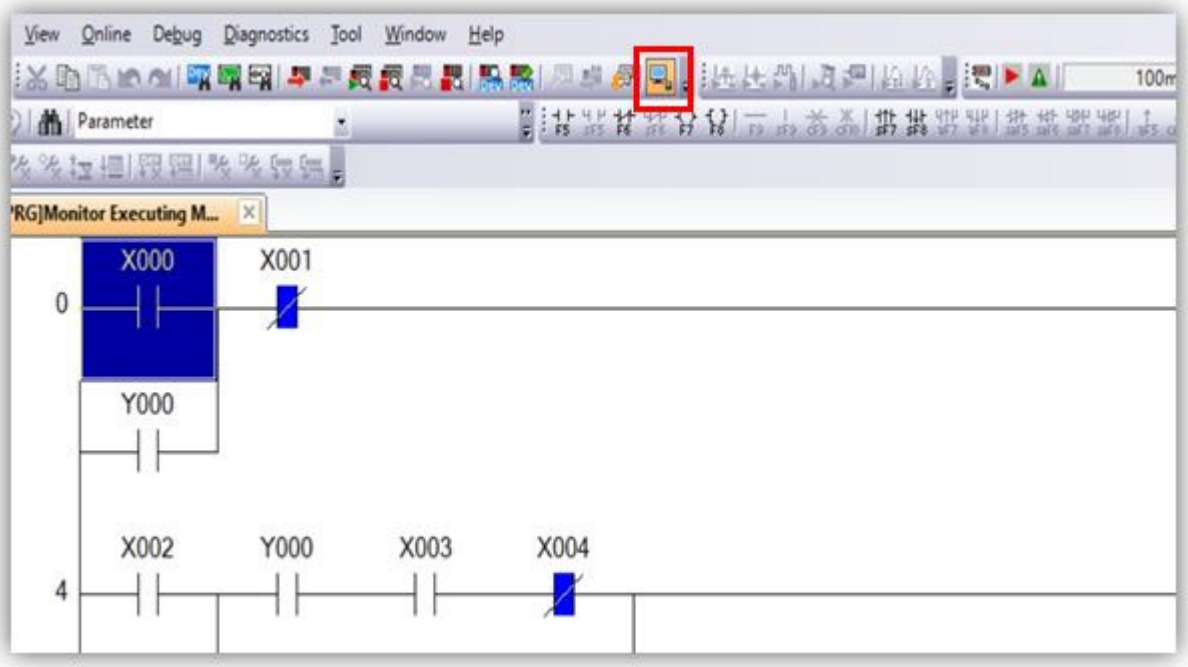


Fig 4.2.1.3: Simulator function

- After testing is completed, the code has to be dumped inside the PLC.
- To dump the code, select 'Connection Destination' in the Navigation pane
- Double click 'Connection1' shown in Fig 4.2.1.4, a new window will open
- Double click the 'Serial USB' option, a new small window will open show in Fig 4.2.1.4
- Select following:
    - RS 232C
    - COM Port -> Desired COM according to PC
    - Transmission rate -> 9.6Kbps
- Click 'Ok' to complete the settings.
- Now click the 'Connection Test' to check the PLC is connected properly or not
- A dialogue box will appear confirming the connection

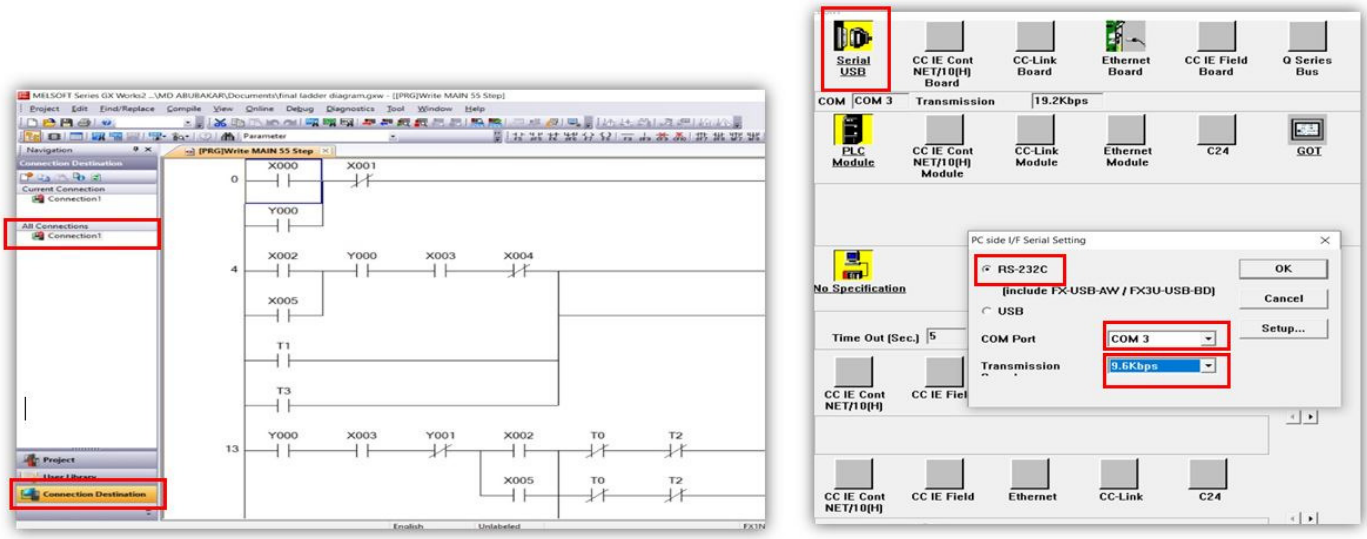


Fig 4.2.1.4: Connection

- After the settings, select 'online > Write to PLC' from menu bar shown in Fig 4.2.1.5, a new window will open
- In the new window select 'Parameter + Program' and click 'Execute', the code will get dumped inside the PLC shown in Fig 4.2.1.5.

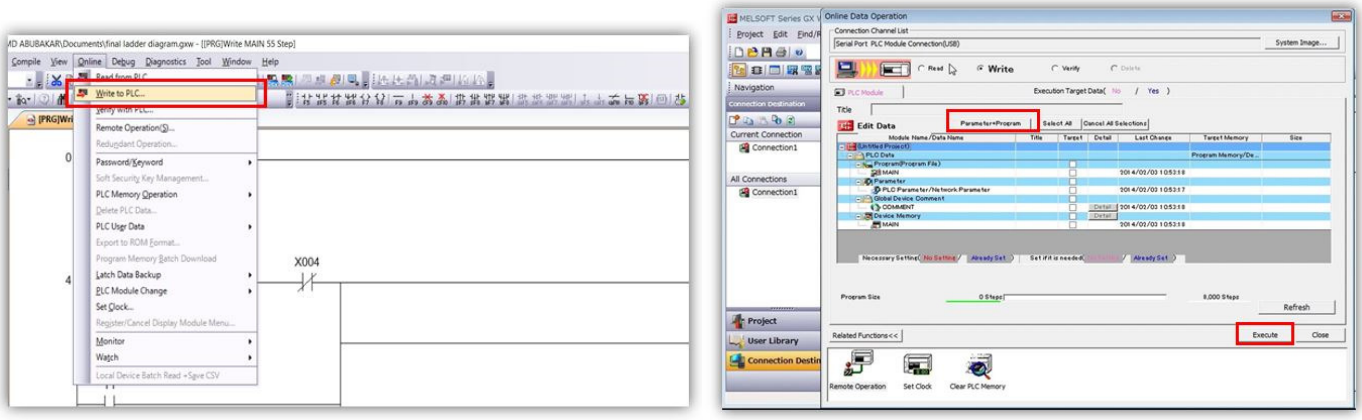


Fig 4.2.1.5: Dumping the code to PLC

CHAPTER 5

HARDWARE AND SOFTWARE DESCRIPTION

5.1 PCB CONNECTION DIAGRAM

A PCB (printed circuit board) is made where the Node MCU ESP8266 is connected to the IR sensor, 16X2 LCD and the 5V relay module. The connections diagram of the PCB is shown in figure 5.1. In our project we have used ESP8266 (NodeMCU) microcontroller to control the level of liquid which is to be filled using the Blynk app. The switches are created on the Blynk app (discussed in chapter 5.5), which is the user interface where the user can provide the system with the required input.

When the user enters the input via the switch in the app, the signal is sent to the NodeMCU at a particular PIN which is in turn connected to the relay.

- When the user selects Quantity1 in the app, the signal is sent to D7 pin of NodeMCU. This triggers the 5V relay at IN3 pin.
- When Quantity2 is selected in the app, the signal is sent to D8 pin of NodeMCU, which triggers the 5V relay at IN2 pin.
- When the motor on/off switch is selected this signal is sent to D0 pin of Node MCU which triggers the 5V relay at IN1 pin (mentioned in Table 5.1).

At each input signal the relay gets triggered. The output of the relay is connected to the PLC input. Therefore, as the user enters the input via the Blynk app, the PLC receives the input via the relay. This is how we provide the input to the PLC over the internet.

The Node-MCU ESP8266 apart from being connected to the relay, it also connects the IR sensor and 16X2 LCD. The Vcc and GND pins of IR sensor are connected to 5V and Ground respectively. The D0 pin of IR sensor is connected to RX pin of NodeMCU. The pin connections of the 16X2 LCD display with the NodeMCU are mentioned in Table 5.1. The 12V to 5V power supply converter circuit discussed in Chapter 3.2.12 is used here in order to provide power supply to the relay, LCD, IR Sensor and Node-MCU.

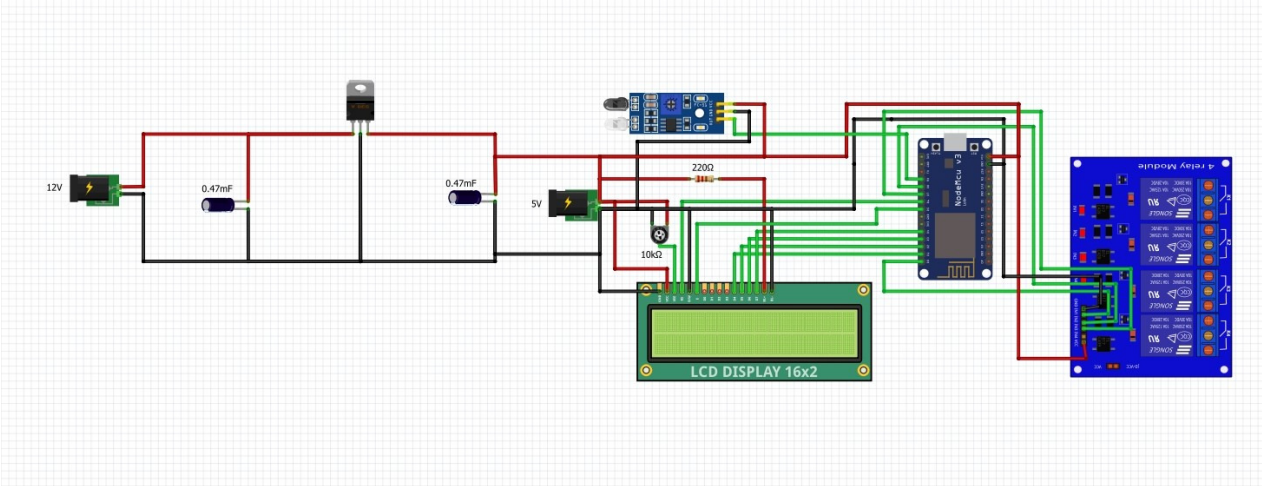


Fig 5.1: Connection diagram for input side

Connections for the above circuit diagram are as follows

Table 5.1 Connections for PCB board

FROM	TO
LCD Ground pin	Ground
LCD Vcc pin	+5V
LCD Vee pin	Potentiometer
LCD RS pin	Node MCU D6 pin
LCD R/W pin	Ground
LCD E pin	Node MCU D5 pin
LCD D4 pin	Node MCU D1 pin
LCD D5 pin	Node MCU D2 pin
LCD D6 pin	Node MCU D3pin
LCD D7 pin	Node MCU D4 pin
LCD BL+ pin	+5V
LCD BL- pin	Ground
Node MCU D0 pin	5V Relay IN1 pin
Node MCU D7 pin	5V Relay IN3 pin
Node MCU D8 pin	5V Relay IN2 pin
Node MCU RX pin	IR sensor D0 pin
Node MCU VIN pin	+5V
Node MCU GND pin	Ground
IR sensor Vcc pin	+5V
IR sensor GND pin	Ground

5.2 PLC CONNECTION DIAGRAM

The sensors and actuators are connected to the PLC board using 12V relay and the 5V relay. Fig 5.2 indicates the connections of the PLC board

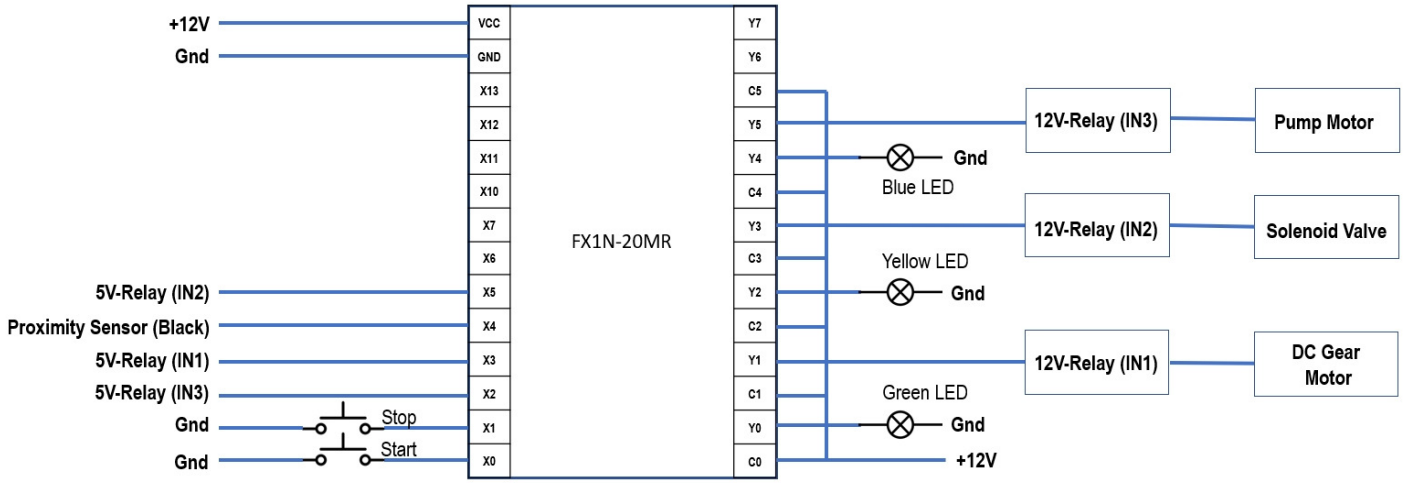


Fig 5.2: PLC connections

The Vcc pin of the PLC is connected to the 12V power supply and the GND pin is connected to GND. X0, X1, X2 etc. represent the input terminals whereas Y0, Y1, Y2 represent the output terminals.

- The X0 terminal is connected to the Green push button which acts as the Master Start button. The ground of the push button is connected to GND.
- The X1 terminal is connected to the Red push button which acts as the Master Stop button. The ground of the push button is connected to GND.
- The X2 terminal is connected to the IN3 pin of the 5V relay which is connected to NodeMCU (discussed in Chapter 5.1). When quantity1 is selected by the user using the Blynk app, it is provided to the PLC through this pin.

- The X3 terminal is connected to the IN1 pin of the 5V relay which is connected to NodeMCU (discussed in Chapter 5.1). When the motor on/off switch is selected by the user using the Blynk app, it is provided to the PLC through this pin.

- The X4 terminal of the PLC is connected to the brown wire of the proximity sensor. The blue and the black wire of the proximity sensor are connected to GND and 12V respectively.

- The X5 terminal is connected to the IN2 pin of the 5V relay which is connected to NodeMCU (discussed in Chapter 5.1). When quantity2 is selected by the user using the Blynk app, it is provided to the PLC through this pin.

- The Y0 terminal is connected to the Green LED which indicates the Pilot light. The ground of the LED is connected to GND

- The Y1 terminal is connected to the IN1 pin of the 12V relay. This is connected to the DC motor which is used to move the conveyor belt. The ground of the motor is connected to GND

- The Y2 terminal is connected to the Yellow LED which indicates the running of the Motor. The ground of the LED is connected to GND

- The Y3 terminal is connected to the IN2 pin of the 12V relay. This is connected to the Solenoid Valve which is used to fill the liquid into the bottles. The ground of the solenoid valve is connected to GND

- The Y4 terminal is connected to the Blue LED which indicates the filling of the liquid. The ground of the LED is connected to GND

- The Y5 terminal is connected to the IN3 pin of the 12V relay. This is connected to the Water pump which is used to pump the liquid from the container to the solenoid valve. The ground of the water pump is connected to GND

- C0, C1, C2 etc. are all common points which are used to complete the circuit. All these terminals are short and connected to 12V supply.

5.3 LADDER LOGIC

The nomenclature for inputs and outputs in the ladder logic diagram differs for the different brands of PLCs that we are using. In case of Allen – Bradley, Siemens PLC inputs are represented using "I" and outputs are represented using "Q" whereas for Delta and Mitsubishi PLC inputs are represented using "X" and outputs are represented using "Y".

The ladder logic is developed using GX-Works2 software. The ladder logic developed for our project is shown in Figure 5.3.

The inputs, outputs and timers used in the ladder logic are as follows

INPUTS
X0 - Master Start
X1 - Master Stop
X2 - Quantity 1 (via app)
X3 - Motor (via app)
X4 - Proximity Sensor
X5 - Quantity 2 (via app)

OUTPUTS
Y0 - System pilot light
Y1 - Motor
Y2 - Motor light
Y3 - Solenoid Valve
Y4 - Solenoid Valve light
Y5 - Water pump

TIMERS
T0 - Timer for filling Quantity 1
T1 - Timer for filling Quantity 1 + Delay
T2 - Timer for filling Quantity 2
T3 - Timer for filling Quantity 2 + Delay

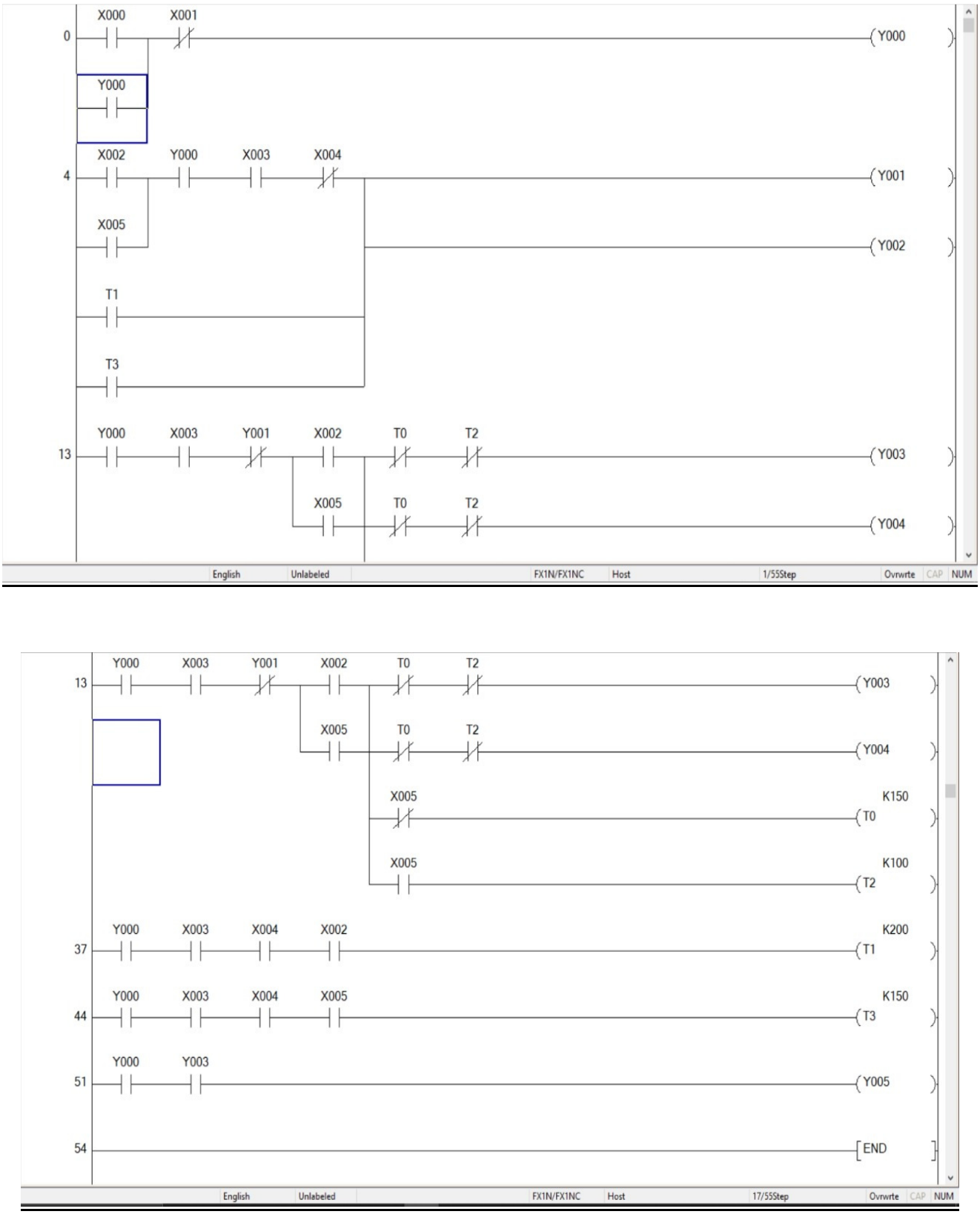


Fig 5.3: Ladder Logic for the project

The system turns ON once the Master Start button (X0) is turned on and the Master Stop button (X1) is in OFF condition. This system ON condition is indicated by a Pilot Light LED (Y0). Latching is used in the first rung of the ladder logic (shown in Fig 5.3) because we are using push buttons.

When the pilot light (Y0) is in ON condition, the quantity of Quantity1 (X2) or Quantity2 (X5) is selected by the user via the app and the motor switch (X3) is turned ON through the app, the motor (Y1) and the motor light (Y2) , both are turned ON. The motor is in the moving condition only when the bottle is not detected by the proximity sensor. Therefore a normally closed state of proximity sensor (X4) is used. The motor automatically starts running once the bottle is filled and delay timing is completed. Therefore T1 and T3 are added so that the motor starts moving once the timing of T1 for Quantity1 or T3 for Quantity2 is completed.

As the motor (Y1) is turned off, when the bottle is sensed by the proximity sensor, the solenoid valve (Y3) and the Filling LED (Y4) are turned ON. Along with the solenoid valve and the filling LED, timers (T0 or T2) are started depending on which quantity is selected. If Quantity1 (X2) is selected then T0 which is for 15 seconds, whereas if Quantity2 (X5) is selected then T2 which is for 10 seconds will turn on.

Simultaneously two more timers T1 and T3 which are for 20 seconds and 15 seconds are started. These two timers are included in order to incorporate a delay of 5 seconds for both the quantities.

When the pilot light (Y0) and the solenoid valve (Y3) are in their ON state, the water pump (X5) is turned ON.

When the Master STOP button (X1) is pressed, the normally closed state of X1 becomes normally open, and the connection is broken in the first rung, which in turn causes all the other process in the system to come to a halt.

5.4 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application which can be used on systems with Windows, Linux or Mac OS. It is used to write and upload programs to Arduino compatible boards. It can also be used for other boards by downloading third party cores. Arduino IDE supports languages C and C++ using special rules for coding structuring.

The user written code just requires two basic functions – for starting the sketch and the main programming loop which is compiled and linked with a program main() into an executable cyclic executive program .

It employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the board by a loader program in the board's firmware.

5.4.1 PROGRAMMING NODEMCU ESP8266 WITH ARDUINO IDE

With respect to our project, we use Arduino IDE in order to program the NodeMCU ESP8266 development board. In order to program NodeMCU using arduino IDE we require arduino IDE, USB cable and the NodeMCU ESP 8266 board.

But before we start programming the NodeMCU development board, we first need to install the ESP8266 core into arduino IDE. This can be done by following the steps below:

- The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed in our PC
- We need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json into the Additional Board Manager URLs text box situated on the bottom of the window (Show in Fig 5.4.1)

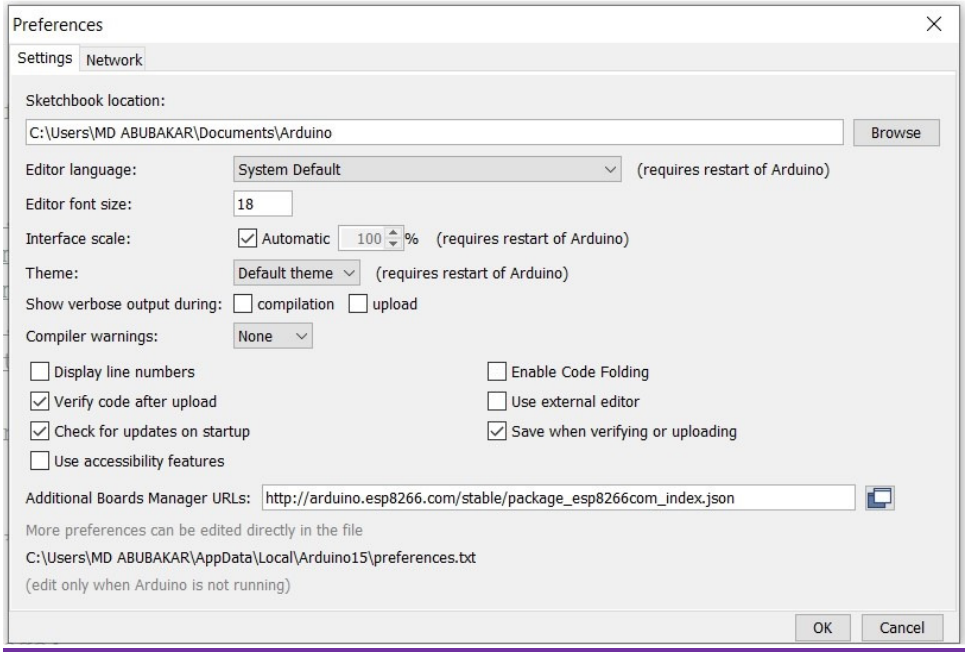


Fig 5.4.1: Updating the board manager Arduino IDE

- Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There will be a couple new entries in addition to the standard Arduino boards. We can filter our search by typing esp8266. Click on that entry and select Install.

- The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small INSTALLED text will appear next to the entry. We can now close the Board Manager.

5.5 BLYNK APPLICATION

Blynk is an Internet of Things platform, which makes controlling hardware remotely and visualizing its data very easy. We can create our own interfaces using the free Blynk App. Every Wi-Fi, Bluetooth/BLE, Ethernet and Serial device is able to connect to the Blynk cloud or a locally running server. Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, store data, visualize it and do many other things.

FEATURES OF BLYNK APP

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using: Wifi, Bluetooth, ethernet, USB(serial) , GSM
- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins
- History data monitoring via SuperChart widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.
- New features are constantly added

5.5.1 BLYNK APP SET-UP

In order to provide the input from the Blynk app, a set-up has to be made in the app. After downloading the app, the below steps were followed to set up the app shown in Fig 5.5.1.1

- After downloading the Blynk app we need to create a New Blynk account using our email ID
- After successfully logging into our account, we start by creating a new project
- On the next window, we choose Device – NodeMCU and Connection Type – WiFi

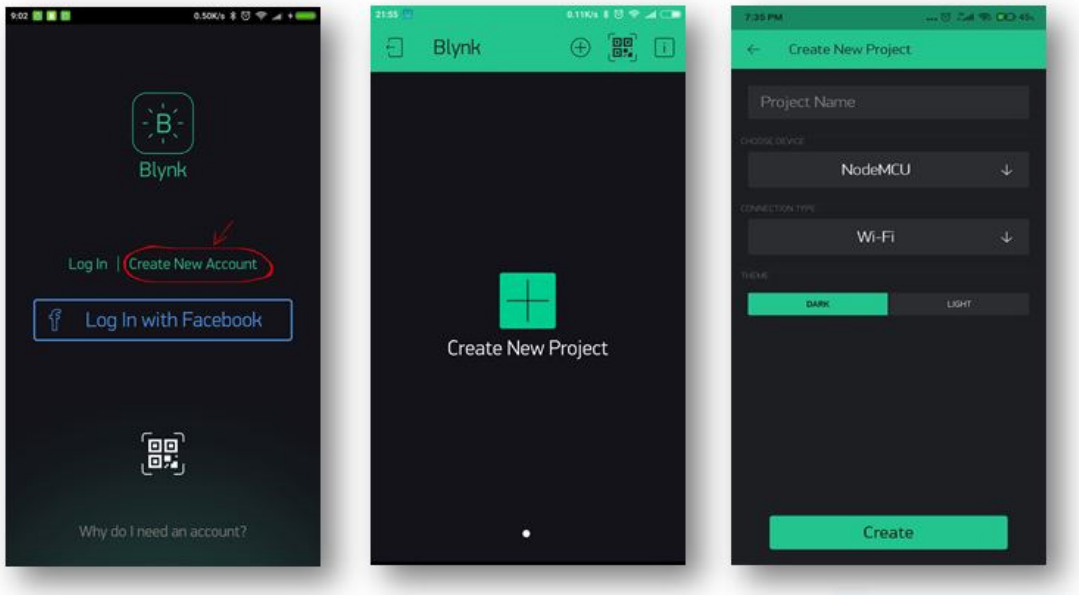


Fig 5.5.1.1: Creating an account and choosing hardware

- Auth Token is a unique identifier which is needed to connect the hardware to our smartphone. Every new project we create will have its own Auth Token. We'll get Auth Token automatically on our email after project creation.
- Click on blank screen to enter inside the New Project window. To add the widgets click the (+) button on top right.
- For our project we added 3 button widgets chosen from the widget box

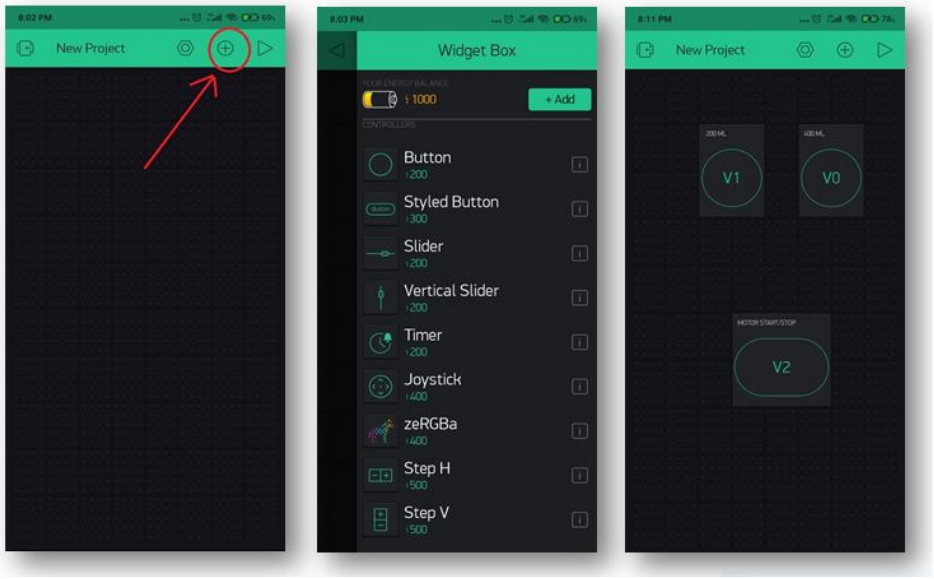


Fig 5.5.1.2: Creating a new project and add buttons

Buttons settings

- Upon clicking the button that we created in the previous step, a new window will open
- We can provide the name for the button as per our requirement. Our project has 3 buttons namely 200ml, 400ml and Motor On/Off
- Once we click on the PIN option, select Virtual and choose V0 for 1st button, V1 for 2nd button and V2 for 3rd button shown in Fig 5.5.1.2
- Change the mode of the button from PUSH to SWITCH

5.6 SETTING UP ARDUINO IDE FOR BLYNK APP

- Open Arduino IDE
- In the menu bar select Sketch > Include Library > Manage Libraries
- In search bar type ‘blynk’ and install the library ‘Blynk by Volodymyr…..’ shown in Fig 5.6.1
- After install, the Arduino IDE is ready for the code to be written

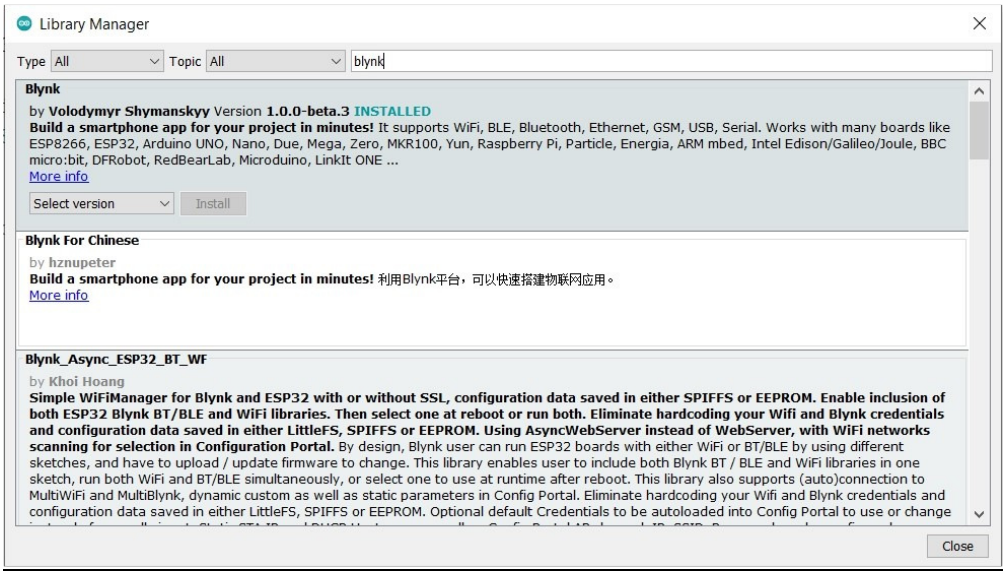


Fig 5.6.1: Blynk library set up

Dumping the code inside the NodeMCU:

- Write the code in the code editor window
- Include two libraries:

  <ESP8266WiFi.h> and <BlynkSimpleEsp8266.h>

- The auth token we received on mail should be pasted inside the code as

  char auth[] = " Auth Token ";

- To connect to desired wi-fi network we include the code as:

  char ssid[] = " Network SSID ";

  char pass[] = " Network Password "; Shown in Fig 5.6.2

- After completing the code, go to Tools > Board: "NodeMCU 1.0… > ESP8266 Boards (2.7.4) > NodeMCU 1.0 (ESP-12E Module)
- Now again we need to select the port by going to tools > Port > COM$ where '$' could be any value depending on which USB Port is used.
- Now click upload code to start dumping
- After dumping the NodeMCU will connect to the wi-fi and we can check the status on serial monitor shown in Fig 5.6.3
- Now the NodeMCU is ready to be used by connecting it to PCB board

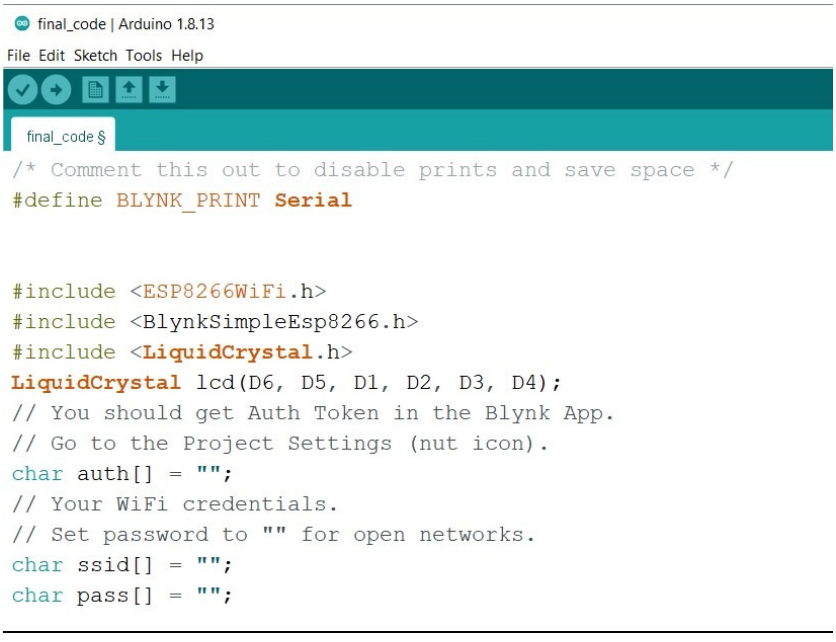


Fig 5.6.2: Setting up auth token and WiFi

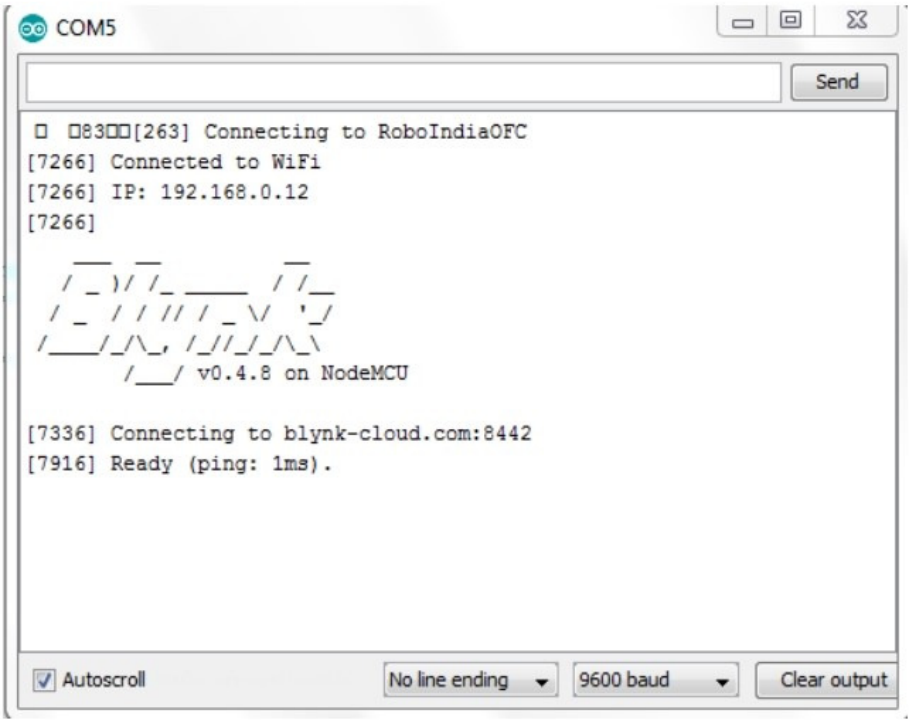


Fig 5.6.3: WiFi status check using Serial monitor

CHAPTER 6

RESULT AND CONCLUSION

6.1 RESULT

- The working model is successfully connected to the Blynk app via WiFi, through which Motor On/Off and the quantity switches can be controlled
- On providing the input through the app, the conveyor belt starts moving and the proximity sensor is able to detect the bottle
- The conveyor belt stops as the proximity sensor senses the bottle
- The solenoid valve is able to fill the pre-set quantity of liquid into the bottle within the pre-set time
- Once the bottle is filled, the conveyor belt start moving with the liquid filled bottle
- In the process of the movement, IR sensor detects the bottle and the count is displayed on the LCD

6.2 CONCLUSION AND FUTURE SCOPE

The main purpose of introducing automation in industries is to improve the productivity, which helps in economic growth. PLCs are used in industries in order to control the whole mechanism of the system.

Though the initial installation cost of PLC is not cheap, they can work efficiently for a very long time. They also help to increase the productivity and the performance.

Since IoT is used in this project, it can be operated via the phone application, reducing the requirement of human intervention.

By using a stronger solenoid valve and a jet nozzle it is possible to increase the productivity, as the time in filling the bottles would get reduced. The part of the mechanism that should be given extra care is the positioning of the nozzle.

The system can be redesigned for different sizes of bottles and also to increase the productivity. A capping section also can be added where the filled bottles are capped using an automatic machine.

We can also add sensors for the alarms which can indicate any mishandling in between the process.

APPLICATIONS

- Packing of bottles for pharmaceutical industries
- Packing of beverage bottles
- Packing of bottles for agricultural industries such as disinfectants, fertilizers
- In milk industries
- Packing of mineral water bottles