

**INDUSTRIAL WATER BOTTLE FILLING MACHINE**

By

**ANI STEEPHAN (1740319)**

**ANNA JOY (1740320)**

**LINDA MARY FRANCIS (1740327)**

**Under the guidance of**

**Prof: MUKUND NARAGUND**

**Prof: VINEETH VALSAN**

**Department of Physics and Electronics**

**A project report submitted in partial fulfilment of the award of degree of**

**B.Sc. (Electronics) of Christ (Deemed to be university), Bengaluru.**

[www.christuniversity.in](http://www.christuniversity.in)

**March 2020**



**CERTIFICATE**

This is to certify that, the project titled “**Industrial automatic water bottle filling machine**” is a bonafide record of the work done by “**ANI STEEPHAN**” in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science (Electronics) of Christ Deemed to be university, Bengaluru during the year 2019-2020.

**Head of the Department** **Project Guide** **Project Guide**

(Prof Benny Sebastian) (Prof: Mukund Naragund) (Prof: Vineeth Valsan)

Examination Date: **Name of the Student**: Ani Steephan

Examination Centre: Christ University **Class** : VI Semester PME

**Register No** : 1740319

**Examiners**

1. 2.



**CERTIFICATE**

This is to certify that, the project titled “**Industrial automatic water bottle filling machine**” is a bonafide record of the work done by “**ANNA JOY**” in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science (Electronics) of Christ Deemed to be university, Bengaluru during the year 2019-2020.

**Head of the Department** **Project Guide** **Project Guide**

(Prof Benny Sebastian) (Prof: Mukund Naragund) (Prof: Vineeth Valsan)

Examination Date: **Name of the Student** : Anna Joy

Examination Centre : Christ University **Class** : VI Semester PME

**Register No** : 1740320

**Examiners**

1. 2.



**CERTIFICATE**

This is to certify that, the project titled “**Industrial automatic water bottle filling machine**” is a bonafide record of the work done by “**LINDA MARY FRANCIS**” in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science (Electronics) of Christ Deemed to be university, Bengaluru during the year 2019-2020.

**Head of the Department** **Project Guide** **Project Guide**

(Prof Benny Sebastian) (Prof: Mukund Naragund) (Prof: Vineeth Valsan)

Examination Date: **Name of the Student:** Linda Mary Francis

Examination Centre: Christ University **Class** : VI Semester PME

**Register No** : 1740327

**Examiners**

1. 2.

**ACKNOWLEDGEMENT**

We thank God almighty for enabling us to complete our project by his blessings. We express our sincere gratitude to Rev. Fr. Dr. Abraham V.M, Vice Chancellor, CHRIST (Deemed to be University) for providing us the required facilities that helped us in accomplishing our project.

We are grateful to the Management and the Head of the Department Prof. Benny Sebastian, for giving us the opportunity to come up with such a project.

On the successful completion of this project, we would like to wholeheartedly thank our guide Prof. Mukund Naragund and Prof. Vineeth Valsan for their guidance, constructive feedbacks and constant encouragement throughout the project.

We take this opportunity to thank or friends for their suggestions which helped in the advancement. We would also like to appreciate the guidance given by the teachers during our project permission that has improved our presentation skills, thanks to their comments and advice. Finally, we thank all our classmates, supporting faculty members and families for extending their support for our development of the project.

**ABSTRACT**

The objective of our project is to design, develop, and monitor “Automatic bottle filling machine using Arduino”. Filling of any kind of liquid in a bottle can be carried out by machine. In industries, this type of work is done by embedded system. This project is based on the Industrial automation and is a vast application used in many industries like milk industries, chemical, food, mineral water industries. A prototype has been developed to illustrate the project. In this project, the filling of the bottle is controlled by using an Arduino. For the conveyor system, a dc motor has been selected for better performance and ease of operation. We use the Arduino program for filling and to place the position. In our project, we used less number of systems hence the overall cost has been reduced to an extent. The results of this project indicate that this system can be further developed and designed on a larger scale to meet the market requirements.

**TABLE OF CONTENT**

CHAPTER 1

1.1 INTRODUCTION

1.2. BLOCK DIAGRAM

1.3. EXPLANATION OF BLOCK DIAGRAM

1.4. FEATURES AND APPLICATION

CHAPTER 2

2.1. COMPONENTS USED

2.2. HARDWARE REQUIREMENTS

2.2.1. STEPPER MOTOR

2.2.2. STEPPER MOTOR DRIVER

2.2.3. ATMEGA328P

2.2.4. ARDUINO UNO

2.2.5. 7812 AND 7805

2.2.6. I2C

2.2.7. D882 AND BC547B

2.3. SOFTWARE REQUIREMENTS

2.3.1. IDE ENVIRONMENT

CHAPTER 3

3.1. CIRCUIT DIAGRAM

3.2. WORKING

CHAPTER 4

4.1 CODE

CHAPTER 5

5.1. ADVANTAGES

5.2. LIMITATIONS

5.3. FUTURE SCOPE

CHAPTER 6

6.1. REFERENCES

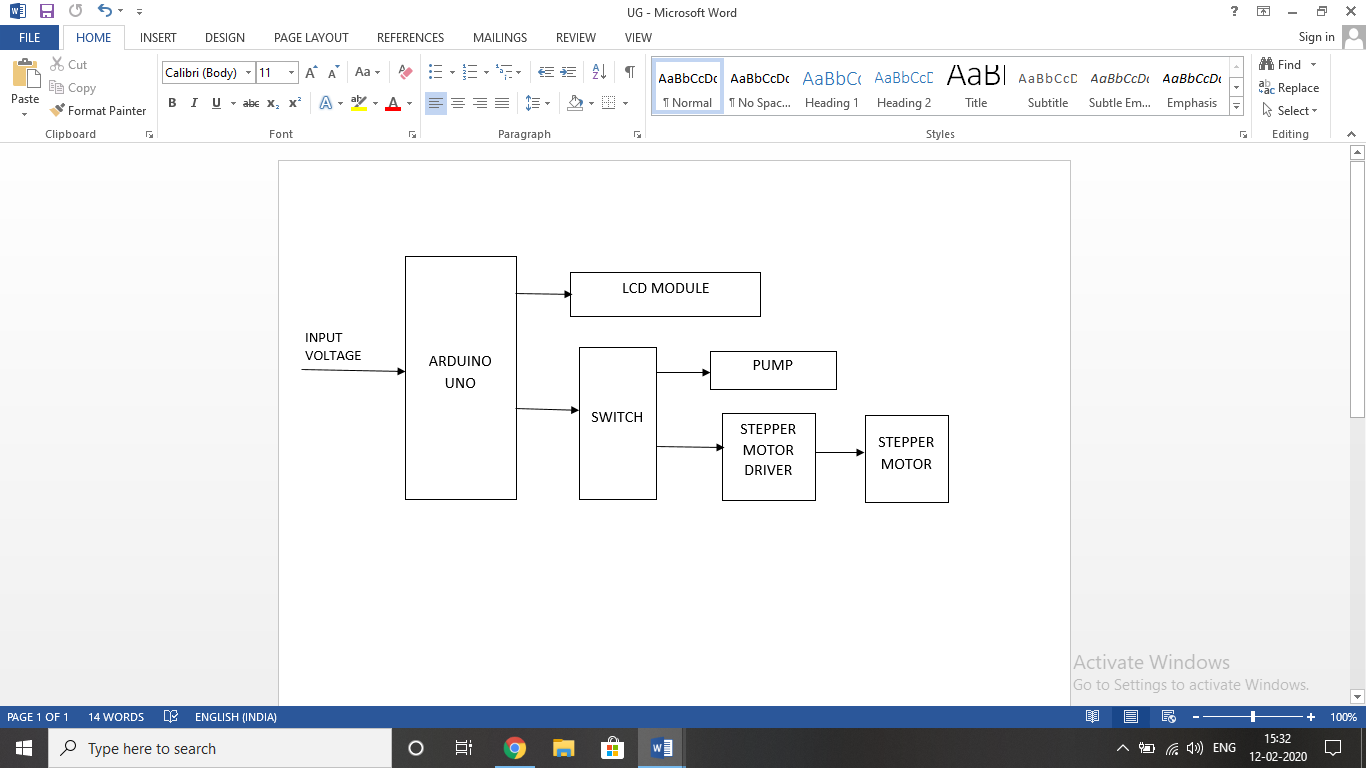
**CHAPTER 1**

* 1. **INTRODUCTION**

In developing countries, the production of beverages, milk, mineral water and cooking oil is a major support of the entire economy. Liquid filling machines are widely used in large scale beverage and bottling industries. In small industries bottle filling operation is done manually. The manual filling process has many shortcomings like spilling of water while filling it in bottle, equal quantity of water may not be filled, delay due to natural activities of human etc. This problem faced by small industries compels us to take up this project. . Our project aims to eliminate problem faced by small scale bottle filling system.

With this system that operates automatically, every process can be smooth and the process of refilling can reduce workers cost and operation time. The system operates by the program that designed to do the operation. The design of a bottle filling machines is multidisciplinary in nature as aspects that relate to mechanical, electrical and software engineering are synergized.

**1.2. BLOCK DIAGRAM**



**1.3. EXPLANATION OF BLOCK DIAGRAM**

A voltage source is first given to the Arduino board. Then the LCD module is activated .The switch initiates the working of the stepper motor driver and thus the stepper motor also starts rotating. The switch also initiates the pump to allow the water to be filled into the bottles.

**1.4. FEATURES AND APPLICATION**

**Features:**

* The volume of water to be filled in the bottles can be altered any time just by making changes in the program code.
* The number of bottles to be filled can be increased or decreased further according to our requirements by altering the source code.
* A buzzer sound is enabled to indicate the user about the end of the process.

**Applications:**

Used in industries like

* Cosmetics
* Chemical
* Food
* Beverages
* Bottled water industries

**CHAPTER 2**

**2.1. COMPONENTS USED**

The components used in this project are stepper motor (NEMA 17), A4988 driver, arduino Uno, 2x16 LCD, I2C, LM7812, LM7805, BC547B, D882, DC jack and pin, 5V buzzer, limit switch, tactile button, heat sink, resistors (1K,10K), capacitor 100µF, 8mm pulley, small pulley, water pump, belt, pipe, spacers, terminal block.

**2.2. HARDWARE REQUIREMENTS**

**2.2.1. STEPPER MOTOR (NEMA 17)**

|  |
| --- |
| https://reprap.org/mediawiki/images/thumb/c/c1/RepRap-NEMA-17.jpg/300px-RepRap-NEMA-17.jpg |

**NEMA 17**is a **hybrid stepping motor** with a 1.8° step angle (200 steps/revolution). Each phase draws 1.2 A at 4 V, allowing for a holding torque of 3.2 kg-cm. NEMA 17 Stepper motor is generally used in Printers, CNC machines and Laser Cutters.

|  |
| --- |
| Image result for nema 17 stepper motor wiring diagram" |

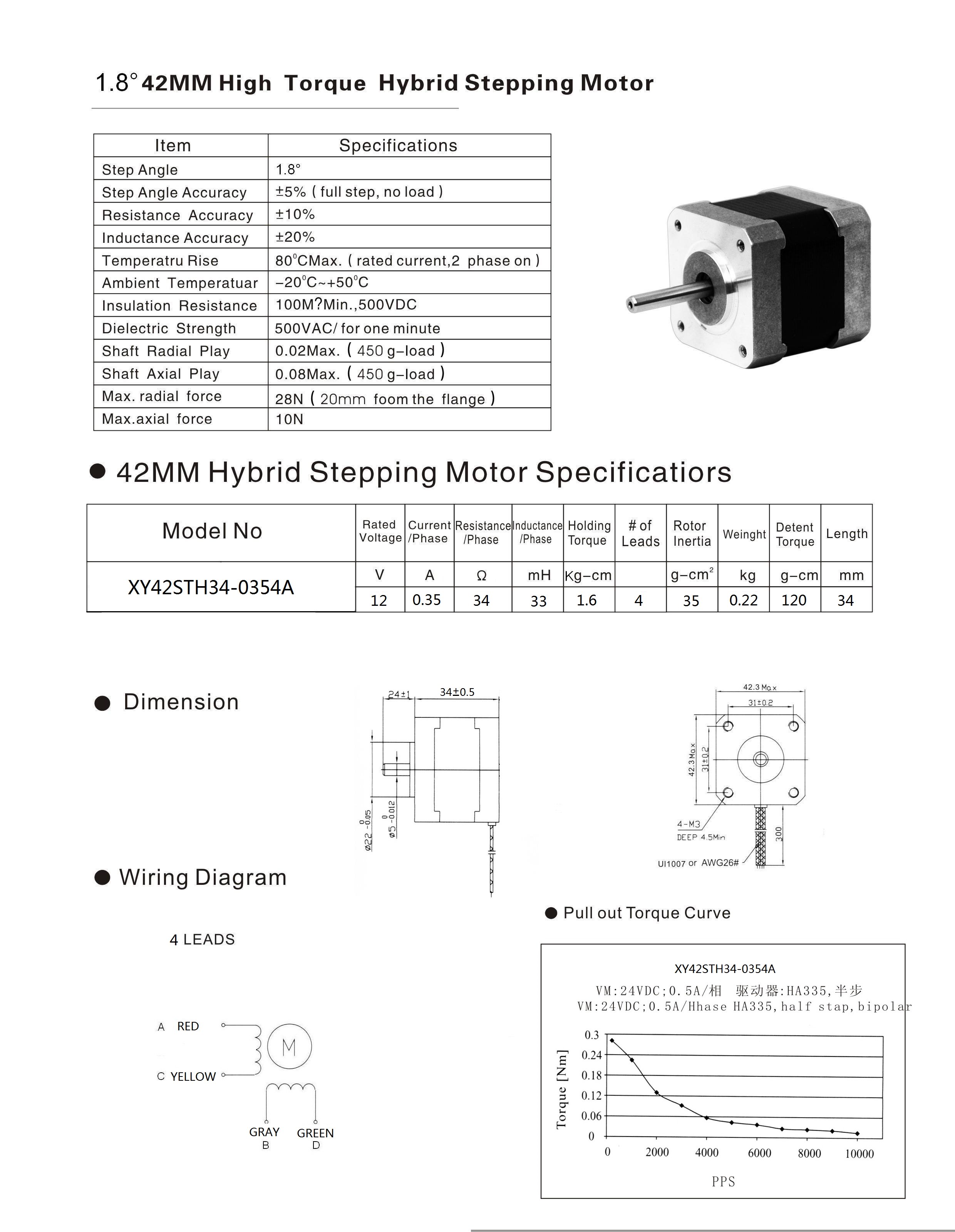
It has 6 lead wires. It can be operated at lower voltage but torque will drop.Since the motor work at 12V, it can provide high torque. Since this motor consumes high current, A4988 driver IC is mandatory. The coil diagram is as shown. The motor has six wires, connected to two split windings as is common for unipolar stepper motors. Black, Yellow, Green wires is part of first winding where Black is centre tap and Yellow and Green are coil end while Red, White and Blue is part of second winding in which White is centre tap and Red and Blue are coil end wires. In use, the centre taps of the windings (Black and White) are typically wired to the positive supply, and the two ends of each winding are alternately grounded through a drive circuit. As shown in the wiring diagram the order of the stator poles in the motor is A, B, A’, B’.

Technical specification:

Voltage - 12V DC

No. of phases – 4

Motor length – 1.54 inches

 **2.2.2. STEPPER MOTOR DRIVER**

|  |
| --- |
| A4988 Stepper Motor Driver Module |

The **A4988** is a complete **Micro stepping Motor Driver** with built-in translator for easy operation. The driver has a maximum output capacity of 35 V and ± 2 A. It has an operating voltage range from 8v-35v. It can operate bipolar stepper motors ie., we can control the stepper motor with just 2 pins from our controller or one for controlling the rotation direction and the other for controlling the steps. It has 5 steps of revolution: Full-step, haft-step, quarter-step, eight-step and sixteenth-step.

Pin Configuration

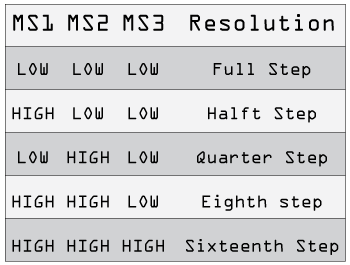
|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VDD & GND | Connected to 5V and GND of Arduino |
| VMOT & GND | Used to power the motor |
| 1A, 1B, 2A, 2B | Connected to the 4 coils of motor |
| DIRECTION | Motor Direction Control pin connected to the digital pin of arduino. |
| STEP | Steps Control Pin connected to digital pin od arduino. |
| MS1, MS2, MS3 | Resolution Selection Pins |
| SLEEP | Pins For Controlling Power States |
| RESET | Sets transistor to a predefined home state. |
| ENABLE | Turning ON/OFF the FET outputs. |

### **Features**

* Max. Operating Voltage: 35V
* Min. Operating Voltage: 8V
* Max. Current Per Phase: 2A
* Micro step resolution: Full step, ½ step, ¼ step, 1/8 and 1/16 step
* Reverse voltage protection: No

|  |
| --- |
| Image result for a4988 stepper motor driver specifications" |

* Dimensions: 15.5 × 20.5 mm (0.6″ × 0.8″)
* Short-to-ground and shorted-load protection
* Low RDS(ON) outputs
* Thermal shutdown circuitry



**2.2.3. ATMEGA328P**

The Atmel Pico Power ATmega328 is a low power CMOS 8bit micro-controller based on the AVR enhanced RISC architecture. **ATMEGA328P** is high performance form Microchip. It is the most popular of all AVR controllers as it is used in ARDUINO boards.

Features:

|  |  |
| --- | --- |
| CPU | 8-bit AVR |
| Number of Pins | 28 |
| Operating Voltage (V) | +1.8 V TO +5.5V |
| Number of programmable I/O lines | 23 |
| Communication Interface | Master/Slave SPI Serial Interface (17,18,19 PINS)  Programmable Serial USART (2,3 PINS) Two-wire Serial Interface (27,28 pins) [Can be used to connect peripheral devices like Servos, sensors and memory devices] |
| JTAG Interface | Not available |
| ADC Module | 6channels, 10-bit resolution ADC |
| Timer Module | Two 8-bit counters with Separate Prescaler and compare mode, one 16-bit counter with Separate Prescaler, compare mode and capture mode. |
| Analog Comparators | 1 (12,13 PINS) |
| DAC Module | Nil |
| PWM channels | 6 |
| External Oscillator | 0-4MHz @ 1.8V to 5.5V  0-10MHz @ 2.7V to 5.5V  0-20MHz @ 4.5V to 5.5V |
| Internal Oscillator | 8MHz Calibrated Internal Oscillator |
| Program Memory Type | Flash |
| Program Memory or Flash memory | 32Kbytes [10000 write/erase cycles] |
| CPU Speed | 1MIPS for 1MHz |
| RAM | 2Kbytes Internal SRAM |
| EEPROM | 1Kbytes EEPROM |
| Watchdog Timer | Programmable Watchdog Timer with Separate On-chip oscillator |
| Program Lock | Yes |
| Power Save Modes | Six Modes [Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby] |
| Operating Temperature | -40°C to +105°C (+105 being absolute maximum,  -40 being absolute minimum) |

**2.2.4. ARDUINO UNO**

|  |
| --- |
| Image result for arduino uno" |

The Arduino Uno is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) and other circuits. The board has 14 digital I/O pins (six capable of [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) output), six analog I/O pins, and is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts.

Technical specifications:

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 8-bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended Input Voltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Boot loader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

General pin functions:

* LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* VIN: The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* 3V3: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* GND: Ground pins.
* IOREF: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
* Reset: Typically used to add a reset button to shields that block the one on the board.

Special pin functions:

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode (), digitalWrite (), and digitalRead () functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labelled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analogReference () function.

In addition, some pins have specialized functions:

* Serial / [UART](https://en.wikipedia.org/wiki/UART): Pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* External interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) (pulse-width modulation): Pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite () function.
* [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* TWI (two-wire interface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): Pin SDA (A4) and pin SCL (A5)-Support TWI communication using the Wire library.
* AREF (analog reference): Reference voltage for the analog inputs.

**2.2.5. 7812 AND 7805**

|  |
| --- |
| Image result for lm7812 pin diagram" |

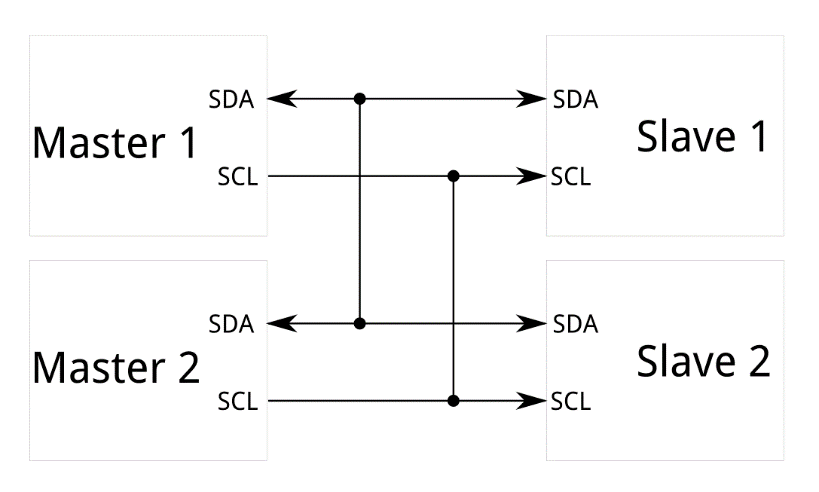
LM7812: This is a voltage regulator. The output voltage and the output current from 7812 are 12V and 1.5A. A heatsink is required in order to avoid the overheating of the voltage regulator. An alternative can be used instead of 7812 is a DC buck switching voltage regulator. It has performance issues with respect to battery life and power loss. It is used as a short circuit protection and thermal overload protection.

|  |
| --- |
| Lm7805-pinout-diagram-300x238 |

LM7805:  A voltage regulator IC maintains the output voltage at a constant value. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink. The input voltage range 7V- 35V, Current rating Ic =1A and the Output voltage range 5.2V-4.8V. This difference between the input and output voltage is released as heat. The greater the difference between the input and output voltage, more the heat generated. If the regulator does not have a heat sink to dissipate this heat, it can get destroyed and malfunction. Hence, it is advisable to limit the voltage to a maximum of 2-3 volts above the output voltage. Either design your circuit so that the input voltage going into the regulator is limited to 2-3 volts above the output-regulated voltage or place an appropriate heatsink that can efficiently dissipate heat.

**2.2.6. I2C**

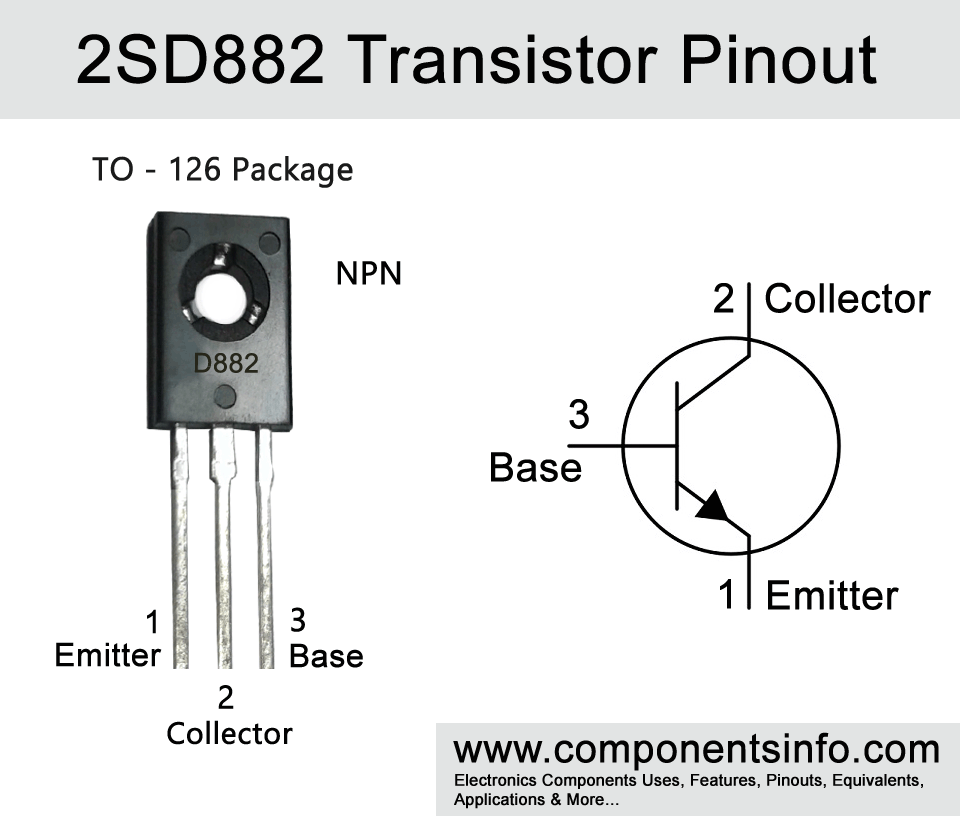
I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems.



I2C bus is popular because it is simple to use, there can be more than one master, only upper bus speed is defined and only two wires with pull-up resistors are needed to connect almost unlimited number of I2C devices. I2C can use even slower microcontrollers with general-purpose I/O pins since they only need to generate correct Start and Stop conditions in addition to functions for reading and writing a byte. The initial I2C specifications defined maximum clock frequency of 100 kHz. This was later increased to 400 kHz as Fast mode. There is also a High speed mode which can go up to 3.4 MHz and there is also a 5 MHz ultra-fast mode.Each slave device has a unique address. Transfer from and to master device is serial and it is split into 8-bit packets. All these simple requirements make it very simple to implement I2C interface even with cheap microcontrollers that have no special I2C hardware controller. You only need 2 free I/O pins and few simple i2C routines to send and receive commands.

**2.2.7. D882 AND BC547B**

D882 TRANSISTOR

FEATURES OF D882:

* Package type (TO-126)
* Transistor type (NPN)
* Maximum Collector current
* Maximum Collector Emitter voltage
* Maximum collector base voltage
* Maximum emitter base voltage
* Maximum transition dissipation(10watts)
* Maximum and minimum DC current gain
* Maximum storage and operating temperature

D882 TRANSISTOR DISCRIPTION

This transistor is a general purpose transistor and it is a quite high performance device, according to its features this transistor is an ideal to use in wide variety of educational, commercial and in electronic projects. D882 can be used for wide variety of switching and amplification purpose. It have maximum collector current is 3A which is enough to drive many relays, LEDs, bulbs, motors, etc. The minimum collector saturation voltage is only 0.3V.Moreover with its 10 watt output feature it can be used in the output stage of an audio amplifier or it can also be used separately as an amplifier.

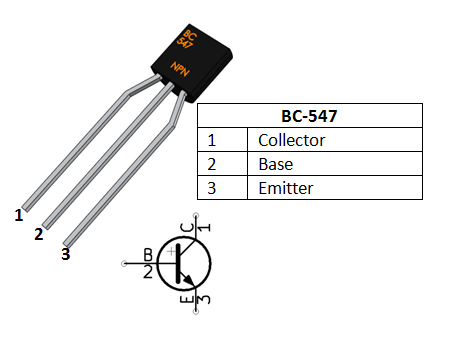
WHERE WE CAN USE IT AND HOW TO USE

D882 can be used in variety of different application. It can be used in power supply circuits, voltage regulator circuits, battery charger circuits, driving motors, switching any load under 3A. A part from the above applications it will also perform when used to amplify audio signals and it can be used in small audio amplifiers, output stage of receiver circuits, bell circuits, audio related circuits etc.

APPLICATIONS OF D882

* Relay driver
* Audio amplifier
* Switching loads under 3A
* Darlington pairs

**BC547B TRANSISTOR**



Collector - Current flows in through collector

Base - Controls the biasing of transistor

Emitter - Current drains out through emitter

### **BC547 Transistor Features**

* Bi-Polar NPN Transistor
* DC Current Gain is 800 maximum
* Continuous Collector current is 100mA
* Emitter Base Voltage is 6V
* Base Current is 5mA maximum
* Available in To-92 Package

### **Brief Description on BC547**

**BC547 is a NPN transistor** hence the collector and emitter will be left open (Reverse biased) when the base pin is held at ground and will be closed (Forward biased) when a signal is provided to base pin. BC547 has a gain value of 110 to 800, this value determines the amplification capacity of the transistor. The maximum amount of current that could flow through the Collector pin is 100mA, hence we cannot connect loads that consume more than 100mA using this transistor. To bias a transistor we have to supply current to base pin, this current (IB) should be limited to 5mA.

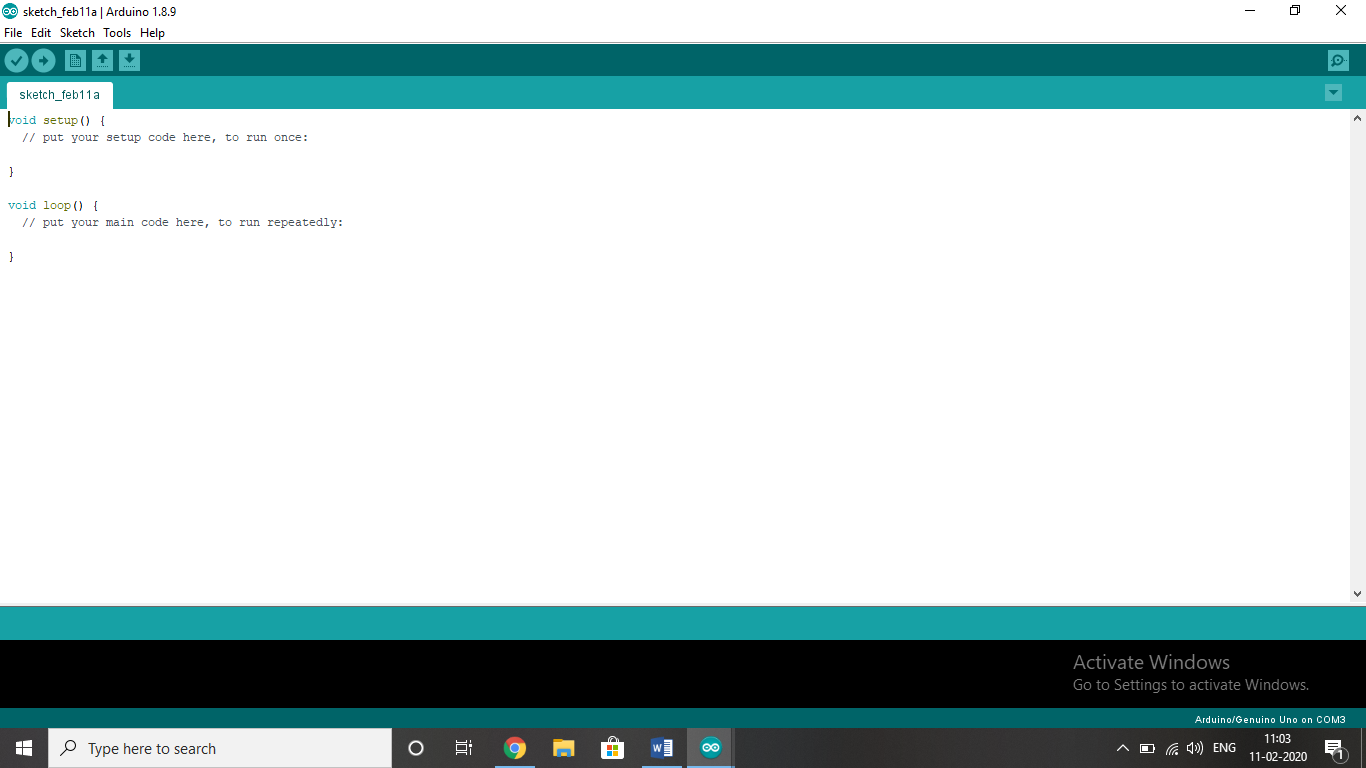
When this transistor is fully biased then it can allow a maximum of 100mA to flow across the collector and emitter. This stage is called **Saturation Region** and the typical voltage allowed across the Collector-Emitter (V­CE) or Base-Emitter (VBE) could be 200 and 900 mV respectively. When base current is removed the transistor becomes fully off, this stage is called as the **Cut-off Region**and the Base Emitter voltage could be around 660 mV.

**Applications**

* Driver Modules like Relay Driver, LED driver etc.
* Amplifier modules like Audio amplifiers, signal Amplifier etc.
* Darlington pair

**2.3. SOFTWARE REQUIREMENT**

**2.3.1. IDE ENVIRONMENT**

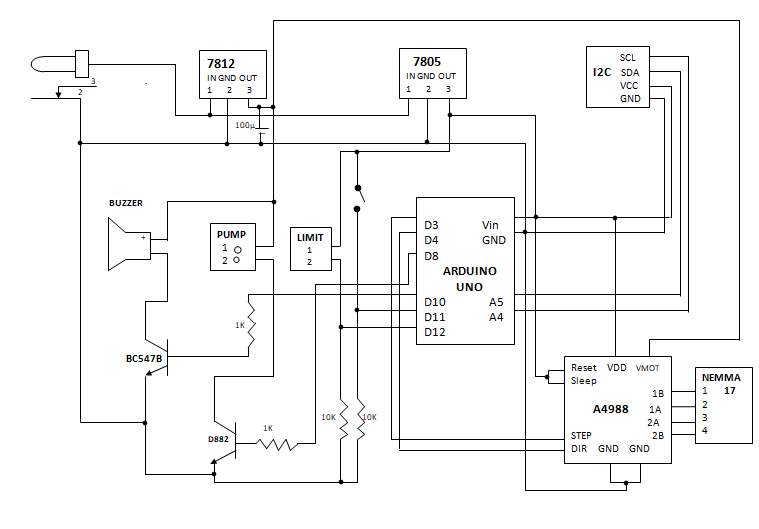


The Arduino integrated development environment (IDE) is a cross-platform application that is written in the programming language Java, C or C++. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

**CHAPTER 3**

**3.1. CIRCUIT DIAGRAM**

**3.2. WORKING**

The working of this system starts when the required voltage is supplied using a 15v DC source and a 5v source for the Arduino board. The LCD module initially displays ‘Press start’. Press the given switch. Then the filling process starts.

Each bottle is brought to the filling position by rotation initiated with the help of a stepper motor, small and large pulleys and belt. Each bottle gets filled to their appropriate levels one by one with water that is supplied with the help of a tube and a pump. Accordingly the number of bottles filled will be displayed on the LCD module. After all the bottles are filled, the process gets completed indicated with a buzzer sound.

**CHAPTER 4**

**4.1 CODE**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd=LiquidCrystal\_I2C(0x27,16,2);

const int steppin=3;

const int dirpin=4;

int buttonpin;

int r;

int buzzer=10;

int m=8;

void setup()

{

buttonpin=11;

r=12;

pinMode(buttonpin,INPUT\_PULLUP);

pinMode(r,INPUT);

lcd.init();

lcd.backlight();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Press start");

pinMode(steppin,OUTPUT);

pinMode(dirpin,OUTPUT);

pinMode(m,OUTPUT);

while(digitalRead(r)==LOW)

{

digitalWrite(dirpin,LOW);

for(int x=0;x<400;x++)

digitalWrite(steppin,HIGH);

delayMicroseconds(2000);

digitalWrite(steppin,LOW);

delayMicroseconds(2000);

if(digitalRead(r)==HIGH)

break;

}

}

void loop()

{

if(digitalRead(buttonpin)==HIGH)

{

tone(buzzer,800);

delay(100);

noTone(buzzer);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Start filling");

{

delay(500);

lcd.setCursor(2,1);

lcd.print("Bottle 1/5");

digitalWrite(m,HIGH);

delay(1850);

digitalWrite(m,LOW);

delay(1000);

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}

{

digitalWrite(dirpin,HIGH);

for(int x=0;x<100;x++){

digitalWrite(steppin,HIGH);

delayMicroseconds(2000);

digitalWrite(steppin,LOW);

delayMicroseconds(2000);

}

delay(500);

lcd.setCursor(2,1);

lcd.print("Bottle 2/5");

digitalWrite(m,HIGH);

delay(1850);

digitalWrite(m,LOW);

delay(1000);

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}

{

digitalWrite(dirpin,HIGH);

for(int x=0;x<100;x++){

digitalWrite(steppin,HIGH);

delayMicroseconds(2000);

digitalWrite(steppin,LOW);

delayMicroseconds(2000);

}

delay(500);

lcd.setCursor(2,1);

lcd.print("Bottle 3/5");

digitalWrite(m,HIGH);

delay(1850);

digitalWrite(m,LOW);

delay(1000);

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}

{

digitalWrite(dirpin,HIGH);

for(int x=0;x<100;x++){

digitalWrite(steppin,HIGH);

delayMicroseconds(2000);

digitalWrite(steppin,LOW);

delayMicroseconds(2000);

}

delay(500);

lcd.setCursor(2,1);

lcd.print("Bottle 4/5");

digitalWrite(m,HIGH);

delay(1850);

digitalWrite(m,LOW);

delay(1000);

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}

{

digitalWrite(dirpin,HIGH);

for(int x=0;x<100;x++){

digitalWrite(steppin,HIGH);

delayMicroseconds(2000);

digitalWrite(steppin,LOW);

delayMicroseconds(2000);

}

delay(500);

lcd.setCursor(2,1);

lcd.print("Bottle 5/5");

digitalWrite(m,HIGH);

delay(1850);

digitalWrite(m,LOW);

delay(1000);

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}

}

**CHAPTER 5**

**5.1. ADVANTAGES**

* High reliability
* Small space required
* Computer capabilities
* Expandability
* High power handling
* Reduced human effort
* User friendly

**5.2. LIMITATIONS**

* Circuit is complex
* Unemployment problem

**5.3. FUTURE SCOPE**

Here in this project, by installation of 5 nozzles, we can fill 5 bottles at a time which reduces the time to fill bottles and can efficiently increase productivity but still some manual interference is required for loading and unloading of the pallets from conveyer belt. This manual interference can be reduced by introduction of the Robotic Arm which can be programmed for loading and unloading operations. Also we can design this device for more than 5 bottles.

**CHAPTER 6**

**6.1. REFERENCES**

* <https://components101.com/modules/a4988-stepper-motor-driver-module>
* <https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>
* <https://components101.com/motors/nema17-stepper-motor>
* <https://www.electronicsforu.com/resources/learn-electronics/7805-ic-voltage-regulator>
* <https://learn.sparkfun.com/tutorials/i2c/all>