WEST BENGAL STATE UNIVERSITY

AUTOMATED SOFT DRINK DISPENSER

(USING ARDUINO)
PROJECT DOCUMENTATION

BSC. PART III (H) – CMSA (2018 – 2019)

SUBHRAJIT KARMAKAR	**REGISTRATION NUMBER – 1011611300098 OF 2016**
DIPAYAN PAL	**REGISTRATION NUMBER – 1011611400100 OF 2016**
ROUNAK SAHA	**REGISTRATION NUMBER – 1011611400124 OF 2016**

GUIDED BY
PROFESSOR PRADYUT SAHA
APC COLLEGE
NEW BARRACKPORE, NORTH 24 PARGANAS, KOLKATA – 700131

INDEX

AUTOMATED SOFT DRINK DISPENSER

DONE BY
SUBHRAJIT KARMAKAR
DIPAYAN PAL
ROUNAK SAHA

ACHARYA PRAFULLA CHANDRA COLLEGE

(GOVT. SPONSORED)
New Barrackpur, North 24 Parganas, Kolkata – 700131
Phone : 033-2537-3297 Email : apc1960@apccollege.ac.in Website : www.apccollege.ac.in

CERTIFICATE OF ORIGINALITY

This is to certify that the Project report entitled **Automated Soft drink Dispenser** submitted to **Acharya Prafulla Chandra College** in partial fulfillment of the requirements for the award of the degree of B.Sc., is an authentic and original work carried out under my guidance by the following students with University Registration Number and Roll Numbers.

Name	University Registration Number	Roll Number
Subhrajit Karmakar	1011611300098 OF 2016	319111308067
Dipayan Pal	1011611400100 OF 2016	319111308075
Rounak Saha	1011611400124 OF 2016	319111308070

The matter embodied in this project is genuine work done by the student and has not submitted whether to this Institute or to any other University /Institute for the fulfilment of the requirements of any course of study.
This project is here by approved as accreditable performance for a group of Final Year BSc students. It is carried out and presented in a manner will satisfactory to warrant its acceptance as a prerequisite to the BSc for which it has been submitted. It is understood that by approval of this project Report the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the Project Report for the purpose it has been submitted.

NAME AND SIGNATURE OF THE STUDENT WITH DATE

NAME : SUBHRAJIT KARMAKAR **SIGNATURE :**

NAME : DIPAYAN PAL **SIGNATURE :**

NAME : ROUNAK SAHA **SIGNATURE :**

SIGNATURE OF THE GUIDE WITH DATE

PROFESSOR - PRADYUT SAHA
(PROJECT GUIDE)
(DEPARTMENT OF CMSA)

ACHARYA PRAFULLA CHANDRA COLLEGE

(GOVT. SPONSORED)
New Barrackpur, North 24 Parganas, Kolkata – 700131
Phone : 033-2537-3297 Email : apc1960@apccollege.ac.in Website : www.apccollege.ac.in

CERTIFICATE OF APPROVAL

This is to certify that SUBHRAJIT KARMAKAR, DIPAYAN PAL, ROUNAK SAHA are the students of Acharya Prafulla Chandra College, have successfully completed a project on "Automated Soft Drink Dispenser" in PART-III final examination at Department of Computer Science.

This report has not been submitted to any other Organization and does not form part of any Course undergone by then, for the award of Master of Computer Science.

SIGNATURE OF THE HOD WITH DATE

PROFESSOR JOYDEB DAS BISWAS
HEAD OF THE DEPARTMENT
DEPARTMENT OF CMSA

ACKNOWLEDGEMENT

It is our privilege to express our sincerest regards to our **Project Coordinator, Professor Pradyut Saha**, for his valuable inputs, guidance, encouragement, whole-hearted co-operation and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our **Head of the Department, Professor Joydeb Das Biswas** for encouraging and allowing us to present the project on the topic "**Automated Soft Drink Dispenser**" at our Department premises for the partial fulfilment of the requirements leading to the award of B.Sc.

We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout the career.

Last but not the least, we express our thanks to our friends for their co-operation and support.

NAME AND SIGNATURE OF THE STUDENT WITH DATE

NAME : SUBHRAJIT KARMAKAR **SIGNATURE :**

NAME : DIPAYAN PAL **SIGNATURE :**

NAME : ROUNAK SAHA **SIGNATURE :**

PROBLEM STATEMENT

Automated Soft drink Dispenser allows the user to select one out of the three options (small / medium / large) for the amount of liquid as desired.

INTRODUCTION

Soft Drink dispensers are the devices that are designed to serve soft drinks. Dispensers store soft drink and oxygen is prevented from entering the bottle when pouring so it preserve drink for long time.

It also reduces the error of spilling soft drink while pouring. The dispensers can be operated only by a nozzle, some by taps, and others by automatic dispensation in response to the button of size.

OBJECTIVE

1. The device will wait for the user to press KEY to activate.
2. KEY “SMALL” will pour 30ml of Soft drink.
3. KEY “MEDIUM” will pour 60ml of Soft drink.
4. KEY “LARGE” will pour 90ml of Soft drink.
5. The device will display according to the KEY pressed
6. The display will be very informative to display when not in use.

ROLE OF ARDUINO

The Arduino microcontroller constantly monitors the switches and looks for the moment when any of the switches is pressed. Then the microcontroller inputs the data and detects which switch is pressed. After it gets to know which switch is pressed, it just will put the air pump on high state for corresponding time.

MEMBERS ROLES

- SUBHRAJIT KARMAKAR
- DIPAYAN PAL
- ROUNAK SAHA

SERIAL NO	TOPIC	NAME OF GROUP MEMBERS
1.	**PROJECT DISCUSSION AND ANALYSIS**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA
2.	**CIRCUIT DESIGN**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA
3.	**HARDWARE IMPLEMENTATION**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA
4.	**CODING**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA
5.	**TESTING**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA
6.	**MODEL MAKING**	• SUBHRAJIT KARMAKAR • DIPAYAN PAL • ROUNAK SAHA

HARDWARE REQUIREMENTS

1. ARDUINO NANO
2. USB ADAPTER FOR ARDUINO NANO
3. MALE TO FEMALE JUMPER WIRE – 10
4. MALE TO MALE JUMPER WIRE – 10
5. TECTILE BUTTON SWITCH – 3
6. AIR PUMP-1
7. I2C OLED DISPLAY 0.96 INCH 128x64
8. PIPES
9. BOTTLE
10. RESISTORS

SOFTWARE REQUIREMENTS

1. ARDUINO IDE FOR CODING
2. CIRCUIT DIAGRAM FOR CIRCUIT DRAWING

MISCELLANEOUS REQUIREMENTS

A. HARDWARE TOOLS:

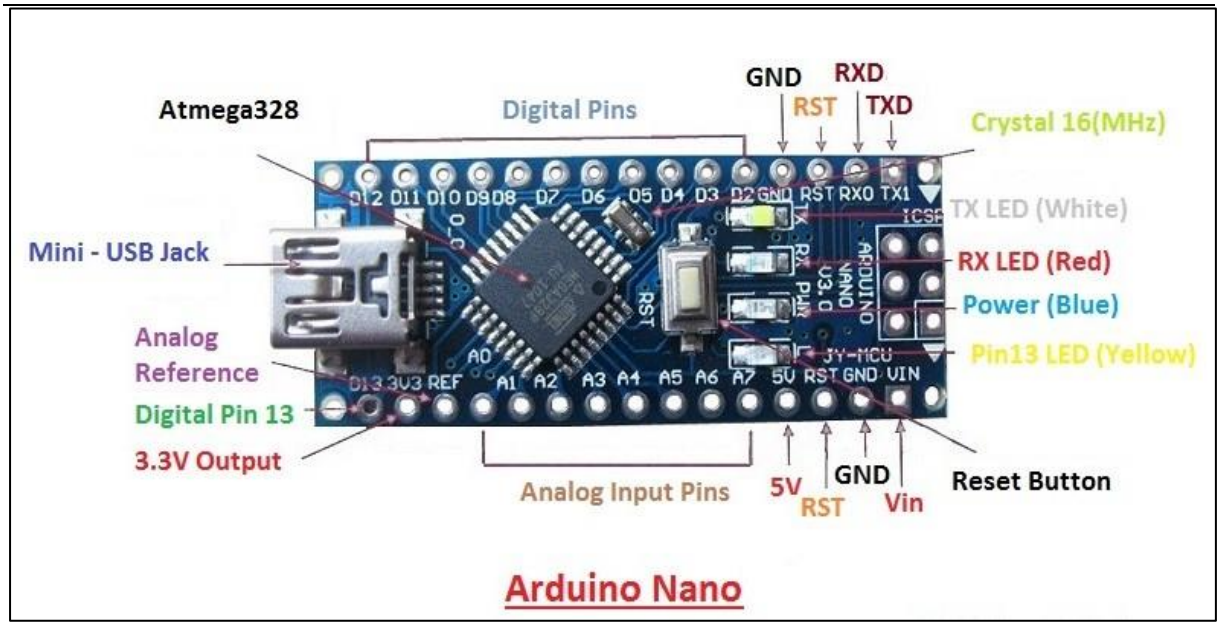
a. SOLDERING IRON
b. SOLDER
c. WIRES
d. WIRECUTTERS
e. BREADBOARD

B. MODELLING TOOLS:

a. CARDBOARD
b. SCISSORS
c. GUM
d. CELLOTAPE
e. RULER
f. PENCILS AND ERASERS
g. CUTTERS

ARCHITECTURE

1. ARDUINO NANO



Arduino Nano is a microcontroller based on Atmel ATmega328 which is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture.
By executing a powerful single clock cycle, the ATmega328 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source

Memory

The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 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 FTDI USB-to-TTL Serial chip.
- External Interrupts: 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. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

- I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication datasheet.

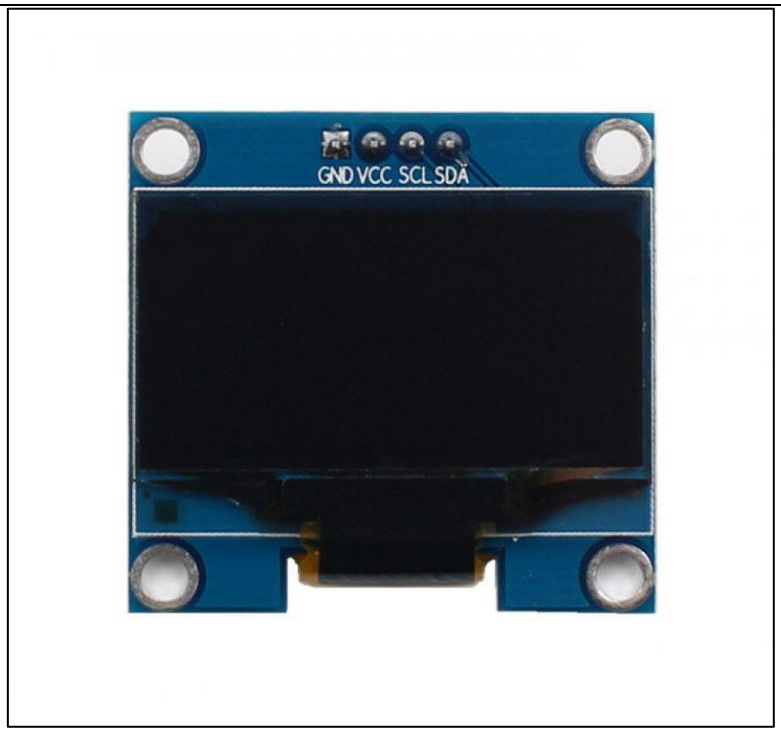
Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

PIN DESCRIPTION:

Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

2. I2C OLED DISPLAY



0.96 inch OLD display module can be interfaced with any microcontroller using IIC protocols. The package include display board, display, 4 pin male header presoldered to board. OLED(Organic Light-Emitting Diode) is a self light -emitting technology composed of a thin, multi-layered organic film placed between an anode and cathode. In contrast to LCD technology, OLED does not require a backlight.

OLEDs basic structure consists of organic materials positioned between the cathode and the anode, which is composed of electric conductive transparent Indium Tin Oxide (ITO). The organic materials compose a multi-layered thin film, which includes the Hole Transporting Layer (HTL), Emission Layer (EML) and the Electron Transporting Layer (ETL). By applying appropriate voltage, holes and electrons are injected into the EML from the anode and the cathode respectively. The holes and electrons combine inside the EML to form excitons, after which electro luminescence occurs. The transfer material, emission layer material and choice of electrode are the key factors that determine the quality of OLED components.

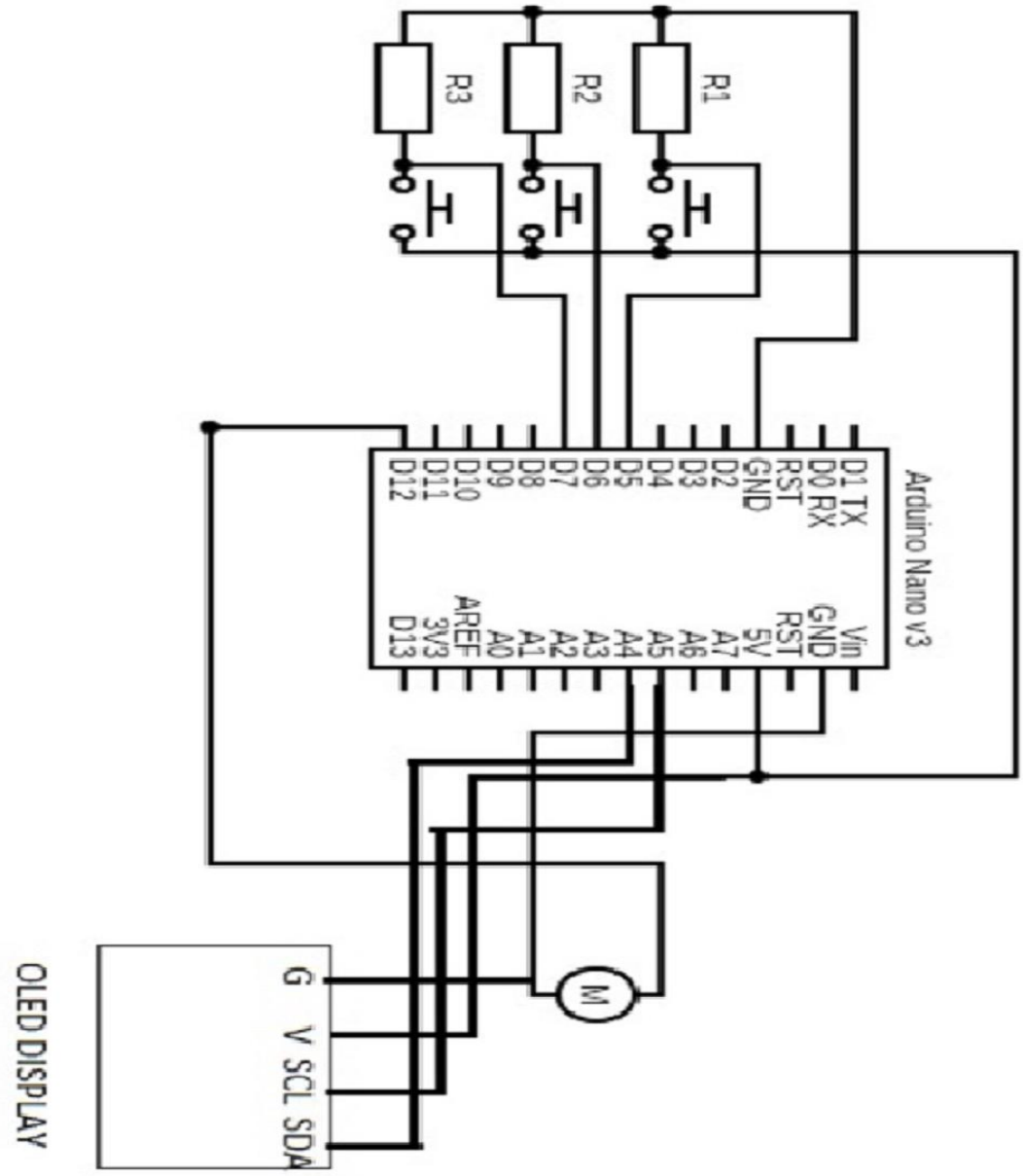
FEATURES OF 0.96 INCH OLED DISPLAY MODULE

- **Viewing angle** : greater than 160 degrees
- **Resolution:** 128 x 64
- Full Compatible with Arduino
- **Low power consumption :** 0.04W during normal operation
- **Support wide voltage** : 3.3V-5V DC
- **Driver IC :** SSD1306
- **Communication :** IIC, only two I / O ports
- **Backlight :** OLED self light, no backlight
- **Interface:** VCC: 3.3-5V
  GND: Ground
  SCL: Serial Clock
  SDA: Serial Data

3. AIR PUMP

Air pump is a device that is used for pumping the air using power supply provided to it. The two of its terminals are used in supplying power and it has one end which is connected to pipe that creates air suction, thus making vacuum.
Working voltage 5V DC and current ~0.25A.

CIRCUIT DIAGRAM



HARDWARE IMPLEMENTATION

Implementation of soft drink dispenser based on Arduino nano, motor and switches. Here we control the Arduino using switches. There are three switches one for small amount of soft drink, another two for medium and large amount of drinks respectively. Switches are connected with ground, vcc and digital input/output pins of Arduino. The air pump is connected to output pin of nano and to ground connection. Air pump's nozzle is connected to bottle with a pipe and another pipe is also put inside the bottle to get drinks from it. We use a oled display to show the amount of drink is dispensed. Oled display is connected to ground, vcc and dedicated analog input/output pins of Arduino. When a switch is pressed the dedicated pin of Arduino gets HIGH voltage and due to appropriate coding it turn the output pin of Arduino nano HIGH 5V. Air pump get power and then it starts pulling water from bottle and water starts pouring from another end of the pipe. In the meanwhile, oled display is also turned on and display the amount of selected drink. After few moments the pouring of drink is accomplished and all the digital input/output pins are at LOW voltage. When we press a switch again the process will start. We use male to male and male to female wires for better reliability.

CODING

```
/* THIS IS THE CODE FOR SOFTDRINK DISPENSOR USING ARDUINO NANO
 *  WHERE SWITCHES ARE THE PINS SPIN(SMALL PIN), MPIN(MEDIUM PIN) AND LPIN(LARGE PIN)
 *  IT WILL CONROL THE PUMP'S RELAY FOR A DELAY WHERE PPIN IS FOR THE PUMP'S RELAY PIN PIN
 *  THE RELAY IS CONNECTED WITH AN AIR PUMP WHICH WILL BE USED TO POUR THE SOFT DRINK
 *  AND ALL THE PROCESSINGS AND VISUALS ARE SHOWN IN OLED I2C 128X64
 *  FOR THIS OLED I2C DISPLAY WE NEED TO INCLUDE HEADER FILES AND DISPLAY DECLARATION AND STARTING DISPLAY WITH AN ADDRESS 0x3C
 *  MATERIALS REQUIRED:- ARDUINO NANO(1 PC), BREADBOARD(1 PC), TACTILE SWITCH(3 PC), RELAY SPDT(1 PC), AIR PUMP(1 PC), PIPES(2 PC), OLED I2C 128X64 DISPLAY(1 PC), WIRES AND JUMPERS
 */
//INITIALIZATION OF REQUIRED HEADER FILES
#include<SPI.h>
#include<Wire.h>
#include<Adafruit_GFX.h>
#include<Adafruit_SSD1306.h>
//END OF REQUIRED HEADER FILES

//INITIALIZATION OF DISPLAY DECLARATION
Adafruit_SSD1306 display(-1);
//END OF DISPLAY DECLARATION

//INITIALIZATION OF GLOBAL VARIABLES
int pPin=12;   //PUMP'S RELAY PIN
int sPin=5;    //SMALL SWITCH
int mPin=6;    //MEDIUM SWITCH
int lPin=7;    //LARGE SWITCH
//END OF GLOBAL VARIABLES

//FUNCTION CALLS
void pausingOtherPins()      //THIS FUNCTION WILL PAUSE THE OTHER PINS WHILE PUMP'S RELAY IS ON
{
  digitalWrite(sPin, LOW);  //SETTING SMALL PIN DEACTIVATED TILL THE PROCESS IS DONE
  digitalWrite(mPin,LOW);   //SETTING MEDIUM PIN DEACTIVATED TILL THE PROCESS IS DONE
  digitalWrite(lPin,LOW);   //SETTING LARGE PIN DEACTIVATED TILL THE PROCESS IS DONE
  digitalWrite(pPin,HIGH);  //ACTIVATING PUMP
}

void showDisplay(char str[], int s, int clp, int cwp) //THIS FUNCTION WILL DISPLAY THE TEXT WHEN THE SOFTDRINK IS POURED
/*str[] IS THE STRING TO BE DISPLAYED
 * s IS THE TEXT SIZE
 * clp IS THE CURSOR'S LENGTH POSITION
 * cwp IS THE CURSOR'S WIDTH POSITION
 */
{
 display.clearDisplay();                  //CLEARING DISPLAY SCREEN
 display.setTextSize(s);                  //SETTING TEXT SIZE ACCORDING TO THE SIZE PROVIDED DURING CALL
 display.setTextColor(WHITE);             //SETTING TEXT COLOR
```

```
  display.setCursor(clp,cwp);                  //SETTING CURSOR POSITION AS PER PROVIDED
  display.print(str);                          //DISPLAYING THE STRING PROVIDED
  display.display();                           //CALLED FOR DISPLAYING IN OLED
}

void removeDisplay()
{
  display.clearDisplay();             //CLEARING DISPLAY SCREEN
  display.setTextSize(2);             //SETTING TEXT SIZE ACCORDING TO THE SIZE 2
  display.setTextColor(WHITE);        //SETTING TEXT COLOR
  display.setCursor(0,2);             //SETTING CURSOR POSITION AT (0,2)
  display.print("REMOVE THE GLASS");   //DISPLAYING THE "REMOVE THE GLASS" STRING
  display.display();                  //CALLED FOR DISPLAYING IN OLED
  display.startscrollleft(0x00,0x0F);  //CALLED FOR SCROLLING THE TEXT LEFT TO RIGHT
  delay(8000);                        //SETTING THE DELAY FOR 8 SECONDS FOR REMOVING GLASS FROM THE
DISPENSOR
  display.stopscroll();               //STOPPING THE SCROLLING THE TEXT
}

void nonWorkingDisplay()
{
  display.clearDisplay();             //CLEARING DISPLAY SCREEN
  display.setTextSize(2.6);           //SETTING TEXT SIZE ACCORDING TO THE SIZE 2.6
  display.setTextColor(WHITE);        //SETTING TEXT COLOR
  display.setCursor(10,10);           //SETTING CURSOR POSITION AT (10,10)
  display.print("WELCOME");           //DISPLAYING THE "WELCOME" STRING
  display.display();                  //CALLED FOR DISPLAYING IN OLED
 }
//END OF FUNCTION CALLS

//STARTING OF MAIN SETUP
void setup()
{
  pinMode(pPin, OUTPUT);              //PUMP IS IN THE OUTPUT MODE
  pinMode(sPin, INPUT);               //SMALL SWITCH IS IN THE INPUT MODE
  pinMode(mPin, INPUT);               //MEDIUM SWITCH IS IN THE INPUT MODE
  pinMode(lPin, INPUT);               //LARGE SWITCH IS IN THE INPUT MODE
  display.begin(SSD1306_SWITCHCAPVCC,0x3C); //STARTING DISPLAYING WITH STARTING ADDRESS
0x3C
  display.clearDisplay();             //CLEARING THE DISPLAY
  display.display();                  //CALLED FOR DISPLAYING TEXT IN OLED
}
//END OF MAIN SETUP

//STARTING OF MAIN LOOP
void loop()
{
  nonWorkingDisplay();            //DISPLAYING "WELCOME" WHEN NOT IN USE
  while(digitalRead(sPin)==HIGH)   //CHECKING IF SMALL SWITCH IS PRESSED
  {
    char sstring[] = "SMALL";      //WANT TO DISPLAY THAT "SMALL" IS PRESSED
```

```
   showDisplay(sstring,3.4,23,10); //WILL DISPLAY "SMALL" IN TEXTSIZE = 3.4 AT CURSOR POSITION
(23,10)
   pausingOtherPins();          //WILL CALL TO DISABLE OTHER PINS AND ENABLE PUMP'S RELAY
   delay(10000);               //DELAYING THE PUMP FOR 10 SEC
   digitalWrite(pPin, LOW);      //DEACTIVATING THE PUMP
   removeDisplay();          //WILL DISPLAY "REMOVE THE GLASS" AFTER THE WORK OF PUMP'S RELAY
IS DONE
  }
  while(digitalRead(mPin)==HIGH)   //CHECKING IF MEDIUM SWITCH IS PRESSED
  {
   char sstring[]="MEDIUM";      //WANT TO DISPLAY THAT "MEDIUM" IS PRESSED
   showDisplay(sstring,3,15,10);   //WILL DISPLAY "MEDIUM" IN TEXTSIZE = 3 AT CURSOR POSITION
(15,10)
   pausingOtherPins();          //WILL CALL TO DISABLE OTHER PINS AND ENABLE PUMP'S RELAY
   delay(20000);               //DELAYING THE PUMP FOR 20 SEC
   digitalWrite(pPin, LOW);      //DEACTIVATING THE PUMP.
   removeDisplay();          //WILL DISPLAY "REMOVE THE GLASS" AFTER THE WORK OF PUMP'S RELAY
IS DONE
  }
  while(digitalRead(lPin)==HIGH)   //CHECKING IF LARGE SWITCH IS PRESSED
  {
   char sstring[]="LARGE";       //WANT TO DISPLAY THAT "LARGE" IS PRESSED
   showDisplay(sstring,3.4,23,10); //WILL DISPLAY "LARGE" IN TEXTSIZE = 3.4 AT CURSOR POSITION
(23,10)
   pausingOtherPins();          //WILL CALL TO DISABLE OTHER PINS AND ENABLE PUMP'S RELAY
   delay(30000);               //DELAYING THE PUMP FOR 30 SEC
   digitalWrite(pPin, LOW);      //DEACTIVATING THE PUMP
   removeDisplay();          //WILL DISPLAY "REMOVE THE GLASS" AFTER THE WORK OF PUMP'S RELAY
IS DONE
  }
}
//END OF MAIN LOOP
```

TESTING

In testing, we have successfully discovered our faults and we took certain measures to avoid it.

The code itself is in the 6th stage because we removed error in each stage.

We checked for:-

a. Code validation.

b. Unit testing for each set.

- At first stage of code we test switch and relay connection.
- In second stage we test oled display connection.
- Next, we test air ump connection.

c. Integrated testing.

- Finally, we integrate all our component and test it.

d. Re-modification according to any failure in above methods.

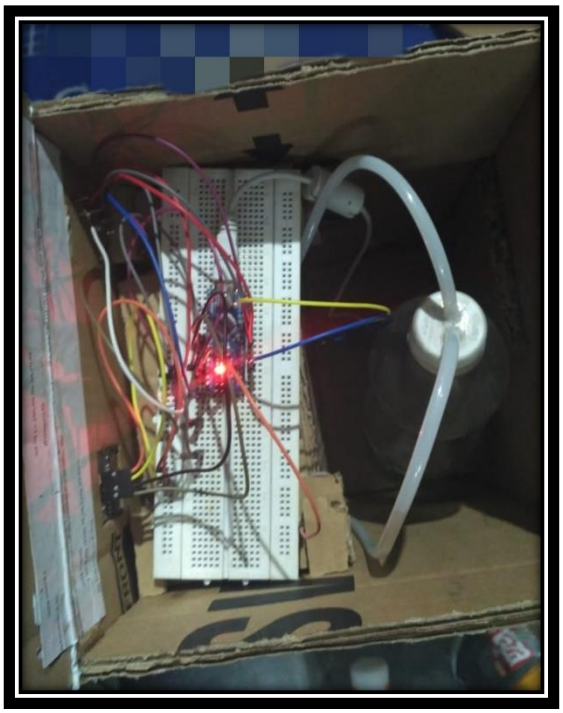
e. Removed redundancy and followed from start.

Here contain some glimpses of our Testing phases.

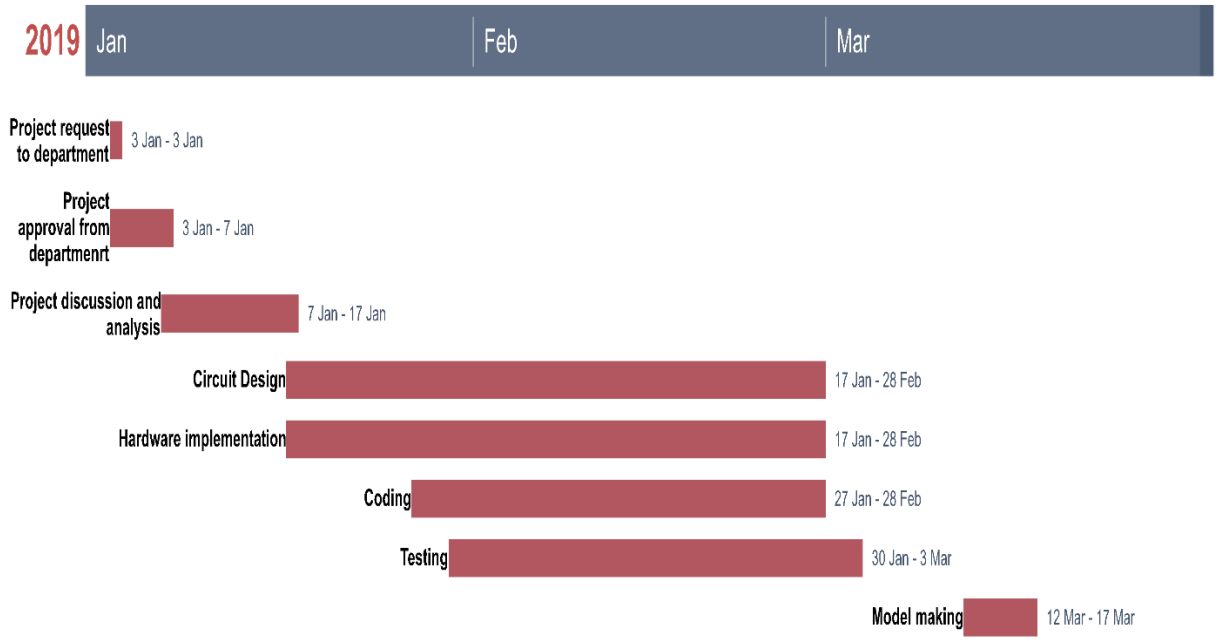
MODEL MAKING

In model making, we used measurable size of cardboard to make it like a real dispenser. We make chamber behind to keep the bottle. We arranged switches and display for better usability and user-friendly manner.

Here contains some of the model pictures.



GANTT CHART


2019
Jan
Feb
Mar
Project request to department
3 Jan - 3 Jan
Project approval from departmenrt
3 Jan - 7 Jan
Project discussion and analysis
7 Jan - 17 Jan
Circuit Design
17 Jan - 28 Feb
Hardware implementation
17 Jan - 28 Feb
Coding
27 Jan - 28 Feb
Testing
30 Jan - 3 Mar
Model making
12 Mar - 17 Mar

DISCUSSIONS

1. In first in our code, we tried to implement the code so that to check the push buttons working or not so that we can adjust the code in future according to the project.

2. We implemented full code but checked that we're mal-utilizing the space.
So, we removed Redundancies as much possible in second version.

3. We checked in future trials with modified push button codes that it was adding more time to it because we're not able to control it.
For eg. – if we press small and while pouring, mistakenly pressed large/medium just adds more time. We cannot stop the control by not accepting orders while it's in use.
So, we blocked interrupts coming to work and making it more responsive and accurate.

4. At this point, the dispenser is working exactly what we wanted. So, at next version we implemented i2c code which was bit complicated for us because we couldn't link the header files and the functions of the display.
It took a lot of time to figure out our fault that we didn't initialize the I2C display with correct information.
5. We have successfully implemented I2C code working simultaneously with the Relay and Pump but here again, we discovered a lot of space was used for I2C implementation.
So, we removed as much redundancy as possible to make it cleaner and better understandability.

6. The dispenser is working as fine as we wanted. But the display readability was not that great.
So, we included scrollable text.

7. Also, the display didn't show anything useful when it's idle/ not working.
So, we include "WELCOME" in the display when the dispenser is idle.

8. We faced the next issue in model making phase.
We had to open all the circuits so that we can make the model. But in that process, we observed Relay has got damaged. Switches got damaged while placing in the board.
So, we had to buy another Relay and switches to make it normal.

FUTURE SCOPES

1. **Making available by wireless Connectivity (IoT) –** The soft drink dispenser can be used by including IoT. For eg, wireless dispenser (Bluetooth, Wi-Fi), online payment gateway for dispensing.
2. **Coin vending Soft drink Dispenser** – The user needs to pay required amount of coin for dispensing soft drink.
3. **Multiple Soft Drink Dispenser in one** – The user can have option to choose any from multiple soft drinks.
4. **Soft drink mixer machine** – Some users like mixing of two or more soft drinks into one drink.
5. **Beverage Dispenser –** This whole project can be used for making beverage dispensers.
    a. **Tea dispenser.**
    b. **Coffee dispenser.**
    c. **Hot/ Cold beverage dispenser.**

And many more…

CREDITS

1. To our guide.
2. To our HOD for approving our project.
3. To all Computer Science teachers of our department whose invisible helps throughout our course helped a lot.
4. To the shopkeeper to provide us genuine products.
5. To our families to allow us to make project in our homes.
6. To our friends whose contributions make it success.
7. To various sites who provided us some valuable information regarding minute details required for the project.
    a. www.google.co.in
    b. https://www.arduino.cc/
    c. www.wikipedia.org
    d. www.circuit-diagram.org
    e. www.tinkercad.com
    f. https://components101.com
    g. https://startingelectronics.org/tutorials/arduino/modules/OLED-128x64-I2C-display/