EMBEDDED SYSTEM DESIGN PROJECT WORK

“INDUSTRIAL AUTOMATION”
Group - 09

Members:
Aanshi Patwari (AU1841004)
Dipika Pawar (AU1841052)
Miracle Rindani (AU1841017)
Bhumiti Gohel (AU1841051)

Guided By: Prof. Anurag Lakhlani

Motivation

In the era of the increasing use of technology and automation in every sector, the industries still do labour work for the production and the manufacturing process which is very tedious and repetitive. So, in order to reduce the amount of labour work we came with the idea of automation in the industrial sector especially in the field of food processing units. Also, the food industry needs to take care of certain levels of hygiene while manufacturing and packing the food items as they can directly affect the health of the common public; so if the packing process is automated then there will be fewer issues of unhygienic or improper packaging.

Description

The labourers will be given ids through which they can check in to the industry and turn on the lighting system. By this id the labourer can switch on the machine which will, in turn, start working. As we will be focusing on the food processing and in that we will be preparing the bottle filling industrial model.

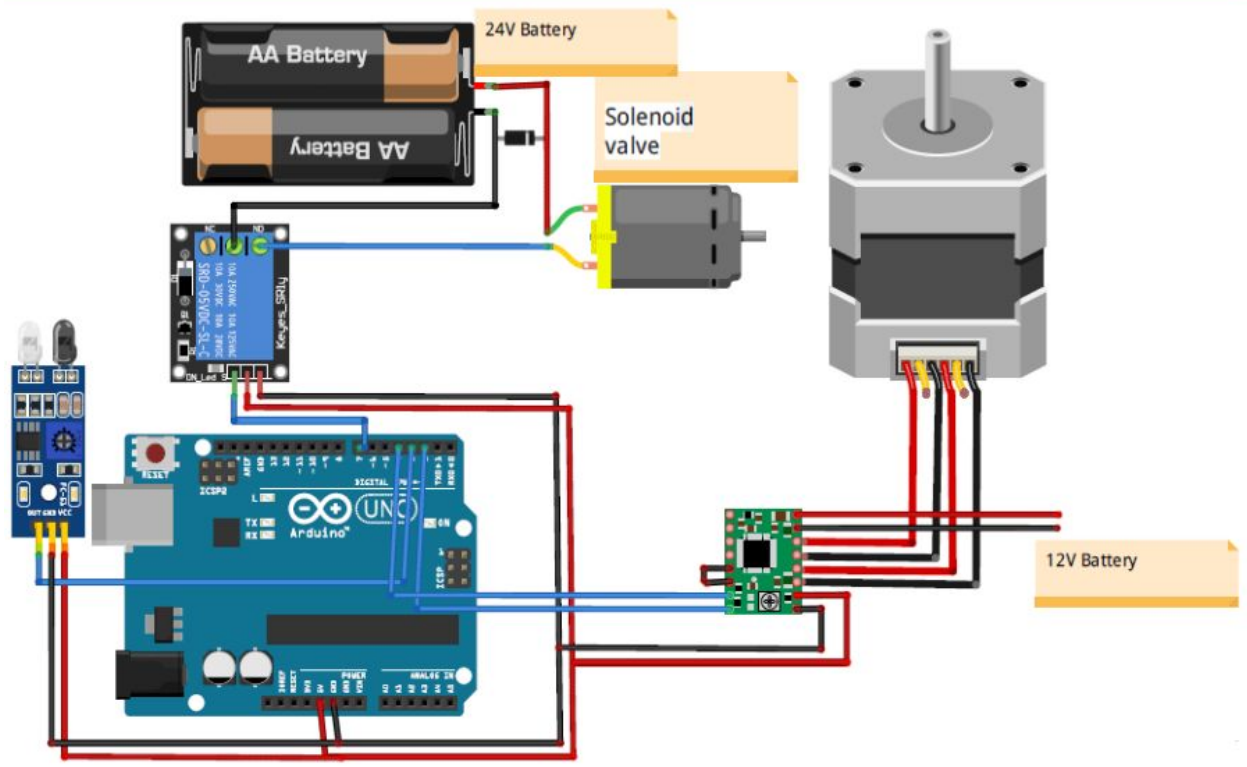
There will be a motor for the roller. On the detection of the bottle moving on the roller by the IR sensor, the roller will stop for a fixed time and the bottle will get filled up and then the roller will start rolling again. The process will then continue until the end of the day is reached.

Final Outcome

At the end of the day, we get the bottles filled up and packed up at the end of the roller and also there will be a display of the count of the number of bottles which have been filled up completely. Also, we will be displaying if there comes out to be a fault or an interrupt during the ongoing process.

The number of bottles filled and packed in the day will be sent to the owner of the industry as a message on his/her cellphone via the GSM module.

Block Diagram

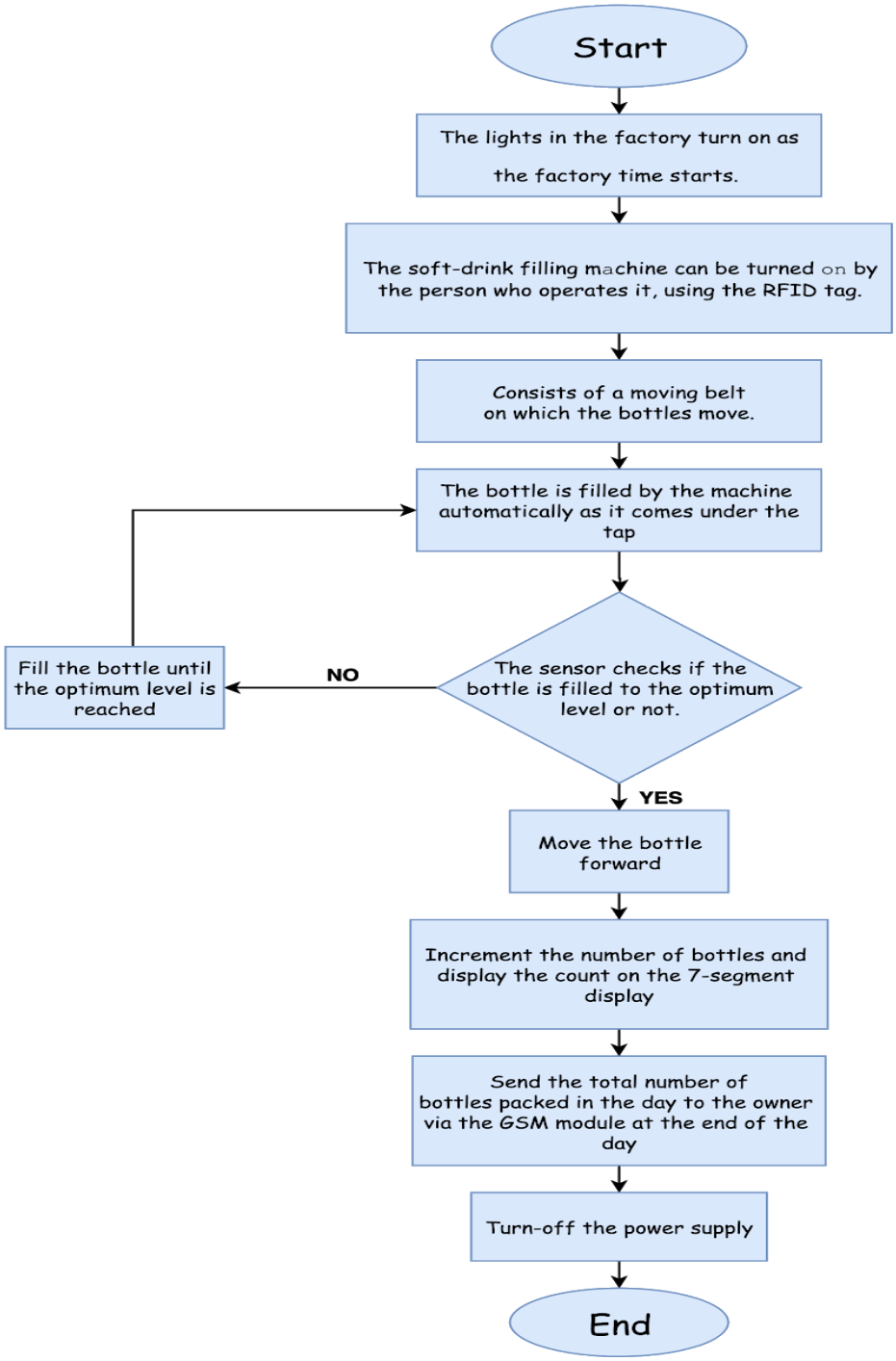


Components required

The components mainly required are:

- Arduino Mega - 1
- LED - 15
- RFID module (RC522 433Hz) - 2
- Jumper wires - 2 sets of each type
- Push-Buttons - 3
- Dc motor - 2
- Relay module - 1
- A4988 Motor driver - 1
- Solenoid valve - 2
- IR sensor - 1
- LCD (2 X 16) - 1
- Ultrasonic sensor(HC - SR04) - 1
- Buzzer - 2
- Water pump - 1
- Power Bank - 1
- Potentiometer - 2
- GSM module - 1

FlowChart of Program



Selection Criteria For Components

1. **Arduino Mega**
    a. We can use the Arduino MEGA for instead of Arduino UNO, as it does not have enough pins for a large project to be implemented on.
    b. It can help in decreasing the use of the number of wires used for the connection.
    c. Also, the MEGA consists of Tx and Rx pins which can be used for serial communication.

2. **RFID module (RC522 433Hz)**
    a. The use is for security purposes.
    b. The tag can be used for the identification of the person.
    c. If a person has the RFID tag then the user is considered as identified or else not.

3. **Push Buttons**
    a. The use of switches is for manually turning on/off the system after the user gets login to the system through the RFID tag.

4. **DC motor**
    a. A DC motor is used for the roller purpose.
    b. The roller can be moved using the DC motor.

5. **Relay module**
    a. It is used for moving the conveyor belt on which the water bottles will be moving.

6. **A4988 Motor driver**
    a. The motor is used for driving the entire roller module.

7. **Solenoid valve**
    a. The fetching of water is done with the help of solenoid valves along with the water pump.

8. **IR sensor**
    a. It is used for detecting the bottle coming on the roller.
    b. The roller gets stopped when a bottle is detected by the IR sensor.

9. LCD (2 X 16)

a. The LCD displays the number of bottles currently filled and cumulatively at the end of the day.

10. Ultrasonic sensor(HC - SR04)

a. The sensor is used for detecting the water level filled inside the bottle.
b. After that, the bottle gets fixed upto a certain extent the roller gets to start to move on.

11. Buzzer

a. It is used when there is an entry to a user without the RFID tag
b. It is also used when a bottle gets filled up to a fixed level so that the user can know that the bottle is filled up.

12. Water pump

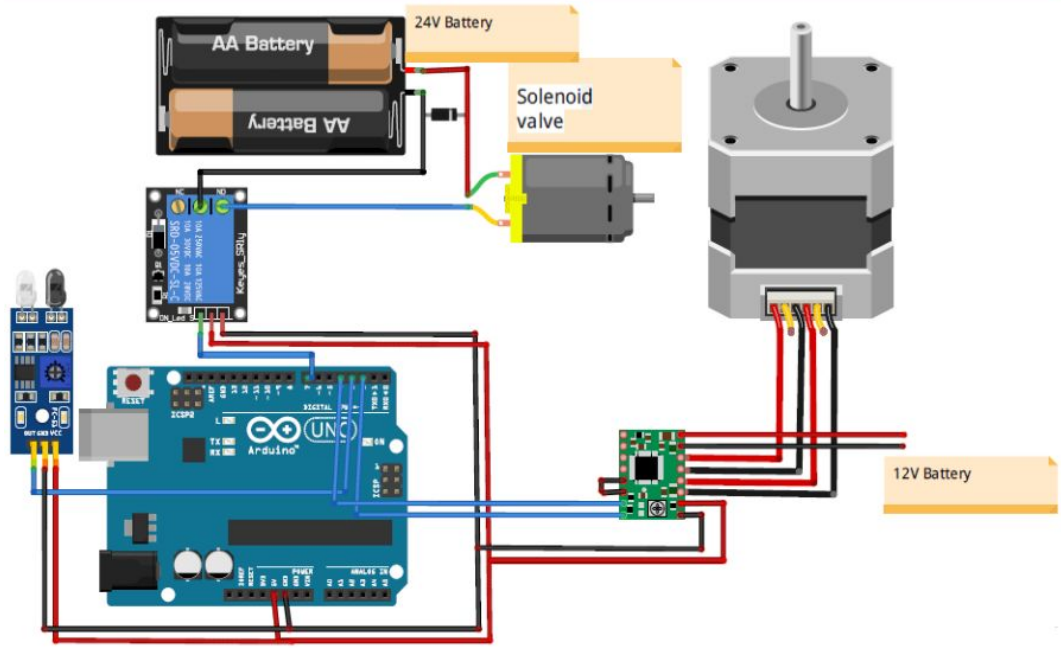
a. It is used for fetching the water from the main container.

13. GSM module

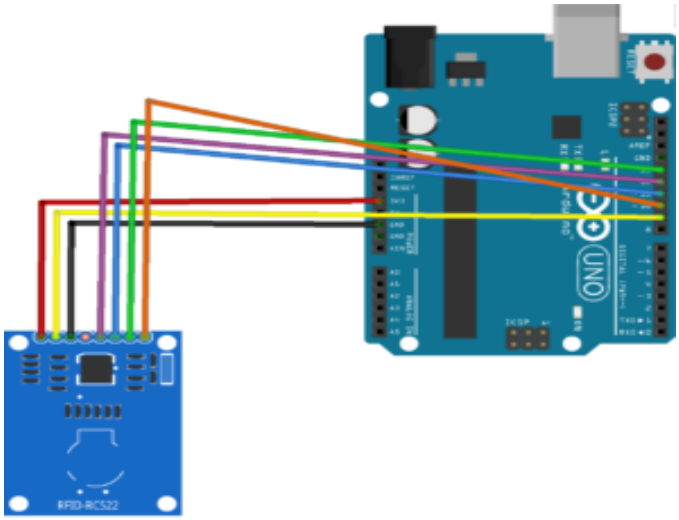
a. It is used for sending a message to the owner of the industry at the end of the day about the number of bottles filled.
b. Also used for sending a message if any interrupt comes into the system.

Circuit Diagram

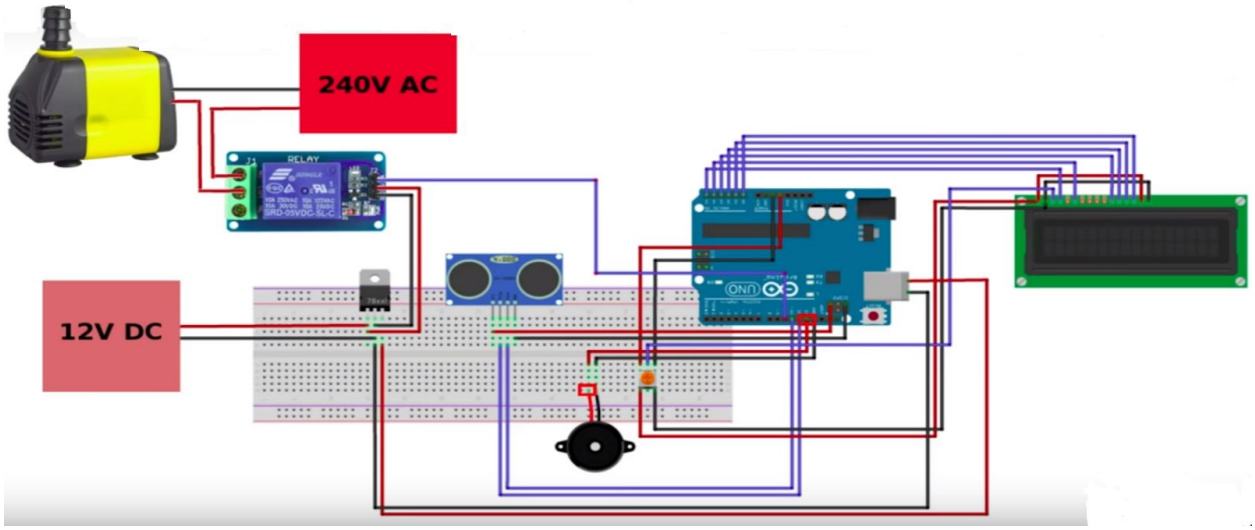
Roller Block:



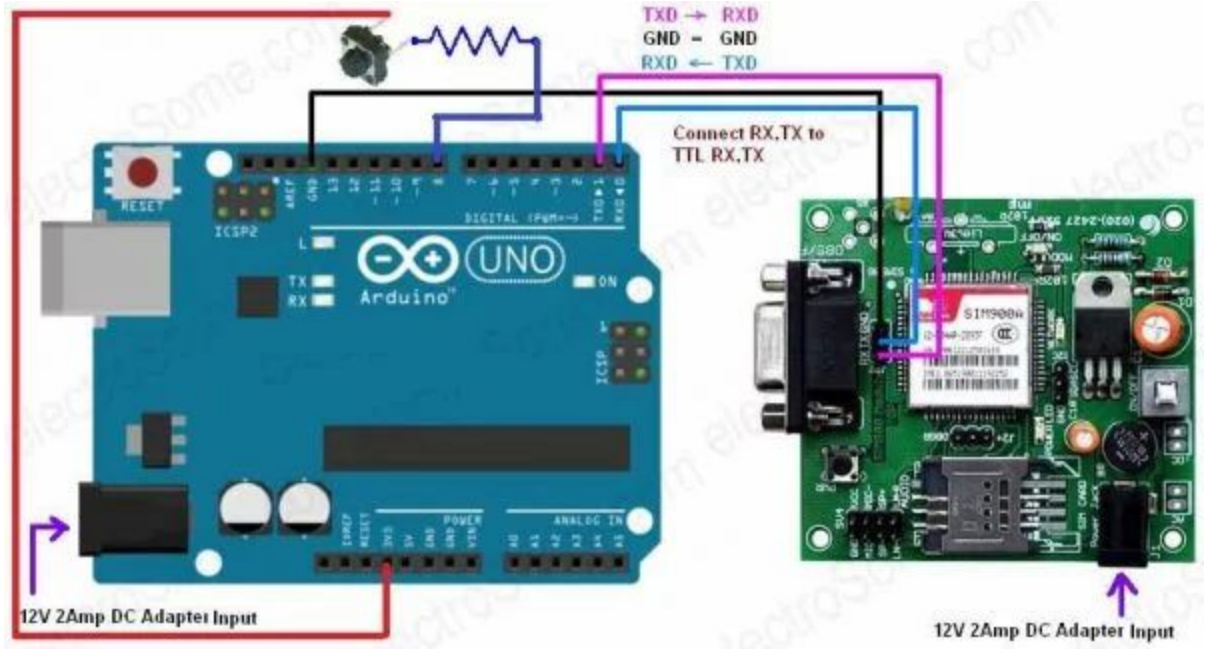
RFID Module:



Water Pump Module:



GSM Module:

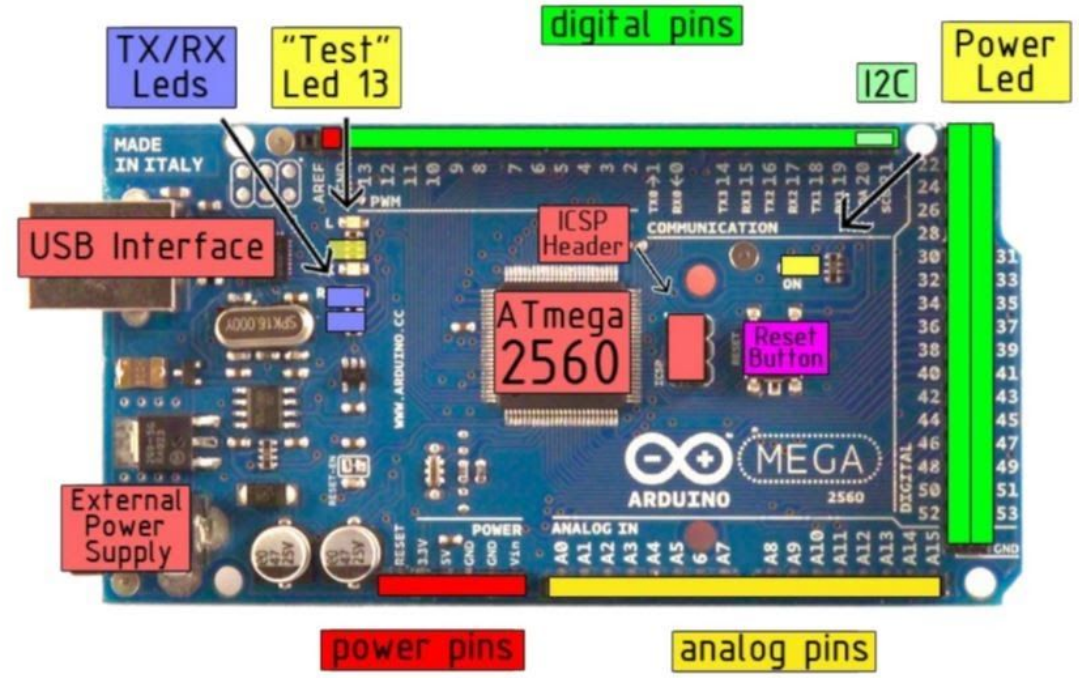


Datasheets:

1. Arduino Mega:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



2.LCD:

1. Features

1. 5x8 dots with cursor
2. 16characters *2lines display
3. 4-bit or 8-bit MPU interfaces
4. Built-in controller (ST7066 or equivalent)
5. Display Mode & Backlight Variations
6. ROHS Compliant

LCD type	☐TN			
	☐FSTN	☑FSTN Negative		
	☐STN Yellow Green	☐STN Gray	☐STN Blue Negative	
View direction	☑6 O'clock	☐12 O'clock		
Rear Polarizer	☐Reflective	☐Transflective	☑Transmissive	
Backlight Type	☑LED	☐EL	☐Internal Power	☑3.3V Input
		☐CCFL	☑External Power	☐5.0V Input
Backlight Color	☑White	☐ Blue	☐ Amber	☐Yellow-Green
Temperature Range	☑Normal	☐Wide		☐Super Wide
DC to DC circuit	☐Build-in		☑Not Build-in	
Touch screen	☐With		☑Without	
Font type	☑English-Japanese	☐English-Europen	☐English-Russian	☐other

2. MECHANICAL SPECIFICATIONS

Module size	80.0mm(L)*36.0mm(W)* Max13.5(H)mm
Viewing area	64.5mm(L)*16.4mm(W)
Character size	3.00mm(L)*5.23mm(W)
Character pitch	3.51mm(L)*5.75mm(W)
Weight	Approx.

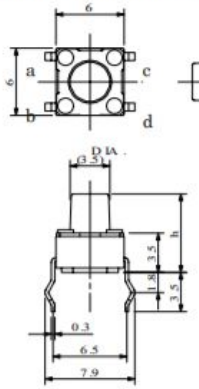
3.Push buttons:

■Features

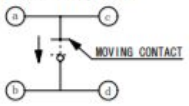
- Crisp clicking by tactile feedback
- Prevent flux rise by insert-molded terminal
- Ground terminal is attached
- Snap-in mount terminal

■Dimensions (Unit : mm)

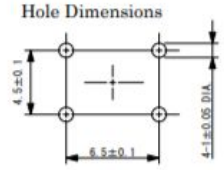
Without ground terminal



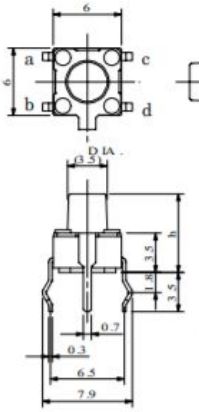
■Circuit Diagram



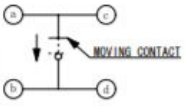
■ Recommended Mounting Hole Dimensions



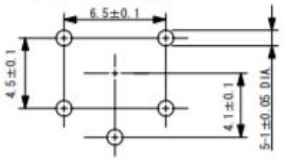
With ground terminal



■Circuit Diagram



■ Recommended Mounting Hole Dimensions



■Specifications

Item	Specification	Unit	Note
Operating Temperature Range	-20～+70	℃	
Storage Temperature Range	-40～+85	℃	
Type of Operation	Tactile Feedback		
Circuit Configuration	Push-On Momentary, 1 pole-1 throw		
Power Rating	MAX 50mA 24VDC		
Contact Resistance	MAX 100	mΩ	
Insulation Resistance	100MΩ Min. at 100VDC		
Dielectric Withstanding Voltage	250VAC for 1 minute		
Contact Bounce	MAX 5	ms	
Operating Force	0.98±0.49	N	A type
	1.57±0.49	N	B type
Return Force	MIN 0.10	N	A type
	MIN 0.49	N	B type
Travel	0.25 +0.2/-0.1	mm	

■Model Designation

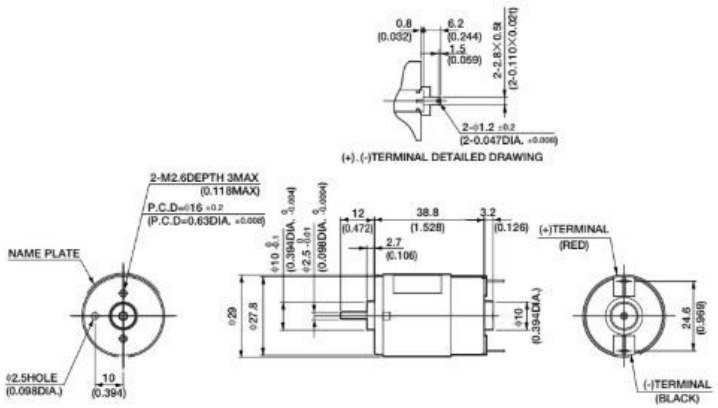
KSMC6 1 2 A

- A: Operation Force
- 2: Mount Height
- 1: Terminal Option (1:Without ground terminal, 2:With ground terminal)
- KSMC6: Model No.

	h (mm)
1	4.3
2	5
3	7
4	9.5

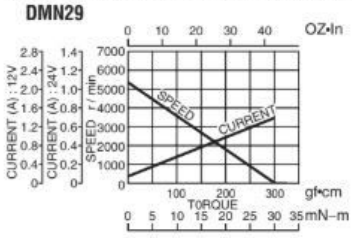
4. DC motor:

●DIMENSIONS Unit mm(inch)

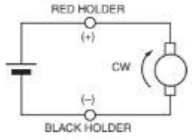


●CURRENT, SPEED-TORQUE CURVE

DMN29



●CONNECTION



●STANDARD SPECIFICATIONS

Model	Rated						No load		Stall torque		Weight	
	Output	Voltage	Torque		Current	Speed	Current	Speed	mN-m	oz·in		
	W	V	mN-m	oz·in	A	r/min	A	r/min			g	lb
DMN29BA	3.0	12	7.8	1.11	0.42	3700	0.07	5000	30	4.17	90	0.20
DMN29BB	3.0	24	7.8	1.11	0.21	3700	0.05	5000	30	4.17	90	0.20

5.A4988 Motor driver:

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

ELECTRICAL CHARACTERISTICS[1] at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ.[2]	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, I_F = –1.5 A	–	–	1.2	V
		Sink Diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		V_{DD}×0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD}×0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD}×0.7	–20	<1.0	20	µA
	$I_{IN(0)}$	V_{IN} = V_{DD}×0.3	–20	<1.0	20	µA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	µs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	µs
		R_{OSC} = 25 kΩ	23	30	37	µs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	µA
Current Trip-Level Error[3]	err_I	V_{REF} = 2 V, $\%I_{TripMAX}$ = 38.27%	–	–	±15	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 70.71%	–	–	±5	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 100.00%	–	–	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold[4]	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

[1]For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

[2]Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

[3]$V_{ERR} = [(V_{REF}/8) – V_{SENSE}] / (V_{REF}/8)$.

6.Relay Module:

Specifications

- On-board EL817 photoelectric coupler with photoelectric isolating anti-interference ability strong
- On-board 5V, 10A / 250VAC, 10A / 30VDC relays
- Relay long life can absorb 100000 times in a row
- Module can be directly and MCU I/O link, with the output signal indicator
- Module with diode current protection, short response time
- PCB Size: 45.8mm x 32.4mm

Pin Configuration

1. VCC: 5V DC
2. COM: 5V DC
3. IN1: high/low output
4. IN2: high/low output
5. GND: ground

7.IR Sensor:

- 5VDC operating voltage.
- I/O pins are 5V and 3.3V compliant.
- Range: Up to 20cm.
- Adjustable Sensing range.
- Built-in Ambient Light Sensor.
- 20mA supply current.
- Mounting hole.

3.Specifications

- Size: 50 x 20 x 10 mm (L x B x H)
- Hole size: ϕ2.5mm

8.RFID Module:

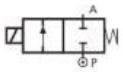
Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DDA}	analog supply voltage	$V_{DD(PVDD)} \leq V_{DDA} = V_{DDD} = V_{DD(TVDD)}$; $V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V	[1][2]	2.5	3.3	3.6	V
V_{DDD}	digital supply voltage			2.5	3.3	3.6	V
$V_{DD(TVDD)}$	TVDD supply voltage			2.5	3.3	3.6	V
$V_{DD(PVDD)}$	PVDD supply voltage		[3]	1.6	1.8	3.6	V
$V_{DD(SVDD)}$	SVDD supply voltage	$V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0$ V		1.6	-	3.6	V

Table 1. Quick reference data *...continued*

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{pd}	power-down current	$V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3$ V					
		hard power-down; pin NRSTPD set LOW	[4]	-	-	5	µA
		soft power-down; RF level detector on	[4]	-	-	10	µA
I_{DDD}	digital supply current	pin DVDD; $V_{DDD} = 3$ V		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; $V_{DDA} = 3$ V, CommandReg register's RcvOff bit = 0		-	7	10	mA
		pin AVDD; receiver switched off; $V_{DDA} = 3$ V, CommandReg register's RcvOff bit = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[5]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[6][7][8]	-	60	100	mA
T_{amb}	ambient temperature	HVQFN32		−25	-	+85	°C

9.Solenoid Valve:

Specifications	
Function	 Flow direction overseat 1 → 2
Maximum Viscosity	Max. 21cST (3 °E)
Body Material (Std)	Stainless Steel 1.4305 EN 10088 (AISI 303)
Orifice Material	Stainless Steel 1.4305 EN 10088 (AISI 303)
Flange Tube[1]	Stainless Steel (AISI 303)
Plunger and Top Stop	Stainless Steel 1.4105 EN 10088 (AISI 430F) or equivalent
Springs	Stainless Steel AISI 302
Seal Material (Std)	Foodgrade FKM
Connection Type (Std)	G parallel thread (ISO 228-1)
Shading Ring	Copper
Electrical Characteristics	
Standard Coil Voltage DC (=)	24 V
Standard Coil Voltage AC 50 Hz (~)	24 V, 110 V, 200 V, 230 V
Standard Coil Voltage AC 60 Hz (~)	24 V, 120 V, 220 V, 240 V
Voltage Tolerance	AC +10% to -15%
	DC +10% to -5%
Duty Cycle	100% ED
Protection Class	IP65 (EN 60529) with plug and gasket correctly fitted *
Electrical Connection	to industrial form B
Coil Insulation	Class F 155 °C
Power Rating (Standard)	AC 10 VA (holding) AC 16 VA (inrush) DC 7W

[1] With special nut, different from Standard.

Features and Benefits

- Direct Acting
- Robust construction for industrial applications
- Stainless steel AISI 430F operators with low residual magnetism
- Coils tested 100% in compliance to RoHS directive and to relevant international standards
- High quality seal materials
- Response time 5 to 25 ms

Pipe Size	Cv (gpm)	Kv (m^3/h)	OPD (bar)		Orifice (mm)	Seal Material	Valve Code
			AC Voltages	DC Voltages			
1/8"	0.09	0.08	0 - 22	0 - 18	1.5	FKM	B298DVC
1/8"	0.13	0.11	0 - 18	0 - 8	2.0	FKM	B298DVE
1/8"	0.19	0.16	0 - 13	0 - 2.5	2.5	FKM	B298DVG
1/8"	0.25	0.21	0 - 8	0 - 1	3.0	FKM	B298DVH
1/8"	0.09	0.08	0 - 24	0 - 24	1.5	KALREZ®	B298DKC
1/8"	0.13	0.11	0 - 18	0 - 15	2.0	KALREZ®	B298DKE
1/8"	0.19	0.16	0 - 15	0 - 3	2.5	KALREZ®	B298DKG

10.Buzzer:

Buzzer Pin Configuration

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

Buzzer Features and Specifications

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard and Perf board friendly

11.Ultrasonic Sensor:

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

12.LED light:

Pararmeter	Symbol	Red	Green	Blue	Unit
Forward current	I_F	20	20	20	mA
Peak forward current(Duty Cycle=$\frac{1}{10}$, 10KHz)	I_{PF}	30	30	30	mA
Reverse current (V_R=5V)	I_R	10	10	10	μA
Operating temp	T_{OPR}	-25~ 85	-25~ 85	-25~ 85	℃
Storage temp	T_{STG}	-30~85	-30~85	-30~85	℃
Peak Emission Wavelength	λP_H	625	520	467.5	nm

* Soldering Bath: not more than 5 seconds @260 ℃.The bottom ends of the plastic reflector should be at least 2mm above the solder surface

Soldering Iron: not more than 3 seconds @300 ℃ under 30W

LED Chip Typical Electircal & Optical Characteristics: (Ta=25℃)

ITEMS	Color	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	Red	V_F	I_F=20mA	1.8	2.0	2.2	V
	Green			3.0	3.2	3.4	
	Blue			3.0	3.2	3.4	
Luminous Intensity	Red	I_V	I_F=20mA	– – –	– – –	800	mcd
	Green			– – –	– – –	4000	
	Blue			– – –	– – –	900	
Wavelenength	Red	$\Delta\lambda$	I_F=20mA	620	623	625	nm
	Green			515	517.5	520	
	Blue			465	466	467.5	
Light Degradation after 1000 hours	Red	-4.68% ~ -8.27%					
	Green	-11.37% ~ -15.30%					
	Blue	-8.23% ~ -16.81%					

13.Potentiometer:

Electric Specifications

These are standard features; other specifications can always be studied on request.

	Through-Hole	SMD
Range of resistance values Lin (A) Log (B) Antilog (C)	 100Ω ... 5MΩ 1 KΩ ... 2,2 MΩ	 100Ω - 1MΩ 1KΩ - 1MΩ
Tolerance Special tolerances available on request	100Ω ... 1MΩ ±20% >1MΩ ... 5MΩ ±30% Out of range: Rn> 5MΩ: +50% -30%	< 1MΩ ± 25%
Variation laws	Lin (A), Log (B), Antilog (C) Other tapers available on request	
Residual resistance	Lin (A), Log (B), Antilog (C) ≤ $5*10^{-3}$*Rn Minimum value 2Ω	
CRV - Contact Resistance Variation (dynamic)	≤3%Rn	
CRV - Contact Resistance Variation (static)	≤5%Rn	
Maximum power dissipation at 40° C. Lin (A) No Lin (B, C)	 0,10W 0,06W	
Maximum voltage at 40°C Lin (A) No Lin (B, C)	 100 VDC 60VDC	
Operating temperature	-25°C ... +70°C	
Temperature coefficient	100Ω - 10KΩ → +200/ -300 ppm. >10KΩ - 5MΩ → +200/ -500 ppm	100Ω - 100KΩ → +200/ -500 ppm. >100KΩ - 1MΩ → +200/ -1000 ppm.

14. Water Pump:

APP Pumps		APP 11 / 1200	APP 11 / 1500	APP 13 / 1200	APP 13 / 1500
Code number		180B3212	180B3211	180B3214	180B3213
Geometric displacement	cm^3/rpm	166	137	197	166
	in^3/rpm	10.1	8.4	12.0	10.1
Rated flow at max. speed 1)	m^3/h	11.0	11.1	13.1	13.5
	gpm	48.4	49	57.7	59.3
Outlet min. pressure 2)	bar	30	30	30	30
	psi	435	435	435	435
Outlet max. pressure, cont. 3)	bar	80	70	80	70
	psi	1160	1015	1160	1015
Inlet min. pressure	bar	2	2	2	2
	psi	29	29	29	29
Inlet max. pressure, cont.	bar	5	5	5	5
	psi	72.5	72.5	72.5	72.5
Inlet max. pressure, peak	bar	10	10	10	10
	psi	145	145	145	145
Max. speed cont.	rpm	1200	1500	1200	1500
Min. speed cont.	rpm	700	700	700	700
Power requirement at max. speed and 60 bar outlet pressure	kW	23	24	27	29
	hp	30.8	32.2	36.2	38.9
Torque at 60 bar outlet pressure	Nm	179	148	212	179
	lbf-ft	132	109	157	132
Weight	Kg	75	75	75	75
	lb	165	165	165	165
Integrated flushing valve		YES	YES	YES	YES

15. GSM Module:

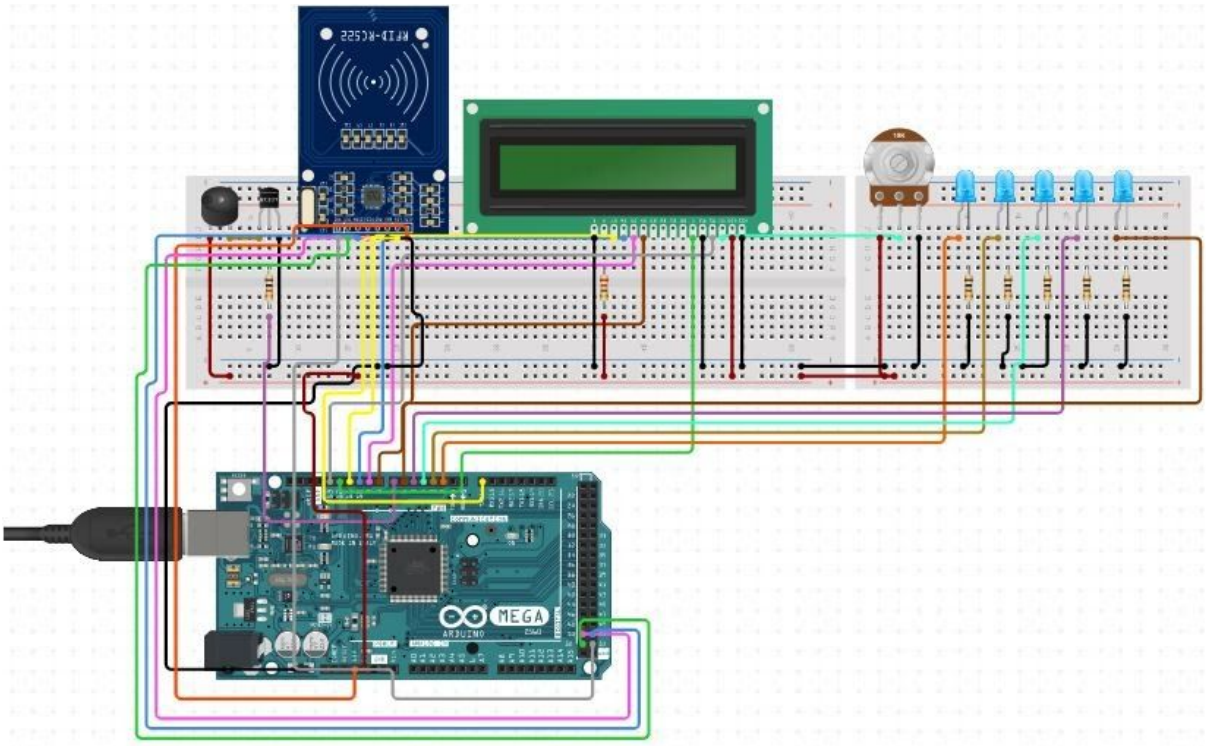
Feature	Implementation
Power supply	Single supply voltage 3.4V – 4.5V
Power saving	Typical power consumption in SLEEP mode is 1.5mA (BS-PA-MFRMS=5)
Frequency Bands	● SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the 2 frequency bands automatically. The frequency bands also can be set by AT command. ● Compliant to GSM Phase 2/2+
GSM class	Small MS
Transmitting power	● Class 4 (2W) at EGSM 900 ● Class 1 (1W) at DCS 1800
GPRS connectivity	● GPRS multi-slot class 10 （default） ● GPRS multi-slot class 8 (option) ● GPRS mobile station class B

Project Modules and Respective codes:

RFID Module for secure access and to turn on the lights in the factory and start the alarms if a person tries to gain unauthorised access

Sensors/Actuators/Displays used:

- RFID tag
- LCD display
- Buzzer
- LED



Code:

```
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
MFRC522 mfrc522(10, 9);

int led1 = 7;
int led2 = 6;
int led3 = 8;
int led4 = 4;
int led5 = 3;
int buzzerPin = 5;
String tagUID = "29 B9 ED 23";

void setup() {
 pinMode(buzzer, OUTPUT);
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);
 pinMode(led5, OUTPUT);
 lcd.begin();
 lcd.backlight();
 SPI.begin();
 mfrc522.PCD_Init();
 lcd.clear();
}

void loop() {
 lcd.setCursor(0, 0);
 lcd.print(" RFID secured bottle-filler.");
 lcd.setCursor(0, 1);
 lcd.print(" Show Your Tag ");
 if ( ! mfrc522.PICC_IsNewCardPresent()) {
  return;
 }
 if ( ! mfrc522.PICC_ReadCardSerial()) {
  return;
 }

 String tag = "";

 for (byte i = 0; i < mfrc522.uid.size; i++) {
```

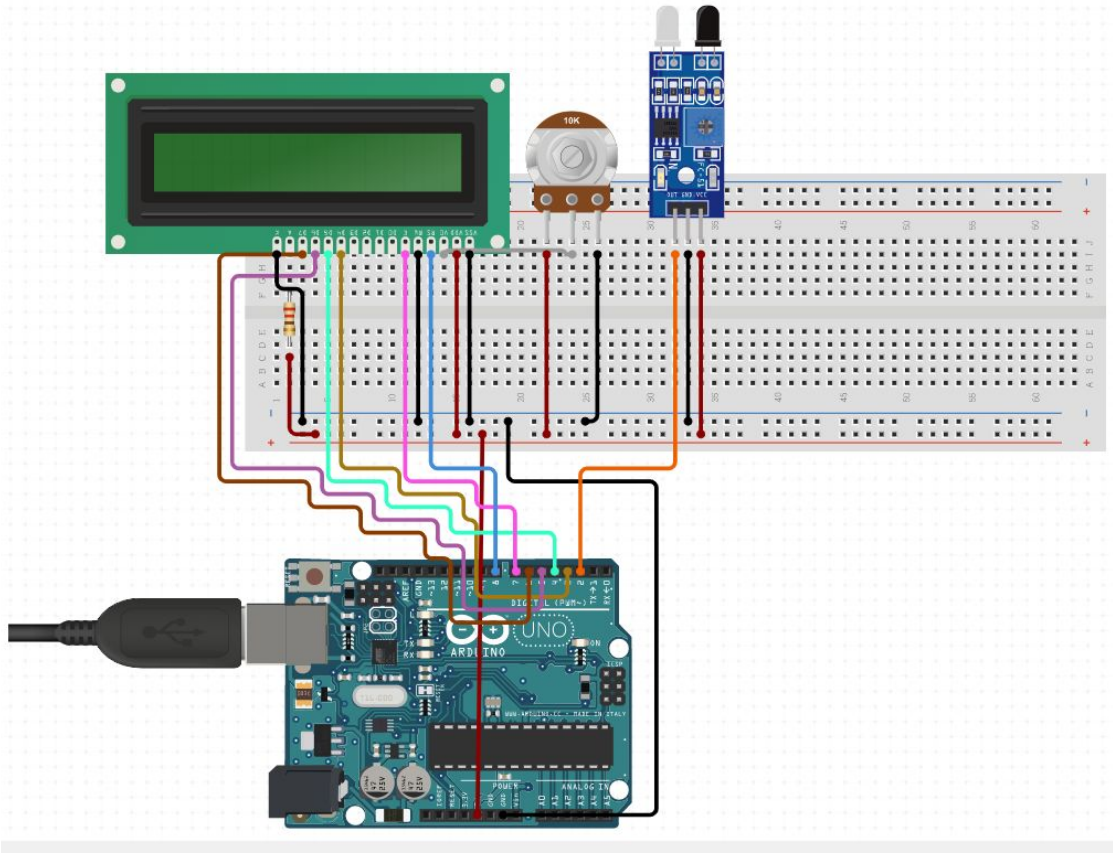
```
    tag.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    tag.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  tag.toUpperCase();

  if (tag.substring(1) == tagUID) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Access Granted");
    lcd.setCursor(0, 1);
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, HIGH);
    digitalWrite(led5, HIGH);
    lcd.clear();
  }
  else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wrong Tag Shown");
    lcd.setCursor(0, 1);
    lcd.print("Access Denied");
    digitalWrite(buzzer, HIGH);
    delay(3000);
    digitalWrite(buzzer, LOW);
    lcd.clear();
  }
}
```

Roller block module to move the bottles forward and IR sensor to display their count on LCD

Sensors/Actuators/Displays used:

- DC motor
- Motor Driver
- LCD Display
- IR Sensor

Sub - Module 1: Use of IR sensor to detect the presence of bottle and display the count of bottles on LCD display

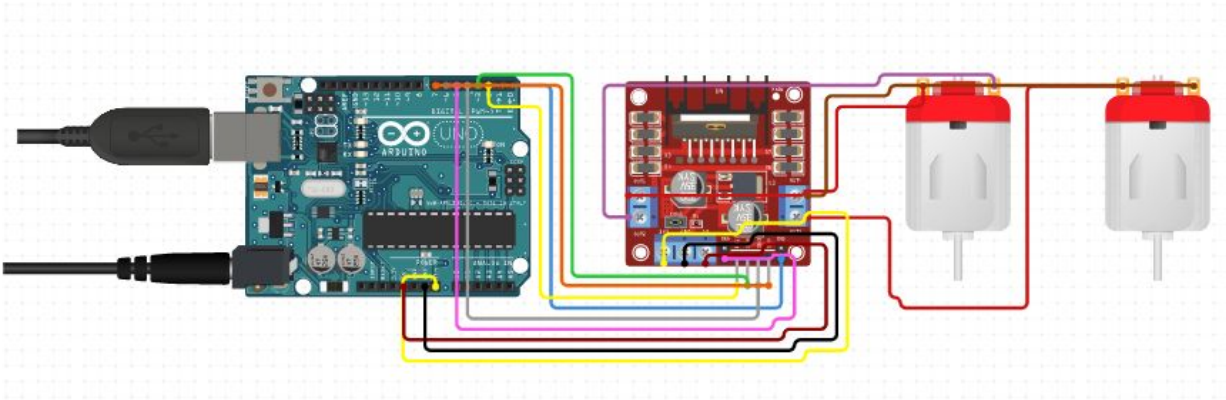


Code:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int count;
const int pin_ir_in = 9;
```

```
int prev_state =1;
int curr_state = 1;
int counter = 0;

void setup() {
 lcd.begin(16, 2);
 pinMode(pin_ir_in, INPUT);
 Serial.begin(9600);
 lcd.print("Bottles filled:");
}

void loop() {
 lcd.setCursor(0, 1);
 int curr_state = digitalRead(pin_ir_in);
 if (curr_state != prev_state) {
  if(curr_state == 0) {
   counter++;
   Serial.println("Obstacle Detected...!");
   Serial.println(counter);
   lcd.print(counter);
  }
  prev_state = curr_state;
 }
 delay(100);
 }
```

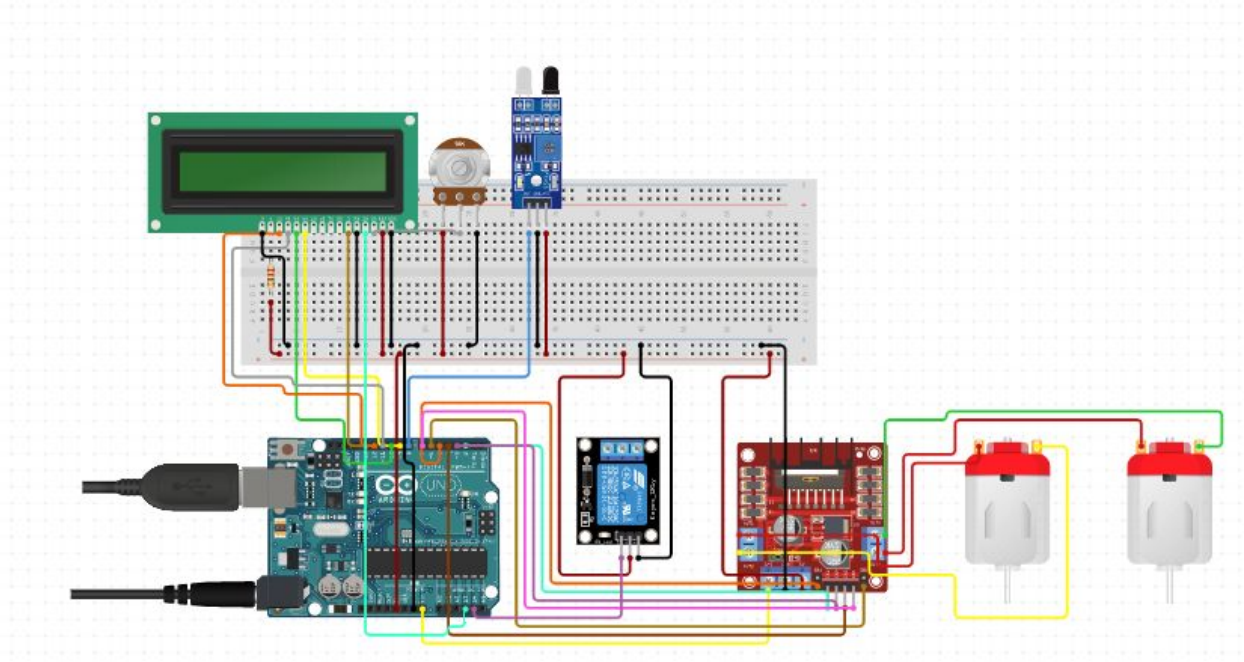
Sub-Module 2: Use of dc motor and motor driver to move the bottles over the roller



Code:

```
const int motorPin1  = 9;
const int motorPin2  = 10;

void setup(){
   pinMode(motorPin1, OUTPUT);
   pinMode(motorPin2, OUTPUT);
 }

void loop(){
   analogWrite(motorPin1, 180);
   analogWrite(motorPin2, 0);
      delay(5000);

   analogWrite(motorPin1, 0);
   analogWrite(motorPin2, 180);
    delay(5000);

    analogWrite(motorPin1, 0);
   analogWrite(motorPin2, 0);
}
```

Main-Module:



Code :

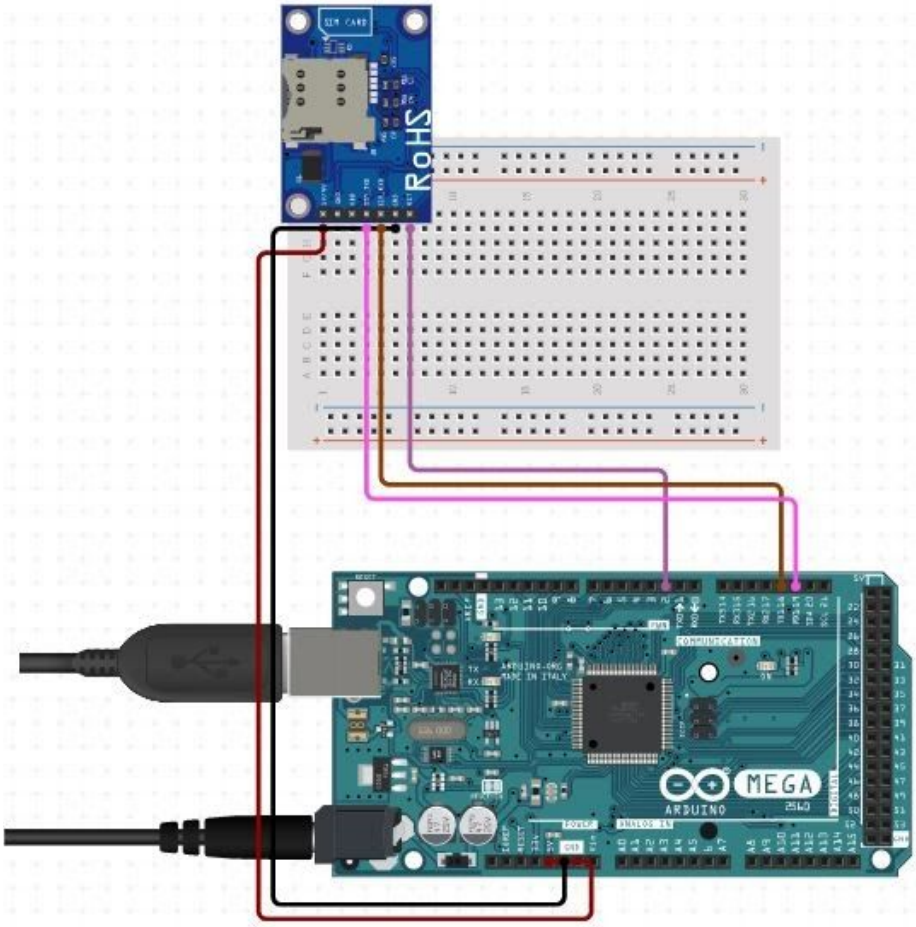
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int count;
int relay=7;
int ir = 3;
const int motorPin1  = 9;
const int motorPin2  = 10;
int prev = 1;
int curr = 1;
int counter = 0;
boolean solenoid_on = false;

void setup() {
  lcd.begin(16, 2);
  Serial.begin(9600);
  pinMode(ir, INPUT);
  pinMode(relay,OUTPUT);
 pinMode(motorPin1, OUTPUT);
 pinMode(motorPin2, OUTPUT);
```

```
  pinMode(4,OUTPUT);
  pinMode(2,OUTPUT);
  lcd.print(“Bottles filled : “);
 }

 void loop() {
  lcd.setCursor(0, 1);
  int curr = digitalRead(ir);
  if (curr != prev) {
    if (curr == 0) {
      counter++;
      Serial.println("Obstacle Detected...!");
      Serial.println(counter);
      lcd.print(counter);
    }
    prev_state = curr_state;
  }
  delay(100);
  if (solenoid_on) {
   delay(6000);
    analogWrite(motorPin1, 0);
    analogWrite(motorPin2, 0);
   solenoid_on = false;
  }
  digitalWrite(relay,LOW);
  analogWrite(motorPin1, 180);
  analogWrite(motorPin2, 0)

}
```

GSM Module to send the total number of soft-drink bottles filled via message to the owner at the end of the day

Sensors/Actuators/Displays used:

- GSM Module



Code:

//This code is in continuation with previous code, we have assumed that the total number of bottles filled in a day is 150.

```
#include <SoftwareSerial.h>

int count = 150;

SoftwareSerial mySerial(9, 10);
```

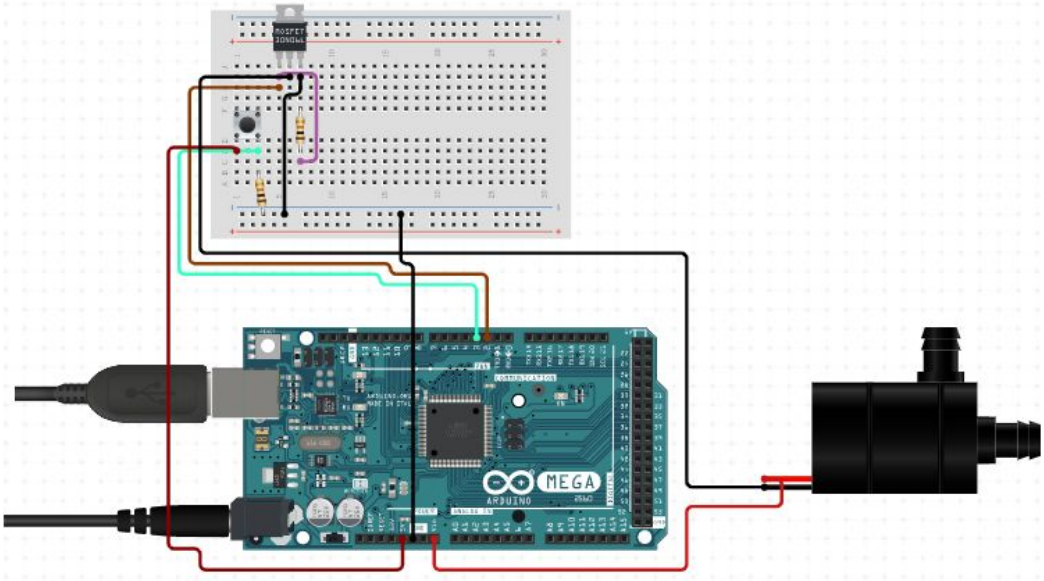
```
void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);
  delay(100);
}

void loop() {
  if (Serial.available()>0) {
    switch(Serial.read) {
      case 's': SendMessage();
                break;
      case 'r': ReceiveMessage();
                break;
    }
  }
  if (mySerial.available()>0) {
   Serial.write(mySerial.read());
  }
}

 void SendMessage() {
  mySerial.println("AT+CMGF=1");
  delay(1000);
  mySerial.println("AT+CMGS=\"+919979481706\"\r");
  delay(1000);
  mySerial.println("Total number of bottles filled today : " + count);
  delay(100);
  mySerial.println((char)26);
  delay(1000);
 }

 void ReceiveMessage() {
  mySerial.println("AT+CNMI=2,2,0,0,0");
  delay(1000);
 }
```

Water Pump Module to fill the soft-drink bottles upto a certain level, and check the water level in the tank using ultrasonic sensor:

Sensors/Actuators/Displays used:

- Ultrasonic sensor
- Buzzer
- Water Pump
- Relay Module

Sub-Module 1: Water pump module, operated using push button

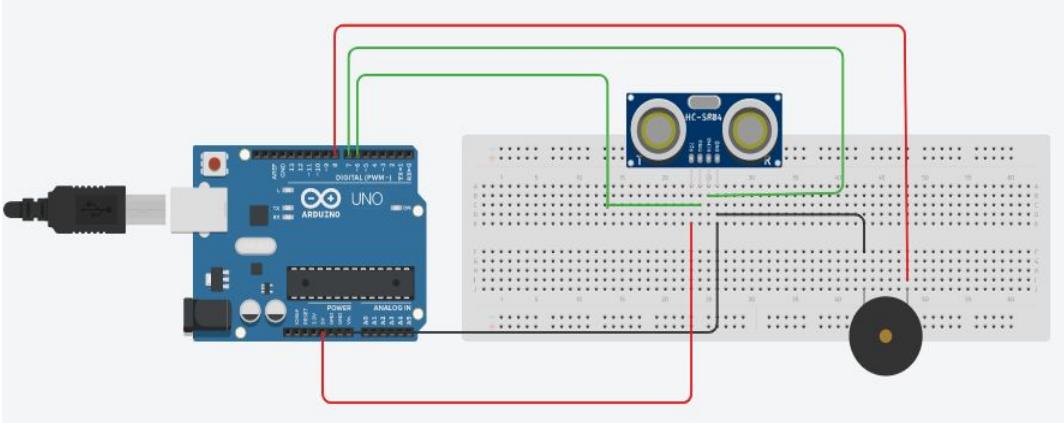


Code:

```
int WATERPUMP = 13;
int button = 8;

void setup() {
  pinMode(13,OUTPUT);
  pinMode(8,INPUT);
  Serial.begin(9600);
  while (! Serial);
  Serial.println("Speed 0 to 255");
}
```

```
void loop() {
  if (Serial.available()) {
    int speed = Serial.parseInt();
    if (speed >= 0 && speed <= 255) {
      analogWrite(WATERPUMP, speed);
    }
  }
  val = digitalRead(8);
  if(val == LOW) {
  digitalWrite(13,LOW);
  }
  else {
  digitalWrite(13,HIGH);
  }
  delay(400);
}
```

Sub-Module 2: Ultrasonic sensor (buzzer will ring when distance increase than threshold level)

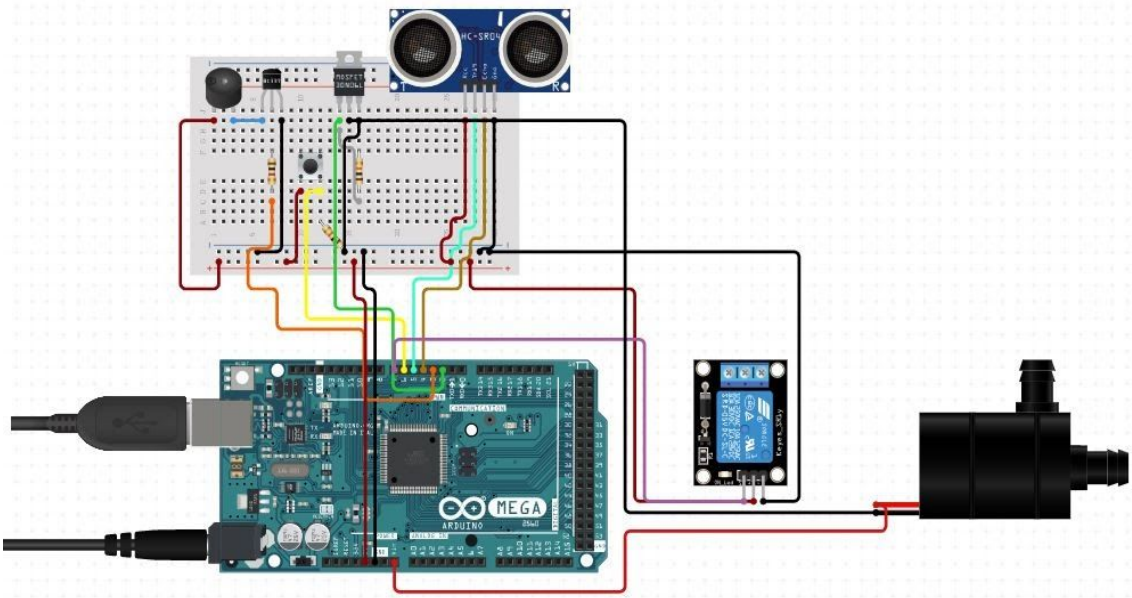


Code:

```
int trig_pin = 6;
int echo_pin = 7;
int buz = 8;
int time;
int dist;
```

```
void setup() {
  pinMode(echo_pin,INPUT);
  pinMode(trig_pin,OUTPUT);
  pinMode(buz,OUTPUT);
  Serial.begin(9600);
}

void loop() {
   digitalWrite(trig_pin,HIGH);
   delayMicroseconds(10);
   digitalWrite(trig_pin,LOW);
   time = pulseIn(echo_pin,HIGH);
   dist = (time * 0.034) / 2;
   Serial.print("Distance :");
   Serial.println(dist);
   if(dist>=150){
     tone(buz,1000);
     delay(1000);
     noTone(buz);
     delay(000);
     tone(buz,1000);
     delay(1000);
     noTone(buz);
   }
}
```

Main-Module:



Code:

```
#include <NewPing.h>
#include <EEPROM.h>

#define TRIGGER_PIN  6
#define ECHO_PIN     7
#define MAX_DISTANCE 500
#define RELAYPIN 8
#define EXTRELAYPIN 13
#define BUZZER 9
#define buttonPin 10
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

byte readval;

int addr = 0;
int addr2 = 1;
int flag;
int buttonState = 0;
int percentage;
int SpmpSensorPin = A0;
int SpmpsensorValue = 0;
```

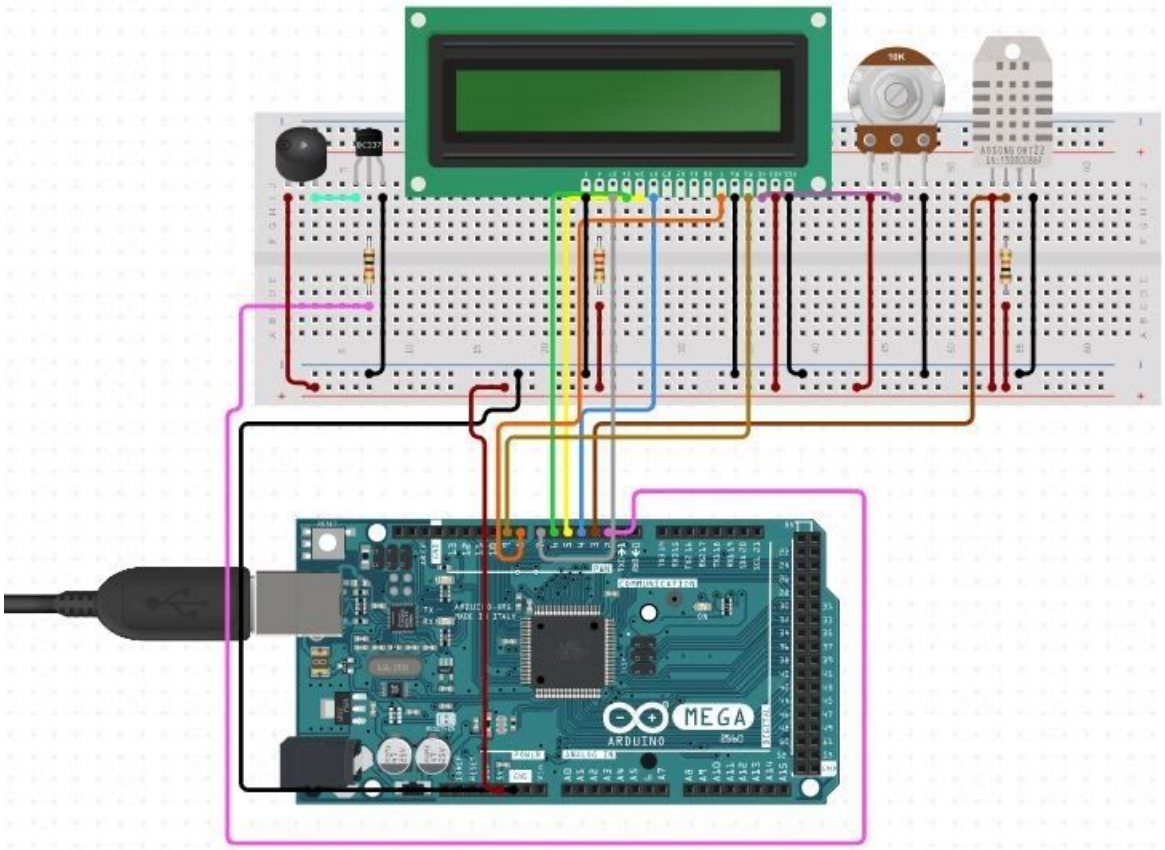
```
 float val;
 float TankHeight, MaxWaterLevel, EmptySpace, SonarReading,
ActualReading,  Temp;

void setup() {
  Serial.begin(9600);
  Serial.println(flag);
  pinMode(RELAYPIN,OUTPUT);
  pinMode(EXTRELAYPIN,OUTPUT);
  digitalWrite(RELAYPIN,LOW);
  digitalWrite(EXTRELAYPIN,HIGH);
  pinMode(BUZZER,OUTPUT);
  digitalWrite(BUZZER,LOW);
  for (int i=0; i<=5; i++) {
   buttonState = digitalRead(buttonPin);
   if (buttonState == HIGH) {
    TankHeight =sonar.ping_cm();
    EEPROM.write(addr, TankHeight);
   }
   delay(1000);
  }

  TankHeight= EEPROM.read(addr);
  MaxWaterLevel=0.85*TankHeight;
  EmptySpace=TankHeight-MaxWaterLevel;
 }

 void loop() {
  delay(50);
  SonarReading=sonar.ping_cm();
  SpmpsensorValue=analogRead(SpmpSensorPin);
  Serial.println(SpmpsensorValue);

  Temp= SonarReading-EmptySpace;
  ActualReading= MaxWaterLevel-Temp;
  percentage=(ActualReading/MaxWaterLevel*100);

  if(SpmpsensorValue>=100) {
   if(percentage<=20) {
```

```
    digitalWrite(RELAYPIN,HIGH);
    digitalWrite(EXTRELAYPIN,LOW);
    flag=1;
    EEPROM.write(addr2, flag);
    flag= EEPROM.read(addr2);
   }
   else if(percentage>20 && percentage<=100) {
    flag= EEPROM.read(addr2);
    if(percentage>20 && percentage<=100 && flag ==1) {
     digitalWrite(RELAYPIN,HIGH);
     digitalWrite(EXTRELAYPIN,LOW);
    }
    else if(percentage>20 && percentage<=100 && flag ==0) {
     digitalWrite(RELAYPIN,LOW);
     digitalWrite(EXTRELAYPIN,HIGH);
    }
   }
   else if(percentage>100) {
    delay(500);
    digitalWrite(RELAYPIN,LOW);
    digitalWrite(EXTRELAYPIN,HIGH);
    flag=0;
    EEPROM.write(addr2, flag);
    flag= EEPROM.read(addr2);
   }
  }

  else if(SpmpsensorValue<=100) {
   flag= EEPROM.read(addr2);
   if(flag==1) {
    digitalWrite(BUZZER,HIGH);
    digitalWrite(RELAYPIN, LOW);
    digitalWrite(EXTRELAYPIN, HIGH);
    delay(100);
    digitalWrite(BUZZER,LOW);
    delay(100);
   }
  }
}
```

Temperature and Humidity Sensor to keep the factory temperature ambient

Sensors/Actuators/Displays used:

- Humidity sensor
- Buzzer
- LCD Display



Code:

```
#include <dht.h>
#include <LiquidCrystal.h>

dht.DHT;

#define DHT11 PIN 7

void setup() {
  serial.begin(9600);
```

```
   lcd.begin(16, 2);
   lcd.print("hello, world!");
  pinMode(buz,OUTPUT);
}

void loop() {
   float h=dht.readHumidity();
   float t= dht.readTemperature();
   int buz=8;
   int chk.DHT.Read11(DHT11_PIN);
   lcd.setCursor(0, 0);
   lcd.print("Temperature =");
   lcd.print(t);
   lcd.setCursor(0,1);
   lcd.print("Humidity =");
   lcd.print(h);
   delay(1000);

    if (t>25) {
        tone(buz,1000);
        delay(1000);
    }
   else {
     noTone(buz);
    }
}
```

Note:

The code for main-modules of **water pump** and **roller block** were searched for before the lockdown, and hence we have included them here. However, the code is not tested in the lab.

Contribution by group members:

Member Name	**Contribution**
1. Aanshi Patwari	• Roller + Counter Block Module • Putting report 3 together
2. Dipika Pawar	• Water Pump Module
3. Miracle Rindani	• RFID Module
4. Bhumiti Gohel	• GSM Module • Temperature and Moisture sensor Module

References

- RFID Module: https://electronicshobbyists.com/rfid-basics-and-rfid-module-interfacing-with-arduino/
- Roller Module: https://circuitdigest.com/microcontroller-projects/automatic-bottle-filling-system-using-arduino
- GSM Module: http://www.circuitstoday.com/interface-gsm-module-with-arduino
- Water pump Module:
    - http://www.circuitstoday.com/water-level-indicator-arduino-ultrasonic-sensor
    - https://www.youtube.com/watch?v=08dcoXZEtqs
- Moisture and Temperature Module: https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/
- IR Sensor based counter: https://www.gadgetronicx.com/visitor-counter-project-using-arduino/
- Circuits designed on:
    - https://www.circuito.io/app
    - https://www.tinkercad.com/dashboard?type=circuits&collection=designs