

# MilkyWay Maker

## Introduction

**Name:** Mara Sofia **STAN**

**Group:** 1221A - FILS

This Arduino-based system offers a convenient and customizable way to make delicious cocoa milk drinks at home. The project contains a smart liquid dispenser with a keyboard that allows the user to input the desired quantity of milk and a distance sensor that detects the presence of a cup to activate the powder dispensing mechanism.

The idea behind this project was to simplify and automate the process of making these drinks and ensure consistent quality every time. MilkyWay Maker is a project that was born out of a love for milk-coffee beverages, such as Nescafe and Nesquik thus I strongly believe that this project is not only useful for coffee lovers, but also for those interested in learning about electronics and programming.



## General Description



The user can input the desired quantity of milk using the keyboard, and the system dispenses the milk accordingly. The distance sensor detects the presence of a cup, activating the powder dispensing mechanism to add the cocoa powder. This ensures that the drink is prepared accurately and consistently every time.

The project is not only aimed at coffee lovers but also at those interested in learning about electronics

and programming. It combines the functionality of Arduino, the convenience of automation, and the enjoyment of delicious milk-coffee beverages. The provided code includes various functionalities, such as reading the keypad inputs, controlling the servo motor, calculating the flow rate of milk, displaying information on an LCD screen, and playing beep sequences for notifications.



## Hardware Design

Components:

- Water Diaphragm Pump, 6-12V, R385
- 400-Point Breadboard
- 3×4 Membrane Keypad
- LCD 1602 + I2C LCD Interface Module 1602
- HC SR-04P Ultrasonic Sensor, 3-5.5V
- SG90 Servo Motor, 180 degrees
- Flow Sensor 0.3-6l/min, YF-S401, 120 degrees
- 5V Relay Module, High-Level Trigger
- 9V Battery
- UNO R3 Arduino Compatible Development Board
- Passive Buzzer 5V
- Water Pump Hose 6x8mm
- Dupont Wires





# Software Design

Here are the features of this setup:

- **User Input:** The keypad allows the user to input the desired volume of liquid. The code reads the input and stores it as the target volume for dispensing.
- **Liquid Dispensing:** The flow sensor measures the flow rate of the liquid. The code calculates the amount of liquid dispensed based on the flow rate and time. It keeps track of the total amount of liquid dispensed until it reaches the target volume.
- **Powder Dispensing:** The ultrasonic sensor detects the presence of a glass near the machine. When the glass is detected, a servo motor is activated to dispense the coffee powder.
- **Display and Feedback:** The LCD display shows various information such as the input volume, flow rate, and total dispensed volume. It provides feedback messages like "Insert volume," "Volume reached," and "Remove cup" to guide the user through the process.
- **Buzzer:** The buzzer is used to generate beeping sounds during certain events, such as when the volume is entered correctly or when the target volume is reached.

Below is the overall flow and explanation of the code. It combines input from the keypad, measurements from the ultrasonic sensor and flow sensor, and control of the water pump and servo motor to automate the process of dispensing milk based on the desired volume.

Include the necessary libraries for the keypad, LCD display, and servo motor( *<Keypad.h>*, *<LiquidCrystal\_I2C.h>*, *<Servo.h>*) and define the pin connections for various components.

```
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

const int ROW_NUM = 4; //4 rows
const int COLUMN_NUM = 3; //3 columns
const int thresholdDistance = 5; // in cm

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

LiquidCrystal_I2C lcd(0x27, 16, 2);
byte pin_rows[ROW_NUM] = {4, 5, 6, 7}; //connect to the row pinouts of the keypad
byte pin_column[COLUMN_NUM] = {8,9, 10}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM);
```

Declare global variables for pins, counters, calibration factors, and flags.

```
const int ultrasonicTrigPin = 12;
```

```
const int ultrasonicEchoPin = 13;
const int servoPin = 11;

int sensorInterrupt = 0; // interrupt 0
int sensorPin = 2; //Digital Pin 2
unsigned int SetPoint = 400; //400 milileter
int buzzerPin = A3;
String code="";

/*The hall-effect flow sensor outputs pulses per second per litre/minute of
flow.*/
float calibrationFactor = 40; //You can change according to your datasheet

volatile byte pulseCount =0;

float flowRate = 0.0;
unsigned int flowMilliLitres =0;
unsigned long totalMilliLitres = 0, volume = 0;

unsigned long oldTime;
const int relais_moteur = 3; // the relay is connected to pin 3 of the
Arduino board
Servo myServo;
bool objectDetected = false;
bool volumeEntered = false;
bool beepSequence = false;
bool beepSequence2 = false;
```

1. Call the `setup()` function to initialize the settings and configurations of the Arduino board and connected components. Set pin modes, attach interrupts, and initialize the LCD display.
2. Call the `pulseCounter()` function, which is an interrupt service routine that increments a pulse count variable. It is triggered by the falling edge of the pulse from the flow sensor, indicating the flow of liquid.
3. Call the `resetProcess()` function to reset the variables and settings related to the flow measurement process. Clear the LCD display, set it to the initial state, and initialize the necessary variables.
4. Call the `getDistance()` function to measure the distance using the ultrasonic sensor. Send a trigger pulse and calculate the duration of the echo pulse to determine the distance. Return the distance value in centimeters.
5. Call the `playBeepSequence()` function to generate a beep sound using a buzzer. Play a sequence of three short beeps with a delay between them.

**Below I added a .zip file containing a demo video of the project:**

[milkywaymaker-demovideo.zip](#)

## Results

I am thrilled and proud of the successful incorporation of my everyday favorite beverage into an Arduino project. By creating the Milky Way Maker coffee maker machine, I have not only combined my passion for coffee with my interest in electronics and programming but also brought a new level of convenience and automation to my daily coffee routine. ❌

Looking forward, the project holds immense potential for customization and expansion. The Arduino platform allows for further enhancements and the addition of new features. For instance, I could explore incorporating temperature control to ensure the perfect brewing temperature or create programmable recipes to cater to different coffee preferences. This adaptability and room for growth make the Milky Way Maker a truly versatile and dynamic coffee maker machine.

## Conclusions

In conclusion, the development of the Milky Way Maker coffee maker machine using Arduino has been a successful project. It has demonstrated the potential of integrating electronics and programming into everyday appliances to automate processes and enhance functionality. The user-friendly interface, sensor integration, and customization options make it a versatile and enjoyable coffee-making experience. The project has provided valuable learning opportunities and highlights the possibilities for future innovation in home automation. ❌

## Download

An archive containing the source code and the additional libraries used: [milkywaymaker.zip](#)

## Jurnal Log

- **15.05:** Researched for a project theme
- **17.05:** Decided on Milky Way Maker as the project (reflecting my love for chocolate drinks in general so I knew this was the one for me)
- **18.05:** Ordered the components required for the project
- **21.05:** Components arrived, began brainstorming the design and took a trip to *IKEA* to purchase a wooden box and recipients
- **23.05:** Enlisted my dad's help in cutting the wood box and creating necessary indents for the pump hose and other components ❌
- **26.05:** Finally assembled all the components and completed the Milky Way Maker coffee maker machine

## Bibliography/Resources

Inspo for the liquid dispensing process:

<https://www.robotique.tech/robotics/intelligent-water-filling-system-controlled-by-arduino/>

Important documentation for calibrating the flow sensor (factor & vertical inclination) :

<https://www.epitran.it/ebayDrive/datasheet/YF-S401.pdf>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/milywaymaker>



Last update: **2023/05/30 12:36**