CHOCOLATE VENDING MACHINE

BY

M.JEYA PRABHA

*1.*Project Title:

Arduino-Based Chocolate Vending Machine.

1.1 Objectives:

- To build an automated vending machine using Arduino.
- Allow users to select from multiple chocolate options via keypad.
- Display cart and payment info on an LCD.
- Dispense chocolate using a servo motor after successful payment.

1.2 Description:

This project simulates a chocolate vending machine controlled by an Arduino Uno. It uses a 4x4 keypad for input, an I2C LCD to show available items and payment instructions, and a servo motor to simulate dispensing chocolate. Users can scroll through multiple items, add them to a cart, enter payment, and receive change if applicable.

1.3 Scope & Features:

✓ Multi-item scrolling menu
✓ Add-to-cart logic
✓ Amount entry via keypad
✓ Balance calculation
✓ Chocolate dispensing via servo
✓ LCD interface

1.4 System Requirements:

Inputs:

- Keypad inputs (chocolate selection, payment entry)

Outputs:

- LCD display (menu, status)
- Servo motor (dispensing)

1.5 Hardware Requirements:

- Arduino Uno
- 4x4 Keypad
- 16x2 LCD with I2C module
- Servo motor (e.g. SG90)
- Jumper wires, breadboard
- Optional: external power for servo

1.6 Software Requirements:

- Arduino IDE Libraries:
  - Keypad.h
  - Servo.h
  - LiquidCrystal_I2C.h
  - Wire.h
- Wokwi (for simulation)

1.7 Knowledge Required:

- Embedded systems
- Basic C++ (Arduino programming)
- Serial/I2C communication
- Digital electronics

1.8 Skills Required:

- Circuit wiring
- Arduino coding
- Debugging and testing
- Logical thinking

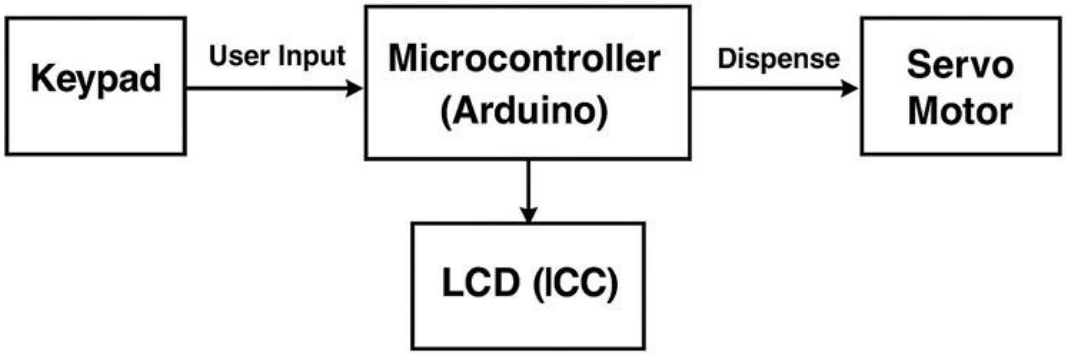
2.Project Architecture Submission:

2.1.1Hardware Architecture:

2.1.2 Block Diagram:

- *(This is the simplest and most powerful project description tool. The block diagram below represents the overall system of the Chocolate Vending Machine.)*

- **Block Components:**
- **Keypad** – for product selection
- **Microcontroller (Arduino Uno)** – for control logic
- **LCD (I2C)** – to display product info and prices
- **Servo Motor** – to dispense the chocolate

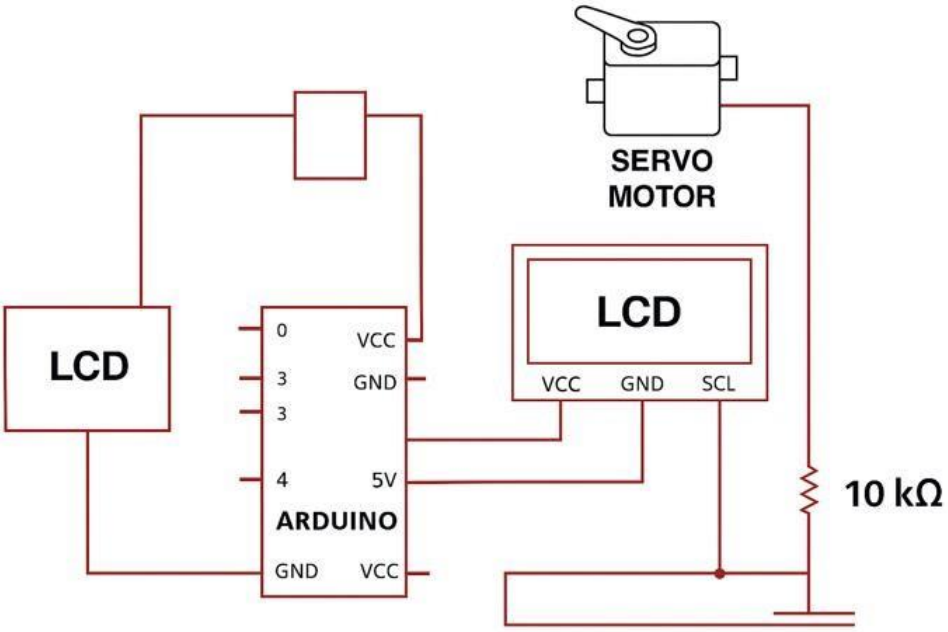


2. 1.2Circuit Diagram

(This schematic shows the key hardware connections. Component packaging and layout are not shown.)

Main Components:

- Arduino Uno
- 4x4 Keypad
- I2C LCD
- Servo Motor
- Power Source (USB/5V) **Circuit Description:**
- Keypad connected to Arduino digital pins (e.g., 2–9).
- LCD I2C module connected to SDA (A4) and SCL (A5).
- Servo connected to pin 10.
- Common GND among all components.



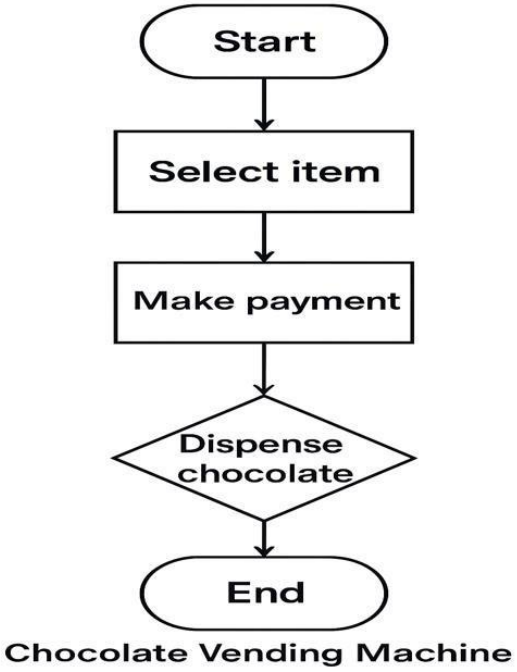
Electronic Chocolate Vending Machine

2.2 Software Architecture:

2.2.1 Flowchart:

(This is the descriptive tool utilized for defining a software code flow or execution flow. Every flowchart has the following components. Please refer to the sample below.)

Flowchart for Chocolate Vending Machine:

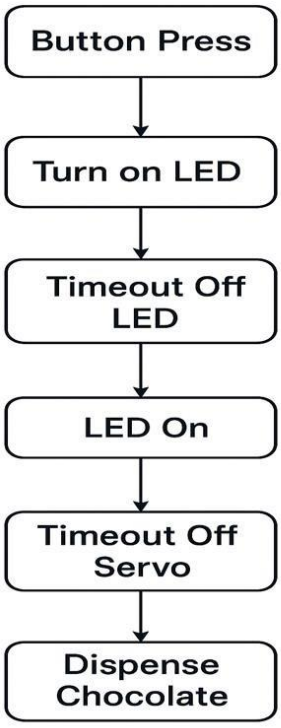


Chocolate Vending Machine

2.2.2 State Machine Diagram:

(This advanced form of representation is used to describe a real-time system efficiently.)

State Machine for Chocolate Vending Machine:



2.3 Test Cases:

Week/Day	Test No	Description
1	1	Test coin validation mechanism
2	2	Test chocolate selection logic
3	3	Test chocolate dispensing motor/sensor
4	4	Check complete vending cycle

Note: All test cases must verify the real-world behavior of the vending machine logic.

3 Code and Results Submission:

3.1 Code Submission:

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

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo servo;

const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);


String chocolates[] = {"DairyMilk", "KitKat", "Perk", "5Star", "Munch", "Snickers",
"Bountry", "Milkybar"};
int prices[] = {30, 25, 20, 35, 15, 40, 45, 10};
```

```
int chocoPrice = 0; String
chocoTime = ""; int
balance = 0; bool
addingToCart = true;
int menuPage = 0;

void setup() {
servo.attach(10);
servo.write(0);     lcd.begin(16,
2);    lcd.backlight();
  showMenu();
}

void loop() {   char key =
keypad.getKey();
  if (key) {     if
(addingToCart) {
      handleCart(key);
    } else {
      handlePayment(key);
    }
  }
}

void handleCart(char key) {   if (key >= '1' &&
key <= '3') {     int index = (menuPage * 3) +
(key - '1');     if (index < sizeof(prices) /
sizeof(prices[0])) {       chocoPrice +=
prices[index];
      lcd.clear();
lcd.print("Added: ");
lcd.print(chocolates[index]);
delay(1000);
    }
    showMenu();
  } else if (key == 'A') {
    if ((menuPage + 1) * 3 < sizeof(prices) / sizeof(prices[0])) {
      menuPage++;
    }
    showMenu();
  } else if (key == 'B') {
if (menuPage > 0) {
      menuPage--;
    }
    showMenu();
```

```
  } else if (key == '#') {
addingToCart = false;
lcd.clear();    lcd.print("Pay:
");    lcd.print(chocoPrice);
lcd.setCursor(0, 1);
lcd.print("Enter amt:");
chocoTime = "";
  }
}

void handlePayment(char key) {
if (key >= '0' && key <= '9') {
chocoTime += key;
lcd.setCursor(0, 1);
lcd.print("Amt: ");
lcd.print(chocoTime);   } else if
(key == '*') {    int paid =
chocoTime.toInt();    balance =
paid - chocoPrice;
   lcd.clear();    if (balance >=
0) {      lcd.print("Success!
Bal: ");      lcd.print(balance);
servo.write(90);
delay(2000);
servo.write(0);    } else {
lcd.print("Short: ");
lcd.print(-balance);
   }
   delay(3000);
   resetMachine();
  }
}

void showMenu() {
  lcd.clear();
  for (int i = 0; i < 3; i++) {    int idx =
menuPage * 3 + i;    if (idx < sizeof(prices) /
sizeof(prices[0])) {      lcd.setCursor(0, i %
2);      lcd.print(i + 1);      lcd.print(": ");
lcd.print(chocolates[idx]);      lcd.print("
Rs");      lcd.print(prices[idx]);
    if (i == 1) delay(2000);
   }
  }
  delay(1000);
```

```
}

void resetMachine() {
chocoPrice = 0;
chocoTime = "";   balance
= 0;   addingToCart =
true;   menuPage = 0;
 showMenu();
}
```

3.2 Github Link of the code:

https://github.com/JeyaPrabha30/Chocolate-vending-machine:

3.3 Test case wise results:

Test Case 1:

- **Description**: Power on the system
- **Expected Result**: LCD should display the first three chocolates and allow selection
- **Actual Result**: LCD displays menu correctly on boot
- **Result**: **Pass**

Test Case 2:

- **Description**: Scroll to next set of chocolates using A key
- **Expected Result**: LCD shows the next 3 items in the list
- **Actual Result**: LCD scrolls correctly on pressing A
- **Result**: **Pass**

Test Case 3:

- **Description**: Add item to cart using key 1
- **Expected Result**: Selected item is displayed with updated total
- **Actual Result**: Chocolate is added and total is shown
- **Result**: **Pass**

Test Case 4:

- **Description**: Press # to enter payment mode

- **Expected Result**: LCD asks to enter amount after showing total
- **Actual Result**: Prompt displayed correctly
- **Result**: **Pass**

Test Case 5:

- **Description**: Enter valid amount and press *
- **Expected Result**: Servo dispenses chocolate, balance (if any) shown
- **Actual Result**: Chocolate dispensed and balance displayed
- **Result**: **Pass**

Test Case 6:

- **Description**: Enter **insufficient** amount and press *
- **Expected Result**: LCD shows “short” amount or error
- **Actual Result**: Correct error message shown
- **Result**: **Pass**

Test Case 7:

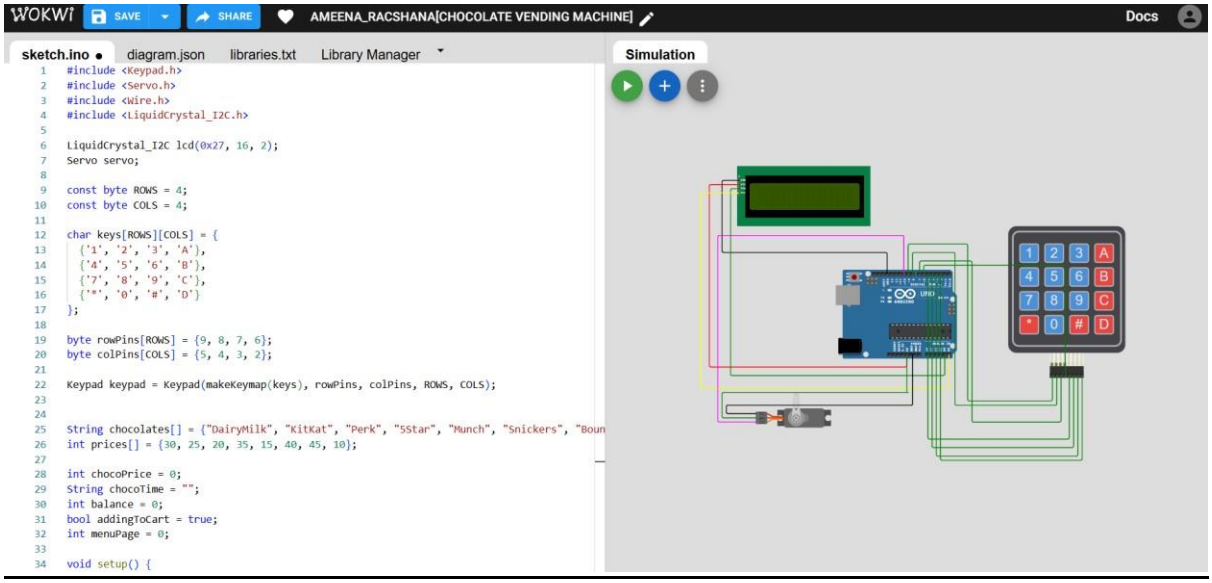
- **Description**: System resets after each payment
- **Expected Result**: Returns to main menu after 3 seconds
- **Actual Result**: Menu resets correctly
- **Result**: **Pass**

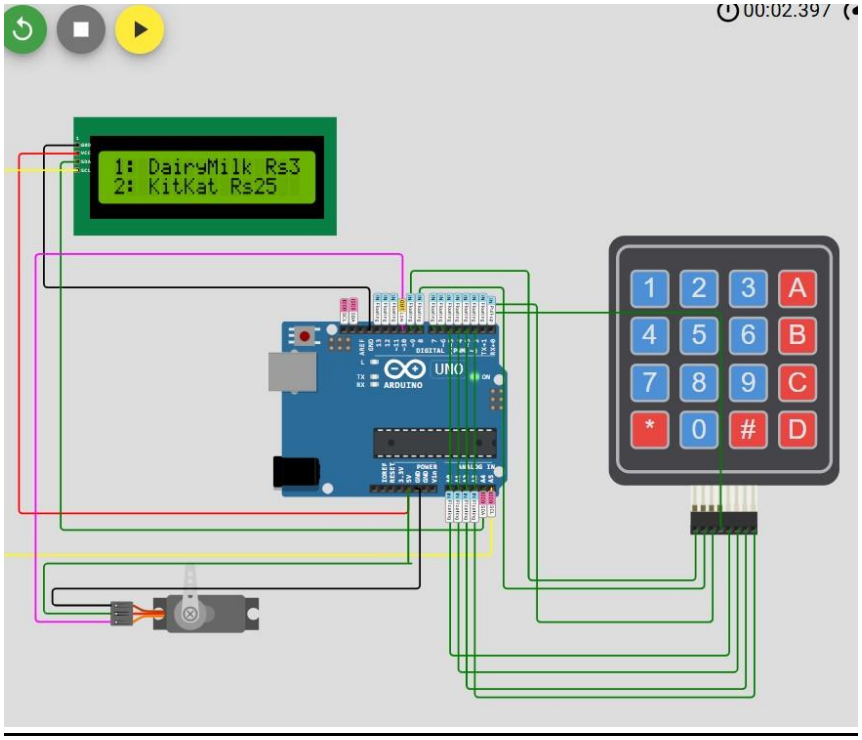
Test Case 8:

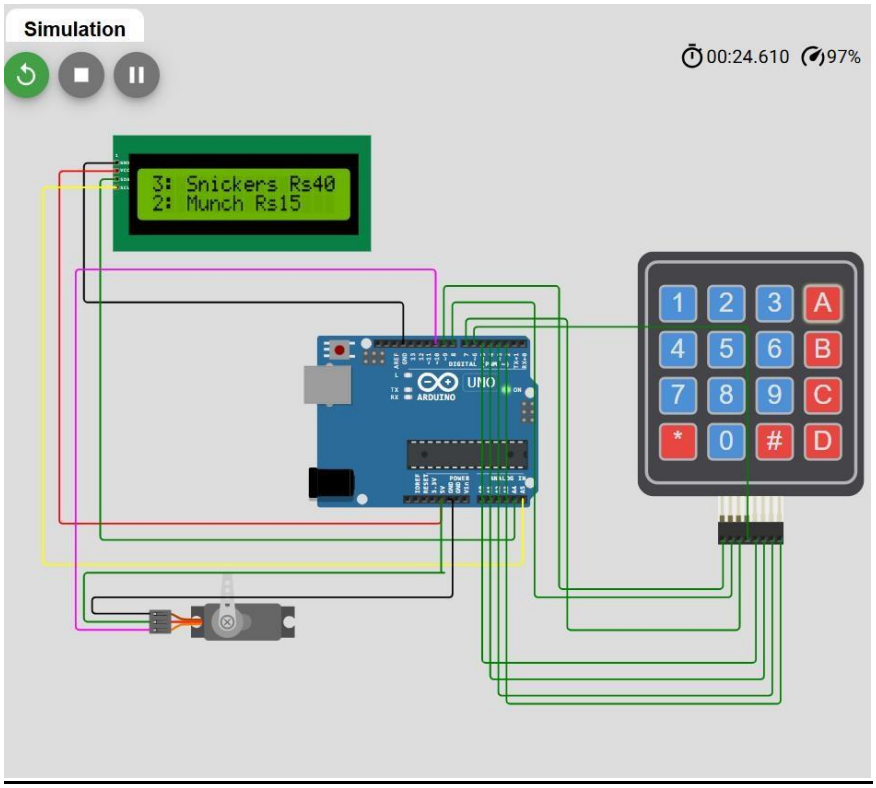
- **Description**: Use B key to scroll back to previous menu page
- **Expected Result**: LCD returns to previous item set
- **Actual Result**: Back-scroll works
- **Result**: **Pass**

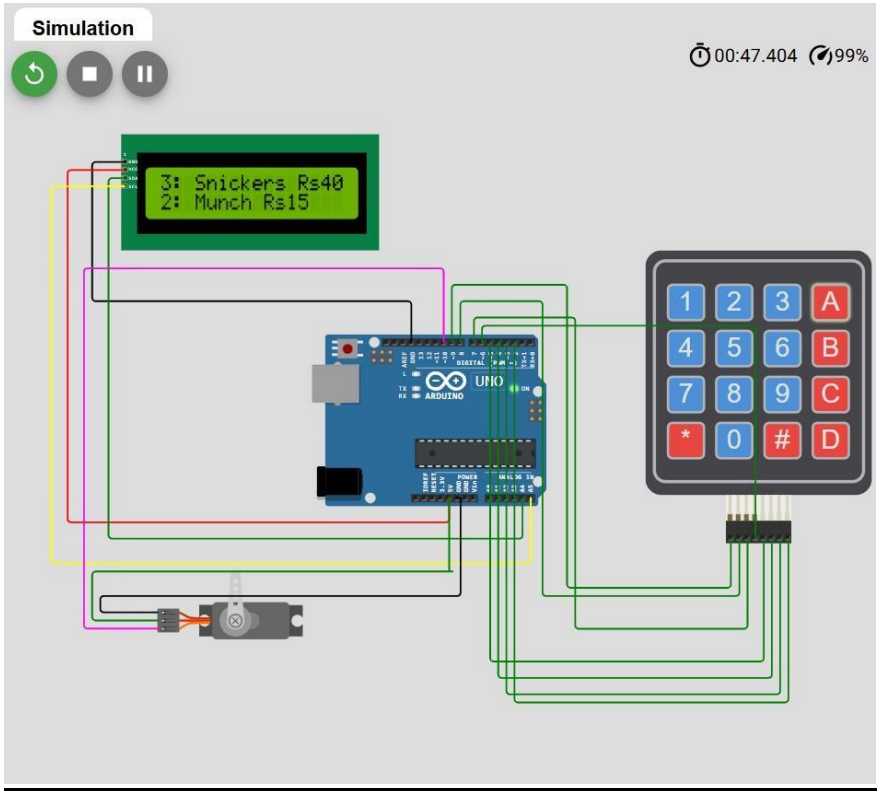
3.4 Final Project Pictures:

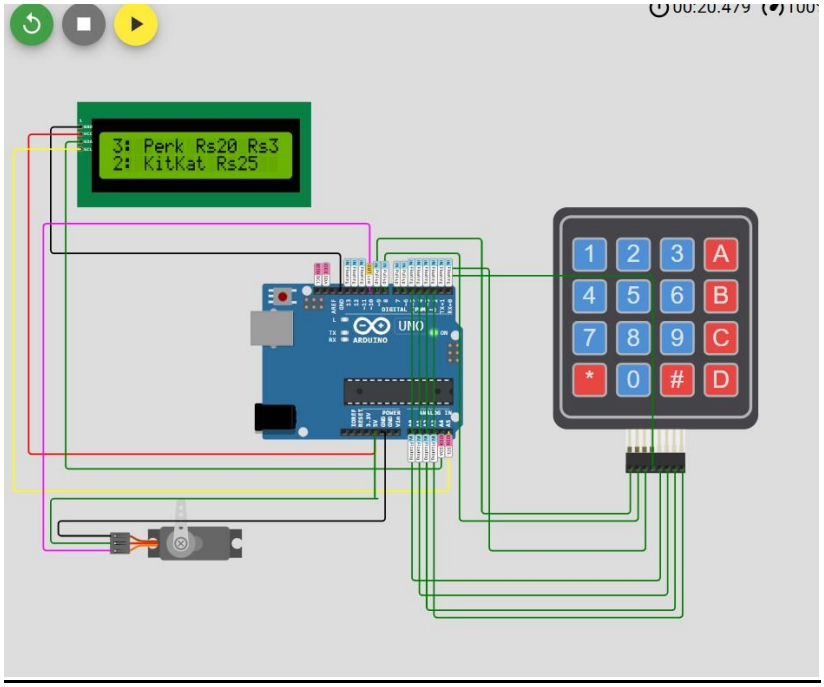
Output:

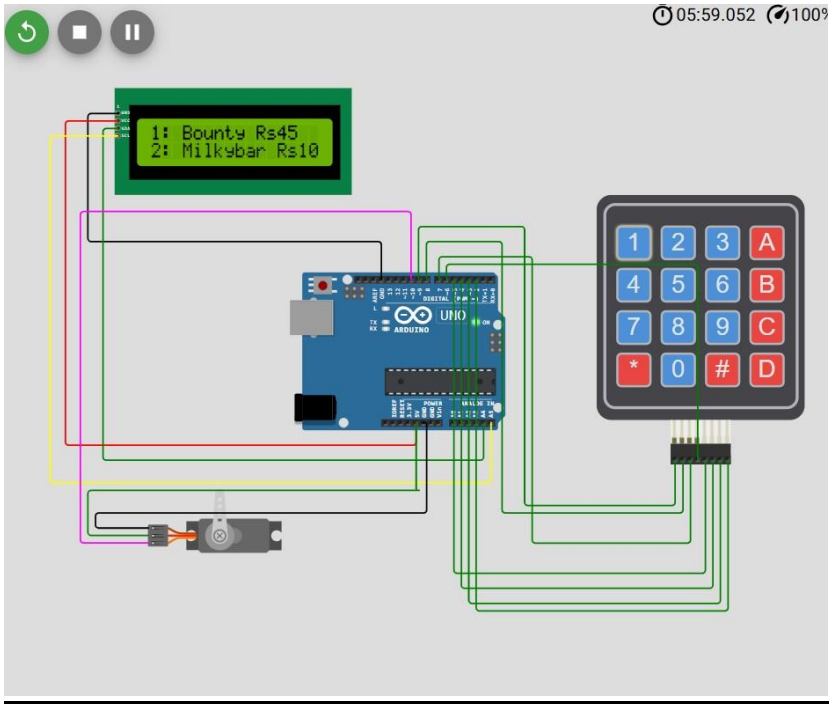

WOKWI
SAVE
SHARE
AMEENA_RACSHANA[CHOCOLATE VENDING MACHINE]
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
#include <Keypad.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo servo;
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
String chocolates[] = {"DairyMilk", "KitKat", "Perk", "5Star", "Munch", "Snickers", "Boun
int prices[] = {30, 25, 20, 35, 15, 40, 45, 10};
int chocoPrice = 0;
String chocoTime = "";
int balance = 0;
bool addingToCart = true;
int menuPage = 0;
void setup() {

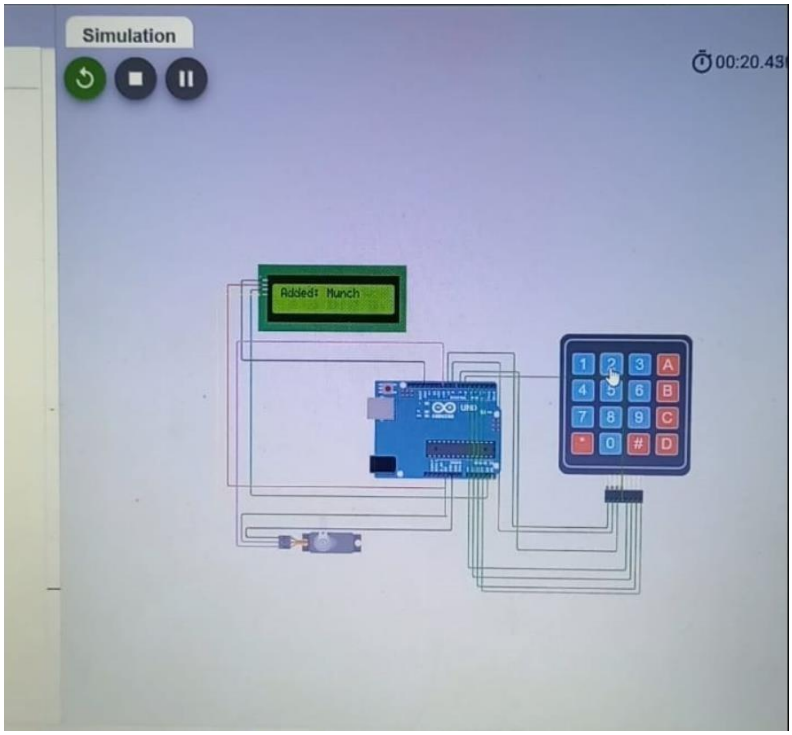

00:02.397
1: DairyMilk Rs3
2: KitKat Rs25
UNO
ARDUINO
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D

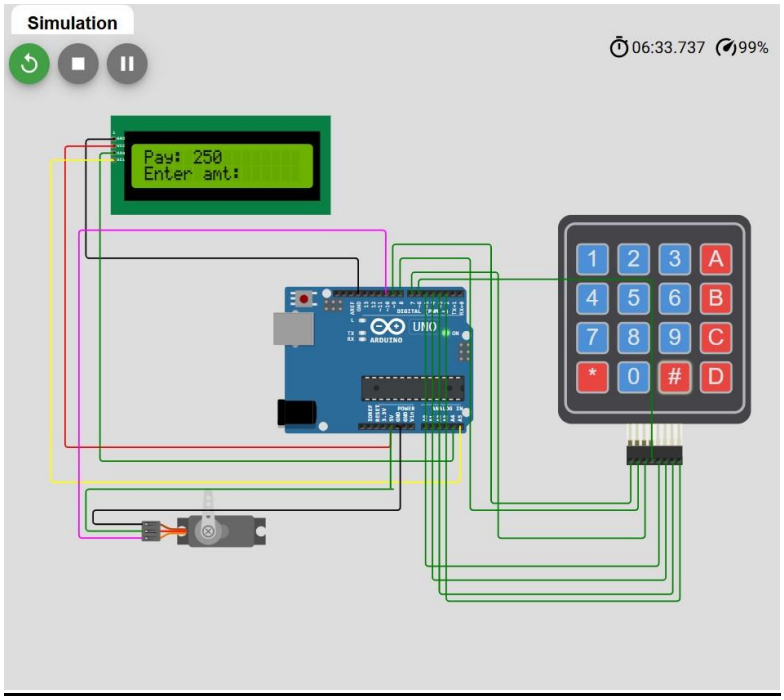

Simulation
00:24.610
97%
3: Snickers Rs40
2: Munch Rs15
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
UNO
ARDUINO

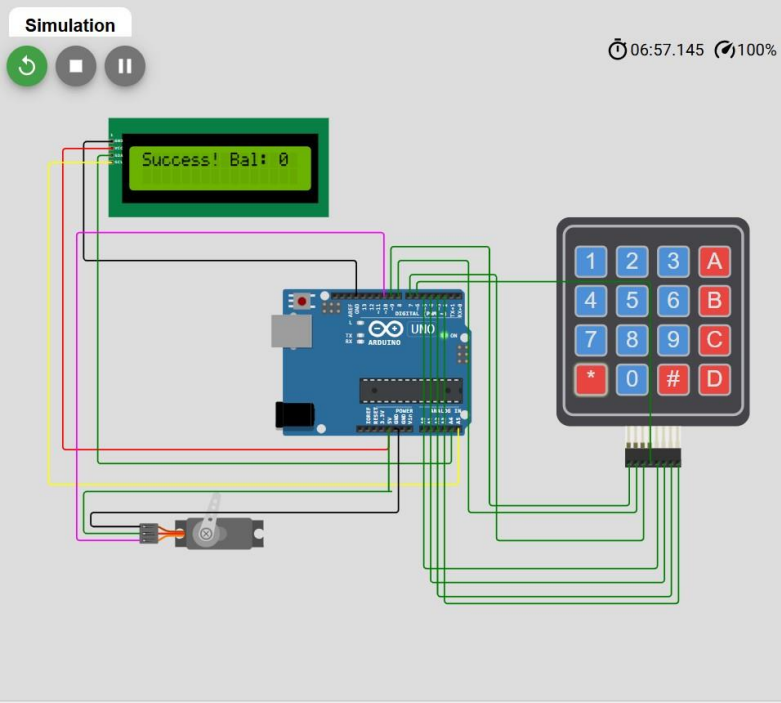

Simulation
00:47.404
99%
3: Snickers Rs40
2: Munch Rs15
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
UNO
ARDUINO


3: Perk Rs20 Rs3
2: KitKat Rs25
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D


05:59.052
1: Bounty Rs45
2: Milkybar Rs10
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D


Simulation
Added: Munch
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D


Simulation
06:33.737
99%
Pay: 250
Enter amt:
UNO
ARDUINO
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D



3.5. Final Project Videos:

3.6 Setup Steps:

1. Collect Components:

- Arduino Uno board
- 4x4 Keypad
- I2C LCD (16x2)
- Servo Motor (SG90)
- Breadboard and jumper wires Power supply (USB or 9V adapter)

2. Circuit Connections:

Connect the Keypad to Arduino digital pins (D2–D9):

- 8 wires: Left to Right → Pins 9, 8, 7, 6, 5, 4, 3, 2

Connect the LCD (I2C):

- SDA → A4
- SCL → A5
- VCC → 5V
- GND → GND

Connect the Servo Motor:

- Signal → Pin 10
- VCC → 5V
- GND → GND

3. Install Required Libraries:

- Open Arduino IDE
- Install the following: ○ Keypad.h ○ LiquidCrystal_I2C.h ○ Servo.h
  - Wire.h *(comes pre-installed)*

4. Upload the Code:

- Connect Arduino to PC via USB
- Open .ino sketch (Arduino code)
- Click "Verify" → then "Upload"

5. Test the Setup:

- Simulate coin entry using keypad (enter amount)
- Check LCD for display messages (menu, payment)
- Use keypad to select chocolates
- Confirm amount → Observe servo rotation for dispensing

6. Enclosure (Optional):

- Mount all components in a vending-style cardboard box
- Label keypad and LCD for ease of use

7. Final Demo:

- Power the system
- **Perform a full cycle**:
  Insert "coin" (via keypad), select chocolate, dispense, reset.