AUTOMATED MINI FRUIT JUICE DISPENSER

Berkan Alsancak (115202027),

Fatih Kıran (115200082),

Mehmet Öztop (116200082),

Berk Şardan (115202053),

Onur Üstün (115202120),

Cahit Kayın (115202075)

AUTOMATED MINI FRUIT JUICE DISPENSER

PROJECT REPORT

Group members: Berkan Alsancak (115202027), Fatih Kıran (115200082), Mehmet Öztop (116200082), Berk Şardan (115202053), Onur Üstün (115202120), Cahit Kayın (115202075)

Roles: Berkan- circuit & structural design, Berk- designing the motors for caps that hold the granulated powder, Fatih-coding & documentation, Mehmet-coding & documentation, Onur- designing the motors for caps that hold the granulated powder, Cahit- motor design & budget research

Aim: Is to create a fruit juice dispenser that uses powdered fruit juice and a water supply. The process of mixing the powder with the water is going to be automated with a mixer division and the finalized fruit juice will be automatically poured into a cup with a simple activation. Therefore, the aim is to end up with fruit juice without manually stirring the mixture and without manually pumping the juice.

TIMELINE

1st week- System design**(picture1)**: We draw the outline, came up with the possible appropriate materials for the project.

Research: estimated an average cost for the project, bought most of the necessary supplies.

2nd week-We designed the motors for the caps that hold the fruit powder, set up the system for the mixer, constituted the water pump. **(picture2)**

Coding: We tested the software to work with Arduino, came up with a set up for the control mechanisms.

3rd week- Building the dispenser**(picture3)**: we put the circuits together, we set up the motors for the caps using threads, connected every division with each other, put the parts in their proper spots in our box.

Testing: We ran test attempts of getting a cup of fruit juice from the dispenser, checked for errors like water leakage around the circuits, checked to see if there is dampening or any similar problems with regard to our powder.**(picture4)**

Finalizing the documents: making sure that they're in proper format, that there are no missing parts and that they're in proper order.

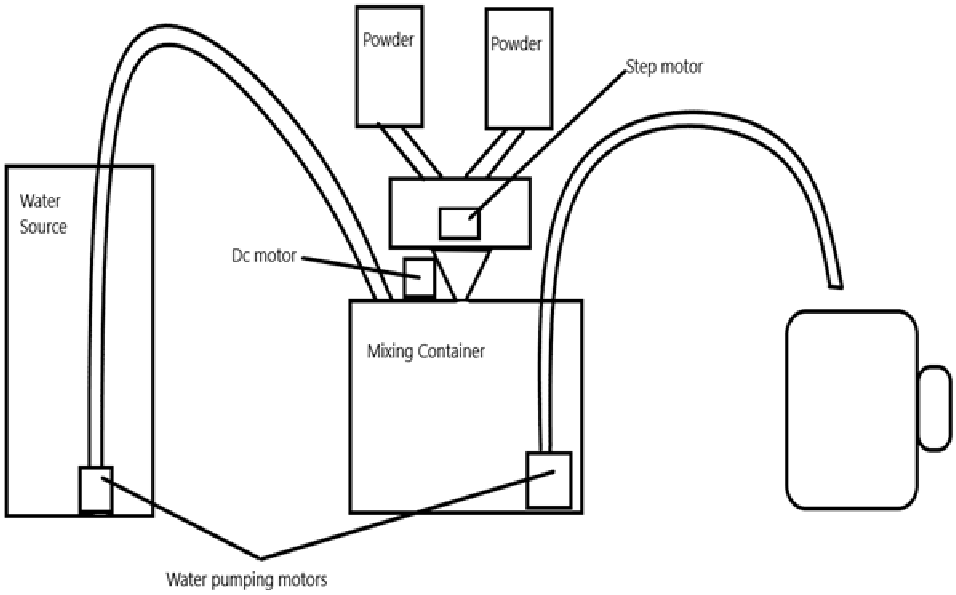
Improving: we've discussed bettering the dispenser with possible additions, and adding new features with regard to software.

STRUCTURE OF THE SYSTEM

We’ve got 2 containers made out of water bottles cut in half that hold the fruit powders, the first one has orange powder, the latter has peach. There is a step motor that controls the cap for the containers. When activated the step motor rotates in accordance with the button assigned to each of the containers. The selected fruit powder gets released into the mixer division.

We’ve got a water supply and a water pump installed in it. When the buttons are pressed, enough water gets pumped into the mixer before any powder is released.

There is a mixer division that uses a DC motor with a small fan attached to it. The motor is installed at the top of the mixing container. There is a second water pump inside the mixer that pours the fruit juice after it’s mixed. To activate/deactivate the mixer, you turn it’s switch on/off. There are led lights that signal when the switch of the mixer should be switched.

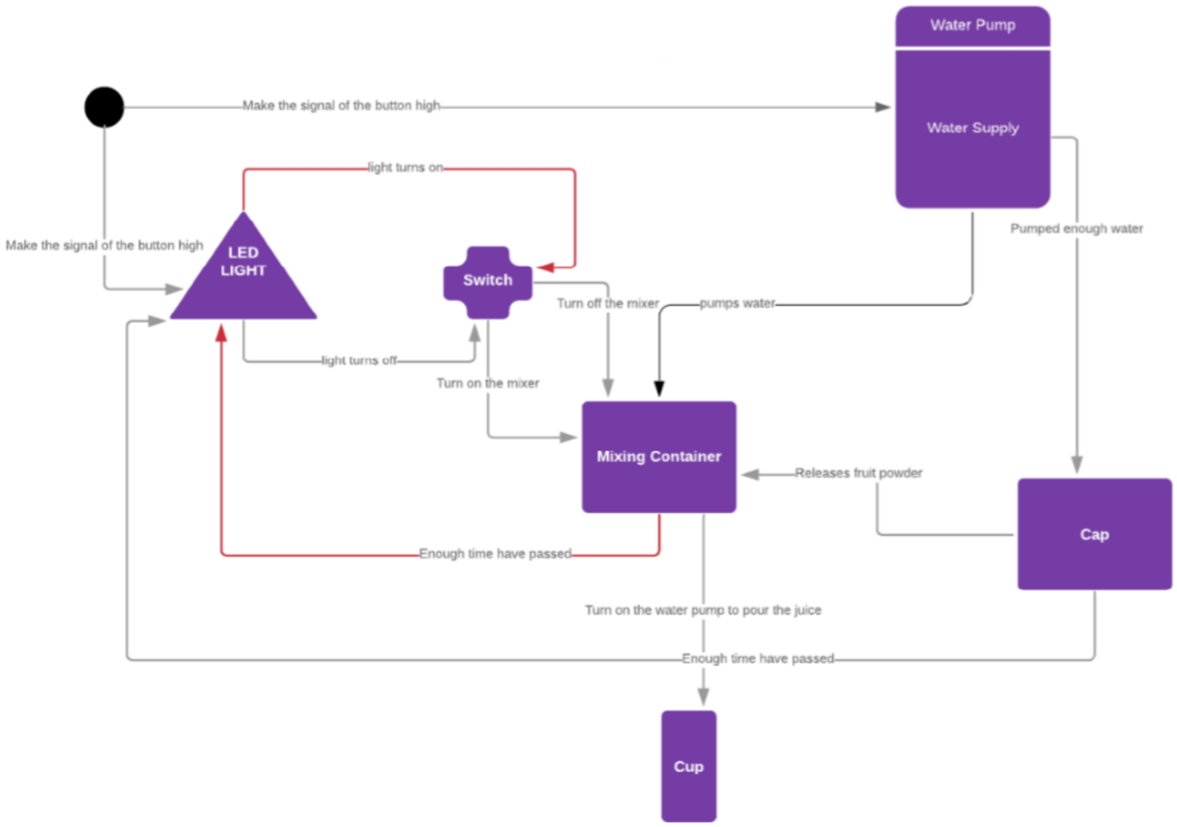


(picture1)

HARDWARE MATERIALS

1x Arduino Mega, 1x 5V step motor, 2x 6v DC motor, 1x 5v DC motor, 1x LED light, 2x push button, 1x switch , 1x Breadboard, jumper and copper cables, 1x DC motor driver, 2x 9V battery, 1x 1.5V battery.

UML STATE DIAGRAM



CODE

```
#include <X113647Stepper.h>
// Pinler
// Calisiyor
int step1[]={4,5,6,7};
static const int STEPS_PER_REVOLUTION = 32 * 64;
tardate::X113647Stepper Stepper1(STEPS_PER_REVOLUTION, 4, 5, 6, 7);
int sumotor1[]={9,10};
int sumotor2[]={11,12};
int ledControl[]={17,18};
int ldr[]={A0,A1};
int ledWarning[]={19,20};
int button[]={2,3};

void setup() {
  // put your setup code here, to run once:
  for (int i = 0 ; i < 2 ; i++)
  {
   pinMode(sumotor1[i], OUTPUT);
  }
  for (int i = 0 ; i < 2 ; i++)
  {
   pinMode(sumotor2[i], OUTPUT);
  }
  for (int i = 0 ; i < 2 ; i++)
  {
   pinMode(button[i], INPUT);
  }
  pinMode(19,OUTPUT);
  Stepper1.setSpeed(4);
}
```

```
void runDC(int pin1, int pin2, int runTime){
  // turn on the motor
  digitalWrite(pin1, HIGH);
  digitalWrite(pin2, LOW);
  //let the motor run
  delay(runTime);
  //turn off the motor
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, LOW);
}
void loop() {
  // put your main code here, to run repeatedly:
    if(digitalRead(2)==HIGH){
      runDC(sumotor1[0], sumotor1[1], 12000);
      Stepper1.step(7*STEPS_PER_REVOLUTION/10);
      digitalWrite(19,HIGH);
      delay(100);
      Stepper1.step(-7*STEPS_PER_REVOLUTION/10);
      delay(35000);
      digitalWrite(19,LOW);
      runDC(sumotor2[0], sumotor2[1], 19000);
      delay(500);
}
  if(digitalRead(3)==HIGH){
      runDC(sumotor1[0], sumotor1[1], 12000);
      Stepper1.step(-5*STEPS_PER_REVOLUTION/10);
      digitalWrite(19,HIGH);
      delay(100);
      Stepper1.step(5*STEPS_PER_REVOLUTION/10);
      delay(35000);
      digitalWrite(19,LOW);
      runDC(sumotor2[0], sumotor2[1], 19000);
      delay(500);
}}
```

UNUSED:

```
bool checkPowder(int btn) {
  if(btn== button[0]){
    digitalWrite(ledControl[0], HIGH);
    delay(500);
    if(analogRead(ldr[0]) < 100){
      digitalWrite(ledWarning[0], LOW);
      return true;
      digitalWrite(ledControl[0], LOW);
    }
    else{
      digitalWrite(ledWarning[0], HIGH);
      return false;
      digitalWrite(ledControl[0], LOW);
      for (int i = 0 ; i < 10 ; i++)
      {
       digitalWrite(ledWarning[0], LOW);
       delay(500);
       digitalWrite(ledWarning[0], HIGH); }}}
  else if(btn== button[1]){
    digitalWrite(ledControl[1], HIGH);
    delay(500);
    if(analogRead(ldr[1]) < 100){
      digitalWrite(ledWarning[1], LOW);
      return true;
      digitalWrite(ledControl[1], LOW);
    }
    else{
      digitalWrite(ledWarning[1],
HIGH);

      return false;

      digitalWrite(ledControl[1], LOW);

      for (int i = 0 ; i < 10 ; i++)

      {

       digitalWrite(ledWarning[1],
LOW);

       delay(500);

       digitalWrite(ledWarning[1],
HIGH);  }}}}

void loop() {

  // put your main code here, to run
repeatedly:

  if(digitalRead(button[0])==HIGH){

    if(checkPowder(button[0])){

      runDC(sumotor1[0], sumotor1[1],
sumotor1[2], 5000);


Stepper1.step(STEPS_PER_REVOLUTION);

      delay(5000);

      Stepper1.step(-
STEPS_PER_REVOLUTION);

      runDC(mixer[0], mixer[1],
mixer[2], 5000);

      runDC(sumotor2[0], sumotor2[1],
sumotor2[2], 5000);}}

  else
if(digitalRead(button[1])==HIGH){

    if(checkPowder(button[1])){

      runDC(sumotor1[0], sumotor1[1],
sumotor1[2], 5000);


Stepper2.step(STEPS_PER_REVOLUTION);

      delay(5000);

      Stepper2.step(-
STEPS_PER_REVOLUTION);
```

```
        runDC(mixer[0], mixer[1],
mixer[2], 5000);

        runDC(sumotor2[0], sumotor2[1],
sumotor2[2], 5000);}}}
```

FUNCTIONS:

runDC: Runs a DC motor for a desired amount of time

checkPowder: Turns on a led and checks the value at the light sensor to see if the container has powder or not. (UNUSED)

We couldn’t use the checkPowder function because we couldn’t set up the necessary light sensor.

The rest of the functions are from the Arduino library.

TESTS AND CRITICAL CASES

Critical cases:
The timing of the motors

Setting up the portions for the water and the fruit powder

Avoiding water leakage

No nasty particles at the end product

Test cases:

Initiating the water flow by filling the water supply

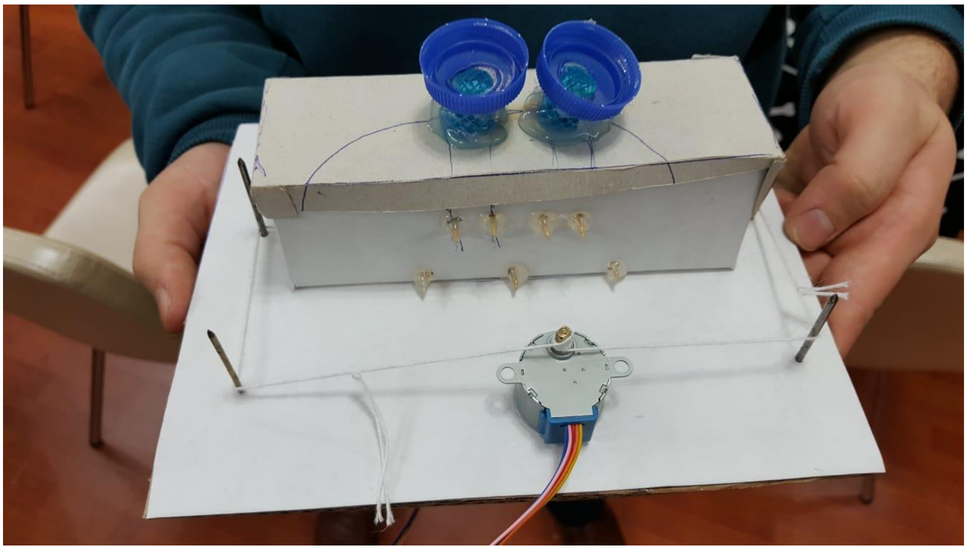
Checking out whether there are leakages on the flow

Starting to pour the powder in case there is no water leakage

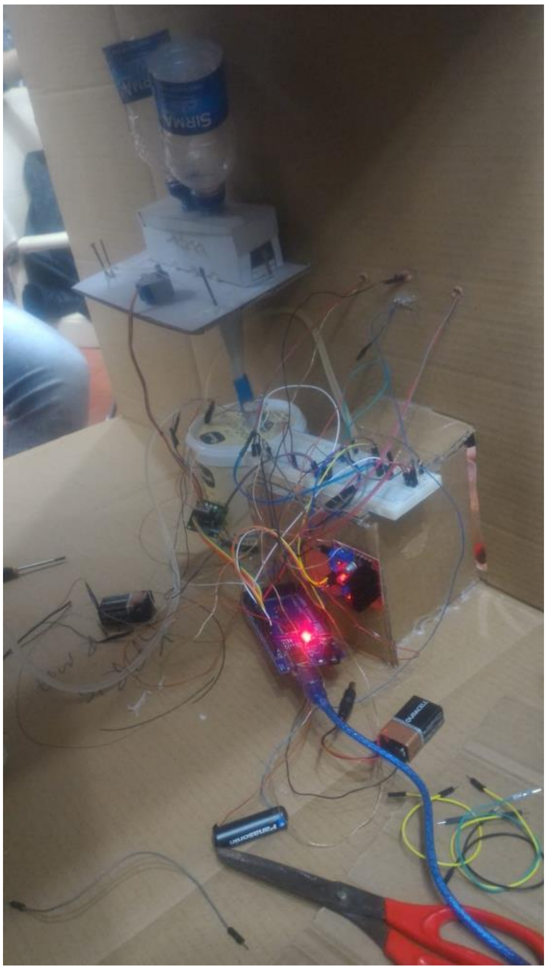
Checking out if the powder is getting released

Trying to run the mixer to see if the powder is getting mixed into the water

Examining whether there is a problem on pouring the juice from the mixer

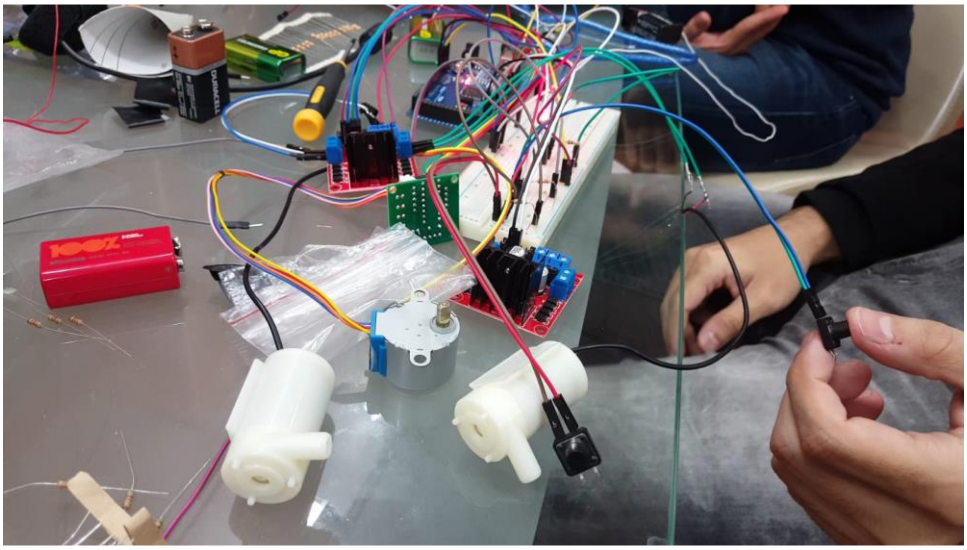


Powder Control Box**(picture2)**

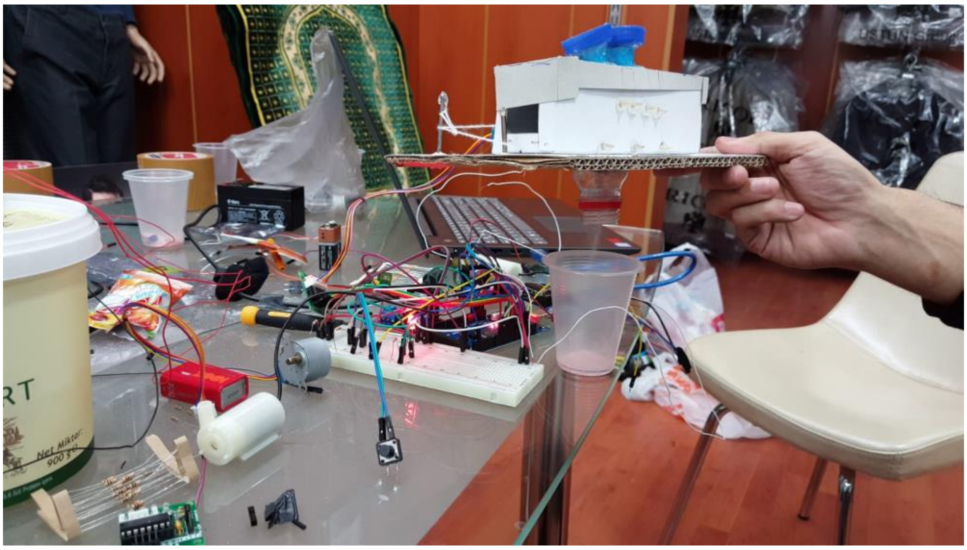


Inside of Machine**(picture3)**

Outside of Machine**(picture3)**



Test process of water pumping motors**(picture2)**



Test process of powder pour control unit **(picture4)**