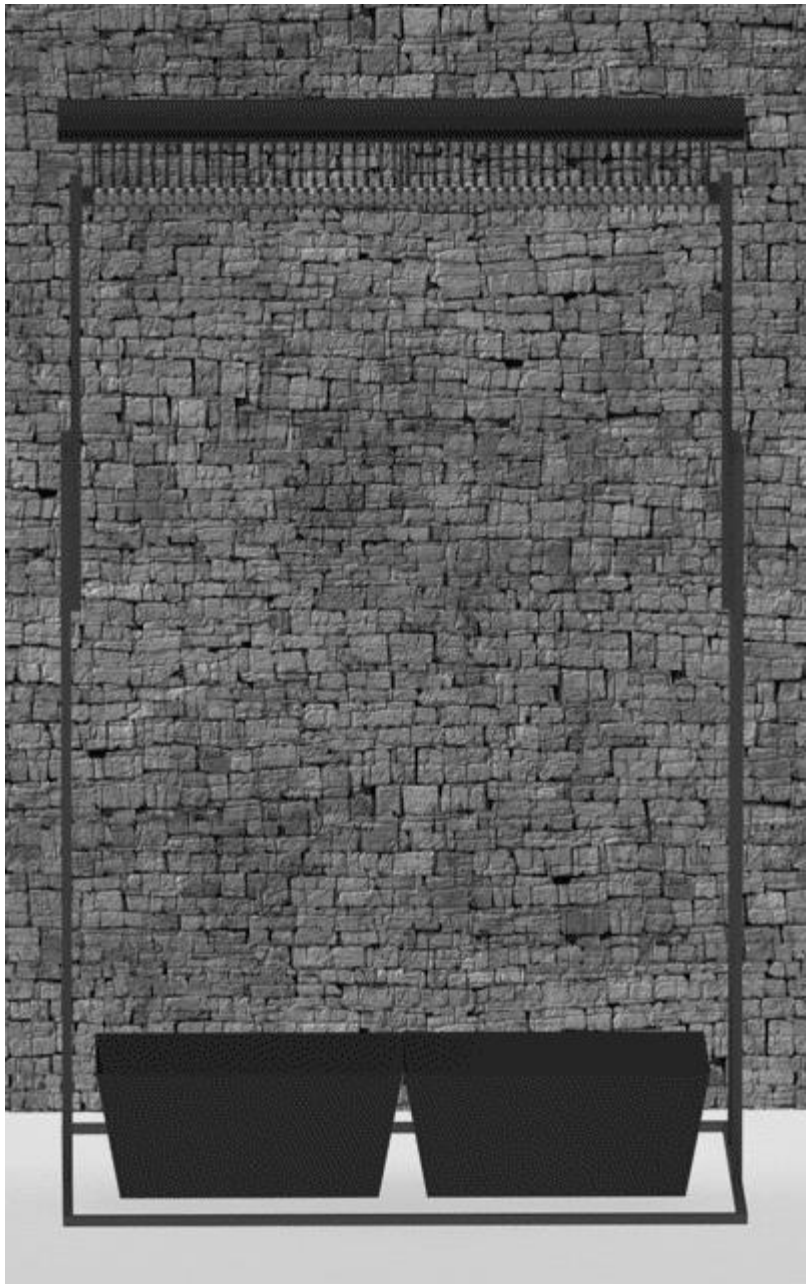


# Graphical Waterfall

## Prototype

By
Farzad Azizi Zade

adio | Arduino 1.8.10

File Edit Sketch Tools Help

adio

```
      /* s=400 roundtrip example function (returns the input)*/

      case 400:
      /* the second value (val) can really be anything here  */

      /* This is an auxiliary function that returns the ASCII
         value of its first argument. It is provided as an
         example for people that want to add their own code  */

      /* your own code goes here instead of the serial print */
      Serial.println(val);

      s=-1;  /* we are done with the aux function so -1      */
      break; /* s=400 taken care of                          */



      /* ******* UNRECOGNIZED STATE, go back to s=-1 ******* */

      default:
      /* we should never get here but if we do it means we
         are in an unexpected state so whatever is the second
         received value we get out of here and back to s=-1  */

      s=-1;  /* go back to the initial state, break unneeded */


    } /* end switch on state s                               */

  } /* end if serial available                              */

} /* end loop statement                                     */
```

Done compiling.

```
Sketch uses 2948 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 192 bytes (9%) of dynamic memory, leaving 1856 bytes for local variables. Maximum is 2048 bytes.
```

263 Arduino/Genuino Uno on COM3

# ADIO Code

**First, we need to upload this code on an Arduino board to control it by using MATLAB**

## On / Off Controller

## Code in MATLAB

```
clear all
clc
board=arduino('COM3')
for j=1:7
for j=2:15;
pinMode(board,j,'OUTPUT');
end
binaryimg=[کد باینری مورد نظر];

for i=length(binaryimg):-1:1;
    digitalWrite(board,2,abs(binaryimg(i,13)-1));
    digitalWrite(board,3,abs(binaryimg(i,12)-1));
    digitalWrite(board,4,abs(binaryimg(i,11)-1));
    digitalWrite(board,5,abs(binaryimg(i,10)-1));
    digitalWrite(board,6,abs(binaryimg(i,9)-1));
    digitalWrite(board,7,abs(binaryimg(i,8)-1));
    digitalWrite(board,8,abs(binaryimg(i,7)-1));
    digitalWrite(board,9,abs(binaryimg(i,6)-1));
    digitalWrite(board,10,abs(binaryimg(i,5)-1));
    digitalWrite(board,11,abs(binaryimg(i,4)-1));
    digitalWrite(board,12,abs(binaryimg(i,3)-1));
    digitalWrite(board,13,abs(binaryimg(i,2)-1));
    digitalWrite(board,14,abs(binaryimg(i,1)-1));
    digitalWrite(board,15,abs(binaryimg(i,14)-1));
    pause(0.1);
end
end
```

## Binary Images

00000000000000
00000111111000
00011100011110
00111000000110
01110000000000
01110000000000
01110000000000
01110000000000
01110000000000
00111000000110
00011100011110
00000111111000
00000000000000

It must be equal to the number of the solenoid valves, which is 14

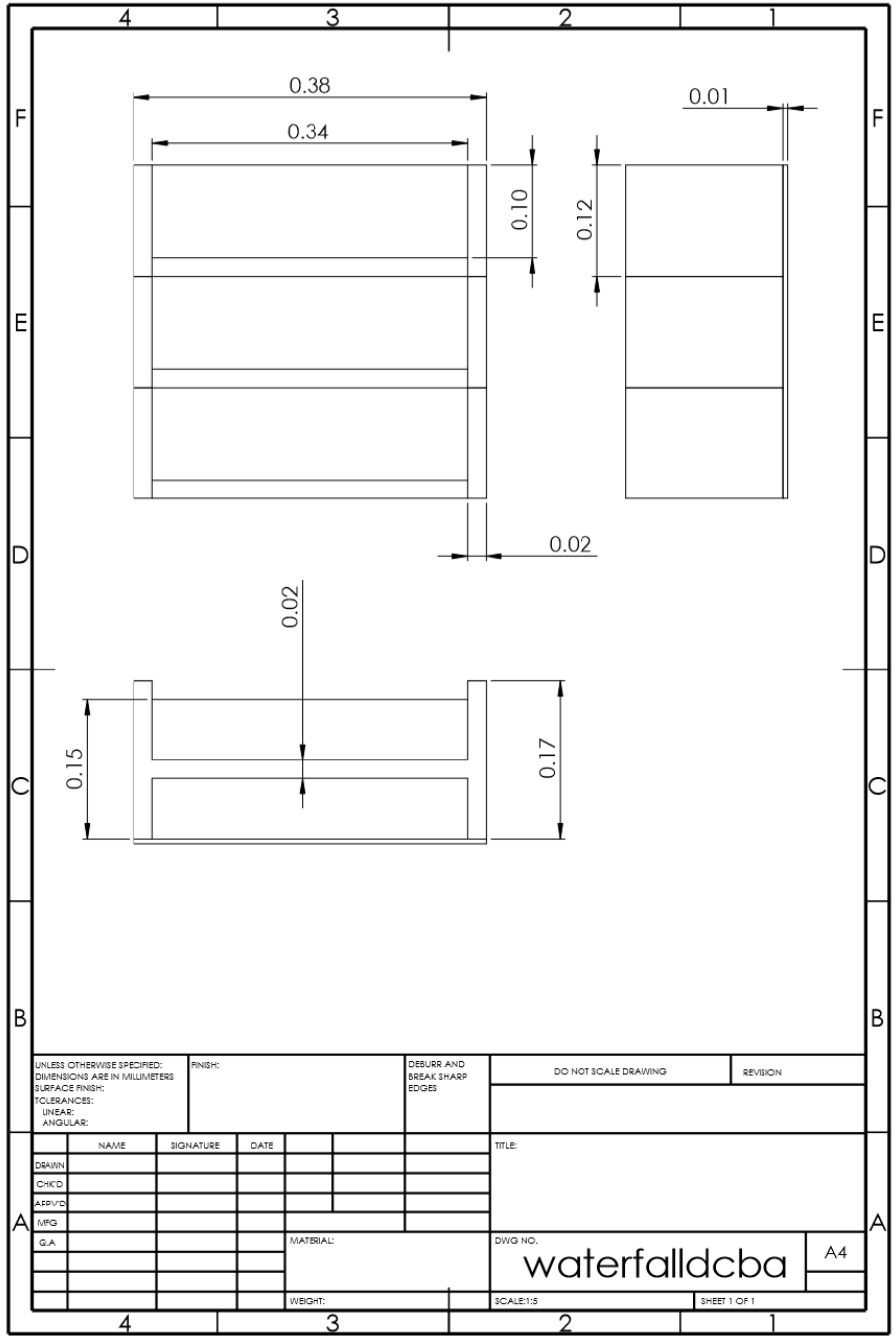
00000000000000
00000111111000
00011100011110
00111000000110
01110000000000
01110000000000
01110000000000
01110000000000
01110000000000
00111000000110
00011100011110
00000111111000
00000000000000

00000000000000
00000011000000
00000111100000
00001100110000
00001100110000
00011000011000
00111000011100
00 1111111111 00
01110000001110
01100000000110
01100000000110
01100000000110
00000000000000

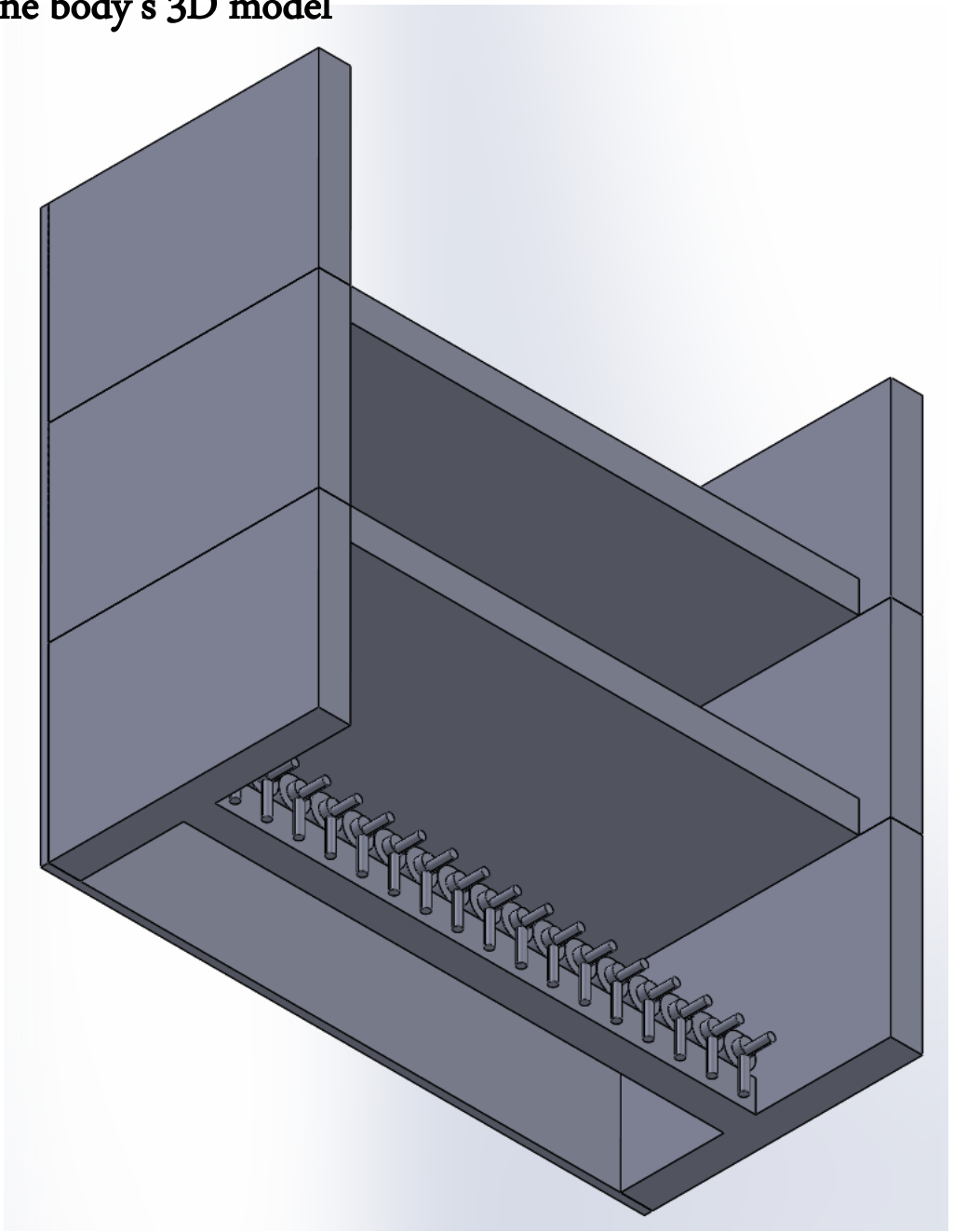
00000000000000
01111111100000
01110001111000
01110000011100
01110000001110
01110000001110
01110000001110
01110000001110
01110000001110
01110000011100
01110001111000
01111111100000
00000000000000

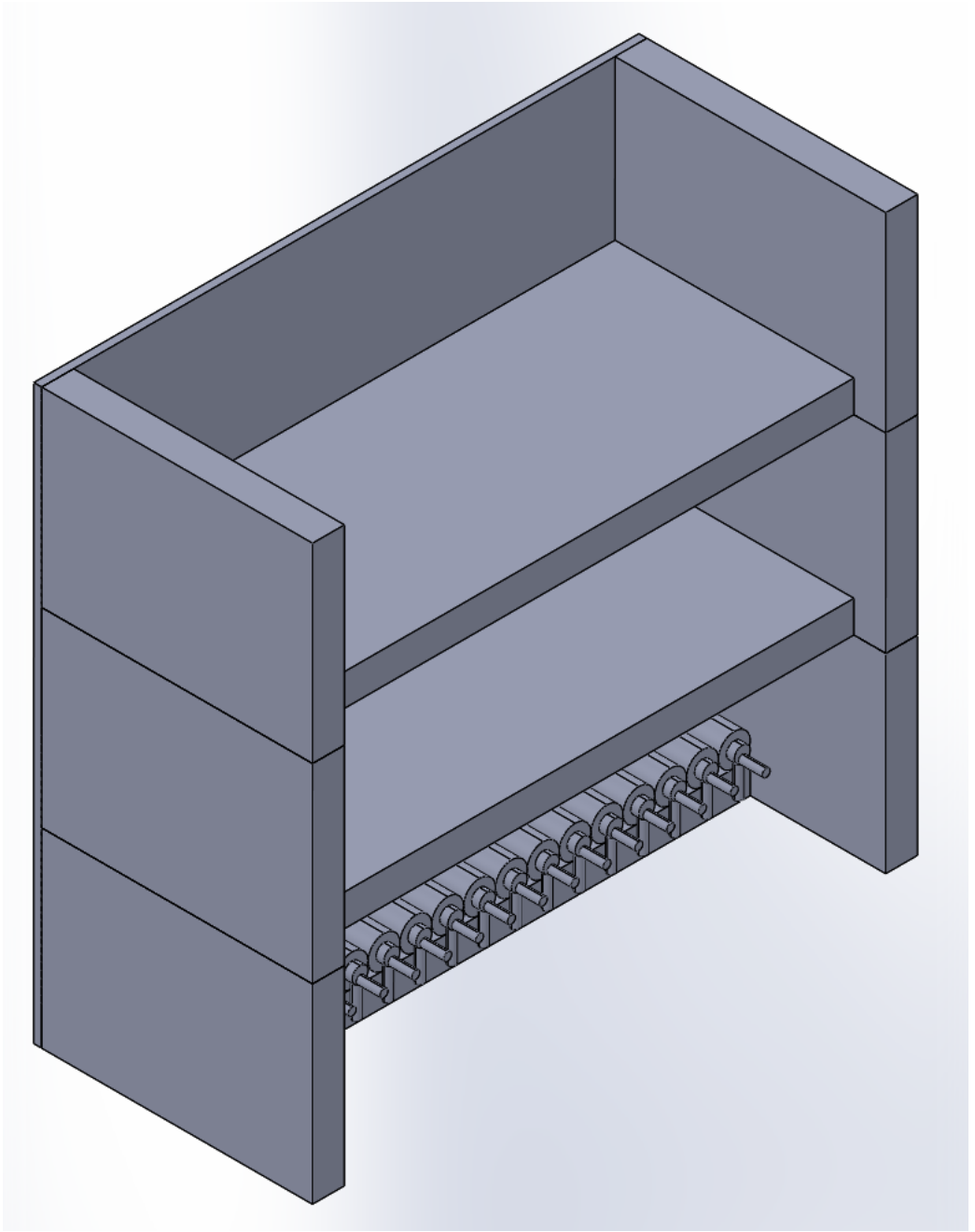
The number of these lines and the reaction speed of the valves determine the required height to install the machine.

# Machine body's 2D drawing

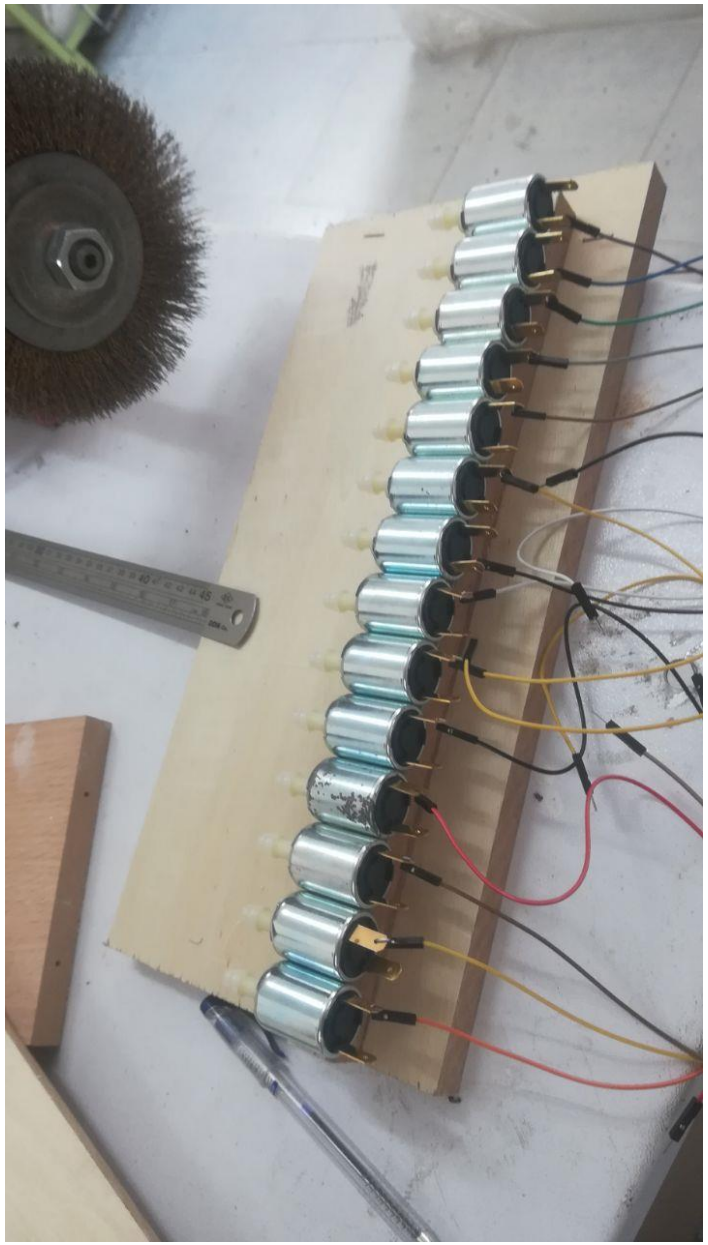


# Machine body's 3D model

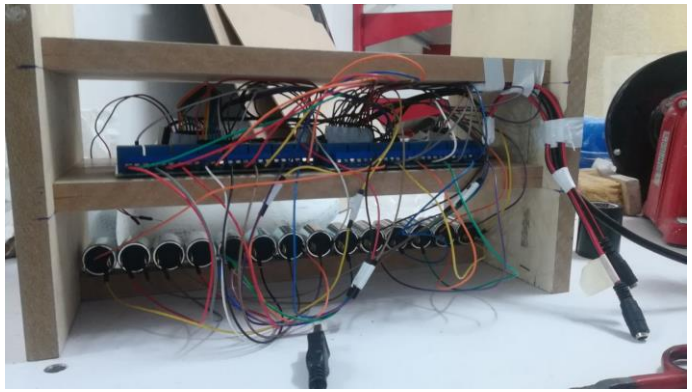


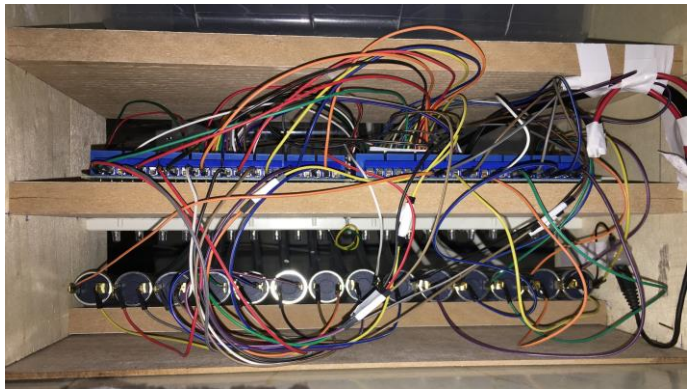


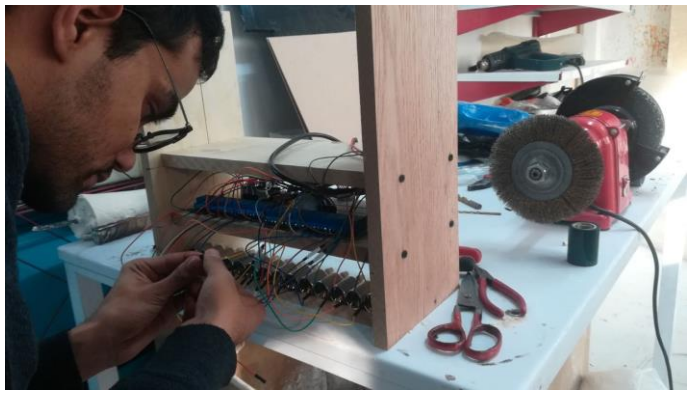
# Construction stages

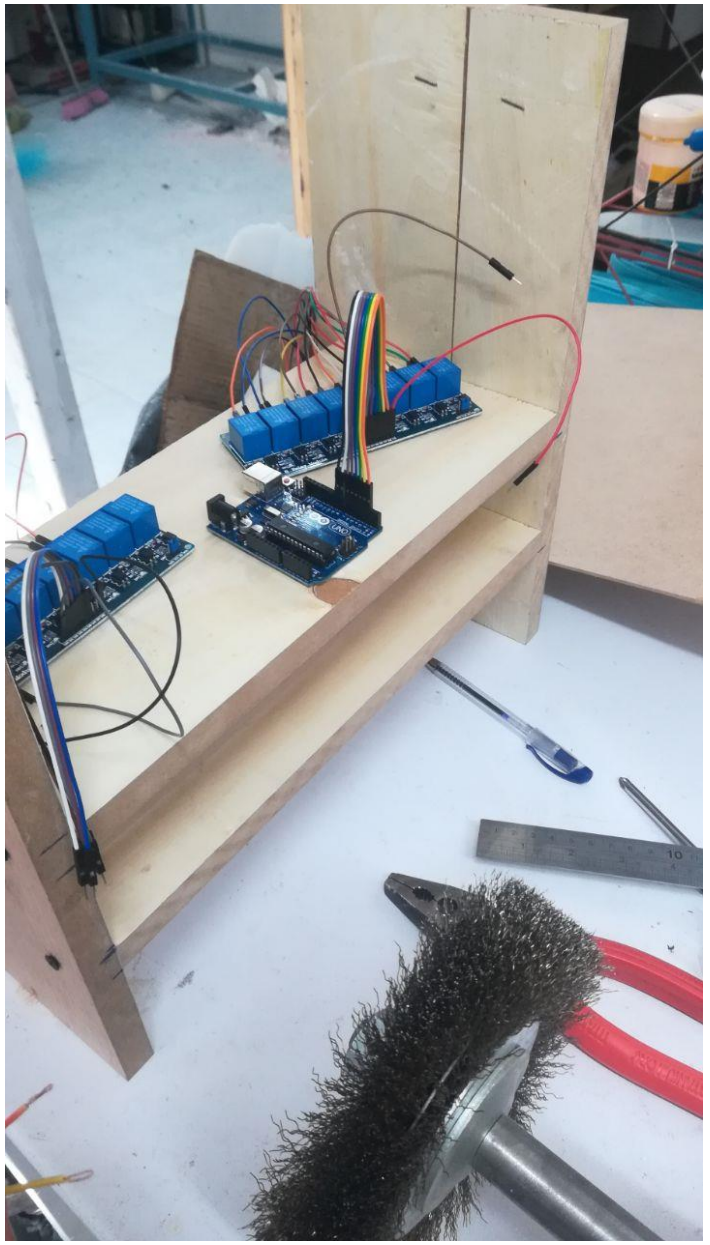


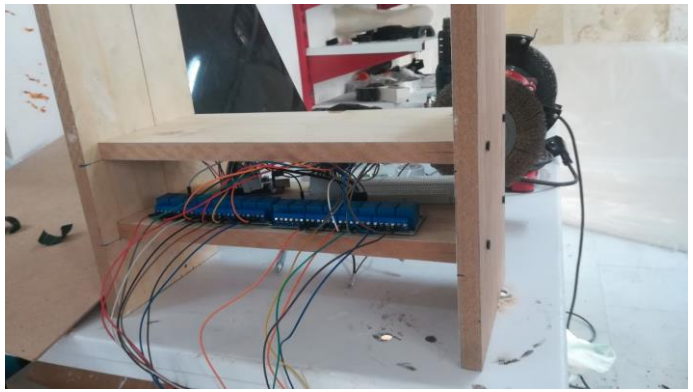
# Construction – wiring and installing electronic components

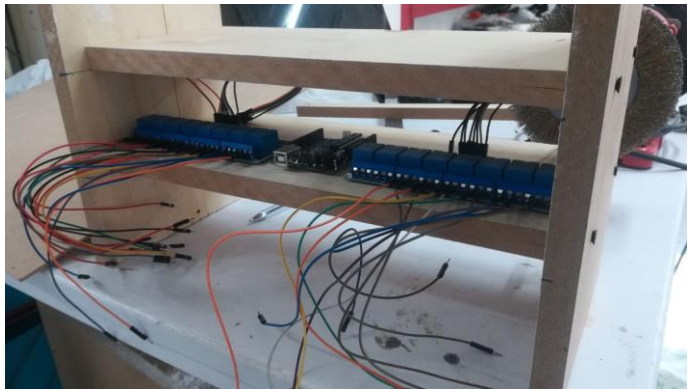


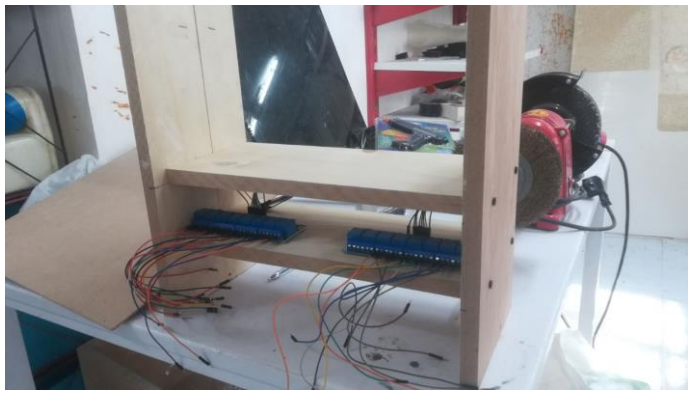




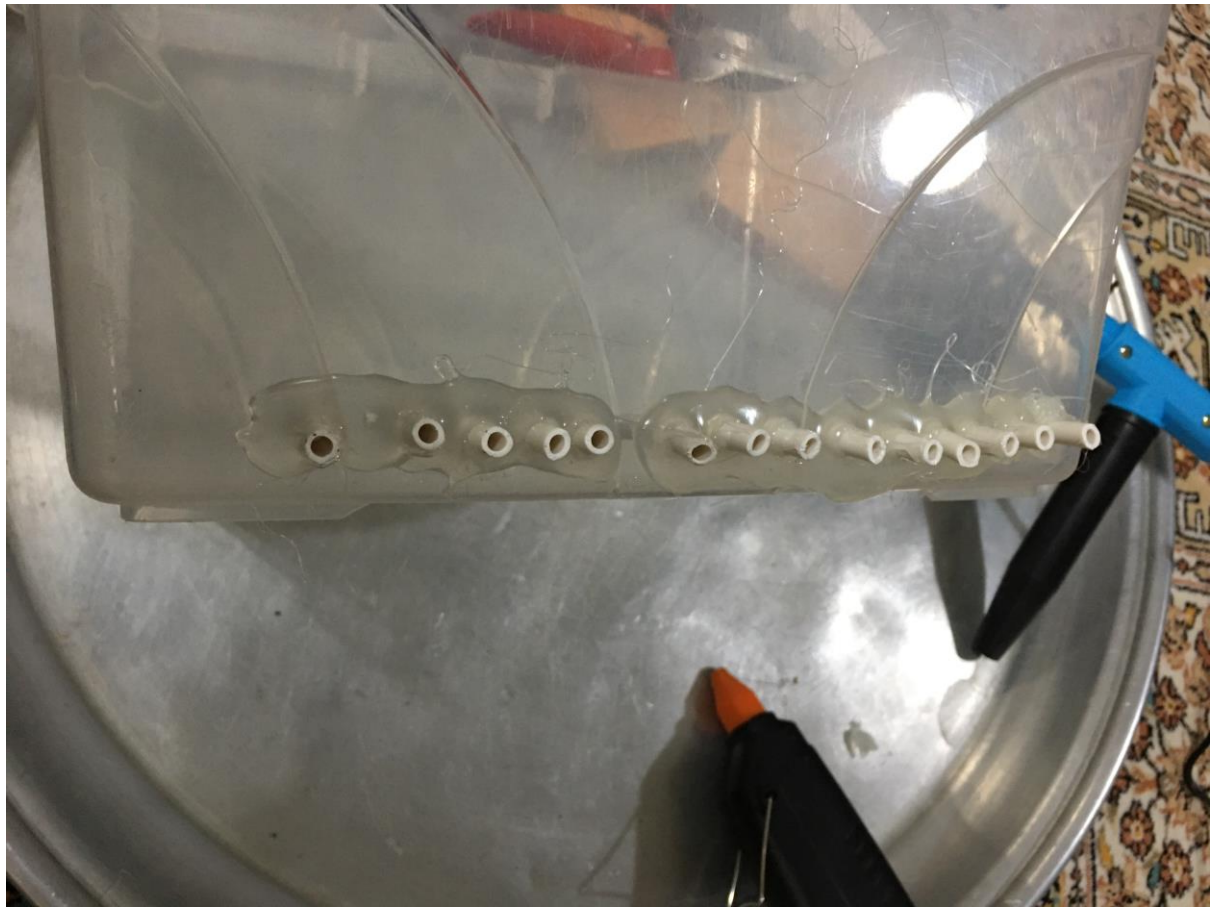




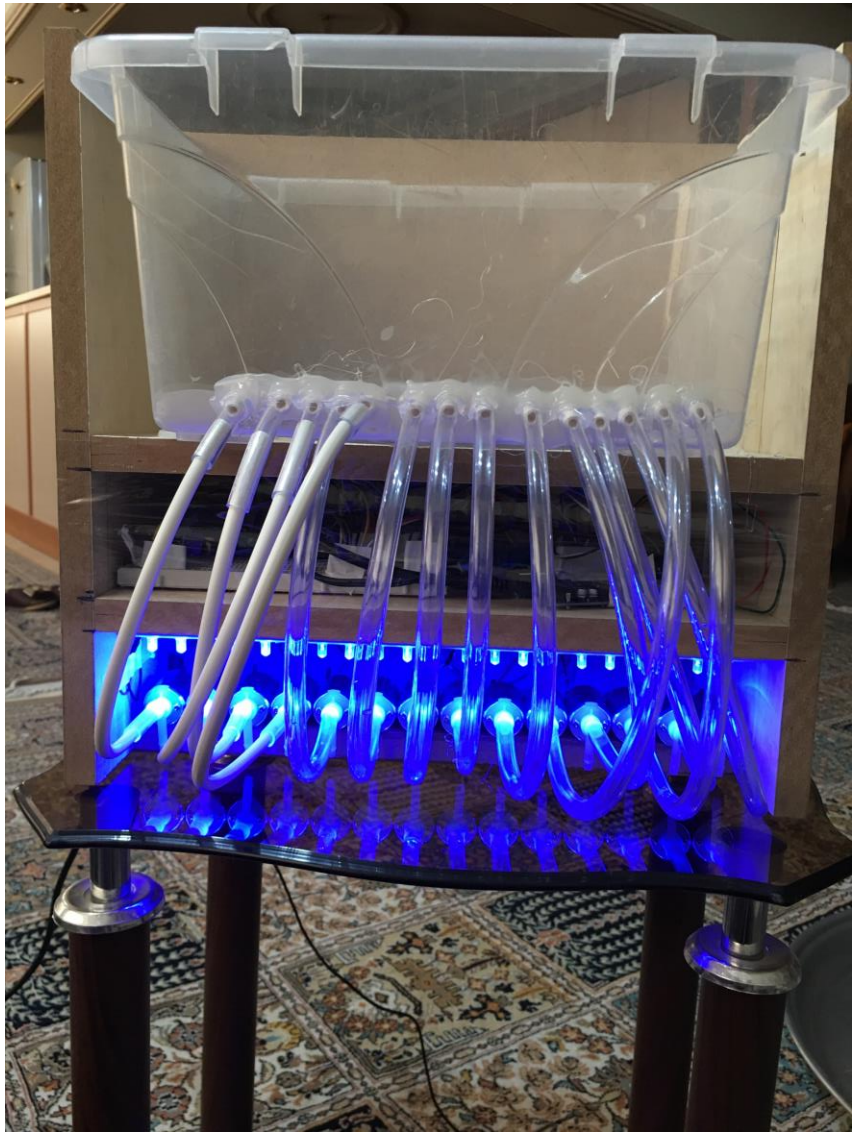


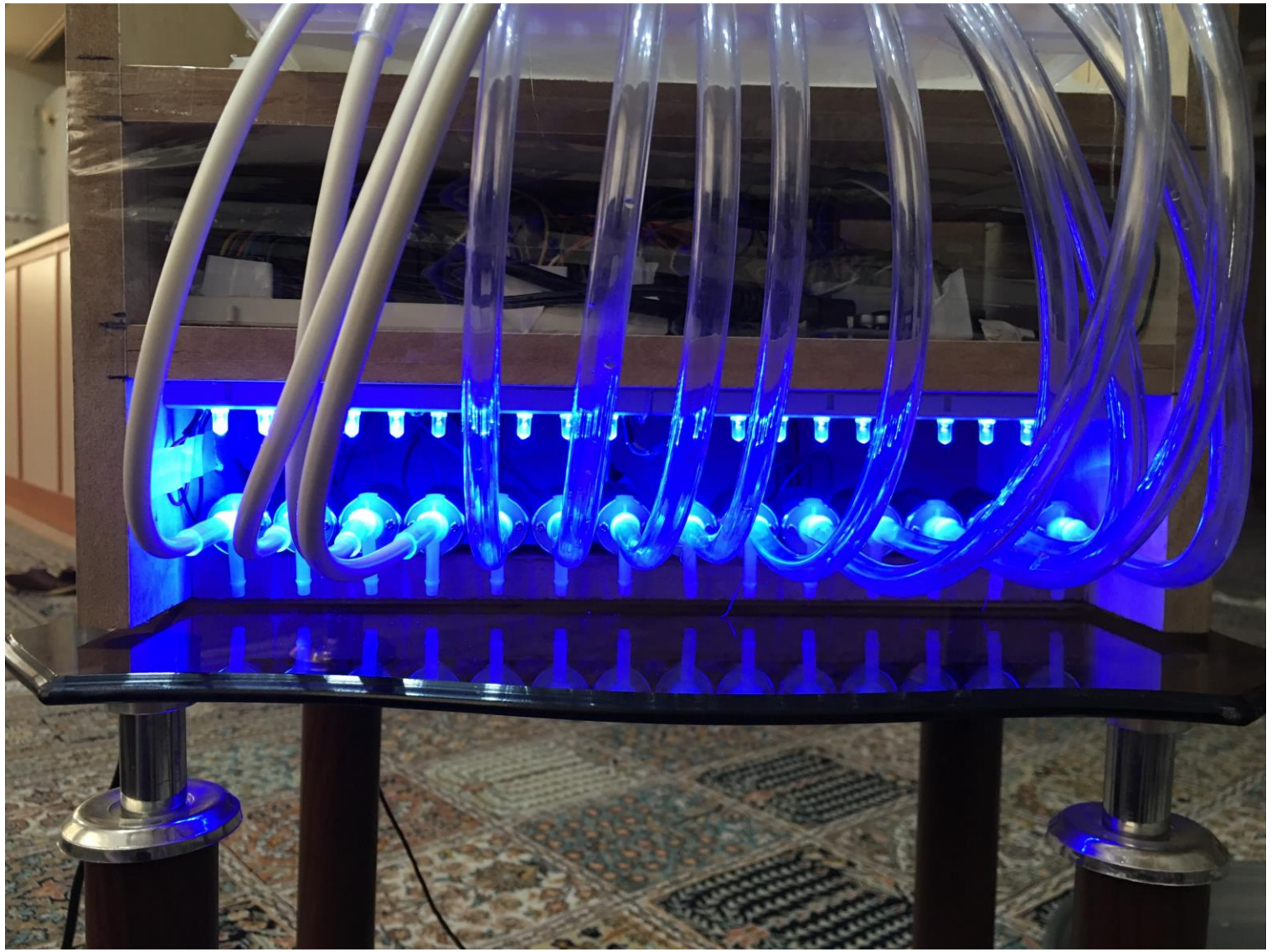


# Construction – installing the tank and the pipes



# Construction – installing and testing LEDs





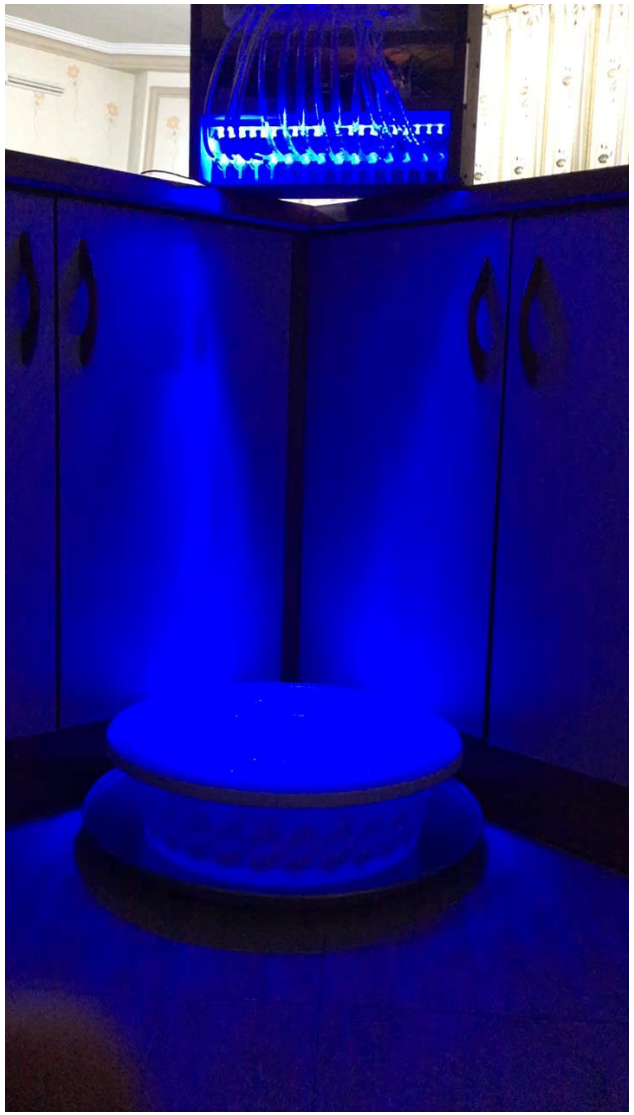

# The power supply

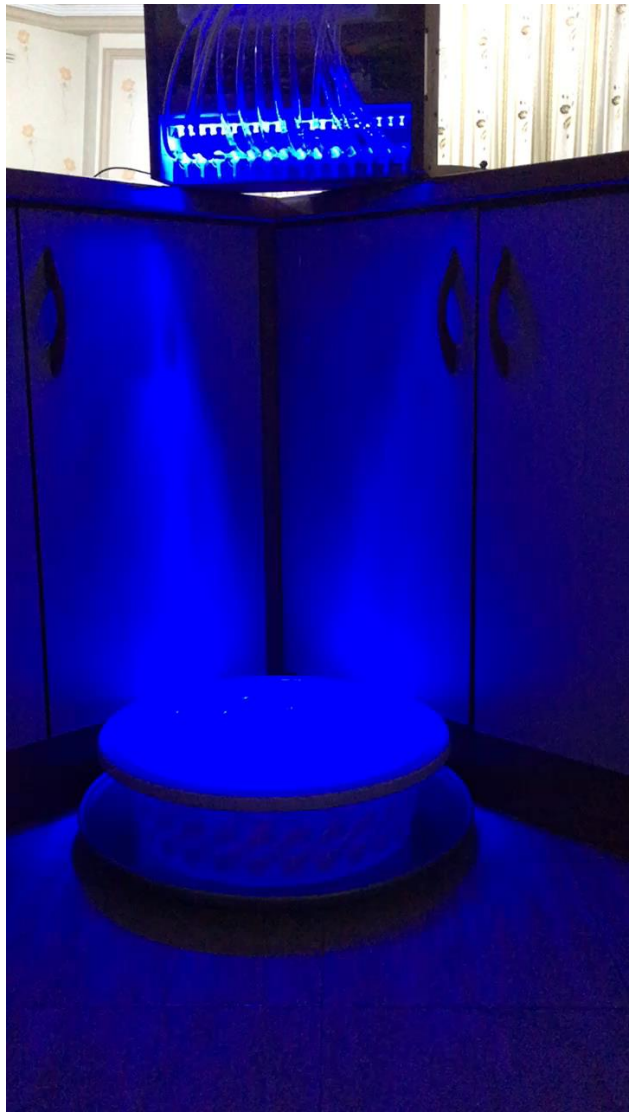
Three power supplies are needed to run the machine:

- A 19 V and 6 Amp for solenoid valves
- 3.3 to 3.7 V and 0.37 Amp for LEDs
- 7 to 12 V for Arduino, you can also use a USB cable

To control the machine using MATLAB, you have to connect it to a computer with a USB cable.

# Machine performance clips





## The testing collector

We used this collector for the first run and testing solenoid valves. Because it leaked, we replaced it with a homemade tank.