Automatic Waste Sorting Machine

By Zemerart_Asani (/member/Zemerart_Asani/) in Circuits (/circuits/) > Robots (/circuits/robots/projects/) 5,378 9 18

Published Dec 29th, 2019 Download Favorite

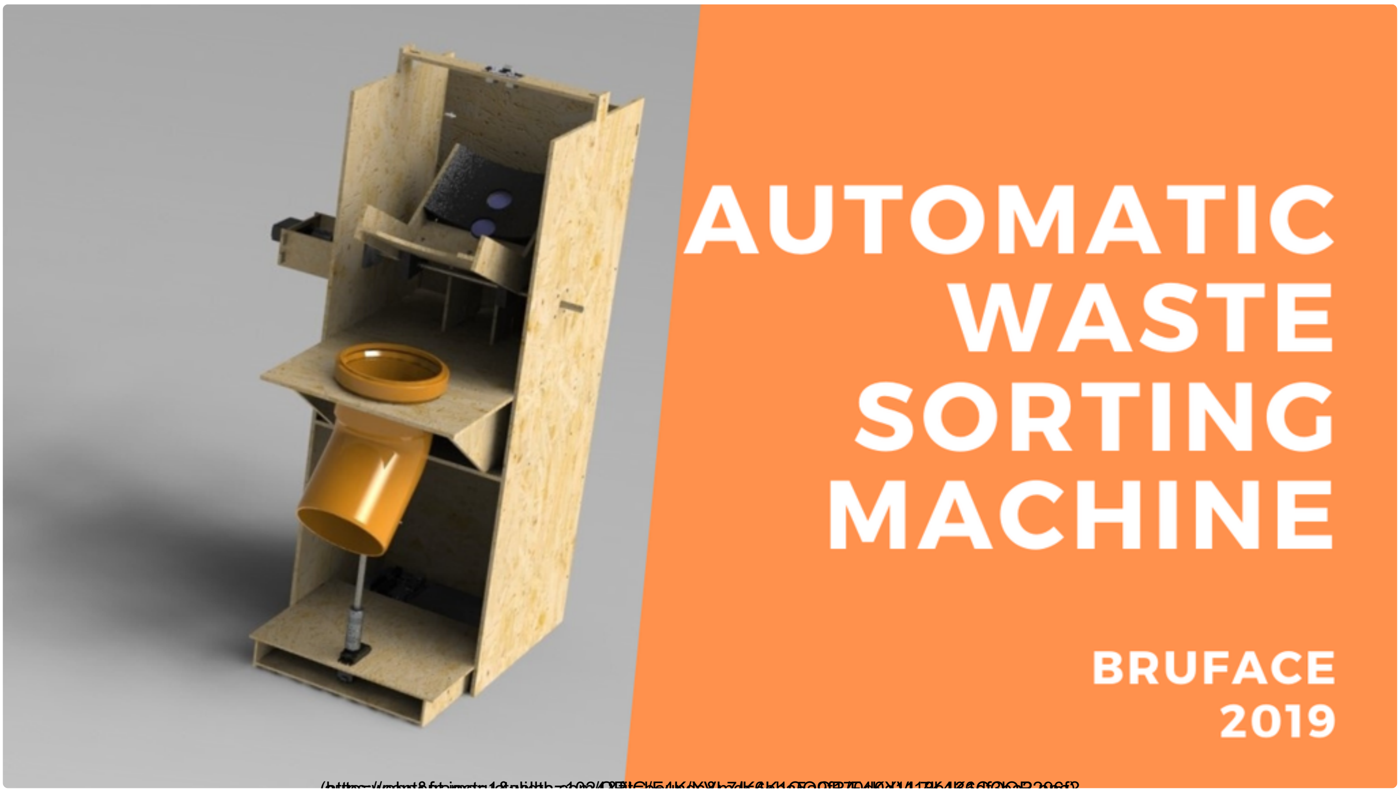


Figure 1. Automatic Waste Sorting Machine

Add Tip Ask Question Comment Download

Step 1: Working Principle of the Automatic Waste Sorting Machine

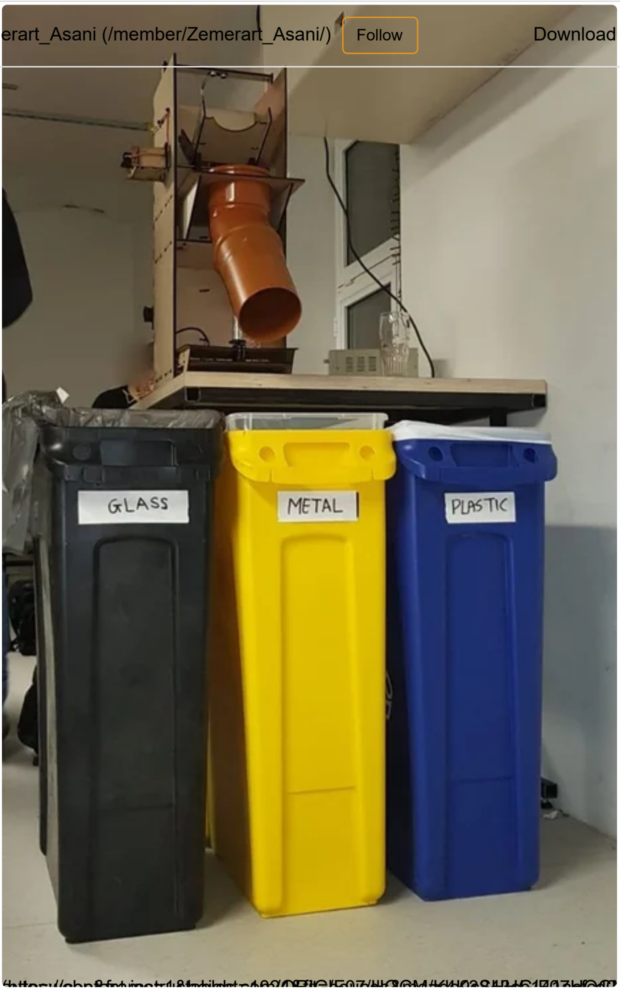


Figure 2. Automatic Waste Sorting Machine

Goal of the project

- Goal of this project is to sort minimum three different types of wastes: Plastic, Metal and Glass.

- The waste is directed by the pipe into their respective waste bins.

Object Detection/Restrictions

- Sorting wastes of different sizes and shapes is possible, but the performance is better with objects of cylindrical shape or cubic shape with limitations of the size (in terms of height) of a Coca-Cola Cans (waste should be limited to a maximum of 1kg).

- The restrictions are due to the design constraints of our waste sorting machine, modifying the size of the sorting machine (e.g. Gate, Pipe diameter) will make it more adaptable to different shapes, sizes and weight of the wastes.

- The capacitive sensor is very sensitive in detecting plastic waste, anything inside/around the plastic object affects the accuracy of the sensor.

Future implementations.

Our waste sorting machine is only a prototype. With modifications and improvements a similar concept of our waste sorting machine can be implemented in house hold, offices and industrial applications.

Add Tip Ask Question Comment Download

Step 2: Photos of the Automatic Waste Sorting Machine

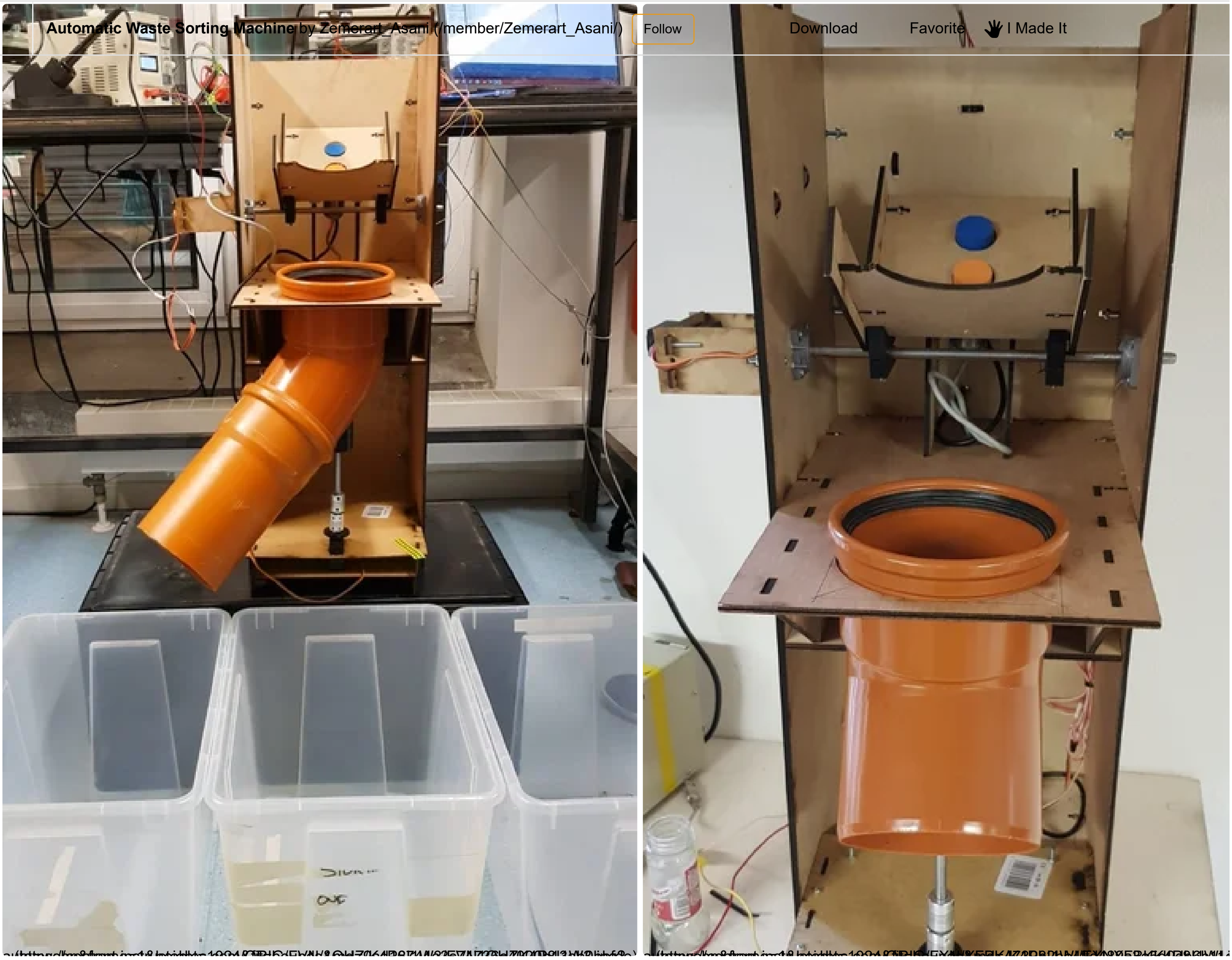
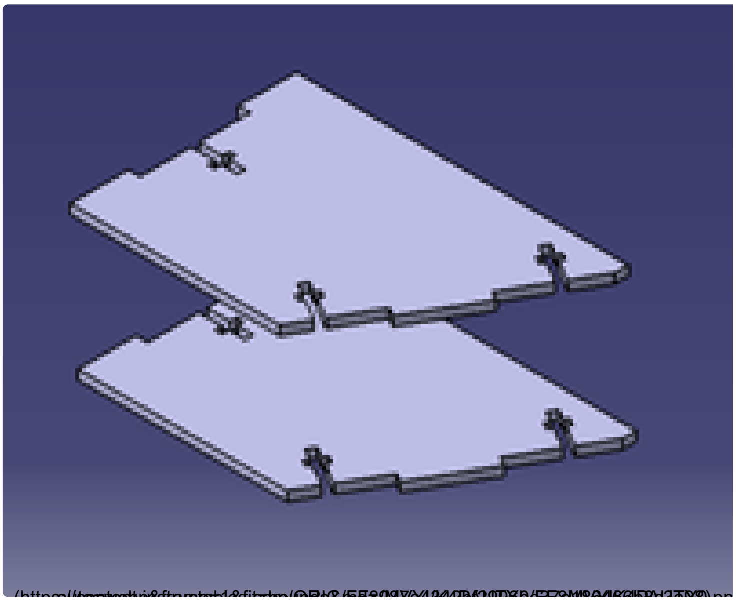
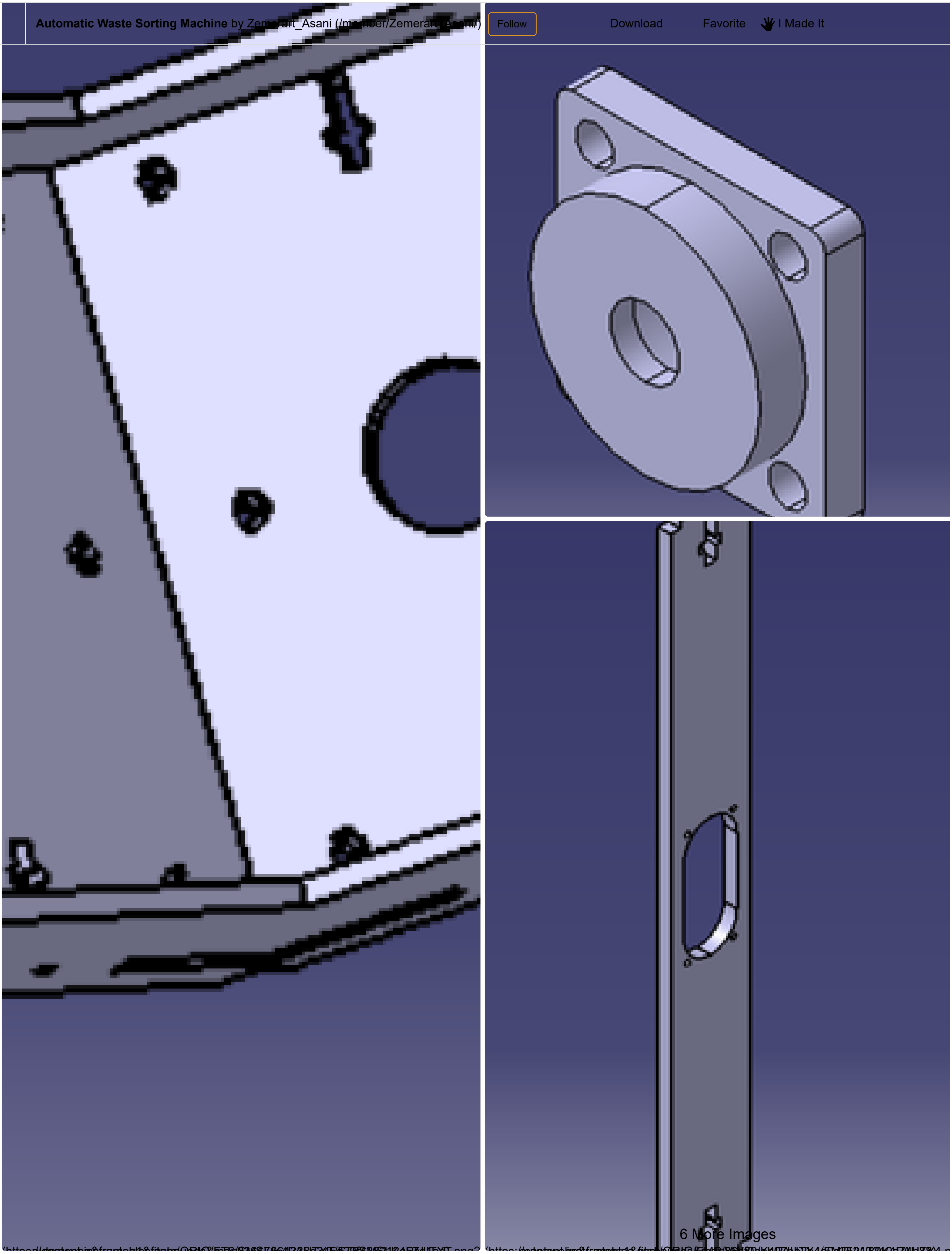


Figure 3. Waste sorting machine pictures.

Add Tip Ask Question Comment Download

Step 3: Mechanical Aspects of the Project





Automatic Waste Sorting Machine by Zemerart_Asani (/member/Zemerart_Asani/)

Figure 4. Mechanical Parts fabricated/printed

3D printed parts (Ultimaker and Prusa)

- Shaft cover
- Pipe support
- V gate Shaft-Bearing support
- Capacitive and Inductive sensor cover support
- V gate support

Laser-cut parts

1. Frame
2. V gate
3. Ultrasonic sensor support
4. Capacitive and Inductive sensor support
5. V gate motor support

frame final.DWG Download (https://content.instructables.com/ORIG/FGW/DUOU/K4PB55M1/FGWDUOUK4PB55M1.dwg)

Final Assembly.DWG Download (https://content.instructables.com/ORIG/FK2/VTFK/K4PB55N9/FK2VTFKK4PB55N9.dwg)

Add Tip Ask Question Comment Download

Step 4: Bill of Materials (Mechanical)

1) 4mm Plywood

2) 125mm Upvc Elbow - 1pcs

3) 125mm Upvc Pipe - 1pcs (1m Length)

4) 8mm Ss shaft for Motor - 1pcs (2m Length)

5) Servo Motor coupler - 2pcs

6) Screws and Nuts

7) Radial Bearing 8mm - 2pcs

Add Tip Ask Question Comment Download

Step 5: CAD DESIGN

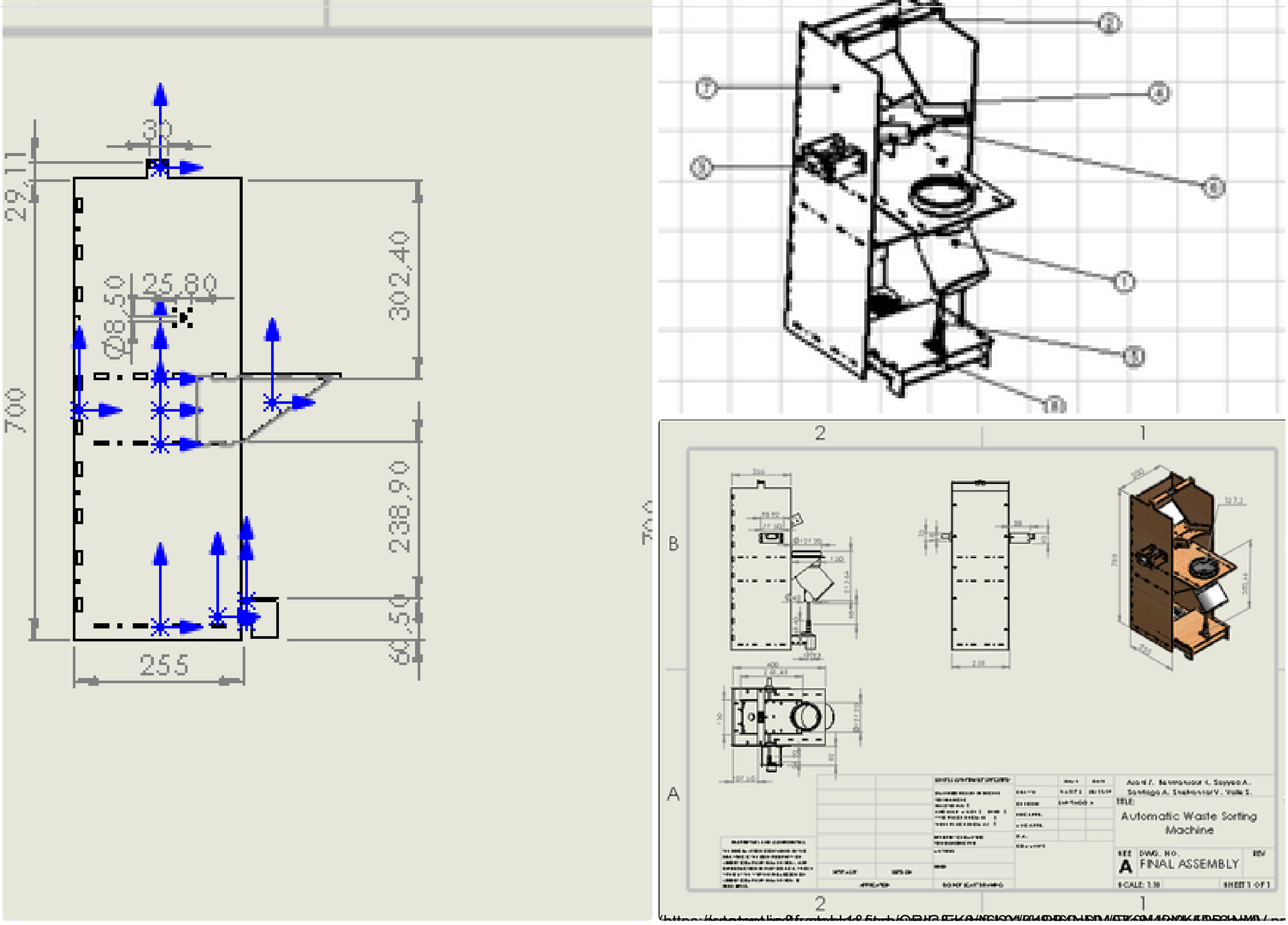


Figure 5. CAD Design

Add Tip Ask Question Comment Download

Step 6: 3D Design (Render)

[illegible]

Figure 6. 3D Design (Render)

Add Tip Ask Question Comment Download

Step 7: Bill of Materials (Electronics)

In this project we have used:

- One Arduino Uno
- One Prototyping Board
- Soldering material + equipments
- Wires, jump wires
- One Inductive Sensor
- One Capacitive Sensor
- One Ultrasonic Sensor
- Two Servo motors
- One Step Down Converter
- One 6V Power Supply

Add Tip Ask Question Comment Download

Step 8: Description of Each Component

Each component as listed below is described, detailing its specification, connections and method of operation.

- Arduino Uno
- Inductive Sensors
- Capacitive Sensors
- Ultrasonic Sensors

- Motors

- Step Down Converter

Arduino Uno

As a microcontroller board we have used the Arduino UNO which is based on the ATmega328P. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button [3 (https://store.arduino.cc/arduino-uno-rev3)].

Inductive Sensor

Inductive proximity sensors have been used to detect metal objects. We have used a larger sensor, based on our requirement of having a larger sensing range.

Type of sensor[Link (https://www.amazon.fr/gp/product/B072QBT43V/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1)]:

• Inductive Proximity Sensor Switch Model: LJ30A3-15-Z/AX

• Wire Type: DC 3 Wire Type

•Output Type: NPN-NC (Normally Closed)

• Detecting Distance: 15mm/ 0.59'

•The threads: M30

• Supply Voltage: DC 6-36V

•Current Output: 300mA

Capacitive Sensor

Capacitive sensors can sense any object within their sensing range. When the target approaches the face of the sensor, the capacitance increases, resulting in an increase in amplitude of the oscillator. Then the solid state output switch detects the increase in amplitude and based on that it is turned on or off.

There are different types of Capacitive Sensors. In our project we have used Capacitive Sensors with a compensation adjustment. The capacitive sensor sensitivity depends on the material dielectric constant. Plastic, in fact, has a lower dielectric constant than glass. We have adjusted our sensor in such a way that it can "see through" the objects of plastic nature (to not detect plastic),and glass is detected by the capacitive sensor alone.

Larger sensors have a larger range of detection.

Capacitive sensor specifications[Link (https://www.amazon.es/capacitiva-proximidad-Interruptor-ljc30-a3-h-z-normalmente/dp/B0714PG5J1/ref=sr_1_8?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=sensor+capacitivo&qid=1572965514&sr=8-8)]:

• Product Name: Capacitive Proximity Switch LJC30A3-H-Z/BX

• Wire Type: DC 3 Wire Type (Black, Brown, Blue)

- Output Type: NPN NC (Normal Closed)

- Diameter of Column Sensor: 30mm; Detecting distance: 15mm

- Supply Voltage: DC 6-36V; Current Output: 300mA

Motors

We have used servo motors for both gate and pipe, specifically,
we have used TowerPro MG996R servo motors.

Specifications of the TowerPro MG996R servo motor[Link (http://www.towerpro.com.tw/product/mg996r/)]:

- Weight: 55g

- Dimension: 40.7×19.7×42.9mm

- Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)

•Operating Speed : 0.13sec / 60 degrees (6.0V no load)

- Operating voltage: 4.8~ 6.6v

- Gear Type: Metal Gear

- servo wire length: 32cm

- Current draw at idle 10MA

- No load operating current draw 170mA

- Stall current draw 1400mA

•Weight: 0.080 kg

Step Down Converter

The DC/DC step-down voltage regulator is capable to convert voltage between 3.2V and 40V into a smaller voltage between 1.25V and 35V. It is capable of driving a 3A load.
We have adjusted this step-down converter for our requirements, respectively the 6V from the power supply has been converted to 5V which feeds the Arduino Uno; this has been done via the potentiometer for voltage adjustment.

Specifications[Link (https://www.antratek.be/adjustable-dc-dc-power-converter-1-25v-35v-3a-lm2596?gclid=CjwKCAiAuqHwBRAQEiwAD-zr3cJe_0MROWoFzcI38YSEkKsHVX3NfjktMYdooQZa33tmFxXW0ebdwhoCUZgQAvD_BwE)]:

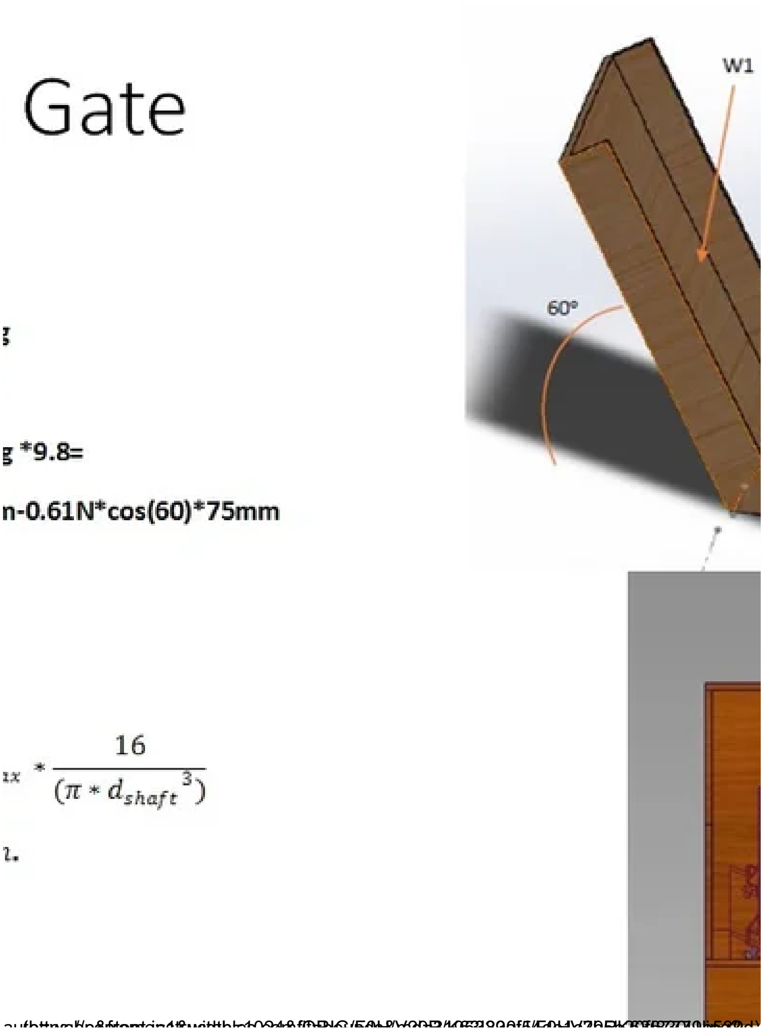
- Input voltage: 3.2V - 40VDC
- Output voltage: 1.25V - 35VDC

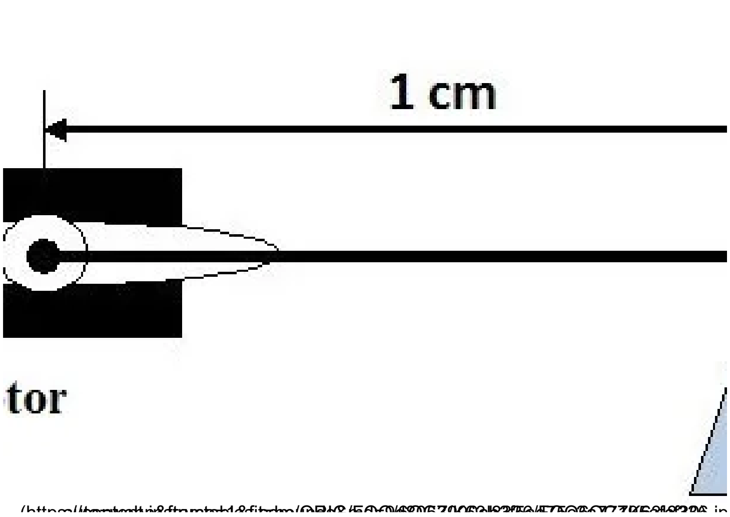
- Max. output Current: 3A

- Dimensions: 43mm x 21mm x 14mm(l) x (w) x (h)

- Weight: 12 g

Step 9: Torque





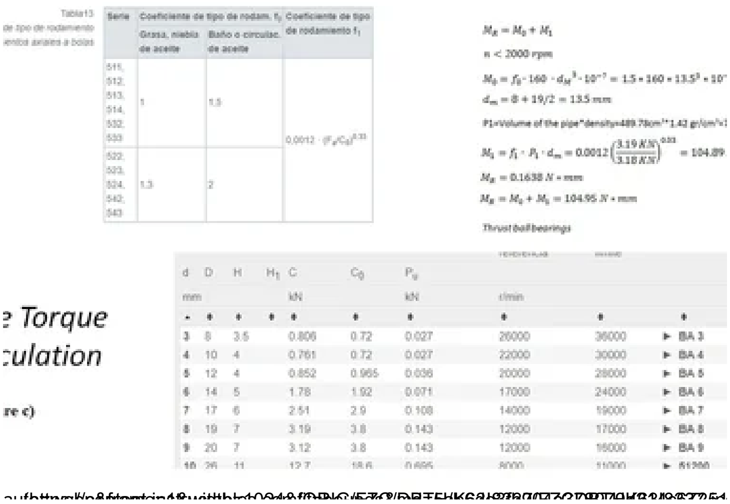


Table13 de tipo de rodamiento ientos axiales a bolas

Serie	Coeficiente de tipo de rodam. f_0		Coeficiente de tipo de rodamiento f_1
	Grasa, niebla de aceite	Baño o circulac. de aceite	
511, 512, 513, 514, 532, 533	1	1.5	$0{,}0012 \cdot (F_a/C_0)^{0.33}$
522, 523, 524, 542, 543	1.3	2	

$M_R = M_0 + M_1$

$n < 2000$ rpm

$M_0 = f_0 \cdot 160 \cdot d_M{}^3 \cdot 10^{-7} = 1.5 * 160 * 13.5^3 * 10^{-7}$

$d_m = 8 + 19/2 = 13.5$ mm

P1=Volume of the pipe*density=489.78cm^3*1.42 gr/cm^3

$M_1 = f_1 \cdot P_1 \cdot d_m = 0.0012\left(\frac{3.19\,KN}{3.18\,KN}\right)^{0.33} = 104.89$

$M_R = 0.1638\ N * mm$

$M_R = M_0 + M_1 = 104.95\ N * mm$

Thrust ball bearings

d	D	H	H_1	C	C_0	P_u			
mm				kN		kN	r/min		
3	8	3.5		0.806	0.72	0.027	26000	36000	► BA 3
4	10	4		0.761	0.72	0.027	22000	30000	► BA 4
5	12	4		0.852	0.965	0.036	20000	28000	► BA 5
6	14	5		1.78	1.92	0.071	17000	24000	► BA 6
7	17	6		2.51	2.9	0.108	14000	19000	► BA 7
8	19	7		3.19	3.8	0.143	12000	17000	► BA 8
9	20	7		3.12	3.8	0.143	12000	16000	► BA 9
10	26	11		12.7	18.6	0.695	8000	11000	► 51200

On the choice of the motor it is very important one to have an idea of the torque required for the specific implementation.

In the figure a) are given the calculations for the gate torque. The torque needed for the motor to rotate the gate is calculated to be 1.72875 kg-cm. The diameter of the shaft is chosen 4 mm.

In the figure c) are given the calculations for the pipe torque. The torque needed for the pipe motor to rotate the pipe is calculated to be 1.0495 kg-cm.

Experimentally: the current needed is 0.11 A, which corresponds to the torque of 0.8635 kg-cm. The experimental results are close to the calculated analytical values.

In the figure b) is given the servo motor with the weight (load) of 11 kg suspended in 1 cm which represents the stall torque of 11 kg-cm as given in the data sheet. The necessary torque for the gate and the pipe has been calculated analytically and we concluded that each of the servo motors; are more than sufficient to provide the required torque. Further to note the the maximum current draw from the system was measured to be 0.28 A (when the Pipe Servo is rotating) which is less than the stall current of the motor 1.4 A.

A simple/fast approximation of the torque needed for the pipe servo can be done as follows:

T=F*d
In terms of kg-cm we have:

T=kg*cm

T=0.8 kg *(12.5/2)

T=0.8*6.25=5 kg-cm

This result gives as an indication that the servo motor used can fulfill the requirements in terms of torque.

Simply the motor can pull a weight of 0.8 kg suspended in a distance of 12.5/2 cm and we get 5 kg-cm. Since the Torque is directly proportionally to the current. And speed of rotation is directly proportional to voltage applied to the servo motor.

Therefore, we can analytically now calculate the torque that corresponds to the case of 0.28 A.

T=Kt*I
Tstall=11kg-cm
Istall=1.4 A
Kt=7.85
For our case I=0.28 A
T=2.198 kg-cm

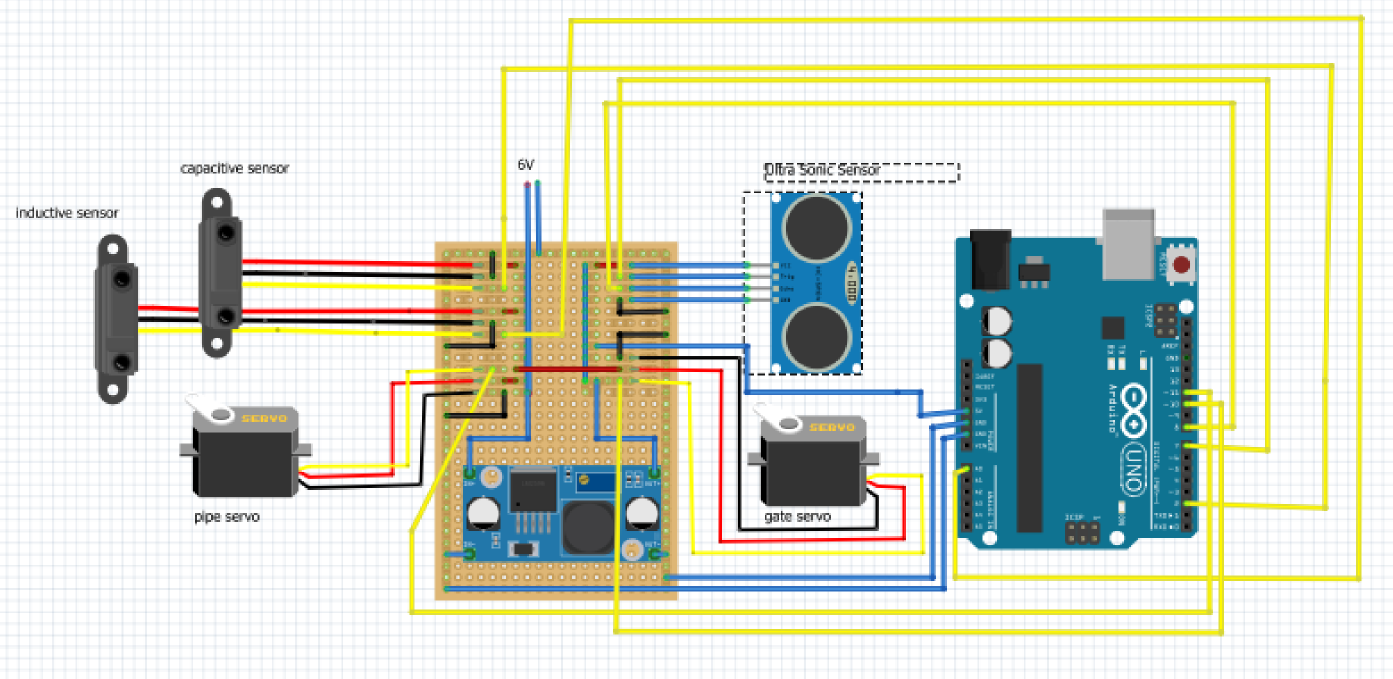
Add Tip Ask Question Comment Download

Step 10: Mounting Proximity Sensors

While placing the Proximity sensors one has to be careful that the sensors do not interfere with each other. One has to consider the cases when mounting the sensors face-to-face or side-by-side; they should be at least two sensor diameters far from each other. These distances have been respected in our design and sensors do not have interference, however, we used also some trial and error to optimize furthermore and place the sensors; which allow us to be more effective and efficient in terms of machine usage[1, 2].

Add Tip Ask Question Comment Download

Step 11: Physical Connections of the Electronic Components



(https://content.instructables.com/ORIG/FIO/M8AJ/K4EE6SDL/FIOM8AJK4EE6SDL.png?auto=webp&frame=1&fit=bounds&md=0c55f28c99494e0828bff6a13a7878c5)

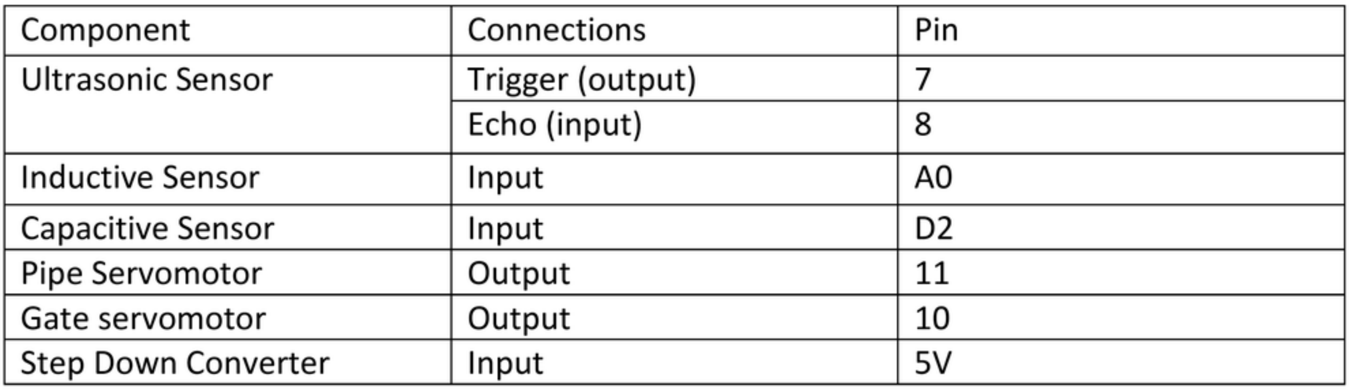
Figure 7. Physical connections of electronic components

As seen in the figure above, the electronic components have been soldered in a prototyping board. The general schema is the following: We utilized one 6V power supply which drives all the components. With this 6 volts provision, we powered the two servos, inductive sensor and the capacitive sensor. Next we used a step down converter to convert the 6V to 5V with which we can drive the Arduino via the 5V pin and power the ultrasonic sensor. When soldering one has to be careful and connect all the components to the same ground.

Add Tip Ask Question Comment Download

Step 12: Pins Connections to Arduino

Table 1. Pins Connections to Arduino



Component	Connections	Pin
Ultrasonic Sensor	Trigger (output)	7
	Echo (input)	8
Inductive Sensor	Input	A0
Capacitive Sensor	Input	D2
Pipe Servomotor	Output	11
Gate servomotor	Output	10
Step Down Converter	Input	5V

[illegible]

Add Tip Ask Question Comment Download

Step 13: Programming

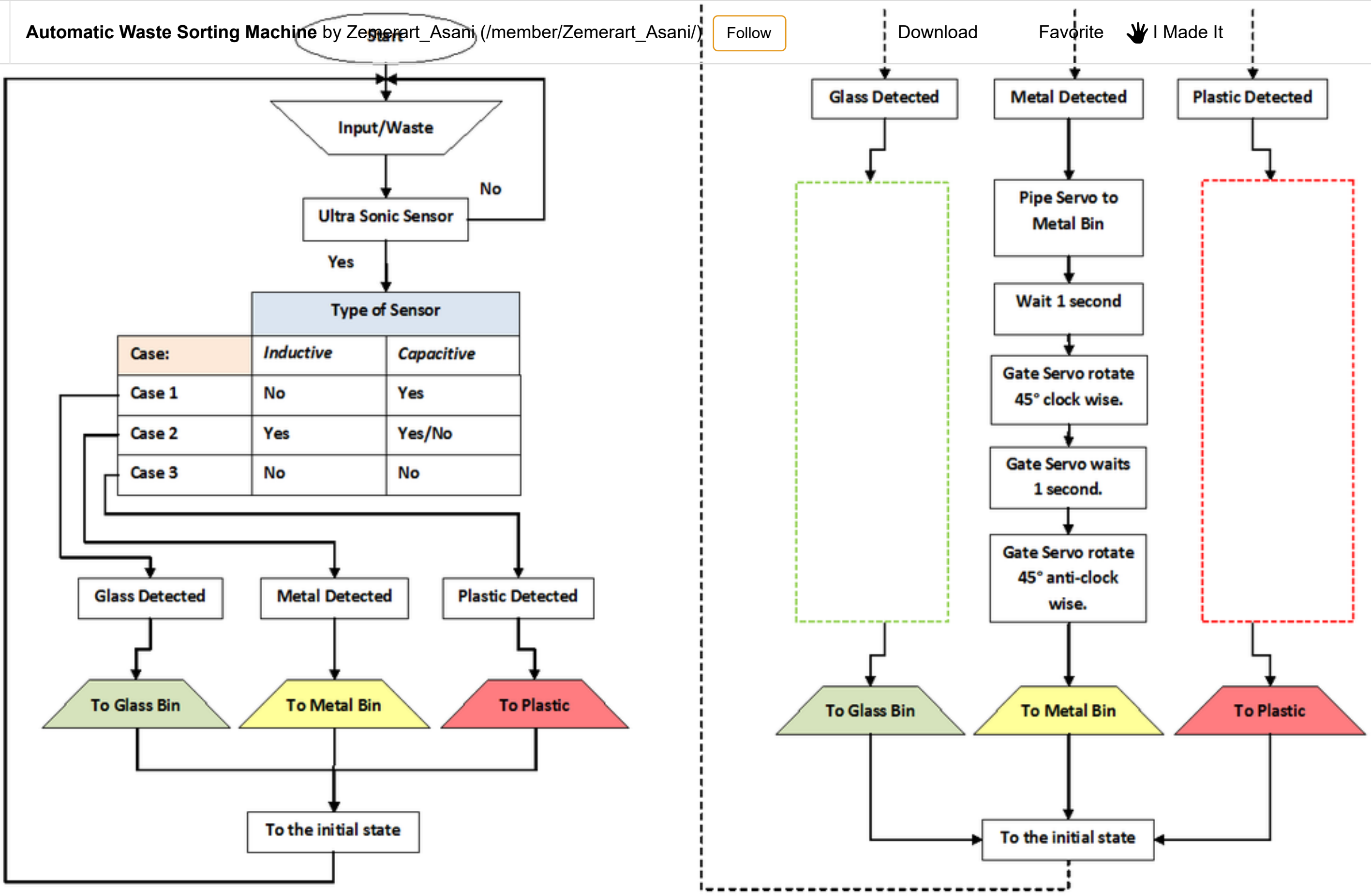


Figure 8. Flowchart

To do the sorting of three different types of materials we have used three sensors. Based on these sensors values we can distinguish the type of material. We have 3 cases: Metal detected, Plastic Detected, Glass detected. As we can notice in the flowchart those three cases correspond to those specific state combinations of the sensors.

The programming is divided into these sections:

1. Initialization of the system

2. Object and Material Detection

3. Servos Movements

1. Initialization

Comment: In startup Gate and Pipe servo will go to a prescribed position, this procedure will be repeated in each start-up or reset. This part of the code will be executed only once in startup.

- Read the values for the actual position of the two Servo motors.

If Pipe position != to the zero position

-move pipe to the zero position

-move gate and then bring it back to the initial position (This is done intentionally each system start up/reset; no matter if it is in the zero position or not)

If Pipe position == the zero position

-move gate and then bring it back to the initial position (This is done intentionally each system start up/reset; no matter if it is in the zero position or not)

-Initialize sensors

2. Object and Material Detection

if (Inductive Sensor<=250 && Capacitive Sensor == 1 && Ultrasonic Sensor Distance<=14 cm)

-Plastic Detected: To Plastic Bin (step 3)

if(Inductive Sensor<=250 && Capacitive Sensor != 1 && Ultrasonic Sensor Distance<=14 cm)

-Glass Detected: To Glass Bin (step 3)

if(Inductive Sensor>=250){

-Metal Detected: To Metal Bin (step 3)

3. Servos Movements

If Plastic Detected:

-Move Pipe Servo to Plastic Bin

-Delay

-Move Gate Servo from position 0 to 1
-Delay

-Move Gate Servo from position 1 to 0

If Glass Detected:
-Move Pipe Servo to Glass Bin

-Delay

-Move Gate Servo from position 0 to 1

-Delay

-Move Gate Servo from position 1 to 0

If Metal Detected:

-Move Pipe Servo to Metal Bin

-Delay

-Move Gate Servo from position 0 to 1

-Delay

-Move Gate Servo from position 1 to 0

In the following section the code will be provided.

Step 14: CODE

```
[code]<br>
```

```
//********************************************
// GROUP 8: AUTOMATIC WASTE SORTING MACHINE  *
// MECHATRONICS 2019                         *
// @GROUP 8                                  *
//*******************************************
```

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

```
Servo Pipe_Servo;
Servo Gate_Servo;
```

```
// define the variables
int sensorInd = A0; //Pin A0 (analog)
int sensorpin = 2; //Digital Pin nr. 2
int indValue;
const int trigPin = 7;
const int echoPin = 8;
int i=0;
float Pipe_Pos=0.0;
float Gate_Pos=0.0;
int n=0;
```

```
long duration;
int distance;
```

```
float metalDetected;
```

```
void setup() {
```

```
pinMode(trigPin, OUTPUT); // trigPin=output
pinMode(echoPin, INPUT); // echoPin = input
```

```
Serial.begin(9600); // Starts the serial communication
```

```
// Codes that run only once[
```

```
//--System Initialization--
```

```
    Pipe_Pos=Pipe_Servo.read();//Read the last value given to Pipe servo
    Gate_Pos=Gate_Servo.read();//Read the last value given to Pipe servo
```

```
    Serial.println("Motor_PIPE Position");
    Serial.println(Pipe_Pos);
    Serial.println("Motor_GATE Position");
    Serial.println(Gate_Pos);
    delay(5000);
```

```


    Pipe_Servo.attach(11);
```

```
    if (Pipe_Pos<90){

```

```
      for(i=Pipe_Pos; i<=90; i=i+1){
      Pipe_Servo.write(i);
      delay(15);
      }
    }

      else {
        for(i=Pipe_Pos; i>=90; i=i-1){
        Pipe_Servo.write(i);
        delay(15);

        }
      }
```

```
delay(1000); //Delay (wait) for servo pipe to go to the bin.
```

```
    //Next rotate the gate servo.
    //And control the speed of Gate Servo.
    Gate_Servo.attach(10);
```

```
     for(n=0; n<=45; n=n+1){  //Choose the right angle for Gate servo
      Gate_Servo.write(n);
      delay(20); //Controll the speeed of Gate servo
```

```
    }
```

```
    delay(1000); //Delay (wait) and then bring back the Gate_Servo to its
     //initial position.
```

```
     //Bring Back Gate_Servo to initial position
     //So from actual position 2-> n=45; we go backwards up to n=0.
```

```
     for(n=45; n>=0; n=n-1){
      Gate_Servo.write(n);
      delay(25); //Control the speed of Gate servo
     }
```

```
delay(5000);
```

```
//--System Initialization--//
// Codes that run only once]
}
```

```
void loop() {
```

```
//delay(100);
delay(3000);
```

```
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(10);//
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(30);//it was 1
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
```

```
indValue=analogRead(sensorInd);//Save value that is read from the analog pin A0 to the variable indValue
delay(10);
int sensorstate = digitalRead(sensorpin); // Digital value for Capacitive S.
delay(500);
metalDetected = (float)indValue*100/1024.0; //from analog value
delay(50);
Pipe_Pos=Pipe_Servo.read();//Read the actual position of Pipe servo
Gate_Pos=Gate_Servo.read();//Read the actual position of Gate servo
```

```
//**METAL DETECTED**/
```

```
/ if(indValue>=250 && sensorstate == 1 && distance<=14)
  if(indValue>=250){
   Serial.println("Metal Detected");
    Pipe_Pos=Pipe_Servo.read();//Read the last value given to Pipe servo
    Gate_Pos=Gate_Servo.read();//Read the last value given to Pipe servo
    Serial.println("Motor_PIPE Position");
    Serial.println(Pipe_Pos);
    Serial.println("Motor_GATE Position");
    Serial.println(Gate_Pos);
```

```
    Serial.println("Motor to metal bin");
    delay(100);
    //Wait for some time after the measurements and move servo to
    //the corresponding bin
```

```
    //For initial position of servo_gate/PIPE in this case the read function will give us the last position
of servo
    //in degree
    //Go to the second position of the PIPE servo @90 degree
    //To control the speed of the servo we do this for loop.
    //We have 2 cases 1; when pipe is somewhere in position greater then 90, in this particular case we want
to go to 90 so we have to loop back.
    //The other case is to loop forward i=i+1, (meaning to increase the angle in incremental manner).

    if (Pipe_Pos<90){
```

```
      for(i=Pipe_Pos; i<=90; i=i+1){
      Pipe_Servo.write(i);
      delay(15);
      }
    }

      else {
        for(i=Pipe_Pos; i>=90; i=i-1){
        Pipe_Servo.write(i);
        delay(15);

        }
      }
```

```
delay(1000); //Delay (wait) for servo pipe to go to the bin.
```

```
    //Next rotate the gate servo.
    //And control the speed of Gate Servo.
```

```
     for(n=0; n<=45; n=n+1){
      Gate_Servo.write(n);
      delay(20);
```

```
    }
```

```
    delay(1000); //Delay (wait) and then bring back the Gate_Servo to its initial position.
```

```
     //Bring Back Gate_Servo to initial position
     //So from actual position 2--> n=45; we go backwards up to n=0.
```

```
     for(n=45; n>=0; n=n-1){
      Gate_Servo.write(n);
      delay(25);
     }

   //**METAL DETECTED**//
```

```
   //**PLASTIC DETECTED**//

   }
```

```
   else if(indValue<=250 && sensorstate == 1 && distance<=15){ //NO NO state
   delay(100);
   Serial.println("Plastic Detected");
```

```
    Pipe_Pos=Pipe_Servo.read();//Read the last value given to Pipe servo
    Gate_Pos=Gate_Servo.read();//Read the last value given to Pipe servo
    Serial.println("Motor_PIPE Position");
    Serial.println(Pipe_Pos);
    Serial.println("Motor_GATE Position");
    Serial.println(Gate_Pos);
    Serial.println("Motor to Plastic Bin");
```

```
    delay(100);
    //Wait for some time after the measurements and move servo to
    //the corresponding bin
    //For initial position of servo_gate/PIPE in this case the read function will give us the last position
of servo
    //in degree
    //Go to the second position of the gate Pipe_Servo in degree (145) for Plastic
    //To control the speed of the servo we do this for loop.
```

```
    //We have 2 cases 1; when pipe is somewhere in position greater then 145, in this particular case we wan
t to go to 145 so we have to loop back.
    //The other case is to loop forward i=i+1, (meaning to increase the angle in incremental manner)until we
reach the desired angle.

    if (Pipe_Pos<145){
```

```
      for(i=Pipe_Pos; i<=145; i=i+1){
      Pipe_Servo.write(i);
      delay(15);
      }
    }

      else {
        for(i=Pipe_Pos; i>=145; i=i-1){
        Pipe_Servo.write(i);
        delay(15);

        }
      }
```

```
delay(1000); //Delay (wait) for servo pipe to go to the bin.
```

```
    //Next rotate the gate servo.//This remains the same for all other cases
    //And control the speed of Gate Servo.
```

```
     for(n=0; n<=45; n=n+1){
      Gate_Servo.write(n);
      delay(20);
```

```
    }
```

```
    delay(1000); //Delay (wait) and then bring back the Gate_Servo to its
     //initial position.
```

```
     //Bring Back Gate_Servo to initial position
     //So from actual position 2--> n=45; we go backwards up to n=0.
```

```
     for(n=45; n>=0; n=n-1){  //Choose the right angle for Gate servo
      Gate_Servo.write(n);
      delay(25); //Controll the speeed of Gate servo (it can be a litle faster now)
     }
//delay(100);

   }
```

```
//**PLASTIC DETECTED**//
```

```
//**GLASS DETECTED**//
```

```
else if(indValue<=250 && sensorstate != 1 && distance<=14){
   Serial.println("Glass Detected");
```

```
    Pipe_Pos=Pipe_Servo.read();//Read the last value given to Pipe servo
    Gate_Pos=Gate_Servo.read();//Read the last value given to Pipe servo
    Serial.println("Motor_PIPE Position");
    Serial.println(Pipe_Pos);
    Serial.println("Motor_GATE Position");
    Serial.println(Gate_Pos);
    Serial.println("Motor to Glass Bin");
```

```
    delay(1000);
    //Wait for some time after the measurements and move servo to
    //the corresponding bin
    //For initial position of servo gate/PIPE in this case the read function will give us the last position
of servo in degree
    //Go to the second position of the Pipe_Servo @25 degree for Glass
    //To control the speed of the servo we do this for loop.
```

```
    //We have 2 cases; when pipe is somewhere in position greater then 90, in this particular case we want t
o go to 25 so we have to loop back.
    //The other case is to loop forward i=i+1, (meaning to increase the angle in incremental manner).
    //Actually in this case we can have only one case when angle is greater then 25, but however! :P
```

```
    if (Pipe_Pos<25){
```

```
      for(i=Pipe_Pos; i<=25; i=i+1){
      Pipe_Servo.write(i);
      delay(15);
      }
    }

      else {
        for(i=Pipe_Pos; i>=25; i=i-1){
        Pipe_Servo.write(i);
        delay(15);

        }
      }
```

```
delay(1000); //Delay (wait) for servo pipe to go to the bin.
```

```
    //Next rotate the gate servo.//This remains the same for all other cases
    //And control the speed of Gate Servo.
```

```
     for(n=0; n<=45; n=n+1){
      Gate_Servo.write(n);
      delay(20);
```

```
    }
```

```
    delay(1000); //Delay (wait) and then bring back the Gate_Servo to its
     //initial position.
```

```
     //Bring Back Gate_Servo to initial position
     //So from actual position n=45; we go backwards up to n=0.
```

```
     for(n=45; n>=0; n=n-1){
      Gate_Servo.write(n);
      delay(25);
     }
```

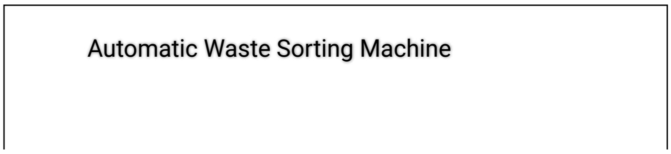
```
    Serial.println("Motor_PIPE Position");
    Serial.println(Pipe_Pos);
    Serial.println("Motor_GATE Position");
    Serial.println(Gate_Pos);
}
}
```

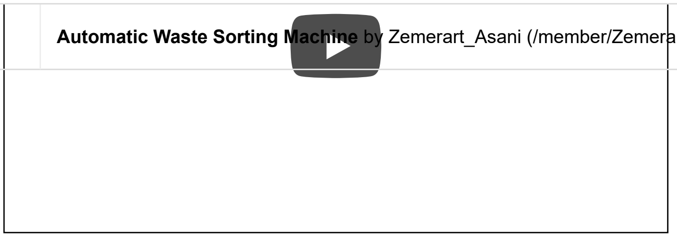
[/code]

*We suggest to use the code attached here (https://drive.google.com/open?id=1Uk-knbf5CxyElHnqZorZ6uISC30FiIlK).

Add Tip   Ask Question   Comment   Download

Step 15: VIDEO 1





The video describes the operation and performance of the Automatic Waste Sorting Machine which has been tested for sorting three different type of materials (metal, glass and plastic). As it can be noticed we have used regular trash bins that can be found in markets.

Add Tip Ask Question Comment Download

Step 16: VIDEO 2

In this video, a reset test has been conducted. As seen the initialization of the system has been done and the machine sets to its zero position. This video is done before finalizing our project (in terms of some mechanical aspects). However, in terms of the codes the same codes have been used as in video 1.

Step 17: Scope for Future Implications and Improvements

•This project can be modified further for detecting an organic waste (wet waste), by adding a moisture sensor.

• With the increase of the number of sensors one could segregate more variety of wastes.

•By modifying the design we can make it more adaptable to detect wastes of different shapes, sizes and weights.

•This project could be made more aesthetically pleasing and ergonomically flexible for betterment of Human Machine interaction.

Add Tip Ask Question Comment Download

Step 18: Challenges Faced

Every project on its own has its difficulties and challenges, for our project challenges are listed below:

Rotating the Pipe: Designing a mechanism, while bettering of the project done last year was a challenge. We wanted to have something that can be implemented in real life (like having a centralized garbage system in a Building). Countering the problem of the last year project we focused on not moving the Garbage can and so we went for a design which included a Pipe which rotates in 3 direction of fixed garbage can.

Designing a Gate: Designing a gate where the sensor detects the type of waste and directs it to the pipe, while the pipe rotates in the meantime to the respective garbage can was very tricky. We had several ideas from having an Objection plate while the garbage is held on the plate for detection and having a tilting bucket etc. Finally, we went for an idea of V-gate in order to avoid the movement of sensors and have less mechanical moving parts.

Using bearings: The original Idea involved using bearing both for V-gate shaft and Pipe support. As seen we used an 8 mm Radial bearing for Gate shaft on the either side. Also the rotating pipe shaft was to be supported by a thrust bearing, eventually we opted out of this idea as the wooden frame constructed for the Pipe was more than sufficient to support it. This saved a lot of time and prevented further any future mechanical complications in the project.

Using only one power supply: It was a challenge to restrict our selves in using only one power supply and having least electronic components as possible. Therefore, in our design phase, we have considered all the possible ways to use only one power supply efficiently; and we opted for the scheme that is represented in the electronics chapter.

Accuracy of sensors: Sensors are not always accurate, having limitation in their distance of measurement. This affects it's accuracy in detection (particularly when the object is small). We intended to modify the gate design by adding a printed part but it effected the performance of the sensor and so for future implications of the system for better detection, shielded sensors can be used.

Position of Sensors: There are many factors that came in to the picture while deciding the position and location of our sensors. We considered making a fixed structure for the sensors because we did not want the sensors to be attached to the gate. Also, the distance between the two sensors had to be considered so that the sensors do not influence each other. This has been done by considering the recommended distance in the datasheet and also by some trial and error we found the optimal locations to place our sensors.

Initialization of the system: In the start-up of the system, we wanted everything to go to the prescribed zero position. However, servos were moving without a command given to them (fast and not as commanded). We "fixed" this by considering the actual position of the motors on the system start-up. Furthermore, one can consider here other ways of solving this issue, for instance, using different servos, other software solutions like using EEPROM; or hardware solutions.

Tuning and adjusting different parameters of the system: Tuning and adjusting different parameters of the system, and making everything work together was rather challenging. We opted for values that are optimal, however, the system response can be increased by changing these parameters (ex. speed of servos, the delays added into the code, etc.). Also, we observed that it will detect more efficiently metals if we only consider the data from the inductive sensor, this can be seen in the codes given in the programming section.

Add Tip Ask Question Comment Download

Step 19: Conclusion

This project has been done during the academic year 2019-2020 as part of the Mechatronics course, in the Brussels Faculty of Engineering; a joint Master's program of the Vrije Universiteit Brussel (VUB) and Université Libre de Bruxelles (ULB). The project was challenging. The requirement was to do an automatic machine to be able to sort minimum two different types of waste. As a team we managed to meet these requirements and developed a machine that sorts three types of waste.

Authors of this project are (A-Z):

Afaque Sayyed (https://www.instructables.com/member/afaksayyed/)

(https://www.instructables.com/member/afaksayyed/)Anibal Santiago

Kawtar Benmansour

Varun Shelvankar

Silvana Valle

Zemerart Asani

Instructors:

Prof. Dr. Ir. Bram Vanderborght

Ir. Albert De Beir

Ir. Marco Rossini

Step 20: Download Files

CAD files.rar Download (https://content.instructables.com/ORIG/FWR/NH81/K4PAUUFF/FWRNH81K4PAUUFF.rar)

Add Tip Ask Question Comment Download

Step 21: References

[1] https://www.edata.omron.com.au/eData/Prox/Proximity_TG.pdf (https://www.edata.omron.com.au/eData/Prox/Proximity_TG.pdf)

[2] https://www.newark.com/pdfs/techarticles/eatonCHD/IproxInductiveProximitySensors-mounting.pdf (https://www.newark.com/pdfs/techarticles/eatonCHD/IproxInductiveProximitySensors-mounting.pdf)

[3] https://store.arduino.cc/arduino-uno-rev3 (https://store.arduino.cc/arduino-uno-rev3)

[4 (http://www.enggjournals.com/ijet/abstract.html?file=17-09-06-127)] (https://www.researchgate.net/publication/322158863_Automatic_Sorting_of_Solid_Wastes_Using_Sensor_Fusion)Khaled Chahine, Bilal Ghazal. Automatic Sorting of Solid Wastes Using Sensor. International Journal of Engineering andTechnology 9.

[5] https://cdn.antratek.nl/media/wysiwyg/pdf/datasheet-pow00900m.pdf (https://cdn.antratek.nl/media/wysiwyg/pdf/datasheet-pow00900m.pdf)

[6] https://howtomechatronics.com/projects/arduino-color-sorter-project/ (https://howtomechatronics.com/projects/arduino-color-sorter-project/)

Add Tip Ask Question Comment Download

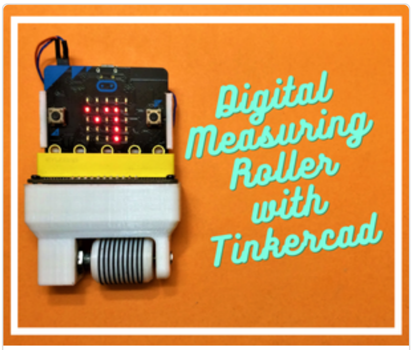
Participated in the
STEM Contest (/contest/stem2019/)

Be the First to Share

Did you make this project? Share it with us!

I Made It!

Recommendations



(/Digital-Measuring-Roller-Using-Microbit-Tinkercad/)

Digital Measuring Roller Using Microbit & Tinkercad (/Digital-Measuring-Roller-Using-Microbit-Tinkercad/) by Makertronics

♥ 6 👁 405

(/Pocket-Dice-Electronic-Dice-for-Liars-Dice-and-Mor/)

Pocket Dice! Electronic Dice for Liars Dice and More (/Pocket-Dice-Electronic-Dice-for-Liars-Dice-and-Mor/) by sunyecz06

♥ 11 👁 1.1K

(/contest/makeitglow21/) (/contest/fta2021/)

Add Tip

Ask Question

Post Comment

We have a **be nice** policy.
Please be positive and constructive.

Add Images Post

18 Discussions

1

(/member/afaksayyed/) afaksayyed (/member/afaksayyed/) Question 2 months ago

Answer Upvote

Hello Sir Asani,
This is one of the best project I see on Instructables for waste sorting. Could you link me to the brilliant team who did this project.

(/member/Lylo/) Lylo (/member/Lylo/) 3 months ago Reply Upvote

Hello sir, I see that in your arduino code, analog values read from the inductive sensor were used to determine whether the waste material was metal, plastic or glass, depending on the High or Low state of the capacitive sensor. I would like to ask
1. why you used the analog values from the inductive sensor and not the capacitive sensor because according to what I know, inductive sensors only detect metal, but capacitive sensors detect metal, plastic, paper, glass, and many other materials.
2. The analog value being greater than or equal to 250, how does it specifically help to determine the waste type? I mean, do different kinds of materials induce different analog reading ranges when they come close to the sensor? If so, what range of values are specific to plastic and glass? Or is there any site I can get information on this from?

5 replies

(/member/Lylo/) Lylo (/member/Lylo/) 3 months ago

Reply Upvote

I have another question sir.
Yes, we know that capacitive sensors detect both glass and plastic, but from the code, in order to detect glass, the condition was that sensor state (state of the capacitive sensor) should not be equal to one. Is it that the capacitive sensor was tuned to not detect glass or...?
I'm really confused with the detection, and I'd love it if you could help me out with an explanation.
Thank you

1

(/member/Antony Jeniston/) Antony Jeniston (/member/Antony Jeniston/) 6 months ago

Reply Upvote

Sir any working video is available

1 reply

(/member/Kartik0004/) Kartik0004 (/member/Kartik0004/) Question 11 months ago on Step 8

Answer Upvote

Hello sir, what will happen if I use the proximity sensors with the output type of NPN -NO instead of NPN-NC.

4 answers

(/member/FlorenceC17/) FlorenceC17 (/member/FlorenceC17/) Question 10 months ago

Answer Upvote

Sir/Maam can i ask for the image of symatic diagram of the electronic component if will it will be a great help to me thanks alot!

2 answers

Post Comment