

A smart Trash Bin **CRAP BAG**

Team Members:

Raj Dhaubanjar
Simran Thapa
Sumit Shrestha





Introduction

- Smart Trash Bin – Crap Bag
- No hassle, No Physical Touch
- Prototype of an automated bin

Motivation



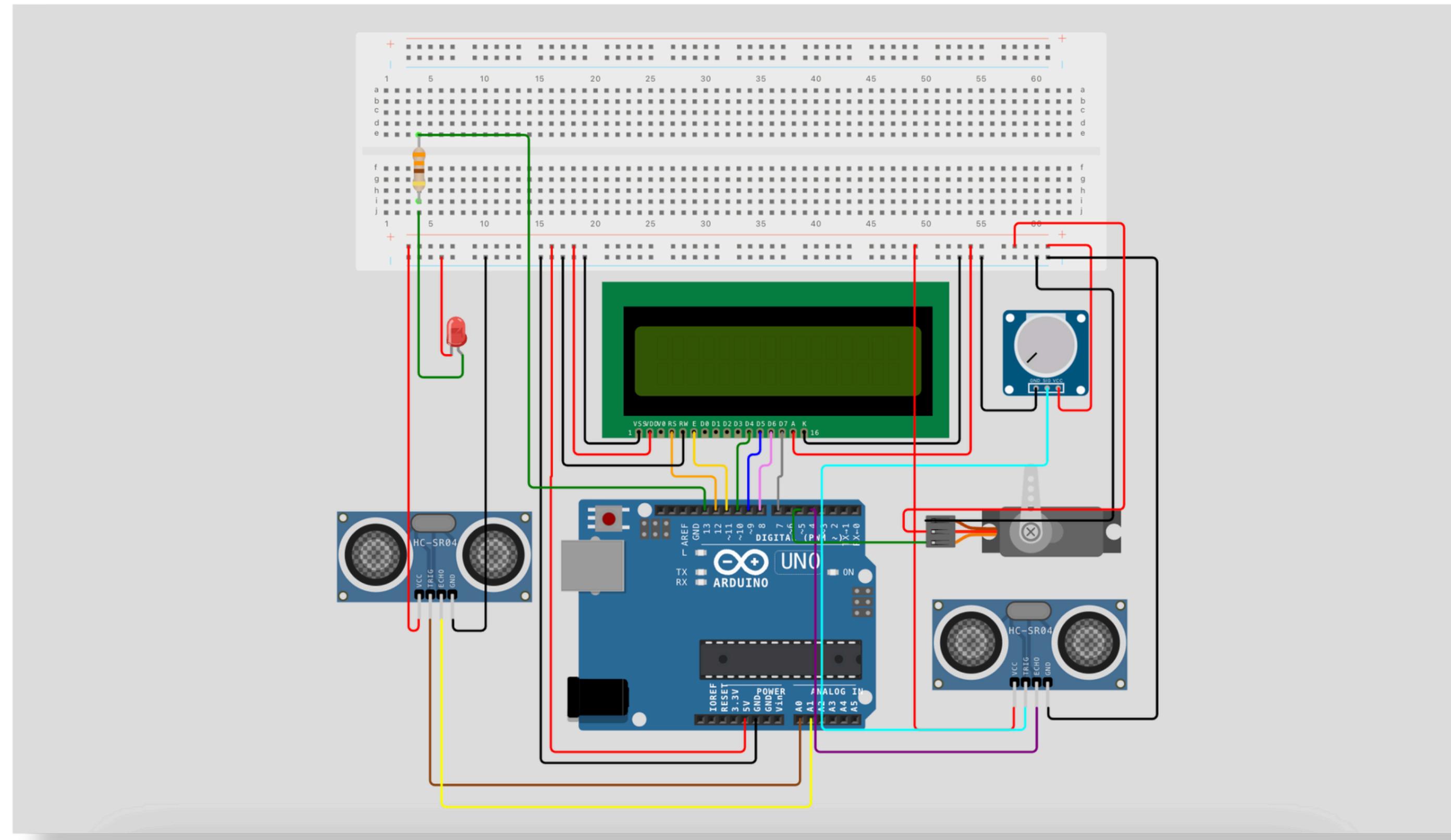
1. Convenience
2. Overflowing Bins
3. Hygiene

Overview

- Opens when you simply approach it
- Lid stays closed when the bin is full
- LCD display shows current waste level
- LED light turns red when the bin is full



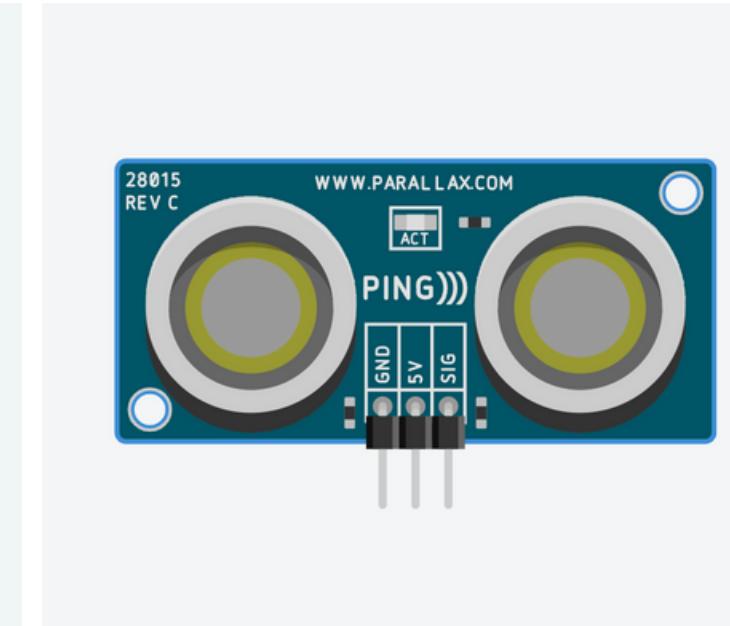
Simulation



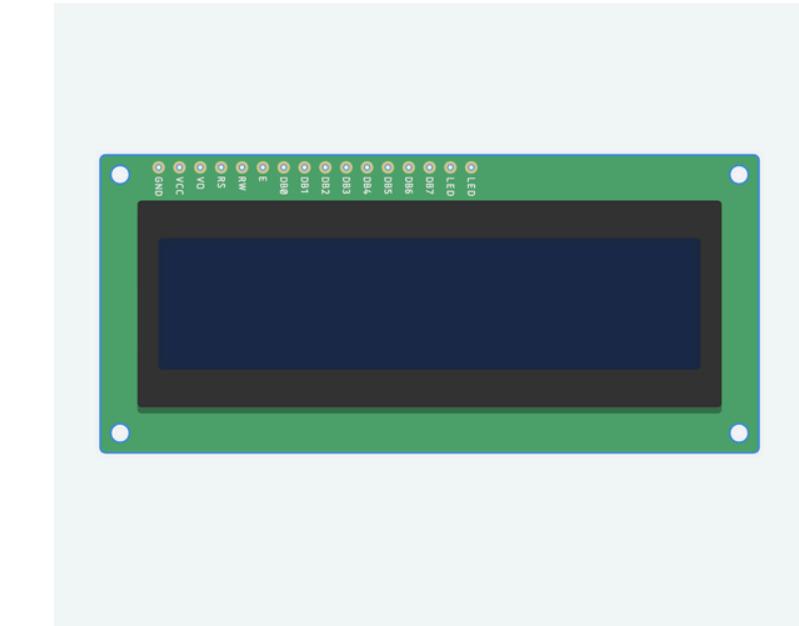
Hardware Used



Arduino UNO x 1



Ultrasound Sensor x 2



LCD x 1



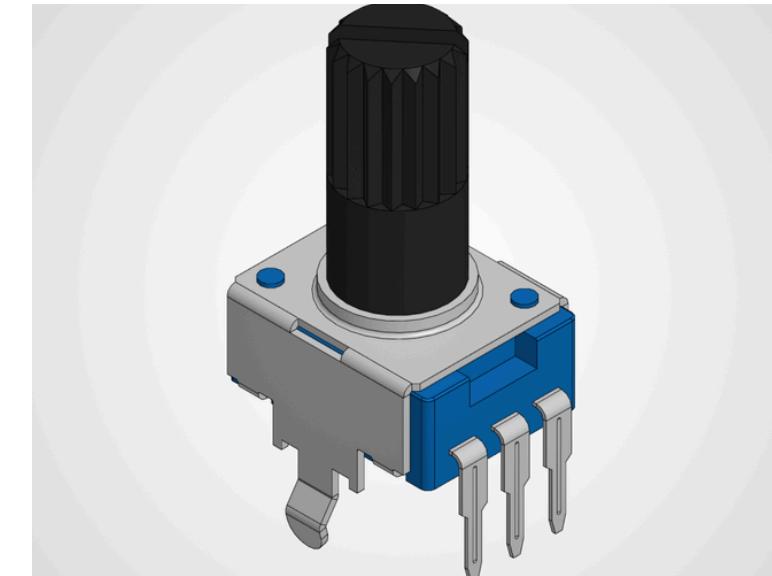
Servo Motor x 1



Jumper Cable x ∞



10mm LED x 1



Potentiometer x 1

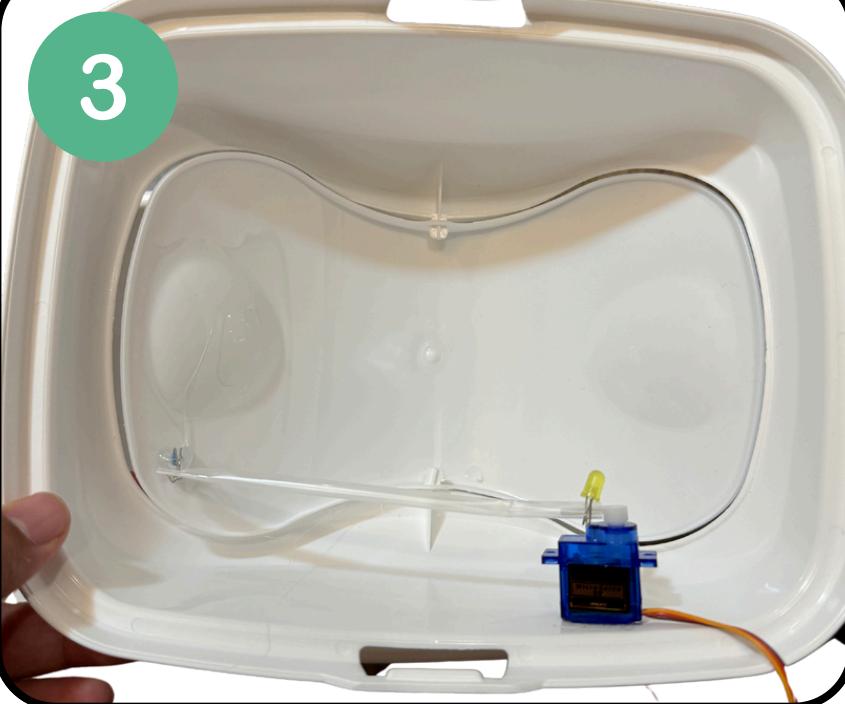
1



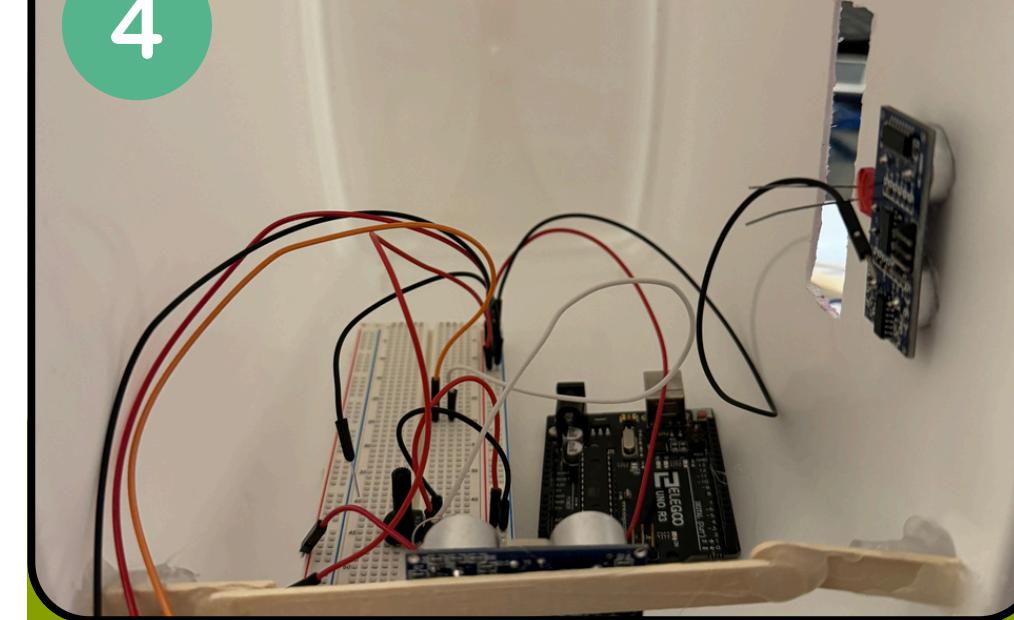
2



3



4



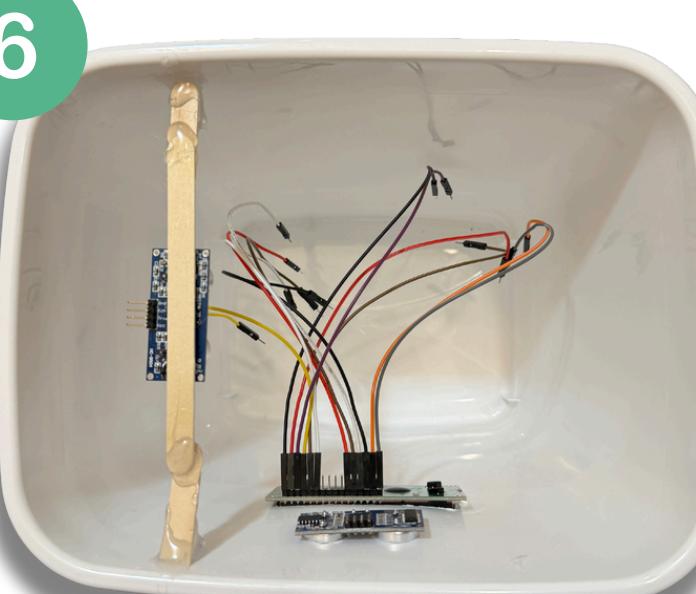
8



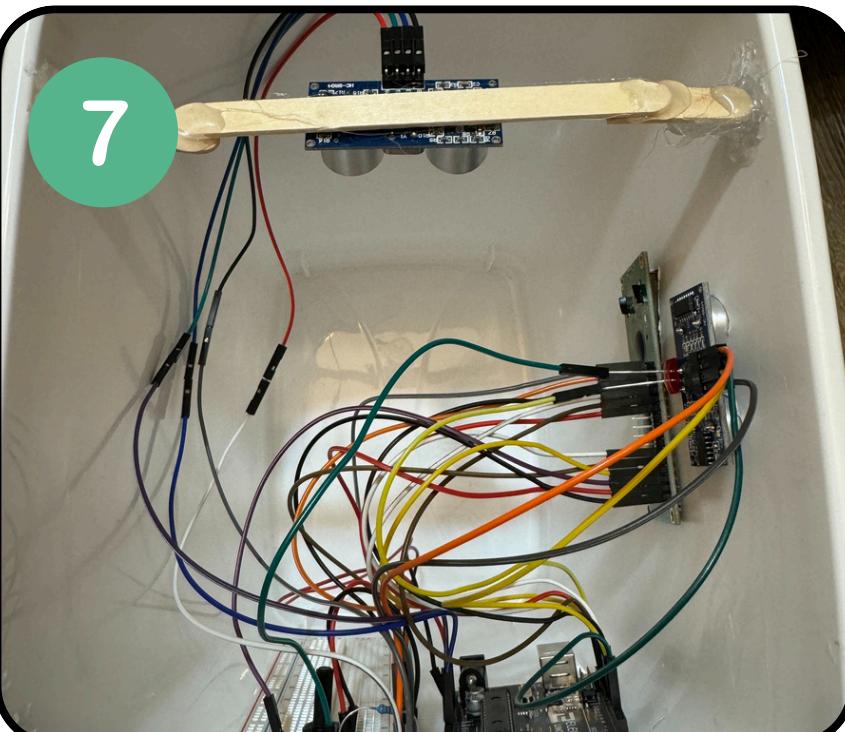
5



6



7



Phases

Technical Difficulties

A

Unexpected behavior and weird characters in LCD.



B

Servo and Timer 1 conflict

C

Timer 2 and New Ping conflict

D

Managing Wires

```
automatic_trash.ino  Servo.h  NewPing.h  X
140 // #include <pins_arduino.h>
141 #endif
142
143 #if defined (__AVR__)
144 #include <avr/io.h>
145 #include <avr/interrupt.h>
146 #endif
147
148 #if defined (PARTICLE)
149 #include <SparkIntervalTimer.h>
150 #endif
151
152 // Shouldn't need to change these values unless you have a specific need to do so.
153 #define MAX_SENSOR_DISTANCE 500 // Maximum sensor distance can be as high as 500cm, no reason to wait for ping longer than sound takes to travel thi
154 #define US_ROUNDTRIP_CM 57 // Microseconds (uS) it takes sound to travel round-trip 1cm (2cm total), uses integer to save compiled code space.
155 #define US_ROUNDTRIP_IN 146 // Microseconds (uS) it takes sound to travel round-trip 1 inch (2 inches total), uses integer to save compiled code
156 #define ONE_PIN_ENABLED true // Set to "false" to disable one pin mode which saves around 14-26 bytes of binary size. Default=true
157 #define ROUNDING_ENABLED false // Set to "true" to enable distance rounding which also adds 64 bytes to binary size. Default=false
158 #define URM37_ENABLED false // Set to "true" to enable support for the URM37 sensor in PWM mode. Default=false
159 #define TIMER_ENABLED false CHANGED THIS, "false" to disable the timer ISR (if getting "__vector_7" compile errors set this to false). Default=true
160
```

Code

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

const int maxWastePercentage = 96;
const int trigPin = 3;
const int echoPin = 4;
const int servoPin = 5;

int wastePercentage;
float durationWaste, distanceWaste;
float maxLevel = 23.0;

// Define pins and max distance:
#define wasteTrig A0
#define wasteEcho A1

#define MAX_DISTANCE 200 // Maximum distance we want to measure

NewPing sonar(wasteTrig, wasteEcho, MAX_DISTANCE);
Servo servo;

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
```

```
void setup() {
    lcd.begin(16, 2);
    lcd.print("...Welcome to...");
    lcd.setCursor(0, 1);
    lcd.print("    Crap Bag    ");
    Serial.begin(9600);
    servo.attach(servoPin);
    DDRB |= B100000; // pinMode(ledPin, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    servo.write(0);
    servo.detach(); // Detach the servo after ensuring
    // maxLevel = sonar.ping_cm();
    cli(); // Disable global interrupt

    // Configure Timer2
    TCNT2 = 0;
    TCCR2A = B00000000; // Set entire TCCR2A register to 0
    TCCR2B = B00000000; // Set entire TCCR2B register to 0

    OCR2A = 252; // Compare match every 16ns
    TCCR2A |= B00000100; // Set CTC Mode
    TCCR2B |= B00000111; // Set Prescaler of 1024
    TIMSK2 |= B00000010; // Set Timer2 CompA Value active
    sei(); // Enable global interrupt
}
```

```
// Checks the object distance from the sensor
long measureSensorDoor() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    float duration = pulseIn(echoPin, HIGH);
    float distdoor = (duration * 0.034) / 2.0;
    return distdoor;
}
```

```
// Check door according to wastePercentage
void checkDoor(long distdoor, int wastePercentage)
{
    if (wastePercentage < maxWastePercentage) {
        if (distdoor < 30 && distdoor > 0) {
            servo.write(360);
            delay(1000);
        } else {
            servo.write(0);
        }
    }
}
```

```
// Distance of Waste
int checkWasteDistance() {
    unsigned int wastePing = sonar.ping();
    Serial.print("Ping: ");
    unsigned int wasteInCm = sonar.convert_cm(wastePing);
    while (wasteInCm < 0) {
        unsigned int wastePing = sonar.ping();
        unsigned int wasteInCm = sonar.convert_cm(wastePing);
    }
    wastePercentage = 100 - ((wasteInCm / maxLevel) * 100);
    checkWasteThreshold(wastePercentage);
    Serial.print("Waste Percentage: ");
    Serial.println(wastePercentage);
    return wastePercentage;
}
```

```
void printWastePercentage(int wastePercentage) {
    if (wastePercentage < maxWastePercentage) {
        lcd.clear();
        lcd.print("Waste percentage:");
        lcd.setCursor(0, 1);
        lcd.print(wastePercentage);
        lcd.print("%");
        PORTB &= B011111;
    } else {
        PORTB |= B100000;
        servo.write(0);
        lcd.clear();
        lcd.print("Bin is FULL!!!");
    }
}
```

```
void checkWasteThreshold(int wastePercentage) {
    if (wastePercentage < 0) {
        wastePercentage = 1;
    } else if (wastePercentage >= 99) {
        wastePercentage = 100;
    }
    // Modify the global variable directly
    ::wastePercentage = wastePercentage;
}
```

```
void loop() {
    float distdoor = measureSensorDoor();
    servo.attach(servоСPin);
    checkDoor(distdoor, wastePercentage);
}

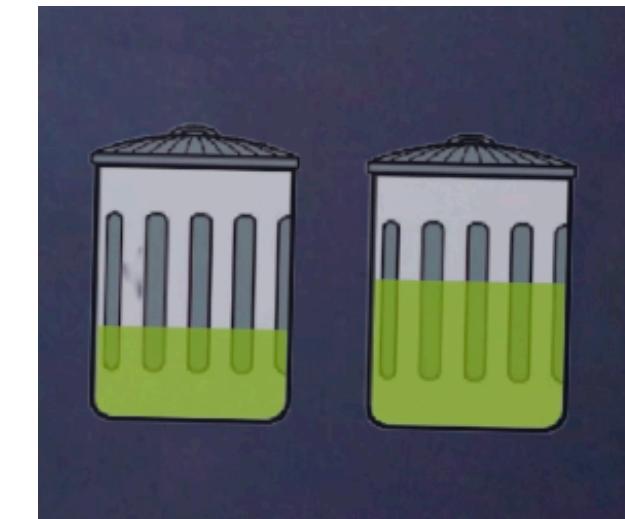
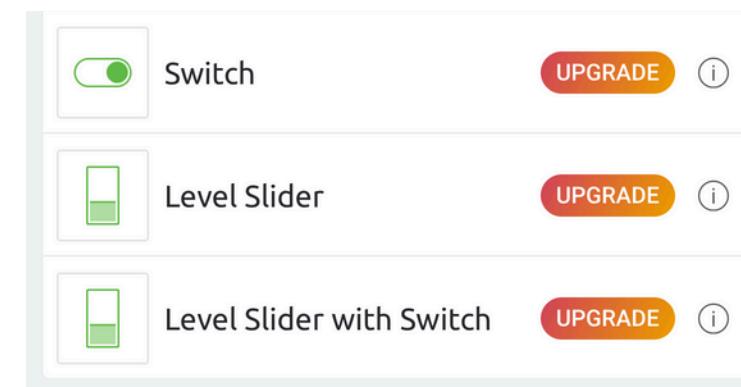
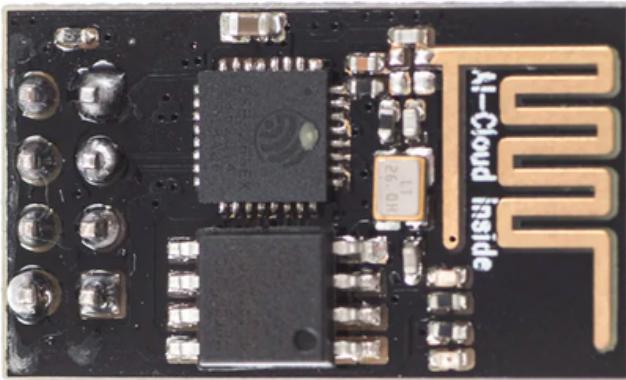
// ISR will trigger in 16ns
ISR(TIMER2_COMPA_vect) {
    int wastePercentage = checkWasteDistance();
    printWastePercentage(wastePercentage);
}
```

Conclusion

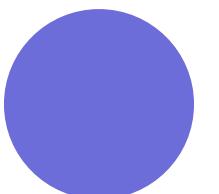
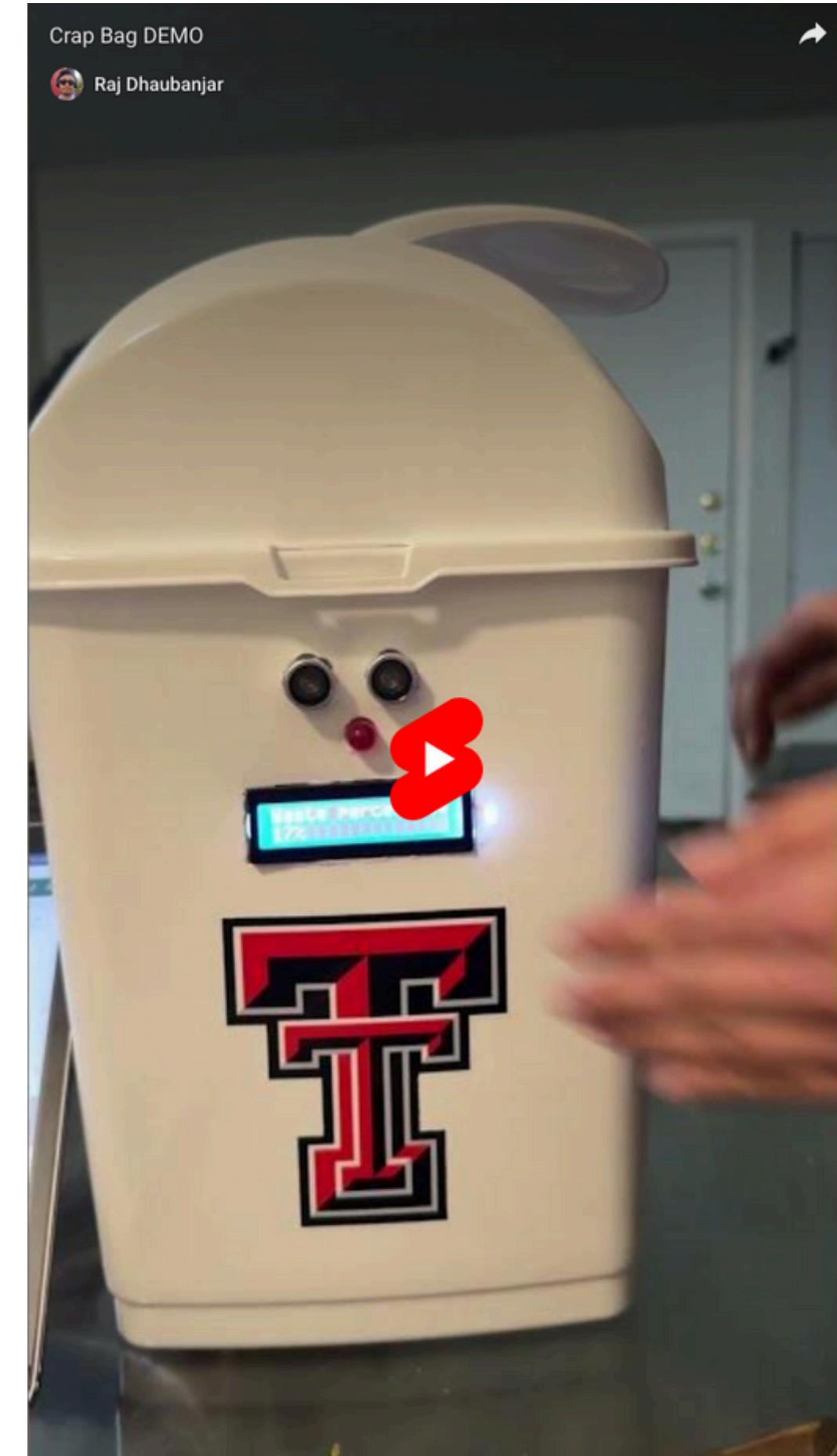
- Simple convenient smart trash bin prototype
- No Physical Contact, Hygienic
- Alert mechanism when it's full

Future Work

- Cooling and Air Freshener System for bad odor
- Adding wheels for movement
- Mobile App Integration for Monitoring



U
Demo



Question?

THANK YOU
For Watching