Electronic Speed Bump

Ip Aplication no. : SA 1020246466

Mohammed Yaser AlAbbas

- Abstract -

Electronic speed Bump aims to decrease speed bumps in traffic by lowering or raising them as needed. By using linear motor we can control the bumps and raise them at peak hours time or at the time students go out/in from school. This will give opportunity to decrease non needed speed bumps.

- Description -

- Background Of The Invention

An speed bump move with mechanism to have the ability to be lowered or high-ered via control system. Also it includes AI system to detect any jam, and send signals to lower or higher the bump.

- Description Of Invention

A deep and multiple control system to raise a speed bump up or move it down. the mechanism of the road built help in achieving the bump arch shape and give the chance to have smart, and electronic speed bump.

- Description Of Drawings Drawings are on page 4*

- shape 1 is about Applying a real-life prototype of a speed bump with the introduction of a linear motor
- shape 2 is about A preliminary 3d model of the parts required to build the speed bump and its mechanics to make it work smoothly
- shape 3 is about A 3d model that specifically explains how the mechanics of raising and lowering the speed bump work
- shape 4 is about The electrical circuit for the motors, lights used and all Electronic parts
- shape 5 is about Follow the electrical circuit

- Detailed Description

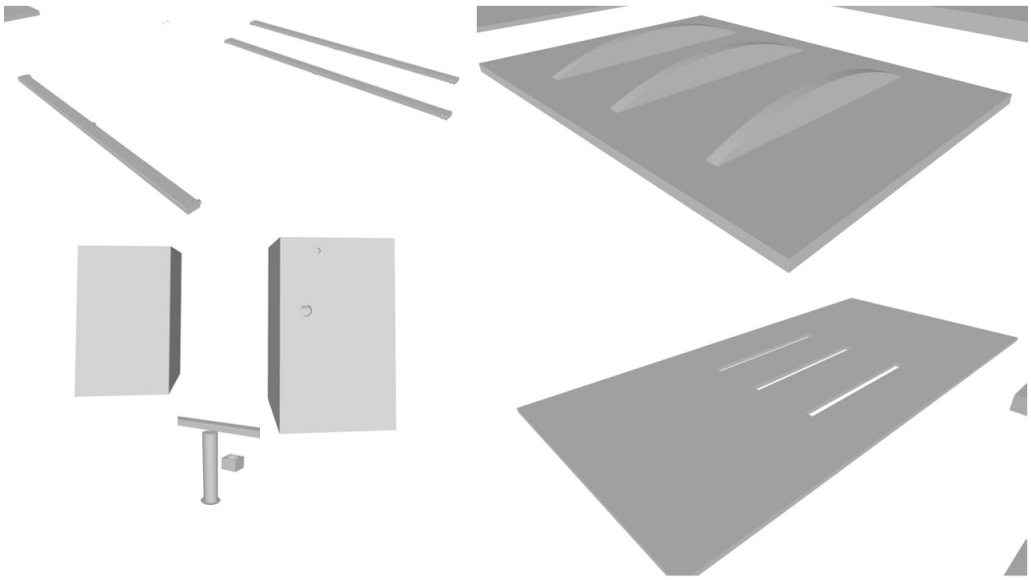
At first, Electronic speed bumps will solve major problems that we face in our daily lives. They can move the bump up and down as needed, allowing us to enhance traffic. By using a linear motor, we can direct the bump either up or down. Also, adding LEDs around the bump will give a clear visual for drivers to note the bump. It can move by schedules or by controlling it with a keypad and LCD display, and while the bump moving, the LED will change continuously to note the traffic that the bump state is changing. Of course, the controlling devices are protected by password to prevent manipulating them. The electric circuit is designed by Arduino Board, which all of the parts connected to it, send the codes to the other parts so the bump works without any issues. The Arduino board works on 5Volt, but the linear motor needs 12 volts to work, so adding an external H-Bridge to send enough power to both the board and motor. The mechanism of the design allows the bump to move up or down. Since the roadblocks above the bump are shuttered into pieces this will allow pushing the bump up while keeping its curve. The linear motor pushes arches toward the bump so it can be raised and shaped as a bump, and when the motor goes down the base will prevent the road from falling and the shape will be straight.

- The protection elements -

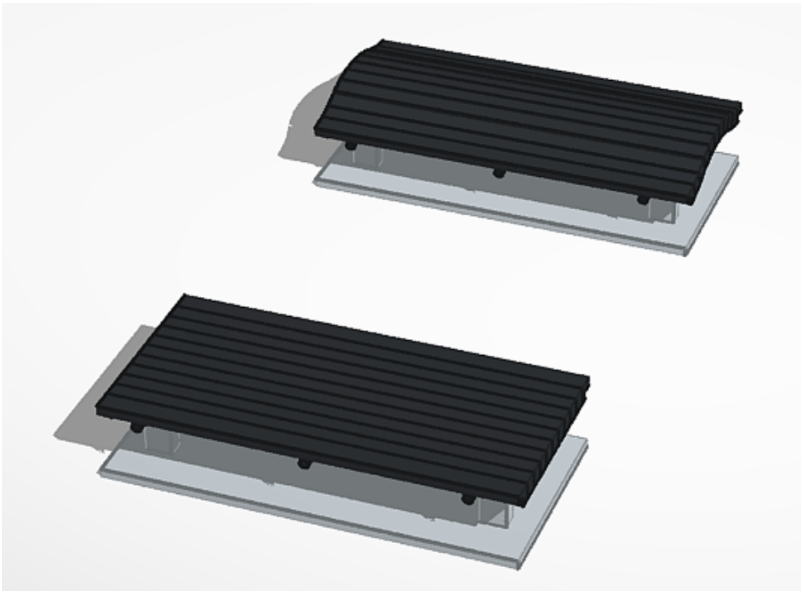
1. The mechanics of raising and lowering the speed bump are the basis of this invention and must be considered in the first place.
2. Tracking techniques (artificial intelligence) related to the speed bump.
3. The possibility of full manual remote control of all the properties of the bump.
4. The mechanism for controlling the height of the bump

- Graphics -

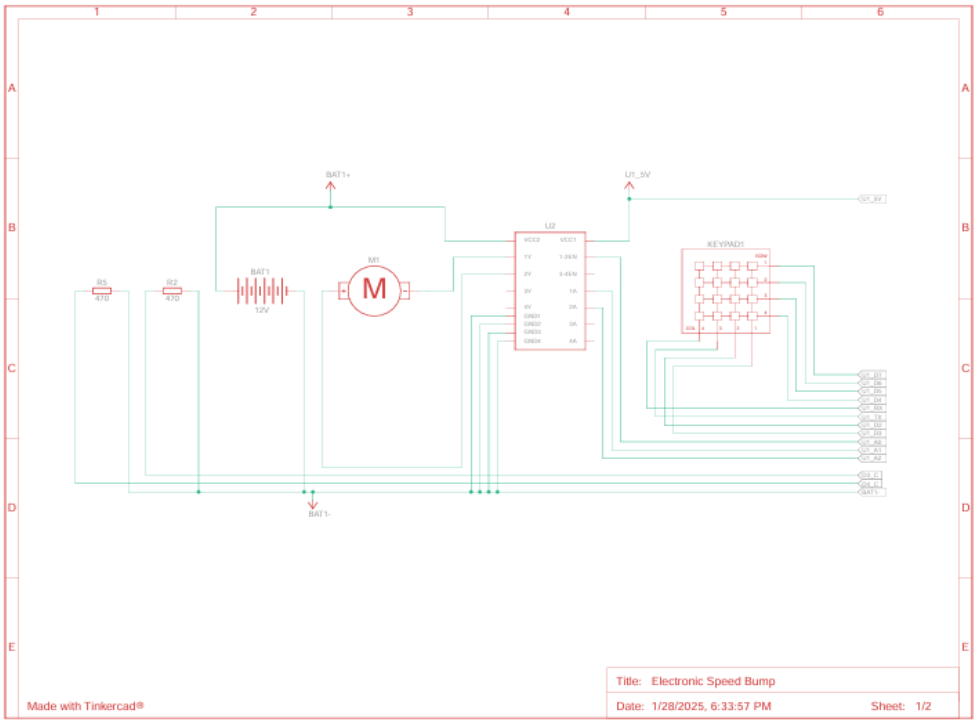
- Shape 1 -



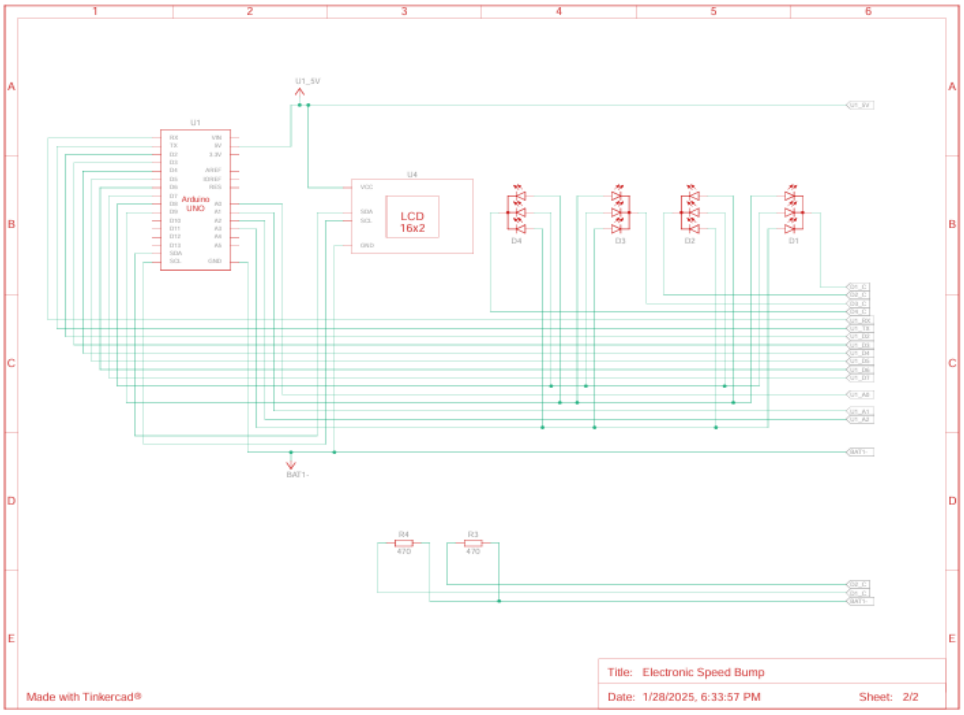
- Shape 2 -



- Shape 3 -



- Shape 4 -



- Shape 5 -

Request Id	SA 1020246466
Request status	Under Classification
Request title	Electronic Speed Bump
Filling Date	06 Jumada al-Ula 1446 | 08 November 2024 | 12:48 PM
Publication date	-
Grant date	-
Registeration number	-
Publication number	-
Publication count	-
Request owner 1	MOHAMMED YASER AL ABBAS
Request type	Prime
Agent	-

- Code -

```
#include <Wire.h>
#include "RTClib.h"
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

// Objects and constants
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
LiquidCrystal_I2C lcd(0x27, 16, 2);

int last_color;
const int red = 9;
const int blue = A3;
const int green = 8;
const int ENA_PIN = A0;
const int IN1_PIN = A1;
const int IN2_PIN = A2;

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] = {7, 6, 5, 4};
byte colPins[COLS] = {3, 2, 1, 0};

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

String inputPassword = "";
const String correctPassword = "1234";
char selectedAction = '0';
char selectedColor = '0';

struct Schedule {
    int hour;
    int minute;
    bool turnOn;
};

Schedule schedules[10];
int scheduleCount = 0;
```

```
void setup() {
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Hello!");
    delay(10);

    pinMode(red, OUTPUT);
    pinMode(blue, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(ENA_PIN, OUTPUT);
    pinMode(IN1_PIN, OUTPUT);
    pinMode(IN2_PIN, OUTPUT);

    digitalWrite(ENA_PIN, HIGH);
    digitalWrite(red, HIGH);
    last_color = 1;

    if (!rtc.begin()) {
        lcd.clear();
        lcd.print("RTC Error!");
        while (1);
    }
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

}

void loop() {
    DateTime now = rtc.now();
    checkSchedules(now.hour(), now.minute(), now.second());

    char key = keypad.getKey();
    if (key) {
        if (key == 'A' || key == 'B' || key == 'C' || key == 'D') {
            selectedAction = key;

            if (selectedAction == 'C') {
                lcd.clear();
                lcd.print("1)Red 2)Green");
                lcd.setCursor(0, 1);
                lcd.print("     3)Blue");

                while (true) {
                    key = keypad.getKey();
                    if (key == '1' || key == '2' || key == '3') {
                        selectedColor = key;
                        break;
                    }
                }
            }
            else {
                delay(1);
            }
```

```
            if (selectedAction) {
                lcd.clear();
                lcd.print("Enter Password:");
                inputPassword = "";

                while (true) {
                    key = keypad.getKey();
                    if (key) {
                        if (key == '*') {
                            checkPassword();
                            break;
                        }
                        else if (key == '#') {
                            if (inputPassword.length() > 0) {
                                inputPassword.remove(inputPassword.length() - 1);
                                lcd.setCursor(0, 1);
                                lcd.print(" ");
                                lcd.setCursor(0, 1);
                                lcd.print(inputPassword);
                            }
                        }
                        else {
                            inputPassword += key;
                            lcd.setCursor(0, 1);
                            lcd.print(inputPassword);
                        }
                    }
                }
            }
        }
        else {
            lcd.clear();
            lcd.print("Invalid Option");
            delay(1000);
            lcd.clear();
            lcd.print("Hello");
        }
    }
}

void checkPassword() {
    if (inputPassword == correctPassword) {
        executeAction();
    }
    else {
        lcd.clear();
        lcd.print("Wrong Password");
        delay(2000);
        lcd.clear();
        lcd.print("Hello");
    }
}
```

```
void executeAction() {
    lcd.clear();
    if (selectedAction == 'A') {
        turnMotorsOn();
    }
    else if (selectedAction == 'B') {
        turnMotorsOff();
    }
    else if (selectedAction == 'C') {
        changeLedColor();
    }
    else if (selectedAction == 'D') {
        if (scheduleCount < 10) {
            addSchedule();
        }
        else {
            lcd.clear();
            lcd.print("Max 10 Schedules");
            delay(2000);
            lcd.clear();
            lcd.print("Hello");
        }
    }

    selectedAction = '0';
    selectedColor = '0';
    lcd.clear();
    lcd.print("Hello");
}

void turnMotorsOn() {
    digitalWrite(IN1_PIN, HIGH);
    digitalWrite(IN2_PIN, LOW);

    lcd.print("Motors ON");
    digitalWrite(red, LOW);
    digitalWrite(green, LOW);
    digitalWrite(blue, LOW);

    for (int i = 0; i < 2; i++) {
        digitalWrite(red, HIGH);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);
        delay(500);

        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
        delay(500);
    }
```

```
    if (last_color == 1) {
        digitalWrite(red, HIGH);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
    }
    else if (last_color == 2) {
        digitalWrite(red, LOW);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);
    }
    else if (last_color == 3) {
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, HIGH);
    }

    delay(2000);
    lcd.clear();
    lcd.print("Hello");
}

void turnMotorsOff() {
    digitalWrite(IN1_PIN, LOW);
    digitalWrite(IN2_PIN, HIGH);

    lcd.print("Motors OFF");
    digitalWrite(red, LOW);
    digitalWrite(green, LOW);
    digitalWrite(blue, LOW);

    for (int i = 0; i < 2; i++) {
        digitalWrite(red, HIGH);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);
        delay(500);

        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
        delay(500);
    }
    digitalWrite(IN1_PIN, LOW);
    digitalWrite(IN2_PIN, LOW);
    if (last_color == 1) {
        digitalWrite(red, HIGH);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
    }
    else if (last_color == 2) {
        digitalWrite(red, LOW);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);
    }
```

```
    else if (last_color == 3) {
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, HIGH);
    }

    delay(2000);
    lcd.clear();
    lcd.print("Hello");
}

void changeLedColor() {
    lcd.clear();
    lcd.print("Changing Color:");

    if (selectedColor == '1') {
        digitalWrite(red, HIGH);
        digitalWrite(green, LOW);
        digitalWrite(blue, LOW);
        lcd.setCursor(0, 1);
        lcd.print("Red");
        last_color = 1;
    }
    else if (selectedColor == '2') {
        digitalWrite(red, LOW);
        digitalWrite(green, HIGH);
        digitalWrite(blue, LOW);
        lcd.setCursor(0, 1);
        lcd.print("Green");
        last_color = 2;
    }
    else if (selectedColor == '3') {
        digitalWrite(red, LOW);
        digitalWrite(green, LOW);
        digitalWrite(blue, HIGH);
        lcd.setCursor(0, 1);
        lcd.print("Blue");
        last_color = 3;
    }

    delay(2000);
}

void addSchedule() {
    lcd.clear();
    lcd.print("Set Hour:");
    int hour = getTimeInput();

    lcd.clear();
    lcd.print("Set Minute:");
    int minute = getTimeInput();
```

```
    lcd.clear();
    lcd.print("Set Action:");
    lcd.setCursor(1, 1);
    lcd.print("1) Up 2) Down");

    char key = '0';
    bool turnOn = false;

    while (key != '1' && key != '2') {
        key = keypad.getKey();
        if (key == '1') {
            turnOn = true;
        }
        else if (key == '2') {
            turnOn = false;
        }
    }

    schedules[scheduleCount].hour = hour;
    schedules[scheduleCount].minute = minute;
    schedules[scheduleCount].turnOn = turnOn;
    scheduleCount++;

    lcd.clear();
    lcd.print("Schedule Added");
    delay(2000);
    lcd.clear();
    lcd.print("Hello");
}

int getTimeInput() {
    String timeInput = "";
    lcd.setCursor(0, 1);

    while (true) {
        char key = keypad.getKey();
        if (key >= '0' && key <= '9') {
            timeInput += key;
            lcd.print(key);
        }
        else if (key == '*') {
            break;
        }
        else if (key == '#') {
            if (timeInput.length() > 0) {
                timeInput.remove(timeInput.length() - 1);
                lcd.setCursor(0, 1);
                lcd.print(" ");
                lcd.setCursor(0, 1);
                lcd.print(timeInput);
            }
        }
    }

    return timeInput.toInt();
}
```

```
void checkSchedules(int currentHour, int currentMinute, int currentSecond) {
    for (int i = 0; i < scheduleCount; i++) {
        if (schedules[i].hour == currentHour && schedules[i].minute == currentMinute && currentSecond
== 0) {
            if (schedules[i].turnOn) {
                turnMotorsOn();
                delay(1);
            }
            else {
                turnMotorsOff();
                delay(1);
            }
        }
    }
}
```

Electronic Speed Bump Using Arduino

Mohammed Yasser AlAbaas

Alkifah Academy

-Abstract

Electronic Speed Bump project aims to decrease traffic jam and save time by creating a speed bump that can be lowered or raised as needed. By using linear actuator motor, we can control the height of the speed bump. This motor is controlled via keypad which can be raised or lowed through. Additionally, RGB LED lights is implanted inside the speed bump and before it which provide visual cues to drivers enhancing awareness and compliance. The motor and LEDs are controlled by Keypad and LCD which user can control the whole system through with the need for password. This project combines designing and technology to improve traffic and save time.

Keywords—Speed Bump, Traffic, Speed System

Introduction

Speed Bumps at non-suitable times had been major concern in the early developing world. This can be improved by adding motor to the Speed Bump. For instance, a Speed Bump near school would not be important lately at the night so we can lower the bump to enhance the traffic. Speed Bumps cause a lot of damage which could be reduced by reducing the speed bumps. Confused.com, who conducted the study based on the replies of 2,000 drivers and Freedom of Information requests sent to local councils, found that almost half (48%) of the cars affected by speed bumps were subsequently found to have damage to tires, while 33% were left with suspension issues.[1]

To increase the safety and decrease traffic jam we must reduce speed bumps using electronic Speed Bump. The electronic speed bump focus mainly at the linear actuator motor which accomplish the main mission which is raising and lowering the speed Bump. This motor is controlled by 12Volt L298d H-Bridge. Which make it suitable and perfect to work for Arduino. Since Arduino support 5 volt as default [2] we use the external power to make it support 12Volt and handle the motor without any issues.

Methodology

This section will illustrate our project which aims to reduce traffic jam via electronic speed bump.

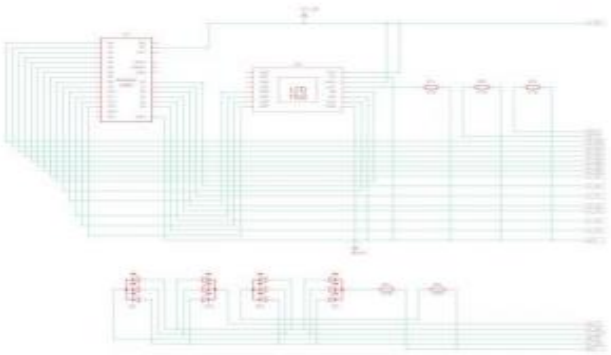
A – Design and Components

Figure 1a

As shown in Figure 1A the Arduino board fit for 20 pins in addition to the ground and power pins which make it suitable for keypad, lcd, linear motor, and 4 LEDs. This is perfect to go

Figure 2a

Figure 2a shows the heart of the system which bump electricity to all of the parts.



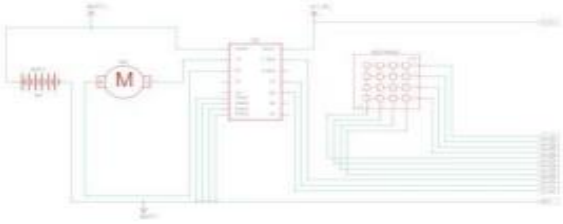


Figure 3a

As shown in Figure 3a the circuit model of the project contains the pins connections and the circuit diagram with the resistors that can handle 12V

B – Operational Process

Schedules: Controller could define and add schedules to control speed bump

Dynamic Adjustment: During non-peak hours or in situations where the bump is not needed, it can retract, leaving the road flat to reduce wear .and unnecessary slowdowns

Communication: The control unit can be programmed with specific schedules for schools (e.g., active during school .hours and retracted during off-hours)

C - Construction

The speed bump is made with strong motor to handle high traffic situations. Also, there would be underground room to save other components as the board to keep the bump work as long as possible.

Results

A – Time to change the state

The motor take 15 second to move all the way up or down which make it suitable for roads and cars moving on.

B – Improving traffic flow

During non-peak hours the bump will disappear with the road which help in moving the traffic

C – Safety

The speed bump effectively slows down vehicles during critical times, such as when students are arriving or leaving school, reducing the risk of accidents.

D – Decreasing Bumps

While using electronic speed bump the rate of bumps in non-peak hours will decrease significantly. This will allow soft movement fir the cars which will help the most in emergency time.

Conclusion

The electronic speed bump offers a flexible, responsive solution to speed control, particularly in sensitive areas like schools. By utilizing linear motors, it provides a more efficient and adaptive approach compared to traditional static speed bumps, improving both safety and traffic .management

References

[1] https://www.winnsolicitors.com/news/road-traffic-news/speed-bumps-cause-damage-to-one-in-five-cars

[2] https://forum.arduino.cc/t/power-max-input-voltage/976448

[3] www.tinkercad.com

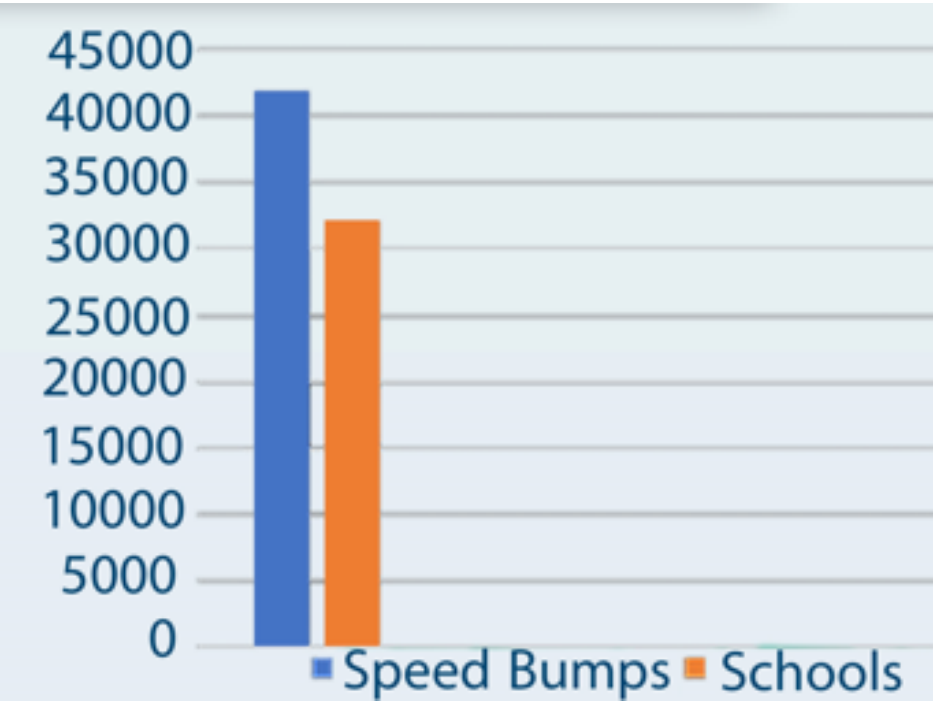
- Cost Analysis -

Prototype		Real-Life	
Motors	50$	Motors	2000$/motor
Lights	5$	Lights	10-20$/light
Material	70$	Material	4$/3.3kg
Work	0$	Work	200$
Electronics	120$	Electronics	160$
Total	245$	Needs for 2 way road	2 Motors - 10 light approx. - 3500 kg steel approx
		Needs for 3 way road	3 motors - 15 light approx. - 5500 kg steel approx.
		Total for 2 way road	8750$ approx.
		Total for 3 way road	13250$ approx.

- Is this high price?

Actually its not high price because, the target audience are governments, and they pay till 6000$ for side walks for 350 sqr feet and 2000-4000$ for street lights. At the road we can find hundreds of lights which will cost more than the Electronic Speed Bump. Off course the bumps doesn't exist as high as lights or side walks, so it will be suitable price and it will gain high profits in a short time.

- Data Analysis -



There are over 42,000 speed bump at UK, and at the same time there are 32163 school at UK. So, if each school had 1 bump at least then 32163 Electronic Speed Bump are needed at UK to enhance traffic.

- References -

1. Sinoconcept
2. Ecosmart
3. Homeguide
4. Steelexpress
5. Madar
6. Thomsonlinear

Electronic Speed Bump

Ip Application no. : SA 1020246466

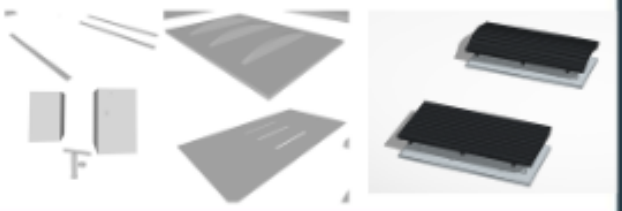
- Problem

At non-peak hours, there are no reason to have speed bumps at many places. They cause traffic jams and decrease vehichles speed for no reason.

- Goals

1 - To decrease speed bumps and create flexible traffic.
2 - To enhance the road development.
3 - removing bumps to save time
4 - create safe and modern road enviroment

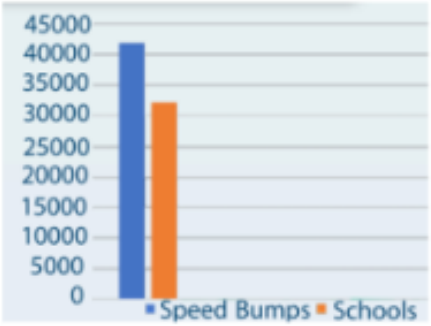
- Models



- Cost Analysis

Prototype		Real-Life	
Motors	50$	Motors	2000$/motor
Lights	5$	Lights	10-20$/light
Material	70$	Material	4$/3.3kg
Work	0$	Work	200$
Electronics	120$	Electronics	160$
Total	245$	Needs for 2 way road	2 Motors - 10 light approx. - 3500 kg steel approx
		Needs for 3 way road	3 motors - 15 light approx. - 5500 kg steel approx
		Total for 2 way road	8750$ approx.
		Total for 3 way road	13250$ approx.

- Data Analysis



- Mechanisim

To ensure that the bump moves smoothly we slice the road at the bump area ino several slides, which make it easy to manipulate the road shape. Then we add linear motor that push archs toward the bump which create the bump shape.

Mohammed AlAbbas Saudi Arabia, AlAhsa'a +966531461146 mohmoq5@hotmail.com