CONTENTS

INTRODUCTION

- Smart Parking system consists of an on-site deployment of an IOT module that is used to monitor and signalize the state of availability of each single parking space.

- Smart Parking is a parking system, usually a new one that is equipped with special structured devices (things) to detect the available parking slots at any parking area.

PROBLEM STATEMENT

➢With the growth of population and economical development, the number of vehicles on the road is increasing day by day. Parking is becoming one of the major problems for cities, and is becoming very costly .so to avoid Problems such as, traffic congestion, limited car parking facilities and road safety there is solution that is being addressed by Smart parking using IOT which is a solution to metropolitan cities to reduce congestion, cut vehicle emission totals and save persons' time by helping them in finding a spot to park.

REQUIREMENTS

1) **Arduino Uno:** This will act as the microcontroller for the project and all the other sensors will connected to it

2) **IR sensors:** It w o sense the presence of vehicle in the pa y sending out IR radiations.

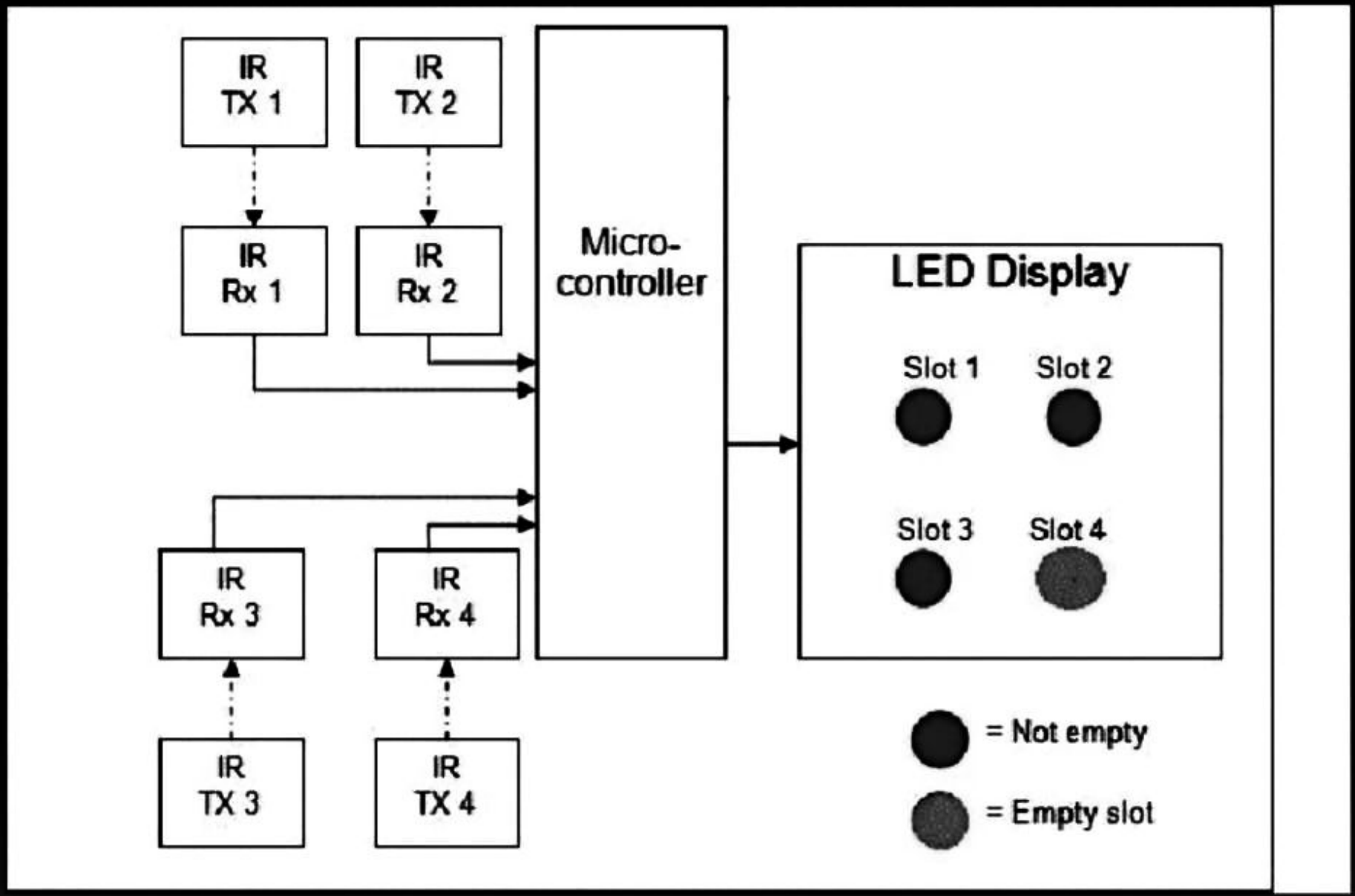
3)LED'S: This will be giving the indication if the particular slot is empty or full screen that is the status of the parking slots and change real-time

- As we are imple we as usually use jumper wires, bread board and USB cable.

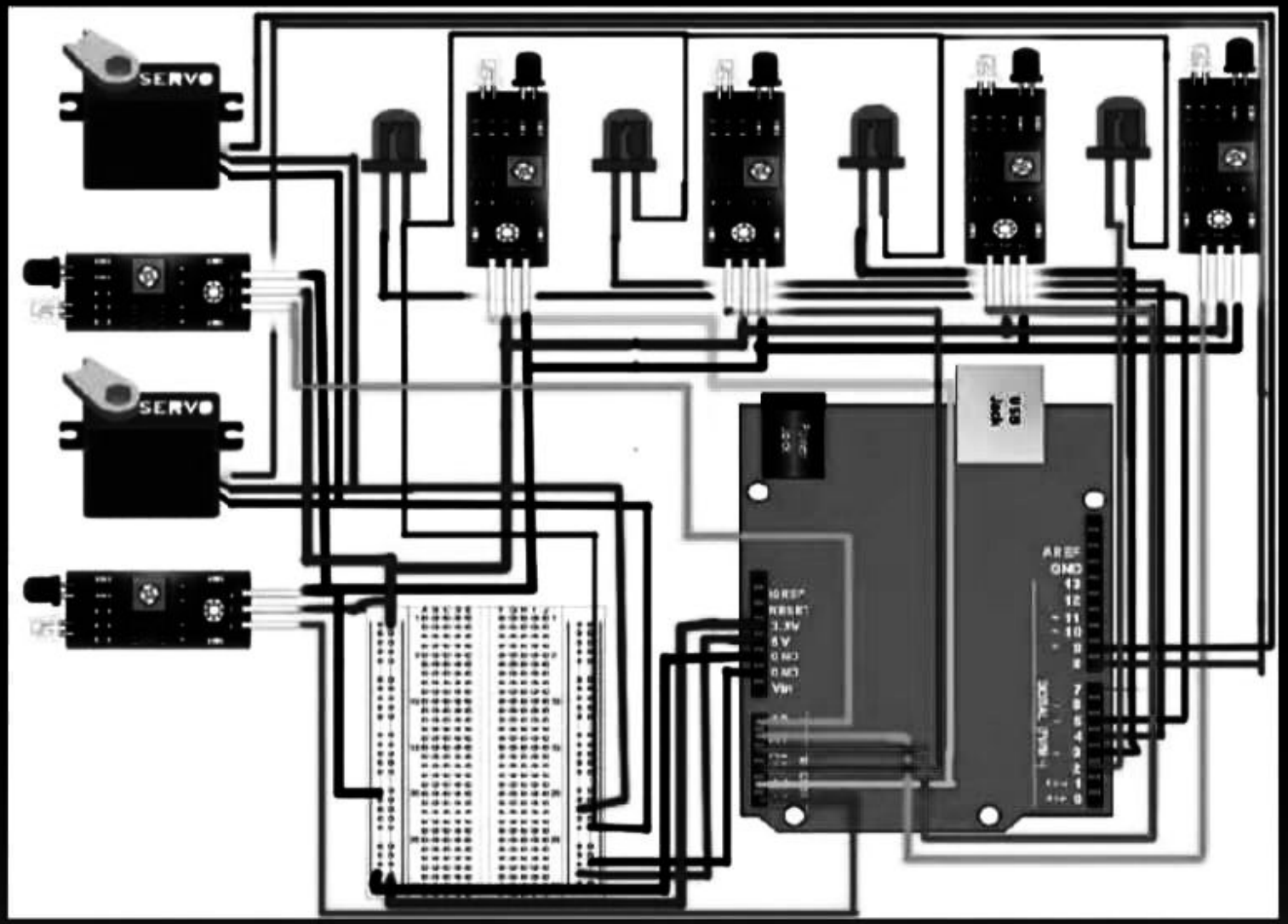
OUR DESIGN

- We are using 4 IR sensors in the particular parking slot. When a car or any vehicle enters the IR sensor in the Parking slot sends signal to Arduino and Arduino triggers the signal to LED which goes ON or OFF accordingly that indicates whether a parking slot is vacant or not.

BLOCK DIAGRAM



System Model



- The project involves a system including infrared transmitter and receiver in every lane and a LED display outside the car parking gate. So the person entering parking area can view the LED display and can decide which lane to enter so as to park the car.

AIMS AND OBJECTIVES

We Aim to create a system that:

- Enhance the security with Simplifying Parking System

- Parking System that pars a number of vehicles with east possible space

METHADOLOGY

- The IR sensors should be placed in the appropriate places to clearly cover all the parking slots
- The parking slots should be appropriately numbered to mark them on the system

- These marked points will act as the control points and will be integrated as slots in the cloud
- Then the setting will be saved and the microcontroller will be programmed to display the data online accordingly

PSEUDO CODE

```
const int analogInPin = A2;  // Analog input pin that the receiver is attached to
int sensorValue = 0;        // value read from the receiver

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  //initialize the indicator LED:
  pinMode(13, OUTPUT);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(A2);
```

```
// print the results to the serial monitor:
 Serial.print("\nsensor = ");
Serial.print(sensorValue);
 if(sensorValue < 383){ //checks if object is there or not
  digitalWrite(13, LOW);
Serial.print("\nObject Detected");
    }
else{
   digitalWrite(13, HIGH);
  Serial.print("\nNo object in Front");
 }
delay(500);
}
```

RESULTS

- INPUT: We implemented a code related to our parking system. the below figures are our code .


sketch_mar20a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_mar20a
const int analogInPin = A0; // Analog input pin that the receiver is attached to
int sensorValue = 0;   // value read from the receiver
void setup() {
// initialize serial communications at 9600 bps:
Serial.begin(9600);
//initialize the indicator LED:
pinMode(13, OUTPUT);
}
void loop() {
// read the analog in value:
sensorValue = analogRead(A0);
// print the results to the serial monitor:
Serial.print("\nsensor = ");
Serial.print(sensorValue);
if(sensorValue < 403)
{ //checks if object is there or not
digitalWrite(13, LOW);
Serial.print("\nObject Detected");
}
else{
digitalWrite(13, HIGH);
Serial.print("\nNo object in front");
}
delay(500);
}
Done uploading.
COM4 (Arduino/Genuino Uno)
Send
sensor = 402
Object Detected
sensor = 403
No object in front
sensor = 401
Object Detected
sensor = 402
Object Detected
sensor = 403
No object in front
sensor = 404
No object in front
sensor = 405
No object in front
sensor = 405
No object in front
Autoscroll
No line ending
9600 baud
Clear output

OUTPUT