PROJECT REPORT

IoT Device Essentials

"Smart Parking System"

By:
Soni Shrishail Salgar
(sonisalgar11298@gmail.com)

SBID: SB20200033727

Project done at SmartInternz

INDEX

1.Introduction

1.1 Overview:

In this project a new parking system called Smart Parking System (SPS) is proposed to assist drivers to find vacant spaces in a car park in a shorter time. The new system uses ultrasonic sensors to detect either car park occupancy or improper parking actions.

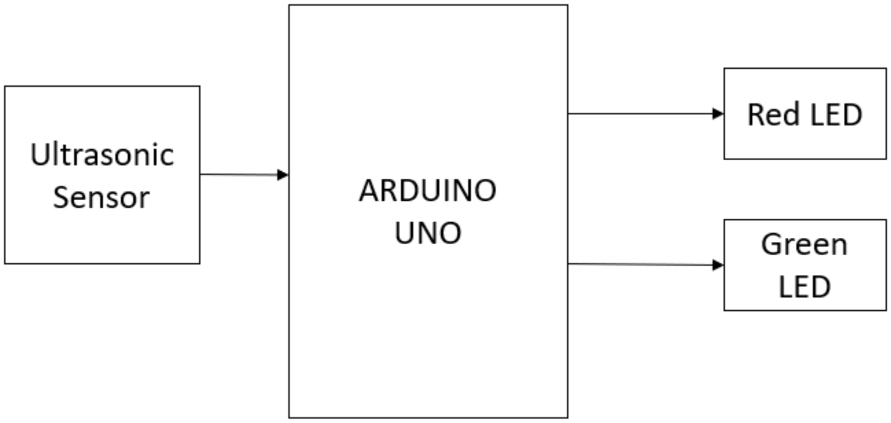
1.2 Purpose:

Due to increase in the number of vehicles on road, traffic problems are bound to exist. This is due to the fact that the current transportation infrastructure and car park facility developed are unable to cope with the influx of vehicles on the road. To alleviate the aforementioned problems, the smart parking system has been developed. With the implementation of the smart parking system, patrons can easily locate and secure a vacant parking space at any car park deemed convenient to them.

Features of SPS include vacant parking space detection through the use of specific LEDs.

2.Theoritical Analysis

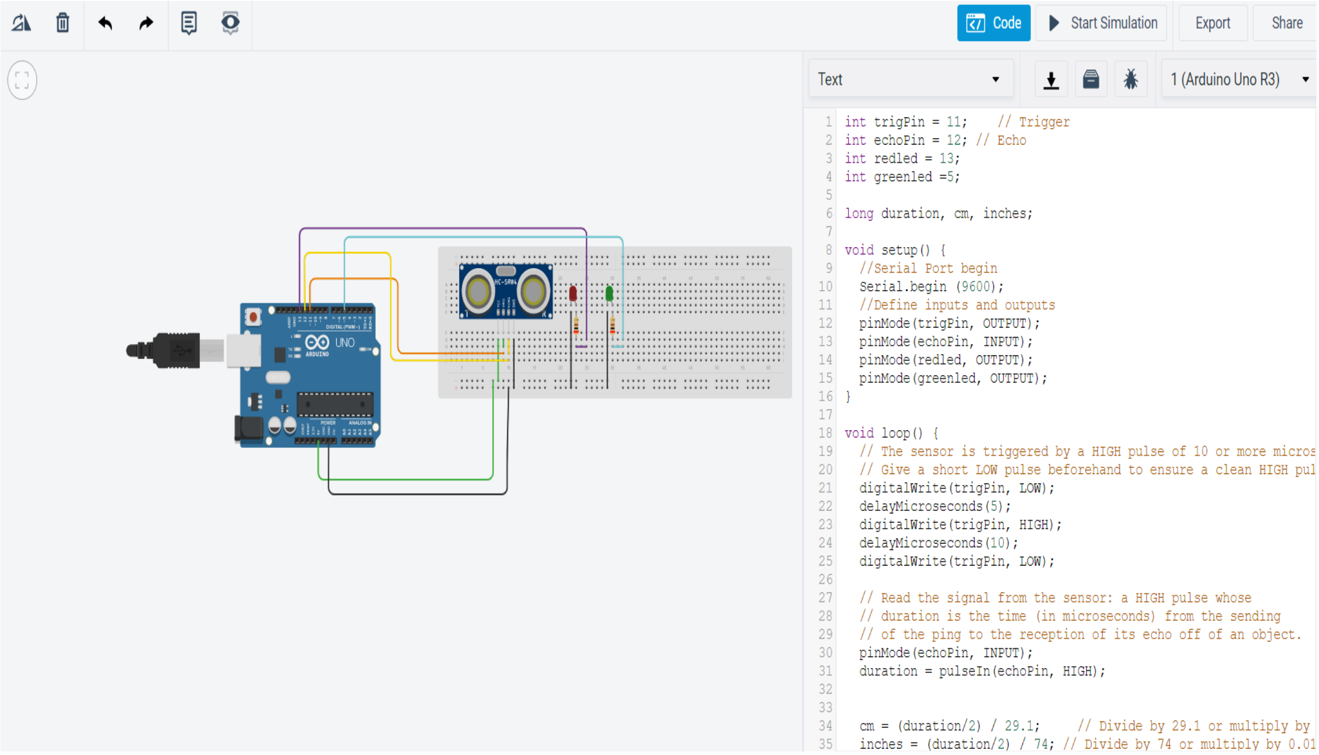
2.1 Block Diagram:



2.2 Hardware / Software Designing:

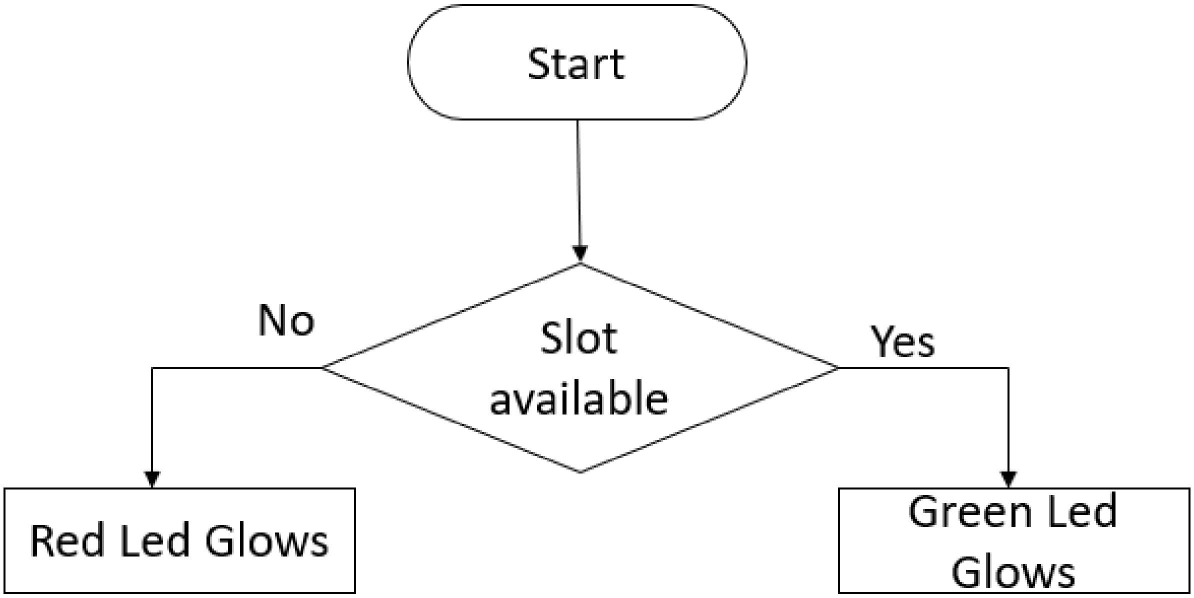
1. Build Circuit on Tinkercad.
2. Make proper connections.
3. Write the required code in text window of tinkercad and start the simulation.

3.Experimental Investigations



```
int trigPin = 11;    // Trigger
int echoPin = 12; // Echo
int redled = 13;
int greenled =5;

long duration, cm, inches;

void setup() {
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redled, OUTPUT);
  pinMode(greenled, OUTPUT);
}

void loop() {
  // The sensor is triggered by a HIGH pulse of 10 or more micros
  // Give a short LOW pulse beforehand to ensure a clean HIGH pul
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);


  cm = (duration/2) / 29.1;     // Divide by 29.1 or multiply by
  inches = (duration/2) / 74; // Divide by 74 or multiply by 0.01
```

4. Flowchart



5. Result:

We have successfully buld a Smart Parking System device which will indicate weather the parking is available or not.

6. Advantages and Dis-advantages:

6.1 Advantages :

1. Easy indication for availability of parking space
2. Less chances for vehicle vandalism.
3. Emissions are greatly brought down and reduced.
4. There is a minimal staff requirement.
5. Can be used in a dark parking lot, facilitating parking in a smooth manner.

6.2 Dis-advantages:

1. There is a greater construction cost per space.
2. It may be a bit confusing for unfamiliar users.
3. It requires a maintenance contract with the supplier
4. Any obstacle that is present along the sides of the car will go unnoticed.

7. Applications:

1. Easily indicates about new parking areas available.
2. Can be used in Public places like shopping malls, theaters, restaurants, etc.
3. Private parking spaces for safe parking.
4. Reduces traffic by detecting empty spaces.

8. Conclusion:

A model for smart parking system has been created on Tinkercad, using Arduino.

9. Future Scope:

A web Application can be created for getting the data on mobile like devices for reducing traffics in public places, roads and more also detecting the parking availability at different locations using GPS.

10. Bibliography:

1. Tinkercad :

   https://www.tinkercad.com/

2. Arduino Uno :

   https://store.arduino.cc/usa/arduino-uno-rev3

3. Ultrasonic sensor :

https://randomnerdtutorials.com/courses/

11.Apendix

11.1 Source Code:

```
int trigPin = 11;    // Trigger
int echoPin = 12; // Echo
int redled = 13;
int greenled =5;

long duration, cm, inches;

void setup() {
  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redled, OUTPUT);
  pinMode(greenled, OUTPUT);
}

void loop() {
  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);


  cm = (duration/2) / 29.1;     // Divide by 29.1 or multiply by 0.0343
  inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135


  if (cm < 20)

   {

   digitalWrite(redled,HIGH);
   delay(250);
   digitalWrite(redled,LOW);
   delay(250);

   }

  else {

    digitalWrite(greenled,HIGH);
    delay(250);
    digitalWrite(greenled,LOW);
    delay(250);

       }
  Serial.print(inches);
  Serial.print("in, ");
```

```
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(250);
}
```