

---

## SMART PARKING SYSTEM

---

### TEAM MEMBERS:-

- 1:-M.Alagu Sathiya -au951221106003
- 2:-M.Abirami -au951221106002
- 3:-S.Rajathi -au951221106030
- 4:-S.Sasi Rekha -au951221106043
- 5:-M.Venuka -au951221106052

### PHASE-3:-Development Part-1:-

#### AI based Smart Parking Assistant:-

Ultrasonic, electromagnetic field detection, and infrared are several types of iot smart parking sensors.

#### 1. **Ultrasonic:**

- 2. The precision of the smart parking sensor is improved by using ultrasonic wave. The disadvantage of this type of sensor is that it can get clogged with dirt.

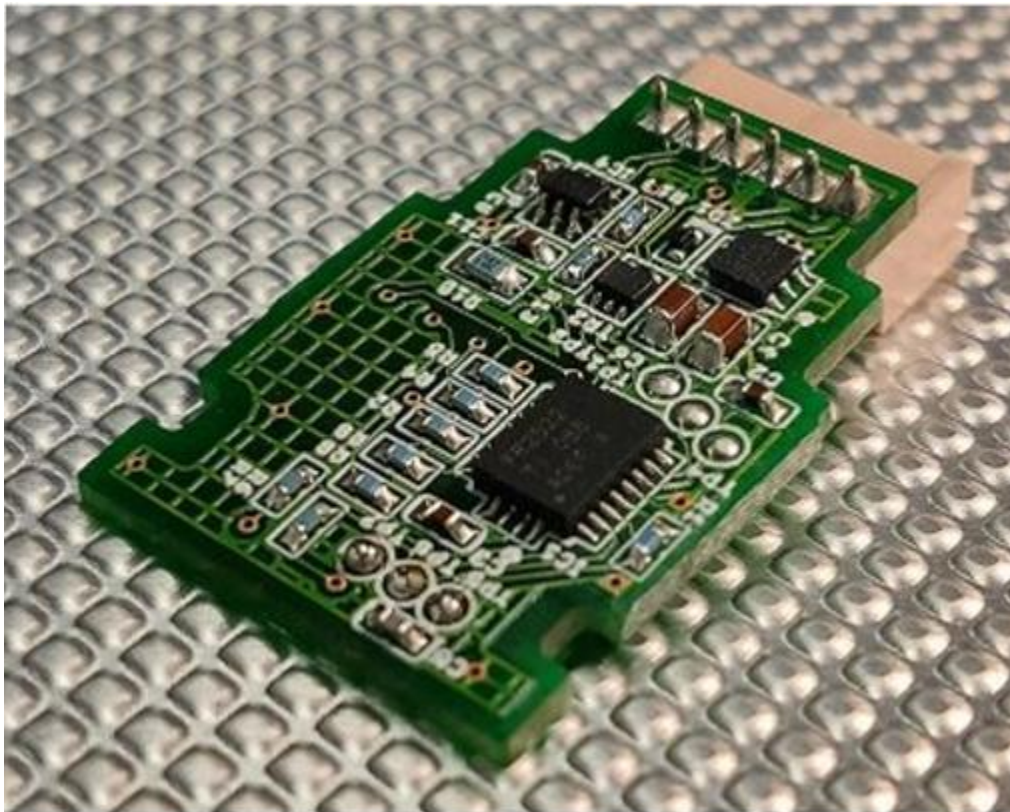


3.

4. **Electromagnetic Field Detection:**

5. The sensor can detect small changes in the magnetic field when a metal object is near it.

6.



7. **Infrared:**

8. This type of sensor measures changes in ambient temperature and detects movement.

**Coding:-**

```
//TECHATRONIC.COM
```

```
// BLYNK LIBRARY
```

```
// https://github.com/blynkkk/blynk-library
```

```
// ESP8266 LIBRARY

// https://github.com/ekstrand/ESP8266wifi

#define TRIGGER D0

#define ECHO D2

// NodeMCU Pin D0 > TRIGGER | Pin D2 > ECHO

#define BLYNK_PRINT Serial // Comment this out to disable prints
and save space

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).

Char auth[] = "Whobi6tSCicbj4W654WdBeo7O4D6Ajwt4"; //Auth
code sent via Email

// Your WiFi credentials.
// Set password to "" for open networks.

Char ssid[] = "DESKTOP"; //Wifi name
Char pass[] = "asdfghjkl"; //Wifi Password

Void setup() {
  Serial.begin (9600);

  Blynk.begin(auth, ssid, pass);

  pinMode(TRIGGER, OUTPUT);

  pinMode(ECHO, INPUT);
```

```
pinMode(BUILTIN_LED, OUTPUT);  
}  
  
Void loop() {  
  Long duration, distance;  
  digitalWrite(TRIGGER, LOW);  
  delayMicroseconds(2);  
  digitalWrite(TRIGGER, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(TRIGGER, LOW);  
  duration = pulseIn(ECHO, HIGH);  
  distance = (duration/2) / 29.1;  
  if (distance <=200) {  
    Blynk.virtualWrite(V0, 255);  
  }  
  Else {  
    Blynk.virtualWrite(V0, 0);  
  }  
  If (distance <= 35) {  
    Blynk.virtualWrite(V1, 255);  
  }  
  Else {  
    Blynk.virtualWrite(V1, 0);  
  }  
}
```

```
}  
  If (distance <= 30) {  
    Blynk.virtualWrite(V2, 255);  
  }  
  Else {  
    Blynk.virtualWrite(V2, 0);  
  }  
  If (distance <= 25) {  
    Blynk.virtualWrite(V3, 255);  
  }  
  Else {  
    Blynk.virtualWrite(V3, 0);  
  }  
  If (distance <= 20) {  
    Blynk.virtualWrite(V4, 255);  
  }  
  Else {  
    Blynk.virtualWrite(V4, 0);  
  }  
  Serial.print(distance);  
  Serial.println("Centimeter:");  
  Blynk.virtualWrite(V5, distance);
```

```
delay(200);  
Blynk.run();  
Serial.print(distance);  
Serial.println("Centimeter:");  
Blynk.virtualWrite(V6, distance);  
delay(100);  
Blynk.run();  
}
```