Phase 2: Innovation

Smart parking systems are beginning to provide answers for urban transporation as a result of digitization.This system enables the acquisition of real-time data about parking availability and information about traffic and road conditions. This is made possible through the Internet of Things and sensor technologies

After through research and analysis,we arrived at an innovative solution to solve the problem.

Microcontroller :

Esp32:

Implementation on smart parking using Esp32 cam will provide information as well as detect cars provided for parking users.Internet of Things or IOT has begum to be widely recognized by humans.

Arduino Uno:

The Arduino Uno is used to create a smart car parking system.The device uses IR sensors mounted in the parking slots to detect emoty slots and assists the driver in finding parking in a new city.

Sensor

The most widespread use of the sensor is,of course,parking assistance systems in modern cars:the sensors are affixed in

rear and front bumpers and help you park by signaling proximity to obstructions in the short range

Wireless sensor technology:To provide real-time parking data to drivers,the Internet of Things is foremost in importance.Smart parking companies have installed more than 10,000 wireless devices worldwide,and many of these systems can include guidance when visitors try to find space in your company's car park or at any other public space.

DETECTOR

Radar sensors for vehicle detection in parking lot.It is suitable for performing a detection if a vehicle parks in a parking spot

connectivity

1) BLE

2) WIFI

3)ZIGBEE

BLUETOOTH

In this work,a smart parking system that operates both indoor and outdoor is introduced.The system is based on Bluetooth Low Energy(BLE) beacons and users particle filtering to improve its accuracy.Through simple BLE connectivity with smartphones, an intuitive parking system is designed and employed.

WIFI

The central server is installed and connected via the wi-fi network,and it receives all the information of all integrated web servers of each parking available in the city.This system uses a mobile application so that drives can get information on vacant parking spaces.

ZIGBEE

It provides connectivity with zigbee end nodes installed at parking slots using zigbee fruquency(2.4GHz or 868/915MHz).It has connectivity with wifi and cellular networks as well.

PROTOCOL

MQTT(Message Queuing Telemetry Transport

MQTT is a messaging protocol for restricted low-bandwidth networks and extremely high-latency IOT devices.

MQTT is well-suited for IOT due to its efficient message distribution system.

MQTT is a lightweight publish-subscribe messaging protocol designed for low-bandwidth,high-latency or fluctuating networks.

CLOUD

Smart parking launched smartcloud after partnering with Geogle during their deployment of the Geogle cloud IOT core platform, and it allows a complete solution for connecting,managing,and reporting on car park usage.cloud platforms are used to store and process data collected by the system.

SMART CAMERAS

Smart cameras are used to monitor parking spots and detect the presence of vehicles.

GPS SYSTEMS:

GPS systems are used to track the location of vehicles and provide drivers with information about available parking spots.

LED

Display the state of the car park (occupied or empty)

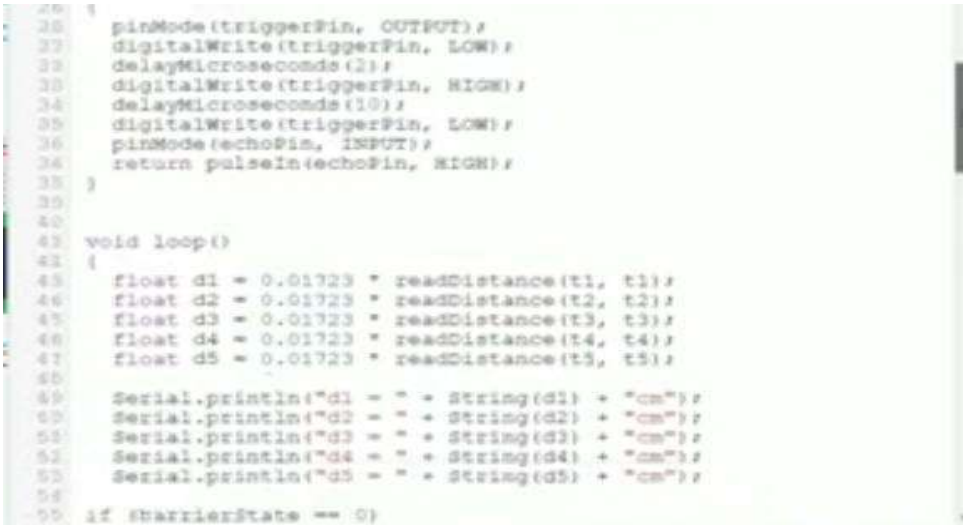
Payment and reservations:

Smart parking systems often offer digital payment options and allow users to reserve parking spaces in advance. This reduces the time and effort required for parking transactions.

DATA COLLECTION:

Data from sensors are collected and transmitted to a central management system.This data can include information on available parking spaces,occupancy rates,and parking durations.

```
#include <Servo.h>

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define t1 10
#define t2 9
#define t3 8

#define t4 7
#define t5 13

Servo myservo;

int distanceThreshold = 100;

int parkingAvailable = 3;
int barrierState = 0;

void setup() {
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  Serial.begin (9600);
  myservo.attach(6);
  myservo.write(0);
}

long readDistance(int triggerPin, int echoPin)
```



```
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}


void loop()
{
  float d1 = 0.01723 * readDistance(t1, t1);
  float d2 = 0.01723 * readDistance(t2, t2);
  float d3 = 0.01723 * readDistance(t3, t3);
  float d4 = 0.01723 * readDistance(t4, t4);
  float d5 = 0.01723 * readDistance(t5, t5);

  Serial.println("d1 = " + String(d1) + "cm");
  Serial.println("d2 = " + String(d2) + "cm");
  Serial.println("d3 = " + String(d3) + "cm");
  Serial.println("d4 = " + String(d4) + "cm");
  Serial.println("d5 = " + String(d5) + "cm");

if (barrierState == 0)
```

```
if (barrierState == 0)
{
  if (d4<100 && parkingAvailable>0)
  {
     parkingAvailable -= 1;
     barrierState = -1;
     myservo.write(90);
  }
  if (d5<100 && parkingAvailable<3)
  {
     parkingAvailable += 1;
     barrierState = 1;
     myservo.write(90);
  }
}
else if (barrierState == -1)
{
  if (d4>=100 && d5<100)
  {
     barrierState = -2;
     myservo.write(0);
  }
}
 else if (barrierState == 1)
{
```

```
    if (d5>=100 && d4<100)
  {
     barrierState = 2;
     myservo.write(0);
  }
}
 else if (barrierState == -2)
{
    if (d5>=100)
  {
     barrierState = 0;
  }
}
else if (barrierState == 2)
{
    if (d4>=100)
  {
     barrierState = 0;
  }
}
lcd.setCursor(0,0);
if (parkingAvailable == 0)
{
  lcd.print("Parking Full  ");
}
else
{
  lcd.print("Parking left ");
```

```
{
  lcd.print("Parking left ");
  lcd.print(parkingAvailable);
}

if (d1>100 & d2>100 & d3>100)
{
    lcd.setCursor(0,1);
    lcd.print("Slot 1 2 3 Free");
    delay(500);
    }
else if((d1>100 & d2>100)||(d2>100 & d3>100)||(d3>100 & d1>100))
  {
    lcd.setCursor(0,1);
    if(d1>100 & d2>100)
        lcd.print("Slot 1 & 2 Free");
    else if(d1>100 & d3>100)
        lcd.print("Slot 1 & 3 Free");
    else
        lcd.print("Slot 2 & 3 Free");
    delay(500);
    }
else if(d1<100 & d2<100 & d3<100)
  {
    lcd.setCursor(0,1);
    lcd.print("Parking Full   ");
```

```
    lcd.print("Parking Full   ");
    delay(500);
    }
else if((d1<100 & d2<100)||(d2<100 & d3<100)||(d3<100 & d1<100))
  {
    lcd.setCursor(0,1);
    if(d1>100)
      lcd.print("Slot 1 is Free ");
    else if (d2>100)
       lcd.print("Slot 2 is Free ");
    else
       lcd.print("Slot 3 is Free ");
  delay(500);
   }
  delay(100);
}
```