Smart Car Parking Mechanism

By Shreerang Mhatre

Introduction:

Urbanization is on the rise, and the complexity of city traffic is growing on a daily basis, especially post-COVID when everyone seems to be out on the road. While public transportation is often unable to cater to the needs of all commuters, finding parking for personal vehicles has become an even more daunting and expensive task. Therefore, the requirement for an Internet of Things (IoT)-powered parking system has gained prominence to minimize ongoing traffic congestion and reduce parking availability's unpredictability. With numerous connected devices on the horizon, devising an IoT-based smart parking system is relatively easy now. In fact, such infrastructure empowers drivers and parking facility managers with advanced information such as available slots and current parking fees.

The implementation of an IoT-based smart parking system also opens up possibilities for integration with other urban systems. For instance, it can be linked with navigation applications to guide drivers directly to available parking spaces, minimizing unnecessary circling around city blocks. Additionally, data collected from the smart parking system can be analyzed to identify parking trends and support urban planning initiatives.

Smart Parking System:

The Smart Parking System aims to provide real-time information about parking space availability in a specific area. This technology makes it easier for drivers to find vacant parking spots, reducing the time and effort spent on parking, and thus contributing to a more efficient urban environment. It involves the use of sensors and data communication to relay information to users, enabling them to make informed decisions regarding their parking needs

Some major daily parking issues include:

- Lack of parking spaces, mainly in the urban area
- Misuse of available parking spaces
- More time and fuel/gas are used to find open parking spaces
- Difficulty in finding vehicles at large parking lots
- Traffic congestion is concentrated on underutilized parking spaces
- Business parking lots are taken over by passenger parking
- Incorrect parking
- Proper management of disabled areas & unused private parking lots
- The natural impact of excessive fuel consumption in search of a parking space
- Unclear parking policies.

Concept of Smart Parking

- Parking demand management and space optimization
- Personalized parking guidance
- parking reservation system
- dynamic parking price and policy optimization
- parking area, charging, illegal parking detection

Types of IoT smart parking sensors:

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

- Electromagnetic Field Detection: The sensor can detect small changes in the magnetic field when a metal object is near it.

- Infrared: This type of sensor measures changes in ambient temperature and detects movement

Arduino:

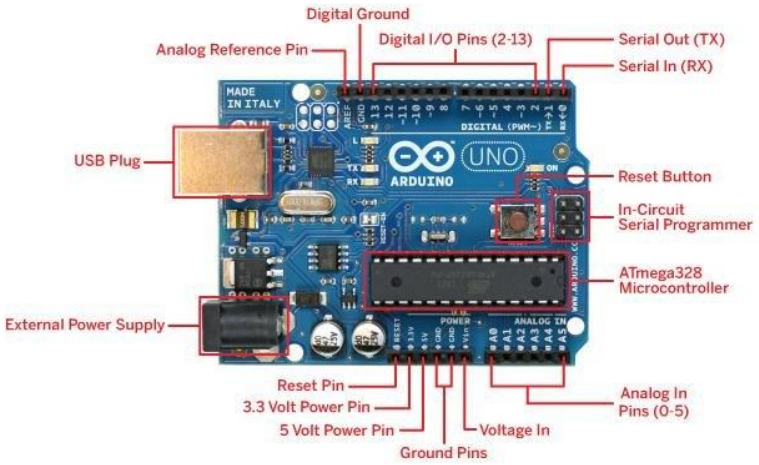
Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

What Does it Do?

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.

Arduino UNO R3:

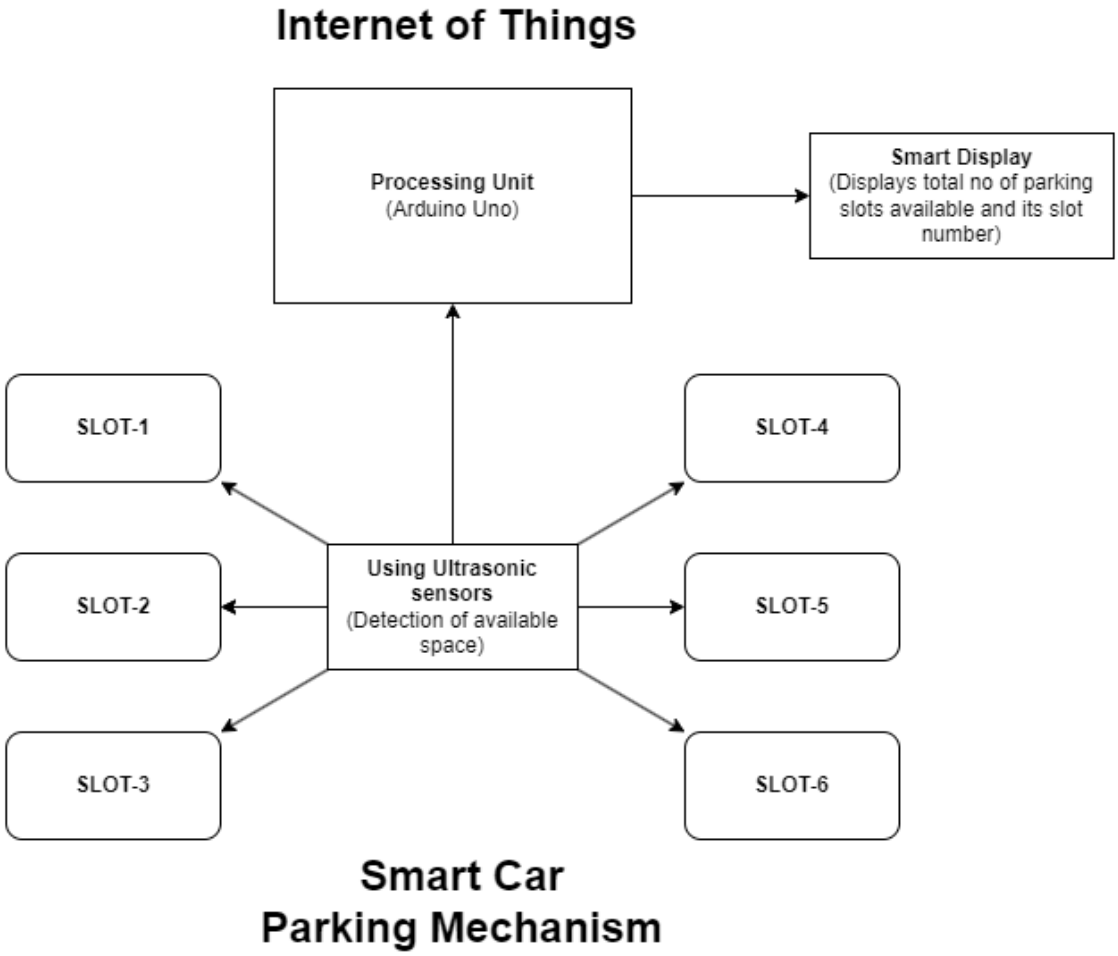


Arduino In Smart Parking System:

Arduino serves as the central control unit in the Smart Parking System, orchestrating all components and functions of the project. It interfaces with the ultrasonic sensor to process data and determine parking space occupancy, utilizing a predefined threshold for decision-making. Arduino controls the LCD display, delivering real-time information to users by updating the screen with "Parking Empty" or "Parking Occupied" based on the sensor data. Its role in real-time monitoring, data processing, and user interaction makes it the indispensable brain of the system, ensuring efficiency and user-friendliness.

Arduino's open-source nature and a rich ecosystem of libraries simplify prototyping and development, making it an ideal choice for IoT applications. Its capabilities enable the project to address urban parking challenges by reducing the effort needed to find available parking spots, contributing to a more efficient and convenient urban environment.

The architecture of Smart Parking System:



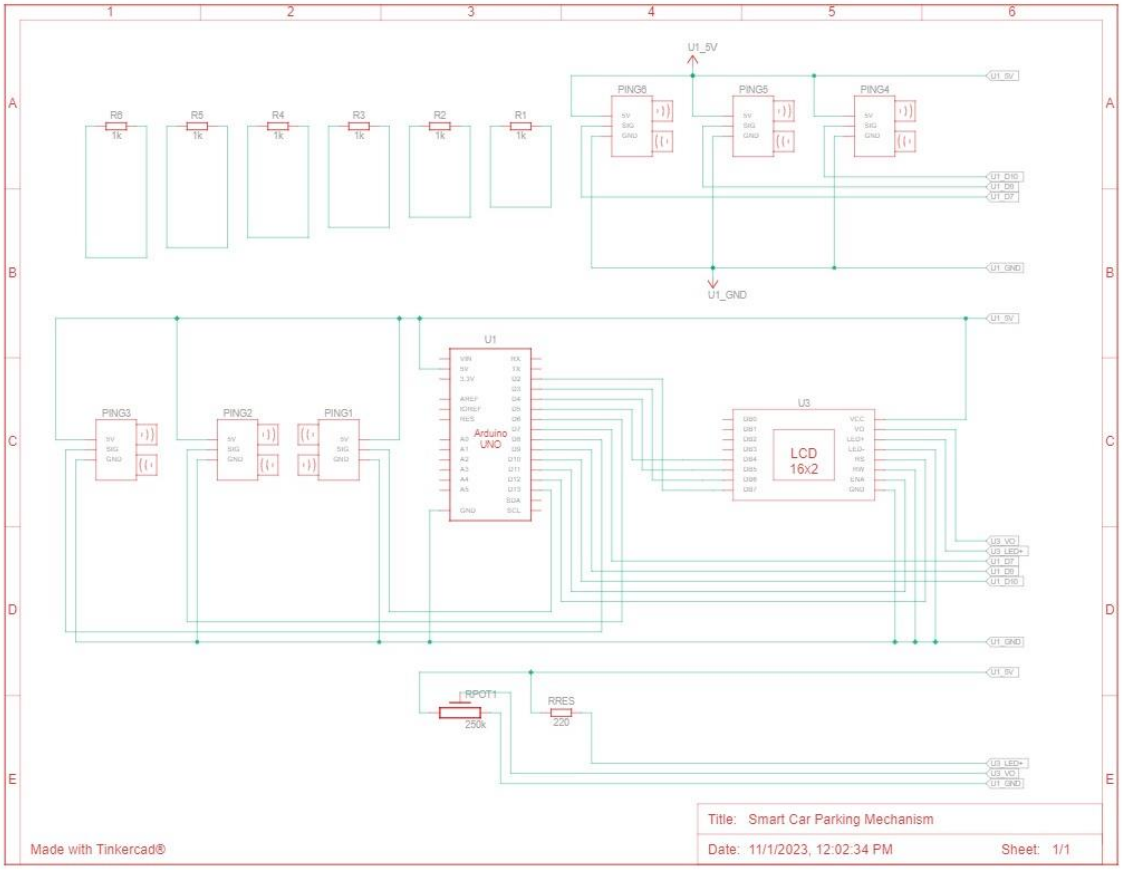
Project Description:

This project involves creating a Smart Parking System using Arduino, an Ultrasonic sensor, and an LCD display. The system will provide real-time information about parking space availability to users, making parking in crowded areas more convenient.

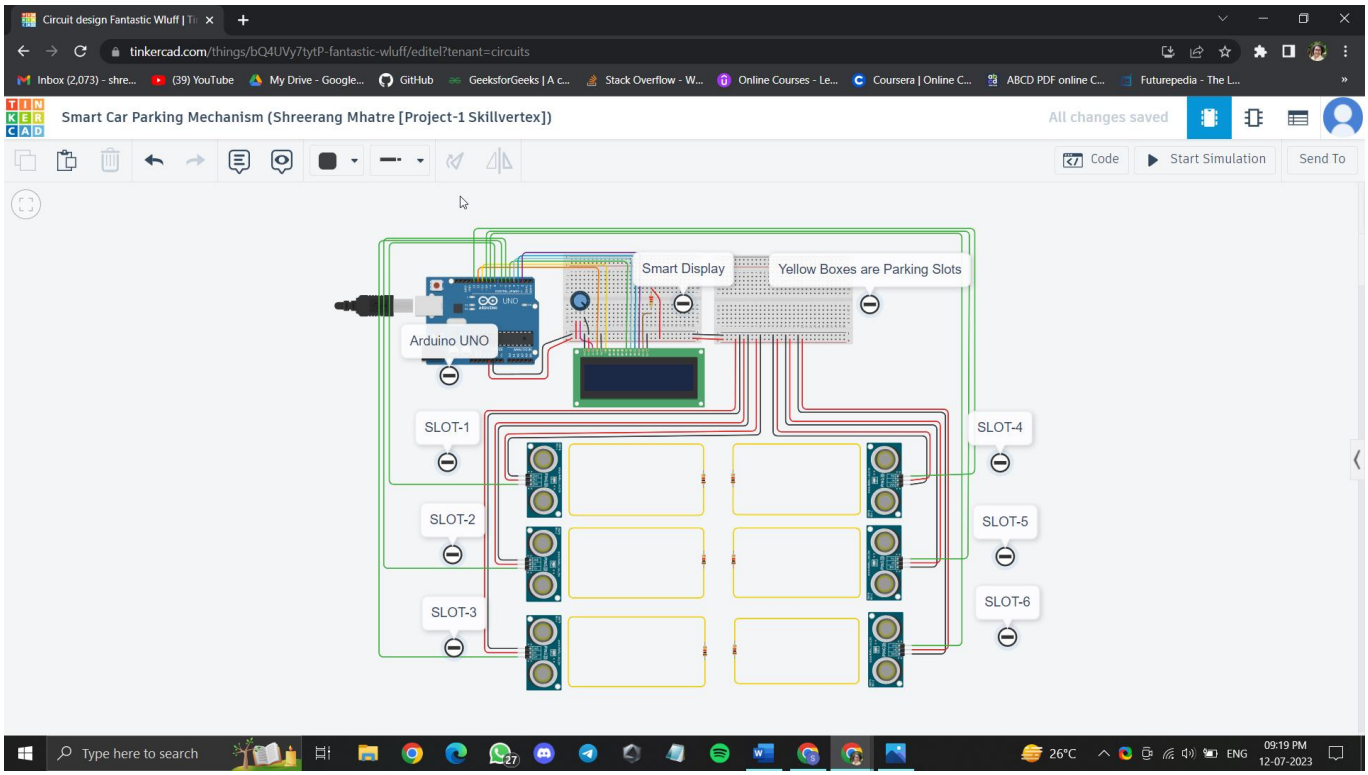
Component List:

Description	Label Name	Quantity
Arduino Uno R3	U1	1
Ultrasonic Distance Sensor	PING1 PING2 PING3 PING4 PING5 PING6	6
1 kΩ Resistor	R1 R2 R3 R4 R5 R6	6
220 kΩ Resistor	RRes	1
250 kΩ Potentiometer	Rpot1	1
LCD 16x2	U3	1

Circuit Diagram:

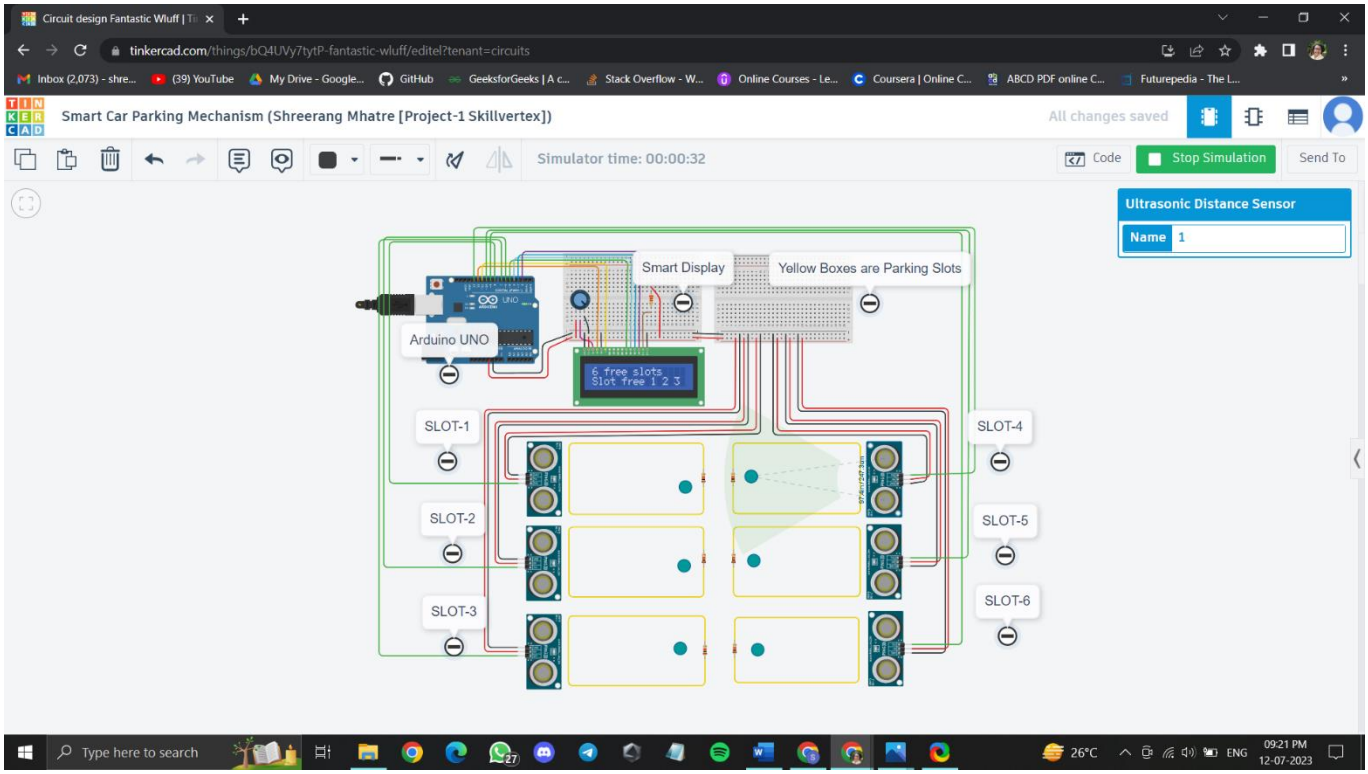


Solution: complete circuit

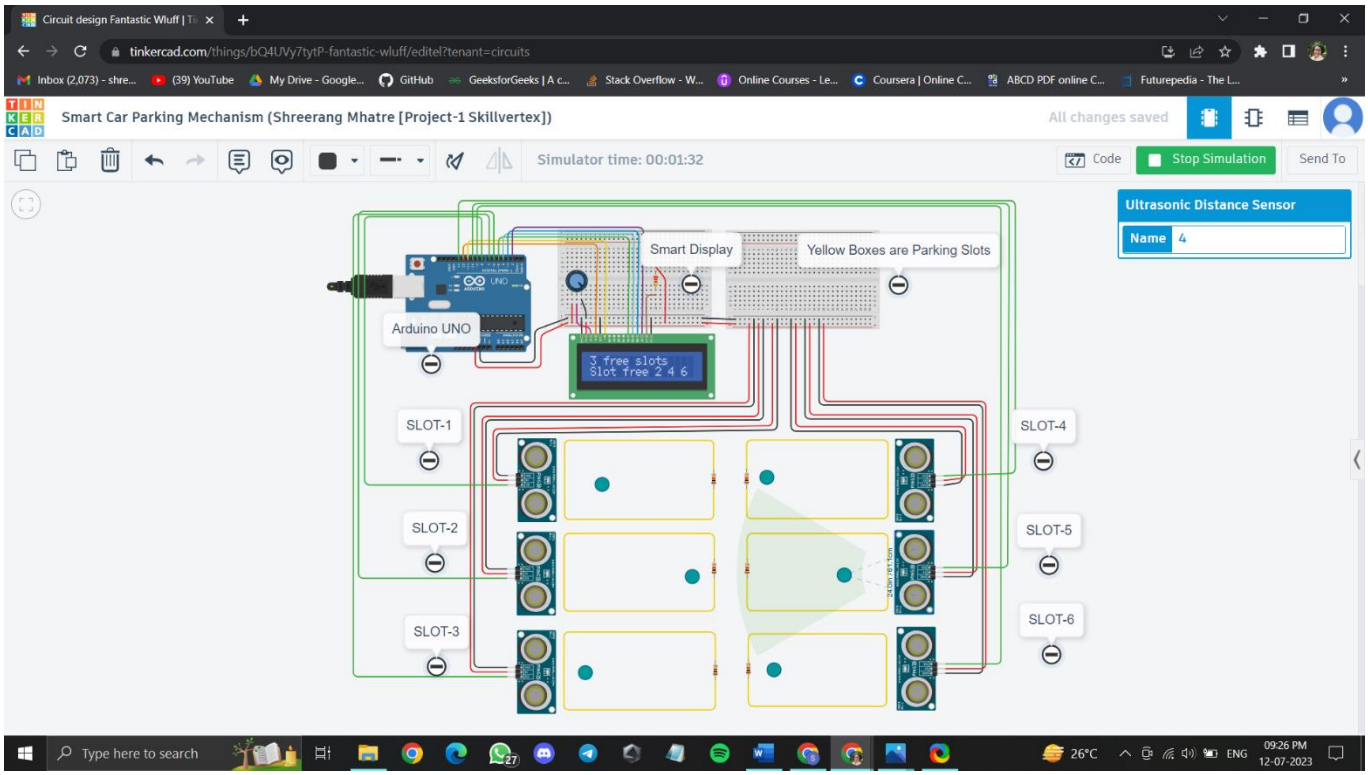


Working:

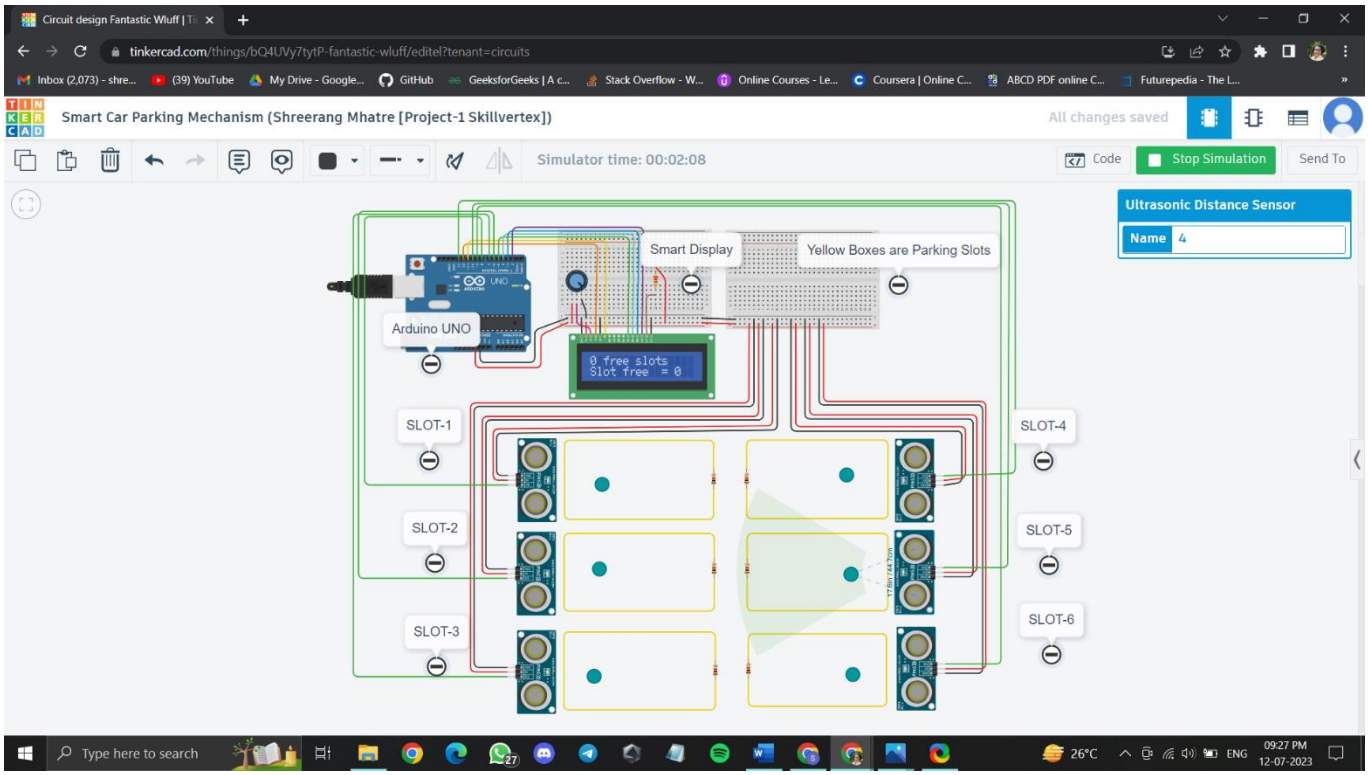
Case1: When all parking slots are free the Smart Display indicates “6 free slots” with the top 3 unoccupied slot name



Case2: When some parking slots are occupied it displays the free space available and its slot number



Case3: When no parking space is available it indicates “0 free slots”



Arduino Code:

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

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in
microseconds
  return pulseIn(echoPin, HIGH);
}
// initialize the library with the numbers of the interface pins


void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);

}
int cm1 = 0;
int cm2 = 0;
int cm3 = 0;
int cm4 = 0;
int cm5 = 0;
int cm6 = 0;
void loop() {
  int count = 0;
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(10, 1); //10th box in lcd display
  cm1 = 0.01723 * readUltrasonicDistance(7, 7);
  cm2 = 0.01723 * readUltrasonicDistance(6, 6);
  cm3 = 0.01723 * readUltrasonicDistance(8, 8);
  cm4 = 0.01723 * readUltrasonicDistance(9, 9);
  cm5 = 0.01723 * readUltrasonicDistance(10, 10);
  cm6 = 0.01723 * readUltrasonicDistance(13, 13);
  if (cm1 > 150)
  {
    count++;
  }
  if(cm2 > 150)
  {
    count++;
  }
  if(cm3 > 150)
  {
    count++;
  }
    if (cm4 > 150)
  {
    count++;
```

```
  }
  if(cm5 > 150)
  {
    count++;
  }
  if(cm6 > 150)
  {
    count++;
  }
  lcd.setCursor(0,0);
  lcd.print(count);
  lcd.print(" free ");
  lcd.print("slots");
  lcd.setCursor(0,1);
  lcd.print("Slot ");
  lcd.print("free ");
  if (cm3 > 150)
    lcd.print("1 ");
  if (cm2 > 150)
    lcd.print("2 ");
  if (cm1 > 150)
    lcd.print("3 ");
  if (cm6 > 150)
    lcd.print("4 ");
  if (cm5 > 150)
    lcd.print("5 ");
  if (cm4 > 150)
    lcd.print("6 ");
  if (cm3<150 & cm2<150 & cm1<150 & cm4<150 & cm5<150 & cm6<150)
    lcd.print(" = 0");

  delay(500);
  lcd.clear();

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

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in
microseconds
  return pulseIn(echoPin, HIGH);
}
// initialize the library with the numbers of the interface pins


void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);
```

```
}
int cm1 = 0;
int cm2 = 0;
int cm3 = 0;
int cm4 = 0;
int cm5 = 0;
int cm6 = 0;
void loop() {
  int count = 0;
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(10, 1); //10th box in lcd display
  cm1 = 0.01723 * readUltrasonicDistance(7, 7);
  cm2 = 0.01723 * readUltrasonicDistance(6, 6);
  cm3 = 0.01723 * readUltrasonicDistance(8, 8);
  cm4 = 0.01723 * readUltrasonicDistance(9, 9);
  cm5 = 0.01723 * readUltrasonicDistance(10, 10);
  cm6 = 0.01723 * readUltrasonicDistance(13, 13);
  if (cm1 > 150)
  {
    count++;
  }
  if(cm2 > 150)
  {
    count++;
  }
  if(cm3 > 150)
  {
    count++;
  }
    if (cm4 > 150)
  {
    count++;
  }
  if(cm5 > 150)
  {
    count++;
  }
  if(cm6 > 150)
  {
    count++;
  }
  lcd.setCursor(0,0);
  lcd.print(count);
  lcd.print(" free ");
  lcd.print("slots");
  lcd.setCursor(0,1);
  lcd.print("Slot ");
  lcd.print("free ");
  if (cm3 > 150)
    lcd.print("1 ");
  if (cm2 > 150)
    lcd.print("2 ");
  if (cm1 > 150)
    lcd.print("3 ");
  if (cm6 > 150)
    lcd.print("4 ");
  if (cm5 > 150)
    lcd.print("5 ");
  if (cm4 > 150)
```

```
    lcd.print("6 ");
  if (cm3<150 & cm2<150 & cm1<150 & cm4<150 & cm5<150 & cm6<150)
    lcd.print(" = 0");

  delay(500);
  lcd.clear();

}
```

Working Principle:

The Smart Parking System utilizes an Ultrasonic Sensor and an Arduino to determine parking space status. Here's a brief working explanation:

1. **Distance Measurement:** The Ultrasonic Sensor sends out ultrasonic waves, which bounce back when they encounter an obstacle, such as a parked car. The sensor measures the time it takes for the waves to return.
2. **Threshold Comparison:** The Arduino compares the measured distance to a predefined threshold value, typically set at 5 cm. If the measured distance is greater than this threshold, it indicates that the parking space is empty.
3. **Display Control:** Depending on the distance comparison, the Arduino sends commands to the LCD display. If the distance is greater than the threshold, the display shows "Parking Empty." If the distance is less than the threshold, it displays "Parking Occupied."
4. **Real-time Monitoring:** The Arduino continuously repeats this process, monitoring the sensor's data and updating the LCD display accordingly. This real-time monitoring ensures that users receive up-to-date information about parking space availability.

Applications:

1. **Urban Parking Management:**
   In densely populated urban areas, managing parking spaces efficiently is crucial to reducing traffic congestion. The Smart Parking System can be deployed in public parking lots or on-street parking spaces, providing real-time information to drivers about available spots. This not only reduces the time and frustration associated with searching for parking but also contributes to less congested roads and improved urban mobility.

2. **Commercial Parking Facilities:**
   Shopping centers, business complexes, and commercial parking lots can benefit from the system by helping customers quickly locate vacant parking spots. This feature enhances the overall experience for visitors and shoppers, increasing foot traffic and, ultimately, boosting business revenues.

3. **Airport Parking:**
   Airports often witness a constant influx of vehicles, especially in busy travel seasons. The Smart Parking System can guide travelers to available parking spaces, reducing the time and effort required to find a spot. This, in turn, enhances the overall airport experience and reduces stress for travelers.

4. **Hospital Parking:**
   Hospitals deal with a steady stream of patients and visitors. The system can be deployed

in hospital parking areas to help patients and visitors find available parking spaces swiftly, which is particularly critical during emergencies when every minute counts.

5. **Smart Cities:**
   As part of a broader smart city initiative, the Smart Parking System contributes to more efficient urban planning and traffic management. It enables authorities to monitor and optimize parking availability, reducing traffic congestion, pollution, and the environmental impact of urban living.

6. **Event Venues:**
   Large event venues, such as stadiums and concert halls, often draw substantial crowds. The system can be used to optimize parking availability for event attendees, ensuring a smoother and more enjoyable experience. It also helps venue operators manage traffic flow efficiently.

7. **Private Parking Facilities:**
   Individual property owners, like hotels or apartment complexes, can use the Smart Parking System to enhance the parking experience for residents, guests, or customers. By guiding them to available parking spaces, it ensures convenience and makes a positive impression.

Conclusion:

In conclusion, the Smart Parking System created using Arduino and an Ultrasonic sensor provides a practical solution for monitoring parking space availability in real-time. This system has the potential to enhance urban parking management by reducing the time and effort required to find a vacant parking spot, ultimately making urban living more efficient and user-friendly. The project demonstrates the power of IoT technology in addressing everyday challenges and improving the quality of life.

Tinkercad link:

https://www.tinkercad.com/things/bQ4UVy7tytP?sharecode=Evw2C6K6T48zDsKNucTFoY7Ic_wJ1SfwEKVp3haT_p0

References:

Arduino Documentation:
https://docs.arduino.cc/hardware/uno-rev3

Research Paper:
https://www.researchgate.net/publication/341870728_Smart_Parking_System_based_on_IOT
https://www.researchgate.net/publication/335807455_Smart_Parking_Management_System

Articles:
https://tomorrow.city/a/smart-parking
https://www.smartparking.com/uk
https://www.airtel.in/blog/business/iot-based-smart-parking-system-a-step-towards-building-smart-city/