

SMART PARKING USING IOT

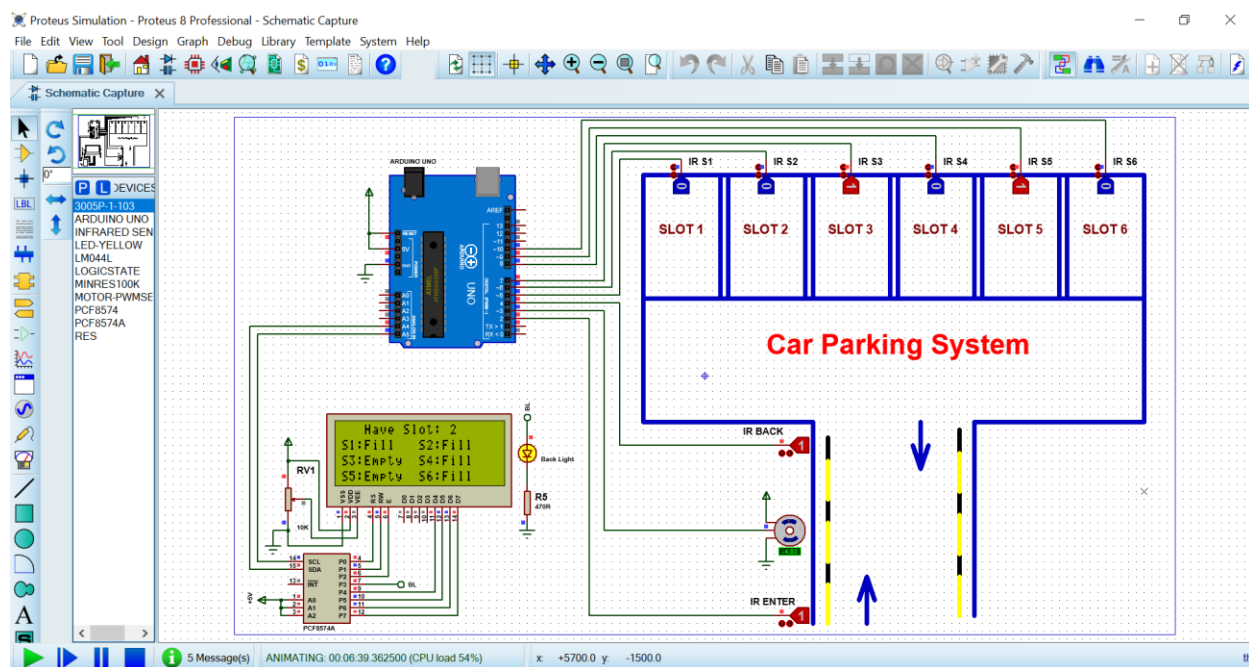
TEAM MEMBER

Phase -3 Document submission

J.ARUNKUMAR

Project:SMART PARKING

510421106003



Smart parking in ultrasonicsensor:

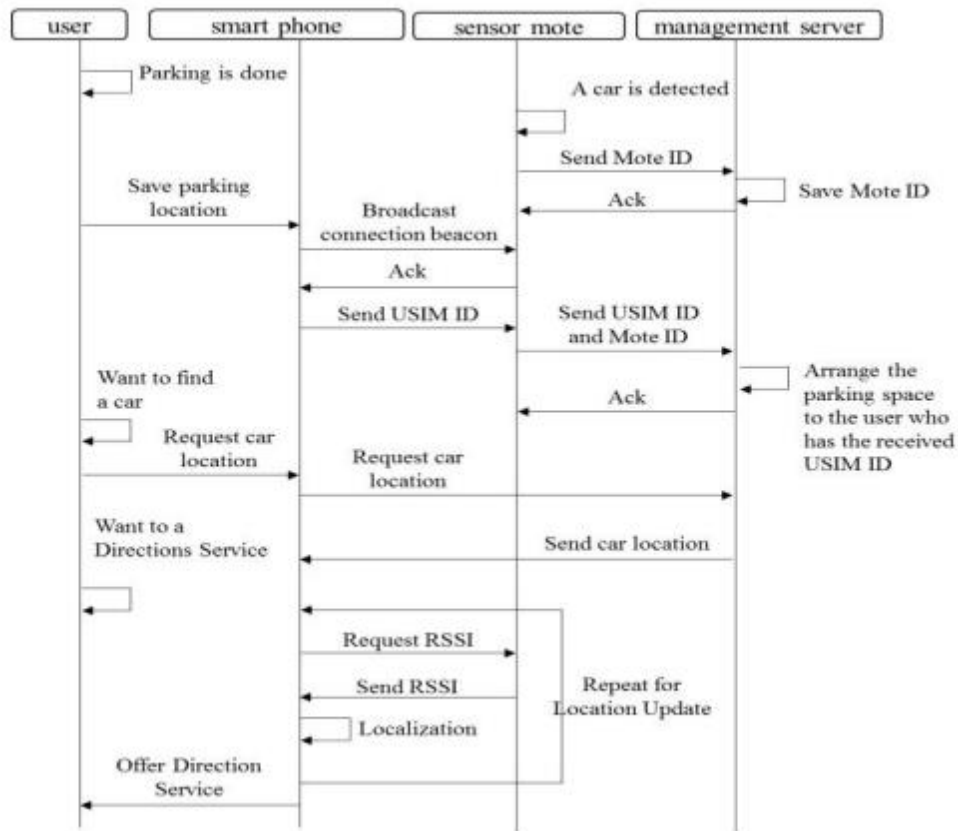
Limited parking spaces in certain areas is a common problem for all drivers. A smart parking system helps drivers efficiently find a parking space without the inconvenience of circling blocks, wasting fuel, and wasting valuable time. Our vehicle detection sensors work well to develop this IoT application

Our Goal in Parking System OEM Applications:

- Provide a competitively priced product to be integrated into OEM parking systems.
- Simplify Operations
- Maximize Profits

Operation of the Smart Parking System

After the user finishes parking their vehicle, the sensor mote above the parking space transmits the mote ID to the management server. The management server updates the parking state for the corresponding space. Next, the user executes the parking application on their smartphone to save the location of their parked vehicle. The parking application employs the RSSI of the BLE module to recognize the location of the parked vehicle. The USIM ID of the user's smartphone is then sent to the sensor mote before the sensor mote transmits the received USIM ID and the mote ID to the management server. The management server saves all the information. When users request parking location information regarding their vehicles, the parking application submits the USIM ID to the management server. In addition, if the user requires the location guidance service to find their parking space, the parking application periodically receives RSSI data from the sensor motes deployed around the user. The RSSI data are then transformed into the distances between the user and the sensor motes. Based on three selected distances, the triangulation method is applied periodically and the current location of the user is recognized so the location guidance service can be provided in real-time.



Program:

```
#define ECHO_PIN1 15 //Pins for Sensor 1
```

```
#define TRIG_PIN1 2 //Pins for Sensor 1
```

```
#define ECHO_PIN2 5 //Pins for Sensor 2
```

```
#define TRIG_PIN2 18 //Pins for Sensor 2
```

```
#define ECHO_PIN3 26 //Pins for Sensor 3
```

```
#define TRIG_PIN3 27 //Pins for Sensor 3
```

```
int LEDPIN1 = 13;
```

```
int LEDPIN2 = 12;

int LEDPIN3 = 14;

void setup() {

    Serial.begin(115200);

    pinMode(LEDPIN1, OUTPUT);

    pinMode(TRIG_PIN1, OUTPUT);

    pinMode(ECHO_PIN1, INPUT);

    pinMode(LEDPIN2, OUTPUT);

    pinMode(TRIG_PIN2, OUTPUT);

    pinMode(ECHO_PIN2, INPUT);

    pinMode(LEDPIN3, OUTPUT);

    pinMode(TRIG_PIN3, OUTPUT);

    pinMode(ECHO_PIN3, INPUT);

}

float readDistance1CM() {

    digitalWrite(TRIG_PIN1, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN1, HIGH);
```

```
    delayMicroseconds(10);

    digitalWrite(TRIG_PIN1, LOW);

    int duration = pulseIn(ECHO_PIN1, HIGH);

    return duration * 0.034 / 2 ;
}

float readDistance2CM() {

    digitalWrite(TRIG_PIN2, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN2, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG_PIN2, LOW);

    int duration = pulseIn(ECHO_PIN2, HIGH);

    return duration * 0.034 / 2;
}

float readDistance3CM() {

    digitalWrite(TRIG_PIN3, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN3, HIGH);
```

```
    delayMicroseconds(10);

    digitalWrite(TRIG_PIN3, LOW);

    int duration = pulseIn(ECHO_PIN3, HIGH);

    return duration * 0.034 / 2;
}

void loop() {

    float distance1 = readDistance1CM();

    float distance2 = readDistance2CM();

    float distance3 = readDistance3CM();

    bool isNearby1 = distance1 > 200;

    digitalWrite(LEDPIN1, isNearby1);

    bool isNearby2 = distance2 > 200;

    digitalWrite(LEDPIN2, isNearby2);

    bool isNearby3 = distance3 > 200;

    digitalWrite(LEDPIN3, isNearby3);

    Serial.print("Measured distance: ");

    Serial.println(readDistance1CM());

    Serial.println(readDistance2CM());
```

```
Serial.println(readDistance3CM());  
  
delay(100);  
  
}
```

Conclusion:

In this study, we implemented a smart parking system based on sensor motes with ultrasonic sensors and BLE communication devices in the IoT. The states of the parking spaces are checked in real-time by the smart parking system. In addition, based on user location awareness technology, the locations of parked vehicles are recognized and a location guidance service is provided to help users find their parking space. The ultrasonic sensor in the sensor mote is employed to determine the states of parking spaces. The ultrasonic sensor calculates the distance between the sensor mote and the floor of the parking space using a backward echo ultrasonic wave. If the distance between the sensor and the floor is below 100 cm, then the space is regarded as being occupied by a vehicle. We also implemented two location awareness methods based on the RSSI of the BLE modules in sensor motes. In order to obtain reliable RSSI data, we measured the fluctuations in the RSSI range according to the signal strength of the BLE module. We found that reliable RSSI data could be received when the signal strength was weak and the fluctuations in the RSSI were small. Therefore, we decided to use a signal strength of 6 dBm in our experiment.