**Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham**

**Chennai Campus**

**Department of ECE**

**Internet of Things Lab – Capstone Report**

|  |  |
| --- | --- |
| **Name** | **BIYAPPU VISHNU VARDHAN REDDY** |
| **Registration Number** | **CH.EN.U4ECE20004** |
| **Name of the project** | **SMART PARKING SYSTEM** |
| **Date of the Experiment - done** | **10TH DECEMBER 2021** |
| **Date of report submission** | **16TH DECEMBER 2021** |

|  |  |
| --- | --- |
| **Abstract** | This concept is based on the smart working process of the gadgets which is the future to us. We know that smart parking system is auto detect system which uses to detect the allotted free slots to park the vehicles.  The recent growth in economy and due to the availability of price down cars in the market, an every average middle-class individual can afford a car, which is good thing, however the consequences of heavy traffic jams, pollution, less availability of roads and spot to drive the motor car. One of the important concerns, which is to be taken in accounting, is that problem of parking those vehicles.  If there is space for parking the vehicle but so much time is squandered in finding that exact parking slot resulting in more fuel intake and not also environment friendly. It will be great deal if in some way we find out that the parking itself can provide the precise vacant position of parking slot then it'll be helpful not limited to the drivers also for the environment. We can easily search for a empty slots to park our vehicle which is securely locked with password and manual protocols.  IoT based Smart Parking Systems for Smart Cities | HIOTRON |
| **Objective** | Many of places have the information in mobile or system where the places is empty slot are no whether there is free slot or full to choose next one .it makes work easy to park the vehicles by operating automatically.  This system can give user the information about parking space, but it won’t be able to give which parking slot is vacant and occupied. Hence, such system cannot smartly handle the issue. Car lifts along with automated robotic system, which automatically takes car to a particular parking spot as soon as the car enters on a platform |
| **Novelty of the Experiment** | The main motivation of this project is to reduce the traffic jam that occurs in the many areas which are caused by vehicles searching for parking. In the newspapers, we saw many articles regarding the parking problem. we are still using the manual vehicle parking system that why we are facing problems like wastage of time and energy finding free space across the parking surface when we need to park our car which requires a good amount of fuel. We proposed an automated system where the parking ground will only open if it has free slots for parking. The user can also check it before arriving there by a website. It will save the time as well as reduce the gathering in front of parking area.  These type of system’s easy on counting how many cars have entered the parking area and calculating the difference between this figure and the maximum number of parking spaces to estimate the number of spaces available. These types of system usually need a person in charge of the location in case something goes wrong due to variations in the number of parking spaces. |
| **Working principle of smart parking system** | * Here the ardunio controls whole part of the system which helps the sensors to acknowledge the slot verification whether the slots is free or not. * It controls and watches over all the components. The ultrasonic sensors will be placed in the parking slots that will encounter the presence of the cars inside the parking slots. One sensor will be placed beside the main entrance of the parking lot. * The sensors get the presence of a car in front of the entrance, it will send signal to the Arduino chip to check if there is an empty slot inside the parking lot. * The smart car parking system works on the simple principle of detecting obstacle and sending a visual feedback. * The resulting potential difference helps complete the circuit. The LEDs are placed along the driveway and switch on based on the input received by the sensor. * When the parking space is occupied, the IR rays emitted by the emitter is bounced back as the vehicle height is within the threshold distance and the rays strike the receiver and these waves are converted into an electrical signal creating a potential difference. The feedback of this result is indicated by the Red LED turning on and thus specifying the driver that the particular parking space is filled. |
| **FLOW CHART OF SMART PARKING SYSTEM** |  |
| **Tinker-cad Procedure**  **(Methodology)** | **WIRING FOR PARKING LOTS:**   * Firstly, a bread board and Arduino Uno-1 is taken, then power & ground connections are given in between them. * And an LCD display is placed then power and GND pins of LCD are connected to breadboard +, - pins. * Register select and enable pins of LCD is connected with input pins of Arduino Uno. * DB-4 to DB-7 pins of LCD is connected to input pins of Arduino. * Then LED anode pin of LCD is connected with power and LCD cathode pin is connected to ground of breadboard. * Read write is pin connected to ground. * Then a potentiometer is taken terminal 1 is connected to power and terminal 2 is connected to ground. And wiper pin of potentiometer is connected to Contrast pin of LCD display.   We should follow the same process in wiring for other 2 LCD screens. The only change is that if we don’t have breadboard directly, we connect to the respective pins.   * Then we took 5 ULTRASONIC DISTANCE SENSORS. And we gave wiring as the pins of each ultrasonic distance sensor power pin and ground pin is connected to +ve rail and -ve rail. And signal pin of these sensor is connected to input pins of Arduino Uno.   **WIRING FOR DISPLAYING THE CHARGE PER LOT:**  The same process should be followed in between Arduino uno and an LCD display as we followed for above process. But for this case there is no need of breadboard.  **WIRING IN BETWEEN KEYPAD AND LCD-3 FOR SETTING SECURITY PIN GENERATING SYSTEM:**  The same process wiring process is done in between Arduino Uno and LCD but in this case DB-4 to DB-7 pins of LCD-3 is connected to analog pins of Arduino uno-3. Now the input pins (D0 – D7) of Arduino is connected to row-1 to row-4 and column-1 to column-4 of Key pad pins.  Then servo motor which acts as a gate, is placed and power pin of this servo motor is connected to 5v pin of Arduino and ground pin of servo motor is connected to GND pin of Arduino Uno-3. At last the signal pin of Servo motor is connected to input pin (D9) of Arduino uno-3. |
| **Arduino Program** | **ARDUINO UNO – 1 CODE:**  #include <LiquidCrystal.h>  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int green=13;  int red=2;  #define t1 10  #define t2 9  #define t3 8  #define t4 7  #define t5 6  int distanceThreshold = 70;  void setup() {  lcd.begin(16,2);  lcd.setCursor(0,0);  Serial.begin (9600);  pinMode(green,OUTPUT);  pinMode(red,OUTPUT);  }  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 (d1>70 & d2>70 & d3>70 & d4>70 & d5>70)  {  lcd.setCursor(0,0);  lcd.print("5 Lots Free");  lcd.setCursor(0,1);  lcd.print("Lots 1,2,3,4,5 Free");  delay(200);  digitalWrite(red,LOW);  digitalWrite(green,HIGH);  }  else if((d1>70 & d2>70 & d3>70 & d4>70)|(d1>70 & d2>70 & d3>70 & d5>70)|(d1>70 & d2>70 & d4>70 & d5>70)|(d1>70 & d3>70 & d4>70 & d5>70)|(d2>70 & d3>70 & d4>70 & d5>70))  {  lcd.setCursor(0,0);  lcd.print("4 Lots are Free");  lcd.setCursor(0,1);  if(d1>70 & d2>70 & d3>70 & d4>70)  lcd.print("Lot 1,2,3,4 Free");  if(d1>70 & d2>70 & d3>70 & d5>70)  lcd.print("Lot 1,2,3,5 Free");  if(d1>70 & d2>70 & d4>70 & d5>70)  lcd.print("Lot 1,2,4,5 Free");  if(d1>70 & d3>70 & d4>70 & d5>70)  lcd.print("Lot 1,3,4,5 Free");  if(d2>70 & d3>70 & d4>70 & d5>70)  lcd.print("Lot 2,3,4,5 Free");  delay(300);  digitalWrite(red,LOW);  digitalWrite(green,HIGH);  }  else if((d1>70 & d2>70 & d3>70)|(d2>70 & d3>70 & d4>70)|(d3>70 & d4>70 & d5>70)|(d4>70 & d5>70 & d1>70))  {  lcd.setCursor(0,0);  lcd.print("3 Lots is Free");  lcd.setCursor(0,1);  if(d3>70 & d4>70 & d5>70)  lcd.print("Lot 3,4,5 Free");  if((d4>70 & d5>70 & d1>70))  lcd.print("Lot 1,4,5 Free");  if(d1>70 & d2>70 & d3>70)  lcd.print("Lot 1,2,3 Free");  if(d2>70 & d3>70 & d4>70)  lcd.print("Lot 2,3,4 Free");  delay(300);  digitalWrite(red,LOW);  digitalWrite(green,HIGH);  }  else if((d1>70 & d2>70)|(d2>70 & d3>70)|(d3>70 & d4>70)|(d4>70 & d5>70)|(d5>70 & d1>70)|(d1>70 & d3>70)|(d1>70 & d4>70)|(d1>70 & d5>70)|(d2>70 & d4>70)|(d2>70 & d5>70)|(d3>70 & d5>70))  {  lcd.setCursor(0,0);  lcd.print("2 Lots are Free");  lcd.setCursor(0,1);  if((d1>70 & d2>70))  lcd.print("Lot 1,2 Free");  if((d2>70 & d3>70))  lcd.print("Lot 2,3 Free");  if((d3>70 & d4>70))  lcd.print("Lot 3,4 Free");  if((d4>70 & d5>70))  lcd.print("Lot 4,5 Free");  if((d5>70 & d1>70))  lcd.print("Lot 1,5 Free");  if((d1>70 & d3>70))  lcd.print("Lot 1,3 Free");  if((d4>70 & d1>70))  lcd.print("Lot 1,4 Free");  if((d2>70 & d4>70))  lcd.print("Lot 2,4 Free");  if((d2>70 & d5>70))  lcd.print("Lot 2,5 Free");  if((d3>70 & d5>70))  lcd.print("Lot 3,5 Free");  delay(400);  digitalWrite(red,LOW);  digitalWrite(green,HIGH);  }    else if((d1>70)|(d2>70)|(d3>70)|(d4>70)|(d5>70))  {  lcd.setCursor(0,0);  lcd.print("1 Lot is Free");  lcd.setCursor(0,1);  if(d1>70)  lcd.print("Lot 1 Free");  if(d2>70)  lcd.print("Lot 2 Free");  if(d3>70)  lcd.print("Lot 3 Free");  if(d4>70)  lcd.print("Lot 4 Free");  if(d5>70)  lcd.print("Lot 5 Free");  delay(500);  digitalWrite(red,LOW);  digitalWrite(green,HIGH);  }  else if((d1<70) & (d2<70) & (d3<70) & (d4<70) & (d5<70))  {  lcd.setCursor(0,0);  lcd.print("No Slot Free");  lcd.setCursor(0,1);  lcd.print("Parking Full");  delay(500);  digitalWrite(green,LOW);  digitalWrite(red,HIGH);    }  delay(100);  }  **ARDUINO UNO – 2 CODE:**  #include <LiquidCrystal.h>  LiquidCrystal lcd(13,12,11,10,8,7);  void setup()  {  lcd.begin(16,2);  lcd.clear();  lcd.setCursor(0,0);  lcd.print("CHARGE PER LOT");  lcd.setCursor(0,1);  lcd.print("Rs: 150/- ONLY");  lcd.write(byte(0));  }  void loop()  {  delay(1000);  lcd.display();  lcd.setCursor(10,0);  }  **ARDUINO UNO – 3 CODE:**  #include <Keypad.h>  #include <LiquidCrystal.h>  #include <Servo.h>  #define Password\_Length 5  Servo myservo;  LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);  int pos = 0;  char Data[Password\_Length];  char Master[Password\_Length] = "1234";  byte data\_count = 0, master\_count = 0;  bool Pass\_is\_good;  bool door = false;  char customKey;  /\*---preparing keypad---\*/  const byte ROWS = 4;  const byte COLS = 4;  char keys[ROWS][COLS] = {  {'1', '2', '3', 'A'},  {'4', '5', '6', 'B'},  {'7', '8', '9', 'C'},  {'\*', '0', '#', 'D'}  };  byte rowPins[ROWS] = {0, 1, 2, 3};  byte colPins[COLS] = {4, 5, 6, 7};  Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);  /\*--- Main Action ---\*/  void setup()  {  myservo.attach(9, 2000, 2400);  ServoClose();  lcd.begin(16, 2);  lcd.print("Lot Entry Gate");  loading("Loading");  lcd.clear();  }  void loop()  {  if (door == true)  {  customKey = customKeypad.getKey();  if (customKey == '#')  {  lcd.clear();  ServoClose();  lcd.print("Gate is closed");  delay(300);  door = false;  }  }  else  Open();  }  void loading (char msg[]) {  lcd.setCursor(0, 1);  lcd.print(msg);  for (int i = 0; i < 9; i++) {  delay(100);  lcd.print(".");  }  }  void clearData()  {  while (data\_count != 0)  {  Data[data\_count--] = 0;  }  return;  }  void ServoClose()  {  for (pos = 90; pos >= 0; pos -= 10) {  myservo.write(pos);  }  }  void ServoOpen()  {  for (pos = 0; pos <= 90; pos += 10) {  myservo.write(pos);  }  }  void Open()  {  lcd.setCursor(0, 0);  lcd.print("Enter Password");  customKey = customKeypad.getKey();  if (customKey)  {  Data[data\_count] = customKey;  lcd.setCursor(data\_count, 1);  lcd.print(Data[data\_count]);  data\_count++;  }  if (data\_count == Password\_Length - 1)  {  if (!strcmp(Data, Master))  {  lcd.clear();  ServoOpen();  lcd.print(" Gate is Open ");  door = true;  delay(3000);  loading("Waiting");  lcd.clear();  lcd.print(" Payment Done! ");  delay(1000);  ServoClose();  door = false;  }  else  {  lcd.clear();  lcd.print(" Wrong Password ");  door = false;  }  delay(1000);  lcd.clear();  clearData();  }  } |
| **Results**  **(Figures, tables and results discussion)** | **CIRCUIT DIAGRAM:** |
| **Applications** | * **Optimized parking** – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities. * **Reduced traffic** – Traffic flow increases as fewer cars are required to drive around in search of an open parking space. * **User Experience** – A smart parking solution will integrate the entire user experience into a unified action. Driver’s payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process. * **System integration**: these systems can integrate in shopping malls, restaurants, theatres where people crowed more will be helpful. * **Usages**: This will be helpful to save money, time, clears traffic jams which is friendly nature of parking. |
| **Conclusion** | Here the data analysis of management platform to show the parking of places whether it is filled or not by using Arduino board and lcd displays. The intelligent parking system with guidance to a free parking space can improve the traffic situation in cities. Drivers find a free parking space faster, which reduces congestion and various other negative externalities.  So, this project is based on smart parking system which is completely based on Arduino board and simulated in Tinker-cad. |