Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessors and Microcontrollers Lab

BECE204P

SMART CAR PARKING SYSTEM

FACULTY : PADMINI T N

DONE BY : **1) SANJAY M S (22BML0106)**

2) HARIHARASUDHAN V (22BEC0955)

3) PUGALARUSU M (22BEC0962)

4) HARIHARAN N (22BEC0965)

SMART CAR PARKING SYSTEM

TABLE OF CONTENTS :

1.Aim/Objective of the Project

2.Introduction

3.Abstract

4.Methodology

- Block diagram (Hardware)
- Flow chart (Software)

5.Program with comments

6.Results

 Screen shot showing program with zero syntax error(KEIL)

 Screen shot of the result (KEIL)

 Screen shot of Proteus simulation

 Photo of the hardware demonstration .

7.Conclusion

1.Aim / Objective of the Project :

To design a smart car parking system using 8051 microcontroller ,ir sensors and servo motors that has a feature of showing number of empty slots in the lcd display.

2.Introduction :

Smart car parking system is a system that offers intelligent and efficient solution to the challenges associated with traditional parking systems.

Smart car parking systems have revolutionized the traditional approach to parking by integrating advanced technologies such as 8051 microcontroller ,infrared (IR) sensors, servo motors, and LCD displays.

These systems provide efficient and hassle-free parking solutions, ensuring optimal utilization of parking space and enhancing the overall parking experience for drivers.

By automating the detection and control processes, these systems optimize parking space utilization, enhance user experience, and contribute to a more organized and streamlined parking environment.

This real-time information facilitates quicker and more convenient parking, reducing congestion and improving overall parking management.

3.Abstract :

The smart car parking system using an 8051 microcontroller, IR sensors, servo motors, and an LCD display presents an intelligent and automated approach to parking management .The core components of this smart parking system include an 8051 microcontroller, IR sensors, servo motors, and an LCD display. The 8051 microcontroller acts as the central processing unit, coordinating the operation of various components. The IR sensors are responsible for detecting the presence or absence of vehicles in parking spots, while the servo motors control the movement of barriers or gates. The LCD display provides real-time information to drivers regarding parking availability and instructions for parking.

The system operates by continuously monitoring the parking spots using IR sensors. Initially the total number of slots will be displayed on the LCD display .When a vehicle enters the entrance spot, the IR sensor kept infront of entrance gate detects its presence and sends a signal to the 8051 microcontroller. The microcontroller processes this information and activates the servo motor, which operates the gate or barrier to allow the vehicle entry. After 5 seconds , the gate closes after the car enters through the entrance gate. And the number of slots available will be decremented by 1 and it will be displayed on the LCD display.

Similarly, when a vehicle leaves the parking spot, the IR sensor just before the exit gate detects its presence and relays the signal to the microcontroller. The microcontroller then triggers the servo motor to open the gate for the exiting vehicle. Similarly the gate closes after 5 seconds. And the count of total number of free slots will be incremented by 1 and it will be displayed on the LCD display. The LCD display connected to the system provides drivers with real-time updates on parking availability, guiding them to vacant spots and displaying relevant instructions.

The integration of an 8051 microcontroller in this smart car parking system offers several advantages. When all slots were full, the gate in the entrance won't open .So even if the driver has failed to see the available slots in the LCD display , the driver won't able to enter the parking area reducing unusual risks for the drivers .And when there is no car inside the parking area, the exit gate won't open.

The smart car parking system provides a reliable and efficient control unit, capable of processing sensor data and managing the operation of servo motors. The use of IR sensors ensures accurate detection of vehicles, preventing unauthorized parking and optimizing space utilization. The servo motors enable automated gate operations, eliminating the need for manual intervention and streamlining the parking process. The LCD display enhances user experience by providing drivers with crucial information, enabling them to find parking spots more quickly and conveniently.

ADVANTAGES :

1. Optimal Space Utilization: The integration of IR sensors in the smart car parking system enables accurate detection of vehicles, allowing for optimal utilization of parking space. This helps to reduce congestion and maximize the number of vehicles that can be accommodated within the parking area.

2. Enhanced Efficiency: By automating the process of vehicle detection and gate operations, the smart car parking system improves overall efficiency. Drivers can quickly locate available parking spots through real-time information displayed on the LCD display, reducing the time spent searching for a parking space.

3. Improved Traffic Flow: The automated gate operations facilitated by servo motors help to streamline traffic flow within the parking area. As the gates open and close automatically in response to vehicle presence or absence, the system minimizes delays and congestion, contributing to a smoother parking experience.

4. Enhanced Security: The combination of IR sensors and servo motors in the smart car parking system enhances security. Unauthorized entry or parking can be detected and prevented, ensuring that only authorized vehicles have access to the parking area. This helps to maintain a secure environment for both vehicles and users.

5. Real-time Information: The LCD display provides real-time updates on parking availability, guiding drivers to vacant spots. Additionally, it can display instructions or notifications, such as parking rules or emergency alerts, enhancing user convenience and awareness.

6. Cost-effective Solution: Implementing a smart car parking system using 8051 microcontroller, IR sensors, servo motors, and an LCD display offers a cost-effective solution compared to traditional parking management systems. The integration of these components provides efficient automation while being relatively affordable and easily maintainable.

7. User-Friendly Experience: The user-friendly interface of the LCD display simplifies the parking process for drivers. Clear instructions and real-time updates eliminate confusion and allow drivers to park their vehicles quickly and conveniently.

8. Data Collection and Analysis: The smart car parking system can collect data on parking patterns, such as the duration of parking and occupancy rates. This data can be analyzed to optimize parking management strategies, identify trends, and improve overall system efficiency.

9. Scalability and Flexibility: The smart car parking system can be easily scaled to accommodate larger parking areas or expanded as needed. The modular design

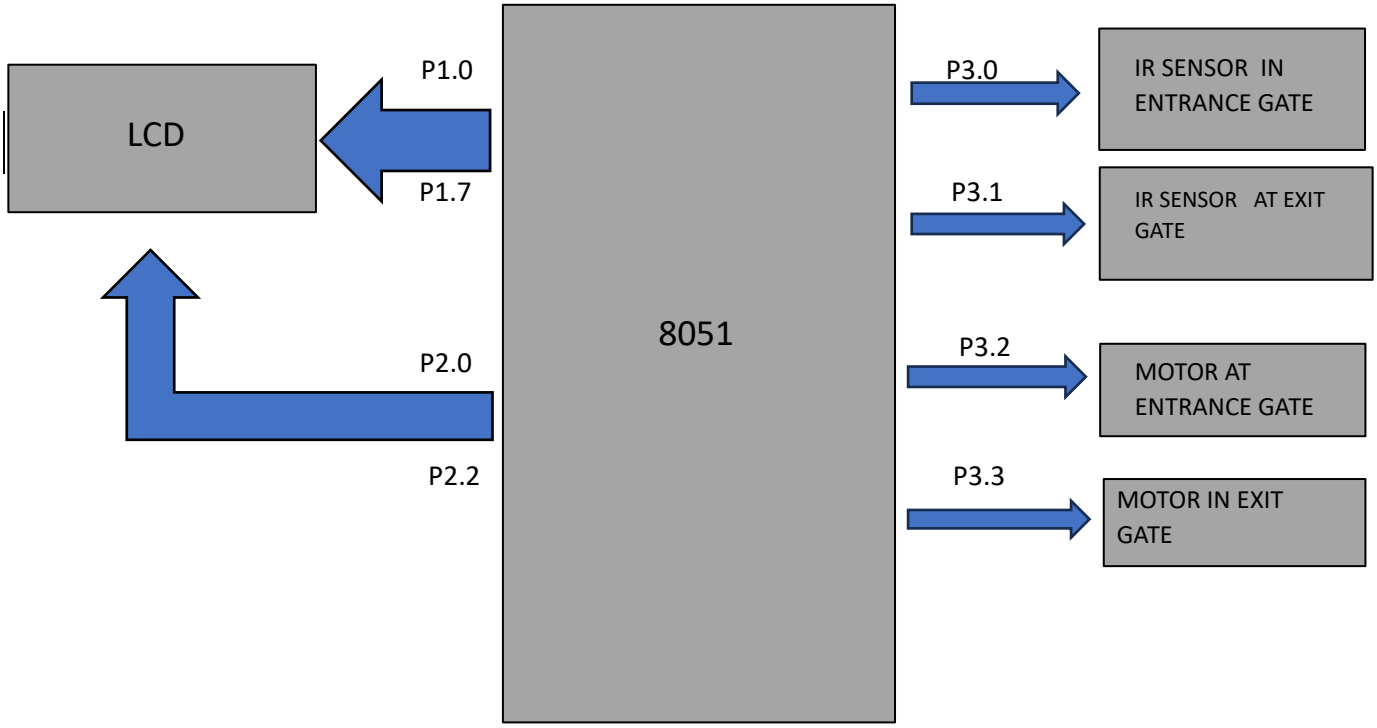
allows for flexibility in system configuration, making it adaptable to various parking environments and requirements.

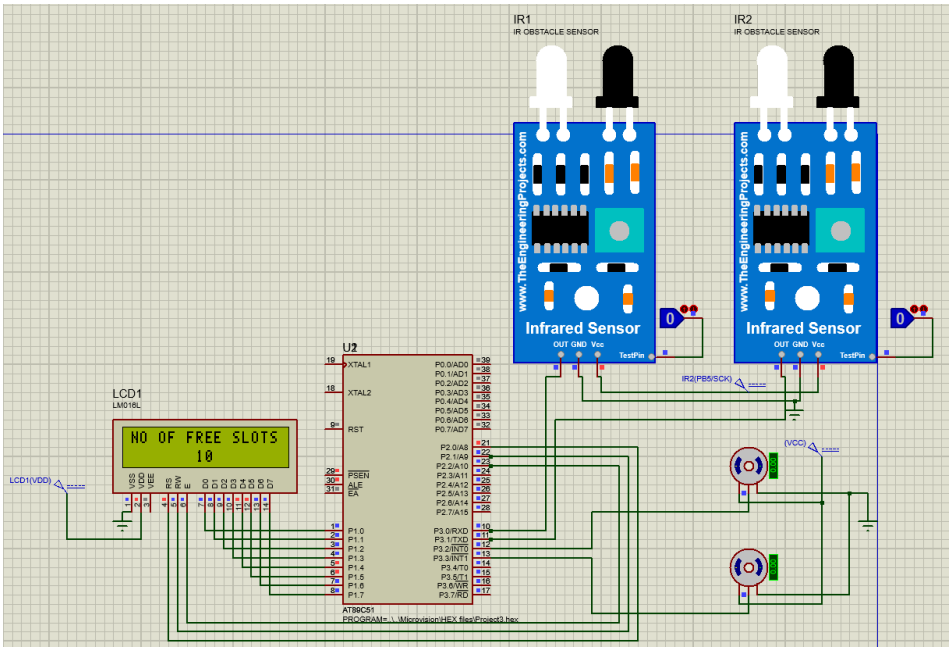
10. Environmentally Friendly: Efficient space utilization and reduced congestion contribute to a more environmentally friendly parking system. By minimizing the time spent searching for parking spots, the system helps to reduce fuel consumption, air pollution, and carbon emissions.

Overall, a smart car parking system utilizing an 8051 microcontroller, IR sensors, servo motors, and an LCD display offers numerous advantages, including optimal space utilization, enhanced efficiency, improved security, real-time information, cost-effectiveness, and user-friendly experience. It represents a technologically advanced solution that enhances parking management while improving the overall parking experience for drivers.

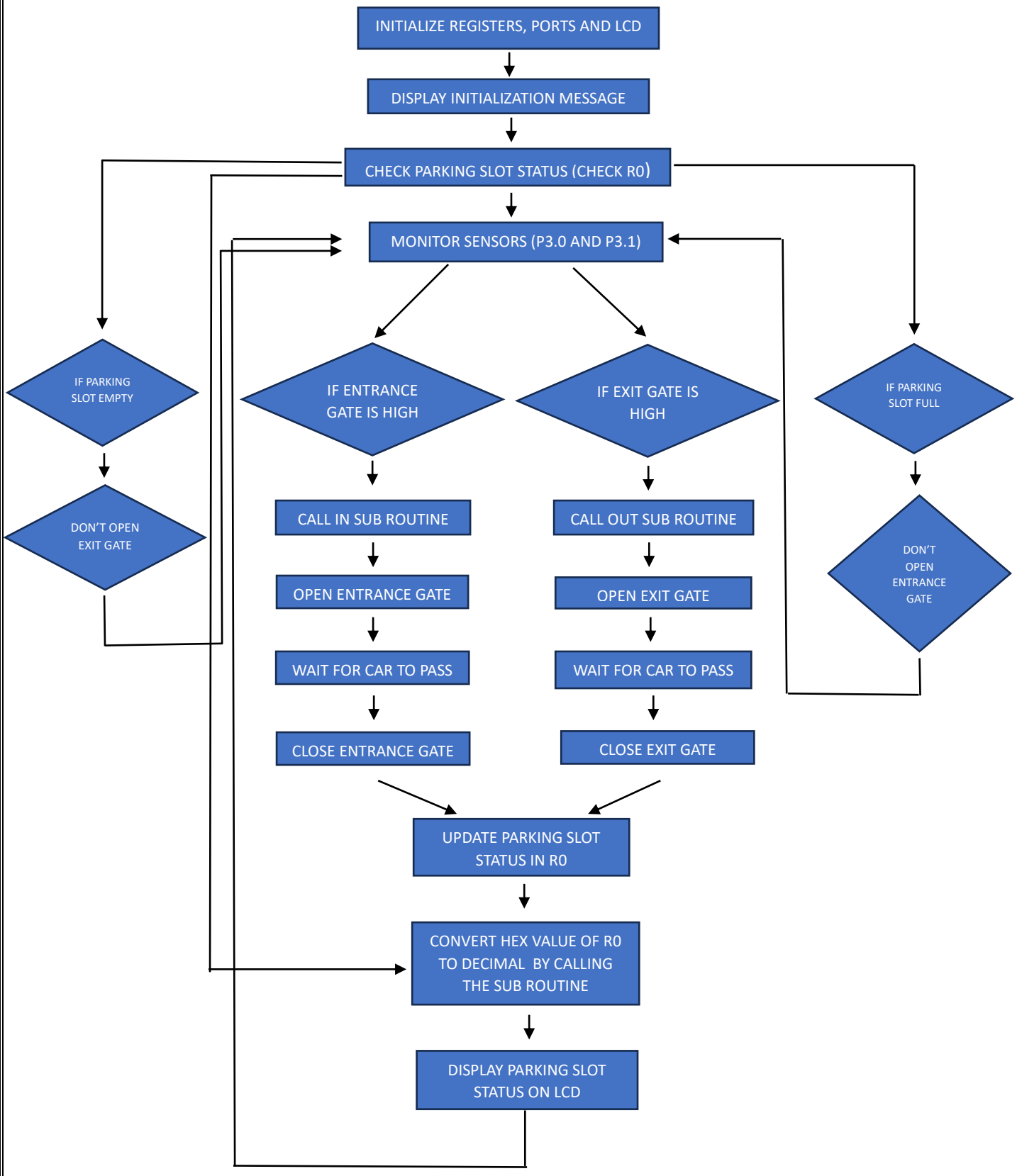
METHODOLOGY:

Block Diagram





Flow Chart.


INITIALIZE REGISTERS, PORTS AND LCD
DISPLAY INITIALIZATION MESSAGE
CHECK PARKING SLOT STATUS (CHECK R0)
MONITOR SENSORS (P3.0 AND P3.1)
IF PARKING SLOT EMPTY
IF ENTRANCE GATE IS HIGH
IF EXIT GATE IS HIGH
IF PARKING SLOT FULL
DON'T OPEN EXIT GATE
CALL IN SUB ROUTINE
CALL OUT SUB ROUTINE
DON'T OPEN ENTRANCE GATE
OPEN ENTRANCE GATE
OPEN EXIT GATE
WAIT FOR CAR TO PASS
WAIT FOR CAR TO PASS
CLOSE ENTRANCE GATE
CLOSE EXIT GATE
UPDATE PARKING SLOT STATUS IN R0
CONVERT HEX VALUE OF R0 TO DECIMAL BY CALLING THE SUB ROUTINE
DISPLAY PARKING SLOT STATUS ON LCD

Program with Comments:

```
ORG 000H

MOV R0,#10;No of free slots for parking, can be extended upto 255

MOV P2,#00H;Setting P2 port as output for controlling the LCD

MOV P3,#00000011B;Making the P3.0 and P3.1 as input to connect sensor,(P3.0 enterance gate and
P3.1 exit gate)

MOV TMOD,#01H;Timer 0 in mode 1 to generate time delay

MOV B,#38H ;init. LCD 2 lines,5x7 matrix

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV B,#0EH ;display on, cursor on

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV B,#06H ;shift cursor right

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV B,#0CH ;display on, cursor on

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV B,#01 ;clear LCD

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV B,#80H ;cursor at line 1,pos.0

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV DPTR,#500H;storing DPTR with 500h to show the message stored in that location to LCD

LOOP:CLR A;clear A to update next location

MOVC A,@A+DPTR;Moving the data from code memory to Accumulator

ACALL DATAWRT;To write the collected above data to LCD

ACALL DELAY;;give LCD some time

INC DPTR;Increment DPTR to fetch next word of the message to be shown
```

```
JZ EXT;if A==0 then the sentence is over, jumps to STAY

SJMP LOOP;if A not equal to 0 the jumps to loop to display next word

// After the Message is shown

EXT:

ACALL CHECKR0;calling CHECKR0 to check the current value of R0(no of slots) to be shown in the LCD


MONITOR:

MOV B,P3;Moving the P3 data to B for monitoring the sensors output

MOV R6,B;Moving to R6 for checking the output

CJNE R6,#00000000B,CHECK01;If the sensor outputs are not equal,jumps to CHECK01 to check the entrance gate whether car is present or not

SJMP MONITOR; if P3==0 then no car has detected so jumps to MONITOR for checking the P3

CHECK01:CJNE R6,#00000001B,CHECK10;If p3.0 not equal to one(entrance gate) then jumps to CHECK10 for exit gate

ACALL IN; If entrane gate is 1 (p3.0) then car is detected and calls IN routine to perform the necessary task

CHECK10:CJNE R6,#00000010B,MONITOR;If p3.1 not equal to one(exit gate) then jumps to MONITOR for checking P3

ACALL OUT; If exit gate is 1 (p3.1) then car is detected and calls OUT routine to perform the necessary task

SJMP MONITOR;then jumps to MONITOR to keep on checking the status of the sensors connected in P3



//If the IR sensor connected to P3.0 gives output as 1,then IN routine is called to perfom task

//like opening the gate in entrance using the motor and update the Register R0

IN:

CJNE R0,#00H,JP;If R0 is 0 then all parking slots are full,so returns from the routine,if not equal then jumps to jp

RET

JP:

DEC R0;When a car is entered R0 is decremented,so one slot is occupied by that car

//A servo motor is connected to P3.2 as a response for entrance gate
```

ACALL NIENTY_DEGREEIN;If a car is entered,then calling NIENTY_DEGREEIN to rotate the motor 90 degree to open the gate,initially the gate will be closed

ACALL SERVO_DELAY;giving some delay, so the motor will be in same state for the passing of car

ACALL ZERO_DEGREEIN;After the delay the motor returns to zero degree so that gate is closed by calling ZERO_DEGREEIN

ACALL CHECKR0;calling CHECKR0 to check the current value of R0(no of slots) and to display in LCD

//After updating the value in LCD,then keep monitoring the entrance and exit sensors connected to P3.0 and P3.1

L1:JNB P3.0,L2;if P3.0 not equals to 0,jumps to L2 to check whether P3.1 is logic high or not

SJMP MONITOR;if P3.0 is 1 then jumps to MONITOR to check the port's input to perform necessary task

L2:JNB P3.1,L1;if P3.1 not equals to 0,jumps to L1 to check whether P3.0 is logic high or not

SJMP MONITOR;if P3.1 is 1 then jumps to MONITOR to check the port's input to perform necessary task

RET

//If the IR sensor connected to P3.1 gives output as 1,then OUT routine is called to perfom task

//like opening the gate in exit using the motor and update the Register R0

OUT:

CJNE R0,#10,JP1;If R0 is 10 then all parking slots are full,so returns from the routine,if not equal then jumps to jp1

RET

JP1:

INC R0;When a car is in exit gate R0 is incremented,so one slot is free by that car

//A servo motor is connected to P3.3 as a response for exit gate

ACALL NIENTY_DEGREEOUT;If a car is in exit gate,then calling NIENTY_DEGREEOUT to rotate the motor 90 degree to open the gate,initially the gate will be closed

ACALL SERVO_DELAY;giving some delay, so the motor will be in same state for the passing of car

ACALL ZERO_DEGREEOUT;After the delay the motor returns to zero degree so that gate is closed by calling ZERO_DEGREEOUT

ACALL CHECKR0;calling CHECKR0 to check the current value of R0(no of slots) and to display in LCD

//After updating the value in LCD,then keep monitoring the entrance and exit sensors connected to P3.0 and P3.1

L4:JNB P3.0,L3;if P3.0 not equals to 0,jumps to L3 to check whether P3.1 is logic high or not

LJMP MONITOR;if P3.0 is 1 then jumps to MONITOR to check the port's input to perform necessary task

L3:JNB P3.1,L4;if P3.1 not equals to 0,jumps to L4 to check whether P3.0 is logic high or not

LJMP MONITOR;if P3.1 is 1 then jumps to MONITOR to check the port's input to perform necessary task

RET

CHECKR0:

ACALL HXT_TO_DEC;Calls this routine to convert the hexadecimal value stored in R0 to decimal value, to display the current status in LCD

//To display the current status in line2

MOV B,#0C7H ;cursor at line 2,pos.7

ACALL COMNWRT ;call command subroutine

ACALL DELAY ;give LCD some time

MOV A,R1;moving the Hundred's place(result of HXT_TO_DEC subroutine) stored in R1 to A,and checking for the corresponding ASCII value of the decimal,so that the number im R1 is displayed in LCD

CJNE R1,#0,D1;If the Hundred's place is 0 the no need to display in LCD,if it is not 0 then need to display so jumps to D1

SJMP D2

D1:ACALL CHECK;Calling the CHECK subroutine to check the value stored in A with the corresponding ASCII value of this decimal so the LCD display can show the number

ACALL DATAWRT;After checking the ASCII value and stored that value in A,calling the DATAWRT routine to display the number in LCD

ACALL DELAY;give LCD some time

D2:MOV A,R2;moving the Ten's place(result of HXT_TO_DEC subroutine) stored in R2 to A and checking for the corresponding ASCII value of the decimal,so that the number im R2 is displayed in LCD

ACALL CHECK;Calling the CHECK subroutine to check the value stored in A with the corresponding ASCII value of this decimal so the LCD display can show the number

ACALL DATAWRT;After checking the ASCII value and stored that value in A,calling the DATAWRT routine to display the number in LCD

ACALL DELAY;give LCD some time

MOV A,R3;;moving the One's place(result of HXT_TO_DEC subroutine) stored in R3 to A and checking for the corresponding ASCII value of the decimal,so that the number im R3 is displayed in LCD

ACALL CHECK;Calling the CHECK subroutine to check the value stored in A with the corresponding ASCII value of this decimal so the LCD display can show the number

ACALL DATAWRT;After checking the ASCII value and stored that value in A,calling the DATAWRT routine to display the number in LCD

ACALL DELAY;give LCD some time and return from this routine

RET

//Converting the Hexadecimal to decimal,to for check the corresponding ASCII values so that the number in R0 is displayed in LCD

HXT_TO_DEC:

MOV A,R0;Moving R0 to A,to perform division

MOV B,#100;Moving 100 to B,to perform division

DIV AB;Dividing the register value by 100,which gives the Quotient value as the Hundreds place of the decimal corresponding to the Hexadecimal

MOV R1,A;storing the hundred's place value in R1

MOV A,B;Moving the Reminder of the above result to A

MOV B,#10;Moving 10 to B,to perform division for obtaining the Ten's and ones's place of the decimal corresponding to the Hexadecimal value in R0

DIV AB;Dividing the Reminder value stored in A by 10 to obtaining the Ten's and ones's place of the decimal corresponding to the Hexadecimal value in R0

MOV R2,A;The resulting Quotient is the Ten's place of the decimal,storing the ten's place to R2

MOV R3,B;The resulting Reminder is the One's place of the decimal,storing the One's place to R2

RET

//Checking the Decimal values with the corresponding ASCII value, so that the LCD can display the numbers

CHECK:

CJNE A,#00,CHECK1;if A(decimal) not equal to 0,jumps to CHECK1

MOV A,#30H;If A(decimal)=0 moves the corresponding ASCII value(30H) to A,for displaying the number 0 in the LCD and returns from this routine

RET

CHECK1:CJNE A,#1,CHECK2;if A(decimal) not equal to 1,jumps to CHECK2

MOV A,#31H;If A(decimal)=1 moves the corresponding ASCII value(31H) to A,for displaying the number 1 in the LCD and returns from this routine

RET

CHECK2:CJNE A,#2,CHECK3;if A(decimal) not equal to 2,jumps to CHECK1

MOV A,#32H;If A(decimal)=2 moves the corresponding ASCII value(32H) to A,for displaying the number 2 in the LCD and returns from this routine

RET

CHECK3:CJNE A,#3,CHECK4;if A(decimal) not equal to 3,jumps to CHECK1

MOV A,#33H;If A(decimal)=3 moves the corresponding ASCII value(33H) to A,for displaying the number 3 in the LCD and returns from this routine

RET

CHECK4:CJNE A,#4,CHECK5;if A(decimal) not equal to 4,jumps to CHECK1

MOV A,#34H;If A(decimal)=4 moves the corresponding ASCII value(34H) to A,for displaying the number 4 in the LCD and returns from this routine

RET

CHECK5:CJNE A,#5,CHECK6;if A(decimal) not equal to 5,jumps to CHECK1

MOV A,#35H;If A(decimal)=5 moves the corresponding ASCII value(35H) to A,for displaying the number 5 in the LCD and returns from this routine

RET

CHECK6:CJNE A,#6,CHECK7;if A(decimal) not equal to 6,jumps to CHECK1

MOV A,#36H;If A(decimal)=6 moves the corresponding ASCII value(36H) to A,for displaying the number 6 in the LCD and returns from this routine

RET

CHECK7:CJNE A,#7,CHECK8;if A(decimal) not equal to 7,jumps to CHECK1

MOV A,#37H;If A(decimal)=7 moves the corresponding ASCII value(37H) to A,for displaying the number 7 in the LCD and returns from this routine

RET

CHECK8:CJNE A,#8,CHECK9;if A(decimal) not equal to 8,jumps to CHECK1

MOV A,#38H;If A(decimal)=8 moves the corresponding ASCII value(38H) to A,for displaying the number 8 in the LCD and returns from this routine

RET

CHECK9:CJNE A,#9,NOTEQ;if A(decimal) not equal to 9,jumps to NOTEQ

MOV A,#39H;If A(decimal)=9 moves the corresponding ASCII value(39H) to A,for displaying the number 9 in the LCD and returns from this routine

```
NOTEQ:RET



COMNWRT: ;send command to LCD
MOV P1,B ;copy reg A to port1
CLR P2.0 ;RS=0 for command
CLR P2.1 ;R/W=0 for write
SETB P2.2 ;E=1 for high pulse
ACALL DELAY ;give LCD some time
CLR P2.2 ;E=0 for H-to-L pulse
RET
DATAWRT: ;write data to LCD
MOV P1,A ;copy reg A to port1
SETB P2.0 ;RS=1 for data
CLR P2.1 ;R/W=0 for write
SETB P2.2 ;E=1 for high pulse
ACALL DELAY ;give LCD some time
CLR P2.2 ;E=0 for H-to-L pulse
RET
//Delay subroutine,to give LCD some time to perfom the given task
DELAY: MOV R4,#20
HERE2: MOV R5,#255
HERE: DJNZ R5,HERE
DJNZ R4,HERE2
RET


//To create a pulse of 1ms for zero degree rotation in servo motor(for closing the gate)
ZERO_DEGREEIN:
MOV TH0, #0FCH
MOV TL0, #66H
SETB P3.2 ;Make P2.0 HIGH
```

```
SETB TR0 ;Start the timer 0,
WAIT1:JNB TF0, WAIT1 ;Wait till the TF0 flag is set
CLR P3.2 ;Make P2.0 LOW
CLR TF0 ;Clear the flag manually
CLR TR0 ;Stop the timer 0
RET
ZERO_DEGREEOUT:
MOV TH0, #0FCH
MOV TL0, #66H
SETB P3.3 ;Make P2.0 HIGH
SETB TR0 ;Start the timer 0
WAIT3:JNB TF0, WAIT3 ;Wait till the TF0 flag is set
CLR P3.3 ;Make P2.0 LOW
CLR TF0 ;Clear the flag manually
CLR TR0 ;Stop the timer 0
RET

//To create a pulse of 1.5ms for 90 degree rotation of sevromotor(to open the gate)
NIENTY_DEGREEIN:
MOV TH0, #0FAH
MOV TL0, #99H
SETB P3.2 ;Make P2.0 HIGH
SETB TR0 ;Start the timer 0
WAIT2:JNB TF0, WAIT2 ;Wait till the TF0 flag is set
CLR P3.2 ;Make P2.0 LOW
CLR TF0 ;Clear the flag manually
CLR TR0 ;Stop the timer 0
RET
NIENTY_DEGREEOUT:
MOV TH0, #0FAH
MOV TL0, #99H
```

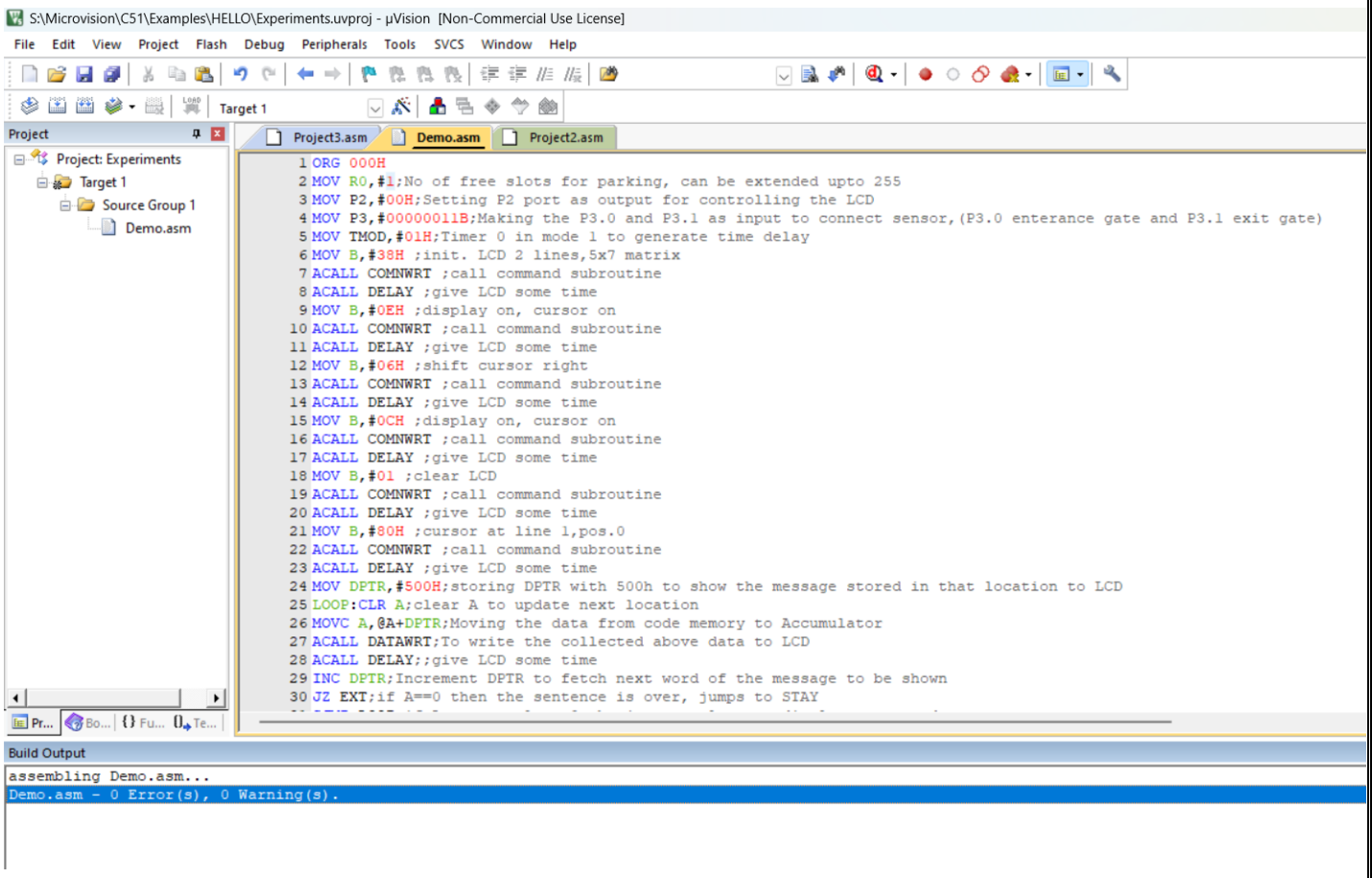
```
SETB P3.3 ;Make P2.0 HIGH
SETB TR0 ;Start the timer 0
WAIT4:JNB TF0, WAIT4 ;Wait till the TF0 flag is set
CLR P3.3;Make P2.0 LOW
CLR TF0 ;Clear the flag manually
CLR TR0 ;Stop the timer 0
RET

//To make servo motor stay in same position for some seconds(until the car passes)
//This routine will provide a delay of 5seconds,which can be modified by changing the value in R7
SERVO_DELAY:
MOV R7,#71
AGAIN4:MOV TH0,#00H
MOV TL0,#00H
SETB TR0
WAIT5:JNB TF0,WAIT5
CLR TF0
CLR TR0
DJNZ R7,AGAIN4
RET

ORG 500H
DB 'NO OF FREE SLOTS',0
END
```

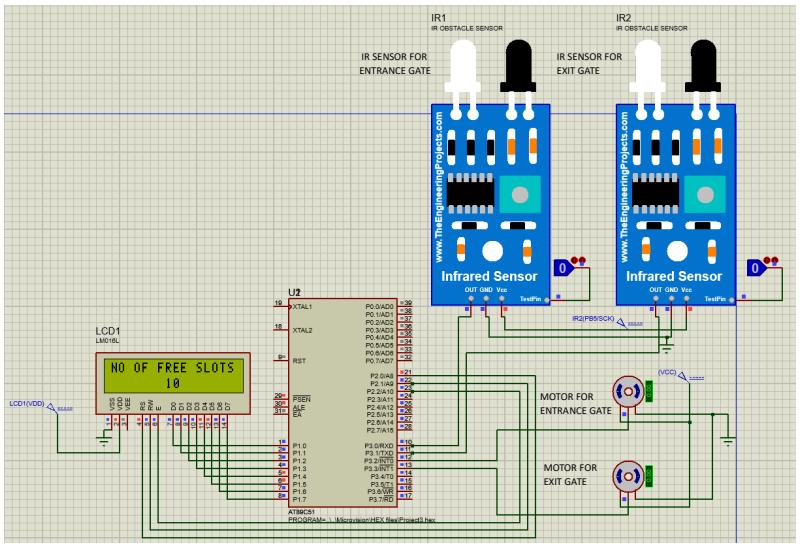
Result:

Program with Zero syntax error.

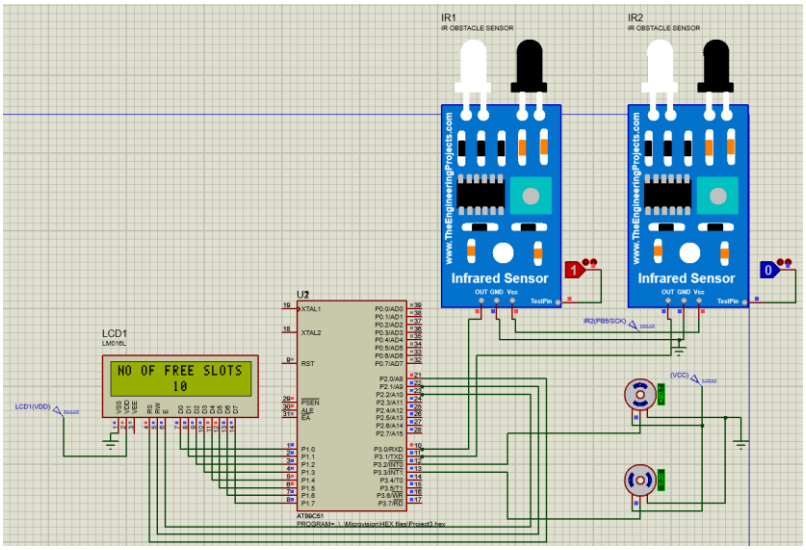


Proteus Simulation Result :

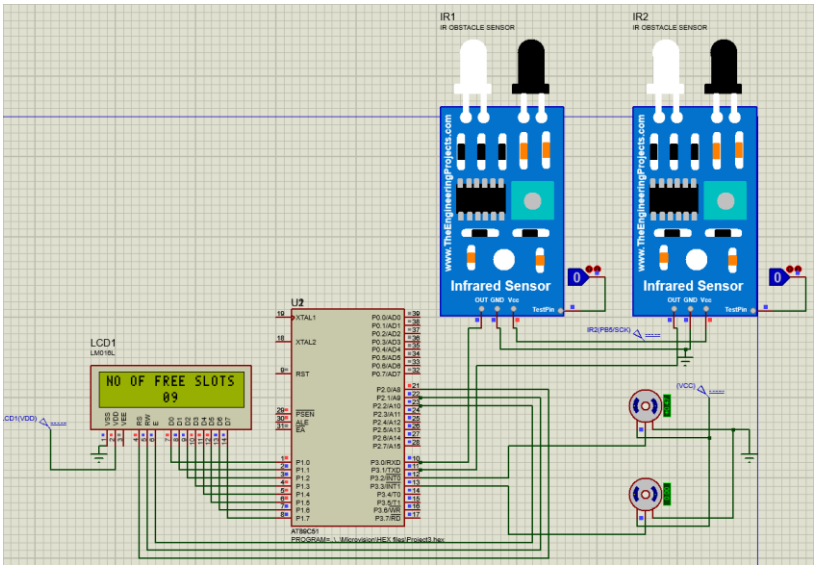
1.Initial stage of working ,2 sensor's output are low and 2 servo motors are closed and LCD displays the total slots i.e 10.



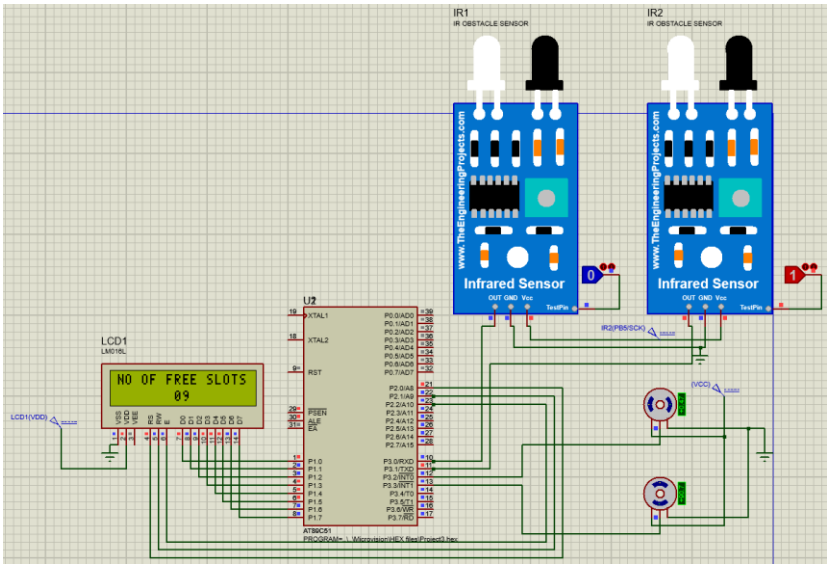
2.When the car is in front of the entrance gate, ir sensor kept in entrance gate will be high and the servo motor rotates 90 degree i.e the gate will be opened.



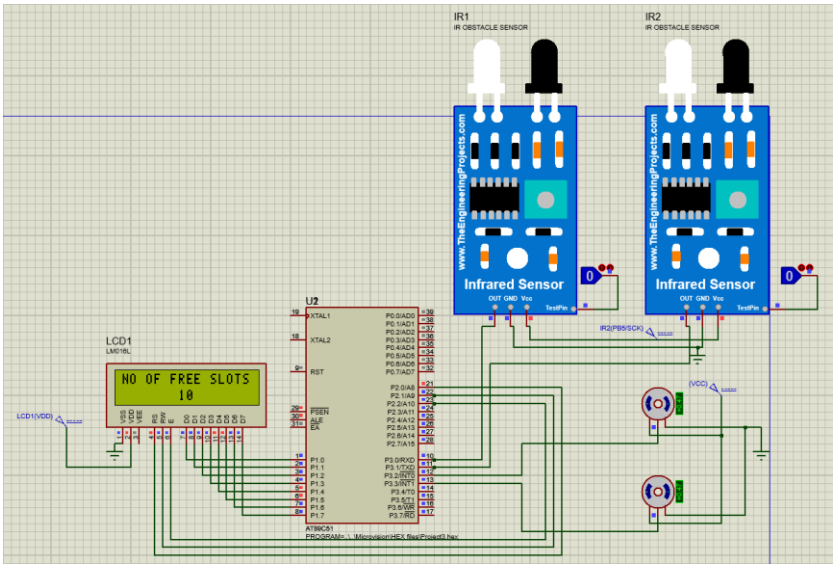
3. After 5 seconds, the gate will be closed and the no .of free slots will be decremented by 1. i.e (10 to 9)



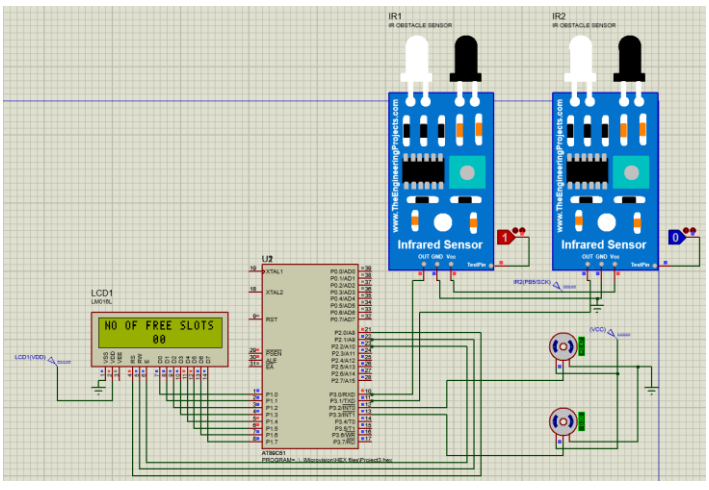
4. When the car is in front of the exit gate, ir sensor kept in exit gate will be high and the servo motor rotates 90 degree i.e the gate will be opened.



5. After 5 seconds, the gate will be closed and the no .of free slots will be incremented by 1.in this case (9 to 10)



6.When there is no free slot , eventhough the sensor senses ,the gate present in entrance gate will not open .



7.Eventhough the sensor in exit gate accidentally senses when there is no car inside the parking area, the exit gate doesn't open.

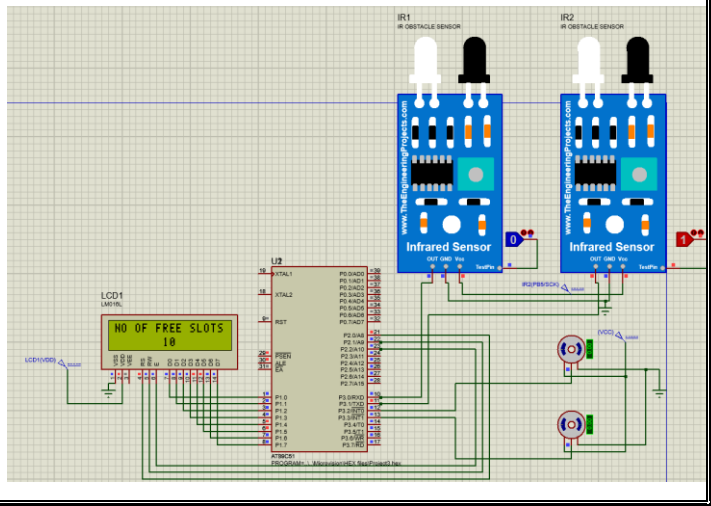
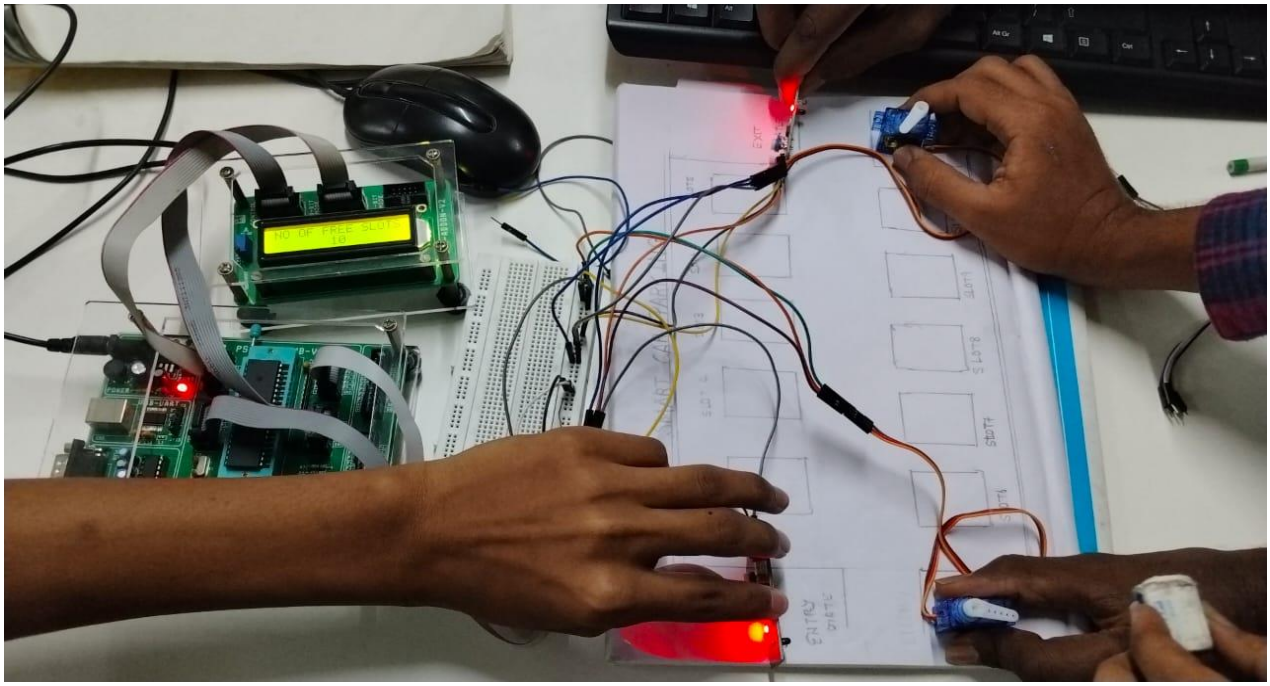
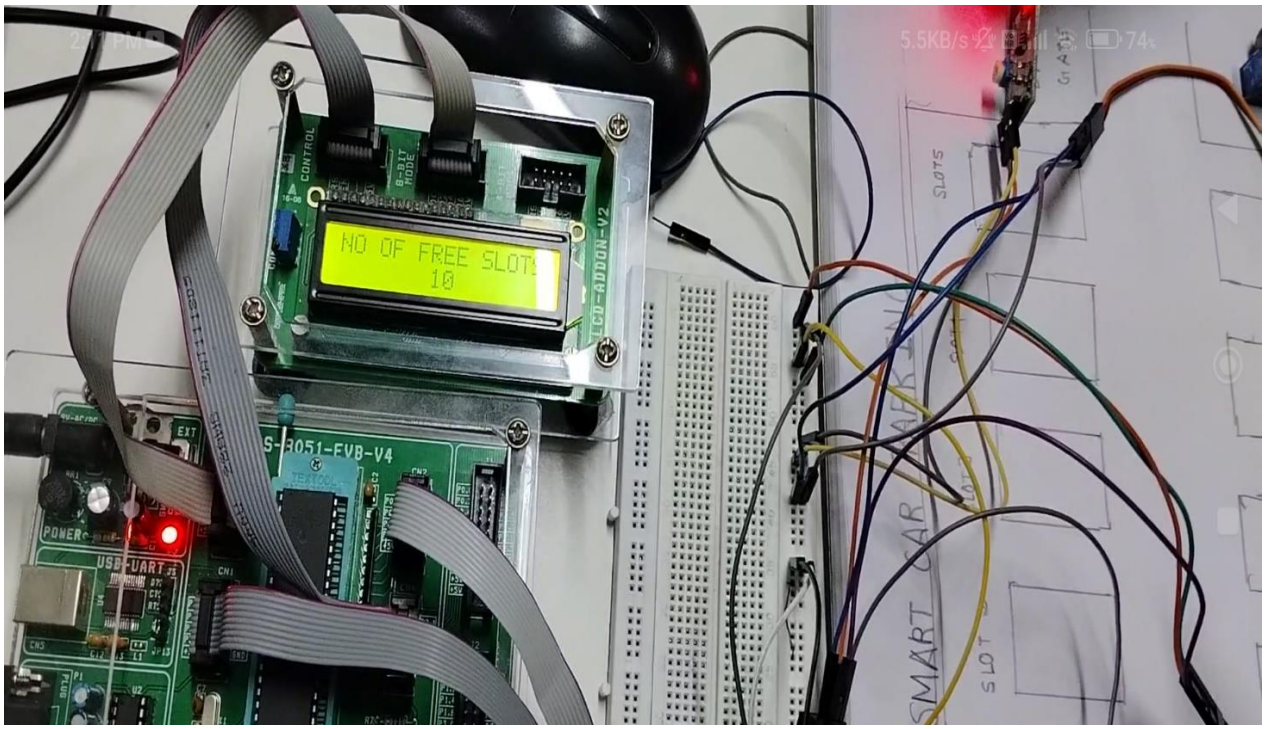


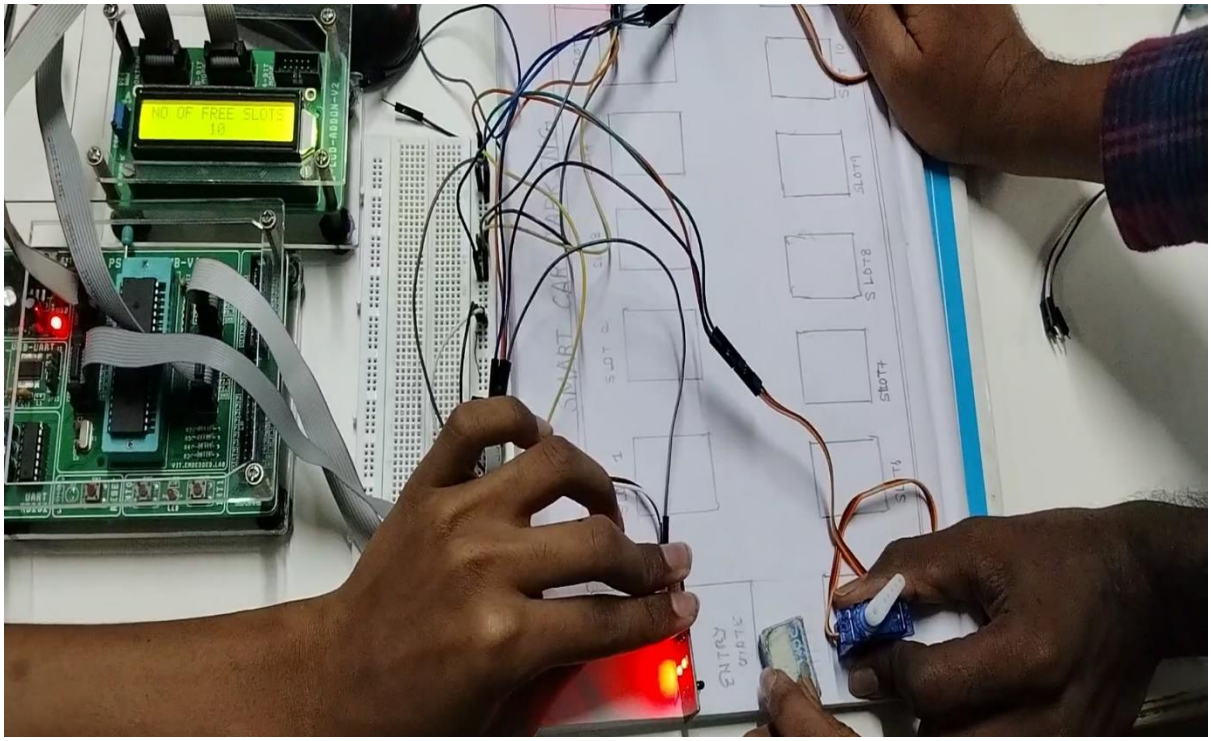
Photo of the hardware demonstration .

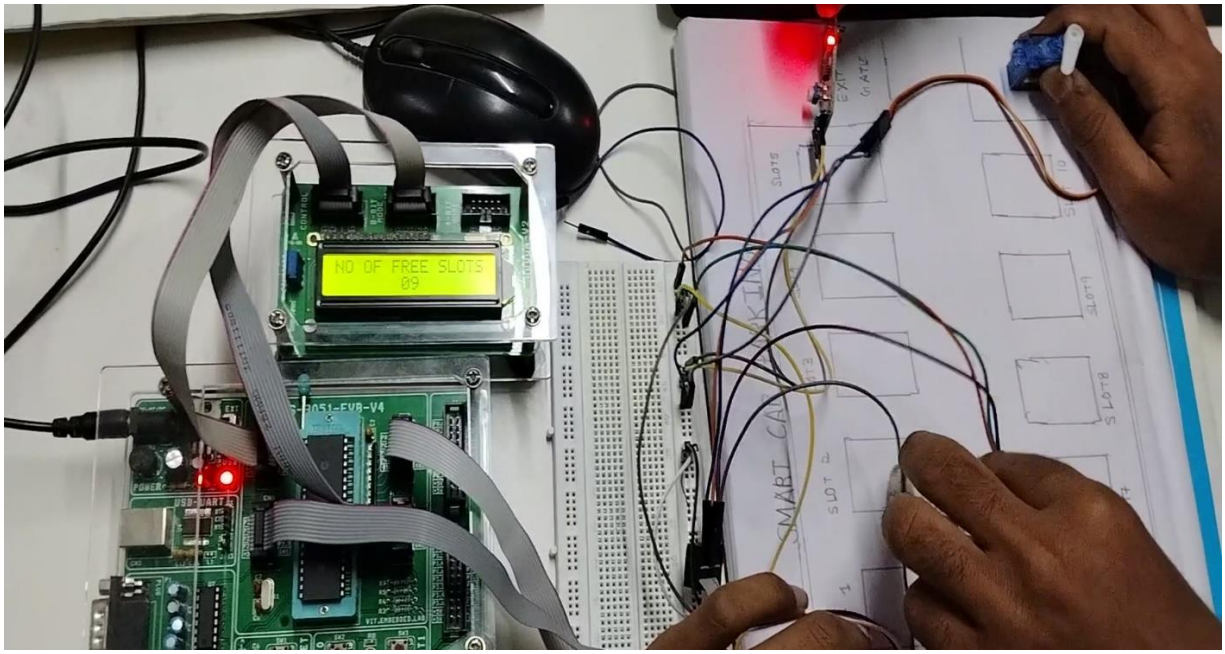
Initial stage:



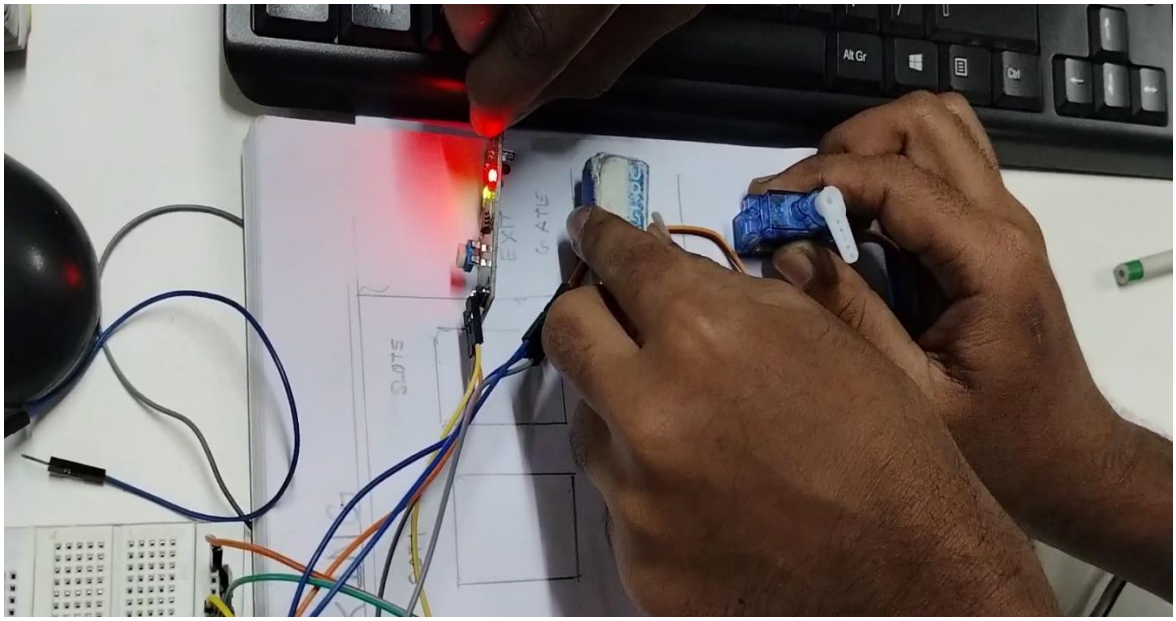


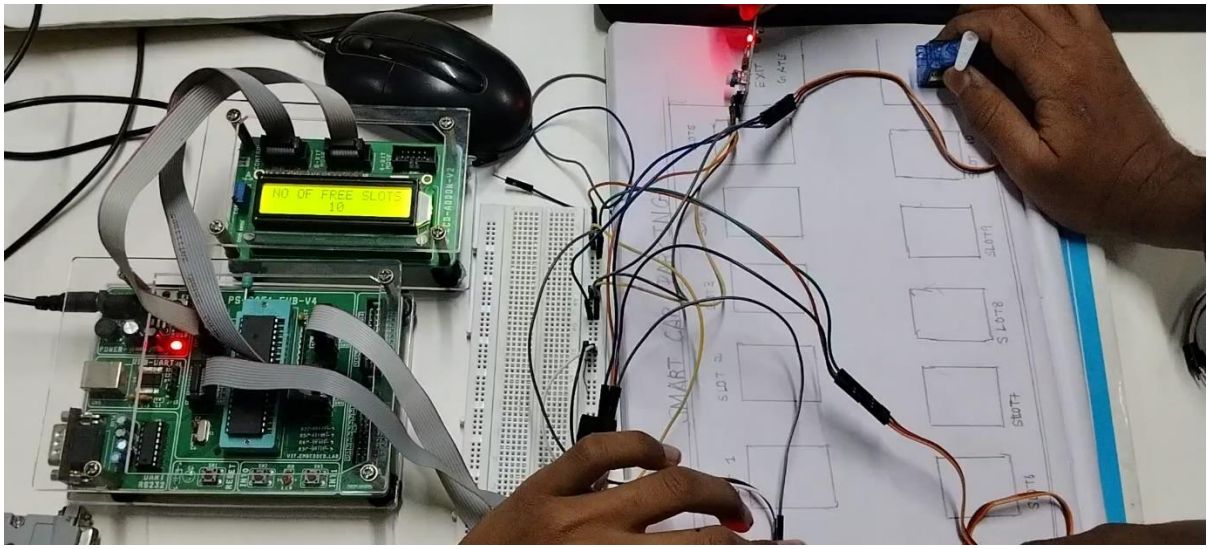
When a car enters

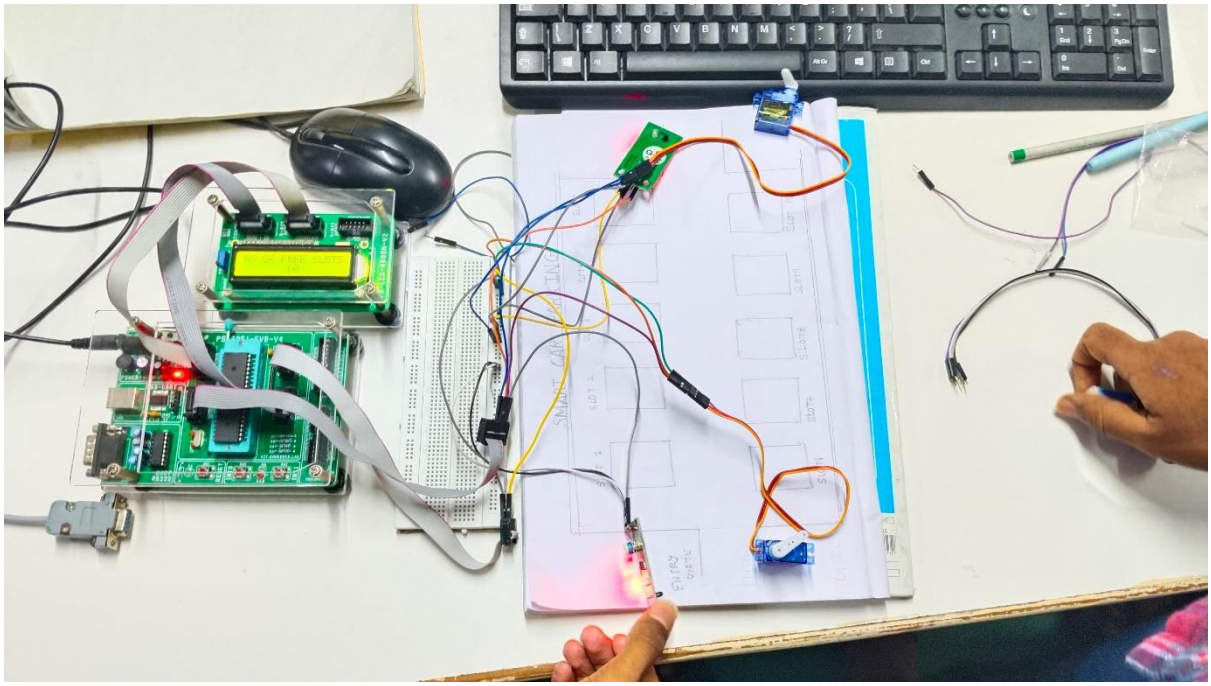




When a car is leaving







Our preparation video:

https://drive.google.com/file/d/1Ye1F6rvkmcXyYnLElvZshQH8VeqsdL7P/view?usp=drivesdk

CONCLUSION :

In conclusion, the smart car parking system utilizing an 8051 microcontroller, IR sensors, servo motors, and an LCD display presents a sophisticated solution to traditional parking challenges. By combining advanced technologies, this system offers efficient parking management, space optimization, and a streamlined parking experience for drivers.

THANK YOU !