WEEK 7

Session 5: A parking plot has a certain capacity of cars/two wheelers. Number of empty spots should be displayed on the display outside the Parking Plot and also display the spots available using Arduino Uno

AIM :A parking plot has a certain capacity of cars/two wheelers. Number of empty spots should be displayed on the display outside the Parking Plot and also display the spots using Arduino Uno.

HARDWIRE REQUIRED

- Arduino UNO
- IR Sensor module- 5 No's
- 20X4 LCD display
- LCD_I2C connector
- Bread Board
- Connecting Wires

SOFTWARE REQUIRED

- Proteus for simulation
- Arduino

Block Diagram

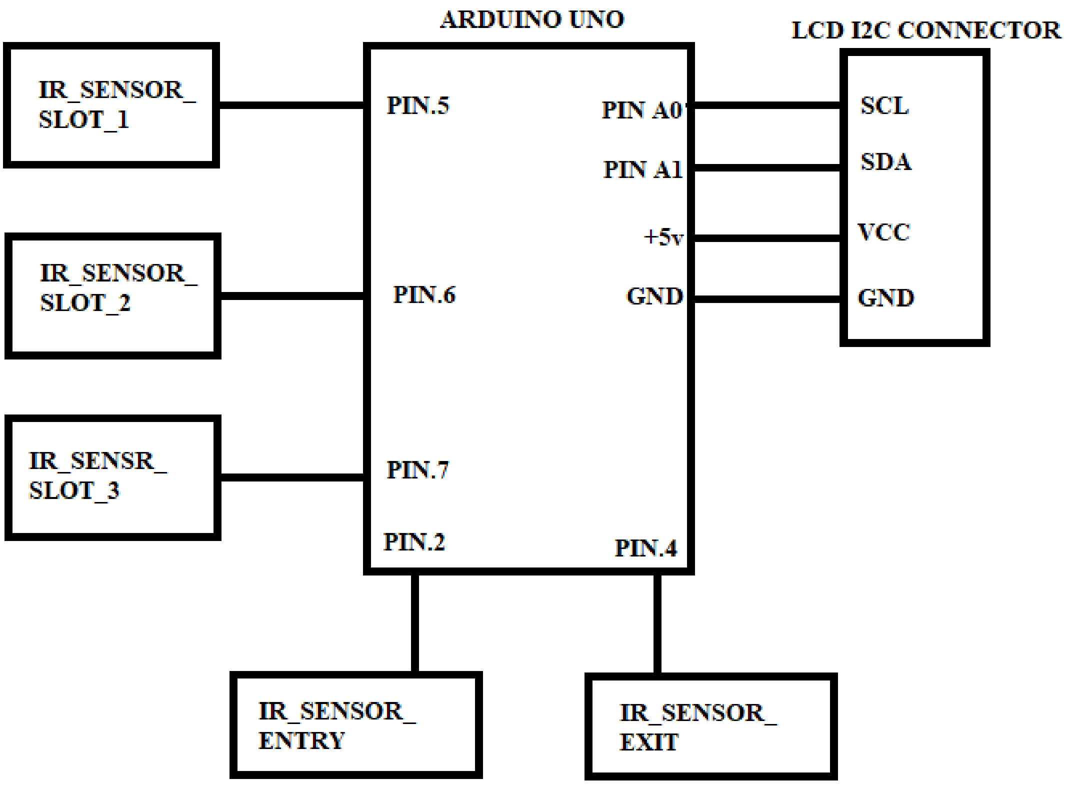


Fig: Parking Slot Arduino Uno block diagram

Circuit Diagram

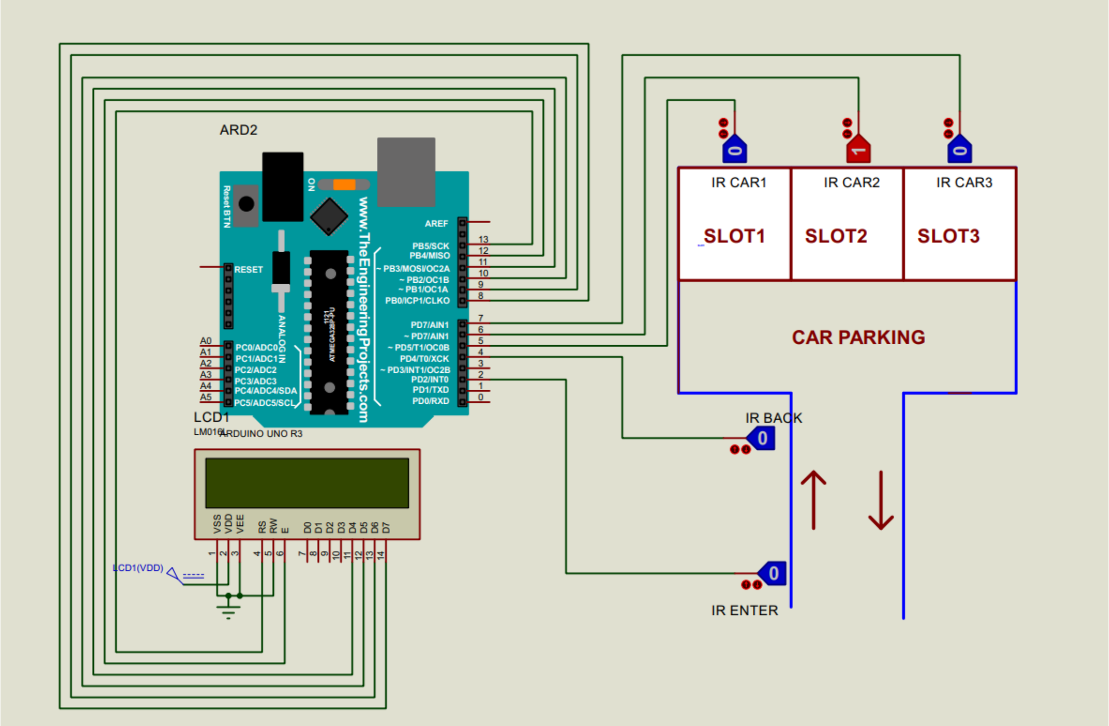


Fig : Circuit diagram showing Connections between Arduino uno, IR sensors,LCD display

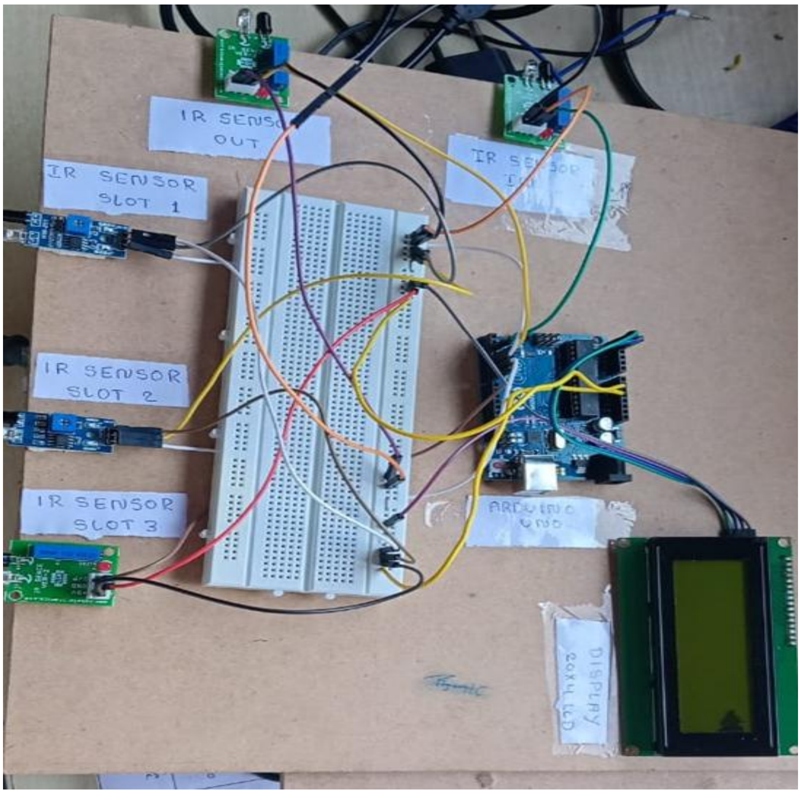


Fig : Parking slot display Arduino uno board

Procedure for Simulation

1.Double click on Proteus 8 icon and open the software.

2.Go to file- Create new project-name the project and save-click on next-choose schematic diagram template and portrait -click next-select do not create PCB layout-select no firmware project-click next-click finish.

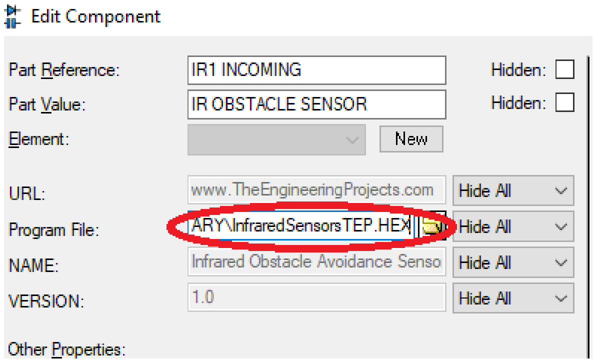
3. Select the following components required by clicking on "Picking Devices"

- Arduino Uno
- IR sensor
- LogicState
- M016L(16x2 LCD Display)
- Ground

NOTE: Add arduino and IR sensor libraries to Proteus Library

4.Make connections as per schematic diagram

5. Double click this Infrared Sensor and in edit Properties Panel will open up. Browse to the file InfraredSensorTEP.HEX which is placed in the Library folder of Proteus and click Ok.



Edit Component

Part Reference: IR1 INCOMING Hidden: □

Part Value: IR OBSTACLE SENSOR Hidden: □

Element: New

URL: www.TheEngineeringProjects.com Hide All

Program File: ARY\InfraredSensorsTEP.HEX Hide All

NAME: Infrared Obstacle Avoidance Senso Hide All

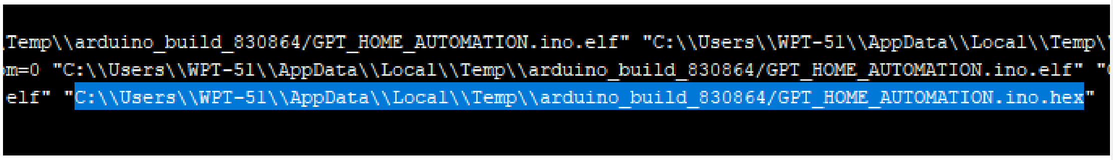
VERSION: 1.0 Hide All

Other Properties:

6.If test Pin of IR sensor is LOW, then sensor will remain normal and if it's HIGH then sensor will behave as it has something in front of it.

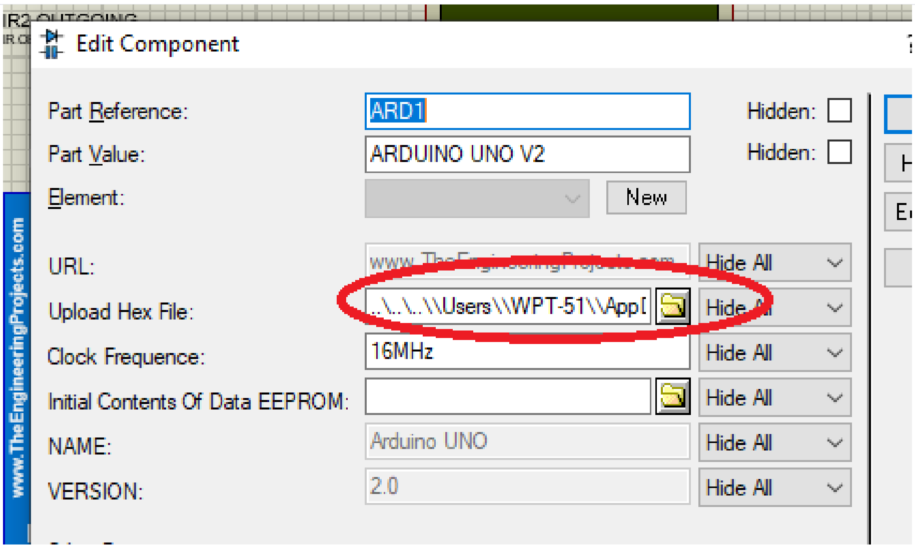
7.Open Arduino IDE ,Type the program and Check the compilation option in File—Preferences to generate the hex file.

8.Compile the code and copy the hex file path.



```
Temp\\arduino_build_830864/GPT_HOME_AUTOMATION.ino.elf" "C:\\Users\\WPT-51\\AppData\\Local\\Temp\\
m=0 "C:\\Users\\WPT-51\\AppData\\Local\\Temp\\arduino_build_830864/GPT_HOME_AUTOMATION.ino.elf" "
elf" "C:\\Users\\WPT-51\\AppData\\Local\\Temp\\arduino_build_830864/GPT_HOME_AUTOMATION.ino.hex"
```

9.Double Click on arduino board to insert hex file code and click ok.



Edit Component

Part Reference:	ARD1	Hidden: □	
Part Value:	ARDUINO UNO V2	Hidden: □	
Element:		New	
URL:	www.TheEngineeringProjects.com	Hide All	
Upload Hex File:	..\..\..\Users\\WPT-51\\AppD	Hide All	
Clock Frequence:	16MHz	Hide All	
Initial Contents Of Data EEPROM:		Hide All	
NAME:	Arduino UNO	Hide All	
VERSION:	2.0	Hide All	

10.Start the simulation and change logic states of IR sensors to" 0" and "1" to check the output.

Procedure for Arduino Kit Downloading

1.Open Arduino ide software and type the program.
2.Compile and check for errors,if no errors go to TOOL—Select Arduino Board and PORT e connections as below

- Use 3-line connector, connect the connector head to IR module of entry
  Connect the VCC pin to 5v power supply section
  Connect the GND pin to GND power supply section
  Connect the signal pin to GPIO of 2 Arduino PIN
- Use 3-line connector, connect the connector head to IR module of exit
  Connect the VCC pin to 5v power supply section
  Connect the GND pin to GND power supply section
  Connect the signal pin to GPIO of 4 Arduino PIN.
- Use 3-line connector, connect the connector head to IR Sensor Slot_1
  Connect the VCC pin to 5v power supply section
  Connect the GND pin to GND power supply section
  Connect the signal pin to GPIO of 5 Arduino PIN
- Use 3-line connector, connect the connector head to IR Sensor Slot_2
  Connect the VCC pin to 5v power supply section
  Connect the GND pin to GND power supply section
  Connect the signal pin to GPIO of 6 Arduino PIN
- Use 3-line connector, connect the connector head to IR Sensor Slot_3
  Connect the VCC pin to 5v power supply section
  Connect the GND pin to GND power supply section
  Connect the signal pin to GPIO of 7 Arduino PIN
- Use 4-line connector, connect the connector head to LCD I2C connector module of pin header [SDA, SCL, VCC, GND] Connect the VCC pin to 5v power supply section, Connect the GND pin to GND of Arduino board. Connect the SCL pin to GPIO of A0 Arduino PIN Connect the SDA pin to GPIO of A1 Arduino PIN

4.Dowload the program to kit.

5.Check the output .

Program:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
#define ir_enter 2
#define ir_back  4
#define ir_car1 5
#define ir_car2 6
#define ir_car3 7
int S1=0, S2=0, S3=0;
int flag1=0, flag2=0;
int slot = 3;
void setup(){
   lcd.init();                      // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("PARKING SLOT DISPLAY");
Serial.begin(9600);
lcd.begin(20, 4);
pinMode(ir_car1, INPUT);
pinMode(ir_car2, INPUT);
pinMode(ir_car3, INPUT);
pinMode(ir_enter, INPUT);
pinMode(ir_back, INPUT);

Read_Sensor();

int total = S1+S2+S3;
slot = slot - total;
}
void loop(){

Read_Sensor();

lcd.setCursor (0,1);
lcd.print("occupied Slot: ");
lcd.print(slot);
lcd.print("    ");

lcd.setCursor (0,2);
if(S1==0){lcd.print("S1:Fill ");}
    else{lcd.print("S1:Empty");}
lcd.setCursor (10,2);
if(S2==0){lcd.print("S2:Fill ");}
     else{lcd.print("S2:Empty");}

lcd.setCursor (0,3);
if(S3==0){lcd.print("S3:Fill ");}
     else{lcd.print("S3:Empty");}

  if(digitalRead (ir_back) == 0 && flag1==0) // read digital data from
IR sensor1
   {
    if(slot>0)
     {
      flag1=1;
      if(flag2==0)
       {
               slot = slot-1;
        }
     }

   }

  if(digitalRead (ir_enter) == 0 && flag2==0) // read degital data from
IR sensor2
   {
    flag2=1;
    if(flag1==0)
     {
          slot = slot+1;
      }
   }
   if(flag1==1 && flag2==1)
   {
    delay (1000);
      flag1=0, flag2=0;
    }
        delay(1);
}

void Read_Sensor(){
S1=1, S2=1, S3=1;

if(digitalRead(ir_car1) == 0){S1=0;}
if(digitalRead(ir_car2) == 0){S2=0;}
if(digitalRead(ir_car3) == 0){S3=0;}

}
```

Program for Proteus simulation

```
#include<LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);
#defineir_enter 2
#define ir_back  4
#define ir_car1 5
#define ir_car2 6
#define ir_car3 7
int S1=0, S2=0, S3=0;
int flag1=0, flag2=0;
int slot = 3;
void setup(){
Serial.begin(9600);

pinMode(ir_car1, INPUT);
pinMode(ir_car2, INPUT);
pinMode(ir_car3, INPUT);


pinMode(ir_enter, INPUT);
pinMode(ir_back, INPUT);
lcd.begin(16, 2);
Read_Sensor();

int total = S1+S2+S3;
slot = slot-total;
lcd.clear();
lcd.setCursor (0,0);
lcd.print("Available Slot:");
lcd.print(slot);
}
void loop(){

Read_Sensor();
delay(500);
lcd.setCursor (0,1);
if(S1==1){lcd.print("1:Fil");}
    else{lcd.print("1:Emp");}

lcd.setCursor (5,1);
if(S2==1){lcd.print("2:Fil");}
    else{lcd.print("2:Emp");}

lcd.setCursor (11,1);
if(S3==1){lcd.print("3:Fil");}
    else{lcd.print("3:Emp");}

if(digitalRead (ir_enter) == 1 && flag1==0){
if(slot>0){flag1=1;
if(flag2==0){ slot = slot-1;}
}else{
```

```
lcd.clear();
lcd.setCursor (0,0);
lcd.print("SorryParkingFull");
delay(500);
}
}

if(digitalRead (ir_back) == 1 && flag2==0){flag2=1;
if(flag1==0){ slot = slot+1;}
}

if(flag1==1 && flag2==1){
delay (1000);

flag1=0, flag2=0;
}

delay(500);
lcd.setCursor (0,0);
lcd.print("Available Slot:");
lcd.print(slot);

}

void Read_Sensor(){
S1=0, S2=0, S3=0;

if(digitalRead(ir_car1) == 1){S1=1;}
if(digitalRead(ir_car2) == 1){S2=1;}
if(digitalRead(ir_car3) == 1){S3=1;}

}
```