SMART PARKIING

Name:Guntamadugu Pavan kumar

Register no: 711121106026

Component details:

Arduino Uno,

20×4 LCD Display,

I2C LCD Module,

Male Header,

Female Header,

IR Sensor x 8,

Mini Servo Motor SG-90 Female DC Power Jack,

5v 2Amp Power Adapter

ARDUINO UNO

The Arduino Uno is a popular microcontroller board that is widely used by hobbyists, students, and professionals for various electronic projects. It is part of the Arduino platform, which is an open-source hardware and software ecosystem designed for easy prototyping and development of interactive electronic projects.

Here are some key features and information about the Arduino Uno:

Microcontroller: The Arduino Uno is built around the ATmega328P microcontroller, which is a low-power, high-performance 8-bit AVR microcontroller. It operates at 16 MHz and has 32KB of flash memory for storing your program code.

Digital and Analog I/O: The board has 14 digital input/output pins, where 6 can be used as PWM (Pulse Width Modulation) outputs. It also has 6 analog input pins for reading analog sensors.

Voltage Levels: The Arduino Uno operates at 5 volts, making it compatible with a wide range of sensors and components. It also has a 3.3V output and an I2C (Inter-Integrated Circuit) compatible header for interfacing with 3.3V devices.

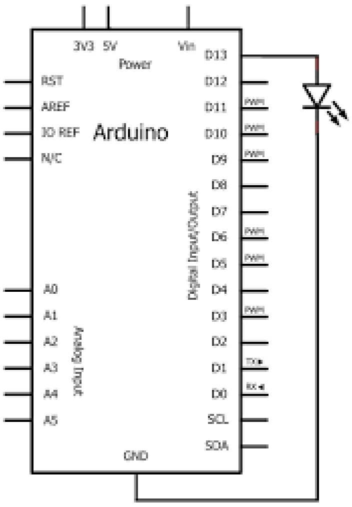
USB Interface: It has a built-in USB interface, which makes it easy to program the board and communicate with your computer. You can upload your code through the USB connection.

Power Supply: You can power the Arduino Uno through the USB connection, an external power supply, or a battery. It has a voltage regulator that can handle a wide range of input voltages (7-12V).

Programming: The Arduino Uno can be programmed using the Arduino IDE (Integrated Development Environment), which is a user-friendly software tool for writing and uploading code to the board. The programming language is based on C/C++.

Open Source: Arduino is an open-source platform, which means that both the hardware and software designs are freely available for anyone to use, modify, and share.

Shields: Arduino Uno is compatible with a variety of "shields," which are add-on boards that can be easily attached to extend its capabilities. For example, there are shields for adding WiFi, Ethernet, Bluetooth, and more.



IR SENSOR x8

It seems you are interested in IR sensors, specifically eight of them. Infrared (IR) sensors are commonly used in various applications, including remote controls, proximity detection, object detection, and more. Here's some information on using eight IR sensors in a project:

Sensor Selection: Choose the type of IR sensors that best suit your project's requirements. Common types include Infrared Reflective Sensors, Infrared Proximity Sensors, and Infrared Obstacle Detection Sensors.

Wiring: Connect the eight IR sensors to your microcontroller, such as the Arduino Uno. Each sensor typically has three pins: VCC (power supply), GND (ground), and OUT (output). Make sure to connect them correctly.

Power Supply: Ensure that the power supply to the IR sensors is stable and appropriate for the sensor's specifications. Typically, IR sensors work with voltages between 3.3V and 5V.

Interfacing: Write code to interface with the IR sensors. You can use digital pins for reading the sensor outputs. Be sure to configure your microcontroller's pins accordingly.

Reading Data: Read the sensor data. Depending on the type of IR sensor, you might get analog or digital data. Infrared Reflective Sensors usually provide digital data, while Infrared Proximity Sensors might give analog data.

Processing Data: Process the data from the sensors as needed for your specific application. For example, you can set thresholds for distance or reflection values, depending on the sensors you are using.

Multiple Sensors: When using eight IR sensors, consider how you'll manage the data from all of them. You may want to use arrays or lists to store and process the data from each sensor.

Application: Depending on your project, you can use the data from these sensors for tasks such as obstacle avoidance in robotics, line following, gesture control, or any application that involves detecting the presence or absence of objects.

Calibration: Calibrate your sensors if needed. Different environmental conditions can affect the performance of IR sensors, so calibration may be necessary to ensure accuracy.

Testing: Thoroughly test your setup to ensure that it performs as expected. Make any necessary adjustments to your code and hardware if issues arise during testing

Mini Servo Motor SG-90:

The SG-90 is a popular and widely used mini servo motor. It's often used in hobbyist projects, robotics, and remote control applications.

Servo motors are devices that can precisely control the position of the motor's shaft. They are designed for small to medium-sized applications that require accurate and repeatable movements.

The SG-90 servo motor is small in size and typically operates at 4.8-6V. It's known for its lightweight, affordability, and compatibility with microcontrollers like Arduino.

These servo motors have three wires: power (VCC), ground (GND), and control (signal) wire. You can use PWM (Pulse Width Modulation) signals to control the position of the servo motor's shaft.

Female DC Power Jack:

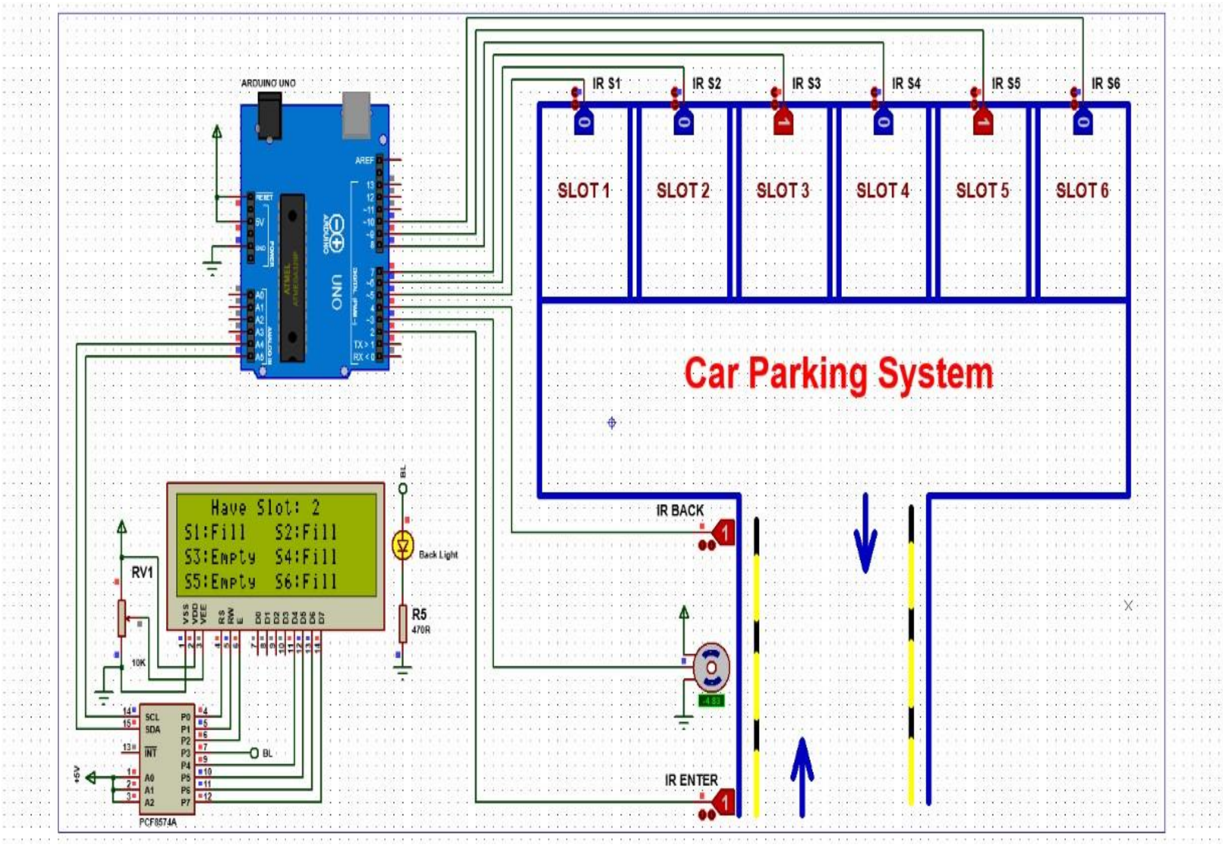
A female DC power jack is a connector that is used to provide power to various electronic devices, circuits, or components.

It is typically used for connecting an external power source, such as a DC power adapter or battery, to a device or circuit.

Female DC power jacks have two or more terminals for connecting to the power source and the device. They are often labeled with positive (+) and negative (-) markings to ensure correct polarity.

The common sizes for these jacks include 2.1mm and 2.5mm, which are often used in small electronic projects and device

Circuit diagram:



Code for smart parking

```
#include <Servo.h> //includes the servo library

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);


Servo myservo;


#define ir_enter 2

#define ir_back  4


#define ir_car1 5

#define ir_car2 6

#define ir_car3 7
```

```
#define ir_car4 8

#define ir_car5 9

#define ir_car6 10


int S1=0, S2=0, S3=0, S4=0, S5=0, S6=0;

int flag1=0, flag2=0;

int slot = 6;


void setup(){

Serial.begin(9600);


pinMode(ir_car1, INPUT);

pinMode(ir_car2, INPUT);

pinMode(ir_car3, INPUT);

pinMode(ir_car4, INPUT);

pinMode(ir_car5, INPUT);

pinMode(ir_car6, INPUT);


pinMode(ir_enter, INPUT);

pinMode(ir_back, INPUT);


myservo.attach(3);

myservo.write(90);


lcd.begin(20, 4);

lcd.setCursor (0,1);

lcd.print("   Car  parking  ");

lcd.setCursor (0,2);

lcd.print("      System     ");

delay (2000);

lcd.clear();
```

```
Read_Sensor();

int total = S1+S2+S3+S4+S5+S6;
slot = slot-total;
}

void loop(){

Read_Sensor();

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

lcd.setCursor (0,1);
if(S1==1){lcd.print("S1:Fill ");}
    else{lcd.print("S1:Empty");}

lcd.setCursor (10,1);
if(S2==1){lcd.print("S2:Fill ");}
    else{lcd.print("S2:Empty");}

lcd.setCursor (0,2);
if(S3==1){lcd.print("S3:Fill ");}
    else{lcd.print("S3:Empty");}

lcd.setCursor (10,2);
if(S4==1){lcd.print("S4:Fill ");}
    else{lcd.print("S4:Empty");}
```

```
 lcd.setCursor (0,3);
if(S5==1){lcd.print("S5:Fill ");}
    else{lcd.print("S5:Empty");}

lcd.setCursor (10,3);
if(S6==1){lcd.print("S6:Fill ");}
    else{lcd.print("S6:Empty");}

if(digitalRead (ir_enter) == 0 && flag1==0){
if(slot>0){flag1=1;
if(flag2==0){myservo.write(180); slot = slot-1;}
}else{
lcd.setCursor (0,0);
lcd.print(" Sorry Parking Full ");
delay(1500);
}
}

if(digitalRead (ir_back) == 0 && flag2==0){flag2=1;
if(flag1==0){myservo.write(180); slot = slot+1;}
}

if(flag1==1 && flag2==1){
delay (1000);
myservo.write(90);
flag1=0, flag2=0;
}

delay(1);
}
```

```
void Read_Sensor(){
S1=0, S2=0, S3=0, S4=0, S5=0, S6=0;

if(digitalRead(ir_car1) == 0){S1=1;}
if(digitalRead(ir_car2) == 0){S2=1;}
if(digitalRead(ir_car3) == 0){S3=1;}
if(digitalRead(ir_car4) == 0){S4=1;}
if(digitalRead(ir_car5) == 0){S5=1;}
if(digitalRead(ir_car6) == 0){S6=1;}
}
```