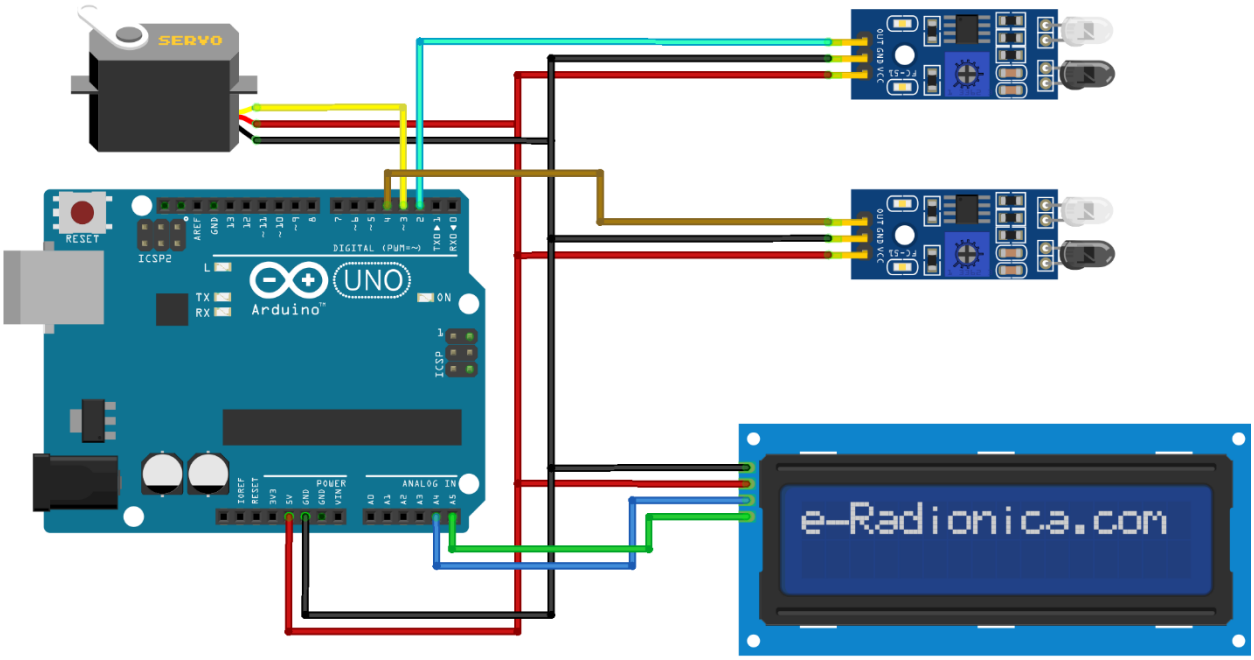
Title: Car Parking System.

Objective: Arduino car parking system.

Diagram:



Code:

```
#include <Arduino.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <Servo.h>


Servo myservo1;


// setting up the arduino pins

int IR1 = 4;

int IR2 = 2;
```

```
// total number of parking slots
int Slot = 4;

int flag1 = 0;
int flag2 = 0;

void setup() {
	lcd.init();
	lcd.backlight();

	pinMode(IR1, INPUT);
	pinMode(IR2, INPUT);

	// determining the pin of the servo motor
	myservo1.attach(3);
	myservo1.write(110);

	lcd.setCursor(0, 0);
	lcd.print("    ARDUINO    ");
	lcd.setCursor(0, 1);
	lcd.print(" PARKING SYSTEM ");
	delay(2000);
}

void loop()
{
```

```
// checking for car entry
if (digitalRead(IR1) == LOW && flag1 == 0)
{
        // checking whether there's empty slots left
        if (Slot > 0) {
                flag1 = 1;
                // decrease the slot when a car enters
                if (flag2 == 0) {
                        myservo1.write(40);
                        Slot = Slot - 1;
                }
        }
        else
        {
                lcd.setCursor(0, 0);
                lcd.print("    SORRY    ");
                lcd.setCursor(0, 1);
                lcd.print("parking is full!");
                delay(3000);
                lcd.clear();
        }
}



if (digitalRead(IR2) == LOW && flag2 == 0)
{
        flag2 = 1;
```

```
            // increase the slot when a car leaves
            if (flag1 == 0) {
                  myservo1.write(40);
                  Slot = Slot + 1;
            }
      }

      // resetting and shutting the door when the entry/exit of the car finishes
      if (flag1 == 1 && flag2 == 1) {
            delay(1000);
            myservo1.write(110);
            flag1 = 0, flag2 = 0;
      }

      // printing the number of slots available
      lcd.setCursor(0, 0);
      lcd.print("    WELCOME!    ");
      lcd.setCursor(0, 1);
      lcd.print("Slots Left:    ");
      lcd.print(Slot);
}
```

Result: After collecting all components according to the circuit diagram and uploading the code to the Arduino board. Now put the sensors and servo engine at accurate positions. There are four parking spaces in this project, IR sensor-1 and 2 are set at the passage and exit entryway separately and a servo engine is utilized to function the common single section and exit entryway. The LCD show is set near the section door. Within the starting, when all stopping openings are purge, at that point the LCD show appears, remaining way, this framework naturally permits 4 vehicles. In case the stopping is full, the framework blocked the entrance entryway by closing the servo boundary. And the Driven show appears that all spaces are full. When a vehicle clears out a space and arrives at the entryway of the stopping range at that

point the IR sensor-2 recognizes it. the stopping region at that point the IR sensor-1 identifies the vehicle and the framework is permitted to enter that vehicle by opening the servo barrier. After entering the stopping range when that vehicle involves a space at that point the Driven show appears that the opening is full. In this, That vehicle and the framework open the servo obstruction. At that point the Driven show appears that the opening is empty. Again the system will permit entering a modern vehicle. There's a denial of entering and clearing out two cars at a time.

CONCLUSION

Our project detects the empty slots and helps the drivers to find parking spaces in new city. The average waiting time of users for parking their vehicles is viably reduced in this system test results show that the execution of the Arduino UNO-based framework can viably fulfill wants of existing car parking hassles thereby minimizing the time expended to discover a empty stopping part and real-time data rendering this keen stopping framework gives betta er execution and low cost.