IOT LAB 1

Name: M. VARUN REDDY

Date: 3rd May 2023

Reg no.: 21BRS1507

Car Parking System using Arduino and Ultrasonic Distance Sensor

Aim: To perform and analysis of smart car parking system with Arduino in TinkerCad and using Arduino hardware

BLOCKS/COMPONENTS REQUIRED:-

Tinker CAD Block Modules
1. Arduino UNO
2. IR Sensor
3. Buzzer (OPTIONAL)

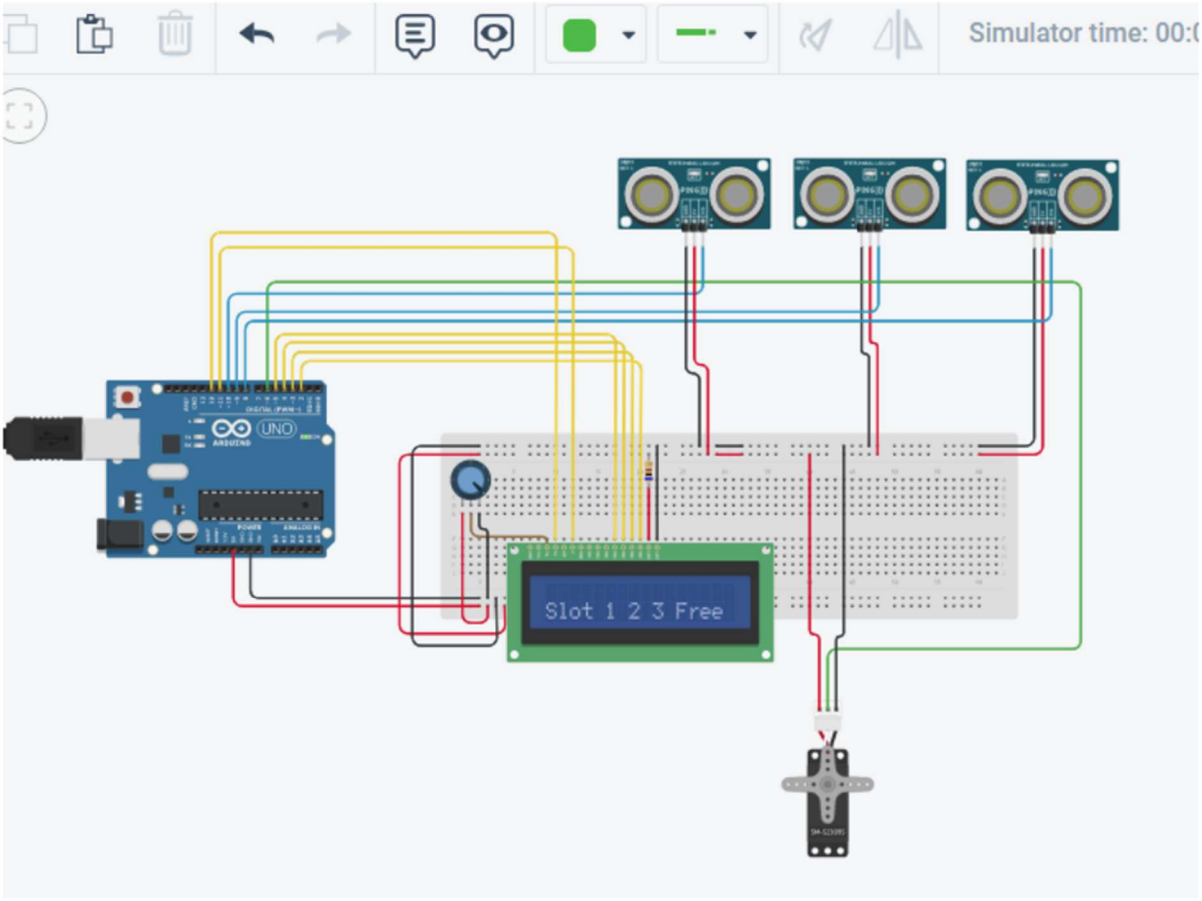
Hardware:
1. Arduino UNO
2. IR Sensor
3. Buzzer

TinkerCad Procedure:

Step-1: Open TinkerCAD.
Step-2: In the Design Suite, Drag and drop the required blocks and its associated components into their respective places.
Step-3: Connect as per the circuit diagram given in the block diagram. Follow the image and place each part as shown. Notice that the breadboard is used to connect the components together. It contains a series of columns and two rails on each side. As you click from pin to pin, you'll create wires, each of which can be color-coded for easy identification.
Step-4: In code section, appropriate Arduino codes are fed. Create the next block, an "if" statement, which is a type of control block that makes a decision.
Step-5: Once you're done, select "Start Simulation" on the toolbar to turn on your Arduino Uno.

Step-6: Troubleshooting: If your program doesn't behave as expected, check your wiring and programming. Ensure that all pins are properly connected and that each block is written correctly.

Block Diagram:



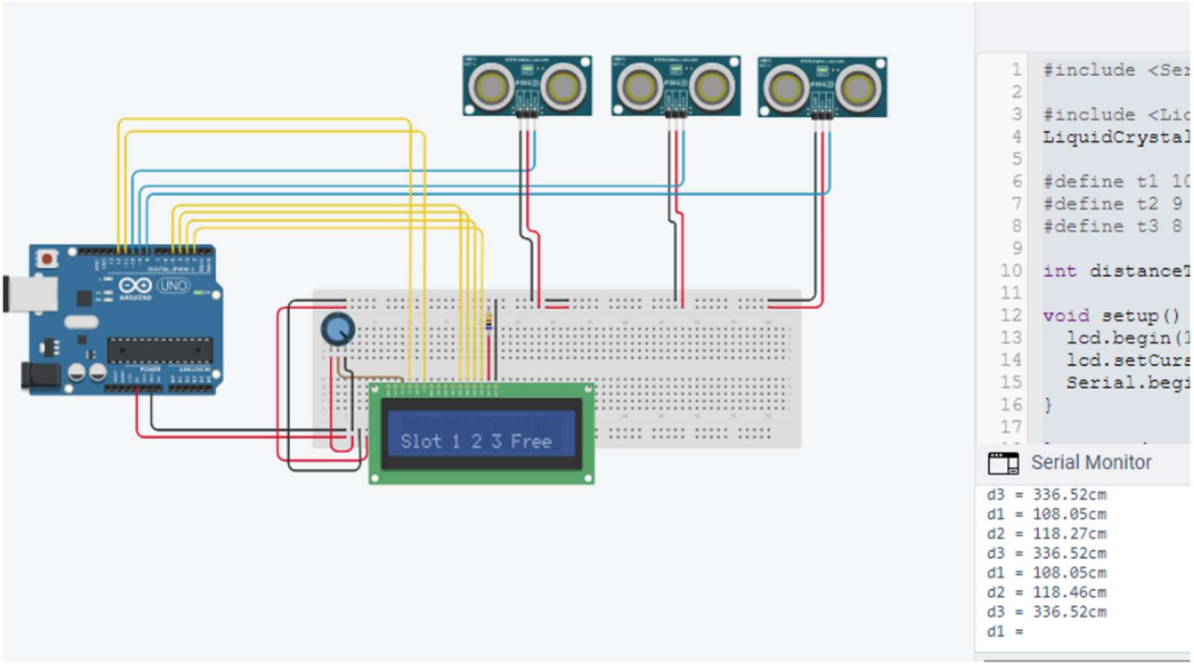
TinkerCad code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define t1 10
#define t2 9
#define t3 8

int distanceThreshold = 100;

void setup() {
lcd.begin(16,2);
lcd.setCursor(0,0);
```

```
Serial.begin (9600);
}

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);

Serial.println("d1 = " + String(d1) + "cm");
Serial.println("d2 = " + String(d2) + "cm");

Serial.println("d3 = " + String(d3) + "cm");

if (d1>100 & d2>100 & d3>100){
lcd.setCursor(0,0);
lcd.print("3 Slots Free");
lcd.setCursor(0,1);
lcd.print("Slot 1 2 3 Free");
delay(500);
}
else if((d1>100 & d2>100)|(d2>100 & d3>100)|(d3>100 & d1>100))
{
lcd.setCursor(0,0);
lcd.print("2 Slots Free");
lcd.setCursor(0,1);
if(d1>100 & d2>100)
lcd.print("Slot 1 & 2 Free");
else if(d1>100 & d3>100)
```

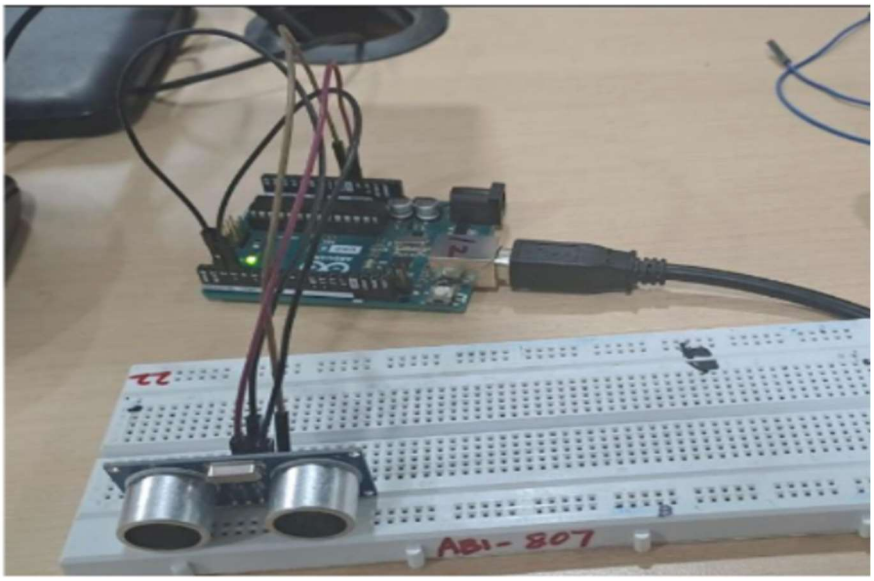
```
lcd.print("Slot 1 & 3 Free");
else
lcd.print("Slot 2 & 3 Free");
delay(500);
}
else if(d1<100 & d2<100 & d3<100)
{

lcd.setCursor(0,0);
lcd.print("No Slot Free");
lcd.setCursor(0,1);
lcd.print("Parking Full");
delay(500);
}
else if((d1<100 & d2<100)|(d2<100 & d3<100)|(d3<100 & d1<100))
{
lcd.setCursor(0,0);
lcd.print("1 Slot Free");
lcd.setCursor(0,1);
if(d1>100)
lcd.print("Slot 1 is Free");
else if (d2>100)
lcd.print("Slot 2 is Free");
Else
lcd.print("Slot 3 is Free");
delay(500);
}

delay(100);
}
```

Output:



Hardware:

Connections:



Arduino Program:

(Without Libraries)

```
const int trigPin = 2;
const int echoPin = 3;
const int buzzer = 6;

// defines variables
long duration;
int distance;
int safetyDistance;
void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(buzzer, OUTPUT);
safetyDistance = ultrasonic();
Serial.begin(9600); // Starts the serial communication
}

void loop() {
int dist = ultrasonic();
if (dist <= safetyDistance){
tone(buzzer, 2500);
delay(500);
tone (buzzer, 2500);
}
else{
noTone (buzzer);
}
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(dist);

}
int ultrasonic(){
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
```

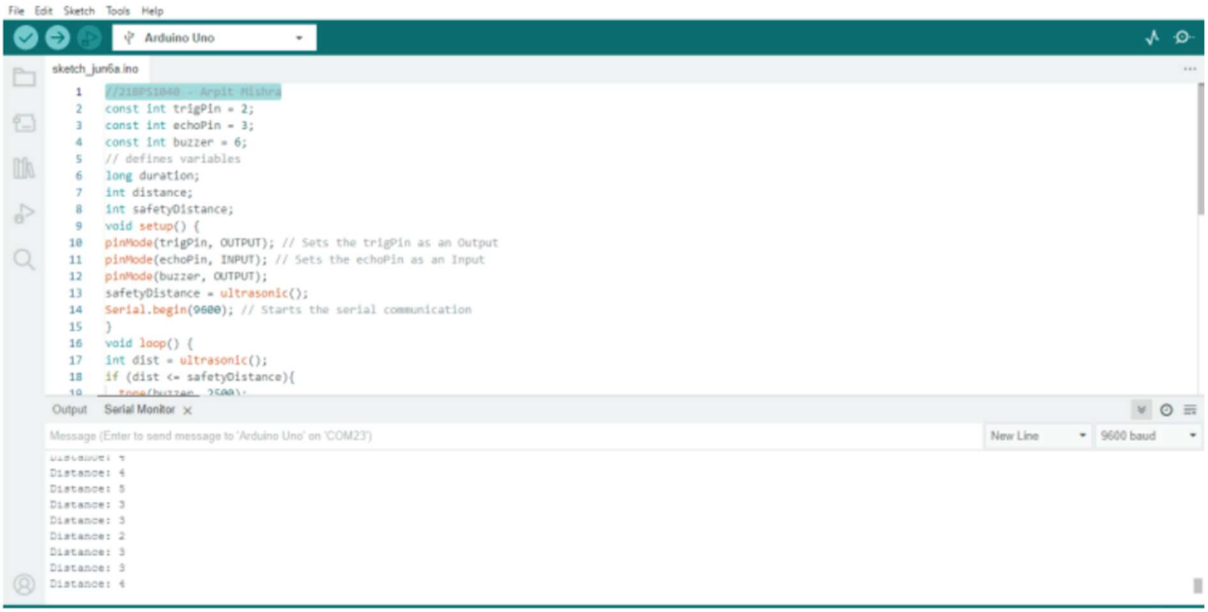
```
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
return distance;
}
```

Output:

```
(With Libraries)
#include <HCSR04.h>
HCSR04 hc1(5, 6);
//initialisation class HCSR04 (trig pin , echo pin)
HCSR04 hc2(9,10);
void setup()
{
Serial.begin(9600);
pinMode(11,OUTPUT);
pinMode(12,OUTPUT);
pinMode(13,OUTPUT);
}
void loop() {
Float dist1,dist2;
Serial.println(hc1.dist());
// return curent distance in serial
delay(200);
Serial.println(hc2.dist());
delay(200);
dist1=hc1.dist();
dist2=hc2.dist();
if (dist1<30){
digitalWrite(11,HIGH);
Serial.println("Parking Slot 1 is full");

digitalWrite(13,HIGH);
}
Else
{
digitalWrite(11,LOW);
digitalWrite(13,LOW);
```

```
}
if (dist2<30){
digitalWrite(12,HIGH);
Serial.println("Parking Slot 2 is full");
digitalWrite(13,HIGH);
}
Else
{
digitalWrite(12,LOW);
digitalWrite(13,LOW);
}
}
```

OUTPUT:-



INFERENCE : We implemented a smart car parking system by using an Arduino and a buzzer. When we place our hand near the sensor, the distance between the sensor and our hand changes, causing the buzzer to make noise. By running the program, the change in distance can be viewed on the Arduino serial monitor.

RESULT: We have successfully performed the experiment using Arduino in TinkerCad and Arduino hardware.