Practical 1

MIni Project:

IOT BASED SMART PARKING SYSTEM

Team Members**:**

1. Akash Bhutka (12102080501002)
2. Dhruv Patel ((12102080501014)
3. Dipam Shah ((12102080501016)
4. Setul Patel ((12102080501065)

Purpose:

Finding a car parking space is a big issue in congested cities. There are too many vehicles on the road but not enough parking spaces. One of the biggest problems is when we enter a parking area then we realize that there are no empty parking slots to park our cars. Another problem is after entering in a parking area we confused to find the empty parking slot to park our car. We all had faced these problems that wasted our time.

That's why we need efficient parking management systems in all parking areas that will provide confusion-free and easy parking.

Hardware Required:

- ➢ INFRARED SENSORS
- ➢ ARDUINO UNO
- ➢ BREADBOARD
- ➢ NODE MCU(ESP8266)
- ➢ POWER SUPPLY
- ➢ CONNECTING WIRES

Working:

The IR sensor at the entrance detects a coming car and will automatically showcases that there is a free slot in the parking lot but if the lot is full, then it'll show empty space.

When a vehicle leaves a slot and arrives at the gate of the parking area then the IR sensor-2 detects that vehicle and the system open the servo barrier. Then it shows that the slot is empty. Again the system will allow entering a new vehicle.

IR Sensor will transmit the data to Arduino Board and further it will pass the data to USER through NodeMCU.

Arduino Code:

```
#include<Servo.h>
#include<SoftwareSerial.h>
SoftwareSerial esp82(6, 9);
Servo entryservo;
Servo exitservo;
#define IR1 2
#define IR2 4
#define IR3 7
#define IR4 8
#define IR5 12
#define IR6 13

bool ir1;
bool ir2;
bool ir3;
bool ir4;
bool ir5;
bool ir6;
int pos;
void setup() {

Serial.begin(9600);
esp82.begin(115200);
entryservo.attach(3);
exitservo.attach(5);
pinMode(IR1, INPUT);
pinMode(IR2, INPUT);
pinMode(IR3, INPUT);
pinMode(IR4, INPUT);
pinMode(IR5, INPUT);
pinMode(IR6, INPUT);
}
void loop() {
ir1 = digitalRead(IR1);
if(ir1 == 0)
{
 Serial.println("Object is present at 1");
 esp82.println("Object is present at 1");
}
else if (ir1 == 1)
{
 Serial.println("Empty");
 esp82.println("Empty");
}
```

```
ir2 = digitalRead(IR2);
if(ir2 == 0)
{
 Serial.println("Object is present at 2");
 esp82.println("Object is present at 2");
}
else if (ir2 == 1)
{
 Serial.println("Empty");
 esp82.println("Empty");
}
ir3 = digitalRead(IR3);
if(ir3 == 0)
{
 Serial.println("Object is present at 3");
 esp82.println("Object is present at 3");
}
else if (ir3 == 1)
{
 Serial.println("Empty");
 esp82.println("Empty");


}
ir4 = digitalRead(IR4);
if(ir4 == 0)
{
 Serial.println("Object is present at 4");
 esp82.println("Object is present at 4");

}
else if (ir4 == 1)
{
 Serial.println("Empty");
 esp82.println("Empty");
}
ir5 = digitalRead(IR5);
if(ir5 == 0)
{
 Serial.println("Object is present at Entry");

 for(pos=0;pos<=90;pos++){
 entryservo.write(pos);
 delay(15);
 }
 delay(2000);
```

```
 for(pos=90;pos>=0;pos--){
 entryservo.write(pos);
 delay(15);
 }
 }

else if (ir5 == 1)
{
 Serial.println("Empty");

}
ir6 = digitalRead(IR6);
if(ir6 == 0)
{
 Serial.println("Object is present at exit");

  for(pos=0;pos<=90;pos++){
 exitservo.write(pos);
 delay(15);
 }
 delay(2000);
 for(pos=90     ;pos>0;pos--){
 exitservo.write(pos);
 delay(15);
 }
}
else if (ir6 == 1)
{
 Serial.println("Empty");

}
}
```

ESP code:

```
#define BLYNK_TEMPLATE_ID "TMPL3QR0cpd8x"
#define BLYNK_TEMPLATE_NAME "JAAYY"
#define BLYNK_AUTH_TOKEN "DTah7G6K3YPAetCA-ISzxJsBm_cpxuBq"

#include<ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include<SoftwareSerial.h>
SoftwareSerial ard(5, 4);
#define BLYNK_PRINT Serial
WidgetLED led1(V0);
WidgetLED led2(V1);
WidgetLED led3(V2);
WidgetLED led4(V3);
WidgetLED led5(V4);
WidgetLED led6(V5);
BlynkTimer timer;
String str1;
char ssid[] = "HAARY";
char pass[] = "dhruvharry1998";
void sensor1(){
  String str1 = ard.readStringUntil('\n');
  str1.trim();
  Serial.println(str1);
  Serial.println(str1.length());
  if(str1=="Object is present at 1"){
  led1.on();
  }
  else if(str1=="Empty"){
  led1.off();
  }
}
 void sensor2(){
  String str2 = ard.readStringUntil('\n');
  str2.trim();
  Serial.println(str2);
  Serial.println(str2.length());
  if(str2=="Object is present at 2"){
  led2.on();
  }
  else if(str2=="Empty"){
  led2.off();
  }
}
```

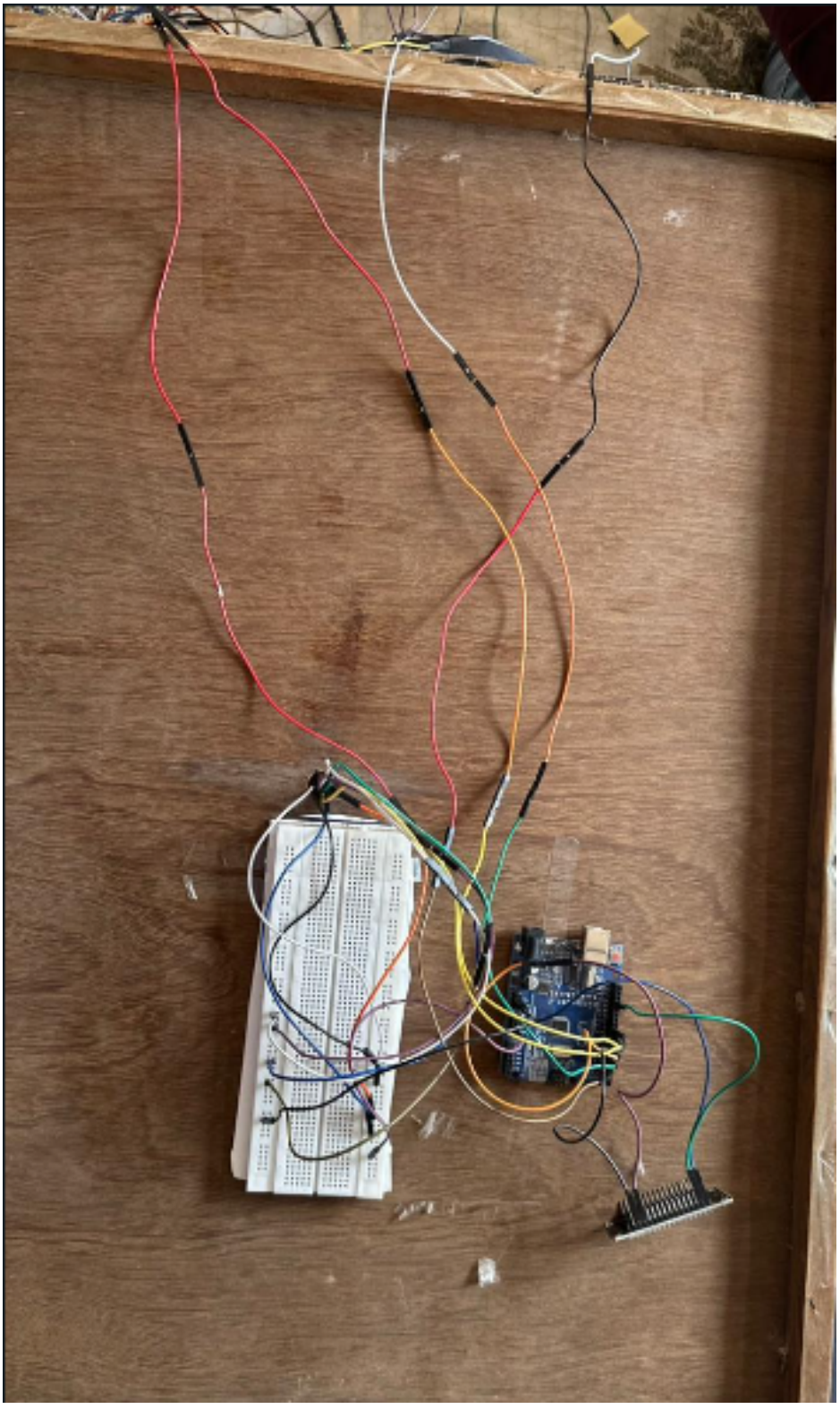
```
void sensor3(){
  String str3 = ard.readStringUntil('\n');
  str3.trim();
  Serial.println(str3);
  Serial.println(str3.length());
  if(str3=="Object is present at 3"){
  led3.on();
  }
  else if(str3=="Empty"){
  led3.off();
  }
}
void sensor4(){
  String str4 = ard.readStringUntil('\n');
  str4.trim();
  Serial.println(str4);
  Serial.println(str4.length());
  if(str4=="Object is present at 4"){
  led4.on();
  }
  else if(str4=="Empty"){
  led4.off();
  }
}
void setup() {
  Serial.begin(9600);
  ard.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(300L,sensor1);
  timer.setInterval(300L,sensor2);
  timer.setInterval(300L,sensor3);
  timer.setInterval(300L,sensor4);

}

void loop() {

  Blynk.run();
  if(ard.available()){
  timer.run();}
}
```

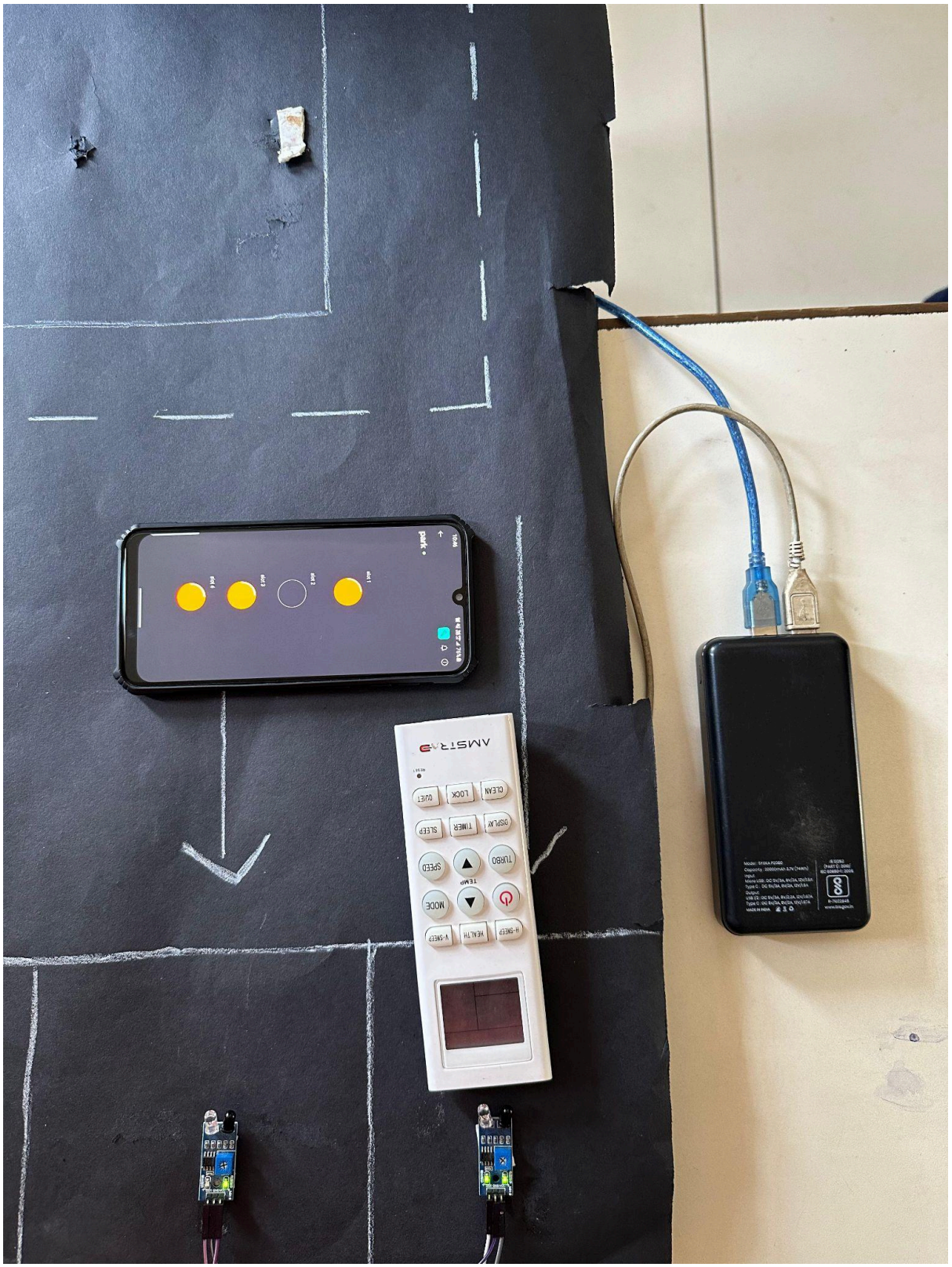
Connections:



Connection between all required hardware equipment.

Figure(i)

Output:



Object detected at parking slots 1,3 and 4 i.e. slot 2 is free

Figure(i)

Object detected at parking slots 2,3 and 4 i.e. slot 1 is free.

Figure(ii)