```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
#include <SoftwareSerial.h>
#include <Servo.h>
Servo dropServo;
#define RX 9
#define TX 10
int IR_1 = A0;
int IR_2 = A1;
int IR_3 = A2;
int IR_4 = A3;
int counter=0;
int currentState=0;
int previousState_1=0;
int previousState_2=0;
int IR_1_op;
int IR_2_op;
int IR_3_op;
int IR_4_op;
String AP = "elegant";       // AP NAME
String PASS = "smartwork"; // AP PASSWORD
String API = "ACHINOFHIZ60Y28R";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
```

```
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;

SoftwareSerial esp8266(RX,TX);
void setup()
{
  lcd.begin(16, 2);
  lcd.print ("  SMART PARKING ");
lcd.setCursor (0,1);
lcd.print (" SLOT USING IOT");
delay(2000);
dropServo.attach(11);
pinMode(IR_1,INPUT);
pinMode(IR_2,INPUT);
pinMode(IR_3,INPUT);
pinMode(IR_4,INPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS
+"\"",20,"OK");
```

```
}
void loop() {
 String getData = "GET /update?api_key="+ API
+"&field1="+get_slots();
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+
PORT,15,"OK");
 sendCommand("AT+CIPSEND=0,"
+String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
}

String get_slots()
{
IR_1_op = digitalRead(IR_1);
IR_2_op = digitalRead(IR_2);
IR_3_op = digitalRead(IR_3);
IR_4_op = digitalRead(IR_4);
Serial.println(digitalRead(IR_2));
if (IR_1_op == LOW && IR_2_op == LOW&&IR_3_op == LOW)
{
 lcd.clear();
 lcd.print ("NO SLOTS AVAIABLE:");
 delay(500);
```

```
 return String(0);
}
else if (IR_1_op == HIGH && IR_2_op == HIGH && IR_3_op == LOW)
{
 lcd.clear();
 lcd.print ("2 SLOTS AVAIABLE:");
 delay(500);
 return String(2);
}
else if (IR_1_op == LOW && IR_2_op == HIGH && IR_3_op == HIGH)
{
 lcd.clear();
 lcd.print ("2 SLOTS AVAIABLE:");
 delay(500);
 return String(2);
}
else if (IR_1_op == HIGH && IR_2_op == LOW && IR_3_op == HIGH)
{
 lcd.clear();
 lcd.print ("2 SLOTS AVAIABLE:");
 delay(500);
 return String(2);
```

```
}
else if (IR_1_op == HIGH && IR_2_op == LOW && IR_3_op == LOW)
{
 lcd.clear();
 lcd.print ("1 SLOTS AVAIABLE:");
 delay(500);
 return String(1);
}
else if (IR_1_op == LOW && IR_2_op == HIGH && IR_3_op == LOW)
{
 lcd.clear();
 lcd.print ("1 SLOTS AVAIABLE:");
 delay(500);
 return String(1);
  }
else if (IR_1_op == LOW && IR_2_op == LOW && IR_3_op == HIGH)
{
 lcd.clear();
 lcd.print ("1 SLOTS AVAIABLE:");
 delay(500);
 return String(1);
}
```

```
else if (IR_1_op == HIGH && IR_2_op == HIGH && IR_3_op == HIGH)
{
  lcd.clear();
  lcd.print ("3 SLOTS AVAIABLE:");
  delay(500);
  return String(3);
}
}
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }
```

```
  countTimeCommand++;
 }
 if(found == true)
 {
  Serial.println("OYI");
  countTrueCommand++;
  countTimeCommand = 0;
 }
 if(found == false)
 {
  Serial.println("Fail");
  countTrueCommand = 0;
  countTimeCommand = 0;
 }
 found = false;
}
```