DAFTAR PUSTAKA

Akmal, S., Ritzkal, dan Hendrawan, A. H., 2018, Implementasi Sistem Ketersediaan Slot Parkir Otomatis Perbantuan Mikrokontroler Menggunakan Arduino Uno R3, *Seminar Nasional Teknologi Informasi Universitas Ibn Khaldun Bogor*, Bogor.

Andang, A., Hiron, N., Chobir, A., dan Busaeri, N., 2019, Investigation of Ultrasonic Sensor Type JSN-SR04T Performance as Flood Elevation Detection, *IOP Publishing*, Vol. 50, No. 5, hal. 1-7.

Anhar, 2010, *Panduan Menguasai PHP & MySQL Secara Otodidak*, MediaKita, Jakarta.

Baharuddin, Fibriasari, H., dan Rajagukguk, J., 2021, Model by Doing in Learning official Volley Ball Games, *Proceedings of the 3rd International Conference on Innovation in Education, Science, and Culture*, Medan.

Barrett, S. F., 2013, *Arduino Microcontroller Processing for Everyone!*, Edisi Ketiga, Morgan & Claypool Publishers, Williston.

Bright, S., 2016, *JavaScript Fundamentals: JavaScript Syntax, What JavaScript is Use for in Website Development, JavaScript Variable, Strings, Popup Boxes, JavaScript Objects, Function, and Event Handlers*, Lulu Press, Morrisville.

Berchmans, D., dan Kumar, S. S., 2014, Optical Character Recognition: An Overview and an Insight, 2014 International Conference on Control, Instrumentation, *Communication and Computational Technologies (ICCICCT)*, Kanyakumari.

Chaudhuri, B. B., 2007, *Digital Document Processing*, Springer, Singapur.

Dharmawan, H. A., 2017, *Mikrokontroler Konsep Dasar dan Praktis*, UB Press, Malang.

Fauzan, M. N., dan Adiputri, L. C., 2019, *Tutorial Membuat Prototipe Prediksi Ketinggian Air (PKA) untuk Pendeteksi Banjir Peringatan Dini Berbasis IoT*, Kreatif Industri Nusantara, Bandung.

Fraden, J., 2010, *Handbook of Modern Sensors: Physics, Design, and Applications*, Edisi Keempat, Springer, New York.

Fogli, D., Tetteroo, D., Barricelli, B. R., Borsci, S., Markopoulos, P., dan Papadopoulos, G. A., 2021, Empowering Stakeholders with Artificial Intelligence, Meta-Design, and Cultures of Participation, *International Symposium on End-User Development*, Limassol.

Greengard, S., 2015, *The Internet of Things*, MIT Press, Cambridge.

Hakim, L., 2018, *Bahasa Pemrograman (C# dan EmguCV)*, Deepublish, Yogyakarta.

Hardjono, B., Widjaja, A. E., Rhizma, M. G. A., Tjahyadi, H., Haryani, C. A., dan Renata, W., 2020, *Komunikasi Nirkabel dengan Aplikasinya di Bidang Telekomunikasi dan Informatika*, Andi, Yogyakarta.

Heri, Irawansyah, Noviyantono, E., dan Prayogi, D., 2017, Perancangan Perangkat Pengelolaan Parkir Otomatis dengan Pemanfaatan Kamera, *Journal of Applied Microcontrollers and Autonomous System*, Vol.3, No.1, hal 21-26.

Ibadillah, A. F., dan Alfita, R., 2021, *Mikrokontroler dan Aplikasinya*, MNC Publishing, Malang.

Imran, M. A., Abbasi, Q. H., Onireti, O., dan Ansari, S., 2021, *Autonomous Airborne Wireless Networks*, Wiley, Chennai.

Kadir, A., 2017, *Pemrograman Arduino dan Processing*, Elex Media Komputindo, Jakarta.

Kadir, A., 2019, *Langkah Mudah Pemrograman OpenCV & Python*, Elex Media Komputindo, Jakarta.

Kurniawan, R., dan Zulius, A., 2018, Sistem Smart Parking Menggunakan Ultrasonik Sensor, *JUSIKOM*, Vol.3, No.1, hal 24-31.

Marko, D., dan Hurby, D., 2020, Distance measuring in vineyard row using ultrasonic and optical sensors, *International Conference of Young Scientists*, Prague Czech Republic.

Nadeem, D., dan Rizvi, S., 2015, Character recognition using template matching. *Project report Department of Computer Science*, Jamia Millia Islamia, New Delhi.

Noer, Z., dan Dayana, I., 2021, *Buku Sistem Kontrol*, Guepedia, Jakarta.

Nugroho, A., Susilo, K. E., Winardi, S., dan Budijanto, A., 2020, *Buku Petunjuk Praktikum Mikrokontroler Arduino*, Scopindo Media Pustaka, Surabaya.

Pambudi, W. S., 2021, *Aplikasi Akuisisi Data Sensor dengan InstrumenLab, PlotLab, Chart pada Arduino Uno*, Andi, Yogyakarta.

Rizal, M., 2020, *Pengukuran Teknik Dasar dan Aplikasi*, Syiah Kuala University Press, Aceh.

Sakti, S. P., 2017, *Pengantar Teknologi Sensor: Prinsip Dasar Sensor Besaran Mekanik*, UB Press, Malang.

Santoso, H., 2015, *Panduan Praktis Arduino untuk Pemula*, Elangsakti, Malang.

Satria, E., 2017, *Modul Elektronika dan Mekatronika Motor Servo*, Direktorat Pembinaan Sekolah Menengah Kejuruan, Jakarta.

Sendari, S., Wirawan, I. M., dan Nasrulloh, M., 2021, *Sensor Transduser*, Ahlimedia Press, Malang.

Setiawardhana, Oktavianto, H., Wasista, S., dan Susanto, E., 2021, *14 Jam Belajar Cepat Internet of Things (IoT)*, Deepublish, Yogyakarta.

Shi, F., Chen, J. I. Z., dan Tavares, J. M. R. S., 2022, *Third International Conference on Image Processing and Capsule Networks*, Springer, Switzerland.

Siahaan, V., dan Sianpar, R., H., 2019, *Python GUI with MySQL: A Step By Step Guide to Database Programming*, SPARTA Publishing, Medan.

Umam, F., Budiarto, H., dan Dafid, 2021, *Motor Listrik*, MNC Publishing, Malang.

Umami, R., 2017, Penerapan Hand Motion Tracking Pengendali Pointer Pada Virtual Mouse Dengan Metode Optical Flow, *Skripsi*, Politeknik Negeri Sriwijaya, Palembang.

Utami, E., dan Sukrisno, 2005, *10 Langkah Belajar Logika dan Algoritma, Menggunakan Bahasa C dan C++ di GNU / Linux*, Andi, Yogyakarta.

Wasista, S., Setiawardhana, Saraswati D. A., dan Susanto E., 2019, *Aplikasi Internet of Things (IoT) Dengan Arduino Dan Android "Membangun Smart*

Home Dan Smart Robot Berbasis Arduino Dan Android", Deepublish, Yogyakarta.

Widodo, B., dan Almasri, 2021, Rancang Sistem Informasi Parkir Otomatis dengan Menentukan Posisi Parkir Berbasis Telegram Menggunakan Arduino Mega2560, *Jurnal Pendidikan Tambusai*, Vol.5, No.3, hal 7055-7074.

Yudaningtyas, E., 2017, *Belajar Sistem Kontrol Soal & Pembahasan*, UB Press, Malang.

Ardiyansyah, R., 2022, Bot Telegram, *https://loop.co.id/bot-telegram/*, diakses Maret 2022.

Arducore webpage, Arduino Mega CH340, *https://www.arducore.com.br/arduino-mega-ch340-cabo-usb*, diakses Desember 2022.

Badan Pusat Statistik, 2021, Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, *https://www.bps.go.id/linkTableDinamis/view/id/1133*, diakses Maret 2022.

Bakker B. D., 2019, Waterproof JSN-SR04T Ultrasonik Distance Sensor with Arduino Tutorial. *https://www.makerguides.com/jsn-sr04t-arduino-tutorial/*, diakses Maret 2022.

Datasheet, 2020, Datasheet JSN-SR04T, *https://www.openbikesensor.org/*, diakses Maret 2022.

Digiware, 2019, Memilih Sensor Ultrasonik dari MaxBotix untuk Berbagai Kebutuhan, *https://digiwarestore.com/id/digiware-news/57_Memilih-Sensor-Ultrasonik-Dari-MaxBotix-untuk*, diakses Mei 2022.

Solawetz, J., 2021, What is Mean Average Precision (mAP) in Object Detection?, *https://blog.roboflow.com/mean-average-precision/*, diakses Desember 2022.

LAMPIRAN-LAMPIRAN

Lampiran A. Data Pengujian

A. 1 Data karakterisasi sensor ultrasonik

Jarak pada Alat Ukur (cm)	Tegangan Keluaran (mV)					Rata-rata (mV)
	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5	
10	54,5	54,4	53,6	53,6	54,1	54,04
20	96,4	95,5	93,8	95,7	96	95,48
30	136,5	136,8	137,9	138,1	140,3	137,92
40	182,3	185,2	183,1	182,5	180,8	182,78
50	221,5	218,7	219,4	220,6	219,2	219,88
60	268,8	267,8	271,7	269,7	270,5	269,7
70	326,3	325,8	325,7	327	326,6	326,28
80	393,7	396,8	389,6	392,1	393	393,04
90	433,8	436,3	431,5	438,4	434,5	434,9
100	488,9	491,5	488,3	489,4	487,2	489,06
110	518,7	520,6	521,5	523	519,6	520,68
120	576,8	577,4	581,4	583,7	582,7	580,4
130	619,4	620,2	623,4	622,6	618,3	620,78
140	668,6	671,2	670,4	672	671,1	670,66
150	719,8	722,5	720,9	723,6	717,4	720,84
160	787,9	786,8	792,3	793,5	790,8	790,26
170	824,1	819,6	816,7	821,2	822,5	820,82
180	887,7	892,4	891,7	888,4	890,6	890,16
190	927,3	931,6	930	929,5	932,1	930,1
200	1001,9	999,7	998,6	1002,2	998,9	1000,26
210	1050,2	1051,5	1052,4	1048,6	1049,7	1050,48
220	1103,6	1108,5	1105,6	1107,4	1102,5	1105,52
230	1157,8	1162,6	1161,8	1160,7	1159,9	1160,56
240	1197,6	1199,2	1201,6	1203,5	1200,8	1200,54
250	1251,1	1249,7	1248,6	1250,7	1252,5	1250,52
260	1289,3	1290,3	1288,7	1293,2	1291,5	1290,6
270	1340,9	1341,6	1340	1339,7	1338,5	1340,14
280	1389,6	1390,4	1394,4	1387,8	1392,2	1390,88
290	1454,2	1455,3	1456,7	1453,9	1456,8	1455,38
300	1501,6	1499,5	1498,6	1502,9	1500,7	1500,66
310	1559,7	1560,4	1558,6	1561,6	1562,5	1560,56

Jarak pada Alat Ukur (cm)	Tegangan Keluaran (mV)					Rata-rata (mV)
	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5	
320	1601,1	1603,2	1599,4	1598,6	1600,7	1600,6
330	1651,7	1653,7	1652,6	1649,5	1647,4	1650,98
340	1711,4	1709,7	1708,5	1710,3	1713,2	1710,62
350	1761,6	1764,6	1767,8	1765,9	1766,5	1765,28
360	1809,8	1811,6	1810,3	1808,7	1812,4	1810,56
370	1890	1893	1889,5	1887,4	1891,7	1890,32
380	1927,5	1925,7	1924,8	1926,4	1923,8	1925,64
390	1978,1	1981,3	1982,4	1980,6	1979,8	1980,44
400	2036,7	2034,6	2035,1	2037,9	2033,8	2035,62
410	2098,5	2097,1	2093,5	2095,7	2091,4	2095,24
420	2138,9	2134,5	2135	2133,6	2136,2	2135,64
430	2185,6	2187,5	2182,1	2183,7	2187,6	2185,3
440	2240,7	2237,6	2238,5	2241,3	2242,4	2240,1
450	2289,7	2286,8	2291,4	2290,6	2292,2	2290,14
460	2364,5	2365	2366,8	2367,4	2363,3	2365,4
470	2511,4	2504,7	2503,3	2510,5	2502,6	2506,5
480	2625,2	2627,4	2624,8	2626,1	2628,5	2626,4
490	2696,1	2697,8	2698,2	2701,3	2699,7	2698,62
500	2737,5	2733,5	2738,8	2735	2736,9	2736,34
510	2788,4	2782,4	2783,7	2786,8	2785,6	2785,38
520	2807,9	2808,8	2803,4	2805,6	2809,6	2807,06
530	2874,2	2872,4	2873,9	2875,7	2871,6	2873,56
540	2939	2938,4	2935,3	2936,6	2940,4	2937,94
550	2962,5	2964,3	2966,7	2965,4	2963,8	2964,54
560	3093,2	3096,3	3095,4	3094,3	3097,5	3095,34
570	3138,6	3137,4	3139,1	3141,5	3133,1	3137,94
580	3235,8	3236,4	3232,5	3233,7	3234,9	3234,66
590	3346,8	3344	3348,3	3345,7	3347,5	3346,46
600	3533,2	3536,2	3538,2	3538,2	3537,2	3536,6

Lampiran B. *List* Program

B. 1 Program karakterisaasi sensor ultrasonik JSN-SR04T

```
int trigPin = 22;     // Trigger
int echoPin = 24;     // Echo
long duration, cm;


void setup() {

  //Serial Port begin
  Serial.begin (9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}


void loop() {

  // The sensor is triggered by a HIGH pulse of 10 or
more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean
HIGH pulse:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the
sending
  // of the ping to the reception of its echo off of an
object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);


  // Convert the time into a distance
  cm = (duration/2) / 29.1;   // Divide by 29.1 or
multiply by 0.0343

  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(250);
}
```

B. 2 Program pengujian motor servo

```
#include <Servo.h> //library servo
Servo myservo; // membuat variabel servo untuk dikendalikan
int pos = 0; // deklarasi variabel untuk posisi sudut


void setup() {
myservo.attach(6); //deklarasi servo pada pin 3
}

void loop() {
for(pos = 0; pos < 90; pos+= 1) // perulangan untuk posisi 0 sampai 180 derajat

  { // step setiap 1 derajat
   myservo.write(pos); // memerintahkan servo ke posisi derajat sesuai nilai variabel pos
   delay(50); // menunggu 15 milidetik
   }
}
```

B. 3 Program ESP32-CAM

Tab 1- Telegram Bot and Wi-Fi

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "***";// wifi name
const char* password = "****";// wifi password
String BOTtoken = "5508343588:****";//token bot telegram
String CHAT_ID = "13978340**";


WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;
```

```
bool sendPhoto1 = false;

int botRequestDelay = 1000; //Cek pesanan telegram
setiap 1 detik.
unsigned long lastTimeBotRan;

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  pinMode(FLASH_LED_PIN, OUTPUT); // Set lampu flash
  digitalWrite(FLASH_LED_PIN, flashState);

  // Config and init the camera
  configInitCamera();

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);    //
Add root certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
```

```
  Serial.println(WiFi.localIP());
  pinMode(2, OUTPUT);
  pinMode(14, OUTPUT);
  pinMode(15, OUTPUT);
}

void loop() {
  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int                  numNewMessages                    =
bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages                                         =
bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }

}
```

Tab 2- Camera

```
void configInitCamera(){
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
```

```
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;  //0-63 lower number means
higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;  //0-63 lower number means
higher quality
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x",
err);
    delay(1000);
    ESP.restart();
  }

  // Drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s,     FRAMESIZE_CIF);           //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}
```

Tab 3- Telegram Menu

```
void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID) {
      bot.sendMessage(chat_id,   "Unauthorized   user",
"");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Hi , " + from_name + "\n" ;
      welcome += "Selamat datang di SLOT PARKIR";
```

```
      welcome += ", gunakan perintah berikut untuk info
lebih lanjut \n";
      welcome += "/kondisi : toggles flash LED \n";
      bot.sendMessage(CHAT_ID, welcome, "");
    }

    if (text == "/kondisi") {
      String kon = "kondisi slot parkir saat ini \n";
      int tom1 = digitalRead(2);
      int tom2 = digitalRead(14);
      int tom3 = digitalRead(15);
      if (tom1 == HIGH) {
        kon += "kon 1= terisi\n";
      }
      else {
        kon += "kon 1= tersedia\n";
      }
      if (tom2 == HIGH) {
        kon += "kon 2= terisi\n";
      }
      else {
        kon += "kon 2= tersedia\n";
      }
      if (tom3 == HIGH) {
        kon += "kon 3= terisi\n";
      }
      else {
        kon += "kon 3= tersedia\n";
      }

      bot.sendMessage(CHAT_ID, kon, "");
      Serial.println("New photo request");
    }
  }
}
```

Tab 4- Connection

```
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
```

```
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));


  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--kotakpaket\r\nContent-Disposition:
form-data;  name=\"chat_id\";  \r\n\r\n"  +  CHAT_ID  +
"\r\n--kotakpaket\r\nContent-Disposition:    form-data;
name=\"photo\";  filename=\"esp32-cam.jpg\"\r\nContent-
Type: image/jpeg\r\n\r\n";
    String tail = "\r\n--kotakpaket--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    clientTCP.println("POST  /bot"+BOTtoken+"/sendPhoto
HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length:        "        +
String(totalLen));
    clientTCP.println("Content-Type:    multipart/form-
data; boundary=kotakpaket");
    clientTCP.println();
    clientTCP.print(head);

    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n=0;n<fbLen;n=n+1024) {
      if (n+1024<fbLen) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        clientTCP.write(fbBuf, remainder);
      }
    }

    clientTCP.print(tail);

    esp_camera_fb_return(fb);
```

```
    int waitTime = 10000;   // timeout 10 seconds
    long startTimer = millis();
    boolean state = false;

    while ((startTimer + waitTime) > millis()){
      Serial.print(".");
      delay(100);
      while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state==true) getBody += String(c);
        if (c == '\n') {
          if (getAll.length()==0) state=true;
          getAll = "";
        }
        else if (c != '\r')
          getAll += String(c);
        startTimer = millis();
      }
      if (getBody.length()>0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
  }
  else {
    getBody="Connected to api.telegram.org failed.";
    Serial.println("Connected    to    api.telegram.org
failed.");
  }
  return getBody;
}
```

B. 4 Program keseluruhan

```
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial ESP(42, 44);
Servo Myservo1;
Servo Myservo2;
int data = 0;


String response = "";

//G1 (Gate1)
const int triggerPin_1 = 22;
const int echoPin_1 = 24;
```