

pLAMPIRAN

1. Datasheet Komponen

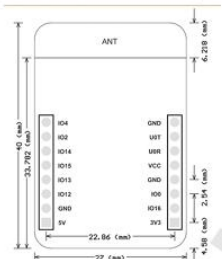


安信可科技
Ai-Thinker

ESP32-CAM Wi-Fi+BT SoC Module V1.0

安信可科技
Ai-Thinker

ESP32-CAM Wi-Fi+BT SoC Module V1.0



ANT

IO4
IO2
IO14
IO19
IO13
IO12
GND
5V

GND
UART
VCC
GND
IO6
2V5

33.702 (mm)
22.86 (mm)
6.218 (mm)
2.54 (mm)
4.50 (mm)
27 (mm)



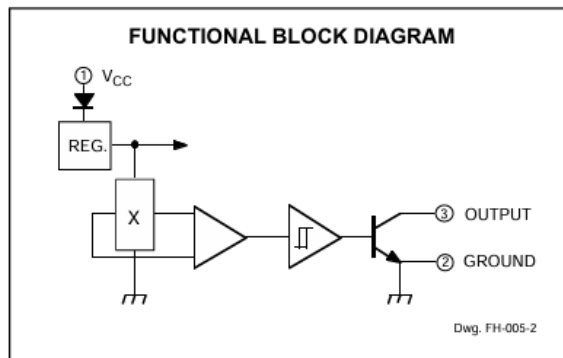
ESP32-CAM

Product Specifications

Module Model	ESP32-CAM
Package	DIP-16
Size	27*40.5*4.5 (±0.2) mm
SPI Flash	Default 32Mbit
RAM	520KB SRAM +4M PSRAM
Bluetooth	Bluetooth 4.2 BR/EDR and BLE standards
Wi-Fi	802.11 b/g/n
Support interface	UART, SPI, I2C, PWM
Support TF card	Maximum support 4G
IO port	9
UART Baudrate	Default 115200 bps
Image Output Format	JPEG(OV2640 support only),BMP,GRAYSCALE
Spectrum Range	2412 ~2484MHz
Antenna	Onboard PCB antenna, gain 2dBi
Transmit Power	802.11b: 17±2 dBm (@11Mbps) 802.11g: 14±2 dBm (@54Mbps) 802.11n: 13±2 dBm (@MCS7)
Receiving Sensitivity	CCK, 1 Mbps: -90dBm CCK, 11 Mbps: -85dBm 6 Mbps (1/2 BPSK): -88dBm 54 Mbps (3/4 64-QAM): -70dBm MCS7 (65 Mbps, 72.2 Mbps): -67dBm
Power Dissipation	Turn off the flash lamp:180mA@5V Turn on the flash lamp and turn on the brightness to the maximum:310mA@5V Deep-sleep: Minimum power consumption can be achieved 6mA@5V Modern-sleep: Minimum up to 20mA@5V Light-sleep: Minimum up to 6.7mA@5V
Security	WPA/WPA2/WPA2-Enterprise/WPS
Power Supply Range	5V
Operating Temperature	-20 °C ~ 85 °C
Storage Environment	-40 °C ~ 90 °C , < 90%RH

ESP32 CAM

3141 THRU 3144 SENSITIVE HALL-EFFECT SWITCHES FOR HIGH-TEMP. OPERATION



ELECTRICAL CHARACTERISTICS at $V_{CC} = 8\text{ V}$ over operating temperature range.

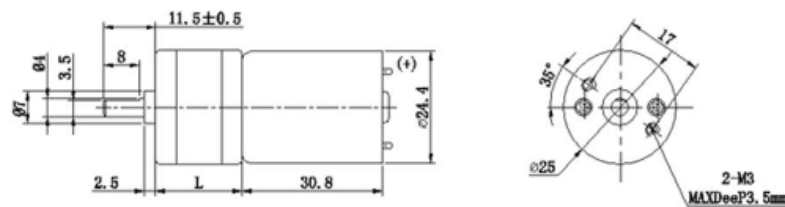
Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Units
Supply Voltage	V_{CC}	Operating	4.5	—	24	V
Output Saturation Voltage	$V_{OUT(SAT)}$	$I_{OUT} = 20\text{ mA}$, $B > B_{OP}$	—	175	400	mV
Output Leakage Current	I_{OFF}	$V_{OUT} = 24\text{ V}$, $B < B_{RP}$	—	<1.0	10	μA
Supply Current	I_{CC}	$B < B_{RP}$ (Output OFF)	—	4.4	9.0	mA
Output Rise Time	t_r	$R_L = 820\ \Omega$, $C_L = 20\text{ pF}$	—	0.04	2.0	μs
Output Fall Time	t_f	$R_L = 820\ \Omega$, $C_L = 20\text{ pF}$	—	0.18	2.0	μs

HALL EFFECT SENSOR A3144

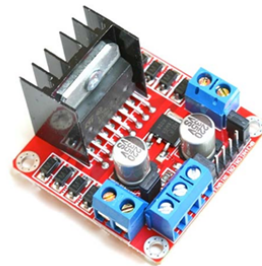
Specification:

Voltage V	No-load		Maximum efficiency pointed				Blockage	
	speed r/min	electric current A	speed r/min	electric current A	Torque Kg.cm	Power W	Torque Kg.cm	electric current A
6	190	0.2	133	0.5	0.75	1.1	4.0	2.1
12	350	0.1	245	0.65	1.4	2.4	5.2	2.2

size:



Motor DC JG25-370 12v

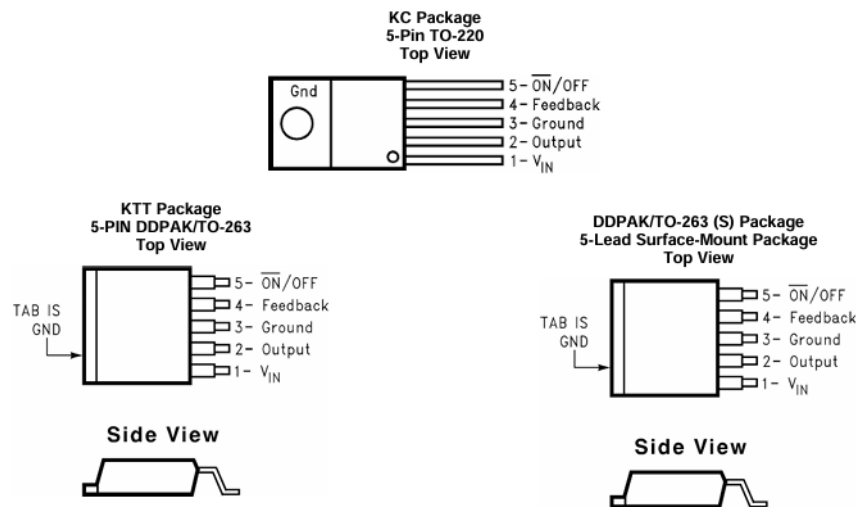


SKU: MDU-1049

Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low: $-0.3V \leq V_{in} \leq 1.5V$.
- High: $2.3V \leq V_{in} \leq V_{ss}$.
- Enable signal input voltage range :
 - Low: $-0.3 \leq V_{in} \leq 1.5V$ (control signal is invalid).
 - High: $2.3V \leq V_{in} \leq V_{ss}$ (control signal active).
- Maximum power consumption: 20W (when the temperature $T = 75^{\circ}C$).
- Storage temperature: $-25^{\circ}C \sim +130^{\circ}C$.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

Driver L298N



Pin Functions

PIN		I/O ⁽¹⁾	DESCRIPTION
NO.	NAME		
1	V _{IN}	I	Supply input pin to collector pin of high-side transistor. Connect to power supply and input bypass capacitors C _{IN} . Path from V _{IN} pin to high frequency bypass C _{IN} and GND must be as short as possible.
2	OUTPUT	O	Emitter pin of the power transistor. This is a switching node. Attached this pin to an inductor and the cathode of the external diode.
3	GROUND	—	Ground pin. Path to C _{IN} must be as short as possible.
4	FEEDBACK	I	Feedback sense input pin. Connect to the midpoint of feedback divider to set V _{OUT} for ADJ version or connect this pin directly to the output capacitor for a fixed output version.
5	ON/OFF	I	Enable input to the voltage regulator. High = OFF and low = ON. Connect to GND to enable the voltage regulator. Do not leave this pin float.
—	TAB	—	Connected to GND. Attached to heatsink for thermal relief for TO-220 package or put a copper plane connected to this pin as a thermal relief for DDPAK package.

(1) I = INPUT, O = OUTPUT

DC Buck Converter LM2596

2. Source Code Arduino IDE ESP32 CAM

```
#include "esp_camera.h"

// --- PIN KONFIGURASI KAMERA (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

// --- KONFIGURASI METODE ROI & INVERTED THRESHOLDING ---
// Format QQVGA memiliki resolusi lebar 160 piksel dan tinggi 120
// piksel.
// Pastikan Anda menyesuaikan 4 nilai ROI di bawah ini berdasarkan
// hasil kalibrasi.
int ROI_X_START = 20; // Geser kotak ke kiri/kanan
int ROI_Y_START = 20; // Geser kotak ke atas/bawah
int ROI_WIDTH   = 120; // Lebar kotak merah
int ROI_HEIGHT  = 65; // Tinggi kotak merah

// KARENA BACKGROUND PUTIH, LOGIKA DIBALIK:
// Nilai intensitas di BAWAH 127 dianggap sebagai bodi
// mobil/bayangan (Objek Gelap)
const int BATAS_THRESHOLD = 170;
// Jika piksel gelap menutupi >= 50% area ROI, maka slot
// diklasifikasikan Terisi (T)
const float BATAS_PERSENTASE = 50.0;

void setup() {
    // Inisialisasi komunikasi Serial ke ESP32 Master
    // Baud rate 9600 sudah sangat cukup untuk mengirim 1 karakter per
    // detik
    Serial.begin(115200);

    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 = 24000000;

    // WAJIB Grayscale untuk mengurangi beban komputasi & RAM memori

```

```

config.pixel_format = PIXFORMAT_GRAYSCALE;
config.frame_size = FRAMESIZE_QQVGA;
config.jpeg_quality = 12;
config.fb_count = 1;

// Cek apakah perangkat keras kamera terdeteksi
if (esp_camera_init(&config) != ESP_OK) {
    // Jika kamera gagal inisialisasi, tunggu 3 detik lalu restart
    sistem
    delay(3000);
    ESP.restart();
}
}

void loop() {
    camera_fb_t * fb = esp_camera_fb_get();
    if (!fb) return; // Jika gagal mengambil frame, abaikan dan ulangi
    loop

    int total_piksel_gelap = 0; // Menggunakan variabel piksel gelap
    int total_piksel_roi = ROI_WIDTH * ROI_HEIGHT;
    int image_width = 160; // Lebar total resolusi QQVGA

    // PEMROSESAN METODE ROI DARI SISI SAMPING
    for (int y = ROI_Y_START; y < (ROI_Y_START + ROI_HEIGHT); y++) {
        for (int x = ROI_X_START; x < (ROI_X_START + ROI_WIDTH); x++) {

            // Mengubah koordinat 2D (x,y) menjadi indeks array 1D
            int index = y * image_width + x;

            // Mengambil nilai warna piksel tunggal (0 = Hitam Murni, 255
            = Putih Murni)
            uint8_t nilai_piksel = fb->buf[index];

            // INVERTED THRESHOLDING & PIXEL COUNTING
            // Mengevaluasi apakah piksel lebih gelap dari batas latar
            belakang putih
            if (nilai_piksel < BATAS_THRESHOLD) {
                total_piksel_gelap++; // Dihitung sebagai bagian dari
                mobil/penghalang
            }
        }
    }
}

```

```

    // Segera kembalikan buffer memori agar ESP32-CAM tidak Crash /
    Brownout
    esp_camera_fb_return(fb);

    // PROSES KLASIFIKASI KEPADATAN OBJEK (PENHALANG CAHAYA)
    float persentase_gelap = ((float)total_piksel_gelap /
total_piksel_roi) * 100.0;

    // Jika area gelap memakan 50% atau lebih dari total kotak ROI,
    nyatakan Terisi
    char dataKirim = (persentase_gelap >= BATAS_PERSENTASE) ? 'T' :
    'K';

    // KIRIM DATA VIA KABEL TX KE ESP32 MASTER
    // Menggunakan println agar data memiliki karakter pembatas
    (newline \n)
    Serial.println(dataKirim);

    // Interval 1 detik (1000ms) untuk mencegah data flooding
    (penumpukan memori) di Master
    delay(300);
}

```

3. Source Code Arduino IDE ESP32 Master

```

#include <Arduino.h>
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>

// =====
// 1. PIN HARDWARE & SERIAL (MASTER)
// =====
#define SS_PIN 21
#define RST_PIN 5
MFRC522 rfid(SS_PIN, RST_PIN);

const int pinHall = 4;

// Pin Kustom untuk Komunikasi Kamera (Serial1)
#define RX_CAM 26
#define TX_CAM 33

// Pin Kustom untuk Komunikasi Slave (Serial2)

```

```

#define RX_SLAVE 14 // Telinga Master dari Slave (Sambung ke TX
Slave)
#define TX_SLAVE 27 // Mulut Master ke Slave (Sambung ke RX Slave)

// =====
// 2. DATABASE ARRAY & VARIABEL SISTEM
// =====
const int JUMLAH_SLOT = 6;
String slotMemory[JUMLAH_SLOT] = {"", "", "", "", "", ""};
int currentSlot = 1;
int targetSlot = -1;

bool isProsesMasuk = false;
bool isProsesKeluar = false;

int lastHallState = HIGH;
unsigned long lastHallTime = 0;

String lastUID = "";
bool waitingDoubleTap = false;
unsigned long timeFirstTap = 0;
const unsigned long BATAS_WAKTU_TAP = 3000;

// =====
// 3. KOMUNIKASI SERIAL
// =====
bool menungguValidasiKamera = false;
unsigned long waktuTungguKamera = 0;

// Kirim perintah ke Slave
void kirimKeSlave(String perintah) {
    Serial.println("[ TRANSMIT ] Ke Slave: " + perintah);
    Serial2.println(perintah);
}

// =====
// 4. LOGIKA UTAMA
// =====
void prosesCheckIn(String uid) {
    for(int i = 0; i < JUMLAH_SLOT; i++){
        if(slotMemory[i] == uid){
            Serial.println("[ VALIDATOR ] DITOLAK: Kendaraan sudah ada di
dalam database!");
            return;
        }
    }
}

```

```

    }

    bool adaSlotKosong = false;
    for(int i = 0; i < JUMLAH_SLOT; i++){
        if(slotMemory[i] == ""){ adaSlotKosong = true; break; }
    }
    if(!adaSlotKosong){
        Serial.println("[ VALIDATOR ] DITOLAK: Parkiran PENUH!");
        return;
    }

    Serial.println("\n-> [ MASUK ] Lolos Validasi. Mengecek kondisi
Slot SAAT INI (" + String(currentSlot) + "...");
    isProsesMasuk = true;
    isProsesKeluar = false;

    while(Serial1.available()) Serial1.read();
    menungguValidasiKamera = true;
    waktuTungguKamera = millis();
}

void prosesCheckOut(String uid) {
    targetSlot = -1;
    for(int i = 0; i < JUMLAH_SLOT; i++){
        if(slotMemory[i] == uid){ targetSlot = i + 1; break; }
    }

    if(targetSlot == -1){
        Serial.println("[ VALIDATOR ] DITOLAK: Kartu tidak terdaftar!");
        return;
    }

    Serial.println("\n-> [ KELUAR ] Menjemput mobil di Slot: " +
String(targetSlot));
    isProsesKeluar = true;
    isProsesMasuk = false;

    if(currentSlot == targetSlot){
        kirimKeSlave("GATE_OUT");
        slotMemory[currentSlot - 1] = "";
        Serial.println("-> Database: Slot " + String(currentSlot) + "
KOSONG");
        isProsesKeluar = false;
    } else {
        kirimKeSlave("MOTOR_ON");
    }
}

```

```

    }
}

// =====
// 5. SETUP UTAMA
// =====

void setup() {
    Serial.begin(9600); // Ke Laptop

    // --- KODINGAN BARU: CUSTOM PIN UNTUK SERIAL 2 ---
    Serial2.begin(9600, SERIAL_8N1, RX_SLAVE, TX_SLAVE); // Ke Slave
via Pin 14 & 27
    // -----

    Serial1.begin(9600, SERIAL_8N1, RX_CAM, TX_CAM); // Ke Kamera
via Pin 26 & 33

    WiFi.mode(WIFI_OFF);
    delay(500);

    SPI.begin();
    rfid.PCD_Init();

    byte rfid_v = rfid.PCD_ReadRegister(rfid.VersionReg);
    if (rfid_v == 0x00 || rfid_v == 0xFF) {
        Serial.println("[ WARNING ] Komunikasi RFID GAGAL! Cek kabel.");
    } else {
        Serial.println("[ OK ] RFID Siap dan Stabil.");
    }
}

pinMode(pinHall, INPUT_PULLUP);
 kirimKeSlave("MOTOR_OFF");

    Serial.println("\n--- SISTEM MASTER (TRIPLE SERIAL AMAN) SIAP ---");
}

// =====
// 6. LOOP UTAMA
// =====

void loop() {

    // A. PENGHITUNG HALL SENSOR
    if(isProsesMasuk || isProsesKeluar) {
        int currentHallState = digitalRead(pinHall);

```

```

    if(currentHallState == LOW && lastHallState == HIGH) {
        if(millis() - lastHallTime > 500) {
            lastHallTime = millis();
            currentSlot++;
            if(currentSlot > JUMLAH_SLOT) currentSlot = 1;

            Serial.println("\n[ SENSOR ] Keranjang fisik nomor " +
String(currentSlot) + " tiba.");

            if (isProsesMasuk) {
                kirimKeSlave("MOTOR_OFF");

                while(Serial1.available()) Serial1.read();

                menungguValidasiKamera = true;
                waktuTungguKamera = millis();
                Serial.println("-> Membaca sinyal ESP-CAM untuk Slot " +
String(currentSlot) + "...");
            }
            else if (isProsesKeluar) {
                if(currentSlot == targetSlot) {
                    kirimKeSlave("MOTOR_OFF");
                    kirimKeSlave("GATE_OUT");
                    slotMemory[currentSlot - 1] = "";
                    Serial.println("-> Mobil keluar. Slot " +
String(currentSlot) + " dikosongkan.");
                    isProsesKeluar = false;
                }
            }
        }
    }
    lastHallState = currentHallState;
}

// B. EKSEKUSI HASIL DARI ESP-CAM
if (menungguValidasiKamera) {
    if (Serial1.available() > 0) {
        char statusCam = Serial1.read();

        if(statusCam == 'K') {
            Serial.println("[ ESP-CAM ] AMAN: Fisik Slot " +
String(currentSlot) + " KOSONG!");
            kirimKeSlave("GATE_IN");

            slotMemory[currentSlot - 1] = lastUID;

```

```

        Serial.println("-> Database menyimpan UID: " + lastUID + "
di Slot " + String(currentSlot));

        menungguValidasiKamera = false;
        isProsesMasuk = false;
    }
    else if(statusCam == 'T') {
        Serial.println("[ ESP-CAM ] TERISI: Slot " +
String(currentSlot) + " ada isinya. Lanjut putar motor mencari...");
        kirimKeSlave("MOTOR_ON");
        menungguValidasiKamera = false;
    }
}

// Timeout
if(millis() - waktuTungguKamera > 5000) {
    Serial.println("[ ERROR ] Kabel Kamera tidak merespons! Mesin
berhenti.");
    kirimKeSlave("MOTOR_OFF");
    menungguValidasiKamera = false;
    isProsesMasuk = false;
    lastUID = "";
}

// C. PEMBACAAN KARTU RFID
if (isProsesMasuk || isProsesKeluar || menungguValidasiKamera)
return;

if (waitingDoubleTap && (millis() - timeFirstTap >
BATAS_WAKTU_TAP)) {
    waitingDoubleTap = false;
    prosesCheckIn(lastUID);
}

if (!rfid.PICC_IsNewCardPresent()) return;
if (!rfid.PICC_ReadCardSerial()) return;

String uid = "";
for (byte i = 0; i < rfid.uid.size; i++) {
    uid += String(rfid.uid.uidByte[i] < 0x10 ? "0" : "");
    uid += String(rfid.uid.uidByte[i], HEX);
}
uid.toUpperCase();
rfid.PICC_HaltA();

```

```

    if (waitingDoubleTap && uid == lastUID) {
        waitingDoubleTap = false;
        prosesCheckOut(uid);
    } else {
        lastUID = uid;
        timeFirstTap = millis();
        waitingDoubleTap = true;
        Serial.println("\n[ DETEKSI ] Tap Pertama UID: " + uid);
        Serial.println("-> Tahan kartu 3 detik untuk Keluar, Lepas untuk
Masuk.");
    }
}

```

4. Source Code Arduino IDE ESP32 Slave

```

#include <Arduino.h>
#include <ESP32Servo.h>

// =====
// 1. PIN KOMUNIKASI (DARI MASTER)
// =====
#define RX_MASTER 16 // Telinga Slave
#define TX_MASTER 17 // Mulut Slave

// =====
// 2. PIN HARDWARE (MOTOR DC L298N 12V)
// =====
// Dinamo Sisi Kiri
#define ENA 33
#define IN1 25
#define IN2 26

// Dinamo Sisi Kanan
#define ENB 33 // PENTING: Karena Anda memparalel pin ENA dan ENB ke
GPIO 33
#define IN3 27
#define IN4 14

// --- KONFIGURASI KECEPATAN (PWM) ---
// Nilai rentang PWM di ESP32 adalah 0 (Berhenti) hingga 255
(Maksimal)
const int KECEPATAN_ROTARY = 160;

// Konfigurasi PWM ESP32 (Core v3.x.x)

```

```

const int PWM_FREQ = 1000;    // Frekuensi 1KHz
const int PWM_RESOLUTION = 8; // Resolusi 8 bit (0-255)

// =====
// 3. PIN HARDWARE (2 MOTOR SERVO)
// =====
#define PIN_SERVO_IN 18    // Servo untuk Pintu Masuk
#define PIN_SERVO_OUT 19   // Servo untuk Pintu Keluar

Servo servoMasuk;
Servo servoKeluar;

// Sudut Gate Out (Berputar 0 ke 90 derajat)
const int SUDUT_TUTUP_OUT = 0;
const int SUDUT_BUKA_OUT  = 90;

// Sudut Gate In (Berputar arah berlawanan: 180 ke 90 derajat /
// rotasi -90 derajat)
const int SUDUT_TUTUP_IN  = 180;
const int SUDUT_BUKA_IN   = 90;

// =====
// 4. SETUP UTAMA
// =====
void setup() {
    // Ditingkatkan ke 115200 agar Anda tidak perlu bolak-balik ubah
    // baud rate di Serial Monitor saat pindah dari Master ke Slave
    Serial.begin(115200);

    // Beri waktu listrik stabil sebelum ESP32 bekerja penuh
    delay(2000);
    Serial.println("\n[ SYSTEM ] Memulai inisialisasi
kelistrikan...");

    // Kecepatan WAJIB 9600 menyesuaikan kodingan di Master (Serial2)
    Serial2.begin(9600, SERIAL_8N1, RX_MASTER, TX_MASTER);

    // Setup L298N (Motor DC) - Pin Arah
    pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);

    // Setup L298N (Motor DC) - Pin Kecepatan (PWM Core v3.x.x)
    // Langsung attach ke Pin ENA.
    ledcAttach(ENA, PWM_FREQ, PWM_RESOLUTION);

```

```

// Pastikan motor rotary diam saat baru dinyalakan
digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
ledcWrite(ENA, 0); // Putuskan daya via PWM (Target ke Pin)

// Alokasikan Timer agar Servo tidak bikin ESP32 Crash/Restart
// KRITIS: Timer 0 ditiadakan agar tidak bentrok dengan ledcAttach
(PWM Motor)
ESP32PWM::allocateTimer(1);
ESP32PWM::allocateTimer(2);
ESP32PWM::allocateTimer(3);

// Setup Servo Masuk
servoMasuk.setPeriodHertz(50);
servoMasuk.attach(PIN_SERVO_IN, 500, 2400);

// Setup Servo Keluar
servoKeluar.setPeriodHertz(50);
servoKeluar.attach(PIN_SERVO_OUT, 500, 2400);

// Kunci kedua gerbang di posisi tutup masing-masing
delay(200);
servoMasuk.write(SUDUT_TUTUP_IN);
servoKeluar.write(SUDUT_TUTUP_OUT);

Serial.println("=====
");
Serial.println(" [ STATUS ] ESP32 SLAVE (OTOT PENGGERAK)
SIAP! ");
Serial.println(" [ STATUS ] Kecepatan PWM di-set pada: " +
String(KECEPATAN_ROTARY));
Serial.println(" [ STATUS ] Menunggu instruksi dari ESP32
Master ");
Serial.println("=====
");
}

// =====
// 5. LOOP UTAMA (Mendengarkan Master)
// =====
String pesanMasuk = "";

void loop() {
    // Jika ada aliran data di kabel komunikasi...
    while (Serial2.available() > 0) {

```

[illegible]

```

    // Matikan via PWM untuk pengereman yang aman
    ledcWrite(ENA, 0);
    // Matikan arah untuk memastikan rem penuh
    digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
}

else if (perintah == "GATE_IN") {
    Serial.println("[-] AKSI: SERVO MASUK terbuka (Mobil Check-
In).");
    servoMasuk.write(SUDUT_BUKA_IN);

    delay(5000); // Tahan gerbang terbuka 5 detik agar mobil bisa
masuk

    servoMasuk.write(SUDUT_TUTUP_IN);
    Serial.println("[-] INFO: SERVO MASUK tertutup kembali secara
otomatis.");
}

else if (perintah == "GATE_OUT") {
    Serial.println("[-] AKSI: SERVO KELUAR terbuka (Mobil Check-
Out).");
    servoKeluar.write(SUDUT_BUKA_OUT);

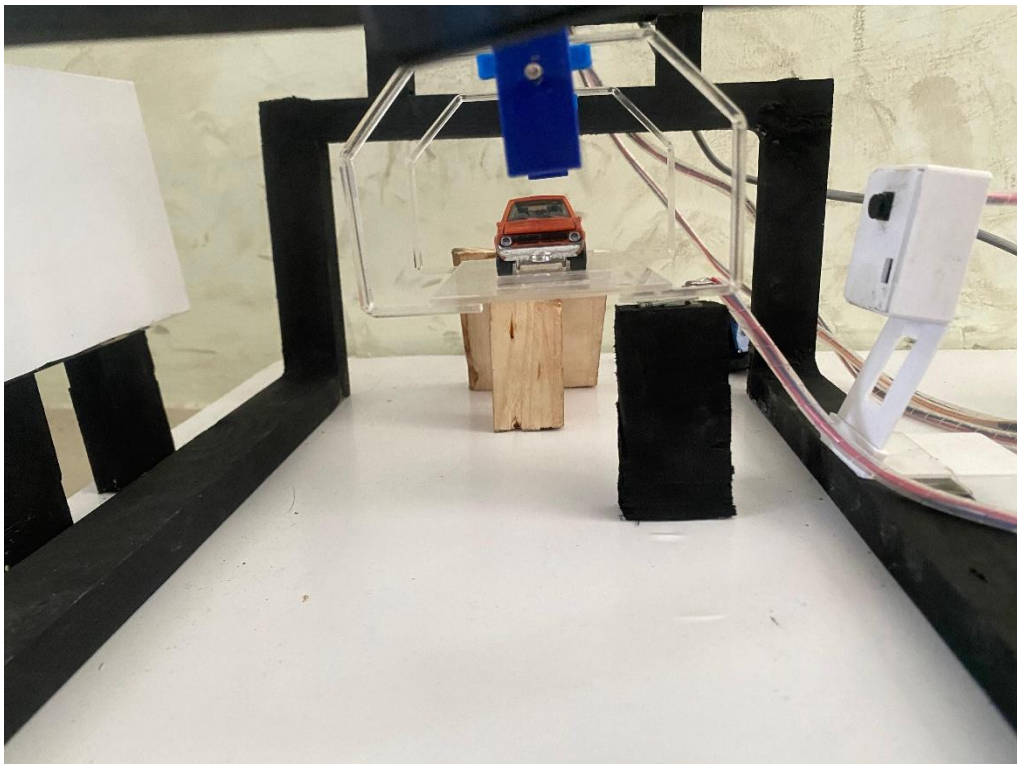
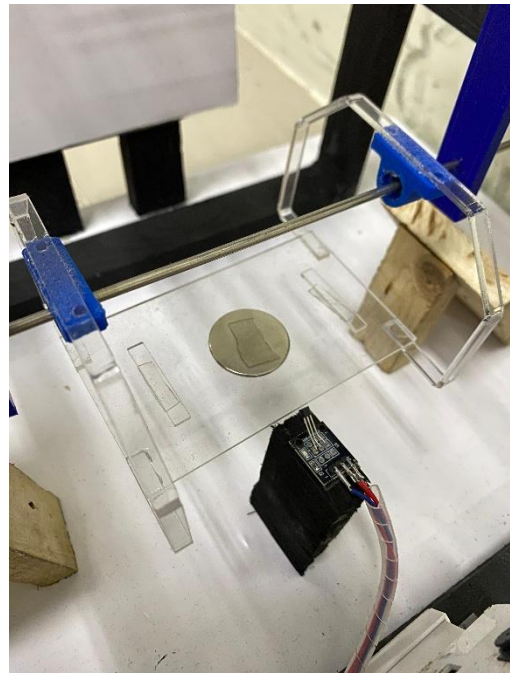
    delay(5000); // Tahan gerbang terbuka 5 detik agar mobil bisa
keluar

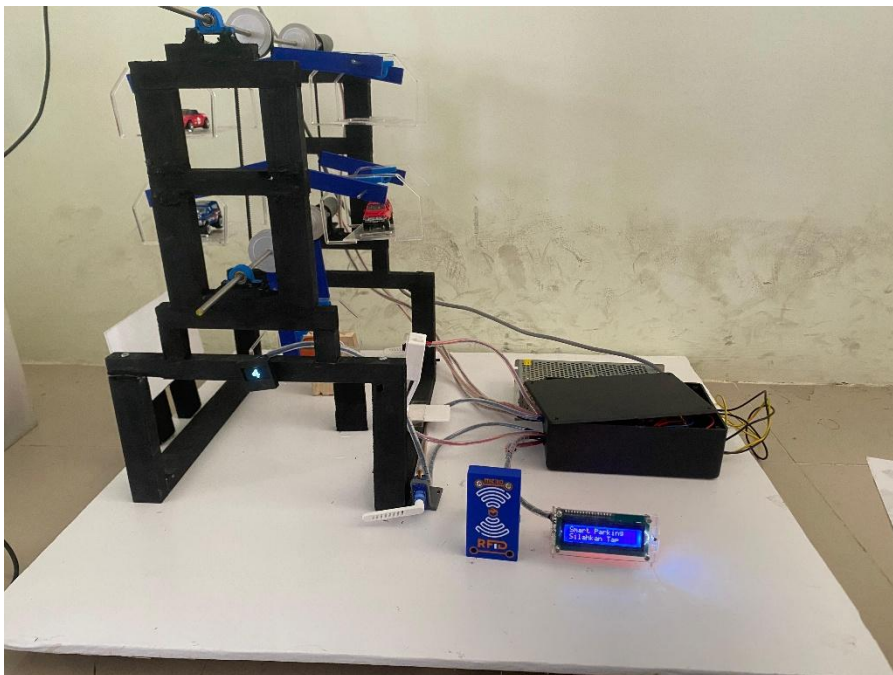
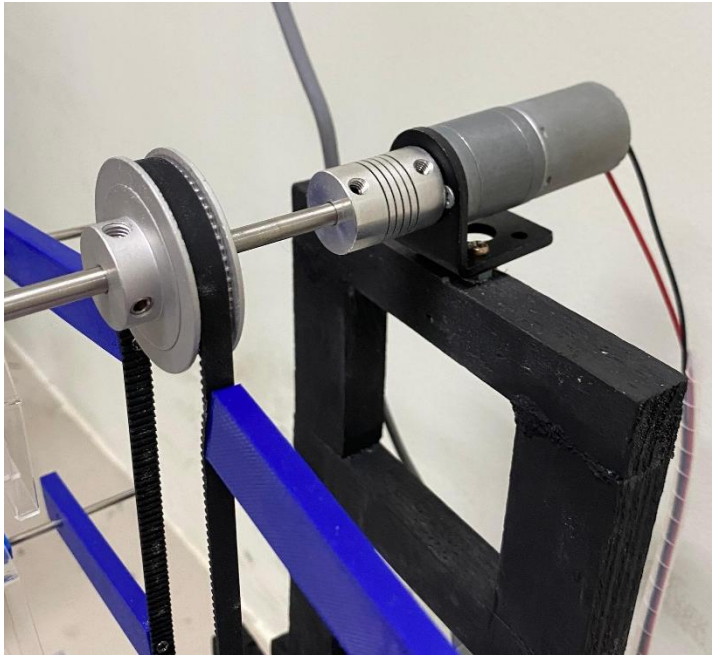
    servoKeluar.write(SUDUT_TUTUP_OUT);
    Serial.println("[-] INFO: SERVO KELUAR tertutup kembali secara
otomatis.");
}

else {
    Serial.println("[!] ERROR: Kata kunci perintah tidak terdaftar
di sistem Slave!");
}
}

```

5. Dokumentasi Proses Pembuatan Alat

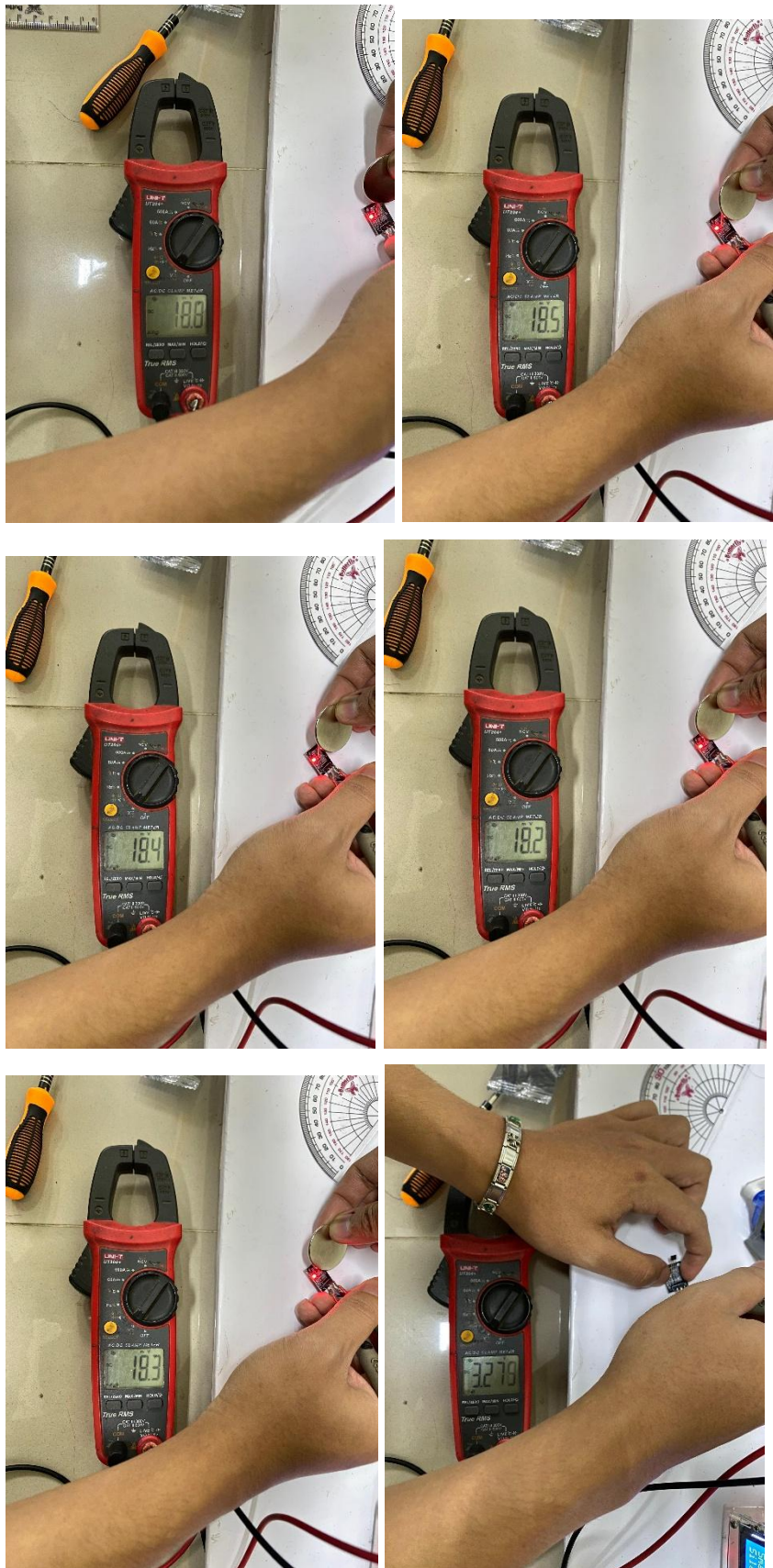




6. Pengukuran Catu Daya (Tegangan output, Input, dan Arus)



7. Pengukuran Jarak *Hall Effect* Sensor (0,6; 0,5; 0,4; 0,3; 0,2; 0,1 cm)



8. Pengukuran Intensitas Cahaya menggunakan Luxmeter pada HP



9. Pengukuran Warna kendaraan dengan ESP32 CAM

```

20:29:14.718 -> ===== PENGUJIAN VISUAL =====
20:29:14.718 -> Rata-rata Grayscale (0-255)      : 141.31
20:29:14.753 -> Kepadatan Piksel Gelap (%)       : 58.76%
20:29:14.753 -> Logika Sinyal Terkirim            : 1
20:29:14.753 -> Karakter Status UART           : T
20:29:14.753 -> =====

20:31:05.893 -> ===== PENGUJIAN VISUAL =====
20:31:05.893 -> Rata-rata Grayscale (0-255)      : 142.18
20:31:05.893 -> Kepadatan Piksel Gelap (%)       : 56.85%
20:31:05.893 -> Logika Sinyal Terkirim            : 1
20:31:05.893 -> Karakter Status UART           : T
20:31:05.893 -> =====

20:33:19.282 -> ===== PENGUJIAN VISUAL =====
20:33:19.282 -> Rata-rata Grayscale (0-255)      : 145.62
20:33:19.282 -> Kepadatan Piksel Gelap (%)       : 54.44%
20:33:19.282 -> Logika Sinyal Terkirim            : 1
20:33:19.282 -> Karakter Status UART           : T
20:33:19.282 -> =====

20:33:51.638 -> ===== PENGUJIAN VISUAL =====
20:33:51.638 -> Rata-rata Grayscale (0-255)      : 148.15
20:33:51.638 -> Kepadatan Piksel Gelap (%)       : 53.49%
20:33:51.638 -> Logika Sinyal Terkirim            : 1
20:33:51.638 -> Karakter Status UART           : T
20:33:51.638 -> =====

```

```

20:35:06.397 -> ===== PENGUJIAN VISUAL =====
20:35:06.397 -> Rata-rata Grayscale (0-255)      : 151.16
20:35:06.397 -> Kepadatan Piksel Gelap (%)           : 51.86%
20:35:06.397 -> Logika Sinyal Terkirim                  : 1
20:35:06.397 -> Karakter Status UART                     : T
20:35:06.397 -> =====

20:36:03.005 -> ===== PENGUJIAN VISUAL =====
20:36:03.005 -> Rata-rata Grayscale (0-255)      : 135.15
20:36:03.005 -> Kepadatan Piksel Gelap (%)           : 62.36%
20:36:03.005 -> Logika Sinyal Terkirim                  : 1
20:36:03.005 -> Karakter Status UART                     : T
20:36:03.005 -> =====

20:24:21.667 -> ===== PENGUJIAN VISUAL =====
20:24:21.667 -> Rata-rata Grayscale (0-255)      : 145.22
20:24:21.713 -> Kepadatan Piksel Gelap (%)           : 54.69%
20:24:21.713 -> Logika Sinyal Terkirim                  : 1
20:24:21.713 -> Karakter Status UART                     : T
20:24:21.713 -> =====

```