



**RANCANG BANGUN SISTEM *MONITORING* KEBISINGAN
PADA KENDARAAN BERMOTOR DI LINGKUNGAN
PERUMAHAN BERBASIS IoT**

“Pembuatan Alat Sistem *Monitoring* Kebisingan Pada Kendaraan Bermotor di
Lingkungan Perumahan”

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga

Disusun oleh:

Syafiq Surya Rucita 1903332094

PROGRAM STUDI TELEKOMUNIKASI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2022



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RANCANG BANGUN SISTEM *MONITORING* KEBISINGAN PADA KENDARAAN BERMOTOR DI LINGKUNGAN PERUMAHAN BERBASIS IoT

“Pembuatan Alat Sistem *Monitoring* Kebisingan Pada Kendaraan Bermotor di
Lingkungan Perumahan”

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga

Disusun oleh:
**POLITEKNIK
NEGERI
JAKARTA**

Syafiq Surya Rucita 1903332094

PROGRAM STUDI TELEKOMUNIKASI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2022



HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Syafiq Surya Rucita

NIM : 1903332094

Tanda Tangan :

Tanggal : 22 Agustus 2022



POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



HALAMAN PENGESAHAN TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama : Syafiq Surya Rucita
NIM : 1903332094
Program Studi : Teknik Telekomunikasi
Judul Tugas Akhir : Rancang Bangun Sistem *Monitoring* Kebisingan
pada Kendaraan Bermotor di Lingkungan Perumahan
Berbasis IoT

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada Senin, 1
Agustus 2022 dan dinyatakan **LULUS**.

Pembimbing : Rifqi Fuadi Hasani, S.T., M.T.
NIP. 199208182019031015

Depok, 22 Agustus 2022

Disahkan oleh

Ketua Jurusan Teknik Elektro



Ir. Sri Danaryani, M.T.

NIP. 19630503 199103 2 001

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



KATA PENGANTAR

Alhamdulillah, puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Sistem *Monitoring* Kebisingan Suara pada Kendaraan Bermotor di Lingkungan Perumahan Berbasis IoT ”.

Adapun tujuan penulisan dari laporan ini adalah untuk memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga di Politeknik Negeri Jakarta. Tanpa bantuan dan bimbingan dari berbagai pihak, penulis menyadari akan sangat sulit untuk menyelesaikan Tugas Akhir ini. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Rifqi Fuadi Hasani, S.T., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing penulis dalam melakukan penyusunan Tugas Akhir ini;
2. Seluruh Dosen Program Studi D3 Telekomunikasi atas segala bantuan dan bimbingannya;
3. Orang tua dan keluarga penulis yang tak henti-hentinya memberikan dukungan dan semangat kepada penulis;
4. Mohammad Lucky Pratama selaku rekan Tugas Akhir dan serta seluruh rekan Program Studi D3 Telekomunikasi Angkatan tahun 2019 yang telah ikut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari, bahwa dalam penyusunan Tugas Akhir ini masih jauh dari kata sempurna karena pengalaman dan pengetahuan penulis yang terbatas. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang.

Depok, Agustus 2022

Penulis



RANCANG BANGUN SISTEM *MONITORING* KEBISINGAN SUARA PADA KENDARAAN BERMOTOR DI LINGKUNGAN PERUMAHAN BERBASIS IoT

ABSTRAK

Maraknya tren memodifikasi knalpot kendaraan bermotor khususnya sepeda motor sehingga menimbulkan kebisingan yang kerap kali mengganggu warga di lingkungan perumahan. Berdasarkan peraturan Menteri Lingkungan Hidup, kebisingan yang diperbolehkan untuk kendaraan di bawah 150cc adalah 80 dB. Upaya mengurangi masalah tersebut, maka dibuatlah Sistem Monitoring Kebisingan pada Kendaraan Bermotor di Lingkungan Perumahan Berbasis IoT yang dapat membantu satpam dalam memberikan peringatan kepada pengendara motor serta menangkap gambar dari sepeda motor yang menyebabkan kebisingan. Pada sistem ini terdapat Arduino UNO yang berfungsi sebagai pengontrol utama dari komponen yang terhubung seperti sensor suara GY-MAX4466, Relay, ESP32 CAM dan Serial mp3 player. Sistem ini terhubung dengan database Firebase agar dapat dimonitoring melalui aplikasi android yang telah dibuat. Alat ini telah dilakukan pengujian dan dapat mendeteksi adanya kebisingan pada kendaraan bermotor dengan kebisingan di atas 80 dB dan setiap komponen dapat bekerja sesuai pemrograman yang telah dibuat. Alat ini juga telah dilakukan pengujian Key Performance Indicators dengan nilai RSRP -74 dBm, RSRQ -13 dB dan SINR 0.0 dB sehingga membuktikan jaringan yang digunakan pada sistem dalam kondisi yang optimal dan nilai RSSI untuk penangkapan sinyal Wi-Fi pada alat dalam kategori baik pada rentang jarak 1-3 meter.

Kata Kunci : *Arduino, Firebase Kebisingan, Kendaraan Bermotor, Key Performance Indicators, Monitoring.*

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA



NOISE MONITORING SYSTEM DESIGN ON MOTOR VEHICLES IN HOUSING ENVIRONMENT BASED ON IoT

ABSTRACT

The rise of the trend of modifying motorized vehicles, especially motorcycles, has caused frequent disturbances to residents in residential areas. Based on the regulation of the Minister of the Environment, what is allowed for vehicles under 150cc is 80 dB. To reduce this problem, a Noise Monitoring System for Motorized Vehicles in an IoT-Based Residential Environment was created that can assist security guards in alerting motorcyclists and capturing images of motorcycles that cause triggers. In this system there is an Arduino UNO which functions as the main controller of the connected components such as the GY-MAX4466 sound sensor, Relay, ESP32 CAM and Serial mp3 player. This system is connected to the Firebase database so that it can be monitored through the android application that has been created. This tool has been tested and can detect any findings in motorized vehicles with above 80 dB and each component can work according to the programming that has been done. This tool has also been tested for Key Performance Indicators with RSRP -74 dBm, RSRQ -13 dB and SINR 0.0 dB so as to prove the network used in the system is in optimal conditions and the RSSI value to capture Wi-Fi signals on the device is in good category at a distance of 1 -3 meters.

Keywords : Arduino, Firebase, Key Performance Indicators, Monitoring, Motor Vehicles, Noise.

POLITEKNIK
NEGERI
JAKARTA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN TUGAS AKHIR	iii
KATA PENGANTAR.....	iv
ABSTRAK	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Kegiatan	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Luaran	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 <i>Internet</i>	3
2.2 <i>Internet of things</i>	3
2.3 Kebisingan	4
2.4 Arduino Uno	5
2.5 ESP32 CAM	5
2.6 Sensor Suara	6
2.7 <i>Relay</i>	7
2.8 <i>Serial mp3 player</i>	7
2.9 <i>Lampu Sorot 50 Watt</i>	8
2.10 Speaker	9
2.11 <i>Power Supply / Catu Daya</i>	10
2.12 Arduino IDE	10
2.13 Sketch	12
2.13.1 Struktur Dasar Penulisan Sketch	12
2.13.2 Sintak dalam Penulisan Program.....	12
2.14 Google Firebase	13
2.15 <i>Key Performance Indicators (KPI)</i>	14
2.15.1 <i>Reference Signal Received Power (RSRP)</i>	14
2.15.2 <i>Reference Signal Received Quality (RSRQ)</i>	15
2.15.3 <i>Signal to Noise Ratio (SINR)</i>	16
2.16 <i>Received Signal Strength Indicator (RSSI)</i>	16
BAB III RANCANGAN DAN REALISASI.....	17
3.1 Rancangan Alat.....	17
3.1.1 Deskripsi Alat	17
3.1.2 Diagram Blok	18
3.1.3 Cara Kerja alat	19
3.1.4 Spesifikasi Alat.....	21
3.2 Realisasi Alat	21
3.2.1 Realisasi Sistem <i>Monitoring</i> Kebisingan pada Kendaraan Bermotor ..	22

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritis atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.2.2	Realisasi Rangkaian Catu Daya.....	26
3.2.3	Realisasi Algoritma Pemrograman.....	29
3.2.4	Realisasi <i>database</i> pada Google Firebase	35
BAB IV PEMBAHASAN.....		40
4.1	Pengujian Catu Daya	40
4.1.1	Deskripsi Pengujian.....	40
4.1.2	Prosedur Pengujian Catu Daya.....	41
4.1.3	Data Hasil Pengujian Catu Daya	42
4.1.4	Analisa Data / Evaluasi Pengujian Catu Daya.....	42
4.2	Pengujian Program Arduino IDE.....	43
4.2.1	Deskripsi Pengujian Program Arduino IDE	43
4.2.2	Prosedur Pengujian Program Arduino IDE	43
4.2.3	Analisa Data / Evaluasi Program Arduino IDE.....	45
4.3	Pengujian Sistem Arduino	45
4.3.1	Deskripsi Pengujian Sistem Arduino	45
4.3.2	Prosedur Pengujian Sistem Arduino.....	46
4.3.3	Data Hasil Pengujian Sensor GY-MAX4466.....	46
4.3.4	Data Hasil Pengujian <i>Relay</i>	51
4.3.5	Data Hasil Pengujian <i>Serial mp3 player</i>	52
4.3.6	Data Hasil pengujian ESP32 CAM	53
4.3.7	Analisa Data Pengujian Sistem Arduino	54
4.4	Pengujian <i>Wi-Fi</i> ESP32 CAM.....	55
4.4.1	Deskripsi Pengujian <i>Wi-Fi</i> ESP32 CAM	56
4.4.2	Prosedur Pengujian <i>Wi-Fi</i> ESP32 CAM.....	56
4.4.3	Data Hasil Pengujian Terkoneksi ke <i>Access Point</i>	56
4.4.4	Data Hasil Pengujian RSSI pada <i>Wi-Fi</i> ESP32 CAM.....	57
4.4.5	Analisa Data Pengujian <i>Wi-Fi</i> ESP32 CAM.....	57
4.5	Pengujian <i>Key Performance Indicator</i> (KPI) pada Jaringan Seluler.....	57
4.5.1	Deskripsi Pengujian KPI pada Jaringan Seluler	58
4.5.2	Prosedur Pengujian KPI pada Jaringan Seluler	58
4.5.3	Data Hasil Pengujian KPI pada Jaringan Seluler	58
4.5.4	Analisa Data Pengujian KPI pada Jaringan Seluler.....	59
BAB V PENUTUP.....		60
5.1	Simpulan.....	60
5.2	Saran	60
DAFTAR PUSTAKA		61
DAFTAR RIWAYAT HIDUP		62

DAFTAR GAMBAR

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 2. 1 Tingkat kebisingan (dB) Kendaraan Bermotor	4
Gambar 2. 2 Arduino UNO	5
Gambar 2. 3 ESP32-CAM	6
Gambar 2. 4 Sensor Suara GY-MAX4466	7
Gambar 2. 5 Relay.....	7
Gambar 2. 6 <i>Serial mp3 player</i>	8
Gambar 2. 7 Lampu Sorot 50 Watt.....	9
Gambar 2. 8 <i>Speaker</i>	9
Gambar 2. 9 perbedaan antara tegangan DC dan AC	10
Gambar 2. 10 <i>Software</i> Arduino IDE.....	11
Gambar 2. 11 Logo Firebase	13
Gambar 2. 12 Ilustrasi Nilai RSRP	15
Gambar 2. 13 Ilustrasi Nilai RSRQ.....	15
Gambar 3. 1 Ilustrasi Sistem <i>Monitoring</i> Kebisingan.....	18
Gambar 3. 2 Diagram Blok Sistem Pendeteksi Kebisingan.....	19
Gambar 3. 3 <i>Flowchart</i> Sistem <i>Monitoring</i> Kebisingan.....	20
Gambar 3. 4 Skematik Keseluruhan Sistem <i>Monitoring</i> Kebisingan	22
Gambar 3. 5 Rangkaian Jalur Catu Daya	23
Gambar 3. 6 Rangkaian Skematik ESP32 CAM ke Arduino UNO.....	24
Gambar 3. 7 Rangkaian Skematik Sensor GY-MAX4466 ke Arduino UNO.	25
Gambar 3. 8 Rangkaian Skematik <i>Relay</i> ke Arduino UNO.....	25
Gambar 3. 9 Rangkaian Skematik <i>Serial mp3 player</i> ke Arduino UNO	26
Gambar 3. 10 Skematik rangkaian catu daya.....	27
Gambar 3. 11 Layout Catu Daya.....	29
Gambar 3. 12 Tampilan Awal Firebase	36
Gambar 3. 13 Pembuatan Project Baru	36
Gambar 3. 14 Pemberian Nama Project.....	37
Gambar 3. 15 Tampilan Create Database	37
Gambar 3. 16 Menambah Variabel pada Database.....	38
Gambar 3. 17 Data Pada Firebase	38
Gambar 3. 18 Isi <i>database</i> History	39
Gambar 4. 1 Pengukuran tegangan input Trafo	41
Gambar 4. 2 Hasil pengukuran tegangan pada <i>output</i> trafo.....	41
Gambar 4. 3 Pengukuran Tegangan Catu Daya	42
Gambar 4. 4 Pengaturan <i>Board</i> pada Arduino IDE	44
Gambar 4. 5 Pengaturan <i>Port</i> pada Arduino IDE	44
Gambar 4. 6 Proses Unggah Selesai	45
Gambar 4. 7 Pengambilan Data Dari Sensor GY-MAX4466	46
Gambar 4. 8 Kondisi <i>relay</i> saat menyala dan tidak menyala.....	51
Gambar 4. 9 Tampilan pada <i>Serial monitor</i>	52
Gambar 4. 10 Tampilan <i>Serial monitor Serial mp3 player</i>	52
Gambar 4. 11 Hasil Pengambilan Gambar ESP32 CAM pada kondisi gelap.....	53

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 12 Hasil Pengambilan Gambar ESP32 CAM pada kondisi terang	54
Gambar 4. 13 Tampilan <i>Serial monitor</i> ESP32 CAM	54
Gambar 4. 14 Hasil Pengujian KPI Menggunakan Aplikasi <i>Cell Tower Locator</i>	58





DAFTAR TABEL

Tabel 2. 1 Rentang Nilai RSRP	15
Tabel 2. 2 Rentang Nilai RSRQ.....	16
Tabel 2. 3 Rentang Nilai SINR	16
Tabel 2. 4 Nilai dan Kategori RSSI	16
Tabel 3. 1 Spesifikasi Alat Sistem Pendeteksi Kebisingan.....	21
Tabel 3. 2 <i>Pin</i> komponen dengan <i>pin</i> Arduino Uno	22
Tabel 3. 3 <i>Pin</i> Catu Daya yang terhubung ke Arduino UNO	23
Tabel 3. 4 <i>Pin</i> ESP32 CAM yang terhubung ke Arduino UNO	24
Tabel 3. 5 <i>Pin</i> Sensor GY-MAX4466 yang terhubung ke Arduino UNO.....	25
Tabel 3. 6 menampilkan <i>Pin Relay</i> yang terhubung ke Arduino UNO	26
Tabel 3. 7 <i>Pin Serial mp3 player</i> yang terhubung ke Arduino UNO.....	26
Tabel 3. 8 Spesifikasi Komponen Catu daya	27
Tabel 3. 9 Fungsi <i>library</i> pada sistem mikrokontroler Arduino	29
Tabel 4. 1 Hasil Pengujian Catu Daya Menggunakan Multimeter	42
Tabel 4. 2 Hasil Pengujian 1	47
Tabel 4. 3 Hasil Pengujian 2	47
Tabel 4. 4 Hasil Pengujian 3	48
Tabel 4. 5 Hasil Pengujian 4	49
Tabel 4. 6 Hasil Pengujian 5	50
Tabel 4. 7 Hasil Pengujian Batas dari Sensor GY-MAX4466.....	50
Tabel 4. 8 Hasil pengujian <i>relay</i>	51
Tabel 4. 9 Kondisi <i>Serial monitor</i> saat menerima data.....	52
Tabel 4. 10 Hasil pengujian <i>Serial mp3 player</i>	52
Tabel 4. 11 Data Hasil Pengujian ESP32 CAM.....	53
Tabel 4. 12 Hasil Pengujian Keseluruhan Sistem	55
Tabel 4. 13 Hasil Pengujian Terkoneksi ke <i>Access Point</i>	56
Tabel 4. 14 Hasil Pengujian Terkoneksi dengan <i>Access Point</i>	57
Tabel 4. 15 Hasil Pengujian <i>Key Performance Indicator</i>	59

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR LAMPIRAN

KEMATIK SISTEM KESELURUHAN	L1
KEMATIK CATU DAYA.....	L2
LAYOUT PCB SHIELD SISTEM KESELURUHAN.....	L3
LAYOUT PCB CATU DAYA	L4
AMPAK BELAKANG, DEPAN DAN SAMPING CASING	L5
KETCH PEMROGRAMAN ARDUINO	L6
KETCH PEMROGRAMAN ESP32.....	L7
DATASHEET ARDUINO UNO	L8
DATASHEET ESP32 CAM	L9
DATASHEET SENSOR GY-MAX4466	L10
DATASHEET <i>SERIAL MP3 PLAYER</i>	L11
DATASHEET <i>RELAY</i>	L12
DOKUMENTASI	L13



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



BAB I PENDAHULUAN

1.1 Latar Belakang Kegiatan

Internet merupakan teknologi yang sangat umum dan penting di era digital seperti zaman sekarang. Pemanfaatan *internet* secara maksimal sangat mempermudah manusia dalam melakukan kegiatan sehari-hari, bahkan membantu manusia dalam melakukan kegiatan-kegiatan kecil seperti menyalakan lampu, AC, dan berbagai macam alat elektronik yang sudah dibuat sedemikian rupa dapat terkoneksi dengan *internet*, sehingga segala macam kontrol atau pemantauan dapat dilakukan pada *smartphone*. Hal tersebut termasuk dalam penerapan *Internet of things* (IoT) yang merupakan sebuah jaringan yang digunakan untuk mewujudkan interkoneksi antara suatu obyek dengan layanan WEB atau Aplikasi pada perangkat *smartphone* sehingga penggunaan IoT akan membuat pekerjaan yang dilakukan manusia menjadi lebih mudah dan efisien.

Internet of things dapat diterapkan pada apa saja tergantung kebutuhan penggunaannya, salah satunya yaitu dalam pemantauan kendaraan motor yang kerap menyebabkan terjadinya kebisingan dan mengganggu warga sekitar karena memiliki knalpot yang berisik. Knalpot berisik sudah menjadi tren pada masyarakat khususnya anak muda yang suka memodifikasi knalpot kendaraan mereka menjadi tidak sesuai standar, sehingga knalpot tidak hanya mengeluarkan polusi emisi gas buang namun juga mengeluarkan polusi suara yang normalnya untuk motor dengan kapasitas mesin 80-175cc memiliki standar kebisingan kurang dari 90 dB sesuai dengan standar yang ditetapkan pada peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia NO P.56/MENLHK/SETJEN/KUM.1/10/2019.

Oleh karena itu, untuk membantu mempermudah *me-monitoring* dan memberikan peringatan kepada pengendara yang memiliki motor yang berisik, maka pada tugas akhir ini akan dibuat sebuah Rancang Bangun Sistem *Monitoring* Kebisingan Pada Kendaraan Bermotor yang dapat diterapkan di lingkungan perumahan untuk membantu satpam dalam menjaga ketertiban dan menciptakan lingkungan yang nyaman dan tenang. Alat ini akan menggunakan sensor suara

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

untuk mendeteksi kebisingan dan menggunakan kamera untuk mengambil gambar pengendara yang memiliki motor dengan knalpot yang berisik. Alat ini akan berkoneksi dengan *internet* melalui *database* Firebase dan dapat dikontrol melalui aplikasi Android untuk melihat hasil *monitoring* dan melihat foto yang ditangkap oleh kamera.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, maka permasalahan yang dibahas dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana cara merancang sistem *monitoring* kebisingan pada kendaraan bermotor di lingkungan perumahan berbasis IoT?
2. Bagaimana pengujian komponen dan pembacaan data kebisingan dari alat *monitoring* kebisingan di lingkungan perumahan berbasis IoT?
3. Bagaimana kualitas penangkapan sinyal Wi-Fi dari alat untuk konektivitas alat dengan *database* Firebase?

1.3 Tujuan

Adapun tujuan dari tugas akhir (TA) ini adalah :

1. Mampu merancang sistem *monitoring* kebisingan pada kendaraan di lingkungan perumahan berbasis IoT.
2. Mampu melakukan pengujian komponen dan pembacaan data kebisingan dari alat *monitoring* kebisingan di lingkungan perumahan berbasis IoT.
3. Mampu melakukan pengukuran kualitas penangkapan sinyal Wi-Fi dari alat untuk konektivitas alat dengan *database* Firebase.

1.4 Luaran

Adapun luaran dari Tugas Akhir ini adalah :

1. Alat untuk me-*monitoring* kebisingan pada kendaraan bermotor
2. Laporan Tugas Akhir
3. Jurnal terakreditasi



BAB V PENUTUP

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.1 Simpulan

Pada bab ini didapatkan simpulan dan saran dari Rancang Bangun Sistem *Monitoring* Kebisingan pada Kendaraan Bermotor di Lingkungan Perumahan berbasis IoT. Adapun simpulan dan saran yang dibuat yaitu :

Sistem monitoring kebisingan suara pada kendaraan bermotor di lingkungan perumahan berbasis IoT dirancang menggunakan Arduino, ESP32 CAM, sensor GY-MAXX4466, *relay*, lampu sorot, *Serial MP3 Player* dan Speaker. Alat telah dapat mendeteksi intensitas kebisingan maksimum sebesar 86 dB. Tegangan *output* dari catu daya yaitu 8.22V, hasil pengujian didapat bahwa sensor suara GY-MAX4466 dapat mendeteksi intensitas kebisingan dengan rentang suara saat pengujian yaitu 53-86 dB. Alat akan bekerja memberikan peringatan ketika sensor mendeteksi intensitas Kebisingan ≥ 80 dB serta memberikan output ke komponen pendukung lainnya seperti kamera, *relay* dan *serial mp3 player* sesuai sistem yang telah diprogram.

3. Pengukuran kualitas penangkapan sinyal Wi-Fi ESP32 dari jarak 1 sampai 10 meter didapat pada jarak 1-3 meter kualitas RSSI dalam kategori baik dengan nilai rata-rata -58.5 dBm, jarak 4-7 meter dengan rata-rata -65.7 dBm termasuk kategori cukup, dan jarak 7-10 meter kualitas RSSI dalam kategori buruk dengan nilai rata-rata -85 dBm.

5.2 Saran

Saran untuk hasil rancang bangun yang dilakukan yaitu diharapkan dapat dikembangkan menjadi lebih baik agar sistem kerja alat menjadi lebih sempurna dalam melakukan pembacaan kebisingan pada kendaraan bermotor.



DAFTAR PUSTAKA

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunsumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Afifah, E. N. (2021, Oktober). *Mengenal Perangkat Lunak Arduino IDE*. Dipetik 08 11, 2022, dari kmtech.id: <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide>
- Amarta, S., Putrada, A. G., & Suwastika, N. A. (2019, April 1). Asesmen Kebisingan di Open Library Telkom University Menggunakan Sistem. *e-Proceeding of Engineering : Vol.6, No.1 April 2019*. Dipetik Juli 14, 2022
- Asuti, N. F. (2020, September 28). *Mengenal Fungsi Speaker, Lengkap dengan Jenis-Jenisnya*. Dipetik Juli 14, 2022, dari merdeka.com: <https://www.merdeka.com/jabar/mengenal-fungsi-speaker-lengkap-dengan-jenis-jenisnya-klm.html>
- Dicoding. (2020, November 25). *Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya*. Dipetik 07 14, 2022, dari dicoding.com: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>
- Efriyendro, R., & Rahayu, Y. (2017). Analisa Perbandingan Kuat Sinyal 4G LTE Antara Operator Telkomsel dan XL AXIATA Berdasarkan Paramater Drive Test. Dipetik Juli 15, 2022
- Faudin, A. (2018, Maret 28). *Cara membuat mp3 player menggunakan Arduino*. Dipetik Juli 15, 2022, dari <https://www.nyebarilmu.com/>: <https://www.nyebarilmu.com/cara-membuat-mp3-player-menggunakan-arduino/>
- Kho, D. (2017, 07 22). *Pengertian Relay dan Fungsinya*. Dipetik 07 15, 2022, dari teknikelektronika.com: <https://teknikelektronika.com/pengertian-relay-fungsi-relay/>
- Nurshela, A. (2020). RANCANG BANGUN SISTEM PENDETEKSI KEBAKARAN MENGGUNAKAN ESP32-CAM BERBASIS IoT. Dipetik Juli 13, 2020
- Pratama, M. R. (2017). RANCANG BANGUN ALAT UKUR TINGKAT KEBISINGAN SUARA KNALPOT MENGGUNAKAN MIKROKONTROLER. Dipetik Juli 18, 2022
- Razor, A. (2021, Februari 21). *Software Arduino IDE: Cara Download, Instal, dan Fungsinya*. Dipetik Juli 13, 2022, dari aldyrazor.com: <https://www.aldyrazor.com/2020/05/software-arduino-ide.html>
- rhie.id. (2022, February 14). *Memahami Fitur Received Signal Strength Indication (RSSI) Pada Wireless dan Implementasinya*. Dipetik Juli 15, 2022, dari <https://www.rhie.id/2020/08/memahami-fitur-received-signal-strength.html>



DAFTAR RIWAYAT HIDUP

Syafiq Surya Rucita

Lahir di Wonosobo 3 April 2001. Lulus dari SDN Lubang Buaya 14 Petang pada tahun 2013. SMP Negeri 157 Jakarta pada tahun 2016. SMK Farmasi Ar-Raisiyah Husada pada tahun 2019. Gelar Diploma Tiga (D3) diperoleh pada tahun 2022 dari jurusan Teknik Elektro, Program Studi D3 Telekomunikasi, Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



LAMPIRAN

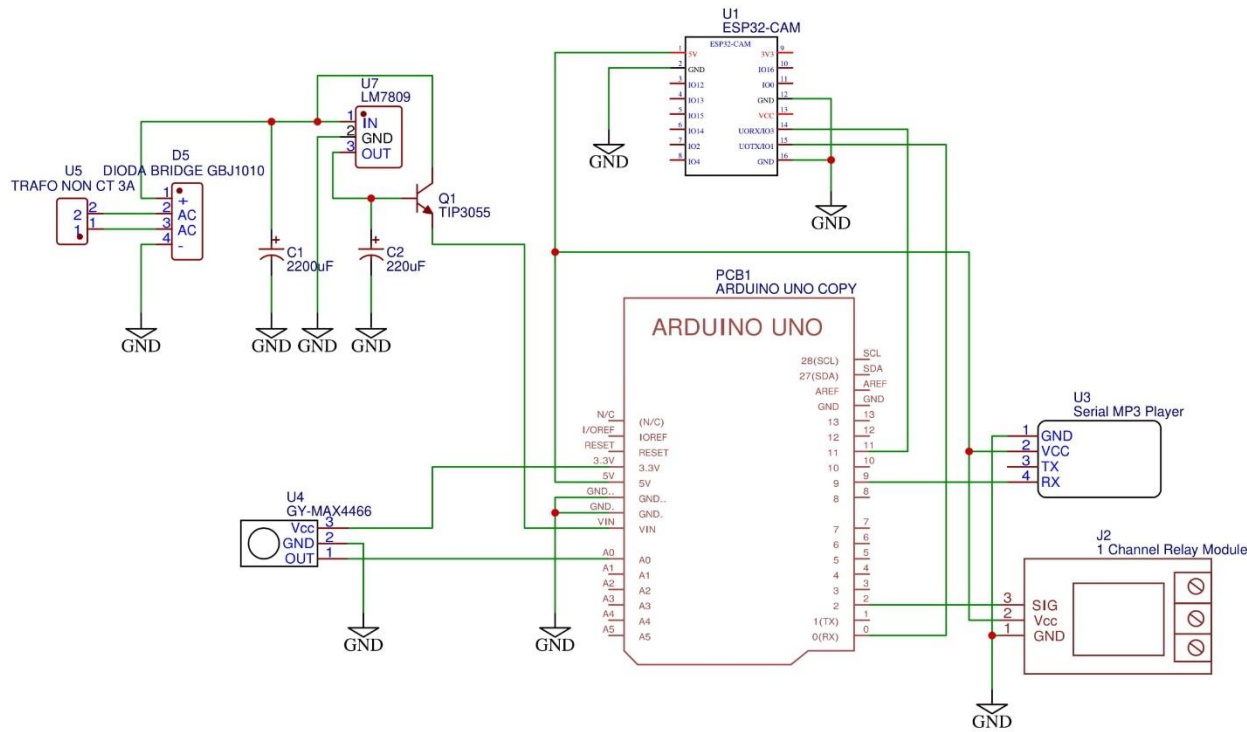


© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





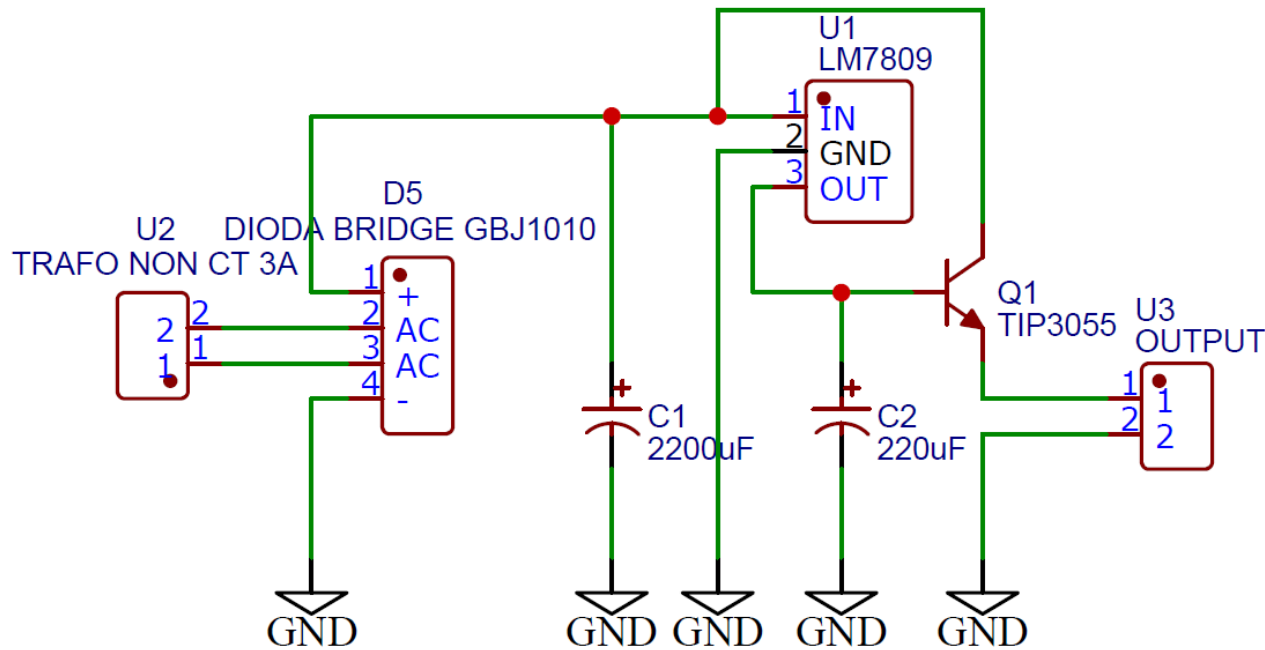
01

SKEMATIK SISTEM KESELURUHAN



PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	Syafiq Surya Rucita
Diperiksa	Rifqi Fuadi Hasani, S.T., M.T.
Tanggal Juli 2022



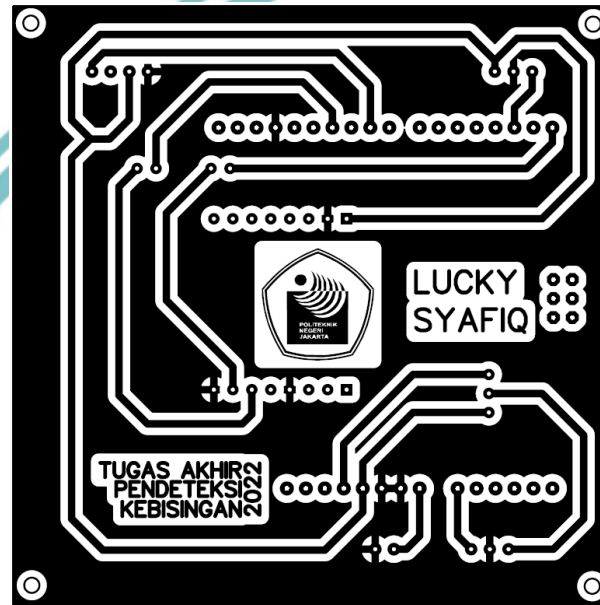
02

SKEMATIK CATU DAYA



PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	Syafiq Surya Rucita
Diperiksa	Rifqi Fuadi Hasani, S.T., M.T.
Tanggal Juli 2022



Politeknik Negeri Jakarta

ak Cipta :

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber. Dilarang menggunakan karya tulis ini untuk penerbitan, penulisan laporan, penelitian, atau tinjauan suatu materi tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA

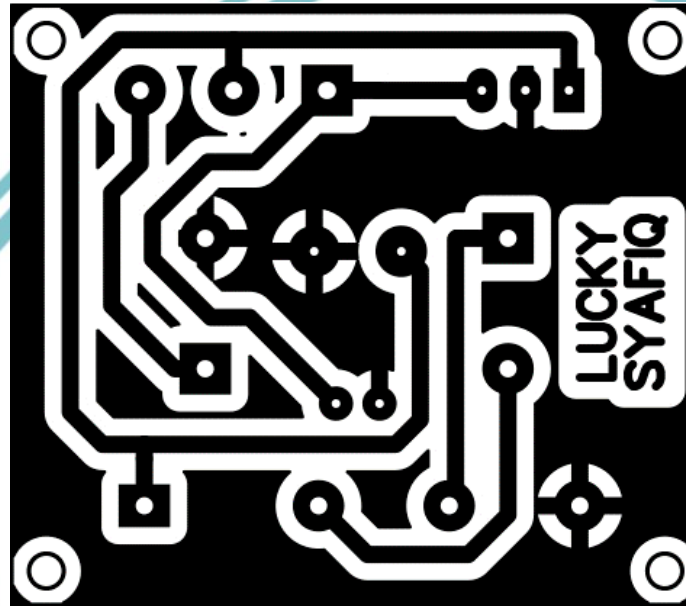
03

LAYOUT PCB SHIELD SISTEM KESELURUHAN 1:1



PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	Syafiq Surya Rucita
Diperiksa	Rifqi Fuadi Hasani, S.T., M.T.
Tanggal Juli 2022



ta milik Politeknik Negeri Jakarta

ak Cipta :

. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber.
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah
 b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 . Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA

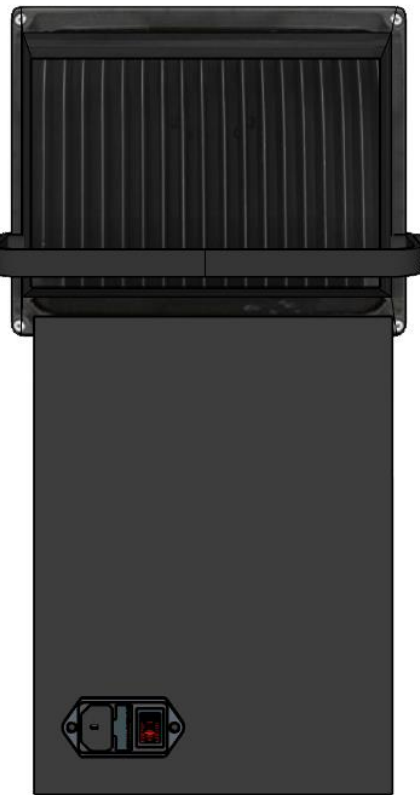
LAYOUT PCB CATU DAYA 2:1

04



**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK
ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar	Syafiq Surya Rucita
Diperiksa	Rifqi Fuadi Hasani, S.T., M.T.
Tanggal Juli 2022



05

TAMPAK BELAKANG, DEPAN DAN SAMPING CASING

PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA



Digambar	Syafiq Surya Rucita
Diperiksa	Rifqi Fuadi Hasani, S.T., M.T.
Tanggal Juli 2022



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
SoftwareSerial linkSerial(10, 11); // komunikasi antara arduino dan ESP32
#include "SerialMP3Player.h"
#define GD3300_TX 9
#define GD3300_RX 8
SerialMP3Player mp3(GD3300_RX, GD3300_TX); // komunikasi antara arduino dan module
Serial mp3 player
#define MAX4466_PIN A0 // Pin Arduino yang terhubung dengan sensor Suara MAX4466
const int sampleWindow = 50; // Sample window width in mS (50 mS = 20Hz)
unsigned int sample;
const int relayPin = 2; //Pin Arduino yang terhubung dengan Relay
unsigned long sendmsgMillis = 0;
int sendmsgInterval = 5000; //Interval pengiriman pesan ke Firebase
int lightFlag = 0; //Pengaturan kontrol lampu
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 115200 bits per second:
  Serial.begin(115200);
  // Initialize the "link" serial port
  // Use the lowest possible data rate to reduce error ratio
  linkSerial.begin(115200);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
  mp3.begin(9600); // start mp3-communication
  delay(500); // wait for init
  mp3.sendCommand(CMD_SEL_DEV, 0, 2); //select sd-card
  delay(500); // wait for init
}
// the loop routine runs over and over again forever:
void loop() {
  unsigned long startMillis = millis(); // Start of sample window
  unsigned int peakToPeak = 0; // peak-to-peak level
  unsigned int signalMax = 0;
  unsigned int signalMin = 1024;
  // collect data for 50 mS
  while (millis() - startMillis < sampleWindow)
  {
    sample = analogRead(MAX4466_PIN);
    if (sample < 1024) // toss out spurious readings
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

{
  if (sample > signalMax)
  {
    signalMax = sample; // save just the max levels
  }
  else if (sample < signalMin)
  {
    signalMin = sample; // save just the min levels
  }
}

// Konversi dari pembacaan sensor analog ke dB
peakToPeak = signalMax - signalMin; // max - min = peak-peak amplitude
double volts = ((peakToPeak * 3.3) / 1024) * 0.707; // konversi dari pembacaan
ensor analog ke tegangan

// konversi dari tegangan ke dB
double first = log10(volts / 0.00631) * 20;

double desibel = first + 94 - 44 - 15;
//https://forums.adafruit.com/viewtopic.php?f=8&t=100462&p=503661

// pengiriman data sensor ke ESP32 jika pembacaan dB > 80 dB
if (desibel > 80) {
  if (lightFlag == 1) digitalWrite(relayPin, LOW);
  // membuat dokumen JSON
  StaticJsonDocument<64> doc;
  doc["muffler"] = 1;
  doc["db"] = desibel;
  doc["volt"] = volts;
  // mengirim dokumen JSON ke ESP32 melalui pin komunikasi serial
  serializeJson(doc, linkSerial);
  serializeJson(doc, Serial);
  Serial.println();
  delay(10);
  Serial.println("Memutar Suara Peringatan");
  mp3.play(); // memutar suara dari module Serial mp3 player
  delay(6000); // menunggu 6 detik
  mp3.stop();
  Serial.println("Suara Peringatan Selesai");
  if (lightFlag == 1) digitalWrite(relayPin, HIGH);
  sendmsgMillis = millis();
}
}

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunsumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

/ pengiriman data sensor ke ESP32
if (millis() - sendmsgMillis >= sendmsgInterval) {
    // membuat dokumen JSON
    StaticJsonDocument<64> doc;

    doc["muffler"] = 0;
    doc["db"] = desibel;
    doc["volt"] = volts;
    // mengirim dokumen JSON ke ESP32 melalui pin komunikasi serial
    serializeJson(doc, linkSerial);
    serializeJson(doc, Serial);
    Serial.println();
    sendmsgMillis = millis();
}
// Pembacaan data yang masuk ke arduino
if (Serial.available()) {
    // read the incoming string:
    String incomingString = Serial.readStringUntil('\n');
    // menampilkan data yang masuk dari Firebase ke serial monitor
    Serial.print("Menerima Data : ");
    Serial.println(incomingString);
    // Kondisi Lampu saat menerima data dari aplikasi
    if (incomingString == "*L0#") lightFlag = 0;
    else if (incomingString == "*L1#") lightFlag = 1;
    else if (incomingString == "*L2#") lightFlag = 2;
}
if (lightFlag == 1) digitalWrite(relayPin, HIGH);
else if (lightFlag == 2) digitalWrite(relayPin, LOW);
else if (lightFlag == 0) digitalWrite(relayPin, HIGH);
}

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#include "Wi-Fi.h"
#include "esp_camera.h"
#include "Arduino.h"
#include "soc/soc.h" // Disable brownout problems
#include "soc/rtc_cntl_reg.h" // Disable brownout problems
#include "driver/rtc_io.h"
#include <SPIFFS.h>
#include <FS.h>
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include <addons/TokenHelper.h>
// Provide the RTDB payload printing info and other helper functions.
#include <addons/RTDBHelper.h>
#include <ArduinoJson.h>
#include <NTPClient.h>
#include <Wi-FiUdp.h>

// Define NTP Client to get time
Wi-FiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "id.pool.ntp.org");

//Week Days
String weekDays[7] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

//Month names
String months[12] = {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};

//Replace with your network credentials
const char* ssid = "tugasAkhirLuckySyafiq";
const char* password = "luckysyafiq";

// Insert Firebase project API Key
#define API_KEY "AIzaSyAEsMONG3dUd0cN5MP6OvbxXnBqmcMe8yk"

// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "syafiqsuryarucita@gmail.com"
#define USER_PASSWORD "taluckysyafiq"

// Insert Firebase storage bucket ID e.g bucket-name.appspot.com
#define STORAGE_BUCKET_ID "ta-syafiq-lucky.appspot.com"

// Insert Firebase RTDB URL e.g <databaseName>.<region>.Firebasedatabase.app
#define RTDB_URL "https://ta-syafiq-lucky-default-rtdb.asia-southeast1.firebaseio.com/"

// Photo File Name to save in SPIFFS
#define FILE_PHOTO "/data/photo.jpg"

// OV2640 camera module pins (CAMERA_MODEL_AI_THINKER)
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#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

boolean takeNewPhoto = false;
//Define Firebase Data objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig configF;
bool taskCompleted = true;
unsigned long sendDataPrevMillis = 0;
int count = 0;
int muffler = 0;
float desibel = 0;
float volts = 0;
unsigned long getFbMillis = 0;
int getFbInterval = 1000;
// Check if photo capture was successful
bool checkPhoto( fs::FS &fs ) {
  File f_pic = fs.open( FILE_PHOTO );
  unsigned int pic_sz = f_pic.size();
  return ( pic_sz > 100 );
}
// Capture Photo and Save it to SPIFFS
void capturePhotoSaveSpiiffs( void ) {
  camera_fb_t * fb = NULL; // pointer
  bool ok = 0; // Boolean indicating if the picture has been taken correctly
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

do {
    // Take a photo with the camera
    Serial.println("Taking a photo...");

    fb = esp_camera_fb_get();

    if (!fb) {
        Serial.println("Camera capture failed");
        return;
    }

    // Photo file name
    Serial.printf("Picture file name: %s\n", FILE_PHOTO);
    File file = SPIFFS.open(FILE_PHOTO, FILE_WRITE);

    // Insert the data in the photo file
    if (!file) {
        Serial.println("Failed to open file in writing mode");
    }
    else {
        file.write(fb->buf, fb->len); // payload (image), payload length
        Serial.print("The picture has been saved in ");
        Serial.print(FILE_PHOTO);
        Serial.print(" - Size: ");
        Serial.print(file.size());
        Serial.println(" bytes");
    }

    // Close the file
    file.close();
    esp_camera_fb_return(fb);

    // check if file has been correctly saved in SPIFFS
    ok = checkPhoto(SPIFFS);
} while ( !ok );

void initWi-Fi() {
    Wi-Fi.begin(ssid, password);

    while (Wi-Fi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to Wi-Fi...");
    }
}

void initSPIFFS() {
    if (!SPIFFS.begin(true)) {

```





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

Serial.println("An Error has occurred while mounting SPIFFS");
ESP.restart();
}
else {
  delay(500);
  Serial.println("SPIFFS mounted successfully");
}

void initCamera() {
  // OV2640 camera module
  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;
  if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }
}

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

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

void setup() {
    // Serial port for debugging purposes
    Serial.begin(115200);
    initWi-Fi();
    initSPIFFS();
    // Turn-off the 'brownout detector'
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    initCamera();
    //Firebase
    // Assign the api key
    configF.api_key = API_KEY;
    //Assign the user sign in credentials
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;
    //Assign the RTDB URL
    configF.database_url = RTDB_URL;
    //Assign the callback function for the long running token generation task
    configF.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
    Firebase.begin(&configF, &auth);
    Firebase.reconnectWi-Fi(true);
    // Initialize a NTPClient to get time
    timeClient.begin();
    timeClient.setTimeOffset(25200);
}

void loop() {
    // Check if the other Arduino is transmitting
    if (Serial.available())
    {
        // Allocate the JSON document
        // This one must be bigger than for the sender because it must store the
        strings
        StaticJsonDocument<128> doc;

```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// Read the JSON document from the "link" serial port
DeserializationError err = deserializeJson(doc, Serial);
if (err == DeserializationError::Ok)
{
  muffler = doc["muffler"];
  desibel = doc["db"];
  volts = doc["volt"];
  if (muffler == 1) {
    takeNewPhoto = true;
    taskCompleted = false;
  }
  if (Firebase.ready()) {
    timeClient.update();
    time_t epochTime = timeClient.getEpochTime();
    Firebase.RTDB.setIntAsync(&fbdo, "/muffler", muffler);
    Firebase.RTDB.setFloatAsync(&fbdo, "/db", desibel);
    Firebase.RTDB.setFloatAsync(&fbdo, "/volt", volts);
    Firebase.RTDB.setIntAsync(&fbdo, "/epoch", (unsigned long)epochTime);
    getfbMillis = millis();
  }
}
else
{
  // Print error to the "debug" serial port
  Serial.print("deserializeJson() returned ");
  Serial.println(err.c_str());
  // Flush all bytes in the "link" serial port buffer
  while (Serial.available() > 0)
    Serial.read();
}
if (takeNewPhoto) {
  capturePhotoSaveSpiffs();
  muffler = 0;
  takeNewPhoto = false;
}
delay(1);
if (Firebase.ready() && !taskCompleted) {
  taskCompleted = true;
}
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

Serial.print("Uploading picture... ");
timeClient.update();

time_t epochTime = timeClient.getEpochTime();
unsigned long ulEpoch = (unsigned long)epochTime;

String strEpoch = "/photos/" + (String)ulEpoch + ".jpg";

//MIME type should be valid to avoid the download problem.

//The file systems for flash and SD/SDMMC can be changed in FirebaseFS.h.

if (Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID /* Firebase Storage
bucket id */, FILE_PHOTO /* path to local file */, mem_storage_type_flash /*
memory storage type, mem_storage_type_flash and mem_storage_type_sd */, strEpoch
* path of remote file stored in the bucket */, "image/jpeg" /* mime type */) {

    String dlurl = fbdo.downloadURL().c_str();
    Serial.println();
    Serial.print("Download URL: ");
    Serial.println(dlurl);
    Serial.println();
    String buff1 = "/history/" + (String)ulEpoch + "/db";
    Firebase.RTDB.setFloatAsync(&fbdo, buff1, desibel);
    String buff2 = "/history/" + (String)ulEpoch + "/epoch";
    Firebase.RTDB.setIntAsync(&fbdo, buff2, ulEpoch);
    String buff3 = "/history/" + (String)ulEpoch + "/link";
    Firebase.RTDB.setStringAsync(&fbdo, buff3, dlurl);
    Firebase.RTDB.setStringAsync(&fbdo, "/image", dlurl);

    getfbMillis = millis();
}
else {
    Serial.println(fbdo.errorReason());
}
}

if (millis() - getfbMillis >= getfbInterval) {
    if (Firebase.RTDB.getInt(&fbdo, "/light")) {
        if (fbdo.dataTypeEnum() == fb_esp_rtdb_data_type_integer) {
            int getLight = fbdo.to<int>();
            String sendToArduino = "*L" + (String)getLight + "#" + '\n';
            Serial.print(sendToArduino);
        }
    } else {
        Serial.println(fbdo.errorReason());
    }
    getfbMillis = millis();
}

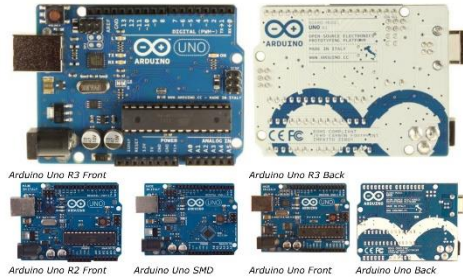
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Arduino Uno



Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the BU2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- **I/O pinout:** added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- ATmega 16U2 replace the BU2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)
 Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an ATmega8, 168, or 328. Current models use an ATmega328, but an ATmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to a corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between [Arduino pins and ATmega328 ports](#). The mapping for the ATmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on [Windows](#), a [.inf file](#) is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial](#) library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire](#) library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (refer to [reference](#), C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the BU2.
 - On Rev2 or later boards: there is a resistor that pulling the BU2/16U2 HWB line to ground, making it easier to put into DFU mode.
- You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

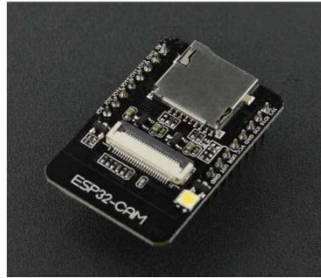
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



ESP32-CAM Development Board

SKU:DFR0602

INTRODUCTION

ESP32-CAM is a low-cost ESP32 based development board with onboard camera, small in size. It is an ideal solution for IoT application, prototypes constructions and DIY projects.

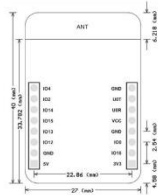
The board integrates WiFi, traditional Bluetooth and low power BLE, with 2 high-performance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz.

Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices

ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.



Schematic Diagram



Dimension Diagram

Notes:
1. Please be sure that the power supply for the module should be at least 5V 2A, otherwise maybe there would be water ripple appearing on the image.

2.ESP32 GPIO32 pin is used to control the power of the camera, so when the camera is in working, pull GPIO32 pin low.

3.Since IO pin is connected to camera XCLK, it should be left floating in using, and do not connect it to high/low level.

4.The product has been equipped with default firmware before leaving the factory, and we do not provide additional ones for you to download. So, please be cautious when you choose to burn other firmwares.

FEATURES

- Up to 160MHz clock speed, Summary computing power up to 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, Built-in Flash lamp.
- Support image WiFi upload
- Support TF card
- Supports multiple sleep modes.
- Embedded Lwip and FreeRTOS
- Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

SPECIFICATION

- SPI Flash: default 32Mbit
- RAM: built-in 520 KB+external 4MPSRAM
- Dimension: 27*40.5*4.5 (±0.2) mm/1.06*1.59*0.18"
- Bluetooth: Bluetooth 4.2 BR/EDR and BLE standards
- Wi-Fi: 802.11b/g/n/e/i
- Support interface: UART, SPI, I2C, PWM
- Support TF card: maximum support 4G
- IO port: 9
- Serial Port Baud-rate: 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 consumption: Turn off the flash: 180mA@5V
Turn on the flash and adjust the brightness to the maximum:
310mA@5V
Deep-sleep: the lowest power consumption can reach 6mA@5V
Modem-sleep: up to 20mA@5V
Light-sleep: 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
- Weight: 10g

SHIPPING LIST

- ESP32-CAM Development Board x1

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Description

Add an ear to your project with this well-designed electret microphone amplifier. This fully assembled and tested board comes with a 20-20KHz electret microphone soldered on. For the amplification, we use the Maxim MAX4466, an op-amp specifically designed for this delicate task! The amplifier has excellent power supply noise rejection, so this amplifier sounds really good and isn't nearly as noisy or scratchy as other mic amp breakouts we've tried!

This breakout is best used for projects such as voice changers, audio recording/sampling, and audio-reactive projects that use FFT. On the back, we include a small trimmer pot to adjust the gain. You can set the gain from 25x to 125x. That's down to be about 200mVpp (for normal speaking volume about 6" away) which is good for attaching to something that expects 'line level' input without clipping, or up to about 1Vpp, ideal for reading from a microcontroller ADC. The output is rail-to-rail so if the sounds gets loud, the output can go up to 5Vpp!

Using it is simple: connect GND to ground, VCC to 2.4-5VDC. For the best performance, use the "quietest" supply available (on an Arduino, this would be the 3.3V supply). The audio waveform will come out of the OUT pin. The output will have a DC bias of VCC/2 so when its perfectly quiet, the voltage will be a steady VCC/2 volts (it is DC coupled). If the audio equipment you're using requires AC coupled audio, place a 100uF capacitor between the output pin and the input of your device. If you're connecting to an audio amplifier that has differential inputs or includes decoupling capacitors, the 100uF cap is not required.

The output pin is not designed to drive speakers or anything but the smallest in-ear headphones - you'll need an audio amplifier (such as our 3.7W stereo amp) if you want to connect the amp directly to speakers. If you're connecting to a microcontroller pin, you don't need an amplifier or decoupling capacitor - connect the OUT pin directly to the microcontroller ADC pin.

For audio-reactive projects, we suggest using an FFT driver library (such as the one in this library) which can take the audio input and 'translate' it into frequencies. Also, check out this awesome Voice Changer project that uses this mic and an Adafruit Wave Shield.

Technical Details



MAX4466 Low-Noise
Microphone Amp Datasheet



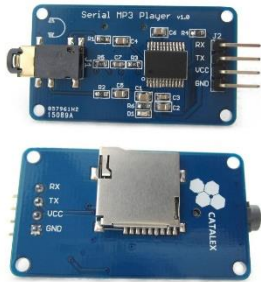
Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Email: catalex_inc@163.com

1 Description



The module is a kind of simple MP3 player device which is based on a high-quality MP3 audio chip—YXS100. It can support 8K Hz – 48K Hz sampling frequency MP3 and WAV file formats. There is a TF card socket on board, so you can plug the micro SD card that stores audio files. MCU can control the MP3 playback state by sending commands to the module via UART port, such as switch songs, change the volume and play mode; and so on. You can also debug the module via USB to UART module. It is compatible with Arduino / AVR / ARM / PIC.

Features:

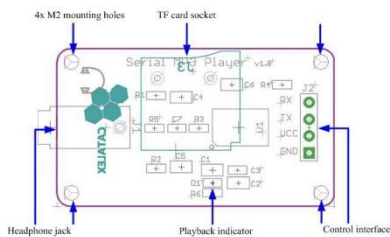
1. Support sampling frequency (MHz): 8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48
2. High quality
3. Support file format: MP3 / WAV
4. Support Micro SD card, Micro SDHC Card
5. 30 class adjustable volume
6. UART TTL serial control playback mode; baud rate is 9600bps
7. Power supply can be 3.2 – 5.2VDC
8. Control logic interface can be 3.3V / 5V TTL
9. Compatible with Arduino UNO / Leonardo / Mega2560 / DUE

2 / 10



Email: catalex_inc@163.com

3 Interface



Control interface: It is UART TTL interface. A total of four pins (GND, VCC, TX, RX). GND to ground, VCC is the power supply, TX is the TX pin of the MP3 chip, RX is the RX pin of the MP3 chip.

TF card socket: The micro sd card can be plugged in it.

Playback indicator: Green light. If it is ready to play or it is paused, it keeps lighting. If playing, it blinks.

Headphone jack: It can be connected with the headphone or external amplifier.

Mounting holes: 4 screw mounting holes whose diameter is 2.2mm, so that the module is easy to install, easy to combine with other modules.

4 / 10



Email: catalex_inc@163.com

2 Specification

Item	Min	Typical	Max	Unit
Power Supply(VCC)	3.2	5	5.2	VDC
Current (@VCC=5V)	/	/	200	mA
Logic interface	3.3V / 5V TTL			/
Supported Card Type	Micro SD card(<=2G); Micro SDHC card(<=32G)			/
File system format	Fat16 / Fat32			/
Uart baud rate	9600			bps
Dimensions	49X24X8.5			mm
Net Weight	5			g

3 / 10





Hak Cipta :

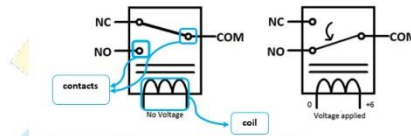
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RELAY MODULES

RELAY WORKING IDEA

Relays consist of three pins normally open pin , normally closed pin, common pin and coil. When coil powered on magntic field is generated the contacts connected to each other.

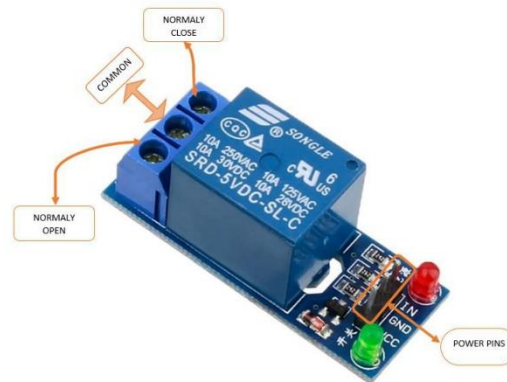


Relay modules 1-channel features

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power isolation between the coil and the switching pins so coils can be easily powered from Arduino by connecting VCC and GND bins from Arduino kit to the relay module kit after that we choose Arduino output pins depending on the number of relays needed in project designed and set these pins to output and make it out high (5 V) to control the coil that allow controlling of switching process.



NOTE : whatever was the relay channels number the pinconfiguration is the same for every channel except the power pins (VCC and GND) are for the board itself. The input signal (IN) pin for every relay.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengunumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Melakukan Pemotongan PCB



Melakukan Penyolderan PCB



Melakukan *Etching* PCB Catu Daya



Melakukan Pengetesan PSU



Melakukan Pengeboran PCB